MULTI-USER SEARCHABLE ENCRYPTION SCHEME WITH USER REVOCATION

Chia-Chi Wu† --- Iuon Lin¹ --- Ya-Hui Liu³
¹ Department of Information Management, National Defense University, Taoyuan, R.O.C.
² Department of Management Information Systems, Nation Chung Hsing University, Taichung, Taiwan R.O.C; Department of Photonics and Communication Engineering, Asia University, Taichung, Taiwan, R.O.C.
³ Department of Management Information Systems, Nation Chung Hsing University, Taichung, Taiwan R.O.C

ABSTRACT
We improve the previous method to add the function of user revocation in searchable encryption scheme. When document owner doesn’t want to share someone he had shared, he can revoke the user who can’t retrieve the specific document. More importantly, the revocation process must not affect other authorized users, the scheme has to keep a low maintenance cost.

Keywords: Access control, Cloud computing, Information security, Multi-user, Searchable encryption, Privacy.

1. INTRODUCTION
In the most of existing searchable encryption schemes, if the document owner wants to share the document with other users called authorized user, the secret key needed to secure transmit to authorized users. Only the authorized user can retrieve the document. The intuitive way shows as follow [1]:

Document owner encrypts the secret key of the document by user’s individual public key. The way increases the document owner’s computation overload which needs to encrypt the document N times. If the document shares with N users. The important issue is user revocation in our scheme. When document owner doesn’t want to share someone he had shared, he can revoke the user who can’t retrieve the specific document. More importantly, the revocation process must not affect other authorized users, the scheme has to keep a low maintenance cost.
2. RELATED WORK

According to a survey [2], the searchable encryption scheme is classified into four architectures. The paper focuses on Single owner/Multiple users (S/M). Single owner uses his secret key to create searchable content. A group of owner-defined can generate search token to search and retrieve the document.

The proposed scheme is based on star-based architecture proposed by Lin, et al. [3]. Our scheme uses the concept to let a document can dynamically authorize a user to retrieve document or revoke a user. The document user uses $e_0$ to encrypt the secret key which encrypted document. As the results, whether the malicious cloud server, any attacker, or other unauthorized user can’t retrieve document. Only the authorized users and owner can retrieve and decrypt the document.

3. OUR METHOD

Some searchable encryption schemes have been proposed which can be classified based on secret key cryptography [4]; [5] or based on public key cryptography [6]; [7]. We use the concept of RSA public key cryptosystem [8] was proposed in 1977.

3.1. Key Assignment

The objective of the scheme is to share document to many users, and making users also can execute searchable encryption. Assume that an owner wants to share documents with $U_1, U_2, ... , U_m$. If the document owners want to share each document for someone among the user group, he needs to assign a secret key $d_1, d_2, ... , d_m$ to each user. The steps of key assignment show as follows:

1. Document owner randomly chooses two $m$ distinct large primes $(p_1, q_1)$ for $U_1$, $(p_2, q_2)$ for $U_2$, ..., $(p_m, q_m)$ for $U_m$. $m$ means the total number of users.
2. Document owner computes $p_m$ multiple $q_m$ to $N_m$. The value of $p_m$ and $q_m$ keeps secret, but $N_m$ makes public. Document owner computes the value of $\varphi(N_m)$ for each $N_m$. The formula for $\varphi(N_m)$ is equal to $(p_m-1)(q_m-1)$.

Next, computing the least common multiple $L_0$ of each $\varphi(N_m)$ which equals to $L_0 = \text{LCM}(\varphi(N_1), \varphi(N_2), ..., \varphi(N_m))$

3. Document owner chooses a large prime $e_0$ that must satisfy two requirements as follows: $e_0 < \min(\varphi(N_1), \varphi(N_2), ..., \varphi(N_m))$

4. Choosing a $d_0$ using the extended Euclidean algorithm. It also must satisfy two requirements as follows: $e_0 \times d_0 = 1 (mod L_0)$
   \[ d_0 > \max(\varphi(N_1), \varphi(N_2), ..., \varphi(N_m)) \]

5. Using $d_0$ to generate secret key for $d_1, d_2, ..., d_m$ for every user $U_1, U_2, ..., U_m$ by the following formula:
   \[ d_m = d_0 \varphi(N_m) \]

6. Document owner sends secret key $d_1, d_2, ..., d_m$ to $U_1, U_2, ..., U_m$ safely.

7. Document owner has a $e_0$ to encrypted the secret key of document only the authorized user can obtain it. Users own a secret key and $N_m$ is public anybody can get it.

3.2. Broadcast Search Key

Assume that the document owner want to authorize some users $AU = \{U_1, U_2, ..., U_m\}$ can search using keywords. To this purpose, document owner needs to broadcast the search key $k_s$ to them. The expression shows as follows:

\[ k_s' = k_s^{e_0} \mod \prod_{U_m \in AU} N_m \]

Every authorized users can use their $d_m$ to get the search key $k$. The following expression shows as follows:

\[ k_s = (k_s' \mod N_m)^{d_m} \mod N_m \]
Finally, the authorized users can get the search key. The people who is not authorized he don’t have the search ability.

3.3. Document Authorize

This step authorizes specific users can retrieve document. If document owner allows $U_{m-1}, U_m$ to retrieve document $D_n$, encrypting the secret key $sk_n$ using $e_0$, and attaching it to the document $D_n$. The following expression shows as follows:

$$sk_n' = sk_n^{e_0} \mod \prod_{U_m \in AU} N_m$$

The unauthorized users can’t use their secret key $d_m$ to retrieve the secret key $sk_n$ and decrypt the document, even if they can retrieve the encrypted content.

$$sk_n = (sk_n' \mod N_m)^{d_m} \mod N_m$$

3.4. Adding User

If the document owner wants to add a user to access $D_n$, it is easy to do that doesn’t bother other original authorized users. Assume the document owner wants to add a user $U_{m+1}$ to retrieve $D_n$. The following steps will execute:

1. Document owner randomly chooses two $m$ distinct large primes $(p_{m+1}, q_{m+1})$ for $U_{m+1}$.
2. Update the value of $L_0$ to $L_0'$

$$L_0' = LCM(\phi(N_1), \phi(N_2), \ldots, \phi(N_m), \phi(N_{m+1}))$$

3. Updating $d_0$ to $d_0'$ that satisfies the two requirements shows as follows:

$$e_0 \times d_0' = 1 (\mod L_0')$$

$$d_0' > \max\{\phi(N_1), \phi(N_2), \ldots, \phi(N_m), \phi(N_{m+1})\}$$

4. Authorize $U_{m+1}$ to retrieve the secret key $sk_n$ of $D_n$.

$$sk_n' = sk_n^{e_0} \mod \prod_{U_m \in AU} N_m$$

3.5. Revoking User

If the document owner wants to revoke some users to access $D_n$, there assumes that document owner wants to revoke $U_{m+1}$ from $D_n$. The $sk_n$ need to update to $sk_n'$. The following expression shows as follows:

$$sk_n' = sk_n^{e_0'} \mod \prod_{U_m \in AU \cup \{U_{m+1}\}} N_m$$

4. EXAMPLE

4.1. Key Assignment

1. Assume the document owner usually share document with $U_1, U_2, U_3, U_4$. At first, giving every user a pair big distinct prime $p$ and $q$. Using the $p$ and $q$ to compute the value of $N_n$ and $\phi(N_n)$. The following table presents the example:

<table>
<thead>
<tr>
<th>User</th>
<th>$(p,q)$</th>
<th>$N$</th>
<th>$\phi(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_1$</td>
<td>$(37,11)$</td>
<td>407</td>
<td>360</td>
</tr>
<tr>
<td>$U_2$</td>
<td>$(23,47)$</td>
<td>1081</td>
<td>1012</td>
</tr>
<tr>
<td>$U_3$</td>
<td>$(35,53)$</td>
<td>1007</td>
<td>936</td>
</tr>
<tr>
<td>$U_4$</td>
<td>$(41,29)$</td>
<td>1189</td>
<td>1120</td>
</tr>
</tbody>
</table>

2. Next, computing the least common multiple $L_0$ of each $\phi(N_m)$ which equals to

$$L_0 = LCM(\phi(N_1), \phi(N_2), \ldots, \phi(N_m))$$

$$= LCM(360,1012,936,1120)$$
Document owner chooses a large prime $e_0$ that must satisfy two requirements as follows:

$$e_0 < \min\{360,1012,936,1120\}$$

$e_0$ is relatively prime to 33153120

In above requirement, there is choosing $e_0 = 89$.

4. Choosing a $d_0$ using the extended Euclidean algorithm. It also must satisfy two requirements as follows:

$$89 \times d_0 = 1 \mod 33153120$$

$$d_0 > \max\{360,1012,936,1120\}$$

In above requirement, there is choosing $d_0 = 11175209$.

5. Using $d_0$ to generate secret key and send secure to every users.

$$d_1 = d_0 \mod \varphi(N_1)$$

$$= 11175209 \mod 360 = 89$$

$$d_2 = d_0 \mod \varphi(N_2)$$

$$= 11175209 \mod 1012 = 705$$

$$d_3 = d_0 \mod \varphi(N_3)$$

$$= 11175209 \mod 936 = 305$$

$$d_4 = d_0 \mod \varphi(N_4)$$

$$= 11175209 \mod 1120 = 969$$

4.2. Broadcast Secret Key

Assume document owner usually share document with four users $U = \{U_1, U_2, U_3, U_4\}$ who will get a search key there assume “234” by the following steps:

$$k_s' = (k_s)^{e_0} \mod N_1 \times N_2 \times N_3 \times N_4$$

$$= (234)^{89} \mod 407 \times 1081 \times 1007 \times 1189$$

$$= 97319753678$$

Then document owner broadcasts $k_s'$ to four users.

When $U_1$ receives $k_s'$:

$$U_1: k_s = (k_s')^{d_1} \mod N_1$$

$$= (97319753678 \mod 407)^{89} \mod 407$$

$$= (367)^{89} \mod 407 = 234$$

When $U_2$ receives $k_s'$:

$$U_2: k_s = (k_s')^{d_2} \mod N_2$$

$$= (97319753678 \mod 1081)^{705} \mod 1081$$

$$= (234)^{705} \mod 1081 = 234$$

When $U_3$ receives $k_s'$:

$$U_3: k_s = (k_s')^{d_3} \mod N_3$$

$$= (97319753678 \mod 1007)^{305} \mod 1007$$

$$= (928)^{305} \mod 1007 = 234$$

When $U_4$ receives $k_s'$:

$$U_4: k_s = (k_s')^{d_4} \mod N_4$$

$$= (97319753678 \mod 1189)^{969} \mod 1189$$

$$= (235)^{969} \mod 1189 = 234$$

Every authorized user can uses their secret key to get search key $k_s'$. 
4.3. Document Authorize

Document owner wants to share $D_1$ with $AU = \{U_1, U_2, U_4\}$. He doesn’t want to share with $U_2$. There will encrypt the secret key $sk_1$ of $D_1$.

$$sk_1' = (sk_1^e \mod N_1) \times N_3 \times N_4$$
$$= (123)^{89} \mod 407 \times 1007 \times 1189$$
$$= 163652033$$

Attaching $sk_1'$ to the content of $D_1$.

When $U_1$ receives $sk_1'$:

$$U_1: sk_1 = (sk_1' \mod N_1)^{d_1} \mod N_1$$
$$= (163652033 \mod 407)^{89} \mod 407$$
$$= (182)^{89} \mod 407 = 123$$

When $U_3$ receives $sk_1'$:

$$U_3: sk_1 = (sk_1' \mod N_3)^{d_3} \mod N_3$$
$$= (163652033 \mod 1007)^{305} \mod 1007$$
$$= (435)^{305} \mod 1007 = 123$$

When $U_4$ receives $sk_1'$:

$$U_4: sk_1 = (sk_1' \mod N_4)^{d_4} \mod N_4$$
$$= (163652033 \mod 1189)^{969} \mod 1189$$
$$= (206)^{969} \mod 1189 = 123$$

$U_1, U_3, U_4$ successfully get $sk_1$ to decrypt document.

When $U_2$ receives $sk_1'$:

$$U_2: sk_1 = (sk_1' \mod N_2)^{d_2} \mod N_2$$
$$= (163652033 \mod 1081)^{705} \mod 1081$$
$$= (182)^{705} \mod 1081 = 708$$

Unfortunately, $U_2$ can’t recover $sk_1$. He can’t decrypt $D_1$ using 708.

4.4. Adding User

Assume that $U_5$ joining to the search group, the steps of the adding user process are as follows.

1. Document owner randomly chooses two $m$ distinct large primes $(p_5, q_5)$ for $U_5$.

Computing $N_5$ and $\varphi(N_5)$ shows as follow:

<table>
<thead>
<tr>
<th>User</th>
<th>$(p,q)$</th>
<th>$N$</th>
<th>$\varphi(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_5$</td>
<td>(71,43)</td>
<td>3053</td>
<td>2940</td>
</tr>
</tbody>
</table>

2. Update the value of $L_0$ to $L_0'$

$$L_0' = LCM(\varphi(N_1), \varphi(N_2), \varphi(N_3), \varphi(N_4), \varphi(N_5))$$
$$= LCM(360,1012,936,1120,2940) = 232071840$$

3. Update $d_0$ to $d_0'$ that satisfies the two requirements shows as follows:

$$e_0 \times d_0' = 1 \mod 232071840$$
$$d_0' > \max\{360,1012,936,1120,2940\}$$

There chooses $d_0' = 4925449$

4. Using $d_0'$ to generate secret key $d_5$ and secure sent to $U_5$, the other original authorized user don’t be bothered.

$$d_5 = d_0 \mod \varphi(N_5)$$
$$= 4925449 \mod 2940 = 949$$
4.5. Revoking User

Assume that the document owner revokes $U_4$ to access $D_1$, so the value of $sk_1$ changes to 99.

$$sk_1' = (sk_1)^{99} \mod N_1 \times N_3 \times N_4$$
$$= (99)^{99} \mod 407 \times 1007 = 4147$$

When $U_1$ receives $sk_1'$:

$$sk_1' = (sk_1' \mod N_1)^{d_1} \mod N_1$$
$$= (4147 \mod 407)^{99} \mod 407$$
$$= 99$$

When $U_3$ receives $sk_1'$:

$$sk_1' = (sk_1' \mod N_3)^{d_3} \mod N_3$$
$$= (4147 \mod 1007)^{305} \mod 100$$
$$= 99$$

$U_4$ and $U_3$ can retrieve the new key $sk_1'$ of $D_1$.

Suppose the revoke user $U_4$ wants to get $sk_1$ using $d_4$

$$sk_1 = (sk_1' \mod N_4)^{d_4} \mod N_4$$
$$= (4147 \mod 1189)^{969} \mod 1189$$
$$= 1044$$

Obviously, $U_4$ can’t recover $sk_1$ to decrypt $D_1$ from now on.

Assume $k_2$ change from 234 to 345.

$$k_2' = (k_2)^{99} \mod N_1 \times N_2 \times N_3$$
$$= (345)^{99} \mod 407 \times 1081 \times 1007$$
$$= 56346964$$

When $U_4$ receives $k_2'$:

$$U_4: k_2 = (k_2' \mod N_4)^{d_4} \mod N_4$$
$$= (56346964 \mod 1189)^{969} \mod 1189$$
$$= (254)^{969} \mod 1189 = 5$$

$U_4$ doesn’t have the ability to search documents.

5. SECURITY ANALYSIS

5.1. Document Confidentiality

If $U_m$ isn’t be authorized to access $D_m$, he can’t receive document. As above example in document authorize section. $U_2$ can’t obtain the secret key of $D_1$ $123(708 \neq 123)$.

$$U_2: sk_1 = (C \mod N_2)^{d_2} \mod N_2$$
$$= (163652033 \mod 1081)^{705} \mod 1081$$
$$= (182)^{705} \mod 1081$$
$$= 708$$

5.2. Revoking User

Continuing the previous example. If the document owner revokes $U_4$ to access $D_1. U_4$ can’t access it, and other authorized users don’t need to re-key.

$$sk_1' = (C \mod N_4)^{d_4} \mod N_4$$
$$= (4147 \mod 1189)^{969} \mod 1189$$
$$= 1044$$

Revoked users also lose their search ability.

When $U_4$ receives $k_2'$:

$$U_4: k_2 = (k_2' \mod N_4)^{d_4} \mod N_4$$
$$= (56346964 \mod 1189)^{969} \mod 1189$$
6. CONCLUSION

Cloud server only acts as a storage space and knows nothing expect for search pattern in our scheme. The multi-user searchable encryption scheme is easy for the document owner adds or revokes a user from user group that don’t bother other authorized users. Document can’t recover expect for the authorized users.

Funding: This study received no specific financial support.

Competing Interests: The authors declare that they have no competing interests.

Contributors/Acknowledgement: All authors contributed equally to the conception and design of the study.

REFERENCES


