Improved Proxy Re-encryption Scheme for Symmetric Key Cryptography

Amril Syalim[1], Takashi Nishide[2], and Kouichi Sakurai[3]

[1] Faculty of Computer Science, Universitas Indonesia
[2] Faculty of Engineering, University of Tsukuba
[3] Department Informatics, Kyushu University

Abstract

A proxy re-encryption scheme can be executed by a semi-trusted proxy, so that we can transform a ciphertext encrypted with a key to another ciphertext encrypted with different key without allowing the proxy to access the plaintext. A method to implement a secure proxy re-encryption is by first converting the plaintext to an intermediate form by using an all or nothing transform (AONT). In this paper, we describe an improved proxy re-encryption scheme for symmetric cipher by advocating the usage of a variant of the AONT function in the proxy re-encryption scheme. We show that the scheme secure under Chosen Plaintext Attack (CPA) for all possible types of attackers.

1 Introduction

In [1], the authors proposed a symmetric encryption scheme that supports proxy re-encryption by first transforming the plaintext into a random sequence using an all or nothing transform (AONT). However, the original security proof assumes that the AONT always produces random sequence of blocks, which is not applicable if the users who have access to the previous encryption keys (i.e. the previous users) are allowed to choose the plaintexts. In this paper, we propose an improved version by introducing the usage of a variant of Rivest' AONT. With this improvement we show that the scheme is secure under Chosen Plaintext Attack (CPA) for all types of attackers.

Some applications of the proxy re-encryption schemes are as follows:

1.1 Fast Key Update in an Encrypted Local Database

To protect his/her data, a user encrypts his/her local files or databases. When the key is unintentionally leaked, the user needs to update the encryption key. The simple method is by decrypting the data with the old key and then encrypting with the new key. The simple method is costly, especially for large data, because we need to execute decryption and encryption function. It is desirable to have a re-encryption function that is comparable to one or less encryption/decryption function.

1.2 Key Update in an Encrypted Data in Cloud

The user may store the data in a cloud for better accessibility and maintenance. To protect the stored data, the user encrypts the data and keeps the encryption keys in a secure storage. Because the data is accesible online, the user can also share the data by sharing the encryption key to another users. However, the problem is when the user needs to revoke the access, the user needs to update the key, which is by using the simple solution (decrypt and then encrypt), the user needs to download the data, decrypt with the old key, encrypt with the new key, and then upload the data to the cloud. A better solution is by securely delegating the re-encryption mechanism to a semi-trusted proxy in the cloud.

2 Related Work

All or nothing transform (AONT) was first proposed by Rivest [2] to convert n blocks message $M = m[1], ..., m[n]$ into a pseudo message $M' = m[1]', ..., m[s]'$ for $s > n$ so that the original message cannot be recovered if any block of the pseudo message is missing. Rivest proposed a concrete AONT's scheme by encrypting the blocks and hiding the encryption key in a block that is produced by xor-ing all encrypted blocks with the encryption key.

The Rivest scheme is similar to the Optimal Asymmetric Encryption Padding (OAEP) proposed by Bellare et.al. [3]. Boyko proved that OAEP is secure in several forms of security definition including adaptive semantic security [4]. Canetti et al. proposed some All-or-Nothing Transform schemes [5] while Dodis el al. proposed exposure resilient cryptography [6]. Desai proposed CTRT, that is a construction of AONT similar to Rivest's scheme based on CTR mode of encryption [7]. Stinson provided an AONT scheme in the context of unconditional security [8]. More recently, Boneh et al. proposed key homomorphic Pseudorandom Function (PRF) that is secure in the standard model [9].

There are some schemes proposed for direct transformation and proxy re-encryption in the asymmetric key encryption setting [10, 11, 12, 13, 14]. A paper by Blaze et al. [14] proposes a bidirectional proxy re-encryption based on ElGamal's encryption. The scheme in [10, 11, 12, 13] improve the Blaze et al.'s scheme. An example is the work of Atenise et al. [10] that uses pairing.

Cook et al. proposed a solution for proxy re-encryption for symmetric ciphers by using double encryption [15]. Another related work is the conversion method for Galois Counter Mode (GCM) mode [16]. It is not intended for proxy re-encryption but rather for fast re-encryption. In this method, the re-encryption needs to execute two encryptions, so that this mechanism is not much more efficient than decrypt-then-encrypt approach.

In [1], the authors proposed a symmetric encryption scheme that supports proxy re-encryption by first transforming the plaintext into the pseudomessage using an all or nothing transform (AONT). However, the authors proved the scheme secure by assuming the AONT has some special characteristics which is not applicable in all attack models. In this paper, we show a security proof which only needs the assumption about the security of encryption and hash functions used by AONT.

3 Preliminaries

Let $m[1], m[2], ..., m[n]$ be a sequence of n blocks message where the size of each $m[i]$ is l bits.The encryption algorithm $\mathcal{PR}$ is a quintiple algorithms = $(\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{RG}, \mathcal{RE})$ where [1]:

- $\mathcal{G}$ is a key generation algorithm that produces the keys to be used by $\mathcal{E}$ and $\mathcal{D}$
- $\mathcal{E}$ is the encryption algorithm that transforms n input blocks $m[1], m[2], ..., m[n]$ to s output ciphertext blocks $c[1], c[2], ..., c[s]$ for $s \geq n$
- $\mathcal{D}$ is the decryption algorithm that transforms the ciphertext $c[1], c[2], ..., c[s]$ into the plaintext $m[1], m[2], ..., m[n]$
- $\mathcal{RG}$ is an algorithm to generate keys to be used by the re-encryption algorithm $\mathcal{RE}$
- $\mathcal{RE}$ is the re-encryption algorithm that transforms the ciphertext $c[1]^A, c[2]^A, ..., c[s]^A$ encrypted with private key K_A into ciphertext $c[1]^B, c[2]^B, ..., c[s]^B$ encrypted with private key K_B

Notion of Security. Security of the symmetric encryption scheme is defined in term of Left-or-Right Indistinguishability [17] as follows .

Definition 1 (LOR-CPA [17]) *Let $\mathcal{SE} = \mathcal{K}, \mathcal{E}, \mathcal{D}$ be a symmetric encryption scheme. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let A_{cpa} be an adversary that has access to the oracle $\mathcal{E}_K(\cdot)$. We consider the following experiment:*

Experiment $\boldsymbol{Exp}^{lor-cpa-b}_{\mathcal{SE},A_{cpa}}(k)$*:*

$K \xleftarrow{R} \mathcal{K}(k)$
$d \leftarrow A^{\mathcal{E}_K(\cdot)}_{cpa}(k)$
return d

We define the advantage of the adversaries as:

$$\begin{aligned} \mathbf{Adv}^{lor-cpa-b}_{\mathcal{SE}}(A) &= \Pr\left[\mathbf{Exp}^{lor-cpa-1}_{\mathcal{SE},A_{cpa}}(k) = 1\right] \\ &- \Pr\left[\mathbf{Exp}^{lor-cpa-0}_{\mathcal{SE},A_{cpa}}(k) = 1\right] \end{aligned}$$

Definition 2 (Pseudorandom function family) *Let F be a pseudorandom function, the advantage of an adversary B to distinguish the outputs of F to a random function R is defined as follows:*

$$\begin{aligned} \mathbf{Adv}^{prf}_{F}(B) &= \Pr[K_i \xleftarrow{\$} K : B^{F_{K_i}(\cdot)} = 1] \\ &- \Pr[R \xleftarrow{\$} \mathcal{F}^{l,L} : B^{R(\cdot)} = 1] \end{aligned}$$

Definition 3 (Pseudorandom permutation family) *Let F be a pseudorandom permutation, the advantage of an adversary B to distinguish the outputs of F to a random permutation P is defined in the following formula:*

$$\begin{aligned}\mathbf{Adv}_F^{prp}(B) &= \Pr[K_i \xleftarrow{\$} K : B^{F_{K_i}(\cdot)} = 1] \\ &- \Pr[P \xleftarrow{\$} \mathcal{P}^l : B^{P(\cdot)} = 1]\end{aligned}$$

Lemma 1 (Difference Lemma [18, 19]) *Let A, B, F be events defined in some probability distribution, and if $A \wedge \neg F \iff B \wedge \neg F$, then $|Pr[A] - Pr[B] \leq Pr[F]$*

Proof:

$$\begin{aligned}Pr[A] - Pr[B]| &= |Pr[A \wedge F] + Pr[A \wedge \neg F] \\ &\quad -Pr[B \wedge F] - Pr[A \wedge \neg F] \\ &= |Pr[A \wedge F] - Pr[B \wedge F]| \\ &\leq Pr[F]\end{aligned}$$

4 The Primitives

4.1 All or Nothing Transform (AONT)

We propose a variant of Rivest's AONT as shown in the following algorithm ($\mathcal{E}$-AONTH and $\mathcal{D}$-AONTH). The main difference to the original Rivest scheme is: we include another "pass" so that the attacker cannot control the outputs of AONT even if he/she knows the key K'. We also use a hash function in the second "pass" rather than an encryption with a "fixed key". The differences are highlighted by boxes.

Algorithm $\mathcal{E}$-AONTH$^{F,F^{-1},H}(ctr, m[1]...m[n])$

$K' \xleftarrow{\$} \{0,1\}^l$
for $i = 1$ to n **do**
  $x[i] \leftarrow m[i] \oplus F_{K'}(ctr + i)$
end for
$m'[n+1] = K' \oplus \boxed{H(x[1]...x[n])}$
for $i = 1$ to n **do**
  $\boxed{m'[i] \leftarrow x[i] \oplus H(m'[n+1] \oplus (ctr + i))}$
end for
return $m'[1]...m'[n+1]$

Algorithm $\mathcal{D}$-AONTH$^{F,F^{-1},H}(ctr, m'[1]...m'[n+1])$

for $i = 1$ to n **do**
  $\boxed{x[i] \leftarrow m'[i] \oplus H(m'[n+1] \oplus (ctr + i))}$
end for
$K' = m'[n+1] \oplus \boxed{H(x[1]...x[n])}$
for $i = 1$ to n **do**
  $m[i] \leftarrow x[i] \oplus F_{K'}(ctr + i)$
end for
return $m[1]...m[n]$

4.2 The functions $\mathcal{PE}, \mathcal{DP}, \mathcal{FC}$, and $\mathcal{PG}$

The functions $\mathcal{PE}, \mathcal{DP}, \mathcal{FC}$ are exactly the same as in [1]. The permutation $\mathcal{PE}$ transforms the second input permuting the sequence according to the first input sequence

(permutation key). The inverse of permutation $\mathcal{DP}$ converts the sequence that has been permuted using the $\mathcal{PE}$ to the original form. $\mathcal{PG}$ finds a conversion key between two permutations.

Algorithm $\mathcal{PE}_{p[1]...p[n]}(x[1]...x[n])$
  for $i = 1$ to n **do**
    $x'[i] \leftarrow x[p[i]]$
  end for
  return $x'[1]...x'[n]$

Algorithm $\mathcal{DP}_{p[1]...p[n]}(x[1]...x[n])$
  for $i = 1$ to n **do**
    $x'[p[i]] = x[i]$
  end for
  return $x'[1]...x'[n]$

Algorithm $\mathcal{FC}(p_A[1]...p_A[n], p_B[1]...p_A[n])$
  for $i = 1$ to n **do**
    for $j = 1$ to n **do**
      if $p_A[i] = p_B[j]$ **then**
        $p_C[j] \leftarrow i$
        break
      end if
    end for
  end for
  return $p_C[1]...p_C[n]$

The permutation key generator $\mathcal{PG}$ generates a permutation key that consists of a sequence of distinct numbers from 1 to n. Our scheme uses an algorithm that takes an input a key k and a number n. The algorithm generates the random sequence by sorting output of an encryption function $F_K()$.

```
Algorithm PG_K(n)
  for i = 1 to n do
    p[i] = i
    tmp[i] = F_K(i)
  end for
  QuickSort(p, tmp, 1, n)
  return p
  Function QuickSort(p, tmp, lo, hi)
  if lo < hi then
    q ←Partition(p, tmp, lo, hi)
    QuickSort(p, tmp, lo, q − 1)
    QuickSort(p, tmp, q + 1, hi)
  end if
  EndFunction
  Function Partition(p, tmp, lo, hi)
  pivot ← tmp[hi]
  i ← lo
  for j = lo to hi − 1 do
    if tmp[j] ≤ pivot then
      swap tmp[i] with tmp[j]
      swap p[i] with p[j]
      i ← i + 1
    end if
  end for
  swap tmp[i] with tmp[hi]
  swap p[i] with p[hi]
  return i
  EndFunction
```

5 The Proposed Scheme

We describe the improved scheme in the algorithm $\mathcal{PR} = (\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{RG}, \mathcal{RE})$ that works on $l \times n$ bits message $m[1]...m[n]$ where the message is divided into n blocks with size l. The key generator $\mathcal{G}$ produces three random permutation keys P_1, P_2, P_3 that are later used by either $\mathcal{E}$ and $\mathcal{D}$ for encryption and decryption. Encryption algorithm $\mathcal{E}$ works by first converting the plaintext into AONT's pseudomessage, and then uses three permutation keys P_1, P_2, P_3 to produce the ciphertext with a random initialization vector (iv). The decryption algorithm $\mathcal{D}$ is the inverse of the $\mathcal{E}$. The reencryption key generator $\mathcal{RG}$ produces the reencryption keys that are later used by the rencryption function $\mathcal{RE}$ to update the encryption key.

Algorithm $\mathcal{G}(g, K_1, K_2, K_3, l, n)$

if g **then**
$K_1 \xleftarrow{\$} \{0,1\}^l$
$K_2 \xleftarrow{\$} \{0,1\}^l$
$K_3 \xleftarrow{\$} \{0,1\}^l$
end if
$P_1 \leftarrow \mathcal{PG}_{K_1}(l)$
$P_2 \leftarrow \mathcal{PG}_{K_2}(l)$
$P_3 \leftarrow \mathcal{PG}_{K_3}(n)$
return P_1, P_2, P_3

Algorithm $\mathcal{E}(K_1, K_2, K_3, ctr, m[1]...m[n], l, n)$

$(P_1, P_2, P_3) \leftarrow \mathcal{G}(0, K_1, K_2, K_3, l, n)$
$iv \xleftarrow{\$} \{0,1\}^l$
$m'[1]...m'[n+1] \leftarrow \mathcal{E}\text{-AONTH}(ctr, m[1]...m[n])$
$m''[1]...m''[n] \leftarrow \mathcal{PE}_{P_3}(m'[1]...m'[n])$
$c[0] \leftarrow \mathcal{PE}_{P_1}(m'[n+1][1...l]) \oplus \mathcal{PE}_{P_2}(iv[1...l])$
for $i = 1$ to n **do**
$c[i] \leftarrow (\mathcal{PE}_{P_1}(m''[i][1...l])$
$\oplus \mathcal{PE}_{P_2}(c[i-1][1...l]))$
end for
return $iv, c[0]...c[n]$

Algorithm $\mathcal{D}(K_1, K_2, K_3, ctr, iv, c[0]..., c[n], l, n)$

$(P_1, P_2, P_3) \leftarrow \mathcal{G}(0, K_1, K_2, K_3, l, n)$
for $i = n$ to 1 **do**
$m[i]'' \leftarrow \mathcal{DP}_{P_1}(c[i] \oplus \mathcal{PE}_{P_2}(c[i-1][1...l]))$
end for
$m'[n+1] = \mathcal{DP}_{P_1}(c[0][1...l] \oplus \mathcal{PE}_{P_2}(iv[1...l]))$
$m'[1]...m'[n] \leftarrow \mathcal{DP}_{P_3}(m''[1]...m''[n])$
$m[1]...m[n] \leftarrow \mathcal{D}\text{-AONTH}(ctr, m'[1]...m'[n+1])$
return $m[1]...m[n]$

Algorithm $\mathcal{RG}(K_1, K_2, K_3)$

$(P_1, P_2, P_3) \leftarrow \mathcal{G}(0, K_1, K_2, K_3, l, n)$
$(P_1', P_2', P_3') \leftarrow \mathcal{G}(1, 0, 0, 0, l, n)$
$CK_1 \leftarrow \mathcal{FC}(P_1, P_1')$
$CK_3 \leftarrow \mathcal{FC}(P_3, P_3')$
return CK_1, K_2, K_2', CK_3

Algorithm $\mathcal{RE}(CK_1, K_2, K_2', CK_3, iv, c[0]...c[n], l, n)$
$P_2 \leftarrow \mathcal{PG}_{K_2}(l)$, $P_2' \leftarrow \mathcal{PG}_{K_2'}(l)$
for $i = n$ to 1 **do**
$c'[i] \leftarrow \mathcal{PE}_{CK_1}(c[i] \oplus \mathcal{PE}_{P_2}(c[i-1][1...l]))$
end for
$c''[1]...c''[n] \leftarrow \mathcal{PE}_{CK_3}(c'[1]...c'[n])$
$c''[0] = \mathcal{PE}_{CK_1}(c[0] \oplus \mathcal{PE}_{P_2}(iv[1...l]))$
$\oplus \mathcal{PE}_{P_2'}(iv[1...l]))$
for i = 1 to n **do**
$c[i] \leftarrow c[i]'' \oplus \mathcal{PE}_{P_2'}(c''[i-1][1...l])$
end for
return $iv, c[0]...c[n]$

Correctness of the Decryption and Re-encryption Functions. It is easy to check that the decryption $\mathcal{D}$ is the inverse of the encryption function $\mathcal{E}$. As shown in [1], the re-encryption algorithm $\mathcal{RE}$ correctly converts a ciphertext encrypted with keys P_1, P_2, P_3 to new keys P_1', P_2', P_3'.

6 Security Analysis

As described in [1], there are three types of attackers: the outsiders, the previous users and the proxy. The authors in [1] argued about the security of the scheme by assuming that the AONT always produces random sequence, which can be false under some circumstances, for example when the previous users is allowed to choose the plaintexts (which is in the original proof [1] is not allowed).

6.1 Security of Encryption

First, we analyze the security of the encryption scheme against the attackers who have no access to any keys (outputs of $\mathcal{G}$ and $\mathcal{RG}$). In the Left-or-Right Indistinguishability, the adversary chooses two message blocks $M[0], M[1]$. The encryption oracle encrypt these blocks, and the adversary should distinguish whether the encrypted blocks belong to the left or right world.

Theorem 1 *Let F in $\mathcal{AONT}$ defined in Section 4.1 be a pseudorandom function and Let $\mathcal{PR}$ be the symmetric encryption that support proxy re-encryption defined in Section 5. The advantages of an adversary A attacking $\mathcal{PR}$ in LOR-CPA as defined in Section 3 is at most:*

$$\mathsf{Adv}^{\mathsf{lor-cpa}}_{\mathcal{PR}}(A) \leq 2 \cdot \mathsf{Adv}^{prf}_{F}(B)$$

Proof 1 *Game based proof.*

G0. *This game represents the original game. In each query q_i, the adversary A chooses two n blocks plaintexts $(M_i[0], M_i[1])$ and given access to the encryption oracle. The oracle encrypts the plaintexts and return the ciphertext, the adversary should guess whether the ciphertexts belong to the left (0) or right (1).*

$P_1 \xleftarrow{\$} P, P_2 \xleftarrow{\$} P, P_3 \xleftarrow{\$} P$
$r \xleftarrow{\$} \{0,1\}, b \xleftarrow{\$} \{0,1\}, K' \xleftarrow{\$} \{0,1\}^l, iv \xleftarrow{\$} \{0,1\}$
for $i \leftarrow 1...q$ ***do***
  $(M_i[0], M_i[1]) \leftarrow A(r, C_1, ..., C_{i-1})$
  $m_i[1]...m_i[n] \leftarrow M_i[b]$
  for $j = 1$ *to* n ***do***
    $\boxed{x[j] \leftarrow m[j] \oplus F_{K'}(i \cdot n + j)}$
  end for
  $\boxed{m'[n+1] \leftarrow K' \oplus H(x[1]...x[n])}$
  for $j = 1$ *to* n ***do***
    $\boxed{m'[j] \leftarrow x[j] \oplus H(m'[n+1] \oplus (i \cdot n + j))}$
  end for
  $\boxed{c_i[0] \leftarrow \mathcal{PE}_{P_1}(m'_i[n+1]) \oplus \mathcal{PE}_{P_2}(iv[1...l])}$
  for $j = 1$ *to* n ***do***
    $\boxed{c_i[j] \leftarrow \mathcal{PE}_{P_1}(m'_i[P_3[j]]) \oplus \mathcal{PE}_{P_2}(c_i[j-1])}$
  end for
  $C_i \leftarrow c_i[0]...c_i[n]$
end for
$d \leftarrow A(r, C_1, ..., C_q)$
return $(b = d)$

G1. *In this game, we assume F is pseudorandom function with a PRF advantage* $Adv^{prf}_{F_i}(B)$. *Because the input to H is random, the outputs of H are also random.*

$P_1 \xleftarrow{\$} P, P_2 \xleftarrow{\$} P, P_3 \xleftarrow{\$} P$
$r \xleftarrow{\$} \{0,1\}, b \xleftarrow{\$} \{0,1\}, K' \xleftarrow{\$} \{0,1\}^l, iv \xleftarrow{\$} \{0,1\}$
for $i \leftarrow 1...q$ ***do***
  $(M_i[0], M_i[1]) \leftarrow A(r, C_1, ..., C_{i-1})$
  $m_i[1]...m_i[n] \leftarrow M_i[b]$
  for $j = 1$ *to* n ***do***
    $\boxed{x[j] \xleftarrow{\$} \{0,1\}^l}$
  end for
  $\boxed{m'[n+1] \xleftarrow{\$} \{0,1\}^l}$
  for $j = 1$ *to* n ***do***
    $\boxed{m'[j] \xleftarrow{\$} \{0,1\}^l}$
  end for
  $\boxed{c_i[0] \leftarrow \mathcal{PE}_{P_1}(m'_i[n+1]) \oplus \mathcal{PE}_{P_2}(iv[1...l])}$
  for $j = 1$ *to* n ***do***
    $\boxed{c_i[j] \leftarrow \mathcal{PE}_{P_1}(m'_i[P_3[j]]) \oplus \mathcal{PE}_{P_2}(c_i[j-1])}$
  end for
  $C_i \leftarrow c_i[0]...c_i[n]$
end for
$d \leftarrow A(r, C_1, ..., C_q)$
return $(b = d)$

Let $Pr[G^A_0 \Rightarrow 1]$ *be the probability that the adversary correctly guess the left or right oracle, then:*

$$\boldsymbol{Adv}^{lor-cpa-b}_{\mathcal{PR}}(A) = 2 \cdot Pr[G^A_0 \Rightarrow 1] - 1$$

Proof:

$$
\begin{aligned}
&Pr[G_0^A \Rightarrow 1] \\
&= Pr[G_0^A \Rightarrow 1|b = 1] \cdot \frac{1}{2} + Pr[G_0^A \Rightarrow 1|b = 0] \cdot \frac{1}{2} \\
&= Pr[G_0^A \Rightarrow 1|b = 1] \cdot \frac{1}{2} + \left(1 - Pr[G_0^A \Rightarrow 0|b = 0] \cdot \frac{1}{2}\right) \\
&= \frac{1}{2} + \frac{1}{2} \cdot (Pr[G_0^A \Rightarrow 1|b = 1] - Pr[G_0^A \Rightarrow 0|b = 0]) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \left(\boldsymbol{Adv}_{\mathcal{SE}}^{lor-cpa-b}(A)\right)
\end{aligned}
$$

*Because in game **G1** the ciphertext is produced randomly, then*

$$Pr[G_1^A \Rightarrow 1] = \frac{1}{2}$$

and

$$
\begin{aligned}
&\mathsf{Adv}_{\mathcal{PR}}^{lor-cpa}(A) = 2 \cdot Pr[G_0^A \Rightarrow 1] - 1 \\
&\leq 2 \cdot \left(Pr[G_1^A \Rightarrow 1] + \mathsf{Adv}_F^{prf}(B)\right) - 1 \\
&\leq 2 \cdot \mathsf{Adv}_F^{prf}(B) + 2 \cdot \frac{1}{2} - 1 \leq 2 \cdot \mathsf{Adv}_F^{prf}(B)
\end{aligned}
$$

6.2 Security Against Previous Users

The previous users, that is the attacker who have access to previous outputs of key generation $\mathcal{G}$ before re-encryption. The difference of the previous user to the normal attacker is he/she may also have access to all plaintext M_i (from previous accesses), intermediate values represented by AONT(M_i), and the key (K') used by AONT. However, the previous user cannot access any outputs of $\mathcal{RG}$ (which can only be accessed by the proxy).

Theorem 2 *Let F in $\mathcal{AONT}$ defined in Section 4.1 be a pseudorandom function and Let $\mathcal{PR}$ be the symmetric encryption that support proxy re-encryption defined in Section 5. The advantages of an adversary A, who have access to previous encryption keys, attacking $\mathcal{PR}$ in LOR-CPA as defined in Section 3 is at most:*

$$
\begin{aligned}
\mathsf{Adv}_{\mathcal{PR}}^{\mathsf{lor-cpa}}(A) \leq \mathsf{Adv}_H^{prf}(B) &\left(2 + \frac{q(q-1)}{2^l}\right) \\
&+ \frac{q(q-1)}{2^l}\left(\frac{1}{2^l} + 3(n+1)\right)
\end{aligned}
$$

where q is the number of queries, n is the number of blocks in each query, and l is the size of each block.

Proof 2 *Game based proof.*

G0/G1. *In this game, the adversary can choose $x[j]$ because it know the encryption key K', but the adversary cannot choose $m'[j]$. To attack the scheme, first that adversary try to distinguish the ciphertext by checking $c[0]$. The adversary needs to find three plaintexts $x_i[1]...x_i[n]$, $x_k[1]...x_k[n]$, $x_{k'}[1]...x_{k'}[n]$ so that $m'_i[n+1] = m'_k[n+1] \oplus m'_{k'}[n+1] \oplus K'$ AND $iv_i = iv_k \oplus iv_{k'}$ for a fixed K', $k < i$, and $k' < i$. The probability of such condition is $\frac{q(q-1)}{2^{l+1}} \cdot \left(\boldsymbol{Adv}_H^{prf}(B) + \frac{1}{2^l}\right)$.*

$P_1 \xleftarrow{\$} P, K_2 \xleftarrow{\$} \{0,1\}^l, P_2 \xleftarrow{\$} P, P_3 \xleftarrow{\$} P$
$r \xleftarrow{\$} \{0,1\}, b \xleftarrow{\$} \{0,1\}, iv \xleftarrow{\$} \{0,1\}, K' \xleftarrow{\$} \{0,1\}^l$
for $i \leftarrow 1...q$ ***do***
  $(M_i[0], M_i[1]) \leftarrow A(r, C_1, ..., C_{i-1})$
  $m_i[1]...m_i[n] \leftarrow M_i[b]$
  for $j = 1$ *to* n ***do***
    $x[j] \leftarrow \{0,1\}^l$ // *the adversary can choose* $x[j]$, *and*
    $m[j] \leftarrow x[j] \oplus F_{K'}(i \cdot n + j)$ // *compute the associated* $m[j]$
  end for
  $m'[n+1] \leftarrow K' \oplus H(x[1]...x[n])$
  for $j = 1$ *to* n ***do***
    $m'[j] \leftarrow x[j] \oplus H(m'[n+1] \oplus (ctr + i))$
  end for
  $c_i[0] \leftarrow \mathcal{PE}_{P_1}(m'_i[n+1]) \oplus \mathcal{PE}_{P_2}(iv[1...l])$
  for $j = 1$ *to* n ***do***
    $c_i[j] \leftarrow \mathcal{PE}_{P_1}(m'_i[P_3[j]]) \oplus \mathcal{PE}_{P_2}(c_i[j-1])$
  end for
  $C_i \leftarrow c_i[0]...c_i[n]$
end for
$d \leftarrow A(r, C_1, ..., C_q)$
return $(b = d)$

In game ***G1****, we assume the probability of the above conditions is 0, so that*

$$Pr[G_0^A \Rightarrow 1] - Pr[G_1^A \Rightarrow 1] \leq \frac{q(q-1)}{2^{l+1}} \left(\boldsymbol{Adv}_H^{prf}(B) + \frac{1}{2^l}\right)$$

G2. *Because $c_i[j] \leftarrow \mathcal{PE}_{P_1}(m'_i[P_3[j]]) \oplus \mathcal{PE}_{P_2}(c_i[j-1])$, in each query q_i, the adversary needs to find the collision $c_k[j] = c_i[j'](0 \leq j \leq n)$, so that he/she can deduce $\mathcal{PE}_{P_1}(m'_k[P_3[j+1]] \oplus m'_i[P_3[j'+1]])$. This information can be used to deduce P_1 by checking the number of 1s or 0s bits of $m'_k[P_3[j+1]] \oplus m'_i[P_3[j'+1]]$. The possibility of collision is defined in the event "*bad*".*

$P_1 \xleftarrow{\$} P, K_2 \xleftarrow{\$} \{0,1\}^l, P_2 \xleftarrow{\$} P, P_3 \xleftarrow{\$} P$
$r \xleftarrow{\$} \{0,1\}, b \xleftarrow{\$} \{0,1\}, iv \xleftarrow{\$} \{0,1\}^l, K' \xleftarrow{\$} \{0,1\}^l$
for $i \leftarrow 1 \ldots q$ ***do***
  $(M_i[0], M_i[1]) \leftarrow A(r, C_1, \ldots, C_{i-1})$
  $m_i[1] \ldots m_i[n] \leftarrow M_i[b]$
  for $j = 1$ *to* n ***do***
    $x[j] \leftarrow \{0,1\}^l$ *// the adversary can choose* $x[j]$*, and*
    $m[j] \leftarrow x[j] \oplus F_{K'}(i \cdot n + j)$ *// compute the associated* $m[j]$
  end for
  $H_{i,0} \xleftarrow{\$} \{0,1\}^l$
  $\boxed{m'[n+1] \leftarrow K' \oplus H_{i,0}}$
  for $j = 1$ *to* n ***do***
    $H_{i,j} \xleftarrow{\$} \{0,1\}^l$
    $\boxed{m'[j] \leftarrow x[j] \oplus H_{i,j}}$
  end for
  $\boxed{c_i[0] \leftarrow \mathcal{PE}_{P_1}(m'_i[n+1]) \oplus \mathcal{PE}_{P_2}(iv[1 \ldots l])}$
  if $c_i[0] \in S$ ***then***
    bad $\leftarrow$ true
  end if
  $S \leftarrow S \cup \{c_i[0]\}$
  for $j = 1$ *to* n ***do***
    $\boxed{c_i[j] \leftarrow \mathcal{PE}_{P_1}(m'_i[P_3[j]]) \oplus \mathcal{PE}_{P_2}(c_i[j-1])}$
    if $c_i[j] \in S$ ***then***
      bad $\leftarrow$ true
    end if
    $S \leftarrow S \cup \{c_i[j]\}$
  end for
  $C_i \leftarrow c_i[0] \ldots c_i[n]$
end for
$d \leftarrow A(r, C_1, \ldots, C_q)$
return $(b = d)$

In game ***G2****, we change the output of H to random output so that*

$$Pr[G_2^A \Rightarrow 1] = \frac{1}{2}$$

and

$$Pr[G_1^A \Rightarrow 1] - Pr[G_2^A \Rightarrow 1] \leq \boldsymbol{Adv}_H^{prf}(B) + Pr[G_2^A \textit{sets bad}]$$

Because each $m'_i[j]$ *is produced randomly and* $\mathcal{PE}$ *is a permutation, the probability of collision is the total probability of bad events, that is the collision of* $m'_i[n+1], iv_i, m'_i[n+1] \oplus iv_i, m'_i[j], c_j$ *and* $m'_i[j] \oplus c_j$*. Because for each query, the algorithm produces three blocks* $m'_i[n+1], iv_i, m'_i[n+1] \oplus iv_i$ *and* $3n$ *blocks* $m'_i[j], c_j, m'_i[j] \oplus c_j$ *totaling* $3(n+1)$ *blocks, the probability of collision is:*

$$\begin{aligned} & Pr[G_2^A \textit{sets bad}] \\ & \leq \frac{3(n+1) + 2 \cdot 3(n+1) + \ldots + (q-1) \cdot 3(n+1)}{2^l} \\ & \leq 3(n+1)\frac{q(q-1)}{2^{l+1}} \end{aligned}$$

So, that

$$Pr[G_1^A \Rightarrow 1] - Pr[G_2^A \Rightarrow 1] \leq \boldsymbol{Adv}_H^{prf}(B) + 3(n+1)\frac{q(q-1)}{2^{l+1}}$$

And,

$$\begin{aligned}
&\mathsf{Adv}_{\mathcal{PR}}^{lor-cpa}(A) = 2 \cdot Pr[G_0^A \Rightarrow 1] - 1 \\
&\leq 2 \cdot \left(Pr[G_1^A \Rightarrow 1] + \frac{q(q-1)}{2^{l+1}} \left(\boldsymbol{Adv}_H^{prf}(B) + \frac{1}{2^l} \right) \right) - 1 \\
&\leq 2 \cdot \frac{q(q-1)}{2^{l+1}} \left(\boldsymbol{Adv}_H^{prf}(B) + \frac{1}{2^l} \right) \\
&+ 2 \cdot \left(\boldsymbol{Adv}_H^{prf}(B) + 3(n+1)\frac{q(q-1)}{2^{l+1}} + Pr[G_2^A \Rightarrow 1] \right) - 1 \\
&\leq 2 \cdot \frac{q(q-1)}{2^{l+1}} \left(\boldsymbol{Adv}_H^{prf}(B) + \frac{1}{2^l} \right) \\
&+ 2 \cdot \left(\boldsymbol{Adv}_H^{prf}(B) + 3(n+1)\frac{q(q-1)}{2^{l+1}} \right) + 2 \cdot \frac{1}{2} - 1 \\
&\leq \boldsymbol{Adv}_H^{prf}(B) \left(2 + \frac{q(q-1)}{2^l} \right) + \frac{q(q-1)}{2^l} \left(\frac{1}{2^l} + 3(n+1) \right)
\end{aligned}$$

6.3 Security Against Proxy

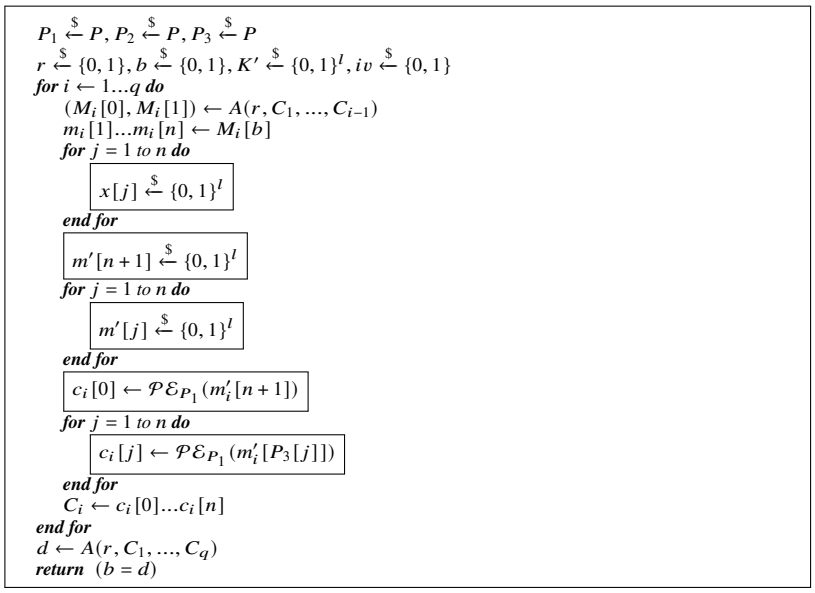
The proxy, that is the attacker who have access to the outputs of $\mathcal{RG}$, but he/she has no access to any outputs of $\mathcal{G}$ and the intermediate values AONT(M_i). The data owner can delegate the re-encryption process to the proxy by providing the re-encryption keys (output of $\mathcal{RG}$) without the need to provide any access to the plaintext and encryption keys.

Theorem 3 *Let F in $\mathcal{AONT}$ defined in Section 4.1 be a pseudorandom function and let $\mathcal{PR}$ be the symmetric encryption that support proxy re-encryption defined in Section 5. The advantages of an adversary A, who have access the re-encryption keys (output of $\mathcal{RG}$), attacking $\mathcal{PR}$ in LOR-CPA as defined in Section 3 is at most:*

$$\mathsf{Adv}_{\mathcal{PR}}^{\mathsf{lor-cpa}}(A) \leq 2 \cdot \mathsf{Adv}_F^{prf}(B)$$

Proof 3 *Game based proof.*

- ***G0***. *In **G0** is similar with the proof in Theorem 1.*
- ***G1***. *In **G1**, F produces random outputs, and we remove $\mathcal{PE}_{P_2}$ because the proxy know the keys P_2 and P_2'.*



Because now the ciphertext is produced randomly, then

$$Pr[G_1^A \Rightarrow 1] = \frac{1}{2}$$

and

$$\begin{aligned}
&\mathsf{Adv}^{lor-cpa}_{\mathcal{PR}}(A) = 2 \cdot Pr[G_0^A \Rightarrow 1] - 1 \\
&\leq 2 \cdot \left(Pr[G_1^A \Rightarrow 1] + \mathsf{Adv}^{prf}_F(B)\right) - 1 \\
&\leq 2 \cdot \mathsf{Adv}^{prf}_F(B) + 2 \cdot \frac{1}{2} - 1 \\
&\leq 2 \cdot \mathsf{Adv}^{prf}_F(B)
\end{aligned}$$

7 Conclusion

In this paper, we have showed an improved proxy re-encryption scheme for the symmetric cipher. The main improvement is the usage of a new variant of AONT, so that the scheme can be proved secure by only assuming the security of the underlying encryption and hash functions used by the AONT function. We show security proofs under Chosen Plaintext Attack (CPA) for all possible type of attackers.

References

[1] A. Syalim, T. Nishide, and K. Sakurai, "Realizing proxy re-encryption in the symmetric world," in *International Conference on Informatics Engineering and Information Science*. Springer Berlin Heidelberg, 2011, pp. 259–274.

[2] R. L. Rivest, "All-or-nothing encryption and the package transform," in *In Fast Software Encryption, LNCS 1267*. Springer-Verlag, 1997, pp. 210–218.

[3] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in *EUROCRYPT*, 1994, pp. 92–111.

[4] V. Boyko, "On the security properties of oaep as an all-or-nothing transform," in *CRYPTO*, 1999, pp. 503–518.

[5] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai, "Exposure-resilient functions and all-or-nothing transforms," in *EUROCRYPT*, 2000, pp. 453–469.

[6] Y. Dodis, A. Sahai, and A. Smith, "On perfect and adaptive security in exposure-resilient cryptography," in *EUROCRYPT*, 2001, pp. 301–324.

[7] A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in *CRYPTO*, 2000, pp. 359–375.

[8] D. R. Stinson, "Something about all or nothing (transforms)," *Des. Codes Cryptography*, vol. 22, no. 2, pp. 133–138, 2001.

[9] D. Boneh, K. Lewi, H. W. Montgomery, and A. Raghunathan, "Key homomorphic prfs and their applications," in *CRYPTO (1)*, 2013, pp. 410–428.

[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.

[11] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 185–194.

[12] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Applied Cryptography and Network Security*. Springer, 2007, pp. 288–306.

[13] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1786–1802, 2011.

[14] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *In EUROCRYPT*. Springer-Verlag, 1998, pp. 127–144.

[15] D. L. Cook and A. D. Keromytis, "Conversion and proxy functions for symmetric key ciphers," in *ITCC*, 2005, pp. 662–667.

[16] S. Hirose, "On re-encryption for symmetric authenticated encryption," in *Computer Security Symposium (CSS) 2010*, 2010.

[17] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*. IEEE, 1997, pp. 394–403.

[18] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs." *IACR Cryptology ePrint Archive*, vol. 2004, p. 332, 2004.

[19] ——, "Oaep reconsidered," *Journal of Cryptology*, vol. 15, no. 4, pp. 223–249, 2002.