Privacy Preserving Mapping Schemes Supporting Comparison

Qiang Tang
DIES, Faculty of EEMCS, University of Twente
Enschede, the Netherlands
q.tang@utwente.nl

ABSTRACT

To cater to the privacy requirements in cloud computing, we introduce a new primitive, namely Privacy Preserving Mapping (PPM) schemes supporting comparison. An PPM scheme enables a user to map data items into images in such a way that, with a set of images, any entity can determine the $<, =, >$ relationships among the corresponding data items. We propose three privacy notions, namely ideal privacy, level-1 privacy, and level-2 privacy, and three constructions satisfying these privacy notions respectively.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Algorithms, Security

Keywords

Cloud computing, secure 2-party computation, privacy

1. INTRODUCTION

1.1 Motivation

With the advances in networking technology, cloud computing has become one of the most exciting IT technologies. Briefly, cloud computing refers to anything that involves delivering hosted services over the Internet, and such services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Today, many well-known IT companies such as Google, Microsoft, Amazon, and Salesforce, have already provided cloud computing services.

When an organization adopts a cloud-oriented business model, typically its data storage and processing will be transferred from within its own organizational perimeter to that of the cloud service provider. As a result of the transit, the organization enjoys many nice features of cloud computing, such as agility, reliability, scalability, cost effectiveness, easy maintenance, and so forth. However, the downside is that, as many security critics have already pointed out, there are potential privacy risks for the outsourced data. For example, Ristenpart *et al.* show that non-provider-affiliated malicious attackers can mount side channel attacks against honest users in Amazonąŕs Elastic Compute Cloud (EC2) service [7]. Clearly, a curious/malicious service provider, such as Amazon, can do a lot more than such non-provider-affiliated malicious attackers.

How to tackle the privacy concerns in cloud computing is a complex issue, and it requires efforts from a number of aspects, such as law enforcement, regulation compliance, network security, cryptography, and so forth. In this paper, we focus on cryptographic techniques.

CCSW'10, October 8, 2010, Chicago, Illinois, USA.

1.2 Contribution

In order to achieve strong privacy guarantees in a hostile environment such as that of cloud computing, a common practice is to keep data always encrypted and perform all operations on the ciphertexts. To this end, we introduce the concept of Privacy Preserving Mapping (PPM) schemes supporting comparison, formally denoted as

$$(\mathsf{KeyGen}, \mathsf{Mapping}, \mathsf{Compare}).$$

A PPM scheme enables a user to map her data items into images in such a way that, with a set of images, any other entity can determine the $<, =, >$ relationships among the corresponding data items. Combined with a standard encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ with semantic security, a PPM scheme enables a user to outsource her data items m_i $(i \geq 1)$ to a cloud service provider in the form of

$$\{\mathsf{Mapping}(\mathsf{m_i}, \cdot), \mathsf{Enc}(m_i, \cdot) \mid i \geq 1\}.$$

As a result, the cloud service provider can sort the data items, generate indexes, and search on them, yet with limited access to the plaintext data items (how much information is accessible depends on the privacy notion of the PPM).

We propose three privacy notions for PPM, namely ideal privacy, level-1 privacy, and level-2 privacy. The ideal privacy guarantees that the images, generated by the Mapping algorithm, reveal no more information about the corresponding data items than their $<, =, >$ relationships. The level-1 privacy guarantees that the images reveal only the mutual distances between the corresponding data items. The level-2 privacy is something between ideal privacy and level-1 privacy, it reveals only the $<, =, >$ relationships of the mutual

distances between the corresponding data items, instead of the true distances as in the case of level-1 privacy. For each privacy notion, we propose a scheme satisfying the privacy property.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we introduce the concept of PPM and formulate three privacy notions, namely ideal privacy, level-1 privacy, and level-2 privacy. In Section 3, we propose a scheme with ideal privacy. In Section 4, we propose a scheme with level-1 privacy. In Section 5, we propose a scheme with level-2 privacy. In Section 6, we briefly review some related work. In Section 7, we conclude the paper.

2. FORMULATIONS OF PPM SCHEMES

Suppose that $\mathcal{S}$ is a public data set. A PPM scheme for $\mathcal{S}$ consists of three algorithms (KeyGen, Mapping, Compare).

- KeyGen$(\ell, \mathcal{S})$: This algorithm takes a security parameter ℓ and the data set $\mathcal{S}$ as input, and outputs a public/private key pair (pk, sk).
- Mapping(x_i, sk): This algorithm takes $x_i \in \mathcal{S}$ and the private key sk as input, and outputs an image for x_i, referred to as T_{x_i}.
- Compare(T_{x_i}, T_{x_j}, pk): This algorithm takes two images T_{x_i}, T_{x_j} and the public key pk as input, and outputs 1 if $x_i > x_j$, 0 if $x_i = x_j$, or -1 if $x_i < x_j$.

For a PPM scheme, the KeyGen and Mapping algorithms are run by a user, while the Compare algorithm can be run by any entity. Straightforwardly, given a set of images T_{x_i} $(1 \leq i \leq n)$ where n is an integer, any entity can repeatedly run the Compare algorithm to generate a permuted set $\{T_{x'_1}, T_{x'_2}, \cdots, T_{x'_n}\}$ such that $x'_1 \leq x'_2 \leq \cdots \leq x'_n$.

2.1 Definition of Ideal Privacy

We first describe the following observation.

Observation. *Compared with symmetric/asymmetric key encryption schemes, a PPM scheme inherently leaks more information about transformed data items since any entity can compare x_i, x_j given T_{x_i}, T_{x_j}. Intuitively, if x_i, x_j are integers and from the data set $\mathcal{S} = [0, M]$, then $x_i < x_j$ implies that $x_i \neq M$ and $x_j \neq 0$. Moreover, the more images are disclosed, the more information about the corresponding data items is leaked. Nonetheless, in the PPM setting, this kind of information leakage is necessary and reasonable.*

With respect to the privacy property of a PPM scheme, an adversary represents all entities other than the user. We assume the adversary is passive, which means that it has access to the public parameters and the images disclosed by the user but is not allowed to submit a data item (chosen by by itself) to the user to obtain the corresponding image. Since the adversary can always sort the corresponding data items with the obtained images, therefore, we assume that the adversary is given $\mathcal{T}_0 = \{T_{x_1}, T_{x_2}, \cdots, T_{x_n}\}$, where n is an integer and $x_1 < x_2 < \cdots < x_n$. It is worth noting the order that these images are disclosed to the adversary does not affect our analysis. Moreover, if the Mapping algorithm is probabilistic, we assume the adversary can have multiple images of a data item.

Inspired by the analysis of public key encryption schemes [2], we adopt a similar approach to evaluate the security of PPM schemes. Given a set of images $\mathcal{T}_0 = \{T_{x_1}, T_{x_2}, \cdots, T_{x_n}\}$, where $x_1 < x_2 < \cdots < x_n$, the privacy leakage is measured by the indistinguishability from $\mathcal{T}_1 = \{T_{y_1}, T_{y_2}, \cdots, T_{y_n}\}$ for any $y_1 < y_2 < \cdots < y_n$. Formally, we give the following definition.

DEFINITION 1. *A PPM scheme achieves ideal privacy, if any polynomial-time adversary has only a negligible advantage in the attack game shown in Fig. 1, where the advantage is defined to be $|\Pr[b' = b] - \frac{1}{2}|$.*

1. Setup phase: the challenger runs the KeyGen algorithm to generate a public/private key pair (pk, sk).
2. Phase 1: The adversary sends $\mathcal{C}_0$ and $\mathcal{C}_1$ to the challenger for a challenge, where
   $\mathcal{C}_0 = \{x_1, x_2, \cdots, x_n\}$ such that $x_1 < x_2 < \cdots < x_n$,
   $\mathcal{C}_1 = \{y_1, y_2, \cdots, y_n\}$ such that $y_1 < y_2 < \cdots < y_n$.
   Without loss of generality, we assume that $x_1 \leq y_1$.
3. Challenge phase: The challenger selects $b \in_R \{0, 1\}$ and sends $\mathcal{T}_b$ to the adversary, where
   $\mathcal{T}_0 = \{T_{x_1}, T_{x_2}, \cdots, T_{x_n}\}, \mathcal{T}_1 = \{T_{y_1}, T_{y_2}, \cdots, T_{y_n}\}$.
4. Phase 2: For each obtained image, the adversary can request more images for the same data item[a]. The adversary outputs a guess bit b'.

[a]This reflects the fact that the user may disclose different images for the same data item when the Mapping algorithm is probabilistic.

Figure 1: The Game for Ideal Privacy

In the above definition, we use the term "ideal privacy" because a secure PPM scheme under this definition reveals no more information about the data items than that implied by the comparison functionality. In practice, some application scenarios may have weaker privacy requirements, so that we propose two relaxed privacy notions for PPM schemes.

2.2 Definition of Level-1 Privacy

Let a and b be two integers. Clearly, their distance, namely $a - b$, is enough to tell which number is larger. For a PPM scheme, if we can construct images in such a way so that given T_{x_i}, T_{x_j} any entity can obtain $x_i - x_j$ but nothing else, then it is straightforward to design the Compare algorithm. Given T_{x_i} $(1 \leq i \leq n)$, such a PPM scheme leaks no more information than the mutual distances between x_i $(1 \leq i \leq n)$. Formally, we give the following definition.

DEFINITION 2. *A PPM scheme achieves level-1 privacy, if any polynomial-time adversary has only a negligible advantage in the attack game shown in Fig. 1, with the following additional requirement on $\mathcal{C}_0$ and $\mathcal{C}_1$: $y_i - y_j = x_i - x_j$ for any $1 \leq i, j \leq n$.*

2.3 Definition of Level-2 Privacy

Suppose that we have a PPM scheme secure under Definition 2 and possesses the following property[1]: an entity can indeed learn $x_i - x_j$ from T_{x_i}, T_{x_j}. For example, the scheme described in Section 4 is such a scheme. Let's consider an extreme situation when such a scheme is used.

Example. *Suppose that the user has disclosed $T_{x_i}, T_{x_j}, T_{x_k}$, where $x_i, x_j, x_k \in [0, M]$, $x_j - x_i = \frac{M}{2}$, and $x_k - x_i = M$. Then any entity can learn that x_i is 0, x_j is $\frac{M}{2}$, and x_k is M.*

Although it is an extreme example, this indicates that a PPM scheme secure under Definition 2 could possibly reveal a lot of information. Moreover, given two PPM schemes secure against Definition 1 and 2 separately, there is a big gap between their privacy guarantees in practice. To bridge the gap, we introduce another privacy notion, namely level-2 privacy, under which the images T_{x_i} $(1 \leq i \leq n)$ leak no more information than the $<, =, >$ relationships of the mutual distances between x_i and x_j for any $1 \leq i, j \leq n$. We illustrate the idea by the following example.

Example. *Suppose that the user has disclosed $T_{x_i}, T_{x_j}, T_{x_k}$, which satisfy $x_i - x_j = x_j - x_k$. Then, for any $y_i - y_j = y_j - y_k$, then an adversary will only succeed with a negligible advantage in distinguishing $\{T_{x_i}, T_{x_j}, T_{x_k}\}$ from $\{T_{y_i}, T_{y_j}, T_{y_k}\}$. However, if $x_i - x_j = x_j - x_k$ and $y_i - y_j \neq y_j - y_k$, which means that the relative relationships of distances between the data items are different, then an adversary may succeed with a non-negligible advantage.*

Formally, we give the following definition.

DEFINITION 3. *A PPM scheme achieves level-2 privacy, if any polynomial-time adversary has only a negligible advantage in the attack game shown in Fig. 1, with the following additional requirement on $\mathcal{C}_0$ and $\mathcal{C}_1$.*

1. *For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j = y_k - y_l$ then $x_i - x_j = x_k - x_l$.*
2. *For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j < y_k - y_l$ then $x_i - x_j < x_k - x_l$.*

From the descriptions of Definition 2 and 3, it is clear that any $\mathcal{C}_0, \mathcal{C}_1$ satisfying the requirements in Definition 2 will also satisfy the requirements in Definition 3, but the vise versa is not true. Consequently, it means that any PPM secure under Definition 3 is always secure under Definition 2, but vise versa is not true.

3. SCHEME WITH IDEAL PRIVACY

3.1 Description of the Scheme

For this PPM scheme, we assume that the public data set $\mathcal{S}$ contains integers $1 \leq i \leq N$. The algorithms are defined as follows.

[1]It is worth noting that a PPM scheme secure against Definition 2 does not necessarily have this property. For example, a PPM with ideal privacy also achieves level-1 privacy.

- $\mathsf{KeyGen}(\ell, \mathcal{S})$: This algorithm generates a symmetric key $sk \in \{0,1\}^\ell$ and selects two hash functions $\mathsf{H}_1 : \{0,1\}^* \to \{0,1\}^\ell$ and $\mathsf{H}_2 : \{0,1\}^{2\ell} \to \{0,1\}^\ell$. This algorithm also generates a public list $\mathcal{L}$ which is a random permutation of the following set

  $$\{\mathsf{H}_2(\mathsf{H}_1(sk||j)||\mathsf{H}_1(sk||i)) \mid 1 \leq i, j \leq N \text{ and } i < j\}.$$

  The public key is $(\mathsf{H}_1, \mathsf{H}_2, \mathcal{L})$.
- $\mathsf{Mapping}(i, sk)$: For any $1 \leq i \leq N$, this algorithm generates an image $T_i = \mathsf{H}_1(sk||i)$.
- $\mathsf{Compare}(T_x, T_y, pk)$: Given T_x and T_y, this algorithm outputs 0 if $T_x = T_y$, outputs 1 if $\mathsf{H}_2(T_x||T_y)$ is in the public list $\mathcal{L}$, and outputs -1 otherwise.

In the next subsection, we prove that this scheme achieves the ideal privacy. Note that, in the execution of the KeyGen algorithm, the user needs to perform $\frac{N(N+1)}{2}$ hash operations to generate the list $\mathcal{L}$, which requires a storage of $\frac{N(N-1)}{2}$ hash values. Due to the computing and storage limitations of current computer systems, for this scheme, the data set can only be polynomial size, namely N is a polynomial in the security parameter ℓ. In practice, this may be a possible drawback for some applications. It is worth noting that, with respect to storage, existing techniques such as Bloom filter [3] can be used to improve the performance.

3.2 Security Analysis

LEMMA 1. *The above PPM scheme achieves ideal privacy (defined in Definition 1) given that H_1 and H_2 are random oracles.*

Proof sketch. Suppose that an adversary has the advantage ϵ in the attack game shown in Fig. 1. The security proof is done through a sequence of games [9].

Game_0: In this game, the challenger faithfully simulates the protocol execution and answers the oracle queries from $\mathcal{A}$. Let $\delta_0 = \Pr[b' = b]$, as we assumed at the beginning, $|\delta_0 - \frac{1}{2}| = \epsilon$.

Game_1: The challenger performs faithfully as in Game_0, except for instantiating the public list $\mathcal{L}$ with values randomly chosen from $\{0,1\}^\ell$. If a query is made to the oracle H_2 with the input $\mathsf{H}_1(sk||j)||\mathsf{H}_1(sk||i)$ where $1 \leq i < j \leq N$, the challenger returns a randomly chosen value from $\mathcal{L}$ given that this value has not been a response to another query. Let $\delta_1 = \Pr[b' = b]$ at the end of this game. If H_2 is modeled as a random oracle, Game_1 is identical to Game_0 so that $\delta_1 = \delta_0$.

Game_2: The challenger performs faithfully as in Game_1, except for the following.

1. In the challenge phase of the game, the challenger randomly chooses N values r_t $(1 \leq t \leq N)$ from $\{0,1\}^\ell$, and returns r_i $(1 \leq i \leq n)$ as the challenge.
2. If a query is made to the oracle H_2 with the input $r_j||r_i$ where $1 \leq i < j \leq n$, the challenger returns a randomly chosen value from $\mathcal{L}$ given that this value has not been a response to another query.

If H_1 and H_2 are modeled as random oracles, Game_2 is identical to Game_1 except that the following event *Ent* occurs:

- a query is made to H_1 with an input of the form $sk||*$, where $*$ can be any string, or

- a query is made to H_2 with an input of the form $r_t||*$ or $*||r_t$, where $*$ can be any string and $n+1 \leq t \leq N$.

Let $\delta_2 = \Pr[b' = b]$ in this game. If the event Ent does not occur, we have $\delta_2 = \frac{1}{2}$ since the challenge returned to the adversary is generated independent from $\mathcal{C}_0$ and $\mathcal{C}_1$. Since H_1 and H_2 are random oracles and r_t $(n+1 \leq t \leq N)$ randomly chosen from $\{0,1\}^\ell$, it is straightforward to verify that $\Pr[Ent]$ is negligible. From the Difference Lemma in [9], we have $|\delta_2 - \delta_1| \leq \Pr[Ent]$ and

$$\epsilon = |\delta_0 - \frac{1}{2}| = |\delta_1 - \frac{1}{2}| \leq |\delta_2 - \delta_1| + |\delta_2 - \frac{1}{2}| = \Pr[Ent].$$

Since $\Pr[Ent]$ is negligible, the lemma now follows. □

4. SCHEME WITH LEVEL-1 PRIVACY

4.1 Description of the Scheme

For this PPM scheme, we also assume that the public data set $\mathcal{S}$ contains integers $1 \leq i \leq N$. The algorithms are defined as follows.

- $\mathsf{KeyGen}(\ell, \mathcal{S})$: This algorithm generates a symmetric key $sk \in \{0,1\}^{\ell+d_N}$, where d_N is the bit-length of N. The public key is an empty string.
- $\mathsf{Mapping}(i, sk)$: For any $1 \leq i \leq N$, this algorithm generates an image $T_i = sk + i$.
- $\mathsf{Compare}(T_x, T_y, pk)$: Note that the values of T_x and T_y are in the following forms

  $$T_x = sk + x, T_y = sk + y.$$

  The algorithm outputs 0 if $T_x = T_y$, outputs 1 if $T_x - T_y > 0$, and outputs -1 otherwise.

In the next subsection, we prove that this scheme achieves the level-1 privacy. Compared with the previous PPM scheme, the data set can be exponential size for this scheme, and the KeyGen algorithm is extremely efficient.

4.2 Security Analysis

LEMMA 2. *The above PPM scheme achieves level-1 privacy (defined in Definition 2) unconditionally.*

Proof sketch. For the above PPM scheme, since $\mathcal{C}_0$ and $\mathcal{C}_1$ satisfy that $y_i - y_j = x_i - x_j$ for any $1 \leq i, j \leq n$, to prove the lemma, it is sufficient to show that the adversary's advantage is negligible in the case where $\mathcal{C}_0 = \{x_1\}$ and $\mathcal{C}_1 = \{y_1\}$.

Since sk is chosen from $\{0,1\}^{\ell+d_N}$ uniformly at random, T_{x_1} is uniformly distributed over

$$\{x_1, x_1+1, \cdots, y_1, y_1+1, \cdots, 2^{\ell+d_N}, \cdots, 2^{\ell+d_N} + x_1 - 1\},$$

while T_{y_1} is uniformly distributed over

$$\{y_1, y_1+1, \cdots, 2^{\ell+d_N}, \cdots, 2^{\ell} + x_1, \cdots, 2^{\ell+d_N} + y_1 - 1\}.$$

Note that we assume $x_1 \leq y_1$. Consequently, given a value from $\{y_1, y_1+1, \cdots, 2^{\ell}, \cdots, 2^{\ell} + x_1 - 1\}$, it is impossible to tell whether it is T_{x_1} or T_{y_1}, namely $\Pr[b' = b] = \frac{1}{2}$ holds unconditionally. As a result, an adversary can distinguish T_{x_1} from T_{y_1} with the advantage ϵ.

$$\begin{aligned}
\epsilon &= |\frac{1}{2} \cdot \frac{2^{\ell+d_N} + x_1 - y_1}{2^{\ell+d_N} + y_1 - x_1} + (1 - \frac{2^{\ell+d_N} + x_1 - y_1}{2^{\ell+d_N} + y_1 - x_1}) - \frac{1}{2}| \\
&\leq |\frac{1}{2} \cdot \frac{2^{\ell+d_N} + x_1 - y_1}{2^{\ell+d_N} + y_1 - x_1} - \frac{1}{2}| + |1 - \frac{2^{\ell+d_N} + x_1 - y_1}{2^{\ell+d_N} + y_1 - x_1}| \\
&\leq |\frac{1}{2} \cdot \frac{2(y_1 - x_1)}{2^{\ell+d_N} + y_1 - x_1}| + |\frac{2(y_1 - x_1)}{2^{\ell+d_N} + y_1 - x_1}| \\
&< 3 \cdot \frac{2^{d_N}}{2^{\ell+d_N}} < \frac{1}{2^{\ell-2}}.
\end{aligned}$$

Since $\frac{1}{2^{\ell-2}}$ is negligible with respect to the security parameter ℓ, the lemma now follows. □

5. SCHEME WITH LEVEL-2 PRIVACY

5.1 preliminary

Suppose that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative groups of prime order p, and h_1 and h_2 are randomly chosen generators respectively. We assume that there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$, but there is an efficiently computable bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

- Bilinear: for any $a, b \in \mathbb{Z}_p$, we have $\hat{e}(h_1^a, h_2^b) = \hat{e}(h_1, h_2)^{ab}$.
- Non-degenerate: $\hat{e}(h_1, h_2) \neq 1$.

In the above pairing setting, we introduce two new problems, namely *extended computational/decisional problems with hidden exponent*. Suppose that α is randomly chosen from $\mathbb{Z}_p$ and g_1 and g_2 are randomly chosen generators for $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Suppose also that N is an integer of polynomial size in the security parameter. The computational problem is to let an adversary compute $\hat{e}(g_1, g_2)^{\alpha^y}$, where $y \in [1, N]$ and is not equal to $x_j - x_i$ for any $1 \leq i, j \leq n$, when given

$$(x_1, x_2, \cdots, x_n; g_1^{\alpha^{x_1}}, g_1^{\alpha^{x_2}}, \cdots, g_1^{\alpha^{x_n}}; g_2^{\alpha^{-x_1}}, g_2^{\alpha^{-x_2}}, \cdots, g_2^{\alpha^{-x_n}}),$$

where $x_1 < x_2 < \cdots < x_n$ are any integers from $[1, N]$. Note that the parameters α, g_1, g_2 are kept secret. The decisional problem is modeled as a three-stage game between a challenger and an adversary.

1. The challenger generates $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e})$ as the public parameters and (α, g_1, g_2) the secret parameters.
2. Given the public parameters $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e})$, the adversary selects $x_1 < x_2 < \cdots < x_n$ and $y_1 < y_2 < \cdots < y_n$ from $[1, N]$ satisfying the following property: for any $1 \leq i, j, k, l \leq n$, $x_i - x_j = x_k - x_l$ iff $y_i - y_j = y_k - y_l$. This property eliminates the situation that an adversary can trivially distinguish the pairs by computing parings of the given group elements.
3. The challenger chooses $b \in_R \{0,1\}$ and sends X_b to the adversary, which returns a guess b'.

$$X_0 = (g_1^{\alpha^{x_1}}, g_1^{\alpha^{x_2}}, \cdots, g_1^{\alpha^{x_n}}; g_2^{\alpha^{-x_1}}, g_2^{\alpha^{-x_2}}, \cdots, g_2^{\alpha^{-x_n}}),$$

$$X_1 = (g_1^{\alpha^{y_1}}, g_1^{\alpha^{y_2}}, \cdots, g_1^{\alpha^{y_n}}; g_2^{\alpha^{-y_1}}, g_2^{\alpha^{-y_2}}, \cdots, g_2^{\alpha^{-y_n}}).$$

The adversary's advantage is defined to be $|\Pr[b' = b] - \frac{1}{2}|$.

DEFINITION 4. *The extended computational problem with hidden exponent is intractable if any polynomial adversary has only a negligible advantage in computing* $\hat{e}(g_1, g_2)^{\alpha^y}$, *while the extended decisional problem with hidden exponent is intractable if any polynomial adversary has only a negligible advantage in distinguishing between* X_0 *and* X_1.

A formal analysis of both assumptions is of independent interest, and will appear in an extended version of this paper.

5.2 Description of the Scheme

For this PPM scheme, we also assume that the public data set $\mathcal{S}$ contains integers $1 \leq i \leq N$. The algorithms are defined as follows.

- $\mathsf{KeyGen}(\ell, \mathcal{S})$: This algorithm generates the pairing parameters specified in Section 5.1, namely $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e})$. Let g_1 and g_2 be randomly chosen generators for $\mathbb{G}_1$ and $\mathbb{G}_2$, and α be randomly chosen from $\mathbb{Z}_p$. The private key is $sk = (\alpha, g_1, g_2)$ and the public key pk is defined to be

  $$pk = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e}, \mathcal{L}, \mathsf{H}),$$

  where $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ is a hash function and $\mathcal{L}$ is a random permutation of the following set

  $$\{\mathsf{H}(\hat{e}(g_1, g_2)^\alpha), \mathsf{H}(\hat{e}(g_1, g_2)^{\alpha^2}), \cdots, \mathsf{H}(\hat{e}(g_1, g_2)^{\alpha^N})\}.$$

- $\mathsf{Mapping}(i, sk)$: For any $1 \leq i \leq N$, this algorithm generates an image T_i where

  $$T_i = (g_1^{\alpha^i}, g_2^{\alpha^{-i}}).$$

- $\mathsf{Compare}(T_x, T_y, pk)$: Note that the values of T_x and T_y are in the following form

  $$T_x = (g_1^{\alpha^x}, g_2^{\alpha^{-x}}),\ T_y = (g_1^{\alpha^y}, g_2^{\alpha^{-y}}).$$

  The algorithm outputs 0 if $T_x = T_y$, outputs 1 if $\mathsf{H}(b) \in \mathcal{L}$, and outputs -1 otherwise.

  $$b = \hat{e}(g_1^{\alpha^x}, g_2^{\alpha^{-y}}) = \hat{e}(g_1, g_2)^{\alpha^{x-y}}.$$

Note that the user needs to perform 1 pairing, N exponentiation, and N hash operations to generate the list $\mathcal{L}$, which requires a storage of N hash values. This scheme is clearly less expensive than the scheme with ideal privacy, but, still, N can only be a polynomial in the security parameter ℓ. Similarly, Bloom filter [3] can be used to improve the storage performance.

5.3 Security Analysis

LEMMA 3. *Given N is a polynomial in the security parameter ℓ, the above PPM scheme achieves level-2 privacy (defined in Definition 3) in the random oracle model if the extended computational/decisional problems with hidden exponent are intractable.*

Proof sketch. Suppose an adversary has the advantage ϵ in the attack game shown in Fig. 1, with the following additional requirement on $\mathcal{C}_0$ and $\mathcal{C}_1$.

1. For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j = y_k - y_l$ then $x_i - x_j = x_k - x_l$.
2. For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j < y_k - y_l$ then $x_i - x_j < x_k - x_l$.

The security proof is done through a sequence of games [9].

Game_0: In this game, the challenger faithfully simulates the protocol execution and answers the oracle queries from $\mathcal{A}$. Let $\delta_0 = \Pr[b' = b]$, as we assumed at the beginning, $|\delta_0 - \frac{1}{2}| = \epsilon$.

Game_1: The challenger performs faithfully as in Game_0, except for instantiating the public list $\mathcal{L}$ with values randomly chosen from $\{0,1\}^\ell$. If a query is made to the oracle H with the input $\hat{e}(g_1, g_2)^{\alpha^i}$ where $1 \leq i \leq N$, the challenger returns a value randomly chosen from $\mathcal{L}$ given that this value has not been a response to another query. Let $\delta_1 = \Pr[b' = b]$ at the end of this game. If H is modeled as a random oracle, Game_1 is identical to Game_0 so that $\delta_1 = \delta_0$.

Game_2: The challenger performs faithfully as in Game_1, except for the following.

1. If a query is made to the oracle H with the input $\hat{e}(g_1, g_2)^{\alpha^\beta}$, the challenger aborts as a failure when the following event Ent occurs.
   (a) If $b = 0$, the $\beta \in [1, N]$ and is not equal to $x_i - x_j$ for any $1 \leq i, j \leq n$.
   (b) If $b = 1$, the $\beta \in [1, N]$ and is not equal to $y_i - y_j$ for any $1 \leq i, j \leq n$.
2. If a query is made to the oracle H with the input $\hat{e}(g_1, g_2)^{\alpha^t}$ where $1 \leq t \leq N$ and t does not fall in the above case, the challenger returns a randomly chosen value from $\mathcal{L}$ given that this value has not been a response to another query.

The value of $\Pr[Ent]$ is negligible based on the extended computational problem with hidden exponent. Let $\delta_2 = \Pr[b' = b]$ at the end of this game. From the Difference Lemma in [9], we have $|\delta_2 - \delta_1| \leq \Pr[Ent]$. Based on the extended decisional problem with hidden exponent, we have $|\delta_2 - \frac{1}{2}| \leq \epsilon'$ where ϵ' is negligible. As a result, we have

$$\epsilon = |\delta_0 - \frac{1}{2}| = |\delta_1 - \frac{1}{2}| \leq |\delta_2 - \delta_1| + |\delta_2 - \frac{1}{2}| \leq \Pr[Ent] + \frac{\epsilon'}{2}.$$

Since $\Pr[Ent]$ and ϵ' are negligible, the lemma now follows. □

6. RELATED WORK

The concept of PPM is closely related to that of Order Preserving Encryption (OPE), which was proposed by Agrawal *et al.* [1] and then further investigated by Boldyreva *et al.* [4]. An OPE scheme $(\mathsf{K}, \mathsf{Enc}, \mathsf{Dec})$ guarantees that if $x < y$ then $\mathsf{Enc}(\mathsf{x}, \cdot) < \mathsf{Enc}(\mathsf{x}, \cdot)$ holds, and it has been considered as a useful primitive because it allows operations such as indexing and queries to be done on the encrypted data in the same way as on the plaintext data. So far, it remains as an open problem to construct an OPE scheme under a conventional security notion such as semantic security. The main difficulty is that such a construction needs to simultaneously achieve three properties, namely plaintext recoverability, plaintext privacy, and ciphertext order-preserving property. Combined with a standard encryption scheme $(\mathsf{K}', \mathsf{Enc}', \mathsf{Dec}')$, we can construct a new encryption scheme $(\mathsf{K}, \mathsf{Enc}, \mathsf{Dec})$ based on a PPM scheme $(\mathsf{KeyGen}, \mathsf{Mapping}, \mathsf{Compare})$.

- Let the output of K to be those of both K' and KeyGen.
- Let the output of $\mathsf{Enc}'(m,\cdot)$ be $(\mathsf{Enc}'(m,\cdot), \mathsf{Mapping}(m,\cdot))$.
- Let Dec be the same as Dec'.

The resulted encryption scheme provides similar functionalities to that of an OPE scheme. Informally, the PPM indirectly provides the ciphertext order-preserving property since any entity can order the plaintexts with the PPM images through $\mathsf{Compare}$ operations, the encryption scheme provides plaintext recoverability, and both PPM and the standard encryption scheme guarantees plaintext privacy. As a result, we argue that, together with standard encryption schemes, PPM is a practical alternative to OPE in practice. However, it is an interesting future work to investigate the detailed security properties of this hybrid construction of OPE.

Besides OPE, the concept of PPM is also related to other cryptographic primitives that supports comparison operations on encrypted data, such as public key encryption with keyword search (PEKS) [5], public key encryption with registered keyword search (PERKS) [10], and encryption schemes supporting conjunctive, subset, or range queries [6, 8]. The main difference is that, in these schemes, private keys or equivalent secrets need to be distributed to an entity in order to enable her to perform the comparison.

More generally, the PPM primitive can be regarded as a special form of secure 2-party computation [11], which has been a fruitful research area with numerous results. The speciality of PPM lies in its non-interactive nature, where any entity can compare the data items in disclosed images without any interaction with the user.

7. CONCLUSION

In this paper, we have introduced the concept of Privacy Preserving Mapping (PPM) schemes supporting comparison, and proposed three privacy notions with three constructions satisfying these privacy notions. Our constructions serve the purposes of successful instantiation of our privacy notions, yet it is an interesting work to investigate new constructions. In particular, it is an interesting work to construct schemes with ideal privacy but without the limitations of a polynomial message space and expensive precomputations. In addition, we have only considered a passive adversary in the security model, it is also an interesting work to consider an active adversary, which may obtain data item and image pairs, and then to investigate the detailed security properties of the hybrid construction of OPE.

Acknowledgement

This is an ongoing work carried out in the Kindred Spirits project, which is sponsored by STW's Sentinels program in the Netherlands. The author would like to thank Steven Galbraith and Hoon Wei Lim for their discussions.

8. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 563–574. ACM, 2004.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.

[3] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[4] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2009.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public Key Encryption with Keyword Search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[6] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC'07: Proceedings of the 4th conference on Theory of cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[7] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 199–212. ACM, 2009.

[8] E. Shi, J. Bethencourt, H. T.-H. Chan, D. X. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *2007 IEEE Symposium on Security and Privacy*, pages 350–364. IEEE Computer Society, 2007.

[9] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. http://shoup.net/papers/, 2006.

[10] Q. Tang and L. Chen. Public-key encryption with registered keyword search. In *Proceeding of Public Key Infrastructure, 5th European PKI Workshop: Theory and Practice (EuroPKI 2009)*, volume ??? of *Lecture Notes in Computer Science*, page ??? Springer, 2009.

[11] A. Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, 1982.

$\{y_1, y_2, \cdots, y_n\}$ such that $y_i - y_j = x_i - x_j$ for any $1 \leq i, j \leq n$

$\{y_1, y_2, \cdots, y_n\}$ such that

- For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j = y_k - y_l$ then $x_i - x_j = x_k - x_l$.
- For any $1 \leq i, j, k, l \leq n$, if $y_i - y_j < y_k - y_l$ then $x_i - x_j < x_k - x_l$.