A Symmetric Authenticated Proxy Re-Encryption Scheme with Provable Security

Zhiniang Peng[1], Shaohua Tang*[1], and Linzhi Jiang[23]
246003@qq.com, shtang@ieee.org, linzjiang@hotmail.com

[1] School of Computer Science & Engineering,
South China University of Technology, Guangzhou 510006, China
[2] University of Electronic Science and Technology of China, Chengdu 611731, China
[3] West Anhui University, Liuan 237012, China

Abstract. In crypto 2013, Dan et al. proposed a symmetric proxy re-encryption scheme based on key homomorphic PRF. It can be used to ensure the data privacy in cloud storage systems. However, it only focuses on preventing a honest-but-curious proxy from learning anything about the encrypted data. Although it can be made to provide integrity without disrupting the key homomorphism property by using MAC then encrypt with counter-mode, it's not a symmetric authenticated proxy re-encryption scheme because only the data owner can verify the integrity of some encrypted data. In this paper, we propose a symmetric authenticated proxy re-encryption scheme which can prevent a malicious proxy from tampering users' data. It can update the authentication tag as well as the ciphertext so that any intended user can verify the integrity of the encrypted data.

Key words: symmetric proxy Re-Encryption, key homomorphic PRF, cloud storage, Encryption-and-Mac.

1 Introduction

Proxy re-encryption (PRE) [1] [7] allows a proxy to transform a ciphertext encrypted under Alices key into one that can be decrypted by Bobs key. It has many useful applications in cloud storage [15] [25]. Many proxy re-encryption schemes have been proposed in the asymmetric crypto world [11] [14] [8].

In many applications, data is always encrypted using symmetric cipher for better efficiency. Although an asymmetric PRE scheme can be efficiently used to re-encrypt the key which is used to encrypt the data with symmetric cipher [1], there is a weakness [23] that re-encrypting the key does not update the actual key used to encrypt data (the key of the symmetric encryption) and users who have the previous key can use the same key to decrypt the data.

For example, a company stores some data encrypted under a key sk_1 on a cloud based storage system and any employee who knows sk_1 has access to it.

* Corresponding author: Shaohua Tang, E-mail: shtang@ieee.org.

As employees leave the company, there is a need to re-encrypt it using a new key sk_2. To prevent the cloud from obtaining any information about the data, a naive way is to download the entire ciphertext from the cloud, re-encrypt under a new key, and upload the new ciphertext to the cloud. Unfortunately, downloading and re-uploading all the data from the cloud just for the purpose of key rotation results in considerable wasted bandwidth and cost. Asymmetric PRE schemes are not practical in this case especially when the size of the data to be encrypted is large.

Typically, there are two ways to solve this problem. The first one is using double encryption strategy proposed in [9], but its computation overhead is doubled and it's inflexible for many scenarios. The second one is to encrypt the data using a symmetric proxy re-encryption scheme (S-PRE) and use the cloud as the proxy holding the companys encrypted data. Now, by simply sending to the cloud the re-encryption key, the cloud can translate the ciphertext from key sk_1 to key sk_2 without doing any large data transfers.

There are S-PRE schemes based on some primitive with algebraic property such as ANOT [23] [21], but those basic primitives are always inefficient. In crypto 2013, Dan et al. proposed a S-PRE [5] based on key homomorphic pseudorandom function (KH-PRF) [2]. It provides IND-CPA security and has various applications in cloud based systems.

The complete system model in cloud computing [10] or clouding storage should involve three different entities: the data owner, the data user and the cloud server [24] [26]. Dan's scheme can be made to provide integrity without disrupting the key homomorphism property by using MAC-then-encrypt with counter-mode [12]. This can ensure that the data owner can verify the integrity of outsourced data. However, data users can not verify the integrity of encrypted data. One way to provide integrity for data users is to use digital signature schemes, but it will introduce asymmetric cryptography and it's not suitable for some applications where the data owner wants to hide his identity. To prevent the malicious proxy (cloud server) to tamper users' data, a better solution is to build a symmetric authenticated proxy re-encryption scheme where every user has independent MAC key and encryption key, the re-encryption algorithm can update the encryption key as well as the MAC key to make sure every intended data user can verify the integrity of some encrypted data with his/her own key.

Our Contributions: In this paper, we propose a symmetric authenticated proxy re-encryption (SA-PRE) scheme with provable security in random oracle model. It's the first SA-PRE scheme based on KH-PRF and it can re-encrypt the authentication tag as well as the ciphertext. Our scheme provides indistinguishability under chosen plaintext attack (IND-CPA) and ciphertext integrity (INT-CTXT) against any malicious user and malicious proxy. Our work also shows that generic Encryption-and-Mac composition [3] can be used to get an authenticated encryption scheme if nonce is used and the MAC is based on PRF.

2 Preliminaries

2.1 Key Homomorphic Pseudorandom Function

Definition 1. *Key homomorphic PRF Consider an efficiently computable function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that $(\mathcal{K}, \oplus)$ and $(\mathcal{Y}, \otimes)$ are both groups. We say that $(F, \oplus, \otimes)$ is a key homomorphic PRF if the following properties hold:*

1) F is a secure pseudorandom function.
2) For every $k_1, k_2 \in \mathcal{K}$ and every $x \in \mathcal{X}$, $F(k_1, x) \otimes F(k_2, x) = F(k_1 \oplus k_2, x)$.

Let $\mathcal{G}$ be a finite cyclic group of prime order q and let $H_1 : \mathcal{X} \to \mathcal{G}$ be a hash function modeled as random oracle. Then we define function $F_m : Z_q \times \mathcal{X} \to \mathcal{G}$ as:

$$F_m(k, x) \leftarrow H_1(x)^k.$$

If the Decision Diffe-Hellman (DDH) [4] assumption holds in $\mathcal{G}$, then F_m is a secure PRF in random oracle model [18]. As we can see that $F_m(k_1 + k_2, x) = F_m(k_1, x) \times F_m(k_2, x)$, this PRF is clearly key homomorphic. KH-PRF has many applications in cloud storage systems [19]. It can sometimes be replaced by a y-almost KH-PRF [6] [13] which can be constructed based on LWE [20] or Ring-LWE problems [16] [17].

2.2 Symmetric Proxy Re-Encryption from KH-PRF

Let $F_e : \mathcal{K}_e \times \mathcal{X} \to \mathcal{Y}_e$ be a KH-PRF, the symmetric proxy re-encryption scheme proposed in [5] consists of the following algorithms.

$Setup(1^k) \to pp$: According to the security parameter 1^k, the setup algorithm chooses appropriate parameters pp for F_e.

$KeyGen(pp) \to sk$: The key generation algorithm outputs a random key $sk \in F_e$.

$ReKeyGen(sk_i, sk_j) \to rk_{ij}$: The re-encryption key generation algorithm takes two secret key sk_i and sk_j as inputs and returns $rk_{ij} = sk_j - sk_i$ as re-encryption key.

$Enc(sk, m) \to C$: The encryption algorithm chooses a random r from X and computes $c = m + F_e(sk, r)$. Then it outputs (r, c).

$ReEnc(rk_{ij}, C_i) \to C_j$: The re-encryption algorithm takes ciphertext $C_i = (r_i, c_i)$ and re-encryption key rk_{ij} as inputs. It Outputs $(r_i, c_i + F_e(rk_{ij}, r_i))$.

$Dec(sk, C) :\to m$: The decryption algorithm takes ciphertext $C = (r, c)$ as inputs, it outputs $c - F(sk, r)$.

If F_e is a secure KH-PRF, then this scheme provides IND-CPA security against various adversaries.

3 Our Scheme

Our proposal is based on the S-PRE in [5]. We introduce an updatable MAC to it and prove its security in random oracle model. Let $F_e : \mathcal{K}_e \times \mathcal{X} \rightarrow \mathcal{Y}_e$ be a KH-PRF and $F_m : \mathcal{K}_m \times \mathcal{X} \rightarrow \mathcal{Y}_m$ be the KH-PRF based on DDH assumption. Suppose message $m \in \mathcal{Y}_e$, our SA-PRE scheme consists of the following six algorithms:

$Setup(1^k) \rightarrow pp$: According to the security parameter 1^k, the setup algorithm chooses appropriate parameters pp for F_e and F_m.

$KeyGen(pp) \rightarrow sk$: The key generation algorithm takes public parameters pp and outputs a random key sk as follows:
- Randomly choose a key $k_e \in \mathcal{K}_e$ for F_e.
- Randomly choose a key $k_m \in \mathcal{K}_m$ for F_m.
- Let $sk = (k_e, k_m)$.

$ReKeyGen(sk_i, sk_j) \rightarrow rk_{ij}$: The re-encryption key generation algorithm takes two secret key sk_i and sk_j as input. It outputs the re-encryption key rk_{ij} as follows:
- Compute $rk_{e_{ij}} = k_{e_j} - k_{e_i}$.
- Compute $rk_{m_{ij}} = k_{m_j} / k_{m_i}$.
- Let $rk_{ij} = (rk_{e_{ij}}, rk_{m_{ij}})$.

$Enc(sk, m) \rightarrow C$: The encryption algorithm takes message m and secret key $sk = (k_e, k_m)$ as input. It computes the ciphertext C as follows:
- Randomly choose nonce r.
- Compute $c = F_e(k_e, r) + m$.
- Compute $t = F_m(k_m, r||m)$.
- Let $C = (r, c, t)$.

$ReEnc(rk_{ij}, C_i) \rightarrow C_j$: The re-encryption algorithm takes ciphertext $C_i = (r_i, c_i, t_i)$ and re-encryption key $rk_{ij} = (rk_{e_{ij}}, rk_{m_{ij}})$ as input. It computes the ciphertext C_j as follows:
- Let $r_j = r_i$.
- Compute $c_j = c_i + F_e(rk_{e_{ij}}, r_j)$.
- Compute $t_j = (t_i)^{rk_{m_{ij}}}$.
- Let $C_j = (r_j, c_j, t_j)$.

$Dec(sk, C) :\rightarrow m$ or $\perp$: The decryption algorithm takes ciphertext $C = (r, c, t)$ and secret key $sk = (k_e, k_m)$ as input. It computes the results as follows:
- Compute $m = c - F_e(k_e, r)$.
- Return $\perp$ if $t \neq F_m(k_m, r||m)$.
- Otherwise return m

Correctness. Let $pp \leftarrow Setup(1^k)$, $sk \leftarrow KeyGen(pp)$ and $m \in \mathcal{Y}_e$. Then $Dec(sk, Enc(sk, m)) = m + F_e(k_e, r) - F_e(k_e, r) = m$ as desired for all nonce r. Let $sk_i \leftarrow KeyGen(pp)$, $sk_j \leftarrow KeyGen(pp)$ and $rk_{ij} \leftarrow ReKeyGen(sk_i, sk_j)$. Then $Dec(sk_j, ReEnc(rk_{ij}, Enc(sk_i, m))) = m$ holds for all nonce r.

If message m is too large to fit in an element of $\mathcal{Y}_e$, we can convert it to a vector over $\mathcal{Y}_e$ and extend the length of $F_e(k_e, r)$ by introducing a counter in r. Similar technique are used in the updatable encryption scheme in [5].

4 Security Analysis

In this section, we will prove that our SA-PRE scheme is a secure authenticated encryption against any malicious user and malicious proxy. Our scheme is designed by generic Encryption-and-Mac composition, meaning that we use an IND-CPA secure symmetric encryption scheme and a SUF-CMA secure MAC scheme in a black box way. In a ciphertext $C = (r, c, t)$, r is a nonce; c can be considered as a ciphertext encrypted by a stream cipher based on PRF F_e; t can be considered as a authentication tag computed by a MAC scheme based on PRF F_m.

However, Bellare and Namprempre proved that generic Encryption-and-Mac can only guarantee plaintext integrity (INT-PTXT) in paper [3]. This is because that MAC could reveal information about the plaintext. But this does not mean that our scheme is insecure. In fact, our work shows that generic Encryption-and-Mac composition can be used to get an INT-CTXT $\wedge$ IND-CPA secure authenticated encryption if nonce is introduced and the MAC is based on PRF.

4.1 Security against any Malicious User

We considered a security model inspired by the work of Canetti and Hohenberger [7]. A malicious user can access re-encryption oracle as well as encryption oracle. The encryption oracle ($C \leftarrow Enc(i, m)$) takes an user's index i and plaintext m as inputs and outputs the ciphertext C encrypted under user i's sk_i. The re-encrypt oracle ($C_j \leftarrow Re(C_i, sk_j)$) takes an ciphertext C_i and an new secret key sk_j as inputs and transforms the ciphertext encrypted under sk_i to a new ciphertext encrypted under sk_j.

INT-CTXT against any Malicious User

Theorem 1. *The advantage of any malicious user $\mathcal{A}$ in INT-CTXT game against our scheme $\mathcal{E}$ is less than ϵ_m, where ϵ_m is the advantage of some efficient algorithm against PRF F_m.*

Proof. We construct an adversary $\mathcal{B}$ which uses $\mathcal{A}$ to break the PRF F_m. The game between the challenger and the adversary $\mathcal{B}$ starts with the challenger first randomly chooses a PRF key $k_m \in K_m$. $\mathcal{B}$ can asks $F_m(k_m, r)$ for different r. Then $\mathcal{B}$ is supposed to output a PRF pair $(r, F_m(k_m, r))$ which has not been asked before. For simplicity, we suppose nonce r will never repeat.

$\mathcal{B}$ works by interacting with $\mathcal{A}$ in an INT-CTXT game as follows ($\mathcal{B}$ simulates the challenger for $\mathcal{A}$):

Setup: $\mathcal{B}$ chooses appropriate parameter pp and gives it to $\mathcal{A}$. $\mathcal{B}$ creates empty lists $\mathcal{SK}$ and $\mathcal{C}$.

Enc: When $\mathcal{A}$ asks the encryption oracle for user i and plaintext m_k, $\mathcal{B}$ responds as follows:

- If i is not in $\mathcal{SK}$, then randomly choose $k_{e_i} \in \mathcal{K}_e$ and $k_{m_i} \in \mathcal{K}_m$ for user i. Add $(i$,k_{e_i},$k_{m_i})$ to $\mathcal{SK}$.

- Randomly choose r_k, and ask PRF challenger for $v_k = F_m(k_m, r_k||m_k)$.
- Let $c_k = m_k + F_e(k_{e_i}, r_k)$ and $t_k = {v_k}^{k_{m_i}}$.
- Add (r_k, c_k, t_k, m_k) to list $\mathcal{C}$ and respond to $\mathcal{A}$ with (r_k, c_k, t_k).

ReEnc: When $\mathcal{A}$ asks the re-encryption oracle to convert ciphertext $C = (r, c, t)$ encrypted under user i's secret key into ciphertext under secret key $sk_j = (k_{e_j}, k_{m_j})$, $\mathcal{B}$ responds as follows:

- If i is not in $\mathcal{SK}$:
  Return $\perp$, and stop the simulation.
- If C is in $\mathcal{C}$:
  Get the corresponding message m.
  Let $c_j = m + F_e(k_{e_j}, r)$, $t_j = F_m(k_{m_j}, r_j||m)$
  Respond to $\mathcal{A}$ with (r, c_j, t_j).
- If (r, t) is not in $\mathcal{C}$:
  Get the user i's key (i,k_{e_i},k_{m_i}) from list $\mathcal{SK}$.
  Compute $m = c - F_e(k_{e_i}, r)$.
  Compute $t = t^{\frac{1}{k_{m_i}}}$.
  Output PRF pair $(r||m, t)$ to PRF challenger.
- If (r, t) is in $\mathcal{C}$, but c is not in $\mathcal{C}$:
  Compute $m = c - F_e(k_{e_i}, r)$.
  Get (m', c') from $\mathcal{C}[r, t]$.
  m certainly not equal to m', otherwise c will equal to c'.
  Compute $t = t^{\frac{1}{k_{m_i}}}$.
  Output PRF pair $(r||m, t)$ to PRF challenger.

Forge: Eventually, $\mathcal{A}$ outputs a forged ciphertext $C = (r, c, t)$ for user i. $\mathcal{B}$ outputs a PRF pair as follows:

- Get the user i's key (i,k_{e_i},k_{m_i}) from list $\mathcal{SK}$.
- Compute $m = c - F_e(k_{e_i}, r)$.
- Compute $t = t^{\frac{1}{k_{m_i}}}$.
- Output PRF pair $(r||m, t)$ to PRF challenger.

Claim: During the simulation, $\mathcal{A}$'s view is identical to that in the real attack. If $\mathcal{A}$ can forge a valid ciphertext during ReEnc phase or Forge phase, $\mathcal{B}$ can use it to break PRF F_m with at least the some advantage. So we can conclude that

$$Adv_{\mathcal{E}}^{INT-CTXT}(\mathcal{A}) \leq Adv_{PRF}(\mathcal{B}) \leq \epsilon_{F_m}.$$

IND-CPA against any Malicious User

Theorem 2. *The advantage of any malicious user in IND-CPA game against our scheme $\mathcal{E}$ is less than $2 * \epsilon_{F_m} + \epsilon_{F_e}$, where ϵ_{F_m} is the advantage of some efficient algorithm against PRF F_m and ϵ_{F_e} is the advantage of some efficient algorithm against PRF F_e.*

Proof. Our proof uses a sequence of games[22].

Game 0: Fix a malicious user $\mathcal{A}$. Let us define Game 0 to be the real attack game in the definition of IND-CPA. We describe the attack game using figure 1

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $c = F_e(k_e, r) + m_b$
  $t = F_m(k_m, r||m_b)$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc($m$):
  $r \leftarrow R$
  $c = F_e(k_e, r) + m$
  $t = F_m(k_m, r||m)$
  $\mathcal{C}[(r, c, t)] = m$
  return: $(r, c, t)$

Re($C$,$\hat{sk}$):
  $(r, c, t) = C$
  $m = c - F_e(k_e, r)$
  If $t \neq F_m(k_m, r||m)$
    return: $\perp$
  $(\hat{k_e}, \hat{k_m}) = \hat{sk}$
  $c = m + F_e(\hat{k_e}, r)$
  $t = F_m(\hat{k_m}, r||m)$
  return: $(r, c, t)$
```

Fig. 1. Game 0.

and define S_i to be the event that Game i return true, then $\mathcal{A}$'s advantage in Game 0 is $|Pr[S_0] - 1/2|$.

Game 1: [This is a transition based on INT-CTXT of $\mathcal{E}$.] We now make a small change to Game 0. The differences between Game 0 and Game 1 are boxed up in figure 2. Instead of re-encrypting every valid ciphertext, we only re-encrypt the ciphertext recorded in a list $\mathcal{C}$. Define E_{01} as the event that $\mathcal{A}$ produces a new valid ciphertext and asks re-encryption oracle to re-encrypt it. Game 0 and Game 1 are identical for $\mathcal{A}$ unless E_{01} happens. If E_{01} happens, it means that INT-CTXT of $\mathcal{E}$ is broken.

If $\mathcal{A}$ can distinguish Game 0 and Game 1, then we can construct an adversary $\mathcal{F}$ that can break the INT-CTXT of $\mathcal{E}$ with at least the same advantage. Then we can get that

$$|Pr[S_0] - Pr[S_1]| \leq Pr[E_{01}] \leq Adv_{\mathcal{E}}^{INT-CTXT}(\mathcal{F}) \leq \epsilon_{F_m} . \qquad (1)$$

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $c = F_e(k_e, r) + m_b$
  $t = F_m(k_m, r||m_b)$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc($m$):
  $r \leftarrow \mathcal{R}$
  $c = F(k_e, r) + m$
  $t = F(k_m, r||m)$
  $\mathcal{C}[(r, c, t)] = m$
  return: $(r, c, t)$

Re($C$,$\hat{sk}$):
  $(r, c, t) = C$
  $\boxed{m = \mathcal{C}[c, r, t]}$
  $\boxed{\text{If } C[c, r, t] = \perp}$
    return: $\perp$
  $(\hat{k_e}, \hat{k_m}) = \hat{sk}$
  $c = m + F_e(\hat{k_e}, r)$
  $t = F_m(\hat{k_m}, r||m)$
  return: $(r, c, t)$
```

Fig. 2. Game 1.

Game 2: [This is a transition based on the indistinguishability of PRF F_e.] In Game 2, we use real random function $\mathcal{U}$ instead of PRF F_e during encryption and challenge ciphertext generation. Differences between Game 1 and Game 2 are boxed up in figure 3. If $\mathcal{A}$ can distinguish Game 1 and Game 2, then we can construct an adversary $\mathcal{F}_e$ that can break the indistinguishability of PRF F_e with at least the same advantage. Then we can get that

$$|Pr[S_2] - Pr[S_1]| \leq \epsilon_{F_e} . \qquad (2)$$

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $\boxed{c = \mathcal{U} + m_b}$
  $t = F_m(k_m, r||m_b)$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc(m):
  $r \leftarrow \mathcal{R}$
  $\boxed{c = \mathcal{U} + m}$
  $t = F(k_m, r||m)$
  $\mathcal{C}[(r, c, t)] = m$
  return: $(r, c, t)$

Re(C, \hat{sk}):
  $(r, c, t) = C$
  $m = \mathcal{C}[c, r, t]$
  If $C[c, r, t] = \perp$
    return: $\perp$
  $(\hat{k_e}, \hat{k_m}) = \hat{sk}$
  $c = m + F_e(\hat{k_e}, r)$
  $t = F_m(\hat{k_m}, r||m)$
  return: $(r, c, t)$
```

Fig. 3. Game 2.

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $c = \mathcal{U} + m_b$
  $\boxed{t = \mathcal{U}}$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc(m):
  $r \leftarrow \mathcal{R}$
  $c = \mathcal{U} + m$
  $\boxed{t = \mathcal{U}}$
  $\mathcal{C}[(r, c, t)] = m$
  return: $(r, c, t)$

Re(C, \hat{sk}):
  $(r, c, t) = C$
  If $C[c, r, t] = \perp$
    return: $\perp$
  $(\hat{k_e}, \hat{k_m}) = \hat{sk}$
  $m = C[c, r, t]$
  $c = m + F_e(\hat{k_e}, r)$
  $t = F_m(\hat{k_m}, r||m)$
  return: $(r, c, t)$
```

Fig. 4. Game 3.

Game 3: [This is a transition based on the indistinguishability of PRF F_m.] In Game 3, we use real random function $\mathcal{U}$ instead of PRF F_m during encryption and challenge ciphertext generation. Differences between Game 2 and Game 3 are boxed up in figure 4. If $\mathcal{A}$ can distinguish Game 2 and Game 3, then we can construct an adversary $\mathcal{F}_m$ that can break the indistinguishability of PRF F_m with at least the same advantage. Then we can get that

$$|Pr[S_3] - Pr[S_2]| \le \epsilon_{F_m} . \qquad (3)$$

Combining 1, 2, and 3, we can get that

$$|Pr[S_0] - Pr[S_3]| \le 2 * \epsilon_{F_m} + \epsilon_{F_e} .$$

In Game 3, all the ciphertexts leak no information about the secret key sk. The challenge ciphertext is independent of messages m_0 and m_1. This means that the adversary's advantage in Game 3 is $|Pr[S_3] - 1/2| = 0$. Then we can conclude that the advantage of any malicious user in IND-CPA game of our scheme is less than $2 * \epsilon_{F_m} + \epsilon_{F_e}$.

4.2 Security against any Malicious Proxy

A malicious proxy can access encryption oracle as well as re-encryption key generation oracle. The re-encryption key generation oracle ($C \leftarrow ReK(i, j)$) takes two users' index i and j and outputs the re-encryption key rk_{ij}, which can be used to convert a ciphertext originally encrypted under sk_i into a new ciphertext encrypted under sk_j.

INT-CTXT against any Malicious Proxy

Theorem 3. *The advantage of a malicious proxy in INT-CTXT game against our scheme $\mathcal{E}$ is less than ϵ_m, where ϵ_m is the advantage of some efficient algorithm against PRF F_m.*

Proof. We construct an adversary $\mathcal{B}$ using $\mathcal{A}$ to break the PRF F_m. The game between the challenger and the adversary $\mathcal{B}$ starts with the challenger first randomly chooses a PRF key $k_m \in \mathcal{K}_m$. $\mathcal{B}$ asks $F_m(k_m, r)$ for different r. Then $\mathcal{B}$ is supposed to output a PRF pair $(r, F_m(k_m, r))$ which was not asked before.

$\mathcal{B}$ works by interacting with $\mathcal{A}$ in an INT-CTXT game as follows ($\mathcal{B}$ simulates the challenger for $\mathcal{A}$):

Setup: $\mathcal{B}$ chooses appropriate parameters pp and gives it to $\mathcal{A}$. Then $\mathcal{B}$ creates empty lists $\mathcal{SK}$ and $\mathcal{C}$.

Enc: When $\mathcal{A}$ asks the encryption oracle for user i and plaintext m_k, $\mathcal{B}$ responds as follows:
– If i is not in $\mathcal{SK}$, then randomly choose $k_{e_i} \in \mathcal{K}_e$ and $k_{m_i} \in \mathcal{K}_m$ for user i and add (i,k_{e_i},k_{m_i}) to $\mathcal{SK}$.
– Randomly choose r_k, and ask PRF challenger for $v_k = F_m(k_m, r_k||m_k)$.
– Let $c_k = m_k + F_e(k_{e_i}, r_k)$ and $t_k = {v_k}^{k_{m_i}}$.
– Respond to A with (r_k, c_k, t_k).

ReK: When $\mathcal{A}$ asks the re-encryption key generation oracle for a re-encryption key from user i to user j, $\mathcal{B}$ responds as follows:
– If i or j is not in $\mathcal{SK}$:
  Return $\perp$, and stop the simulation.
– Get (k_{e_i}, k_{m_i}) and (k_{e_j}, k_{m_j}).
– Let $rk_{e_{ij}} = k_{e_j} - k_{e_i}$ and $rk_{m_{ij}} = k_{m_j}/k_{m_i}$.
– Respond to $\mathcal{A}$ with $(rk_{e_{ij}}, rk_{m_{ij}})$.

Forge: Eventually, $\mathcal{A}$ outputs a forged ciphertext $C = (r, c, t)$ for user i. $\mathcal{B}$ outputs a PRF pair as follows:
– Get the user i's key (k_{e_i},k_{m_i}) from list $\mathcal{SK}$.
– Compute $m = c - F_e(k_{e_i}, r)$.
– Compute $t = t^{\frac{1}{k_{m_i}}}$.
– Output PRF pair $(r||m, t)$ to PRF challenger.

Claim: During the simulation, $\mathcal{A}$'s view is identical to that in the real attack. If $\mathcal{A}$ can forge a valid ciphertext, $\mathcal{B}$ can use it to break PRF F_m. So we can conclude that

$$Adv_{\mathcal{E}}^{INT-CTXT}(\mathcal{A}) \leq Adv_{PRF}(\mathcal{B}) \leq \epsilon_{F_m}.$$

IND-CPA against any Malicious Proxy

Theorem 4. *The advantage of a malicious proxy in IND-CPA game against our scheme $\mathcal{E}$ is less than $\epsilon_{F_m} + \epsilon_{F_e}$, where ϵ_{F_m} is the advantage of some efficient algorithm against PRF F_m and ϵ_{F_e} is the advantage of some efficient algorithm against PRF F_e.*

Proof. We use a sequence of games in our proof.

Game $\hat{0}$: Fix a malicious proxy $\mathcal{A}$. Let us define Game $\hat{0}$ to be the real attack game against $\mathcal{A}$ in the definition of IND-CPA. We describe it in figure 5. We define $\hat{S}_i$ to be the event that Game $\hat{i}$ return true, then $\mathcal{A}$'s advantage in Game $\hat{0}$ is $|Pr[\hat{S}_0] - 1/2|$.

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $i, m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $c = F_e(k_e, r) + m_b$
  $c = c + F(k_{e_i}, r)$
  $t = F_m(k_m, r||m_b)^{k_{m_i}}$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc(i, m):
  If $SK[i] = \perp$
    $(k_{e_i}, k_{m_i}) \leftarrow \mathcal{K}$
    $SK[i] = (k_{e_i}, k_{m_i})$
  $r \leftarrow \mathcal{R}$
  $c = F(k_e, r) + m$
  $c = c + F(k_{e_i}, r)$
  $t = F(k_m, r||m)^{k_{m_i}}$
  return: $(r, c, t)$

Re(i, j):
  If $SK[i] = \perp$ or $SK[j] = \perp$
    return: $\perp$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $(k_{e_j}, k_{m_j}) = SK[j]$
  $rk_{e_{ij}} = k_{e_j} - k_{e_i}$
  $rk_{m_{ij}} = k_{m_j} / k_{m_i}$
  return: $(rk_{e_{ij}}, rk_{m_{ij}})$
```

Fig. 5. Game $\hat{0}$.

Game $\hat{1}$: [This is a transition based on the indistinguishability of PRF F_e.] In Game $\hat{0}$, we use real random function $\mathcal{U}$ instead of PRF during encryption and challenge ciphertext generation. Differences between Game $\hat{0}$ and Game $\hat{1}$ are boxed up in figure 6. If $\mathcal{A}$ can distinguish Game $\hat{0}$ and Game $\hat{1}$, then we can construct an adversary $\mathcal{F}_m$ that can break the indistinguishability of PRF F_e with at least the same advantage. Then we can get that

$$|Pr[\hat{S}_0] - Pr[\hat{S}_1]| \leq \epsilon_{F_m} . \tag{4}$$

```
Game(1^k,b):
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $i, m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $\boxed{c = \mathcal{U} + m_b}$
  $c = c + F_e(k_{e_i}, r)$
  $t = F_m(k_m, r||m_b)^{k_{m_i}}$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$

Enc(i, m):
  If $SK[i] = \perp$
    $(k_{e_i}, k_{m_i}) \leftarrow \mathcal{K}$
    $SK[i] = (k_{e_i}, k_{m_i})$
  $r \leftarrow R$
  $\boxed{c = \mathcal{U} + m_0}$
  $c = c + F_e(k_{e_i}, r)$
  $t = F(k_m, r||m)^{k_{m_i}}$
  return: $(r, c, t)$

Re(i, j):
  If $SK[i] = \perp$ or $SK[j] = \perp$
    return: $\perp$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $(k_{e_j}, k_{m_j}) = SK[j]$
  $rk_{e_{ij}} = k_{e_j} - k_{e_i}$
  $rk_{m_{ij}} = k_{m_j} / k_{m_i}$
  return: $(rk_{e_{ij}}, rk_{m_{ij}})$
```

Fig. 6. Game $\hat{1}$.

Game $\hat{2}$ [This is a transition based on the indistinguishability of PRF F_m.] In Game $\hat{2}$, we use real random function $\mathcal{U}$ instead of PRF F_m during encryption and challenge ciphertext generation. Differences between Game $\hat{2}$ and Game $\hat{1}$ are boxed up in figure 7. If there is an adversary $\mathcal{A}$ which can distinguish Game

$\hat{2}$ and Game $\hat{1}$, then we can construct an adversary $\mathcal{F}_m$ that can break the indistinguishability of PRF F_m with at least the same advantage. Then we can get that

$$|Pr[\hat{S}_2] - Pr[\hat{S}_1]| \leq \epsilon_{F_m}. \tag{5}$$

```
$\underline{\textbf{Game}(1^k,\textbf{b})}$:
  $pp \leftarrow Setup(1^k)$
  $(k_e, k_m) \leftarrow \mathcal{K}$
  $i, m_0, m_1 \leftarrow \mathcal{A}^{Re,Enc}(pp)$
  $r \leftarrow \mathcal{R}$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $c = \mathcal{U} + m_b$
  $c = c + F_e(k_{e_i}, r)$
  $\boxed{t = \mathcal{U}^{k_{m_i}}}$
  $b' \leftarrow \mathcal{A}(r, c, t)$
  return $(b = b')$
```

```
$\underline{\textbf{Enc}(i, m)}$:
  If $SK[i] = \perp$
    $(k_{e_i}, k_{m_i}) \leftarrow \mathcal{K}$
    $SK[i] = (k_{e_i}, k_{m_i})$
  $r \leftarrow R$
  $c = \mathcal{U} + m_0$
  $c = c + F_e(k_{e_i}, r)$
  $\boxed{t = \mathcal{U}^{k_{m_i}}}$
  return: $(r, c, t)$
```

```
$\underline{\textbf{Re}(i, j)}$:
  If $SK[i] = \perp$ or $SK[j] = \perp$
    return: $\perp$
  $(k_{e_i}, k_{m_i}) = SK[i]$
  $(k_{e_j}, k_{m_j}) = SK[j]$
  $rk_{e_{ij}} = k_{e_j} - k_{e_i}$
  $rk_{m_{ij}} = k_{m_j}/k_{m_i}$
  return: $(rk_{e_{ij}}, rk_{m_{ij}})$
```

Fig. 7. Game $\hat{2}$.

Combining 4 and 5 , we can get that

$$|Pr[\hat{S}_0] - Pr[\hat{S}_2] \leq |Pr[\hat{S}_0] - Pr[\hat{S}_1] + Pr[\hat{S}_1] - Pr[\hat{S}_2]| \leq \epsilon_{F_m} + \epsilon_{F_e}.$$

In Game $\hat{2}$, all the ciphertexts leak no information about secret key sk. The challenge ciphertext is independent of messages m_0 and m_1. This means that $\mathcal{A}$'s advantage in Game $\hat{2}$ is $|Pr[\hat{S}_2] - 1/2| = 0$. Then we can conclude that the advantage of any malicious proxy in IND-CPA game of our scheme is less than $\epsilon_{F_m} + \epsilon_{F_e}$.

4.3 IND-CCA security

According to theorem 3.2 in [3], an encryption scheme that is both IND-CPA secure and IND-CTXT secure is also IND-CCA secure (INT-CPA $\wedge$ INT-CTXT $\rightarrow$ IND-CCA). Combining above theorems, we can conclude that our SA-PRE $\mathcal{E}$ is IND-CCA secure against any malicious user and any malicious proxy.

In fact, our security reduction shows that generic Encryption-and-Mac composition can be used to get an authenticated encryption scheme if nonce is used and the MAC is based on PRF.

5 Comparison and Conclusions

The most costly parts of our scheme are evaluations of KH-PRFs. So we compare our SA-PRE with Dan's S-PRE by counting the number of KH-PRFs need to be evaluated. Suppose we want to encrypt a message with size of n times the size of

Table 1. Comparison of our SA-PRE with Dan's S-PRE.

	Dan's S-PRF	Our SA-PRF
Security	IND-CPA	IND-CCA
Key size	size of a KH-PRF key	size of two KH-PRF keys
Encryption	n KH-PRF evaluations	n+1 KH-PRF evaluations
Decryption	n KH-PRF evaluations	n+1 KH-PRF evaluations
ReKeyGen	1 KH-PRF basic operation	2 KH-PRF basic operations
Re-encryption	n KH-PRF evaluations	n+1 KH-PRF evaluations

KH-PRF, we need to evaluate n KH-PRFs to mask the plaintext and 1 KH-PRF to compute the authentication tag. We give a comparison of our SA-PRE with Dan's S-PRE in Table 1.

From Table 1, we can observe that the key size and re-encryption key generating time of our SA-PRE are doubled compared with the Dan's S-PRE and our SA-PRE need one more KH-PRE evaluations in encryption, decryption and re-encryption compared with Dan's S-PRE. When n become larger, the performance gap between our SA-PRE will be negligible. So we can conclude that our SA-PRE is comparable to Dan's S-PRE in performance.

In this paper, we propose a SA-PRE by introducing an updatable MAC into Dan's S-PRE to provide authentication. The re-encryption process can update the authentication key as well as the encryption key. We prove that it's IND-CPA and INT-CTXT secure against any malicious user or any malicious proxy. Our work also shows that the generic Encryption-and-Mac composition can be used to get an authenticated encryption scheme if nonce is used and the MAC is based on PRF.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Nos. 61632013, U1135004 and 61170080), 973 Program (No. 2014CB360501), Guangdong Provincial Natural Science Foundation (No. 2014A030308006), and Guangdong Provincial Project of Science and Technology (no. 2016B090920081).

References

1. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transactions on Information and System Security (TISSEC) 9(1), 1–30. ACM, New York (2006)
2. Banerjee, A., Peikert, C.: New and improved key-homomorphic pseudorandom functions. In: International Cryptology Conference. pp. 353–370. Springer, Heidelberg (2014)
3. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 531–545. Springer, Heidelberg (2000)

4. Boneh, D.: The decision diffie-hellman problem. In: International Algorithmic Number Theory Symposium. pp. 48–63. Springer, Heidelberg (1998)
5. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Advances in Cryptology–CRYPTO 2013, pp. 410–428. Springer, Heidelberg (2013)
6. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions. In: Theory of Cryptography Conference. pp. 1–30. Springer, Heidelberg (2015)
7. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM conference on Computer and communications security. pp. 185–194. ACM, New York (2007)
8. Chow, S.S., Weng, J., Yang, Y., Deng, R.H.: Efficient unidirectional proxy re-encryption. In: International Conference on Cryptology in Africa. pp. 316–332. Springer, Heidelberg (2010)
9. Cool, D., Keromytis, A.D.: Conversion and proxy functions for symmetric key ciphers. In: International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II. vol. 1, pp. 662–667. IEEE (2005)
10. Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F.: Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE transactions on parallel and distributed systems 27(9), 2546–2559 (2016)
11. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: Applied Cryptography and Network Security. pp. 288–306. Springer (2007)
12. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Annual International Cryptology Conference. pp. 310–331. Springer, Heidelberg (2001)
13. Lewi, K., Montgomery, H., Raghunathan, A.: Improved constructions of PRFs secure against related-key attacks. In: International Conference on Applied Cryptography and Network Security. pp. 44–61. Springer, Cham (2014)
14. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: International Workshop on Public Key Cryptography. pp. 360–379. Springer, Heidelberg (2008)
15. Liu, Q., Wang, G., Wu, J.: Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. Information Sciences 258, 355–370. Elsevier (2014)
16. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 1–23. Springer, Heidelberg (2010)
17. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 35–54. Springer, Heidelberg (2013)
18. Naor, M., Pinkas, B., Reingold, O.: Distributed pseudo-random functions and KDCs. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 327–346. Springer, Heidelberg (1999)
19. Parra, J.R., Chan, T., Ho, S.W.: A noiseless key-homomorphic PRF: Application on distributed storage systems. In: Australasian Conference on Information Security and Privacy. pp. 505–513. Springer, Cham (2016)
20. Regev, O.: The learning with errors problem. Invited survey in CCC p. 15 (2010)
21. Rivest, R.L.: All-or-nothing encryption and the package transform. In: International Workshop on Fast Software Encryption. pp. 210–218. Springer, Heidelberg (1997)
22. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004)

23. Syalim, A., Nishide, T., Sakurai, K.: Realizing proxy re-encryption in the symmetric world. In: International Conference on Informatics Engineering and Information Science. pp. 259–274. Springer, Heidelberg (2011)
24. Xia, Z., Wang, X., Sun, X., Wang, Q.: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems 27(2), 340–352 (2016)
25. Xu, L., Wu, X., Zhang, X.: CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. pp. 87–88. ACM, New York (2012)
26. Zhangjie, F., Xingming, S., Qi, L., Lu, Z., Jiangang, S.: Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Transactions on Communications 98(1), 190–200 (2015)