특집논문-08-13-1-05

암호화된 데이터에서의 OT(Oblivious Transfer)를 이용한 효율적인 검색 기술

이 현 숙[a], 박 종 환[a], 이 동 훈[a]‡

Efficient Oblivious Keyword Search on Encrypted Data

Hyun Sook Rhee[a], Jong Hwan Park[a], and Dong Hoon Lee[a]‡

요 약

암호화된 다양한 멀티미디어 컨텐츠(multimedia contents)를 공개하고 사용자가 서버에게 검색어에 대한 정보를 들어내지 않고 검색어를 포함하고 있는 멀티미디어 컨텐츠를 검색하는 키워드 검색에서의 문제점을 연구한다. 최근 Ogata와 Kurosawa는 Oblivious Transfer의 개념을 이용하여 키워드 검색 프로토콜을 제안하였다. 하지만 그들의 방식은 하나의 검색어를 이용하여 데이터를 검색할 때 공개된 모든 암호화된 데이터 수만큼의 비교를 위한 계산량이 요구된다. 이러한 문제점을 해결하기 위해 본 논문에서는 사용자가 모든 데이터를 비교 검색하지 않아도 되는 Oblivious Transfer기술을 이용한 효율적인 검색 기술을 제안한다. 본 논문에서는 제안된 프로토콜이 RSA known target inversion 문제의 어려움에 기반을 두고 안전하다는 것을 보인다.

Abstract

We study the problem of keyword search in which a server contains various multimedia contents and a user wishes to retrieve some multimedia items containing a specific keyword without revealing to the server which items they are. Recently, Ogata and Kurosawa introduced a keyword search scheme by using the notion of oblivious transfer. In their scheme, a user must inefficiently search and compare all the data stored in the server for each keyword search query. In this paper, we propose an efficient oblivious keyword search by using the oblivious transfer, in which a user needs not to search and compare all the data. We formally prove that the proposed scheme is secure under the hardness of RSA known target inversion problem.

Keyword : Keyword Search, Oblivious Transfer

Ⅰ. 서 론

인터넷에서 관리되고 저장되어지는 정보의 양이 급격하게 증가되어짐에 따라, 데이터베이스와 같은 저장 시스템의 중요성은 증가되었다. 그 결과로 저장시스템에 저장된 자료들에 대한 프라이버시를 보호하는 것은 데이터베이스 산업 분야에 가장 시급히 해결해야 할 과제이다.

a) 고려대학교 정보경영공학전문대학원
Graduate School of Information Security Korea University
‡ 교신저자 : 이동훈(donghlee@korea.ac.kr)
※본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0025))

최근, 데이터베이스 보안에서의 이슈는 외부 공격자로부터의 보호와 시스템 관리자와 같은 내부 사용자들로 부터의 자료의 보호를 들 수 있다. 내부 공격자로부터의 보호를 위해서는 사용자들은 역시 그들의 자료를 암호화 하여 저장하고 검색어에 대한 정보를 숨겨야 한다. 하지만 암호화는 자료를 랜덤하게 만들어 자료의 검색을 비효율적으로 만드는 단점도 존재한다. 최근에, 이러한 단점을 해결하기 위해 암호화된 문서상에서 효율적인 자료 검색 프로토콜들이 많이 제안되었고 [1, 3, 4, 5, 7, 8, 10, 11] 이들은 응용환경에 따라서 크게 3가지로 구분된다. 첫째는 Golle et. al.[5]에 의해서 제안된 스킴으로 사용자가 직접 암호화된 데이

터를 웹하드와 같이 공개된 서버에 공개한 후 서버관리자에게 검색어에 대한 정보를 들어내지 않고 검색하는 기술이다. 둘째는 Boneh et. al.[1]에 의해서 메일환경에서 제안된 스킴으로 메일 송신자가 암호화된 메일을 메일서버에 전송하고 수신자가 검색어에 대한 정보를 암호화된 형태로 제공하였을 때, 메일서버에게 메일이나 검색어에 대한 정보의 노출이 거의 없이 수신자에게 원하는 정보를 제공하는 환경이다. 셋째는 Ogata와 Kurosawa[7]에 의해서 제안된 스킴으로 데이터 제공자가 문서를 암호화해서 공개된 서버에 올려놓고 사용자들이 효율적으로 검색하는 기술이다. 이 방법이 앞에서 소개한 두 가지 방법과 상이한 점은 검색을 위한 비교 계산량이 사용자 단에서 이루어지기 때문에 서버의 과부하를 막을 수 있다는 점이다. 반면에 사용자 단에서의 비교를 위한 계산량이 요구되어지기 때문에 검색을 위한 비교 계산량의 효율성이 중요한 부분이다. 하지만 Ogata와 Kurosawa가 [7]에서 언급하였듯이, OKS 프로토콜에서 검색 가능한 키워드의 수는 오직 하나이고 검색어를 이용하여 데이터를 검색할 때 공개된 모든 암호화된 데이터 수만큼의 비교를 위한 계산량이 요구된다.

본 논문과 OKS (oblivious keyword search) 프로토콜에서 이용하고 있는 기술인 OT는 Rabin [9] 에 의해서 소개된 개념이며, 송신자와 수신자 사이의 양자간(two party) 프로토콜이다. 송신자 S는 두 개의 비트들을 가지고 있으며, 수신자 R은 다음의 두 가지 조건을 만족시키며, 두 비트중의 하나의 비트를 얻기를 원한다. (1) 송신자 S는 R이 어느 비트를 얻었는지 알 수 없어야 한다. (2) 수신자 R은 얻은 비트가 아닌 다른 비트의 정보를 알 수 없어야 한다. OKS 프로토콜 역시 T와 사용자 U에 대한 양 자간 프로토콜이다. (1)의 성질에 의해 사용자 U는 자신이 직접 선택한 키워드 w를 포함하고 있는 문서 및 자료를 정보의 노출 없이 검색할 수 있다. (2)의 성질에 의해 사용자 U는 오직 키워드 w를 포함한 문서만 검색 할 수 있다.

본 논문에서는 Ogata와 Kurosawa에 의해서 제안된 OT(oblivious transfer)를 이용한 OKS (oblivious keyword search) 프로토콜을 기반으로 효율적인 EOKS(Efficient Oblivious Keyword Search) 프로토콜을 제안한다. EOKS 프로토콜은 사용자가 데이터 제공자로부터 권한을 얻어서 검색어를 이용하여 공개되어져 있는 암호화된 데이터로 부터 원하는 데이터를 얻는 과정은 OKS프로토콜과 동일하다. 하지만 사용자 측면의 검색을 위한 비교연산을 위한 계산량을 줄여서 효율성을 증가시켰다. 새롭게 제안된 EOKS 프로토콜에서는 비교를 통하여 원하는 데이터를 얻을 때 Test 알고리즘을 통과하는 데이터를 얻으면 알고리즘 수행을 멈추어도 검색어에 관련된 모든 데이터를 얻는 것이 가능하도록 하였다. 이를 위해서 우리는 저장되어지는 암호화 데이터의 구조를 다음과 같이 변형하였다.

$$KD_i = \{SI_i \| C_{i,1},\ldots,C_{i,n_i}\}$$

여기서, w_i는 키워드이고 SI_i는 검색어 w_i에 대한 정보를 숨기고 있는 searchable information 이다. 여기서, E가 안전한 대칭키 암호 알고리즘일 때, $M_{i,j}$가 키워드 w_i와 연관된 데이터일 때 $C_{i,j} = E(M_{i,j})$이다.

일반적으로, 데이터에 관련된 검색어의 수는 여러 개이고 데이터의 수 n이 검색어의 수 m보다 훨씬 많다. 따라서, 데이터를 중심으로 관련된 검색어를 필드(field)구조로 저장하는 OKS 프로토콜에서의 구조보다 검색어를 중심으로 관련된 데이터를 저장하는 것은 사용자가 원하는 정보를 얻은 후에 검색 과정을 멈추어도 원하는 모든 데이터를 얻는 것이 가능하게 하여 검색 속도의 효율성을 증가시킨다. 다만, 데이터에 관련된 검색어가 다수인 경우 저장을 여러 번 해야 하기 때문에 저장공간 측면에서의 효율성은 낮다. 하지만, 검색어를 중심으로 관련된 데이터를 KD_i에 직접 포함시키는 것이 아니라 데이터가 저장된 주소를 KD_i에 포함시켜서 사용자가 해당 주소에서 암호화된 데이터를 얻는 방법을 이용하면 저장공간과 검색속도 측면에서의 효율성을 모두 개선할 수 있다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 Ogata 와 Kurosawa에 의해서 제안되었던 OKS프로토콜을 살펴본 후, 3장에서는 본 논문에서 제안한 프로토콜들의 안전성 정의들과 안전성의 가정(computational assumption)들을 설명한다. 4장과 5장에서는 본 논문에서 제안한 EOKS와 EOKS-I프로토콜을 설명하고 안전성을 증명한다. 마지막으

로 6장에서는 결론을 맺는다.

II. OKS 프로토콜

본 논문에서 제안한 EOKS 프로토콜과 EOKS-I프로토콜의 기반이 되는 Ogata와 Kurosawa 가 제안했던 OKS 프로토콜은 다음과 같다.

1. k-out-of-n OKS 프로토콜

[7]에서는 k-out-of-n OT (OT_k^n) 프로토콜에 근거를 두고 OKS 프로토콜을 제안하였다. OKS_k^n 프로토콜은 데이터 제공자T(database supplier)와 사용자U사이의 양자간 프로토콜로 commitment 단계와 transfer단계로 구성된다. 이 때, W는 키워드의 집합이고 l은 보안변수(security parameter)이다.

commit 단계에서는 데이터 제공자 T가 다음과 같은 형태의 n개의 데이터 $B_1, ..., B_n$을 commit 한다. 이 때, $w_i \in W$이고 c_i는 암호화된 데이터이다.

$$B_i = (w_i, c_i)$$

transfer 단계는 k개의 보조단계로 구성되는데, 각각의 j $(1 \le j \le k)$번째 보조단계는 다음과 같다.

j번 째 보조 단계에서 사용자U는 적당한 키워드 $w_j^* \in W$를 선택하여 검색 결과로 $Search(w_j^*)$를 얻게 된다. 이 때, 키워드 값 w에 대한 검색 결과는 $Search(w) = \{c_i \mid w_i = w \quad \text{for some } i\}$로 정의된다. 그러나 데이터 제공자 T는 사용자 U가 검색에 사용한 검색어 $w_1^*, w_2^*, ..., w_k^*$ 에 대한 정보를 얻을 수 없다. 이에 반하여 사용자 U는 자신이 선택한 검색어 $w_j^* \in W$로 검색하여 얻은 결과 $Search(w_j^*)$를 제외한 어떠한 정보도 얻을 수 없어야 한다. 다음에 사용되는 함수 H는 해쉬 함수이고 G는 유사 랜덤 함수이다. 다음은 OKS 프로토콜에 대한 구체적인 설명이다.

1.1 commit 단계

데이터 제공자T 는 RSA 공개키 (N, e)와 개인키 d 를 생성하여 공개키 (N, e)를 공개한다. 데이터 제공자T는 다음과 같은 형태의 $C_i = \{K_i, E_i\}(1 \le i \le n)$를 계산하여 사용자 U에게 commit 한다.

$$K_i = (H(w_i)^d) \mod N$$

$$E_i = G(w_i \parallel K_i \parallel i) \oplus (0^l \parallel c_i)$$

이 때, $\parallel$ 는 연접(concatenation)이고 0^l은 0의 l비트 열이다.

1.2 transfer 단계

transfer 단계는 k 개의 보조 단계로 구성된다. 각각의 $j(1 \le j \le k)$ 번 째 보조단계는 다음과 같다.

(1 단계) 사용자 U는 키워드 값 w_j^*를 선택한다.

(2 단계) 사용자 U는 랜덤 값 r을 선택하여 다음의 Y 값을 데이터 제공자 T 에게 보낸다.
$Y = r^e H(w_j^*) \mod N$

(3 단계) 데이터 제공자 T는 $Y^d = K' \mod N$ 을 계산하여 사용자 U에게 보낸다.

(4 단계) (a) 사용자 U는 다음의 K 값을 계산한다.
$K = K'/r = H(w_j^*)^d \mod N$

(b) 사용자 U는 $G(w_j^* \parallel K \parallel i)$를 계산하여 commit 된 n개의 데이터 $E_1, ..., E_n$ 에 대하여 다음의 K 값을 계산한다.
$(a_i \parallel b_i) = E_i \oplus G(w_j^* \parallel K_i \parallel i)$

(c) $a_i = 0^l$ 가 되면 사용자 U는 키워드 검색에 성공하게 되고 데이터로 $c_i = b_i$ 를 얻는다.

등호가 성립하지 않으면 사용자U는 키워드 검색에 실패하게 된다.

Ⅲ. 안전성 정의

본 장에서는 EOKS 프로토콜과 EOKS-I프로토콜에 필요한 안전성을 정의하려고 한다. EOKS 프로토콜에서의 안전성은 [7]에서 제안한 OKS 프로토콜에서의 안전성의 모델을 그래도 이용한다.

1. EOKS 프로토콜의 안전성

EOKS프로토콜은 두 가지 안전성의 목적을 갖는다. 첫째는 사용자 안전성(User Security)로 데이터 제공자 T는 사용자가 $i\,(1 \leq i \leq k)$번째 보조 단계에 키워드 검색을 위하여 요청한 query (requesting query)로부터 검색어에 대한 정보를 얻을 수 없도록 한다. 두 번째는 데이터베이스 안전성(Database Security)으로 어떠한 악의적인 사용자$\tilde{U}$도 검색 결과를 제외한 어떠한 정보도 얻을 수 없도록 한다. 데이터베이스의 안전성의 정의는 Universally composable security에 그 기반을 둔다. 이러한 안전성의 성질을 다음과 같이 정의할 수 있다.

정의 1 [EOKS 프로토콜에서의 사용자 안전성]
임의의 i 번 째 키워드 w_i 와 w_i'에 대하여 $w_i \neq w_i'$라고 하자. 어떠한 악의적인 데이터베이스 제공자(database supplier)T도 w_i를 이용하여 생성하여 사용자가 데이터베이스 제공자에게 전송한 결과와 w_i'를 이용하여 생성하여 사용자가 데이터베이스 제공자에게 전송한 결과를 구별하는 것이 계산적으로 불가능(computationally indistinguishable)하면 EOKS 프로토콜은 사용자에 대한 안전성을 보장한다고 정의한다.

정의 2 [EOKS 프로토콜에서의 데이터베이스 안전성]
데이터베이스(Database)의 안전성을 정의하기 위해서 다음과 같은 ideal world를 가정한다.

[Ideal world] 신뢰기관(trusted third party)인 TTP는 데이터 제공자 T로 부터 n개의 데이터 $B_1,\ldots, B_n$를 제공받는다. 사용자 U가 $j\ (1 \leq j \leq k)$ 번 째 검색 단계에서 검색어 w_j 로 검색하고자 할 때, TTP는 사용자 U에게 검색결과인 $Search(w_i)$를 제공한다.

이 때, TTP는 사용자가 검색 할 수 있는 횟수를 k이하로 제한하게 된다. OCKS 프로토콜이 이러한 ideal world 가정하에서 다음의 구별불가능성(Indistinguishability)을 만족하면 database에 대한 안전성을 보장한다고 한다.

[구별불가능성(Indistinguishability)] 임의의 악의적인 사용자 $\tilde{U}$와 임의의 다항식 시간 구별자(distinguisher) D에 대하여 다음의 등식을 만족하는 real world에서의 사용자의 역할을 하는 ideal world의 시뮬레이터(simulator) Δ_S가 존재한다면 구별불가능성(indistinguishability)을 갖는다고 한다.

$$\begin{aligned}&|\,(\Pr(D(\textit{the output of }(\tilde{U}))\\&=1)-(\Pr(D(\textit{the output of }\Delta_S))=1)\,|<\epsilon(l)\end{aligned}$$

이 때, l은 security parameter이고 $\epsilon(l)$은 무시할수 있는(negligible) 함수이다.

정의 3 [EOKS 프로토콜의 안전성]
앞에서 정의한 OCKS 프로토콜에서 사용자의 안전성과 database의 안전성이 만족된다면 EOKS 프로토콜은 안전하다고 정의한다.

Ogata와 Kurosawa는 transfer 단계의 각각의 보조단계에서 검색어들에 대한 정보를 숨기기위해서 RSA 블라인드 서명을 사용한다. 따라서, 데이터 제공자(database supplier) T는 사용자가 검색을 위해서 전송하는 질의(query)를 보고 검색어에 대한 어떠한 정보도 얻을 수 없다. 우리의 EOKS 프로토콜에서도 OKS 프로토콜과 동일하게 전송(transfer) 단계의 각각의 보조단계에서 RSA 블라인드 서명을 사용한다.

2. RSA 블라인드 서명의 안전성과 관련 문제

Bellare et. al은 [6]에서 RSA 블라인드 서명의 안전성과 RSA-KTI 문제의 어려움이 동일하다는 것을 증명했다[6, 19]. 다음은 RSA 블라인드 서명의 안전성에 대한 정의이다. 이때, N은 RSA modulus이고 함수 $H: 0,1^{*} \rightarrow Z_N^{*}$는 일방향 해쉬 함수라 가정한다. 또, KeyGen 은 k를 입력 값으로 받아서 N, e, d를 결과 값으로 주는 RSA 키 생성 알고리즘이다.

정의 4 [RSA블라인드 서명의 안전성]

안전성 파라메터 (security parameter) $k \in N$에 대하여 함수 $m,h: N \rightarrow N$은 k에 대한 함수이고 공격자 F는 RSA- 역(inversion) 오라클 $(\bullet)^d \bmod N$와 해쉬 오라클 $H(\bullet)$에 접근가능하다고 가정할 때, 다음의 Experiment 에 대하여 공격자 F의 advantage는 $Adv_{F,h,m}^{rsa-omf}(k) = pr[E_{F,h,m}^{rsa-omf}(k)=1]$로 정의한다. 안전성 파라메터 k에서 복잡도(time-complexity)가 다항식시간인 공격자 F에 대하여 함수 $Adv_{F,h,m}^{rsa-omf}(k)$가 무시할수있는 함수(negligible)이면 RSA 블라인드 서명은 하나이상의 서명 위조공격에 대하여 다항식시간 내에 안전하다(polynomial-secure against one-more forgery)고 한다.

Experiment $EXP_{F,h,m}^{rsa-omf}(k)$

$(N,e,d) \leftarrow KeyGen(k)$

$((M_1,x_1),...,(M_{m(k)+1},x_{m(k)+1})) <- F^{(\cdot)^d \bmod N, H(\cdot)}(N,e,k)$

만약 다음의 조건들을 모두 만족한다면 결과로 1을 내놓고, 그렇지 않으면 결과로 0을 내놓는다.

- 임의의 $i \in \{1,...,m(k)+1\}$에 대하여, $H(M_i) = x_i{}^e \bmod N$
- 메시지 스트링 M_1, M_2, ..., $M_{m(k)+1}$ 은 모두 다르다.
- 공격자 F눈 RSA-inversion 오라클에 기껏해야 $m(k)$개의 쿼리를 만들수 있다.
- 이 experiment에서 해쉬 오라클 쿼리의 수는 기껏해야 $h(k)$개이다.

RSA-KTI 문제에서 공격자에게 허락된 오라클 호출(calls) 수는 RSA-역(inverse) 값을 계산하고자 하는 대상이 되는 목적값들(target points) 보다 적어도 하나는 적고 공격자는 이러한 가정에서 모든 목적값들(target points)의 RSA-역(inverse) 를 계산하면 공격에 성공하게 된다. 다음은 [6]에서 제시한 RSA-KTI 문제의 개념이다. RSA-역(inverse) 오라클 $(\bullet)^d \bmod N$은 입력 값 $y \in Z_N^{*}$에 대하여 RSA-역(inverse) 값 y^d을 결과로 내놓는다. RSA-KTI 문제를 푸는 공격자에게는 오라클$(\bullet)^d \bmod N$에 접근하는 횟수는 많아야 $m(k)$번으로 제한되고 $m(k)+1$개의 목적값들(target points)이 주어져서 모든 목적값들(target points)에 대한 RSA-역(inverse) 값을 계산하게 된다.

정의 5 [RSA-KTI 문제의 어려움]

안전성 파라메터 (security parameter) $k \in N$에 대하여 함수 $m,h: \mathrm{N} \rightarrow \mathrm{N}$ 은 k에 대한 함수이고 공격자 A는 RSA- 역(inversion) 오라클 $(\bullet)^d \bmod N$에 접근가능하다고 가정할 때, 다음의 Experiment 에 대하여 공격자 A의 성advantage는 $Adv_{A,m}^{rsa-kti}(k) = pr[EXP_{A,m}^{rsa-kti}(k)=1]$로 정의한다. 이 때, 안전성 파라메터 k에서 복잡도(time-complexity)가 다항식시간인 공격자 A에 대하여 함수 $Adv_{F,h,m}^{rsa-kti}(k)$가 무시할수 있는(negligible) 함수이면 RSA-KTI 문제는 어렵다고 정의한다.

Experiment $EXP_{A,m}^{rsa-kti}(k)$

$(N,e,d) \leftarrow KeyGen(k)$

$i=1$에서 $m(k)+1$까지 $(y_i <- Z_N^{*}(N,e,k,y_1,...,y_{m(k)+1})$를 실행한다.

만약 다음의 조건들을 모두 만족한다면 결과로 1을 내놓고, 그렇지 않으면 결과로 0을 내놓는다.

- 임의의 $i \in \{1,...,m(k)+1\}$에 대하여, $y_i = x_i{}^e \bmod N$
- 공격자 A눈 기껏해야 $m(k)$개의오라클 쿼리를 만들수 있다.

정리 1 [6]. RSA-KTI 문제가 어렵다면 RSA 블라인드 서명은 하나 이상의 서명 위조공격(one-more forgery)에 대

하여 다항식 안전성을 갖는다(polynomially-secure).

Ⅳ. EOKS 프로토콜

EOKS프로토콜은 commit 단계와 전송(transfer) 단계로 구성된다. commit 단계에서는 데이터 제공자 T는 다음과 같은 형태의 m개의 데이터 블럭 $KD_1,...,KD_m$을 commit 한다. 여기서, $KD_i=\{SI_i \| C_{i,1},...,C_{i,n_i}\}$, W_i는 키워드, SI_i는 검색어 w_i에 대한 정보를 숨기고 있는 searchable information 그리고 $C_{i,j}=E(M_{i,j})$이다. transfer 단계는 k개의 보조단계로 구성되는데, 각각의 $j(1 \le j \le k)$번째 보조단계는 다음과 같다. j번 째 보조 단계에서 사용자U는 적당한 키워드 w^*를 선택하여 검색결과로 $Search(w^*)=\{C_{i,1},...,C_{i,n_i} | w_i=w^*\}$를 얻게 된다.

다음에 사용되는 함수 H는 해쉬 함수이고 G는 유사 랜덤 함수이다. EOKS프로토콜에 대한 구체적인 설명이다.

1. EOKS 프로토콜

1.1 Commit 단계

데이터 제공자T 는 RSA 공개키 (N,e)와 개인키 d 를 생성하여 공개키 (N,e)를 공개한다. 데이터 제공자T는 다음과 같은 형태의 $KD_i=\{SI_i \| C_{i,1},...,C_{i,n_i}\}(1 \le i \le m)$를 구성하여 사용자 U에게 $KD_1,...,KD_m$를 commit 한다.

$$SI_i = G(w_i \| H(w_i^d \| i) \oplus (0^l \| key_i)$$

$$C_{i,j} = E(M_{i,j}),\ (1 \le j \le n_i)$$

이 때, $\|$ 는 연접(concatenation), 0^l은 0의 l비트 열, $Key_i \in \{0,1\}^{l_1-1}$는 복호화 키이다.

1.2 Transfer 단계

transfer 단계는 k 개의 보조 단계로 구성된다. 각각의 $j(1 \le j \le k)$ 번 째 보조단계는 다음과 같다.

(1 단계) 사용자 U는 키워드 값 w_j^*를 선택한다.

(2 단계) 사용자 U는 랜덤 값 r을 선택하여 다음의 Y값을 데이터 제공자 T에게 보낸다.

$$Y = r^e H(w_j^*) \mod N$$

(3 단계) 데이터 제공자 T는 $Y^d = K' \mod N$ 을 계산하여 사용자 U에게 보낸다.

(4 단계) (a) 사용자 U는 다음의 K값을 계산한다.

$$K = K'/r = H(w_j^*)^d \mod N$$

(b) 사용자 U는 $G(w_j^* \| K \| i)$를 계산하여 commit 된 m개의 데이터 $KD_1,...,KD_m$에 대하여 다음의 값을 계산한다.

$$(a_i \| b_i) = E_i \oplus G(w_j^* \| K_i \| i)$$

(c) 적당한 t $(1 \le t \le m)$에 대해서 $a_t = 0^l$ 가 되면 사용자 U는 키워드 검색에 성공하게 되고 데이터로 복호화 키 key_t를 얻어서 검색어에 관련된 모든 데이터를 얻는다.

$$Search(w_j^*) = \left\{(t, Key_t), M_{t,1},...,M_{t,n_t} \middle| M_{t,\xi} = E_{Key_t}^{-1}(C_{t,\xi}) for 1 \le \xi \le n_t\right\}$$

모든 i에 대하여 $a_i = 0^l$의 등호가 성립하지 않으면 사용자U는 키워드 검색에 실패하게 된다.

2. EOKS 프로토콜의 안전성

정리 **2.** RSA 블라인드 서명 스킴이 안전하다면 OCKS 프로토콜은 사용자의 안전성을 제공한다.

데이터 제공자 T는 모든 전송단계에서 사용자가 보낸 쿼리가 RSA 블라인드 서명으로 블라인드 되었기 때문에 키워드 $w_j(1 \le j \le k)$ 에한 정보를 얻을 수 없다. 즉, RSA-KTI 문제의 안전성에 기반을 두고 사용자의 안전성이 제공된다.

정리 **3.** RSA-KTI 문제가 어렵다면 데이터베이스 안전성(Database Security)을 제공한다.

(증명) 본 논문은 RSA-KTI문제가 어렵다는 가정 하에 데이터베이스 안전성을 보인다. 악의적인 사용자 $\tilde{U}$는 k개의 메시지와 서명의 쌍을 가지고 RSA서명을 위조하려는 공격자이다.

데이터베이스 안전성을 보이기 위해서 real world에서 EOKS프로토콜을 공격하려는 악의적인 사용자 $\tilde{U}$를 이용하려는 ideal world에서의 시뮬레이터 $\triangle_U$를 구성한다. 이 때, ideal world에서의 TTP는 데이터베이스 관리자 T로부터 m개의 메시지 $KD_1,...,KD_m$를 얻어서 $\triangle_U$에게 보낸다. 각각의 j번째 전송(transfer) 보조단계에 TTP는 다음과 같은 결과를 $Search(w_j)$를 $\triangle_U$에게 보낸다.

시뮬레이션을 위해서 $\triangle_U$는 T의 공개키인 (N,e)와 $KD_1,...,KD_m$를 악의적인 사용자 $\tilde{U}$에게 보낸다. $\triangle_U$는 다음과 같이 전송(transfer) 보조단계를 시뮬레이션하고 real world에서의 악의적인 사용자 $\tilde{U}$ 의 결과를 출력한다.

***H* queries.** 해쉬 쿼리를 시뮬레이션하기 위해서 시뮬레이터 $\triangle_U$는 $(w,H(w))=(w,y_w)$로 구성된 H-list를 만든다. 우선 H-list는 공집합이라 가정한다. real world에서의 악의적인 사용자 $\tilde{U}$가 $w\in\{0,1\}^*$를 랜덤오라클 H에 쿼리하면 시뮬레이터 $\triangle_U$는 다음을 수행한다.

1. $w\in\{0,1\}^*$가 이미 H-list에 포함되어 있다면 시뮬레이터 $\triangle_U$는 $(w,H(w))=(w,y_w)$인 y_w를 찾아서 y_w로 대답한다.
2. 그렇지 않다면, 시뮬레이터 $\triangle_U$는 랜덤한 $y_w\in Z_N$를 선택하여 $H(w)=y_w$로 고정하고 해쉬 리스트 H-list를 업데이트한다 .

***G* queries.** real world에서의 악의적인 사용자 $\tilde{U}$는 $(w\parallel H(w)^{\tilde{d}}\parallel i)$를 G함수에 쿼리하기 전에 w를 해쉬함수 H에 쿼리한다고 가정하여도 무리가 없다. 또, 시뮬레이터 $\triangle_U$는 $(w,Search(w))$로 구성된 R-list를 만든다. $\tilde{U}$가 $(w\parallel H(w)^{\tilde{d}}\parallel i)$를 G에 쿼리하면 시뮬레이터 $\triangle_U$는 ideal 해쉬함수 G를 다음과 같이 시뮬레이션(simulation)한다. 이 때, 카운트 수는 $cnt=0$ 그리고 R-list는 공집합으로 초기화 한다.

1. $H(w)^d\neq H(w)^{\tilde{d}} \mod N$이라면, 시뮬레이터 $\triangle_U$는 랜덤한 수 $g\in\{0,1\}^{l_1}$을 선택하여 $G(w\parallel H(w)^{\tilde{d}}\parallel i)=g$로 시뮬레이션한다 .
2. $H(w)^d=H(w)^{\tilde{d}} \mod N$ 라면 시뮬레이터 $\triangle_U$는 R-list를 체크하여 w가 R-list에 포함되어 있지 않다면 $cnt=cnt+1$를 수정하고 다음 단계 3으로 간다. 만약 w가 R-list에 포함되어 있고 인덱스 i가 $Search(w)$에 속한다면, 즉, $(i,Key_i)\in Search(w)$, 시뮬레이터 $\triangle_U$는 $G(w\parallel H(w)^{\tilde{d}}\parallel i)=SI_i\oplus(0^l\parallel Key_i)$로 결과를 내고 단계 3을 생략한다. 만약 w가 R-list에 포함되어 있지만 인덱스 i가 $Search(w)$에 속하지 않는다면, $G(w\parallel H(w)^{\tilde{d}}\parallel i)$를 랜덤하게 결과를 내고 다음 단계 3을 생략한다 .
3. $cnt>k+1$이면 시뮬레이터 $\triangle_U$는 $G(w\parallel H(w)^{\tilde{d}}\parallel i)$를 랜덤하게 결과를 낸다. 그렇지 않다면 시뮬레이터 $\triangle_U$는 TTP에게 w를 쿼리하여 $Search(w)$를 결과로 얻고,$(w,Search(w))$를 R-list에 추가한다.

이 때 $cnt>k+1$가 발생하는 사건을 BAD사건이라고 한다. BAD사건이 발생한다는 것은 real world에서의 악의적인 사용자 $\tilde{U}$가 정당한 $k+1$개의 RSA-blind서명을 생성하는데 성공했다는 것을 뜻한다. 그런데, 앞에서 RSA-KTI 문제가 어렵다고 가정하였고, RSA-KTI 문제가 어려우면 하나이상의 RSA블라인드 서명 위조공격에 대하여 안전하다. 결과적으로, 시뮬레이터 (simulator) A의 결과와 악의적인 사용자 $\tilde{U}$의 결과를 구별할 확률은 무시할수 있는 (negligible) 함수가 된다. 더 나아가서, BAD사건이 발생하지 않았다는 것은 real world에서의 $\tilde{U}$의 결과와 ideal world에서의 시뮬레이터 $\triangle_U$의 결과와 같게 된다.

V. EOKS-I 프로토콜

EOKS-I프로토콜은 키워드를 중심으로 저장할 때, 데이

터를 직접 저장하는 것이 아닌, 데이터가 저장되어져 있는 주소 혹은 인덱스 값을 저장하여 여러 검색어에 관련된 데이터의 경우에 중복 저장되어서 저장의 효율성을 저하시키는 EOKS프로토콜의 단점을 개선하였다. EOKS 프로토콜에 대한 자세한 설명은 다음과 같다.

1. EOKS-I 프로토콜

1.1 Commit 단계

데이터 제공자T는 RSA 공개키 (N,e)와 개인키 d를 생성하여 공개키 (N,e)를 공개한다. 데이터 제공자T는 다음과 같은 형태의 $KI_i=\{SI_i \| IM_i\}(1 \le i \le m)$를 구성하여 사용자 U에게 $KI_1,...,KI_m$를 commit 한다.

$$SI_i = G(w_i \| H(w_{i)}^d \| i) \oplus (0^l \| key_i)$$
$$IM_i = E_{Key_i}((i_1, K_{i,1}),...,(i_{n_i}, K_{i_{n_i}}))$$

이 때, $\|$ 는 연접(concatenation), 0^l은 0의 l비트 열, $i_j(1 \le j \le n_i)$는 키워드 w_i와 연관된 데이터 M_{i_j}에 대한 암호화된 데이터 $C_{i_j}=E_{K_{i_j}}(M_{i_j})$가 저장되어져 있는 주소 혹은 인덱스이다. 또, $Key_i \in \{0,1\}^{l_1-l}$은 IM_i $(1 \le i \le m)$의 복호화 키이다.

1.2 Transfer 단계

transfer 단계는 k 개의 보조 단계로 구성된다. 각각의 j $(1 \le j \le k)$ 번 째 보조단계는 다음과 같다.

(1 단계) 사용자 U는 키워드 값 w_j^*를 선택한다.

(2 단계) 사용자 U는 랜덤 값 r을 선택하여 다음의 Y 값을 데이터 제공자 T 에게 보낸다.
$Y = r^e H(w_j^*) \mod N$

(3 단계) 데이터 제공자 T는 $Y^d = K' \mod N$ 을 계산하여 사용자 U에게 보낸다.

(4 단계) (a) 사용자 U는 다음의 K값을 계산한다.
$K = K'/r = H(w_j^*)^d \mod N$

(b) 사용자 U는 $G(w_j^* \| K \| i)$를 계산하여 commit 된 m개의 데이터 $KI_1,...,KI_m$에 대하 여 다음의 값을 계산한다.
$(a_i \| b_i) = SI_i \oplus G(w_j^* \| K_i \| i)$

(c) 적당한 t $(1 \le t \le m)$에 대해서 $a_t = 0^l$ 가 되면 사용자 U는 키워드 검색에 성공하게 되고 데이터로 복호화 키 key_t를 얻어서 $E_{K_{t_\xi}}^{-1}(C_{t_\xi})$ $(1 \le \xi \le n_t)$ 를 계산하여 w_i와 연관된 데이터 M_{i_j}에 대한 암호화된 데이터 $C_{i_j}=E_{K_{i_j}}(M_{i_j})$가 저장되어져 있는 주소 혹은 인덱스와 각각의 복호화키를 얻는다.

모든 i에 대하여 $a_i = 0^l$의 등호가 성립하지 않으면 사용자U는 키워드 검색에 실패하게 된다.

2. EOKS-I 프로토콜의 안전성

정리 **4.** ***RSA*** *블라인드 서명 스킴이 안전하다면* ***EOKS-I*** *프로토콜은 사용자의 안전성과 데이터베이스 안전성을 제공한다.*

(증명) EOKS-I 프로토콜의 안전성의 증명은 EOKS프로토콜의 증명과 동일하다.

Ⅵ. 결 론

본 논문에서는 Ogata 와 Kurosawa가 제안했던 OKS프로토콜을 기반으로 랜덤 오라클모델에서 검색 속도를 향상시킨 EOKS 프로토콜과 EOKS-I 프로토콜을 제안하였다. OKS 프로토콜은 원하는 검색어에 관련된 데이터를 검색하기 위해서 전체 데이터에 대한 비교연산을 요구한다. 하지만, 본 논문에서 제안한 EOKS 프로토콜과 EOKS-I 프로토콜은 검색어에 대한 비교연산을 수행하다가 원하는 데이터를 얻게되면 중간에 멈추어도 관련된 모든 데이터를 검색할 수 있다. 또, 일반적으로 데이터에 관련된 검색어의

수는 여러개일 때 효율적이다. 하지만 OKS 프로토콜은 모든 데이터마다 검색이 허락된 검색어의 수가 단 하나이다. 본 논문에서 제안한 두 프로토콜에서는 데이터에 관련된 검색어의 수를 여러개로 확장하는 것이 가능하다. 하지만, EOKS 프로토콜에서는 데이터가 중복저장되어야 하는 단점을 갖는다. 이 점을 보완한 것이 두 번째 프로토콜인 EOKS-I 프로토콜이다. EOKS-I 프로토콜에서는 데이터에 관련된 검색어의 수가 여러개이면서 검색속도를 빠르게 효율성을 증가시켰다. 본 논문에서 제안한 프로토콜들은 RSA-KTI 문제에 안전성의 기반을 둔다.

참 고 문 헌

[1] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key Encryption with Keyword Search", EUROCRYPT'04, 2004.

[2] M. Bellare, C. Namprempre, D. Pioncheval,``The Power of RSA Onversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme", Proc. of Finandcial Cryptography 2001, LNCS vol. 2339, pp. 319-338.

[3] Y. C. Chang , M. Mitzenmacher, ``privacy preserving keyword searches on remote encrypted data", ePrint, October 7th 2003.

[4] E. J. Goh, ``secure index", Cryptology ePrint Archive, October 7th 2003.

[5] P. Golle, J. Staddon and B. Waters, ``Secure Conjunctive Keyword Search Over Encrypted Data", Proceedings of the Second International Conference on ACNS:Applied Cryptography and Network Security, 2004.

[6] M. Noar and B. Pinkas, ``Efficient Oblivious trnasfer protocols", 12th Annual Symposium on Discrete Algorithms(SODA), pp 448-457(2001).

[7] W. Ogata, K. Kurosawa, ``Oblivious Keyword Search", Journal of complexity'04, Vol 20. April/Jun 2004.

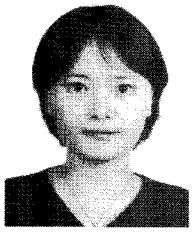
[8] D. Park, K. Kim and P. Lee, "Public key Encryption with Conjunctive Field Keyword Search", WISA'04, LNCS 3325, pp73-86, 2004.

[9] M. Rabin, ``How to exchange secrets by oblivious transfer", Technical Report TR 81, Aiken computation Lab, Harvard University.

[10] D. Song, D. Wagner, and A. Perrige, "Practical Techniques for searches on Encrypted Data", In Proc. of the 2000 IEEE Security and Privacy Symposium, May 2000.

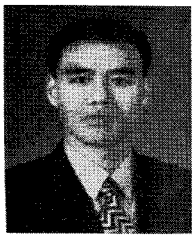
[11] B. R. Waters, D. Balfanz, G. Durfee and D. K. Smetters, ``Building an Encrypted and Searchable Audit Log", 11th Annual Network and Distributed Security Symposium (NDSS '04)}; 2004.

저 자 소 개



이 현 숙

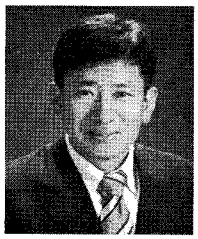
- 1998년 2월 : 단국대학교 수학과 (학사)
- 2000년 2월 : 단국대학교 수학과 (석사)
- 2008년 2월 졸업예정 : 고려대학교 정보경영공학전문대학원 정보보호학과 암호프로토콜 (박사)
- 주관심분야 : 데이터 프라이버시, 사용자 익명성, 프라이버시 관련분야 등



박 종 환

- 1999년 2월 : 고려대학교 수학과 (학사)
- 2004년 2월 : 고려대학교 정보보호대학원 암호프로토콜 (석사)
- 2004년 3월 ~ 현재 : 고려대학교 정보경영공학전문대학원 정보보호전공 박사과정
- 주관심분야 : Pairing-based 암호, 브로드캐스트 암호, 전자서명 등

저 자 소 개



이 동 훈

- 1984년 : 고려대학교 경제학 학사
- 1987년 : University of Oklahoma 전산학과 석사
- 1992년 : University of Oklahoma 전산학과 박사
- 1993년~1997년 : 고려대학교 전산학과 조교수
- 1997년~2001년 : 고려대학교 전산학과 부교수
- 2001년~현재 : 고려대학교 정보보호대학원 교수
- 주관심분야 : 암호이론, 암호프로토콜, USN 이론, 키 교환, 익명성 연구, PET 기술