Oblivious Conjunctive Keyword Search *

Hyun Sook Rhee, Jin Wook Byun, Dong Hoon Lee, and Jongin Lim

Center for Information Security Technologies (CIST),
Korea University, Anam Dong, Sungbuk Gu, Seoul, Korea
{math33, byunstar, donghlee, jilim}@korea.ac.kr

Abstract. We study the problem of keyword search in which a server contains various multimedia contents and a user of server wishes to retrieve some multimedia item containing specific keywords without revealing to the server which item it is. Recently, Ogata and Kurosawa introduced an interesting keyword search scheme called oblivious keyword search by using the notion of oblivious transfer. However, only one keyword can be searched in each query, hence the scheme cannot provide a conjunctive keyword search which finds items containing each of several keywords. In this paper, we firstly design a conjunctive keyword search by using the oblivious transfer, and present *oblivious conjunctive keyword search* (for short, OCKS). We prove that OCKS protocol is secure under the intractability of RSA known target inversion problem.

Keywords : Conjunctive Keyword Search, Storage system, Oblivious Transfer, Anonymity, Privacy.

1 Introduction

As the amount of information to be stored and managed on the Internet rapidly increases, the importance of storage system such as a database is increasingly growing. As a result, ensuring privacy for the stored data on the storage system is becoming one of the most urgent challenges in database research and industry.

Today, there are two important privacy issues in database research. The first important issue is that a user needs to be assured that his data stored on the database is protected against data thefts from outsiders. To simply resolve this problem, we may implement a cryptographic encryption module in the database management system (DBMS), and the module in the DBMS performs encryptions and decryptions for the stored data. Actually, the most real database systems such as `Oracle` $8i$ and `MS Access` provide themselves a cryptographic module in their DBMS, hence the first issue can be resolved without any additional implementation costs such as private key management module certainly.

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)

The second issue is that the data need to be protected even from an insider system manager such as a system administrator of database if the database system cannot be trusted. When the storage system is not trustworthy, users may ensure the privacy of their data by storing it encrypted. The users of system should securely manage its encryption keys without revealing to the insider system manager. Originally, storing data in an encrypted form was proposed in Blaze's Cryptographic File System (CFS) [2], and expanded on later systems [2, 8, 23, 13]. If data are stored in an encrypted form, it can be protected from leaking by the server because it does not know the decryption key. In addition, there is no need to encrypt data again for confidentiality when it is sent on the public network. However, secure encryption makes documents look random, and unreadable to anyone other than the users holding the decryption keys, hence the server is unable to determine which encrypted documents contain specific keywords. To resolve this problem, many efficient search protocols over encrypted data have been suggested in various scenarios [4, 7, 10, 12, 17, 18, 21, 22].

In this paper, we study the problem of keyword search in a setting where a database supplier uploads data onto its database. Hence a system manager is also a database supplier and has no motivation to be an inside attacker. For example, consider a setting of keyword search where a server itself contains various multimedia contents and a user of server wishes to retrieve some multimedia item containing specific keywords without revealing to the server which item it is. Recently, in this setting, Ogata and Kurosawa first introduced an interesting keyword search scheme called oblivious keyword search (for short, OKS) by using the notion of oblivious transfer (for short, OT) [17]. Originally, the notion of OT was introduced by Rabin [20], and have been subsequently defined in different forms [9].[1] The OT protocol is a two party protocol between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. $\mathcal{S}$ has two bits, and $\mathcal{R}$ wishes to get one of them satisfying the followings properties: (1) $\mathcal{S}$ does not know which bit $\mathcal{R}$ obtained and (2) $\mathcal{R}$ does not know any information about the bit that he did not obtain. The OKS protocol is also a two party protocol between a database supplier $\mathcal{T}$ and a user $\mathcal{U}$ based on the above properties. By the property of (1), the user $\mathcal{U}$ is able to search and retrieve the data containing keyword w chosen by user $\mathcal{U}$ without revealing the keyword w to $\mathcal{T}$. On the other hand, the user $\mathcal{U}$ learns only the data containing keyword by the property (2).

1.1 Our Contributions

As mentioned by Ogata and Kurosawa themselves, the number of keywords for searching in their OKS protocol is restricted by only one. Hence a user cannot use boolean combinations of keywords, which should be an available functionality. One may think that a conjunctive keyword search can be derived from the multiple executions of only single keyword search scheme. In this case, however, the server should find all documents containing each individual keyword by using the single keyword search, check the intersection set of all documents, then

[1] All these OT variants were shown to be equivalent to one another [9, 5].

return the results to the user. This approach requires the server high computational cost and redundancy due to duplicated comparisons and search, hence the design of conjunctive keyword search ensuring efficiency and security is never an easy task.

In this paper, we propose an oblivious conjunctive keyword search protocol(OCKS) by using the notion of OT. First, we present a new security model for OCKS, and define its security for OCKS. We then prove that our suggested OCKS protocol is secure if the RSA known target inversion problem [6] is hard.

1.2 Organization

This paper is organized as follows. In Section 2, we overview an OKS protocol presented by Ogata and Kurosawa. In Section 3, we describe our security model, security definition and computational assumption for constructing OCKS protocol. In Section 4, we present an OCKS protocol and show that the OCKS protocol is secure based on the hardness of RSA known target inversion problem. We conclude in Section 5.

2 OKS protocol

Now we review the OKS protocol proposed by Ogata and Kurosawa [17], which is a building block of the proposed scheme.

2.1 Description of k-out-of-n OKS

The OKS protocol is based on the k-out-of-n OT protocol (OKS_k^n) suggested in [17]. The OKS_k^n protocol consists of database supplier $\mathcal{T}$ and a user $\mathcal{U}$. Let W be the set of keywords and l be a security parameter. The OKS_k^n protocol consists of a commit phase and a transfer phase.

In the commit phase, $\mathcal{T}$ commits n data $B_1, ..., B_n$ such that

$$B_i = (w_i, c_i)$$

where $w_i \in W$ and c_i is an encrypted data. We define the search result for a keyword w by

$$\text{Search}(w) = \{c_i \mid w_i = w \ \text{ for } \ \text{some } i\ \}$$

The transfer phase consists of k subphases. At each subphase j $(1 \leq j \leq k)$, $\mathcal{U}$ can choose a keyword $w_j^* \in W$ adaptively and learn $Search(w_j^*)$. However, $\mathcal{T}$ should gain no information on w_1^*, w_2^*, ..., w_k^* and $\mathcal{U}$ should not learn anything except the result of $Search(w_j^*)$. Let G be a pseudo-random generator and H be a cryptographic secure hash function. The OKS_k^n protocol works as follows.

- **[Commitment Phase]** $\mathcal{T}$ generates a public key (N, e) and a secret key d of RSA. $\mathcal{T}$ publishes (N, e). Next, $\mathcal{T}$ computes $\mathcal{C}_i = \{K_i, E_i\}$ for $1 \leq i \leq n$, as follows.

$$\mathcal{C}_i = \begin{cases} K_i = (H(w_i))^d \ mod \ N \\ E_i = (G(w_i \| K_i \| i)) \oplus (0^l \| c_i) \end{cases}$$

where $\|$ denotes concatenation and 0^l is a l-bit string with 0's. $\mathcal{T}$ sends $E_1, ..., E_n$ to $\mathcal{U}$

- **[Transfer Subphase]** At each transfer subphase j $(1 \leq j \leq k)$, $\mathcal{U}$ performs the followings.
  (1) $\mathcal{U}$ chooses a keyword w_j^*.
  (2) $\mathcal{U}$ chooses a random element r and computes Y where,

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

  $\mathcal{U}$ sends Y to $\mathcal{T}$.
  (3) $\mathcal{T}$ computes $K' = Y^d \ \ mod \ N$ and sends it to $\mathcal{U}$.
  (4) $\mathcal{U}$ computes K as follows.

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

  For $i = 1, ..., n$, $\mathcal{U}$ computes $G(w_j^* \| K \| i)$, and gets the following.

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

  If $a_i = 0^l$ $(1 \leq i \leq n)$ then $\mathcal{U}$ succeeds and gets $c_i = b_i$. Otherwise, $\mathcal{U}$ outputs a failure message.

3 Security Definitions

In this section, we describe the security definitions for OCKS protocol. We slightly extend the security model of [17] to be suitable for conjunctive keyword search. In order to treat several keywords, we assume an untrusted storage system has an actual database which has several records, each of which contains keyword fields. We assume that there are m keyword fields for each encrypted documents.

3.1 Security Definition of OCKS

We define two security goals based on the definition of [17]. One is a user security. It guarantees that an untrusted database supplier $\mathcal{T}$ should not learn any information about keywords from the user's requesting query in the i-th subphase $(1 \leq i \leq k)$. The other is a database security. It guarantees that a malicious user $\tilde{\mathcal{U}}$ should get nothing except any search results. We formally define as follows.

Definition 3.1.1 [User Security in OCKS protocol] Suppose that $\boldsymbol{w_i} = w_{ij_1}, w_{ij_2}, ..., w_{ij_{d_i}}$ and $\boldsymbol{w_i^*} = w^*_{ij_1}, w^*_{ij_2}, ..., w^*_{ij_{d_i}}$ are arbitrary keyword strings in the i-th subphase and $(\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_k}) \neq (\boldsymbol{w_1^*}, \boldsymbol{w_2^*}, ..., \boldsymbol{w_k^*})$. The OCKS protocol is secure for the user if for any malicious database supplier $\mathcal{T}$, the view of $\mathcal{T}$ for any keyword strings $\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_k}$ and the view of $\mathcal{T}$ for any keyword strings $\boldsymbol{w_1^*}, \boldsymbol{w_2^*}, ..., \boldsymbol{w_k^*}$ are computationally indistinguishable.

Definition 3.1.2 [Database Security in OCKS protocol] Consider the following *ideal world*. A trusted third party (TTP) first receives $(B_1, B_2, ..., B_n)$ from $\mathcal{T}$. TTP next tells the user $\mathcal{U}$ the search result $\bigcap_{t=1}^{d_j} Search(w^*_{j_t})$ on the request $\boldsymbol{w_j^*} = w^*_{j_1}, w^*_{j_2}, ..., w^*_{j_{d_j}}$ for any $(1 \leq j \leq k)$. We say that a protocol is secure for the database if the following condition is satisfied:

- **[Indistinguishability]** For any malicious user $\tilde{\mathcal{U}}$, there exists a simulator $\triangle_S$ that plays the role of a user in the ideal world such that for any polynomial time distinguisher $\mathcal{D}$,

  $$|Pr(\mathcal{D}(the\ output\ of\ \tilde{\mathcal{U}}) = 1) - Pr(\mathcal{D}(the\ output\ of\ \triangle_S) = 1)| < \epsilon(l).$$

  where $\epsilon(l)$ is a negligible function and l is a security parameter.

Definition 3.1.3 [OCKS Security] We say that a protocol is a secure OCKS protocol if it satisfies both user and database security in OCKS protocol.

Ogata and Kurosawa use the RSA blind signature scheme to hide keywords in the transfer phase. Thus, the user's requesting keywords are blinded from the malicious database supplier $\mathcal{T}$. We also use the RSA blind signature scheme in each transfer subphase.

3.2 Security of RSA Blind Signature and its related problem

In [6], Bellare et al. proved that the RSA-blind signature scheme is secure as long as the RSA known-target inversion problem is hard [6, 19]. The following is a definition about the security of RSA-blind signature scheme. The function H is an one-way hash function, where $H : \{0,1\}^* \longrightarrow Z_N^*$. We let KeyGen be the RSA key generation algorithm which takes k as input and returns the values N, e and d, where N is a k-bit RSA modulus.

Definition 3.2.1 [Blind Signature Security] Let $k \in \mathsf{N}$ be the security parameter, and let $m, h : \mathsf{N} \rightarrow \mathsf{N}$ be functions of k. Let $\mathcal{F}$ be a *forger* who has access to RSA-inversion oracle and a hash oracle, denoted $(\cdot)^d \bmod N$ and $H(\cdot)$, respectively. Consider the following experiment.

Experiment $\mathsf{EXP}^{rsa\text{-}omf}_{\mathcal{F},h,m}(k)$.

$(N, e, d) \stackrel{R}{\leftarrow} \mathsf{KeyGen}(k)$

$((M_1, x_1), ..., (M_{m(k)+1}, x_{m(k)+1})) \longleftarrow \mathcal{F}^{(\cdot)^d modN, H(\cdot)}(N, e, k)$

If the following are all true, return 1 else return 0:

- $\forall i \in \{1, ...t, m(k)+1\}$: $H(M_i) = x_i^e \bmod N$
- Message strings $M_1, M_2, ..., M_{m(k)+1}$ are all distinct.
- $\mathcal{F}$ makes at most $m(k)$ quries of message to its RSA-inversion oracles.
- The number of hash-oracle queries made in this experiment is at most $h(k)$

We define the advantage of the forger $\mathcal{F}$ via

$$\mathbf{Adv}^{rsa\text{-}omf}_{\mathcal{F},h,m}(k) = \Pr\ [\ \mathsf{EXP}^{rsa\text{-}omf}_{\mathcal{F},h,m}(k)=1].$$

The RSA blind signature scheme is *polynomial-secure against one-more forgery* if the function $\mathbf{Adv}^{rsa\text{-}omf}_{\mathcal{F},h,m}(\cdot)$is negligible for any forger $\mathcal{F}$ whose time-complexity is polynomial in the security parameter k.

In RSA known target inversion (for short, **RSA-KTI**) problem, the number of oracle calls allowed to the adversary is just one fewer than the number of target points, so that to win it must compute the RSA-inverse of all target points. The following is a notion of **RSA-KTI** problem in [6]. We denote by $(\cdot)^d \bmod N$ the oracle that takes input $y \in Z_N^*$ and returns its RSA-inverse y^d. An adversary solving the known-target inversion problem is given oracle access to $(\cdot)^d \bmod N$ and is given $m(k)+1$ targets where $m : N \longrightarrow N$. Its task is to compute the RSA-inverses of all the targets while submitting at most $m(k)$ queries to the oracle.

Definition 3.2.2 [Hardness of RSA-KTI problem] Let $k \in \mathsf{N}$ be the security parameter, and let $m, h : \mathsf{N} \rightarrow \mathsf{N}$ be functions of k. Let $\mathcal{A}$ be an adversary who has access to an $(\cdot)^d \bmod N$. Consider the following experiment:

Experiment $\mathsf{EXP}^{rsa\text{-}kti}_{\mathcal{A},m}(k)$.

$(N, e, d) \overset{R}{\leftarrow} \mathsf{KeyGen}(k)$

For $i = 1$ to $m(k)+1$ do $y_i \overset{R}{\leftarrow} Z_N^*(N, e, k, y_1, ..., y_{m(k)+1})$

If the following are both true then return 1 else teturn 0

- $\forall i \in \{1, ..., m(k)+1\}$: $y_i = x_i^e \ mod\ N$
- $\mathcal{A}$ makes at most $m(k)$ oracle queries.

We define the advantage of $\mathcal{A}$ via

$$\mathbf{Adv}^{rsa\text{-}kti}_{\mathcal{A},m}(k) = \Pr\ [\mathsf{EXP}^{rsa\text{-}kti}_{\mathcal{A},m}(k)=1]$$

The **RSA-KTI** problem is said to be hard if the function $\mathbf{Adv}^{rsa\text{-}kti}_{\mathcal{A},m}(k)$ is negligible for any adversary $\mathcal{A}$ whose time-complexity is polynomial in the security parameter k.

Proposition 3.2.3 [6]. If **RSA-KTI** is hard, then **RSA** blind signature scheme is polynomially-secure against one-more *forgery*. □

4 OCKS protocol

In this section, we construct an OCKS protocol. We show that OCKS protocol is secure if the one-more RSA-inversion problem is hard. The OCKS protocol is illustrated in Fig.1, and works as follows.

- **[Commitment Phase]** $\mathcal{T}$ generates a public key (N, e) and a secret key d of RSA. We assume that the value k_i is a decryption key of encrypted data $c_i = E_{k_i}(D_i)$, where $D_i = \{w_{i1}, ..., w_{im}\}$. Let $keygen_i(x) = (x - r_{i1}) \times \cdot \cdot \cdot \times (x - r_{im}) + k_i$ be a key generation function which provides a decryption key k_i corresponding to data c_i, where m is the number of keyword fields. Let $\mathsf{leng} : Z_N \longrightarrow Z_q$ is an ideal hash function and $f_k : Z_N \longrightarrow Z_N$ is a pseudo random function. For every $i = 1, ..., n$ and $j = 1, ..., m$, $\mathcal{T}$ computes the followings.
(**Step 1**) $\mathcal{T}$ randomly chooses $r'_i \in Z_N$.
(**Step 2**) $\mathcal{T}$ makes $\mathsf{OCSI}_{ij} = f_k(w_{i,j})^d r'_i \ \ mod\ N$, where OCSI_{ij} is the oblivious conjunctive keyword searchable information for w_{ij}.
(**Step 3**) $\mathcal{T}$ computes $\mathsf{leng}({r'}_i^j) = r_{ij}$ and sets $y_{ij} = g^{r_{ij}}$.
(**Step 4**) $\mathcal{T}$ constructs $B_i = c_i \parallel y_{i1}, y_{i2}, ..., y_{im} \| \ keygen_i(x) \parallel \mathsf{OCSI}_{i1}, ..., \mathsf{OCSI}_{im}$.
$\mathcal{T}$ commits $B_1, B_2, \cdot \cdot \cdot, B_n$.

$\mathcal{T}$		$\mathcal{U}$
	$\xrightarrow{B_1,...,B_n}$	
		Randomly choose $w^*_{j_1}, w^*_{j_2}, ..., w^*_{j_{d_j}} \longleftarrow W$ Choose a random value $r \in Z_N$ Compute $Y = r^e \times f_k(w^*_{j_1}) \times \cdot \cdot \cdot \times f_k(w^*_{j_{d_j}})$
	$\xleftarrow{Y}$	
Compute $K' = Y^d \ \ mod\ N$		
	$\xrightarrow{K'}$	
		Compute $K = K'/r$ Compute $\mathsf{OCSI}_{i_{j1}} \times \cdot \cdot \cdot \times \mathsf{OCSI}_{i_{jd_j}}/K$, and set it as $(r'_i)^{d_j}$. Compute $\mathsf{leng}((r'_i)^{d_j}) = (r_{id_j})$ If $g^{r_{id_i}} = y_{id_i}$ then user $\mathcal{U}$ gets the decryption key $k_i = keygen_i(r_{id_j})$

Fig.1. OCKS Protocol

- **[Transfer Phase]** The transfer phase consists of k subphases. $\mathcal{U}$ learns a conjunctive keyword search result $\bigcap_{t=1}^{d_j} search(w^*_{j_t})$ as follows.
(**Step 1**) $\mathcal{U}$ choose d_j keywords $\boldsymbol{w^*} = w^*_{j_1}, w^*_{j_2}, ..., w^*_{j_{d_j}}$ on W adaptively

(**Step 2**) $\mathcal{U}$ chooses a random element $r \in Z_N$ and computes Y as follows. $\mathcal{U}$ sends Y to $\mathcal{T}$.

$$Y = r^e \times f_k(w^*_{j_1}) \times \cdots \times f_k(w^*_{j_{d_j}}).$$

(**Step 3**) $\mathcal{T}$ computes $K' = Y^d \ \ mod\ N$ and sends it to $\mathcal{U}$.
(**Step 4**) $\mathcal{U}$ computes $K = K'/r$.

(**Step 5**) For every i from 1 to $n, \mathcal{U}$ computes r_{id_j} as follows.

$$\begin{cases} \mathsf{OCSI}_{i_{j1}} \times \cdots \times \mathsf{OCSI}_{i_{jd_i}}/K \Longrightarrow (r'_i)^{d_j}. \\ \mathsf{leng}((r'_i)^{d_j}) \Longrightarrow r_{id_j}. \end{cases}$$

If the following equation (1) is satisfied, then we determine that the data D_i contains the keywords $w^*_{j1}, w^*_{j2}, ..., w^*_{jd_j}$.

$$g^{r_{id_j}} = y_{id_j}. \tag{1}$$

And $\mathcal{U}$ can get a decryption key k_i for c_i by the equation (2).

$$k_i = keygen_i(r_{id_j}). \tag{2}$$

User $\mathcal{U}$ can get data D_i as decrypting the data c_i with k_i.

4.1 Security of OCKS Protocol

Theorem 1. The above OCKS protocol satisfies *user security* if the security of RSA blind signature scheme.

Proof. For every subphase i $(i = 1, 2, ..., k\ )$, $\mathcal{T}$ has no information on $\boldsymbol{w^*_1}, \boldsymbol{w^*_2}, \cdots, \boldsymbol{w^*_k}$, where $\boldsymbol{w^*_i} = w_{ij_1}, w_{ij_2}, ..., w_{ij_{d_i}}$ since they are blinded by the RSA blind signature scheme. Therefore, user security is satisfies under an intractability of RSA known target inversion problem. □

Theorem 2. The above OCKS protocol satisfies *database security* if the **RSA-KTI** is hard.

Proof. We next prove the database security by assuming that **RSA-KTI** is hard. We show that there exists a simulator $\triangle_S$ that plays the role of a user in ideal world as follows.

[**Commitment Phase**] $\triangle_S$ generates (N, e, d) and sends (N, e) to $\tilde{u}$. $\triangle_S$ also chooses $B_1, ..., B_n$ and sends them to $\tilde{u}$.

- For every $i = 1, ..., n$, $\triangle_S$ randomly chooses c_i and $\mathsf{OCSI}_{i1}, ..., \mathsf{OCSI}_{im}$.
- For every i, j $(1 \le i \le n,\ 1 \le j \le m)$, $\triangle_S$ randomly chooses r_{ij} and decryption key k_i, and computes $y_{ij} = g^{r_{ij}}$ and sets $keygen_i(x) = (x - r_{i1}) \times \cdots \times (x - r_{im}) + k_i$.

- $\triangle_S$ commits $B_i = c_i \parallel y_{i1}\ , y_{i2}\ , ..., y_{im} \parallel keygen_i(x) \parallel \mathsf{OCSI}_{i1}, ..., \mathsf{OCSI}_{im}$.

[Transfer Phase] $\triangle_S$ can perfectly simulate $\mathcal{T}$ because $\triangle_S$ knows (N, e, d).

$\triangle_S$ simulates f_k as follows. If $\tilde{u}$ queries $w^*_{i_{j1}}, w^*_{i_{j2}}, ..., w^*_{i_{jd_i}}$ to f_k for the first time, then $\triangle_S$ chooses a random value over Z_N, return it as the value of $f_k(w^*_{i_{j1}}) * f_k(w^*_{i_{j2}}) * \cdots * f_k(w^*_{i_{jd_i}})$, where the vector of keywords $\boldsymbol{w_i} = w^*_{i_{j1}}, w^*_{i_{j2}}, ..., w^*_{i_{jd_i}}$ in i-th subphase.

We assume that ***QA-list*** contains the pair of query and the set of result ($\boldsymbol{w_i}, \bigcap_{t=1}^{d_i} Search(w^*_{ij_t})$) $(1 \le i \le k)$. To simulate an ideal hash function leng, we need a ***QA-list***. $\triangle_S$ simulates an ideal hash function leng as follows.

- Let $cnt = 0$.
- If $cnt = 0$ then ***QA-list*** is empty.
- WLOG, before $\tilde{u}$ queries Y to $\mathcal{T}$, $\tilde{u}$ queries $w^*_{i_{jt}}$ to f_k, for every $t = 1, ..., d_i$.

  (a) If $(\boldsymbol{w_i}, \bigcap_{t=1}^{d_i} Search(w^*_{ij_t})) \in$ ***QA-list***, then go to step (c).
  else $cnt = cnt + 1$

  (b) If $cnt > cnt + 1$, then a simulator $\triangle_S$ sets $\bigcap_{j=1}^{d_i} Search(w^*_{ij})$ at random. Otherwise, $\triangle_S$ queries $\boldsymbol{w_i}$ to the TTP and receives $\bigcap_{j=1}^{d_i} Search(w^*_{ij})$. $\triangle_S$ adds $(\boldsymbol{w_i}, \bigcap_{j=1}^{d_i} Search(w^*_{ij}))$ to ***QA-list***.

  (c) For i $(1 \le i \le n)$, if c_i is included in $\bigcap_{j=1}^{d_i} Search(w^*_{ij})$, then sets the values of *leng* as follows.
  - $\triangle_S$ computes $(r'_i)^t = \mathsf{OCSI}_{ij_1} \times \mathsf{OCSI}_{ij_2} \times \cdots \times \mathsf{OCSI}_{ij_{d_i}} / (f_k(w_{ij_1}))^d \times (f_k(w_{ij_2}))^d \times \cdots (f_k(w_{ij_{d_i}}))^d$ and then $\triangle_S$ sets $\mathsf{leng}((r'_i)^{j_t}) = r_{ij_t}$.
  - For all $t = d_{i+1}, ..., m$, $\triangle_S$ sets $\mathsf{leng}((r'_i)^j t)$ at random.

  Otherwise, $\triangle_S$ sets the value leng at random.

Let **BAD** be the event that $cnt > cnt + 1$. If **BAD** dose not occur, then $\triangle_S$ simulates hash function leng perfectly. Note that $\Pr(\mathbf{BAD})$ means the probability that $\tilde{\mathcal{U}}$ succeeds in the one-more forgery attack on the **RSA** blind signature scheme. From Proposition 3.2.3, it is negligible if the **RSA-KTI** is hard. Consequently, the output of $\triangle_S$ and $\tilde{u}$ are indistinguishable if the **RSA-KTI** is hard. $\square$

5 Concluding Remarks

In this paper, we first gave an oblivious conjunctive keyword search to the scheme of [17] in the random oracle model. We prove its security under the RSA known target inversion problem. It would be a good future work to design a secure OCKS protocol without random oracle assumption.

Acknowledgement

We deeply thank Hyun-A Park for helpful discussions of an earlier version of this paper. We also very thank anonymous referees of WISA05' for valuable comments.

References

1. R. Agrawal and R. Srikant,"Privacy-Preserving Data Mining", *In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 439-450, 2000.
2. M. Balze, "A Cryptographic file system for UNIX.", *Processings of 1st ACM Conference om Communications and Computing Security*, 1993.
3. S. Bellovin , W. Cheswick, "Privacy-enhanced searches using encrypted bloom filters", *Cryptology ePrint Archive*, Report 2004/022, Feb 2004.
4. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key Encryption with Keyword Search", *EUROCRYPT'04*, 2004.
5. D. Brassard, C. Crepeau, and J. M. Robert, "All-or-Nithing Disclosure of Secrets", *Crypto'86*, Springer-Verlag, 1987, pp. 234-238.
6. M. Bellare, C. Namprempre, and 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.
7. Y. C. Chang, M. Mitzenmacher, "privacy preserving keyword searches on remote encrypted data",*ePrint*, October 7th 2003.
8. G. Cattaneo, G. Persiano, A. Del Sorbo, A. Cozzolino, E. Mauriello, and R. Pisapia, "Design and implementation of a transparent cryptographic file system for UNIX", *Techincal Report, University of Salerno*, 1997.
9. S. Even, O. Goldreich, and A. Lempel, "A Randomized Protocol for Signing Contracts", *comm. of ACM*, 28:637-647, 1985.
10. E. J. Goh, "secure index", *ePrint*, October 7th 2003.
11. P. Golle, M. Jakobsson, A. Juels, and Paul Syverson, "Universal Re-encryption for Mixnets", *In proceedings of CT-RSA 2004*, 2004.
12. 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.
13. J. Hughes and D. Corcoran, "A nuiversal access, smart-card-based, secure fiel system.", *Atlanta Linux Showcase* , October 1999.
14. A. John , R. Peter, "Electric Communication Development", *Communications of the ACM* ,40,May 1997, pp. 71-79. 48-63, 2002.
15. K. Kurosawa, "Multi-recipient Public-Key Encryption with Shortened Ciphertext", *In proceedings of PKC 2002*, LNCS 2274, pp. 48-63, 2002.
16. M. Noar and B. Pinkas, "Efficient Oblivious trnasfer protocols", *12th Annual Symposium on Discrete Algorithms(SODA)*, pp 448-457(2001).
17. W. Ogata and K. Kurosawa, "Oblivious Keyword Search", *Journal of complexity'04*, Vol 20. April/Jun 2004.
18. D. Park, K. Kim, and P. Lee, "Public key Encryption with Conjunctive Field Keyword Search", *WISA'04*, LNCS 3325, pp73-86, 2004.
19. D. Pointcheval and J. P. Stern, "Provably secure blind signature schemes", *Proc. of Asiacrypt'96*, LNCS Vol. 1163, pp 252-265, 1996.

20. M. Rabin, "How to exchange secrets by oblivious transfer", *Technical Report TR 81*, Aiken computation Lab, Harvard University.
21. 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.
22. 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.
23. E. Zadok, I. Badulescu, and A. Shender, "Cryptfs : A stackable vnode level encryption fiel system.", *Technical Report CUCS-021-98*: 1998.