Semantic Secure Public Key Encryption with Filtered Equality Test *PKE-FET*

Kaibin Huang[1], Yu-Chi Chen[2] and Raylin Tso[1]
[1]*Department of Computer Science, National Chengchi University, Taipei, Taiwan*
[2]*Institute of Computer Science, Academic Sinica, Taipei, Taiwan*

Keywords: Cloud Storage, Equality Test, Filtered Equality Test, Public Key Encryption, Secret Sharing, Semantic Security.

Abstract: Cloud storage allows users to outsource their data to a storage server. For general security and privacy concerns, users prefer storing encrypted data to pure ones so that servers do not learn anything about privacy. However, there is a natural issue that servers have worked some analyses (i.e. statistics) or routines for encrypted data without losing privacy. In this paper, we address the basic functionality, equality test, over encrypted data, which at least can be applied to specific analyses like private information retrieval. We introduce a new system, called *filtered equality test*, which is an additional functionality for existing public key encryption schemes. It satisfies the following scenario: a ciphertext-receiver selects several messages as a set and produces its related warrant; then, on receiving this warrant, an user is able to perform equality test on the receiver's ciphertext without decryption when the hidden message belongs to that message set. Similar to the attribute based encryption, ABE. In ABE schemes, those ones who match the settled conditions could get the privilege of decryption. In FET schemes, those 'messages inside selected set' can be equality tested. Combining PKE schemes and filtered equality test, we propose a framework of *public key encryption scheme with filtered equality test*, abbreviated as PKE-FET. Then, taking ElGamal for example, we propose a concrete PKE-FET scheme based on secret sharing and bilinear map. Finally, we prove our proposition with semantic security in the standard model.

1 INTRODUCTION

With the development of cloud computing, users can carry on devices with weak computational abilities instead of powerful ones, since computational resource and power come from the cloud server. In particular, cloud storage (an application of cloud services) provides space which users can store data on cloud and retrieve their data when they need. For example, Dropbox, Google Drive, and iCloud, are well-known cloud storage services, but these systems only allow users to access their own data. In this case, users must trust the server without doubt. Straightly, there is a natural privacy issue – the server can see data in the clear. For privacy, data encryption is employed to overcome this issue. Finally, servers receive large amount of encrypted data everyday, but they cannot do any statistical analysis because data is encrypted.

Complicated data analysis may not be realized for encrypted data, but it is plausible to obtain data equality which is the easiest analysis to check whether two encrypted data are the same or not. For this purpose, encrypted data with equality test extracts significant attention in cloud storage applications. In the original equality testable setting like (Peng et al., 2005), it allows the storage server to play the tester role who has the ability to work equality test on two ciphertexts of the same receiver. To make PKE-ET more flexible, Yang et al. add the multi-user setting (Bellare et al., 2000)(Fouque et al., 2014) to propose the public key encryption with equality test, PKE-ET (Yang et al., 2010), which the tester can check ciphertexts of different receivers. However, PKE-ET (both in the multi-user setting or single-user setting) has a drawback that all users including adversaries can play the tester role to arbitrarily test ciphertexts, since this is the main goal of PKE-ET. Public key encryption with authenticated equality test, PKE-AET, (Huang et al., 2015) is presented to overcome above issue by limiting the availability of being a tester. In PKE-AET, an user becomes a tester after obtaining the warrant from the corresponding ciphertext-receiver. The war-

Huang K., Chen Y. and Tso R..
Semantic Secure Public Key Encryption with Filtered Equality Test - PKE-FET.
DOI: 10.5220/0005550303270334
In *Proceedings of the 12th International Conference on Security and Cryptography* (SECRYPT-2015), pages 327-334
ISBN: 978-989-758-117-5

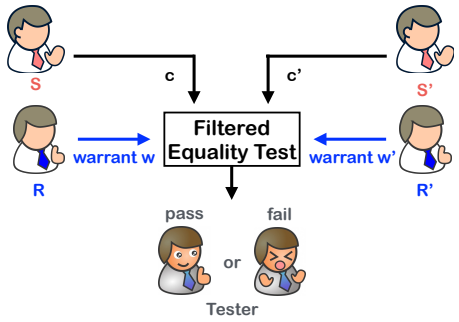

Figure 1: Public key encryptions with filtered equality test.

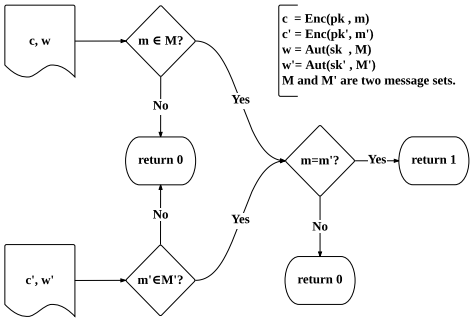

Figure 2: Flow chart of filtered equality test.

rant is referred to an authority that allows the tester to perform equality test.

1.1 Related Works

In the literature, the notion of PKE-ET was first introduced by Yang et al. (Yang et al., 2010). Based on bilinear map, they constructed the framework in which everyone is able to perform equality test on two ciphertexts encrypted under different public keys. Generally speaking, it is impossible to achieve standard semantic security or IND-CPA security due to equality testability. However, Tang partially fixed the above security issue by adding the authority of equality testability (Tang, 2012b)(Tang, 2012a). The receiver gives warrants to trusted testers, and these testers can perform equality test on ciphertexts of this receiver. Therefore, Tang's method provides IND-CCA security against outsider attacker (not the tester) and one-way security against the testers. Tang defined Type-I/II adversary as the party with/without the warrant. Later, Ma et al. propose an efficient public key encryption with delegated equality test in a multi-user setting, PKE-DET (Ma et al., 2014). Both of Tang and Ma et al.'s schemes make testers able to perform equality test on *all* ciphertexts of the receiver who gives the warrant. Recently, Huang et al. introduced a new PKE-ET with ciphertext-binded authorities (Huang et al., 2014) which makes testers only able to work equality test on a *specific* ciphertext. The receiver takes ciphertexts as input to generate warrants, and the functionality of warrants is rigorously restricted.

1.2 Contributions

In this paper, we present a new notion called *filtered equality test*, it provides additional functionality to existing public key encryption schemes: in a selected message set, it makes equality tests work. There are three entities, sender, equality tester (server), and receiver. The sender uses the receiver's public key to produce encrypted data, and delivers them to the tester (server). The receiver designates a set of messages to generate its warrant, and delivers to the tester. Once the tester holds warrants and ciphertexts, it does the check in Figure 1 without decryption; the detail data flow is described in Figure 2. To help understanding, we take attribute-based encryption (ABE) schemes for example. In ABE schemes, those quantified users who match settled conditions can decrypt the ciphertexts. In FET schemes, those quantified 'messages' can be equality tested.

We construct the framework of PKE-FET , and its security model. Taking ElGamal (Gamal, 1985) for instance, we propose an efficient PKE-FET scheme based on the properties of secret sharing and bilinear map, and then prove the its semantic security in the standard model.

The rest of paper is organized as follows. In Section 2, some preliminaries are briefly introduced, i.e. bilinear map, secret sharing, and hardness assumptions. In Section 3, we show the notion of PKE-FET in details. We present our PKE-FET scheme in Section 4 and its security proof in Section 5. Finally, the conclusions of this paper are given in Section 6.

2 PRELIMINARIES

Now we will introduce some preliminaries, including the bilinear map, a well-defined primitive, and then review the concept of secret sharing. Moreover, we attach some hardness assumptions which will be used to analyze the security of the proposed scheme.

2.1 Bilinear Map (A.K.A Pairing)

The bilinear map (Chatterjee and Menezes, 2011) works as follows. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups with the same prime order q. Define the mapping as a function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow$

$\mathbb{G}_T$. If $\mathbb{G}_1 = \mathbb{G}_2$, it is called Type-I pairing in (Galbraith et al., 2008). Otherwise, it is called asymmetric pairing while $\mathbb{G}_1 \neq \mathbb{G}_2$. In asymmetric pairing setting, if there is an efficiently-computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, it is called Type-II pairing. However, if it is in asymmetric setting without an efficiently-computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, it is called Type-III pairing.

Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generators respectively. $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ holds the following properties.

1. Bilinear: for all $x, y \in \mathbb{Z}_q^*$, we have
$$e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$$
2. Non-degenerate: let I be the identity of group $\mathbb{G}_T$; for all generators g_1 and g_2, we have
$$e(g_1, g_2) \neq I$$
3. Computable: $e(g_1, g_2)$ can be computed in polynomial run-time.

2.2 Secret Sharing

The notion of secret sharing (Shamir, 1979) is introduced to share a secret data $\mathcal{D}$ to n users. In secret sharing, sufficient k ($1 \leq k \leq n$) users can reconstruct the secret $\mathcal{D}$ on receiving their sharing fragments; on the other hand, any less than $k-1$ pieces reveal no information about $\mathcal{D}$. In the following, there is a simple k-out-of-n secret sharing scheme provided in (Shamir, 1979).

For sharing a secret $\mathcal{D}$ to n users, a dealer (trusted party who holds the secret) first picks $k-1$ random numbers, r_1 to r_{k-1}, to form k points on a 2-dimensional plane, which are $\{(0, D), (1, r_1), ..., (k-1, r_{k-1})\}$. Then, along with these points, there should be one and only one polynomial function ψ with $k-1$ degree determined. Next, he computes following points $(i, \psi(i))$ for user $i \in [k, n]$. Now there are total n points which all of them satisfy $y = \psi(x)$. By distributing these points, it formalizes a k-out-of-n secret sharing scheme. As a result, k users are able to rebuild the polynomial function ψ by linear combination, and then compute $\mathcal{D} = \psi(0)$ to acquire the secret. However, if there are less than k users, they cannot recover ψ so that the secret $\mathcal{D}$ is perfectly protected. Hence, secret sharing is perfect secure.

2.3 Hardness Assumptions

In this section, some hardness assumptions will be introduced since we will use them to argue the security of the proposed PKE-FET scheme. In general, we assume the probability of breaking these assumptions is negligible. We describe the universal one-way hash function (Diffie and Hellman, 1976)(Naor and Yung, 1989) as follows.

Universal One-way Hash Function. A function F is one-way if a random input x, given $F(x)$, it is hard to compute x. In other words, a one-way hash function F is easy to find $F(x)$ given an input x, but it is computational difficult to extract x from $F(x)$. Based on one-way hash function, an universal one-way hash function is proposed by Naor and Yung (Naor and Yung, 1989). It address on the problem that find x, y in the domain such that $y \neq x$ and $F(x) = F(y)$. Let $\mathcal{A}$ be an polynomial-time adversary to solve the one-way hash function. We define $\mathcal{A}$'s advantage as $Adv^{OW}_{\mathcal{A},F} = \Pr[x \leftarrow \mathcal{A}(F(x))]$.

Besides, there are a series of computationally hardness assumptions.

- **Discrete Logarithm Problem (DLP).** given $g, y \in \mathbb{G}$ it is hard to output an integer x such that $y = g^x$ where $\mathbb{G}$ can be $\mathbb{G}_1$ or $\mathbb{G}_2$. Let $\mathcal{A}$ be an polynomial-time adversary to solve DLP. We define $\mathcal{A}$'s advantage as $Adv^{DLP}_{\mathcal{A},\mathbb{G}} = \Pr[x \leftarrow \mathcal{A}(g, g^x)]$.
- **Computational Diffie-Hellman (CDH) Problem.** given $g^x \in \mathbb{G}$ and $g^y \in \mathbb{G}$ it is hard to output g^{xy}. Let $\mathcal{A}$ be an polynomial-time adversary to solve the CDH problem. We define $\mathcal{A}$'s advantage as $Adv^{CDH}_{\mathcal{A},\mathbb{G}} = \Pr[g^{xy} \leftarrow \mathcal{A}(g^x, g^y)]$. According to (Sakurai and Shizuya, 1995), the only known solution to solve CDH problem is to solve DL problem.
- **Decisional Diffie-Hellman (DDH) Problem.** given $g^x \in \mathbb{G}$, $g^y \in \mathbb{G}$, and $Z \in \mathbb{G}$, it is hard to decide whether $Z = g^{xy}$ or not. Let $c \in \{0, 1\}$ be a fair coin, both $c = 1$ and $c = 0$ appear with half probability; if $c = 1$, $Z = g^{xy}$; otherwise, Z is a random number. Let $\mathcal{A}$ be an polynomial-time adversary to break the DDH problem. We define $\mathcal{A}$'s advantage as $Adv^{DDH}_{\mathcal{A},\mathbb{G}} = \Pr[c^* \leftarrow \mathcal{A}(g^x, g^y, Z) : c^* = c] - \frac{1}{2}$.
- **Symmetric External Diffie-Hellman (SXDH).** By the definition of (Ghadafi et al., 2010), the DDH problem in $\mathbb{G}_1$ or $\mathbb{G}_2$ in the type-III pairing environment is called SXDH problem; and SXDH problem is as hard as DDH problem.

3 MODELS OF PKE-FET

PKE-FET is composed of a PKE scheme and FET functionality, which FET denotes the following scenario: the receiver picks n messages (denoted as a set $\mathcal{M} = \{m_1, \cdots m_n\}$) from message space $\mathbb{M}$; then she

generates a warrant w using these n messages and her private key. Following, she can delegate the equality testability to someone by delivering the warrant w. When one user gets the warrant, he can run equality tests on the receiver's ciphertexts if the messages inside the ciphertexts belong $\mathcal{M}$. For security issues, we have to limit $n << |\mathbb{M}|$. The formal description of PKE-FET is listed below:

3.1 Framework

Let PKE-FET be a public key encryption with filtered equality test. Formally, PKE-FET is composed of following polynomial-time algorithms:

- Setup: on input a secure parameter λ, it generates a series of public parameters pp.
- Key generation: on input the public key pp, it returns the receiver's key pair (sk, pk).
- Encryption: on input a public key pk and a message m, the encryption algorithm, run by the sender, generates a probabilistic ciphertext $c = Enc(pk, m)$.
- Decryption: on input a ciphertext c and the secret key sk, the decryption algorithm, run by the receiver, outputs the message m hidden in the ciphertext.
- Authorization: on input the secret key sk and a set of n messages, $m_1, ..., m_n \in \mathcal{M}$, it generates a warrant w for the message set $\mathcal{M}$. Let $w = Aut(sk, \mathcal{M})$. The receiver will give the warrant w to a trusted tester, and thus the tester is able to perform the filtered equality test on those ciphertext encrypted under the receiver's public key.
- Filtered equality test: (see Figure 2) on input two ciphertexts $c = Enc(pk, m)$ and $c' = Enc(pk', m')$ and two warrants $w = Aut(sk, \mathcal{M})$ and $w' = Aut(sk', \mathcal{M}')$, this algorithm returns $1 \leftarrow FET(c, c', w, w')$ if and only if all of the following three conditions hold:

  $$m \in \mathcal{M}, m' \in \mathcal{M}', \text{and } m = m'$$

  Otherwise, it returns 0.

3.2 Properties of PKE-FET

Define $\varepsilon(\lambda)$ as a negligible probability based on the secure parameter λ. Referring to (Yang et al., 2010)(Huang et al., 2014)(Huang et al., 2015), a PKE-FET scheme is considered to be valid if and only if it satisfies correctness, perfect consistency, and computational soundness.

- *Correctness*: PKE-FET is a public key encryption scheme in which the receiver can recover the correct message m. That is, for all m,

  $$\Pr[Dec(sk, Enc(pk, m)) = m] = 1$$

- *Perfect consistency*: for three true conditions, $m \in \mathcal{M}$, $m' \in \mathcal{M}'$, and $m = m'$, $FET(c, c', w, w')$ must return 1.
- *Computational soundness*: with at least one false condition, the probability $\Pr[1 \leftarrow FET(c, c', w, w')]$ is bounded at most $\varepsilon(\lambda)$.

3.3 Semantic Security

To discuss about security issues of PKE-FET , first we recall the definition of semantic security, or so called indistinguishability (IND) style of security, which composed of an adversary $\mathcal{A}$ and a challenger.

On input a public key pk selected by the challenger, the probabilistic polynomial time (PPT) adversary $\mathcal{A}$ is allowed to access decryption oracle $\mathcal{D}_1$ for polynomial times, then it outputs two messages m_0 and m_1. The challenger randomly picks one of them (m_b, $b \in \{0, 1\}$) and encrypts it as a challenge $c_b = Enc(pk, m_b)$. $\mathcal{A}$ is still allowed to access decryption oracle $\mathcal{D}_2$, then it has to output a guess of b. If it successfully guesses with a non-negligible probability in advance, then $\mathcal{A}$ breaks this game; otherwise, this encryption scheme achieves semantic security, or called indistinguishability styles of security.

Decryption oracle $\mathcal{D}_1$ and $\mathcal{D}_2$ differ from different attack models, where: in the chosen-plaintext attack (CPA) model, neither $\mathcal{D}_1$ nor $\mathcal{D}_2$ works; in the chosen-ciphertext attack (CCA) model, $\mathcal{D}_1$ works, but $\mathcal{D}_2$ does not; and in the adaptive chosen-ciphertext attack (CCA2) model, they both work, but the challenge $c_b = Enc(pk, m_b)$ is forbidden to be requested to $\mathcal{D}_2$. Along with the IND game, semantic secure is classified into IND-CPA, IND-CCA and IND-CCA2 secure.

3.4 Semantic Security for PKE-FET Schemes

Extending from above section, we define semantic security for PKE-FET schemes. In the beginning, the challenger generates a legal key pair (sk, pk) and a warrant $w \leftarrow Aut(sk, \mathcal{M})$ which $\mathcal{M}$ is randomly selected from message space $\mathbb{M}$; and meanwhile, $\mathcal{M}$ is unknown to $\mathcal{A}$. Then, pk and w are delivered to $\mathcal{A}$ to start a semantic secure game. With the aid of decryption oracle $\mathcal{D}_1$, $\mathcal{A}$ outputs his two messages

m_0 and m_1 selected in $\mathbb{M}$. The challenger picks one of them (m_b, $b \in \{0,1\}$) and encrypts it into a challenge $c_b = Enc(pk, m_b)$. Along with the help of decryption oracle $\mathcal{D}_2$, $\mathcal{A}$ outputs a guess b^* to terminate this game. Let $Adv^{IND}_{\mathcal{A},FET}$ be the probability of $\mathcal{A}$ wins the game in advance (more than universally guessing). If $Adv^{IND}_{\mathcal{A},FET}$ is non-negligible, then we say $\mathcal{A}$ breaks this game; otherwise, we say a PKE-FET scheme is semantic secure. We say a PKE-FET scheme is semantic secure if $Adv^{IND}_{\mathcal{A},FET}$ is negligible, where it is defined as follows:

$$\Pr\left[\begin{array}{l}(sk,pk) \leftarrow KeyGen(\lambda); \mathcal{M} \leftarrow_R \mathbb{M};\\ w \leftarrow Aut(sk, \mathcal{M});\\ (m_0,m_1) \leftarrow \mathcal{A}^{\mathcal{D}_1}(pk,w);\\ b \leftarrow_R \{0,1\}; c_b \leftarrow Enc(pk,m_b);\\ b^* \leftarrow \mathcal{A}^{\mathcal{D}_2}(pk,w,c_b) : b^* = b\end{array}\right] - \frac{1}{2} \leq \varepsilon(\lambda)$$

4 PROPOSED PKE-FET SCHEME

In this section, we will describe the proposed PKE-FET scheme from secret sharing to realize filtered equality test over encrypted data. Taking ElGamal as a building block, we construct our PKE-FET scheme below.

4.1 Construction

- Setup: Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a Type-III bilinear map operation which are mentioned in Section 2.1. Then, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three multiplicative cyclic group with the same prime order q. Randomly choose $g \in_R \mathbb{G}_1$ and $g_2 \in_R \mathbb{G}_2$ as two generators respectively. The message space $\mathbb{M}$ is set to be a subgroup of $\mathbb{G}_1$; i.e., $\mathbb{M} \subseteq \mathbb{G}_1$. $H_1 : \mathbb{M} \rightarrow \mathbb{G}_T$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ are two universal one-way hash functions. Here, the auxiliary information $n = n(\lambda)$ is needed, and we emphasize that $n << |\mathbb{M}|$. The public parameter is composed of $pp = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g, g_2, H_1, H_2, n\}$.
- Key generation: Sample $(u, v, s_0, s_1, ..., s_n) \leftarrow_R \mathbb{Z}_q^*$ as sk. Compute $U = g^u$, $V = e(g, g_2)^{uv}$ and $S_i = g^{s_i}$ for all $i \in [0,n]$; then publish $pk = (U, V, S_0, S_1, ...S_n)$.
- Encryption: Taking a message $m \in \mathbb{M}$, the sender first randomly selects $r \leftarrow_R \mathbb{Z}_q^*$, then computes $h = H_2(m)$ and $c = (A, B, C, D)$ where

$$A = g^r, B = m \cdot U^r, C = V^r \cdot H_1(m)$$

$$D = ((S_0)^r, (S_1)^{rh}, (S_2)^{rh^2}, ..., (S_n)^{rh^n}) \quad (1)$$

- Decryption: The receiver computes $m = B/A^u$. Let $D = (D_0, D_1, ..., D_n)$ and $h = H_2(m)$, he verifies both $C = e(A, g_2)^{uv} \cdot H_1(m)$ and $D_i = A^{s_i h^i}$ for all $i \in [0,n]$. If they both hold, it returns m; otherwise, it returns $\perp$ as decryption failed.
- Authorization: on input $\mathcal{M} = \{m_1, ..., m_n\}$ and the secret key $sk = (u, v, s_0, s_1, ..., s_n)$, the authorization algorithm computes a n-degree polynomial function $f(x)$ following:

$$f(x) = \prod_{i=1}^{n}(x - H_2(m_i)) + uv = \sum_{i=0}^{n} a_i x^i. \quad (2)$$

Furthermore, the receiver computes $w_i = g_2^{a_i/s_i}$ for all $i \in [0,n]$; and then, he destroys $(a_0, ..., a_n)$ and sends the warrant $w = (w_0, w_1, ..., w_n)$.

Note that the message set $\mathcal{M}$ is hidden in the generated warrant, which is unknown for all users even he or she gets the warrant.

- Filtered equality test: upon receiving two ciphertexts $c = Enc(pk, m)$ and $c' = Enc(pk', m')$ and two warrants $w = Aut(sk, \mathcal{M})$ and $w' = Aut(sk', \mathcal{M}')$, the filtered equality test algorithm works the following steps.
  1. Parse $c = (A, B, C, D)$, $D = (D_0, D_1, ..., D_n)$ and $w = (w_0, w_1, ..., w_n)$.
  2. Compute

$$z = C / \prod_{i=0}^{n} e(D_i, w_i) \quad (3)$$

  3. Compute z' from (c', w') following Steps 1 and 2.
  4. Check whether $z = z'$ or not. If $z = z'$, then it returns 1, which means $m \in \mathcal{M}$, $m' \in \mathcal{M}'$, and $m = m'$. If not, it returns 0 instead.

4.2 Analysis

We now verify our scheme to satisfy three significant properties mentioned in Section 3.2.

- *Correctness*: The decryption algorithm computes

$$B/A^u = mg^{ur}/g^{ur} = m$$

Then, let $h = H_2(m)$, it checks both

$$\begin{aligned} e(A, g_2)^{uv} \cdot H_1(m) &= e(g^r, g_2)^{uv} \cdot H_1(m) \\ &= e(g, g_2)^{uvr} \cdot H_1(m) \\ &= V^r \cdot H_1(m) = C \end{aligned}$$

and

$$\forall i \in [1,n], D_i = (S_i)^{rh^i} = g^{s_i r h^i} = A^{s_i h^i}$$

It is straightforward that the correctness holds along with the decryption algorithm.

Table 1: Comparison with previous works.

		(Yang et al., 2010)	(Ma et al., 2014)	(Huang et al., 2014)	Proposed
Efficiency	KeyGen	$O(1)$	$O(1)$	$O(1)$	$O(n)$
	Enc	$O(1)$	$O(1)$	$O(1)$	$O(n)$
	Dec	$O(1)$	$O(1)$	$O(1)$	$O(n)$
	Aut	-	$O(1)$	$O(1)$	$O(n)$
	Test	$O(1)$	$O(1)$	$O(1)$	$O(n)$
Storage	Key	$O(1)$	$O(1)$	$O(1)$	$O(n)$
	Cipher	$O(1)$	$O(1)$	$O(1)$	$O(n)$
	Warrant	-	$O(1)$	$O(1)$	$O(n)$
Testable	messages	$\mathbb{M}$	$\mathbb{M}$	1	n
Security	With Aut	-	OW-CCA	OW-CCA	IND-CCA
	W/O Aut	OW-CCA	IND-CCA	IND-CCA	IND-CCA

- *Perfect consistency*: On input (c,w) and (c',w'), the filtered equality test algorithm obtains z by computing equation (3).

$$\begin{aligned} z &= C/\Pi_{i=0}^{n} e(D_i, w_i) \\ &= C/\Pi_{i=0}^{n} e(g^{s_i r H_2(m)^i}, g_2^{a_i/s_i}) \\ &= C/\Pi_{i=0}^{n} e(g,g_2)^{r a_i H_2(m)^i} \\ &= C/e(g,g_2)^{r\Sigma_{i=0}^{n} a_i H_2(m)^i} \\ &= C/e(g,g_2)^{r f(H_2(m))} \end{aligned}$$

If $m \notin \mathcal{M}$, z will be a random number locates statistically random in $\mathbb{G}_T$ because $f(H_2(m))$ is statistically random in $\mathbb{Z}_q^*$. Otherwise, in case that m is in the message set $\mathcal{M}$, then we have

$$f(H_2(m)) = \sum_{i=0}^{n} a_i H_2(m)^i = uv \qquad (4)$$

Therefore, equation (3) will be

$$\begin{aligned} z &= C/e(g,g_2)^{uvr} \\ &= V^r \cdot H_1(m)/V^r = H_1(m) \end{aligned}$$

Analogously, it returns $z' = H_1(m')$ if $m' \in \mathcal{M}'$ through the same computations. Finally, it verifies whether $z = z'$ or not. It returns 1 if and only if the $m \in \mathcal{M}$, $m' \in \mathcal{M}'$ and $m = m'$, and therefore the perfect consistency holds.

- *Computational soundness*: We consider the following two conditions.
  1. $m \in \mathcal{M}$ and $m' \in \mathcal{M}'$: By the inference of consistency, z and z' will be computed as $z = H_1(m)$ and $z' = H_1(m')$ respectively. We can reduce computational soundness to the universal property of hash function H_1. If collision happens with negligible probability, then the probability of $1 \leftarrow FET(c,c',w,w')$ in this condition is also negligible.
  2. $m \notin \mathcal{M}$ or $m' \notin \mathcal{M}'$: At least one of z and z' will be close to a uniformly random number in $\mathbb{G}_T$, so the probability of $1 \leftarrow FET(c,c',w,w')$ in this condition is definitely $1/q$ (negligible).

  The computational soundness holds due to the negligible probability.

We compare PKE-FET to some existing protocols and record them on Table 1. It is clear that PKE-FET performs not good on considering the efficiency and storage space. Nevertheless, it provides the 'filtered equality test' functionality; and meanwhile, it is semantic secure against those adversaries who own warrants. The security analysis will be discussed in the next section.

5 SECURITY PROOF

In this section, we are going to prove the security of our PKE-FET scheme through a series of hybrid games in the standard model. We claim that if there is a polynomial time adversary can break our PKE-FET with non-negligible probability, then the challenger is able to take advantage of $\mathcal{A}$ to break SXDH problem with non-negligible probability. First, we define game 0 which is equal to our PKE-FET scheme, and then it will be inferred step by step.

Game 0:

The challenger works key generation and authorization algorithms to generate a key pair $(sk,pk) = KeyGen(\lambda)$ and a warrant $w = Aut(sk,\mathcal{M})$, which $\mathcal{M}$ is randomly selected from message space $\mathbb{M}$. Then, he delivers pk and w to the adversary $\mathcal{A}$. With the aid of decryption oracle $\mathcal{D}_1$, $\mathcal{A}$ outputs m_0 and m_1 after polynomial times of decryption requests. Then, the challenger picks one of two messages (m_b, $b \in \{0,1\}$), encrypts it into $c_b = Enc(pk,m_b)$, and then returns c_b to $\mathcal{A}$. With polynomial times requests to decryption oracle $\mathcal{D}_2$, $\mathcal{A}$ finally outputs a guess b^*. Decryption oracles $\mathcal{D}_1$ and $\mathcal{D}_2$ are defined as follows: on receiving a decryption query $Dec(sk,c)$, the chal-

lenger returns $m = Dec(sk, c)$ to $\mathcal{A}$. Only c_b is forbidden to be requested. Let $\Pr[game_0]$ be the probability to break the semantic security in game 0, it is oblivious that $Adv^{IND}_{\mathcal{A},FET} = \Pr[game_0] - \frac{1}{2}$.

Game 1:

All settings in game 1 is identical to those in game 0 except for the following condition. If $m_0 \in \mathcal{M}$ or $m_1 \in \mathcal{M}$, the challenger terminates and claims fail; otherwise, game 1 works as game 0 does. According to difference lemma (Shoup, 2004), we have $\Pr[game_0] \leq \Pr[game_1] + \frac{2n}{|\mathbb{M}|}$; where $\frac{2n}{|\mathbb{M}|}$ is negligible in $\Pr[game_1]$ since $n << |\mathbb{M}|$.

Game 2:

Extends from game 1, we import the SXDH problem defined in section 2 to help proof. The challenger runs $(sk, pk) = KeyGen(\lambda)$ and $w = Aut(sk, \mathcal{M})$ ($\mathcal{M}$ is randomly picked from $\mathbb{M}$). Following, he replaces $S_0 = g^y$ and

$$\begin{aligned} V &= e(g, g_2)^{uv - a_0} \cdot e(g^y, g_2)^{a_0/s_0} \\ &= e(g, g_2)^{uv - a_0 + a_0 y/s_0} \end{aligned}$$

Then, $pk = (U, V, S_0, S_1, ..., S_n)$ and w are sent to $\mathcal{A}$. After polynomial times of decryption queries to $\mathcal{D}_1$, $\mathcal{A}$ returns two messages m_0 and m_1. After that, the challenger randomly picks message m_b, $b \in_R \{0,1\}$, then he computes $h = H_2(m_b)$ and $c_b = (A, B, C, D)$, where

$$A = g^x, C = (e(g^x, g_2)^{uv - a_0} \cdot e(Z, g_2)^{a_0/s_0}) \oplus H_1(m_b),$$

$$B = m_b g^{ux}, D = (Z, g^{xs_1 h}, g^{xs_2 h^2}, ... g^{xs_n h^n})$$

After receiving c_b and polynomial times of decryption queries to $\mathcal{D}_2$, $\mathcal{A}$ outputs a guess b^*. If $b^* = b$, the challenger guesses $c = 1$; otherwise, he guesses $c = 0$.

Theorem 1. *Game 2 is indistinguishable from game 1.*

Proof. There are two major differences between game 1 and game 2: the first one comes from the substitution of S_0 and V; and the second one is Z in part C or D of challenge c_b. S_0 is originally g^{s_0}, and it becomes g^y. It is oblivious that S_0 is still in the correct form, but y becomes unknown to the challenger; moreover, since secret key v is unknown in $\mathcal{A}$'s viewpoint, any V belongs to $\mathbb{G}_T$ is considered as a regular parameter. On the hand, both two substitutions of Z in c_b are distinguishable because of the intractable of SXDH problem. The probability that $\mathcal{A}$ can distinguish game 2 from game 1 is estimated as $Adv^{SXDH}_{\mathcal{A},\mathbb{G}_1}$.

Theorem 2. $\Pr[game_2]$ *can be polynomially reduced to the SXDH problem.*

Proof. Let $\mathcal{A}$ has a non-negligible probability ε in advance to break PKE-FET scheme (total $\frac{1}{2} + \varepsilon$ probability). We discuss game 2 on considering whether $Z = g^{xy}$ or not. The first case, when $c = 1$, scenario in game 2 is actually a PKE-FET scheme so that $\mathcal{A}$ has $\frac{1}{2} + \varepsilon$ probability to break PKE-FET scheme. The second case, $c = 0$, scenario in game 2 is different from PKE-FET scheme; the probability that $\mathcal{A}$ break game 2 is estimated as $\frac{1}{2}$. After plenty of games, considering on those games that $\mathcal{A}$ has won, first case must be more than second case; and their rate will be $(1/2 + \varepsilon) : 1/2$. It means when $\mathcal{A}$ wins the game, the challenger has more probability on answering $Z = g^{xy}$. The accurate advanced probability of breaking SXDH problem is estimated as:

$$Adv^{SXDH}_{\mathcal{A},\mathbb{G}_1} = \frac{1/2 + \varepsilon}{1/2 + \varepsilon + 1/2} - \frac{1}{2} = \frac{\varepsilon}{2 + 2\varepsilon}$$

In other words, if $\mathcal{A}$ has non-negligible advanced probability ε to break PKE-FET scheme; then, the challenger can take advantage of $\mathcal{A}$ to break SXDH problem with non-negligible advanced probability $\frac{\varepsilon}{2+2\varepsilon}$, which is a little smaller than $\frac{\varepsilon}{2}$, but it is still non-negligible. Therefore, we can say $\Pr[game_2]$ is close to but a little bigger than $2Adv^{SXDH}_{\mathcal{A},\mathbb{G}_1} + \frac{1}{2}$. Then, $Adv^{IND}_{\mathcal{A},FET}$ is inferred as follows:

$$\begin{aligned} Adv^{IND}_{\mathcal{A},FET} &= \Pr[game_0] - \frac{1}{2} \\ &\leq \Pr[game_1] + \frac{2n}{|\mathbb{M}|} - \frac{1}{2} \\ &\leq \Pr[game_2] + Adv^{SXDH}_{\mathcal{A},\mathbb{G}_1} + \frac{2n}{|\mathbb{M}|} - \frac{1}{2} \end{aligned}$$

Combining $\Pr[game_2]$, $Adv^{IND}_{\mathcal{A},FET}$ is estimated bounded by a real number d, which $d \approx 3Adv^{SXDH}_{\mathcal{A},\mathbb{G}_1} + \frac{2n}{|\mathbb{M}|}$ is negligible in λ. We say PKE-FET is semantic secure or IND-CCA2 secure based on the intractability of SXDH problem.

6 CONCLUSIONS

Computations over ciphertext has extracted research attention. In this paper, a new notion of filtered equality test has been presented. We show a framework and some security requirements of filtered equality test as an additional functionality to existing encryption protocols; and then propose an instantiation, the PKE-FET scheme, from secret sharing and bilinear map. In addition, we prove its semantic security in the standard model. Finally, the efficiency of PKE-FET is not well due to massive bilinear mapping operations. We keep the efficiency improvement as an open problem.

ACKNOWLEDGEMENTS

This research is supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 103-2221-E-004-009. We appreciate the anonymous reviewers for their valuable suggestions.

REFERENCES

Bellare, M., Boldyreva, A., and Micali, S. (2000). Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, pages 259–274.

Chatterjee, S. and Menezes, A. (2011). On cryptographic protocols employing asymmetric pairings - the role of revisited. *Discrete Applied Mathematics*, 159(13):1311–1322.

Diffie, W. and Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

Fouque, P., Joux, A., and Mavromati, C. (2014). Multi-user collisions: Applications to discrete logarithm, even-mansour and PRINCE. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, pages 420–438.

Galbraith, S. D., Paterson, K. G., and Smart, N. P. (2008). Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121.

Gamal, T. E. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472.

Ghadafi, E., Smart, N. P., and Warinschi, B. (2010). Groth-sahai proofs revisited. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, pages 177–192.

Huang, K., Tso, R., Chen, Y., Li, W., and Sun, H. (2014). A new public key encryption with equality test. In *Network and System Security - 8th International Conference, NSS 2014, Xi'an, China, October 15-17, 2014, Proceedings*, pages 550–557.

Huang, K., Tso, R., Chen, Y.-C., Rahman, S. M. M., Almogren, A., and Alamri, A. (2015). Pke-aet: Public key encryption with authorized equality test. *The Computer Journal*, page bxv025.

Ma, S., Zhang, M., Huang, Q., and Yang, B. (2014). Public key encryption with delegated equality test in a multi-user setting. *The Computer Journal*.

Naor, M. and Yung, M. (1989). Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 33–43.

Peng, K., Boyd, C., Dawson, E., and Lee, B. (2005). Ciphertext comparison, a new solution to the millionaire problem. In *ICICS*, pages 84–96.

Sakurai, K. and Shizuya, H. (1995). Relationships among the computational powers of breaking discrete log cryptosystems. In *EUROCRYPT*, pages 341–355.

Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

Shoup, V. (2004). Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332.

Tang, Q. (2012a). Public key encryption schemes supporting equality test with authorisation of different granularity. *IJACT*, 2(4):304–321.

Tang, Q. (2012b). Public key encryption supporting plaintext equality test and user-specified authorization. *Security and Communication Networks*, 5(12):1351–1362.

Yang, G., Tan, C. H., Huang, Q., and Wong, D. S. (2010). Probabilistic public key encryption with equality test. In *CT-RSA*, pages 119–131.