LETTER

# Off-Line Keyword Guessing Attacks on Searchable Encryption with Keyword-Recoverability

**Eun-Jun YOON**[†a)] *and* **Kee-Young YOO**[†b)], ***Members***

**SUMMARY** In 2009, Jeong et al. proposed a new searchable encryption scheme with keyword-recoverability which is secure even if the adversaries have any useful partial information about the keyword. They also proposed an extension scheme for multi-keywords. However, this paper demonstrates that Jeong et al.'s schemes are vulnerable to off-line keyword guessing attacks, where an adversary (insider/outsider) can retrieve information of certain keyword from any captured query message of the scheme.

***key words:*** *keyword search, keyword-recoverability, cryptanalysis, keyword guessing attacks*

## 1. Introduction

The notion of searchable encryption was first suggested by Boneh et al. in [1]. With a searchable encryption scheme, a sender makes a ciphertext by encrypting a keyword with the public key of a receiver. The receiver can make a trapdoor for a keyword with a private key. Then any party can test whether or not the ciphertext and the trapdoor were made with the same keyword without knowing the keyword itself.

Bellare et al. [2] first proposed an SEKR (searchable encryption scheme with keyword-recoverability) in 2007. The SEKR scheme provides keyword-recoverability as well as keyword-testability. Keyword-testability means that a receiver of a ciphertext can test whether the ciphertext contains a specific keyword. Keyword-recoverability means that a receiver can extract the keyword from a ciphertext. Bellare et al.'s SEKR scheme provides only these two properties compared with the previous searchable encryption schemes.

In 2009, Jeong et al. [3] pointed out that Bellare et al.'s SEKR scheme does not provide IND-CKA (indistinguishability against chosen keyword attacks) since their SEKR scheme is constructed to be an "efficiently-searchable" encryption scheme. Furthermore, Jeong et al. proposed a new SEKR scheme which is secure even if the adversaries have any useful partial information about the keyword. They also proposed the mSEKR scheme for multi-keywords.

However, this paper demonstrates that Jeong et al.'s SEKR schemes [3] are not secure to off-line keyword guessing attacks [4], which an adversary (insider/outsider) can retrieve information of certain keyword from any captured query message of the scheme.

Manuscript received February 16, 2010.

[†]The authors are with the School of Electrical Engineering and Computer Science, Kyungpook National University, 1370 Sankyuk-Dong, Buk-Gu, Daegu 702–701, South Korea.

a) E-mail: ejyoon@knu.ac.kr

b) E-mail: yook@knu.ac.kr (corresponding author)

DOI: 10.1587/transinf.E93.D.1995

## 2. Review of Jeong et al.'s Schemes

The following algorithms are used in the schemes [3].

- Bilinear Map. Let $\mathbb{G}_1$ be a group of prime order $q$. $e$ is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties: (1) For all $u, v \in \mathbb{G}_1$ and $a, b \in Z$, $e(u^a, v^b) = e(u, v)^{ab}$. (2) If $g$ is a generator of $\mathbb{G}_1$, $e(g, g)$ is a generator of $\mathbb{G}_2$.
- Computational Diffie-Hellman (CDH) Assumption. Given $g, g^{u_1}, g^{u_2} \in \mathbb{G}_1$ as input, where $u_1, u_2 \leftarrow [1, q]$, compute $g^{u_1 u_2} \in \mathbb{G}_1$.
- Message Authentication Code (MAC). MAC consists of $\mathsf{M} = (\mathsf{Mac}, \mathsf{Vfy})$. Given a random key $k$, Mac computes a tag $\tau$ for a message $m$; $\tau \leftarrow \mathsf{Mac}_k(m)$. Vfy verifies the message-tag pair using the key $k$, and returns 1 if the tag is valid or 0 otherwise; $m$, $\mathsf{Vfy}_k(m, \mathsf{Mac}_k(m)) \stackrel{?}{=} 1$.
- Random Oracle Model. Let $H$ be a hash function such that $H : \{0,1\}^* \rightarrow \{0,1\}^\theta$, where $\theta$ is the length of the results of the hash function.

### 2.1 SEKR Scheme

Let the keyword $KW \in \{0,1\}^l$. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^{log_2 q}$, $H_3 : \mathbb{G}_1 \rightarrow \{0,1\}^l$ and $H_4 : \mathbb{G}_1 \rightarrow \{0,1\}^{log_2 q}$ be hash functions.

- $\mathsf{SEKR.key}(1^\theta)$. The algorithm picks a random $\alpha \in \mathbb{Z}_q^*$ and a generator $g$ of $\mathbb{G}_1$. It outputs a pair of public key $pk = [g, h = g^\alpha]$ and private key $sk = \alpha$.
- $\mathsf{SEKR.enc}(pk, KW)$. The algorithm first computes $a = e(H_1(KW), h^r)$ and $k = H_4(h^r)$ for a random $r \in \mathbb{Z}_q^*$. Then it outputs
$$A = g^r, B = H_2(a), C = H_3(h^r) \oplus KW$$
$$D = \mathsf{Mac}_k(A \| B \| C)$$
- $\mathsf{SEKR.td}(sk, KW)$. The algorithm outputs $t_{KW} = H_1(KW)^\alpha \in \mathbb{G}_1$.
- $\mathsf{SEKR.test}(pk, c, t_{KW})$. Let $c = [A, B, C, D]$. The algorithm tests if
$$H_2(e(t_{KW}, A)) \stackrel{?}{=} B$$
If so, the algorithm outputs 1; if not, the algorithm outputs 0.

- $\mathsf{SEKR.dec}(sk, c)$. Let $c = [A, B, C, D]$. The algorithm calculates $k = H_4(A^\alpha)$. Then the algorithm tests if

$$\mathsf{Vfy}_k(A\|B\|C, D) \stackrel{?}{=} 1$$

If so, the algorithm outputs $KW \leftarrow C \oplus H_3(A^\alpha)$. Otherwise, it outputs $\perp$.

### 2.2 mSEKR Scheme for Multi-Key Words

- $\mathsf{mSEKR.key}(1^\theta)$. The algorithm picks a random $\alpha \in \mathbb{Z}_q^*$ and a generator $g$ of $\mathbb{G}_1$. It outputs a pair of public key $pk = [g, h = g^\alpha]$ and private key $sk = \alpha$.
- $\mathsf{mSEKR.enc}(pk, \mathbf{KW})$, where $\mathbf{KW} = (KW_1, \ldots, KW_n)$. The algorithm first computes $a_i = e(H_1(KW_i), h^r)$ and $k = H_4(h^r)$ for a random $r \in \mathbb{Z}_q^*$, where $1 \le i \le n$. Then it outputs

$$A = g^r, B_i = H_2(a_i), C_i = H_3(h^r) \oplus KW_i$$
$$D = \mathsf{Mac}_k(A\|B_1\|\ldots\|B_n\|C_1\|\ldots\|C_n)$$

for $1 \le i \le n$.
- $\mathsf{mSEKR.td}(sk, \mathbf{KW})$. The algorithm outputs $t_{KW} = H_1(KW)^\alpha \in \mathbb{G}_1$.
- $\mathsf{mSEKR.test}(pk, c, t_{KW})$. Let $c = [A, B_1, \ldots, B_n, C_1, \ldots, C_n, D]$. The algorithm tests if

$$H_2(e(t_{KW}, A)) \stackrel{?}{=} B_i$$

for some $i$. If so, the algorithm outputs 1; if not, the algorithm outputs 0.
- $\mathsf{mSEKR.dec}(sk, c)$. Let $c = [A, B_1, \ldots, B_n, C_1, \ldots, C_n, D]$. The algorithm calculates $k = H_4(A^\alpha)$. Then the algorithm tests if

$$\mathsf{Vfy}_k(A\|B_1\|\ldots\|B_n\|C_1\|\ldots\|C_n, D) \stackrel{?}{=} 1$$

If so, the algorithm outputs $KW_i \leftarrow C_i \oplus H_3(A^\alpha)$ for $1 \le i \le n$. Otherwise, it outputs $\perp$.

## 3. Off-Line Keyword Guessing Attacks

In general, keywords are chosen from much smaller space than passwords and users usually use well-known keywords (low entropy) for search of document [4]. For example, in an e-mail search system which is a major application area of keyword search scheme based on public key encryption, users are interested to search for their e-mails sent by "Supervisor" or "Lover" in the From field or they may concern well-known keywords such as "Urgent", "Exam", and "Hello" in the Title fields. Usually, when users fill in a title of e-mail, they use a simple and representative sentence composed of very short keywords to make receivers easily grasp the content of e-mail. Sufficiently, this fact can give rise to keyword guessing attacks where an malicious adversary is able to guess some candidate keywords, and verify his/her guess is correct or not in an off-line manner. By performing this off-line keyword guessing attack, malicious outsider/insider adversary can get relevant information of encrypted e-mail, and intrude on a users' e-mail privacy. The off-line keyword guessing attack on the Jeong et al.'s SEKR scheme [3] can be performed by an adversary $Adv$ as follows.

Let $\mathbb{D}$ be a dictionary of keywords whose size is bounded by some polynomial. Let $pk = [g, h = g^\alpha]$ be a public key for a party. Assume that an adversary $Adv$ is given $t_{KW} = H_1(KW)^\alpha$ such that $\mathsf{SEKR.test}(pk, c, t_{KW}) = 1$, and $t_{KW}$ was made with keywords in $\mathbb{D}$. Then $Adv$ can determine which keyword was used in $t_{KW}$ as follows:

1. $Adv$ guesses an appropriate keyword $KW^*$ in $\mathbb{D}$, and computes $H_1(KW^*)$.
2. $Adv$ tests if
$$e(H_1(KW^*), h) \stackrel{?}{=} e(\hat{t}_{KW}, g) \tag{1}$$
If so, the guessed keyword is a valid keyword. Otherwise, go to Step 1.

We know that $t_{KW}$ is equal to $H_1(KW)^\alpha$ from the $\mathsf{SEKR.td}(sk, KW)$ algorithm of SEKR scheme. Therefore, if $KW$ is equal to $KW^*$, then Eq. (1) always holds since

$$\begin{aligned} e(H_1(KW^*), h) &= e(H_1(KW^*), g^\alpha) \\ &= e(H_1(KW^*)^\alpha, g) \\ &= e(t_{KW}, g) \end{aligned}$$

Similarly, this off-line keyword guessing attack works on the Jeong et al.'s mSEKR scheme to the multi-keywords settings [3]. As a result, Jeong et al.'s schemes are not secure to off-line keyword guessing attacks.

## Acknowledgements

This work is supported by the 2nd Brain Korea 21 Project in 2010.

### References

[1] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," Eurocrypt 2004, LNCS 3089, pp.31–45, 2004.
[2] M. Bellare, A. Boldyreva, and A. Öneill, "Deterministic and efficiently searchable encryption," CRYPTO 2007, LNCS 4622, pp.535–552, 2007.
[3] I.R. Jeong, J.O. Kwon, D. Hong, and D.H. Lee, "Searchable encryption with keyword-recoverability," IEICE Trans. Inf. & Syst., vol.E92-D, no.5, pp.1200–1203, May 2009.
[4] J.W. Byun, H.S. Rhee, H. Park, and D.H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," SDM 2006, LNCS 4165, pp.75–83, 2006.