A Study on the Security of Privacy Homomorphism*

Yu Yu, Jussipekka Leiwo, and Benjamin Premkumar
(Corresponding author: Yu Yu)

Nanyang Technological University, School of Computer Engineering
Block N4, Nanyang Avenue, Singapore 639798

(Selected paper from ITNG 2006)

Abstract

Informally, Privacy Homomorphism (PH) refers to encryption schemes with a homomorphic property allowing to obtain $E_k(a+b)$ or $E_k(a \times b)$ from ciphertexts $E_k(a)$ and $E_k(b)$ without the knowledge of the decryption key. Privacy homomorphisms, especially algebraic ones, have a wide range of applications in information security due to the homomorphic property. In this paper, we correct a misunderstanding regarding the security of additive PH, give a definition for efficient PH, and discuss the security of algebraic PH in the black-box model to show that any PH is at most semantically secure under non-adaptive chosen-ciphertext attacks (i.e. IND-CCA1 secure), which also implies that we can simulate an IND-CCA1 secure algebraic PH with a small amount of hardware.

Keywords: Chosen-ciphertext attacks, private computation, privacy homomorphism, semantic security

1 Introduction

Privacy homomorphism (hereafter referred to as PH) is first introduced by Rivest et al. [21] as a tool for processing encrypted data. Basically, an additive PH is a tuple (G, E, D, $\boxplus$) of polynomial-time algorithms satisfying the following conditions:

1) On input 1^n, probabilistic key generator G outputs a pair of strings (e, d), where n is the security parameter, e is the encryption key and d is the decryption key.

2) There exists a polynomially bounded function $\ell : \mathbb{N} \to \mathbb{N}$, called the block length, such that for every pair (e, d) produced by $G(1^n)$ and for every x, $y \in \{0,1\}^{\ell(n)}$, encryption algorithm E and decryption algorithm D satisfy

$$\Pr[D(d, E(e,x)) = x] = 1 \qquad (1)$$

$$\Pr[D(d, \boxplus(E(e,x), E(e,y))) = x+y] = 1, \qquad (2)$$

where "+" denotes addition over the plaintext space and the probability is taken over the internal coin tosses of E (note that D is deterministic).

The above definition does not distinguish between private-key block ciphers and public-key ones. In private-key schemes, e and d can be inferred from each other. Thus, for simplicity, we assume that e=d=k and that k is kept secret. In public-key schemes, e is publicly known and it is computationally infeasible to infer d from e. In the rest of this paper, we write $E_e(x)$ instead of $E(e, x)$ and $D_d(c)$ instead of $D(d, c)$. Equations (1) and (2) are written in terms of probability since we do not distinguish probabilistic ciphers from deterministic ones. Analogously, we can define a multiplicative PH, (G, E, D, $\boxtimes$), using almost the same definition as the additive PH except that Equation (2) is replaced by

$$\Pr[D_d(\boxtimes(E_e(x), E_e(y))) = x \times y] = 1, \qquad (3)$$

where "$\times$" denotes multiplication over the plaintext space. If a PH, (G, E, D, $\boxplus$, $\boxtimes$), is both additive and multiplicative (i.e., satisfies Equations (1), (2) and (3)), it is called an algebraic PH.

PH would be useful in a number of applications such as secret sharing schemes, software protection, multiparty computation and electronic voting (e.g. [3, 9, 10, 17, 22]). Rivest et al. [21] have presented four basic PHs, which are later shown to be vulnerable to either ciphertext-only attacks or known-plaintext attacks [7]. Goldwasser and Micali [16] present a probabilistic additive PH (the GM crypto-system) with the block length $\ell(n)$=1 (i.e. regardless of the security parameter n) and it is semantically secure if the Quadratic Residuosity Assumption holds. More efficient GM crypto-system variants are proposed in [4, 25]. There are also other semantically secure additive PHs (e.g. [19, 20]) whose "$\boxplus$" simply takes as input two ciphertexts and outputs the product of them. Domingo-Ferrer [12, 13] proposed two algebraic PHs targeted at combating known-plaintext attacks and the corresponding cryptanalysis is given in [2, 8, 24]. Sander,

*A preliminary version of this work appeared in proceedings of international conference on Information Technology: New Generations (ITNG 2006).

Young and Yung [23] proposed an unconditionally secure algebraic PH, which is inefficient as the length of ciphertexts is increased by a constant factor after each semigroup operation and thus it is used for computing only log-depth circuits. Boneh, Goh and Nissim [6] presented an algebraic PH based on the subgroup decision problem but their PH allows only one multiplication on ciphertexts.

Feigenbaum and Merritt [14] doubt the existence of secure algebraic PH. Boneh and Lipton [5] show that any deterministic algebraic PH can be broken in sub-exponential time under a reasonable assumption. We stress that this result is not surprising since it is well-known that stateless deterministic encryption schemes are not secure under the standard privacy notion. Therefore, in spite of those negative results, the best achievable security of PH remains an open question.

In this paper, we first correct a misunderstanding regarding the security of PH, then give a definition for "efficient" PH, which rules out most existing algebraic PHs. We proceed to exploring the best security of efficient algebraic PH in the black-box model where only oracle access is allowed to the homomorphic functions $\boxplus$ and $\boxtimes$. As shown in Figure 1, this can be achieved by implementing $\boxplus$ and $\boxtimes$ using a portable weak power device (e.g. a smart card) and thus it offers a hardware-based solution to construct secure PH. We show that the best achievable security of the resulting PH is semantic security under non-adaptive chosen-ciphertext attacks (IND-CCA1 security). Consequently, any PH (either hardware-based or algorithm-based) is at most IND-CCA1 secure.

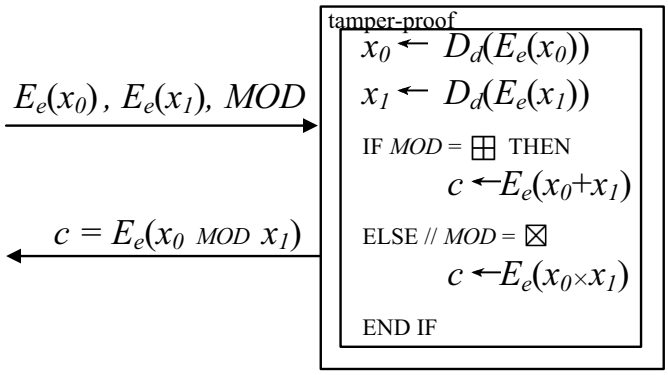


Figure 1: The oracle access to $\boxplus$ (resp., $\boxtimes$) can be emulated by a tamper-proof device that takes $E_e(x_0)$ and $E_e(x_1)$ as argument, decrypts them and outputs the encrypted sum (resp., product) of x_0 and x_1.

2 A Misunderstanding Regarding the Security of PH

Ahituv, Lapid and Neumann [1] showed a chosen-plaintext attack at additive PH and this result is sometimes mistakenly cited as that no additive PH can be secure against chosen-plaintext attacks. We point out that the attack does not work for all additive PHs.

The attack can be summarized as follows: Let E_k: $\{0,1\}^n \rightarrow \{0,1\}^m$ be an additively (or algebraically) homomorphic block cipher. For each a and $b \in \{0,1\}^n$, it holds that

$$D_k(\boxplus(E_k(a), E_k(b))) = a + b$$

where "$\boxplus$" is an addition function and we assume that $[a_n,\cdots,a_1]$ (resp., $[b_n,\cdots,b_1]$) is the binary representation of a (resp., b). An attacker chooses the following m plaintext-ciphertext pairs:

$$\begin{aligned} &E_k([c_{11}, c_{12}, ..., c_{1n}]) = [1, 0, \cdots, 0] \\ &E_k([c_{21}, c_{22}, ..., c_{2n}]) = [0, 1, \cdots, 0] \\ &\vdots \\ &E_k([c_{m1}, c_{m2}, ..., c_{mn}]) = [0, 0, \cdots, 1]. \end{aligned}$$

Then given any $E_k([x_1,\cdots,x_n])=[a_1,\cdots,a_m]$, he may obtain $[x_1,\cdots,x_n]$ by computing:

$$[x_1, \cdots, x_n] = \sum_{i=1}^{m} a_i E_k([c_{i1}, c_{i2}, \cdots, c_{in}]).$$

Nevertheless, the above attack is successful only when the "$\boxplus$" computes addition or similar linear functions. By the definition of PH, "$\boxplus$" is a polynomial-time algorithm that takes E_k (a) and E_k (b) as argument and outputs E_k $(a+b)$. Thus, if "$\boxplus$" is a non-linear function (e.g. multiplication), then the attack would not be successful.

3 Efficient Privacy Homomorphism

In this section, we show a trivial and inefficient all-operation-support privacy homomorphism and provide a definition for efficient privacy homomorphism.

The notion of PH was introduced to process encrypted data [21], namely, to allow an untrustworthy party to perform operations on ciphertexts without revealing him anything substantial. However, it turns out that most existing algebraic PHs (e.g. [12, 13, 23]) are not length-preserving, namely, there is at least one homomorphic function that inputs two ciphertexts and outputs a new one whose length is the sum of their lengths. In other words, the length of resulting ciphertext goes exponentially with regard to the depth of such operations. Furthermore, decryption takes time at least proportional to the length of ciphertext. In this case, we take no advantage of the untrustworthy party as we would spend more time on decrypting the final ciphertext than on performing the computation in plaintext all by ourselves. Carried to the extreme, the inefficiency can be illustrated with a trivial PH.

Suppose that we have a semantically secure encryption scheme $(G,\ E,\ D)$, then for any operation "$\star$"

on plaintexts, we write the homomorphic operation on ciphertexts as "★", which takes c_1 and c_2 as inputs and outputs ciphertext denoted by $c_1 \bigstar c_2$, and define $D_d(c_1 \bigstar c_2) = D_d(c_1) \star D_d(c_2)$. It is not hard to prove that the resulting PH $(G, E, D, \bigstar)$, but it is trivial, for example, if we want an untrustworthy Bob to perform a computation on plaintexts $x_1, \cdots, x_n$, then we need to encrypt those plaintexts and send their ciphertexts to Bob, and after Bob returns the final ciphertext of the result, we still need to recover $x_1, \cdots, x_n$ and perform the computation on them, which takes more time than we do the computation in the first place without Bob. As the inefficiency is caused by non-length-preserving homomorphic function(s), we define efficient PH as follows:

Definition 1. *(Efficient PH): A PH is efficient if all its homomorphic functions are length-preserving, namely, for any homomorphic function ★, and for any ciphertext-pair (c_1,c_2), the length of the resulting ciphertext $\bigstar(c_1,c_2)$ is not larger than the maximal length of c_1 and c_2.*

It is relatively easy to construct an efficient PH that supports only one homomorphic function, and most existing additive or multiplicative (but not both) PHs are efficient in the sense of Definition 1. In contrast, most algebraic PHs are not efficient with an exception being the one introduced in [6], which unfortunately supports only one multiplication on ciphertexts and hence is not truly multiplicative. Therefore, in the rest of the paper, we consider only the security of efficient algebraic PH.

4 IND-CCA1 Secure Block Ciphers

In this section, we introduce the notion of IND-CCA1 security and point out that block ciphers of this type exist under reasonable assumptions.

Informally, a block cipher is IND-CCA1 secure if and only if it has indistinguishable encryptions[1] (i.e. IND) and it is secure under nonadaptive chosen-ciphertext attacks (i.e. CCA). IND-CCA1 security can be described using a "guess" game between an adversary Malice and an oracle $\mathcal{O}$ [18, Protocol 14.1]:

Game 1.

1) *$\mathcal{O}$ and Malice agree on a target (G,E,D) and $\mathcal{O}$ chooses a pair of keys by $(e,d) \leftarrow G(1^n)$, where the encryption key e is revealed to Malice if (G,E,D) is a public-key encryption scheme.*

2) *$\mathcal{O}$ allows Malice to have oracle access to E_e and D_d. After several rounds, they proceed to the next step.*

3) *Based on the information achieved, Malice selects two distinct messages x_0 and x_1 of the same length and sends them to $\mathcal{O}$.*

4) *$\mathcal{O}$ tosses a fair coin $b \in_U \{0,1\}$ and selects a value c^* from the distribution of $E_e(x_b)$.*

5) *After receiving c^*, Malice can only have oracle access to E_e. Then, Malice guesses b by answering either 0 or 1.*

If Malice has no strategy to win Game 1 better than random guessing, then (G,E,D) is IND-CCA1 secure for any single message. We stress that the single-message IND-CCA1 security is equivalent to the multiple-message one in which multiple messages are encrypted [15, Section 5.4.4.1]. The rigorous definition for IND-CCA1 is given by Goldreich as follows [15, Definition 5.4.14]:

Definition 2. *(indistinguishability of encryptions under non-adaptive chosen-ciphertext attacks): For public-key schemes: A public-key block cipher, (G,E,D), is said to be **IND-CCA1 secure** if for every pair of probabilistic polynomial oracle machines, A_1 and A_2, for every positive polynomial p, and all sufficiently large n and $z \in \{0,1\}^{poly(n)}$ it holds that*

$$|p^{(0)}_{n,z} - p^{(1)}_{n,z}| < \frac{1}{p(n)},$$

where

$$p^{(i)}_{n,z} \stackrel{def}{=} \Pr\left[\begin{array}{c} v = 0, where \\ ((x_0,x_1),\sigma) \leftarrow A_1^{E_e,D_d}(e,z) \\ c^* \leftarrow E_e(x_i) \\ v \leftarrow A_2^{E_e}(\sigma, c^*). \end{array}\right]$$

$(e,d) \leftarrow G(1^n)$, $|x_0|=|x_1|=n$ and the probability is taken over the internal coin tosses of G, E_e, A_1 and A_2.

For private-key schemes: The definition is identical except that A_1 gets the security parameter 1^n instead of the encryption key e.

In the above definition, adversary Malice is decoupled into a pair of oracle machines $(A_1^{E_e,D_d}, A_2^{E_e})$. That is, A_1 has oracle access to both E_e and D_d (see the step 2 of Game 1) while A_2 is restricted to E_e (step 5). In the joint work of A_1 and A_2, σ denotes the state information A_1 passes to A_2. Since we use non-uniform formulations z is a non-uniform auxiliary input of A_1 (A_2's counterpart is included in σ). Finally, A_2 outputs v as its guess.

We can construct block ciphers that is IND-CCA1 secure under reasonable assumptions, e.g., the private-key IND-CCA1 secure block cipher in [15, Construction 5.4.19] and the public-key one in [15, Construction 5.4.23]. In both cases, the encryption function E_e is probabilistic (or stateful deterministic), namely, if we invoke E_e on the same input x polynomially (in n) many times, its outputs will be different from each other with an overwhelming probability. Otherwise, Malice can win Game 1 with ease. In practice, $\boxplus$ (resp., $\boxtimes$) can be emulated by tamper-proof hardware that takes as input two ciphertexts $E_e(m_1)$ and $E_e(m_2)$ and outputs $E_e(m_1+m_2)$ (resp., $E_e(m_1 \times m_2)$) by invoking E_e and D_d. Such hardware can be smart cards as some smart cards can do both private-key encryptions and public-key encryptions with the help of cryptographic co-processors.

[1] In most cases, indistinguishability of encryptions is equivalent to semantic security. Thus, we use IND as the shorthand for semantic security.

5 IND-CCA1 Secure PH

In this section, we assume an IND-CCA1 secure block cipher (G, E, D) having oracle access to $\boxplus$ and $\boxtimes$ and prove that the resulting PH, (G, E, D, $\boxplus$, $\boxtimes$), preserves IND-CCA1 security. Analogously, the IND-CCA1 security for PH (G, E, D, $\boxplus$, $\boxtimes$) can be modelled with Game 2:

Game 2.

1) *$\mathcal{O}$ and Malice agree on a target (G,E,D,$\boxplus$,$\boxtimes$) and $\mathcal{O}$ chooses a pair of keys by $(e,d)\leftarrow G(1^n)$, where the encryption key e is revealed to Malice if (G,E,D) is a public-key encryption scheme.*

2) *$\mathcal{O}$ allows Malice to have oracle access to E_e, D_d, $\boxplus$ and $\boxtimes$. After several rounds, they proceed to the next step.*

3) *Based on the information achieved, Malice selects two distinct messages x_0 and x_1 of the same length and sends them to $\mathcal{O}$.*

4) *$\mathcal{O}$ tosses a fair coin $b \in_U \{0,1\}$ and selects a value c^* from the distribution of $E_e(x_b)$.*

5) *After receiving c^*, Malice is only allowed to have oracle access to E_e, $\boxplus$ and $\boxtimes$. After that, Malice guesses b by answering either 0 or 1.*

Thus, the formal definition of IND-CCA1 security for (G, E, D, $\boxplus$, $\boxtimes$) is almost same as Definition 2 except that the pair of oracle machines is $(A_1^{E_e,D_d,\boxplus,\boxtimes},A_2^{E_e,\boxplus,\boxtimes})$ instead of $(A_1^{E_e,D_d},A_2^{E_e})$.

We proceed to proving that an IND-CCA1 secure (G, E, D) preserves the same security when we allow adversaries to have additional oracle access to $\boxplus$ and $\boxtimes$. Before presenting the formal proof, we sketch it informally in Figure 2. By definition, (G, E, D) is IND-CCA1 secure iff no polynomial-time adversary can win Game 1 better than random guessing. Note that the only difference between Game 1 and Game 2 is that in the latter case Malice can have oracle access to $\boxplus$ and $\boxtimes$ (see step 2 and step 5). We will show that the oracle access to (E_e, D_d, $\boxplus$, $\boxtimes$) in step 2 is equivalent to accessing only (E_e, D_d). In addition, the oracle access to (E_e, $\boxplus$, $\boxtimes$) can be made computationally indistinguishable to accessing only E_e by emulating $\boxplus$ and $\boxtimes$ with E_e. Recall that the indistinguishable property (i.e. IND) implies that any polynomial-time adversary is unable to distinguish between any pair of distinct ciphertexts, e.g., $E_e(t_n)$ and $E_e(0)$.

Lemma 1. *(Indistinguishablity between $E_e(t_n)$ and $E_e(0)$): For public-key schemes: Let (G, E, D) be an IND-CCA1 secure public-key (resp., private-key) block cipher, then for every plaintext $t_n \in \{0,1\}^n$, for every pair of probabilistic polynomial oracle machines A_1 and A_2, every positive polynomial p, all sufficiently large n's and $z\in\{0,1\}^{poly(n)}$, it holds that*

$$|p_{n,z}^{(0)} - p_{n,z}^{(1)}| < \frac{1}{p(n)}$$

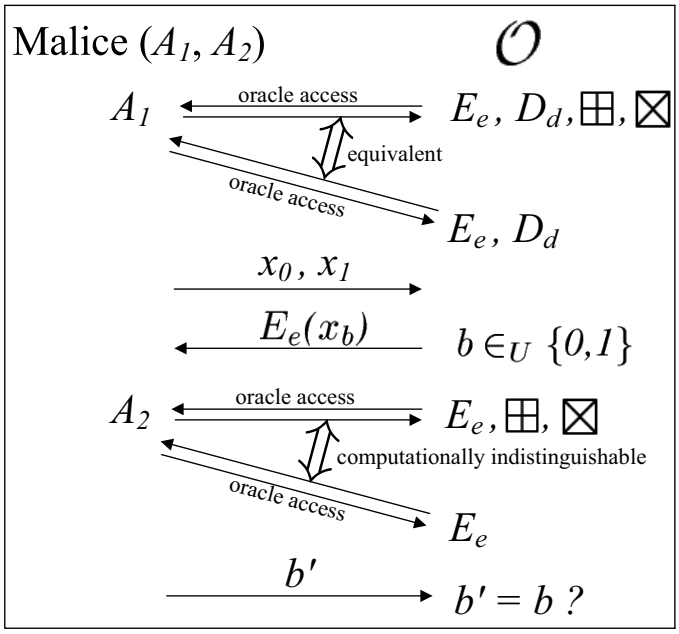


Figure 2: The proof sketch

where

$$p_{n,z}^{(i)} \stackrel{def}{=} \Pr\left[\begin{array}{c} v=0, where \\ \sigma \leftarrow A_1^{E_e,D_d}(e,z) \\ c^* \leftarrow \left\{\begin{array}{l} E_e(0),\ i=0 \\ E_e(t_n),\ i=1 \end{array}\right. \\ v \leftarrow A_2^{E_e}(\sigma,c^*) \end{array}\right]$$

$(e,d)\leftarrow G(1^n)$ and the probability is taken over the internal coin tosses of G, E_e, A_1 and A_2.

For private-key schemes: The indistinguishability also holds except that A_1 gets the security parameter 1^n instead of the encryption key e.

Proof sketch. For the sake of contradiction, we assume that $E_e(t_n)$ and $E_e(0)$ are distinguishable by $A_2^{E_e}$ within polynomial time. It follows that Definition 2 does not hold in case that x_0=0 and x_1=t_n, which contradicts the fact that (G, E, D) is IND-CCA1 secure. □

Theorem 1. *(IND-CCA1 secure PH): Let (G, E, D) be an IND-CCA1 secure block cipher, let $\boxplus$ and $\boxtimes$ be as in (2) and (3) respectively and assume that only oracle access is allowed to $\boxplus$ and $\boxtimes$, then the PH (G, E, D, $\boxplus$, $\boxtimes$) is also IND-CCA1 secure.*

Proof. As discussed, we only need to prove that Definition 2 still holds if we replace $(A_1^{E_e,D_d},A_2^{E_e})$ by the corresponding $(A_1^{E_e,D_d,\boxplus,\boxtimes},A_2^{E_e,\boxplus,\boxtimes})$. The equivalence of $A_1^{E_e,D_d}$ and $A_1^{E_e,D_d,\boxplus,\boxtimes}$ is quite straightforward since either $\boxplus$ or $\boxtimes$ is implied by E_e and D_d. In other words, $\mathcal{O}$ can efficiently emulate $\boxplus$ (resp., $\boxtimes$) by decrypting the input ciphertexts, computing the sum (resp., product) of the corresponding plaintexts and encrypting the result. The remaining difficulty is the reduction from $A_2^{E_e,\boxplus,\boxtimes}$ to $A_2^{E_e}$. Obviously, the input-output behaviors of $\boxplus$ and $\boxtimes$

cannot be emulated using only E_e, but $\mathcal{O}$ can implement a function $\mathcal{O}^{+,\times}$ using E_e such that their input-output behaviors are computationally indistinguishable to those of $\boxplus$ and $\boxtimes$. For example, we can let $\mathcal{O}^{+,\times}$ be as follow:

```
O^{+,×} (E_e(x_0), E_e(x_1)){
    return E_e(0);
}
```

By the definition of $\boxplus$ and $\boxtimes$, it holds that

```
⊞ (E_e(x_0), E_e(x_1)){
    return E_e(x_0+x_1);
}

⊠ (E_e(x_0), E_e(x_1)){
    return E_e(x_0×x_1);
}
```

By Lemma 1, $A_2^{E_e}$ cannot distinguish between $E_e(0)$ and $E_e(x_0+x_1)$ (or $E_e(x_0\times x_1)$), namely, whatever can be efficiently computed from the oracle access to $\boxplus$ and $\boxtimes$ can also efficiently computed from scratch. Thus, replacing $(\boxplus,\boxtimes)$ by $\mathcal{O}^{+,\times}$ has no effect on the decision of A_2 and as a result, $A_2^{E_e,\boxplus,\boxtimes}$ and $A_2^{E_e,\mathcal{O}^{+,\times}}$ compute almost the same function (with a negligible difference). Therefore, it suffices that $A_2^{E_e,\mathcal{O}^{+,\times}}$ can be reduced to $A_2^{E_e}$ in a computational sense and the conclusion immediately follows. □

In the above proof, we make use of the simulation paradigm, namely, if the view of an adversary can be efficiently simulated by a PPT from scratch (or what the adversary already knows), then the adversary gain nothing substantial from his view. The knowledge of $\boxplus$ and $\boxtimes$ only allows him to produce semantically correlated ciphertext triplets $(E_e(x_0), E_e(x_1), E_e(x_0+x_1))$ and $(E_e(x_0), E_e(x_1), E_e(x_0\times x_1))$, but does not help to distinguish between $E_e(x_0)$ and $E_e(x_1)$ for any distinct x_0 and x_1. In addition, we note that E_e is probabilistic and hence Malice cannot even distinguish between the two instances of $E_e(x)$, where x is an arbitrary plaintext. Thus, Malice cannot tell whether two ciphertexts have the same semantics.

6 Beyond IND-CCA1 Security

We have shown that it is possible to obtain IND-CCA1 secure PH given only oracle access to $\boxplus$ and $\boxtimes$. In this section, we show that IND-CCA1 is the best achievable security for any PH, namely, any security beyond IND-CCA1 is not attainable.

6.1 Non-Malleable Security

Informally, IND security against some-type attacks requires that the ciphertexts reveal nothing to passive adversaries conducting the some-type attacks, where "some-type" can be ciphertext-only, chosen-plaintext, chosen-ciphertext, etc. This notion is enough in most cases where the adversary only hopes to gain the information from ciphertexts, but it does not prevent Malice from replacing the ciphertext with a semantically related one. Let us consider the well-known private key encryption of the one-time pad (Vernam cipher). Given an n-bit message $m=m_1\cdots m_n$, the key generator outputs an n-bit key $k=k_1\cdots k_n$ uniformly chosen from $\{0,1\}^n$. The encryption is done by bitwise XORing m_i with k_i, that is, for $1\le i\le n$, $c_i=m_i\oplus k_i$. It is well-known that this cipher is unconditionally secure in that ciphertexts disclose nothing (except the length n) in an information-theoretic sense. However, the cipher is malleable. Malice is able to flip each c_i by replacing c_i with $c_i'=c_i\oplus 1$ such that the one that decrypts c_i' will get the complement of m_i. In this way, Malice can flip the semantics of each ciphertext.

$(G, E, D, \boxplus, \boxtimes)$ is also malleable. Given any plaintext-ciphertext pair $(a, E_e(a))$ with $a \neq 0$ (resp., $a \neq 1$), Malice is able to increment (resp., multiply) by a the semantics of any ciphertext c by replacing c with c', where $c'\leftarrow\boxplus(c,E_e(a))$ (resp., $c'\leftarrow\boxtimes(c,E_e(a))$). Therefore, non-malleable (NM) security is not achievable for any PH. Although there are techniques (e.g. message authentication code) that prevents active adversaries from faking any valid ciphertext, these techniques contradict the definition of PH by which any party is allowed to create valid ciphertext using existing ciphertexts and $\boxplus$ (or $\boxtimes$).

6.2 Security under Adaptive Chosen-ciphertext Attacks

So far, we know that $(G, E, D, \boxplus, \boxtimes)$ cannot have NM security. That is, it is at most IND secure against some-type attacks and this "some-type" can be CCA. However, can this "some-type" be something more advanced? In other words, can $(G, E, D, \boxplus, \boxtimes)$ be IND secure under adaptive chosen-ciphertext attacks (CCA2)? To answer this question [2], we first model the notion of IND-CCA2 security for PH with the following game:

Game 3.

1) *$\mathcal{O}$ and Malice agree on a target $(G,E,D,\boxplus,\boxtimes)$ and $\mathcal{O}$ chooses a pair of keys by $(e,d)\leftarrow G(1^n)$, where the encryption key e is revealed to Malice if (G,E,D) is a public-key encryption scheme.*

2) *$\mathcal{O}$ allows Malice to have oracle access to E_e, D_d, $\boxplus$ and $\boxtimes$. After several rounds, they proceed to the next step.*

3) *Based on the information achieved, Malice selects two distinct messages x_0 and x_1 of the same length and sends them to $\mathcal{O}$.*

4) *$\mathcal{O}$ tosses a fair coin $b \in_U \{0,1\}$ and selects a value c^* from the distribution of $E_e(x_b)$.*

[2] Alternatively, we can answer the question using the results by Dolev et al. [11], who showed that under CCA2 semantic security and non-malleable security are equivalent. Since we have already shown that PH is malleable under CCA2, it cannot be IND-CCA2 secure.

5) Upon receiving c^*, Malice is only allowed to have oracle access to E_e, D_d, $\boxplus$ and $\boxtimes$ with the exception that the decryption query of c^* will be denied. Then, Malice guesses b by answering either 0 or 1.

Now we show that Malice can always win Game 3 efficiently. For any c^*, Malice decrypts it using $\boxplus$ (or $\boxtimes$) to obtain b with the following steps:

1) Malice queries E_e with 0 (resp., 1) to receive $E_e(0)$ (resp., $E_e(1)$).

2) Malice queries $\boxplus$ (resp., $\boxtimes$) with c^* and $E_e(0)$ (resp., $E_e(1)$) to obtain c^{**}.

3) Malice queries D_d with c^{**} to obtain x_b.

4) Malice determines b by looking up x_b in (x_0,x_1).

To conclude, no matter how secure (G, E, D) is, the homomorphic property of $\boxplus$ or $\boxtimes$ will make the PH vulnerable under CCA2. Thus, the best achievable security of PH is no more than IND-CCA1 security.

7 Concluding Remarks

After correcting a misunderstanding regarding the security of additive PH and defining efficient PH, we show that any algebraic (or additive or multiplicative) PH can be at most IND-CCA1 secure in the black-box model, but it still remains an open question whether IND-CCA1 security is a tight upper bound for algorithm-based PH. We believe it is extremely hard to find such efficient PHs as we have put some restrictions: (1) homomorphic functions are length-preserving. (2) the additively homomorphic function cannot be addition or any other linear function. Nevertheless, we can construct IND-CCA1 secure hardware-based PH using low-cost portable devices (e.g. smart cards with cryptographic co-processors) and they might be useful in applications where no algorithm-based solution is available and interactions are not so intensive. For example, in software protection, most algorithm-based solutions are heuristic and have no complexity-theoretic foundations. Thus, with hardware-based PH, we can use a small amount of hardware to achieve provable security.

References

[1] N. Ahituv, Y. Lapid, and S. Neumann, "Processing encrypted data," *Communications of the ACM (CACM)*, vol. 30, no. 9, pp. 777-780, 1987.

[2] F. Bao, "Cryptanalysis of a provable secure additive and multiplicative privacy homomorphism," in *Proceedings of the International Workshop on Coding and Cryptography (WCC 2003)*, pp. 43-50, 2003.

[3] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret sharing," in *Advances in Cryptology - CRYPTO '1986*, pp. 251-260, Springer-Verlag, 1986.

[4] M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information," in *Advances in Cryptology - CRYPTO '1984*, pp. 289-302, Springer-Verlag, 1984.

[5] D. Boneh and R. J. Lipton, "Algorithms for black-box fields and their application to cryptography (Extended abstract)," in *Advances in Cryptology - CRYPTO 1996*, pp. 283-297, Springer-Verlag, 1996.

[6] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Proceedings of the 2nd Theory of Cryptography Conference (TCC 2005)*, pp. 325-341, 2005.

[7] E. F. Brickell and Y. Yacobi, "On privacy homomorphisms (Extended abstract)," in *Advances in Cryptology - EUROCRYPT '1987*, pp. 117-125, Springer-Verlag, 1987.

[8] J. H. Cheon and H. S. Nam, *A Cryptanalysis of the Original Domingo-Ferrer's Algebraic Privacy Homomophism*, Cryptology ePrint Archive, Report 2003/221, 2003. (http://eprint.iacr.org/)

[9] R. Cramer, I. Damgård, and J. B. Nielsen, "Multiparty computation from threshold homomorphic encryption," in *Advances in Cryptology - EUROCRYPT '2001*, pp. 280-299, Springer-Verlag, 2001.

[10] R. Cramer, R. Gennaro, and B. Schoenmakers, "A Secure and optimally efficient multi-authority election scheme," in *Advances in Cryptology - EUROCRYPT '1997*, pp. 103-118, Springer-Verlag, 1997.

[11] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pp. 542-552, 1991.

[12] J. D. Ferrer, "A new privacy homomorphism and applications," *Information Processing Letters*, vol. 60, no. 5, pp. 277-282, 1996.

[13] J. D. Ferrer, "A provably secure additive and multiplicative privacy homomorphism," in *Proceedings of the 5th International Conference on Information Security (ISC 2002)*, pp. 471-483, Springer-Verlag, 2002.

[14] J. Feigenbaum and M. Merritt, "Open questions, talk abstracts, and summary of discussions," in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 2, pp. 1-45, 1991.

[15] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2, Cambridge University Press, 2004.

[16] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.

[17] A. Kiayias and M. Yung, "The vector-ballot e-voting approach," in *Financial Cryptography (FC 2004)*, pp. 72-89, Springer-Verlag, 2004.

[18] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2004.

[19] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in Cryptology - EUROCRYPT '1998*, pp. 308-318, Springer-Verlag, 1998.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT 1999*, pp. 223-238, Springer-Verlag, 1999.

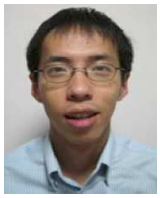
[21] R. L. Rivest, L. M. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, pp. 169-180, 1978.

[22] T. Sander and C. F. Tschudin, "Towards mobile cryptography," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 215-224, IEEE CS Press, 1998.

[23] T. Sander, A. Young, and M. Yung, "Non-interactive cryptocomputing for NC1," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS 1999).*, pp. 554-567, 1999.

[24] D. Wagner, "Cryptanalysis of an algebraic privacy homomorphism," in *Proceedings of the 5th International Conference on Information Security (ISC 2003)*, pp. 234-239, Springer-Verlag, 2003.

[25] A. C. Yao, "Theory and applications of trapdoor functions," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pp. 80-91. IEEE CS Press, 1982.



Yu Yu received his B.S. degree in Computer Science from Fudan University (China) in 2003. He is currently a PhD candidate in the school of Computer Engineering, Nanyang Technological University, Singapore. His research interests include private computation, obfuscation and cryptanalysis.

Jussipekka Leiwo was born in Finland. He obtained an M.Sc. from the University of Oulu (Finland) in 1995 and a Ph.D. from Monash University (Melbourne, Australia) in 2000. He is currently an Assistant Professor in the school of Computer Engineering, Nanyang Technological University (Singapore). His major research interests include security engineering, in particular methodologies, tools and techniques for engineering high assurance IT security products and systems.

Benjamin Premkumar received his Bachelor of Science degree in Physics and Math from Bangalore University (India) and a Bachelor's degree in Electrical communication Engineering from the Indian Institute of Science (India). He briefly worked in large communication industry in Bangalore in their Research and Development division before proceeding to the US to earn his M.S. from North Dakota State University. His MS research was in the area of digital speech processing. He taught as a graduate teaching fellow at NDSU. He then went on to obtain his PhD from University of Idaho. His PhD thesis was in the area of Synthetic Aperture Radar Signal Processing, a project funded by NASA. He has held various teaching positions since 1991 both in the US and Singapore. Currently he is an Associate Professor in the school of Computer Engineering (NTU). His research interests include digital signal processing and its applications in wireless communication, software defined radio and impulse radio. He also works in the area of multirate signal processing and number theory and its applications to signal analysis.