Security Analysis and Enhancement for Prefix-Preserving Encryption Schemes

Liangliang Xiao
University of Texas at Dallas
xll052000@utdallas.edu

I-Ling Yen
University of Texas at Dallas
ilyen@utdallas.edu

Abstract

Prefix-preserving encryption (PPE) is an important type of encryption scheme, having a wide range of applications, such as IP addresses anonymization, prefix-matching search, and rang search. There are two issues in PPE schemes, security proof and single key requirement. Existing security proofs for PPE only reduce the security of a real PPE scheme to that of the ideal PPE object by showing their computational indistinguishability [1, 14]. Such security proof is incomplete since the security of the ideal encryption object is unknown. Also, existing prefix-preserving encryption schemes only consider a single encryption key, which is infeasible for a practical system with multiple users (Implying that all users should have the single encryption key in order to encrypt or decrypt confidential data).

In this paper we develop a novel mechanism to analyze the security of the ideal PPE object. We follow the modern cryptographic approach and create a new security notion IND-PCPA. Then, we show that such weakened security notion is necessary and the ideal PPE object is secure under IND-PCPA. We also design a new, security-enhanced PPE protocol to support its use in multi-user systems, where no single entity in the system knows the PPE key. The protocol secret shares and distributes the PPE key to a group of key agents and let them "distributedly encrypt" critical data. We develop a novel distributed PPE algorithm and the corresponding request and response protocols. Experimental results show that the protocol is feasible in practical systems.

Key Words: Prefix-preserving encryption, security notion, IND-CPA, IND-PCPA, cryptographical security proof, multi-user systems, distributed prefix-preserving encryption algorithm.

1 Introduction

Prefix-preserving encryption (PPE) supports search on ciphertexts. In PPE, the longest common prefix of any two ciphertexts is of the same length as the longest common prefix of the corresponding plaintexts. Two PPE algorithms have been proposed in the literature. In [14], a PPE was first proposed for securely processing real-world Internet traffic traces without disclosing the IP addresses in them. The PPE is constructed bit by bit, where the i-th bit of the ciphertext is constructed by applying an instantiating function to the previous $i-1$ bits of the plaintext to preserve the prefix consistency. In [10], the authors suggested that PPE constructed in [14] can also be used to support range search on encrypted data. For a given range $[a, b]$, the range query can be transformed into at most $2\log_2 b - 1$ prefix-matching queries. In [1], the authors designed a PPE to support secure processing of prefix-matching queries (such as searching area-code starting with 310), where the prefix is generalized to a sequence of blocks (e.g. 64 bits or 4 UTF-16 characters) instead of a sequence of bits in [14]. To search for matching entries with a prefix x, the system encrypts x into $\mathcal{E}(x)$ and use $\mathcal{E}(x)$ as the prefix to perform prefix matching on the ciphertexts. Since the plaintext has a matching prefix of x if and only if the corresponding ciphertext has a matching prefix of $\mathcal{E}(x)$, the prefix matching computation can be achieved in logarithmic-time if the ciphertexts are organized in some standard tree data structures.

Although PPE can be applied to various applications, the security of PPE is weakened since some prefix information of plaintexts is leaked from ciphertexts. Thus, security analysis of PPE becomes crucial. Unfortunately, existing work does not offer sufficient security analysis of the PPE schemes. Most of the existing security analyses of the PPE schemes are informal: either they prove the security against the author-defined attacks, or they illustrate the security based on experiments. The authors in [1] initiated the cryptographic study of PPE scheme. They defined the security notion by the real PPE scheme and the ideal PPE object. In the ideal PPE object, the encryption function is uniformly randomly selected from the set of all prefix-preserving functions. A real PPE scheme is defined to be "secure" if it is computationally indistinguishable from the ideal PPE object. According to this security definition, the authors proved that their real PPE construction based on HCBC is "secure" (i.e. computationally indistinguishable from the ideal PPE object). In fact, the authors in [14] have also proved that their PPE scheme is computationally indistinguishable from the ideal PPE object, except that they did not user the crypto terminologies.

These security proofs are incomplete because the security of the ideal PPE object is unknown. If the security of the ideal object is unacceptable, then the proof of indistinguishability between the real scheme and the ideal object is not very indicative in security assurance. In modern cryptography, the security notions IND-CPA (indistinguishability under the chosen-plaintext attack) and IND-CCA (indistinguishability under the chosen-ciphertext attack) [9] have been developed to qualify the security of various encryption schemes. In [2], a weakened security notion IND-DCPA (indistinguishability under the distinct chosen-plaintext attack) is defined for the security qualification of the deterministic symmetric encryption scheme. However, the attempt for finding a qualifying security notion for the OPE (order preserving encryption, a similar approach as PPE) schemes has not been successful. In [3], the weakened security notion IND-OCPA (indistinguishability under the ordered chosen-plaintext attack) is defined to qualify the security of OPE scheme, but no OPE scheme that can satisfy this security notion has been found. Such attempt for PPE has not been considered due to the same challenge. In this paper, we develop a novel mechanism to analyze the security of the ideal PPE object. We follow the same approach to seek a necessarily and sufficiently weakened security notion to qualify the security of the ideal PPE object. First, we prove that no PPE scheme is secure under IND-CPA by designing a DLLCP attack (let the adversary query two plaintext pairs with different lengths of longest common prefix strings). Then we weaken the security notion from IND-CPA to IND-PCPA (indistinguishability under the prefixed chosen-plaintext attack) and prove that (1) such weakened security notion is necessary (otherwise the DLLCP attack will be successful), and (2) the ideal PPE object is secure under IND-PCPA. From (1) and (2), we conclude that the security notion IND-PCPA exactly qualifies the security of the idea PPE object.

Another limitation in all the existing PPE schemes is that there is no consideration of users. Generally, a system would deal with multiple users with different access privileges. Consider a database hosted in the cloud and is accessed by many users. The critical data in the database are encrypted by a PPE scheme using a master encryption key. The server should not have the knowledge of the master key. When a user sends a query to the server, critical data in the query need to be encrypted and the returned results need to be decrypted. PPE schemes allow the database server to perform lookup operations on encrypted data without knowing the encryption key. However, in order for the server to have interactions with users, some entities in the system need to know the encryption key. In conventional PPE schemes, it is implicitly assumed that the users know the master key and, hence, are able to encrypt and decrypt the corresponding data. However, in practice, giving the master key to all the users is insecure. There is a significant probability for the server (or an adversary who compromises the server) to collude with one of the users and compromise the entire database.

A potential solution to the problem above is to use different encryption keys for different data. But it may not be easy to design a PPE to support search on data that are encrypted using different keys. Another possible solution is to use a key agent to host the keys to perform encryption and decryption to bridge the users and the server. However, in this case, the key agent will know the master key and the client data and, hence, defeat the purpose of the PPE scheme. Thus, we need provide a secure protocol to support multiple users communicating with the server while satisfying that (1) no entity in the system has the knowledge of the master encryption key, and (2) the users' data is only known to the users, no one else. Our solution to achieve the goal is to use multiple key agents and secret share the master encryption key. Consider a PPE system consisting of a server DB hosting data encrypted by PPE using a master encryption key k. Assume that a user sends a query to DB which contains a confidential data x. In our PPE protocol, k is secret shared and distributed to the group of key agents. The user secret shares its confidential data x and passes the shares to the key agents. The key agents then "distributedly" encrypt the data shares into cipher shares, which in turn, are assembled into the ciphertext by the DB.

However, "distributed encryption" for PPE is not straightforward. In this paper, we develop a novel distributed PPE algorithm based on the PPE algorithm $\mathcal{E}$ constructed in [14]. In the original PPE algorithm, the i-th bit of the ciphertext $y = y_1 \cdots y_l$ is computed from the first i bits of the plaintext $x = x_1 \cdots x_l$ such that $y_i = x_i \oplus L(R(x_1, \cdots, x_{i-1}, k))$, where L denotes the least significant bit and R can be any pseudorandom function. If we directly make this algorithm distributed, it is necessary for the key agents to "distributedly" compute the shares of $R(x_1, \cdots, x_{i-1}, k)$ and then $L(R(x_1, \cdots, x_{i-1}, k))$ based on the shares of $x_1, \cdots, x_{i-1}$ and the shares of k. First, we invent a method based on the threshold technique developed in [11] to "distributedly" evaluate the shares of $R(x_1, \cdots, x_{i-1}, k)$ based on the shares of $x_1, \cdots, x_{i-1}$ and k. However, we still have the problem for "distributedly" evaluating the shares of $L(R(x_1, \cdots, x_{i-1}, k))$ based on the shares of $R(x_1, \cdots, x_{i-1}, k)$. Note that $L(R(x_1, \cdots, x_{i-1}, k))$ and $R(x_1, \cdots, x_{i-1}, k)$ are in different domains. It is very difficult to directly map the shares of $R(x_1, \cdots, x_{i-1}, k)$ to the domain of $L(R(x_1, \cdots, x_{i-1}, k))$ consistently. It is also insecure for any entity to reconstruct $R(x_1, \cdots, x_{i-1}, k)$. In fact, L is used to extract the least significant bit. Thus, instead of distributedly computing L, we eliminate L and let the bit x_i be encrypted to a block $z_i = H'(x_i) \cdot R(x_1 \cdots x_{i-1}, k)$, where H' is a hash function, hashing x_i to a block.

Although $z = z_1 \cdots z_l$ can be used to realize the desired PPE protocol, its large size will increase the overhead during search (comparisons). We design a reduction algorithm RA to reduce $z = z_1 \cdots z_l$ to the final ciphertext $y = y_1 \cdots y_l$, where y_i is a single bit, $1 \leq i \leq l$. Note that If some plaintexts have a common prefix $x_1 \cdots x_{i-1}$, then they have the same $R(x_1 \cdots x_{i-1}, k)$ value. Thus, their z_i can only be two values since $H'(x_i)$ can only have two values. Thus, we map z_i to an arbitrary y_i value if z_i has not appeared before; otherwise, it is mapped to the same y_i assigned previously. As long as we maintain the mapping history, the reduction can be done consistently.

The response can be processed in a reverse way of the request processing. However, in many data centric applications, it is likely that the response would contain many confidential data objects. The cost for "distributedly decrypt" many confidential data objects may be high. Thus, we use a simple, but very effective solution, which maintains two ciphers, one encrypted using PPE with the master encryption key and the other encrypted using a regular encryption scheme (different objects are encrypted with different keys). Search can be performed on the ciphers from PPE. In a response, we only need to include the ciphertexts encrypted using the regular encryption algorithm, greatly reducing the overhead.

Here we summarize our contributions in this paper.

- This paper is the first to successfully define a security notion, IND-PCPA, to exactly qualify the security of PPE. Specifically, we design the DLLCP attack to show that it is necessary to weaken the security notion from IND-CPA to IND-PCAP for the ideal PPE object. We also proof that the ideal PPE object is secure under IND-PCPA.

- We develop a PPE protocol to support the multi-user systems, making PPE feasible for practical use. The major invention in this protocol is the distributed PPE encryption by a group of key agents. We cryptographically prove the security of our protocol by defining an ideal model for PPE protocols and showing that our PPE protocol is computationally indistinguishable from the ideal model. We also conduct experiments to study the performance of the protocol, showing that it has a reasonable overhead.

The rest of the paper is organized as follows. In Section 2, we present the concept of PPE scheme and the ideal PPE object, and define the security notion IND-CPA. In Section 3 we design the DLLCP attach and show that PPE scheme is not secure under IND-CPA due to this attack. Then, we weaken the security notion from IND-PCA to IND-PCPA for PPE and prove that the ideal PPE object is secure under IND-PCPA. In Section 4 we extend PPE to multi-user systems. Experimental studies and performance results for our multi-user PPE protocol is presented in Section 5. Finally, Section 6 concludes the paper.

2 Background

This section introduces the background of the PPE scheme and the security analysis techniques. In Subsection 2.1, we present the concept of PPE schemes and introduce the ideal PPE object definition given in [1]. Then, in Subsection 2.2, we define the standard security notion IND-CPA.

Here we define some preliminary notations. Let λ be the security parameter and ν be a negligible function. Let $x \xleftarrow{\$} A$ denote that x is uniformly randomly selected from set A, $x \xleftarrow{\$} \mathcal{X}$ denote that randomized algorithm $\mathcal{X}$ returns value x, and $\mathcal{X}^{\mathcal{Y}}$ denote that algorithm $\mathcal{X}$ is accessible to oracle $\mathcal{Y}$.

2.1 PPE Schemes and the Ideal PPE Object

PPE algorithm should have the prefix-preserving property: the longest common prefix of any two ciphertexts is of the same length as the longest common prefix of the corresponding plaintexts. Assume that the plaintexts and ciphertexts are in $\{0,1\}^l$, where $\{0,1\}^l$ denotes the set of binary strings of length l. Let $LCP(x_1, x_2)$ denote the longest common prefix function which returns the longest common prefix of two binary strings x_1 and x_2, and $|LCP(x_1, x_2)|$ denote the length of $LCP(x_1, x_2)$. Then the PPE scheme can be defined as follows.

Definition 2.1 (PPE scheme [14]). A PPE scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a deterministic symmetric-key encryption scheme, where $\mathcal{K} : \{0,1\}^* \to \{0,1\}^*$ is a key generation algorithm, $\mathcal{E} : \{0,1\}^l \times \{0,1\}^* \to \{0,1\}^l$ is a deterministic symmetric-key encryption algorithm, and $\mathcal{D} : \{0,1\}^l \times \{0,1\}^* \to \{0,1\}^l$ is a decryption algorithm. The encryption algorithm $\mathcal{E}$ satisfies the "prefix-preserving" property:

$$|LCP(x_1, x_2)| = |LCP(\mathcal{E}(x_1, k), \mathcal{E}(x_2, k))|$$

for any $x_1, x_2 \in \{0,1\}^l$ and key k. □

For the ideal PPE object, the encryption function is uniformly randomly selected from all the prefix-preserving functions. Let

$$F^{PPE}_{\{0,1\}^l,\{0,1\}^l} \triangleq \{f : \{0,1\}^l \to \{0,1\}^l \mid |LCP(x_1, x_2)| = |LCP(f(x_1), f(x_2))|, \forall x_1, x_2 \in \{0,1\}^l\}$$

be the set of prefix-preserving functions mapping $\{0,1\}^l$ to $\{0,1\}^l$. In Lemma 2.2, we (1) prove that the prefix-preserving functions are invertible and, hence, the ciphertexts of the ideal PPE can be decrypted, and (2) compute the cardinality of $F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$ which will be used to prove the equivalence of the prefix-preserving function and the tree-based function in Proposition 3.6.

Lemma 2.2. *f is a bijection for any $f \in F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$ and $\left|F^{PPE}_{\{0,1\}^l,\{0,1\}^l}\right| = 2^{2^l - 1}$.*

Proof. For $f \in F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$, since the domain and range of f have the same (finite) cardinality, it suffices to prove that f is injective. Assume that $f(x_1) = f(x_2)$. Then $|LCP(x_1, x_2)| = |LCP(f(x_1), f(x_2))| = l$. Hence $x_1 = x_2$.

Let $N(l)$ denote the number of prefix-preserving functions with domain and range $\{0,1\}^l$. For $l = 1$ there are two prefix-preserving functions, which are $f(0) = 0$ and $f(1) = 1$; $f(0) = 1$ and $f(1) = 0$. Thus, $N(1) = 2$.

Let f_1 and f_2 denote any two prefix-preserving functions with domain and range $\{0,1\}^{l-1}$. Then it can be used to construct the prefix-preserving function f and g with domain and range $\{0,1\}^l$. For $x \in \{0,1\}^l$, let $x = x_1 \cdots x_l$ where $x_i \in \{0,1\}$, $1 \le i \le l$. We define

$$f(x_1x_2\cdots x_l) \triangleq \begin{cases} 0f_1(x_2\cdots x_l) & \text{if} \quad x_1 = 0 \\ 1f_2(x_2\cdots x_l) & \text{if} \quad x_1 = 1 \end{cases} \quad \text{and} \quad g(x_1x_2\cdots x_l) \triangleq \begin{cases} 1f_1(x_2\cdots x_l) & \text{if} \quad x_1 = 0 \\ 0f_2(x_2\cdots x_l) & \text{if} \quad x_1 = 1 \end{cases}.$$

It can be verified that f and g are different prefix-preserving functions and any prefix-preserving functions with domain and range $\{0,1\}^{l-1}$ must agree with the form of f or g. Hence, $N(l) = 2N(l-1)^2$. We can derive $N(l) = 2^{2^l-1}$ by solving the close form of $N(l)$ from the established equations $N(l) = 2N(l-1)^2$ and $N(1) = 2$. □

Now we present the formal definition of the ideal PPE object in Definition 2.3.

Definition 2.3 (Ideal PPE object). We say that $\mathcal{SE}^*(\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ is the ideal PPE object if

- $\mathcal{K}^*$ uniformly randomly selects $f \in F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$;

- $\mathcal{E}^*$ encrypts x to $f(x)$;

- $\mathcal{D}^*$ decrypts y to $f^{-1}(y)$. □

Remark 1. The ideal PPE object is computationally infeasible since it involves choosing f uniformly randomly from the set $F^{PPE}_{\{0,1\}^\lambda,\{0,1\}^\lambda}$, which is on the order of $2^{2^\lambda-1}$. In [1], the authors construct a real PPE scheme $\mathcal{SE}(\mathcal{K}, \mathcal{E}, \mathcal{D})$, and prove that the real PPE scheme is computationally indistinguishable from the ideal PPE object.

2.2 Security Notion IND-CPA

Let $\mathcal{LR}$ denote a left-or-right encryption oracle which knows the description of a given symmetric-key encryption scheme $\mathcal{SE}$ and the encryption key k. Also, let $\mathcal{A}$ denote an adversary who also knows the descriptions of $\mathcal{SE}$, but not the key k. The definition of the security notion IND-CPA is defined as follows.

Definition 2.4 (IND-CPA). Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme and $b \in \{0,1\}$. Let $\mathcal{E}_k(\mathcal{LR}(\cdot,\cdot,b))$ be a left-or-right encryption oracle such that for queries $\{(x_u^0, x_u^1)\}_{u=1}^h$, it returns

$$\mathcal{E}(x_u^b, k) \xleftarrow{\$} \mathcal{E}_k(\mathcal{LR}(x_u^0, x_u^1, b))$$

for $1 \le u \le h$. Let $\mathcal{A}$ be an adversary that can access $\mathcal{E}_k(\mathcal{LR}(\cdot,\cdot,b))$ and finally returns a bit b' as a guess of b. Consider the following experiment.

$$\textbf{\textit{Experiment}}\ \mathbf{Exp}^{\text{IND-CPA-}b}_{\mathcal{SE},\mathcal{A}}$$
$$k \xleftarrow{\$} \mathcal{K};\, b' \xleftarrow{\$} \mathcal{A}^{\mathcal{E}_k(\mathcal{LR}(\cdot,\cdot,b))};\, \text{Return } b'$$

The encryption scheme $\mathcal{SE}$ is said to be secure under IND-CPA if for every probabilistic polynomial time (PPT) adversary $\mathcal{A}$, the advantage of $\mathcal{A}$, defined by

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{SE},\mathcal{A}} = \Pr[\mathbf{Exp}^{\text{IND-CPA-1}}_{\mathcal{SE},\mathcal{A}} = 1] - \Pr[\mathbf{Exp}^{\text{IND-CPA-0}}_{\mathcal{SE},\mathcal{A}} = 1],$$

is bounded by a negligible function of the security parameter. □

3 Security Proof for PPE Schemes

Existing cryptographic security proofs for PPE schemes only reduce the security of real PPE schemes to the security of the ideal PPE object by showing that they are computationally indistinguishable. However it is not a complete security proof since the security of the ideal PPE object is unknown and there has been no security analysis in the literature to show its security level. In this section, we complete the existing security proof by developing a security notion IND-PCPA and showing that it exactly qualifies the ideal PPE object.

In Subsection 3.1, we design a DLLCP attack and use it to prove that PPE is not secure under IND-CPA. Then we define a weakened security notion IND-PCPA in Subsection 3.2. Such weakening is necessary because otherwise the PPE will be broken by the DLLCP attack. In Subsection 3.3.2, we prove that the ideal PPE object is secure under IND-PCPA.

3.1 Security Notion IND-CPA and DLLCP Attack

IND-CPA is a well established security notion in cryptography. However, PPE schemes require the ciphertexts to preserve the prefix of the plaintexts and cannot be qualified by IND-CPA. Consider the following DLLCP (differentiated length of longest common prefix) attack against the PPE scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with respect to IND-CPA.

Adversary $\mathcal{A}$

In the experiment $\mathbf{Exp}_{\mathcal{SE},\mathcal{A}}^{\text{IND-CPA-}b}$, $\mathcal{A}$ chooses the set of plaintext pairs

$$\{(x_1^0, x_1^1), (x_2^0, x_2^1) \mid LCP(x_1^0, x_2^0) \neq LCP(x_1^1, x_2^1)\},$$

and sends it to $\mathcal{LR}$;

$\mathcal{LR}$ computes the set of ciphertexts $\{\mathcal{E}(x_i^b, k)\}_{i=1}^{2}$, and sends it back to $\mathcal{A}$;

Finally $\mathcal{A}$ outputs $b' = \begin{cases} 0 & \text{if} \quad |LCP(x_1^0, x_2^0)| = |LCP(\mathcal{E}(x_1^b, k), \mathcal{E}(x_2^b, k))|; \\ 1 & \text{otherwise.} \end{cases}$

In the DLLCP attack, the adversary queries (x_1^0, x_1^1) and (x_2^0, x_2^1), where $LCP(x_1^0, x_2^0) \neq LCP(x_1^1, x_2^1)$. If $b = 0$, x_1^0 and x_2^0 will be encrypted; if $b = 1$, x_1^1 and x_2^1 will be encrypted. Since PPE preserves prefix, the adversary can distinguish whether the plaintexts are x_1^0 and x_2^0 or x_1^1 and x_2^1 by comparing $LCP(y_1, y_2)$ with $LCP(x_1^0, x_2^0)$ and $LCP(x_1^1, x_2^1)$, where y_1 and y_2 are the returned ciphertexts of the encryption oracle. If $LCP(y_1, y_2) = LCP(x_1^0, x_2^0)$, then the plaintexts are x_1^0 and x_2^0 and, hence, $b = 0$. If $LCP(y_1, y_2) = LCP(x_1^1, x_2^1)$, then the plaintexts are x_1^1 and x_2^1 and, hence $b = 1$. Thus, the advantage of the adversary $\mathcal{A}$ is 1. We summarize the conclusion in the following lemma.

Lemma 3.1. *PPE is not secure under IND-CPA.*

3.2 Security Notion IND-PCPA for the Ideal PPE object

In [3], it has been shown that an OPE scheme (order-preserving encryption scheme, requiring the ciphertexts to preserve the order of the plaintexts) does not satisfy IND-CPA and a weakened security notion IND-OCPA has been defined to qualify the security of OPE schemes (though no OPE schemes satisfy IND-OCPA either and no security notion has been found to properly qualify OPE yet). Inspired by this approach, we define a weaken security notion IND-PCPA to qualify the security of PPE schemes. According to the DLLCP attack, the adversary should only be allowed to query the plaintext pairs in the set

$$PPP_h = \{(x_i^0, x_i^1) \in \{0,1\}^l \times \{0,1\}^l, 1 \leq i \leq h \mid |LCP(x_u^0, x_v^0)| = |LCP(x_u^1, x_v^1)|, 1 \leq u, v \leq h\}.$$

Accordingly, we define the security notion IND-PCPA (indistinguishability under prefixed chosen-plaintext attack) in Definition 3.2.

Definition 3.2 (IND-PCPA)**.** IND-PCPA has the same definition as that of IND-CPA in Definition 2.4 except that the adversary is only allowed to query the prefixed plaintext pairs in the set PPP_h. □

It is obvious that IND-PCPA is the necessarily weakened security notion (with respect to indistinguishability and left-or-right encryption oracle) for PPE. We show that it is also the sufficiently weakened security notion for PPE by proving that the ideal PPE object is secure under IND-PCPA.

3.3 Security Proof of the Ideal PPE Object Under IND-PCPA

To prove that the ideal PPE object is secure under IND-PCPA, we need to show the number of the prefix-preserving functions mapping x_i^0 to $\mathcal{E}^*(x_i^b, k)$ equals to that of the prefix-preserving functions mapping x_i^1 to $\mathcal{E}^*(x_i^b, k)$, where (x_i^0, x_i^1) are the plaintext pairs the adversary queries, $1 \leq i \leq h$. In other words, there is no bias for the adversary's guess. However, the proof is not straightforward because it needs to use the prefix-preserving property to count the number of prefix-preserving functions mapping x_i^0 (resp. x_i^1) to $\mathcal{E}^*(x_i^b, k)$, where x_i^0, x_i^1, and $\mathcal{E}^*(x_i^b, k)$ are indeterminates, $1 \leq i \leq h$.

To overcome the difficulties, we represent the prefix-preserving function by the tree-based function. The tree-based function consists of a plaintext tree and a ciphertext tree. The plaintext tree is a complete binary tree. Each edge connecting a parent node to its left child node is labeled by 0, and each edge connecting a parent node to its right child node is labeled by 1. Each leaf node in the plaintext tree is labeled by the binary string composed of the labels of the edges from the root to itself (the label represents the plaintext string). The ciphertext tree is the same as the plaintext tree except for its labels. Each edge connecting a parent node to its left child node could be labeled by 0 or 1. If it is labeled by 0 (resp. 1), then the corresponding edge connecting the parent node to its right child node must be labeled by 1 (resp. 0). A tree-based function maps the i-th leaf node in the plaintext tree to the i-th leaf node in the cipertext tree. It implies that the labels of each path in the ciphertext tree (from the root node to the leaf node) represent the ciphertext of the plaintext represented by the corresponding path in the plaintext tree. In other words, the label of the i-th leaf node in the cipertext tree represents the cipertext of the label of the i-th leaf node in the plaintext tree.

Once the prefix-preserving function is represented by the tree-based function, it suffices to show that the number of the tree-based functions mapping x_i^0 to $\mathcal{E}^*(x_i^b, k)$ equals to that of the tree-based functions mapping x_i^1 to $\mathcal{E}^*(x_i^b, k)$, $1 \leq i \leq h$. An important observation is that given h plaintext ciphertext pairs, some labels of the edges in the ciphertext tree will be determined while others will not. Also, the number of the undetermined labels of the edges in the ciphertext tree decides the number of the tree-based functions. Therefore the security proof can be reduced to show that the number of the undetermined labels of the edges in the ciphertext tree given $(x_i^0, \mathcal{E}^*(x_i^b, k))$ equals to that of the undetermined labels of the edges in the ciphertext tree given $(x_i^1, \mathcal{E}^*(x_i^b, k))$, $1 \leq i \leq h$. We use mathematical induction on h to prove the equality of these two numbers.

3.3.1 Tree-Based Function Definition

Before defining the tree-based function, we first define some preliminary concepts. Then, the tree-based function is formally defined in Definition 3.5.

Definition 3.3. Let $T = (V^T, E^T)$ be a tree where V^T denotes the set of nodes and E^T denotes the set of edges. The nodes in V^T can be partitioned into the set of internal nodes V_I^T and the set of leave nodes V_L^T, where $V^T = V_I^T \bigcup V_L^T$ and $V_I^T \bigcap V_L^T = \emptyset$.

Let $v_i^{L,T}$ denote the i-th leaf node in T where the leave nodes are indexed from the left most leaf node (the first) to the right most leaf node (the $|V_L^T|$-th), $1 \leq i \leq |V_L^T|$. For $v \in V_L^T$ with depth n, let $P(v)$ denote the path from the root to v and $P(v)[1] \cdots P(v)[n+1]$ denote the nodes on the path, where $P(v)[1]$ is the root, $P(v)[n+1] = v$, and $P(v)[2], \cdots, P(v)[n]$ are internal nodes connecting root and v. Let $PI(v) = \{P(v)[i] \mid 1 \leq i \leq n\}$ denote the set of internal nodes on the path $P(v)$, $PL(v) = \{v\}$ denote the set of leaf node in the path $P(v)$, and $PE(v) = \{P(v)[i]P(v)[i+1] \mid 1 \leq i \leq n\}$ denote the set of edges on the path $P(v)$. □

In the tree-based function, the domain of the plaintexts and the range of the ciphertexts are two labeled trees. The labeling rules (defined in Definition 3.4) can guarantee the prefix-preserving property of the tree-based function. In Definition 3.4, we first define the internal nodes labeled (INL) tree where the internal nodes are labeled with 0 or 1, and define the nodes and edges labeled (NEL) tree where the labels are extended from the internal nodes to the edges and leave nodes.

Definition 3.4 (INL and NEL trees)**.** Internal nodes labeled (INL) tree is defined to be a pair $(T, \mathcal{L})$, where $T = (V^T, E^T)$ is a tree and

$$\mathcal{L} : V_I^T \to \{0, 1\}$$

is a label function over internal nodes, which is called INL function. Given an INL tree $(T, \mathcal{L})$, it uniquely defines the nodes and edges labeled (NEL) tree $(T, \mathcal{L}^*)$, where the NEL function

$$\mathcal{L}^* : V^T \bigcup E^T \to \{0, 1\}^*$$

is defined by the following rules.

(1) For $v \in V_I^T$, $\mathcal{L}^*(v) \triangleq \mathcal{L}(v)$.

(2) Let $e \in E^T$ where $e = v_{1e}v_{2e}$ and v_{1e}, v_{2e} denote the two endpoints of e. Without loss of generality, assume that v_{1e} is the parent node and v_{2e} is the child node. Then $\mathcal{L}^*(e) \triangleq \mathcal{L}(v_{1e})$ if v_{2e} is the left child node of v_{1e}; $\mathcal{L}^*(e) \triangleq 1 \oplus \mathcal{L}(v_{1e})$ if v_{2e} is the right child node of v_{1e}.

(3) For $v \in V_L^T$ with depth n, $\mathcal{L}^*(v)$ is a string of n bits. Let $PE(v) = \{P(v)[i]P(v)[i+1] \mid 1 \le i \le n\}$ denote the set of edges on the path $P(v)$, where $P(v)[1]$ is the root, $P(v)[n+1] = v$, and $P(v)[2], \cdots, P(v)[n]$ are internal nodes connecting root and v. Then

$$\mathcal{L}^*(v) \triangleq \mathcal{L}^*(P(v)[1]P(v)[2]) \cdots \mathcal{L}^*(P(v)[n]P(v)[n+1]). \qquad \Box$$

Now we are ready to define the tree-based function, which is given in Definition 3.5. The tree-based function is defined with respect to two NEL trees, which are called the plaintext tree and the ciphertext tree, respectively. It maps the label of the i-th leaf node in the plaintext tree to the i-th leaf node in the ciphertext tree.

Definition 3.5 (tree-based function)**.** The tree-based function is defined with respect to two NEL trees: the plaintext tree $PT_l = (T_{PT_l}, \mathcal{L}^*_{PT_l})$ and the ciphertext tree $CT_l = (T_{CT_l}, \mathcal{L}^*_{CT_l})$, where T_{PT_l} and T_{CT_l} are two complete binary trees with heights l. In the plaintext tree PT_l, the INL function $\mathcal{L}_{PT_l}(v) \equiv 0$ for any internal node $v \in V_I^{T_{PT_l}}$. But in the ciphertext tree CT_l, the INL function $\mathcal{L}_{CT_l}(v)$ could be 0 or 1 for any internal node $v \in V_I^{T_{CT_l}}$. The INL functions $\mathcal{L}_{PT_l}$ and $\mathcal{L}_{CT_l}$ uniquely define the NEL functions $\mathcal{L}^*_{PT_l}$ and $\mathcal{L}^*_{CT_l}$ following the rules defined in Definition 3.4.

Given $PT_l = (T_{PT_l}, \mathcal{L}^*_{PT_l})$ and $CT_l = (T_{CT_l}, \mathcal{L}^*_{CT_l})$, we define the corresponding tree-based function

$$\begin{aligned} &f_{PT_l,CT_l} : \{0,1\}^l \to \{0,1\}^l \\ &f_{PT_l,CT_l}(\mathcal{L}^*_{PT_l}(v_i^{L,T_{PT_l}})) \triangleq \mathcal{L}^*_{CT_l}(v_i^{L,T_{CT_l}}), \end{aligned}$$

where $v_i^{L,T_{PT_l}}$ and $v_i^{L,T_{CT_l}}$ denote the i-th leave nodes in the plaintext tree and ciphertext tree, respectively, $1 \le i \le 2^l$. Let TBF_l denote the set of all tree-based functions, i.e.,

$$TBF_l = \{f_{PT_l,CT_l} \mid \mathcal{L}_{CT_l} : V_I^{T_{CT_l}} \to \{0,1\}\}. \qquad \Box$$

Remark 2. In the definition of the tree-based function f_{PT_l,CT_l}, the plaintext tree PT_l is fixed since $\mathcal{L}_{PT_l}$ is fixed; but the ciphertext tree is not fixed. Since $\mathcal{L}_{CT_l}$ uniquely defines $\mathcal{L}^*_{CT_l}$, it also determines the ciphertext tree CT_l. Therefore, the INL function of ciphertext tree $\mathcal{L}_{CT_l}$ uniquely determines the tree-based function f_{PT_l,CT_l}.

We show the equivalence of the tree-based function and the prefix-preserving function in Proposition 3.6, but omit the proof due to the space limitation.

Proposition 3.6. $TBF_l = F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$.

Remark 3. According to the proof of Proposition 3.6, the **prefix-preserving property** of the tree-based function can be **geometrically interpreted** as follows. For any $x \in \{0,1\}^l$, it corresponds to the j-th leaf node $v_j^{L,T_{PT_l}}$ in the plaintext tree $V_L^{T_{PT_l}}$. Actually $j = B(x) + 1$ where $B(x)$ denotes the binary number of x. For $x_1, x_2 \in \{0,1\}^l$, the paths $P(v_{j_1}^{L,T_{PT_l}})$ and $P(v_{j_2}^{L,T_{PT_l}})$ on the plaintext tree share $|LCP(x_1, x_2)|$ many common edges. The tree-based function has the prefix-preserving property such that the paths $P(v_{j_1}^{L,T_{CT_l}})$ and $P(v_{j_2}^{L,T_{CT_l}})$ on the ciphertext tree also share $|LCP(x_1, x_2)|$ many common edges.

Based on Definition 2.3 and Proposition 3.6, we give an alternative definition for the ideal PPE object in Definition 3.7.

Definition 3.7 (alternative definition of the ideal PPE object)**.** It has the same definition as that of Definition 2.3 except that $\mathcal{K}^*$ uniformly randomly selects f from TBF_l instead of $F^{PPE}_{\{0,1\}^l,\{0,1\}^l}$. □

3.3.2 Security Proof

Now we prove that the ideal PPE object is secure under IND-PCPA. It also implies that the real PPE schemes, which are computationally indistinguishable to the ideal PPE object, achieve the highest security notion for PPE. Essentially, we need to show that in the security notion IND-PCPA, the number of the tree-based functions mapping x_i^0 to $\mathcal{E}^*(x_i^b, k)$ equals to that of the tree-based functions mapping x_i^1 to $\mathcal{E}^*(x_i^b, k)$, where (x_i^0, x_i^1) are the queried plaintexts pairs, $1 \le i \le h$. Since $\mathcal{L}_{CT_l}$ uniquely determines the tree-based function, in order to count those numbers, we need to consider the effect towards $\mathcal{L}_{CT_l}$ (partial mapping will be determined) when given the plaintext ciphertext pairs $(x_i^0, \mathcal{E}^*(x_i^b, k))/(x_i^1, \mathcal{E}^*(x_i^b, k))$, $1 \le i \le h$.

Lemma 3.8. *Given h plaintext ciphertext pairs (x_i, y_i) of f_{PT_l,CT_l}, then the labels of the internal nodes on h paths $P(v_{j_i})$ are determined, where v_{j_i} is decided by x_i and the labels are decided by y_i, $1 \le i \le h$.*

Proof. Consider plaintext ciphertext pair $(x_i, y_i) \in \{0,1\}^l \times \{0,1\}^l$ such that $f_{PT_l,CT_l}(x_i) = y_i$. We assume that $x_i = \mathcal{L}^*_{PT_l}(v_{j_i}^{L,T_{PT_l}})$ where $v_{j_i}^{L,T_{PT_l}}$ denotes the j_i-th leaf node in the plaintext tree, and $y_i = y_{i1} \cdots y_{il}$, $y_{iu} \in \{0,1\}$, $1 \le u \le l$. According to the definition of tree-based function,

$$y_i = f_{PT_l,CT_l}(x_i) = f_{PT_l,CT_l}(\mathcal{L}^*_{PT_l}(v_{j_i}^{L,T_{PT_l}})) = \mathcal{L}^*_{CT_l}(v_{j_i}^{L,T_{CT_l}}).$$

Therefore, $\mathcal{L}^*_{CT_l}(P(v_{j_i}^{L,T_{CT_l}})[u]P(v_{j_i}^{L,T_{CT_l}})[u+1]) = y_{iu}$ for $1 \le u \le l$ according to the definition of $\mathcal{L}^*_{CT_l}$. It implies that the labels of the edges on the path $P(v_{j_i}^{L,T_{CT_l}})$ are determined by y_i. Since the labels of the internal nodes on the path and the labels of the edges on the same path can be mutually decided, the labels of the internal nodes on the path $P(v_{j_i}^{L,T_{CT_l}})$ are decided by y_i. Hence, given the plaintext ciphertext pairs (x_i, y_i) where $x_i = \mathcal{L}^*_{PT_l}(v_{j_i}^{L,T_{PT_l}})$, for the INL function $\mathcal{L}_{CT_l}$, the labels of the internal nodes on the path $P(v_{j_i}^{L,T_{CT_l}})$ are determined, $1 \le i \le h$. □

Consider the adversary counting the number of tree-based functions mapping x_i^0/x_i^1 to $\mathcal{E}^*(x_i^b, k)$, $1 \le i \le h$. Since the tree-based function is uniquely determined by the INL function $\mathcal{L}_{CT_l}$ (Remark 2), it is equivalent to count the number of INL functions. The **important observation** is: according to Lemma 3.8, the labels of the internal nodes on the corresponding h paths are determined. Therefore it suffices to count the rest undetermined labels since they decides the number of INL functions. Following such idea, we use mathematical induction on h to prove that the two numbers are identical in Lemma 3.9.

Lemma 3.9. *The number of the tree-based functions mapping x_i^0 to $\mathcal{E}^*(x_i^b, k))$ equals to that of the tree-based functions mapping x_i^1 to $\mathcal{E}^*(x_i^b, k))$, $1 \le i \le h$.*

Proof. Let $x_i^0 = \mathcal{L}^*_{PT_l}(v_{j_i^0}^{L,T_{PT_l}})$ where $v_{j_i^0}^{L,T_{PT_l}}$ denotes the j_i^0-th leaf node in the plaintext tree, and $x_i^1 = \mathcal{L}^*_{PT_l}(v_{j_i^1}^{L,T_{PT_l}})$ where $v_{j_i^1}^{L,T_{PT_l}}$ denotes the j_i^1-th leaf node in the plaintext tree, $1 \le i \le h$. For tree-based functions mapping x_i^0 to $\mathcal{E}^*(x_i^b,k)$, the labels of the internal nodes on the path $P(v_{j_i^0}^{L,T_{CT_l}})$ in the ciphertext tree are determined; for tree-based functions mapping x_i^1 to $\mathcal{E}^*(x_i^b,k)$, the labels of the internal nodes on the path $P(v_{j_i^1}^{L,T_{CT_l}})$ in the ciphertext tree are determined, $1 \le i \le h$ (Lemma 3.8). Hence it suffices to prove that the determined labels of the internal nodes in that two ciphertext trees are assigned consistent values and the number of the undetermined labels of the internal nodes in that two ciphertext trees are identical, i.e.

$$|\cup_{1\le i\le h} PI(v_{j_i^0}^{L,T_{CT_l}})| = |\cup_{1\le i\le h} PI(v_{j_i^1}^{L,T_{CT_l}})| \tag{1}$$

where $PI(v)$ (defined in Definition 3.3) denotes the set of internal nodes on the path $P(v)$.

We use mathematical induction on h to prove it. For $h=1$, it is obvious that the labels in $PI(v_{j_1^0}^{L,T_{CT_l}})/PI(v_{j_1^1}^{L,T_{CT_l}})$ are assigned consistent values with respect to $\mathcal{E}^*(x_1^b,k)$ according to the proof of Lemma 3.8. Also $|PI(v_{j_1^0}^{L,T_{CT_l}})| = l-1 = |PI(v_{j_1^1}^{L,T_{CT_l}})|$. So we assume that the induction assumption holds for $h<h'$ and consider the situation for $h=h'$. According to the inductional assumption, the labels in $\cup_{1\le i\le h'-1}PI(v_{j_i^0}^{L,T_{CT_l}})/\cup_{1\le i\le h'-1}PI(v_{j_i^1}^{L,T_{CT_l}})$ are assigned consistent values. Also, $|\cup_{1\le i\le h'-1} PI(v_{j_i^0}^{L,T_{CT_l}})| = |\cup_{1\le i\le h'-1} PI(v_{j_i^1}^{L,T_{CT_l}})|$ and $|PI(v_{j_{h'}^0}^{L,T_{CT_l}})| = |PI(v_{j_{h'}^1}^{L,T_{CT_l}})|$. Since $(x_i^0,x_i^1) \in PPP_{h'}$, $LCP(x_{h'}^0,x_i^0) = LCP(x_{h'}^1,x_i^1)$ for $1\le i\le h'-1$ according to the definition of $PPP_{h'}$. Note that $x_i^0 = \mathcal{L}_{PT_l}^*(v_{j_i^0}^{L,T_{PT_l}})$ and $x_i^1 = \mathcal{L}_{PT_l}^*(v_{j_i^1}^{L,T_{PT_l}})$ for $1\le i\le h'$, we have $|PI(v_{j_{h'}^0}^{L,T_{PT_l}}) \cap PI(v_{j_i^0}^{L,T_{PT_l}})| = |PI(v_{j_{h'}^1}^{L,T_{PT_l}}) \cap PI(v_{j_i^1}^{L,T_{PT_l}})|$ for $1\le i\le h'-1$ according to the definition of NEL function $\mathcal{L}_{PT_l}^*$. Therefore

$$|PI(v_{j_{h'}^0}^{L,T_{CT_l}}) \cap PI(v_{j_i^0}^{L,T_{CT_l}})| = |PI(v_{j_{h'}^1}^{L,T_{CT_l}}) \cap PI(v_{j_i^1}^{L,T_{CT_l}})| \tag{2}$$

for $1\le i\le h'-1$ according to the conclusions in Remark 3. Without loss of generality, we assume $b=0$. So the labels in $PI(v_{j_{h'}^0}^{L,T_{CT_l}})/PI(v_{j_i^0}^{L,T_{CT_l}})$ are assigned consistent values for $1\le i\le h'-1$, i.e., the labels in $PI(v_{j_{h'}^0}^{L,T_{CT_l}}) \cap PI(v_{j_i^0}^{L,T_{CT_l}})$ are assigned the same values no matter with respect to $\mathcal{E}^*(x_{h'}^b,k)$ or $\mathcal{E}^*(x_i^b,k)$ for $1\le i\le h'-1$. Consequently, the labels in $PI(v_{j_{h'}^1}^{L,T_{CT_l}}) \cap PI(v_{j_i^1}^{L,T_{CT_l}})$ are assigned the same values no matter with respect to $\mathcal{E}^*(x_{h'}^b,k)$ or $\mathcal{E}^*(x_i^b,k)$ for $1\le i\le h'-1$ based on the proof in Lemma 3.8 and (2), which implies that the labels in $PI(v_{j_{h'}^1}^{L,T_{CT_l}})/PI(v_{j_i^1}^{L,T_{CT_l}})$ are assigned consistent values for $1\le i\le h'-1$. Therefore, the labels in $\cup_{1\le i\le h'}PI(v_{j_i^0}^{L,T_{CT_l}})/\cup_{1\le i\le h'}PI(v_{j_i^1}^{L,T_{CT_l}})$ are assigned consistent values. Also, let $1\le i_0\le h'-1$ such that

$$|PI(v_{j_{h'}^0}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^0}^{L,T_{CT_l}})| = \max_{1\le i\le h'-1}\{|PI(v_{j_{h'}^0}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^0}^{L,T_{CT_l}})|\}$$

and

$$|PI(v_{j_{h'}^1}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^1}^{L,T_{CT_l}})| = \max_{1\le i\le h'-1}\{|PI(v_{j_{h'}^1}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^1}^{L,T_{CT_l}})|\}.$$

Then

$$\begin{aligned}
|\cup_{1\le i\le h'} PI(v_{j_i^0}^{L,T_{CT_l}})| &= |(\cup_{1\le i\le h'-1}PI(v_{j_i^0}^{L,T_{CT_l}})) \cup PI(v_{j_{h'}^0}^{L,T_{CT_l}})| \\
&= |\cup_{1\le i\le h'-1} PI(v_{j_i^0}^{L,T_{CT_l}})| + |PI(v_{j_{h'}^0}^{L,T_{CT_l}})| - |PI(v_{j_{h'}^0}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^0}^{L,T_{CT_l}})| \\
&= |\cup_{1\le i\le h'-1} PI(v_{j_i^1}^{L,T_{CT_l}})| + |PI(v_{j_{h'}^1}^{L,T_{CT_l}})| - |PI(v_{j_{h'}^1}^{L,T_{CT_l}}) \cap PI(v_{j_{i_0}^1}^{L,T_{CT_l}})| \\
&= |(\cup_{1\le i\le h'-1}PI(v_{j_i^1}^{L,T_{CT_l}})) \cup PI(v_{j_{h'}^1}^{L,T_{CT_l}})| = |\cup_{1\le i\le h'} PI(v_{j_i^1}^{L,T_{CT_l}})|.
\end{aligned}$$

It completes the induction. $\square$

In Theorem 3.10, we prove the security of the ideal PPE object.

Theorem 3.10. *The ideal PPE object $\mathcal{SE}^*$ is secure under IND-PCPA.*

Proof. According to Proposition 3.6 and Lemma 3.9, the number of the prefix-preserving functions mapping x_i^0 to $\mathcal{E}^*(x_i^b, k))$ equals to that of the prefix-preserving functions mapping x_i^1 to $\mathcal{E}^*(x_i^b, k))$, $1 \leq i \leq h$. Therefore $\Pr(\mathbf{Exp}_{\mathcal{SE}^*,\mathcal{A}}^{\text{IND-PCPA-}b} = 1) = \frac{1}{2}$ for $b = 0, 1$. Hence,

$$\mathbf{Adv}_{\mathcal{SE}^*,\mathcal{A}}^{\text{IND-PCPA}} = \Pr(\mathbf{Exp}_{\mathcal{SE}^*,\mathcal{A}}^{\text{IND-PCPA-1}} = 1) - \Pr(\mathbf{Exp}_{\mathcal{SE}^*,\mathcal{A}}^{\text{IND-PCPA-0}} = 1) = 0,$$

which implies that the ideal PPE object $\mathcal{SE}^*$ is secure under IND-PCPA. $\square$

4 PPE for Multi-user Systems

In this section we develop a security-enhanced protocol to support PPE in multi-user systems. The multi-user system we consider consists of a single server hosting a database and a set of users. Let DB denote the server and $U = \{U_j \mid j \geq 1\}$ denote the set of users. The system operations consist of a request protocol Q, in which a user U_i issues a request (query) q to DB, where q may contain some secret data x that needs to be transmitted with q to DB, and a response protocol P, in which the DB sends back the response r to the user, where r may include a returned data object y in encrypted form. Note that a request or a response may include multiple data objects, but the processing will be the same. For simplicity, we assume that there is only one secret data item in q or r.

The PPE protocol should guarantee functionality requirements including: (1) When q reaches DB, x should have been encrypted by the PPE using the key k; (2) When r is returned to the user, the user should be able to obtain the plaintext y of r. The protocol should also satisfy some security requirements, such as no entity in the system should have the knowledge of the key k and x should be protected against all system entities but the owner. To avoid informal security descriptions and facilitate formal security proofs, we define the security requirements via an ideal model, in which (1) encryption and decryption are performed by a trusted party TP who holds the key k (key agents replaced by TP); (2) the communication channels between TP and users/DB are secure. The ideal model implies the highest security level that the real PPE protocol (including Q and P) can achieve and we will prove that The real protocol is "equivalent" to the ideal model.

The system entities, users, DB, and key agents, may collude to acquire additional information. We unify the possible collusions and construct a passive adversary $\mathcal{A}$ who tries to gain extra information by compromising some entities in the system. We assume that the key agents and DB are better protected than the users, and the adversary can compromise less than t key agents ($t \leq \frac{m}{2} + 1$) simultaneously. Thus, the adversary structure is defined as

$$\mathcal{Z} = \{U_\mathcal{A} \cup KA_\mathcal{A}, U_\mathcal{A} \cup KA_\mathcal{A} \cup \{DB\} \mid U_\mathcal{A} \subset U, |KA_\mathcal{A}| < t \leq \frac{m}{2} + 1\}, \tag{3}$$

where $U_\mathcal{A}$ is the set of compromised users and $KA_\mathcal{A}$ is the set of compromised key agents (note that $U_\mathcal{A}$ and $KA_\mathcal{A}$ could be empty).

In Subsection 4.1, we introduce the general system design. Then, we discuss the response and request protocols in the following two subsections. The proof that shows our PPE protocol achieves the functionality requirements is given in Subsection 4.4. In Subsection 4.4 we formally define the security requirement and prove that our PPE protocol achieves the requirement.

4.1 General System Design

For convenience, we assume that there are only numerical data in DB. Data of other types can be represented by numerical data easily. For each critical data item x, the DB maintains the ciphertexts $(C_{PPE}(x), C_{CE}(x))$.

$C_{PPE}(x)$ is encrypted using a PPE scheme with a master key k. $C_{CE}(x)$ is encrypted using a classical encryption scheme (e.g. AES). The purpose of storing $C_{CE}(x)$ is to support efficient transmission of responses. For each data item x, a different data key dk_x is used to generate $C_{CE}(x)$. A user with access privilege to data item x will be granted key dk_x. In real implementation, the data items with the same access privileges can be grouped together into an access domain and only one key is needed for each access domain. For example, if data items x and y can be accessed by exactly the same set of users, then x and y can be in the same access domain, i.e., we can have $dk_x = dk_y$.

Request protocol Q should transfer q to DB while ensuring the correct and secure encryption of $C_{PPE}(x)$ and $C_{CE}(x)$ in the request transmission process. (Note that U_i can encrypt x using dk_x and obtain $C_{CE}(x)$. But since U_i does not have the PPE master key k, it is not possible for U_i to compute $C_{PPE}(x)$. Thus, a set of key agents (KA) are introduced to perform PPE encryption in the request protocol. Let $KA = \{KA_j \mid 1 \leq j \leq m\}$ denote the set of key agents. The key k is shared among the key agents such that no single entity in the system knows the master encryption key k. The user shares x and sends the shares to the key agents in KA. The key agents distributedly encrypt the shares of x with the shares of k and send the encrypted shares to DB. DB reconstructs the shares and get the ciphertexts $C_{PPE}(x)$. The "distributed" encryption process is similar to the decryption process in the threshold public-key crypto system [13, 5, 6, 12, 7, 8]. For the response protocol P, we use a simple but innovative design to achieve efficiency without going through the key agents.

4.2 Response Protocol

In the response protocol P, w simply include $C_{CE}(y_1), C_{CE}(y_2), \cdots, C_{CE}(y_t)$ in r. The user should have access rights to $y_1, y_2, \cdots, y_t$ and, hence, should have the encryption keys to decrypt the data items in r. Consider the security of the system against adversary $\mathcal{A}$ (assume that $\mathcal{A}$ has not compromised the users in $U_{y_1}, \cdots, U_{y_t}$, where U_{y_j} is the set of users who can access y_j). Since the protocol only transfers $C_{CE}(y_1), C_{CE}(y_2), \cdots, C_{CE}(y_t)$, $\mathcal{A}$ cannot get the encryption keys and cannot compromises $y_1, y_2, \cdots, y_t$. Note that the design of P is fully discussed here and will not be discussed further.

4.3 Request Protocol

We design the request protocol Q which consists of the distributed PPE protocol $P_{\mathcal{E}_d}$ and the reduction algorithm RA. In $P_{\mathcal{E}_d}$, the key agents "distributedly" evaluated PPE $\mathcal{E}_d$ and the DB assembles the result shares into the intermediate ciphertext z. In RA, z is reduced (in size) to the single-bit ciphertext y based on a mapping function f. In the following subsections, we introduce the primitives used in the protocol and discuss the details of $P_{\mathcal{E}_d}$ and RA.

4.3.1 Primitives

Here we introduce the primitives used for constructing $P_{\mathcal{E}_d}$, including the secret sharing algorithm Π and reconstructing algorithm Re over $\mathbb{Z}_p$ where p is a prime number, another secret sharing algorithm Π' and reconstructing algorithm Re' over a multiplicative group G satisfying that the decisional Diffie-Hellman (DDH) problem is hard over G, the hash function H mapping strings to $\mathbb{Z}_p$, and the hash function H' mapping strings to G.

Let G be a cyclic group where $|G| = p$ and p is a prime number, and $g \in G$ be a generator. Without loss of generality, let G be a multiplicative group with the identity 1. We assume that the decisional Diffie-Hellman (DDH) problem is hard over G, i.e. (g, g^u, g^v, g^{uv}) and (g, g^u, g^v, g^w) are computationally indistinguishable for randomly selected u, v, w from $\mathbb{Z}_p$. Let $H : \{Null\} \cup \{0,1\}^* \to \mathbb{Z}_p$ be a cryptographic hash function, where $Null$ denotes the empty string. We assume that H is a random oracle, and then $R(x, k) \triangleq g^{H(x) \cdot k}$ is a pseudorandom function according to [11]. Let $H' : \{0,1\}^* \to G$ be a cryptographic hash function. Since a cryptographic hash function should be collision-free, we assume that $H'(0) \neq H'(1)$.

Let Π and Re be the sharing and reconstructing algorithms of the (t, m) threshold secret sharing scheme over $\mathbb{Z}_p$ [13]. For secret $x \in \mathbb{Z}_p$,

$$\Pi(x) = (s_1, \cdots, s_m) \in \mathbb{Z}_p^m,$$

where $s_i = f(i)$ and f is a randomly selected polynomial over $\mathbb{Z}_p$ with degree $t-1$ satisfying $f(0) = x$. The reconstructing algorithm

$$Re(s_{i_1}, \cdots, s_{i_n}) = x$$

for any n $(n \geq t)$ shares $s_{i_1}, \cdots, s_{i_n} \in \{s_1, \cdots, s_m\}$ by using the Lagrange's interpolation formula. The (t, m) threshold secret sharing scheme can be extended to the group G [6, 11]. Let Π' and Re' be the sharing and reconstructing algorithms of the (t, m) threshold secret sharing algorithm over G. For any secret $x' \in G$

$$\Pi'(x') = (s'_1, \cdots, s'_m) \in G^m,$$

where $s'_i = x' \cdot g^{f'(i)}$ and f' is a randomly selected polynomial over $\mathbb{Z}_p$ with degree $t-1$ satisfying $f'(0) = 0$. The reconstructing algorithm

$$Re'(s'_{i_1}, \cdots, s'_{i_n}) = x'$$

for any n $(n \geq t)$ shares $s'_{i_1}, \cdots, s'_{i_n} \in \{s'_1, \cdots, s'_m\}$ by using the Lagrange's interpolation formula to the exponents.

4.3.2 Protocol $P_{\mathcal{E}_d}$

Suppose that the master encryption key $k \in \mathbb{Z}_p$ is shared by Π, i.e., $\Pi(k) = (k_1, \cdots, k_m)$, and the share k_i is distributed to KA_i for $1 \leq i \leq m$. We describe the protocol $P_{\mathcal{E}_d}$ to "distributedly" evaluate the PPE algorithm $\mathcal{E}_d$ in Algorithm 1. It encrypts the plaintext $x = x_1 \cdots x_l$ to the intermediate ciphertext $z = z_1 \cdots z_l$.

Algorithm 1 Protocol $P_{\mathcal{E}_d}$

```
goal: distributedly encrypt $x = x_1 \cdots x_l$ to $z = z_1 \cdots z_l$
for $i = 1$ to $l$ do
    the user shares $H'(x_i)$ and $H(x_1, \cdots, x_{i-1})$ by $\Pi'$ and $\Pi$, respectively;
    let $\Pi'(H'(x_i)) = (h'_{i1}, \cdots, h'_{im})$ and $\Pi(H(x_1, \cdots, x_{i-1})) = (h_{i1}, \cdots, h_{im})$;
    for $j = 1$ to $m$ do the user sends $(h'_{ij}, h_{ij})$ to $KA_j$; end for
end for
for $i = 1$ to $l$ do
    for $j = 1$ to $m$ do $KA_j$ computes $h''_{ij} = h'_{ij} \cdot g^{h_{ij} \cdot k_j}$ and sends it to the $DB$; end for
end for
for $i = 1$ to $l$ do
    the $DB$ selects $n$ $(n \geq 2t-1)$ shares $h''_{ij_1}, \cdots h''_{ij_n}$ and computes $z_i = Re'(h''_{ij_1}, \cdots h''_{ij_n})$;
end for
the $DB$ retrieves the intermediate ciphertext $z = z_1 \cdots z_l$;
```

As shown in Algorithm 1, for plaintext $x = x_1 \cdots x_l$, the user shares $H'(x_i)$ and $H(x_1, \cdots, x_{i-1})$ to the key agents, the key agents distributedly encrypt it, and DB assembles the encrypted shares into an intermediate ciphertext $z = z_1 \cdots z_l$. We prove the correctness of this protocol in Lemma 4.1.

Lemma 4.1. *The DB retrieves the ciphertext encrypted by $\mathcal{E}_d$ in the end of the distributed protocol $P_{\mathcal{E}_d}$.*

Proof. According to $P_{\mathcal{E}_d}$, $H'(x_i)$ and $H(x_1, \cdots, x_{i-1})$ are shared by the user. Let $h'_{ij} = H'(x_i) \cdot g^{f'_i(j)}$ be the shares of $H'(x_i)$, where f'_i is a randomly selected polynomial over $\mathbb{Z}_p$ with degree $t-1$ satisfying $f'_i(0) = 0$, $1 \leq j \leq m$. Let $h_{ij} = f_i(j)$ be the shares of $H(x_1, \cdots, x_{i-1})$, where f_i is a randomly selected polynomial over $\mathbb{Z}_p$ with degree $t-1$ satisfying $f_i(0) = H(x_1, \cdots, x_{i-1})$, $1 \leq j \leq m$. The key agent KA_j will compute $h''_{ij} = h'_{ij} \cdot g^{h_{ij} \cdot k_j} = H'(x_i) \cdot g^{f'_i(j)} \cdot g^{f_i(j) \cdot k_j} = g^{(\log_g H'(x_i)) + f'_i(j) + f_i(j) \cdot k_j}$, $1 \leq j \leq m$. Notice that $(\log_g H'(x_i)) + f'_i(j)$ is the share of $\log_g H'(x_i)$ using a polynomial over $\mathbb{Z}_p$ with degree $t-1$, $1 \leq j \leq m$. And $f_i(j) \cdot k_j$ is the share of $H(x_1, \cdots, x_{i-1}) \cdot k$ using a polynomial over $\mathbb{Z}_p$ with degree $2t-2$, $1 \leq j \leq m$. Therefore $(\log_g H'(x_i)) + f'_i(j) + f_i(j) \cdot k_j$ is the share of $(\log_g H'(x_i)) + H(x_1, \cdots, x_{i-1}) \cdot k$ using a polynomial over $\mathbb{Z}_p$ with degree $2t-2$, $1 \leq j \leq m$. Hence the DB reconstructs

$$z_i = H'(x_i) \cdot R(x_1, \cdots, x_{i-1}, k),$$

where $R(x_1, \cdots, x_{i-1}, k) = g^{H(x_1, \cdots, x_{i-1}) \cdot k}$ by using n $(n \geq 2t-1)$ shares based on the Lagrange's interpolation to the exponents, $1 \leq i \leq l$. □

We show that $\mathcal{E}_d$ preserves prefix and it is secure under IND-PCPA in Lemma 4.2, but omit the proof due to the space limitation.

Lemma 4.2. *The encryption $\mathcal{E}_d$ preserves prefix (i.e., two plaintexts share i common prefix if and only if the ciphertexts share i common pre-blocks). Furthermore, it is secure under IND-PCPA.*

4.3.3 Reduction Algorithm

Since $\mathcal{E}_d(x) = z$ preserves prefix, the ciphertext $z = z_1 \cdots z_l$ can support prefix search already. But $\mathcal{E}_d$ increases the size of the ciphertext since z_i is a block instead of a bit. This can impact the search performance significantly. We develop a reduction algorithm RA to reduce the intermediate ciphertext $z = z_1 \cdots z_l$ to the final single-bit ciphertext $y = y_1 \cdots y_l$, $1 \leq i \leq l$. We use a mapping function f to record the mapping between z_i and y_i. For a node v, let $l(v)$ denote the left child node, $r(v)$ denote the right child node, $vl(v)$ denotes the edge connecting v and $l(v)$, and $vr(v)$ denotes the edge connecting v and $r(v)$. RA is designed in Algorithm 2.

Algorithm 2 The Reduction Algorithm RA

```
goal: reduce the intermediate ciphertext $z = z_1 \cdots z_l$ to the final ciphertext $y = y_1 \cdots y_l$
initialization: the mapping function $f$ = null
$v$ = root;
while $v \neq$ leaf node do
  if $f(vl(v)) = f(vr(v))$ = null then
    $b \xleftarrow{\$} \{0, 1\}$
    if $b = 0$ then $f(vl(v)) = z_i$; $y_i = 0$; $v = l(v)$;
    else $f(vr(v)) = z_i$; $y_i = 1$; $v = r(v)$;
    end if
  end if
  if $f(vl(v)) \neq$ null & $f(vr(v))$ = null then
    if $z_i = f(vl(v))$ then $y_i = 0$; $v = l(v)$;
    else $f(vr(v)) = z_i$; $y_i = 1$; $v = r(v)$;
    end if
  end if
  if $f(vr(v)) \neq$ null & $f(vl(v))$ = null then
    if $z_i = f(vr(v))$ then $y_i = 1$; $v = r(v)$;
    else $f(vl(v)) = z_i$; $y_i = 0$; $v = l(v)$;
    end if
  end if
  if $f(vl(v)) \neq$ null & $f(vr(v)) \neq$ null then
    if $z_i = f(vl(v))$ then $y_i = 0$; $v = l(v)$;
    else $y_i = 1$; $v = r(v)$;
    end if
  end if
end while
return $y = y_1 \cdots y_l$;
```

In Algorithm 2, if f has recorded that z_i has already been mapped to a bit, then the mapping should be retained. Otherwise, z_i is mapped to a chosen bit and f records this new mapping. Lemma 4.3 proves the security and prefix preserving properties of the algorithm, the proof is omitted due to the space limitation.

Lemma 4.3. *RA is efficient. Furthermore, the encryption algorithm $RA \circ \mathcal{E}_d$ preserves prefix and is secure under IND-PCPA.*

4.4 Functionalities and Security Requirements Proofs for the Protocols

In this section, we prove that our PPE protocol satisfies the functionality and security requirements. In Theorem 4.4 we prove that the request protocol Q and response protocol P satisfy the functionality requirements (1) and (2), respectively.

Theorem 4.4. *The request protocol Q realizes the functionality requirement (1) and the response protocol P realizes the functionality requirement (2).*

Proof. According to Lemmas 4.1, 4.2, and 4.3, the DB receives the ciphertext of x encrypted by the PPE $RA \circ \mathcal{E}_d$ in Q. Therefore the request protocol Q realizes the functionality requirement (1). In P the returned data object y will be encrypted. The recipient user will have the encryption key and, hence, can decrypt the ciphertext and obtain y. Hence the response protocol P realizes the functionality requirement (2). □

We adopt the security definition for multiparty computation [4] to define the security requirement for our system, which is based on real model and ideal model defined in Definition 4.5.

Definition 4.5 (Real model and ideal model)**.** The real model is exactly the request protocol Q and response protocol P. In the ideal model, there are users, the DB, and a trusted (incorruptible) party TP who holds the key. There are secure communication channels between the TP and users/DB. In the ideal model the TP receives/sends the message from/to users/DB, and does all the encryptions/decryptions needed in the protocols Q and P. □

Now we define the security requirement in Definition 4.6. Essentially, it requires that the real model is "equivalent" to the ideal model.

Definition 4.6 (Security requirement)**.** Let $VIEW_R(Z)$ be the instance event randomly selected from the event space of what the adversary $\mathcal{A}$ can observe in the real model by compromising entities in the set $Z \in \mathcal{Z}$. Let $VIEW_I(Z)$ be the instance event randomly selected from the event space of what the adversary $\mathcal{A}$ can observe in the ideal model by compromising the entities in the set $Z - KA$, the real model is secure if the adversary cannot retrieve more information from the real model than the ideal model, or equivalently, if there exists a PPT simulator $\mathcal{S}$ such that $VIEW_R(Z)$ is computationally indistinguishable from $\mathcal{S}(VIEW_I(Z))$, i.e. the advantage of $\mathcal{A}$, defined by

$$\mathbf{Adv}_{\mathcal{A}} \triangleq \Pr[\mathcal{A}(VIEW_R(Z)) = 1] - \Pr[\mathcal{A}(\mathcal{S}(VIEW_I(Z))) = 1],$$

is bounded by a negligible function of the security parameter for any $Z \in \mathcal{Z}$. □

We prove that our system achieves the security requirement in Theorem 4.7.

Theorem 4.7. *Our system achieves the security requirement in Definition 4.6.*

Proof. First we consider the security of Q. In both the real model and the ideal model, the adversary $\mathcal{A}$ can compromise some users and view the same thing. In the real model $\mathcal{A}$ can compromise less than t key agents; while in the ideal model $\mathcal{A}$ cannot compromise the trusted party TP. Since the user shares $H'(x_i)$ and $R(x_1, \cdots, x_{i-1}, k)$ to the key agents by using (t, m) secret sharing scheme and $\mathcal{A}$ compromises less than t shares, the view of $\mathcal{A}$ is random numbers and, hence, can be simulated by $\mathcal{S}$. In the real model $\mathcal{A}$ can compromise the DB and view the intermediate ciphertext $z = z_1 \cdots z_l$, the final ciphertext $y = y_1 \cdots y_l$, and the mapping function f; while in the ideal model $\mathcal{A}$ can only view the final ciphertext y. Since the difference between the intermediate ciphertext z and final ciphertext y is that z_i is a random block and y_i is a random bit. Therefore z can be simulated by $\mathcal{S}$ based on y. The mapping function f can be simulated accordingly based on z and y. Hence, $VIEW_R$ can be simulated by $\mathcal{S}$ based on $VIEW_I$.

Then we consider the security of P. In P only the users who can access rights to the data will have the key. Thus, the adversary $\mathcal{A}$ cannot get the encryption keys unless $\mathcal{A}$ compromises the corresponding users. Therefore P achieves the security requirement because the adversary cannot achieve more information in P than in the ideal model. □

5 Experimental Study

We conduct experiments to study the performance of the request protocol Q. Specifically, we study the performance of $\mathcal{E}_d$ since it is the dominant factor. First, we consider the secret sharing factor in $\mathcal{E}_d$. In $\mathcal{E}_d$, two secret sharing schemes have been used, one is Π and Re over $\mathbb{Z}_p$, and the other is Π' and Re' over G. Various groups can be used for G and here we use the Schnorr group, i.e., G is a multiplicative subgroup of $\mathbb{Z}_q^*$, where $|G| = p$ and p is a 256-bit prime number. We implemented the algorithms and ran them 10^4 trials on a PC with 2.50GHz Intel Core 2 Duo Processor. The average execution times are shown in Figure 1.

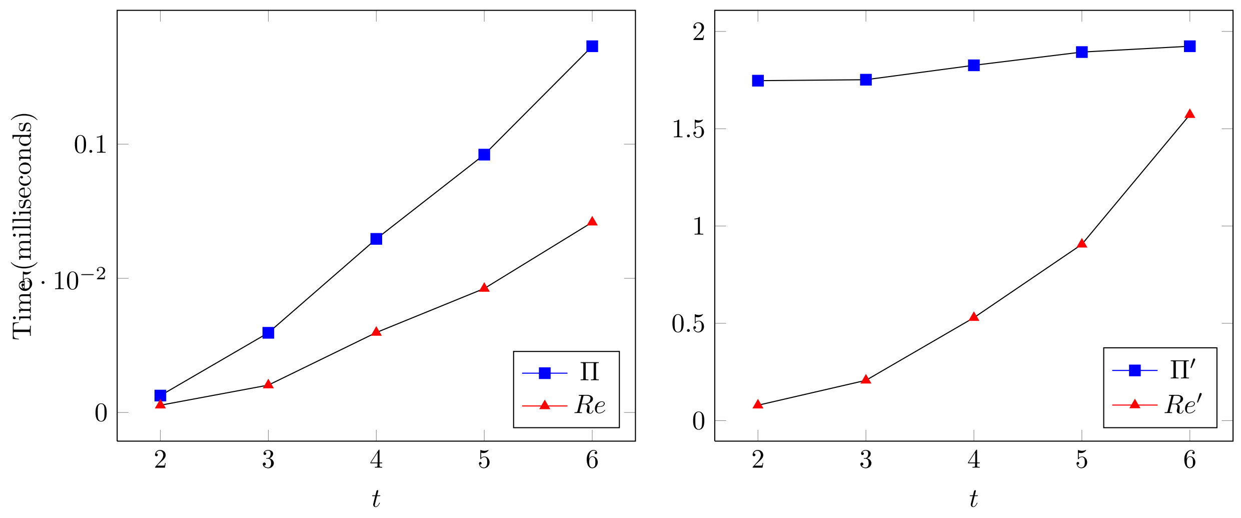

Figure 1: Computation Cost of Secret Sharing over $\mathbb{Z}_p$ and G (Share Number $m = 6$).

As shown in Figure 1, Π' and Re' has a higher computation cost than Π and Re because the computation of Π' and Re' needs extra group operations. Since the Lagrange's interpolation is linear, the reconstruction algorithm has a lower computation cost than the sharing algorithm which requires polynomial evaluation. Both the sharing time and the reconstruction time increase when the threshold t increases (which is obvious from the sharing and reconstruction approach).

To factor in the communication latencies between the system entities, we allocate the user, the key agents and the DB to different PlanetLab computers and measure the communication latencies between them. The user is in Dallas and the DB is in Los Angeles. Six key agents (i.e., $m = 6$) are allocated to Phoenix (Arizona), Salt Lake City (Utah), Carson City (Nevada), Eugene (Oregon), Albuquerque (New Mexico), and Denver (Colorado). Both hash functions H and H' are SHA-2. We assume that the request message without the critical data is of size 170 bytes (based on the average size of some common queries). The critical data size is l bits and we set $l = \{8, 16, 32, 64, 128, 256, 512, 1024\}$. Since the threshold t should satisfy the condition $t \leq \frac{m}{2} + 1$ (Condition (3)), we set $t = 2, 3, 4$. For comparison purpose, we also consider a "No Encryption" protocol and a "PPE" protocol. In "No Encryption", the user directly sends the query (with the critical data) to the DB without any encryption. In "PPE", we assume that the user has the encryption key and encrypts the critical data by the PPE constructed in [14], and then sends the query (with the encrypted critical data) to the DB. The experimental results are given in Figure 2 and summarized in Table 1.

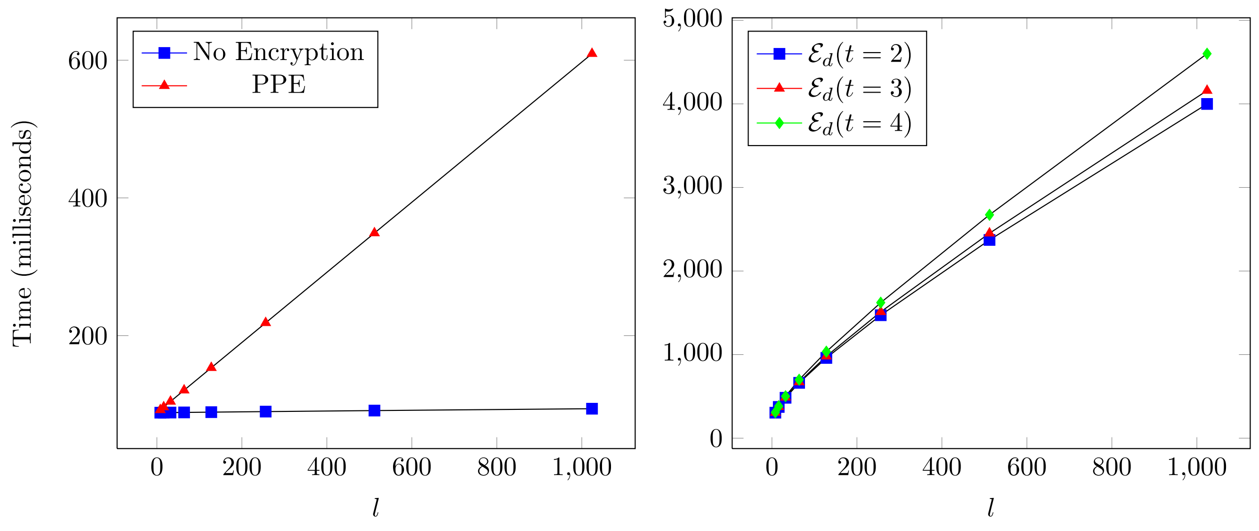

Figure 2: Encryption Cost Comparisons for Different Protocols.

l	"No Encryption"	"PPE"	$\mathcal{E}_d(t = 2)$	$\mathcal{E}_d(t = 3)$	$\mathcal{E}_d(t = 4)$
8	88.11	92.14	305.01	306.27	309.71
16	88.16	96.22	373.55	376.05	382.94
32	88.27	104.38	483.21	488.21	502.00
64	88.47	120.70	661.87	671.88	699.46
128	88.88	153.33	960.05	980.07	1035.24
256	89.67	218.56	1471.30	1511.34	1621.67
512	91.16	348.95	2371.91	2451.98	2672.65
1024	93.86	609.45	3999.07	4159.23	4600.57

Table 1: Encryption Cost (in milliseconds) Comparisons for Different Protocols.

As shown in Figure 2 and Table 1, the encryption cost of "No Encryption" $<$ the encryption cost of "PPE" $<$ the encryption cost of $\mathcal{E}_d$ for $t = 2 <$ the encryption cost of $\mathcal{E}_d$ for $t = 3 <$ the encryption cost of $\mathcal{E}_d$ for $t = 4$. The encryption costs of all protocols increase when l increases because the length of the critical data increases. "No Encryption" requires approximately 90 millisecond, and its encryption time increases slowly when the length of the critical data increases because it does not incur encryption overhead but only incurs communication overhead. "PPE" requires 92 milliseconds to 609 milliseconds when l increases from 8 to 1024. The encryption costs of PPE and $\mathcal{E}_d$ grow linearly with the increase of the critical data size. Relatively, $\mathcal{E}_d$ incurs a much higher encryption cost than pure "PPE", from 3 folds when data size is 8 bits to 6 folds when data size becomes 1024. This is because it also incurs the sharing and reconstructing cost as well as a higher communication cost due to the use of intermediate key agents. However, a multi-user PPE protocol is essential and the cost is bearable. When t increases, the computation cost of secret sharing increases and, hence, the encryption cost of $\mathcal{E}_d$ increases, but the increase is relatively slow. Thus, using additional key agents to enhance security can be a feasible method.

6 Conclusion

Existing cryptographic security proofs for PPE schemes are based on the indistinguishability of the real PPE scheme and the ideal PPE object. This is insufficient because the security of the ideal PPE object is unknown. We developed the first complete security proof for PPE by qualifying the security of the ideal PPE object. We created a new security notion, IND-PCPA, and proved that the ideal PPE object is secure under IND-PCPA and can at the best reach IND-PCPA security.

We also built a protocol and extended an existing PPE scheme to support multi-user systems, in which users do not know the master encryption key (in fact, no single entity in the system knows the master encryption key) for encrypting and decrypting the data to be sent to and received from the server, respectively. We solve the challenge and designed a distributed PPE encryption scheme for the multi-user protocol. The correctness and security of the multi-user protocol have been proven rigorously. The performance of the protocol is studied experimentally to illustrate its feasibility.

References

[1] G. Amanatidis, A. Boldyreva and A. O'Neill, *Provably-secure schemes for basic query support in outsourced databases*, *Working Conference on Data and Applications Security*, 2007, pp. 14-30.

[2] M. Bellare, T. Kohno, and C. Namprempre, *Authenticated encryption in SSH: provably fixing the SSH binary packet protocol*, *Proceedings of the 9th ACM conference on Computer and Communications Security (CCS-02)*, 2002, pp. 1-11.

[3] A. Boldyreva, N. Chenette, Y. Lee, A. O'Neill, *Order-preserving symmetric encryption*, *Eurocrypt* 2009, pp. 224-241.

[4] R. Cramer, I. Damgard, J. Nielsen, *Multiparty Computation, an Introduction*, 2009, http://cs.au.dk/ jbn/smc.pdf.

[5] Y. Desmedt, *Society and group oriented cryptography: an new concept*, *Advances in Cryptography - CRYPTO '87*, 1987, Springer-Verlag LNCS 293, pp. 120-127.

[6] Y. Desmedt and Y. Frankel, *Threshold Crypto-Systems*, *Advances in Cryptography - CRYPTO '89*, 1989, Springer-Verlag LNCS 435, pp. 307-315.

[7] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, *Robust and efficient sharing of RSA functions*, *Advances in Cryptology - CRYPTO '96*, 1996, Springer-Verlag LNCS 1109, pp. 157-172.

[8] P. Gemmell, *An introduction to threshold cryptography*, *Cryptobytes*, 1997, pp. 7-12.

[9] J. Katz, Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, *Chapman & Hall/CRC*, 2007.

[10] J. Li, E. R. Omiecinski, *Efficiency and security trade-off in supporting range queries on encrypted databases*, *Data and Applications Security* 2005, pp. 69-83.

[11] M. Naor, B. Pinkas, O. Reingold, *Distributed Pseudo-Random Functions and KDCs*, *Advances in Cryptology EUROCRYPT'99* 1999, pp. 327-346.

[12] T. Pederson, *A threshold crypto-system without a trusted dealer*, *Advances in Cryptology - EUROCRYPT '91*, 1991, Springer-Verlag LNCS 547, 522-526.

[13] A. Shamir, *How to share a secret*, *Communications of the ACM* 1979, 22, pp. 612-613.

[14] J. Xu, J. Fan, M.H. Ammar, and S.B. Moon, *Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme*, *IEEE International Conference on Network Protocols*, pp. 280-289, 2002.