情報保護學會論文誌
第 23 卷 第 2 號, 2013. 4

비교가능 암호화의 허점*

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

A Security Hole in Comparable Encryption*

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

요 약

확률적 공개키 시스템에서 두 암호문이 주어졌을 때 이들을 복호화하지 않고 동일한 메시지를 암호화한 것인지 확인할 수 있는 암호기법을 비교가능 암호화(comparable encryption)라 한다. 최근에 Yang 등이 이와 같은 암호기법을 제안하였으며, 이영민 등과 Tang은 확인자를 제한할 수 있도록 Yang 등이 제안한 기법을 수정하였다. 하지만 Yang 등이 제안한 시스템은 주장된 것과 달리 암호화된 메시지가 서로 다른 경우에도 같다는 결과를 주는 허점을 가지고 있으며, 이 허점은 이영민 등과 Tang 시스템에도 동일하게 나타난다. 이 논문에서는 이와 같은 허점을 제시하며, 이 허점이 응용에 미칠 수 있는 파급효과를 분석한다.

ABSTRACT

Comparable encryption allows a verifier to test whether given two ciphertexts from a probabilistic public key cryptosystem are encryption of the same message without decrypting them. Recently, Yang et al. proposed such scheme and Lee et al. and Tang independently modified Yang et al.'s system to restrict the entity who can perform the verification. However, the original Yang et al.'s scheme has a flaw that enables two ciphertexts which are not encryption of the same message to pass the test. In this paper, we concretely show the faults in all three schemes considered and analyze the effect of this flaw in the use of such schemes in applications.

Keywords: Comparable Encryption, Searchable Encryption

I. 서 론

데이터의 프라이버시와 기밀성을 보장하기 위해 데이터를 암호화된 상태로 유지할 필요성이 높아지고 있다. 특히, 데이터를 신뢰할 수 없는 외부에 아웃소싱할 때에는 반드시 요구되는 기능 중 하나이다. 이 때문에 최근에 암호화된 데이터에 대한 연산에 관심이 높아지고 있으며, 이와 같은 연산 중 Yang 등이 최근에 주장한 비교가능 암호화(comparable encryption)[1]가 있다. 비교가능 암호화는 확률적 공개키 시스템에서 서로 다른 공개키로 암호화된 두 개의 암호문이 주어졌을 때 같은 메시지를 암호화한 것인지 복호화하지 않고 검사할 수 있도록 해준다. 이 연산이 가능하면 Yang 등이 제시한 것처럼 다음과 같은 응용에 사용할 수 있다.

- 검색가능 암호화(searchable encryption): 암호화된 문서와 이 문서 연관된 암호화된 키워

접수일(2012년 8월 9일), 수정일(1차: 2012년 12월 20일, 2차: 2013년 1월 31일), 게재확정일(2013년 2월 18일)
* 이 논문은 2011년도 한국기술교육대학교 교수교육연구진흥비 지원에 의하여 연구되었음.
* 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2012-R1A2A2A01046986).
† 주저자, sangjin@koreatech.ac.kr
‡ 교신저자, hkoh@hanyang.ac.kr

〔표 1〕 표기법

기호	의미
$G=\langle g\rangle, G_T$	CDH(Computational Diffie-Hellman) 문제가 어려운 위수가 소수 q인 곱셈군, 여기서 g는 군 G의 생성자
$\hat{e}: G\times G\rightarrow G_T$	겹선형 쌍함수(bilinear map)
$H_1 : \{0,1\}^*\rightarrow\{0,1\}^{\lvert G\rvert+\lvert Z_q^*\rvert}, H_2 : \{0,1\}^*\rightarrow Z_q^*$, $H_3 : \{0,1\}^*\rightarrow\{0,1\}^{\lvert G\rvert}$, $H_4 : \{0,1\}^*\rightarrow G$	충돌회피 해쉬함수
$\|$, $\oplus$	비트결합, XOR 연산

드들이 있을 때, 후자를 비교가능 암호화로 암호화한 경우, 누구나 동일 키워드를 자신의 공개키로 암호화하여 트랩도어(암호화된 질의 키워드)로 사용할 수 있다. 물론 이 경우 트랩도어를 아무나 생성할 수 있지만 검색가능 암호화의 경우 검색자를 확인하기 위한 별도의 인증 메커니즘이 필요하기 때문에 이것이 문제가 되지는 않는다.

- 암호문 분류: 위와 동일한 환경에서 암호화된 키워드들을 서로 비교하여 관련 문서들을 분류할 수 있다.

하지만 Yang 등의 기법은 서로 다른 메시지를 암호화한 두 암호문의 경우에도 확인 결과가 참이 되는 허점을 가지고 있다.

이영민 등[2]은 Yang 등의 시스템은 누구나 두 암호문이 같은 메시지를 암호화한 것인지 확인할 수 있기 때문에 이를 제한하기 위한 기법을 제안하였다. 즉, 확인할 때 이를 수행할 확인자의 개인키가 필요하도록 하였다. 하지만 원 기법의 단점인 서로 다른 암호문을 암호화한 경우에도 같다는 결과를 주는 단점이 그대로 남아 있다.

Tang[3]도 이영민 등과 유사하게 확인할 수 있는 주체를 제한하는 기법을 제안하였다. 차이점은 이영민 등의 시스템은 확인자의 개인키가 필요하도록 하여 제한을 한 반면에 Tang은 복호화를 할 수 있는 주체들이 확인자에게 토큰을 발급하여 줄 경우에만 가능하도록 제한을 하였다. 그러나 Tang도 여전이 동일한 단점을 지니고 있다.

이 논문에서는 이와 같은 단점을 제시하고 이 단점이 해당 응용에 어떤 파급효과가 있는지 분석한다. 참고로 Canard 등[4]은 비교가능 암호화와 유사한 평문 확인가능 암호화(plaintext-checkable encryption)을 제안하였다. 평문 확인가능 암호화는 공개키로 암호화된 암호문과 평문이 주어지면 해당 암호문을 복호화하지 않고 해당 평문을 암호화한 암호문인지 확인할 수 있는 암호화를 말한다.

이 논문의 구성은 다음과 같다. 2장에서는 이 논문에서 주장하는 비교가능 암호화의 단점을 각 시스템별로 제시한다. 3장에서는 이 논문에서 발견한 단점의 파급효과에 대해 분석하며, 4장에서는 결론과 향후 연구방향을 제시한다.

II. 비교가능 암호화의 허점

2.1 표기법

이 논문에서 다루는 3개의 프로토콜을 일관성 있게 설명하기 위해 공통적으로 사용되는 기호에 대해서는 표 1에 제시된 표기법을 사용한다.

2.2 Yang 등의 시스템

Yang 등이 제안한 시스템[1]의 암호화 연산과 검사 연산은 다음과 같다. 여기서 사용자 A의 개인키는 $x_A\in Z_q^*$이며, 공개키는 $y_A=g^{x_A}$이다.

- $E(y_A, m)$: $r\in_R Z_q^*$를 선택하고, $U=g^r$, $V=m^r$, $W=H_1(U\|V\|y_A^r)\oplus\{m\|r\}$을 계산한다. 결과 암호문은 $C=(U,V,W)$이다.
- $Test(C_A, C_B)$: $C_A=(U_A, V_A, W_A)$는 사용자 A가 생성한 암호문이고, $C_B=(U_B, V_B, W_B)$는 사용자 B가 생성한 암호문일 때, 만약 $\hat{e}(U_A, V_B)$와 $\hat{e}(U_B, V_A)$가 같으면 1을 반환하고, 아니면 0을 반환한다.

$C_A=(U_A=g^{r_A},\ V_A=m^{r_A},\ W_A=H_1(U_A\|V_A\|y_A^{r_A})\oplus\{m\|r_A\})$이고 $C_B=(U_B=g^{r_B},\ V_B=m^{r_B}, W_B=H_1(U_B\|V_B\|y_B^{r_B})\oplus\{m\|r_B\})$이면 $\hat{e}(U_A, V_B)$와 $\hat{e}(U_B, V_A)$는 다음과 같이 동일함을 알 수 있다.

$$\hat{e}(U_A, V_B)=\hat{e}(g^{r_A}, m^{r_B})=\hat{e}(g,m)^{r_A r_B}$$
$$=\hat{e}(g^{r_B}, m^{r_A})=\hat{e}(U_B, V_A)$$

하지만 $Test$ 연산에서 W_A와 W_B가 전혀 사용되지 않기 때문에 C_A와 C_B가 각각 서로 다른 m과 m'을 다음과 같이 암호화하였을 경우에도 1을 반환하게 된다.

$$C_A = (U_A = g^{r_A}, V_A = m^{r_A}, W_A = H_1(U_A \| V_A \| y^{r_A}) \oplus \{m \| r_A\})$$

$$C_B = (U_B = g^{r_B}, V_B = m^{r_B}, W_B = H_1(U_B \| V_B \| y^{r_B}) \oplus \{m' \| r_B\})$$

다음은 Yang 등[1]이 제시한 비교가능 암호화의 강건성(soundness) 정의이다.

정의 1. 〔강건성〕 모든 다항시간 알고리즘 M에 대해 다음이 성립해야 한다.

$$\Pr\begin{bmatrix}(C_A, C_B, x_A, x_B) \rightarrow M(1^k), m \leftarrow D(x_A, C_A), \\ m' \leftarrow D(x_B, C_B) : m \neq \perp \wedge m' \neq \perp \wedge \\ m \neq m' \wedge Test(C_A, C_B) = 1\end{bmatrix} \leq \epsilon(k)$$

여기서 k는 보안 파라미터이며, $\epsilon(k)$는 무시할 수 있을 정도로 작은 값으로 수렴하는 함수이다.

앞서 지적한 바와 같이 서로 다른 평문을 암호화한 C_A와 C_B에 대해 $Test$ 연산은 1을 반환하여 주지만 강건성 정의에 위배되지는 않는다. Yang 등의 강건성 정의는 복호화가 정상적으로 이루어지는 경우만을 고려하기 때문이다. 제시된 예의 경우 V_B와 W_B에 서로 다른 m을 사용하고 있다는 것이 복호화 과정에서 발견되기 때문에 복호화 연산은 $\perp$을 반환한다.

하지만 비교가능 암호화에서 비교를 수행하는 개체와 복호화를 수행하는 개체는 다르며, 복호화를 한 후에 비교하는 것은 아니다. 더욱이 서로 다른 공개키로 암호화된 암호문의 비교가 가능하기 때문에 비교를 수행하는 사용자는 두 암호문을 복호화할 수 있는 개인키들을 모두 가지고 있을 수가 없다. 즉, 비교가능 암호화의 핵심은 복호화하지 않고 비교하는데 있기 때문에 이 허점을 가진 상태로는 어떤 응용에서도 그 역할을 하기 어렵다.

2.3 Tang의 시스템

Tang이 제안한 시스템[3]의 암호화 연산과 검사 연산은 다음과 같다. 여기서 사용자 A의 개인키는 $x_A, z_A \in Z_q^*$이며, 공개키는 $y_A = (g^{x_A}, g^{z_A})$이다.

- $E(y_A, m)$: $u, v \in_R Z_q^*$를 선택하고, $U = g^u$, $V = g^v$, $W = H_1(g^{ux_A}) \oplus \{m \| u\}$, $Z = g^{H_2(g^{vz_A})+m}$, $S = H_3(U \| V \| W \| Z \| m \| u)$을 계산한다. 결과 암호문은 $C = (U, V, W, Z, S)$이다.
- $Test(C_A, C_B, T_A, T_B)$: $C_A = (U_A, V_A, W_A, Z_A, S_A)$는 사용자 A가 생성한 암호문이고, $C_B = (U_B, V_B, W_B, Z_B, S_B)$는 사용자 B가 생성한 암호문이며, $T_A = z_A$, $T_B = z_B$가 각 사용자가 프록시에게 제출한 토큰일 때, 만약 $Z_A \cdot g^{-H_2(V_A^{T_A})}$와 $Z_B \cdot g^{-H_2(V_B^{T_B})}$가 같으면 1을 반환하고, 아니면 0을 반환한다.

이 시스템의 경우에도 C_A와 C_B가 규칙대로 생성되었다고 가정하였을 때 다음과 같이 $Test$ 결과가 1임을 알 수 있다.

$$\begin{aligned} Z_A \cdot g^{-H_2(V_A^{T_A})} &= g^{H_2(g^{v_A z_A})+m} \cdot g^{-H_2(V_A^{T_A})} \\ &= g^{H_2(g^{v_A z_A})+m} \cdot g^{-H_2(g^{v_A z_A})} = g^m \end{aligned}$$

$$\begin{aligned} Z_B \cdot g^{-H_2(V_B^{T_B})} &= g^{H_2(g^{v_B z_B})+m} \cdot g^{-H_2(V_B^{T_B})} \\ &= g^{H_2(g^{v_B z_B})+m} \cdot g^{-H_2(g^{v_B z_B})} = g^m \end{aligned}$$

하지만 $Test$ 연산에서 Yang 등과 마찬가지로 W_A와 W_B가 전혀 사용되지 않기 때문에 C_A와 C_B가 각각 서로 다른 m과 m'을 다음과 같이 암호화하였을 경우에도 1을 반환하게 되어 Yang 등과 동일한 문제점을 가지고 있다.

$$\begin{aligned} C_A = (&U_A = g^{u_A}, V_A = g^{v_A}, W_A = H_1(g^{u_A x_A}) \oplus \{m \| u_A\}, \\ &Z_A = g^{H_2(g^{v_A z_A})+m}, S_A = H_3(U_A \| V_A \| W_A \| Z_A \| m \| u_A)) \end{aligned}$$

$$\begin{aligned} C_B = (&U_B = g^{u_B}, V_B = g^{v_B}, W_B = H_1(g^{u_B x_B}) \oplus \{m' \| u_B\}, \\ &Z_B = g^{H_2(g^{v_B z_B})+m}, S_B = H_3(U_B \| V_B \| W_B \| Z_B \| m \| u_B)) \end{aligned}$$

2.4 이영민 등의 시스템

이영민 등이 제안한 시스템[2]의 암호화 연산과 검사 연산은 다음과 같다. 여기서 사용자 A의 개인키는 $x_A \in Z_q^*$이며, $y_A = g^{x_A}$이고, 서버의 개인키 $x_T \in Z_q^*$이며, $y_T = g^{x_T}$이다.

- $E(y_A, y_T, m)$: $u, v \in_R Z_q^*$를 선택하고, $U = g^u$, $V = H_4(m)^v$, $W = g^v \oplus H_4(y_T^u)$, $Z = m \oplus H_4(y_A^u)$을 계산한다. 결과 암호문은 $C = (U, V, W, Z)$이다.
- $Test(C_A, C_B, x_T)$: $C_A = (U_A, V_A, W_A, Z_A)$는 사용자 A가 생성한 암호문이고, $C_B = (U_B, V_B, W_B, Z_B)$는 사용자 B가 생성한 암호문일 때, 서버는 $g^{v_A} = H_4(U_A^{x_T}) \oplus W_A$와 $g^{v_B} = H_4(U_B^{x_T}) \oplus W_B$를 계산한 후, 만약 $\hat{e}(g^{v_A}, V_B)$와 $\hat{e}(g^{v_B}, V_A)$가 같으면 1을 반환하고, 아니면 0을 반환한다.

이 시스템의 경우에도 C_A와 C_B가 규칙대로 생성되

었다고 가정하였을 때 다음과 같이 $Test$ 결과가 1임을 알 수 있다.

$$\hat{e}(g^{v_A}, V_B) = \hat{e}(g^{v_A}, H_4(m)^{v_B}) = \hat{e}(g, H_4(m))^{v_A v_B}$$
$$= \hat{e}(g^{v_B}, H_4(m)^{v_A}) = \hat{e}(g^{v_B}, V_A)$$

하지만 $Test$ 연산에서 Z_A와 Z_B가 전혀 사용되지 않기 때문에 기존 두 개의 시스템과 마찬가지로 C_A와 C_B가 각각 서로 다른 m과 m'을 다음과 같이 암호화하였을 경우에도 1을 반환하게 되어 Yang 등과 동일한 문제를 가지고 있다.

$$C_A = (U_A = g^{u_A}, V_A = H_4(m)^{v_A}, W_A = g^{v_A} \oplus H_4(y_T^{u_A}),$$
$$Z_A = m \oplus H_4(y_A^{u_A}))$$

$$C_B = (U_B = g^{u_B}, V_B = H_4(m)^{v_B}, W_B = g^{v_B} \oplus H_4(y_T^{u_B}),$$
$$Z_B = m' \oplus H_4(y_B^{u_B}))$$

III. 고찰

검색가능 암호화에서 키워드를 암호화하고 트랩도어를 생성할 때 이 논문에서 살펴본 비교가능 암호화 기법을 사용하면 키워드 추측 공격[5]에 더 취약해지는 문제점이 있다. Yang 등의 기법[1]은 아무나 확인할 수 있는 기법이기 때문에 암호화된 키워드를 확보하면 추측 키워드를 아무 사용자의 공개키로 암호화하여 비교함으로써 누구나 키워드 추측 공격을 할 수 있다. 이영민 등[2]과 Tang[3]처럼 검색서버만 할 수 있도록 제한하더라도 검색가능 암호화 환경에서 검색서버는 정직하지만 궁금한(curious-but- honest) 서버이므로 여전히 문제가 된다.

이와 같은 문제를 고려하지 않을 경우 실제 검색가능 암호화에서는 암호화된 키워드의 복호화가 필요하지 않기 때문에 비교가능 암호화의 일부분만을 사용할 수는 있다. 즉, w가 키워드일 때 Yang 등의 기법[1]을 응용하여 $U_A = g^{r_A}$, $V_A = H_4(w)^{r_A}$을 키워드 암호화를 위해 사용하고, 트랩도어로 $U_B = g^{r_B}$, $V_B = H_4(w)^{r_B}$를 사용하면 $\hat{e}(U_A, V_B)$와 $\hat{e}(U_B, V_A)$의 비교를 통해 검색이 가능하다. 따라서 검색자에 대한 별도 인증이 필요하다는 것(트랩도어만을 가지고는 검색자의 권한을 확인할 수 없음)을 고려하면 다중사용자 환경(여러 사용자가 데이터를 저장하고 검색할 수 있는 환경)을 위한 수단으로 고려할 수 있다. 그러나 키워드 추측 공격에 매우 취약하기 때문에 현실적으로 적용하는 것은 어렵다.

Yang 등[1]의 논문에서 주장된 것처럼 키워드가 아닌 메시지를 직접 이용하여 검사하는 것도 고려해 볼 수 있으나 검색가능 암호화는 외부에 아웃소싱된 데이터를 효과적으로 얻는 것이 목표이므로 데이터 자체를 가지고 검색하는 것은 응용에 한계가 있다.

검색가능 암호화 환경에서 비교가능 암호화를 활용할 경우 이 논문에서 제시된 허점을 공격자들이 실제 활용할 가능성은 크지 않다. 특히, 데이터를 암호화하여 저장하는 측이나 이를 검색하는 측에서 부정을 할 이유가 없기 때문이다. 하지만 이 허점을 이용하여 Baek 등[6]이 제시한 연관된 키워드를 조작하는 공격은 가능하다. 연관된 키워드를 조작하는 공격은 암호화된 문서와 연관되어 있는 암호화된 키워드를 삭제하거나 다른 문서에 있는 것과 바꾸는 공격을 말한다. 이 공격은 암호화된 키워드가 문서와 바인딩되어 있지 않고 이것을 바인딩하기 어렵기 때문에 가능한 공격이다. 즉, 연관된 키워드 조작 공격은 비교가능 암호화를 사용하지 않고 다른 기법을 통해 키워드를 암호화하더라도 기존 검색가능 암호화 기법들이 가지고 있는 취약점이다.

이 논문에서 지적한 문제점은 비교가능 암호기법의 암호문을 구성하는 값 중 비교할 때만 사용되는 값과 복호화할 때만 사용되는 값이 동일한 메시지를 이용하여 생성된 값인지 확인하는 메커니즘이 없기 때문에 발생한다. 즉, Yang 등의 시스템[1]에서 V와 W가 같은 m을 이용하여 생성된 값인지 확인할 수 있어야 이 허점을 제거할 수 있다. 하지만 이것은 복호화할 수 없는 사용자에게 숨겨진 값에 대한 접근이 필요하므로 제공하는 것이 쉽지 않다. 즉, 기본 ElGamal 기법 $U = g^r$, $W = y^r \cdot m$에 $V = m^r$를 추가한 비교가능 암호기법을 생각할 수 있고, 이 때 W와 V가 같은 m을 이용하여 생성되었다는 것을 영지식 형태로 증명할 수 있는지 여부를 검토하면 필요한 해결책의 난이도를 예측할 수 있다.

IV. 결 론

이 논문에서는 최근에 제안된 비교가능 암호화가 모두 동일한 취약점을 가지고 있다는 것을 보였다. 즉, 서로 다른 메시지를 암호화한 두 암호문의 경우에도 확인 검사를 통과하는 문제점이 있다. 검색가능 암호화 환경에서 비교가능 암호화 기법을 키워드 암호화에 활용할 경우 이 문제점을 활용한 의미 있는 공격은 없지만 키워드 추측 공격에 더 취약해지기 때문에 이 문제점과 상관없이 활용가치가 적다. 더욱이 복호화하

지 않고 서로 다른 공개키로 암호화된 암호문의 비교가 가능해야 하기 때문에 현재의 허점이 해결되지 않으면 이 기법은 어떤 응용에서도 활용되기 힘들며, 이 문제 자체를 해결하는 것도 쉽지 않을 것으로 보인다.

참고문헌

〔1〕 G. Yang, C.H. Tan, Q. Huang, and D.S. Wong, "Probabilistic public key encryption with equality test," Proceedings of the Cryptographis' Track at the RSA Conference, LNCS 5985, pp. 119-131, Mar. 2010.

〔2〕 이영민, 구우권, 이현숙, 이동훈, "고정된 검사자를 고려한 메시지 동일성 검사 공개키 암호시스템," 정보보호학회논문지, 21(5), pp. 3-13, 2011년 10월.

〔3〕 Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," Security and Communication Networks, vol. 5, no. 12, pp. 1351-1362, Dec. 2012.

〔4〕 S. Canard, G. Fuchsbauer, A. Gouget, and F. Laguillaumie, "Plaintext-checkable encryption," Proceedings of the Cryptographis' Track at the RSA Conference, LNCS 7178, pp. 332-348, Mar. 2012

〔5〕 I.R. Jeong, J.O. Kwon, D.W. Hong, and D.H. Lee, "Constructing PEKS schemes secure against keyword guessing attacks is possible?," Computer Communications, vol. 32, no. 2, pp. 394-396, Feb. 2009.

〔6〕 J. Baek, R. Safavi-Naini, and W. Susilo, "On the integration of public key data encryption and public key encryption with keyword search," Proceedings of the 9th International Conference on Information Security, LNCS 4176, pp. 217-232, Sept. 2006.

〈著者紹介〉

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

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