Journal of The Institute of Electronics and Information Engineers Vol.53, NO.3, March 2016 http://dx.doi.org/10.5573/ieie.2016.53.3.029
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2016-53-3-4

프록시 재암호화 기반의 보안 클라우드 저장장치를 위한 분실된 비밀번호 변경 기법

(Method of Changing Password for Secure Cloud Storage based on Proxy Re-encryption Scheme)

박 영 훈*, 서 승 우**

(Young-Hoon Park and Seung-Woo Seo©)

요 약

클라우드에서 저장된 데이터의 보안 및 사용자의 프라이버시가 중요해짐에 따라 암호화하여 파일을 저장하되 사용자만 복호화할 수 있고, 클라우드 서비스 제공업체 조차도 암호화된 파일을 열어볼 수 없는 보안 클라우드가 개발되었다. 하지만, 이러한 제약 때문에 사용자가 비밀번호를 분실하면 저장된 데이터를 아무도 열어보지 못 하는 부작용이 발생하게 된다. 본 논문에서는 상기 문제점을 해결하기 위하여 프록시 재암호화 기법을 이용하여 보안 클라우드에서 사용자가 비밀번호를 분실하더라도 비밀번호를 갱신할 수 있는 기법을 제안하고자 한다. 본 기술을 사용하면 제 3자 및 클라우드 서비스 제공업체 조차도 파일 내용을 볼 수 없기 때문에 오로지 파일 소유자만이 안전하게 비밀번호를 업데이트 하고 암호화된 파일을 재암호화 할 수 있다.

Abstract

In cloud storages, as security of stored files and privacy of users become regarded as important concerns, secure cloud storages have been proposed, where stored files are encrypted with file owner's password and even the cloud service provider can not open the file contents. However, if the file owner forgets one's password, one can no longer access the file. To solve this problem, we propose a scheme for changing password for the secure cloud based on proxy re-encryption, which make the file owner enable to change password even when one forgets it. With the proposed scheme, only the file owner can change the password and re-encrypt the files securely because other user and even the service provider can not see the file contents.

Keywords : 보안 클라우드, 프록시 재암호화, 공개키 암호화

Ⅰ. 서 론

클라우드 저장장치는 파일을 언제 어디서든지 엑세스할 수 있다는 장점 때문에 폭발적으로 그 수요가 증가하고 있다. 그러나, 클라우드에 저장되는 파일에는 개인정보나 기업 기밀사항이 포함되어 있을 수 있기 때문에 파일의 보안성이 매우 중요하게 부각되고 있다[1]. 따라서 파일을 저장할 때 사용자의 비밀번호로 암호화하는 보안 클라우드 저장장치 서비스가 제안되었고, 현재 운영되고 있다. 이러한 보안 클라우드 서비스 제공 업체는 비밀번호와 그로부터 유도된 암/복호화 키를 저장하지 않고 있기 때문에, 파일 주인 이외에 다른 사용자, 심지어 클라우드 서비스 제공 업체조차도 파일을 열어볼 수 없다고 주장하고 있다[2~3].

보안 클라우드는 사용자의 비밀번호로 암/복호화 키를 생성하여 파일을 암/복호화한다. 이 때, 암/복호화

* 정회원, 숙명여자대학교
(Sookmyung Women's University)
** 정회원, 서울대학교
(Seoul National University)
Corresponding Author(E-mail: sseo@snu.ac.kr)
※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2009-0083495).
Received ; November 23, 2015 Revised ; December 7, 2015
Accepted ; March 8, 2016

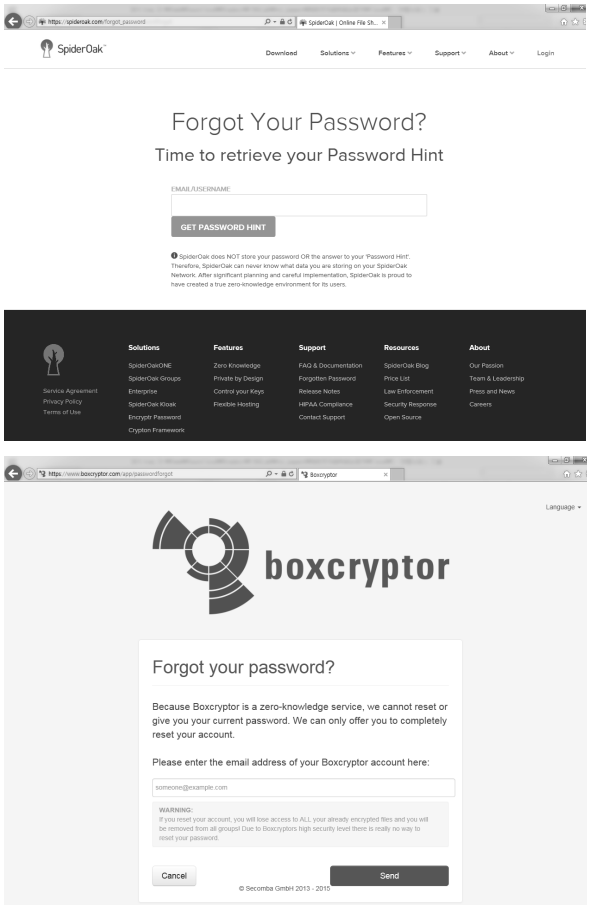


그림 1. 보안 클라우드에서 비밀번호 분실시 경고문
Fig 1. Warning Messages in Resettig Password Process.

키는 클라우드단에는 저장되어 있지 않다. 파일을 저장할 때는 파일을 클라이언트에서 암호화를 한 후, 그 결과를 클라우드 서버에 보내준다. 또한, 파일을 열 때는 암호화된 파일을 클라우드에서 다운로드 한 후 클라이언트에서 파일을 복호화한다.

클라우드 저장장치 시스템에서 클라이언트는 클라우드 관련 프로그램이 설치되어 있는 클라이언트와 그렇지 않은 클라이언트로 나눌 수 있다. 전자를 참여 클라이언트, 후자를 비참여 클라이언트라고 하자. 참여 클라이언트에는 클라우드에서 제공하는 싱크 프로그램이나 스마트기기용 어플리케이션이 설치되어 있다. 참여 클라이언트에는 사용자의 편의를 위하여 아이디 및 비밀번호가 저장되어 있어서 사용자가 일일이 로그인 하지 않아도 된다. 비참여 클라이언트에는 상기 서술한 프로그램이 전혀 설치되어 있지 않은 것을 말한다. 대표적인 예로 공공 PC나 본인 소유가 아닌 스마트기기 등이 있다. 이 클라이언트에서 사용자가 클라우드를 사용하려면 클라우드 웹페이지에 접속하여 로그인을 하여야 한다.

현재, 보안 클라우드에서 비밀번호를 잊어버리면 비밀번호는 재설정할 수 있지만, 저장된 데이터는 모두 포기해야 한다. 그 이유는 비밀번호는 사용자만이 알고 있는데, 이를 잊어버린 것이므로 암/복호화 키를 만들어낼 수 없게 된다. 실제로 현존하는 보안 클라우드 서비스에는 비밀번호를 재설정하면, 그림 1과 같이 데이터를 영영 사용할 수 없다고 경고문을 띄운다.[4~5] 물론 참여 클라이언트의 경우 비밀번호나 암/복호화 키가 저장되어 있기 때문에 클라우드로부터 암호화된 파일들을 받은 후 복호화 하고 새로운 키로 암호화 하면 된다. 혹은 참여 클라이언트와 클라우드가 동기화 되어 있는 경우, 클라이언트에 저장된 파일들을 새로운 키로 암호화하고 클라우드에 보내면 된다. 하지만, 이 방법들은 클라우드에 수십 혹은 수백 GB의 파일이 있을 경우 클라이언트에서 통신 및 연산 오버헤드가 엄청나가 소모된다. 또한 비참여 클라이언트에서는 비밀번호를 잊어버렸을 경우 파일을 이용할 수 있는 방법이 없다.

본 논문에서는 프록시 재암호화 기법을 이용하여 위에서 언급한 비밀번호를 잊어버렸을 경우 비밀번호를 바꿈과 동시에 파일들을 온전히 재암호화하는 기법을 소개하고자 한다. 이를 이용하면, 사용자가 비밀번호를 잊어버렸을 때 클라우드 서버에게 사용자가 재암호화 키만 안전하게 보내주면 된다. 그러면, 클라우드 서버는 이 키로 새로운 복호화 키로 풀 수 있게끔 파일들을 재암호화 한다. 이 과정에서 서버에 저장된 파일은 복호화되지 않으면서 클라우드 서버는 사용자의 복호화 키를 갖고 있지 않으므로 사용자의 파일은 안전하게 보호될 수 있다.

Ⅱ. 배경 설명

가. 겹선형사상

최근 들어, 겹선형 사상은 보안이나 프라이버시가 요구되는 네트워크에서 차세대 암호화 기법에 널리 쓰이고 있다.[6~8] 또한, 점차 이동 통신 기기의 성능이 향상되고 있고, 겹선형사상의 한 기법인 Tate pairing은 연산량과 소요 시간이 다른 겹선형사상에 비해 매우 적기 때문에 이동 통신 기기용으로 가능하다고 보고 있다.[9~10]

$\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$를 각각 소수 p를 기반으로 하는 곱셈 순환군이라 하자. 또한, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$는 생성자이며, 동형 사상 $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$이 존재하여, $\psi(g_2) = g_1$이라 하자. 이 때, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$가 다음을 만족할 때 e를

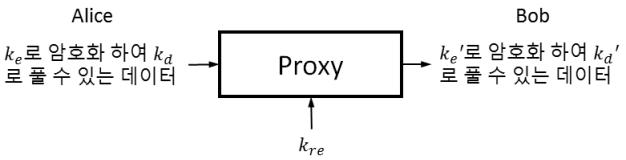


그림 2. 프록시 재암호화 기술
Fig. 2. Proxy Re-encryption Scheme.

겹선형사상이라 부른다.

a) 이진성: 임의의 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$와 임의의 정수 a, $b \in \mathbb{Z}_p$에 대하여, $e(u^a, v^b) = e(u,v)^{ab}$를 만족한다.

b) 일반성: $e(g_1, g_2) \neq 1$

c) 계산 가능성: $e(u,v)$를 계산할 수 있는 알고리즘이 존재

본 논문에서 후술할 프록시 재암호화 기법 역시 겹선형 사상을 이용하고 있다.

나. 프록시 재암호화 기법

프록시 재암호화 기법은 어떤 키로 복호화할 수 있는 암호화된 데이터를 다른 키로 복호화할 수 있게끔 바꿔주는 기법이다. 이 과정에서 암호화된 데이터는 복호화되지 않는다. 따라서 재암호화 동작을 다른 기기에게 맡기는 **위임성**이라는 성질과 재암호화 해서 다른 키를 가진 사용자에게 안전하게 보낼 수 있는 **전송용이성**이 있다. 프록시 재암호화 기법의 기본적인 방식은 다음 그림 2와 같다.

위의 그림 2에서 k_e와 k_d는 각각 Alice의 암호화키와 복호화키이고, k_e'와 k_d'는 각각 Bob의 암호화키와 복호화키이다. 또한, k_{re}는 재암호화키이다. Alice가 Proxy에게 k_e로 암호화된 파일을 보내서 재암호화를 요청하면 Proxy는 k_{re}를 이용하여 파일을 k_d'로 복호화할 수 있는 암호화된 파일로 바꾸어 준다[11].

본 논문에서는 보안 클라우드에서 비밀번호를 잊어버렸을 때에도 파일을 이용할 수 있도록 하는 기술을 제안한다. 사용자가 암/복호용 비밀번호를 분실했을 경우, 아무도 사용자의 암/복호용 비밀번호를 저장하지 않았기 때문에 암/복호용 비밀번호를 복구할 수 없다. 또한, 암/복호용 비밀번호를 갱신한다면, 이미 이전의 암/복호용 비밀번호로 생성된 키로 암호화된 클라우드의 데이터에 대하여, 이것의 암호를 풀고 새로운 비밀번호로 재암호화해야 하는데, 이전 암/복호용 비밀번호를 알지 못하므로 암호를 풀 수 없다. 이를 해결하기 위하여 복호화 과정 없이 암호화된 데이터를 재암호화하는 프록시-재암호화 기법을 사용한다.

프록시 재암호화 기술은 단방향과 양방향으로 나눌 수 있다.[12] 단방향은 암호화된 파일을 한번 재암호화하면 더 이상 재암호화를 할 수 없는 기술이고, 양방향은 여러 번 재암호화 할 수 있는 기술이다. 본 논문에서는 비밀번호를 계속 바꿀 수 있어야 하므로 양방향 프록시 재암호화 기법을 사용할 것이다. 그 대표적인 예로써 Canetti와 Hohenberger이 제안한 프록시 재암호화 기술을 이용한다[13].

Ⅲ. 제안 기술

가. 용어 설명

본 논문에서 사용할 용어와 기호들과 그 의미는 다음과 같다.

참여 클라이언트: 사용자의 복호화키를 갖고 있고, 미리 등록되어 있는 클라이언트. (예: 싱크 프로그램이 설치되어 있는 컴퓨터, 클라우드 앱이 설치되어 있는 스마트폰. 여기서 싱크 프로그램이란, 클라우드 저장 장치 서비스가 제공하는 어플리케이션으로 클라우드의 파일 및 폴더 구조와 컴퓨터의 파일 및 폴더 구조를 동일하게 유지시켜준다.)

비참여 클라이언트: 사용자의 복호화키를 갖고 있지 않고, 미리 등록되지 않은 클라이언트. (예: 싱크 프로그램이 설치되어 있지 않은 공용 컴퓨터)

M: 클라우드에 저장될 값 (평문)

k_s: 시드 키. 클라우드 서비스 제공자가 생성하고, 암/복호용 비밀번호를 갱신할 때마다 바뀐다.

ID, pw_{login}: 사용자의 아이디 및 로그인용 비밀번호. 사용자는 ID와 비밀번호를 모두 알고 있으며, 클라우드에는 ID값과 비밀번호의 해시값이 저장되어있다.

pw_{enc} : 사용자의 파일 암/복호화용 비밀번호. 사용자만이 알고 있다.

$f(x,y)$: 다항식 시간 안에 계산할 수 있는 입력값이 두 개인 함수(즉, 정해진 시간에 내에 결과를 내어주는 함수). 단, $z = f(x,y)$일 때, x, y 중 어느 하나와 z를 알고 있더라도 x, y 중 다른 하나를 매우 알기 어려운 성질을 갖고 있다. 클라우드 서비스 제공자와 클라이언트에 모두 저장되어 있다.

$\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$: Multiplicative cyclic group

$g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}$: 생성자. 클라우드 서비스 제공자와 클라이언트 모두 갖고 있다.

k_d: 사용자의 복호화 키. $k_d = f(pw_{enc}, k_s)$를 만족하여, 클라이언트에만 저장되어 있다.

k_e: 사용자의 암호화 키. $k_e = g_2^{k_d}$를 만족하고, 클라이언트에만 저장되어 있다.

k_{re}: 재암호화 키. 기존의 키로 암호화된 데이터를 새로운 키로 암호화된 데이터로 바꾸기 위하여 사용되는 키이다.

r: 난수

나. 제안하는 모델

본 논문에서는 우선 로그인 비밀번호 (pw_{login})와 암/복호용 비밀번호 (pw_{enc})와 같이 두 종류의 비밀번호를 사용하였다. 전자는 로그인을 위하여 사용자를 인증할 때 사용하고, 후자는 로그인 이후 파일의 암/복호용 비밀번호를 생성할 때 사용한다.

사용자가 처음 가입할 때 아이디와 두 종류의 비밀번호를 만들고 나면 클라우드가 생성한 시드 키와 생성자를 전송받아 암호화 키와 복호화 키를 계산한다. 이 때, 비밀번호, 암/복호화 키는 클라우드로 절대 전송되지 않는다.

클라이언트는 참여 클라이언트와 비참여 클라이언트가 있다. 사용자는 참여 클라이언트가 될 기기 (ex. 컴퓨터 혹은 스마트 기기)를 정한다. 그러면 암/복호화 키가 저장된다. 참여 클라이언트는 항상 인터넷에 연결 되어 있는 것이 좋다. 왜냐하면 비참여 클라이언트에서 이용할 때 비밀번호를 잊어버리면 참여 클라이언트의 도움이 필요하기 때문이다.

로그인 과정은 참여 클라이언트와 비참여 클라이언트에서의 로그인 과정으로 구분할 수 있는데, 전자의 경우는 참여 클라이언트에는 암/복호화 키가 저장되어 있기 때문에 별도의 로그인 과정이 필요 없다. 후자의 경우는 사용자가 직접 ID와 비밀번호를 입력하여 로그인을 하고, 클라우드로부터 시드 키와 생성자를 받아서 암/복호용 키를 유도해야 한다.

파일을 업로드할 때는 우선 클라이언트에서 암호화 키로 암호화 하고 업로드 하게 된다. 또한, 다운로드할 때는 다운로드 후 클라이언트에서 복호화를 하게 된다.여기서 쓰이는 암/복호화 과정은 Ⅱ-나 에 소개된 Canetti와 Hohenberger이 제안한 프록시-재암호화 기법을[13] 적용할 수 있는 특수한 암/복호화 기법이 사용된다.

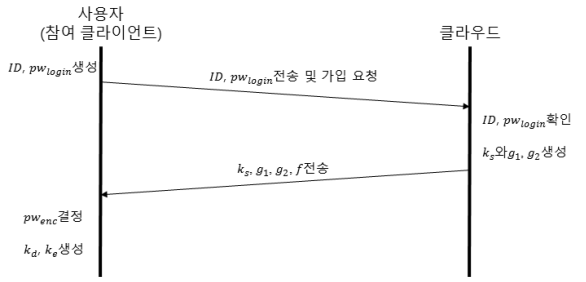


그림 3. 사용자 가입 프로토콜
Fig. 3. Protocol for New User's Registration

사용자가 암호화용 비밀번호를 바꾸고자 할 때는 사용자의 기존 암호화 키와 새로 만든 암호화 키를 이용하여 재암호화 키를 생성한 후, 이를 클라우드 서버에 보내서 재암호화를 요청한다. 이 때, 참여 클라이언트에서 재암호화를 요청하거나 비참여 클라이언트에서 암/복호용 비밀번호를 입력하면 재암호화 키를 생성할 수 있지만, 비참여 클라이언트에서 암/복호용 비밀번호를 잊어버렸을 경우에는 온라인 상태의 참여 클라이언트로부터 암호화용 키를 받아와서 재암호화 키를 만들게 된다.

다. 상세한 기술 설명

이 절에서는 Ⅲ-나. 에서 제시한 모델을 상세히 설명할 것이다. 구체적인 기술은 다음과 같다.

(1) 사용자 가입

이 과정은 사용자가 처음 클라우드 저장 서비스에 가입할 때 적용된다. 이는 참여 클라이언트에서만 이루어진다.

1) 사용자가 참여 클라이언트에서 자신의 로그인 정보(ID, pw_{login})를 생성하고, 이를 클라우드에 안전하게 전송한다.

2) 클라우드는 시드 키 k_s를 임의로 생성하고, 이를 참여 클라이언트에게 안전하게 전송한다.

3) 클라우드가 함수 f와 두 generator인 g_1, g_2를 참여 클라이언트에게 전송한다.

4) 사용자가 자신의 암/복호화용 비밀번호 pw_{enc}를 생성한다.

5) 사용자가 $k_d = f(pw_{enc},\ k_s)$, $k_e = g_2^{k_d}$를 이용하여 복호화 키, 암호화 키를 각각 유도한다.

(2) 로그인

이 과정은 사용자가 로그인을 할 때 적용된다. 참여

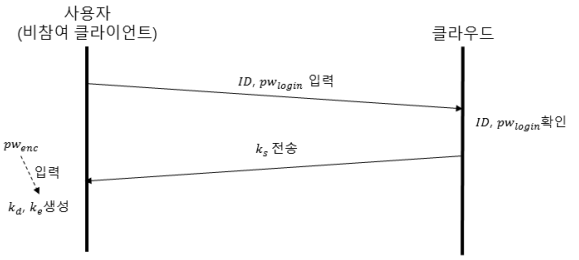


그림 4. 비참여 클라이언트에서의 로그인 프로토콜
Fig. 4. Protocol for Login at Unparticipated Client.

클라이언트에서 로그인할 때는 클라이언트에 이미 로그인 정보와 암/복호화에 필요한 값들이 이미 저장되어 있으므로 별도의 로그인 과정이 필요 없다.

이제 비참여 클라이언트에서 로그인 할 때를 보자. 비참여 클라이언트의 로그인 과정은 g_2와 함수 f를 전송받고, 암/복호용 비밀번호를 유도하는 과정이다.

1) 사용자가 로그인 정보 (ID, pw_{login})를 이용하여 자기 자신을 인증한다.

2) 인증이 완료되면 클라우드는 비참여 클라이언트에게 seed key인 k_s를 안전하게 보내준다. 그리고 함수 f와 generator g_2도 보내준다.

3) 사용자는 자신이 알고 있는 암/복호화용 비밀번호 pw_{enc}와 k_s를 이용하여 복호화 키 k_d를 유도한다.

4) 사용자는 k_d와 g_2를 이용하여 암호화 키 k_e를 유도한다.

(3) 데이터 저장 및 읽기

사용자가 클라우드에 로그인 하면, 복호화키와 암호화키를 모두 갖게 된다. 따라서 데이터를 저장하는 경우 암호화키로 암호화하여 클라우드로 업로드하면 되고, 데이터를 읽는 경우 클라우드에서 암호화된 데이터를 다운로드받아서 복호화키로 복호화하면 된다.

파일 암호화를 위하여 다음과 같은 연산을 한다. 파일 원본 데이터를 M이라 하고, 암호화된 데이터를 (C,Z)라고 하면,

$$(C,Z) \leftarrow \left(M \cdot e(g_1,k_e)^r,\ g_1^r\right)$$

이 때, r은 난수이며, 암호화를 할 때마다 새로 생성한다. 또한, 위의 식과 같이 암호화된 데이터는 C와 Z 두 값의 순서쌍으로 이루어져 있다.

복호화는 다음 식을 이용한다. 암호화된 데이터가 (C,Z)일 때,

$$M' = \frac{C}{e(Z,g_2)^{k_d}}$$

만일 사용되는 키들에 아무 이상이 없다면 다음과 같은 이유로 $M = M'$이 된다.

$$M' = \frac{C}{e(Z,g_2)^{k_d}} = \frac{M \cdot e(g_1,k_e)^r}{e(Z,g_2)^{k_d}} = \frac{M \cdot e(g_1,g_2)^{k_d r}}{e(g_1,g_2)^{r k_d}} = M$$

파일을 암/복호화할 때는 파일을 단위 데이터로 나눈 후, 각각의 단위에 대하여 위의 식들을 적용한다. 이 때, 연산량을 줄이기 위하여, 같은 파일에 대해서는 모든 단위 데이터들에 같은 난수 r을 이용할 수 있다. 즉, 파일을 단위 데이터로 쪼갠 결과가 다음과 같다고 하자.

$$M_1 \| M_2 \| M_3 \| \cdots \| M_\ell$$

그러면, 암호화의 결과는 다음과 같다.

$$C_1 \| C_2 \| C_3 \| \cdots \| C_\ell \| Z$$

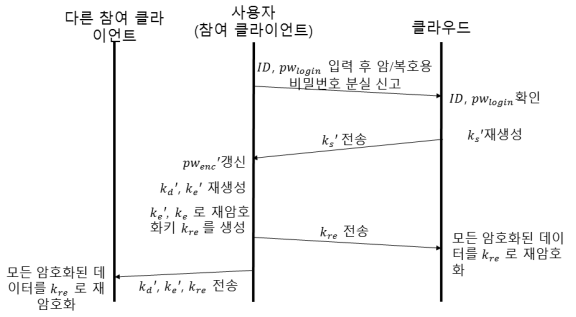
단, $i = 1, 2, \cdots, \ell$에 대하여, $C_i = M_i \cdot e(g_1,k_e)^r$이고, $Z = g_1^r$이 성립한다. 또한, 선택적으로, 크기가 큰 파일에 대하여, 파일 전체를 프록시 재암호화 방식에 쓰이는 암호화 방식으로 암호화 하지 않고, AES와 같은 기존의 암호화 방식으로 암호화를 한 후, 이 때 사용되었던 키를 프록시 재암호화 방식에 사용되는 암호화 방식으로 암호화 해도 된다.

(4) 아이디 혹은 비밀번호를 분실시 갱신 과정

이제, 사용자가 아이디나 비밀번호를 잃어버렸을 때 갱신 과정을 알아볼 것이다. 비밀번호는 로그인용 비밀번호 pw_{login}과 암/복호화용 비밀번호 pw_{enc}가 있다.

우선, 아이디나 로그인용 비밀번호를 잃어버렸을 경우에는 기존의 아이디 찾기나 비밀번호 갱신 기술을 이용하면 된다.

이제, 암/복호화용 비밀번호를 잃어버렸을 경우를 보자. 여기서는 사용자가 로그인 아이디와 비밀번호는 모두 알고 있다고 가정한다. (3)항에서 클라우드 서비스



참여 클라이언트에서 비밀번호 갱신할 때

그림 5. 참여 클라이언트에서 암/복호용 비밀번호 갱신 프로토콜

Fig. 5. Protocol for Updating Password for Encryption/ Decryption at Participated Client.

제공자가 파일을 열어보게 하지 못하게 하기 위하여 사용자의 암/복호용 비밀번호로부터 생성된 키로 암호화 하였다. 이제, 새로운 암호화 키와 복호화 키를 각각 $k_e{}'$와 $k_d{}'$라 할 때, $k_d{}'$로 풀 수 있는 파일로 갱신하는 과정을 살펴보자. 편의상 (3)항에서 사용했던 기호를 이용하자. 암호화된 파일이

$$C_1 \| \ C_2 \| \ C_3 \| \cdots \| \ C_\ell \| \ Z$$

일 때, 재암호화 키를 $k_{re} = \dfrac{k_e{}'}{k_e}$로 정의하면, $i = 1$, 2, …, ℓ에 대하여,

$$C_i' = C_i \cdot e(Z, k_{re})$$

를 적용하여, 다음과 같이 바꾼다.

$$C_1' \ \| \ C_2' \ \| \ C_3' \ \| \cdots \| \ C_\ell' \ \| \ Z$$

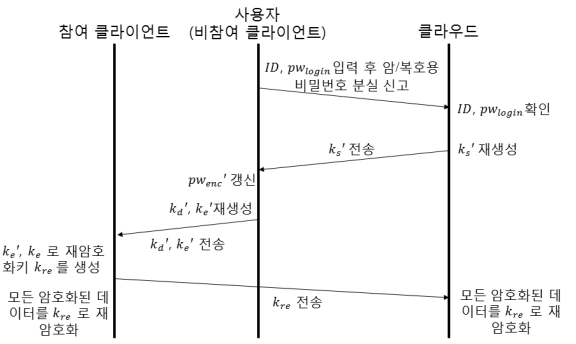
이렇게 하면,

$$\begin{aligned} C_i' &= M \cdot e(g_1, k_e)^r \cdot e(Z, k_{re}) \\ &= M \cdot e(g_1, g_2)^{k_d r} \cdot e(g_1^r, g_2^{k_d' - k_d}) \\ &= M \cdot e(g_1, g_2)^{k_d r} \cdot e(g_1, g_2)^{r(k_d' - k_d)} \\ &= M \cdot e(g_1, g_2)^{k_d' r} \\ &= M \cdot e(g_1, k_e')^r \end{aligned}$$

이 되어 $k_d{}'$로 복호화 할 수 있게 된다.

암/복호용 비밀번호 갱신 과정은 클라이언트가 참여 클라이언트일 때와 비참여 클라이언트일 때로 나뉜다. 각각에 대한 상세한 과정은 다음과 같다.

참여 클라이언트에는 암/복호용 비밀번호가 저장되어 있기 때문에 이를 이용하여 재암호화키를 쉽게 유도할 수 있다. 구체적인 방법은 다음과 같다.



비참여 클라이언트에서 비밀번호 갱신할 때

그림 6. 비참여 클라이언트에서 암/복호용 비밀번호 갱신 프로토콜

Fig. 6. Protocol for Updating Password for Encryption/ Decryption at Unparticipated Client.

1) 사용자가 참여 클라이언트에서 로그인 정보(ID, pw_{login})를 이용하여 로그인 후 클라우드에 암/복호용 비밀번호 분실 신고를 한다.

2) 클라우드는 사용자의 seed key를 업데이트하고, 업데이트된 seed key $k_s{}'$를 사용자에게 안전하게 보내준다.

3) 암/복호용 비밀번호를 갱신하고 이를 $pw_{enc}{}'$라 한다.

4) 참여 클라이언트가 $k_s{}'$와 $pw_{enc}{}'$를 이용하여 새로운 복호화키인 $k_d{}'$를 유도한다.

5) 참여 클라이언트가 $k_d{}'$를 이용하여 새로운 암호화키 $k_e{}'$를 유도한다.

6) 참여 클라이언트에서 k_e와 $k_e{}'$를 이용하여 k_{re}을 계산한다.

7) 유도한 k_{re}을 각 참여 클라이언트 및 클라우드에게 모두 안전하게 전송하고, 이들은 재암호화키로 저장된 암호화된 데이터들을 모두 재암호화한다.

재암호화키를 만들어 내기 위해서는 k_e가 있어야 하는데, 이것이 저장되어 있는 곳은 참여 클라이언트 뿐이다. 따라서 이 경우는 참여 클라이언트와의 통신을 이용하여 k_e를 비참여 클라이언트로 받아오고, 이것과 새로 만든 암호화키를 이용하여 재암호화키를 만든다. 구체적인 방법은 다음과 같다.

1) 사용자가 참여 클라이언트에서 로그인 정보(ID, pw_{login})를 이용하여 로그인 후 클라우드에 암/복호용 비밀번호 분실 신고를 한다.

2) 클라우드는 사용자의 seed key를 업데이트하고, 업데이트된 seed key $k_s{}'$를 사용자에게 안전하게 보내

준다.

3) 암/복호용 비밀번호를 갱신하고 이를 pw_{enc}'라 한다.

4) 비참여 클라이언트가 k_s'와 pw_{enc}'를 이용하여 새로운 복호화키인 k_d'를 유도한다.

5) 비참여 클라이언트가 k_d'를 이용하여 새로운 암호화키 k_e'를 유도한다.

6) 비참여 클라이언트가 k_d'와 k_e'를 모든 자신의 참여 클라이언트들에게 안전하게 전송한다. 이 때, 직접 클라이언트끼리 인터넷 망을 통하여 통신할 수도 있고, 아니면 클라우드 서비스 제공 업체의 서버를 통하여 통신할 수도 있다.

7) 참여 클라이언트에서 k_e와 k_e'를 이용하여 재암호화 키 k_{re}를 유도한다.

8) 유도한 k_{re}를 이용하여 참여 클라이언트들은 자신들이 갖고 있던 파일들을 재암호화한다.

9) 참여클라이언트 중 하나가 k_{re}를 클라우드로 전송하고, 클라우드도 재암호화키를 이용하여 파일들을 재암호화한다.

Ⅳ. 토의 및 결론

우선, 제안된 기술을 사용하면 파일을 재암호화하는 과정에서 파일을 복호화하지 않으므로 파일 내용이 다른 사람에게 노출될 우려는 없다. 또한, 재암호화하기 위해서는 기존의 암호화키가 필요한데, 이는 참여 클라이언트에만 저장되어 있고, 비참여 클라이언트에서 비밀번호를 갱신할 때는 비참여 클라이언트에서 생성한 암/복호화 키를 참여클라이언트에게 넘겨주고, 참여 클라이언트에서 재암호화키를 생성하므로 암/복호화 키는 클라우드로 가지 않는다. 따라서 클라우드는 복호화도, 자신이 풀 수 있게끔 재암호화도 하지 못 한다.

한편, 재암호화 과정에서 각 파일마다 난수가 다르게 설정되어 있고, 같은 파일 안에서는 단위 데이터들을 암호화 할 때 난수를 모두 같게 정했기 때문에 겹선형 사상의 연산 횟수는 파일의 개수와 같다. 나머지는 모두 곱셈 연산이다. 따라서 연산 오버헤드에서 겹선형 사상을 계산에 의한 연산 비용은 큰 비중을 차지하지 않을 것이다.

본 논문에서는 보안 클라우드에서 사용자가 비밀번호를 잊어버렸을 때에도 파일들을 사용할 수 있게 하는 기술을 제안하였다. 제안된 기술을 이용하면 파일들을 온전히 이용할 수 있을 뿐 아니라 다른 사용자와 클라우드 서비스 제공업체에게조차도 파일의 내용들이 노출이 되지 않으면서 재암호화를 할 수 있다.

REFERENCES

[1] Fei Chen, Tao Xiang, Yuanyuan Yang, and Chow S.S.M., “Secure cloud storage meets with secure network coding,” in INFOCOM, 2014 Proceedings IEEE, vol., no., pp.673-681, April 27 2014-May 2 2014

[2] Cong Wang, Chow, S.S.M., Qian Wang, Kui Ren, and Wenjing Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” in Computers, IEEE Transactions on , vol.62, no.2, pp.362-375, Feb. 2013

[3] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM conference on Computer and communications security (CCS '07). ACM, New York, NY, USA, 598-609.

[4] https://spideroak.com/

[5] https://www.boxcryptor.com/

[6] D. Boneh, and M. Franklin, “Identity-based encryption from the weil pairing,” SIAM J. Comput., 32(3):586-615, Mar 2003.

[7] X. Lin, X. Sun, P. Ho, and X. Shen. “GSIS: A secure and privacy-preserving protocol for vehicular communications,” Vehicular Technology, IEEE Transactions on, 56(6):3442-3456, Nov. 2007.

[8] Young-Hoon Park, and Seung-Woo Seo, “Scheme for Verification Between Mobile Devices in a Service with Expiration Time by Using Zero-knowledge Proof”, in Journal of the Institute of Electronics Engineers of Korea Vol. 50 No.3, pp. 23-32, Mar. 2013

[9] Y. Kawahara, T. Takagi, and E. Okamoto. “Efficient Implementation of Tate Pairing on a Mobile Phone Using Java,” In Computational Intelligence and Security, Lecture Notes In Artificial Intelligence, Vol. 4456. Springer-Verlag, Berlin, Heidelberg 396-405.

[10] S. D. Galbraith, K. Harrison, and D. Soldera. 2002. “Implementing the Tate Pairing,” In Proceedings of the 5th International Symposium on Algorithmic Number Theory (ANTS-V), Claus Fieker and David R. Kohel (Eds.). Springer-Verlag, London, UK, UK, 324-337.

[11] WooKwon Koo, JungYeon Hwang, Hyoung-Joong

Kim, and DongHoon Lee, “ID-Based Proxy Re-encryption Scheme with Chosen-CiphertextSecurity”, in Journal of the Institute of Electronics Engineers of Korea Vol. 46 No. 3, pp. 64-77, Jan. 2009

[12] Anca-Andreea Ivan and Yevgeniy Dodis. Proxy Cryptography Revisited. In NDSS. The Internet Society, 2003.

[13] Ran Canetti and Susan Hohenberger. 2007. Chosen-ciphertext secure proxy re-encryption. In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07). ACM, New York, NY, USA, 185-194.

저 자 소 개

박 영 훈(정회원)

2006년 서울대학교 전기공학부 학사 졸업.

2008년 서울대학교 전기컴퓨터공학부 석사 졸업.

2013년 서울대학교 전기컴퓨터공학부 박사 졸업.

2016년～현재 숙명여자대학교 컴퓨터과학부 조교수

<주관심분야: 컴퓨터 보안, 네트워크 보안, 암호, 최적화>

서 승 우(정회원)

1987년 2월 서울대학교 전기공학과 학사 졸업

1989년 2월 서울대학교 전기공학과 석사 졸업

1993년 12월 펜실베니아 주립대학 박사 졸업

1996년～현재 서울대학교 공과대학 전기컴퓨터공학부 교수

2009년 9월～현재 서울대학교 지능형자동차IT연구센터 센터장

<주관심분야 : 자동차IT, 시스템 최적화, 보안>