# Oblivious Signatures

Lidong Chen

Aarhus University, Denmark

**Abstract.** Two special digital signature schemes, oblivious signatures, are proposed. In the first, the recipient can choose one and only one of $n$ keys to get a message signed without revealing to the signer with which key the message is signed. In the second, the recipient can choose one and only one of $n$ messages to be signed without revealing to the signer on which message the signature is made.

Key words: oblivious signatures

## 1 Introduction

A digital signature scheme is a protocol of a signer and a recipient (see [DH76]). In a public key system, the protocol has a secret key as a secret auxiliary input of the signer. By executing the protocol, the recipient gets a message $m$ signed. The signature $\sigma(m)$ can be verified with a corresponding public key.

In some of cryptology schemes, digital signatures are used as subroutines of the scheme. In order to protect the privacy of the recipient of a signature, in a certain stage, the information about with which key the recipient wants to get the message signed or which message the recipient wants to be signed should not be revealed. Blind signature (see [Ch82]) is a beautiful solution for this kind of problems. But sometimes it requires more restrictive for users' choice.

This note proposes a special kind of digital signature schemes: oblivious signatures. This name is from the fact that, theoretically, it can be implemented by an oblivious transfer (see [Ra81], [Cr87]). The signature schemes here are more efficient. We will consider two oblivious signature schemes.

The first scheme could be considered a complement of group signature (see [ChHe91]). The scheme is a multiparty protocol. The participants are a group of signers $S_1, S_2, ..., S_n$ and a recipient $R$. There are $n$ pairs of public and secret keys involved. Each signer has one of the secret keys as a secret auxiliary input. The scheme has the following characteristics.

- By executing the protocol, the recipient can get a message signed with one of $n$ keys which is chosen by himself and is called accepted key in this executing.
- The signers, even the holder of accepted key, can not find out with which key the signature is got by the recipient.
- If it is necessary, the recipient can show that he has got a signature with one of $n$ keys without revealing with which special one.

One example of application of the oblivious signature with $n$ keys is that in order to access a database, the user must pay certain amount of money to get a permit which is possibly a signature from the manager of the database. But the information about which database interests the user is sensitive. So he can choose $n$ databases which he is eligible to access. By executing oblivious signing protocol with the managers, he can get the permit for only one of $n$ databases without revealing which one.

The second scheme involves a signer $S$ and a recipient $R$. This oblivious signature scheme has $n$ messages as a part of common input. The scheme has the following characteristics.

- By executing the protocol, the recipient can choose only one of $n$ messages to get signed.
- The signer cannot find out on which message the recipient has got the signature.
- If it is necessary, the recipient can show that he has got a signature of one of $n$ messages without revealing which special one.

Such an oblivious signature can be used to protect the privacy of users. For example, the user will buy a software from the seller. The software can be used if and only if it is signed by the seller. But the information about which software interests the user may be sensitive in some stage. So the user can choose $n$ softwares and get one and only one signed by the seller without revealing which one.

Both oblivious signatures can be converted to designated confirmer signatures (see [Cha94]) such that

- only the recipient is able to convincingly show the signature afterwards.

## 2 Basic Protocol and Its Divertibility

### 2.1 Basic protocol

First we consider the basic three move protocol proposed in [ChaPe92]. Suppose $p$ is a prime, $q$ is the largest prime factor of $p-1$, and $g$ is a generator of $G_q$, the multiplying group of order $q$. The participants of the protocol are a prover $\mathcal{P}$ and a verifier $\mathcal{V}$.

The common input for $\mathcal{P}$ and $\mathcal{V}$ is

$$(g, h, m, z),$$

and the secret auxiliary input for $\mathcal{P}$ is

$$x = \log_g h.$$

We call $(g, h)$ the public key and $x$ the secret key of the protocol.

For given $h, m, z \in G_q$, the protocol is a proof of knowledge of $x = \log_g h$ and $\log_g h = \log_m z$.

The whole process is shown in Figure 1.

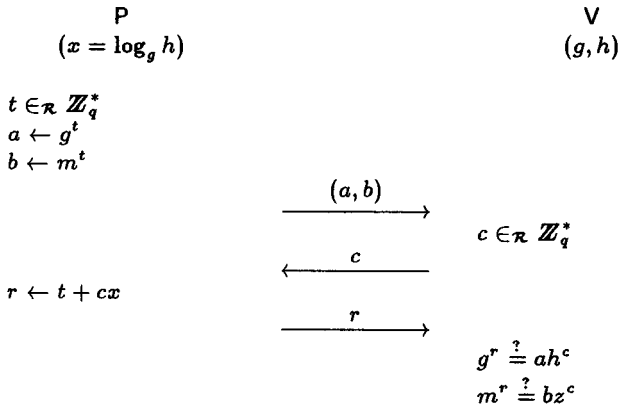


**Fig. 1.** $\mathcal{P}$ proves that $\log_g h = \log_m z$

### 2.2 The signature based on basic protocol

If the basic three move protocol is a proof of knowledge, then a class of signature schemes can be established. This kind of signature scheme is first proposed by Fiat and Shamir (see [FS87]). So it is called Fiat-Shamir style signature in the literature.

Let $\mathcal{H}$ be a hash function. The signature based on the basic protocol on message $m$ with secret key

$$x = \log_g h$$

is

$$\sigma_{(g,h)}(m) = (z, a, b, r).$$

It is correct if $c = \mathcal{H}(m, z, a, b)$ and

$$g^r = ah^c \qquad \text{and} \qquad m^r = bz^c.$$

*Remark.* Here we suppose that the message $m$ is in $G_q$. If it is not the case, a hash function will be used to map the message to $G_q$.

The signature is secure, if the basic protocol is witness hiding (see [FS90]) and the hash function $\mathcal{H}$ satisfies the following assumption.

**Assumption 1** *$\mathcal{H}$ has the property that if the basic protocol is a proof of knowledge, then it is as difficult to convince a verifier, who chooses $c = \mathcal{H}(m, z, a, b)$, as a verifier who chooses $c$ at random.*

### 2.3 Divertibility

The basic protocol has a very important property: the verifier, without the secret key as an auxiliary input, can divert the protocol to a third party when executing the protocol with the prover. This property is called divertibility (see [CheDaPe94]). The protocol is shown in Figure 2. For a history reason, we will call the middle one warden (see [Sim84]) denoted as $\mathcal{W}$.

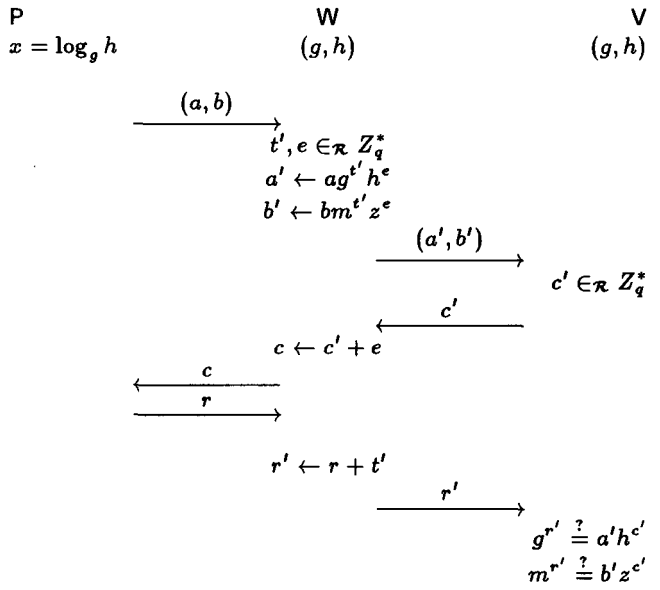


**Fig. 2.** Divertibility of the basic protocol

¿From Figure 2, it is easy to see that the warden $\mathcal{W}$ can play the role of prover to execute the protocol with the verifier $\mathcal{V}$. Furthermore, neither $\mathcal{P}$ nor $\mathcal{V}$ can perceive what the warden has done. This property will be used to construct the oblivious signatures in the following sections.

## 3 Oblivious Signature with $n$ Keys

### 3.1 Divertibility for different secret keys

In the previous section, we have seen a possibility to divert the basic protocol to a third party, in which both $\mathcal{P}$ and $\mathcal{W}$ prove the same secret key $x = \log_g h$. In this section, we will introduce another possibility to divert the basic protocol.

Suppose the input to $\mathcal{P}$ and $\mathcal{W}$ is

$$(g, h, m, z),$$

when $\mathcal{W}$ diverts it to $\mathcal{V}$, the common input to $\mathcal{W}$ and $\mathcal{V}$ is

$$(g, k, m, w),$$

where $k = h^y$ and $w = z^y$. $\mathcal{P}$ has secret input $x$ and $\mathcal{W}$ chooses $y$ by himself. In this protocol, $\mathcal{P}$ and $\mathcal{W}$ prove knowledge of different secret keys, $\log_g h$ and $\log_g k$ respectively. The divertibility is shown in Figure 3.

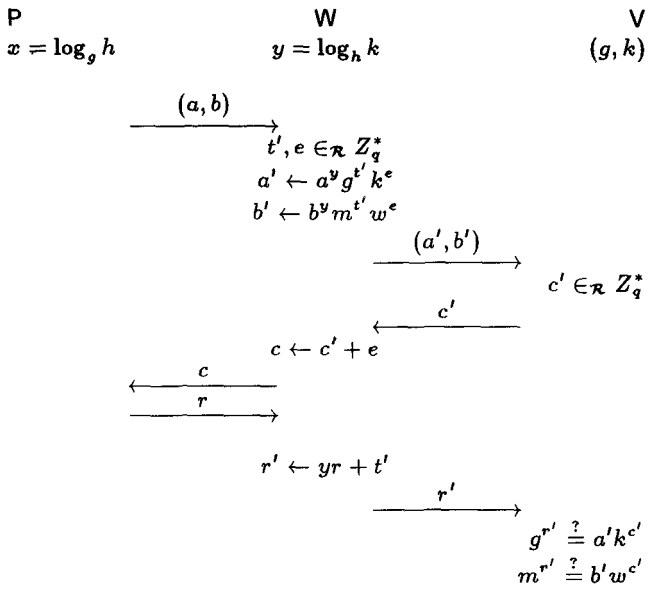


**Fig. 3.** Divertibility for different secret keys

If, instead of getting the random challenge $c'$ from $\mathcal{V}$, $\mathcal{W}$ computes $c'$ as a value of a hash function

$$c' = \mathcal{H}(m, w, a', b'),$$

then $\mathcal{W}$ gets a signature with the secret key $\log_g k$ as defined in Section 2.2.

### 3.2 Oblivious signing protocol (I)

The oblivious signature is a kind of random signature defined as follows.

**Definition 1.** (Random signature (I)) A random signature on a message $m$ with the public key $(g, h)$ and a random element $k \in Z_p^* - \{1\}$ is defined as

$$\Gamma_{(g,h)}(k, m) = \{\sigma_{(g,k)}(m), \sigma_{(h,k)}(m)\},$$

where $\sigma_{(g,h)}(m)$ and $\sigma_{(h,k)}(m)$ are defined in Section 2.2.

Here we will not specify the hash functions used in the signatures. We only suppose that the hash functions are given and with the property stated in Assumption 1.

It is clear that by executing the basic protocol with $\mathcal{P}$, $\mathcal{W}$ can get a random signature on $m$ with the public key $(g, h)$ and a random element $k$.

**Definition 2.** (Oblivious signature (I)) Suppose $\Omega$ is a group of signers (public-secret keys). An oblivious signature on message $m$ from $\Omega$ is a random signature $\Gamma_{(g,h_i)}(k, m)$ for some public key $(g, h_i)$ in $\Omega$.

Suppose that the public key for signer $S_i$ is $(g, h_i)$ and secret key is $x_i = \log_g h_i$, $i = 1, 2, ..., n$. In order to get the random signature on the message $m$ with one of the secret keys, say, $\log_g h_1$, $R$ chooses $y \in_R Z_q^*$ and computes $k = {h_1}^y$. The signing process goes as follows.

1. $R$ starts the protocol by broadcasting the message $m$.
2. Each $S_i$ computes $z_i = m^{x_i}$ and sends $z_i$ to $R$.
3. $S_i$ sends $(a_i, b_i)$ to $R$, $i = 1, 2, ..., n$.
4. $R$ chooses $t, e \in_R Z_q^*$ and computes $a = a_1^y g^t h_1^{ye}$ and $b = b_1^y m^t z_1^{ye}$.
5. $R$ broadcasts
$$c = \mathcal{H}(m, z, a, b) + e.$$
6. $S_i$ sends $r_i$ for the challenge $c$ to $R$, $i = 1, 2, ..., n$.
7. $R$ verifies $r_i$'s. If all of them are correct, he computes $r = yr_1 + t$, otherwise halts.

By executing the protocol above $R$ gets

$$\sigma_{(g,k)}(m) = (w, a, b, r),$$

where $w = z_1^y$. He can compute $\sigma_{(h_1,k)}(m)$ by himself. So he gets a random signature $\Gamma_{(g,h_1)}(k, m)$.

*Remark.* In fact, we can suppose either that $n$ different signers hold different keys or only one signer holds all $n$ keys.

### 3.3 Security of oblivious signature (I)

In this section, we suppose that the signature scheme defined in Section 2.2 is secure with the definition of [GMR88]. The security for the signer is partly based on a limitation of divertibility of the basic protocol. In [CheDaPe94], similar kind of limitation has been proved. However it is weaker than what we need here to prove the security of the oblivious signature (I).

First we must extend the divertibility stated in Section 2.3.

Suppose a warden $\mathcal{W}$ executes the basic protocol with $\mathcal{P}_i$, $i = 1, 2, \ldots, n$, with common input $(g, h_i, m_i, z_i)$ parallely. The rule for $\mathcal{W}$ is that he can only send to all the $\mathcal{P}_i$'s a same challenge $c$. At the same time, he can divert the protocol to a verifier with some common input, say, $(g, h, z, m)$. The limitation is that $\mathcal{W}$, with limited computational power, cannot divert it to two verifiers $\mathcal{V}_1$ and $\mathcal{V}_2$ parallely. We will state this limitation as a conjecture.

**Conjecture 1** *By executing the basic protocol in Section 2 parallely with $\mathcal{P}_i$, $i = 1, 2, \ldots, n$ with the restriction that only a same challenge $c$ can be sent to all $\mathcal{P}_i$'s, any warden $\mathcal{W}$ with limited computational power cannot divert the basic protocol to two independent verifiers $\mathcal{V}_1$ and $\mathcal{V}_2$ with the input $(g, h, m, z)$ and $(g, h', m', z')$ separately with nonnegligible probability unless he knows one of* $\log_g h$ *and* $\log_g h'$.

This conjecture can only be proved when the challenge set is a subset $E$ of $Z_q^*$ in the basic protocol such that

$$|E| < k^c,$$

for some $c > 0$, where $k$ is the length of input.

***Theorem 3.*** *By executing protocol in Section 3.2, the recipient, with limited computational power, cannot get more than one oblivious signature.*

*Proof.* In order to get two signatures from one execution of the protocol in Section 3.2, $R$ must work as the warden to divert the protocol to two independent verifiers if we suppose that the hash function is a random oracle as in Assumption 1, which is impossible by the conjecture. □

Sometimes, it is necessary to be sure that $R$ does get a random signature with one of the keys. This can be done by requiring $R$ to prove that he knows one of $\log_{h_i} k$, $i = 1, 2, ..., n$ for $\sigma_{(g,k)}(m)$ without revealing which one by the protocol proposed by Schoenmakers (see [Sch93]).

The next theorem is about the security for the recipient.

***Theorem 4.*** *From the transcripts of the protocol in Section 3.2, and from the signer's proof that he has got the message signed by one of the n keys, it cannot be recognized with which key the recipient got the signature even with unlimited computational power.*

*Proof.* Suppose that $\log_g h_i = x_i$ is held by $S_i$, $i = 1, 2, ..., n$. $R$ chooses $k$ as a random element in the protocol. For any $i$, $k = h_i^{y_i}$ and $\log_g k = x_i y_i$. If $a_i = g^{s_i}$, $b_i = m^{s_i}$, $i = 1, 2, ..., n$, and $a = g^s$, $b = m^s$, then denoting $s - s_i y_i = t_i$, $r = y_i r_i + t_i = c(x_i y_i) + s$, $i = 1, 2, ..., n$. Since the proof that the signer knows one of $\log_{h_i} k$, $i = 1, 2, \ldots, n$, is witness hiding, from $\sigma_{(g,k)}(m)$ and the transcripts of the protocol, no Shannon information about which key has been chosen by $R$ is revealed. □

*Remark.* By a small change of the protocol in Figure 3, the protocol can be diverted to a blind message $m' = m^{v_1} g^{v_2}$. In this case, both key and message are blinded. The oblivious signatures are untraceable even though they are shown afterwards.

# 4 Oblivious signature on $n$ messages

## 4.1 Divertibility for different messages

In order to describe the oblivious signature scheme on $n$ messages, we first introduce the divertibility of the basic protocol for different messages. In this case, the common input for $(\mathcal{P}, \mathcal{W})$ is

$$(g, h, m, z),$$

and for $(\mathcal{W}, \mathcal{V})$ is

$$(g, h, m', z'),$$

where

$$m' = m^y g^s$$

for some $y, s$, $y \neq 0$, which $\mathcal{W}$ knows. In this protocol, $\mathcal{P}$ and $\mathcal{W}$ prove the same secret key $\log_g h$ but for different $m$ and $m'$. The protocol is shown in Figure 4.

If for some $y, s$, $m' = m^y g^s$ is also a message, with $c' = \mathcal{H}(m', z', a', b')$, $\mathcal{W}$ can get a blind signature on message $m'$ by executing the basic protocol. However we cannot use a blind signature to construct an oblivious signature on $n$ messages since in this case, the recipient is not necessarily to get one of $n$ predetermined messages signed. Instead, he may construct some other message on which the signer is not going to offer the signature.

## 4.2 Oblivious signing protocol (II)

In order to restrict the recipient getting one of the given messages signed, we assume the signature scheme stated in Section 2.2 is on a hash value of the message $\mathcal{H}_1(m)$. The hash function $\mathcal{H}_1$ satisfies the following assumption.

**Assumption 2** *Assume that $\mathcal{H}_1$ is a hash function on message space $\mathcal{M}$. $\mathcal{H}_1$ has the property that for any polynomial time Turing machine $M$, by choosing the input $m \in \mathcal{M}$ randomly, $M$ cannot output $m' \in \mathcal{M}$, $m' \neq m$, and $y, s$ such that*

$$\mathcal{H}_1(m)^y g^s = \mathcal{H}_1(m')$$

*with nonnegligible probability.*

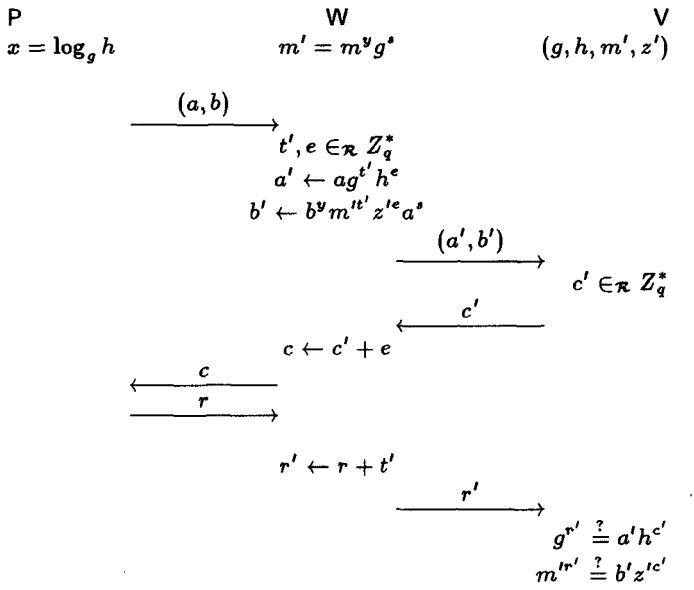


**Fig. 4.** Divertibility for different messages

For oblivious signature on $n$ messages, we use another kind of random signature which is defined as follows.

**Definition 5.** (Random signature (II)) A random signature on a message $m$ with the public key $(g, h)$ and a random element $m' \in Z_p^* - \{1\}$ is defined as

$$\Sigma_{(g,h)}(m', m) = (\sigma_{(g,h)}(m'), \sigma_{(\mathcal{H}_1(m), m')}(m)),$$

where $\sigma_{(g,h)}(m')$ and $\sigma_{(\mathcal{H}_1(m), m')}(m)$ are defined in Section 2.2.

**Definition 6.** (Oblivious signature (II)) Suppose $\mathcal{H}_1$ is a hash function. An oblivious signature on message $m_1, m_2, ..., m_n$ with public key $(g, h)$ and secret key $x = \log_g h$ is a random signature $\Sigma_{(g,h)}(m', m_i)$ on one of $m_i$'s.

The recipient can get it by executing the following protocol with the signer. Without loss of generality, we assume that $R$ would like to get the signature on $m_1$. He chooses $y$ at random and computes $m' = \mathcal{H}_1(m_1)^y$.

1. The recipient $R$ starts the protocol by sending $n$ messages $m_1, m_2, ..., m_n$ to the signer $S$.
2. $S$ computes $z_i = \mathcal{H}_1(m_i)^x$, and sends $z_i$, $i = 1, 2, ..., n$ to $R$.
3. $S$ chooses $t_i \in_R Z_q^*$, computes $a_i = g^{t_i}$, $b_i = \mathcal{H}_1(m_i)^{t_i}$, and sends $(a_i, b_i)$, $i = 1, 2, ..., n$ to $R$.

4. $R$ chooses $t, e$ in $Z_q^*$ randomly, computes $a = a_1 g^t h^e$, $b = b_1^y m'^t z_1^{ye}$, $c' = \mathcal{H}(m', z_1^y, a, b)$, and sends $c = c' + e$ to $S$.
5. $S$ computes $r_i = xc + t_i$, and sends $r_i$, $i = 1, 2, ..., n$ to $R$.
6. $R$ verifies $r_i$'s. If all of them are correct, he computes $r = r_1 + t$. Otherwise halts.

By executing the protocol, $R$ gets

$$\sigma_{(g,h)}(m') = (z_1^y, a, b, r).$$

So he gets an oblivious signature

$$\Sigma_{(g,h)}(m', m_1).$$

In order to prove that $R$ has got a signature of one of $n$ messages, he shows $\sigma_{(g,h)}(m')$ and proves that he knows one of $\log_{\mathcal{H}_1(m_i)} m'$, $i = 1, 2, ..., n$, by the witness hidding protocol in [Sch93].

### 4.3 Security of oblivious signature (II)

The security of the oblivious signature (II) partly depends on some kind of limitation about the common input between $\mathcal{W}$ and $\mathcal{V}$ of divertibility. The following conjecture has been used in the literature (see [Bran94a]).

**Conjecture 2** *For any polynomial time warden $\mathcal{W}$, if the basic protocol with input $(g, h, m, z)$ can be diverted to $\mathcal{V}$ by $\mathcal{W}$ for input $(g, h, m', z')$, then either $\mathcal{W}$ knows the secret key $x = \log_g h$, or $m' = m^y g^s$ for some $y, s$, $y \neq 0$, that $\mathcal{W}$ knows.*

There is no formal proof for this, even though it is believed to be true and no counterexample has been found. A proof of the conjecture appears to require an assumption which is seemingly stronger than the discrete logarithm assumption.

***Theorem 7.*** *By executing the protocol in Section 4.2, the recipient, with limited computational power, can get at most one of $m_1, m_2, ..., m_n$ signed.*

*Proof.* By Assumption 2 about the hash function $\mathcal{H}_1$ and Conjecture 2, it is impossible to get a signature on message $m \neq m_i$, $i = 1, 2, ..., n$ by executing the protocol in Section 4.2. By Assumption 1 and Conjecture 1, $R$ cannot get more than one signature in executing the protocol in Section 4.2 once. □

The privacy of the recipient is clear.

***Theorem 8.*** *From the transcript of protocol in Section 4.2, and from the proof that the recipient has got a signature on one of n messages, it cannot be recognized on which message the recipient has got the signature even with unlimited computational power.*

*Proof.* In executing the protocol, all the messages of $R$ are blinded by random factors. Even with unlimited computational power, it is impossible to find out on which message $R$ will get the signature. Also the proof that $R$ knows one of $\log_{\mathcal{H}_1(m_i)} m'$, $i = 1, 2, \ldots, n$, is witness hiding. So no Shannon information about on which message $R$ will get signature is revealed. □

## 5 Oblivious Signature with the Recipient as a Confirmer

The oblivious signature defined in Section 3 and Section 4 are digital signatures. It is not only recipient but also anyone who has got a copy of signature can convince the verifiers. If the signature is bought by a recipient, then sometimes he will not lost his privilege of convincing the correctness of the signature. Chaum proposed a kind of signature called *designated confirmer signatures* (see [Cha94]). After the recipient gets the signature, instead of the signer, some designated confirmer can convince the verifier that this is a correct signature with signer's key.

In this section, the oblivious signature will be constructed as the signature which can only be confirmed by the recipient.

In order to make the oblivious signatures with $n$ keys as designated confirmer signatures, the oblivious signature on message $m$ with public key $(g, h_i)$ will be

$$\sigma_{(g,k)}(m)$$

for random factor $k = h_i^y$ together with a proof of the knowledge

$$\log_{h_i} k.$$

For the oblivious signatures on $n$ messages, the oblivious signature is

$$\sigma_{(g,h)}(m')$$

where $m' = \mathcal{H}_1(m_i)^y$ together with a proof of the knowledge

$$\log_{\mathcal{H}_1(m_i)} m'.$$

After getting an oblivious signature, the recipient is the only one who can show its correctness.

## 6 Conclusion and Open Problems

A class of oblivious signature schemes can be established based on divertibility of a three move basic protocol. The security of oblivious signatures partly depends on some limitations of divertibility of the protocol. The proof of the limitations is an open problem.

The oblivious signature can be constructed based on almost all known Fiat-Shamir style signature which is based on three move proof of knowledge without special difficulties. Another open problem is how to construct the oblivious signature by RSA signature scheme.

## References


[Bran94a] S. Brands. Untraceable Off-line Cash in Wallet with Observers. In *Advances in Cryptology – Proceedings of CRYPTO 93.* Lecture Notes in Computer Science #773, Springer-Verlag, 1994, pp. 302–318.

[Ch82] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology - Proceedings of Crypto '82*, Plenum Press, 1983, pp. 199–203.

[Cha94] D. Chaum. Designated confirmer signatures. In *Advances in Cryptology - proceedings of EUROCRYPT'94*,

[ChHe91] D. Chaum, E. van Heyst. Group Signatures. In *Advances in Cryptology - proceedings of EUROCRYPT 91*, Lecture Notes in Computer Science, pages 257-265. Springer-Verlag, 1991.

[ChaPe92] D. Chaum and T. P. Pedersen. Wallet Databases with observers. In *Advances in Cryptology - proceedings of CRYPTO 92*, Lecture Notes in Computer Science, pages 89 – 105. Springer-Verlag, 1993.

[CheDaPe94] L. Chen, I. Damgaard and T. P. Pedersen. Parallel divertibility of proofs of knowledge. In *Advances in Cryptology - proceedings of EUROCRYPT 94*,

[Cr87] C. Crepeau Equivalence between two flavours of oblivious transfer. In *Advances in Cryptology - proceedings of CRYPTO 87*, Lecture Notes in Computer Science, pages 350 – 354. Springer-Verlag, 1988.

[DH76] W. Diffie and M. E. Hellman New Directions in Cryptography. In *IEEE Trans. Inform.*, IT-22(6):644–654, November, 1976.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - proceedings of EUROCRYPT 86*, Lecture Notes in Computer Science, pages 186 – 194. Springer-Verlag, 1987.

[FS90] U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pages 416 – 426, 1990.

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attack. *SIAM Journal on Computing*, 17(2):281 – 308, April 1988.

[Ra81] M. Rabin. How to exchange secrets by oblivious transfer Tech. Memo TR-81, Aiken Computation Laboratory, Harvard University,1981.

[Sch93] B. Schoenmakers. Efficient Proofs of Or. Manuscript, 1993.

[Sim84] G. J. Simmons. The Prisoner's Problem and the Subliminal Problems. In *Advances in Cryptology - proceedings of CRYPTO 83*, Plenum Press, pages 51–67. 1984.