情報保護學會論文誌
第 23 卷 第 2 號, 2013. 4

조건값의 개수에 독립적인 조건부 프록시 재암호화 기법*

손 정 갑,[1†] 오 희 국,[1] 김 상 진[2‡]
[1]한양대학교, [2]한국기술교육대학교

A Single Re-encryption key based Conditional Proxy Re-Encryption Scheme*

Junggab Son,[1†] Heekuck Oh,[1] SangJin Kim[2‡]
[1]Hanyang University, [2]Korea University of Technology and Education

요 약

프록시 재암호화 기법은 데이터를 재암호화 하는 과정에서 평문이 노출되지 않는 장점이 있다. 이 기법은 최근 클라우드 컴퓨팅, 모바일 오피스 등 서버를 이용하는 환경에서 저장된 데이터를 안전하게 공유하기 위한 기법으로 각광받고 있다. 하지만 기존 기법은 재암호화 키를 반복적으로 사용할 수 있어 재암호화 오남용 문제가 발생한다. 이를 해결하기 위해 조건부 프록시 재암호화 기법이 제안되었지만 조건값의 수만큼 재암호화키를 생성해야하는 부담이 있다. 본 논문에서는 재암호화 키 생성 측면에서 효율적인 조건부 프록시 재암호화 기법을 제안하였다. 제안하는 기법은 조건값의 사용을 암호화와 복호화 과정으로 제한하여 조건값이 많아지더라도 재암호화 키를 추가로 생성하지 않아도 되는 장점이 있다. Weng 등의 기법이 재암호화 키 생성에 따른 시간 복잡도가 $O(n)$인 것에 비해, 제안하는 기법은 $O(1)$의 시간 복잡도를 가진다. 또한, 제안하는 기법은 선택적 암호문 공격에 안전하도록 설계되었다.

ABSTRACT

Proxy re-encryption scheme has advantage where plaintext does not get exposed during re-encryption process. This scheme is popular for sharing server-saved data in case of cloud computing or mobile office that uses server to save data. Since previous proxy re-encryption schemes can use re-encryption key over and over again, it may abuse re-encryption. To solve this problem, conditional proxy re-encryption scheme was proposed. But, it is computationally expensive generate the same number of re-encryption key with the number of condition values. In this paper, we propose an efficient conditional proxy re-encryption scheme in terms of re-encryption key generation. The proposed scheme uses only encryption and decryption process. Therefore it has advantage to generate one re-encryption key for one person. The proposed scheme is secure against chosen-ciphertext attack.

Keywords: Proxy re-encrpytion, conditional proxy re-encryption, data sharing

접수일(2012년 9월 7일), 수정일(2012년 12월 20일),
게재확정일(2012년 12월 23일)
* 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2012-R1A2A2A01046986).
* 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임 (No. 2012-R1A1A2009152).
† 주저자, jgson@infosec.hanyang.ac.kr
‡ 교신저자, sangjin@kut.ac.kr

I. 서 론

프록시 재암호화는 사용자 A의 공개키로 암호화된 암호문을 사용자 B의 개인키로 복호화할 수 있도록 암호문을 변환하는 기법이다[5]. 프록시는 재암호화 키를 이용하여 암호문을 재암호화 하며, 이 과정에서 기존의 암호문을 복호화 하지 않고 암호문을 변환할 수 있으므로 프록시는 평문이나 A의 개인키를 알지 못한다는 장점이 있다.

최근 클라우드 컴퓨팅, 스마트 오피스 등 서버를 통해 데이터를 저장하고 공유하는 환경이 많아지고 있다. 이러한 환경은 비용 절감의 측면에서 큰 장점을 지니게 되지만, 모든 데이터가 서버에 저장된다는 점은 보안 측면에서 단점으로 작용할 수 있다[1][2]. 서버에는 다양한 형태의 데이터가 저장되게 되는데 특히, 개인정보, 기업정보 등 적절한 보호기법을 적용해야 하는 민감한 데이터도 서버에 저장될 수 있다. 사용자는 데이터를 전송 및 저장하는 과정에서 민감한 데이터가 악의적인 외부 공격자에게 노출되는 것을 막기 위해 이를 암호화 한다. 적절한 암호기법으로 데이터를 암호화 하면, 외부 공격자가 데이터를 획득하더라도 이를 복호화할 수 없으므로 안전하다. 데이터를 공유하기 위한 기법으로는 그룹키, 속성기반 암호화 등 다양한 기법을 적용할 수 있지만, 가장 간단한 예로 암호화된 데이터를 복호화하여 다른 사용자의 키로 암호화한 후 전달하는 방법이 있다. 이 방법을 사용하면 암호문을 복호화하여 다시 암호화하는 과정에서 평문이 드러나므로 서비스 제공자에 의한 데이터의 노출은 막을 수 없다. 데이터의 소유자가 공유하고자 하는 사용자의 키로 데이터를 암호화하여 전송할 수 있지만 소유자가 부재일 경우 공유가 불가능하고, 공유 대상이 많을 경우 소유자가 암호화를 여러번 수행해야 한다는 단점이 있다. 프록시 재암호화기법은 이러한 환경에 대한 하나의 해결책이 될 수 있다[8]. 소유자는 다른 사용자에 대한 재암호화 키를 생성하여 서버에 저장하고, 공유하고자 하는 파일을 암호화하여 서버에 저장하면, 서버는 다른 사용자의 요청에 의해 데이터를 재암호화 하여 전송한다. 이 경우, 데이터 소유자는 자신의 키를 이용하여 데이터를 암호화한 후 서버에 전송하면 된다. 재암호화에 필요한 연산은 서버에 위임할 수 있으며, 공유를 위해 데이터를 재암호화하는 과정에서 서비스제공자가 원본 데이터를 확인할 수 없으므로 내부 공격자에게도 안전하다는 장점이 있다.

1997년 Mambo 등[6]에 의해 암호문의 복호 권한을 위임하는 방식이 제안되었다. Mambo 등의 기법은 A의 공개키로 암호화된 암호문을 B에 대한 암호문으로 변환한다. 하지만, A만 암호문을 변환할 수 있기 때문에 A가 다른 사용자들에 대한 암호문을 생성하는 단순한 방법과 차이가 없다. 1998년 Blaze 등은 사용자가 암호화된 데이터와 재암호화 키를 프록시에게 전달하면, 프록시가 이들을 이용하여 재암호화를 수행하는 방식을 제안하였다[7]. 이 기법은 A의 개인키와 B의 개인키를 이용하여 재암호화 키를 생성하기 때문에 개인키를 공유해야 하는 문제가 발생한다. 또한, 프록시와 B가 공모하면 다른 사용자의 재암호화 키를 생성할 수 있다. 2005년 Ateniese 등은 A의 개인키와 B의 공개키를 결합하여 재암호화 키를 생성하는 프록시 재암호화 기법을 제안하였다[8]. 이 기법은 재암호화 키를 생성하기 위해 개인키를 공유하지 않으며, 프록시와의 공모를 통해 다른 사용자의 재암호화키 생성을 방지할 수 있다.

지금까지의 프록시 재암호화 기법은 발행된 재암호화 키를 통한 재암호화를 제어할 수 없다. 즉, A가 B를 위한 재암호화 키를 발행한다는 것은 A의 모든 암호문을 B의 암호문으로 재암호화할 수 있다는 것을 의미한다. 이는 재암호화 발행 이전의 모든 암호문뿐만 아니라 발행 이후의 모든 암호문을 포함한다. 이러한 문제를 해결하기 위해 2011년 Weng 등은 조건값을 통해 재암호화를 제어할 수 있는 조건부 프록시 재암호화 기법(Conditional Proxy Re-encryption, CPRE)을 제안하였다[9]. CPRE 기법은 데이터 소유자가 조건값을 생성하여 재암호화 키 생성과정과 데이터 암호화 과정에서 생성한 조건값을 입력한다. 그러면, A가 발행한 암호문과 같은 조건이 포함된 재암호화 키를 사용하였을 때에만 원하는 결과를 얻을 수 있다. A는 데이터를 암호화할 때마다 다른 조건값을 입력함으로써 동일한 재암호화 키가 반복적으로 사용되는 것을 방지할 수 있다.이 기법을 사용하면 재암호화의 오남용 문제를 막을 수 있고, 조건값을 통해 복호화 권한을 제어할 수 있다. 하지만, 이 기법은 한 조건값에 대해 재암호화 키 한 개가 필요하다. 조건값을 변경할 때마다 재암호화 키를 새로 발급해야 하므로 결과적으로 재암호화키 생성 및 유지 측면에서의 오버헤드가 많이 발생하게 된다.

본 논문에서는 암호화, 복호화 과정에서만 조건값이 필요한 CPRE 기법을 제안한다. 제안하는 기법은 조건값을 사용하여 재암호화의 오남용을 방지할 수 있으며, 재암호화키를 최초 1회만 생성한다. 조건값이

바뀌더라도 재암호화 키를 다시 생성할 필요가 없으므로 이전 기법에 비해 조건값을 통해 복호화 권한을 효율적으로 제어할 수 있다. 논문의 구성은 다음과 같다. 2장에서 관련 연구에 대해 살펴보고 3장에서 제안하는 기법에 대해 서술한다. 4장에서 제안하는 기법을 분석한 후, 5장에서 결론을 맺는다.

II. 관련 연구

2.1절에서는 논문에서 사용하는 표기법과 기반 지식에 대해 서술한다.

2.1 표기법과 기반 지식

〔표 1〕은 이 논문에서 사용하는 표기법을 나타낸다.

〔표 1〕 표기법

표기	내용
q	k-bit 소수
Z_q	q를 위수로 갖는 군
G, G_T	q를 위수로 갖는 순환군
e	겹선형 쌍함수, $e: G\times G\rightarrow G_T$
pk_A, sk_A	사용자A의 공개키/개인키 쌍
m	메시지
w	조건값
C_A	A의 암호문
C_A^w	조건값 w가 포함된 A의 암호문
$RK_{A\rightarrow B}$	$A\rightarrow B$로의 재암호화키
$RK_{A\rightarrow B}^w$	조건w일 때 $A\rightarrow B$로의 재암호화키
s	Z_q에 속하는 임의의 값
g	G에 속하는 임의의 값
R	G_T에 속하는 임의의 값
H_1	해쉬 함수, $\{0,1\}^*\rightarrow Z_q$
H_2	해쉬 함수, $\{0,1\}^*\rightarrow G$
H_3	해쉬 함수, $G\rightarrow\{0,1\}^n$
H_4	해쉬 함수, $\{0,1\}^*\rightarrow G$
H_5	해쉬 함수, $G\rightarrow Z_q$

2.1.1 겹선형 맵(Bilinear Maps) [12]

G와 G_T가 같은 위수를 갖는 곱셉 순환군일 때, $e: G\times G\rightarrow G_T$는 다음을 만족해야 한다.

- Bilinearity
  $\forall g_1, g_2\in G, \forall a,b\in Z_q^*$에 대해서
  $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$를 만족해야 한다.
- Non-degeneracy
  $g_1, g_2\in G$에 대해서 $e(g_1, g_2)\neq 1$를 만족해야 한다.
- Computability
  $\forall g_1, g_2\in G$에 대해 $e(g_1, g_2)$를 효율적으로 계산하는 알고리즘이 존재한다.

결정적 겹선형 디피-헬만 가정 (Decisional Bilinear Diffie-Hellman assumption) 〔13〕

순환군 (G, G_T)에서 DBDH문제란 $(g, g^a, g^b, g^c, Q)\in G^4\times G_T$일 때, 알 수 없는 $a,b,c\in Z_q$에 대해 $Q = e(g,g)^{abc}$를 만족하는 Q를 찾는 것을 말한다. 다항시간 알고리즘을 B, 공격자의 이득을 ε라 할 때, 이 둘의 관계는 다음과 같아야 한다.

$$|\Pr[B(g, g^a, g^b, g^c, Q = e(g,g)^{abc}) = 1] - \Pr[B(g, g^a, g^b, g^c, Q = e(g,g)^d) = 1)]| \geq \varepsilon$$

이 때, 확률 Pr은 각각 Z_q에서 a,b,c,d를 임의로 선택할 확률과 G에서 임의로 g를 선택할 확률이며, B에서 임의의 비트를 소비할 확률을 말한다. 순환군 (G, G_T)에서 DBDH가정은 공격자가 DBDH문제를 해결하기 위해 B를 계산할 때 얻는 이득이 최소 ε보다 커야 한다는 것이다.

2.2 Blaze 등의 기법 〔7〕

1997년 Mambo가 암호문의 복호 권한을 위임하는 방식을 제안한 후, 이 기법이 A만 암호문을 변환할 수 있다는 점을 개선하여 프록시가 재암호화를 수행할 수 있도록 설계하였다. 프록시 역할은 연산 능력이 있는 사용자 또는 서버가 수행할 수 있으며, 프록시는 재암호화 키로 A의 암호문을 재암호화 함으로써 B의 암호문으로 변환할 수 있다. Blaze 등의 기법은 Elgamal 암호시스템에 기반을 두고 있으며, 다음과 같이 구성된다.

- 키 생성: 사용자 A에 대한 공개키/ 개인키 쌍을 생성하기 위해 $a\in_R Z_q$를 선택하고, $pk_A = g^a$, $sk_A = a$로 설정한다.
- 재암호화 키 생성: A는 B로부터 B의 개인키 b를 전송받아 A와 B사이의 재암호화 키를 생성한다.
  $RK_{A\rightarrow B} = b\cdot a^{-1} (\mathrm{mod} q)$
- 암호화: A는 $r\in_R Z_q$를 선택하고, 다음과 같이 m을 암호화한다.
  $C_A = (g^r\cdot m, g^{ar})$
- 재암호화: 프록시는 전달받은 C_A를 $RK_{A\rightarrow B}$로 재

암호화 한다.

$$C_B = (g^r \cdot m, (g^{ar})^{RK_{A\to B}}) = (g^r \cdot m, (g^{ar})^{ba^{-1}}) = (g^r \cdot m, g^{br})$$

- 복호화: B는 C_B를 자신의 개인키로 복호화한다.

$$g^r = (g^{br})^{b^{-1}}$$

$$m = \frac{g^r \cdot m}{g^r}$$

Blaze 등의 기법은 A와 B의 개인키를 이용하여 재암호화 키를 생성하므로 A의 암호문을 재암호화하여 B의 암호문으로 변환할 수 있으며, 반대로 B의 암호문을 재암호화하여 A의 암호문을 생성할 수도 있다. 프록시 재암호화 기법이 이러한 양방향성을 가지기 위해서는 재암호화 키를 두 사용자의 개인키를 이용하여 생성하여야 한다. 이는 개인키의 전달 및 공유 문제로 이어진다. 또한, 재암호화 키의 생성과정이 TTP (Trust Third Party)에 의해 안전하게 수행되었다고 가정하더라도 프록시와 B가 공모하여 B가 재암호화 키를 획득하면 B는 재암호화 키로부터 A의 개인키를 추출할 수 있다.

2.3 Ateniese 등의 기법 [8]

2005년 Ateniese등은 일방향성을 가지는 프록시 재암호화 기법을 제안하였다. 일방향성을 위해 암호화 단계를 1레벨 암호화와 2레벨 암호화로 나누어 설계하였다. 데이터 소유자는 데이터가 재암호화 되는 것을 원하지 않을 경우 1레벨 암호화를 사용한다. 재암호화를 통해 복호 권한의 위임을 허용하는 경우 2레벨 암호화를 사용한다. A의 2레벨 암호문을 $RK_{A\to B}$로 재암호화 하면 B의 1레벨 암호문이 되며, 이 특성 때문에 암호문을 반복적으로 재암호화 할 수 없다. 2레벨 암호문을 1레벨 암호문으로 변경하기 위해 겹선형 맵을 사용한다. Ateniese 등의 기법은 다음과 같다.

- 키 생성: 사용자 A에 대한 공개키/ 개인키 쌍을 생성하기 위해 $a \in_R Z_q$를 선택하고, $pk_A = g^a$, $sk_A = a$로 설정한다.
- 재암호화 키 생성: A는 B의 공개키 g^b를 전송받아 A와 B사이의 재암호화 키를 생성한다. $RK_{A\to B} = g^{b/a} \in G$
- 1레벨 암호화: A는 $r \in_R Z_q$를 선택하고, 다음과 같이 $m \in G_T$을 암호화한다. 1레벨 암호문은 sk_A를 소유하고 있는 사용자만 복호화할 수 있다.

$$C_A = (E^{ar}, m \cdot E^r), \quad E = e(g,g)$$

- 1레벨 암호문의 복호화: A는 다음과 같이 1레벨 암호문을 복호화할 수 있다.

$$C_A = (E^{ar}, m \cdot E^r)$$

$$m = \frac{m \cdot E^r}{(E^{ar})^{1/a}}$$

- 2레벨 암호화: 데이터의 재암호화를 허용하는 경우, 2레벨 암호화를 사용한다. A는 $r \in_R Z_q$를 선택하고, 다음과 같이 $m \in G_T$을 암호화한다.

$$C_A = (g^{ar}, m \cdot E^r)$$

- 재암호화: 프록시는 전달받은 C_A를 $RK_{A\to B}$로 재암호화 한다.

$C_A = (g^{ar}, m \cdot E^r)$로부터

$e(g^{ar}, RK_{A\to B}) = e(g^{ar}, g^{b/a}) = E^{br}$을 계산한다. C_B는 다음과 같다.

$$C_B = (E^{br}, m \cdot E^r)$$

- 재암호화 된 암호문의 복호화: B는 C_B를 전달받아 다음과 같이 복호화할 수 있다.

$C_B = (E^{br}, m \cdot E^r)$로부터 $m = \frac{m \cdot E^r}{(E^{br})^{1/b}}$이 된다.

Ateniese 등의 기법은 sk_A와 pk_B를 이용하여 $RK_{A\to B}$를 생성하므로 사전에 개인키를 공유하지 않아도 된다. A의 2레벨 암호문을 재암호화하면 B의 1레벨 암호문이 되므로 B는 이를 다시 재암호화할 수 없다. $RK_{A\to B}$로는 A의 암호문만 재암호화할 수 있으며, B의 암호문은 재암호화할 수 없어 일방향성을 갖는다. 프록시는 $g^{a/b}$로부터 $g^{b/a}$를 계산할 수 없으며, 이를 이용하여 다른 사용자의 재암호화키 역시 생성할 수 없다. 따라서, 프록시는 단순히 재암호화 기능만 수행할 수 있다. 또한, 프록시와 B가 공모하더라도 A의 개인키를 알아내는 것은 어렵다.

2.4 Weng 등의 기법 [9]

Ateniese 등의 기법이 암호학적인 측면에서 이전 기법들에 비해 상당히 안전하지만 A가 $RK_{A\to B}$를 발행하게 되면 A의 모든 암호문을 B의 암호문으로 재암호화할 수 있다는 문제가 발생한다. A가 $RK_{A\to B}$생성에 사용한 자신의 개인키 a로 암호화한 모든 암호문은 B의 암호문으로 재암호화할 수 있게 된다. 이러한 재암호화 오남용 문제를 해결하기 위해 2009년 Weng 등은 조건부 프록시 재암호화 기법(Conditional Proxy Re-Encryption)을 제안하였다.

CPRE 기법은 A가 추가적으로 조건값을 생성하여 이를 암호화와 재암호화키 생성 과정에 사용한다. A가 조건값 w_1을 생성하여 $RK_{A\to B}^{w_1}$를 발행하면, 프록시는 w_1이 포함된 A의 암호문 $C_A^{w_1}$만 정상적으로 재암호화 할 수 있다. Weng 등의 기법은 다음과 같다.

- 키 생성: A에 대한 공개키, 개인키 쌍 생성을 위해 $a\in_R Z_q$를 선택하고 $pk_A = g^a, sk_A = a$로 설정한다.
- 재암호화 키 생성: A는 B의 공개키 pk_B를 전송받고 $r,s\in_R Z_q$를 선택한 후, 조건값 w일 때 B에 대한 재암호화키를 다음과 같이 생성한다.

$$RK_{A\to B}^{w} = (rk_1, rk_2)$$
$$rk_1 = (H_2(pk_A, w)pk_B^{s\cdot H_5(pk_B^{s\cdot sk_A})})^{-sk_A}$$
$$rk_2 = pk_A^s$$

- 1레벨 암호화: A는 $r,s\in_R Z_q$를 선택하고, 다음과 같이 $m\in\{0,1\}^n$을 암호화한다.

$$R = H_1(m,r)$$
$$C_A = (C_1, C_2, C_3, C_4)$$
$$= (g^r, Re(g, pk_A)^{-r\cdot s\cdot sk_A H_5(pk^{s\cdot sk_A})}, m\oplus H_3(R), g^{s\cdot sk_A})$$

- 1레벨 암호문의 복호화: A는 다음과 같이 C_A를 복호화할 수 있다.

$$R = C_2\cdot e(C_1, C_4)^{sk_A H_5(C_4^{sk_A})}$$
$$m = C_3\oplus H_3(R)$$

$g^{H_1(m,R)} = C_1$를 확인하여 같을 경우 결과는 m이 되며, 다를 경우 $\perp$가 된다.

- 2레벨 암호문: 재암호화가 가능한 2레벨 암호문은 다음과 같이 생성된다.

$$C_A^w = (C_1, C_2, C_3, C_4)$$
$$= (g^r, Re(pk_A, H_2(pk, w))^r, m\oplus H_3(R), H_4(C_1, C_2, C_3)^r)$$

- 재암호화: 프록시는 재암호화키를 이용하여 A의 2레벨 암호문을 B의 1레벨 암호문으로 변경할 수 있다.

$$\overline{C_1} = C_1, \overline{C_2} = C_2\cdot e(C_1, rk_1), \overline{C_3} = C_3, \overline{C_4} = rk_2$$
$$C_B = (\overline{C_1}, \overline{C_2}, \overline{C_3}, \overline{C_4})$$
$$= (g^r, Re(g, pk_B)^{-r\cdot s\cdot sk_A H_5(pk^{s\cdot sk_A})}, m\oplus H_3(R), g^{s\cdot sk_A})$$

- 재암호화된 암호문의 복호화: B는 C_B를 전송받아 다음과 같이 복호화 할 수 있다.

$$R = \frac{\overline{C_2}}{e(C_1, C_4))^{sk_B}}$$
$$m = \overline{C_3}\oplus H_3(R)$$

$g^{H_1(m,R)} = C_1$를 확인하여 같을 경우 결과는 m이 되며, 다를 경우 $\perp$가 된다.

Weng 등의 기법은 조건값을 통해 복호권한의 위임을 제어할 수 있지만 조건값의 개수만큼 재암호화 키를 생성해야 한다. 예를 들어, A가 B에 대해 두 조건을 할당하면 프록시는 B에 대해 $RK_{A\to B}^{w_1}, RK_{A\to B}^{w_2}$ 두 개의 재암호화 키를 유지해야 한다. 이는 사용자와 조건값이 많아질수록 서버에 대한 큰 부담이 될 수 있다.

III. 제안하는 기법

기존 Weng등의 기법이 조건값을 통해 재암호화 오남용 문제를 적절히 해결하였지만, 한 사용자에 대한 재암호화 키를 조건값의 수 만큼 생성해야 되는 부담이 있다. 본 절에서는 재암호화키 생성의 측면에서 효율적인 조건부 프록시 재암호화 기법을 제안한다. 재암호화 키에 조건값을 포함시키면 Weng 등의 기법과 같이 조건값의 수 만큼 재암호화 키를 생성해야 하며, 프록시가 조건값을 알 때에만 재암호화가 가능하도록 설계하면 재암호화 키를 한번만 생성해도 되지만 조건값을 아는 프록시가 이를 남용할 수 있게 되는 문제가 있다. 따라서, 제안하는 기법에서는 조건값이 필요한 부분을 암호화, 복호화 과정으로 제한한다. 이를 통해 얻을 수 있는 이점은 데이터를 생성하는 각 사용자가 전송 대상의 재암호화 키를 한번만 생성하면 된다는 점이다. 따라서 각 사용자는 암호문 생성 시 다른 조건값을 사용함으로써 복호 권한을 제어할 수 있게 된다. 조건값은 H_2해쉬함수에 의해 G 순환군으로 사상되므로 조건값의 선택에 제약이 없다.

3.1 제안하는 기법의 설계

제안하는 기법 역시 1레벨 암호화와 2레벨 암호화로 구성되며, 선택적 평문 공격에 안전하도록 설계하였다.

- 키 생성: A에 대한 공개키, 개인키 쌍 생성을 위해 $a\in_R Z_q$를 선택하고 $pk_A = g^a, sk_A = a$로 설정한다.
- 재암호화 키 생성: A는 B의 공개키 pk_B를 전송받고 $r,s\in_R Z_q$를 선택한 후, 조건값 w일 때 B에 대한 재암호화키를 다음과 같이 생성한다.

$RK^{w}_{A\rightarrow B}=(rk_1, rk_2)$
$rk_1=(pk_B^{s\cdot H_5(pk_B^{s\cdot sk_A})})^{sk_A}$
$rk_2=pk_A^{s}$

- 1레벨 암호화: A는 $r, s\in_R Z_q$를 선택하고, 다음과 같이 $m\in\{0,1\}^n$을 암호화한다.

$R=H_1(m,r)$
$C_A=(C_1, C_2, C_3, C_4)$
$=(g^r, Re(g, pk_i)^{-r\cdot s\cdot sk_A H_5(pk^{s\cdot sk_A})},$
$m\oplus H_3(R), g^{s\cdot sk_A})$

- 1레벨 암호문의 복호화: A는 다음과 같이 C_A를 복호화할 수 있다.

$R=C_2\cdot e(C_1, C_4)^{sk_A H_5(C_4^{sk_A})}$
$m=C_3\oplus H_3(R)$
$g^{H_1(m,R)}=C_1$를 확인하여 같을 경우 결과는 m이 되며, 다를 경우 $\perp$가 된다.

- 2레벨 암호문: 재암호화가 가능한 2레벨 암호문은 다음과 같이 생성된다.

$C^{w}_A=(C_1, C_2, C_3, C_4)$
$=(g^r, Re(pk_A, H_2(pk_A, w))^r,$
$m\oplus H_3(R), H_4(C_1, C_2, C_3)^r)$

- 재암호화: 프록시는 재암호화키를 이용하여 A의 2레벨 암호문을 B의 1레벨 암호문으로 변경할 수 있다.

$\overline{C_1}=C_1, \overline{C_2}=C_2\cdot e(C_1, rk_1), \overline{C_3}=C_3, \overline{C_4}=rk_2$
$C^{w}_B=(\overline{C_1}, \overline{C_2}, \overline{C_3}, \overline{C_4})$
$=(g^r, Re(g, H_2(w)+pk_B^{s\cdot sk_A H_5(pk_B^{s\cdot sk_A})}),$
$m\oplus H_3(R), g^{s\cdot sk_A})$

- 재암호화된 암호문의 복호화: B는 C^{w}_B를 전송받아 w를 갖고 있을 경우, 다음과 같이 복호화 할 수 있다.

$$R=\frac{\overline{C_2}}{e(C_1, H_2(w))\cdot e(C_1, C_4^{sk_B H_5(C_4^{sk_B})})}$$
$m=\overline{C_3}\oplus H_3(R)$
$g^{H_1(m,R)}=C_1$를 확인하여 같을 경우 결과는 m이 되며, 다를 경우 $\perp$가 된다.

3.2 조건값의 전달

제안하는 기법을 이용하여 데이터를 공유하기 위해서는 데이터 소유자인 A가 공유 대상인 B에게 조건값을 전달하여야 한다. 결국, 기존 프록시 재암호화 기법에 비해 하나의 키를 더 사용하는 셈이 된다. A가 재암호화 키를 생성하기 위해서는 B의 공개키를 받아야 한다. B의 공개키로 조건값을 암호화하여 전달하면 되므로 조건값 전달의 측면에서는 큰 문제가 되지 않는다. 그러면 결국 사용자 간 대칭키를 확립하여 데이터를 전송하는 간단한 방법에 비해 CPRE 기법을 사용하였을 때 어떤 장점이 있는가가 중요하다. 장점은 다음과 같다.

- 부재시에도 파일의 공유가 가능하다. 대칭키 방식을 사용하면 A가 다른 사용자의 요청이 있을 때마다 데이터를 다시 암호화 하여 전송해야 한다. 클라우드와 같이 공용 스토리지를 사용하는 환경에서는 이러한 방법이 비효율적일 수 있다. CPRE기법을 사용하면 암호화된 데이터를 스토리지를 통해 공유할 수 있으며, A가 부재중인 상황에서 공유가 가능하다.
- 재암호화 비용을 서버에게 위임할 수 있다. 일반적으로 프록시 재암호화 기법에서 프록시는 연산 능력을 가진 사용자 또는 서버가 수행할 수 있다. 서버는 재암호화를 수행하여 데이터 공유를 지원할 수 있으며, 서버가 재암호화 과정을 수행하더라도 평문이 드러나지 않으므로 안전하다.
- 데이터 소유자는 한번의 암호화만 수행한다. 데이터 소유자는 공유하고자 하는 대상이 가진 조건값을 이용하여 데이터를 암호화한다. 재암호화 과정은 서버가 수행하므로 사용자는 연산량 측면에서 효율적이다. 이는 동일한 데이터를 공유하는 사용자가 많아질수록 증가한다.
- 조건값의 제약이 없다. 조건값을 통해 여러 기능을 추가할 수 있다. 예를 들어, 해쉬 체인을 생성하여 그 값들을 조건값으로 사용하면 계층적 복호권한을 부여할 수 있다. 또한, 타임스탬프를 사용하여 유효기간을 설정하는 것도 가능하다. 최근에는 조건값을 속성 기반 암호화를 사용하여 생성하는 기법도 제안되었다[14].

IV. 분 석

본 절에서는 제안하는 기법을 안전성과 효율성 측면에서 분석한다. 안전성 분석은 조건값의 사용과 선택적 평문 공격에 대해 고려한다.

4.1 안전성

4.1.1 선택적 암호문 공격에 대한 안전성

제안하는 기법이 선택적 평문 공격에 대한 안전성을 가지기 위해서는 다음의 세 가지 조건을 만족해야

한다.

• 2레벨 암호문이 유효하다는 것은 공개적으로 검사할 수 있어야 한다. 그렇지 않을 경우, [11]에서 제시한 공격에 취약할 수 있다.

• 공격자는 2레벨 암호문을 악의적으로 조작할 수 없어야한다.

• 1레벨 암호문 역시 공격자가 악의적으로 조작할 수 없어야 한다.

제안하는 기법은 Weng 등의 기법을 기반으로 설계하였으며, 선택적 평문 공격에 강건하기 위해 적용한 방법들을 그대로 따른다. Weng 등의 기법이 선택적 평문 공격에 안전함을 증명하였으므로, 제안하는 기법은 동일한 안전성을 가진다. 제안하는 기법의 선택적 평문 공격에 대한 안전성은 DBDH 가정에 기반하며, DBDH의 복잡도에 의해 안전하다. 재암호화 키는 rk_1, rk_2로 구성되며, rk_1에 $H_5(pk_B^{s \cdot sk_A}) = H_5(rk_2^{sk_B})$을 포함함으로써 rk_1과 rk_2가 연관성을 가지도록 설계한다. 이러한 방법을 통해 제안하는 기법이 선택적 평문 공격에 대해 안전성을 가질 수 있다[9].

4.1.2 조건값에 대한 안전성

제안하는 기법에서는 조건값을 사용하여 복호 권한의 위임을 제어한다. 수신자에 대한 재암호화 키를 발급하였더라도 수신자가 조건값을 알지 못하면 데이터를 복호화 할 수 없다. 제안하는 기법을 사용하는 환경에서 재암호화 키를 발급받은 대상이 조건값에 대한 전사공격을 통해 평문을 얻는 경우를 고려해야 한다. 충분한 길이의 조건값을 사용함으로써 조건값이 전사공격에 안전성을 가질 수 있다.

4.2 효율성

제안하는 기법은 조건부 프록시 재암호화 기법을 기반으로 설계하였다. 따라서, 논문에서 제시한 이전의 프록시 재암호화 기법에 비해 기능적으로는 뛰어나지만 효율성은 떨어진다. 암호화에 드는 비용만 보더라도 Blaze 등의 기법[7]은 지수연산을 기반으로 하고 있고, Atenese 등의 기법[8]은 페어링 연산을 기반으로 하지만 미리 계산할 수 있다는 장점이 있다. 이에 비해 제안하는 기법은 조건값을 위한 추가 메시지와 페어링을 기반으로 하고 있으므로 이전 기법에 비해 효율성 측면에선 뛰어나다고 할 수 없다. 하지만, 제안하는 기법은 조건값을 통해 이전 프록시 재암호화 기법에서 발생하는 재암호화키 오남용 문제를 해결한다. 또한, Weng 등의 기법[9]이 조건값의 수 만큼 재암호화 키를 생성해야하므로 재암호화 키 생성에 대한 시간 복잡도는 $O(n)$ (n은 조건값의 수) 이지만, 제안하는 기법은 조건값과 재암호화키가 무관하므로 시간 복잡도는 $O(1)$이 된다.

제안하는 기법은 서버를 통하여 데이터를 공유할 때 보다 뛰어난 효율성을 기대할 수 있다. 많은 비용이 드는 재암호화 과정을 서버에게 위탁함으로써 사용자는 재암호화 연산에 드는 비용을 줄일 수 있다. 실제로 파일 공유가 빈번하게 이루어지는 환경에서 사용자는 데이터를 암호화하여 서버에 전송하는 것만으로 역할을 다 하게 된다. 사용자가 데이터 공유 요청을 많이 받더라도 이를 위해 필요한 연산은 서버가 수행하므로 사용자의 부담을 덜어줄 수 있다. 이전에 제안된 프록시 재암호화 기법들 역시 같은 장점을 가지지만, 제안하는 기법은 재암호화키의 오남용 문제를 해결한다는 점에 의미가 있다. 따라서, 이러한 부분들을 고려하였을 때, 제안하는 기법의 연산량 오버헤드는 충분이 납득할 수 있는 수준이다.

V. 결 론

본 논문에서는 프록시 재암호화 기법의 흐름에 대해 살펴보고 중요한 몇가지 기법에 대해 분석하였다. 기존 프록시 재암호화 기법에서 재암호화 키를 통한 반복적인 재암호화가 가능하다는 오남용 문제가 발생함에 따라 이를 해결하기 위해 조건부 프록시 재암호화 기법이 등장하였다. 이 기법은 조건값을 암호문에 포함된 조건값과 같은 값을 가진 재암호화 키로만 재암호화가 가능하므로 재암호화 오남용 문제를 완화할 수 있다. 하지만 조건값이 바뀔 때 마다 재암호화 키를 새로 생성해야 하는 문제가 있다. 제안하는 기법은 암호화와 복호화 과정에서만 조건값을 사용하도록 설계하여 재암호화 키를 반복적으로 생성해야 하는 문제를 해결하였다. 조건값의 전달에 관한 문제를 생각해 볼 수 있지만, 전달받은 공개키로 조건값을 암호화하여 전송할 수 있으며, 기존 기법에 비해 다음과 같은 장점이 있다. 첫째, 기존 기법과 동일하게 서버를 이용한 재암호화가 가능하면서 재암호화키 오남용 문제를 해결한다. 둘째, 조건값과 재암호화키는 무관하므로 조건값의 개수가 많아지더라도 재암호화키를 추가로 생성할 필요가 없다. Weng 등의 기법이 재암호화 키 생성에 따른 시간 복잡도가 $O(n)$인 것에 비해, 제

안하는 기법은 $O(1)$의 시간 복잡도를 가진다. 또한, 제안하는 기법은 선택적 암호문 공격에 안전하게 설계되었다.

참고문헌

[1] K.M. Khan, and Q. Malluhi, "Establishing trust in cloud computing," IT Professional, vol. 12, issue. 5, pp. 20-27, Sept. 2010.

[2] H. Takabi, J. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, issue. 6, pp. 24-31, Nov. 2010.

[3] K. Ren, C. Wang, and Q. Wang, "Security chanllenges for the public cloud," IEEE Internet Computing, vol. 16, issue. 1, pp. 59-73, Jan. 2012.

[4] L.M. Kaufman, "Data security in the world of cloud computing," IEEE Security & Privacy, vol. 7, issue. 4, pp. 61-64, July. 2009.

[5] 송유진, 박광용, "Proxy Re-encryption 기술," 정보보호학회지, 10(5), pp. 95-104, 2009년 10월.

[6] M. Mambo, and E. Okamoto, "Proxy cryptosystems: delegation of the power to decrypt ciphertext," IEICE Transactions on Fund Electronics Communications and Computer Science, vol. E80-A, no. 1, pp. 54-63, Jan. 1997.

[7] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," In Advances in Cryptology. EUROCRYPT'98, vol. 1403 of LNCS, pp. 127-144, May. 1998.

[8] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, issue. 1, pp. 1-30, Feb. 2006.

[9] J. Weng, Y. Yang, Q. Tang, R. H. Deng, and F. Bao, "Efficient conditional proxy re-encryption with Chosen-Ciphertext security," Proceedings of the 12th International Conference on Information Security, pp.151-166, Sept. 2009.

[10] C.K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional proxy broadcast re-encryption," Proceedings of the 14th Australasian Conference on Information Security and Privacy pp. 327-342, July. 2009.

[11] R.H. Deng, J. Weng, S. Liu, and K. Chen, "Chosen-ciphertext secure proxy re-encryption without pairings," Lecture Notes in Computer Science, vol. 5339, pp. 1-17, Dec. 2008.

[12] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," Lecture Notes in Computer Science, vol. 2248, pp. 514-532, Dec. 2001.

[13] D. Cash, E. Kiltz, and V. Shoup, "The Twin diffie-hellman problem and applications," Lecture Notes in Computer Science, vol. 4965, pp. 127-145, Apr. 2008.

[14] J. Zhao, D. Feng, and Z. Zhang, "Attribute-based conditional proxy re-encryption with chosen-ciphertext security," IEEE Global Telecommunications Conference (GLOBECOM 2010), pp. 1-6, Dec. 2010.

〈著者紹介〉

손 정 갑 (Junggab Son) 학생회원
2009년 2월: 한양대학교 컴퓨터공학부 학사
2011년 2월: 한양대학교 컴퓨터공학부 석사
2011년 3월~현재: 한양대학교 컴퓨터공학과 박사과정
〈관심분야〉 암호기술 응용, 클라우드 컴퓨팅 보안

오 희 국 (Heekuck Oh) 종신회원
1983년: 한양대학교 전자공학과 학사
1989년: 아이오와주립대학 전자계산학과 석사
1992년: 아이오와주립대학 전자계산학과 박사
1993년~1994년: 한국전자통신연구원 선임연구원
1995년 3월~현재: 한양대학교 컴퓨터공학과 교수
〈관심분야〉 암호프로토콜, 네트워크 보안

김 상 진 (SangJin Kim) 종신회원
1995년: 한양대학교 컴퓨터공학과 학사.
1997년: 한양대학교 컴퓨터공학과 석사.
2002년: 한양대학교 컴퓨터공학과 박사.
2003년 3월~현재: 한국기술교육대학교 컴퓨터공학부 부교수.
〈관심분야〉 프라이버시 보호, 애드혹 네트워크 보안, 클라우드 컴퓨팅 보안