KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science

Anonymous Credential Systems: From Theory Towards Practice

Jorn LAPON

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

July 2012

Anonymous Credential Systems: From Theory Towards Practice

Jorn LAPON

Jury:
Prof. dr. ir. Carlo Vandecasteele, chair
Prof. dr. ir. Bart De Decker, supervisor
Dr. Vincent Naessens, co-supervisor
Prof. dr. ir. Bart Preneel
Dr. ir. Lieven De Strycker
Dr. ir. Markulf Kohlweiss
(Microsoft Research - Cambridge)
Dr. Jaap Henk Hoepman
(Radboud University Nijmegen)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

July 2012

D/2012/7515/54
ISBN 978-94-6018-518-2

Acknowledgments

I would like to thank all the people that helped me during these four years to realize my Ph.D.

First of all, I would like to thank my co-supervisor, Dr. Vincent Naessens, who gave me the guidance and the opportunities that led me to the fulfillment of this Ph.D, for finding funds for my research and the many trips abroad, for the many interesting discussions we had and for the many times we moved around the campus.

Secondly, I would also like to thank Prof. Bart De Decker for accepting to be the supervisor of this thesis, for the detailed feedback, for the many discussions and for his wise advise.

Next, I would like to express my gratitude to Dr. Lieven De Strycker, Dr. Jaap Henk Hoepman, Dr. Markulf Kohlweiss and Prof. Bart Preneel for accepting to be members of the jury and for their observations which have improved the quality of this text, and Prof. Carlo Vandecasteele for chairing the jury.

This thesis is the result of the collaboration with many people, for which I'm thankful: Patrik Bichsel, Jan Camenisch, Kristiyan Haralambiev, Milica Milutinovic, Franz-Stefan Preiss, Tahir Sandikkaya, Stefaan Seys, Dieter Sommer, Bram Verdegem, Pieter Verhaeghe, Kristof Verslype, and many more. I would explicitly like to thank Markulf Kohlweiss for the many and long skype-discussions and the patience and help on all the questions I fired at him.

My thanks further go to the many colleagues I got to know in both KAHO Sint Lieven and KU Leuven. I am indebted to my colleagues in the MSEC research group, Faysal Boukayoua, Koen Decroix and Jan Vossaert, for backing me up with food and entertainment in times of need and helping me to find answers to the many questions I presented to them.

Last but definitely not least, I would like to thank my wife Nathalie Zoete for helping me to fulfill my dreams, for giving me support and advice when things are not going like I want them to go, and together with my daughter Amélie, for being the most important in my life.

Jorn Lapon,
Ghent, June 2012

Abstract

In today's society, privacy is subject to a lively debate. The growing connectivity and new technologies make linking and profiling easier and more accurate. Hence, the protection of privacy becomes a necessity unless we believe that our *privacy is lost*. Therefore, it is important to work on solutions that can improve our privacy.

In this dissertation, we start with a critical assessment of electronic identity technology currently deployed, in particular the Belgian electronic identity card. The results clearly show that the protection of privacy is inadequate, especially when the card is used across both the public and private domains.

Anonymous credential systems promise an alternative, supporting privacy and strong authentication. Unfortunately, these credential systems are still mainly a research topic, and have not yet found their way towards the general public. A major drawback of anonymous credential systems is that they are considerably more complex than the technologies currently used. In this dissertation, we provide a solution based on mobile devices as a platform, possibly extended with a secure element, for hosting the anonymous credential system. Another issue with anonymous credential systems is the lack of an efficient and practical revocation strategy. Multiple schemes have been presented, but their complexity is much more involved than revocation schemes used in traditional PKI-based systems. We present a pragmatic assessment of revocation schemes for anonymous credential schemes, of which some have been implemented as a basis for an in depth evaluation.

Anonymous credential systems are complex systems supporting privacy-friendly transactions. However, to make sense, anonymous credentials should be accompanied by an infrastructure that supports privacy-preserving applications and new protocols will need to be designed. We analyze how simulation-based security models can be applied for building such larger complex systems. We provide a number of building blocks in order to help and guide protocol designers. As a validation of the framework and of our building-blocks, we model the concept of Oblivious Trusted Third Parties and present an actual implementation.

Beknopte samenvatting

Vandaag is privacy een veelbesproken onderwerp. Een alsmaar toenemende connectiviteit en het gebruik van nieuwe technologieën zorgen ervoor dat persoonlijke informatie makkelijk gelinkt kan worden en profielen eenvoudiger en nauwkeuriger kunnen opgesteld worden. Het belang van de bescherming van de privacy is duidelijk. Hoewel sommigen reeds aannemen dat privacy een verloren zaak is, blijft het sowieso belangrijk om te werken aan oplossingen die onze privacy kunnen verbeteren.

In dit proefschrift, beginnen we met een kritische beoordeling van de elektronische identiteitskaarten die momenteel reeds in gebruik zijn genomen, met in het bijzonder de Belgische elektronische identiteitskaart. De resultaten laten duidelijk zien dat de bescherming van de privacy onvoldoende is, vooral wanneer de kaart wordt gebruikt in zowel het publieke als private domeinen.

Anonieme credential systemen beloven een alternatief. Ze combineren sterke authenticatie met privacyvriendelijke eigenschappen. Helaas treffen we deze systemen nog vooral aan in onderzoekslaboratoria, en hebben deze nog niet hun weg gevonden naar het grote publiek. Een belangrijk nadeel van anonieme credential systemen is dat ze aanzienlijk complexer zijn dan de technologieën die momenteel worden gebruikt. In dit proefschrift demonstreren we een oplossing die gebruik maakt van mobiele toestellen, eventueel in combinatie met een veilige component, voor het beheren en gebruiken van de anonieme credentials. Een ander probleem gerelateerd aan anonieme credential systemen is het ontbreken van een efficiënte en haalbare revocatiestrategie. In de literatuur werden reeds verschillende oplossingen voorgesteld, maar ze zijn vele malen complexer dan de revocatie strategieën voor systemen gebaseerd op traditionele PKI-technologieën. We geven een pragmatische beoordeling van de revocatiestrategiën voor anonieme credentials, waarvan een aantal zijn geïmplementeerd als basis voor een diepgaande evaluatie.

Anonieme credential systemen zijn complexe systemen ter ondersteuning van privacyvriendelijke transacties. Echter, het gebruik van anonieme credentials volstaat niet. Het is belangrijk om een nieuwe infrastructuur te voorzien met nieuwe protocollen en

systemen die toelaten om privacyvriendelijke toepassingen te ontwikkelen. We gaan na hoe een raamwerk voor simulatiegebaseerde modellen kan worden toegepast voor het ontwerpen van dergelijke grote complexe systemen die veilig en privacyvriendelijk zijn. Wij voorzien een aantal bouwblokken om het modelleren te vereenvoudigen. Om het raamwerk en onze bouwblokken te valideren modelleren we het concept 'Oblivious Trusted Third Parties' en demonstreren we hoe dit concept kan gerealiseerd worden.

Abbreviations

ABC	Attribute Based Credentials
Acc	Accumulator
API	Application Programming Interface
CARL	Card-based Access control Requirements Language
CCA	(Adaptive) Chosen-Ciphertext Attack
CPA	Chosen-Plaintext Attack
CRL	Certificate Revocation List
DAA	Direct Anonymous Attestation
EEPROM	Electrically Erasable Programmable Read-Only Memory
eID	Electronic Identity
GUC	Generalized Universal Composability
GWT	Google Web Toolkit
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IITM	Inexhaustible Interactive Turing Machine
IP	Internet Protocol
ITM	Interactive Turing Machine
LL	Limited Lifetime
MAC	Media Access Control
MULTOS	Multi-application smart card Operating System
NFC	Near Field Communication
NP	nondertministic polynomial time
NRN	National Registration Number
OCSP	Online Certificate Status Protocol
OTP	Oblivious Trusted third Party
PBC	Pairing Based Cryptography
PET	Privacy Enhancing Technology
PIN	Personal Identification Number

PK	proof of knowledge
PKI	Public Key Infrastructure
PPT	probabilistic polynomial time
QR	Quick Response code
RAM	Random-Access Memory
RBAC	Role Based Access Control
REST	REpresentational State Transfer
RFC	Request For Comments
RFID	Radio-Frequency IDentification
RL	Revocation List
RSA	Ron Rivest, Adi Shamir and Leonard Adleman
SAML	Security Assertion Markup Language
SE	Secure Element
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SPK	Signature proof of knowledge
SSL	Secure Sockets Layer
SSO	Single-Sign-On
TCG	Trusted Computing Group
TLS	Transport Layer Security
TPM	Trusted Platform Module
TTP	Trusted Third Party
UC	Universal Composability
URI	Uniform Resource Identifier
VAT	Value Added Tax
VE	Verifiable Encryption
VLR	Verifier Local Revocation
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
ZK	zero-knowledge

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's service-oriented world, users regularly come across situations where they need to authenticate to a clerk, a terminal, a door, or an on-line service to gain access to a desired resource. Often such authentications are performed by identifying the user using physical keys, cryptographic credentials (e.g., X.509 certificates), or an authentication tuple consisting of a username and password. This leads to an excessive release of personal information, often not even required for the underlying business processes, but released due to the authentication technology being used. Even worse, performing multiple identifying interactions with the same or different service providers makes all those interactions linkable to those service providers.

This research started with a survey of the Belgian electronic identity card (eID), which uses X.509 certificates for authentication. Verhaeghe et al. [VLN+08, VLDD+09] identified exactly such privacy issues in the Belgian eID. We will shortly review existing eID schemes, in particular the Belgian eID and present some solutions to reduce these problems. Nevertheless, the main conclusion is that traditional certificate based systems, which are commonly used for eIDs, do not sufficiently protect the privacy of citizens, especially, when used in the private sector. One solution is to develop eID architectures, based on privacy enhancing technologies.

During the last decades, many privacy enhancing technologies have been proposed and developed. Examples are *privacy enhancing service architectures* [ABD+03, BJ05], *anonymous communication* [GRS96, RR98, DMS04, BFK01, KZG07], *anonymous storage* [DFM01, DDD05] and *anonymizing user location data* [CZBP06, KFF+07, HH10]. They all aim at offering a higher level of privacy (or anonymity) in the digital world.

Another such technology is *Anonymous Credential systems*, which allows for

anonymous yet accountable transactions. Only the attributes – or properties of attributes – that are required to access a service, are proved to a service provider, while the client is preferably not being identified through those. Then, business processes operate on those properties instead of identities, realizing *data-minimization*. For instance, users can prove to belong to a certain age category to get discounts on public transport tickets. Similarly, they only need to prove to live in the city in order to get access to the local waste recycling center.

One drawback of anonymous credential systems is that they are considerably more complex than traditional certificate based systems, making their correct integration by implementers even more challenging. Moreover, solving requirements such as credential revocation are much more involved than in traditional authentication systems and require special attention.

With their increasing computational power, mobile devices have been emerging as potential target platforms for the wide-spread deployment of anonymous credential systems. A mobile device, optionally extended with a secure element, can store the user's credentials and act as a host to perform credential-based authentication protocols. Mobile devices are particularly suited as a host platform for credential protocols because many users already carry a mobile device with them, the possibility of realizing intuitive graphical user interfaces, and the required short-range channels to connect to other devices being or becoming available. Currently, ongoing developments [GRB03, GM07, SKK08, DW09] in the area of trusted execution environments go into the direction of strengthening future mobile devices to make them better suited for hosting the credentials of a user.

To support an acceptable level of privacy, anonymous credentials particularly make sense in large scale environments such as nation-wide electronic identities. Several countries already issued electronic identities, for both public and private transactions. Looking at the intricacies of those systems gives us an advantage in defining the requirements for an electronic identity based on anonymous credentials.

In this thesis, we try to solve the following question:

> *Is it feasible to use anonymous credentials as a nation-wide electronic identity, offering both enhanced privacy and security properties?*

1.1 Privacy and Accountability

Information privacy. Privacy is a hot topic, debated in both technical and non-technical literature. Giving a simple definition is not straightforward. In fact, the concept of privacy changes over time [Lan01]. While in the early days, privacy was

mostly associated with ensuring that governments cannot spy on citizens and privacy was often considered as *the right to be let alone* [WB90], nowadays, it is more believed to be *the right to select what personal information about an individual is known to what people* [Wes70].

While some aspects of privacy (such as territorial, communication and bodily privacy [Lan01]) have been regulated in constitutional rights, new and rapidly evolving technologies make privacy even more challenging. With the advent of the Internet and online services, *information privacy* has become an important aspect of privacy. Information privacy deals with the protection of personal information: it addresses when, how and to what extent an individual shares personal information. In other words, it puts restrictions on the collection of personal information, its use, its retention and its disclosure.

Defining privacy is difficult, protecting it is worse.

One of the reasons why information privacy is not easy to protect is that people disclose information to services without being aware of the consequences. Although surveys show people are concerned about their privacy, research and experiments have evidenced that individuals are willing to disclose information for small rewards [Acq04a]. Moreover, people are getting used to clicking *Accept* upon so-called *privacy policies*. Privacy policies are a sort of "agreements" on what can and cannot be done with the personal information disclosed to the service provider. However, they merely protect the company against lawsuits than to adhere to fair datahandling practices [Pol07]. Moreover, these privacy policies are often written in *legalese* language, protecting the service provider's concerns, while hard to read and understand by laypeople. Hence, even if they read the policies, they still do not know what they agree upon.

A second important reason is that, once information is disclosed, it is hard to control what happens with it. Although there may be policies or legal restrictions on, for instance, data-retention or sharing, it remains a matter of trust in the service provider to actually apply them.

People are the product. As e-commerce activities are growing worldwide, companies collect more and more personal information of their customers. Companies are trusted to carefully handle the information gathered and only use this information to run their *business* properly. The problems start when there is a difference in the interpretation of '*business*' from the viewpoint of the company and that of the users. Moreover, as many services pretend to be *free*, the business model behind these services blurs away in the eye of the user. Hence, they do not know what their information is used for. Often companies offering such free services make selling personal information their business. They turn customers into their product, and sell

the information gathered, for instance, to advertising companies. This shows that we cannot wait for the companies to become privacy-friendly. As long as privacy is not restricted by regulations, privacy will be handled as a trade-off between costs and benefits. Hence, if privacy cannot be monetized, companies will not provide it.

Towards a better privacy. Some consider privacy, and in its extreme form *anonymity*, is a way to hide clandestine and illegal actions: *'People that care about privacy have something to hide'*. Often such actions may be easily prevented even in an anonymous setting, however, sometimes appropriate countermeasures should be put in place. In that case, there should be a means to take the individual *accountable* for his actions. Usually, service providers therefore gather a plethora of personal information.

The current solutions do not offer sufficient protection of privacy: privacy policies are hard to read and may change over time, legislation is generic, often country-specific and slow to act in a quick evolving environment, and it is hard to verify that rules are respected (e.g., using audits). Therefore, Privacy Enhancing Technologies (PETs) may help in counterfeiting the deterioration of privacy. Anonymous credential systems is such a technology, offering a solution in which both privacy is preserved (e.g., anonymity and selective disclosure), and appropriate actions may be taken in case of abuse. Hence, service providers should no longer collect excessive personal information to obtain accountability. In this dissertation, we analyze how anonymous credentials can be used in real-world applications from a technical point of view and show that anonymous credentials may indeed be useful to protect the user's privacy.

Unfortunately, supporting anonymous credential systems in real-world applications requires the collaboration of the service providers, which may have other interests. Hence, it remains to find a way to convince service providers to implement such technologies.

1.2 Approach & Scope

This research started with a study of the Belgian electronic identity card. Belgium introduced electronic identity cards as one of the first countries in Europe; in 2003. The card offers strong authentication to a variety of applications. However, the card also has a number of limitations and weaknesses, of which the lack of privacy protection is one of the major ones. We contrive solutions which may reduce the impact of these issues.

*As a **first main contribution**, we summarize the main properties of the current Belgian eID and design a number of advanced applications that take advantage of the benefits of the card, while mitigating the privacy & security issues involved.*

However, the application domains discussed in this contribution have been chosen specifically with that in mind. To increase the user's privacy in electronic transactions, anonymous credentials may provide better features. They allow for anonymous, yet accountable transactions. Moreover, selective disclosure allows the user to limit the amount of personal information being revealed to service providers. Since anonymous credential systems are still mainly a research topic, we evaluate their applicability in real world environments, particularly, in the large scale setting of electronic identities. Despite their beneficial security and privacy properties, they include a number of drawbacks, mostly related to efficiency. The computational requirements are well-known problems of anonymous credential systems. For instance, current Java Card technology available for the mass-market is not yet powerful enough to fully support anonymous credential systems (e.g., including range proofs). The increasing computational resources of mobile devices may offer a solution to make anonymous credential systems effective. Moreover, embedding them in mobile devices allows for ubiquitous secure electronic transactions.

*As a **second main contribution**, we present a number of building blocks for advancing the development of mobile applications: we demonstrate how they may be used to support secure and privacy friendly applications on mobile devices and we evaluate the usability of anonymous credentials in this setting.*

Another important issue in existing anonymous credential systems, is the lack of support for a proper revocation strategy. Many revocation schemes have been presented in the literature, with different approaches, usability and efficiency.

*As a **third main contribution**, we present a pragmatic assessment of revocation schemes for anonymous credential schemes, of which a number of schemes have been implemented as a basis for an in depth evaluation.*

Anonymous credentials allow to implement electronic transactions that are unlinkable and selectively disclose the minimal amount of information about the user. At the same time these transactions have to be accountable. When using anonymous credentials, transactions are automatically accountable in the sense that the verifier is ensured that what is being proved during the credential show, is indeed vouched for by the issuer. However, many real-life applications have to consider exceptional cases in which additional information is required in case of a malicious transaction. Moreover, sometimes service providers may require more guarantees in order to properly run their businesses. Therefore, new schemes will have to be developed that provide sufficient guarantees towards service providers while preserving the privacy of their customers. However, building such complex systems is not trivial.

*As a **fourth main contribution**, we evaluate a framework for proving the security of complex applications. We present a number of building blocks and a common approach that features the design of new systems and evaluate our approach, the building blocks and the framework in an application named Oblivious Trusted Third Parties. In this application we provide higher trust and efficiency for services by supporting trusted third parties that are kept as oblivious as possible to the task they perform. We model these privacy preserving services and present protocols that allows its realization.*

1.3 Traditional Electronic Identities

Currently, most electronic identities that provide strong authentication, such as the Belgian eID, are based on X.509 certificate technology. Inherent to this technology is that it is linkable, and all information included in the certificate is disclosed to the verifier. Particularly, if such a certificate is assigned to a single citizen, all electronic transactions of that individual may be linked, allowing for composing of extensive profiles.

Furthermore, most electronic identity solutions employ smart cards. Even if there is no government issued citizen card, smart cards (e.g., SIM or banking card) are often used as a bearer for electronic identities (e.g., FineID [PS01] and Austrian Bürgerkarte [LHP02]). These tamper resistant cards protect the secret keys from being revealed. Nevertheless, smart cards also entail some issues, of which the most important is the required trust in the host environment. For instance, usually PIN codes must be entered in the host application, requiring trust in the host for not storing or leaking the PIN. In addition, the user has virtually no control which messages are sent to the card. Hence, the user has no guarantees on which transactions are in fact performed (e.g., which documents are being signed, what information is being revealed or, to which site is the card authenticating).
These are important properties that need special attention when designing electronic identities, especially, since they encourage the development of new online services with potentially high security risks.

The German eID [PWVT11], also based on X.509 certificate technology, already tries to take care of most of these issues. Linkability is prevented by having a batch of users share the same certificate and private key. Nevertheless, privacy may be limited due to a possibly small batch size (i.e., documents issued during a 3 months period get the same key pair [Mar11]). Moreover, access to information on the card requires approval by the certification authority. Unfortunately, it is economically not attractive for service providers to support the German eID in their applications. An initial investment is required for technical support and service approval, and there are also

recurring costs, such as fees for the certification authority. Moreover, the security of the scheme is entirely based on the tamper resistance of the card. If a single card is broken, and the secret key is obtained, an attacker may impersonate citizens without even being detectable. This is a substantial security threat, which may be resolved with more advanced technologies such as anonymous credentials.

1.4 Anonymous Credentials

In an increasingly information driven society, protecting the privacy becomes essential. However, privacy or anonymity is sometimes abused to perform clandestine and illegal actions. Anonymous credentials promise a solution. They protect the user's privacy, while ensuring accountability towards the service provider. Anonymous credential systems [Cha85, CL01, CL03, Bra00], which are closely related to group signatures and identity escrow schemes, allow for anonymous yet accountable transactions between users and organizations. They are attribute-based, user-centric mechanisms in which users can *anonymously* prove assertions about themselves and their relations with others. Moreover, selective disclosure allows the user to reveal only a subset of possible properties of the attributes embedded in the credential. For instance, a credential, with the user's date of birth as an attribute, can be used to prove that the owner is over 18, without disclosing the exact date of birth or other attributes.

In essence, with an *anonymous credential*, the holder may authenticate without disclosing any information but the fact that she possesses a valid credential.

In the literature, there are two major strategies for solving this problem, based on how linking of user interactions with the issuer on the one hand, and user interactions with relying parties on the other hand, is prevented. Hence we define two types:

- TYPE 1: The first technology [Bra00], is based on *blind signatures* [Cha83] in order to break the link between the issuer and the user's credential. In short, the issuer signs the user's credential, without knowing the resulting signature value. As a result, when the credential is used, even if the issuer and relying party share information, it cannot be linked to the issuance phase. In order to make multiple transactions unlinkable, a batch of credentials is issued and for each anonymous transaction, a new credential is used.
- TYPE 2: The second technology [CL01, CL03], takes a totally different approach. Here, the link between the credential issuance and its use, is prevented by using so-called *zero-knowledge proofs*. During authentication, the user proves in zero-knowledge to the relying party, that she has a genuine credential (i.e., certified by the trusted issuer). Here, zero-knowledge means that, in the general case, nothing is disclosed but the fact that the credential

is genuine, and the prover is the holder of the credential (i.e., she knows the corresponding private key). These proofs are generally more involved than, for instance, showing a credential of the first technology. On the other hand, since multiple shows are unlinkable, only a single credential is required.

In addition, both systems support selective disclosure of attributes and properties thereof. Obviously, the more information is disclosed, the more the level of anonymity decreases.

For each of these strategies, an implementation is available. `U-Prove` [19], implemented by Microsoft, adheres to the first technology, while the `Identity Mixer` [11], implemented by IBM, uses the second technology. In this thesis, we will focus on anonymous credential systems of the second type, unless stated differently.

To fully benefit from the privacy features provided by anonymous credentials, they should be used in combination with anonymous communication [GRS96, RR98, DMS04, BFK01, KZG07] in order to prevent linking or identification through for instance IP address, MAC address, cookies, or browser identification. Nevertheless, even without anonymous communication, anonymous credentials are superior and more privacy friendly than traditional authentication technologies.

1.5 Relation to Research Projects

This dissertation is strongly related to larger projects. There are contributions in different projects, and results of those projects are used as input for this research. We briefly present the most important relations:

STORK. The STORK 1.0 project aimed at establishing a European eID Interoperability Platform that allows citizens to establish new e-relations across borders, just by presenting their national eID.[1]
Although we did not discuss in detail the concept of BeID proxy certificates, discussed in Chapter 4, is easily extended to several other national eIDs that use X.509 certificates.

adapID. The adapID project[2] aimed at the development of a framework for secure and privacy-preserving applications and investigated technologies for future enhanced generations of the eID.

[1] Secure Identities Across Borders Linked, Project co-funded by the European Commission (INFSO-ICT-PSP-224993) `https://www.eid-stork.eu`.

[2] Advanced applications for electronic IDentity cards in Flanders, funded by the Flemish government (IWT-SBO 040072) `http://www.cosic.esat.kuleuven.be/adapid/`.

The BeID proxy certificates [LVV+09] and petition application [LVV+08] presented in Chapter 4 are contributions to the adapID project. In addition, in Chapter 6, we have developed a stripped down version of the framework [VVL+10] that was designed during the adapID project.

PRIME/PrimeLife. The PRIME project[3] aimed at developing a working prototype of a privacy-enhancing Identity Management System. PrimeLife[4] is a follow-up project that builds upon and expands the sound foundation of the PRIME project.
The analysis of accumulator-based revocation technologies [LKDDN10] has contributed to the PrimeLife project. We employ CARL policies [CMN+10], which contributed to the PrimeLife project. Finally, the Oblivious Trusted Third Party application model is also part of the main research results of the PrimeLife project [CDK+11]. In fact, most of this thesis provides contributions in the research on privacy-enhancing identity technologies.

ABC4Trust ABC4Trust[5] aims at defining a common, unified architecture for attribute based credential (ABC) systems and delivering an open reference implementation demonstrating the practical use of these systems.
The mobile authentication application presented in Chapter 7 demonstrates the practical use of `Identity Mixer` anonymous credentials, which is an ABC system and will be published as a contribution to the ABC4Trust project.

1.6 Outline and Summary of the Contribution

1.6.1 Summary of the Contribution

This dissertation has four main parts corresponding to the four main contributions presented above. Most of this work was presented and published in the proceedings of peer-reviewed international conferences [LVNV08, LVV+08, LVV+09, LKDDN10, LKDDN11, CHK+11a], in *Security and Communication Networks* [LNV+10], and as part of the book *Privacy and Identity Management for Life* [CDK+11].

In Part I, we evaluate the properties of the current Belgian eID as an example of traditional strong authentication technologies. The Belgian eID also comes with some limitations and weaknesses. Therefore, we present some example applications in which we take advantage of the benefits of the eID, while mitigating the drawbacks. Since in many applications mitigation is not possible, we have to reside to other

[3] Privacy and Identity Management for Europe, Project co-funded by the European Commission (IST-507591) `https://www.prime-project.eu`

[4] PrimeLife: Bringing sustainable privacy and identity management to future networks and services, Project co-funded by the European Commission (216483) `http://www.primelife.eu/`

[5] Attribute-based Credentials for Trust, Project co-funded by the European Commission (ICT-2009-5) `https://abc4trust.eu/`

solutions. Therefore, we introduce anonymous credentials as a strong anonymous authentication mechanism, offering both security and privacy.

However, with respect to traditional authentication technologies, these credentials are much more complex and may, in fact, require a whole new approach. We analyze, in Part II and III, how these anonymous credential systems perform in real world applications requiring user authentication, what is still lacking, and what is required to make them effective.

Finally, anonymous credentials offer new possibilities for privacy-enhancing applications. However, developing such applications with both privacy and security in mind, is not straightforward. Simulation-based strategies may offer a way out. These strategies allow for proving the security of complex systems, based on an idealized (and possibly simpler) version of the system. In Part IV, we evaluate such a general simulation-based framework, by validating it through the realization of an application called oblivious third parties.

1.6.2 Outline

The rest of this dissertation is structured as follows:

Chapter 2 : Preliminaries presents the notation, and introduces a number of building blocks used throughout this thesis.

Part I : Traditional Electronic Identities

Chapter 3 : Belgian Electronic Identity Technology evaluates the advantages, but also some of the drawbacks of the current Belgian eID.

Chapter 4: eID Applications presents three application domains in which we try to benefit from the strengths and try to mitigate some of the limitations and weaknesses involved in the Belgian eID.

Chapter 5: eID Requirements Study presents the requirements of an improved electronic identity, based on the findings of the current Belgian eID, which will be used as a base for comparison when we want to use anonymous credentials for electronic identities.

Part II : Mobile Anonymous Authentication

Chapter 6: Building Secure and Privacy-friendly Mobile Applications provides a number of building blocks, to advance the development of secure and privacy friendly mobile applications.

Chapter 7: Mobile Authentication towards a Terminal demonstrates the use of those building blocks in an example application, in which a mobile device is used to

authenticate towards a terminal. It also shows the feasibility of using anonymous credentials in mobile environments.

Chapter 8: Evaluation evaluates the solution based on anonymous credentials with respect to the requirements derived in Chapter 5.

Part III : Revocation Strategies

Chapter 9: State of the Art presents an overview and classification of revocation strategies for anonymous credential systems available in the literature.

Chapter 10: Analysis of Revocation Strategies for `Identity Mixer` Credentials extends the `Identity Mixer` library with multiple revocation schemes, each addressing a specific type of revocation. Based on these implementations, we provide a pragmatic evaluation of the different schemes.

Chapter 11: Analysis of Accumulator-based Revocation Mechanisms reviews and evaluates three accumulator-based revocation schemes available in the literature. An actual implementation allows a practical evaluation of those schemes.

Chapter 12: Evaluation evaluates the research presented on revocation strategies and provides a number of guidelines towards both researchers and application developers.

Part IV : Secure Application Modeling

Chapter 13: Modeling Secure Applications provides a brief introduction to the simulation-based security framework, in particular the framework by Küsters [Küs06]. Based on this framework, we provide a number of building blocks that may be used as components for building larger applications.

Chapter 14: Oblivious Trusted Third Parties provides an ideal model of the Oblivious Trusted Third Parties concept and demonstrates how the functionalities presented in the previous chapter, can be used to build an actual instantiation of our model.

Chapter 15: Evaluation evaluates the OTP model, the building blocks and the simulation-based framework in general.

Chapter 16: General Conclusions.

Chapter 2

Preliminaries

In this chapter, we first present the notation used throughout this thesis. The remainder mainly aims at making this thesis self-contained and may be used as a reference for further reading. We summarize some cryptographic building blocks, of which most are employed in the `Identity Mixer` library, namely the CL signature scheme, commitments and some example proofs of knowledge. For more details, we refer to the corresponding publications.

2.1 Notation

For clarity, as depicted in Fig. 2.1, we define a *credential* as a signed set of attributes of which some are only known by the user, and others are known by both the user and the issuer of the credential. In order to authenticate, a token, asserting certain statements, is sent to the relying party. In the case of X.509, the token consists of the certificate and a signature on a fresh *message*, stating among other things that the user knows the private key of the corresponding public key embedded in the certificate. In the case of anonymous credentials, the token is more complex.

2.1.1 Terminology

To have a common understanding, we first clarify some terms used throughout the text.

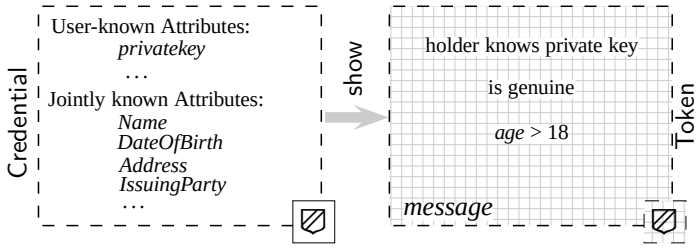


Figure 2.1: Definition of a Credential.

Strong Authentication. We define *Strong Authentication* as a cryptographic process based on a challenge-response protocol providing a strong assurance (computationally secure) on the authenticity of the claim (e.g., proof of identity).

SSL/TLS. We define an *SSL/TLS* secure communication as communication providing both confidentiality (e.g., no eavesdropping) and integrity protection (e.g., no tampering), with at least server authentication. SSL (*Secure Sockets Layer*) was originally defined by Netscape Communications [9]. TLS (*Transport Layer Security*), an IETF standard track protocol [4], is based on the earlier SSL specifications. Note that SSL/TLS does not provide non-repudiation for messages sent over the channel. We use SSL/TLS as an abbreviation for secure communication. SSL is a commonly known acronym. However, due to stronger security properties, TLS is the preferred choice.

2.1.2 Roles & Interactions

Similar to standard credential systems (e.g., X.509), we initially define three roles, as depicted in Fig. 2.2: a user U, a relying party RP, and an issuing party IP (also called issuer). The user obtains a credential during an issue-transaction with an issuer IP. Later, the user may authenticate to a relying party, during a show-transaction (also called a credential show), and if successful, the user gets access to the relying party's services. Note that this setup can easily be extended with multiple users, issuers and relying parties.

We will sometimes use the roles *prover* and *verifier* in transactions in which a party, the prover, has to cryptographically prove statements to another party, the verifier, and the latter verifies the correctness of these statements.

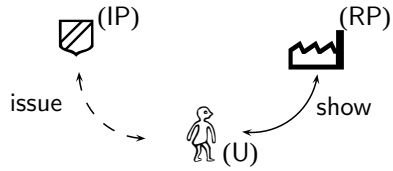


Figure 2.2: Overview of the Roles and Interactions in the Anonymous Credential System.

2.1.3 Groups, Protocols and Proofs of Knowledge

Assumptions

When proving the security of a protocol, we make assumptions. An adversary with unlimited power can break almost any cryptographic protocol. Therefore, to make sense, the power of the adversary is restricted. Here, we consider probabilistic polynomial time adversaries. We call assumptions believed to be hard for such adversaries, *complexity theoretic* assumptions. We briefly present the most important assumptions used in this dissertation:

Definition 1 (Discrete Logarithm Assumption)**.** *Let $\mathbb{G}$ be a (multiplicative) group Given $g \in \mathbb{G}$ and $y \in \langle g \rangle$, the discrete logarithm assumption requires that it is hard to find an integer x such that $g^x = y$.*

Definition 2 (RSA Assumption [RSA78])**.** *Given an RSA modulus n, prime e and a random element $y \in \mathbb{Z}_n^*$, the RSA assumption requires that it is hard to compute $x \in \mathbb{Z}_n^*$ such that $x^e \equiv y \bmod n$.*

Definition 3 (Strong RSA Assumption [BP97, FO97])**.** *Given an RSA modulus n and a random element $y \in \mathbb{Z}_n^*$, the Strong RSA assumption (SRSA) requires that it is hard to compute $x \in \mathbb{Z}_n^*$ and integer $e > 1$ such that $x^e \equiv y \bmod n$.*

Groups

Let $\{0,1\}^l$ denote the set of bitstrings with length l. 1^k is the bitstring of k ones. Let x be a bitstring, then $|x|$ denotes the length of the bitstring. Let S be a finite set, then $|S|$ denotes the size of the set. Let $y \in_R S$ denote that y is chosen uniformly at random from the set S. We use $=$ for equality, and $\leftarrow$ for an assignment.

Group. Informally, a group is a set of objects with an operation defined upon any pair of objects in the set. A (multiplicative) group $\mathbb{G}$ is a set $\mathbb{G}$ satisfying the following conditions:

Closure $\forall a,b \in \mathbb{G} : ab \in \mathbb{G}$

Associative $\forall a,b,c \in \mathbb{G} : a(bc) = (ab)c$

Identity $\exists$ a unqiue (identity) element $e \in \mathbb{G} : \forall a \in \mathbb{G} : ae = ea = a$

Inverse $\forall a \in \mathbb{G} : \exists a^{-1} \in \mathbb{G} : aa^{-1} = a^{-1}a = e$

A group is *finite*, if the set of objects is finite. If for all $a,b \in \mathbb{G}, ab = ba$, the group is called *abelian* (also *commutative*). The size of the set $|\mathbb{G}|$ is also called the *order* of the group. A group is called *cyclic* if there exists an element $g \in \mathbb{G}$ such that for any $b \in \mathbb{G}$, there exists an integer x such that $b = g^x$. g is called a *generator* of $\mathbb{G}$, which can be written as $\langle g \rangle$.

Prime order group $\mathbb{G}_p$. For prime order groups, for instance as used in the Pedersen commitments [Ped92], we generate a multiplicative group $\mathbb{Z}_q^*$, with large primes q and p, such that p divides $q-1$, resulting in a unique cyclic (multiplicative) subgroup $\mathbb{G}_p$ of prime order p, in which the discrete logarithm problem is hard.

Composite order/RSA groups. In several protocols, groups are based on a special RSA modulus n, being the product of two safe primes ($p = 2p' + 1$ and $q = 2q' + 1$). It forms a multiplicative group $\mathbb{Z}_n^*$ of integers less than n with a subgroup QR_n of quadratic residues modulo n, a cyclic group under multiplication. In short, a quadratic residue q is an integer for which there exists an integer x such that $x^2 \equiv q \bmod n$. In these cyclic groups with sufficiently large n, the RSA assumption holds.

Bilinear Maps. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be (multiplicative) groups of prime order p. A bilinear map (also known as a pairing) from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$ is a computable map e: $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

1. *Bilinearity*: for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$ and $a,b \in \mathbb{Z}_p : \mathsf{e}(u^a, v^b) = \mathsf{e}(u,v)^{ab}$.
2. *Non-degeneracy*: for all generators $g \in \mathbb{G}_1, h \in \mathbb{G}_2$: $\mathsf{e}(g,h)$ generates $\mathbb{G}_T$.
3. *Efficiency*: there is an efficient algorithm $\mathsf{BMGen}\ (1^k)$ that outputs (p, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, e, g, h) to generate the bilinear map (with k the security parameter and g and h generators) and an efficient algorithm to compute $\mathsf{e}(u,v)$ for any $u \in \mathbb{G}_1, v \in \mathbb{G}_2$.

Often symmetric pairings are used. In that case, $\mathbb{G}_1$=$\mathbb{G}_2$.

Sometimes, for clarity, we will add $\bmod x$ to the notation of the protocol when it might not be entirely clear from the context. If both groups of prime and composite order are applied in the same protocol, we denote the computations on group elements of the group of prime order in $\mathfrak{fraktur}$ font, for instance, $\mathfrak{y} = \mathfrak{g}^x$, and computations on group elements of the composite order group, in standard math font (e.g., $y = g^x$).

Notation for Protocols.

The following notation will be used to denote a local computation performed by X, a local computation by X after which the result is sent to Y and an interactive protocol between X and Y respectively:

$$\mathsf{X} : (out_x) \leftarrow f(in_x) \tag{2.1a}$$

$$\mathsf{Y} \leftarrow \mathsf{X} : (out_x) \leftarrow f(in_x) \tag{2.1b}$$

$$\mathsf{X} \leftrightarrows \mathsf{Y} : (out_c; out_x; out_y) \leftarrow f(in_c; in_x; in_y)\,. \tag{2.2}$$

The computations take common input in_c and secret input in_x, in_y from X and Y respectively and result in outputs out_x and out_y to X and Y respectively; out_c is common output. Empty inputs and outputs are represented by -. In the case of interactive protocols as in (2.2), the top arrow points away from the initiator of the protocol, in this example case Y.

Notation for Proofs of Knowledge.

We use the notation put forward by Camenisch and Stadler [CS97] for various proofs of knowledge of discrete logarithms and proofs of the validity of statements about discrete logarithms. For example:

A *zero-knowledge proof of knowledge* of integers α, β and δ, such that $y = g^\alpha h^\beta$ and $\tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta$ holds is denoted as follows:

$$PK\{(\alpha, \beta, \delta) : y = g^\alpha h^\beta \ \wedge \ \tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta\}\,,$$

where y, g, h are elements of a cyclic group $\mathbb{G}$ with generators g and h, and $\tilde{y}, \tilde{g}, \tilde{h}$ are elements of a cyclic group $\tilde{\mathbb{G}}$ with generators $\tilde{g}$ and $\tilde{h}$. The variables represented by Greek letters, denote the quantities of which knowledge is proved, while the other

parameters are public (i.e., known by both the prover and the verifier). Note, however, that for clarity, we sometimes use the actual name of the quantities being proved, instead of a Greek letter.

Interactive proofs can be converted into non-interactive ones, using the Fiat-Shamir heuristic [FS87]. This non-interactive version may then also be used for signing a message m and is denoted as follows:

$$SPK\{(\alpha,\beta,\delta) : y = g^{\alpha}h^{\beta} \ \wedge \ \tilde{y} = \tilde{g}^{\alpha}\tilde{h}^{\delta}\}(m)\,.$$

2.2 Cryptographic Building Blocks

Following Camenisch and Lysyanskaya [CL04], the credential systems (i.e., TYPE 2) require a signature scheme, a commitment scheme and efficient protocols for: (1) proving equality of two committed values; (2) getting a signature on a committed value; and (3) proving knowledge of a signature on a committed value.

In order to make this thesis self-contained, we first briefly summarize the main properties or the `Identity Mixer` library, followed by the main building blocks used in order to implement those properties, with a special focus on zero-knowledge proofs of knowledge.

Identity Mixer. The `Identity Mixer` library is a Java™-based anonymous credential system developed by IBM Research - Zurich, providing both strong authentication and privacy. It uses the CL-signature scheme by Camenisch and Lysyanskaya [CL03]. Next to proving knowledge of a valid signature on committed attributes, the library also supports proving statements about attributes contained in one or more credentials. Currently, the library [11] allows proving equalities (i.e., two attributes have the same value), inequalities (i.e., an attribute is less than or greater than a constant or another attribute), set membership (i.e., an attribute is included or not included in a given set of values), the generation of a *credential-specific* (i.e., based on the master secret) or *domain-specific pseudonym* (i.e., based on the master secret and a domain identifier), verifiable encryption (i.e., the verifier is ensured that a ciphertext includes a specific secret such that a certain third party can decrypt it) and *credential updates*.

2.2.1 The CL Signature Scheme

In a number of publications [CL01, CL03, CL04], Camenisch and Lysyanskaya put forward CL-signature schemes and a number of efficient protocols for implementing a proper anonymous credential scheme.

We briefly recall the CL-signature scheme, with a signer S and verifier V, for blocks of L messages as presented in [CL03] and implemented in [11]:

$\mathsf{V}: (pk_{Sig}, sk_{Sig}) \leftarrow \mathsf{setup}_{\mathsf{CL}}(1^k)$
Choose a special RSA modulus $n = pq$ of length $l_n = 2k$ with $p = 2p' + 1, q = 2q' + 1$ where p, q, p' and q' are prime. Choose, uniformly at random $h \in_R QR_n$ and $g, h_1, \ldots h_L \in_R \langle h \rangle$ with public key $pk_{Sig} = (n, g, h, h_1, \ldots, h_L)$ and private key $sk_{Sig} = (p)$.

$\mathsf{S}: (\sigma) \leftarrow \mathsf{sign}_{\mathsf{CL}}(m_1, \ldots, m_L, sk_{Sig})$
Let l_m be a parameter defining the message space as $m_i \in \pm\{0,1\}^{l_m}$ for $0 < i \leq L$. Choose a random prime e of length $l_e > l_m + 2$ and a random number $v \in_R \pm\{0,1\}^{l_n + l_m + l_r}$, with l_r a security parameter, and compute the signature $\sigma = (A, e, v)$ such that $A^e \equiv \frac{g}{h_1^{m_1} \ldots h_L^{m_L} h^v} \bmod n$. The latter requires knowledge of the order of the subgroup to compute the inverse of e.

$\mathsf{V}: (Bool) \leftarrow \mathsf{verify}(\sigma, m_1, \ldots, m_L, pk_{Sig})$
Parse σ as a tuple (A, e, v) and return true if $g \equiv A^e h_1^{m_1} \ldots h_L^{m_L} h^v \bmod n$, $2^{l_e - 1} < e < 2^{l_e}$ and $m_i \in \pm\{0,1\}^{l_m}$ for $0 < i \leq L$ holds, else return false.

2.2.2 Commitments

The second requirement of an anonymous credential scheme is a commitment scheme [Ped92, FO97, DF02]. In the context of computer science, committing is making the effects of a transaction permanent. In cryptography, however, a commitment scheme is the digital analogue of sealed envelopes. Committing then refers to *binding* a party (e.g., prover P) to a value such that she cannot alter this value, while keeping it *hidden* from other parties (e.g., verifier V). Later on, the commitment can be *opened*, revealing the contents of the commitment. Hence, the *binding* property of a commitment scheme denotes that one cannot successfully open the same commitment *Com* to two different values, while the *hiding* property denotes that the value that was committed, remains unrevealed.

The following commitment methods are relevant:

- P: $(params_{Com}) \leftarrow$ CommitSetup (1^k), a setup algorithm, returning public parameters of the commitment scheme.

- V $\leftarrow$ P: (Com, $open$) $\leftarrow$ Commit ($params_{Com}$, $value$), a new commitment for $value$, is generated using the randomness $open$.
- V: ($boolean$) $\leftarrow$ CommitOpen ($params_{Com}$, Com, $open$, $value$), returns true if the commitment Com corresponds to the committed $value$ with opening $open$.

Commitment schemes. Proofs of knowledge heavily rely on commitment schemes. Furthermore, the commitments of Pedersen [Ped92] and Fujisaki and Okamoto [FO97] present additional functionality, which will be useful in the proofs of knowledge. For instance, for proving, in zero-knowledge that the prover knows the opening (i.e., $value$ and $open$) of the commitment.

We briefly recall the Pedersen [Ped92] commitment scheme for prime order groups, which is based on the discrete logarithm problem:

CommitSetup. Generate group parameters describing a group $\mathbb{G}$ of prime order p with generators g and h, such that the discrete logarithm problem is hard and return $params_{Com} \leftarrow (p,g,h)$.

Commit. Commit to an arbitrary large integer $value$ by choosing a random $open \in_R \mathbb{Z}_p$, compute $Com \leftarrow g^{value}h^{open}$ and return $(Com, open)$.

Damgård et al. [DF02] present a generalization of the commitment scheme of Fujisaki and Okamota [FO97], which is essentially the same as above, but in a group of unknown order:

CommitSetup. Generate group parameters describing a hidden order group $\mathbb{G}$. Therefore, choose an l_n-bit RSA modulus $n = pq$ with p and q two safe primes. Choose $g,h \in_R QR_n$ with $g \in \langle h \rangle$ and return $params_{Com} \leftarrow (n,g,h)$.

Commit. Commit to an arbitrary large integer $value$ by choosing a random $open \in_R [0, \lfloor n/4 \rfloor]$, compute $Com \leftarrow g^{value}h^{open} \bmod n$ and return $(Com, open)$.

2.2.3 Proofs of Knowledge (PK)

Definition. Informally, a proof of knowledge is a proof in which the prover convinces a verifier that it *knows* a certain statement. Moreover, if the proof does not reveal anything but the truth of the statement, the proof of knowledge turns into a zero-knowledge proof of knowledge. In particular, it will not allow the verifier to convince a third party that the prover knows the statement.

We represent these statements in terms of language membership. Let L be a NP language, x an element of the language and $W(x)$ the set of witnesses for proving

that x belongs to the language, then we define the witness relation as $R = \{(x,w) : x \in L, w \in W(x)\}$. We use the Camenisch and Stadler [CS97] notation as in Sect. 2.1 for specifying proofs of knowledge. However, more concisely, we can denote such a proof as:

$$\mathsf{P} \rightleftarrows \mathsf{V} : (\mathit{Bool}; \text{-}; \text{-}) \leftarrow \mathsf{prove}_{\mathsf{PK}}(x; w; \text{-})$$

A proof of knowledge exhibits the following properties:

Definition 4 (Completeness)**.** *For every $(x,w) \in R$, the verifier V accepts after interacting with an honest prover P:*

$$\forall (x,w) \in R : Pr[(\mathit{true}; \text{-}; \text{-}) \leftarrow \mathsf{prove}_{\mathsf{PK}}(x; w; \text{-})] = 1\,.$$

Definition 5 (Soundness)**.** *A cheating prover P^* cannot convince a honest verifier V to accept a proof for which the prover does not know a corresponding witness, except for some small probability ε.*

$$Pr[(\mathit{true}; \text{-}; \text{-}) \leftarrow \mathsf{prove}_{\mathsf{PK}}(x; \text{-}; \text{-})] < \varepsilon\,.$$

A zero-knowledge (ZK) proof of knowledge has the additional property of zero-knowledgeness:

Definition 6 (Zero-knowledgeness)**.** *No cheating verifier V^* learns anything but the truth of the statement. Formally, for all probabilistic polynomial time (PPT) verifiers V^*, there exists a PPT simulator S, such that for all $(x,w) \in R, x \in L, w \in W(x)$, and auxiliary input $z \in \{0,1\}^*$, the following distributions are identical:*

$$\mathit{View}_{\mathsf{V}^*}[(\mathit{Bool}; \text{-}; \text{-}) \leftarrow \mathsf{prove}_{\mathsf{PK}}(x; \text{-}; z)] = S(x,z)\,.$$

In this thesis we use several protocols in order to prove statements about discrete logarithms (in both known and hidden order groups). These are often defined as Σ-protocols. Σ-protocols, visualized in Fig. 2.3, are three-move protocols in which the prover first *commits* to the randomness values corresponding to the values she wants to prove knowledge of. These commitments are sent to the verifier, who in turn replies with a *challenge*. Finally, the prover returns its *response*. The Σ visualizes the flow of the messages between the prover and verifier. In order to make the protocol non-interactive, the challenge is replaced by a hash [FS87] of the commitments of the first move, common inputs and a common string (context), consisting of a list of public parameters and the issuer's public key. If a message is included in the hash, the proof protocol is called a signature proof of knowledge (SPK). Similar to [11], the commitments of the first move are denoted as t-values, while the responses in the third move will be referred to as s-values.

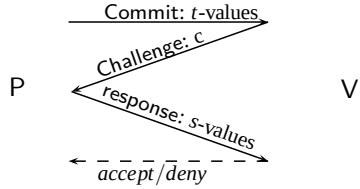


Figure 2.3: Visualization of a Σ-proof protocol.

Note that for both groups of known and unknown order, proofs exist, and may be combined into a larger proof. More details about how to combine these proofs can be found in [CKY09].

Protocols. Several protocols for different proofs of knowledge have been presented in the literature. We give a brief overview of some of those proofs of knowledge that are useful for building more complex proofs. We refer to the original papers for details of their implementation.

Proof of knowledge of a discrete logarithm. This proof has been shown for groups of known [Sch91] and unknown order [FO97, BCM05].

$$PK\{(\alpha) : y = g^{\alpha}\}.$$

As an example, if we assume a group of known order, the above proof of knowledge can be translated into the following three move protocol:

$$1)\ \mathsf{P} \rightarrow \mathsf{V} : \mathsf{P} \text{ chooses } r_\alpha \in_R \text{ from } \mathbb{Z}_p \text{ and returns as } t\text{-value } T \leftarrow g^{r_\alpha} \quad (2.3)$$

$$2)\ \mathsf{P} \leftarrow \mathsf{V} : \mathsf{V} \text{ replies with a challenge } c \in_R \mathbb{Z}_p \quad (2.4)$$

$$3)\ \mathsf{P} \rightarrow \mathsf{V} : \mathsf{P} \text{ sends as } s\text{-value } s_\alpha \leftarrow r_\alpha + c\alpha \bmod p$$

$$\text{to } \mathsf{V} \text{ who checks that } g^{s_\alpha} \equiv y^c T. \quad (2.5)$$

Proof of knowledge of a CL-signature. The following proof defines the proof of knowledge of a CL-signature (A, e, v), based on an RSA group, as it is used in

the `Identity Mixer` library.

$$
\begin{aligned}
PK\{(e,\{m_i : i \in A_h\},v) : \\
&\frac{g}{\prod_{i\in A_r} h_i^{m_i}} \equiv \pm A^e h^v \prod_{j\in A_h} h_j^{m_j} \\
&\forall i \in A_h : m_i \in \{0,1\}^{l_m+l_\phi+l_H+2} \\
&e - 2^{l_e-1} \in \{0,1\}^{l'_e+l_\phi+l_H+2} \\
\},
\end{aligned}
\tag{2.6}
$$

with A_h and A_r the set of hidden, resp. revealed attributes, l_m, l_H, l_e the bit length of the attributes, the challenge and e respectively, l_ϕ a security parameter that governs the statistical zero-knowledgeness and l'_e the size of the interval the e values are taken from.

Proof of knowledge of equality. In order to prove knowledge of equality, Chaum and Pedersen [CP93] present a protocol on how to prove that two public keys w.r.t. two different bases are equal:

$$PK\{(\alpha) : y = g^\alpha \wedge \tilde{y} = \tilde{g}^\alpha\}.$$

Which is generalized by Camenisch [CS97] as follows. A proof of knowledge of representations of $y_1,\ldots,y_n$ with respect to some of the bases $g_1;\ldots,g_k$ and that, additionally, some of the elements of the representations are equal. It is denoted:

$$PK\{(\alpha_1,\ldots,\alpha_u) : y_1 = \prod_{j\in I_1} g_j^{\alpha_{e_{1,j}}} \wedge \ldots \wedge y_n = \prod_{j\in I_n} g_j^{\alpha_{e_{n,j}}}\}.$$

with indices $e_{i,j} \in 1,\ldots,u$ referring to secrets $\alpha_1,\ldots,\alpha_n$ and the elements of I_i being indices referring to base elements $g_1,\ldots,g_k$.

We refer to the literature for several other proofs of knowledge: proofs for proving polynomial relations [FO97, Cam98]; proving that a value lies in a tight interval [Bou00] or non-tight interval [CFT98, BCDvdG06]; proof that a number is the product of two safe primes [CM99]; proofs for and/or/not-relations [CG08];...

Part I

Traditional Electronic Identities

The last decade, governments have started issuing electronic identities to their citizens. In general, electronic identities allow for digital document signing and strong authentication. In this thesis, we focus on the *strong authentication* capabilities of electronic identities. A substantial part of strong authentication applications using these electronic identities are towards government regulated services in the *public sector*. Examples are, but not limited to, online tax-declaration, reporting crimes and requesting official documents.

Electronic identities also offer perspectives for applications in the *private sector*. In this sector, there is a whole range of applications with strong authentication and authorization requirements.

However, using these government issued cards for all kinds of transactions requiring strong authentication is not straightforward. There are multiple reasons for that. The technologies used in eID cards, nowadays, require a substantial trust in multiple parties. Especially privacy may become a crucial concern. Furthermore, most electronic identity solutions provided to citizens are card-based, while there is a shift in the consumer electronics market towards mobile computing, that often do not provide the hardware required to read these smart cards.

We first look at existing electronic identities, particularly, the current Belgian eID and identify a number of concerns in Chapter 3, followed by Chapter 4, in which we present three eID based applications, that demonstrate how the current eID may be applied to offer secure, mobile and privacy-preserving eID applications. Finally, based on the analysis of the Belgian eID (i.e., both its advantages and drawbacks), we extract the requirements for future electronic identities offering improved properties (Chapter 5). This part of the thesis is mainly provided for setting the scene for electronic identities based on anonymous credentials.

Contributions: The evaluation of the Belgian electronic identity card briefly summarizes the joint work presented in [VLN+08, VLDD+09]. With respect to the applications, the secure home automation was peer-reviewed and presented at the *International European Conference on the Use of Modern Information and Communication Technologies* [LVNV08]. The ePetition application was peer-reviewed, presented and published in the pre-proceedings of the *IFIP/FIDIS-Summer School* at Brno [LVNV08] and further elaborated in a report [VLV+08]. Finally, the use of proxy certificates extending the Belgian eID, has been published in the proceedings of the *International Conference on Security and Privacy in Mobile Information and Communication Systems* [LVV+09], and was further validated in an article published in *Security and Communication Networks* [LNV+10].

Chapter 3

Belgian Electronic Identity Technology

3.1 Introduction

Belgium introduced an electronic identity card as one of the first countries in Europe, in 2003. The card allows Belgian citizens to identify, authenticate and sign electronic documents [CWP06]. It is clear that many application developers benefit from this evolution. However, the use of the eID card involves a few security and privacy pitfalls [VLN+08, VLDD+09], [Dum05], which become prominent as the use of the card is moving from governmental applications towards commercial applications.

Meanwhile, many other countries [FID06] also provide card-based eID solutions to their citizens. Examples are: Portugal [CNdS10], that introduced an electronic identity based on the Belgian eID; Italy [ACF+04], with a similar card containing a digital certificate for authentication and one for digital signatures; Germany [TR-11], in which the eID is part of a more complex infrastructure and allows a more fine-grained access control to attributes in the card; and the Netherlands, that introduced the DigiD which will in the future be a certificate-based eID card.

In contrast to the card-based solution, in which the eID is bound to a specific government issued citizen card, some countries [FID06] opted to provide a more flexible solution. For instance, as for the Austrian Bürgerkarte [LHP02] and the Finnish FineID [PS01], these eIDs can be implemented by a variety of entities, both public and private. Hence, the electronic identity can be issued to many tokens such as membership cards, banking cards and even mobile SIM cards.

We take the Belgian eID as a comparative case study for our further research, and therefore, have a brief look at some of the possibilities but also issues involved. These issues are mostly inherent to the technology used and are largely applicable to several other electronic identity schemes as well.

3.2 Brief Summary of the Design

The Belgian eID [CWP06] is a legible identity card, presenting identification information such as the card holders name, date of birth, place of birth, citizenship, a picture and also her national registration number (NRN). The latter is a unique nation-wide identification number, assigned to each citizen. In addition, it includes a Java Card chip allowing electronic transactions.

The chip contains the same personal information as printed on the card with in addition, the address of the card holder. This information is stored into 3 separate files, certified by the National Registration Office: an identity file, an address file and a picture file. The latter two are cryptographically linked to the former using digital signatures. Furthermore, the card contains 2 personal X.509 certificates and corresponding private keys, one for authenticating and one for making legally binding signatures. A (single) PIN is required to activate those private keys. Note that for usability reasons (i.e., SSL/TLS may request the user to enter his PIN unexpectedly), authentication is single-sign-on (SSO), i.e., the PIN is required only for the first authentication, as long as the card is not removed from the card reader or reset. In contrast, for digital signatures, a PIN is required for each signing operation.

3.3 Analysis

Below, we summarize the major advantages, but also constraints of the current Belgian eID. Though the card also supports physical identification and digital signatures, we focus on authentication. We refer to [VLN+08, VLDD+09] and [Dum05], for a more in depth analysis of the Belgian eID.

3.3.1 Security

Strong Authentication. A major advantage of the Belgian eID is that the card offers strong authentication, based on well-established and widely accepted standard building blocks, proved to be secure, such as RSA signatures, RSA authentication [13] and X.509 certificates [2]. Unlike the German eID [TR-11] in which an attacker

breaking the tamper resistance of the card may impersonate arbitrary citizens (i.e., attributes are not certified), breaking the tamper resistance of the Belgian eID only allows to impersonate the owner of the broken card [PWVT11].

User consent. For eID authentication and signatures, it is mandatory that the user enters his PIN. For digital signatures, this PIN is required for each signature, however, the single-sign-on property for authentication, only requires this once and subsequent authentications are performed transparently. Moreover, certified personal information may be downloaded from the card without user consent.

User control. Inherent to card-based solutions is the trusted path problem [GST95, BF99]. There is no trusted interface with the card. Hence, the user does not know what is going on at the card. For instance, she does not know what is being signed or which information is downloaded from the card, especially if the host is not trustworthy [JPH02, SCL01a]. In order to get more control, additional security measures must be put in place. For instance, ensuring a signer that what she signs is indeed what she intends to sign (WYSIWYS), may require additional trusted hardware [JA08]. Moreover, an untrusted host (e.g., infected with malware or trojans) could easily capture the PIN of the user and surreptitiously request signatures or authentication. Actually, for user authentication, because of the single-sign-on (SSO) property the PIN is not even required, if the user already authenticated before. This may be partially solved by using a card reader with a built-in pin-pad and display or using hardware-enabled Trusted Computing on the host [SCL01b, BCPP01, Bal09].

Similarly, if no proper mutual authentication and key agreement is used (e.g., the user authenticates only after an SSL/TLS session was initialized), a malicious service provider may relay the authentication towards another service, and hence gain access to those services.

Revocation. In order to provide a secure and accountable system, the possibility to revoke electronic identities is necessary. In other words, once a certificate can no longer be trusted (e.g., due to loss or theft), its rights for signing or authenticating should be withdrawn. Verifying the revocation status of these standard X.509 certificates is straightforward and easy to understand. There are two standard solutions for doing this.

The first solution uses a type of blacklist named **C**ertificate **R**evocation **L**ist (CRL) [2]. Such a list contains the serial numbers of revoked certificates and is signed by a trusted revocation authority. During authentication, the service provider checks for the presence of the certificate's serial number in the CRL. If the serial number was found, the certificate has been revoked, and the authentication should fail.

Table 3.1: Format of the Belgian National Registration Number (NRN).

NRN Format: YYMMDDXXXCC
→ YYMMDD : DATE OF BIRTH
→ XXX : DAYCOUNTER OF BIRTHS (ODD: MALE, EVEN: FEMALE)
→ CC : CONTROL NUMBER

Another solution to support revocation, is to use the **O**nline **C**ertificate **S**tatus **P**rotocol (OCSP) [16]. Here, a dedicated server, called OCSP responder, carries out this check for the verifier. The latter has the primary benefit of requiring less network bandwidth and thus enabling near real-time status checks, offering a higher security. However, for OCSP, the verifier must be online.

Although revocation may be handled efficiently, architectural decisions in the Belgian eID (e.g., signing certificates of citizens under 18 are suspended and citizens may choose to deactivate authentication and signing functionality) resulted in an inefficient implementation, having CRLs of a few hundreds of megabytes. Therefore, the use of delta CRLs, that only list the revoked certificates since a previously issued CRL, is advisable for verifying the revocation status.

3.3.2 Privacy

Controlled Release of personal data. Probably the most important weakness, when using the Belgian eID, especially in the private sector, is privacy. It is undeniable that the release of certified personal information can be a privacy threat as certified and hence, verifiable information is more valuable for third parties. Personal information included in the Belgian eID, such as the holder's name, address and picture, is certified by the National Registration Bureau and can be read from the card without any restrictions.

Furthermore, even if this information is not read from the card, important privacy intrusive information (e.g., the holder's *NRN* and full name) is revealed during authentication or when presenting digital signatures. For example, authentication using the Belgian eID will usually lead to the divulgement of important personal data such as the national registration number and the name of the card holder. Moreover, the date of birth and gender can easily be derived from this NRN (see Table 3.1).

Linkability. All signatures created with the Belgian eID (be it for authentication or for a digital signature) can be linked to the same person. These issues become

apparent when the eID card is being used across multiple domains. All transactions by the same citizen (e.g., eHealth, banking and commercial transactions) can be linked to one another. Hence, colluding service providers can easily link the information and compile extensive profiles of citizens. Moreover, in contrast to tracing based on identifying information such as IP address, MAC address, cookies, web bugs [AM03] and browser fingerprinting [Eck10, BFGI11], this information is certified, making a stronger link between different transactions.

Besides the possible linking by colluding service providers, if OCSP is used for revocation checking, the OCSP responder knows which certificate was used with a particular host at a particular time. This is a threat to the certificate holder's privacy, as extensive profiles can be compiled, but also to the service provider, as possibly important strategic business information (i.e., who is connecting when), is leaked towards the OCSP responder. Moreover, standard OCSP does not require encryption of requests, hence, others may possibly obtain this information as well.

Note that privacy threats become larger if the electronic identity card is used across multiple domains, as colluding parties (e.g., service providers or OCSP responders), may compile extensive user profiles.

3.3.3 User-friendliness.

Using the Belgian eID is very simple and straightforward. It is similar to other card technologies such as payment cards. On the other hand, citizens need to acquire a card reader and properly install the middleware, if they want to use their eID at home.

3.3.4 Mobility.

The Belgian eID is a Java Card-based solution. They are in a sense mobile, as citizens carry them with them everywhere they go. However, usage of the card requires a card reader connected to a preferably trustworthy host. Most mobile devices do not have an embedded card reader or cannot connect to an external one, and even if they do, it is often a cumbersome and unhandy solution.

Recently, in order to improve mobility, Van Damme et al. [VDWDCD11], demonstrated an integration of the Belgian eID on a secure microSD card. It was deployed on an Android compatible mobile device, resulting in a mobile and attractive electronic identity solution. Nevertheless, the drawbacks related to privacy remain.

3.4 Conclusion

The Belgian eID offers strong authentication, for both governmental and non-governmental applications. However, expanding the use of government issued electronic identities towards the private sector comes with considerable security and privacy threats. Moreover, card-based electronic identities also have inherent disadvantages, such as wear and tear of contacts (i.e., for contact cards), no trusted user interface and requiring a card reader, but also being less portable in a world where mobile devices become the standard for electronic transactions.

In the following chapter, we will present a number of application domains in which we build upon the strong properties of the eID, while tempering its drawbacks.

Chapter 4

eID Applications

Based on our analysis in the previous chapter, we present in this chapter three different strategies in which we try to alleviate the weaknesses, and take full advantage of the benefits. In Sect. 4.1, secure access to personal resources is based on the strong authentication of the eID, followed by Sect. 4.2, in which proxy certificates increase the mobility of the Belgian eID. In Sect. 4.3, in order to increase privacy when using the Belgian eID, the eID is used as a bootstrap to obtain a more privacy friendly credential for use in further transactions.

4.1 Secure eID Applications – Home Automation

Strong authentication is one of the major advantages of the Belgian eID card. In this section we demonstrate how applications may gain advantage in using the eID for strong authentication. An important limitation of using the Belgian eID in the private sector is the lack of privacy. In order to mitigate this drawback, we focus on secure access to personal resources (e.g., a personal server), which imposes only minor privacy concerns. As an example, we present a *Secure Home Automation* application (presented in [LVNV08]) using the Belgian eID for secure remote access.

4.1.1 Secure Home Automation

Home automation provides the automation of different tasks in private homes to increase comfort, security and lower energy consumption. The ultimate goal is to make life more convenient.

Traditional home automation systems (we call them building automation systems), typically consist of multiple hardware modules that are controlled by a dedicated hardware *controller*. The controller receives input signals from input modules and reacts by sending outputs to output modules. For instance, a controller may detect that someone pressed a light switch. The controller will react by sending output signals to certain lights.

However, besides building automation, current systems include multi-media and entertainment functionality, automation of recurring tasks and alarm functions. These new features require a new and flexible approach for administering the system. Moreover, many scenarios can be found where remote control may be desirable [DPG06]. The owner of the system may want to start the micro-wave or set the temperature in her house remotely. Or the owner may want to allow other persons to enter the house while being away. Users may only change the state of the home automation system after being authenticated, but also logging may help to detect misbehavior and to impose appropriate control measures.

Related work. Many home automation systems do not offer appropriate support for modifying the state of the home automation system remotely [KCKP02]. Moreover, enabling remote control requires appropriate security measures [SG01, BDK01, Pro05]. Although in theory, secure solutions for remotely accessing home automation systems have been discussed [ST03, MMS+07, MRB09], in reality secure remote access is often only a secondary requirement, implemented using simple password based authentication [AB05, JCC06]. Our solution, based on the strong authentication properties of the Belgian eID, may offer both increased security and usability, as it is carried along anyhow. Moreover, it is simple to temporarily provide access to others based on information revealed during authentication (i.e., name and date of birth).

Construction

We present a software assisted home automation system that makes those features possible. Fig. 4.1 illustrates the architecture of a software assisted home automation system. An *access control module* controls access to the Secure Home Automation system, which consists of two servers, namely a *home automation server* and a *management server*. We explain their functionality below.

Home Automation Server. The home automation server is a software component that communicates with the building automation system, controlling the hardware in- and outputs. It allows to appropriately react on different actions coming from both

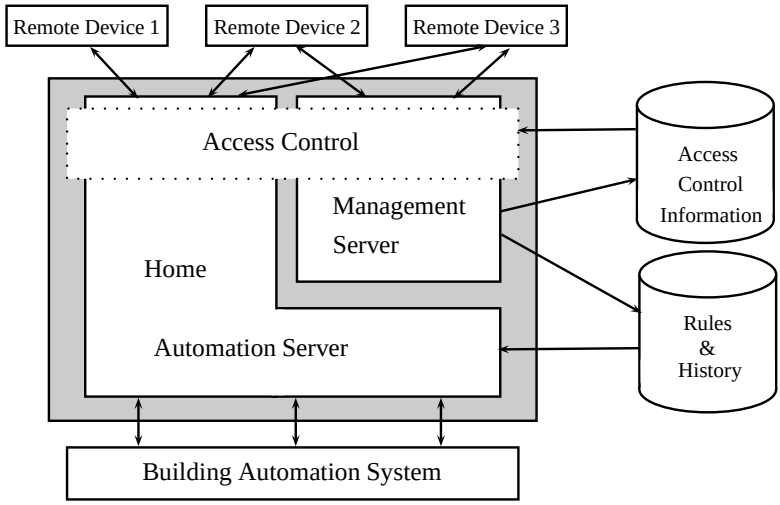


Figure 4.1: Overview of the Secure Home Automation System.

inside and outside the system, such as switching on a light switch, the automatic activation of scheduled tasks, or switching on the heating through a mobile device.

Management Server. The management server enables the reconfiguration of the system and the modification of the access control policies to services. Privileged entities (typically the owner of the system and possibly some additional persons) can use the management server for two purposes:

Configuration: privileged users can (re)configure the types of actions that must be performed if the state of one or more input modules has changed. For instance, pressing a button may have a different outcome after reconfiguration.

Role Based Access Control: privileged users (typically the owner of the building or house) can add and remove users and assign roles to these users or change them. For instance, the owner of the building can add her partner as an additional administrator and assign roles with more restricted privileges to the children. For registering new users in our prototype, it is sufficient to only enter the user's name, and date of birth. The administrator does not require the eID. Moreover, privileged users can define an access policy and assign that policy to certain roles. An access policy typically consists of a set of rules or permissions that define the types of actions that can be performed by users within that role.

The first functionality aims at easing reconfigurability of the home automation system. The latter aims at supporting secure remote access through role based access control (RBAC). As a result, we obtain a more consistent access management. Role based access control [FSG+01], provides a flexible approach to model access control policies. Permissions are associated with roles, and users are made members of relevant roles, hereby acquiring the roles' permissions. A role's permissions can be updated without updating privileges for every user individually.

Access control module. Remote access to the home automation system has to be restricted. Only privileged users may (de)activate modules. Similarly, access to the management server has to be restricted. Only a few users may change policies and add or remove users. The access control module supports SSL/TLS client authentication based on the authentication certificate in the Belgian eID and further authorizes the user, found based on the subject in the authentication certificate, by enforcing the policies defined using RBAC, before they are actually executed.

To demonstrate the feasibility of our approach, we applied the proposed architecture to the development of a software server for the Contatto HAS from Duemmegi [6].

Evaluation

Our secure home automation system supports remote authentication based on the Belgian eID, which provides strong authentication. Even if an adversary succeeds to steal a valid identity card, it is useless as a PIN code is required to go through the authentication phase successfully.

Since the home automation is actually under the control of the owner, the privacy concerns raised for the Belgian eID are not as important. However, security issues still prevail. Hence, owners are advised to be cautious when using the eID on an untrusted system. For instance, prevent SSO by removing the card, as soon as authentication has succeeded. Furthermore, only Belgian citizens can authenticate, and a card reader is required.

Although the card provides a powerful means for strong authentication, it is not sufficient to make the system secure. As part of the access control, we also require a proper way to define and enforce authorization. The prototype, therefore, implements role-based access control (RBAC). However, simple role-based access control has other disadvantages. For instance, in home automation systems, we may want to enable or disable roles periodically (e.g., temporal RBAC [BBF01, JBLG05] or make them context aware (e.g., context based access control [CLS+01, NS03, McD03]). For instance, access from untrusted networks may impose more constraints than access from the local home network.

4.2 Mobile eID Applications – Proxy Certificates

Nowadays, people rely on their mobile devices for a growing number of applications. eID-based applications should not be any different and, hence, should support mobile environments. However, the latter often have no built-in card reader for connecting with the Belgian eID. Below, we present BeID proxy certificates [LNV+10, LVV+09]. These are proxy certificates based on the Belgian eID supporting delegation, encryption and decryption. In order to make the Belgian eID technology more interoperable, proxy certificates may be transferred to mobile devices, expanding the eID functionality to mobile environments. Note that the non-card-based eID cards (i.e., the Austrian Bürgerkarte [LHP02] and the Finnish FineID [PS01]) may already provide most of the flexibility we get with our solution using proxy certificates.

Related Work. Pitkanen and Mikkonen [PM08] present a solution to combine the Shibboleth federated identity management with mobile environments, using short-lived certificates that allow for authentication, even in absence of the trusted parties (i.e., offline scenarios). Gome et al. [GHHF05] present a delegation solution using a centralized delegation authority using SAML assertions. Takaaki et al. [KNH+09] use SAML assertions in combination with a smart card making the delegation more user-centric. Similarly, Peeters et al. [PSCP08] present a generalized concept to support user-centric delegation in a federated setting. Although our solution is focused on the Belgian eID, it fits in their scheme.

4.2.1 Proxy Certificates

Although the eID card itself cannot encrypt nor decrypt data, the ability and right to do this can be delegated to the system to which the card reader is connected. This restricted delegation and proxying to another entity can be achieved through proxy certificates [20].

In the sequel we will abbreviate the phrase *"signed with the private key that corresponds to the public key certified in a certificate"* into *"signed with the certificate"*.

Proxy certificates. A proxy certificate is an extended version of a normal X.509 certificate. Proxy certificates are derived from and, hence, signed with a normal X.509 certificate or with another proxy certificate. For every proxy certificate, a new key pair is generated of which the public key is included in the proxy certificate. The proxy certificate and its corresponding private key can be used for asymmetric encryption, resp. , decryption or signing, resp. , verifying.

Modified standard. The standards for proxy certificates, as defined in the RFC 3820 [20], present some problems when used in the context of the Belgian eID. Therefore, some modifications have to be made (see Fig. 4.1).

First, according to the RFC, proxy certificates should be signed with the authentication certificate, since only this certificate of the Belgian eID contains the required value for the *key-usage* attribute.[1] The *key-usage* attribute defines the purpose of the corresponding private key. However, using the private authentication key sk_{Auth} is less secure than using the private signature key sk_{Sig}, since the PIN is only required for the first authentication, while it is required for every creation of a signature. Therefore, we propose to use sk_{Sig} for signing proxy certificates, despite the incorrect *key-usage* attribute.

Second, the Belgian legislation prohibits to keep track of the national registration number of Belgian citizens. However, the subject field in both eID certificates, contains the owner's name concatenated with her *NRN*. This implies that the eID certificates may not be stored in the file system or in a database. Nevertheless, the proxy certificate standard specifies that the subject of the issuing certificate is to be copied into the issuer and subject fields of new proxy certificates. To solve this problem, we propose to copy only the name of the owner into the subject field and to copy the hash of the signing certificate into the issuer field instead of the subject of the eID certificate. For validation purposes, the serial number of the issuer certificate is also included in the certificate (i.e., *issuerSN*). Additionally, another extra attribute indicates the certificate type: `BEID-PROXY`. In this scheme, we assume that when the eID certificate is revoked (e.g., in case of loss or theft), all issued proxy certificates are also no longer valid. Moreover, every proxy certificate must expire before the expiration date of the issuing eID certificate (i.e., $\Delta_1 \geq 0$), but obviously only after the issue date of the eID certificate (i.e., $\Delta_2 \geq 0$).

The protocol in Listing 4.2 demonstrates the creation of a BeID proxy certificate. The user U generates an asymmetric key-pair (1). A serial number is generated from the hash of the subject and the *issueDate* of the new proxy certificate (2-3). The subject in the hash avoids collisions, while the *issueDate* enables users to have more than one proxy certificate. A proxy certificate *proxyCert* is then generated and signed with the eID card (4-5). Finally, the signature is inserted in the certificate.

Revocation of proxy certificates. A special purpose server RA can keep track of revoked proxy certificates and publish the appropriate CRLs. To revoke a proxy certificate (cf. Table 4.3), the user authenticates with her eID card (1) and sends the proxy certificate she wants to revoke (2). If the user corresponds to the issuer of

[1] In the Belgian eID card, the authentication certificate has as key usage *Digital Signature*, while the signature certificate contains *Non-repudiation*!

Table 4.1: Content of the Modified Proxy Certificate.

	Signature Certificate	Proxy Certificate
SerialNumber	5874...2345	hash ($cert_{Sig}$.*Subject* || *issueDate*)
Subject	Name; *NRN*; ...	Name
SubjectPK	52:05:11:21:...:d2:7b	23:2b:24:a4:...:83:c5
ExpiryDate	*ExpiryDate*	min($cert_{Sig}.ExpiryDate - \Delta_1$, *issueDate* $+ \Delta_2$)
IssueDate	xxxx.xx.xx xx:xx	issueDate
CRL	crl.eid.belgium.be/...	crlLocation
Issuer	Citizen CA;BE;...	hash ($cert_{Sig}$)
Signature	15:f5:55:ff:...:20:f6	d5:fe:23:b4:...:4b:ab
Extensions		
IssuerSN	[not present]	$cert_{Sig}$.*serialNb*
Type	[not present]	`BEID-PROXY`

Table 4.2: Protocol for Creating a New Proxy Certificate.

createBeIDProxy([*attributes*]):

(1) U : (sk_U, pk_U) ← generateKeyPair(-)
(2) U : (*issueDate*) ← getDate(-)
(3) U : (*serialNb*) ← hash($cert_{Sig}$.*Subject*||*issueDate*)
(4) U : (*fullName*) ← substring($cert_{Sig}$.*Subject.Name*,";")
(5) U : (*issuer*) ← hash($cert_{Sig}$)
(6) U : (*expiryDate*) ← min($cert_{Sig}.ExpiryDate - \Delta_1$, *issueDate* $+ \Delta_2$)
(7) U : (*proxyCert*) ← generateProxy(
serialNb, *fullName*, *expiryDate*,
crlLocation, *issuer*,pk_U
`BEID-PROXY`,$cert_{Sig}$.*serialNb*)
(8) U ⇄ C : (*Sig*) ← sign(hash(*proxyCert*), sk_{eID})
(9) U : *proxyCert.Sig* ← *Sig*

the proxy certificate (i.e., hash($cert_{Sig}$) $\stackrel{?}{=}$ *proxyCert.Issuer*), the proxy certificate is revoked by adding its serial number to the latest CRL.

Table 4.3: Revoking a Proxy Certificate.

revokeBeIDProxy(*proxyCert*):			
(1)	U $\rightleftarrows$ RA :	(*cert*$_{Sig}$;-;-) $\leftarrow$ authenticate(*pk*$_{Sig}$;*sk*$_{Sig}$;-)
(2)	U $\rightarrow$ RA :	revokeCertificate(*proxyCert*)
(3)	RA :	if (hash (*cert*$_{Sig}$) $\neq$ *proxyCert.Issuer*) abort
(4)	U $\leftarrow$ RA :	(true) $\leftarrow$ addToCRL(*proxyCert.serialNb*)

Validation. A receiver validates a new proxy certificate by checking its signature, the validity period, its revocation status and by verifying the remaining certificate chain. Since only the hash of *cert*$_{Sig}$ is kept in the issuer field, name chaining (cf. RFC 3280 [2]) for certification path validation will fail. However, the extra attribute *issuerSN* included in the certificate binds the eID certificate to the proxy certificate. The first step in creating the certification path needs thus to be modified: the serial number of its issuing eID certificate must match the *issuerSN* in the proxy certificate.

To comply with Belgian legislation, the eID certificate is deleted after validation. Hence, future validation is not possible. However, verifying the validity period and the revocation status suffices if the proxy certificate was stored on a trusted location after its validation. Additionally, the revocation status of the eID certificate can be verified by checking the *issuerSN* of the proxy certificate in the CRLs of the Belgian eID.

The scheme described in this section allows for creating legitimate *proxy certificates* (created with the eID card), that can be used in many applications. Moreover, once a proxy certificate has been created, the Belgian eID is no longer required.

Evaluation. The *proxy certificate* mechanism allows owners of an eID card to set up mutually authenticated secure channels without the need for an additional trusted third party. Secure communication is even no longer restricted to SSL/TLS. The proposed system with proxy certificates makes it more flexible and extensible. Although not completely complying with the standards, the proposed scheme also supports asymmetric *encryption* with the eID card. Moreover, the proxy certificates can be used for asynchronous communication (i.e., recipients can decrypt confidential messages after the communication channel has been closed).

Once the receiver of a proxy certificate has deleted the eID certificate with which the proxy certificate has been signed (as is imposed by Belgian legislation), she can still check the validity of the proxy certificate and the revocation status of the eID certificate. Although other certificates in the validation path (i.e., *Citizen_CA*,

Belgium_Root_CA, ...) may have been revoked, the proxy certificate can include the serial numbers and the CRL-locations of these certificates as well as extra attributes to make the verification of the complete certificate chain possible. Note that, as defined in the RFC, optional attributes in the proxy certificate may further restrict its use.

The storage of the private key corresponding to the public key in the proxy certificate requires more trust in the device storing it. A developer must pay special attention on how to use this technology. For instance, when using proxy certificates on mobile phones, the developer should store the private key inside a secure element or SIM card. Also, certificate revocation and a limited validity period may further reduce the implications of a stolen private key. Revoking a proxy certificate causes less burden for the citizen than revoking her eID certificates, as in the latter case she will have to apply for a new eID card.

4.2.2 Applications

The Belgian proxy certificate scheme bootstraps lots of practical applications. The use of an existing nation-wide Belgian PKI infrastructure, maintained by the government considerably decreases the implementation costs of applications requiring strong security. Moreover, rights can be delegated to other devices and/or individuals. Usage constraints (e.g., purpose, location and time intervals) and liabilities can be included in proxy certificates.

Proxy certificates may facilitate the interoperability of eID infrastructures of different countries. More recently Sanchez et al. [SGGO10] presented a similar solution using proxy certificates based on electronic identities combined with SAML assertion to support cross-country interoperability. Nevertheless, legislation may impose restrictions, for instance, the Belgian eID certificate needs to be stored to verify the validity of a proxy certificate which is not allowed by Belgian legislation. As a result, the scheme cannot be used for services in the private sector. For instance, creating a proxy certificate that serves as a server certificate for a secure public website is not possible since its validity cannot be verified. Moreover, it is important to carefully handle the private keys associated with the proxy certificates. Below, three different application domains are presented in which the proxy certificates can be used, namely delegation of rights to another device, delegation of rights to another person and secure communication channels.

Delegation to another device

Experts anticipate that by 2020, mobile phones will be the primary Internet devices for most people in the world [ARR09]. It is to be expected that mobile devices

will become the main guardians and managers of our multiple electronic identities for a broad range of applications and services which include payments, e-health, e-government, etc. The Belgian eID card may be used to delegate electronic identities to mobile devices. Hence, Belgian proxy certificates can be stored and used on mobile devices to secure access to corporate email and resources, personal health data or e-government applications based on a nationwide infrastructure.

A major reason to implement single-sign-on for authentication is that a private key operation on the card is required to renew the session key during an SSL/TLS session. The server typically fixes the time interval between two session renewals. However, the single-sign-on feature in the Belgian eID card induces severe security and privacy threats as discussed in [VLDD+09]. Proxy certificates may solve this problem as the user may temporarily delegate the right to access –that specific service– to the browser.

Further, it supports access control to physical buildings, offices, etc. A building automation system may allow access to a building or a secure area based on the identity of individuals. Using proxy certificates in combination with, for instance, a mobile device, the identity of the individual can be proved more easily since a card reader is not required.

Delegation to other individuals

A second application domain is where one person delegates her rights to another individual by means of a proxy certificate to allow that individual to sign or authenticate in the name of the delegator. The following gives two examples:

A first example concerns an online-tax-declaration or transactions in a company. Currently, an online-VAT-declaration requires a digital certificate allowing an employee to submit the declaration, instead of the director of the company. If the Belgian eID card were used, it would require the director of the company to hand over her eID card to the employee. However, using a proxy certificate, the employee can declare the taxes or transactions *in the name* of the director. Moreover, the proxy certificate can include restrictions. For instance, it may include that it is only valid for specific actions, during office hours, by a specific employee.

A second example in which this kind of delegation may be useful is for transferring privileges to another person, such as the right to enter a building. Authorization may be based on information included in the certificate (e.g., building/room nr, time interval, dates)

Secure communication

In contrast with the Belgian eID card, proxy certificates facilitate encryption and decryption. As such, based on an existing nation-wide PKI infrastructure, it is possible to implement *peer to peer* SSL/TLS communication between two Belgian citizens. For instance, in a secure chat application, trust can be established between two persons based on these proxy certificates, as both are signed with their respective eID cards.

4.3 Privacy-Preserving eID Applications – PetAnon

When directly used to access electronic services, the eID card involves some important privacy issues. A possible work-around is to use the eID as a bootstrap (similar to [VDDN+08] and [DKD+09]) in order to obtain a more privacy-preserving credential. In this section, we present an electronic petition application [LVV+08], in which the eID is used to obtain a petition credential. Petition credentials are implemented as `Identity Mixer` anonymous credentials, which present improved privacy and security properties.

4.3.1 Electronic Petitions

In a petition, opinions of people are collected and processed. In paper-based petitions the collection and processing takes a lot of time and effort. Electronic petition systems (ePetition), however, offer several benefits with respect to the paper-based petitions. ePetitions enable users to sign petitions anywhere at any time and now reach wider sections of society. Moreover, automatic processing of the results can make the petitions more reliable.

On the other hand, electronic petition systems introduce new problems. Sometimes they may present unreliable results if they cannot prevent that a user signs a petition more than once. Other systems request personal information from the signer to prevent multiple signatures by the same user. However, these systems are usually not privacy friendly.

Related Work. Diaz et al. [DKD+09] where the first to present (2008) an anonymous electronic petition system based on e-tokens, obtained using the Belgian eID. Whereas Diaz et al. focused on its conceptual construction and apply the `Identity Mixer` library as a validation of their concept, we focused more on its practical implementation and, exploit the capabilities of the `Identity Mixer`. Our solution focuses on *electronic polls*, in which the user may select one of

several options and we allow her to optionally disclose information in order to allow *privacy friendly* statistics. To prevent a citizen to sign the same petition twice, the authors of [DKD+09] employ periodically spendable e-tokens presented in [CHK+06], whereas we use the construction presented in [VD09]. Finally, we added an extra verification during issuance, such that in case of loss of her credential, the requirement that the user can sign a petition only once, remains valid. More recently, an electronic petition system has been developed [Cas11] in which DAA anonymous credentials [BCC04] where implemented on a smart card.
Electronic online petitions are closely related to electronic online voting [BT94, Acq04b, CCM08, DKR06, LL11]. However, the requirements of electronic voting are harder to achieve. For instance, receipt-freeness [BT94, DKR06], in which a voter cannot prove anything about his vote, or the stronger notion of coercion-resistance [JCJ05, DKR06] cannot be achieved by the use of anonymous credentials and provable pseudo-random functions alone. In anonymous electronic petitions, we do not require these strong properties, resulting in a simpler and more efficient solution.
Also the `Identity Mixer`-based anonymous reviewing system [NDDD06] has similar requirements, in which a reviewer can only review the same paper once. However, their solution is less efficient, as it requires the credential to be updated after a review.

We now present PetAnon, a privacy-preserving petition system using `Identity Mixer` anonymous credentials. PetAnon combines good privacy properties with reliable results. First, we present the requirements of the system, followed by the protocols and an evaluation of the solution.

Requirements

The requirements of the privacy-preserving ePetition system are discussed below. They are classified according to security and privacy requirements.

Security requirements

(S1) A user can sign a certain petition only once.

(S2) A petition may address only a subset of the potential signers; therefore the signer may be required to prove that she belongs to that subset.

(S3) A user can verify that her signature is included in the petition's database.

(S4) Everyone can verify the correctness of the petition results.

Privacy requirements

(P1) Signers are anonymous.

(P2) Signatures cannot be linked to a user. Moreover, signatures of the same user, in different petitions, cannot be linked to each other.

(P3) A petition may request optional attributes that the user can release in order to get more differentiated results. The user has the choice if she wants to disclose these attributes or not.

Protocols

Roles and setting. A user U possesses an eID card, which is used when U authenticates towards the registration server IP. This authentication is required before IP issues a *voting credential* to U that can be used to sign an ePetition on a server of a petition organizer O. This voting credential can be used for multiple petitions of different petition organizers. The registration server IP has a certificate containing the public cryptographic information used in the anonymous credential-issue and credential-show protocols.

Setting up an ePetition. Table 4.4 shows the protocol for setting up an ePetition. The petition organizer O contacts the registration server IP and obtains a petition certificate, certifying petition specific information, that signers use for signing petitions.

Therefore, the petition organizer contacts IP. IP issues a certificate to O that contains O's public key pk_{pet}, a unique provable one-way function f_{pet} and petition specific information (e.g., name, participant info, ...). This function, as well as the petition-info are included in a (X.509) *petition certificate* $cert_{pet}$ that is issued by IP to O. As a result, the latter obtains a corresponding public key pk_{pet}.

Table 4.4: Setting up an ePetition.

setupPetition()			
(1)	$\mathsf{O} \rightleftarrows \mathsf{IP}$	:	$(\mathsf{true};\text{-};\text{-}) \leftarrow \mathsf{authenticate}(f_{pet}, pk_{pet}, info_{pet}; sk_{pet}; \text{-})$
(2)	$\mathsf{O} \leftarrow \mathsf{IP}$	:	$(cert_{pet}) \leftarrow \mathsf{issueCert}(f_{pet}, pk_{pet}, info_{pet}, sk_{\mathsf{IP}})$

Note that a *provable one-way function* $out \leftarrow f(inp)$ is a pseudorandom function such that the party knowing inp, can easily prove, in a zero-knowledge proof, that she

knows an *inp* such that $out = f(inp)$ holds. We refer to [VD09] for an instantiation of this provable pseudorandom function.

Retrieving a voting credential. In order to sign petitions, U has to obtain a voting credential (i.e., an `Identity Mixer` anonymous credential). Table 4.5 presents the protocol steps for doing so. The user authenticates using her eID card (1). This action reveals the personal data contained in the eID card to IP.

Table 4.5: Retrieving a Voting Credential.

getPetitionCredential()

(1)	U $\rightleftarrows$ IP	:	$(cert_{aut}, props_{eID}; \text{-}; \text{-}) \leftarrow \mathsf{authenticate}(cert_{aut}; sk_{aut}; \text{-})$
(2a)	IP	:	if (!credExists(hash($cert_{aut}.Subject$)))
(2a.1)	U	:	$rand_{\mathsf{U}} \leftarrow \mathsf{genSecureRand}(\text{-})$
(2a.2)	U	:	$(Com, open) \leftarrow \mathsf{Commit}(cert_{pet}.params, rand_{\mathsf{U}})$
(2a.3)	U $\rightarrow$ IP	:	$(Com) \leftarrow \mathsf{prove}_{\mathsf{PK}}($ $\{x \mid Com = \mathsf{Commit}(cert_{pet}.params, x)\}, Com; open; \text{-})$
(2a.4)	IP	:	$rand_{\mathsf{IP}} \leftarrow \mathsf{genRand}()$
(2b)	IP	:	else
(2b.1)	IP	:	$(serialOld, Com, rand_{\mathsf{IP}})$ $\leftarrow \mathsf{retrieveCredInfo}(\mathsf{hash}(eID.cert_{aut}.Subject))$
(2b.2)	IP	:	$\mathsf{revokeCred}(serialOld)$
(3)	IP	:	$serialNew \leftarrow \mathsf{getNewSN}()$
(4)	U $\leftrightarrows$ IP	:	$(\text{-}; cred; \text{-}) \leftarrow \mathsf{issueCred}(serialNew, Com.rand_{\mathsf{U}}, rand_{\mathsf{IP}},$ $\mathsf{subset}(props_{eID}))$
(5)	IP	:	$\mathsf{store}(serialNew, \mathsf{hash}(cert_{aut}.Subject), Com, rand_{\mathsf{IP}})$
(6)	U	:	$\mathsf{store}(cred)$

Every citizen is only allowed to have one voting credential. This is first checked by IP. If the user did not register previously (2a), the user generates a (long) secure random number (2a.1), puts it in a commitment (2a.2), which is sent to IP, and proves that she knows the committed value (2a.3). IP also generates a (potentially shorter) random value (2a.4). These two random values will be used to generate petition specific pseudonyms to prevent voting multiple times for the same petition (cf. further below).

If U previously had been issued a voting credential (2b), *Com* and $rand_{\mathsf{IP}}$ are retrieved from IP's storage and will be reused (2b.1). Also the credential's serial number is retrieved, which allows to revoke this credential before issuing a new one (2b.2).

After a serial number for the new credential is generated (3), all the parameters for the credential issuance are known and the voting credential is issued (4). It contains

the two random values (RN, $rand_{\mathsf{U}}$), the serial number (*serial*) and a subset of the attributes (or properties thereof) that were extracted from the eID card. Note that IP never gets hold of the user's secure random number.

Finally, U stores the credential (6), and IP stores the commitment, the other random number and the serial number, as well as the hash of the subject in the authentication certificate (5). This will allow IP to check whether a user already has been issued a voting credential, to revoke a voting credential and to issue new ones.

Signing a petition. Table 4.6 gives the protocol steps for signing a petition. Initially, the petition organizer O, authenticates towards the user using her petition specific certificate $cert_{pet}$. O additionally sends a policy specifying an overview of required and optional personal properties that must or can be proved when signing the petition. For instance, to sign a certain petition the participant must be older than 18 years. However, it is up to the voter whether or not she reveals her gender or zip code. Finally, O sends a list to U of choices for which the user can vote (1).

Table 4.6: Signing Petitions.

signPetition			
(1)	$\mathsf{U} \rightleftarrows \mathsf{O}$	:	$(policy, choices[]; \text{-}; \text{-}) \leftarrow \mathsf{authenticate}(cert_{pet}; sk_{pet}; \text{-})$
(2)	U	:	$(Nym) \leftarrow cert_{pet}.f_{pet}(cred.rand_{\mathsf{U}}, cred.rand_{\mathsf{IP}})$
(3)	$\mathsf{U} \rightarrow \mathsf{O}$	:	Nym, $(props) \leftarrow \mathsf{select}(policy)$, $(choice) \leftarrow \mathsf{select}(choices[])$
(4)	$\mathsf{U} \rightleftarrows \mathsf{O}$	:	$(proof; \text{-}; \text{-}) \leftarrow \mathsf{showCred}(props \&\& cert_{pet}.f_{pet}(cred.rand_{\mathsf{U}}, cred.rand_{\mathsf{IP}}) = Nym; cred; \text{-})\{choice\}$
(5)	O	:	if $(\mathsf{petitionSigned}(Nym))$ $\mathsf{abort}()$
(6)	$\mathsf{U} \leftarrow \mathsf{O}$	:	$(voteNr) \leftarrow \mathsf{getNextVoteNr}()$
(7)	$\mathsf{U} \leftarrow \mathsf{O}$	:	$(signature) \leftarrow \mathsf{sign}(voteNr, \mathsf{hash}(proof), Nym, sk_{pet})$
(8)	O	:	$\mathsf{store}[voteNr, Nym, proof, signature]$
(9)	U	:	$\mathsf{store}[signature, \mathsf{hash}(proof), voteNr]$

With the help of the petition's pseudorandom function and the two random values embedded in the voting credential, the user generates her petition specific nym (2), and sends it to O, together with the description of the personal properties that U is willing to disclose and her choice for which she wants to vote (3).

Now, the credential show protocol is run (4): U proves the selected properties, as well as that the petition specific nym for that user is correctly formed based on the random values contained in the credential, thereby anonymously signing the user's choice .

If that nym has not been used in that specific petition (5), the protocol continues by generating a vote number. This is a reference to the petition-record that is being generated. The vote number, the hash of the proof and the user's nym are signed with the petition secret key, and stored by O together with that signature. The resulting record is made public. The signature is sent to and stored by U and allows U to check that her anonymous signature is included in the petition's database and to file a complaint otherwise and to verify whether the record has been modified by O.

Verification. The user can request from O the record with index *voteNr*, which was signed by the user. If the vote was tampered with, either the O-provided signature will no longer be the same as the signature stored by U, or the O-provided signature will no longer match the $(proof, nym, voteNr)$-tuple made public by O.

If all the records are made publicly available, everyone can verify the correctness of the petition by verifying for each record the proof and the respective signature.

4.3.2 Prototype

The prototype consists of two applets and two servers. The first applet is used to obtain a new credential. PetAnon uses the Belgian eID card to authenticate to the registration server and retrieve identity information of the owner. However, other eID technologies could be used. The *attributes* embedded in the credential are date of birth, zip code and gender. The second applet allows users to sign a petition using their voting credentials. We now enumerate some implementation details of PetAnon.

First, since privacy legislation prohibits the storage of *NRN*, a pseudorandom function is used to mask *NRN*. Second, in the prototype, the provable pseudorandom function f_{pet} was implemented using a discrete logarithm commitment. Third, signatures of knowledge are not available in the version of the `Identity Mixer` library,[2] at the time the prototype was implemented. As a result, U cannot sign the chosen option anonymously. The problem was solved by embedding extra *choice*-attributes in the credential, containing the values 1 up to n, with n the maximum number of choices in the petitions for which the credential can be used. To sign a specific choice, the *choice*-attribute with the value equal to the choice number is disclosed to the petition organizer, while the signer only proves knowledge of the other options. However, this solution is less efficient than using anonymous signatures. In order to ensure that O does not change the order of the choices of the petition, the order may be defined in the petition certificate.

[2]The implementation was done using a pre-released version (in 2008) of the `Identity Mixer` library.

Performance results. The performance of our system is evidently dominated by the `Identity Mixer` protocols. Table 4.7 shows measurements of the time needed by `Identity Mixer`. Signing a petition where one releases ones age interval and zip code, using a 1.83 GHz processor by both user and petition server requires about four seconds if a 1024-bit RSA modulus is used. On average, between 50 and 60 percent of the computations is done at the user side, meaning that a server with four 1.83 GHz processors would need about 5.5 days to handle 1,000,000 signatures. Counting and verifying the votes would require a similar amount of time. Moreover, in order to deploy PetAnon properly, anonymous communication is to be used, which will cause additional communication latency.

Table 4.7: Measurements of the Performance of the `Identity Mixer` in PetAnon on an Intel(R) Core(TM) CPU T5600 @ 1,83GHz.

(ms)	**512-bit**	**1024-bit**	**2048-bit**
Vote cred. issue	459	1082	3907
show: Nym & vote choice	343	828	2974
additional show: zip	266	741	2556
additional show: age interval	611	2565	12470
Verification show proof	246	344	1375
Verification zip proof	118	291	1099
Verification age int. proof	345	1317	5589

The proofs that are stored in the PetAnon system, contain the proof, as well as the XML description of what has been proved. To verify such a stored proof, only the public key info of IP is required. We see that a petition record will have a size of about 12.6 kB, meaning that a petition with one million signers requires about 12 GB of storage. The size of proofs could be optimized somewhat, e.g., by compressing the XML descriptions.

Table 4.8: Required Storage Space for Proofs.

(kB)	**512-bit**	**1024-bit**	**2048-bit**
Nym	512-bit	1024-bit	2048-bit
Signature	512-bit	1024-bit	2048-bit
Show	3.1	3.4	3.8
+ zip	2.2	2.3	2.5
+ age int.	5.1	6.8	10
Total	10.5	12.7	16.8

4.3.3 Evaluation

We first evaluate the proposed solution with respect to the requirements we initially put forward:

(S1) is easily fulfilled, as for each petition the user is known by O under a petition specific nym. If that nym has already been used in that specific petition, the vote is cancelled.

(S2) and **(P3)** are fulfilled. Some attributes in the credential show may be required by O, while for others it is up to whether she wants to disclose the information or not.

(S3) is fulfilled. U can detect if her vote was tampered with based on the O-provided signature.

(S4) If the records are made public, everyone can verify the correctness of the petition by verifying the proofs and signatures.

(P1) Using the `Identity Mixer` credential show protocol, as long as no identifying attributes are revealed, the user U remains anonymous, and different shows are unlinkable. Moreover, privacy is preserved in case of collusion of IP and O.

(P2) is fulfilled. To sign a petition, U authenticates anonymously using her credential (*cred*). Signing the petition is done anonymously, and there are no identifiable actions linked to the signature.

The construction results in a fair and anonymous petition system using anonymous `Identity Mixer` credentials. The eID card is used to obtain a voting credential. Thereafter, petitions can be signed. The system has already been used in small-scale settings. The performance of the `Identity Mixer` system can become a bottleneck when a huge amount of users participate. Hence, a more distributed architecture will be necessary. Note that more recent versions of the `Identity Mixer` library offer an improved performance (see Chapter 7).

4.4 Conclusion

In this chapter, we presented three different strategies in which we try to alleviate the weaknesses, and fully benefit of the advantages of the Belgian eID. Nevertheless, the application domains were carefully studied to minimize the limitations of the current eID and cannot simply be applied to other domains. In the following chapter, we will make a more in-depth evaluation and consider what an actual eID should provide in order to make it really of use in today's society.

Chapter 5

eID Requirements Study

In the previous chapters, we analyzed the advantages and drawbacks of the Belgian eID. The major advantage of the technology is that it provides *strong authentication*. We also mentioned the *simple and efficient revocation* mechanism that may be employed. The major disadvantage of the eID card is the lack of *privacy*, which is inherent to the technology used. More specifically, different authentications and signatures are linkable. Hence, colluding service providers may compile extensive user profiles. In addition, in a growing mobile world, the current card-based solutions do not offer a feasible solution for using the eID on mobile devices. Hence, its use for services in the private sector is limited and may in the future probably get less adopted, as nowadays more and more applications and services are accessed through mobile devices.

Nevertheless, in certain domains next to the government regulated services, the eID card is a valuable technology to offer more secure environments. Therefore, we presented three different application domains in which the eID card can be employed, while mitigating its weaknesses. In the first domain, privacy is only of little concern, and we can directly employ the strong security properties of the eID card. In the second domain, in order to expand the eID setting to mobile environments, we proposed BeID proxy certificates, and in the last application domain, to get a more privacy-friendly electronic identity, we actually use the eID card as a bootstrap for obtaining a more privacy-friendly credential.

As may be noticed, these application domains were carefully studied to minimize the limitations of the current eID. But also many other applications could benefit from a strong authentication mechanism. However, the current state-of-the-art electronic identities are not appropriate to do so. Anonymous credential systems, on the other hand, may offer better properties for a secure and also privacy-preserving electronic

identity.

In the rest of this dissertation, we will evaluate the use of anonymous credentials for implementing electronic identities. As a basis for comparison, we first list the requirements for electronic identities, present an attack model and corresponding assumptions. Finally, we review the threats and issues for the Belgian eID based on the above model.

5.1 eID Requirements

Based on our findings in the previous chapters, we put forth the following requirements for future electronic identities.

Security. Evidently, an electronic identity should support secure electronic transactions. First, it should support *strong authentication*. In addition, requiring knowledge of the PIN is also used as a form of *user consent* such that transactions cannot be performed surreptitiously. Finally, a last, maybe obvious requirement is an *efficient revocation* mechanism. As we will see later, this is not always the case.

Privacy. Most eIDs currently do not provide sufficient privacy. However, this is an important requirement, as most of them are used to support transactions in both the public (e.g., government services) and the private domain (e.g., commercial applications). Therefore, an important requirement is to prevent *linkability*, unless required for personalized services. Moreover, the *selective disclosure and controlled release of personal information* help to keep the user's personal data protected and support data-minimization at the side of the service provider.

User-friendliness. To stimulate users to actually use their eID, usability and user-friendliness are important requirements. Unfortunately, these requirements compete with the security and privacy requirements and actions taken to improve user-friendliness sometimes lead to insecure systems and vice versa.

Mobility. Nowadays, electronic transactions are increasingly being performed from mobile devices such as smartphones and tablet computers. Hence, electronic identities should be more flexible and thus allow secure transactions also in these environments.

5.2 Attack Model

Electronic identities are used in a variety of ways. Currently, the Belgian eID is used in both trusted (e.g., personal home computer) and untrusted environments (e.g., at a shop, Internet café, insurance company, banks). But also the services for which the electronic identity is used, may require different trust properties.

For the setting of an electronic identity, we consider the following adversaries:

Communication (*C*). A passive adversary may eavesdrop the communication both globally (i.e., for remote communication) and locally (i.e., communication with the eID). An active adversary can have control of the communication system. Messages can be dropped, modified or inserted.

eID (*eID*). The functionality of the eID could be emulated by an adversary.

Host (*H*). The adversary can gain full or partial control over the local host (e.g., malware).

Service Provider (*SP*). The service provider may get corrupted, and possibly attack eID users.

5.3 Assumptions

We take the following the assumptions:

Issuance. The issuer and trusted parties are assumed to be trusted w.r.t. the authenticity of the information and correctness of the applet on the eID, hence, only valid eIDs are issued.

Cryptography. Cryptographic primitives (e.g., RSA) used for user authentication are considered to be computationally secure, and cannot be broken. An adversary cannot generate valid eID certificates or signed identity files.

Tamper Resistance. It is hard to extract private keys or the PIN from the card. Moreover, the applets on the card cannot be changed by an unauthorized party. In the unlikely event that information is extracted (e.g., through side channel attacks), the security breach should be limited.

5.4 Threats and Issues for the Current Belgian eID

Based on the requirements, the attacker model and the assumptions above, the following threats remain. The abbreviations between brackets denote the attack profiles.

Identification.

(*H*) Identity, address and picture file may be read without the user's consent.

(*eID*) If no authentication is required, the identity, address and picture file of another user may be used to identify.

Authentication.

(*H*, *SP*) Surreptitious authentication (by abusing SSO or PIN caching). Moreover, the user may authenticate with the wrong key and, hence, sign a document instead.

(*SP*) The service provider may implement man-in-the-middle attacks towards another service if no appropriate mutual authentication mechanism is used, in which both the user and the service provider are ensured to be authenticating to the correct party, and only to that party.

(*C*) Secure communication is not a requirement, hence, insecure communication may be eavesdropped or tampered with.

(*SP*) Multiple service providers, and possibly also OCSP responders may collude and aggregate extensive user profiles based on linkable information being revealed during each authentication.

Digital Signature.

(*H*) The user may sign another document than the one intended.

(*H*, *SP*) The user may sign a document, while she thinks she is authenticating as the same PIN is used for both authenticating and signing.

In addition, although using the Belgian eID is simple and straightforward, in other words user-friendly, it has limited mobility properties as it requires a card reader, which is often not available for mobile devices.

5.5 Conclusion

In the previous chapters, we learned that traditional electronic identities lack a number of properties that should be fulfilled when they are being used in cross-domain settings (i.e., both public and private domain). We presented the requirements for a new electronic identity, an attack model and assumptions that allows us to analyze such electronic identities. As a basis for comparison, we evaluated the Belgian eID based on these results.

In the remainder of this thesis, we will analyze how anonymous credentials may replace the current state-of-the-art electronic identities. Within the anonymous credentials setting, we look at specific topics, required to bring them into practice.

Part II

Mobile Anonymous Authentication

Anonymous credential systems are, from a privacy point of view, the most suitable technology for realizing privacy-preserving authentications. A drawback of these credential systems is that they require substantially more computational effort than traditional authentication technologies. With the increasing computational power of mobile devices such as smartphones and mobile tablets, they have been emerging as potential target platforms for the wide-spread deployment of anonymous credential systems.

In this part, we analyze how mobile devices can feature anonymous credentials (i.e., `Identity Mixer` credentials), in order to protect our personal information, the advantages we gain, but also which problems still need to be solved to make anonymous credentials effective.

In Chapter 6, we provide a number of building blocks in order to make mobile authentication simple and easy to implement. In Chapter 7, these building blocks are used in a prototype implementation on an Android mobile device. Finally, in Chapter 8, we evaluate the use of anonymous credentials in this setting and reflect on the results in Chapter 5, in which we identified the basic properties we search for in future electronic identities.

Contributions: An article [BDDL+12] is accepted for the *IFIP TC6 and TC11 Conference on Communication and Multimedia Security*, presenting a privacy-friendly and secure authentication application for mobile environments. This application is presented in Chapter 7. In addition, a number of building blocks are presented to help the development of secure mobile applications. These building blocks are discussed in Chapter 6.

Chapter 6

Building Secure and Privacy-friendly Mobile Applications

6.1 Introduction

Smartphones are quickly becoming the standard for mobile phones. In Belgium, 1.2 million smartphones were sold last year. One out of five citizens already possesses a smartphone.[1] Furthermore, in 2011, worldwide end-users bought 1.8 billion units, an 11.1 percent increase compared to 2010.[2] But also the tablet market is growing explosively, since the introduction of the iPad in 2010. In the first nine months of 2011, in the Benelux almost 450,000 tables were sold.[1]

Currently these mobile devices (i.e., smartphones and tablet PCs) already provide lots of features, offering new opportunities for all kinds of applications. However, in order to be taken up by a majority of users, simplicity and attractiveness are key concerns. Hence, for secure and privacy-friendly mobile applications, solutions have to be found that offer these qualities.

We envision a mobile authentication application, taking advantage of the features of mobile devices. Therefore, we seek for a non-intrusive but secure solution, which

[1] Source: GFK Retail & Technology (February 2012).
[2] Source: Gartner (February 2012).

can be deployed today, relying only on what is provided on commercially available devices.

Communication is required for many applications. Currently, high-bandwidth network connectivity is provided by telecom providers through, for instance, 3G connections with the Internet, or WLAN with a local network, and Bluetooth offering lower-bandwidth, short range communication. The latter is often used for connecting with devices such as car kits or wireless headsets. For short range communication, near field communication (NFC) is a more promising technology for simple transactions. Back in 2005, Nokia was one of the first, to introduce NFC on mobile devices. NFC enabled devices must be in close proximity, usually no more than a few centimeters, to establish a radio communication. However, the rise of NFC is not as expected, and only a few commercially available devices actually implement NFC. Hence, it cannot be used for mobile authentication today.

Therefore, we present an appealing solution using *visual communication*. It leverages the available standard hardware of web cams and displays, without the need to rely on the availability of NFC as a short-range channel. All mobile devices, but also terminals (e.g., workstations), have a display, often with high resolution, and most of them embody a high resolution camera. Visual communication can then be realized as a combination of two uni-directional channels. Retrieving data is achieved by scanning the information presented on a display and sending data is possible by displaying information, which is scanned by another device. Many formats are available today for visualizing information. A simple readable format, with wide-spread support is the Quick Response Code (QR). As one of the contributions in this chapter, we show how to apply QR codes to obtain a visual communication channel. Reusable components have been built, allowing a simple and quick introduction into new or existing applications.

Another requirement for many applications is security. Several technologies have been developed for building secure and privacy-friendly applications. However, application developers, willing to use these technologies, do not always have enough background on their correct use. Hence, bugs and vulnerabilities are easily introduced. To address this problem, we apply a *security and privacy framework* that takes care of the complex protocols, while keeping it transparent for developers. This framework, which is a stripped-down version of the `Priman` framework [VVL+10], offers the application developer a uniform interface to use, store and combine different types of privacy enhancing technologies. In addition, privacy aware policies, based on the CARL policy language [CMN+10], are used to specify server-side authentication requirements for relying parties.

The capacity and capabilities of current mobile devices, however, make them vulnerable to threats similar to the ones on traditional PCs. They will be more and more targeted by attackers, especially, since mobile devices store lots of personal

information. Keeping this information secure, is crucial. Moreover, in case of loss or theft, it is essential to prevent unauthorized use of private resources. Therefore, we provide an *extension to the `Identity Mixer`* library for anonymous authentication, in which the master secret is kept on a PIN protected smart card. All computations involving this master secret are performed on this smart card. Our solution provides a higher level of security against theft of the master secret than without the secure element, preventing impersonation of the user, and offers a higher performance than existing alternatives, such as a full deployment of the `Identity Mixer` anonymous credential system on a Java Card. Note that this solution gives no guarantees on the privacy protection of the user's attributes towards the host, as would be the case in solutions fully implemented on the card [BCGS09]. However, due to the complexity of anonymous credential systems (i.e., a simple proof on the card takes more than seven seconds), a full smart card implementation is not yet practical. Nevertheless, as the mobile may be seen as partially trusted, we do not require the smart card implementation to offer full anonymity.

In Sect. 6.2, we present our visual communication solution based on QR codes, followed by Sect. 6.3 in which we provide a privacy and security framework suitable for mobile environments. In Sect. 6.4, we implement an extension to the `Identity Mixer`, keeping the master secret secure, on a smart card. Finally, we end with some concluding remarks in Sect. 6.5.

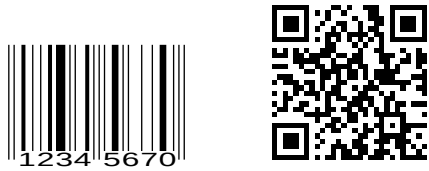
6.2 Visual Communication

6.2.1 QR

For visualizing information, we use Quick Response Codes [12] because of its broad adoption and desirable properties such as error correction and readability from different angles and rotations. Nevertheless, other formats may be suitable as well. As shown in Fig. 6.1(b), a Quick Response Code, short QR, is a two-dimensional symbology that can easily be interpreted by optical scanning equipment (like cameras). In contrast to barcodes (Fig. 6.1(a)), QR codes contain black and white squares (also called modules) in both the vertical and horizontal directions.

Hence, a QR code can contain considerably more information than a bar code. Whereas conventional bar codes can store a maximum of approximately 20 digital digits, up to 2953 bytes can be encoded in one QR symbol. Multiple data types (e.g., binary, numeric, alphanumeric), different symbol versions and several error correcting levels are supported.

A symbol version refers to the number of modules contained in a QR symbol, starting



(a) (b)

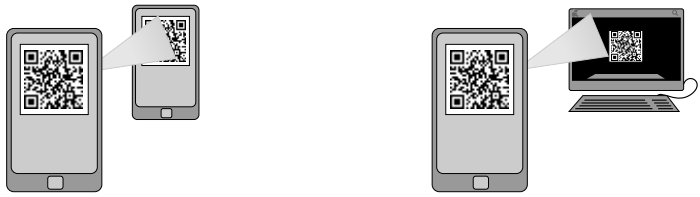
Figure 6.1: (a) a Bar Code, (b) a QR Code.

with version one (21×21 modules) up to version 40 (177×177 modules). Each higher version number comprises four additional modules per side. Four error correcting levels (L, M, Q and H) are defined with error correction capabilities of 7, 15, 25 and 30 percent, respectively. This feature is realized by using Reed-Solomon encoding. Level Q or H should be selected in factory environments where QR codes can get dirty whereas level L and M may be selected in clean environments where a large amount of data needs to be encoded. Raising the level improves error correction capabilities, but also increases the size of the data encoding. Software libraries that generate QR codes typically select a feasible version based on the amount of data, the data type and the selected error correcting level. Moreover, the user can define an upper bound to the version number (to allow for readings with low resolution scanners). If the data cannot be kept in a single QR symbol, the information may be split over multiple symbols.

6.2.2 Communication

We will use QR codes as a format for sending and receiving data through a bi-directional visual channel resulting from two uni-directional channels. As illustrated in Fig. 6.2, communication with a mobile or a desktop(the first uni-directional channel), is realized by displaying the data and having it scanned by the receiving party. The second uni-directional channel is simply the opposite: the other party presents the data on her display, which can be scanned to obtain the data. The QR code format is a standardized format, easy to scan and interpret. Moreover, the error correction covers possible transmission errors due to, for instance, a dirty display, scratches or reflections. For the communication towards a mobile device, the lowest error correction level is often sufficient for correcting a few errors. In that case, we can store up to about 3 kB of binary data in a single QR code. Compression techniques may possibly decrease the size of the data to be encoded. However, sometimes one QR code will not suffice for sending a message.

One solution is to present a sequence of QR codes and cycle through them. The speed of cycling will, of course, depend on the properties (e.g., resolution, size, speed) of



(a) (b)

Figure 6.2: Scanning QR codes (a) from a Mobile or (b) from a Desktop.

the display and the camera. However, this solution decreases the speed-of-scanning, making it less user-friendly. Another possibility is to combine a uni-directional QR-channel with a communication channel with higher capacity such as WLAN or even Bluetooth. In this case, the QR code must contain, among others, a reference to the verifier, allowing a secure out-of-band connection with the other party.

6.2.3 Implementation

We developed reusable components,[3] supporting the Android™ and Java™ environment, to act as both sender (i.e., generating and displaying QR codes) and receiver (i.e., scanning and parsing QR codes). Hence, a channel can be set up in both directions between mobile devices or between a PC and a mobile device. There is a common component, implementing general functionality, and platform specific components, featuring functionality specific to the platform. For instance, for scanning QR codes, access to the built-in camera on an Android device or connecting to a webcam attached to a desktop PC is implemented in a platform specific component.

These new communication components were added to the *Representational State Transfer* architecture (REST) [RR07], a resource-oriented architecture (in contrast to object oriented systems) provided by RESTlet [15], a lightweight and comprehensive open source REST framework for the Java™ platform. Fortunately, it also supports the Android™ environment. In order to make the Java™ based implementation as flexible as possible, we provided a Java™ Applet which can be used both as a stand-alone application or embedded as an applet in a web-page.

[3] We use the ZXing library [21] for scanning, parsing and generating QR codes.

6.3 Privacy & Security Framework

A lightweight version of the `Priman` framework [18] has been instantiated, with support for multiple credential technologies, such as anonymous credentials. It facilitates the development of privacy-enhanced applications and presents a uniform technology-agnostic interface making the complex security protocols transparent to the application developer [VVL+10]. The framework further assists the user in choosing the most appropriate technology. For instance, selective disclosure supporting technologies will likely be more preferred than standard technologies such as X.509 certificates.

As depicted in Fig. 6.3, it includes a credential manager, a persistence manager and a policy manager. In order to support multiple technologies, the managers delegate their requests towards technology specific handlers.

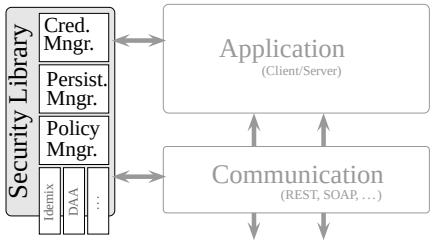


Figure 6.3: Privacy & Security Framework.

Similar to the QR code implementation, our framework is split into a platform independent part, common to all parties (i.e., issuer, prover, verifier), and a platform dependent part. This allows, for instance, the persistence manager to use platform specific functionalities for storing or loading objects, and for the credential manager to communicate with a Java Card embedded in a secure micro SD card.

In contrast to the `Priman` framework, the communication manager is omitted. Hence, communication is handled outside the framework. This offers a more flexible and easy integration of the privacy & security framework in existing communication frameworks. As an example, our framework is used together with the RESTlet [15] library. One drawback of leaving out the communication handler is that the framework cannot force the use of specific communication channels (i.e., anonymous communication channels) or make decisions depending on the type of channel used.

Below, we summarize the tasks of each manager and present some of the technologies that have been included in the prototype.

6.3.1 Credential & Persistance Manager

The credential manager handles both client and server side for the issuance of credentials and authentication. In other words, an entity using the credential manager can handle both proving and verifying. Similarly, the persistence manager handles the storage and loading of credentials.

Currently, the supported credential technologies are anonymous credentials, provided by IBM in the `Identity Mixer` library [11], and a DAA based credential on Java Card [3], provided by COSIC.[4]

We refer to Appendix A.2, for more information on the DAA credentials and the modifications applied to the `Identity Mixer` library in order to make them available on the Android™ platform and make our framework consistent, and less dependent on the technology and platform being used.

6.3.2 Policy Manager

Privacy preserving technologies, such as anonymous credentials, allow for privacy-friendly authentication. However, to make these technologies practical, an adequate policy language is needed, offering a way to specify the access control requirements of the service provider.

Currently, we support the CARL [CMN+10] policy language. CARL is a **C**ard-based **A**ccess control **R**equirements **L**anguage enabling privacy-preserving access control. The language offers adequate semantics to address advanced authentication, and allows for privacy-preserving, i.e., data minimizing statements while at the same time allowing for user accountability. The listing below shows a simple example policy specifying that the service may be consumed by a requester owning a Belgian eID certifying that the requester's age is over 18 and the eID is not expired; the gender needs to be revealed and a message *m* must be signed:

```
01 own p :: eID issued-by BEGov
02 reveal c.gender
03 sign m
04 where p.dateOfBirth ≤ dateMinusYears(today(), 18)
05       ∧ p.expDate > today()
```

The language is independent of the authentication technology, so suitable for our authentication framework presented above. In other words, the same specification

[4] **Co**mputer **S**ecurity and **I**ndustrial **C**ryptography Group, ESAT - KU Leuven

could be used for authenticating with the current Belgian eID, based on X.509 certificates, or with a more advanced credential technology. Of course, depending on the technology used, the authentication may provide better privacy and security properties.

A module has been implemented [1], for parsing and converting CARL policies.[5] The former is to parse policies received from the service provider, while the latter allows to convert these policies into a technology specific proof specification for the `Identity Mixer` library and the DAA credential library. For DAA credentials, however, the policy is restricted to the first and third clause of the example policy (01 & 03), specifying the type, issuer and message to be signed, since no attributes are available for proving statements about them. Note that the graphical user interface for displaying the policies to the user is not provided by the framework.

6.4 Secure Element

6.4.1 Description

Currently, the protocols in the `Identity Mixer` library require a lot of computational resources, making a full Java Card implementation impractical. Therefore, the protocols can only be run on the mobile device. However, credentials may get intercepted by malicious software running on the mobile and used elsewhere. To increase security, we propose an extension to the `Identity Mixer` library, in which only a small portion of the protocol is run on a tamper resistant module.

In our extension, the anonymous credential is bound to a smart card, by keeping the user's master secret on the secure element. That is, storing this key and performing all computations, that involve the master secret, on the card. In addition, the smart card requires a PIN code to be unlocked whenever (parts of) a proof protocol is to be executed.

The secure element adds substantial security to the overall system in that it prevents the user's secret key from being surreptitiously obtained by an attacker, unless he breaks the tamper resistance of the card. The user is protected by preventing illegitimate access by others to personalized services, as it requires the smart card and it also restrains the user from sharing the credential. Sharing the credentials requires to share the PIN code and access to the smart card.

[5] An implementation of a framework supporting CARL policies is available [1], however, the implementation has a tight coupling with the Eclipse Framework, which cannot be used on Android.

Requirements

Since a mobile device may be corrupted, we share the computations of the anonymous credential protocols between the possibly untrusted mobile device M, acting as a host, and a tamper resistant smart card (SC). We therefore extend the anonymous credential system, with the following requirements:

Security. The computations in the anonymous credential scheme involving the master secret are performed on the card, without leaking the master secret to the host.

Authentication. Invocations on the card, require proper access control. For each credential show, the card requires the user to authenticate, for instance, using his PIN or fingerprint.

Efficiency. Since a smart card is a chip with limited resources, the operations carried out on the card must be as simple as possible. Moreover, since communication with the card is slow, data transfer should be kept to a minimum.

Trust assumptions

T1 Authentication towards the smart card is assumed to be secure (e.g., no PIN caching).

T2 It is not possible to extract the master secret from the tamper resistant card through, for instance, side channel attacks.

6.4.2 Related Work

There have been several implementations of anonymous credentials on standard Java Cards presented in the literature.

Bichsel et al. [BCGS09] presented a full Java Card implementation of anonymous credentials similar to the ones used in the `Identity Mixer` library. Showing a credential takes about 7.4 s for a 1280-bit modulus, up to 16.5 s for a 1984-bit modulus. These results are promising, but still too slow for practical use. Moreover, this solution only allows to prove knowledge of a valid credential and optionally reveal a pseudonym which can be used for revocation verification. Adding support for multiple attributes and more extensive proofs such as range proofs, will make it even less practical.

Other implementations, require partial trust on the host. Balasch [Bal08] made an implementation of the DAA protocol, which was further enhanced by Sterckx et al. [SGPV09], taking about 4.2 s for a credential show using a 1024-bit modulus. Alpar

et al. [ABV12] present a solution in which anonymous credentials on a contactless smart card are used in combination with a mobile device. Dietrich [Die10] compared the runtime of the DAA protocol on more resourceful Java™ enabled devices ranging from 0.07s on a Lenovo T61 PC, up to 34.23s on a Nokia 6131 mobile device. Heupel [Heu10] ported and implemented the `Identity Mixer` library onto the Android environment. In contrast, for our solution and in cooperation with IBM, the `Identity Mixer` library was updated and made compatible with the Android environment, such that porting is no longer required. In addition, we also implemented a privacy friendly requirements policy (CARL) and run the protocols partially on a smart card.

Danes [Dan07] presented another approach, similar to ours, in which only the master secret is kept on the card. The author presents an estimation of 6s for a 2048-bit modulus. However, he did not take into account the limitations to the size of the exponents and the computational resources required for performing the modular multiplications on a standard Java Card. The large size of the modulus may require an additional separate exponentiation.

In contrast to the CL based schemes, there are also prototypes implementing `U-Prove` [19] anonymous credentials, which take about 5s for showing a credential. Later, Mostowski and Vullers [MV11], implemented the same protocol on a MULTOS [8] card with better support for modular arithmetic, resulting in only about 0.5s. This is an interesting result, and it may be future work to implement CL based anonymous credentials on a MULTOS card as well. Note that in order to remain unlinkable, the `U-Prove` system requires the issuance of a new credential for each transaction, which may quickly exhaust the EEPROM of the card [BCGS09].

6.4.3 Construction

In order to ensure that authentication without the smart card is infeasible, we start from the fact that knowledge of the attributes in the credential is required for showing a valid credential. Requiring the involvement of a card is then achieved by keeping the master secret, which is one of those attributes, on the card. Moreover, no information about the master secret may be leaked by the card. Computations using the other attributes and the CL signature are executed on the host.

Basic Idea

In 2007, Danes [Dan07] presented a construction to keep the master secret secure on a tamper resistant smart card. It is based on a proof of knowledge of a so called Damgård-Fujisaki-Okamoto commitment [DF02]. In fact, the smart card implements

the prover side of the protocol. Since protocols in the `Identity Mixer` library have been slightly changed, we refer to Appendix B, for an up-to-date specification of the smart card protocols and show their correctness.

However, in order to further increase efficiency of the smart card extension, we present a solution based on a proof of knowledge of a discrete logarithm in hidden order groups. Actually, we base our construction on the protocol for *Showing that a Discrete Logarithm lies in an Interval* [CM98], which was further improved for use in the `Identity Mixer` protocols (see Appendix C.6 of the `Identity Mixer` specification [11]).

Hence, in our *attack model* an adversary cannot extract the master secret from outputs generated by the card, being the results of the proof of knowledge for showing that a discrete logarithm lies in an interval, with the assumption that the discrete log problem holds in the hidden order group (i.e., it is hard to compute the master secret *ms* from $C_{ms} = h_1^{ms} \bmod n$).

Protocols

To simplify our construction, we assume that the card only knows one issuer with its corresponding public key, initialized during the activation of the card, and only a single credential is issued to the card. However, extending the protocols to support multiple credentials and issuers, even after the card was issued, is straightforward.

Our extension consists of the following algorithms running on the card:

Listing 1. Protocols running on the smart card

initCard(..) initializes the card with fixed system parameters l_m, l_n, l_ϕ and l_c. l_m defines the length of the master secret, l_n the size of the modulus, l_ϕ the statistical zero-knowledgeness, and l_c the length of the challenge. The master secret is chosen uniformly at random $ms \in_R [1, 2^{l_m}]$ and the required parts of the issuer's public key $pk_{\mathsf{IP}} = (n, g, h_1)$ is stored.

verifyPIN(..) verifies the PIN provided by the user and returns true in case of a correct PIN. After a fixed number of invalid tries, the card is blocked.

getCommon(..) returns $C_{ms} = h_1^{ms} \bmod n$.

getTValue(..) sets $r_{ms} \in_R \pm\{0,1\}^{l_m + l_\phi + l_c + 1}$ and returns $T_{ms} = h_1^{r_{ms}} \bmod n$.

getSValue(..) receives the challenge c and returns $s_{ms} = r_{ms} + c \cdot ms$.

When the card is first activated, initCard initializes the card by setting the system parameters, the master secret and the issuer's key. The master secret is generated on the card. The card ensures that the order in which the algorithms are invoked is fixed.

During a credential issuance, the correct protocol order is: verifyPIN, getCommon, getTValue, getSValue and getCommon. This last call is used to verify the correctness of the credential. Note that since its result is constant, the host could simply cash the result during the transaction. During a credential show, the algorithms are invoked in this order: verifyPIN, getTValue followed by getSValue. If the order is not respected, any of these fail, or if the card is removed from the reader, the entire sequence must be redone.

In the following, we only present the modifications in the `Identity Mixer` library, required to incorporate our smart card extension. So, extended proofs included in the library that do not use the master secret, remain unchanged. Moreover, the issuer and verifier protocols also remain the same.

Credential Issuance. In order to obtain a credential, the user first has to commit to the self-chosen and thus hidden attributes and prove knowledge of these. One of these attributes is the master secret *ms*, kept on the card. Without loss of generality, we assume that the attribute with index 1 is the master secret *ms*.

The proof of knowledge towards the issuer is denoted as follows:

$$\begin{aligned} PK\{(\{m_i : i \in A_h\}, v_c) : \\ U \equiv \pm g^{v_c} h_1^{ms} \cdot \prod_{j \in A_h \setminus \{1\}} h_j^{m_j} \bmod n \\ m_i \in \pm\{0,1\}^{l_m + l_\phi + l_c + 1} \forall i \in A_h \\ \}, \end{aligned} \tag{6.1}$$

with A_h the set of user chosen but hidden attribute indices.

When converting this to actual protocols, we first compute the commitment U, which is partially computed on the smart card by invoking getCommon. The host may then compute the commitment to the hidden attributes $U = C_{ms} \cdot g^{v_c} \prod_{j \in A_h \setminus \{1\})} h_j^{m_j} \bmod n$.

Then, in order to compute the proof of knowledge, getTValue is invoked on the card and the t-value T_U is computed as follows: $T_U = T_{ms} \cdot g^{r_{v_c}} \prod_{j \in A_h \setminus \{1\})} h_j^{r_j} \bmod n$, with $r_j \in \pm\{0,1\}^{l_m + l_\phi + l_c + 1}$ and $r_{v_c} \in \pm\{0,1\}^{l_n + 2l_\phi + l_c}$.

The host computes the challenge based on the Fiat-Shamir heuristic, sends it to the card and invokes getSValue, which returns the s-value related to the master secret on the card. The s-values for the remaining hidden attributes are computed locally.

Credential Show. During a credential show, multiple proofs of knowledge may be performed, such as knowledge of committed values and interval proofs. However, the

master secret is only involved in the proof of knowledge of a valid CL-signature. Note that we currently do not support pseudonyms, which also uses the master secret, but requires calculations in a prime order group. Hence, we only present what is changed in the CL proof of knowledge.

As in the original `Identity Mixer` protocol, in order to prove knowledge, the host first computes a randomized signature $(\tilde{A}, e, \tilde{v})$:

$$r_A \in_R \{0,1\}^{l_n + l_\phi} \tag{6.2}$$

$$\tilde{A} = A.g^{r_A} \bmod n \tag{6.3}$$

$$\tilde{v} = v - e \cdot r_A . \tag{6.4}$$

$$\tag{6.5}$$

The proof of knowledge of a valid CL signature is then given by formula 6.6:

$$\begin{aligned} PK\{(e, \{m_i : i \in A_h\}, v) : \\ & \frac{h}{\prod_{i \in A_r} h_i^{m_i}} \equiv \pm A^e \underbrace{h_1^{ms}} g^v \prod_{j \in A_h \setminus \{1\}} h_j^{m_j} \bmod n \\ & m_i \in \{0,1\}^{l_m + l_\phi + l_c + 2} \forall i \in A_h \\ & e - 2^{l_e - 1} \in \{0,1\}^{l'_e + l_\phi + l_c + 2} \\ \}, \end{aligned} \tag{6.6}$$

with A_h and A_r are the sets of hidden, resp. , revealed attribute indices.

To construct this proof of knowledge, the host first invokes verifyPIN, with the correct PIN, followed by invoking getTValue. As a result, the host receives T_{ms}. Now, the protocol proceeds by computing the commitment T_Z, which in the original protocol is computed as follows:

$$T_Z = \tilde{A}^{\tilde{e}} \cdot g^{\tilde{v}} \prod_{j \in A_h} h_j^{m_j} \bmod n . \tag{6.7}$$

However, since *ms* is unknown to the host, we re-order some computations resulting in:

$$T_Z = \tilde{A}^{\tilde{e}} \cdot T_{ms} \cdot g^{\tilde{v}} \prod_{j \in A_h \setminus \{1\}} h_j^{m_j} \bmod n . \tag{6.8}$$

On the host, the protocol proceeds as usual and after computing the challenge, the host invokes getSValue on the card, obtaining the s-value s_{ms}.

The show protocol further proceeds as would be the case without the smart card.

Proof. As mentioned before, the show protocol running on the smart card is actually the interactive proof for showing that a discrete logarithm lies in an interval [11]. Based on this and the fact that the discrete logarithm problem is hard, no information is leaked (i.e., information theoretically) to the service provider. For the proof, we refer to the `Identity Mixer` specification [11].

Note that in contrast to the proof of a discrete logarithm in a hidden order group (see [11] Appendix C.5), the length of the random r_{ms} is $l_m + l_\phi + l_c + 1$ bits,[6] while in the former, the length must be $l_n + l_\phi + l_c + 1$ bits, which is substantially larger (e.g., l_n = 1024-bit vs. l_m = 256-bit) and no longer corresponds to the requirements of the `Identity Mixer` library. To be correct, the host should, therefore, check the size of the s-value retrieved from the card and abort in case of failure. The verifier and issuer already perform this check, hence we do not need changes to their respective protocols.

6.4.4 Evaluation

Despite the latency in the computation of the protocols, our extension provides a higher level of security than running the protocols entirely on the host by keeping the master secret on a secure and tamper resistant device. Moreover, our construction only requires one exponentiation per credential show with an exponent of only $l_m + l_\phi + l_c + 1 = 256 + 80 + 256 + 1 = 593$ bits for a 2048-bit modulus. Moreover, showing a credential requires the user to authenticate to the card, using a correct PIN. However, other authentication mechanisms could be used.

Furthermore, with a proper issuing process being in place, (e.g., the issuer does not issue credentials that do not use the card), the relying party may be ensured that only credentials with the corresponding master secret contained in a secure element, are used in protocols. For instance, the card issuer could pre-install the credential on the card. This, as well as the PIN protection, increases the assurance for the relying party that the credential is indeed owned by the one making the proof.

However, we have to make clear that in contrast to a full Java Card extension, our solution does not offer anonymity towards the host. Hence, it should be used in combination with a 'trustworthy' host, such as a mobile device.

[6] Note the additional bit as instead of only positive values, we also allow negative values

6.5 Conclusion

In this chapter, we presented a set of building blocks that help the development of secure mobile applications. Short-range communication for mobile devices allows for a broad range of applications. However, since NFC, one of the principal technologies developed for short-range communication, has still not found its way to commercially available mobile devices, we present a simple short-range communication channel based on QR codes. This construction is directly applicable in most of the current mobile devices. Though not entirely new, it is a step towards already making anonymous credentials practical today, as we will demonstrate in the next chapter.

As a second contribution, we provide a simple security & privacy framework facilitating the development of secure and privacy-friendly authentication. It hides the technology specific intricacies from application developers, decreasing the number of implementation flaws due to insufficient security background. Three technology specific handlers have been provided, namely credential handlers for `Identity Mixer` anonymous credentials and DAA credentials, and a policy handler using CARL-based policies.

The last contribution is an extension to the `Identity Mixer` library. Since the protocols in the library are currently too complex and resource demanding to be fully implemented on a smart card, we have to run the protocols on the mobile device, requiring trust in the protection mechanisms of the operating system. However, this trust is not always justified. Therefore, we presented a solution in which the credential is bound to a smart card, preventing malicious software to simply copy the credential. A credential show requires possession of the card and knowledge of the correct PIN code.

In the following chapter, we will show how the building blocks presented in this chapter are combined to build a secure and privacy-preserving authentication application for mobile devices.

Chapter 7

Mobile Authentication towards a Terminal

7.1 Introduction

In this chapter, we combine the building blocks introduced in the previous chapter, into a prototype application. We show how mobile devices may facilitate the use of anonymous credentials, as an electronic identity, for data-minimizing authentication. We envision scenarios where users authenticate to terminals using their mobile, with the requirement that it is ensured that the user at the terminal is the one performing the authentication. We denote these scenarios *user-to-terminal authentications*.

Examples in the real world are: the age verification of customers or buyers in a bar when selling alcoholic beverages, access control to the premises of one's employer, and a broad variety of electronic ticketing solutions. For the age check, an important property is that no third party can perform the authentication instead of the user at the terminal.

We present the requirements (Sect. 7.2) and protocol constructions (Sect. 7.3), based on short-range QR channels between a user's handheld device and a terminal to establish an authenticated channel between the user's handheld and the terminal. As key property, the user is authenticated based on her properties instead of identifying attributes. We therefore use the `Identity Mixer` credential system on the mobile device. Through the properties of the short-range channels, we acquire a higher level of trust in that the device executing the protocol is the one interacting with the terminal, and vice versa. As a particularly important security feature, the prototype employs a

Secure microSD card as a secure element for handling secret cryptographic tokens throughout their life cycle. That is, storing them and performing all computations related to them. We have implemented a prototype system on top of an Android mobile phone. Implementation specific details can be found in Sect. 7.4. In Sect. 7.5, we present measurements of the key metrics that determine protocol runtimes. We found the results encouraging for a practical application of anonymous authentication technologies on standard mobile devices. We discuss the properties of our prototype; however, we refer to the next chapter for a more in depth evaluation. Finally, we conclude in Sect. 7.6.

Related Work. Chari et al. [CKST01] presented a taxonomy of different m-commerce scenarios with mobile devices. Our user-to-terminal scenarios fit in their *Kiosk-Centric* and *Full Connectivity* model respectively. Similarly, our solution also fits in the model of Claessens [Cla02], in which the mobile is combined with a workstation in order to have a higher degree of security and mobility.

Next to manual authentication [GN04] or authentication based on image comparison [PS99, DE02], many schemes have been presented that use physically constrained channels in order to obtain a secure communication channel between two nearby devices. Although they are often related to device pairing, they also apply to our user-to-terminal authentication. Examples are based on physical contact [SA00, BSSW02], motion [MG07], infrared [BSSW02], audio [GSS+06, STU08], bar- and QR codes [MPR09, LLSL09, SFG09], Bluetooth [JGK05] and radio profiling [VSLDL07].

In contrast to the schemes above, where privacy is often of a lesser concern (e.g., for device pairing), our solution combines the short-range communication channel with a more privacy-friendly authentication mechanism, namely anonymous credentials. In our prototype, we apply a QR based communication channel. Nevertheless, our solution also supports other short-range channels, as mentioned above.

7.2 Requirements

In Part I, we identified a number of requirements for an effective electronic identity. We will take those requirements as a starting point and will later reflect whether or not our construction fulfills these requirements. For the sake of clarity, we recall these requirements: strong authentication; user consent; user control; efficient revocation; controlled release of personal data; linkability; user-friendliness; and mobility.

In addition, we define the *proximity* requirement, in which the terminal is ensured that the device/user performing the authentication is indeed the one at the terminal, and vice versa.

Note that in this solution we do not prevent relay attacks such as *mafia fraud attacks* and *terrorist attacks* [BBD+91]. In a mafia fraud attack, the attacker forwards messages between an honest prover and an honest verifier. Whereas in a terrorist attack, the prover is dishonest and collaborates with the attacker. The latter could allow an adult, to help a youngster to prove that she is older than 18, based on the credentials of the adult. Mafia fraud attacks may be prevented using *distance bounding* [BC94, HK05, SP07, KAK+09, AT09, ABK+11], a mechanism mostly discussed in the context of RFID. Distance bounding in combination with the tamper resistance of the smart card also makes the terrorist attacks harder to achieve, as the prover has less control on the actual protocol running on his device. Unfortunately, in contrast to radio based channels such as NFC, when using a QR-based visual channel or an out-of-band channel, distance bounding is hard to achieve.

7.3 Construction

We now discuss the general architectural concepts used, as well as protocols for realizing our mobile authentication application.

Roles and setting. Figure 7.1 illustrates the parties in our setting. A user U is the holder of a mobile device M. An issuer IP provides credentials to the user U. U keeps the credentials on the mobile; they can later be used when U authenticates towards a terminal T. In order to allow a more flexible setup, with multiple terminals, verification is not performed at the terminal itself. The terminal is in direct connection with a trusted authorization server AS, who performs all verifications. In this case the terminal simply acts as gateway between the user and the authorization server. We denote the relying party RP as consisting of both T and AS. Note that communication between the terminal and the authorization server, is SSL/TLS protected, in order to obtain a higher level of security.

To address the *proximity* requirement, it is sufficient to have the first request sent through a short-range channel and cryptographically bind the response to that event, leaving open how the response is returned (e.g., using an out-of-band channel).

System Setup. The issuer IP publishes the public key information used in the anonymous credential-issue and credential-show protocols. To ensure entities that the public key is indeed the one of the trusted issuer, this information can be certified

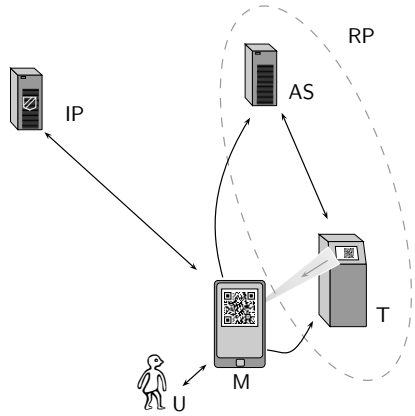


Figure 7.1: Parties in User-To-Terminal authentication. The User U, the Mobile M, the Issuer IP, the Authorization Server AS and the Terminal T. The Terminal T and Authorization Server AS together Form the Relying Party RP.

in a PKI-infrastructure, using standard X.509 certificates [2]. IP can then use this certificate for server authentication. Similarly, AS has a standard X.509 certificate for server authentication.

Setting Authentication Requirements. The authentication requirements are defined using the CARL policy language. Multiple policies may be defined for a single resource being protected. For instance, the user is allowed to prove, based on the date of birth included in a credential, that she has actually reached the age of majority, without revealing her date of birth. On the other hand, she is also allowed to prove possession of a valid driver's license, implying age majority. In addition, the relying party may also specify a *response channel*, and a destination address (e.g., URI or Bluetooth address), over which and to which the response should be returned.

Registration of U with IP. The user first registers with the registration server IP. The system can be bootstrapped by the user when getting issued an anonymous credential on M in a variety of ways. A guard protects the issuing of credentials at IP, specified in a policy on the requirements for obtaining a credential. Note that guards for multiple authentication schemes may be used. For instance, a practical use case for Belgium, where eID cards are issued to all citizens as of 12 years and older, is that the guard requires an attribute statement based on the user's eID card. The attributes obtained in such way can then be issued as attributes of an anonymous credential. IP, relying on the correctness of attributes originating from the Belgian eID card, then acts

as a (re-)certifier of these attributes. The guard-based and policy-based architecture of the issuer leaves the authentication of U with IP open to a concrete deployment of our system, thereby ensuring flexibility.

Authentication of U at T. Figure 7.2 illustrates how the user may authenticate towards the terminal, based on whether or not an out-of-bound channel is used. In both cases, the terminal receives an *authentication request* (linked to the HTTP session) from the authorization server (1), and hands it over to the mobile over the short-range channel (2).

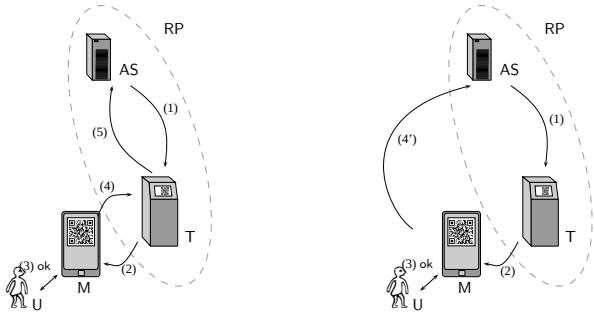


Figure 7.2: Route (a) over T to AS, and (b) Directly to AS.

M parses the received authentication request, containing a challenge and the server policies. It may also include the relying party's certificate to encrypt the response, if it is to be kept secret. The applicable policies are selected, based on the availability of credentials, and rendered by the application to a for humans easy-to-understand presentation format, which is then displayed on the screen of M. Each policy comprises a data request that needs to be disclosed and proved with the `Identity Mixer` or other technologies in our framework. U is required to choose how to fulfill the policy and to give her consent to the information release (3). If the master secret is protected by a secure element, the user is also challenged to enter her PIN code.

The privacy and security framework computes a credential proof and the *authentication response*, which includes this proof and a reference to the chosen policy, is returned to AS. Now, depending on the response channel, as specified in the authentication request, the response may follow a different path.

(a) From the user's perspective, the most straightforward scenario is to reply to the terminal from which the user received its authentication request (4) (see Fig. 7.2(a)). This could be over another or the same short-range channel as used in (2). Then,

the terminal forwards the authentication response to the authorization server over a secure channel (5). The latter parses the response and verifies the correctness of the included proof. Finally, the authorization server notifies the terminal about the result.

(b) In the second case, illustrated in Fig. 7.2(b), the reply is sent directly to the authorization server over an out-of-band channel (4'). Next, the server notifies the terminal (of the corresponding HTTP session) about the verification result.

7.4 Implementation

We have designed and implemented a prototype for validating our constructions and showing the practical feasibility of our ideas.[1]

Specifications. The mobile application was written in Java™ running on an Android mobile (i.e., Samsung Galaxy i9000: 1 GHz ARM Cortex-A8 with 512MB RAM, a 480x800 WVGA Super AMOLED screen and as 2592 x 1944 Camera) and the relying party was implemented as a Java™ EE web service combined with a GWT web application running on a desktop (i.e., DELL E4300: Intel Core2 Duo P9600 @2.54GHz with 4GB RAM running Windows 7(64) with a HD 1280x720 Webcam). We refer to Appendix A.1 for details on the implementation.

In addition, we present how, in the case of `Identity Mixer` anonymous credentials, the authentication is implemented using existing standards.

Standard Authentication Protocol. The authentication protocol based on anonymous credentials, is realized according to the HTTP/1.1 standard [7].

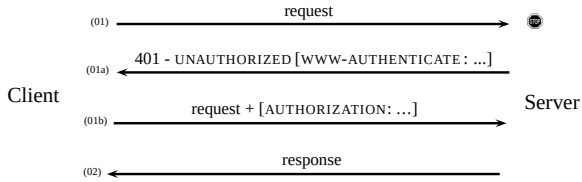


Figure 7.3: Flow of Messages in the HTTP Authentication Protocol.

Fig. 7.3 illustrates the message flow of the standard authentication protocol. In short, whenever a request is made (01) and client authentication is required, the server replies

[1] Lines Of Code: ≈19000 (incl. middleware, mobile app, web services, terminal website and widgets).

with a [401 UNAUTHORIZED] response (01a). The header of this response contains a [WWW-AUTHENTICATE] header field, in which it passes the *authentication info*, containing challenge information in order to allow proper authentication. If the client decides to authenticate, she includes an [AUTHORIZATION] header field into the new request (01b), containing the *proof claim*. The server may then parse the response and verify the proof.

In the case of authentication based on anonymous credentials, the [WWW-AUTHENTICATE] header field contains the policies accepted by the server, and a challenge, allowing the client to authenticate correctly. The client may then compute a proof based on a chosen policy. The proof and a reference to the chosen policy is included into the [AUTHORIZATION] header field.

For our mobile to terminal authentication application, there is a slight difference between the two scenarios (see Fig. 7.2(a)). In the first scenario (a), the terminal plays the role of the client. It therefore sends a request to AS, obtaining the [401 UNAUTHORIZED] response, and converts this into a QR code. In order to limit the size of the QR code, only the [WWW-AUTHENTICATE] header field (i.e., the authentication info) is converted into a QR code, which is displayed on the screen and should be scanned by the mobile device's camera. Next, the mobile displays the policy to the user, waits for the user's consent, computes the proof and displays its proof and choice in a QR code, which is then scanned by the terminal's webcam. This QR code is parsed by the terminal and its content is included into the [AUTHORIZATION] header field of a new HTTP request to AS. In this case, the terminal is immediately notified about the verification result.

In the second scenario (b), after having scanned the QR code displayed on the terminal, and processing the user interaction, the mobile itself includes the proof and choice into the [AUTHORIZATION] header field of an HTTP request and sends it directly to AS. Note that in order to get notified, T has to frequently poll AS for the verification result related to a specific authentication request.

7.5 Results and Analysis

Using the implementation of our prototype, a number of measurements were obtained. In this section we present and discuss these measurements. We looked at three different metrics that have a major effect on the overall system performance:

1. The execution time for a cryptographic `Identity Mixer` proof, that is, the time for constructing a proof and for verifying the proof;
2. The additional overhead introduced by the use of the secure element;

3. The encoding size of the `Identity Mixer` proof, which is relevant in the context of the bandwidth-limited QR channels we employ.

7.5.1 `Identity Mixer` Proof Execution Times

We present measurements for a spectrum of different variants of `Identity Mixer` proofs by using three sizes of the RSA modulus used for the protocol computations, credentials with zero and with three attributes, and proof specifications which hide or reveal all attributes, which need an inequality proof over one attribute, and a proof over one enumeration-type attribute. For these tests, the protocols are entirely computed on the mobile device (without a smart card). This leads to the proofs (a), (b), (c), (d), and (e) that are each constructed with the different modulus sizes 1024 bits, 1536 bits, and 2048 bits. We use the following encoding triplets as a shorthand notation for the structure (i.e., attributes and used features) of a proof: A_t, A_r, F with A_t the total number of integer attributes contained in the credential, A_r the number of revealed attributes, and F a feature to be proved or $\emptyset$ for no feature.

Proofs (a) and (b) perform a basic credential show in which only the fact that the user has a valid credential, is revealed. In case (a), the credential contains no attributes $(0,0,\emptyset)$, while in (b) it contains three $(3,0,\emptyset)$, none of them being disclosed. The proofs in (c), (d) and (e) use the same credential with 3 attributes as in (b), though, compute different proofs on them. In proof (c), all attributes are revealed $(3,3,\emptyset)$. In (d) and (e), the attributes remain hidden and additional proofs are performed in addition to the basic credential show, resulting in more complex cryptographic protocols.

Table 7.1 summarizes the median values we have measured for the prover and verifier side of the protocol and the overall runtime for the proof variants (a) - (e) with the used modulus sizes, without considering communication overhead, nor user-interaction. We may notice that for (c) the resulting timings closely resemble those of (a), which can be explained by how the cryptographic proof is computed as shown in [11] Sect. 6.2.3: *Revealed attributes are realized with modular exponentiations with rather small exponents corresponding to the actual attribute sizes and thus have no major influence on the overall protocol runtime.*

Besides the above mentioned cases, without attributes and with three attributes, more extensive tests have been run with different numbers of attributes, and they clearly show a linearity in the computational overhead with respect to the number of attributes in the credential. Overall, the figures show the expected dependencies of the proof runtime on the key size and the number of non-released attributes of the credential.

Proof (d) illustrates the overhead caused by an inequality proof $(3,0,\text{ineq})$. Proofs like this allow, for instance, to convince the verifier that the user's date of birth is more

than 18 years back in the past. For such predicate to be shown with the `Identity Mixer` library, a substantial protocol runtime overhead is incurred, which confirms the expectations deduced from the protocol specification [11].

Similarly, in proof (e) $(3,0,\text{enum})$, additional overhead results from the proof that an attribute of type enumeration (e.g., Drivers License Category $[A,B,EB]$) contains a specific value (e.g., B).

The modulus size of 2048 bits as recommended for high-security applications leads to a credential show having a runtime of at least about 0.9 s. Note that the computation time on the mobile phone is comparable to that on the server, although the prover side of the protocol needs to perform more computations [11]. This is an unexpected result as the CPU of the phone has a lower clock frequency and less RAM than the one of the PC we used. However, further investigations related to the implementation of the BigInteger class in the Android environment showed that it invokes native code, while on the PC, the class is entirely implemented in Java™. This explains the excellent performance of the mobile phone with respect to arbitrary precision arithmetic. The benchmarks of the petition application in Chapter 4 are worse because of a less efficient version of the `Identity Mixer` library.

Table 7.1: Timing Results (median over 100 runs), in Milliseconds, for Proving and Verifying a Credential Show with a Modulus of 1024, 1536 and 2048 bits

(ms)	1024			1536			2048		
(A_t,A_r,F)	prove	verify	total	prove	verify	total	prove	verify	total
(a) $0,0,\emptyset$	103	78	181	240	187	427	495	375	870
(b) $3,0,\emptyset$	139	125	264	323	265	588	634	515	1149
(c) $3,3,\emptyset$	102	78	180	243	187	430	495	375	870
(d) $3,0,\text{ineq}$	481	436	917	1182	1077	2259	2358	2184	4542
(e) $3,0,\text{enum}$	247	213	460	617	510	1127	1259	1014	2273

A_t: number of attributes in the credential
A_r: number of revealed attributes
F: feature to be proved

Configuration
Mobile (Java™, Android 2.3 - Dalvik Virtual Machine,
Samsung Galaxy i9000:1 GHz ARM Cortex-A8, 512MB RAM)
Desktop (Java™, Windows 7(64), J2SE 1.6 HotSpot Client,
DELL Latitude P9600 @ 2.53GHz with 4GB RAM

7.5.2 Overhead through the Secure Element

We have measured the overhead incurred during a credential show, because of the use of the optional Secure microSD card as a secure element for protecting the user's master secret key.

The figures in Table 7.2 show a substantial additional overhead compared to the timing results in Table 7.1. The overhead for each key length is fixed and independent of the proof specification. Moreover, it has no influence on the performance of the verification of the proof. As an example, a basic proof, case (a) with a 1024-bit modulus, now lasts about 1.44 s, of which about 0.18 s comes from the software and 1.26 s from the secure element fraction of the prover's protocol, compared to the 0.18 s without the secure element.

Compared to the full Java Card implementation of Bichsel et al. [BCGS09], taking 7.4 s, and the DAA implementation of Sterckx et al. [SGPV09] (which is partially run on the host) requiring 4.2 s, our protocol is substantially faster. Note that the DAA implementation provides protection against corrupted TPMs when issuing, which is not the case in our scheme in which we need a trusted setup of the card, during which a credential is issued to the card. In addition, the full card implementation offers enhanced privacy properties with regard to the host.

Table 7.2 also shows that a significant share of the overhead amounts to communication between the host and the secure element. This is partially explained by the current implementation requiring four rounds of communication. This can be reduced to only two cryptographic protocol requests of the `Identity Mixer` library, by combining the PIN verification and protocol selection rounds (i.e., issue or prove) with the first of those requests.

Note that for the 1024 and 1536-bit keys the delay due to the communication is the same, while for the 2048-bit modulus, the communication takes longer. The reason for this is that communication with the secure element happens in message blocks of 254 bytes. In case of 2048-bit keys, it does no longer fit into a single message block, resulting in an extra block and, hence, additional overhead.

7.5.3 Size of QR Codes

Proofs generated by the `Identity Mixer` library are formatted in XML, a verbose syntax. As the QR code-based channels are severely limited in bandwidth, and the `Identity Mixer` proof is the largest part of the content to be transferred back to the terminal, we created a customized space-efficient binary format for representing `Identity Mixer` proofs. The format is based on an ordered length-value encoding. Note that it is straightforward to replace the formatting with a more standardized

Table 7.2: Overhead, in Milliseconds, Incurred by the Secure Element, for a Modulus of 1024, 1536 and 2048 bits. The Overhead is Split Up into the Overhead Due To the Communication, and the Overhead Due To the Computation in the Secure Element.

(ms)	1024	1536	2048
build proof	1262	1606	2082
communication	310	310	375
computation	952	1296	1707

Configuration

Language :	Java™
OS :	Android 2.3 - Dalvik Virtual Machine
Libs :	MSC Smart Card Service 2.1.1
Mobile :	Samsung Galaxy i9000: 1 GHz ARM Cortex-A8, 512MB RAM
Secure Element :	Mobile Security Card SE 1.0 by G&D

formats such as ASN.1 [5]. Table 7.3, presents the average size of the authentication response, containing as its major part the proof generated on the mobile. In the table, the message size is decomposed into: the *proof* size, being the theoretical number of bytes of the `Identity Mixer` proof; the *header info* size, being additional information required to encode the `Identity Mixer` proof in our custom format such as attribute names and lengths of proof values; and the *response info* size, being additional information such as a reference to the chosen policy. Table 7.3 also shows that different proof specifications result in quite different proof sizes.

For the more complex proofs, such as proof (d), the size of the proof becomes too big to be encoded in a single display-readable QR code: as defined by the QR standard, only about three kilobytes of binary data may be included in a single QR code. To work along those constraints, one solution is to use the out-of-band channel and send the proof through the radio channel to AS. Another solution, also implemented, is splitting the message into multiple chunks and cycle through the resulting QR codes until the reader has scanned them all.

Note that the generation of the QR codes on the mobile device currently takes a substantial fraction of the overall protocol runtime. When showing a credential without attributes, the QR code is generated in about 0.8 s. For the case of an interval proof with a 2048-bit modulus size, it requires two (larger) QR codes generated in about 2.5 s.

Table 7.3: Average Size of the Authentication Response (in bytes), for a Modulus of 1024, 1536 and 2048 bits, for Proofs with Credentials without Attributes (a), with Three Attributes (b, d) and with an Inequality Proof (d). The Total Proof Size is Divided into the Theoretical Proof Size, the Size of Header Info (e.g., names of attributes) and Response Info (e.g., session information).

(bytes)	1024	1536	2048
(a) $0, 0, \emptyset$	793	878	1005
proof	589	675	802
header info	147	148	148
response info	57	56	56
(b) $3, 0, \emptyset$	1053	1138	1267
proof	811	897	1024
header info	186	185	187
response info	56	57	57
(d) $3, 0, ineq$	3243	4031	4855
proof	2867	3657	4488
header info	319	317	311
response info	57	57	56

Configuration
Language : Java™
OS : Android 2.3 - Dalvik Virtual Machine
Mobile : Samsung Galaxy i9000: 1 GHz ARM Cortex-A8, 512MB RAM

7.6 Discussion

For increased security and assurance, our system architecture and implementation comprises an optional secure element based on a Java Card microSD token. This achieves not only sharing prevention and theft protection for the user's master secret, but also a stronger binding between a user and her device through the PIN-based authentication. Those properties give the relying party a stronger assurance of the authenticating person having the claimed properties.

The short-range communication channel offers a higher level of security, in that the required *proximity* decreases the chance that another party is communicating with the terminal. However, the terminal is still not fully ensured about this, as the mobile could simply forward the messages to another mobile that performs the authentication instead.

Mobile devices are carried along most of the time, and therefore are an ideal

target as a deployment platform. Though, today's mobile devices suffer from vulnerabilities that may make the software-based computations or the I/O between the user and her device untrusted (e.g., captured or provided by a virus). Trusted Execution Environments [GRB03, GM07, SKK08, DW09] allow certain processes to be executed with a higher level of assurance, thereby ensuring that no malicious software can change computations or intercept the I/O of this process to the user. Developments on this are ongoing and can be employed as an orthogonal mechanism in our system architecture once they will be deployed on mainstream platforms.

7.7 Conclusion

We provide a solution to the authentication dilemma of users being required to identify themselves in most of their authentications today. We have brought anonymous credential systems to mobile devices as a privacy-preserving authentication solution and defined protocols for establishing secure channels between a user's mobile device and a terminal, based on short-range channels. This allows us to handle user-to-terminal authentication solutions through an easy-to-use system. While our protocol constructions apply to a range of short-range channels, we employ QR code technology to establish visual short-range channels in our prototype.

Our system is applicable to a wide range of practically-relevant authentication scenarios which users come across on a daily basis, ranging from user-to-terminal authentication such as age verification in a bar, over access to premises, to authentication to Web services from a home computer.

Future extensions on the protocol level may comprise including the introduction of the user accountability property [BCS05, CSZ06] through the use of verifiable encryption [CD00], or the support of credential revocation mechanisms, e.g., based on dynamic accumulators. Those features are not conceptually changing the constructions or architecture which are the main focus of this paper, but rather require some additions, like for key management.

With our implementation, we demonstrated the feasibility of the building blocks presented in previous chapter and obtained encouraging results regarding the protocol runtimes. In the following chapter, we assess our solution based on the requirements and findings on the Belgian eID previously discussed.

Chapter 8

Evaluation

The mobile authentication prototype illustrates that mobile anonymous authentication based on anonymous credentials is indeed feasible. In this chapter, we further evaluate this solution and compare it to the settings based on eID technologies as discussed in Chapter 5. We evaluate our new solution and compare it with the Belgian eID.

8.1 Requirements

We now verify our construction with the requirements we identified for future electronic identities.

8.1.1 Security

Strong Authentication. Our prototype uses anonymous credentials for authenticating to the relying party. Based on the security of the `Identity Mixer` anonymous credential system, our construction presents strong authentication.

User consent. When showing a credential, the mobile presents the possible alternatives for releasing personal information. The user has to select his choice and additionally, enter his PIN, if a secure element is used.

User control. Inherent to smart cards is that the user has no control of what is going on at the card. There is no trusted interface with the card. However, as the card is embedded in the user's mobile device, the latter may gain a higher level of trust, in contrast with the Belgian eID, which is also used with untrusted hosts (e.g., when identifying at a service desk). We will discuss these trust requirements further below.

Efficient Revocation. In contrast to the Belgian eID where revocation may be efficient, the `Identity Mixer` library does not provide a proper revocation mechanism. Moreover, anonymous credentials in general are lacking an efficient revocation scheme.

In Part III, we will make a pragmatic evaluation of several anonymous credential revocation scheme and strategies, in order to gain a better insight in which schemes are better suited for which settings.

8.1.2 Privacy

Controlled release of personal data & linkability. The unlinkability and selective disclosure properties of `Identity Mixer` anonymous credentials offer far better privacy properties than traditional eID technologies allowing the same credential to be used across multiple domains. In order to fully enjoy the anonymity features provided by these credentials, anonymous communication is required. However, even if no anonymized communication is used, anonymous credentials exhibit better privacy properties. For instance, less data is to be gathered by service providers in order to get sufficient guarantees about the user. Furthermore, linkability is not mandatory, but may be allowed depending on the particular scenario.

Nevertheless, without proper user control on what is being proved, its use makes no sense. Therefore, we use CARL policies to specify these requirements. In fact, these policies may also be used for other, less privacy preserving technologies such as the X.509 based eID cards, with the probable consequence of disclosing more than is required. Note that to be able to enjoy the full possibilities of anonymous credentials, the CARL policy language should be further extended to support (both global and domain-specific) pseudonyms and revocation.

8.1.3 User-friendliness

The prototype demonstrates that using anonymous credentials on a mobile device can be simple and easy-to-use. The latency caused due to the computations is acceptable.

However, for quick access control at, for instance, a terminal of the underground, the current implementation may be too slow, especially if a visual channel is to be used.

Compared to card-based solutions (e.g., the Belgian eID and banking cards), this solution may be less easily adopted, as it requires a more active role of the user.

8.1.4 Mobility

As mobile devices are personal and carried along almost everywhere, they are a possible target platform to support electronic transactions. In contrast, using card-based solutions such as the Belgian eID requires a card-reader, connected to a host, making it less mobile. On the other hand, the BeID proxy extension presented in Section 4.2 or the Belgian eID on a secure microSD [VDWDCD11] offers similar mobility properties.

Furthermore, the features present in mobile devices offer extended scenarios. For instance, in the case of an offline terminal, the terminal could use the mobile to get Internet access in order to obtain the latest revocation information.

Finally, maybe the most important drawback with respect to mobility is the battery lifetime. If the battery is exhausted, there is no way to authenticate. The card-based eID does not have this drawback.

8.2 Assessment of Possible Attacks

We evaluate the attacks that we are able to protect against and which attacks that are not addressed.

Lost or stolen device. If a device is lost or stolen, an adversary is not able to impersonate the user, as the user has to authenticate towards the smart card. In the case of a lost or stolen device, personal and credential information may be kept secret if it is stored on the smart card, and only retrieved from the card when it is required for creating a proof. After a transaction, the host deletes the personal information. We may even go further and keep certain attributes on the card, even during a credential show: since in `Identity Mixer` anonymous credentials, only the hash of text attributes is included in the credential, the text may be kept on the card, and only the hash is revealed to the host; only in credential shows releasing the text attribute, the card returns this value.

Note, however, that in order to make this properly work, it would require a construction such that the card may verify the requirement to release the info. This could be achieved by having the relying party authenticate towards the card, with a certificate stating the requirements. Note that, for the card, it is difficult to verify the revocation status of the server certificate. Nevertheless, even if this protection is not set up the personal information on the host is not certified by a trusted party (in contrast to the identity files on the Belgian eID), and hence, is less valuable.

Malicious host/middleware. The only attack that cannot be performed by malicious software is the extraction of the master secret from the smart card. Hence, credentials cannot be copied and showing a credential requires the possession of, or at least, communication with the smart card and the correct PIN code. As in the case of the Belgian eID, removing the malicious software and changing the PIN code is sufficient to securely use the card again.

Corrupt relying party. A malicious relying party cannot obtain more personal information than what is being proved. Showing a credential requires user consent on what is being proved. A possible attack is that a corrupt relying party acts as a man-in-the-middle to authenticate towards another relying party. As in the case of the Belgian eID, a proper mutual authentication mechanism may help to counter this attack [OHB06].

Corrupt prover. Only users with a valid credential of which the master secret is kept on the smart card, are able to properly authenticate. Attributes or properties thereof are provably correct.
Moreover, in addition to the *all-or-nothing non-transferability* [CL01], the construction prevents the user from sharing her credential with others. However, as also discussed in [Pap09], she could provide remote access to the card. In that case also the PIN must be shared or cached. The same holds for the Belgian eID, where sharing the card requires sharing the PIN. On the other hand, identification based solely on the identity files in the Belgian eID (i.e., without authentication), is insecure as those can simply be copied. This is not possible with our solution.

8.3 Summary of Threats and Issues

The application using anonymous credentials on a mobile device fulfills most of the requirements discussed in Sect. 5.1. Nevertheless, some issues remain. Moreover, although the user has more control over the host, since the secure element is

permanently available, the trust requirements in the host (i.e., the mobile device) may be larger than for the Belgian eID.

We now list the threats and issues unsolved in our solution:

Authentication/Identification.

(*H***)** Surreptitious authentication due to PIN caching.

(*H***)** A corrupt host could reveal attributes/personal information. However, in contrast to the Belgian eID, they are not certified, thus less valuable. Proving knowledge of those attributes requires proper authentication to the secure element.

(*H***)** The mobile device is in charge of parsing and enforcing the service policies, hence, there is no actual user control on the information being disclosed and trust in the host is required.

(*SP***)** The service provider may implement relay attacks [BBD+91] towards another service if no appropriate authentication mechanism is used.

(*C***)** Secure communication is not a requirement, hence, communication may be attacked.

Our prototype currently does not support digital signatures on arbitrary messages.

Our mobile authentication applications provides a solution that requires far less trust in the service provider, particularly concerning privacy.

Unfortunately, trust in the host is substantial: PIN caching, policy enforcement, verification of the service provider, etc. Since the host is a personal mobile device, trust may be higher, but on the other hand, the mobile becomes a single point of attack, and the user's credentials are permanently available. Moreover, the connectivity capabilities of mobile devices makes certain attacks more feasible. Hence, the advantages of using a personal mobile device may actually become a disadvantage.

8.4 Conclusion

We have shown that mobile devices can indeed feature anonymous credentials (i.e., `Identity Mixer` credentials), in order to protect our personal information. This solution exhibits a number of advantages with respect to the Belgian eID, most notably the privacy-preserving properties and its ubiquitous use. On the other hand, some problems, mostly inherent to the smart card environment (such as PIN caching), remain. Moreover embedding the smart card in a mobile device may make certain

settings less favorable and the permanent availability of the smart card makes it an interesting target for adversaries. In fact, mobile devices are already the target of plenty malicious applications.

As mentioned before, Trusted Execution Environments [GRB03, GM07, SKK08, DW09] may increase the trust in the mobile device. They allow certain processes to be executed with a higher level of assurance, thereby ensuring that no malicious software can change computations or intercept the I/O of this process with the user. Developments in this area are ongoing and can be employed as orthogonal mechanism in our system architecture once they will be deployed on mainstream platforms.

Currently, a more important drawback of anonymous credential systems in general, is the lack of a proper revocation strategy (e.g., in the `Identity Mixer` library). In order to have anonymous credentials to be really accountable, a proper revocation system must be put in place. Especially, in high-security environments, the verification of the revocation status must be based on up-to-date revocation information. Therefore, in the following part, we analyze and evaluate different revocation strategies that have been presented in the literature. We try to find an optimal solution to make our mobile anonymous authentication solution effective.

Part III

Revocation Strategies

Previously, we identified *credential revocation* as one of the major features still lacking in existing anonymous credential systems (e.g., `Identity Mixer`). Credential revocation is a crucial part for keeping the system secure. There are many reasons why revocation is desirable. The evident reason is to revoke authentication when the user's credential got stolen. However, the credential may also get lost, or it could simply get inaccessible due to a broken device on which it was stored. Even in those cases it may be appropriate to revoke the credential. Furthermore, in the case of misbehavior, it may be required that an authority can revoke some or all of the user's rights.

For instance, in Belgium in 2011, 1,020,220 bank cards were revoked:[1] 673,345 cards were (sometimes temporarily) blocked due to loss; 62,233 cards were revoked related to fraud, preventative or as a reaction to effective fraud; 179,559 cards were blocked due to theft; and the rest was revoked due to other reasons such as decease or bankruptcy. Note that these numbers cover multiple types of bank cards and multiple issuers. Hence, in case a wallet gets lost, multiple cards will get revoked at the same time.

In traditional credential systems, verifying the revocation status of a credential is straightforward and involves a simple lookup of a revealed credential specific identifier in a list. Well-known examples are *OCSP* [16] and *CRL* [2] based schemes. This strategy can be used for both *local* and *global* revocation. A revocation authority controls the validity of the credential globally, while services can use the identifier for local access control. We can distinguish two types of lists: *blacklists*, in which only revoked credentials are listed; and *whitelists*, in which only valid credentials are listed. Moreover, the time between credential revocations and a service still accepting the credential as valid (latency) can be limited.

In anonymous credential systems, on the other hand, this credential specific identifier may no longer be revealed to the verifier, since it would allow linking. In fact, unlinkability is one of the key requirements of anonymous credentials. Multiple revocation strategies have already been proposed in the literature, often with a theoretical security and performance analysis. However, a pragmatic assessment of revocation schemes for anonymous credentials is still lacking. Hence, it is very difficult to compare results due to varying security parameters and alternative software implementations. However, a critical and pragmatic comparison is crucial to bring those technologies to practice. Although some revocation mechanisms perform well for small groups, we focus on revocation schemes suitable for large scale settings such as electronic identity cards and e-passports. Particularly, efficiency in processing and communication is crucial.

In this part of this dissertation, we reflect on the revocation solutions presented in the literature. We therefore analyze both the functional properties (i.e., how is revocation achieved) and non-functional ones (e.g., cost, reliability and usability) of

[1] source: Atos – based on Card Stop-call center registrations in 2011.

the solutions that have been presented in the literature. As a validation, a prototype implementation was made for different strategies, suitable for the `Identity Mixer` anonymous credential system. In addition, we compare a number of accumulator based revocation strategies independent of the underlying credential system. Hence, also pairing-based solutions, which are not applicable to the `Identity Mixer` library, are compared. We further present some directions on how to realize revocation in order to enable large scale deployment of anonymous credential systems (i.e., as a nationwide eID).

In Chapter 9, we present an overview of revocation strategies for anonymous credential systems, discussed in the literature. We then present and analyze some revocation strategies we added to the `Identity Mixer` anonymous credential system in Chapter 10, while in Chapter 11, we do this for a number of accumulator based revocation strategies. Finally, in Chapter 12, we evaluate the revocation mechanisms available today and provide some future directions.

Contributions: This part presents the evaluations of anonymous credential systems resulting from two separate publications, both published in the peer-reviewed proceedings of international conferences. Chapter 10 presents the work evaluated in [LKDDN11] on revocation schemes suitable for the `Identity Mixer` credential system, while Chapter 11 presents the results of an analysis of accumulator based revocation strategies as they were evaluated in [LKDDN10].

Chapter 9

State of the Art

9.1 Introduction

In the literature, revocation mechanisms suitable for anonymous credential systems are mostly discussed in the context of group signatures. In a sense, group signatures are the non-interactive counterpart of anonymous credential systems, allowing members to sign a message in the name of the group, while preserving the signer's privacy. Proving that the signer's credential is not revoked should not break this. The same is true for anonymous credential systems. Hence, in our overview of anonymous credential revocation schemes, we will mostly talk about membership revocation in group signatures.

In 1999, Ateniese and Tsudik [AT99] were the first to present membership revocation as an open problem to group signatures. They already identified two important properties of membership revocation. The verifier must not learn anything but the fact that the signer is not a deleted member. On the other hand, in order to preserve anonymity, signatures must be backward unlinkable to keep past signatures unlinkable. As from then various solutions were proposed, each with its particular signature and CRL size, and costs for signing, proving and verifying.

Although the ultimate goal is to make the overhead caused by the revocation strategy as little as possible, most strategies assign a substantial workload to one of these parties. Moreover, for some strategies there may be an additional overhead for other parties as well. Based on this, we identify three classes (i.e., User, Verifier and Issuer) in the scheme. Fig. 9.1 shows an overview of the most important papers for the revocation of group signatures/anonymous credentials classified according to the classes defined above.

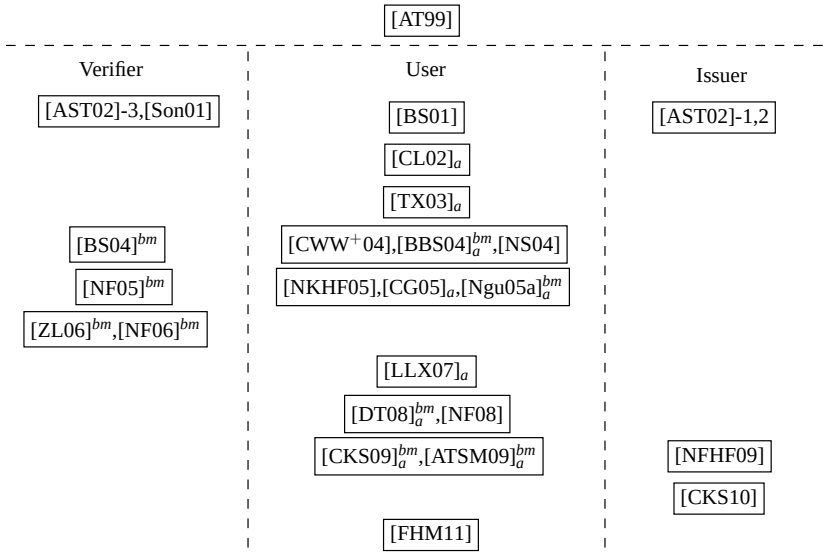


Figure 9.1: Literature Overview on Revocation Mechanisms Classified According to which Party (i.e., Verifier, User or Issuer) Gets the Most Overhead ($\blacksquare_a$: accumulator-based, $\blacksquare^{bm}$: pairing based).

9.2 Overview

We now present a comprehensive, but non-exhaustive overview of the most important revocation schemes based on this classification. Note that we are primarily interested in the construction of the revocation mechanism and focus less on the efficiency of the accompanying signature scheme.

9.2.1 User

Bresson and Stern [BS01] present a witness based solution in which the user has to prove that her membership key is not present in the revocation list. In order to achieve this, they require the user to prove non-membership in a revocation list for each item in the blacklist separately, making the signature grow linearly with the number of revoked members.

In 2002, Camenisch and Lysyanskaya [CL02] presented an important development in credential revocation. The authors put forth a new notion of *dynamic accumulators*, based on the accumulator scheme by Barić and Pfitzmann [BP97]. Their construction

allows the accumulation of a number of elements into one value, with a short size, independent of the number of elements accumulated. When computing a signature, the signer has to prove that her certificate is contained in the accumulator. It allows an efficient membership revocation scheme, with constant cost for signing and verification (i.e., not growing linearly with the number of revoked members). However, it requires the user to make a number of witness updates (i.e., exponentiations) linear in the number of revocations since the last witness update. Tsudik and Xu [TX03] propose a more efficient solution based on the accumulation of composites, however, its proof of security is based on the availability of a trusted third party and requires witness updates for both joining and leaving the group. Later, Camenisch and Groth [CG05] used the scheme of Camenisch and Lysyanskaya [CL02], in combination with a more efficient group signature scheme [CL03].

Along the lines of the accumulator scheme of Camenisch and Lysyanskaya, Boneh et al. [BBS04], presented a revocation scheme for a pairing-based group signature. Nguyen [Ngu05a] was the first to actually define a dynamic accumulator scheme using bilinear maps. Note that Zhang and Chen [ZC09] attacked this scheme. However, the attack was not practical and no longer worked after a change in definition of one of the security requirements [Ngu05b].

Later on, Li et al. [LLX07] extended the notion of dynamic accumulators into *universal dynamic accumulators*. Using this new type of accumulators, one can prove both membership or non-membership in the accumulator, supporting both white- and blacklists. This same notion has been implemented in the pairing-based setting by both Damgård and Triandopoulos [DT08], and Au et al. [ATSM09], based on the scheme by Nguyen [Ngu05b].

More recently, Camenisch et al. [CKS09] proposed an appealing pairing-based solution, improving the efficiency of the witness updates in accumulator-based revocation schemes. Contrary to earlier schemes that require a number of *exponentiations* linear in the number of revocations, this scheme only requires the more efficient *multiplications*, also linear in the number revocations. It therefore accumulates group elements instead of exponents. At a cost, however, of a much larger public key, the so called state-information. Based on this revocation scheme, with slight modifications and the group signature scheme of Boneh et al. [BBS04], Fan et al. [FHM11] recently proposed a new signature scheme with membership revocation.

Meanwhile, Nakanishi and Sugiyama [NS04] took another approach. In this approach, the i-th certificate contains an attribute m, with all bits set to 0, except for the i-th bit, which is set to 1. Then, the issuer issues a value $\tilde{m}$, with for each valid member j, the j-th bit set to 1, implementing a white list. To make a valid signature, the prover now has to prove that a bit specified by m in the certificate is 1 in $\tilde{m}$. To do this, the authors apply a range proof. This scheme, however, is only useful for reasonable-sized groups, (i.e., groups with a size comparable to the bitsize of the RSA modulus of the group

signature scheme).

This scheme is further extended by Nakanishi et al. [NKHF05], to make it suitable for large groups. The authors therefore apply a system of sub-groups with each a corresponding $\tilde{m}_k$. For each sub-group, the issuer generates a sub-group certificate. Later, when a user wants to prove membership, she selects the certificate of the sub-group she belongs to and proves knowledge of the certificate and that the bit in the prover's certificate is also set in $\tilde{m}_k$, without revealing anything else. In order to keep the number of recomputations of sub-group certificates small, especially in the case of large groups, they further propose a tree-based approach for making the sub-groups.

Chen et al. [CWW+04] claim to have a more efficient revocation scheme based on the proof of knowledge of co-primeness. However, as pointed out by Zhou et al. [ZL06], careful examination of the protocol shows that the computation complexity for generating a signature grows with the number of revocations.

Similar to the scheme of Chen et al. [CWW+04], Nakanishi et al. [NF08] present a scheme based on the product of prime numbers. Since the size of the prime numbers and consequently their product is smaller, the scheme is also more efficient. To safeguard efficiency even for larger groups, they apply sub-groups (as in [NKHF05]).

9.2.2 Verifier

Ateniese et al. [AST02], presented three revocation schemes of which the third scheme is the first *verifier local revocation* scheme (VLR). Informally, the user provably generates a randomized pseudonym allowing the verifier, to check for each item in the revocation list that there is no matching item. Therefore, the revocation authority selects a random base u and publishes a revocation list with for each revoked certificate $v_i = u^{e_i}$, with e_i the secret prime in the certificate. During a signature, the user verifiably reveals $T_r = g^r$ and $T_e = g^{r u^{e_j}}$ and the verifier then checks that $T_e \stackrel{?}{\neq} T_r^{v_i}$ for each item in the revocation list. If a match is found, the certificate was revoked. Similar to this third scheme, Song [Son01] presents two signature schemes with a revocation scheme based on such a revocation list.

A few years later, Boneh and Shacham [BS04] formalized the VLR scheme and presented a pairing-based version of it. Nakanishi and Funabiki [NF05] observed that this scheme was not backward unlinkable and presented a modified but backward unlinkable VLR revocation scheme, followed by two publications [ZL06, NF06] making improvements to the underlying group signature.

VLR has been applied in Direct Anonymous Attestation in the context of Trusted Computing [BCC04].

9.2.3 Issuer

The most naive solution for membership revocation requires the issuer to reissue new certificates after each revocation or after a short time period (epoch based). Hence, revocation has little or no impact on the signature size and complexity of the proof, but the issuer gets a lot of overhead on reissuing credentials. Although this idea may be naive, several interesting schemes have been presented based on the reissuance of certificates.

Ateniese et al. [AST02], presented two solutions based on the reissuance of certificates. The first schemes requires the issuer to issue, non-interactively, a new certificate to each member, while in the second scheme each user has to update his certificate individually, with a number of exponentiations growing linearly in the number of revocations.

In 2009, Nakanishi et al. [NFHF09] proposes a blacklist approach, in which the issuer computes a list of issuer-certificates, which can be used for proving non-revocation. To construct this blacklist, all identifiers of revoked credentials are sorted and pairwise certified (i.e., a signature on RID_i and RID_{i+1}) and the resulting list is published. For proving that the user's certificate is not revoked, the user fetches the public certificate from the list for which her credential's identity id, lays in the interval formed by the pair of revoked identifiers (i.e., $RID_i < id < RID_{i+1}$) and proves, next to the knowledge of this public certificate, that the identifier in her credential is strictly in the interval formed by the pair of revoked identifiers. Since the list is sorted, it ensures that only non-revoked credentials can successfully make this proof. Note that next to the burden for the issuer, the user gets a substantial workload due to the complex, tight inequality proofs.[1]

Recently, Camenisch et al. [CKS10] proposed a similar approach to the first scheme in Ateniese et al. [AST02], but now a more efficient group signature was used. Moreover, a more fine-grained revocation strategy may be introduced as certificate updates may address specific attributes in the certificate.

9.3 Conclusion

In the literature, various strategies have been suggested, trying to find an efficient and non-intrusive solution for the revocation of group signatures, and consequently, anonymous credentials. We categorized these strategies into three classes, and described for each its properties. In the following chapters, we will further analyze these classes and show the advantages and drawbacks for of each of these.

[1] In contrast to the more efficient but non-tight interval proofs as presented in [CFT98, BCDvdG06].

Chapter 10

Analysis of Revocation Strategies for Anonymous `Identity Mixer` Credentials

10.1 Introduction

As shown in the literature overview, multiple revocation strategies have already been proposed in the literature, often with a theoretical security and performance analysis. However, a pragmatic assessment of revocation schemes for anonymous credentials is still lacking. Hence, it is very difficult to compare results due to varying security parameters and alternative software implementations.

Based on the classification in previous chapter, we further break down the classes into 6 strategies. In this chapter, one variant of each strategy has been implemented with comparable security parameters and added to an existing library, namely the `Identity Mixer` library [11]. We give a detailed analysis and pragmatic evaluation of the implemented strategies. Amongst others, the security and anonymity properties, the connectivity and performance of the schemes are compared. Usable performance results are presented in the sense that all schemes were implemented within the same library and run on the same platform.

10.2 Revocation Strategies

Next to the classification presented in Chapter 9 (i.e., User, Verifier and Issuer), we add an extra class of revocation schemes. Namely, the class with Limited Overhead, in which none of the parties gets a big payload to handle revocation. In fact we will see that none of those solutions are satisfactory for anonymous credential revocation.

10.2.1 Limited Overhead

Pseudonymous Access [Nym]. Though, more related to service usage [BDDD07], a simple and efficient solution requires the owner to provably disclose a domain specific pseudonym [CMS10, BC10]. The service provider or a trusted party of that domain is in charge of creating and modifying the list of accepted or revoked pseudonyms. Although the domain specific pseudonym can be used for local access control, it cannot be used for a global revocation of the credential. Moreover, the user's transactions in the same domain are linkable.

Verifiable Encryption [VE]. Although verifiable encryption is often cited in anonymous credential schemes related to anonymity revocation [CS03, BCS05], it could be used for credential revocation as well. Hence, the user verifiably encrypts the credential's identifier with the public key of the issuer. To verify the revocation status, the service provider sends the ciphertext to the issuer, who decrypts the ciphertext. The issuer can now use the obtained identifier to do a simple lookup of the revocation status of the corresponding credential and report it to the service provider. This solution is closely related to the OCSP protocol in traditional credential schemes, with only little overhead. However, the user requires a lot of trust in the issuer, since it is able to monitor the usage of the credential (i.e., to which service providers the credential is shown). A possible solution is to require the service provider to make this request over an anonymous channel. Furthermore, replacing the public key of the issuer with the public key of another trusted third party, allows to have a separate authority in charge of the revocation tasks. Moreover, if the encrypted identifier is replaced with a domain specific pseudonym, a domain specific revocation authority may take care of the revocation status in a certain domain.

In spite of the Nym and VE strategies, a practical and privacy friendly revocation strategy with limited (constant) overhead is not yet available.

10.2.2 Issuer

In the most naive solution, both the group public key and the credentials of each user are reissued whenever a party is revoked or added to the group. This solution results in an unacceptable overhead for both users and issuers in large scale settings, hence, it is impractical. The Limited Lifetime and Signature Lists, discussed below, are two schemes requiring the issuer to frequently generate updates for users.

Limited Lifetime [LL]. In this scheme, an attribute expressing the lifetime of the credential, is enclosed. During each authentication, the user proves that the credential has not expired. The lifetime of a credential highly determines the usability of the revocation scheme. A short lifetime requires the user to frequently re-validate the credential, while a long lifetime makes the scheme insecure. Instead of reissuing new credentials, Camenisch et al. [CKS10] pointed out that non-interactive credential updates are a useful replacement. The issuer generates credential update info for all valid credentials before the end of the credential's lifetime is reached. Before the user can authenticate, the user has to download this information and update his credential.

Signature Lists [RL]. Similar to CRLs in traditional schemes, it is possible to maintain revocation lists in anonymous credential schemes. However, the verification is more complicated. Instead of the service provider performing the verification, the user has to prove that the credential is not revoked. In the case of whitelists, the list consists of signatures on the identifiers of each valid credential and a list's identifier. The user selects the signature in the whitelist containing the identifier of his credential and then proves knowledge of the identifier (without revealing the signature) together with the proof that the credential identifier in the signature is the same as the one contained in the credential being validated. Additionally, the list identifier is revealed, such that the service provider can verify that the latest list was used. Note that instead of the whitelist RL, it may be more efficient to simply reissue the user's credential. However, the signatures in the revocation list have no attributes, and can be kept simple such that their issuance is more efficient. Moreover, this strategy allows the revocation authority and the issuer to be distinct parties.

For blacklists, proving non-membership is more complex. Nakanishi et al. [NFHF09] propose an elegant solution by ordering the list of revoked identifiers. For each consecutive pair of identifiers, the issuer publishes a signature on the pair, together with an identifier of the list. During a credential show, the user then proves knowledge of his credential and a signature from the blacklist, such that the identifier in the credential lies between two revoked identifiers in the ordered blacklist without revealing any of the identifiers. Similar as in the case of whitelists, the disclosed list identifier shows that the latest revocation list was used. If this proof verifies

successfully, the service provider is ensured that the credential is valid with respect to the latest blacklist.

In the latter two schemes, the effort of the issuer is significant. For every change that requires the removal of a signature from a whitelist or addition to the blacklist, the issuer has to rebuild the entire revocation list with a new list of identifiers. On the other hand, in the case of an addition in the whitelist, it is sufficient to add only one signature to the latest whitelist. Likewise, removing a previously revoked credential from a blacklist can be done by replacing two consecutive signatures by one new signature. Nevertheless, authentication in both schemes proving (non-)membership results in a non-negligible, but constant overhead.

10.2.3 User

Accumulators [Acc]. A more complex, but possibly more efficient solution for credential revocation is based on so-called *dynamic accumulators* [CL02, Ngu05a, CKS09]. The user needs to prove membership or non-membership in the accumulator, during authentication for whitelist, resp. blacklist revocation. The service provider therefore fetches the latest accumulator value from the revocation authority. If the proof of the credential show verifies correctly w.r.t. that accumulator value, the service provider is ensured that the credential has not been revoked. Except for the verification of a more elaborate proof, the service provider has no additional overhead. On the other hand, although building this proof can be done quite efficiently, it requires the user to be online to first update its witness, which is time-consuming. The latter enables proving (non-)membership in the accumulator. Moreover, since revoking and possibly also adding credentials to the group change the value of the accumulator, a witness update is required. These updates require resources (e.g., exponentiations [CL02, Ngu05a], storage [CKS09]) linear in the number of added or revoked credentials from the accumulator.

10.2.4 Verifier

Verifier Local Revocation [VLR]. For many applications, the resources available to users to perform these witness updates, are very limited. In this case, verifier local revocation [BS04, AST02] first introduced by Ateniese et al. [AST02], may come to the rescue.

Service providers download a list of items each linked to a revoked credential. During authentication, the user provably reveals a token allowing the verifier to check that the token is not related to any of the items in the list. Therefore, as the service provider has to check each item in the list, verification takes a (maximum)

number of resources linear in the number of revoked credentials. Batch verification techniques [ZS10, BGR98] are referred to for making this check more efficient. Note that in some VLR schemes [AST02, BS04], all signatures ever made with the same credential become linkable after its revocation. Therefore, more recent schemes ensure *backward unlinkability* [NF05] such that former credential shows remain unlinkable.

This strategy has been adapted by the Trusted Computing Group for the use in trusted platform modules (TPM) [BCC04]. Note that in this case, revocation is only possible if the private key is revealed to the public. As long as the corrupted private key is kept secret by the adversary, revocation of the corrupted TPM is not possible.

10.3 Discussion

As we focus on strategies rather than on specific revocation schemes, the analysis of the strategies makes abstraction of scheme-specific details. Nevertheless, we do not hesitate to pinpoint the advantages of some specific schemes.

Complexity. All strategies try to tackle the same problem in a different way. For some strategies, the complexity analysis is obvious, in others it is rather subtle. Table 10.1 shows the complexity of the most expensive computations for each scheme. We assume that the average number of valid users ($\#\tilde{U}$), is constant. The table also illustrates the frequency of occurrence of these complex computations.

The table confirms the classification in Sect. 10.2. For both Nym and VE the workload is constant for every party. Further, the LL and RL strategies require the issuer to frequently compute updates, resp. , signatures for valid or revoked credentials. As mentioned before, updating the list in the RL strategies is not required as long as no identifiers are removed from the list. As opposed to LL, in which after each time-interval, the issuer computes for every valid credential a new credential update.

Accumulator based strategies (Acc), on the other hand, alleviate the work of the issuer by moving a part of the computation to the users. In fact accumulator updates can be done quite efficiently and in batch by the issuer (e.g., 1 multibase exponentiation in the case of [CL02]). However, now the user has to perform a number of complex computations (i.e., exponentiations in [CL02, Ngu05a]) linear in the number of added and removed credentials. The accumulator scheme by Camenisch et al. [CKS09] is in this sense quite efficient. Using the so-called state information, users can efficiently update their witness by a number of multiplications. However, in large scale settings, the amount of information required to perform the update is very large. Hence, special update servers are required to compute the updates efficiently, since they may keep the

Table 10.1: Total Complexity of the Most Computationally Intensive Processing During an Interval Δ.

	Complexity	Frequency	Description
Limited Overhead			
Nym	$O(1)$	–	
VE	$O(1)$	–	
Issuer			
LL	$O(\#\tilde{U})$	$\frac{1}{\Delta_t}$	creation of credential update info
RL_w	$O(\#\tilde{U})$	$\frac{1}{max(\Delta_t,\Delta_c)}$	creation of whitelist
RL_b	$O(\#R)$	$\frac{1}{max(\Delta_t,\Delta_c)}$	creation of blacklist
User			
Acc	$O(\Delta_R[+\Delta_J])$	$\frac{1}{\Delta_c}$	update of the user's witness
Verifier			
VLR	$O(\#R)$	each verify	checking the revocation list

$\#\tilde{U}$: average number of members $\#R$: number of revocations
Δ_R : revoked members (since last update) Δ_c: revocation/join interval
Δ_J : joined members (since last update) Δ_t: list update interval

state information in memory. To keep the number of changes of the accumulator in whitelist-based accumulators to a minimum, during setup the issuer can accumulate a large set of unused identifiers. Once the issuer issues a credential, it fetches a free identifier from the set and includes it in the credential. As such, the accumulator does not change whenever new users join the group. Instead of updating the accumulator after each addition or removal, it is possible to update the accumulator value only after a certain time, similar to the case of RL schemes. However, to increase flexibility and decrease latency, a list of the latest accumulators can be published, and allow the service provider to decide which accumulator values are still acceptable. Hence, the service provider may decide to accept proofs with older accumulators. Finally, often the issuer can perform the witness updates more efficiently [CL02]. However, in this case, the user is subject to timing attacks in cases the issuer and service provider collude.

Finally, in the VLR strategy, the verifier carries the burden. In the case of a valid credential, the verifier has to perform a computation for every item in the revocation list. There exist VLR schemes [DDD06] that improve efficiency of the verification; however, for large scale settings the complexity of the credential show and the memory load become significant. Batch verification techniques are sometimes mentioned to resolve this problem. Note that in the literature, there is no batch verification

scheme available that is tuned for the verification that a credential is not in the list of the VLR. For VLR, the batch verification should allow the verification of revocation lists (i.e., *none* of the tokens in the list match with the one being verified), while in the literature the authors often refer to batch verification of the validity of signatures [BGR98, ZS10], determining whether *all* signatures are valid.

Functional Properties. Table 10.2 gives an overview of some functional properties of the strategies with respect to the basic scheme without revocation. It illustrates that there is no straightforward winner. Schemes that score clearly better with certain properties, perform worse on others, and vice versa. For instance, it is clear that the Nym and VE strategies are less privacy friendly. In fact, all other strategies allow unlinkability. However, to obtain unlinkability in LL, RL and Acc, the user should download the entire set of update information, since otherwise timing attacks could allow to link a transaction with the download of user specific update information, making transactions linkable. Alternatively, a private information retrieval scheme may allow the user to download the required data more efficiently, while maintaining unlinkability. Of course, in large scale settings, with many service providers and users, and since the download may be done well before the actual credential show, the dangers of timing attacks may be negligible.

Table 10.2: Summary of Functional Properties for the Revocation Schemes Based on Pseudonyms Nym, Verifiable Encryption VE, Limited Lifetime LL, Revocation Lists RL, Accumulators Acc, and Verifier Local Revocation VLR (👎: worse than the basic credential scheme without revocation).

	Nym	VE	LL	RL	Acc	VLR
Linkability	👎	👎				
Latency			👎	👎		
Netw. Conn.			U	U (SP)	U (SP)	SP
Download (U/SP)	-/-	-/-	👎 / -	👎 / -	👎 / -	-/ 👎
Global/Local	L	$G[L]$	G	G	G	G

The latency for the LL and RL strategy makes them less secure w.r.t. the other schemes. Note that to decrease communication overhead, Acc and VLR can accept a non-zero latency, by accepting older accumulators, resp. revocation lists. To decrease latency in the case of LL and RL, the frequency of issuing update information, resp. revocation lists should be higher than the frequency of revoking credentials. This is computationally expensive, especially in the large scale settings that we envision. Nevertheless, both LL and RL can be useful in environments with lower security requirements.

VLR schemes use blacklisting. RL and Acc, on the other hand, allow for both black- and whitelisting. For RL schemes, while a proof of membership may be more efficient in the case of whitelists, some settings advocate for blacklist based schemes with possibly more efficient updates. Especially, the *valid to revoked credentials* ratio determines which strategy is the better choice. In the case of accumulator based revocation, the difference between white- and blacklists is rather subtle.

The table further shows that the user is required to be online for LL, RL and Acc. The service provider may have to download information for RL, Acc and VLR. However, for both RL and Acc it is possible to avoid downloads. In the case of RL, the service provider can simply verify the revealed validity time of the shown signature. If it lies in an acceptable (small) time interval, it accepts the credential show. Otherwise, it requires the use of a newer revocation list. In the case of Acc, the user could provide the signed accumulator to the service provider. Note that the amount of data to be downloaded by the user in the case of Acc and by the service provider in the case of VLR may be substantial. For some VLR schemes, such as the one of Ateniese et al. [AST02], to obtain high security the revocation list requires frequent updates, resulting in even more data traffic.

Combining strategies. As already discussed, the six schemes have different properties. To maximize the advantage of those properties, multiple strategies can be combined in the same credential scheme. For instance, an updatable lifetime may be used in parallel with accumulators. The lifetime may be sufficient in low-security environments, while a service requiring high-security may require the same user to prove membership in the latest accumulator. In another example, Nym could be used for local access control, while another strategy is used for verifying the global revocation status. In fact all strategies discussed are compatible and only require the issuer to include the appropriate attributes in the credential.

10.4 Implementation

10.4.1 Implementation Notes

One of the most versatile anonymous credential systems available to date is the `Identity Mixer` library [11]. Some of the schemes (i.e., LL, Nym and VE) are readily available in this Java™-based library. We extended the library with the other revocation strategies mentioned.[1] For RL and Acc both a white- and a blacklist scheme is implemented as well as a VLR scheme. More details are given below.

[1] Lines Of Code: ≈5250.

Note that our choice of schemes was restricted by the cryptographic schemes used in the `Identity Mixer` library. For instance, the library does not implement pairings, heavily limiting the number of possible schemes. Note that the implementation was done respecting the architecture and design of the library as much as possible. In fact, all extensions can be optionally activated depending on the proof specification. Most of the implementation effort went to the extended proofs of the credential shows. Except for the declaration and parsing of the appropriate attributes in the credential specifications, there are no major additions to the issuance of the credentials. An optional <*Revocation*> element has been added to the proof specification, in which <*VLR*>, <*Accu*> and <*RevocationList*> elements allow to declare the revocation scheme applied during the credential show.

Similar to the library-calls to the extensions for generating and verifying proofs (e.g., inequality and commitments), we added calls to revocation extensions (i.e., VLR-*Prover*/*Verifier*, Acc-*Prover*/*Verifier* and RL-*Prover*/*Verifier*), in the *Prover* and *Verifier* class. These handle the revocation scheme specific proofs. The credential shows in the `Identity Mixer` library are implemented as common three-move zero-knowledge protocols, made non-interactive using the Fiat-Shamir heuristic [FS87]. We refer to Sect. 2.2.3 for more details. The extensions use the security parameters used in the original `Identity Mixer` library for the construction of the proofs.

Signature Lists. The signature lists for both white- and blacklists are instantiated by CL signatures, which are also used in the library. They allow to prove knowledge of the signature and its attributes, without revealing them. Moreover, it allows to prove relations such as equality of the identifier in the signature and the identifier in the credential in the case of whitelists.
For blacklist revocation, an implementation was made based on the scheme of Nakanishi et al. [NFHF09]. As mentioned before, the revocation list consists of an ordered list of revoked identifiers, which are pairwise signed by the revocation authority together with a list identifier. Additionally, an unused minimum and maximum identifier is included in the list. For the implementation we recover the inequality provers available in the `Identity Mixer` library to build the interval proof as discussed in Sect. 10.2.2.

CL-Accumulator scheme. Several accumulator based revocation schemes exist. An implementation in C++, comparing three of them will be presented in the following chapter. The schemes implemented there, are all whitelist revocation schemes. One of the schemes compatible with the `Identity Mixer` library (in Java™) is the construction by Camenisch et al. [CL02]. Building on this construction Li et al. [LLX07] extended the scheme with a non-membership proof, allowing the same accumulator construction to be used for blacklisting as well. Both schemes have been

implemented based on the membership proof in Sect. 3.3 *"Efficient Proof That a Committed Value Was Accumulated"* presented in [CL02] and the non-membership proof defined in Protocol 1 in Sect. 5 *"Efficient Proof That a Committed Value Was Not Accumulated"* in [LLX07].

DAA-VLR scheme. Finally, the VLR scheme adopted by the TCG group [BCC04] has been implemented. In contrast to what is implemented in TPMs, in which the private key is required for revocation, a separate random identity attribute *id* is enclosed in the credential. The latter is then used to perform the verification. This allows the issuer to revoke the credential based on this identity, and does not require the private key of the credential to be compromised. The protocol presented in [BCC04] defines the issuance and proof of an entire DAA anonymous credential. Our implementation extends the credential show in the `Identity Mixer` library with the proof of knowledge $\mathrm{PK}\{(id) : N_v = \zeta^{id} \wedge \zeta \stackrel{?}{\in}_R \langle\gamma\rangle\}$ with *id* the identity of the user, and ζ a randomly chosen base. The verification of the list of revoked values is then achieved by verifying that $\zeta^{id_i} \neq N_v$ for each id_i in the revocation list.

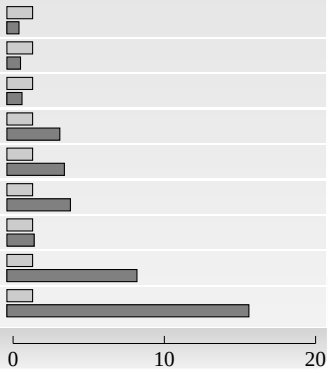
10.4.2 Experiments

This section reports the results of two experiments. The first experiment deals with the issuance and showing of a single credential. The second experiment analyzes the time required for the complex computations as in the complexity analysis (see Table 10.1). The experiments use the default security parameters (i.e., $k = 160$-bit) proposed in Appendix A, Table 2 of the `Identity Mixer` library specification [11], and are executed using the J2SE 1.6 HotSpot Client Virtual Machine on a DELL Latitude P9600 @ 2.53GHz with 4GB RAM. Note that since most algorithms are probabilistic, and random primes in specific ranges are computed, large variations in timings are possible. To make the measurements as realistic as possible and minimize overhead caused for instance by class loading, the given numbers are averages over a large number of runs. Moreover, the communication overhead is not included.

Table 10.3 presents for each implemented scheme, the total time required to issue and show a credential, which is independent of the number of revocations. The credential-show includes the verification of the revocation status. Since for all schemes issuing a credential does not require complex calculations w.r.t. the Basic scheme (i.e., a credential show without revocation), issuing a credential is about the same for most schemes. A small time difference may be noticed for all but the Nym scheme, caused by an additional attribute required by the revocation strategy. However, as could be expected, there is more variance in showing a credential. Only the time for a credential show in the Nym and VLR scheme lies close to the Basic scheme. For these schemes, the small overhead is caused by the computation and disclosure of a pseudonym. Note

Table 10.3: Timing Analysis (in ms) for Issuing and Showing a Single Credential (average over 200 rounds).

(in sec.)	Issue □	Show ■
Basic	2.5	0.8
Nym	2.5	0.9
VLR	2.6	1.0
Acc_b	2.6	3.5
Acc_w	2.6	3.8
LL	2.6	4.2
RL_w	2.6	1.8
RL_b	2.6	8.6
VE	2.6	16.0



Configuration

Language :	Java™
Virtual Machine :	J2SE 1.6 HotSpot Client
OS :	Windows 7(64)
Processor :	DELL Latitude P9600 @ 2.53GHz with 4GB RAM

that in VLR this pseudonym is randomized. For the whitelist based RL_w scheme, the time is doubled w.r.t. the Basic scheme. Here, showing a credential implies two proofs, namely one proof for proving the knowledge of a credential, and an additional proof for proving the knowledge of a signature from the revocation list, with the same identifier as in the credential. The overhead for the credential show in the white- and blacklist accumulator based schemes, is induced by the complex membership, resp. non-membership proof. A more detailed analysis may be found in the next chapter. It is a bit surprising that showing a credential in the LL scheme takes even more time. The reason for this is that the scheme (as implemented in the `Identity Mixer` library) requires an expensive range proof to show that the credential's expiration time, is larger than or equal to the current time. This way the epoch strategy is very flexible, as not all users have to update as frequently as others. However, if the lifetime attribute in credentials is synchronized and the same for all credentials, it is possible to simply disclose the lifetime value. As such, the credential show takes about as much time as in the case of the Basic scheme. Similarly, showing a credential in the RL_b scheme requires an additional signature proof and two range proofs. The signature proof, proves knowledge of a signature in the revocation list and the range proofs prove that the identifier in the credential lies between the revoked identifiers in the proved signature. The worst scheme is the one based on verifiable encryption. This scheme may not be practical for revocation. Moreover, this result shows that using verifiable encryption for *anonymity revocation* implies a very large overhead as well. Note that

Table 10.4: Time Analysis (in seconds) of the Most Complex Computations for the Implemented Revocation Schemes, Corresponding to Our Classification.

(sec.)	Issuer	User	Verifier
LL	$1.4 * \#\tilde{U}$		
$\mathsf{RL_w}$	$1.31 * \#\tilde{U}$		
$\mathsf{RL_b}$	$1.50 * \#R$		
$\mathsf{Acc_b}$	0.16	$0.02 * \Delta_R + 0 * \#J$	
$\mathsf{Acc_w}$	0.16	$0.02 * \Delta_R + 0 * \#J$	
VLR			$0.003 * \#R$

$\#\tilde{U}$: average number of members $\#R$: number of revocations
Δ_R : revoked members (since last update) $\#J$: number of joined members

Configuration

Language :	Java™
Virtual Machine :	J2SE 1.6 HotSpot Client
OS :	Windows 7(64)
Processor :	DELL Latitude P9600 @ 2.53GHz with 4GB RAM

the measurement of 16 s contains both the building and verifying the proof, but also the encryption and decryption.

In the second experiment, summarized in Table 10.4, the most complex computations as discussed in Sect. 10.3 have been verified in practice. Since the total amount of valid users, in our setting will be much larger than the number of revoked users, it is clear that LL and $\mathsf{RL_w}$ require a lot of computations by the issuer. Hence, $\mathsf{RL_b}$ might be more interesting. However, as noted in the previous experiment, showing a credential in the $\mathsf{RL_b}$ scheme is expensive, and may seem impractical. The accumulator based schemes have practically no overhead at the issuer's side. However, before showing his credential, a user has to update his witness. The witness update takes approximately 20 ms per revoked credential, since the previous update. As stated before, it is possible to avoid witness updates as a result of the joining of new credentials. If it is possible to let the user have frequent witness updates, then this overhead is spread over time and may be acceptable for some applications. Finally, the VLR solution shows that it only takes approximately 3 ms per revoked credential, to verify the validity of a credential. The VLR scheme could be practical if the number of revocations can be kept limited, for instance combined with LL, and the verification is optimized. The only drawback is that efficient VLR schemes often do not allow for backward unlinkability, heavily limiting their use.

For the Belgian eID card,[2] there are about ten million users, and about 375,000

[2]Results obtained from http://godot.be/eidgraphs.

revocations a year.[3] Applying the schemes to this large-scale setting, we have the following results.
Generating update information in the LL scheme would take about 160 days. For the RL_b scheme with 375,000 revocations, it takes about 6.5 days, while the accumulator based scheme takes about two hours. Similarly, for the VLR scheme, verifying a credential show takes 18 minutes.

Although great improvements can be reached by faster implementations and processors, these numbers show that for large scale settings, the RL, LL and VLR schemes are impractical. For the accumulator, an implementation in C++ of the accumulator takes for a single witness update only 1.5 ms (see next chapter), instead of 20 ms in Java™ in which the `Identity Mixer` library is implemented, resulting in a witness update of only about 10 minutes.

10.5 Conclusion

In this chapter, we classified existing revocation strategies for anonymous credential systems into six categories. The analysis shows that there is no straightforward winner, and the effectiveness and efficiency of a specific strategy heavily relies on the setting in which the mechanism is used. To maximize the applicability of anonymous credentials, only a combination of multiple strategies may provide some relieve.

Currently, for large-scale deployment, accumulator based revocation schemes provide relatively better features than the other schemes. Accumulators may be practical when using an improved revocation strategy (i.e., the number of revocations $<<$ 375,000), and possibly combined with witness updates performed by the issuer in case there are too many witness updates (e.g., no updates since a year).

In the implementation and comparison presented here, we focused on schemes that are suitable within the `Identity Mixer` library. In the following chapter, we analyze a number of accumulator based revocation schemes, without the restrictions put forth by the `Identity Mixer` library. For instance, revocation schemes based on pairing based cryptography, may be better alternatives.

[3] We have to note though that the certificates of youngsters and kids in Belgium are automatically revoked, giving an incorrect image of the number of actual revocations resulting from lost or stolen credentials. Moreover, Belgian citizens may opt to revoke their digital certificates themselves.

Chapter 11

Analysis of Accumulator-based Revocation Mechanisms

In the previous chapter, we analyzed a number of revocation schemes applicable in the `Identity Mixer` library. The result was that accumulator based schemes may be the only practical strategy for large-scale settings, offering the highest security (i.e., a minimal latency between revocation and accepting the revoked credential as genuine).

A *cryptographic accumulator*, first introduced by Benaloh and de Mare [BdM94], is a construction that allows the accumulation of a number of elements into one value. The size of this value is independent of the number of elements incorporated. For each accumulated element, there is a witness that allows to prove that the element is contained in the accumulator. It must be infeasible, for the adversary, to find a membership witness for an element that is not included in the accumulator. Camenisch and Lysyanskaya [CL02] further extended this notion to *dynamic* accumulators. In dynamic accumulators adding and removing values and updating individual witnesses can be done dynamically [CL02]. Finally, Li et al. [LLX07] defined the notion of *dynamic universal* accumulators, allowing for both proving that an element is, or is not accumulated. It must be computationally infeasible to find a membership witness for a value that was included in the accumulator or to find a non-membership witness for a value that was accumulated.

When applied to the revocation of anonymous credentials, a dynamic accumulator can be used as a *white-list*, accumulating only unrevoked credentials. Hence, proving that a credential was not revoked, requires a proof of membership. Similarly, *black-*

list revocation can be implemented using a universal accumulator, containing only revoked credentials. In this case proving a genuine credential requires a proof of non-membership.

In this chapter we evaluate and compare three accumulator schemes for the revocation of anonymous credentials based on white-listing: the scheme proposed by Camenisch and Lysyanskaya CL, [CL02]; the scheme due to Nguyen LN [Ngu05a]; and the construction due to Camenisch, Kohlweiss and Soriente CKS [CKS09]. We compare their computational and storage performance and discuss their suitability for massive deployment (e.g., in a national eID infrastructure).

11.1 Accumulator Schemes

This section briefly discusses the schemes in [CL02, Ngu05a, CKS09] (i.e., CL, LN and CKS) and summarizes their properties. We give a common interface for accumulator based revocation of anonymous credentials based on these systems. For a more detailed discussion, we refer to the original papers.

The common interface defines the protocols required for processing anonymous credentials with accumulator-based revocation. The schemes under evaluation all specify these protocols, hence, we did not modify the protocols in any major way. We do, however, implement a common book-keeping approach that deviates slightly from the one given in the referred papers. An archive table H records the history of the accumulator and allows to derive the list of added and revoked elements (L_a, resp. L_r) at a given time.

The entities participating in the protocols are: the registration server IP, responsible for the creation and revocation of credentials; the user U, the owner of an anonymous credential; and the verifier V. The verifier checks the revocation status of the user with the help of a zero knowledge proof (authenticate). In the schemes below, we use the notation presented in Sect. 2.1.

$\mathsf{IP} : (pk_{\mathsf{IP}}, acc, H = \emptyset, sk_{\mathsf{IP}}) \leftarrow \mathsf{setup}(1^k, N)$
is a probabilistic key generation algorithm that is executed by the issuer. It initializes the environment for the credential scheme for a given security level k. The second input is the capacity of the accumulator N, i.e., the maximum number of elements that can be accumulated. The public key pk_{IP} also fixes the set X of all elements that can be accumulated (with $|X| = N$). acc is the initial cryptographic accumulator. The history H, is initially empty.

$\mathsf{U} \leftrightarrows \mathsf{IP} : (acc', H'; cred_U; \text{-}) \leftarrow \mathsf{issueCred}(acc, H, pk_{\mathsf{IP}}; \text{-}; sk_{\mathsf{IP}})$
is a probabilistic interactive algorithm run by the issuer and a user. The issuer issues

the credential $cred_U$ to the user and adds the credential's identifier $id_C \in X$ to the accumulator acc. The credential includes witness information wit_C and a private key, that is unknown to the issuer. Only the issuer can add new credentials, as the secret key sk_{IP} is required for issuing. The new history $H' = H \cup \{\langle id_C, \text{“}\texttt{add}\text{”}\rangle\}$ is updated accordingly.

$\mathsf{IP} : (acc', H') \leftarrow \mathsf{revokeCred}(acc, H, pk_{\mathsf{IP}}, sk_{\mathsf{IP}}, id_C)$
is a probabilistic algorithm that is executed by the issuer to revoke the credential id_C. The new history becomes $H' = H \cup \{\langle id_C, \text{“}\texttt{delete}\text{”}\rangle\}$.

$\mathsf{U} : (wit_C{}') \leftarrow \mathsf{updateWit}(acc', H', pk_{\mathsf{IP}}, wit_C)$
is a deterministic algorithm, usually executed by the user, that updates the witness to correspond with the latest accumulator value acc'. However, as no secret data is required, this protocol can be performed by another, possibly untrusted, entity. The duration of witness updates depends on the number of elements added or revoked since the last witness update. The latter can be inferred from the book-keeping information H.

$\mathsf{U} : (boolean) \leftarrow \mathsf{verify}(acc, pk_{\mathsf{IP}}, id_C, wit_C)$
is a deterministic algorithm to verify that id_C is indeed accumulated in acc based on the up-to-date witness information wit_C.

$\mathsf{U} \rightarrow \mathsf{V} : (boolean) \leftarrow \mathsf{authenticate}(acc, pk_{\mathsf{IP}}, cred_U)$
is a two-party non-interactive zero-knowledge proof protocol that allows the user to prove to the verifier, that $cred_U$ is a valid credential (i.e., genuine and not revoked).

Next, we describe the construction of the accumulator, how to update a witness and how it is combined with a credential scheme for the CL, LN and CKS scheme. For the latter, the proofs of knowledge are all compiled into the notation introduced by Camenisch and Stadler [CS97].

11.1.1 CL Scheme

Camenisch and Lysyanskaya [CL02] were the first to introduce an accumulator scheme for the revocation of anonymous credentials. The scheme extends the collision-resistant accumulator defined by Baric and Pfitzman, based on the *strong RSA* assumption (see Definition 3 in Sect. 2.1.3), allowing dynamic updates of the accumulated set. The core of the accumulator uses a composite order group with an RSA modulus and is constructed as follows:

Accu :

$$acc = u^{\prod_i id_i} \tag{11.1}$$

Witness :

$$wit_{Ct} = wit_{C_{t-1}}^{b\cdot \prod_{id_i \in \Delta a} id_i} \cdot acc_t^a \quad \text{with } 1 = a \cdot id_C + b \cdot \prod_{id_i \in \Delta_r} id_i \tag{11.2}$$

verify :

$$acc \stackrel{?}{=} wit_C{}^{id_C}, . \tag{11.3}$$

with $u \in_R QR_n$, the group of quadratic residues modulo n, an RSA modulus; id_C the credential's accumulated value; wit_C the corresponding witness; and Δ_a and Δ_r the list of added, resp. revoked ids since previous update.

Equations (11.1) and (11.3) show the construction of the accumulator. It is clear that finding a witness for an *id* not accumulated, comes down to finding the *id*-th root of *acc* and is hard for a sufficiently large *id*, without knowledge of the factorization of *n*. Next, Eqn. (11.2) shows how to update a witness after a number of revocations Δ_r and additions Δ_a. It is clear for additions, that the time required for updating a witness grows linearly. In the case of revocations using the extended GCD algorithm, it is simple to compute a and b. However, $|a|$ will grow with a growing number of revocations, resulting again in a linear growth.

Finally, Eqn. (11.3) allows the user to verify that wit_C is the corresponding witness for id_C in accumulator *acc*. Nevertheless, when applying the accumulator for authentication, the latter proof will have to be performed in zero knowledge, without revealing the actual value of wit_C and id_C.

The authors applied the accumulator scheme to the identity escrow scheme due to Ateniese et al. [ACJT00]. Later on, the efficiency of the protocol has been further improved. One of these schemes, based on the so-called *SRSA*-CL-signatures [CL03], is used in the credential scheme proposed in [BCL06]. This scheme, that is also used in the `Identity Mixer` library, can be easily combined with the proof that a committed value has been accumulated, using Pedersen commitments. This proof was also mentioned in the paper [CL02]. For the Pederssen commitments, we use a multiplicative group $\mathbb{Z}_q^*$, with a large subgroup of prime order p (see Sect. 2.1). In the following, we will apply a combination, of the accumulator scheme with the CL-credential scheme, in which we integrate the accumulated value id_C as an attribute of the credential and release a commitment to the attribute. A credential is a signature from the issuer on the master secret and a credential identifier: $(\sigma, e, v, atts = \{ms, id_C\})$ with $g = \sigma^e h^v h_0^{ms} h_1^{id_C}$ and id_C chosen by the issuer from a predefined range.

The prover and verifier carry out the following signature of knowledge that a CL-credential is genuine and not revoked:

$$
\begin{aligned}
&SPK\{(\iota,\mu,\alpha,\kappa,\psi,\beta,\gamma,\varepsilon,\zeta,\eta,\delta):\\
&\quad g \equiv \pm C_\sigma^{\iota} h^{\kappa} h_0^{\mu} h_1^{\alpha} && \text{(11.4a)}\\
&\quad \wedge\ \mathfrak{C}_{id_C} = \mathfrak{g}^{\alpha}\mathfrak{h}^{\psi} \bmod q \ \wedge\ \mathfrak{g} = (\frac{\mathfrak{C}_{id_C}}{\mathfrak{g}})^{\gamma}\mathfrak{h}^{\psi} \bmod q && \text{(11.4b)}\\
&\quad \wedge\ \mu,\alpha \in \{0,1\}^{l_m+l_\phi+l_H+2} \ \wedge\ e - 2^{l_e-1} \in \{0,1\}^{l'_e+l_\phi+l_H+2} && \text{(11.4c)}\\
&\quad \wedge\ C_r = g_1^{\varepsilon} g_2^{\zeta} \ \wedge\ C_{id_C} = g_1^{\alpha} g_2^{\eta} \ \wedge\ acc = C_{wit}^{\alpha}(\frac{1}{g_1})^{\beta} && \text{(11.4d)}\\
&\quad \wedge\ 1 = C_r^{\alpha}(\frac{1}{g_1})^{\delta}(\frac{1}{g_2})^{\beta} \ \wedge\ \alpha \in [-B2^{k'+k''+2}, B2^{k'+k''+2}] && \text{(11.4e)}\\
&\quad \}(n_1),
\end{aligned}
$$

with public values $h \in_R QR_n$ and $g, g_1, g_2, h_0, h_1 \in_R \langle h \rangle$; $\mathfrak{g}, \mathfrak{h} \in_R \mathbb{Z}_q^*$ and commitments $\mathfrak{C}_{id_C} = \mathfrak{g}^{id_C}\mathfrak{h}^r \bmod q$; $C_\sigma = \sigma g_2^{r_0}$; $C_{id_C} = g_1^{id_C} g_2^{r_1}$; $C_{wit} = wit_C g_1^{r_2}$; $C_r = g_1^{r_2} g_2^{r_3}$ with $r_0, r_1, r_2, r_3 \in_R \mathbb{Z}_{\lfloor n/4 \rfloor}$ and $r \in_R \mathbb{Z}_p$

The proof (11.4a) proves knowledge of a valid credential, while (11.4b), proves knowledge of a commitment to the accumulated value id_C. It also proves that id_C is not equal to one and that the secrets are in the correct range (Eq. (11.4c)). The proofs in (11.4d) and (11.4e) prove that the committed value id_C has been accumulated in *acc*, and that the accumulated value is in the correct range. A step by step guide for implementing the above proof can be found in [CL02].

Note that a more efficient solution is to use the prime e, which is part of the CL-credential. In that case, the Pedersen commitment $\mathfrak{C}_{id_C}$ and proof that the prime is of the correct form can be left out, as this is already ensured during issuance. This solution, with a slight modification of the credential signature has been presented in [CG05]. Though the latter scheme will allow a more efficient proof of knowledge, the computationally expensive parts (i.e., witness updates) are the same for both schemes.

11.1.2 LN Scheme

Nguyen [Ngu05a], was the first to use bilinear maps to implement a dynamic accumulator for revocation. The security of the accumulator is based on the *q-SDH* assumption, with q an upper-bound on the number of elements to be accumulated. The

scheme employs a symmetric bilinear map $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ with $\mathbb{G}_1$ and $\mathbb{G}_T$ both groups of prime order p.[1]

Accu :

$$acc = u^{s \cdot \prod_{id_i} (id_i + sk_{r,a})} \tag{11.5}$$

Witness :

$$wit_{Ct} = wit_{Ct-1}^{id_i - id_C} acc_{t-1} \quad \text{(add)} \tag{11.6a}$$

$$wit_{Ct} = \left(\frac{wit_{Ct-1}}{acc_t} \right)^{1/(id_i - id_C)} \quad \text{(revoke)} \tag{11.6b}$$

verify :

$$\mathsf{e}(u, acc) \stackrel{?}{=} \mathsf{e}(pk_{r,a} u^{id_C}, wit_C), \tag{11.7}$$

with $u \in_R \mathbb{G}_1$, credential ID $id_C \in_R \mathbb{Z}_p$ and wit_C its corresponding witness, the issuer's accumulator secret $sk_{r,a} \in_R \mathbb{Z}_p$ with $pk_{r,a} = u^{sk_{r,a}}$ and random $s \in_R \mathbb{Z}_p$.

As in the previous scheme equations (11.5) and (11.7) show how to construct the accumulator and verify the correctness of a witness corresponding to a specific identity. Equations (11.6a) and (11.6b) show how to update the witness wit_C at a time t, after a single join, resp. revocation of id_i at a time $t-1$, without the knowledge of the issuer's secret $sk_{r,a}$. For multiple additions or revocations, these calculations are repeated iteratively. Note that this requires a clear bookkeeping of all accumulator values, witness values and id_is.

The credential scheme, used by the authors, is based on the signature scheme due to Boneh and Boyen [BBS04], resulting in a signature (σ, id_C, ms) with $\sigma = (h_0 h^{ms})^{1/(id_C + sk_{r,s})}$, h_0, h generators of $\mathbb{G}_1$, master secret $ms \in_R \mathbb{Z}_p$, accumulated value id_C and issuer's secret $sk_{r,s}$. This scheme is proved secure [NSN04] under the *q-SDH* assumption [BBS04].

[1] In the original paper, the group operations were expressed using the additive notation.

The proof of knowledge of a genuine unrevoked credentials is defined as follows:

$$
\begin{aligned}
&SPK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\phi,\psi,\eta):\\
&\quad E = g^{\eta} \wedge C_r = g_1^{\beta} g_2^{\gamma} {h_1}^{\delta} \wedge 1 = \frac{g_1^{\varepsilon} g_2^{\zeta} {h_1}^{\phi}}{C_r^{\psi}} && (11.8a)\\
&\quad \wedge \frac{\mathsf{e}(pk_{r,s}, C_\sigma)}{\mathsf{e}(h,h_0)} = \frac{\mathsf{e}(h,h)^{\alpha}\mathsf{e}(h,h_1)^{\varepsilon}\mathsf{e}(pk_{r,s},h_1)^{\beta}}{\mathsf{e}(h,\sigma_r)^{\psi}} && (11.8b)\\
&\quad \wedge \frac{\Lambda}{\mathsf{e}(h,C_\sigma)} = \frac{\Theta^{\eta}}{\mathsf{e}(h,h_1)^{\beta}} && (11.8c)\\
&\quad \wedge \frac{\mathsf{e}(pk_{r,a}, C_{wit})}{\mathsf{e}(u,acc)} = \frac{\mathsf{e}(u,h_1)^{\zeta}\mathsf{e}(pk_{r,a},h_1)^{\gamma}}{\mathsf{e}(u,C_{wit})^{\psi}} && (11.8d)\\
&\}(n_1),
\end{aligned}
$$

with public data $pk_{r,s} = h^{sk_{r,s}}$; $\Theta = \mathsf{e}(g,g)^{sk_{r,o}}$; $pk_{r,a} = \tilde{g}^{sk_{r,a}}$ and $g, g_1, g_2, u, h, h_0, h_1 \in_R \mathbb{G}_1$, commitments $C_\sigma = \sigma {h_1}^{r_1}$; $C_{wit} = wit_C {h_1}^{r_2}$; $C_r = g_1^{r_1} g_2^{r_2} {h_1}^{r_3}$; $E = g^r$; $\Lambda = \mathsf{e}(h,\sigma)\Theta^r$ with $r_i \in_R \mathbb{Z}_p$ and the issuer's signing secret $sk_{r,s}$, its opening secret $sk_{r,o}$ and accumulator secret $sk_{r,a}$.

Equations (11.8a) to (11.8c) prove knowledge of a valid credential, while (11.8d) proves that the credential has been accumulated into *acc*, hence, not revoked. Note that the proof of knowledge, as presented above, is the corrected version as in the full version of the paper [Ngu05b].

11.1.3 CKS Scheme

A more recent scheme implementing dynamic accumulators was proposed by Camenisch, Kohlweiss and Soriente [CKS09]. Similar to the LN-scheme, this scheme uses a bilinear map $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. However, the construction of the accumulator is different and is based on another assumption, the *n-DHE* assumption. Additionally, the *n-HSDHE* assumption is required for the proof that a hidden value is accumulated. The accumulator is constructed as follows:

Accu :

$$acc = \prod_{id_i} (g_{N+1-id_i}) \qquad (11.9)$$

Witness $:(wit_C, \sigma_C, U_C)$

$$\sigma_C = g^{1/(sk_{r,a}+\gamma^{id_C})} \tag{11.10a}$$

$$u_C = u^{\gamma^{id_C}} \tag{11.10b}$$

$$wit_{Ct} = wit_{Ct-1} \cdot \frac{\prod\limits_{id_i \in \Delta_a}^{id_i \neq id_C} (g_{N+1-id_i+id_C})}{\prod\limits_{id_i \in \Delta_r}^{id_i \neq id_C} (g_{N+1-id_i+id_C})} \tag{11.10c}$$

verify :

$$z \stackrel{?}{=} \frac{\mathsf{e}(g_{id_C}, acc)}{\mathsf{e}(g, wit_C)} \quad \wedge \quad \mathsf{e}(pk_{r,a} g_{id_C}, \sigma_C) \stackrel{?}{=} \mathsf{e}(g,g)\,, \tag{11.11}$$

with $id_i \in [1..N]$, g a generator of the group $\mathbb{G}_1$, N, the capacity of the accumulator such that $X = [g_1 = g^{\gamma^1}, \ldots, g_N = g^{\gamma^N}]$, state information $[g_1, \ldots, g_N, g_{N+2}, \ldots, g_{2N}]$, issuer's secret $sk_{r,a}$ and corresponding public key $pk_{r,a} = g^{sk_{r,a}}$ and the sets Δ_a and Δ_r of accumulated, resp. revoked values.

Here, contrary to the schemes above, the elements accumulated are group elements and the accumulator is a product of those (11.9). Updating the witness, based on state information, only requires a number of multiplications (11.10c). A property of accumulators is that it is infeasible to compute a witness for an element not accumulated. Therefore, the scheme uses a signature σ_C. Hence, to compute a witness for a revoked element, the adversary would need to compute a signature forgery.

The credential scheme presented in the paper originates from the same Boneh and Boyen signatures as the LN scheme, that was further modified by Camenisch et al. [CL04] for the issuance of anonymous credentials and proven secure in [ASM06] under the *q-SDH* assumption. We obtain a credential (id_C, σ, c, ms) with $\sigma = (g_{id_C} h_0 h_1^{ms})^{1/(c+sk_{r,s})}$ with h_0, h_1 generators of $\mathbb{G}_1$, master secret ms, issuer's secret $sk_{r,s}$, random number c, and the accumulated value g_{id_C}.

To prove knowledge of a valid unrevoked credential the following signature proof of knowledge is performed:

$$
\begin{aligned}
&SPK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\phi,\psi,\eta,\rho,\chi,\omega,\iota):\\
&\quad C_r = h^{\alpha}\tilde{h}^{\varepsilon} \ \wedge\ 1 = C_r^{\gamma}h^{-\zeta}\tilde{h}^{-\phi} && \text{(11.12a)}\\
&\quad \wedge\ \frac{\mathsf{e}(h_0 C_i, h)}{\mathsf{e}(C_\sigma, pk_{r,s})} = \mathsf{e}(C_\sigma,h)^{\gamma}\mathsf{e}(\tilde{h},h)^{\psi}\mathsf{e}(\tilde{h},pk_{r,s})^{-\alpha}\mathsf{e}(\tilde{h},h)^{-\zeta}\mathsf{e}(h_1,h)^{-\eta} && \text{(11.12b)}\\
&\quad \wedge\ \frac{\mathsf{e}(C_i, acc)}{\mathsf{e}(g, C_{wit})z} = \mathsf{e}(\tilde{h},acc)^{\psi}\mathsf{e}(1/g,\tilde{h})^{\beta} && \text{(11.12c)}\\
&\quad \wedge\ C_{r_1} = g^{\psi}\tilde{h}^{\omega} \ \wedge\ 1 = C_{r_1}^{\gamma}g^{-\chi}\tilde{h}^{-\iota} && \text{(11.12d)}\\
&\quad \wedge\ \frac{\mathsf{e}(C_i, C_{\sigma_C})}{\mathsf{e}(g,g)} = \mathsf{e}(pk_{r,a}C_i,\tilde{h})^{\rho}\mathsf{e}(\tilde{h},\tilde{h})^{-\chi}\mathsf{e}(\tilde{h},C_{\sigma_C})^{\psi} && \text{(11.12e)}\\
&\quad \wedge\ \frac{\mathsf{e}(h_0 C_i, h)}{\mathsf{e}(g, C_{s_x})} = \mathsf{e}(\tilde{h},h)^{\psi}\mathsf{e}(1/g,\tilde{h})^{\delta} && \text{(11.12f)}\\
&\quad \}(n_1),
\end{aligned}
$$

with $h,\tilde{h} \in_R \mathbb{G}_1$; $C_r = h^r\tilde{h}^{open}$; $C_{r_1} = G^{r_1}h^{open'}$; $C_\sigma = \sigma\tilde{h}^r$; $C_i = G_i\tilde{h}^{r_1}$; $C_{wit} = wit_C\tilde{h}^{r'}$; $C_{\sigma_C} = \sigma_C\tilde{h}^{r''}$; $C_{U_C} = U_C\tilde{h}^{r'''}$ and $r, r_1, open, open', r', r'', r''' \in_R \mathbb{Z}_q$

Equations (11.12a) and (11.12b) prove knowledge of a genuine credential, while (11.12c) down to (11.12f) extends the proof with a proof that the credential has been accumulated, thus, not revoked. In fact, the latter equations implement a proof of knowledge of the signature σ_C

11.2 Implementation

Various protocols are used to prove knowledge of a valid credential. Most of the papers use the notation introduced by Camenisch and Stadler [CS97]. Nevertheless, the implementation of these schemes was not straightforward. We had to deal with many details and small differences: e.g., some schemes use a group of known order, others of hidden order; interactive versus non-interactive proofs of knowledge; the length of random values and nonces. In the implementation, the proofs of knowledge were made non-interactive using the SHA-2 hash function. Interactive proofs can be converted to non-interactive ones, using the Fiat Shamir heuristic [FS87].

For the anonymous credential schemes, only the required minimal set of attributes are added to the credentials (i.e., the master secret). Thus, the overhead in storage and computation, resulting from additional attributes embedded in the anonymous credential, is not reflected in the results. Likewise, for interactive protocols, the communication overhead is not considered.

To compare the schemes discussed above, they are all implemented in C++.[2] We refer to Appendix A.3 for more details on the implementation and configuration.

11.3 Results

This section reports the results of three experiments. The storage analysis deals with the size of key-material in the scheme. The computational complexity analysis illustrates the complexity of the protocols and the timing analysis validates the results of the complexity analysis by measuring the actual protocols.

11.3.1 Storage Analysis

For each of the implementations, Table 11.1 summarizes the bit-sizes of the private and public key of the issuer, one credential, and the accumulator. Additionally, the size of one accumulated value is listed. Pairings generally allow better results with respect to the size of cryptographic keys than other schemes. This is reflected in the paper of Nguyen [Ngu05a]. However, as can be seen in Table 11.1, the difference is less extreme than the paper suggests. Since the PBC Library does not provide the pairing proposed in Nguyen's paper, another type of pairing was used, resulting in a larger subgroup $\mathbb{G}_1$.

A more important observation is the that public key of the issuer (pk_{IP}) in the CKS scheme contains state information that depends on the capacity of the accumulator. Even if this information can be omitted for most of the protocols, it is required to make witness updates. This will have an impact on how this scheme is used in practice. In the case of massive deployment, for instance, witness updates will require special purpose update servers.

Finally, the elements accumulated in the CL and LN scheme are exponents, while in the CKS scheme they are group elements. This has an impact on the implementation of the credential scheme as it makes the proof of credential ownership more expensive for the CKS scheme, as the credential needs to be extended to bind this group element to the other credential attributes.

[2] Lines Of Code: ≈7500.

Table 11.1: Bit-sizes of the Private and Public Key of the Issuer, the User's Credential, the Accumulator and a Single Element for the Three Accumulator Schemes.

	sk_{IP}	pk_{IP}	$cred_U$	acc	id_C
CL	l_n	$7l_n + l_\rho + 3l_\gamma$	$2l_n + k + l_v + l_\rho$	l_n	l_{id}
	2048	19,730	7719	2048	160
LN	$3l_r$	$16l_q + l_r$	$6l_q + 2l_r$	$2l_q$	l_r
	576	16,576	6528	2048	192
CKS	$3l_r$	$(16 + 4N_t)l_q + l_r$	$10l_q + 2l_r$	$2l_q$	$2l_q$
	576	16,576+4096N_t	10,624	2048	2048

N_t: maximum number of elements accumulated
l_n: size of the RSA modulus (2048)
l_γ: size of the commitment group modulus (1632)
l_q: size of the field used for the pairing (2048)
k: security parameter (160)
l_ρ: size of prime order subgroup (498)
l_v: size of v values in the certificate (2965)
l_r: order of the pairing (192)

11.3.2 Computational Complexity Analysis

Table 11.2 presents the most computationally expensive operations. As shown in the table, the complexity of the *witness update* protocol significantly differs for the three schemes. As each call of authenticate requires an up-to-date witness, the efficiency of witness updates is very important.

The CL scheme only requires one exponentiation for newly accumulated elements, and one for newly revoked elements. However, as the size of the exponents is growing linearly with the number of accumulated, respectively revoked elements (i.e., $N_a \cdot l_{id_i}$, resp. $N_r \cdot l_{id_i}$), the performance decreases considerably (see Timing Analysis). The LN scheme, on the other hand, requires an exponentiation for a base in $\mathbb{G}_1$ for every element accumulated (N_a) or revoked (N_r), since the last witness update. Updating a witness is more efficient in the CKS scheme, as the most expensive operations are a number of multiplications linear in the number of accumulated and revoked elements. Moreover, the scheme requires less expensive operations during the issuance of the credential. However, proving knowledge of a valid credential requires slightly more exponentiations and pairings than in the other schemes. This is because id_C is a group element. The credential proof of possession needs to be extended to show that this group element is bound to the other credential attributes. Finally, the table reveals that optimizations of the authenticate protocol are possible, especially in the LN scheme, in which twelve pairing operations can be precomputed as they do not alter during the lifetime of the credential. Unfortunately, this requires more storage space. Thus, a balance must be found between storage space and processing efficiency.

Table 11.2: The Most Expensive Operations (i.e., exponentiations, pairings, multiplications) for the Protocols in the Credential Scheme, with Δ_t the Number of Accumulated and Revoked Values. The Numbers Between Brackets Denote Operations That Can Be Precalculated.

	CL	LN			CKS			
	$\blacksquare^x_{\mathbb{Z}_n}$	$\blacksquare^x_{\mathbb{G}_1}$	$\blacksquare^x_{\mathbb{G}_T}$	$e(x,y)$	$*/_{\mathbb{G}_1}$	$\blacksquare^x_{\mathbb{G}_1}$	$\blacksquare^x_{\mathbb{G}_T}$	$e(x,y)$
issueCred	18	17		5		10		2
Receiver	*10*	*8*		*5*		*4*		*2*
Issuer	*8*	*9*				*6*		
revokeCred	1	1			1			
updateWit	*1+1*	Δ_t			Δ_t+1			
verify	1	1		2				2
authenticate	52 [+2]	25	24	9 [+12]		31	28	24 [+4]
Prover	*25 [+2]*	*14*	*10*	*3 [+5]*		*19*	*12*	*9 [+2]*
Verifier	*27*	*11*	*14*	*6 [+7]*		*12*	*16*	*15 [+2]*

11.3.3 Timing Analysis

Table 11.3 shows the results of the experiments for all the protocols in the three schemes. The results are averaged over 200 test-runs in an accumulator scheme with a maximum capacity of 2500 elements. The witness update results are presented separately.

The results clearly reflect the analysis of the computational complexity. The setup of the CL scheme takes substantially more time than the schemes using bilinear pairings. CL requires the generation of an RSA-modulus as a product of two safe primes, which is dominating the setup. Note that the setup of the CL scheme takes on average 1.5 minutes, while the same algorithm takes about 2.5 minutes in the `Identity Mixer` library (which was implemented in Java™). The setup time of the CKS scheme, however, includes the creation of state information, which is computed by a large number of exponentiations (twice the capacity of the accumulator). For accumulators with a large capacity (N_t), this may take a substantial amount of the initialization time. Another interesting fact, not shown in the table, is that for the CL scheme, the generation of the prime e takes about $1/3$ of the time needed for the issueCred protocol. For authentication, the CL scheme scores best. Nevertheless, an implementation of the CG scheme [CG05] shows an even better performance, with only 127 ms in total for an authentication. This is due to a more efficient proof of knowledge of the accumulated value. In the CL scheme presented here, the proof of

Table 11.3: Performance Results for the Three Schemes for Initialization of the Scheme (initScheme), Issuing a Credential (issueCred), Revoking a Credential (revokeCred), Verifying the Correctness of the Accumulator (verify) and Showing a Credential (authenticate). The issueCred and authenticate Protocols Have Also Been Measured for Each Party Separately.

(ms)	CL □	LN ■	CKS ■
initScheme	97*s*	1,26*s*	1,26*s* $+N_t.4ms$
issueCred	617	365	219
Receiver	274	230	110
Issuer	343	135	109
revokeCred	23	14	0,09
verify	1,90	130	93
authenticate	684	754	938
Prover	389	346	448
Verifier	296	408	490

Configuration
Language : C++
Compiler : Cygwin C++
OS : Windows 7(64)
Processor : DELL Latitude P9600 @ 2.53GHz with 4GB RAM

knowledge is a combination of a CL signature with the proof that a committed value included in the CL signature is accumulated, while in the CG scheme the accumulated prime is also the prime used in the credential signature resulting in a simplified and efficient proof of knowledge.

Fig. 11.1 shows the time required for updating a witness, depending on the number of elements (from 1 up to 10,000) that have been revoked since the previous witness update. It clearly shows the linear relation with respect to the number of revoked

elements. Similar results are found when elements are added to the accumulator. The figure reveals that the CKS scheme clearly outperforms the others. Nevertheless, the CL and LN scheme may still be useful in specific settings.

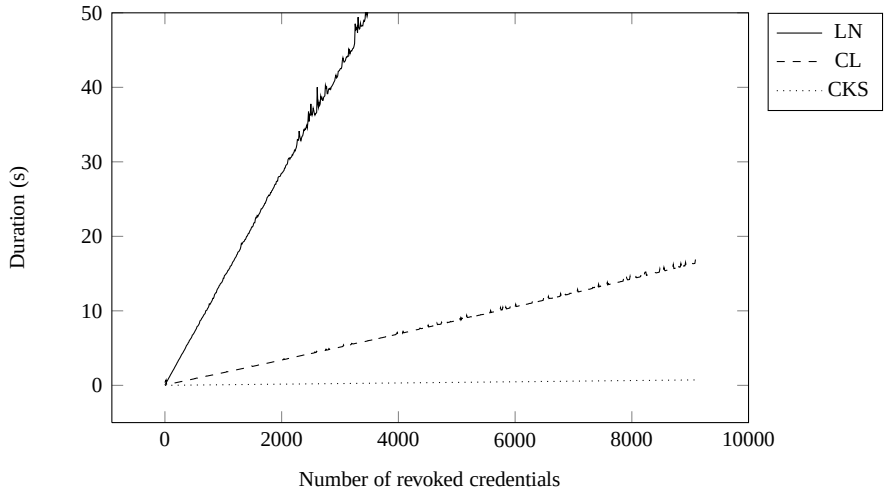


Figure 11.1: Performance Results for Witness Updates with Respect To The Number of Revoked Credentials, Shown Graphically.

11.4 Discussion

11.4.1 Current Bottlenecks

As the three schemes are based on different security assumptions, a straightforward comparison is difficult. While the CL scheme is based on the *strong RSA* assumption, both CKS and LN schemes are based on the *q-SDH* assumption. However, the CKS accumulator scheme requires two additional assumptions: the *n-DHE* and *n-HSDHE* assumption. According to [Che06], the *q-SDH* assumption is a weaker assumption than the *n-DHE* assumption. As a result, the CKS scheme could have a weaker security than the LN scheme. Additionally, the efficiency of the pairing based systems (i.e., LN and CKS) strongly depends on the efficiency of the selected pairing and its implementation.

When we analyze the signatures, we can observe that the LN and CKS signature schemes have a similar construction. The most important difference is that in the CKS

version, the accumulated value is added as a group element, while in the LN scheme it is an exponent. As a consequence, showing a CKS credential requires the proof of a group element. This makes the proof of knowledge (in the authenticate protocol) more complicated than the LN version. Nevertheless, the CL scheme outperforms the others for this protocol.

The benefits and drawbacks of the individual schemes are clearly distinct. The construction of the accumulator is important, with a major impact on the updateWit protocol. On the other hand, the efficiency of issueCred and authenticate heavily depends on the design of the credential scheme accompanying the accumulator scheme. Table 11.4 summarizes the most important bottlenecks of the schemes.

Table 11.4: Bottlenecks (👎) and Benefits (👍) of the Schemes for Each Protocol Separately.

	CL	LN	CKS
initScheme	👎 safe prime generation	👍	👎 state info
issueCred	👎 prime generation		👍
updateWit		👎 exponentiations	👍 (👎 size of state info)
authenticate	👍		👎 exp's + pairings

As for efficiency in time and processing, the CL, LN and CKS schemes are comparable, with CL scoring the best on the authenticate protocol. However, the LN scheme is faster at the prover side for the same protocol with smaller credentials.

Though still acceptable for most practical applications, the CKS scheme is the slowest for proving a valid credential. On the other hand, this scheme clearly outperforms the others with respect to witness updates: it is the only scheme that is practical for massive deployment. It is about 180 times faster than the LN scheme, and 22 times faster than the CL scheme. Yet, there is a snag in it. For witness updates, the CKS scheme requires state information, the size of which is linear in the capacity of the accumulator. For instance, with the configuration above, an accumulator for 10 million elements, requires about 4.8 GB of storage. However, since the update of the witness does not require any secret information, special purpose (possibly untrusted) services may perform the update remotely. With respect to storage, the credentials are comparable in size, with the LN-credential the smallest with only 6528 bits (i.e., 816 bytes).

Large versus small scale environments. The scheme that will be selected depends on the characteristics of the application. In small scale environments with a

limited number of revocations or additions, the efficiency of the authenticate protocol may be more important than the efficiency of the updateWit protocol. However, an important reason for doing this experiment is to explore the applicability of this technology for the use with an electronic identity card (eID) in a nationwide environment. Similar to the previous chapter, we compare with the Belgian eID for which the accumulator size should contain about 10 million elements with about 375,000 revocations a year. As noted before, this large number of revocations is largely due to architectural decisions in the Belgian eID infrastructure.

Suppose in an 'extreme' case, the eID card is used only once a year; this is a valid assumption as a recent survey on the use of the Belgian eID in corporate environments[3] reveals that 56% of the respondents *never* used it. If we can make an interpolation, this would mean that in the best case (i.e., update time grows linearly) an update of 375,000 revoked elements takes about 0.5 minutes, using the fastest update scheme (CKS) and 10 minutes with the CL scheme. While the former may be acceptable in applications such as eID authentication, the latter certainly is not. In the example, we only take the revocations into account, as the accumulator can be precalculated (see *Preissuance-Accumulation* below).

11.4.2 Practical Solutions

Together with improving the efficiency of the protocols, some relevant application-level optimizations can render the schemes practical:

Preissuance-Accumulation. During the introduction of an electronic identity infrastructure, many users are added to the accumulator. To reduce the number of updates, the accumulator could be precomputed. This means that every required element is added to the accumulator and stored securely by the issuer, together with its respective witness.
The CL scheme does not require this precomputation. As already pointed out by the authors [CL02], the witness can simply be calculated from the current accumulator by calculating the id_xth root of the accumulator, with id_x the new 'accumulated' value. This is possible when the factorization of the modulus is known.

Delegation of witness updates. To make the CKS scheme practical without loss of privacy, the witness update should be performed by special purpose update servers. This same strategy may be useful for the other schemes as well. For instance, a resource constrained device, such as a smart card, can delegate the calculation to a more resourceful host.

[3] SAP Survey: Belgen verdeeld over gebruik van eID op het werk (Sept. 2009 by Indigov).

11.5 Conclusion

Similar to the previous chapter, these experiments do not yield a straightforward winner. The revocation of anonymous credentials takes a considerable share in the efficiency of the overall system.

The efficiency of witness updates is an important property in accumulator based revocation systems, and becomes critical in applications with a substantial amount of revocations or additions. The construction of the accumulator has a major impact on the efficiency of the update. Nevertheless, the witness update is also affected by the design of the credential scheme accompanying the accumulator scheme.

Moreover, the computations for proving a valid credential often get a substantial overhead due to the additional proofs of knowledge required for proving that the credential was indeed accumulated. Therefore, although the accumulators can be used as a building block for anonymous credentials, care must be taken when combining it with an actual credential scheme.

Chapter 12

Evaluation

The practicality and applicability of anonymous credential schemes in real-life settings are an on-going discussion and important aspects require further analysis. Specifically, a proper solution for practical revocations is still missing. In this part of the dissertation, we analyze a number of revocation mechanisms, in order to provide a better view on the current state of the art and to be able to define some guidelines on which mechanisms should be used in a particular setting.

12.1 Revocation – Observations

12.1.1 Crypto Primitives

Groups. Protocols using RSA based groups are often easier to implement, using simple primitives. Pairing (i.e., bilinear map) based cryptography, on the other hand, is a relatively young area of cryptography that in contrast to the RSA based groups, requires more complex primitives. A straightforward comparison of RSA and pairing based cryptosystems is not possible.

For the protocols using bilinear maps, we use the PBC library [17] (in C++), which is one of the few currently available. For Java™ it is even more cumbersome. Only very recently a Java™ library [14] implementing bilinear maps has been published. Actually, it is ported from the PBC library. Currently, implementing bilinear maps on a standard Java Card is even more challenging, if not impossible. Hence, using pairings for card-based electronic identities is difficult.

Security. All schemes are proved secure with similar properties. However, those proofs are based on a broad range of assumptions, that are frequently hard to compare. This also implies that selecting concrete system parameters that offer an equivalent security level can be very challenging. Analogously, also in actual implementations, finding comparable system parameters based on a specific security parameter is a research topic by itself.

Implementation. As expected, Java™, as used for the implementation of the `Identity Mixer` library, is substantially slower than the same protocols written in C++. For instance, the same CL accumulator scheme was implemented in both Java™ and C++ in Chapter 10, resp. , 11. In the former, showing a credential takes approximately 3.8 s, while the latter only takes about 0.7 s with the same security parameters and test environment. Moreover, for the Java™ implementation, the implementation of the virtual machine is also important. For instance, running the show protocol partially on an Android mobile device only takes 0.9 s (see Sect. 7.1). The main reason for this is that the Dalvik virtual machine in Android implements the `BIGINTEGER` class in native code, while the Java™ virtual machine implements it entirely in Java™ managed code.

12.1.2 Strategies

For revocation, based on the payload of the parties, we identified three classes: in a first class, comprising VLR-based schemes, the *verifier* has to check the revocation status during a credential show; the second class comprises of schemes in which the *issuer* gets the overhead through the generation of credential updates (i.e., credentials with a limited lifetime LL) or revocation lists RL; and finally the third class comprises the accumulator based schemes Acc, in which the *user* has to update his credential. Each of these strategies have different properties. We now present the most important observations.

Credential Updates. LL, Acc and RL based schemes all require updates, be it of the credential, witnesses or the revocation list. While LL and RL based schemes require the user to only download a small message, accumulator based schemes also require the user to make additional computations. These schemes present an important *change in strategy* with respect to standard revocation schemes (e.g., CRL and OCSP). This may have a major influence on the architecture in which the credentials may be applied. For instance, witness updates are harder to be implemented on Java Cards as they require a substantial amount of computations. To mitigate this problem, some accumulator based schemes allow other parties to perform the witness update. In some schemes [CKS09], this may be done by a possibly untrusted party, while other

schemes require the private key of the issuer. Nevertheless, the card needs some kind of communication with the issuer. On the other hand, in Part II we showed that a combination of a tamper resistant element with a mobile device may make these schemes feasible.

In contrast, VLR schemes do not need the credential to be updated and allow credentials to be fixed at issuance, while a public list of tokens related to revoked credentials is frequently updated. Hence, these constructions allow an architecture similar to the current standard revocation systems.

Credential Show. To preserve the properties of the credential system, revocation mechanisms often require complex proofs of knowledge, implying a negative influence on the efficiency of a credential show. In the credential show of VLR based schemes, this overhead is limited for the user. Moreover, in some schemes, the extra token released during a credential show may be used as a pseudonym. Unfortunately, the verifier gets a substantial overhead. Current VLR schemes require a number of computations linear in the number of revoked credentials. Although some papers mention batch verification as an optimization, and therefore refer to batch verification of signatures [BGR98], it is not clear how this could be achieved for the verification of the revocation status.

The other strategies are more related to each other, resulting in an up-to-date credential or revocation information. Since current anonymous credential schemes are often optimized for efficient proofs of knowledge and LL do not essentially alter the construction of the credential, LL based schemes allow for efficient credential shows. On the other hand, revocation lists and accumulator based schemes often result in more elaborate proofs during a credential show protocol. In revocation lists, it requires at least an additional proof of knowledge of a certificate in the revocation list, while in most accumulator based schemes, credential shows are a combination of an existing and efficient signature scheme with a new construction of an accumulator. Only in the scheme presented by Camenisch and Groth [CG05], the accumulator introduces practically no overhead in the credential show protocol.

Revocation information. An important observation with respect to efficient revocation schemes is that, revocation information in VLR schemes is the same for all verifiers, while for the other strategies, revocation information is user specific. On the other hand, in VLR schemes, the verifier has to perform revocation checks during each credential show, while in the other classes, users only have to update their credential or revocation information once per revocation or time interval.

Table 12.1: Feasibility of The Schemes w.r.t. Latency, Connectivity and Resources (👍: positive; 👎: negative; else: neutral).

	Latency	Offline		Low Resources	
		U	SP	U	SP
Nym	👎	👍	👍	👍	👍
VE	👎	👍	👍		
LL		👎	👍		👍
RL		👎	👍		👍
Acc	👍	👎	👍	👎	👍
VLR	(👍)	👍	👎	👍	👎

12.2 Applying Revocation Schemes – Guidelines

It is clear that there is not one strategy superior to all the others. Therefore, we end the analysis with an overview of which strategies are useful in which settings. Nevertheless, a combination of multiple strategies may sometimes offer the best trade-off. The guidelines are summarized in Table 12.1 and discussed below.

Latency. For high security environments (i.e., requiring low latency) accumulator based revocation is the most secure and privacy-friendly strategy, closely followed by some verifier local revocation schemes. For the latter, one has to select a VLR scheme carefully that provides adequate anonymity. On the other hand, for lower security environments, LL provides a reasonable trade-off. RL offers a similar solution but is not restricted to the issuer to act as revocation manager.

Processing Environments. Often a user's credential is kept in resource constrained environments (e.g., a smart card). In this case, VLR schemes require the least computational overhead for the user. Also LL is a possible alternative. Most RL and Acc schemes, however, require complex computations, making these scenarios less effective in resource constrained environments. On the other hand, the accumulator based scheme by Camenisch and Groth [CG05] in combination with witness updates performed by another party, may be a good trade-off, with high security and efficient verification. In other settings, the verifier has limited resources (e.g., a door lock). In this case VLR is not an option.

Connectivity. In the case of RL and Acc the user requires frequent communication with the issuer. On the other hand, for the service provider in the case of LL and RL, it is sufficient to keep track of time to be able to verify the revocation status. This is especially important for service providers with limited connectivity (e.g., no network coverage). In this case, when computing power is not an issue, the more secure accumulators may provide an alternative. Then the user should provide the latest accumulator, signed by the revocation authority. The verifier then simply checks the validity time of the revocation list.

Online environments offer more freedom. In some schemes, computation may be outsourced to other possibly trusted environments. For instance, verification in the VLR setting may be done by an external more powerful party. When the verifier outsources this verification to a more powerful trusted party, it actually implements a kind of OCSP scenario. Related to accumulator schemes, some schemes [CKS09] also take advantage of remote witness updates.

Clustering. Instead of having a single group for all users, the group may be split into N smaller groups resulting in less revocations per group. During a credential show, the verifier only gets to know that a certain prover is part of a specific group. For instance, users could be classified per region or even at random into a specific group. As a result, the average number of revocations will be about N times less than would be the case with a single group. However, this may entail important privacy consequences. A service will always be able to link an anonymous user to an accumulator. In the worst case, if only one customer of that service is assigned to a particular accumulator, then the service can link all the user's actions.

Combinations. Combinations of multiple strategies may provide solutions for certain settings. For instance, credential updates can easily be combined with accumulator based schemes or VLR schemes. Smaller intervals between credential intervals allow for less witness updates or smaller lists in VLR and RL based schemes, hence, less computations.

12.3 Conclusion and Future Directions

Our analysis shows that there is no straightforward winner. However, using the table above, for a specific architecture a number of strategies can be ruled out.

Nevertheless, for the large scale settings we envision, the current revocation strategies do not provide an easy answer. VLR schemes are closest to standard CRL and OCSP, but they are only efficient when combined with a strategy to keep the number of

revocations minimal. The other schemes require substantially more communication between the user and issuer. This cannot be neglected, and requires a new approach. For instance, electronic identity cards require frequent connectivity for updates and a proper bookkeeping system in order to keep the credential valid. Moreover, allowing updates on the card requires extra security measures on the smart card. In that sense, our mobile authentication application may be an interesting setup.

Based on this research an important conclusion is that the main advantage of TYPE 2 anonymous credentials (see Sect. 1.4), of only requiring a single credential, is possibly lost if a revocation scheme (e.g., Acc, LL, RL) is introduced. On the other hand, revoking TYPE 1 credentials is also a difficult problem.

Probably the best solution, though still an open problem is the following:
Is there an efficient batch verification mechanism for efficiently verifying the revocation status in a backward unlinkable VLR *scheme?*

Part IV

Secure Application Modeling

Anonymous credential systems are complex systems supporting privacy-friendly transactions. However, it makes no sense to use such advanced technologies, if the larger system in which the credentials are used, does not protect the privacy of the user, for instance, by requiring additional information (i.e., address or credit card number) to be disclosed in order to make use of their services. Hence, to make anonymous credentials really useful, they should be accompanied by an infrastructure that provides sufficient guarantees for the service provider to properly run its business. However, with the increasing complexity and constraints of these systems, it is often not straightforward to set up such infrastructures.

The use of cryptographic protocols is not sufficient to build secure and privacy-friendly applications. Moreover, showing that a system is secure, is generally a hard and tedious task. Standard strategies (i.e., game-based proofs) commonly used for proving the security of simple cryptosystems often do not provide sufficient guarantees in larger settings. Simulation-based strategies may offer a way out. Their composability properties allow cryptosystems proved secure in this model, to be re-used as components in a larger setting. Thus, once such a component has been proved secure, it can be used as a building block for building new and more advanced systems, without the need to re-prove its properties.

In this part, we analyze how simulation-based security models can be applied for building larger complex systems. Particularly, we use the Inexhaustible Interactive Turing Machines (IITM) model by Küsters [Küs06], which extends and generalizes existing simulation-based models [Can01, PW01, CLOS02, BPW07]. We provide a number of components/building blocks in Chapter 13, and as a validation of the framework, and our building blocks, we model the concept named *Oblivious Trusted Third Parties (OTP)*, first presented by Camenisch et al. [CGHB08]. The authors only provide a high-level construction for such a protocol but do not present a concrete instantiation. In fact, it is not clear whether their strategy indeed fulfills the requirements. In Chapter 14, we formalize an improved version of this concept and present an actual implementation using the building blocks presented before.

Contributions: This part details a subset of the joint work with Jan Camenisch, Kristyian Haralambiev, Markulf Kohlweiss and Vincent Naessens, published in the proceedings of the *Conference on the Theory and Application of Cryptology and Information Security* [CHK+11a]. In [CHK+11b], we bring a more detailed description of our research. In this research a structure preserving encryption scheme is presented, which is used to implement oblivious trusted third parties. We present a (simulation-based) model of the OTP functionality and prove the realization based on the public key encryption scheme, secure with respect to this model.
We focus on the contributions in which I was mainly involved, namely the modeling of the ideal and real protocols in the IITM simulation-based model. The parts in

which I was less involved such as the structure preserving encryption scheme and the efficient zero-knowledge proofs, are for completeness included in Appendix C. For more details on the structure preserving encryption scheme, we refer to the original publications [CHK$^+$11a, CHK$^+$11b].

Chapter 13

Modeling Secure Applications

13.1 Introduction

Historically, cryptography was mainly used in military. Today, however, partially due to the increasing connectivity of appliances, cryptography is used anywhere: for the protection of communication, authentication of users, data and software protection, and many more. Cryptography is that part of information security that deals with the development and analysis of protocols to secure data.

Classical cryptography was more like an art, relying on creativity and personal skill. Unfortunately, such schemes were eventually broken. Since the late 20th century, (*modern*) cryptography has radically changed, resting on stronger and more scientific foundations, into an actual science active in multiple fields, such as electronic engineering, computer science and mathematics. As from then, ad-hoc systems were being replaced by systems with proven security guarantees (i.e., depending on the model and assumptions [KM07]).

Shannon was the first to define *perfect security*. However, perfectly secure systems have fundamental limitations. Instead, this notion of security is relaxed into a notion of *computational security*. In such cryptosystems security is based on the computational infeasibility of breaking the system.

A common strategy in proving the security of a cryptosystem is based on a *reduction* from the security of the cryptosystem at hand, to some computational hardness assumption, for instance, the assumption that factoring or computing discrete logs is hard. As a result, if an adversary can break the cryptosystem, then it can break the hard problem.

Though, in order to proof a system secure, we first need to define what *secure* actually means. In other words, we need a formal security model (security notion). In addition, we need to define the computational assumptions, to which we reduce our system, followed by the actual proof.

Currently, there are two main approaches to model security, namely *game-based* notions of security and *simulation-based* notions of security.
In the former, security is phrased as a game, played between a hypothetical challenger and an attacker. The security model is problem specific and defined by the responses of the challenger. It defines what is considered to break the system, and also the power of the adversary. The advantage of game-based definitions is that they are often simple to understand and manipulate. However, when systems become too complex, it is hard to come-up with a game-based definition that properly defines the security requirements. Moreover, a proof based on such definitions does not say anything about the security when it is applied in a larger system.

In simulation based security, on the other hand, security is defined in terms of an *ideal system*. A real cryptosystem is then assumed to be secure if any attack in the real system can be translated into an equivalent attack in the ideal system. Proving the security of the system then comes down to proving that both the real attack and the ideal attack are indistinguishable.

An interesting property of these models is that simulation-based definitions guarantee security under composition. Security is preserved even in larger settings, where multiple protocols may run concurrently. Thus, secure protocols can be used as building blocks for building more advanced secure systems, based on the security of those building blocks.

In this chapter, we propose a general approach to simplify the modeling of ideal systems. In addition, we provide a number of ideal systems for which realizations exist, as building blocks for more advanced systems.

This chapter is structured as follows. Sect. 13.2 recalls the simulation-based model being used, followed by Sect. 13.3 proposing some simplifications in order to make the modeling easier. Finally, we present a number building blocks in Sect. 13.4.

13.2 The IITM model

Several simulation-based models have been developed [Can01, PW01, BPW07, Küs06]. Küsters [Küs06] has proposed a general computational model, that generalizes most of these existing simulation based models. Moreover, the model allows to present the relationship between the different simulation-based notions of

security [KDMR08]. We now recall the general computational model as presented by Küsters [Küs06].

13.2.1 The General Computational Model

In [Küs06] both ideal systems $\mathcal{I}$ and their realizations as cryptographic protocols $\mathcal{P}$ are configurations of so-called inexhaustible interactive Turing machines (IITMs). An IITM M is a probabilistic polynomial-time Turing machine with named in put and output tapes. They are called inexhaustible, as the runtime may depend on the length of the data received on input tapes so far, and in every activation of the IITM it may perform a polynomial-time computation.

The first IITM to be activated in a *run of the system* is called a master IITM. It is also triggered if no other IITM was triggered. An IITM is triggered by another IITM if the latter writes a message on an output tape that corresponds to an input tape of the former. Note that on each activation of an IITM, it can write to at most one output tape. If no message was produced at the end of its activation, the master IITM is triggered. If the master IITM does not produce output, or an IITM has written a message to an output tape named *decision*, the run stops.

The names of the tapes define how IITMs may be connected into a system of items $S = M_1|\dots|M_k|!M'_1|\dots|!M'_k$, with M_i and M'_j IITMs such that there are no common names for input tapes. Moreover, IITMs M'_j may contain an unbounded number of copies of IITMs as indicated by the bang operator ('!'). Küsters therefore proposes a flexible and generic mechanism for addressing those copies of IITMs. An IITM may run in two modes: in the CheckAddress mode the IITM runs a deterministic algorithm to verify that a message is in fact addressed to it; in the Compute mode the IITM will do the actual processing of the input and possibly writes output to one of its output tapes. If no current instance of the banged IITM accepts the input, and the default instance accepts in the CheckAddress mode, a new copy is created.

Input tapes can be either *consuming* ($\rightarrow$) or *enriching* ($\twoheadrightarrow$), of which the length of the inputs on the latter is a bounding factor for the size of the current configuration and the output produced by that IITM. In order to ensure that such systems run in polynomial time, *well-formed* systems require a graph defined by the enriching tapes to be acyclic.

The model of [Küs06] further guarantees that well-formed systems of polynomial time bounded IITMs can be simulated by a single IITM. This allows us to interpret an ideal system and a protocol either as an interconnected system that communicates via input/output tape pairs shared between component IITMs, or as a single IITM that manages all external tapes. This is an important difference with other simulation-based models, in which the ideal system is only presented as a single ITM.

13.2.2 Simulation-Based Security Notion

For simulation-based security definitions, we consider three types of systems: real and ideal protocols, simulators, and environments. The types are grouped into network and I/O interfaces. Protocol systems and environments both have an I/O and network interface and adversarial systems (i.e., simulators) only have a network interface.

We recall two definitions in [Küs06]. The notion of negligible function is standard and follows [Can01, Küs06].

Definition 7. *Two systems $\mathcal{P}$ and $\mathcal{Q}$ are called indistinguishable* ($\mathcal{P} \approx \mathcal{Q}$) iff *the function*

$$f(1^k, a) = |\Pr[\mathcal{P}(1^k, a) = 1] - \Pr[\mathcal{Q}(1^k, a) = 1]| \quad \textit{is negligible.}$$

The security notion of strong simulatability is depicted in Fig. 13.1 and formally defined as follows:

Definition 8 (Strong Simulatability)**.** *Let $\mathcal{P}$ and $\mathcal{I}$ be a real, resp. ideal protocol system with the same I/O interface. Then $\mathcal{P}$ realizes $\mathcal{I}$ ($\mathcal{P} \leq \mathcal{I}$)* iff *there exists a simulator $\mathcal{S}$ such that the systems $\mathcal{P}$ and $\mathcal{S}\,|\mathcal{I}$ have the same external interface and for all environmental systems $\mathcal{E}$, connecting only to the external interface of $\mathcal{P}$ (and hence, $\mathcal{S}\,|\mathcal{I}$), it holds that $\mathcal{E}|\mathcal{P} \approx \mathcal{E}|\mathcal{S}|\mathcal{I}$.*

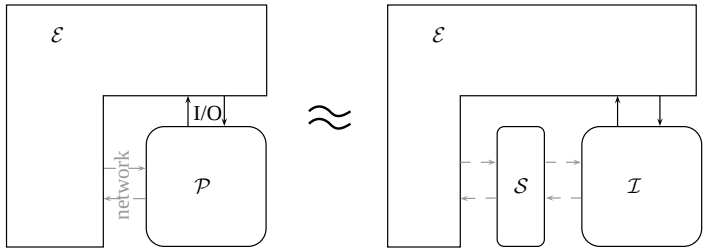


Figure 13.1: A Run of the Environment $\mathcal{E}$ and the Real Protocol $\mathcal{P}$ is Indistinguishable from a Run of the Environment $\mathcal{E}$, the Ideal Protocol $\mathcal{I}$ and a Simulator $\mathcal{S}$, where $\mathcal{S}\,|\mathcal{I}$ have the Same External Interface (i.e., network and I/O) as $\mathcal{P}$

13.3 Simplified Modeling

The framework provided by Küsters is very flexible. As mentioned before, both ideal and real protocols may be presented as a single ITM but also an interconnected system

of ITMs. However, this flexibility makes the modeling of actual application also challenging, as there are multiple ways to represent the same functionality.

Before we provide a number of building blocks, we put some restrictions on how we model these ideal and real functionalities. We first present some extensions to the notation presented by Küsters, followed by a strategy to model (static) corruption of *ideal systems* based on a combination of a virtual incorruptible functionality representing the (ideal) functionality of the system and corruptible dummy functionalities representing the roles in the system that may be corrupted. Communication in this ideal system is then modeled using *delayed communication*. Furthermore, we define *real systems*/realizations as a combination of ideal and real protocols (i.e., a hybrid system). Therefore, we present a number of ideal building blocks: secure communication, a generic zero-knowledge ideal functionality for which efficient realizations exist, and an ideal functionality for a generic secure two-party computation. For the latter, we also present a realization for a joint ciphertext computation, based on the structure preserving encryption scheme (see Appendix C.1), the secure channel functionality, and the generic zero-knowledge ideal functionality.

13.3.1 Notation

As a convention, we bundle communication tapes into interfaces *inf* where an interface consists of named input/output tape pairs. An input/output tape pair is named $inf.\mathsf{R}$ after a combination of the interface name *inf* and a role name R. We refer to the set of all roles of an interface as $inf.\mathcal{R}$. If a system of IITMs implementing an interface *inf* is connected to another IITM M then as a convention, we refer to the swapped input/output tape pair of M connected to role R as $\overline{inf}.\mathsf{R}$.

For each system $\mathcal{S}_{inf}$ implementing a functionality *inf*, we distinguish between the *API inf* (called IO interface in Küsters terminology), defining the environmental/trusted connections ($\rightarrow$) of the system and *network interface ninf*, defining the adversarial/untrusted connections ($\dashrightarrow$) of the system.

For example, if an IITM M wants to send a message to role R of a system of IITMs $\mathcal{S}_{inf}$ implementing *inf*, M would write the request to the output tape of $\overline{inf}.\mathsf{R}$ and $\mathcal{S}_{inf}$ would read it on the input tape of $inf.\mathsf{R}$. To answer the request $\mathcal{S}$ would write the response on the output tape of $inf.\mathsf{R}$ and M would read the request on the input tape of $\overline{inf}.\mathsf{R}$. Similarly, an adversary $\mathcal{A}$ would send messages to the network output tape of $\overline{ninf}.\mathsf{R}$ and $\mathcal{S}_{inf}$ would read it on the input tape of $ninf.\mathsf{R}$.

For simulation-based security definitions the ideal protocol $\mathcal{I}$ and the real protocol $\mathcal{P}$ that emulates this ideal system have to present the same API *inf* towards their environment, i.e., they must be *API compatible*. We refer to an ideal system and a protocol that is API compatible with respect to interface *inf* as $\mathcal{I}_{inf}$ and $\mathcal{P}_{inf}$

respectively. In addition $\mathcal{I}_{inf}$ and $\mathcal{P}_{inf}$ must expose different network interfaces $nintf_1$ and $nintf_2$.

In our notation the definition of strong simulatability can be rewritten as:

Definition 9. *A protocol system $\mathcal{P}_{inf}$ strongly emulates $\mathcal{I}_{inf}$,* iff *there exists a simulator $\mathcal{S}$ connected to $\mathcal{E}$ on interface $nintf_2$ and to $\mathcal{I}_{inf}$ on interface $nintf_1$, such that for all environments $\mathcal{E}$ that connect to inf and $nintf_2$: $\mathcal{E}|\mathcal{P}_{inf} \approx \mathcal{E}|\mathcal{S}|\mathcal{I}_{inf}$*

Küsters [Küs06] describes how to turn every system $\mathcal{S}$ of ITMs into a multi-session system $\underline{\mathcal{S}}$, by programming each ITM instance to accept only messages prefixed with a specific session id, and adding the same session id to all outputs produced by that instance. This is denoted by the session operator $\underline{\ }$. For polynomially many sessions the composition theorem guarantees that given $\mathcal{P}_{inf} \leq \mathcal{I}_{inf}$, $!\underline{\mathcal{P}_{inf}} \leq !\underline{\mathcal{I}_{inf}}$. Informally, the bang operator '!' denotes on demand creation of session specific instances.

The default way of obtaining a multi-session version of a protocol by the bang and session operator requires a fresh copy of all ITMs in a system for every session. However, the sessions of a protocol can often make use of joint resources. For an adequate joint-state realization $\mathsf{P}_1|\mathcal{I}_{sc}|\mathsf{P}_2|\mathcal{I}_{\mathsf{crs}}$ of $\mathcal{P}_{inf}$ that, for instance, makes use of a common reference string functionality,[1] we can write $!\underline{\mathsf{P}_1|\mathcal{I}_{sc}|\mathsf{P}_2}|\mathcal{I}_{\mathsf{crs}} \leq !\mathcal{I}_{inf}$. For further information on the joint state theorem for the model of [Küs06] we refer to [KT08].

13.3.2 Corruption

Küsters [KT08] presents a standard corruption model for ITMs formalized in Listing 13.2. Each corruptible party implements this protocol, independent of whether the party is part of an ideal or a real protocol system. A corrupted role, as depicted in Fig. 13.2, forwards all inputs on I/O tapes $T \in \mathcal{T}_{\mathsf{U}}$ to $ninf_i.\mathsf{R}$ and acts as a proxy that allows the environment to send arbitrary messages to any of its tapes in $\mathcal{T}_{\mathsf{U}}$, by sending control messages on the network tape $\overline{ninf_i}.\mathsf{R}$.

In our exposition, we consider only static corruption. Therefore, after the first activation of a corruptible party (i.e., a message (`Ready`) was received), *corruptible* is set to true, and as soon as the ITM is activated again (i.e., a new message is evaluated), *corruptible* is set to false.
The (`Resources`) message is more a technicality, to ensure that there are sufficient resources for the ITM to forward messages, since the computation time depends on

[1] See Listing 9 for the details of the $\mathcal{I}_{\mathsf{crs}}$ functionality.

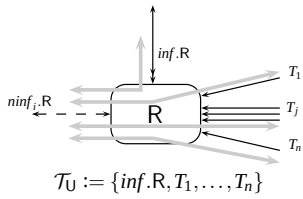


Figure 13.2: Flow of Messages in Case of Corruption.

the size of messages received on enriching tapes. For more information, we refer to the original paper [Küs06].

Listing 2. Macro Corr$(corrupted \in Bool, corruptible \in Bool, initialized \in Bool, msg, inf.R, ninf.R, \mathcal{T}_U)$

Tapes tapes $inf.R$ and $\mathcal{T}_U$ are enriching, while $ninf.R$ is consuming.
Initialization: $res \leftarrow 0$
Compute:
(**Corruption Request**)
On $(\texttt{Corrupted?})$ from inf.R where *initialized*:

– send $(corrupted)$ to inf.R

(**Corruption**)
On $(\texttt{Corrupt})$ received from $ninf.R$ where $corruptible, initialized$ and not $corrupted$

– let $corrupted \leftarrow true$; send $(\texttt{Corrupted}, msg)$ to $ninf.R$

(**Forward to ninf.R (Rule takes precedence over all other rules)**)
On m received from $T \in \mathcal{T}_U$ where $corrupted$

– let $res \leftarrow 0$; send $(\texttt{LeakRecv}, m, T)$ to $ninf.R$

(**Forward to tape**)
On $(\texttt{Send}, m, T)$ received from $ninf.R$, $T \in \mathcal{T}_U$, $corrupted$, $0 < |m| \leq res$

– let $res \leftarrow 0$; send m to T

(**Resources**)
On $(\texttt{Resources}, r)$ received from $inf.R$ where $corrupted$

– let $res \leftarrow |r|$ and send $(\texttt{Resources}, r)$ to $ninf.R$

Note that the IITM framework, although not considered here, also supports more extensive corruption models, for instance, passive corruption, in which the adversary only sees the messages sent by or to the corrupted party, but who cannot modify or introduce new messages.

13.3.3 Functionalities for Modeling the Ideal System.

To simplify the construction of the ideal system, we present three different types of components: *a virtual incorruptible party, corruptible dummy parties, and incorruptible delayed communication*.

Fig. 13.3 illustrates an example ideal system with interface inf, two dummy parties representing the roles P_1 and P_2, and two delayed communication channels with network tapes C_{P_1} and C_{P_2}. This approach allows us to break down the tasks of the ideal system, allowing us to concentrate on the security critical parts of the system. Another advantage of this construction is, that it advances the construction of the simulator in order to proof that a realization securely emulates the ideal system. We now briefly explain the tasks addressed to each functionality.

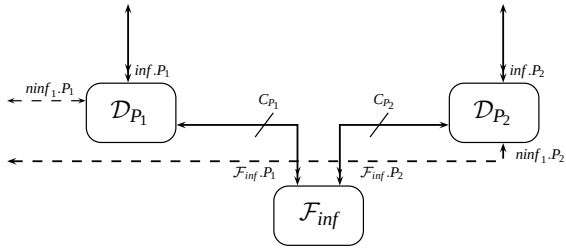


Figure 13.3: Modeling an Ideal System.

Virtual Incorruptible Party $\mathcal{F}_{inf}$. Cryptography has a particular interest in ideal systems that model a virtual incorruptible party $\mathcal{F}_{inf}$. The functionality $\mathcal{F}_{inf}$ implements the security critical parts of an ideal system, without considering corruption or communication.

Dummy Parties $\mathcal{D}_{\mathsf{R}}$. The parties representing the different roles of the interface only need to implement forwarding and corruption. We refer to such a dummy party for role R as $\mathcal{D}_{\mathsf{R}}$. The IITM modeling $\mathcal{D}_{\mathsf{R}}$, for static corruption, is then described as follows:

Listing 3. Dummy functionality: $\mathsf{Dummy}(inf.R, ninf.R, \overline{\mathcal{F}}_{inf}.R)$:

Tapes $inf.R \leftrightarrow\!\!\!\!\rightarrow \overline{inf}.R$, $ninf.R \leftarrow\!\!\!\!\dashrightarrow \overline{ninf}.R$, $\overline{\mathcal{F}}_{inf}.R \leftarrow\!\!\!\!\not\rightarrow\!\!\!\!\rightarrow \mathcal{F}_{inf}.R$
Initialization: $state \leftarrow \varepsilon; corrupted, corruptible \leftarrow false$
Compute:
On (`Ready`) from $inf.R$ where $state = \varepsilon$:
- let $state \leftarrow$ "ready"; let $corruptible \leftarrow true$

– send (Ready) to $ninf.R$

On m from $inf.R$ where $state =$ "ready"

– let $corruptible \leftarrow false$; send m to $\overline{\mathcal{F}}_{inf}.R$

On m from $\overline{\mathcal{F}}_{inf}.R$ where $state =$ "ready"

– let $corruptible \leftarrow false$; send m to $inf.R$

Corruption:

Corr $(corrupted, corruptible, state \neq \varepsilon, \varepsilon, inf.R, ninf.R, \{inf.R, \overline{\mathcal{F}}_{inf}.R\})$

Delayed Communication. Both ideal and real protocols have to model communication. Ideal protocols model both ideal cryptography, as well as ideal communication. A common situation is when the adversary is ideally only able to arbitrarily delay the delivery of results. This models the restriction that cryptography cannot prevent denial of service attacks against an adversary that is in control of communication resources. We model this commonly recurring pattern as an IITM that, on the (Continue) command on adversarial channel C, copies messages from one tape to another.

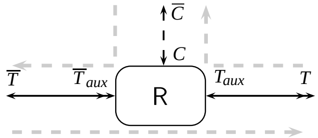


Figure 13.4: Tapes and Message Flow of Functionality Delay$(T, \overline{T}, C)$.

We model enriching delayed communication $T \overset{C}{\twoheadleftarrow\!\!\!/\!\!\!\rightarrow} \overline{T}$ which leaks to network tape C by the IITM presented in Listing 4. In short, we delay all messages in the non-enriching direction (see Fig. 13.4). In our approach for modeling ideal systems, this means that messages send out from the virtual incorruptible party $\mathcal{F}_{inf}$ towards dummy parties is delayed.

Listing 4. Functionality Delay$(T, \overline{T}, C) \equiv T \overset{C}{\twoheadleftarrow\!\!\!/\!\!\!\rightarrow} \overline{T}$

Tapes: $T_{aux} \longleftrightarrow\!\!\!\!\!\rightarrow T$, $\overline{T}_{aux} \twoheadleftarrow\!\!\!\longrightarrow \overline{T}$, $C \dashleftarrow\dashrightarrow \overline{C}$

Initialization: $buffer \leftarrow \varepsilon$.

Compute:

On $m = (\dots, \langle$MsgName$\rangle, \dots)$ or $m = (\langle$MsgName$\rangle, \dots)$ from T_{aux}:

– let n_C be a fresh $\mathbb{C}$ nonce

– store (n_C, m) in *buffer* and leak $(n_C,$Leak$\langle$MsgName$\rangle, |m|)$ to C

On $(n_C,$ Continue$)$ from C:

– if $(n_C, m) \notin buffer$ abort

– remove (n_C, m) from *buffer* and send m to $\overline{T}_{aux}$

On m from $\overline{T}_{aux}$: forward m to T_{aux}

13.4 Building Blocks for Real Systems

We now present a number of ideal systems that may be used as building blocks in the construction of real protocol systems. As mentioned before, our real protocol systems are hybrid systems, combinations of ideal and real protocol systems. This allows us to build protocols without restrictions on the actual realization of the sub-protocols. In order to practically implement such a hybrid protocol, the ideal functionalities must be replaced by real protocols that securely realize these ideal functionalities.

Secure Communication

To make abstraction from communication details in a real system, we model communication as functionalities. One important mechanism is end-to-end authenticated secure (i.e.confidential and integrity protected) communication. Key exchange protocols and public-key infrastructures allow for the construction of such secure channels. For simplicity, we model secure channels through an ideal incorruptible functionality.

Secure Channel $\mathcal{I}_{sc}$. We model an ideal secure channel, as a channel in which both the receiver and sender is authenticated. The ideal channel functionality $\mathcal{I}_{sc}$ supports only request/response communication and only a single message can be sent at a time. We model corruption of sender and receiver through dummy users $\mathcal{D}_{\mathsf{S}_1} = \mathsf{Dummy}(sc.\mathsf{S}_1, nsc.\mathsf{S}_1, \overline{\mathcal{F}_{sc}}.\mathsf{S}_1)$ and $\mathcal{D}_{\mathsf{S}_2} = \mathsf{Dummy}(sc.\mathsf{S}_2, nsc.\mathsf{S}_2, \overline{\mathcal{F}}_{sc}.\mathsf{S}_2)$: $\mathcal{I}_{sc} = \mathcal{D}_{\mathsf{S}_1} | \mathcal{F}_{sc} | \mathcal{D}_{\mathsf{S}_2}$.

Listing 5. Functionality $\mathcal{F}_{sc}$

Tapes: $\mathcal{F}_{sc}.\mathsf{S}_1 \nleftrightarrow \overline{\mathcal{F}}_{sc}.\mathsf{S}_1$, $\mathcal{F}_{sc}.\mathsf{S}_2 \nleftrightarrow \overline{\mathcal{F}}_{sc}.\mathsf{S}_2$
Initialization: $active \leftarrow 1$.
Compute:
On $(\texttt{Send}, m)$ on $\mathcal{F}_{sc}.\mathsf{S}_{2-i}$ where $active = 2-i$
- set $active \leftarrow 1+i$; send $(\texttt{Send}, m)$ to $\mathcal{F}_{sc}.\mathsf{S}_{1+i}$

On $(\texttt{Skip})$ on $\mathcal{F}_{sc}.\mathsf{S}_{2-i}$ where $active = 2-i$
- set $active \leftarrow 1+i$; send $(\texttt{Skipped})$ to $\mathcal{F}_{sc}.\mathsf{S}_{2-i}$

Modeling Zero-Knowledge Proofs of Knowledge

For the types of relations required in our protocols, there exist practical ZK protocol realizations. We refer to Camenisch et al. [CCGS10, CKS11] for details. We will be

proving statements of the form

$$\aleph w_1, \ldots, w_n : \phi(w_1, \ldots, w_n, bases). \tag{13.1}$$

Here, we use the symbol "ℵ" instead of "∃" to indicate that we are proving "knowledge" of a witness, rather than just its existence. w_i are exponents and ϕ is a predicate defining discrete logarithm representations — we presently place restrictions on the form of the domains and the predicate. A witness for a statement of the form (13.1) is a tuple $(w_1, \ldots, w_n)$ of integers such that $\phi(w_1, \ldots, w_n, bases)$ holds.

The predicate $\phi(w_1, \ldots, w_n, bases)$ is given by a formula that is built up from "atoms" using arbitrary combinations of ANDs and ORs. An atom may express several types of relations among the w_i-s: *(i) integer relations*, such as $\mathcal{H} = 0$, $\mathcal{H} \geq 0$, $\mathcal{H} \equiv 0 \pmod{m}$, or $\gcd(\mathcal{H}, m) = 1$, where $\mathcal{H}$ is an integer polynomial in the variables $w_1, \ldots, w_n$, and m is a positive integer; *(ii) group relations*, such as $\prod_{j=1}^{k} g_j^{\mathcal{H}_j} = 1$, where the $g_j \in bases$ are elements of an abelian group, and the $\mathcal{H}_j$'s are integer polynomials in the variables $w_1, \ldots, w_n$.

We define the proof instance *inst* to consist of the set of *bases* and of descriptions of the abelian groups. The proof relation $((w_1, \ldots, w_n), inst) \in \mathfrak{R}$ holds *iff* the predicate $\phi(w_1, \ldots, w_n, bases)$ holds. We call a relation $\mathfrak{R}$ tractable, if such a predicate ϕ and consequently an efficient proof protocol, exists. Camenisch et al. [CCGS10, CKS11] show how to construct efficient protocols for these types of statements that, under reasonable assumptions, multi-realize an ideal functionality with joint access to a common reference string. We refer to the original paper for more details.

Zero-Knowledge Functionality $\mathcal{F}_{\mathsf{zk}}(\mathfrak{R})$. We use a zero-knowledge ideal functionality as defined by Listing 6 that is a simplification of the $\mathcal{F}_{ZK}^{R,R'}$ functionality of [CCGS10] for which we consider only static corruption. This allows us to reuse their ZK protocol compiler to obtain efficient multi-session instantiations $\mathcal{P}_{\mathsf{zk}}$ of $\mathcal{I}_{\mathsf{zk}}(\mathfrak{R})$ in the hybrid $\mathcal{I}_{sc}$ and joint-state $\mathcal{I}_{\mathsf{crs}}$ model. The multi-session version of the real protocol $\mathcal{P}_{\mathsf{zk}}(= \mathsf{Pv}|\mathcal{I}_{sc}|\mathsf{Vf}|\mathcal{I}_{\mathsf{crs}})$ securely realizes the multi-session version of the ideal protocol $\mathcal{I}_{\mathsf{zk}}(\mathfrak{R})(= \mathcal{D}_{\mathsf{Pv}}|\mathcal{F}_{\mathsf{zk}}(\mathfrak{R})|\mathcal{D}_{\mathsf{Vf}})$ or more formally: $!\underline{\mathsf{Pv}|\mathcal{I}_{sc}|\mathsf{Vf}}|\mathcal{I}_{\mathsf{crs}} \leq !\underline{\mathcal{I}_{\mathsf{zk}}(\mathfrak{R})}$.

Listing 6. Functionality $\mathcal{F}_{\mathsf{zk}}(\mathfrak{R})$:

Tapes: $\mathcal{F}_{\mathsf{zk}}.\mathsf{Pv} \leftrightarrow\!\!\!\!\rightarrow \overline{\mathsf{nzk}}.F$, $\mathcal{F}_{\mathsf{zk}}.\mathsf{Vf} \not\leftrightarrow\!\!\!\!\rightarrow \overline{\mathcal{F}_{\mathsf{zk}}}.\mathsf{Vf}$
Initialization: $state \leftarrow$ "ready".
Compute:
On $(\texttt{Prove}, inst, wit)$ from $\mathcal{F}_{\mathsf{zk}}.\mathsf{Pv}$ where $state =$ "ready" and $(inst, wit) \in \mathfrak{R}$
– let $state \leftarrow$ "final"; send $(\texttt{Prove}, inst)$ to $\mathcal{F}_{\mathsf{zk}}.\mathsf{Vf}$

Modeling joint ciphertext computation

Before providing the functionality for joint ciphertext computation, we first present a more generic functionality. We define the ideal functionality for the joint computation of any function f of verifiable inputs inp_1 and inp_2. When performing such a two-party computation, party P_{1+i} is guaranteed that P_{2-i} knows a witness wit_{2-i} for its input inp_{2-i} such that $(inst,(wit_{2-i},inp_{2-i})) \in \mathfrak{R}_{2-i}$. We restrict ourselves to tractable relations $\mathfrak{R}_i$ for which we can give efficient zero-knowledge proofs of knowledge as discussed above.

Secure two-party Computation $\mathcal{I}_{\mathsf{tpc}}(f,\mathfrak{R}_1,\mathfrak{R}_2)$. We model an ideal secure two-party computation system $\mathcal{I}_{\mathsf{tpc}}(f,\mathfrak{R}_1,\mathfrak{R}_2)$ with interface tpc as the combination of two dummy Parties $\mathcal{D}_{\mathsf{P}_1}$ and $\mathcal{D}_{\mathsf{P}_2}$ and an ideal two party computation functionality $\mathcal{F}_{\mathsf{tpc}}$, more formally $\mathcal{I}_{\mathsf{tpc}}(f,\mathfrak{R}_1,\mathfrak{R}_2) = \mathcal{D}_{\mathsf{P}_1}|\mathcal{F}_{\mathsf{tpc}}(f,\mathfrak{R}_1,\mathfrak{R}_2)|\mathcal{D}_{\mathsf{P}_2}$.

Listing 7. Functionality $\mathcal{F}_{\mathsf{tpc}}(f,\mathfrak{R}_1,\mathfrak{R}_2)$

Tapes: $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1 \leftrightarrow\!\!\!\!\not\rightarrow \overline{\mathcal{F}_{\mathsf{tpc}}}.\mathsf{P}_1$, $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2 \leftrightarrow\!\!\!\!\not\rightarrow \overline{\mathcal{F}_{\mathsf{tpc}}}.\mathsf{P}_2$
Initialization: $inp_1, pub, inst \leftarrow \varepsilon$; $state \leftarrow$ "ready"
Compute:
On $(\mathtt{Input}_1, inst', pub', wit_1', inp_1')$ from $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1$ where $state =$ "ready" and $(inst',(wit_1',inp_1')) \in \mathfrak{R}_1$
– let $inp_1 \leftarrow inp_1'$, $inst \leftarrow inst'$, $pub \leftarrow pub'$, and $state \leftarrow$ "input1"; send $(\mathtt{Input}_1, inst, pub)$ to $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2$
On $(\mathtt{Input}_2, wit_2, inp_2)$ from $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_2$ where $state =$ "input1" and $(inst,(wit_2,inp_2)) \in \mathfrak{R}_2$
– let $state \leftarrow$ "final"; send $(\mathtt{Result}, f(pub,inp_1,inp_2))$ to $\mathcal{F}_{\mathsf{tpc}}.\mathsf{P}_1$

Joint ciphertext computation $\mathcal{F}_{\mathsf{jcc}}$. For the protocol in the next chapter, we consider a two-party protocol for the *joint computation of a ciphertext* under a third-party public key pk. The encrypted value is a function of two secrets, each of which remains secret from the other protocol participant. We study the case where only the first party learns the ciphertext, whereas the second has no output.

The model of our joint ciphertext computation, is fully described by a secure two party computation as in Listing 7, where $inp_i := (l_i, \vec{x}_i)$, $pub := pk$, and $f := f_{\mathsf{JC}}(pk, (l_1,\vec{x}_1), (l_2,\vec{x}_2)) = \mathsf{Enc}(pk; g^{l_1+l_2}, (g^{x_{1,1}+x_{2,1}}, \ldots, g^{x_{1,n}+x_{2,n}}))$. We apply the structure preserving encryption scheme (with labels l_i) as presented in Appendix C.1.

Implementing $\mathcal{P}_{\mathsf{jcc}}$. We present the protocol for the special case where the jointly computed ciphertext encrypts a single message (i.e., $n = 1$). This can trivially be extended to the multi-message case.

The idea of the protocol is as follows. The first party computes a partial and blinded encryption of her secret, she proves that the computation is carried out correctly, and sends the partial encryption to the other party. The second party takes the values from the first flow of the protocol and, using its secret and some randomness, computes a blinded full encryption of the agreed function of the two plaintext contributions. Then, the second party sends these values and proves that they are computed correctly. Finally, the first party unblinds the ciphertext and updates the consistency element to obtain a valid encryption of the function of the two secrets under jointly chosen randomness. The function can be a constant to the power of any polynomial of the two secrets; for simplicity, we consider the function $g^{x_1+x_2}$ where g is a fixed group element and x_1, x_2 are the two secrets.

Listing 8. Protocol $\mathcal{P}_{\mathsf{jcc}}(\mathfrak{R}_1,\mathfrak{R}_2) = \mathsf{P}_1(\mathfrak{R}_1,\mathfrak{R}_2)|\mathcal{I}_{\mathsf{zk}_1}(\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))|\mathcal{I}_{\mathsf{zk}_2}(\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))|\mathsf{P}_2(\mathfrak{R}_1,\mathfrak{R}_2)$

Party P_1 and P_2 receive input from $\mathsf{jcc}.\mathsf{P}_1$ and $\mathsf{jcc}.\mathsf{P}_2$ respectively and communicate over $\mathcal{I}_{\mathsf{zk}_1}$ and $\mathcal{I}_{\mathsf{zk}_2}$.

On $(\mathtt{Input}_1, inst, pk, wit_1, (l_1, x_1))$ from $\mathsf{jcc}.\mathsf{P}_1$

- if $(inst, (wit_1, l_1, x_1)) \notin \mathfrak{R}_1$, P_1 aborts
- P_1 computes $(msg_1, aux_1) \leftarrow \mathsf{BlindEnc}_1(pk; l_1, x_1)$ and proves $((msg_1, pk, inst), (wit_1, l_1, x_1, aux_1)) \in \mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1)$ to P_2 using $\mathcal{I}_{\mathsf{zk}_1}(\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$
- P_2 learns $(msg_1, pk, inst)$ from $\mathcal{I}_{\mathsf{zk}_1}$ and outputs $(\mathtt{Input}_1, inst, pk)$ to $\mathsf{jcc}.\mathsf{P}_2$

On $(\mathtt{Input}_2, wit_2, (l_2, x_2))$ from $\mathsf{jcc}.\mathsf{P}_2$

- if $(inst, (wit_2, l_2, x_2)) \notin \mathfrak{R}_2$, P_2 aborts
- P_2 runs $(msg_2, aux_2) \leftarrow \mathsf{BlindEnc}_2(pk; l_2, x_2, msg_1)$
- P_2 proves $((msg_2, pk, inst), (wit_2, l_2, x_2, aux_2)) \in \mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2)$ to P_1 using $\mathcal{I}_{\mathsf{zk}_2}(\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$
- P_1 learns $(msg_2, pk, inst)$ from $\mathcal{I}_{\mathsf{zk}_2}$, computes $(ct) \leftarrow \mathsf{UnblindEnc}(pk; msg_2, aux_1)$, and outputs $(\mathtt{Result}, ct)$ to $\mathsf{jcc}.\mathsf{P}_1$

Where abstractly, relations $\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1)$ and $\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2)$ are defined as

$$\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1) = \{(msg_1, pk, inst), (wit_1, l_1, x_1, aux_1)) \mid (msg_1, aux_1) = \mathsf{BlindEnc}_1(pk; l_1, x_1) \wedge (inst, (wit_1, l_1, x_1)) \in \mathfrak{R}_1\}$$

$$\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2) = \{((msg_2, pk, inst), (wit_2, l_2, x_2, aux_2)) \mid (msg_2, aux_2) = \mathsf{BlindEnc}_2(pk; l_2, x_2, msg_1) \wedge (inst, (wit_2, l_2, x_2)) \in \mathfrak{R}_2\}\,.$$

We show how to efficiently prove the relations $\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$ and $\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$ in a zero-knowledge proof by giving a $\aleph$ language statement in Listing 18 in Appendix C.4.

Theorem 1. *The joint ciphertext computation protocol (Listing 8) strongly emulates the ideal two-party computation protocol (Listing 7) for function* f_{JC}*:* $\mathcal{P}_{\mathsf{jcc}}(\mathfrak{R}_1, \mathfrak{R}_2) \leq \mathcal{I}_{\mathsf{tpc}}(f_{\mathsf{JC}}, \mathfrak{R}_1, \mathfrak{R}_2)$.

We refer to Appendix C.2.1 for the details on the construction of the BlindEnc and UnblindEnc protocols and to Appendix C.3 for the proof of Theorem 1.

13.4.1 Other Functionalities

Finally, we describe two additional functionalities for the IITM model: a common reference string $\mathcal{I}_{\mathsf{crs}}$ and a key registration functionality $\mathcal{I}_{\mathsf{reg}}$.

Listing 9. Functionality $\mathcal{I}_{\mathsf{crs}}(\mathcal{R}, \{D_k\}_{k\in\mathbb{N}})$

Tapes: $\{\mathsf{crs}.R \leftrightarrow\!\!\!\!\rightarrow \overline{\mathsf{crs}}.R\}_{R\in\mathcal{R}}$
Initialization: $params \leftarrow \varepsilon$.
Compute:
On (`GetParams`) on $\mathcal{I}_{\mathsf{crs}}.R$, $R \in \mathcal{R}$
- if $params = \varepsilon$ sample $params \leftarrow D_k$
- send (`Params`, $params$) to $\mathcal{I}_{\mathsf{crs}}.R$

Listing 10. Functionality $\mathcal{I}_{\mathsf{reg}}(\mathcal{R})$

Tapes: $\{\mathsf{reg}.\mathsf{R} \leftrightarrow\!\!\!\!\rightarrow \overline{\mathsf{reg}}.\mathsf{R}\}_{\in\mathcal{R}}$
Initialization: $state \leftarrow \varepsilon$.
Compute:
On $(\texttt{Register}, v)$ from $\mathsf{reg}.\mathsf{R} \in \mathcal{R}$
- Records the value (R, v)

On $(\texttt{Retrieve}, \mathsf{R})$ from $\mathsf{reg}.\mathsf{R}' \in \mathcal{R}$
- If (R, v) is recorded then return $(\texttt{Retrieve}, v)$ to $\mathsf{reg}.\mathsf{R}'$
- Otherwise send $(\texttt{Retrieve}, \perp)$ to $\mathsf{reg}.\mathsf{R}'$

Chapter 14

Oblivious Trusted Third Parties

14.1 Introduction

Anonymous credentials allow to implement electronic transactions that are unlinkable and selectively disclose the minimal amount of information about the user. At the same time these transactions have to be accountable. When using anonymous credentials, transactions are automatically accountable in the sense that the verifier is ensured that what is being proved during the credential show, is indeed vouched for by the issuer. However, many real-life applications have to consider exceptional cases in which additional information is required in case of a malicious transaction.

When the conditions for detecting such abuse can be expressed mathematically and can be detected inside of the electronic system, one can often mitigate such malicious transactions cryptographically. Examples for such transactions are e-cash systems that can resist double spending and money laundering [CFN90, CHK+06], as well as the ePetition system described in Chapter 4.

In other situations, e.g., when a suspect might have used an anonymous credential to get physical access to a crime scene, the evidence for allowing to recover additional information (e.g., the identity of all users that accessed the premise during a certain time period), lies outside of the system. The most simple solution is to reveal a verifiable encryption of this information during the credential show.

In particular, a user U would encrypt her true identity with the public key of the anonymity revocation authority RA, a kind of trusted third party (TTP) and provides

this encrypted data to a service provider SP. She then convinces SP in a zero-knowledge proof of knowledge that this encrypted data contains her valid user identity that can be opened by the authority RA if it decides that the opening request is legitimate.

This solution, however, raises several concerns:

- It involves a fully trusted party, the revocation authority, that is able to link all transactions with no graceful degradation of privacy and security, should the revocation authority become compromised.
- Additionally, the solution does not provide the best achievable accountability properties, as especially powerful users could bribe or threaten the RA such that it would refuse to open particular ciphertexts.
- Honest service providers find the traditional system cumbersome because of the need to involve such highly trusted authorities for even minor dispute cases. For example, to bring a case to law enforcement in the real world is likely to have a non-trivial cost, both in the time required, and in support from legal counsel.

There are two avenues that can be followed to reduce the trust into a trusted third party like the revocation authority. One is to distribute the TTP such that it does not run on a single machine but on multiple machines. Each machine is owned by an organization that is unlikely to collaborate with the other organizations against the user (e.g., a privacy office, the interior ministry, and the justice ministry). The cryptographic protocol that replaces the TTP guarantees that as long as one of these multiple machines is uncompromised and operates correctly, the other machines cannot infringe the user's privacy.

Oblivious Anonymity Revocation. The other approach that we apply here is to design the protocol in such a way that the TTP is as oblivious as possible to the task it performs, e.g., it does not know which user's identity it helps to reveal: in our implementation the identity of the user would be protected by two layers of encryption. The revocation authority can only remove the outer layer of the encryption. The second layer is removed by the service provider it self once it receives the partial decryption from the revocation authority.

This Oblivious Trusted Third Parties (OTP) mechanism helps to achieve some amount of graceful degradation. Even if the revocation authority is compromised, it cannot learn any useful information. Here, we assume that there is no collaboration between the service provider and the revocation authority.

Another aspect in which the revocation authority can be made oblivious, is in terms of the information it receives from the service provider. We want to make sure that the

original ciphertexts are labeled with the revocation condition but are otherwise only known to the service provider, i.e., they look random to all possible collusions between users and the revocation authority. This guarantees that powerful users with special interests have no way of influencing the revocation authority to selectively open only some of the opening requests.

In contrast to the fully trusted third party as discussed above, this scheme alleviates the trust assumptions on the TTP, and provides both stronger privacy and stronger accountability. The OTP revocation authority is a weaker TTP, whose only trust requirement is to revoke the anonymity of users only in those situations in which the revocation condition indeed holds. To achieve this, the scheme restricts the revocation authority to only process blinded information, unknown to users, and to output blinded information that can only be decrypted by the service provider.

As a result, RA cannot block requests of SP selectively and cannot collude against any specific user, nor can it link the transactions of users in the system. Furthermore, a compromised authority remains restricted in the information it could possibly gather, i.e., it can only gather information if the service provider of a particular transaction consents to remove the remaining blinding.

Essentially, oblivious anonymity revocation resolves most of our concerns stated above. Nevertheless, in many scenarios, the cost of proving that a request for anonymity revocation is legitimate, is not proportional with the compensation that the service provider gets.

A simple example is the following: to use a service, an anonymous user has to pay a small fee within 30 days. If the user, however, failed to do this, the service provider has to prove the non-payment towards the revocation authority in order to obtain the user's identity and take action. Distributing the revocation authority across multiple machines owned by different organizations does not solve this problem, on the contrary, now all of these organizations have to check non-payment which further increases the costs for the service provider.

Oblivious Satisfaction Authority In scenarios similar to the aforementioned example, it is often easier for the user to prove satisfaction, than for the service provider to do the opposite. Therefore, we shift some responsibilities from the service provider towards the user. Instead of the service provider having to prove to the revocation authority that the revocation conditions have been met, it is the user's responsibility to prove that the satisfaction conditions have been fulfilled! This change facilitates a far less complicated resolution of disputes and conflicts, which is both more economical for the service provider and more privacy-preserving for the user.

The approach is as follows: upon the user's request, an Oblivious Satisfaction Authority (SA) verifies the satisfaction of some condition with respect to a specific

service, and provides the user with a satisfaction token. The satisfaction authority can be made oblivious in the sense that SA must not be able to link a user's satisfaction transaction with the user's transaction at the service provider. Moreover, even if the oblivious satisfaction authority and the oblivious revocation authority collude, they should not be able to link satisfaction requests with opening requests. This is achieved in a similar way as for the oblivious RA, the satisfaction token is in fact double encrypted, and the satisfaction authority is only able to remove the outer layer, while only the user is able to remove the final blinding.

After unblinding the satisfaction token received from SA, the user publishes this token, proving satisfaction towards the revocation authority. Namely, when the service provider requests the user's identity, he has to provide the same satisfaction token to the revocation authority. Now, the revocation authority only discloses the (blinded) identity to the service provider if the corresponding satisfaction token has *not* been published before some predefined date. If the user, however, decides not to fulfill the contract, and as such cannot publish the corresponding satisfaction tokens, the revocation authority discloses the blinded user's identity towards the service provider.

Since the satisfaction tokens can be machine verified, the involvement of the revocation authority can be reduced significantly and expensive external authorities such as law enforcement become obsolete. This combined approach with oblivious revocation and oblivious satisfaction authorities, better serves the needs of service providers as it keeps the process of revocation and the dependency on external revocation authorities minimal. Furthermore, it provides better privacy guarantees towards the user than the solution with a fully trusted revocation authority.

To achieve this, the scheme restricts the revocation authority to only process blinded information, unknown to service providers, and to output blinded information that can only be decrypted by the user. As a result, SA cannot block requests of U selectively even when under pressure by the service provider and it cannot collude against any specific user, nor can it link the transactions of users in the system.

These strong guarantees do not only protect the user, but they also simplify privacy-friendly transactions. In particular, we can implement a form of anonymous payment based on credit cards rather than anonymous e-cash. When satisfying the payment condition towards the satisfaction authority the user is identified (through her credit card number), however, because of the unlinkability guarantee, her transaction with the service provider remains anonymous.

Camenisch et al. [CGHB08] were the first to propose this concept of *oblivious trusted third parties (OTP)*. Unfortunately, the authors only provide a high-level construction for such a protocol but do not present a concrete instantiation. Their construction has a number of limitations, e.g., the TTP is required to be online during user enrollment, and in fact it is unclear whether a full realization of their ambitious program is possible

along the lines they propose. In particular, our realization [CHK+11a] relies crucially on the CCA security of the encryption scheme, as the TTP essentially acts as a decryption oracle.

14.2 Modeling Oblivious Third Parties

We now formally model the OTP system that involves both an oblivious satisfaction authority and an oblivious revocation authority. In our example scenario, after a service enrollment between a user U and a service provider SP, the user ought to make a payment for the service before t_{due}. Upon request, the satisfaction authority SA checks that the user indeed made the payment and provides the user with a blinded transaction token. The user unblinds the token and publishes it to prove the satisfaction of the payment. Finally, the revocation authority RA reveals the user's identity to the service provider if no payment has been made before the payment deadline (i.e., no token corresponding to the enrollment was published).

We model the security and privacy requirements of such a system with the help of an ideal functionality $\mathcal{F}_{\mathsf{otp}}$. As usual, corruption is modeled via dummies $\mathcal{D}_{\mathsf{U}}$, $\mathcal{D}_{\mathsf{SP}}$, $\mathcal{D}_{\mathsf{SA}}$, $\mathcal{D}_{\mathsf{RA}}$ that allow to access the functionality both over the environment interface (before corruption) and the network interface (after corruption).

The Ideal System $\mathcal{I}_{\mathsf{otp}}$. The ideal system $\mathcal{I}_{\mathsf{otp}}$ is depicted in Fig. 14.1 and consists of the ideal functionality connected to the dummy parties over delayed communication tapes.

The system exports an environment interface named otp with roles $\mathcal{R} := \{\mathsf{U}, \mathsf{SP}, \mathsf{SA}, \mathsf{RA}\}$ and a network interface named notp_1 with roles $\mathcal{R} \cup \{\mathsf{C}_\mathsf{R}\}_{\mathsf{R}\in\mathcal{R}}$. Roles C_R are for the delays on the channel, while roles $\mathsf{U}, \mathsf{SP}, \mathsf{SA}, \mathsf{RA}$ allow to corrupt dummy parties and remotely control their behavior.

Listing 11 specifies the reactive behavior of $\mathcal{F}_{\mathsf{otp}}$. A user that can prove her identity with the help of a witness such that $(inst, (id, wit)) \in \mathfrak{R}$, is allowed to enroll. In particular, this interface supports the case where *wit* and *inst* are the secrets and the public key of a CL-signature [CL03] on the user's identity, i.e., an anonymous credential [CL01, BCKL08], or they are the opening and a commitment to the user's identity, i.e., a pseudonym [CL01]. For all these cases, the relation $\mathfrak{R}$ is tractable (i.e., there exists an efficient universally composable proof of knowledge).

Enrollment consists of three rounds. The first round commits the user to her identity. The second round provides the user with a random satisfaction label with respect to which she can satisfy the condition, e.g., make the necessary payment. In this

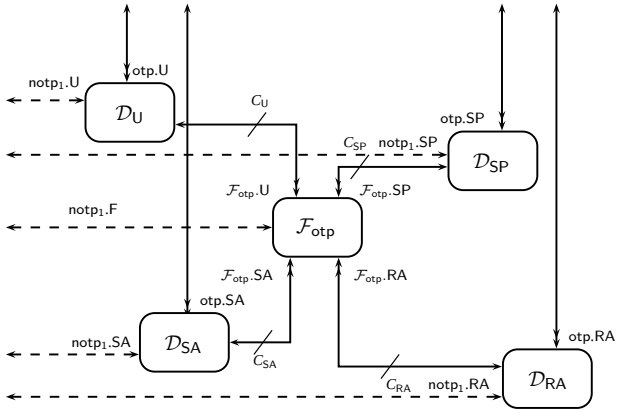


Figure 14.1: The Ideal OTP system $\mathcal{I}_{\mathsf{otp}} = \mathcal{D}_{\mathsf{U}}|\mathcal{F}_{\mathsf{otp}}|\mathcal{D}_{\mathsf{SP}}|\mathcal{D}_{\mathsf{SA}}|\mathcal{D}_{\mathsf{RA}}$.

round the user is also made aware of the due date t_{due} for the payment. Note that the user has to check that t_{due} fulfills reasonable uniformity constraints to protect her privacy. The last round gives the service provider the possibility to ask the identity revocation authority for the user's identity. As a common limitation with other escrow mechanisms for anonymous credentials, we cannot extract the identity itself, but only the image of a bijection of it. We model this by giving the simulator the possibility to choose the bijection. As the identity space of realistic systems is small enough to allow for exhaustive search, this is not a serious limitation.

The client interface towards the ideal oblivious parties, i.e., the interface of the user and the service provider respectively, consists of two messages `ReqAction` and `TestAction`, with $\texttt{Action} \in \{\texttt{Satisfy}, \texttt{Open}\}$. The obliviousness requirement guarantees that oblivious parties do not learn anything about the transactions of their clients. Indeed the decision of an oblivious party cannot be influenced in a transaction specific way, even if the other transaction participant colludes with the oblivious party. This is modeled with the help of test requests that are not related to any transaction. As these requests are indistinguishable from real requests, they allow the user to check whether the oblivious party indeed operates as required. [1]

Consequently, the decision of an oblivious party can only depend on explicit and relevant information. For *satisfaction*, this is the user known satisfaction label L with respect to which she makes her payment. For the *opening*, it is the transaction token T that is secret until after satisfaction, when it is learned by the user. We abstract from the way through which users make T available to the revocation authority, but envision

[1] An extension that allows not only the requester, but arbitrary external parties, e.g. an auditor, to make test requests is a useful and cryptographically straightforward extension to this interface.

some kind of anonymous publicly available bulletin board. It is in the responsibility of the user to make the token, learned during satisfaction, available to RA, and in the responsibility of RA to check its existence. All the protocol guarantees is that RA learns the same T value during *opening* as the user learned during *satisfaction*.

Listing 11. Functionality $\mathcal{F}_{\mathsf{otp}}$

Tapes: see Fig. 14.1
Initialization: $state \leftarrow$ "ready"; $L, T, id, \widehat{T}, \widehat{id}, F, \mathbb{T}, \mathbb{L}, t_{due} \leftarrow \varepsilon$.
Compute:

On $(\texttt{SetF}, F', \mathbb{T}', \mathbb{L}')$ from $\texttt{notp}_1$.F where $state =$ "ready":

- abort if F' is not an efficient bijection or $\mathbb{T}'$ or $\mathbb{L}'$ are not of sufficient size; set $F \leftarrow F'$, $\mathbb{T} \leftarrow \mathbb{T}'$, and $\mathbb{L} \leftarrow \mathbb{L}'$

On $(\texttt{EnrollU}, inst, (id', wit'))$ from $\mathcal{F}_{\mathsf{otp}}$.U where $state =$ "ready":

- if $(inst, (id', wit')) \notin \mathfrak{R}$) abort;
- set $state \leftarrow$ "enrollu"; set $id \leftarrow id'$; send $(\texttt{EnrollU}, inst)$ to $\mathcal{F}_{\mathsf{otp}}$.SP

On $(\texttt{DeliverEnrollU}, {t_{due}}')$ from $\mathcal{F}_{\mathsf{otp}}$.SP where $state =$ "enrollu":

- set $t_{due} \leftarrow {t_{due}}'$; set T, L to random values from $\mathbb{T}$ and $\mathbb{L}$ respectively;
- set $state \leftarrow$ "deliverenrollu"; send $(\texttt{DeliverEnrollU}, L, t_{due})$ to $\mathcal{F}_{\mathsf{otp}}$.U

On $(\texttt{DeliverEnrollSP})$ from $\mathcal{F}_{\mathsf{otp}}$.U where $state =$ "deliverenrollu":

- set $state \leftarrow$ "enrolled"; send $(\texttt{DeliverEnrollSP})$ to $\mathcal{F}_{\mathsf{otp}}$.SP

On $(\texttt{ReqSatisfy})$ from $\mathcal{F}_{\mathsf{otp}}$.U where $L \neq \varepsilon$ and $\widehat{T} = \varepsilon$:

- set $\widehat{T} \leftarrow T$; send $(\texttt{ReqSatisfy}, L)$ to $\mathcal{F}_{\mathsf{otp}}$.SA.

On $(\texttt{TestSatisfy}, L', T')$ from $\mathcal{F}_{\mathsf{otp}}$.U where $\widehat{T} = \varepsilon$:

- set $\widehat{T} \leftarrow T'$; send $(\texttt{ReqSatisfy}, L')$ to $\mathcal{F}_{\mathsf{otp}}$.SA

On $(\texttt{Satisfy}, satisfied)$ from $\mathcal{F}_{\mathsf{otp}}$.SA where $\widehat{T} \neq \varepsilon$:

- if *satisfied*, set $m \leftarrow (\texttt{Satisfy}, \widehat{T})$, otherwise set $m \leftarrow (\texttt{Satisfy}, \varepsilon)$; set $\widehat{T} \leftarrow \varepsilon$; send m to $\mathcal{F}_{\mathsf{otp}}$.U

On $(\texttt{ReqOpen})$ from $\mathcal{F}_{\mathsf{otp}}$.SP where $state =$ "enrolled" and $\widehat{id} = \varepsilon$:

- set $\widehat{id} \leftarrow id$; send $(\texttt{ReqOpen}, T, t_{due})$ to $\mathcal{F}_{\mathsf{otp}}$.RA

On $(\texttt{TestOpen}, T', id', {t_{due}}')$ from $\mathcal{F}_{\mathsf{otp}}$.SP where $\widehat{id} = \varepsilon$:

- set $\widehat{id} \leftarrow id'$; send $(\texttt{ReqOpen}, T', {t_{due}}')$ to $\mathcal{F}_{\mathsf{otp}}$.RA

On $(\texttt{Open}, open)$ from $\mathcal{F}_{\mathsf{otp}}$.RA where $\widehat{id} \neq \varepsilon$:

- if *open*, set $m \leftarrow (\texttt{Open}, F(\widehat{id}))$, otherwise set $m \leftarrow (\texttt{Open}, \varepsilon)$; set $\widehat{id} \leftarrow \varepsilon$; send m to $\mathcal{F}_{\mathsf{otp}}$.SP

14.3 Implementing Oblivious Third Parties

In this section, we present an implementation of the oblivous third parties scheme based on the model above. First, we present an outline of the protocol, followed by a more detailed discussion.

To construct a protocol that securely emulates the above functionality we make essential use of (adaptive chosen-ciphertext attack secure) encryption. As depicted in Fig. 14.2 the protocol makes use of several cryptographic building blocks. But at the core of the protocol are two joint-ciphertext computations, that, as described in the previous chapter, can be efficiently realized thanks to structure preserving encryption [CHK$^+$11a].

The enrollment protocol has a few more communication rounds, because of the zero-knowledge proofs, but otherwise closely follows the three phases of the ideal system. In the first phase the user commits to and proves her identity. Both the user and the service provider commit to randomness that they will use to jointly compute the transaction token T. The user proves knowledge of the opening of her commitment as part of the joint computation of the satisfaction ciphertext $ct_1 \leftarrow \mathsf{Enc}(pk_{\mathsf{SA}}, L, T \cdot g^r)$. In the second phase, the service provider transfers t_{due}, completes the joint ciphertext computation, and starts the computation of the revocation ciphertext $ct_2 \leftarrow \mathsf{Enc}(pk_{\mathsf{RA}}, g^{t_{due}}, (g^{id+r'}, T))$. In both cases, he proves knowledge of the opening to his commitment to guarantee that the transaction token is embedded correctly into both ciphertexts. The user outputs the label of ct_1 as the random satisfaction label L. In the last phase the user again proves knowledge of openings for her commitments in the computation of ct_2 to guarantee that it contains the transaction token T and a blinded user identity g^{id} under label $g^{t_{due}}$.

To satisfy her financial obligations, the user makes a payment with respect to label L and then asks the satisfaction authority to decrypt ct_1. The user receives the blinded transaction token, that she unblinds using her locally stored randomness to learn T. She makes T available to the revocation authority, through some out-of-band anonymous bulletin board mechanism. Test satisfaction requests are just encryptions of blinded T' under label L'. To request the opening of a user identity, the service provider sends the ciphertext ct_2 to the revocation authority, which checks the label $L' = g^{t_{due}}$, decrypts the ciphertext to learn T and verifies whether T was posted by the user. If not, the revocation authority returns the blinded identity $g^{id+r'}$ to the service provider, which can unblind the identity. Test opening requests are just encryptions of T' and blinded $g^{id'}$ under label $g^{t_{due}'}$.

The Real System $\mathcal{P}_{\mathsf{otp}}$. The real protocol $\mathcal{P}_{\mathsf{otp}}$ implements the same API interface as $\mathcal{I}_{\mathsf{otp}}$ (see Fig. 14.2), but is realized as a distributed cryptographic protocol with

parties U, SP, SA, and RA each with their corresponding pairs of API tapes otp.U, otp.SP, otp.SA, otp.RA, towards the environment.

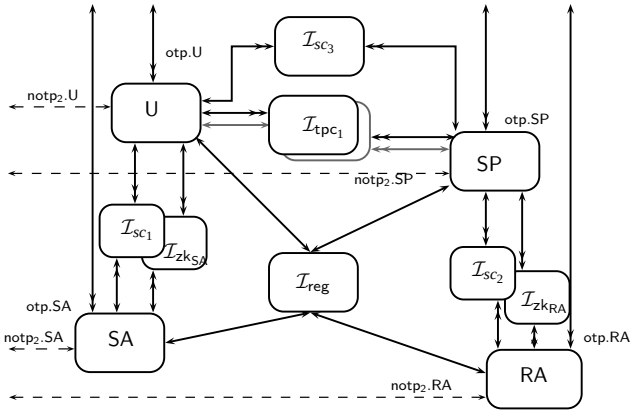


Figure 14.2: $\mathcal{P}_{\mathsf{otp}} = \mathsf{SA}|\mathcal{I}_{sc_1}|\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}|\mathsf{U}|\mathcal{I}_{sc_3}|\mathcal{I}_{\mathsf{tpc}_1}|\mathcal{I}_{\mathsf{tpc}_2}|\mathsf{SP}|\mathcal{I}_{sc_2}|\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}|\mathsf{RA}|\mathcal{I}_{\mathsf{reg}}$, the Real OTP System: The Realization Makes Use of Ideal Resources $\mathcal{I}_{sc_i}$, $\mathcal{I}_{\mathsf{zk}_{\mathsf{R}}}$, $\mathcal{I}_{\mathsf{reg}}$, $\mathcal{I}_{\mathsf{jcc}_i}$ for Secure Communication, Proofs of Knowledge, Key Registration, and Joint Ciphertext Computation Respectively.

The core security guarantees are achieved through the use of secure two-party computation, secure communication, and zero-knowledge proof protocols. We model secure communication through ideal systems $\mathcal{I}_{sc_1}$, $\mathcal{I}_{sc_2}$ and $\mathcal{I}_{sc_3}$, zero-knowledge proofs through $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$, $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$, and two-party computation through $\mathcal{I}_{\mathsf{tpc}_1}$, $\mathcal{I}_{\mathsf{tpc}_2}$, which are instances of respectively $\mathcal{I}_{sc}$, $\mathcal{I}_{\mathsf{zk}}$, and $\mathcal{I}_{\mathsf{tpc}}$ with renamed tapes. Like in the ideal system, we model corruption via an adversarial interface to the protocol parties U, SP, SA, RA that allows to control corrupted parties. During initialization, protocol parties SA and RA register public keys pk_{SA} and pk_{RA} with a key registration authority $\mathcal{I}_{\mathsf{reg}}$.

The real protocol has a few more rounds but follows the same three phases as the ideal system. In the first phase the user commits to and proves her identity. Both the user and service provider commit to the randomness that they will use to compute the revocation token T in commitments $Com_{x_1'}$ and Com_{x_2}. The user proves knowledge of the opening of $Com_{x_1'}$ as part of the joint computation of the satisfaction ciphertext ct_1. In the second phase, the service provider transfers t_{due}, completes the joint ciphertext computation, and proves that his contribution to the blinded revocation token corresponds to the value in Com_{x_2}. The user outputs the label of this ciphertext as her random satisfaction label. The last phase does a joint ciphertext computation of the revocation token T and the user's identity g^{id} under label $g^{t_{due}}$.

Listing 12. Protocol $\mathcal{P}_{\mathsf{otp}} = \mathsf{SA}|\mathcal{I}_{sc_1}|\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}|\mathsf{U}|\mathcal{I}_{sc_3}|\mathcal{I}_{\mathsf{tpc}_1}|\mathcal{I}_{\mathsf{tpc}_2}|\mathsf{SP}|\mathcal{I}_{sc_2}|\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}|\mathsf{RA}|\mathcal{I}_{\mathsf{reg}}$

Tapes: See Fig. 14.2

Initilization: Upon initialization SA and RA generate keys $(sk_{\mathsf{SA}}, pk_{\mathsf{SA}})$ and $(sk_{\mathsf{RA}}, pk_{\mathsf{RA}})$ for the structure preserving encryption scheme and register pk_{SA} and pk_{RA} with $\mathcal{I}_{\mathsf{reg}}$. U and SP retrieve these keys on demand.

Compute:

On $(\texttt{EnrollU}, (id, wit), inst)$ from otp.U:

- if $(inst, (id, wit)) \notin \mathfrak{R}$), U aborts, else U generates commitment parameters $params_{\mathsf{U}}$ and sends them to SP over $\mathcal{I}_{sc_3}$.
- SP receives $params_{\mathsf{U}}$, generates random x_2 and $open_{x_2}$, computes $(Com_{x_2}) \leftarrow \mathsf{Commit}(params_{\mathsf{U}}; x_2, open_{x_2})$, generates commitment parameters $params_{\mathsf{SP}}$, and sends Com_{x_2}, $params_{\mathsf{SP}}$ to U over $\mathcal{I}_{sc_3}$.
- U receives Com_{x_2} and $params_{\mathsf{SP}}$, generates random $open_{id}$, x'_1 and $open_{x'_1}$, computes $(Com_{id}) \leftarrow \mathsf{Commit}(params_{\mathsf{SP}}; id, open_{id})$, $(Com_{x'_1}) \leftarrow \mathsf{Commit}(params_{\mathsf{SP}}; x'_1, open_{x'_1})$ and sends Com_{id}, $Com_{x'_1}$ to SP over $\mathcal{I}_{sc_3}$.
- SP receives Com_{id}, $Com_{x'_1}$ and sends an acknowledgement over $\mathcal{I}_{sc_3}$.
- U generates random x_1 and l_1 and sends $(\texttt{Input}_1, (inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{SA}}, (id, wit, open_{id}, x'_1, open_{x'_1}), (l_1, x_1))$ to $\mathcal{I}_{\mathsf{tpc}_1}.\mathsf{P}_1$.
- SP receives $(\texttt{Input}_1, (inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{SA}})$ on $\mathcal{I}_{\mathsf{tpc}_1}.\mathsf{P}_2$ and sends $(\texttt{EnrollU}, inst)$ to otp.SP.

On $(\texttt{DeliverEnrollU}, t_{due})$ from otp.SP:

- SP sends t_{due} over $\mathcal{I}_{sc_3}$ and U replies with an acknowledgment.
- SP generates random l_2 and sends $(\texttt{Input}_2, (open_{x_2}, (l_2, x_2)))$ to $\mathcal{I}_{\mathsf{tpc}_1}.\mathsf{P}_2$.
- U receives $(ct_1) \leftarrow \mathsf{Enc}(pk_{\mathsf{SA}}, g^{l_1+l_2}, (g^{x_1} \cdot g^{x_2}))$ and $(L) \leftarrow g^{l_1+l_2}$ from $\mathcal{I}_{\mathsf{tpc}_1}.\mathsf{P}_1$ and sends $(\texttt{DeliverEnrollU}, L, t_{due})$ to otp.U.

On $(\texttt{DeliverEnrollSP})$ from otp.U:

- U sends an acknowledgment to SP over $\mathcal{I}_{sc_3}$ and SP generates random x'_2 and sends $(\texttt{Input}_1, (params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}, T_{due}), pk_{\mathsf{RA}}, open_{x_2}, (t_{due}, x_2, x'_2))$ to $\mathcal{I}_{\mathsf{tpc}_2}.\mathsf{P}_1$.
- U receives $(\texttt{Input}_1, (params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{RA}})$ from $\mathcal{I}_{\mathsf{tpc}_2}.\mathsf{P}_2$ and sends $(\texttt{Input}_2, (open_{id}, open_{x'_1}), (0, x'_1, id))$ to $\mathcal{I}_{\mathsf{tpc}_2}.\mathsf{P}_2$.
- SP receives ct_2 with $(ct_2) \leftarrow \mathsf{Enc}(pk_{\mathsf{RA}}, g^{t_{due}}, (g^{x'_1+x_2}, g^{id+x'_2}))$ from $\mathcal{I}_{\mathsf{tpc}_2}.\mathsf{P}_1$ and sends (DeliverEnrollSP) to otp.SP.

On $(\texttt{ReqSatisfy})$ from otp.U where $L \neq \varepsilon$:

- U sends (ct_1, L) to SA over $\mathcal{I}_{sc_1}$.
- SA receives ct_1 from $\mathcal{I}_{sc_1}$ and if the ciphertext with label L validates correctly, SA sends $(\texttt{ReqSatisfy}, L)$ to otp.SA.

On $(\texttt{TestSatisfy}, \widehat{L}, \widehat{T})$ from otp.U:

- U generates a new ciphertext $(\widehat{ct}_1) \leftarrow \mathsf{Enc}(pk_{\mathsf{SA}}, \widehat{L}, (\widehat{T} \cdot g^{x_1 - x_1'}))$ with random x_1 and x_1' and sends $(\widehat{ct}_1, \widehat{L})$ to SA over $\mathcal{I}_{sc_1}$.
- SA receives $\widehat{ct}_1$ from $\mathcal{I}_{sc_1}$ and if the ciphertext with label $\widehat{L}$ validates correctly, SA sends $(\texttt{ReqSatisfy}, \widehat{L})$ to otp.SA.

On $(\texttt{Satisfy}, \textit{satisfied})$ from otp.SA

- SA skips a communication round for $\mathcal{I}_{sc_1}$.
- if *satisfied*, SA decrypts ct_1 and proves correct decryption of the blinded token $(m) \leftarrow \mathsf{Dec}(L, ct_1, sk_{\mathsf{SA}})$ to U using $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$. Otherwise, SA proves $m = \varepsilon$ with an otherwise random instance and witness of correct size to U using $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$.
- U receives m' as the instance of $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$.
- if $m' \neq \varepsilon$, U unblinds $T \leftarrow m' \cdot g^{x_1' - x_1} = g^{x_1' + x_2}$ and sends $(\texttt{Satisfy}, T)$ to otp.U; otherwise U sends $(\texttt{Satisfy}, \varepsilon)$ to otp.U.

On $(\texttt{ReqOpen})$ from otp.SP where *state* = "enrolled":

- SP sends (ct_2, t_{due}) to RA over $\mathcal{I}_{sc_3}$.
- RA receives (ct_2, t_{due}) from $\mathcal{I}_{sc_3}$, decrypts ct_2 under label $g^{t_{due}}$ into $(T, g^{id + x_2'})$, it sends $(\texttt{ReqOpen}, T)$ to otp.RA.

On $(\texttt{TestOpen}, \widehat{T}, \widehat{id}, \widehat{t_{due}})$ from otp.SP:

- SP generates ciphertext $(\widehat{ct}_2) \leftarrow \mathsf{Enc}(pk_{\mathsf{RA}}, g^{t_{due}}, (\widehat{T}, g^{\widehat{id} + x_2'}))$ with random x_2' and sends $(\widehat{ct}_2, \widehat{t_{due}})$ to RA over $\mathcal{I}_{sc_2}$.
- RA receives $(\widehat{ct}_2, \widehat{t_{due}})$ from $\mathcal{I}_{sc_2}$, decrypts the ciphertext under label $g^{\widehat{t_{due}}}$ into $(\widehat{T}, m)$ and sends $(\texttt{ReqOpen}, \widehat{T})$ to otp.SP.

On $(\texttt{Open}, \textit{open})$ from otp.RA:

- RA skips a communication round for $\mathcal{I}_{sc_2}$.
- if *open*, RA proves correct decryption of the blinded identity m to SP using $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$. otherwise, RA proves $m = \varepsilon$ with an otherwise random instance and witness of correct size to SP using $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$;
- SP receives $(ct_1, pk_{\mathsf{SA}}, L, m')$ as the instance of $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$.
- if $m' \neq \varepsilon$, SP unblinds $ID \leftarrow g^{id + x_2'} \cdot g^{-x_2'} = g^{id}$ and sends $(\texttt{Open}, ID)$ to otp.SP; otherwise it sends $(\texttt{Open}, \varepsilon)$ to otp.SP.

The two-party computation $\mathcal{I}_{\mathsf{tpc}_1} = \mathcal{I}_{\mathsf{tpc}}(f_{\mathsf{JC}_1}(pk_{\mathsf{SA}}, (l_1, x_1), (l_2, x_2)), \mathfrak{R}_{1,1}, \mathfrak{R}_{1,2})$ is parameterized by the function f_{JC_1} and two relations $\mathfrak{R}_{1,1}$ and $\mathfrak{R}_{1,2}$ for computing the satisfaction ciphertext ct_1 that contains an encryption of $g^{x_1 + x_2}$ under a jointly

chosen label $L = g^{l_1+l_2}$:

$$\begin{aligned}\mathfrak{R}_{1,1} = \{((&inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), (id, wit, open_{id}, x'_1,\\ &open_{x'_1}, l_1, x_1)) \mid (inst, (id, wit)) \in \mathfrak{R} \wedge\\ &(Com_{id}) = \mathsf{Commit}(params_{\mathsf{SP}}, id, open_{id}) \wedge\\ &(Com_{x'_1}) = \mathsf{Commit}(params_{\mathsf{SP}}; x'_1, open_{x'_1})\}\\ \mathfrak{R}_{1,2} = \{(&inst, params_{\mathsf{U}}; params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), (open_{x_2}, l_2, x_2)) \mid\\ &(Com_{x_2}) = \mathsf{Commit}(params_{\mathsf{U}}; x_2, open_{x_2})\}\end{aligned}$$

Similarly, the two-party computation $\mathcal{I}_{\mathsf{tpc}_2} = \mathcal{I}_{\mathsf{tpc}}(f_{\mathsf{JC}_2}(pk_{\mathsf{RA}}, (\varepsilon, (x'_1, id)), (t_{due}, (x_2, x'_2)), \mathfrak{R}_{1,1}, \mathfrak{R}_{1,2})$ is parameterized by the function f_{JC_2} and relations $\mathfrak{R}_{1,1}, \mathfrak{R}_{1,2}$ for computing the identity ciphertext ct_2 that contains an encryption of $(g^{x'_1+x_2}, g^{id+x'_2})$ under key pk_{RA} with public label $g^{t_{due}}$:

$$\begin{aligned}\mathfrak{R}_{2,1} = \{((&params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}, T_{due}), (open_{x_2}, t_{due},\\ &x_2, x'_2)) \mid (Com_{x_2}) = \mathsf{Commit}(params_{\mathsf{U}}; x_2, open_{x_2}) \wedge T_{due} = g^{t_{due}}\}\ ,\\ \mathfrak{R}_{2,2} = \{((&params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}, T_{due}), (open_{id}, open_{x'_1},\\ &0, x'_1, id)) \mid (Com_{id}) = \mathsf{Commit}(params_{\mathsf{SP}}; id, open_{id}) \wedge\\ &(Com_{x'_1}) = \mathsf{Commit}(params_{\mathsf{SP}}; x'_1, open_{x'_1})\}\ .\end{aligned}$$

The commitment scheme can be realized as a simple Pedersen commitment. Given a tractable relation $\mathfrak{R}$ the relations $\mathfrak{R}_{1,1}$, $\mathfrak{R}_{1,2}$, $\mathfrak{R}_{2,1}$, and $\mathfrak{R}_{2,2}$ are themselves tractable.

Satisfaction and opening make use of proofs of correct decryption. In case SA or RA rejects a request by U and SP respectively, we abuse the functionality $\mathcal{I}_{\mathsf{zk}_i}$ as a secure channel, by proving a statement with an arbitrary instance, and witness. We assume that the instance is of the correct size to thwart traffic analysis. The relations $\mathfrak{R}_{\mathsf{SA}}$ and $\mathfrak{R}_{\mathsf{RA}}$ for proving correct decryption are defined as follows:

$$\begin{aligned}\mathfrak{R}_{\mathsf{SA}} &= \{((ct_1, pk_{\mathsf{SA}}, L, m), sk_{\mathsf{SA}}) \mid (m = \mathsf{Dec}(L, ct_1, sk_{\mathsf{SA}}) \wedge m \neq \varepsilon) \vee m = \varepsilon\}\ ,\\ \mathfrak{R}_{\mathsf{RA}} &= \{((ct_2, pk_{\mathsf{RA}}, g^{t_{due}}, m), (sk_{\mathsf{RA}}, T) \mid ((m, T) = \mathsf{Dec}(g^{t_{due}}, ct_2, sk_{\mathsf{RA}}) \wedge\\ &\qquad m \neq \varepsilon) \vee m = \varepsilon\}\ ,\end{aligned}$$

An efficient realization of $\mathcal{P}_{\mathsf{zk}_{\mathsf{SA}}} \leq \mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}(\mathfrak{R}_{\mathsf{SA}})$ and $\mathcal{P}_{\mathsf{zk}_{\mathsf{RA}}} \leq \mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}(\mathfrak{R}_{\mathsf{RA}})$ is presented in Appendix C.4.

Theorem 2. *Given the CCA security of the encryption scheme, our oblivious third party protocol (see Listing 12) strongly emulates the ideal oblivious third party system (see Listing 11):* $\mathcal{P}_{\mathsf{otp}}(\mathfrak{R}) \leq \mathcal{I}_{\mathsf{otp}}(\mathfrak{R})$.

A note on using the same group setup. The proofs of Sect. 13.4 can efficiently deal with different abelian groups. This means that we can compose tractable relations that make use of different group setups and still obtain a tractable relation. This, however, comes with a cost on the performance of the proofs. To achieve optimal performance, parties should use common group parameters as much as possible. Such group parameters need to exist both in the real world and the ideal world, so they can be used by the identity certification system for implementing the relation $(inst, (wit, id)) \in \mathfrak{R}$. Two ways of achieving this are: 1) to describe a deterministic procedure for deriving adequate pairing parameters based on the security parameter alone. 2) use a global setup (e.g., $\mathcal{I}_{\mathsf{crs}}$) that exists both in the real world and the ideal world, i.e., we prove $\mathcal{P}_{\mathsf{otp}}(\mathfrak{R})|\mathcal{I}_{\mathsf{crs}} \leq \mathcal{I}_{\mathsf{otp}}(\mathfrak{R})|\mathcal{I}_{\mathsf{crs}}$. Where $\mathcal{I}_{\mathsf{crs}}$ only provides a pairing setup. This can be seen as a variant of the GUC model [CDPW07]. We note, however, that this $\mathcal{I}_{\mathsf{crs}}$ does not allow us to overcome the impossibility results that have been shown for GUC. We still make use of UC common reference strings for the proofs of knowledge. We leave the construction of an OTP protocol based on an augmented common reference string as further work, but point to [DSW08] as a starting point.

Multi-session version of the protocol. In a realistic deployment, a large number of users will be interacting with a slightly smaller number of service providers, the latter needing to accept multiple enrollment transactions in parallel. Moreover, to achieve real unlinkability between the different transactions of a user, secure channels need to be replaced with secure anonymous channels. The latter require a separation between network identifiers and session identifiers. However, the multi-session functionalities $\underline{\mathcal{I}_{\mathsf{zk}}}$ and $\underline{\mathcal{I}_{\mathsf{tpc}}}$ do not provide anonymity and cannot be realized without $\underline{\mathcal{I}_{sc}}$ which outputs the same session id/address that it receives as input.

To see that a proof for the single session version of the OTP protocol is sufficient to guarantee the cryptographic property of the multi-session protocol with anonymous channels, we apply the split functionality theorem of [BCL$^+$05, CCGS10] that states that for every functionality realizable with authenticated/secure channels, there exists a corresponding split functionality that is realizable with split authenticated/secure channels. Intuitively in the split functionality it is the adversary that in a multi-session version controls which parties communicate together over which functionality. By applying the split functionality theorem and the composition theorem multiple times, a hybrid protocol with multiple split functionalities can be transformed into a protocol, that contains only split secure channels. After proving implicit session disjointness,

one can achieve a multi-session version of the OTP protocol that has only local session ids [KT11].

14.4 Proof of the Oblivious Third Party Protocol

To prove that $\mathcal{P}_{\mathsf{otp}}$ emulates $\mathcal{I}_{\mathsf{otp}}$ (Theorem 2), we need to prove existance of a simulator $\mathcal{S}$ that translates messages between the interfaces notp_1 and notp_2.

The simulator needs to do some trivial forwarding for every corrupted role R: it forwards all messages from the environment leaked through notp_1.R to notp_2.R; all messages from notp_2.R, addressed to the environment are forwarded to the corrupted party on notp_1.R. The simulator internally simulates most of the real world ideal functionalities to simulate delays and corruption of submodules. All messages addressed to another corrupted real world entity are forwarded to an internal simulation of that entity.

For ideal communication between honest roles, the simulator simply simulates the delays of the real communication internally based on the delays in the ideal communication. The simulator creates and registers the keys of honest SA and RA. After the keys of RA are registered, $\mathcal{S}$ sends $(\texttt{SetF}, F, \mathbb{G}, \mathbb{G})$ to $\mathcal{F}_{\mathsf{otp}}$ to set $F(id) = g^{id}$.

As we will see, the two most interesting cases of the simulation are when either the user or the service provider, but not both are corrupted. We cover the other corner cases first.

Listing 13. $\mathcal{S}$ if both user and service provider are corrupted

We only need to simulate for an honest SA or RA.

- Upon receiving (ct_1, L) from $\mathcal{I}_{sc_1}$, the simulator checks whether the ciphertext correctly decrypts under label L to some value m, picks a random T and sends $(\texttt{TestSatisfy}, L, T)$ to notp_1.U.
- Upon receiving $(\texttt{Satisfy}, \bot)$ or $(\texttt{Satisfy}, T)$, it skips a communication round for $\mathcal{I}_{sc_1}$ and either proves $m = \bot$ with an otherwise random instance and witness of correct size or $((ct_1, pk_{\mathsf{SA}}, L, m), sk_{\mathsf{SA}}) \in \mathfrak{R}_{\mathsf{SA}}$ respectively.
- Upon receiving (ct_2, t_{due}) from $\mathcal{I}_{sc_2}$, the simulator decrypts the ciphertext under label $g^{t_{due}}$ into (T, m), picks a random id, and sends $(\texttt{TestOpen}, T, id, t_{due})$ to notp_1.SP.
- Upon receiving $(\texttt{Open}, \bot)$ or $(\texttt{Open}, id)$, it skips a communication round for $\mathcal{I}_{sc_2}$ and either proves $m = \bot$ with an otherwise random instance and witness of correct size or $((ct_2, pk_{\mathsf{RA}}, g^{t_{due}}, m), (sk_{\mathsf{RA}}, T)) \in \mathfrak{R}_{\mathsf{RA}}$ respectively.

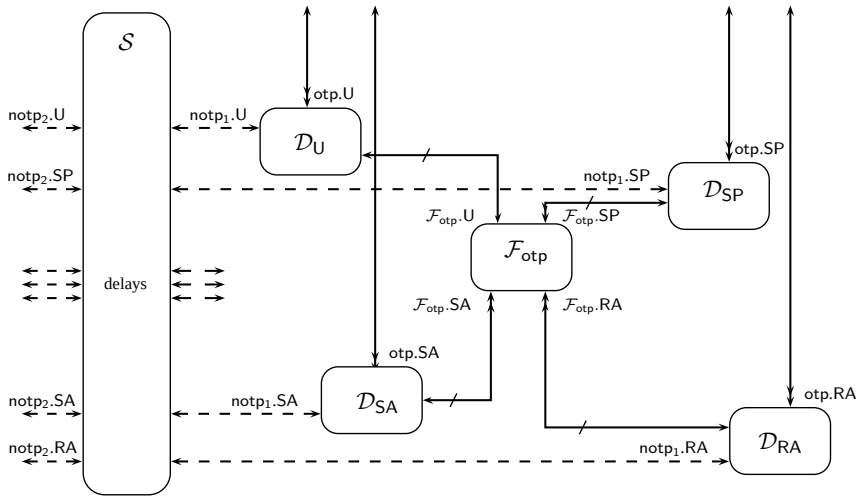


Figure 14.3: OTP Simulator.

Listing 14. $\mathcal{S}$ if both user and service provider are honest

We only need to simulate for a corrupted SA or RA.

- Upon receiving $(\texttt{ReqSatisfy}, L)$, the simulator picks a random message m and sends $(\mathsf{Enc}(pk_{\mathsf{SA}}, L, m), L)$ to $\mathcal{I}_{sc_1}$.
- When receiving $(\texttt{Prove}, \perp)$, it sends $(\texttt{Satisfy}, \mathit{false})$ to $\mathsf{notp}_1.\mathsf{SA}$.
- When receiving $(\texttt{Prove}, ct_1, pk_{\mathsf{SA}}, L, m)$, it sends $(\texttt{Satisfy}, \mathit{true})$.
- Upon receiving $(\texttt{ReqOpen}, T, t_{due})$, the simulator picks a random message m and send $\mathsf{Enc}(pk_{\mathsf{RA}}, g^{t_{due}}; T, m)$ to $\mathcal{I}_{sc_2}$.
- When receiving $(\texttt{Prove}, \perp)$, it sends $(\texttt{Open}, \mathit{false})$ to $\mathsf{notp}_1.\mathsf{RA}$. When receiving $(\texttt{Prove}, ct_2, pk_{\mathsf{RA}}, g^{t_{due}}, m)$ it sends $(\texttt{Open}, \mathit{true})$.

Because of the use of ideal functionalities, the simulation for all of these cases is perfect. We now consider a corrupted user and an honest service provider.

Listing 15. $\mathcal{S}$ if the user is corrupted, the service provider is honest

The simulator $\mathcal{S}$ sets $\mathit{state} \leftarrow$ "ready" and follows the instructions for SP of the real world protocol.

On input $\mathit{params}_{\mathsf{U}}$ on $\mathcal{I}_{sc_3}$ with $\mathit{state} =$ "ready";

- generate commitment parameters $\mathit{params}_{\mathsf{SP}}$ and $\mathit{Com}_{x_2} \leftarrow \mathsf{Commit}(\mathit{params}_{\mathsf{U}}, x_2, \mathit{open}_{x_2})$ on random open_{x_2}, and return both to $\mathcal{I}_{sc_3}$.

– wait for Com_{id} and $Com_{x'_1}$ over $\mathcal{I}_{sc_3}$ and reply with an acknowledgment.

– wait for $(\texttt{Input}_1, (inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{SA}}, (id, wit, open_{id}, x'_1, open_{x'_1}), (l_1, x_1))$ on $\mathcal{I}_{\mathsf{tpc}_1}.\mathsf{P}_1$, store id, wit, x'_1, x_1 and forward the message to the simulated $\mathcal{F}_{\mathsf{tpc}_1}.\mathsf{P}_1$.

– wait for $(\texttt{Input}_1, (inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{SA}})$ on $\mathcal{F}_{\mathsf{tpc}_1}.\mathsf{P}_2$, send $(\texttt{EnrollU}, inst, (id, wit))$ to $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$, and continue the delay on $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$.

– let $state \leftarrow$ "enrollu".

On a delay on $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ with $state =$ "enrollu":

– confirm the delay and wait for $(\texttt{DeliverEnrollU}, L, t_{due})$ from $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$.

– send t_{due} over $\mathcal{I}_{sc_3}$ to $\mathsf{notp}_2.\mathsf{U}$ and wait for an acknowledgment.

– if SA corrupted, send $(\texttt{ReqSatisfy}, L)$ to $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$, confirm satisfaction, learn $(\texttt{Satisfy}, T)$ and set $m \leftarrow T \cdot g^{x_1 - x'_1}$.

– otherwise set $m \leftarrow 1$.

– send $(\texttt{Input}_2, open_{x_2}, (l_2, x_2))$ to the simulated $\mathcal{I}_{\mathsf{tpc}_2}$, in which we set $ct_1 \leftarrow \mathsf{Enc}(pk_{\mathsf{SA}}, L, m)$ and store ct_1. Finally, set $state \leftarrow$ "deliverenrollu".

On the acknowledgment over $\mathcal{I}_{sc_3}$ with $state =$ "deliverenrollu"

– send $(\texttt{Input}_1, (params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}, T_{due}), pk_{\mathsf{RA}}, open_{x_2}, (t_{due}, x_2, x'_2))$ to $\mathcal{I}_{\mathsf{tpc}_2}.\mathsf{P}_1$.

– wait for $(\texttt{Input}_2, (\widehat{open}_{id}, \widehat{open}_{x'_1}), (0, x'_1, id))$ on $\mathsf{notp}_2.\mathsf{U}$, simulate $\mathcal{I}_{\mathsf{tpc}_2}$ resulting in the ciphertext ct_2.

– finally, set $state \leftarrow$ "enrolled" and send $(\texttt{DeliverEnrollSP})$ to $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$ with a confirmation on the delay of $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$.[2]

As the user is corrupted and the service provider is honest, no extra simulation is needed for a corrupted SA, or a honest RA. If SA is honest, we have to handle satisfaction requests.[3]

On $(\widehat{ct}_1, \widehat{L})$ on $\mathsf{notp}_2.\mathsf{U}$ with $state =$ "enrolled";

– simulate $\mathcal{I}_{sc_1}$ and if $ct_1 = \widehat{ct}_1$, send $(\texttt{ReqSatisfy})$ to the corrupted $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$, otherwise if the ciphertext validates with label L, pick a random T and send $(\texttt{TestSatisfy}, L, T)$. Finally, confirm the delay on $\mathcal{F}_{\mathsf{otp}}.\mathsf{SA}$.

On a delay on $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$

– confirm the delay and wait for $(\texttt{Satisfy}, \widehat{T})$ on $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$;

– if $\widehat{T} = \varepsilon$ prove $m = \perp$ with an otherwise random instance and witness of correct size using a simulated $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$.

[2] If $\widehat{open}_{id}$, and $\widehat{open}_{x'_1}$ correspond to the $open_{id}$, and $open_{x'_1}$ sent by the user during the $\texttt{EnrollU}$ phase, this simulation step is perfect. We will show in Lemma 3 that given the binding property of the commitment scheme this is the case except with negligible probability.

[3] Note that in this case, as we did not know the value of T yet, we used a fake encryption of 1 and rely on the CCA security of the ciphertext for the indistinguishability of the simulation. We describe the reduction in Lemma 5.

– if $\widehat{T} \neq \varepsilon$, if this is in reply to a `TestSatisfy` request, prove correct decryption of the blinded token $m = \mathsf{Dec}(\widehat{L}, \widehat{ct}_1, sk_{\mathsf{SA}})$ using $\mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}$ to $\mathsf{notp}_2.\mathsf{U}$, otherwise (this is in reply to a `ReqSatisfy`) prove correct decryption of the blinded token $m = \mathsf{Dec}(L, ct_1, sk_{\mathsf{SA}}) = T \cdot g^{x_1 - x'_1}$).

If RA is corrupted, we have to simulate opening requests towards it. This is done in the same way as for the case of an honest U and an honest SP.[4]

Lemma 3. *Given the DLIN assumption,*[5] *if* U *is corrupted,* SP *is honest, and* SA *and* RA *are either honest or corrupted,* $\mathcal{P}_{\mathsf{otp}}(\mathfrak{R}) \leq \mathcal{I}_{\mathsf{otp}}(\mathfrak{R})$.

For the proof of this Lemma, we refer to Appendix C.5.

Listing 16. $\mathcal{S}$ when the service provider is corrupted, the user is honest

The simulator sets *state* ← "ready" and follows the instructions for U of the real world protocol.

On a delay on $\mathcal{F}_T.\mathsf{SP}$ with *state* = "ready":

– confirm the delay and wait for $(\texttt{EnrollU}, inst)$ from the corrupted $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$, store *inst*.

– generate $params_{\mathsf{U}}$ and send them over $\mathcal{I}_{sc_3}$.

– wait for Com_{x_2} and $params_{\mathsf{SP}}$ on $\mathcal{I}_{sc_3}$, generate random $l_1, id, x_1, x'_1, open_{id}$ and $open_{x'_1}$, compute $Com_{id} \leftarrow \mathsf{Commit}(params_{\mathsf{SP}}, id, open_{id})$, $Com_{x'_1} \leftarrow \mathsf{Commit}(params_{\mathsf{SP}}, x'_1, open_{x'_1})$ and send Com_{id} and Com_{x_1} over $\mathcal{I}_{sc_3}$.

– upon receiving an acknowledgement over $\mathcal{I}_{sc_3}$, set *state* ← "enrollu", and send $(\texttt{Input}_1, (inst, params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{SA}}, (id, wit, open_{id}, x'_1, open_{x'_1}), (l_1, x_1))$ to $\mathcal{I}_{\mathsf{tpc}_1}$.

On t_{due} on $\mathcal{I}_{sc_3}$ where *state* = "enrollu":

– reply with an acknowledgment.

– receive $(\texttt{Input}_2, (open_{x_2}, (l_2, x_2, x'_2)))$ on $\mathsf{notp}_2.\mathsf{SP}$ and forward it to the simulated $\mathcal{I}_{\mathsf{tpc}_1}$ resulting in ct_1 and $l \leftarrow g^{l_1 + l_2}$.

– set *state* ← "deliverenrollu", and send $(\texttt{DeliverEnrollU}, t_{due})$ to $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$ with a confirmation on the delay of $\mathcal{F}_{\mathsf{otp}}.\mathsf{U}$.

On a delay on $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$ where *state* = "deliverenrollu":

– confirm the delay, wait for $(\texttt{DeliverEnrollSP})$ from $\mathcal{F}_{\mathsf{otp}}.\mathsf{SP}$ and send an acknowledgement over $\mathcal{I}_{sc_3}$

– wait for $(\texttt{Input}_1, (params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}, T_{due}), pk_{\mathsf{RA}}, \widehat{open}_{x_2}, (t_{due}, x_2, x'_2))$, simulate $\mathcal{I}_{\mathsf{tpc}_2}$ and receive message $(\texttt{Input}_1, (params_{\mathsf{U}}, params_{\mathsf{SP}}, Com_{id}, Com_{x'_1}, Com_{x_2}), pk_{\mathsf{RA}})$.[6]

[4] This aspect of the simulation is perfect.

[5] See Appendix C.1 for its definition.

[6] If $\widehat{open}_{x_2}$ correspond to the $open_{x_2}$ sent by the service provider during the `EnrollU` phase, this simulation step is perfect. We will show in Lemma 4 that given the binding property of the commitment scheme this is the case except with negligible probability.

– if RA is corrupted, send $(\texttt{ReqOpen})$ to $\mathcal{F}_{\mathsf{otp}}$.SP, learn T, confirm the opening, learn $(\texttt{Open},ID)$ and set $m \leftarrow (T, ID \cdot g^{x'_2})$.

– otherwise set $m \leftarrow (1,1)$.

– send $(\texttt{Input}_2, (open_{id}, open_{x'_1}), (0, x'_1, id))$ to the simulated $\mathcal{I}_{\mathsf{tpc}_2}$ in which we set $ct_2 \leftarrow \mathsf{Enc}(pk_{\mathsf{SA}}, t_{due}, m)$.

– set *state* $\leftarrow$ "enrolled".

As the service provider is corrupted and the user is honest, no extra simulation is needed for an honest SA, or a corrupted RA. If RA is honest, we have to handle opening requests.[7]

On $(\widehat{ct}_2, t_{due})$ on notp_2.SP where *state* = "enrolled";

– simulate $\mathcal{I}_{sc_2}$ and if $ct_2 \leftarrow \widehat{ct}_2$, send $(\texttt{ReqOpen})$ to the corrupted $\mathcal{F}_{\mathsf{otp}}$.SP, otherwise if the ciphertext validates with label $g^{t_{due}}$, pick a random T, *id* and send $(\texttt{TestSatisfy}, T, id, t_{due})$. Finally, confirm the delay on $\mathcal{F}_{\mathsf{otp}}$.RA.

On a delay on $\mathcal{F}_{\mathsf{otp}}$.SP where *state* = "enrolled":

– confirm the delay and wait for $(\texttt{Open}, \widehat{ID})$ on $\mathcal{F}_{\mathsf{otp}}$.SP.

– if $\widehat{ID} = \varepsilon$ prove $m = \perp$ with an otherwise random instance and witness of correct size using a simulated $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$.

– if $\widehat{ID} \neq \varepsilon$, if this is in reply to a $\texttt{TestOpen}$ request, prove correct decryption of the blinded identity $m = \mathsf{Dec}(t_{due}, \widehat{ct}_2, sk_{\mathsf{RA}})$ using $\mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}$ to notp_2.SP, otherwise (this is in reply to a $\texttt{ReqOpen}$) prove correct decryption of the blinded identity $m = \mathsf{Dec}(t_{due}, ct_2, sk_{\mathsf{RA}}) = ID \cdot g^{x'_2}$.

If SA is corrupted, we have to simulate satisfaction requests towards it. This is done in the same way as for the case of an honest U and an honest SP.[8]

Lemma 4. *Given the DLIN assumption, if* SP *is corrupted,* U *is honest, and* SA *and* RA *are either honest or corrupted,* $\mathcal{P}_{\mathsf{otp}}(\mathfrak{R}) \leq \mathcal{I}_{\mathsf{otp}}(\mathfrak{R})$.

The proof follows the proof of Lemma 3.

14.5 Conclusion

Oblivious third parties is a useful mechanism to relieve the trust of users and service providers towards third parties. Moreover, it allows for more efficient systems, in which the user proves satisfaction of certain requirements instead of the revocation authority having to verify if the user really satisfied the requirements (e.g., paid for the service).

[7] Note that in this case, as we did not know the value of T and *id* yet, we used a fake encryption of $(1,1)$ and rely on the CCA security of the ciphertext for the indistinguishability of the simulation. We describe the reduction in Lemma 4.

[8] This aspect of the simulation is perfect.

This complex application is modeled using the general computational model of Küsters [Küs06]. Based on the structure preserving encryption scheme [CHK+11a], we provide an actual instantiation of the oblivious third parties. The protocols were proved secure in the IITM model. Nevertheless, the results carry over to the universal composability model.

Chapter 15

Evaluation

In the previous chapters, a number of ideal functionalities were provided as building blocks for modeling applications. An application named *oblivious trusted third parties*, has been modeled and an instantiation was provided. This is the first realization of the OTP concept. Based on this research, we now evaluate the simulation-based modeling and analyze the way we handle some issues.

15.1 Modeling

The simulation-based model by Küsters [Küs06], is very flexible, gives the modeler a large degree of freedom, and offers security even when a system is used in a larger system. Moreover, it is sufficient to prove the security of a single session, to reason about the security of a multi-session system.

Ideal System. The framework allows a more intuitive approach for modeling the ideal protocol than in other simulation based models [Can01, CLOS02] in which the ideal protocol consists of only a single ITM.

In our approach, we use this flexibility to divide the tasks over well-defined IITMs. We briefly recall the different IITMs we use:

Dummy Parties $\mathcal{D}_{\mathsf{R}}$. For each corruptible role involved in the protocol, a single incorruptible ideal ITM $\mathcal{D}_{\mathsf{R}}$ is used.

Ideal Functionality $\mathcal{F}_{inf}$. All security critical parts and the functionality of the protocol is implemented by a single virtual incorruptible party. This party considers neither corruption, nor communication.

Delayed Communication. For each output tape of the ideal functionality $\mathcal{F}_{inf}$ an IITM $(\mathsf{Delay}(T,\overline{T},C))$ allows the adversary to arbitrarily delay messages. This models the fact that cryptography cannot prevent denial of service attacks against an adversary that is in control of communication resources.

This approach allows us to focus on the ideal functionality, such that it actually implements the requirements of the system. It is, for instance, sufficient to look at the outputs in order to verify that no personal information is retrieved by other parties, or that transactions are unlinkable.

Also corruption and communication is kept simple. Upon corruption, the dummy party simply forwards all messages from I/O tapes to the adversary and allows the adversary to send arbitrary messages on behalf of the corrupted party.

Real System. Real systems can be presented as hybrid systems of real and ideal functionalities. Hence, ideal functionalities can be used as building blocks for building real protocols. Later, upon implementation, the ideal sub-functionalities may be replaced by any real protocol that securely emulates that functionality. This strategy should make it easier to develop new applications/systems.

15.2 Concerns

Nevertheless, there remain a number of concerns with respect to our approach, but also with respect to the framework in general.

15.2.1 Our Approach

Corruption of Roles

We use the standard corruption macro for (static) corruption of dummy parties. Corrupting a role then comes down to a full corruption of the role, with an adversary that may influence the corrupted party. However, sometimes other corruption models are appropriate. For instance, in some settings, an adversary may only passively corrupt a party, and only receive information, but has no direct influence on the corrupted party.

Static Corruption

In our approach, we considered only static corruption. However, it is not fully clear what this means in terms of security in real-world settings, in which the corrupted parties have not decided upon in advance (i.e., adaptive corruption).

Ideal communication

Currently, we use ideal delayed communication for modeling our ideal system. However, sometimes this may be too restrictive. In that case, the delay functionality should allow for more fine grained definitions of communication (e.g., allowing some amount of leakage or influence). Therefore, existing or new ideal functionalities, for instance, for secure and authenticated communication may be used instead.

Moreover, to make simulation easier, the ideal delay functionality currently leaks some amount of information, namely, the type of message being sent. This in a sense is acceptable as the length of ciphertexts could reveal this information as well, unless ciphertext messages have a fixed length. Nevertheless, in order to make the multi-session environment work in an anonymous setting, these messages have to be considered as well.

15.2.2 General Concerns

Simulation-based models such as the model of Küsters offers some important advantages for proving the security of real systems, especially, with respect to game-based definitions. Nevertheless, simulation-based modeling in general also poses some concerns.

Defining Ideal Functionalities. The most important conclusion when proving the security of systems is the following: *simulation-based proofs do not prove security*. In fact, all it proves is that some real system is – at least as secure as – the ideal system. Hence, one should prove in addition to the proof that the real system securely realizes the ideal system, that the ideal system is secure.

The underlying idea is that the model of the ideal system is very simple and straightforward to understand, such that it is easy to reason about its security properties. But in many cases, ideal functionalities become rather complex, making it hard to see if an ideal system indeed fulfills the requirements. For instance, currently we introduce a bijection into the ideal functionality. Here, its use is rather

straightforward, but other systems may introduce other constructions which may make the ideal system insecure.

Moreover, the availability of many different models have resulted in many different definitions for the same ideal functionality, largely depending on the underlying model. In fact, one may expect a similar evolution for finding proper definitions of basic functionalities in simulation-based models, as was the case for the definitions of attack models (e.g., CPA, CCA) used in game-based models.

As the number of corruptible parties (n) grows, the number of simulations grows rapidly ($n!$), making it hard to construct the proofs manually. In fact, if one considers adaptive corruption a lot more simulations may have to be considered.

Implementation – efficiency. Another concern in simulation-based models is that, even if they allow for composition, developing systems based on building blocks may result in inefficient systems. Since efficiency in cryptography is a very important aspect, it would be useful to have some kind of metrics on the *compatibility* of ideal functionalities.

For the construction of primitives (e.g., authenticated and secure communication), it could be better to make a realization of an ideal protocol entirely as a real protocol (i.e., no hybrid protocol), rather than constructing it based on other ideal functionalities. On the other hand, for larger systems, it may be sufficient to reuse those primitives.

15.3 Future Directions

Currently, our OTP application only supports static corruption. It would be interesting to see the implications on the protocol to also support *adaptive corruption*.

Another interesting direction is to *formalize the relations between protocols with different communication requirements* (e.g., secure, authenticated, anonymous). For instance, if we have an ideal system that runs over an insecure channel, it should be possible to reason about the improved security properties when the same system is used over a secure or anonymous channel.

In addition, it would be interesting to set up a *library of (ideal) primitives*, possible realizations and their proofs, such that they can be re-used by protocol developers to build new systems.

Simulation-based proofs are quite extensive and requires knowledge of many different fields. Many publications use some kind of simulation-based model for proving their protocols secure, often by developing new ideal functionalities. Unfortunately, since

publications are often limited in size, the proofs are often very compressed, leaving out details or decisions that may be important, and could possibly make the results invalid. Even if a protocol strongly simulates some ideal protocol, if the latter is defined inappropriately, the proof says nothing about the security. As a first step, to support the comparison and composition of different protocols, some *good practice guidelines* should be provided with some kind of *standardization*.

15.4 Conclusion

In the previous chapters, we have developed a number of building blocks to support the construction of larger protocol systems in the simulation-based setting. Nevertheless, building new protocol systems is not simple and still requires both knowledge of the underlying model and its implications, and also knowledge of cryptographic constructions.

An important observation is that since simulation-based security only proves that attacks in the real world can also occur in the ideal world, it does not ensure security. Security is only obtained in combination with the fact that in the ideal case, it should be much simpler to design a secure protocol.

Unfortunately, the latter is often not the case. Furthermore, it is often a trade-off between an ideal system and what is realizable. A simple example is the following: in the ideal case, secure communication would only leak the fact that an arbitrary message is sent from A to B. However, in the real world, this is not possible, as a ciphertext reveals at least some information about the length of the message. Hence, the ideal protocol has to be relaxed such that it also leaks the length of the message. The same strategy with multiple iterations of relaxations is often used when developing new systems, to allow the real protocol to securely realize the ideal protocol. However, it is not always straightforward to see the implications to the security of the scheme.

Simulation-based security has very interesting properties to help proving a protocol secure. Nevertheless, there are still several concerns left open that should be addressed. Moreover, if used inappropriately, it may allow to prove anything 'secure'. In fact, if an ideal functionality is used in the construction of multiple other protocols, and it turns out to be insecure, all those protocols may get insecure as well.

Chapter 16

General conclusions

16.1 Overview

In contrast to what one could expect from the title, this thesis actually takes off with more practical research, and gradually moves towards the more theoretical work.

This follows more or less the research track I followed during my research. In that sense, my personal background made me take a more practical approach to start from. For instance, practical results showed that there were still a number of issues to be dealt with to make anonymous credentials really practical. In theory there are already some solutions provided, for instance, to solve revocation, but no overview was available on the usability of those solutions.

For each of the major parts of this thesis, we summarize the conclusions and end with the general conclusion on the research presented in this thesis.

Traditional Electronic Identities. The major drawback in traditional eID technologies is the lack of privacy. Currently, the only implementation that supports privacy, though very limited, is the German eID. However, the infrastructure is rather closed, and makes it difficult for service providers to support the technology. Other important drawbacks, such as PIN caching and surreptitious authentications are mainly due to the smart card environment used to carry the credentials.

We provided a number of application domains in which these threats are mitigated. They showed how applications may benefit from strong authentication. However, the

application domains in which traditional card-based eIDs can be used securely, are rather limited.

We used the conclusions from this research as the starting point for the research on electronic identities based on anonymous credentials.

Anonymous Mobile Authentication. To tackle the biggest concern in traditional eIDs, we evaluated the possibility to use anonymous credentials for electronic identities, offering better privacy properties. Anonymous credentials are fairly new, and mostly a theoretical research topic. In this dissertation, we focused on anonymous credentials of TYPE 2, namely, using zero-knowledge proofs for proving the possession of a valid credential (e.g., `Identity Mixer` anonymous credentials). An important drawback in this type of credentials is their complexity. They require a substantial amount of computational resources. In fact, full-fledged `Identity Mixer` anonymous credentials (e.g., with interval proofs, verifiable encryption and proper revocation support), cannot efficiently run entirely on a smart card. Therefore, we analyze the possibility of using mobile devices to assist the smart card in the computations. The results are promising, and show that it in fact could be a good alternative.

Unfortunately, the drawbacks we encountered in traditional eID solutions due to the smart card environment (e.g., trust in the host), remain for the mobile solution we present. Moreover, since mobile devices carry a lot of personal information, and provide increased connectivity, they are a more interesting target for hackers and malicious organizations. Only a combination with Trusted Execution Environments on mobile devices, could make our solution practical. Developments on this are still ongoing.

Revocation of Anonymous Credentials. In traditional eID solutions, revocation is simple and efficient. A service provider may simply check the revocation status of a credential used for authentication by verifying that the identifier of the credential is not in the revocation list.

However, for anonymous credentials the case is much more complex. Since anonymous credentials do not allow such an identifier to be released, other solutions had to be found. Therefore, we evaluated a number of revocation strategies discussed in theory, based on both the approach they follow to solve revocation and the efficiency of the solution.

The results show that none of the solutions is fully satisfactory for the use in an electronic identity infrastructure, especially when using smart cards. In the *smart card-based setting*, in which the card is used as a standalone technology, the best solution

is to combine VLR with credential updates. However, in order to keep it practical, the number of credential updates should be minimal, but sufficient to keep the number of tests by the verifier minimal. Nevertheless, VLR requires a substantial amount of additional resources for the verifier to verify the revocation status. Moreover, with respect to user-friendliness and convenience, this solution is not satisfactory for eIDs as a standalone smart card.

Most strategies, also the one discussed above, require frequent communication of the user with the issuer, for instance, to update the credential or gather revocation information. Moreover, some strategies (i.e., accumulators) may require an additional amount of complex computations. Fortunately, the updates could be carried out by a possibly untrusted party, such as a mobile device. Moreover, if the issuer carries out the witness update, it is in fact similar to a credential update, as used for limited lifetime credentials. For these strategies, a *secure element embedded in a mobile device*, as in our mobile authentication scenario, comes back into the picture. The device may provide both connectivity and possibly assist the secure element for certain computations. A combination of issuer generated credential updates and accumulator based revocation may be the better solution, at least if we assume an issuer with sufficient computational power to generate frequent credential updates. The accumulator may optionally be used for services that require high security. For low-security environments, it may be sufficient to only prove the validity of the credential (e.g., $dateUntil < now$).

Nevertheless, credential updates should occur in batch (i.e., everyone gets the same *Valid-Until* attribute) such that the lifetime may be revealed, instead of proving that it is more than the current time, as the latter would take substantially more computation when showing the credential.

Modeling Secure Applications. Once anonymous credentials are ready for practice, many other concerns still need attention. To fully support privacy-friendly services, service providers may need additional guarantees and new systems may need to be built. Due to the complexity of these systems it gets more difficult to prove them secure. We therefore analyzed the simulation-based models, and in particular the model by Küsters [Küs06]. We have modeled such a new system, named oblivious third parties. In the latter, the service provider may get more trust in the system since he is assured that unless certain requirements are fulfilled, appropriate countermeasures may be taken, even without expensive court decisions.

Due to the complexity in the modeling of such systems and the discrepancy between ideal and real systems, modeling systems is not simple and requires in-depth knowledge of the model. These frameworks are still subject to change or get extended to support new constructions. To be really of use, they should provide a common strategy on how to model ideal and real systems. We provided a first attempt using

dummy parties and a virtual incorruptible ideal functionality. It tries to simplify the definition of the ideal functionality by separating corruption and communication from the actual functionality. It has shown to be an interesting approach, however, it remains to be verified if the approach remains valid for other systems, or when considering adaptive adversaries.

16.2 Anonymous Credential systems: From Theory towards Practice

As in traditional eID infrastructures, standalone smart cards embedding anonymous credentials may become practical in the near future. However, implementing all the functionality provided by the `Identity Mixer` library on the smart card is currently impossible. Thus, only a stripped down version with a very efficient implementation (i.e., smart cards that support efficient modular arithmetic such as MULTOS [8]), could partially support anonymous credentials.

Nevertheless, since the smart card does not provide a trusted user-interface, the user does not know what is going on. Hence, the user will never know what is really being proved. It will require even more trust in the host. Furthermore, next to the computational requirements, connectivity is also a major problem. To support revocation, we need frequent credential updates or up-to-date revocation information, making a smart card solution less favorable, especially if these updates are required frequently.

On the other hand, anonymous credentials would gain a lot if they were used in combination with mobile devices. The computational resources and mobility makes them very attractive for eID-based transactions. In fact, they may be a real game-changer for the adoption of anonymous credentials.

In the introduction, we posed the following question:

> *Is it feasible to use anonymous credentials as a nation-wide electronic identity, offering both enhanced privacy and security properties?*

Summarizing the conclusions above, we could positively answer this question, if anonymous credentials are used in combination with mobile devices.

But, we have to make some side-notes. Even though mobile devices may support the use of anonymous credentials, they are already the target of plenty of malicious applications. Even though one could assume that mobile devices can gain more trust, as it is a personal device, this may be in fact the major reason why it is so attractive: the device being personal makes it valuable for attackers. Hence, in order to securely

support anonymous credentials as an electronic identity, we should use mobile devices with enhanced security protection against malware by, for instance, the use of trusted execution environments.

So, the final answer is: if we can provide trustworthy mobile devices, anonymous credentials as a nation-wide electronic identity, offering both enhanced privacy and security properties are indeed practical.

Appendix A

Implementation Notes

A.1 Implementation Details of Mobile Authentication towards a terminal

Entities. M is realized with an up-to-date Android 2.2-based smartphone.[1] For simplicity of the showcase, T, AS, and IP are realized as services on a single PC,[2] while those services can be easily distributed to multiple machines as required in real-world deployments. On top of the privacy and security framework, M runs an Android authentication app. Communication is handled by the client-side implementation of the extended RESTlet framework(cf. Sect. 6.2). For the prototype, depending on the destination specified in the authentication request, the response will automatically get redirected to the correct channel.

With respect to the storage of the credentials, the prototype supports both credentials stored on the device as well as credentials embedded on the tamper resistant chip of a Secure microSD. We therefore leverage the extended `Identity Mixer`, as discussed in Sect. 6.4.
As secure element, we selected the secure Mobile Security Card SE 1.0 by G&D, a microSD card comprising a tamper resistant Java Card chip. We have implemented the algorithms discussed above as an applet instantiated on the secure element. The `Identity Mixer` library on the host has been adapted to invoke the correct algorithms on the Java Card. The implementation uses the *OV-Chip 2.0 Bignat library* for computing with arbitrary precision integers on a Java Card.

[1] Samsung Galaxy i9000: 1 GHz ARM Cortex-A8, 512MB RAM, 480x800 WVGA Super AMOLED screen, 2592 x 1944 Camera.

[2] DELL E4300: Intel Core2 Duo P9600 @2.54GHz, 4GB RAM, Windows 7(64), 1280x720 Webcam.

T is realized as a GWT (v. 2.1.1) browser application, using the RESTlet-GWT module. A Java™-based GWT Widget was developed for setting up the visual communication channel. Particularly, displaying and scanning QR codes uses the PC's display, resp. webcam. The terminal communicates over an SSL/TLS channel with the authorization server.

The servers, AS and IP, are both realized as Tomcat (v. 6.0) applications featuring the RESTlet communication framework and our privacy and security framework. In the RESTlet framework each resource of a party is protected by a *guard*, also called *ChallengeAuthenticator*. This guard confirms the *authentication requirements* an access requester needs to fulfill in order to get access to this resource. We implemented a guard that delegates the authentication protocols to the security and privacy framework. Access control at the entities IP and AS is technically implemented through such guards.

Communication. We use the extended RESTlet framework, presented in Sect. 6.2 for the communication between the different entities. For both scenarios (a) and (b), the short-range channel is then realized as a QR code-based visual channel between M and T. For scenario (b), we use a network connection.

A.2 `Identity Mixer` and DAA in Android™

`Identity Mixer`. The `Identity Mixer` library is a cryptographic library written for the Java™ platform implementing the credential system of Camenisch and Lysyanskaya [CL01].

For demonstrating secure and privacy-friendly mobile applications, we made the library compatible with the Android™ platform. Actually, only very few changes were required (some were related to XML, and a few other unsupported functionalities on the Android platform). These changes have been requested to IBM, and updated in version 2.3.3 of the library. Meanwhile, we fixed a bug in the inequality prover of the library, since this proof was not linked to the proof of the credential.

A drawback in the library is that using objects (i.e., parameters, specifications and credentials) in the library, requires those objects to be loaded from an XML file, reachable using a Uniform Resource Identifier (URI). In order to make the library more flexible and compatible with other technologies supported by our framework, we added an engine that circumvents this requirement. As an example, a credential object in our framework has a technology agnostic and a technology specific part. Since the `Identity Mixer` library requires the technology specific part to be provided as an XML file, the engine compiles the XML file on-the-fly and loads it into the library.

Although it adds some additional overhead, it makes our framework more consistent, and less dependent on the technology and platform being used.

DAA. Direct Anonymous Attestation [BCC04] (DAA), can be seen as a stripped-down version of `Identity Mixer` anonymous credentials, with no attributes. As a result, with DAA, one can only prove that it is genuine, and in addition that it is not revoked (based on the Verifier Local Revocation technique [BCC04]). In fact, the Trusted Computing Group (TCG) group has included DAA into Trusted Platform Modules (TPM), supporting, for instance, anonymous device attestation of personal computers.

To plug the DAA-based anonymous credentials into the framework, a DAA credential object implements an interface towards the Java Card (embedded on a microSD card), featuring the algorithms required for implementing DAA. Note that for DAA, the computation for showing a credential is partially done on the host (in the DAA credential handler).

A.3 Implementation of Accumulators in C++

A.3.1 Implementation Notes

To compare the schemes discussed above, they are all implemented in C++. The bilinear maps applied in the LN and CKS scheme, are initialized using the PBC Library [17]. This library is built on top of the GNU Multiple Precision Arithmetic Library (GMP [10]), which performs the underlying mathematical operations. To make the comparison as fair as possible, the CL scheme, which does not use bilinear maps, is also implemented using the GMP library. Where applicable, optimized versions for applying pairings and multi-exponentiations in both libraries are used. However, further optimizations may still be possible.

A.3.2 Configuration

The pairing-based schemes, LN and CKS are constructed using symmetric pairings. The pairing used, is a '*Type A*' pairing, as defined in the PBC Library. This pairing is the fastest, available in the library, and allows the user to select the field-size and subgroup-size. However, as the implementation is independent of the type of pairing, other symmetric pairings could be used as well. The '*Type A*' pairing has the following properties: $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ on a supersingular curve $E : y^2 = x^3 + x$ with embedding degree 2, field-size $l_q = 1024$ bits, and a subgroup of size $l_r = 192$ bits.

For the CL scheme, based on the `Identity Mixer`, the following system parameters are used:[3] $l_n = 2048, k' = 80, k'' = 80, l_e = 838, l'_e = 120, l_v = 2965, l_\Gamma = 1632, l_\rho = l_m = 498, l_H = 256, l_{id_i} = 160$.

The experiments were executed with these security parameters on a DELL Latitude P9600 @ 2.53GHz, 4GB RAM.

[3] Note that the parameters proposed for the CL scheme in [Ngu05a] do not satisfy the constraints posed by the respective scheme.

Appendix B

Smart Card Extension based on Commitments

The following presents a smart card extension in which the smart card computes a commitment and proves knowledge of it, hence keeping the master secret secure on the smart card. It is based on the commitment scheme presented in [DF02].

It is similar to the construction of Danes [Dan07], but updated to the latest `Identity Mixer` protocol specification. Note that the pseudonym construction is now based on prime order groups, which we will not consider for our smart card extension.

Our extension consists of the following algorithms running on the card:

initCard(..) initializes the card with fixed system parameters l_m, l_n, l_ϕ and l_c. l_m defines the length of the master secret, l_n the size of the modulus, l_ϕ the statistical zero-knowledgeness, and l_c the length of the challenge. The master secret is chosen uniformly at random $ms \in_R [1, 2^{l_m}]$ and the issuer's key $pk_{\mathsf{IP}} = (n, g, h_1)$ is stored.

verifyPIN(..) verifies the PIN provided by the user and returns true in case of a correct PIN. After a fixed number of invalid tries, the card is blocked.

getCommon(..) choose random $v_c \in_R \pm\{0,1\}^{l_n+l_\phi}$, stores it and returns $C_{ms} = h_1^{ms} g^{v_c} \bmod n$.

getTValue(..) sets $r_{ms} \in_R \pm\{0,1\}^{l_m+l_\phi+l_c+1}$, $r_{v_c} \in_R \pm\{0,1\}^{l_n+2l_\phi+l_c}$ and returns $T_{ms} = h_1^{r_{ms}} g^{r_{v_c}} \bmod n$.

getSValues(..) receives the challenge c and returns $s_{ms} = r_{ms} + c \cdot ms$ and $s_{v_c} = r_{v_c} + c \cdot v_c$.

Credential Issuance. The credential issuance is then changes as follows: the card computes a commitment to the master secret and sends the proof to the host.

Without loss of generality, we assume that the attribute with index 1 is the master secret *ms*.

The proof of knowledge towards the issuer is denoted as follows:

$$\begin{aligned} PK\{(\{m_i : i \in A_h\}, v_c) : \\ U \equiv \pm g^{v_c} h_1^{ms} \cdot \prod_{j \in A_h \setminus \{1\}} h_j^{m_j} \bmod n \\ m_i \in \pm\{0,1\}^{l_m + l_\phi + l_c + 1} \forall i \in A_h \\ \}, \end{aligned} \tag{B.1}$$

with A_h the set of hidden attribute indices.

When converting this to actual protocols, we first compute the commitment U, which is partially computed on the smart card by invoking getCommon. The host may then compute the commitment to the hidden attributes $U = C_{ms} \prod_{j \in A_h \setminus \{1\})} h_j^{m_j} \bmod n$.

Then, in order to compute the proof of knowledge, getTValue is invoked on the card and the t-value T_U is computed as follows: $T_U = T_{ms} \cdot \prod_{j \in A_h \setminus \{1\})} h_j^{r_j} \bmod n$, with $r_j \in \pm\{0,1\}^{l_m + l_\phi + l_c + 1}$.

The host computes the challenge based on the Fiat-Shamir heuristic, sends it to the card and invokes getSValue, which returns the s-values related to the secrets on the card. The s-values for the remaining hidden attributes are computed locally.

Note that in the credential issuance protocol, when the recipient obtains a credential signature (A, e, v), it actually receives v_h from which it may compute $v = v_c + v_h$ However, in our case $v = v_c + v_h$ is now split over the host (i.e., v_h) and the smart card (i.e., v_c).

Credential Show. The proof of knowledge of a valid CL-signature changes as follows: As in the original `Identity Mixer` protocol, in order to prove knowledge, the host first computes a randomization of its signature (A, e, v):

$$r_A \in_R \{0,1\}^{l_n + l_\phi} \tag{B.2}$$

$$\tilde{A} = A.g^{r_A} \bmod n \tag{B.3}$$

$$\tilde{v} = v - e \cdot r_A \tag{B.4}$$

$$\tilde{e} = e - 2^{l_e - 1}. \tag{B.5}$$

However, since v is partially kept on the Java Card, we slightly modify the computation: the host computes $\tilde{v_h} = v_h - e \cdot r_A$ using only the v_h value, as received during issuance.

The proof of knowledge of a valid CL signature is given by formula B.6:

$$\begin{aligned} PK\{&(e, \{m_i : i \in A_h\}, v) : \\ &\frac{h}{\prod_{i \in A_r} h_i^{m_i}} \equiv \pm A^e h_1^{ms} g^v \prod_{j \in A_h \setminus \{0\}} h_j^{m_j} \bmod n \\ &\forall i \in A_h : m_i \in \{0,1\}^{l_m + l_\phi + l_c + 2} \\ &e - 2^{l_e - 1} \in \{0,1\}^{l'_e + l_\phi + l_c + 2} \\ \}&, \end{aligned} \tag{B.6}$$

with A_h and A_r are the sets of hidden, resp. , revealed attribute indices.

To prove this, the host first invokes verifyPIN, with the correct PIN, followed by invoking getTValue. As a result, the host receives T_{ms}. Now, the protocol proceeds by computing the commitment T_Z, which in the original protocol is computed as follows:

$$T_Z = \tilde{A^e} \cdot g^{r_v} \prod_{j \in A_h} h_j^{r_j} \bmod n. \tag{B.7}$$

However, since *ms* is unknown to the host, we reorder some computations resulting in:

$$T_Z = \tilde{A^e} \cdot T_{ms} \cdot g^{r_{v_h}} \prod_{j \in A_h \setminus \{0\}} h_j^{r_j} \bmod n. \tag{B.8}$$

This is easily verified as follows:

$$T_Z = \tilde{A^e} \cdot T_{ms} \cdot g^{r_{v_h}} \cdot \prod_{j \in A_h \setminus \{0\}} h_j^{r_j} \bmod n \tag{B.9}$$

$$T_Z = \tilde{A^e} \cdot h_1^{r_{ms}} g^{r_{v_c}} . g^{r_{v_h}} \cdot \prod_{j \in A_h \setminus \{0\}} h_j^{r_j} \bmod n \tag{B.10}$$

$$T_Z = \tilde{A^e} \cdot h_1^{r_{ms}} g^{r_{v_c} + r_{v_h}} \cdot \prod_{j \in A_h \setminus \{0\}} h_j^{r_j} \bmod n \tag{B.11}$$

$$T_Z = \tilde{A^e} \cdot g^{r_v} \cdot \prod_{j \in A_h} h_j^{r_j} \bmod n. \tag{B.12}$$

On the host, the protocol proceeds as usual and after computing the challenge, the host invokes getSValues on the card, obtaining the s-values s_{ms} and s_{v_c}. The host computes $s_{v_h} = r_{v_h} + c \cdot v_h$

Finally, the host computes $s_v = s_{v_c} + s_{v_h}$, which is exactly the same as in the `Identity Mixer` library. This is easily verified as follows:

$$\begin{aligned} s_v &= s_{v_c} + s_{v_h} && \text{(B.13)} \\ &= (r_{v_c} + c \cdot v_c) + (r_{v_h} + c \cdot v_h) && \text{(B.14)} \\ &= (r_{v_c} + c \cdot v_c) + (r_{v_h} + c \cdot (v_h - e \cdot r_A)) && \text{(B.15)} \\ &= (r_{v_c} + r_{v_h} + c \cdot (v_c + v_h - e \cdot r_A)) && \text{(B.16)} \\ &= (r_v + c \cdot \tilde{v}) . && \text{(B.17)} \end{aligned}$$

The show protocol further proceeds as would be the case without the Java Card. Note that neither s_{v_c} nor s_{v_h} are sent to the verifier.

Proof. The protocol running on the smart card is actually the interactive proof of knowledge of the opening of a commitment in a hidden order group [DF02], as it is used in `Identity Mixer`. In fact, the same, but non-interactive, protocol is part of the issuance protocol in `Identity Mixer`, where the user proves knowledge of hidden attributes, which are to be included in the credential. Hence, no information is leaked to the service provider. We refer to the original paper [DF02] for the proof.

Appendix C

Oblivious Third Parties

C.1 Structure Preserving Encryption

Camenisch et al. [CHK+11a] present a structure preserving encryption scheme. First, recall their definition of structure preserving encryption:

Definition 10. Structure Preserving Encryption. *An encryption scheme is said to be structure-preserving if (1) its public keys, messages, and ciphertexts consist entirely of elements of a bilinear group, (2) its encryption and decryption algorithm perform only group and bilinear map operations, and (3) it is provably secure against chosen-ciphertext attacks.*

Also, recall the well-known DLIN assumption [BBS04]:

Definition 11. *Decisional Linear Assumption (DLIN). Let $\mathbb{G}$ be a group of prime order p. For randomly chosen $g_1, g_2, g_3 \in_R \mathbb{G}$ and $r, s, t \in_R \mathbb{Z}_p$, the following two distributions are computationally indistinguishable:*

$$(\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^t) \approx (\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^{r+s}) .$$

The algorithms of their structure-preserving encryption scheme are given below. The scheme uses a group $\mathbb{G}$ of prime order p generated by g and equipped with a non-degenerate efficiently computable bilinear map $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. For simplicity, we describe the scheme when encrypting a message that is a single group element in $\mathbb{G}$, but it is easily extended to encrypt vectors of group elements.

- Keygen (1^k): Choose random group generators $g_1, g_2, g_3 \in_R \mathbb{G}^*$. For randomly chosen $\vec{\alpha} \in_R \mathbb{Z}_p^3$, set $h_1 = g_1^{\alpha_1} g_3^{\alpha_3}$ and $h_2 = g_2^{\alpha_2} g_3^{\alpha_3}$. Then, select

$\vec{\beta}_0,\ldots,\vec{\beta}_5 \in_R \mathbb{Z}_p^3$, and compute $f_{i,1} = g_1^{\beta_{i,1}} g_3^{\beta_{i,3}}$, $f_{i,2} = g_2^{\beta_{i,2}} g_3^{\beta_{i,3}}$, for $i = 0,\ldots,5$. Output $pk = (g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^5)$ and $sk = (\vec{\alpha}, \{\vec{\beta}_i\}_{i=0}^5)$.

- $\mathsf{Enc}(pk, L, m)$: To encrypt a message m with a label L, choose random $r, s \leftarrow \mathbb{Z}_p$ and set

$$u_1 = g_1^r,\ u_2 = g_2^s,\ u_3 = g_3^{r+s}, \quad c = m \cdot h_1^r h_2^s,$$

$$v = \prod_{i=0}^{3} \mathsf{e}(f_{i,1}^r f_{i,2}^s, u_i) \cdot \mathsf{e}(f_{4,1}^r f_{4,2}^s, c) \cdot \mathsf{e}(f_{5,1}^r f_{5,2}^s, L),$$

  where $u_0 = g$. Output $ct = (u_1, u_2, u_3, c, v)$.

- $\mathsf{Dec}(sk, L, ct)$: Parse ct as (u_1, u_2, u_3, c, v). Then check whether

$$v \stackrel{?}{=} \prod_{i=0}^{3} \mathsf{e}(u_1^{\beta_{i,1}} u_2^{\beta_{i,2}} u_3^{\beta_{i,3}}, u_i) \cdot \mathsf{e}(u_1^{\beta_{4,1}} u_2^{\beta_{4,2}} u_3^{\beta_{4,3}}, c) \cdot \mathsf{e}(u_1^{\beta_{5,1}} u_2^{\beta_{5,2}} u_3^{\beta_{5,3}}, L),$$

  where $u_0 = g$. If the latter is unsuccessful, reject the ciphertext as invalid. Otherwise, output $m = c \cdot \left(u_1^{\alpha_1} u_2^{\alpha_2} u_3^{\alpha_3}\right)^{-1}$.

C.2 Joint Ciphertext Computation

C.2.1 Algorithms of $\mathcal{P}_{\mathsf{jcc}}$

Below, we give the details for the $\mathsf{BlindEnc}_1$, $\mathsf{BlindEnc}_2$, and $\mathsf{UnblindEnc}$ algorithms.

Listing 17. Algorithms of $\mathcal{P}_{\mathsf{jcc}}$

$(msg_1, aux_1) \leftarrow \mathsf{BlindEnc}_1(pk, l_1, x_1)$

- parse pk as $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^5)$.
- pick $\{\gamma_i\}_{i=1}^5$, $\{\delta_i\}_{i=1}^2$, r_1, and s_1 at random and compute

$$\vec{u_1'} = g^{\gamma_1} \cdot g_1^{r_1}, \quad \vec{u_2'} = g^{\gamma_2} \cdot g_2^{s_1}, \quad \vec{u_3'} = g^{\gamma_3} \cdot g_3^{r_1+s_1},$$
$$\vec{u_4'} = g^{\gamma_4} \cdot g^{x_1} \cdot h_1^{r_1} h_2^{s_1}, \quad \vec{u_5'} = g^{\gamma_5} \cdot g^{l_1},$$
$$\vec{v_1'} = \mathsf{e}(g_1, g^{\delta_1}) \cdot \prod_{i=1} \mathsf{e}(f_{i,1}, g^{\gamma_i}), \quad \vec{v_2'} = \mathsf{e}(g_2, g^{\delta_2}) \cdot \prod_{i=1} \mathsf{e}(f_{i,2}, g^{\gamma_i}).$$

- output $msg_1 = (\vec{u_1'}, \vec{u_2'}, \vec{u_3'}, \vec{u_4'}, \vec{u_5'}, \vec{v_1'}, \vec{v_2'})$ and $aux_1 = (\{\gamma_i\}_{i=1}^5, \{\delta_i\}_{i=1}^2, r_1, s_1)$.

$(msg_2, aux_2) \leftarrow \mathsf{BlindEnc}_2(pk, l_2, x_2, msg_1)$

- parse pk as $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^5)$ and msg_1 as $(\vec{u_1'}, \vec{u_2'}, \vec{u_3'}, \vec{u_4'}, \vec{u_5'}, \vec{v_1'}, \vec{v_2'})$.

- pick r_2 and s_2 at random and compute

$$\begin{gathered}\bar{u}_1 = \vec{u}_1 \cdot g_1^{r_2}, \quad \bar{u}_2 = \vec{u}_2 \cdot g_2^{s_2}, \quad \bar{u}_3 = \vec{u}_3 \cdot g_3^{r_2+s_2}, \\ \bar{u}_4 = \vec{u}_4 \cdot g^{x_2} \cdot h_1^{r_2} h_2^{s_2}, \quad \bar{u}_5 = \vec{u}_5 \cdot g^{l_2}, \\ \bar{v} = \left(\textstyle\prod_{i=0} \mathsf{e}(f_{i,1}, \bar{u}_i)/\vec{v}_1\right)^{r_2} \cdot \left(\textstyle\prod_{i=0} \mathsf{e}(f_{i,2}, \bar{u}_i)/\vec{v}_2\right)^{s_2},\end{gathered}$$

  where $\bar{u}_0 = g$.

- output $msg_2 = (\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5, \bar{v})$ and $aux_2 = (r_2, s_2)$.

$ct \leftarrow \mathsf{UnblindEnc}(pk, msg_2, aux_1)$

- parse pk as $(g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0}^5)$, msg_2 as $(\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5, \bar{v})$ and $aux_1 = (\{\gamma_i\}_{i=1}^5, \{\delta_i\}_{i=1}^2, r_1, s_1)$.

- compute

$$\begin{gathered}u_1 = \bar{u}_1/g^{\gamma_1} = g_1^r, \quad u_2 = \bar{u}_2/g^{\gamma_2} = g_2^s, \quad u_3 = \bar{u}_3/g^{\gamma_3} = g_3^{r+s}, \\ u_4 = \bar{u}_4/g^{\gamma_4} = g^{x_1+x_2} \cdot h_1^r h_2^s, \quad u_5 = \bar{u}_5/g^{\gamma_5} = g^{l_1+l_2}, \\ v = \bar{v} \cdot \mathsf{e}(u_1 g_1^{-r_1}, g^{\delta_1}) \cdot \mathsf{e}(u_2 g_2^{-s_1}, g^{\delta_2}) \cdot \textstyle\prod_{i=0} \mathsf{e}(f_{i,1}^{r_1} f_{i,2}^{s_1}, u_i),\end{gathered}$$

  where $u_0 = g$.

- output $ct = (u_1, u_2, u_3, u_4, v)$ encrypted with label u_5.

Correctness. Recall the structure of the ciphertext of the public-key encryption scheme described in Appendix C.1: for a public key $pk = (g_1, g_2, g_3, h_1, h_2, \{f_{i,1}, f_{i,2}\}_{i=0})$, label u_5, and randomly chosen $r, s \leftarrow \mathbb{Z}_q$, the ciphertext is computed as

$$(u_1, u_2, u_3, u_4, v) = \left(g_1^r,\ g_2^s,\ g_3^{r+s}, \quad m \cdot h_1^r h_2^s, \quad \prod_{i=0}^{5} \mathsf{e}(f_{i,1}^r f_{i,2}^s, u_i)\right), \text{where } u_0 = g.$$

Note that the protocol in Listing 8 computes a valid ciphertext because $u_1 = g_1^r$ for $r = r_1 + r_2$, $u_2 = g_2^s$ for $s = s_1 + s_2$, $u_3 = g_3^{r+s}$, $u_4 = m \cdot h_1^r h_2^s$ for $m = g^{x_1+x_2}$, and $v = \prod_{i=0} \mathsf{e}(f_{i,1}^r f_{i,2}^s, u_i)$. To see v is indeed computed this way, note that:

$$\bar{v} = \left(\prod_{i=0} \mathsf{e}(f_{i,1}, \bar{u}_i)/\vec{v}_1\right)^{r_2} \cdot \left(\prod_{i=0} \mathsf{e}(f_{i,2}, \bar{u}_i)/\vec{v}_2\right)^{s_2} = \frac{\prod_{i=0} \mathsf{e}(f_{i,1}^{r_2} f_{i,2}^{s_2}, u_i)}{\mathsf{e}(g_1, g^{\delta_1})^{r_2} \cdot \mathsf{e}(g_2, g^{\delta_2})^{s_2}}$$

and

$$\bar{v} \cdot \mathsf{e}\left(\frac{u_1}{g_1^{r_1}}, g^{\delta_1}\right) \cdot \mathsf{e}\left(\frac{u_2}{g_2^{s_1}}, g^{\delta_2}\right) = \bar{v} \cdot \mathsf{e}(g_1^{r_2}, g^{\delta_1}) \cdot \mathsf{e}(g_2^{s_2}, g^{\delta_2}) = \prod_{i=0} \mathsf{e}(f_{i,1}^{r_2} f_{i,2}^{s_2}, u_i).$$

C.3 Proof of Theorem 1

Proof sketch of Theorem 1: To prove security of Theorem 1 in Sect. 13.4, we show that there exists a simulator $\mathcal{S}$ connected to $\mathcal{E}$ on interface $\mathtt{ntpc}_2$ and to $\mathcal{I}_{\mathsf{tpc}}$ on interface $\mathtt{ntpc}_1$ such that $\mathcal{E}|\mathcal{P}_{\mathsf{jcc}} \approx \mathcal{E}|\mathcal{S}|\mathcal{I}_{\mathsf{tpc}}$. The main cases to be considered for the security proof are when P_1 is corrupted and P_2 is honest, and vice versa.

For the case when P_1 is corrupted by $\mathcal{E}$, in the first step $\mathcal{S}$ receives $\overline{u}'_1$, $\overline{u}'_2$, $\overline{u}'_3$, $\overline{u}'_4$, $\overline{u}'_5$, $\overline{v}'_1$, $\overline{v}'_2$, pk as well as $x_1, l_1, r_1, s_1, \delta_1, \delta_2$ as a part of $(\mathtt{Prove}, (msg_1, pk, inst), (wit_1, l_1, x_1, aux_1))$ being send to the simulated $\mathcal{I}_{\mathsf{zk}_1}$. Then, $\mathcal{S}$ sends $(\mathtt{Input}_1, inst, pk, wit_1, (l_1, x_1))$ to $\mathcal{I}_{\mathsf{tpc}}$ and receives back $(\widehat{u}_1, \widehat{u}_2, \widehat{u}_3, \widehat{u}_4, \widehat{u}_5, \widehat{v})$ which is the ciphertext $(\widehat{u}_1, \widehat{u}_2, \widehat{u}_3, \widehat{u}_4, \widehat{v})$ to be computed at the end by P_1 with a label $\widehat{u}_5$. Using the values $r_1, s_1, r_1, s_1, \delta_1, \delta_2$ obtained earlier, $\mathcal{S}$ computes:

$$\overline{u}_1 = \widehat{u}_1 \cdot \overline{u}'^{r_1}_1 / g_1^{r_1}, \qquad \overline{u}_2 = \widehat{u}_2 \cdot \overline{u}'^{s_1}_2 / g_2^{s_1}, \qquad \overline{u}_3 = \widehat{u}_3 \cdot \overline{u}'^{r_1+s_1}_3 / g_3^{r_1+s_1},$$
$$\overline{u}_4 = \widehat{u}_4 \cdot \overline{u}'^{x_1}_4 / g_1^{x_1}, \qquad \overline{u}_5 = \widehat{u}_5 \cdot \overline{u}'^{l_1}_5 / g^{l_1},$$
$$\overline{v} = \widehat{v} \Big(\mathsf{e}\left(u_1 g_1^{-r_1}, g^{\delta_1}\right) \mathsf{e}\left(u_2 g_2^{-s_1}, g^{\delta_2}\right) \textstyle\prod_{i=0} \mathsf{e}\left(f_{i,1}^{r_1} f_{i,2}^{s_1}, u_i\right) \Big)^{-1},$$

and sends those to P_1 as part of the instance sent to $\mathcal{I}_{\mathsf{zk}_2}$. Thus, the jointly computed ciphertext obtained by P_1 is the one which was produced by the ideal functionary $\mathcal{I}_{\mathsf{tpc}}$.

In the case when P_2 is corrupt, $\mathcal{S}$ chooses random $\overline{u}'_1, \overline{u}'_2, \overline{u}'_3, \overline{u}'_4, \overline{u}'_5, \leftarrow \mathbb{G}$ and $\overline{v}'_1, \overline{v}'_2 \leftarrow \mathbb{G}_T$, and delivers those to P_2 via $\mathcal{I}_{\mathsf{zk}_1}$. In the next step, $\mathcal{S}$ receives from P_2 the values $\overline{u}_1, \overline{u}_2, \overline{u}_3, \overline{u}_4, \overline{u}_5, \overline{v}$ as well as x_2, l_2 as a part of the message $(\mathtt{Prove}, (msg_2, pk, inst), (wit_2, l_2, x_2, aux_2))$ sent to the simulated $\mathcal{I}_{\mathsf{zk}_2}$ by P_2. Finally, $\mathcal{S}$ submits $(\mathtt{Input}_2, wit_2, (l_2, x_2))$ to $\mathcal{I}_{\mathsf{tpc}}$ and P_1 obtains the correct ciphertext.

For the case when both P_1 and P_2 are honest, simulation is easy due to the use of $\mathcal{I}_{\mathsf{zk}_1}$ and $\mathcal{I}_{\mathsf{zk}_2}$, which only requires $\mathcal{E}$ to receive certain notifications. No meaningful messages have to be exchanged between the two parties as the statements are not revealed to the environment over the network interfaces.

□

C.4 Efficient Realization of Zero-Knowledge Proofs

Verifiable Encryption. We show how to efficiently prove the relations $\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$ and $\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$. Note that $aux_1 = (\{\gamma_i\}_{i=1}^5, \{\delta_i\}_{i=1}^2, r_1, s_1)$ and $aux_2 = (r_2, s_2)$. We write ϕ_1, ϕ_2, and *bases* to refer to the formulas of the tractable relations $\mathfrak{R}_1$, $\mathfrak{R}_2$ and the bases in *inst* respectively.

Listing 18. Efficient realization of $\mathcal{P}_{\mathsf{zk}_1} \leq \mathcal{I}_{\mathsf{zk}_1}(\mathfrak{R}_{\mathsf{P}_1}(\mathfrak{R}_1))$ and $\mathcal{P}_{\mathsf{zk}_2} \leq \mathcal{I}_{\mathsf{zk}_2}(\mathfrak{R}_{\mathsf{P}_2}(\mathfrak{R}_2))$

The proofs of correct encryption are as follows:

$$\pi_1 = \nu\, wit_1, l_1, x_1, \gamma_1, \ldots, \gamma_5, r_1, s_1, \delta_1, \delta_2 : \phi_1(wit_1, l_1, x_1, bases) \wedge \vec{u_1'} = g^{\gamma_1} \cdot g_1^{r_1} \wedge$$
$$\vec{u_2'} = g^{\gamma_2} \cdot g_2^{s_1} \wedge \vec{u_3'} = g^{\gamma_3} \cdot g_3^{r_1+s_1} \wedge \vec{u_4'} = g^{\gamma_4} \cdot g^{x_1} \cdot h_1^{r_1} h_2^{s_1} \wedge \vec{u_5'} = g^{\gamma_5} \cdot g^{l_1} \wedge$$
$$\vec{v_1'} = \mathsf{e}(g_1, g^{\delta_1}) \cdot \prod_{i=1}^{5} \mathsf{e}(f_{i,1}, g^{\gamma_i}) \wedge \vec{v_2'} = \mathsf{e}(g_2, g^{\delta_2}) \cdot \prod_{i=1}^{5} \mathsf{e}(f_{i,2}, g^{\gamma_i})$$

and

$$\pi_2 = \nu\, wit_2, l_2, x_2, r_2, s_2 : \phi_2(wit_2, l_2, x_2, bases) \wedge \bar{u_1} = \vec{u_1'} \cdot g_1^{r_2} \wedge \bar{u_2} = \vec{u_2'} \cdot g_2^{s_2} \wedge$$
$$\bar{u_3} = \vec{u_3'} \cdot g_3^{r_2+s_2} \wedge \bar{u_4} = \vec{u_4'} \cdot g^{x_2} \cdot h_1^{r_2} h_2^{s_2} \wedge$$
$$\bar{u_5} = \vec{u_5'} \cdot g^{l_2} \wedge \bar{v} = \left(\prod_{i=0}^{5} \mathsf{e}(f_{i,1}, \bar{u_i}) / \vec{v_1'}\right)^{r_2} \left(\prod_{i=0}^{5} \mathsf{e}(f_{i,2}, \bar{u_i}) / \vec{v_2'}\right)^{s_2}$$

where $\bar{u_0} = g$.

Verifiable Decryption. Below we present a proof that $ct = (u_1, u_2, u_3, \vec{c}, v)$ decrypts to $\vec{m}$ for a label L with a secret key $sk = (\{\vec{x}\}_{i=1}^{n}, \{\vec{y}\}_{i=0}^{(n+4)})$ corresponding to $pk = (\{h_{i,1}, h_{i,2}\}_{i=1}^{n}, \{f_{i,1}, f_{i,2}\}_{i=0}^{(n+4)})$.

Listing 19. Efficient realization of $\mathcal{P}_{\mathsf{zk}_{\mathsf{SA}}} \leq \mathcal{I}_{\mathsf{zk}_{\mathsf{SA}}}(\mathfrak{R}_{\mathsf{SA}})$ and $\mathcal{P}_{\mathsf{zk}_{\mathsf{RA}}} \leq \mathcal{I}_{\mathsf{zk}_{\mathsf{RA}}}(\mathfrak{R}_{\mathsf{RA}})$

The proof of correct decryption is as follows:

$$\pi = \nu\, \{\vec{x}\}_{i=1}^{n}, \{\vec{y}\}_{i=0}^{n+4} : \{\, h_{i,1} = g_1^{x_{i,1}} g_3^{x_{i,3}} \,\}_{i=1}^{n} \wedge \{\, h_{i,2} = g_2^{x_{i,2}} g_3^{x_{i,3}} \,\}_{i=1}^{n} \wedge$$
$$\{\, f_{i,1} = g_1^{y_{i,1}} g_3^{y_{i,3}} \,\}_{i=0}^{(n+4)} \wedge \{\, f_{i,2} = g_2^{y_{i,2}} g_3^{y_{i,3}} \,\}_{i=0}^{(n+4)} \wedge$$
$$v = \prod_{j=1}^{3} \left(\prod_{i=0}^{3} \mathsf{e}(u_j, u_i)^{y_{i,j}} \cdot \prod_{i=4}^{n+3} \mathsf{e}(u_j, c_{i-3})^{y_{i,j}} \cdot \mathsf{e}(u_j, L)^{y_{(n+4),j}} \right) \wedge$$
$$\{\, m_i = c_i \cdot \prod_{j=1}^{3} u_j^{-x_{i,j}} \,\}_{i=1}^{n}$$

C.5 Proof of Lemma 3

Proof sketch of Lemma 3: We proceed in a sequence of games. We start with a game where the environment interacts with the real protocol, and end up with a game where

the environment interacts with the simulator and the ideal system. Then we show that all those games are computationally indistinguishable. Let W_i denote the event that the environment $\mathcal{E}$ outputs 1 in Game i.

Game 0**.** This is the real protocol run.

Game 1**.** This game is the same as Game 0, except that the game aborts if the environment controlling the corrupted user sends two different openings for one of the commitments Com_{id} or $Com_{x_1'}$ as part of its input to $\mathcal{I}_{\mathsf{tpc}_1}$ and $\mathcal{I}_{\mathsf{tpc}_2}$. An environment that distinguishes between Game 0 and Game 1 breaks the binding property of the commitment scheme which for Pedersen commitments would contradict the Discrete Logarithm assumption. Therefore $|\Pr[W_1] - \Pr[W_0]| = \mathrm{negl}(k)$.

Game 2**.** This game is the same as Game 1, except that the checks of relations in the zero-knowledge functionality and the two party computation are turned off for honest parties, and that the real commitment of the service provider is replaced by a random commitment. Honest users never do proofs that wouldn't verify, and commitments are perfectly hiding. Therefore $\Pr[W_2] = \Pr[W_1]$.

Game 3**.** This game is the same as Game 2, except that, if SA is honest, the ciphertext ct_1 is replaced with an encryption of 1. By Lemma 5, $|\Pr[W_3] - \Pr[W_2]| = \mathrm{negl}(k)$.

Game 4**.** Game 4 replaces the control logic of the real protocol with the control logic of the simulator and the ideal functionality. No further cryptographic values need to be changed. Therefore $\Pr[W_4] = \Pr[W_3]$.

□

Lemma 5. *If* U *is corrupted,* SP *and* SA *are honest, and* RA *is either honest or corrupted* $|\Pr[W_3] - \Pr[W_2]| = \mathrm{negl}(k)$*, if* Keygen, Enc, Dec *is a CCA secure encryption scheme.*

Proof. The proof is by contradiction, by showing a reduction from a distinguishing environment $\mathcal{E}$ to a successful CCA adversary $\mathcal{A}$. $\mathcal{A}$ receives the public key pk as input and playing the role of the honest SA, registers it with $\mathcal{I}_{\mathsf{reg}}$. Depending on the bit b of the CCA challenger, the adversary will (without knowing it himself) either simulate Game 2 or Game 3 towards $\mathcal{E}$.

$\mathcal{A}$ follows the instructions of the games but uses the decryption oracle to decrypt messages. When the ciphertext ct_1 needs to be created, the CCA adversary $\mathcal{A}$ asks for a challenge ciphertext ct by sending $m_0 = g^{x_1+x_2}$ and $m_1 = 1$ to the CCA challenge oracle and uses the result as ct_1. For the rest of the interactions with $\mathcal{E}$, $\mathcal{A}$ follows the joint instructions of the games and forwards the output of $\mathcal{E}$ as its guess.

If the bit b chosen by the CCA challenge game is 0 the behavior of the CCA adversary perfectly follows the behavior of Game 2, otherwise it corresponds to Game 3. Consequently, $\mathcal{A}$ has the same advantage as $\mathcal{E}$. □

Standards & Software Libraries

[1] Credential-based authentication framework – user guide, June 2011. IBM Research – Zurich.

[2] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. RFC 5280 - Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, May 2008.

[3] Efficient implementation of DAA on Java card smart cards, 2009. KU Leuven – COSIC.

[4] T. Dierks and E. Rescorla. The transport layer security (TLS) protocol version 1.2, August.

[5] Olivier Dubuisson and Philippe Fouquart. *ASN.1: communication between heterogeneous systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[6] Duemmegi s.r.l. home and building automation.

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC 2616, hypertext transfer protocol – HTTP/1.1, 1999.

[8] Tim France-Massay. MULTOS - the high security smart card OS. Technical report, MAOSCO Limited, 2005.

[9] A. Frier, P. Karlton, and P. Kocher. The ssl 3.0 protocol. *Netscape Communications Corp*, 18:2780, 1996.

[10] GNU multiple precision arithmetic library (GMP), 2009. `http://gmplib.org/`.

[11] Specification of the Identity Mixer cryptographic library – version 2.3.2, 2010. IBM Research – Zurich.

[12] ISO 18004:2006. Information technology – automatic identification and data capture techniques – QR code 2005 bar code symbology specification, September 2006.

[13] J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. RFC 3447 (Informational), February 2003.

[14] Java pairing based cryptography library (jPBC), 2012. `http://gas.dia.unisa.it/projects/jpbc/index.html`.

[15] J. Louvel and T. Boileau. *Restlet: Official Developer's Guide to Restful Web Applications in Java*. Apress, 1 edition, 2010.

[16] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet public key infrastructure online certificate status protocol - OCSP, 1999.

[17] Pairing-based cryptography library (PBC) by Ben Lynn, 2009. `http://crypto.stanford.edu/pbc/`.

[18] PriMan: a privacy-preserving identity framework, 2009.

[19] Christian Paquin Stefan Brands. U-Prove cryptographic specification v1.0, 2010. Microsoft Corporation.

[20] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. RFC 3820 - Internet X.509 public key infrastructure (PKI) proxy certificate profile., 2004.

[21] ZXing - multi-format 1D/2D barcode image processing library with clients for Android, Java, 2011. `http://code.google.com/p/zxing`.

Bibliography

[AB05] Ali Alkar and Umit Buhur. An internet based wireless home automation system for multifunctional devices. *IEEE Transactions on Consumer Electronics*, 51(4):1169 – 1174, November 2005.

[ABD+03] Tero Alamäki, Margareta Björksten, Peter Dornbach, Casper Gripenberg, Norbert Gyorbiro, Gabor Márton, Zoltan Nemeth, Timo Skytta, and Mikko Tarkiainen. Privacy enhancing service architectures. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies*, volume 2482 of *Lecture Notes in Computer Science*, pages 204–208. Springer Berlin / Heidelberg, 2003.

[ABK+11] Gildas Avoine, Muhammed Ali Bingöl, Süleyman Kardaş, Cédric Lauradoux, and Benjamin Martin. A framework for analyzing RFID distance bounding protocols. *Journal of Computer Security*, 19(2):289–317, April 2011.

[ABV12] Gergely Alpár, Lejla Batina, and Roel Verdult. Using NFC phones for proving credentials. In Jens Schmitt, editor, *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, volume 7201 of *Lecture Notes in Computer Science*, pages 317–330. Springer Berlin / Heidelberg, 2012.

[ACF+04] Franco Arcieri, Mario Ciclosi, Fabio Fioravanti, Enrico Nardelli, and Maurizio Talamo. The Italian electronic identity card: a short introduction. In *The National Conference on Digital Government Research*, 2004.

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer Berlin / Heidelberg, 2000.

[Acq04a] Alessandro Acquisti. Privacy in electronic commerce and the economics of immediate gratification. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 21–29, New York, NY, USA, 2004. ACM.

[Acq04b] Alessandro Acquisti. Receipt-free homomorphic elections and write-in ballots. IACR Cryptology ePrint Archive, Report 2004/105, 2004. `http://eprint.iacr.org/2004/105`.

[AM03] Adil Alsaid and David Martin. Detecting web bugs with bugnosis: privacy advocacy through education. In *Proceedings of the 2nd international conference on Privacy enhancing technologies*, PET'02, pages 13–26, Berlin, Heidelberg, 2003. Springer-Verlag.

[ARR09] Janna Anderson, Harrison Rainie, and Lee Rainie. *Ubiquity, Mobility, Security: The Future of the Internet III*. Future of the Internet. Cambria Press, 2009.

[ASM06] Man Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer Berlin / Heidelberg, 2006.

[AST02] Giuseppe Ateniese, Dawn Song, and Gene Tsudik. Quasi-efficient revocation of group signatures. In *Proceedings of the 6th international conference on Financial cryptography*, pages 183–197. Springer-Verlag, 2002.

[AT99] Giuseppe Ateniese and Gene Tsudik. Some open issues and new directions in group signatures. In Matthew Franklin, editor, *Financial Cryptography*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211. Springer Berlin / Heidelberg, 1999.

[AT09] Gildas Avoine and Aslan Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Ardagna, editors, *Information Security*, volume 5735 of *Lecture Notes in Computer Science*, pages 250–261. Springer Berlin / Heidelberg, 2009.

[ATSM09] Man Au, Patrick Tsang, Willy Susilo, and Yi Mu. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009*, volume 5473 of *Lecture Notes in*

Computer Science, pages 295–308. Springer Berlin / Heidelberg, 2009.

[Bal08] Josep Balasch. *Smart Card Implementation of Anonymous Credentials*. Master's thesis, KU Leuven, 2008.

[Bal09] Shane Balfe. *Secure Payment Architectures and Other Applications of Trusted Computing*. PhD thesis, Royal Holloway, London, 2009.

[BBD+91] Samy Bengio, Gilles Brassard, Yvo G. Desmedt, Claude Goutier, and Jean-Jacques Quisquater. Secure implementation of identification systems. *Journal of Cryptology*, 4:175–183, 1991.

[BBF01] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security*, 4(3):191–233, August 2001.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 227–242. Springer Berlin / Heidelberg, 2004.

[BC94] Stefan Brands and David Chaum. Distance-bounding protocols. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer Berlin / Heidelberg, 1994.

[BC10] Patrik Bichsel and Jan Camenisch. Mixing identities with ease. In Elisabeth de Leeuw, Simone Fischer-Hübner, and Lothar Fritsch, editors, *Policies and Research in Identity Management*, volume 343 of *IFIP Advances in Information and Communication Technology*, pages 1–17. Springer Boston, 2010.

[BCC04] Ernest Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145. ACM, 2004.

[BCDvdG06] Ernest Brickell, David Chaum, Ivan Damgård, and Jeroen van de Graaf. Gradual and verifiable release of a secret (extended abstract). In Carl Pomerance, editor, *Advances in Cryptology - CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 156–166. Springer Berlin / Heidelberg, 2006.

[BCGS09] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard java card. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 600–610, New York, NY, USA, 2009. ACM.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *Proceedings of the 5th conference on Theory of cryptography*, TCC'08, pages 356–374, Berlin, Heidelberg, 2008. Springer-Verlag.

[BCL+05] Boaz Barak, Ran Canetti, Yehuda Lindell, Rafael Pass, and Tal Rabin. Secure computation without authentication. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 361–377. Springer Berlin / Heidelberg, 2005.

[BCL06] Endre Bangerter, Jan Camenisch, and Anna Lysyanskaya. A cryptographic framework for the controlled release of certified data. In Bruce Christianson, Bruno Crispo, James Malcolm, and Michael Roe, editors, *Security Protocols*, volume 3957 of *Lecture Notes in Computer Science*, pages 20–42. Springer Berlin / Heidelberg, 2006.

[BCM05] Endre Bangerter, Jan Camenisch, and Ueli Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 154–171. Springer Berlin / Heidelberg, 2005.

[BCPP01] Boris Balacheff, Liqun Chen, David Plaquin, and Graeme Proudler. A trusted process to digitally sign a document. In *Proceedings of the 2001 workshop on New security paradigms*, NSPW '01, pages 79–86, New York, NY, USA, 2001. ACM.

[BCS05] Michael Backes, Jan Camenisch, and Dieter Sommer. Anonymous yet accountable access control. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 40–46, New York, NY, USA, 2005. ACM.

[BDDD07] Stefan Brands, Liesje Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *Information Security and Privacy*, volume 4586 of *Lecture Notes in Computer Science*, pages 400–415. Springer Berlin / Heidelberg, 2007.

[BDDL+12] Patrik Bichsel, Bart De Decker, Jorn Lapon, Vincent Naessens, and Dieter Sommer. Data-minimizing authentication goes mobile. In *Communications and Multimedia Security*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012. (Accepted).

[BDK01] Peter Bergstrom, Kevin Driscoll, and John Kimball. Making home automation communications secure. *Computer*, 34(10):50–56, October 2001.

[BdM94] Josh Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285. Springer Berlin / Heidelberg, 1994.

[BF99] Dirk Balfanz and Edward W. Felten. Hand-held computers can be better smart cards. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 2–2, Berkeley, CA, USA, 1999. USENIX Association.

[BFGI11] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. In Peeter Laud, editor, *NordSec*, volume 7161 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2011.

[BFK01] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable internet access. In *Designing Privacy Enhancing Technologies*, pages 115–129. Springer, 2001.

[BGR98] Mihir Bellare, Juan Garay, and Tal Rabin. Fast batch verification for modular exponentiation and digital signatures. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 236–250. Springer Berlin / Heidelberg, 1998.

[BJ05] Sandford Bessler and Oliver Jorns. A privacy enhanced service architecture for mobile users. In *Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops*, pages 125 – 129, March 2005.

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer Berlin / Heidelberg, 2000.

[BP97] Niko Baric and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer Berlin / Heidelberg, 1997.

[BPW07] Michael Backes, Birgit Pfitzmann, and Michael Waidner. The reactive simulatability RSIM framework for asynchronous systems. *Information and Computation*, 205(12):1685 – 1720, 2007.

[Bra00] Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.

[BS01] Emmanuel Bresson and Jacques Stern. Efficient revocation in group signatures. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 190–206. Springer Berlin / Heidelberg, 2001.

[BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 168–177, New York, NY, USA, 2004. ACM.

[BSSW02] Dirk Balfanz, Diana K. Smetters, Paul Stewart, and H. Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the Network and Distributed System Security Symposium*. The Internet Society, 2002.

[BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, STOC '94, pages 544–553, New York, NY, USA, 1994. ACM.

[Cam98] Jan Camenisch. *Group signature schemes and payment systems based on the discrete logarithm problem*. PhD thesis, Technische Wissenschaften ETH Zürich, Zürich, 1998.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 2001 IEEE International Conference on Cluster Computing*, number 2000/067, pages 136–145. IEEE Comput. Soc, 2001.

[Cas11] Antonio Diaz Castaño. Anonymous e-petitions with secure hardware for smart phones. Master's thesis, KU Leuven - ESAT, 2011. Bart Preneel and Juan Carlos Burguillo-Rial (promotors).

[CCGS10] Jan Camenisch, Nathalie Casati, Thomas Gross, and Victor Shoup. Credential authenticated identification and key exchange. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 255–276. Springer Berlin / Heidelberg, 2010.

[CCM08] Michael Clarkson, Stephen. Chong, and Andrew Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy, 2008. SP 2008*, pages 354 –368, May 2008.

[CD00] Jan Camenisch and Ivan Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer Berlin / Heidelberg, 2000.

[CDK+11] Jan Camenisch, Maria Dubovitskaya, Markulf Kohlweiss, Jorn Lapon, and Gregory Neven. Cryptographic mechanisms for privacy. In Jan Camenisch, Simone Fischer-Hübner, and Kai Rannenberg, editors, *Privacy and Identity Management for Life*, pages 117–134. Springer Berlin Heidelberg, 2011.

[CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In Salil Vadhan, editor, *Theory of Cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 61–85. Springer Berlin / Heidelberg, 2007.

[CFN90] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings on Advances in cryptology*, CRYPTO '88, pages 319–327, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[CFT98] Agnes Chan, Yair Frankel, and Yiannis Tsiounis. Easy come - easy go divisible cash. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 561–575. Springer Berlin / Heidelberg, 1998.

[CG05] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer Berlin / Heidelberg, 2005.

[CG08] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 345–356, New York, NY, USA, 2008. ACM.

[CGHB08] Jan Camenisch, Thomas Groß, and Thomas S. Heydt-Benjamin. Rethinking accountable privacy supporting services: extended abstract. In *ACM DIM – Digital Identity Management*, pages 1–8, 2008.

[Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald Rivest, , and Alan Sherman, editors, *Advances in Cryptology Proceedings of Crypto*, volume 82, pages 199–203. Plenum Publishing, 1983.

[Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commununications of the ACM*, 28(10):1030–1044, October 1985.

[Che06] Jung Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin / Heidelberg, 2006.

[CHK+06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 201–210, New York, NY, USA, 2006. ACM.

[CHK+11a] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Structure preserving CCA secure encryption and applications. In Dong Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 89–106. Springer Berlin / Heidelberg, 2011.

[CHK+11b] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Structure preserving CCA secure encryption and its application to oblivious third parties. IACR Cryptology ePrint Archive, Report 2011/319, 2011. `http://eprint.iacr.org/2011/319`.

[CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, Irvine, pages 481–500, Berlin, Heidelberg, 2009. Springer-Verlag.

[CKS10] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. Solving revocation with efficient update of anonymous credentials. In Juan Garay and Roberto De Prisco, editors, *Security and Cryptography for Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 454–471. Springer Berlin / Heidelberg, 2010.

[CKS11] Jan Camenisch, Stephan Krenn, and Victor Shoup. A framework for practical universally composable zero-knowledge protocols. IACR Cryptology ePrint Archive, Report 2011/228, 2011. http://eprint.iacr.org/2011/228.

[CKST01] Suresh Chari, Parviz Kermani, Sean Smith, and Ros Tassiulas. Security issues in m-commerce: A usage-based taxonomy. e-commerce agents. In *LNAI*, pages 264–282. Springer, 2001.

[CKY09] Jan Camenisch, Aggelos Kiayias, and Moti Yung. On the portability of generalized Schnorr proofs. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 425–442. Springer Berlin / Heidelberg, 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Berlin / Heidelberg, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 101–120. Springer Berlin / Heidelberg, 2002.

[CL03] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer Berlin / Heidelberg, 2003.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 1–6. Springer Berlin / Heidelberg, 2004.

[Cla02] Joris Claessens. *Analysis and design of an advanced infrastructure for secure and anonymous electronic payment systems on the Internet*. PhD thesis, Katholieke Universiteit Leuven, 2002. Bart Preneel and Joos Vandewalle (promotors).

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation.

In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 494–503, New York, NY, USA, 2002. ACM.

[CLS+01] Michael Covington, Wende Long, Srividhya Srinivasan, Anind Dev, Mustaque Ahamad, and Gregory Abowd. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, SACMAT '01, pages 10–20, New York, NY, USA, 2001. ACM.

[CM98] Jan Camenisch and Markus Michels. A group signature scheme with improved efficiency (Extended Abstract). In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology - ASIACRYPT'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 160–174. Springer Berlin / Heidelberg, 1998.

[CM99] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122. Springer Berlin / Heidelberg, 1999.

[CMN+10] Jan Camenisch, Sebastian Mödersheim, Gregory Neven, Franz-Stefan Preiss, and Dieter Sommer. A card requirements language enabling privacy-preserving access control. In *Proceedings of the 15th ACM symposium on Access control models and technologies*, SACMAT '10, pages 119–128, New York, NY, USA, 2010. ACM.

[CMS10] Jan Camenisch, Sebastian Mödersheim, and Dieter Sommer. A formal model of identity mixer. In Stefan Kowalewski and Marco Roveri, editors, *Formal Methods for Industrial Critical Systems*, volume 6371 of *Lecture Notes in Computer Science*, pages 198–214. Springer Berlin / Heidelberg, 2010.

[CNdS10] Paul Crocker, Vasco Nicolau, and Simão Melo de Sousa. Sniffing with the Portuguese identify card for fun and profit. In *Proceedings of the 9th European Conference on Information Warfare and Security*, pages 43–55. Academic Conferences Limited, 2010.

[CP93] David Chaum and Torben Pedersen. Wallet databases with observers. In Ernest Brickell, editor, *Advances in Cryptology - CRYPTO' 92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer Berlin / Heidelberg, 1993.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In Burton Kaliski, editor, *Advances*

in Cryptology - CRYPTO '97, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer Berlin / Heidelberg, 1997.

[CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer Berlin / Heidelberg, 2003.

[CSZ06] Jan Camenisch, Dieter Sommer, and Roger Zimmermann. A general certification framework with applications to privacy-enhancing certificate infrastructures. In Simone Fischer-Hübner, Kai Rannenberg, Louise Yngström, and Stefan Lindskog, editors, *Security and Privacy in Dynamic Environments*, volume 201 of *IFIP International Federation for Information Processing*, pages 25–37. Springer Boston, 2006.

[CWP06] Danny De Cock, Christopher Wolf, and Bart Preneel. The Belgian electronic identity card (Overview). In Jana Dittmann, editor, *Sicherheit*, volume 77 of *LNI*, pages 298–301. GI, 2006.

[CWW+04] Zewen Chen, Jilin Wang, Yumin Wang, Jiwu Huang, and Daren Huang. An efficient revocation algorithm in group signatures. In Jong-In Lim and Dong-Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 339–351. Springer Berlin / Heidelberg, 2004.

[CZBP06] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer Berlin / Heidelberg, 2006.

[Dan07] Luuk Danes. *Smart Card Integration in the pseudonym system idemix*. Master's thesis, University of Groningen, 2007.

[DDD05] Liesje Demuynck and Bart De Decker. Privacy-preserving electronic health records. In Jana Dittmann, Stefan Katzenbeisser, and Andreas Uhl, editors, *Communications and Multimedia Security*, volume 3677 of *Lecture Notes in Computer Science*, pages 150–159. Springer Berlin / Heidelberg, 2005.

[DDD06] Liesje Demuynck and Bart De Decker. How to prove list membership in logarithmic time. Technical report, KU Leuven, Department of Computer Science, 2006.

[DE02] Steve Dohrmann and Carl Ellison. Public-key support for collaborative groups. In *Proceedings of the 1st Annual PKI Research Workshop*, pages 139–148, 2002.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 77–85. Springer Berlin / Heidelberg, 2002.

[DFM01] Roger Dingledine, Michael Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies*, volume 2009 of *Lecture Notes in Computer Science*, pages 67–95. Springer Berlin / Heidelberg, 2001.

[Die10] Kurt Dietrich. Anonymous credentials for Java enabled platforms: A performance evaluation. In Liqun Chen and Moti Yung, editors, *Trusted Systems*, volume 6163 of *Lecture Notes in Computer Science*, pages 88–103. Springer Berlin / Heidelberg, 2010.

[DKD+09] Claudia Diaz, Eleni Kosta, Hannelore Dekeyser, Markulf Kohlweiss, and Girma Enideg Nigusse. Privacy preserving electronic petitions. *Identity in the Information Society*, 1(1):203–209, 2009.

[DKR06] S. Delaune, S. Kremer, and M. Ryan. Coercion-resistance and receipt-freeness in electronic voting. In *Computer Security Foundations Workshop, 2006. 19th IEEE*, pages 12 pp. –42, 2006.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium-Volume 13*, pages 21–21. USENIX Association, 2004.

[DPG06] Armando Roy Delgado, Rich Picking, and Vic Grout. On context in authorization policy. In *Proceedings of the 6th International Network Conference (INC 2006)*, pages 357–366, 2006.

[DSW08] Yevgeniy Dodis, Victor Shoup, and Shabsi Walfish. Efficient constructions of composable commitments and zero-knowledge proofs. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 515–535. Springer Berlin / Heidelberg, 2008.

[DT08] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. IACR Cryptology ePrint Archive, Report 2008/538, 2008. http://eprint.iacr.org/2008/538.

[Dum05] Jos Dumortier. eID en de paradoks van het Rijksregisternummer, 2005.

[DW09] Kurt Dietrich and Johannes Winter. Implementation aspects of mobile and embedded trusted computing. In Liqun Chen, Chris Mitchell, and Andrew Martin, editors, *Trusted Computing*, volume 5471 of *Lecture Notes in Computer Science*, pages 29–44. Springer Berlin / Heidelberg, 2009.

[Eck10] Peter Eckersley. How unique is your web browser? In Mikhail Atallah and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin / Heidelberg, 2010.

[FHM11] Chun-I Fan, Ruei-Hau Hsu, and Mark Manulis. Group signature with constant revocation costs for signers and verifiers. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *Cryptology and Network Security*, volume 7092 of *Lecture Notes in Computer Science*, pages 214–233. Springer Berlin / Heidelberg, 2011.

[FID06] FIDIS. D3.6 Study on ID Documents. Technical report, Future of Identity in the Information Society, 2006.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton Kaliski, editor, *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin / Heidelberg, 1997.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew Odlyzko, editor, *Advances in Cryptology - CRYPTO' 86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Berlin / Heidelberg, 1987.

[FSG+01] David Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3):224–274, August 2001.

[GHHF05] Hidehito Gomi, Makoto Hatakeyama, Shigeru Hosono, and Satoru Fujita. A delegation framework for federated identity management. In *Proceedings of the 2005 workshop on Digital identity management*, DIM '05, pages 94–103, New York, NY, USA, 2005. ACM.

[GM07] Eimear Gallery and Chris Mitchell. Trusted mobile platforms. In *Foundations of security analysis and design IV*, pages 282–323. Springer-Verlag, 2007.

[GN04] Christian Gehrmann and Kaisa Nyberg. Manual authentication for wireless devices. *RSA Cryptobytes*, 7:2004, 2004.

[GRB03] Tal Garfinkel, Mendel Rosenblum, and Dan Boneh. Flexible OS support and applications for trusted computing. In *Proceedings of the 9th conference on Hot Topics in Operating Systems-Volume 9*, pages 25–25. USENIX Association, 2003.

[GRS96] David Goldschlag, Michael Reed, and Paul Syverson. Hiding routing information. In Ross Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer Berlin / Heidelberg, 1996.

[GSS+06] Michael Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. Loud and clear: Human-verifiable authentication based on audio. In *26th IEEE International Conference on Distributed Computing Systems, 2006. ICDCS 2006*, pages 10–10. IEEE, 2006.

[GST95] Howard Gobioff, Sean Smith, and J. D. Tygar. Smart cards in hostile environments. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pages 23–28, 1995.

[Heu10] Marcel Heupel. *Porting and evaluating the performance of idemix and tor anonymity on modern smartphones*. Master's thesis, University of Siegen, 2010.

[HH10] Jaap-Henk Hoepman and George Huitema. Privacy enhanced fraud resistant road pricing. In Jacques Berleur, Magda Hercheui, and Lorenz Hilty, editors, *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, volume 328 of *IFIP Advances in Information and Communication Technology*, pages 202–213. Springer Boston, 2010.

[HK05] Gerhard Hancke and Markus Kuhn. An RFID distance bounding protocol. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005*, pages 67 – 73, September 2005.

[JA08] Audun Jøsang and Bander AlFayyadh. Robust WYSIWYS: a method for ensuring that what you see is what you sign. In *Proceedings of the sixth Australasian conference on Information security - Volume 81*, AISC '08, pages 53–58, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc.

[JBLG05] James Joshi, Elisa Bertino, Usman Latif, and Arif Ghafoor. A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):4–23, January 2005.

[JCC06] Jongpil Jeong, Min Chung, and Hyunseung Choo. Secure user authentication mechanism in digital home network environments. In Edwin Sha, Sung-Kook Han, Cheng-Zhong Xu, Moon-Hae Kim, Laurence Yang, and Bin Xiao, editors, *Embedded and Ubiquitous Computing*, volume 4096 of *Lecture Notes in Computer Science*, pages 345–354. Springer Berlin / Heidelberg, 2006.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 61–70, New York, NY, USA, 2005. ACM.

[JGK05] Wayne Jansen, Serban Gavrila, and Vlad Korolev. Proximity-based authentication for mobile devices. In Hamid R. Arabnia, editor, *Security and Management*, pages 398–404. CSREA Press, 2005.

[JPH02] Audun Jsang, Dean Povey, and Anthony Ho. What you see is not always what you sign. In *AAUG 2002*, 2002.

[KAK+09] Chong Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID distance bounding protocol. In Pil Lee and Jung Cheon, editors, *Information Security and Cryptology - ICISC 2008*, volume 5461 of *Lecture Notes in Computer Science*, pages 98–115. Springer Berlin / Heidelberg, 2009.

[KCKP02] Vassilis Kapsalis, Konstantinos Charatsis, Athanasios Kalogeras, and George Papadopoulos. Web gateway: A platform for industry services over internet. In *2002 IEEE International Symposium on Industrial Electronics IEEE-ISIE'2002*, pages 73–77, 2002.

[KDMR08] Ralf Küsters, Anupam Datta, John Mitchell, and Ajith Ramanathan. On the relationships between notions of simulation-based security. *Journal of Cryptology*, 21:492–546, 2008.

[KFF+07] Markulf Kohlweiss, Sebastian Faust, Lothar Fritsch, Bartek Gedrojc, and Bart Preneel. Efficient oblivious augmented maps: Location-based services with a payment broker. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 77–94. Springer Berlin / Heidelberg, 2007.

[KM07] Neal Koblitz and Alfred J. Menezes. Another look at "provable security". *Journal of Cryptology*, 20:3–37, 2007. 10.1007/s00145-005-0432-z.

[KNH+09] Takaaki Komura, Yasuhiro Nagai, Shoichi Hashimoto, Makiko Aoyagi, and Kenji Takahashi. Proposal of delegation using electronic certificates on single sign-on system with SAML-protocol. In *Ninth Annual International Symposium on Applications and the Internet, 2009. SAINT '09*, pages 235 –238, July 2009.

[KT08] Ralf Küsters and Max Tuengerthal. Joint state theorems for public-key encryption and digital signature functionalities with local computation. In *Proceedings of the 2008 21st IEEE Computer Security Foundations Symposium*, CSF '08, pages 270–284, Washington, DC, USA, 2008. IEEE Computer Society.

[KT11] Ralf Küsters and Max Tuengerthal. Composition theorems without pre-established session identifiers. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 41–50, New York, NY, USA, 2011. ACM.

[Küs06] Ralf Küsters. Simulation-based security with inexhaustible interactive turing machines. In *Computer Security Foundations Workshop, 2006. 19th IEEE*, pages 12–320. IEEE, 2006.

[KZG07] Aniket Kate, Greg Zaverucha, and Ian Goldberg. Pairing-based onion routing. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4776 of *Lecture Notes in Computer Science*, pages 95–112. Springer Berlin / Heidelberg, 2007.

[Lan01] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In Gregory Abowd, Barry Brumitt, and Steven Shafer, editors, *Ubicomp 2001: Ubiquitous Computing*, volume 2201 of *Lecture Notes in Computer Science*, pages 273–291. Springer Berlin / Heidelberg, 2001.

[LHP02] Herbert Leitold, Arno Hollosi, and Reinhard Posch. Security architecture of the Austrian citizen card concept. In *Computer*

Security Applications Conference, 2002. Proceedings. 18th Annual, pages 391–400, 2002.

[LKDDN10] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Performance analysis of accumulator-based revocation mechanisms. In Kai Rannenberg, Vijay Varadharajan, and Christian Weber, editors, *Security and Privacy - Silver Linings in the Cloud*, volume 330 of *IFIP Advances in Information and Communication Technology*, pages 289–301. Springer Boston, 2010.

[LKDDN11] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Analysis of revocation strategies for anonymous Idemix credentials. In Bart De Decker, Jorn Lapon, Vincent Naessens, and Andreas Uhl, editors, *Communications and Multimedia Security*, volume 7025 of *Lecture Notes in Computer Science*, pages 3–17. Springer Berlin / Heidelberg, 2011.

[LL11] Lezhen Ling and Junguo Liao. Anonymous electronic voting protocol with traceability. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 59 –66, December 2011.

[LLSL09] Kuan-Chieh Liao, Wei-Hsun Lee, Min-Hsuan Sung, and Ting-Ching Lin. A one-time password scheme with QR-code based on mobile phone. In *Fifth International Joint Conference on INC, IMS and IDC, 2009. NCM '09.*, pages 2069 –2071, August 2009.

[LLX07] Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 4521 of *Lecture Notes in Computer Science*, pages 253–269. Springer Berlin / Heidelberg, 2007.

[LNV+10] Jorn Lapon, Vincent Naessens, Bram Verdegem, Pieter Verhaeghe, and Bart De Decker. Building advanced applications with the Belgian eID. *Security and Communication Networks*, 3(5):439–451, 2010.

[LVNV08] Jorn Lapon, Koen Vangheluwe, Vincent Naessens, and Annemie Vorstermans. A generic architecture for secure home automation servers. In Luc De Backer, editor, *Proceedings of the Third International European Conference on the Use of Modern Information and Communication Technologies,*. Nevelland vzw, 2008.

[LVV+08] Jorn Lapon, Kristof Verslype, Pieter Verhaeghe, Bart De Decker, and Vincent Naessens. PetAnon: a fair and privacy-preserving petition

system. *The Future of Identity in the Information Society. Challenges for Privacy and Security: Pre-proceedings*, 1:73–78, 2008. Presented at IFIP Summer School on Internet Security and Privacy 2008 in Brno, Czech Republic.

[LVV+09] Jorn Lapon, Bram Verdegem, Pieter Verhaeghe, Vincent Naessens, and Bart Decker. Extending the Belgian eID technology with mobile security functionality. In Andreas U. Schmidt, Shiguo Lian, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson, editors, *Security and Privacy in Mobile Information and Communication Systems*, volume 17 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 97–109. Springer Berlin Heidelberg, 2009.

[Mar11] Marian Margraf. The new German ID card. In Norbert Pohlmann, Helmut Reimer, and Wolfgang Schneider, editors, *ISSE 2010 Securing Electronic Business Processes*, pages 367–373. Vieweg+Teubner, 2011.

[McD03] Patrick McDaniel. On context in authorization policy. In *Proceedings of the ACM Symposium on Access Control Models and Technologies*, pages 80–89, 2003.

[MG07] Rene Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In Anthony LaMarca, Marc Langheinrich, and Khai Truong, editors, *Pervasive Computing*, volume 4480 of *Lecture Notes in Computer Science*, pages 144–161. Springer Berlin / Heidelberg, 2007.

[MMS+07] Andres Marin, Wolfgang Mueller, Robbie Schaefer, Florina Almenarez, Daniel Diaz, and Max Ziegler. Middleware for secure home access and control. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07*, pages 489 –494, March 2007.

[MPR09] Jonathan McCune, Adrian Perrig, and Michael Reiter. Seeing-Is-Believing: using camera phones for human-verifiable authentication. *International Journal of Security and Networks*, 4(1):43–56, 2009.

[MRB09] Arpita Mondal, Kaushik Roy, and Prabir Bhattacharya. Secure and simplified access to home appliances using iris recognition. In *IEEE Workshop on Computational Intelligence in Biometrics: Theory,*

Algorithms, and Applications, 2009. CIB 2009, pages 22 –29, April 2009.

[MV11] Wojciech Mostowski and Pim Vullers. Efficient U-Prove implementation for anonymous credentials on smart cards. In George Kesidis and Haining Wang, editors, *7th International ICST Conference on Security and Privacy in Communication Networks, SecureComm 2011, London, UK, September 7-9, 2011. Proceedings*, volume 96 of *Lecture Notes of the Institute for ComputerSciences, Social-Informatics and Tele-communications Engineering (LNICST)*, pages 243–260. Springer-Verlag, 2011. (to appear).

[NDDD06] Vincent Naessens, Liesje Demuynck, and Bart De Decker. A fair anonymous submission and review system. In *Proceedings of the 10th IFIP TC-6 TC-11 international conference on Communications and Multimedia Security*, CMS'06, pages 43–53, Berlin, Heidelberg, 2006. Springer-Verlag.

[NF05] Toru Nakanishi and Nobuo Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In Bimal Roy, editor, *Advances in Cryptology - ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 533–548. Springer Berlin / Heidelberg, 2005.

[NF06] Toru Nakanishi and Nobuo Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In Hiroshi Yoshiura, Kouichi Sakurai, Kai Rannenberg, Yuko Murayama, and Shinichi Kawamura, editors, *Advances in Information and Computer Security*, volume 4266 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin / Heidelberg, 2006.

[NF08] Toru Nakanishi and Nobuo Funabiki. Efficient revocable group signature schemes using primes. *Journal of Information Processing*, 16:110–121, 2008.

[NFHF09] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography - PKC 2009*, volume 5443 of *Lecture Notes in Computer Science*, pages 463–480. Springer Berlin / Heidelberg, 2009.

[Ngu05a] Lan Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer Berlin / Heidelberg, 2005.

[Ngu05b] Lan Nguyen. Accumulators from bilinear pairings and applications to id-based ring signatures and group membership revocation. IACR Cryptology ePrint Archive, Report 2005/123, 2005. http://eprint.iacr.org/2005/123.

[NKHF05] Toru Nakanishi, Fumiaki Kubooka, Naoto Hamada, and Nobuo Funabiki. Group signature schemes with membership revocation for large groups. In Colin Boyd and Juan Gonzalez Nieto, editors, *Information Security and Privacy*, volume 3574 of *Lecture Notes in Computer Science*, pages 193–226. Springer Berlin / Heidelberg, 2005.

[NS03] Gustaf Neumann and Mark Strembeck. An approach to engineer and enforce context constraints in an RBAC environment. In *Proceedings of the eighth ACM symposium on Access control models and technologies*, SACMAT '03, pages 65–79, New York, NY, USA, 2003. ACM.

[NS04] Toru Nakanishi and Yuji Sugiyama. A group signature scheme with efficient membership revocation for reasonable groups. In *Proc. 9th Australasian Conference on Information Security and Privacy (ACISP 2004), LNCS 3108, pp.336-347*, pages 336–347. Springer-Verlag, 2004.

[NSN04] Lan Nguyen and Rei Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In Pil Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 89–102. Springer Berlin / Heidelberg, 2004.

[OHB06] Rolf Oppliger, Ralf Hauser, and David Basin. SSL/TLS session-aware user authenticationŞ or how to effectively thwart the man-in-the-middle. *Computer Communications*, 29(12):2238 – 2246, 2006.

[Pap09] Sebastian Pape. A survey on non-transferable anonymous credentials. In Vashek MatyáŽ, Simone Fischer-Hübner, Daniel Cvrcek, and Petr Łvenda, editors, *The Future of Identity in the Information Society*, volume 298 of *IFIP Advances in Information and Communication Technology*, pages 107–118. Springer Boston, 2009.

[Ped92] Torben Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK, 1992. Springer-Verlag.

[PM08] M. Pitkanen and H. Mikkonen. Initalizing mobile user's identity from federated security infrastructure. In *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM '08*, pages 390 –394, October 2008.

[Pol07] Irene Pollach. What's wrong with online privacy policies? *Commun. ACM*, 50(9):103–108, September 2007.

[Pro05] Tony Proctor. Data security threats in the home environment. In Andy Sloane, editor, *Home-Oriented Informatics and Telematics*, volume 178 of *IFIP International Federation for Information Processing*, pages 133–144. Springer Boston, 2005.

[PS99] Adrian Perrig and Dawn Song. Hash visualization: a new technique to improve real-world security. In *International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.

[PS01] Antti Partanen and Markku Sievänen. FineID specification (S1/v2.1). Technical report, Population Register Centre, 2001.

[PSCP08] Roel Peeters, Koen Simoens, Danny De Cock, and Bart Preneel. Cross-context delegation through identity federation. In Arslan Brömme, Christoph Busch, and Detlef Hühnlein, editors, *BIOSIG*, volume 137 of *LNI*, pages 79–92. GI, 2008.

[PW01] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, SP '01, pages 184–, Washington, DC, USA, 2001. IEEE Computer Society.

[PWVT11] Andreas Poller, Ulrich Waldmann, Sven Vowe, and Sven Turpe. Electronic identity cards for user authentication : Promise and practice. *IEEE Security and Privacy*, 99:46 – 54, 2011.

[RR98] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.

[RR07] L. Richardson and S. Ruby. *RESTful web services*. O'Reilly Series. O'Reilly, 2007.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[SA00] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In Bruce Christianson, Bruno Crispo, James Malcolm, and Michael Roe, editors, *Security Protocols*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–182. Springer Berlin / Heidelberg, 2000.

[Sch91] Claus Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.

[SCL01a] Adrian Spalka, Armin Cremers, and Hanno Langweg. The fairy tale of ‚what you see is what you sign' - trojan horse attacks on software for digital signatures. In *AUUG Conference Proceedings*, 2001.

[SCL01b] Adrian Spalka, Armin Cremers, and Hanno Langweg. Protecting the creation of digital signatures with trusted computing platform technology against attacks by trojan horse programs. In *Proceedings of the 16th international conference on Information security: Trusted information: the new decade challenge*, Sec '01, pages 403–419, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[SFG09] Guenther Starnberger, Lorenz Froihofer, and Karl Goeschka. QR-TAN: Secure mobile transaction authentication. In *International Conference on Availability, Reliability and Security, 2009. ARES '09*, pages 578 –583, March 2009.

[SG01] Umar Saif and David Greaves. Communication primitives for ubiquitous systems or RPC considered harmful. In *ICDCSW '01: Proceedings of the 21st International Conference on Distributed Computing Systems*, page 240, Washington, DC, USA, 2001. IEEE Computer Society.

[SGGO10] Sergio Sánchez García and Ana Gómez Oliva. Improvements of pan-European IDM architecture to enable identity delegation based on X.509 proxy certificates and SAML. In *Proceedings of the 4th IFIP WG 11.2 international conference on Information Security Theory and Practices: security and Privacy of Pervasive Systems and Smart Devices*, WISTP'10, pages 183–198, Berlin, Heidelberg, 2010. Springer-Verlag.

[SGPV09] Michaël Sterckx, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Efficient implementation of anonymous credentials on Java Card smart cards. In *First IEEE International Workshop on Information Forensics and Security, 2009. WIFS 2009*, pages 106 – 110, December 2009.

[SKK08] Andreas Schmidt, Nicolai Kuntze, and Michael Kasper. On the deployment of mobile trusted modules. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 3169–3174. IEEE, 2008.

[Son01] Dawn Xiaodong Song. Practical forward secure group signature schemes. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 225–234, New York, NY, USA, 2001. ACM.

[SP07] Dave Singelée and Bart Preneel. Distance bounding in noisy environments. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, volume 4572 of *Lecture Notes in Computer Science*, pages 101–115. Springer Berlin / Heidelberg, 2007.

[ST03] Christian Schwaiger and Albert Treytl. Smart card based security for fieldbus systems. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, volume 1, pages 398 – 406 vol.1, September 2003.

[STU08] Claudio Soriente, Gene Tsudik, and Ersin Uzun. HAPADEP: Human-assisted pure audio device pairing. In Tzong-Chen Wu, Chin-Laung Lei, Vincent Rijmen, and Der-Tsai Lee, editors, *Information Security*, volume 5222 of *Lecture Notes in Computer Science*, pages 385–400. Springer Berlin / Heidelberg, 2008.

[TR-11] TR-03127. Architecture electronic identity card and electronic resident permit. Technical report, Federal Office for Information Security, 2011.

[TX03] Gene Tsudik and Shouhuai Xu. Accumulating composites and improved group signing. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 269–286. Springer Berlin / Heidelberg, 2003.

[VD09] Kristof Verslype and Bart De Decker. Service and timeframe dependent unlinkable one-time pseudonyms. In Eduardo Fernandez-Medina, Manu Malek, and Javier Hernando, editors, *SECRYPT 2009, Proceedings of the International Conference on Security and Cryptography, Milan, Italy, July 7-10, 2009, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*, pages 13–20. INSTICC Press, 2009.

[VDDN+08] Kristof Verslype, Bart De Decker, Vincent Naessens, Girma Nigusse, Jorn Lapon, and Pieter Verhaeghe. A privacy-preserving ticketing system. In Vijay Atluri, editor, *Data and Applications Security XXII*, volume 5094 of *Lecture Notes in Computer Science*, pages 97–112. Springer Berlin / Heidelberg, 2008.

[VDWDCD11] Gauthier Van Damme, Karel Wouters, Danny De Cock, and Schellekens; Dries. Integrating the Belgian e-ID into Android, 2011. Presented at DROIDCON 2011.

[VLDD+09] Pieter Verhaeghe, Jorn Lapon, Bart De Decker, Vincent Naessens, and Kristof Verslype. Security and privacy improvements for the Belgian eID technology. In Dimitris Gritzalis and Javier Lopez, editors, *Emerging Challenges for Security, Privacy and Trust*, volume 297 of *IFIP Advances in Information and Communication Technology*, pages 237–247. Springer Boston, 2009.

[VLN+08] Pieter Verhaeghe, Jorn Lapon, Vincent Naessens, Bart De Decker, Kristof Verslype, and Girma Enideg Nigusse. Security and privacy threats of the Belgian electronic identity card and middleware, June 2008. Presented at EEMA European e-Identity conference, Den Haag, 10-11 June 2008.

[VLV+08] Kristof Verslype, Jorn Lapon, Pieter Verhaeghe, Vincent Naessens, and Bart De Decker. PetAnon: A privacy-preserving e-petition system based on Idemix. CW Reports CW522, Department of Computer Science, K.U.Leuven, October 2008.

[VSLDL07] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal De Lara. Amigo: proximity-based authentication of mobile devices. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 253–270, Berlin, Heidelberg, 2007. Springer-Verlag.

[VVL+10] Kristof Verslype, Pieter Verhaeghe, Jorn Lapon, Vincent Naessens, and Bart De Decker. PriMan: a privacy-preserving identity framework. In *Proceedings of the 24th annual IFIP WG 11.3 working conference on Data and applications security and privacy*, DBSec'10, pages 327–334, Berlin, Heidelberg, 2010. Springer-Verlag.

[WB90] Samuel Warren and Louis Brandeis. The right to privacy. *Harward Law Review*, 4(5):193–220, December 1890.

[Wes70] Alan Westin. *Privacy and freedom*. Atheneum, New York, 1970.

[ZC09] Fangguo Zhang and Xiaofeng Chen. Cryptanalysis and improvement of an id-based ad-hoc anonymous identification scheme at CT-RSA 05. *Inf. Process. Lett.*, 109(15):846–849, July 2009.

[ZL06] Sujing Zhou and Dongdai Lin. Shorter verifier-local revocation group signatures from bilinear maps. In David Pointcheval, Yi Mu, and Kefei Chen, editors, *Cryptology and Network Security*, volume 4301 of *Lecture Notes in Computer Science*, pages 126–143. Springer Berlin / Heidelberg, 2006.

[ZS10] Gregory Zaverucha and Douglas Stinson. Group testing and batch verification. In *Information Theoretic Security: 4th International Conference, ICITS 2009, Shizuoka, Japan, December 3-6, 2009. Revised Selected Papers*, page 140. Springer-Verlag New York Inc, 2010.

Curriculum Vitae

May 17th, 1979	Born in Veurne, Belgium
1991-1997	Student at V.T.I. Veurne. Received a High School Degree in Industrial Science.
1997-2001	Student at KAHO St. Lieven, Ghent, Department of Engineering. Received a Degree in Electronic Engineering.
2001-2002	Student at KU Leuven, Faculty of Engineering Sciences. Received a Degree of Advanced studies Master in Artificial Intelligence.
2002-2005	Software Engineer at Eculine nv, Antwerp.
2006	Security Coordinator at Prevebo bvba, Veurne,
2006-2007	Software Engineer & Research at the 8ight Day bvba, Ostend.
2008-2010	Student at EHSAL Management School, Brussels. Received a Postgraduate Degree on Industrial Management.
2008-2012	PhD Student at KU Leuven, Faculty of Engineering.
2007-present	Research Assistant at KAHO St. Lieven, Ghent, Department of Engineering, Computer Science
2008-present	Affiliated Researcher at KU Leuven, Department of Computer Science.

List of Publications

Main Publications

Articles in internationally reviewed journals

[LNV+10] Jorn Lapon, Vincent Naessens, Bram Verdegem, Pieter Verhaeghe, and Bart De Decker. Building advanced applications with the Belgian eID. *Security and Communication Networks*, 3(5):439–451, 2010.

Books, internationally recognized scientific publisher – as editor

[DLNU11] Bart De Decker, Jorn Lapon, Vincent Naessens, and Andreas Uhl, editors. *Communications and Multimedia Security, 12th IFIP TC 6 / TC 11 International Conference, CMS 2011, Ghent, Belgium, October 19-21,2011. Proceedings*, volume 7025 of *Lecture Notes in Computer Science*. Springer, 2011.

Article in book, internationally recognized scientific publisher

[CDK+11] Jan Camenisch, Maria Dubovitskaya, Markulf Kohlweiss, Jorn Lapon, and Gregory Neven. Cryptographic mechanisms for privacy. In Jan Camenisch, Simone Fischer-Hübner, and Kai Rannenberg, editors, *Privacy and Identity Management for Life*, pages 117–134. Springer Berlin Heidelberg, 2011.

Papers at international conferences and symposia, published in full in proceedings

[BDDL+12] Patrik Bichsel, Bart De Decker, Jorn Lapon, Vincent Naessens, and Dieter Sommer. Data-minimizing authentication goes mobile. In *Communications and Multimedia Security*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012. (Accepted).

[CHK+11a] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Structure preserving CCA secure encryption and applications. In Dong Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 89–106. Springer Berlin / Heidelberg, 2011.

[LKDDN11] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Analysis of revocation strategies for anonymous Idemix credentials. In Bart De Decker, Jorn Lapon, Vincent Naessens, and Andreas Uhl, editors, *Communications and Multimedia Security*, volume 7025 of *Lecture Notes in Computer Science*, pages 3–17. Springer Berlin / Heidelberg, 2011.

[LKDDN10] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Performance analysis of accumulator-based revocation mechanisms. In Kai Rannenberg, Vijay Varadharajan, and Christian Weber, editors, *Security and Privacy - Silver Linings in the Cloud*, volume 330 of *IFIP Advances in Information and Communication Technology*, pages 289–301. Springer Boston, 2010.

[LVV+09] Jorn Lapon, Bram Verdegem, Pieter Verhaeghe, Vincent Naessens, and Bart Decker. Extending the Belgian eID technology with mobile security functionality. In Andreas U. Schmidt, Shiguo Lian, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson, editors, *Security and Privacy in Mobile Information and Communication Systems*, volume 17 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 97–109. Springer Berlin Heidelberg, 2009.

[LVNV08] Jorn Lapon, Koen Vangheluwe, Vincent Naessens, and Annemie Vorstermans. A generic architecture for secure home automation servers. In Luc De Backer, editor, *Proceedings of the Third International European Conference on the Use of Modern Information and Communication Technologies,*. Nevelland vzw, 2008.

Meeting abstracts, presented at international conferences and symposia

[CHK+11] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Oblivious third parties in the IITM setting. 2011 Grande Region Security and Reliability Day, Trier, 25 March 2011, 2011.

[LKDDN10] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Comparison of accumulator based revocation schemes. Communications and Multimedia Security, Linz, 31 May - 2 June, 2010.

[LDDN09] Jorn Lapon, Bart De Decker, and Vincent Naessens. Trustworthy authentication using Anonymous Proxy Credentials. 4th Benelux Workshop on Information and System Security, Louvain-La-Neuve, 19-20 Novembre 2009, November 2009.

[LVV+08] Jorn Lapon, Kristof Verslype, Pieter Verhaeghe, Bart De Decker, and Vincent Naessens. PetAnon: a fair and privacy-preserving petition system. *The Future of Identity in the Information Society. Challenges for Privacy and Security: Pre-proceedings*, 1:73–78, 2008. Presented at IFIP Summer School on Internet Security and Privacy 2008 in Brno, Czech Republic.

Internal reports

[CHK+11b] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Structure preserving CCA secure encryption and its application to oblivious third parties. IACR Cryptology ePrint Archive, Report 2011/319, 2011.
`http://eprint.iacr.org/2011/319`.

Publications with Contributions as Co-author

Articles in internationally reviewed journals

[VLDDN11a] Jan Vossaert, Jorn Lapon, Bart De Decker, and Vincent Naessens. Symmetric key infrastructure for authenticated key establishment between resource constrained nodes and powerful devices. *Security and Communication Networks*, 2011.

Papers at international conferences and symposia, published in full in proceedings

[VLDDN11b] Jan Vossaert, Jorn Lapon, Bart De Decker, and Vincent Naessens. User-centric identity management using trusted modules. In Jan Camenisch and Costas Lambrinoudakis, editors, *Public Key Infrastructures, Services and Applications,*, pages 155–170. Springer, 2011.

[NLVDD10] Vincent Naessens, Jorn Lapon, Bram Verdegem, and Bart De Decker. A comparison of mechanisms for controlled delegation of electronic tickets. In Luc De Backer, editor, *Proceedings of the Fourth European Conference on the Use of Modern Information and Communication Technologies,*, 2010. Presented at the European conference on the use of modern information and communication technologies, Ghent, 25-26 March 2010.

[VVL+10] Kristof Verslype, Pieter Verhaeghe, Jorn Lapon, Vincent Naessens, and Bart De Decker. PriMan: a privacy-preserving identity framework. In *Proceedings of the 24th annual IFIP WG 11.3 working conference on Data and applications security and privacy*, DBSec'10, pages 327–334, Berlin, Heidelberg, 2010. Springer-Verlag.

[VLDDN10] Jan Vossaert, Jorn Lapon, Bart De Decker, and Vincent Naessens. Personalized mobile services with lightweight security in a sports association. In Andreas U. Schmidt, Giovanni Russello, Antonio Lioy, Neeli R. Prasad, and Shiguo Lian, editors, *Security and Privacy in Mobile Information and Communication Systems,*, pages 3–14. Springer, 2010.

[NSL+09] Vincent Naessens, Tahir Mehmet Sandikkaya, Jorn Lapon, Kristof Verslype, Pieter Verhaeghe, Girma Enideg Nigusse, and Bart De Decker. Privacy policies, tools and mechanisms of the future. In *InetSec 2009,*, pages 125–138. Springer, April 2009.

[VLDD+09] Pieter Verhaeghe, Jorn Lapon, Bart De Decker, Vincent Naessens, and Kristof Verslype. Security and privacy improvements for the Belgian eID technology. In Dimitris Gritzalis and Javier Lopez, editors, *Emerging Challenges for Security, Privacy and Trust*, volume 297 of *IFIP Advances in Information and Communication Technology*, pages 237–247. Springer Boston, 2009.

[VDDN+08] Kristof Verslype, Bart De Decker, Vincent Naessens, Girma Nigusse, Jorn Lapon, and Pieter Verhaeghe. A privacy-preserving ticketing system. In Vijay Atluri, editor, *Data and Applications Security XXII*,

volume 5094 of *Lecture Notes in Computer Science*, pages 97–112. Springer Berlin / Heidelberg, 2008.

Meeting abstracts, presented at international conferences and symposia

[VLV+10] Jan Vossaert, Jorn Lapon, Pieter Verhaeghe, Bart De Decker, and Vincent Naessens. A smart card based solution for user-centric identity management. PrimeLife/IFIP Summer School 2010, Helsingborg, 2 - 6 August, 2010.

[VLN09] Jan Vossaert, Jorn Lapon, and Vincent Naessens. Towards a cross-domain identity card. 4th Benelux Workshop on Information and System Security, Louvain-La-Neuve, 19-20 Novembre 2009, November 2009.

[VLN+08] Pieter Verhaeghe, Jorn Lapon, Vincent Naessens, Bart De Decker, Kristof Verslype, and Girma Enideg Nigusse. Security and privacy threats of the Belgian electronic identity card and middleware, June 2008. Presented at EEMA European e-Identity conference, Den Haag, 10-11 June 2008.

Internal reports

[DDNLV08] Bart De Decker, Vincent Naessens, Jorn Lapon, and Pieter Verhaeghe. Kritische beoordeling van het gebruik van de Belgische eID kaart. CW Reports CW524, Department of Computer Science, K.U.Leuven, May 2008.

[VLV+08] Kristof Verslype, Jorn Lapon, Pieter Verhaeghe, Vincent Naessens, and Bart De Decker. PetAnon: A privacy-preserving e-petition system based on Idemix. CW Reports CW522, Department of Computer Science, K.U.Leuven, October 2008.

[VVL+08] Kristof Verslype, Pieter Verhaeghe, Jorn Lapon, Girma Enideg Nigusse, Vincent Naessens, and Bart De Decker. A privacy-preserving ticketing system. Leuven, Belgium CW523, KU Leuven, Department of Computer Science, October 2008.

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science
Distributed Systems and Computer Networks
Celestijnenlaan 200A box 2402
B-3001 Heverlee