Analysis of Forward Private Searchable Encryption and Its Application to Multi-Client Settings

Hyeongseob Kim, Changhee Hahn and Junbeom Hur
Dept. Computer Science and Engineering, Korea University
Seoul, Republic of Korea
Email: {hyeongseob, hahn850514, jbhur}@korea.ac.kr

***Abstract*—Searchable encryption (SE) supports privacy-preserving searches over encrypted data. Recent studies on SE have focused on improving efficiency of the schemes. However, it was shown that most of the previous SE schemes could reveal the client's queries even if they are encrypted, thereby leading to privacy violation. In order to solve the problem, several forward private SE schemes have been proposed in a single client environment. However, the previous forward private SE schemes have never been analyzed in multi-client settings. In this paper, we briefly review the previous forward private SE schemes. Then, we conduct a comparative analysis of them in terms of performance and forward privacy. Our analysis demonstrates the previous forward secure SE schemes highly depend on the file-counter. Lastly, we show that they are not scalable in multi-client settings due to the performance and security issue from the file-counter.**

***Index Terms*—searchable encryption, forward privacy;**

I. Introduction

Modern data outsourcing to cloud service providers (CSPs), despite its widespread usage, raises security concerns. Since data may contain sensitive information, they should be encrypted prior to outsourcing. Unfortunately, encryption typically hinders some important functionalities such as data searches.

Searchable encryption (SE) has been known as a promising solution to it because SE supports privacy-preserving searches over encrypted data [1]. Unfortunately, however, recent studies showed that SE cannot guarantee the data confidentiality [2]–[4]. Specifically, Zhang et al. proposed how to recover the queried data by injecting a few files to the cloud server [3]. In order to solve the problem, recent works proposed countermeasure schemes, called forward private SE, which protect queries against injected files [5]–[12]. These SE schemes depend on the file-counter, which allows the honest client to perform correct searches while prohibiting the file-injection adversaries.

In this paper, we briefly review and conduct a comparative analysis of the previous forward private SE schemes. We then show that they are not scalable in multi-client settings due to the performance and security issue incurred by the file-counter.

II. Background

In this section, we will explain the background of SE and forward privacy. Then, we will describe trapdoor permutation and pseudo-random function, the two main cryptographic techniques used for constructing forward private SE.

A. Searchable Encryption

Since Song et al.'s seminal work on SE based on symmetric cryptography (searchable symmetric encryption, or SSE for brevity) [1], SSE has been an interesting research topic over the past decades [13], [14]. However, due to the symmetric key based design, SSE is applicable only to specific scenarios where a single client is allowed to search [15].

In contrast, Boneh et al. [16] proposed SE based on asymmetric cryptographic primitive, known as public key encryption with keyword search (PEKS). Later, Baek et al. [17] enhanced PEKS by removing the need for secure channels. Crenscenzo et al. [18] presented the PEKS without relying on computationally intensive cryptographic primitives. In 2013, Xu et al. [19] proposed the PEKS scheme with fuzzy keyword search to prevent keyword guessing attacks which can infer the search queries. Tang et al. [20] also proposed the public key encryption with registered keyword search (PERKS) which is resilient to the offline keyword guessing attack.

B. Forward Privacy

Since 2009, many SE schemes were proposed to support updates on the encrypted database. Such schemes, called dynamic SE, support dynamic data updates but raises privacy concerns. Specifically, updating data f might reveal whether f contains a keyword that was searched before by the client. For example, a malicious server can reuse the previous search queries on the newly injected data. Then, the server simply observes whether the search queries match newly injected data. This information leakage may seem negligible but Zhang et al. [3] showed that such leakage can reveal all of the client's previous queries. To this end, forward private SE schemes were proposed [5]–[12], and they are typically based on either trapdoor permutations (TDP) or pseudo-random functions (PRF) to enhance security.

C. Trapdoor Permutation

A trapdoor permutation π is a permutation over a set $\mathcal{D}$ such that, using a public key PK, π can be easily evaluated, but the inverse π^{-1} can be efficiently computed only with the secret SK. More formally, π with the key generation function Gen_π is a trapdoor permutation if for all PPT adversaries $\mathrm{Adv}, \Pr[y \xleftarrow{\$} \mathcal{M}; (\mathrm{SK},\mathrm{PK}) \leftarrow Gen_\pi(1^\lambda); x \leftarrow \mathrm{Adv}(1^\lambda, \mathrm{PK}, y) : \pi_{\mathrm{PK}}(x) = y]$ (π is one-way) while for every

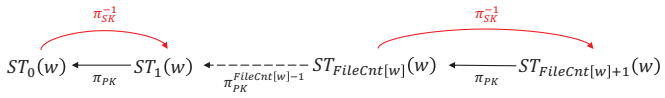

Fig. 1. Trapdoor permutation π

Index	*File Counter*
w_1	c_1
w_2	c_2
...	...
$w_{(\|W\|-1)}$	$c_{(\|W\|-1)}$
$w_{(\|W\|)}$	$c_{(\|W\|)}$

Update
$w_1, w_{(\|W\|)}$

Index	*File Counter*
w_1	c_1+1
w_2	c_2
...	...
$w_{(\|W\|-1)}$	$c_{(\|W\|-1)}$
$w_{(\|W\|)}$	$c_{(\|W\|)}+1$

Fig. 2. State of a file-counter table before and after data update

$x \in \mathcal{D}$. $\pi_{\text{PK}}(\pi_{\text{SK}}^{-1}(x)) = x$ and $\pi_{\text{SK}}^{-1}(\pi_{\text{PK}}(x)) = x$ and $\pi_{\text{PK}}(\cdot)$ and $\pi_{\text{SK}}^{-1}(\cdot)$ can be computed in polynomial time.

In this paper, we also use the notation $\pi_{\text{PK}}^{(c)}(x)$ (resp. $\pi_{\text{SK}}^{(-c)}(x)$) for the iterated application of π_{PK} (resp. π_{SK}^{-1}) c times, as shown in Figure 1.

D. Pseudo-Random Function

Let $Gen_G(1^\lambda) \in \{0,1\}$ be a key generation function, and $G : \{0,1\}^\lambda \times \{0,1\}^l \to \{0,1\}^{l'}$ be a pseudo-random function (PRF) family. $G_K(x)$ denotes $G(K,x)$. G is a secure PRF family if for all PPT adversaries Adv, $|\Pr[K \leftarrow Gen_G(1^\lambda); \text{Adv}^{G_K(\cdot)}(1^\lambda) = 1] - \Pr[\text{Adv}^{R(\cdot)}(1^\lambda) = 1]| \leq \nu(\lambda)$, where $R : \{0,1\}^l \to \{0,1\}^{l'}$ is a truly random function.

E. File Counter

A file-counter table is used for managing the file-counters of keywords, as shown in Figure 2. The file-counter table FileCnt comprises a set of pairs $\langle w_i c_i \rangle$, where $w_i \in W$ is an index of FileCnt and c_i denotes a file-counter corresponding to w_i. when a client updates the data containing w_i, FileCnt will be updated by increasing c_i by 1.

Most forward private SE schemes use the file-counter. It is important to note that the file-counter should be managed in secret. Otherwise, SE may incur incorrect searches or violation of privacy.

III. Forward Private Searchable Encryption

In this session, we will introduce and analyze the recent forward private SE schemes.

A. Forward Private SE with TDP

Bost et al. [5] first proposed a forward private SE scheme, called $\Sigma o\phi o\varsigma$, with trapdoor function. Later, Zuo et al. [8] presented the forward private scheme which supports backward privacy (backward privacy refers to revoking a server's searching ability on deleted data) from symmetric puncturable encryption. Zuo et al.'s scheme also used the trapdoor function for forward privacy like $\Sigma o\phi o\varsigma$. Subsequent works based on trapdoor functions were proposed which offer additional functions or enhanced performance [7], [9], [10].

In forward private SE with trapdoor function, file-counters are used to properly trace the address of the outsourced data. These schemes use the file-counter with the state value ST. When the file-counter is NULL, initial ST_0 is randomly picked. If the client wants to update the data f containing the keyword w, he checks the current state value ST_c of w. He computes $ST_{(c+1)}$ from $\pi_{\text{SK}}^{-1}(ST_c)$ using his private key SK. Then, he updates FileCnt by increasing c of w by 1. Using the new state value $ST_{(c+1)}$, he computes the address and encrypts the identifier id of data. Then he sends them to the server.

If the client wants to search the data with keyword w^*, then he checks the current ST^* of w^* and file count $\text{FileCnt}[w^*] = c^*$. Then, both are sent to the server which will find the encrypted data identifier using ST^*. Note that the server can compute $\pi_{\text{PK}}(ST^*)$ easily using PK. The server recovers subsequent STs thereafter. The properly encrypted file identifiers are founded by those values.

B. Forward Private SE with PRF

Bost et al. [6] also presented forward private SE with backward privacy in 2017. However, unlike $\Sigma o\phi o\varsigma$, Bost et al. used constrained PRFs to provide forward private SE. Since then, many forward private schemes have used PRF as well. In 2018, Etemad et al. [12] proposed the efficient forward SE which supports parallel computation. Ghareh et al. [11] also presented forward and backward SE with PRF as primitive.

Similar to SE schemes with trapdoor function, those schemes with the pseudo-random function used the file-counters to progressively update and search the data. The file-counters of keywords are initialized to 0. If the client wants to update the data f associated with a keyword w, then the client checks the file-counter $\text{FileCnt}[w] = c$. With c and w, he computes the address from $G_K(w, c+1\|0)$. Next, he encrypts the identifier id of the data as $val = id \oplus G_K(w, \text{FileCnt}[w]\|1)$. Due to the property of PRF, the results are computationally indistinguishable. Then he sends them to the server.

If the client wants to search the data with keyword w^*, then he checks the current $\text{FileCnt}[w] = c^*$. With w^* and c^*, he computes and sends the previous addresses to the server. Using these addresses, the server can find the encrypted identifiers.

C. Analysis of Forward Private SE schemes

The existing forward private SE schemes are designed only for single client settings. Our analysis shows that, in extensions to a multi-client setting, they have performance and privacy problems. These issues are specifically due to (1) the usage of file-counters and (2) symmetric key based designs.

Limitations of file-counter. In the multi-client settings, clients are divided into two groups: a data owner group who update their data in the server and a data user group who are authorized to search from the server. In this setting, because the data owners update the file-counters whenever they update data, they must manage the file-counters. However, the data users also need the file-counters in order to perform

searches. Thus, it incurs additional round trip time to retrieve the file-counters from the data owners. Unfortunately, such an additional round trip cost can cause significant performance degradation. For example, search time of $\Sigma o \phi o \varsigma$ can be within 5ms, while the additional round trip cost can be up to 270ms depending on the environment [10], [21].

In addition, managing file-counters in the presence of multiple owners is problematic. First, if the data owners manage their own file-counters separately, then the data users should retrieve every file-counter from each data owner. This means that the client's storage overhead is affected by the number of data owners. It was reported in [6] that the storage overhead per client is up to 720MB. Suppose there are n data owners, then each client's storage overhead will grow to n times, which is unacceptable in practice. One may think that it is solvable by letting the data owners share the single file-counter table. However, it causes information leakage such as which data are updated by which data owner. Specifically, each time a data owner updates data, he also updates the file-counter which is related with the data. Thus, another data owner can guess what the data is by observing the shared file-counter table.

Key distribution. The existing forward private SE schemes are designed on the basis of SSE. Thus, these schemes inherit the drawbacks of SSE. Specifically, SSE has a key distribution issue when the number of clients increase. That is, the forward private SE schemes also share the same issue. Moreover, when the client group is dynamic, i.e., the membership is frequently changed, the frequent key distribution is also required.

IV. Conclusion

As the effective adaptive file-injection attack was presented in 2016, forward private searchable encryption has been considered as a countermeasure to it. Most forward private SE schemes proposed so far are based on symmetric cryptographic primitive and the file-counter. Using only the symmetric cryptographic primitive causes the key distribution problem and scalability issues when SE is extended for supporting multiple clients. Moreover, sharing and synchronizing the file-counter between clients cause privacy and performance issues. Therefore, constructing a forward private SE for multi-client settings without degrading security and performance is an open problem.

V. Acknowledgment

This work has been supported by the Future Combat System Network Technology Research Center program of Defense Acquisition Program Administration and Agency for Defense Development. (UD160070BD)

References

[1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*. IEEE, 2000, pp. 44–55.

[2] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 668–679.

[3] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: the power of file-injection attacks on searchable encryption," in *Proceedings of the 25th USENIX Conference on Security Symposium*. USENIX Association, 2016, pp. 707–720.

[4] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in *in Network and Distributed System Security Symposium*. Citeseer, 2012.

[5] R. Bost, "$\Sigma o \varphi o \varsigma$: Forward secure searchable encryption," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1143–1154.

[6] R. Bost, B. Minaud, and O. Ohrimenko, "Forward and backward private searchable encryption from constrained cryptographic primitives," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1465–1482.

[7] S. F. Sun, X. Yuan, J. K. Liu, R. Steinfeld, A. Sakzad, V. Vo, and S. Nepal, "Practical backward-secure searchable encryption from symmetric puncturable encryption," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 763–780.

[8] C. Zuo, S. F. Sun, J. K. Liu, J. Shao, and J. Pieprzyk, "Dynamic searchable symmetric encryption schemes supporting range queries with forward (and backward) security," in *European Symposium on Research in Computer Security*. Springer, 2018, pp. 228–246.

[9] X. Song, C. Dong, D. Yuan, Q. Xu, and M. Zhao, "Forward private searchable symmetric encryption with optimized i/o efficiency," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[10] L. Wu, B. Chen, K. K. R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based internet of things," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 152–161, 2018.

[11] J. Ghareh Chamani, D. Papadopoulos, C. Papamanthou, and R. Jalili, "New constructions for forward and backward private symmetric searchable encryption," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 1038–1055.

[12] M. Etemad, A. Küpçü, C. Papamanthou, and D. Evans, "Efficient dynamic searchable encryption with forward privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 1, pp. 5–20, 2018.

[13] C. Bösch, P. H. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 18–1, 2014.

[14] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage." in *in Network and Distributed System Security Symposium*, vol. 71, 2014, pp. 72–75.

[15] G. S. Poh, J. J. Chin, W. C. Yau, K. K. R. Choo, and M. S. Mohamad, "Searchable symmetric encryption: designs and challenges," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, p. 40, 2017.

[16] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 506–522.

[17] J. Baek, R. Safavi Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *International conference on Computational Science and Its Applications*. Springer, 2008, pp. 1249–1259.

[18] G. Di Crescenzo and V. Saraswat, "Public key encryption with searchable keywords based on jacobi symbols," in *International Conference on Cryptology in India*. Springer, 2007, pp. 282–296.

[19] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Transactions on computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[20] Q. Tang and L. Chen, "Public-key encryption with registered keyword search," in *European Public Key Infrastructure Workshop*. Springer, 2009, pp. 163–178.

[21] I. F. Akyildiz, Ö. B. Akan, and G. Morabito, "A rate control scheme for adaptive real-time applications in ip networks with lossy links and long round trip times," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 3, pp. 554–567, 2005.