Future Generation Computer Systems 55 (2016) 353–361

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Secure hybrid-indexed search for high efficiency over keyword searchable ciphertexts

Wei Wang [a,b,*], Peng Xu [c], Hui Li [a,b], Laurence Tianruo Yang [d]

[a] Shenzhen Engineering Lab of Converged Networks Technology, Shenzhen Graduate School, Peking University, Shenzhen, China
[b] Shenzhen Key Lab of Cloud Computing Technology & Application, Shenzhen Graduate School, Peking University, Shenzhen, China
[c] Services Computing Technology and System Lab, Cluster and Grid Computing Lab, Huazhong University of Science and Technology, Wuhan, China
[d] Embedded and Pervasive Computing Lab, Huazhong University of Science and Technology, Wuhan, China

HIGHLIGHTS

- We propose our high efficient Secure Hybrid Indexed Search (SHIS) scheme.
- We model semantic secure SHIS with universal transformations from PEKS and DE.
- We show the complexity of SHIS is much lower than PEKS and convergent.
- We universally extend SHIS by PKE schemes towards multiple-receiver applications.

ARTICLE INFO

Article history:
Received 18 November 2013
Received in revised form
25 May 2014
Accepted 29 July 2014
Available online 14 August 2014

Keywords:
Public-key encryption with keyword search
Secure search complexity
Dynamic index
Static index

ABSTRACT

With a significant advance in ciphertext searchability, public-key encryption with keyword search (PEKS) guarantees both security and convenience for outsourced keyword search over ciphertexts. In this paper, we establish static index (SI) and dynamic index (DI) for PEKS to make search efficient and secure in the state of the art. Suppose there are u senders to generate n searchable ciphertexts for w keywords. The search complexity of PEKS always is $O(n)$ for each query, even if the keyword has been searched for multiple times. It is obviously inefficient for massive searchable ciphertexts. Fortunately, SI and DI help PEKS lowering the burden respectively in two phases: if the queried keyword is the first time to be searched, apply SI to reduce the complexity from $O(n)$ to $O(u \cdot w)$; otherwise, apply DI to reduce the complexity from $O(n)$ to $O(w)$. Because DI is invalid for the first time search on any keyword, SI and DI are simultaneously applied with PEKS to complete our work as the secure hybrid indexed search (SHIS) scheme. Since $u \ll w \ll n$ in practice, our SHIS scheme is significantly more efficient than PEKS as demonstrated by our analysis. In the end, we show the extension of SHIS to multi-receiver applications, which is absent for pure PEKS.

1. Introduction

Searchability of online privacy in serving storage is the kernel of applicability beyond security and privacy in service platforms like "cloud". Nevertheless, it also made these files hard to be searched even secure keyword searchability is essential to ciphertexts to run business services. Public-key encryption with keyword search (PEKS) explored in previous work [1] elegantly achieves the searchability.

Let us reveal the functions of PEKS by bothering our two old friends: Alice the sender and Bob the receiver. To create a keyword searchable ciphertext for an encrypted file, Alice (the owner of the file) extracts a keyword of the file and generates a PEKS ciphertext of this keyword, then outsources the PEKS ciphertext and the encrypted file to an online storage. To retrieve the encrypted files of the wanted keyword, Bob sends a keyword search trapdoor to the storage. The storage responds with the encrypted files containing the matching keyword by applying PEKS. During the whole processes, no one except Alice and Bob knows the content of the required keyword such that the privacy of the keyword is guaranteed. Moreover, PEKS suits multiple senders acting as Alice,

* Correspondence to: School of Computer Science and Technology, Huazhong University of Science and Technology, Luoyu Street, #1037, Wuhan, 430074, China. Tel.: +86 18627792102.
E-mail address: viviawang@126.com (W. Wang).

http://dx.doi.org/10.1016/j.future.2014.07.008
0167-739X/

since a search getting triggered and processed rely on the receiver but not the senders.

Lack of high efficiency is the drawback of all existed PEKS schemes. PEKS is the first keyword searchable encryption scheme based on the probabilistic public-key encryption. In the aspect of provable security, it particularly achieves the stronger IND-CKA (indistinguishability of ciphertexts under chosen keyword attack) security than almost all previous schemes, who are based on the deterministic encryption such as [2–4]. In applications, PEKS is more convenient than the searchable symmetric-key encryption schemes like [2], since there is no more needs of it to share a symmetric key between each pair of sender and receiver. However, PEKS has high search complexity that is linear with the number of searchable ciphertexts. The risk may be huge when such searches are repeated over massive searchable ciphertexts. For instance, an online healthcare database must suffer large mounts of authorized searches over patients' files with the keyword "bird flu" in a certain period. Let n denote the number of keyword searchable files in the database and m for authorized queries arrive. For each query, PEKS has to traverse all files to match the keyword searchable trapdoor of "bird flu" due to the lack of efficient indexes. So the number of matching times for each search is $O(n)$ with PEKS. With m authorized queries, the search complexity is $O(m \cdot n)$ by counting the total matching times. As being world-wide applied, it is considerably huge burden with complexity $O(m \cdot n)$ to large-scaled service platforms. With the consideration of efficiency, we offload such overwhelming burden by building secure indexes to files.

1.1. Our ideas and challenge

We notice that using indexes will fasten the searching speed of PEKS, and two kinds of indexes will be used in this paper. We respectively call these two indexes static index (SI) and dynamic index (DI). Let us partition PEKS searches into the first-time search and the repeated searches for each keyword. Supposing there are u senders to generate n searchable ciphertexts for w keywords, the original PEKS search complexity is $O(n)$ for one search. However, the two kinds of indexes we apply enhance the efficiency in different ways respectively with two different searching situations, which we call the first-time search and the repeated search.

SI allows a sender locally generating a static index for each PEKS ciphertext. It allows that PEKS ciphertexts generated exactly by the same sender and for the same keyword have the same static index. Hence, with the help of static indexes, the first-time search of a keyword needs the complexity $O(u \cdot w)$. It brings the lower complexity with SI than without it, since usually $u \ll w \ll n$ in a big data scenario.

DI allows the storage globally generating a dynamic index for the PEKS ciphertexts of the same keyword, if the keyword has been queried before. After a keyword's first-time search is triggered, the storage obtains all the matching PEKS ciphertexts with the received keyword search trapdoor. Then the received keyword search trapdoor is taken as a dynamic index of these matching PEKS ciphertexts. When the keyword is searched again, the dynamic index lowers the search complexity to $O(w)$. Moreover, the dynamic index can label any new generated PEKS ciphertext if it belongs to the index. So DI also is valid for the dynamic update of PEKS ciphertexts.

To achieve our goal, there is a key challenge we need to face: SI and DI cannot leak any keyword to anyone except for the related sender and receiver as it is hard to be avoided with index. Ciphertexts being indexable means some of them hold the same part of information. However indexes must be visible to all clients and servers that could impose the secrets of ciphertexts. So we need to figure out how to avoid secret leaking from indexes which is also the biggest obstacle before our goals.

1.2. Our contributions

We extend PEKS by SI and DI to propose our complete work as the secure hybrid indexed search (SHIS) scheme. We model SHIS with formal definitions, and construct a universal transformation from two cryptographic primitives PEKS and the deterministic encryption (DE) [4,5] to SHIS. In this universal transformation, DE is used to securely realize SI. According to the universal transformation, any instances of PEKS and DE can combine to a SHIS instance. We prove that if any PEKS and DE instances are consistent, their universally transformed SHIS instance also is consistent. So the universally transformed SHIS can correctly work, if its building blocks PEKS and DE correctly work. In the aspect of security, we prove that the universally transformed SHIS scheme is semantically secure if its building blocks PEKS and DE are semantically secure. Given that DE is semantically secure only if its plaintext space has the high prior-entropy [4], in the universal transformation, we take the combination of the keyword space and a random space as the plaintext space of DE to keep its semantic security.

In the aspect of efficiency, we show that the search complexity of SHIS is apparently more efficient than PEKS. Moreover, the search complexity of SHIS quickly converges to a considerably lower value when queries increase comparing to that of PEKS.

Since our original version of SHIS only suits the application of one receiver, to apply SHIS towards multiple receivers, we universally extend SHIS by public-key encryption (PKE) scheme. In the extension, keyword search trapdoors are authorized by a trusted third part (TTP) for receivers or users. The communication between TTP and any user is in a secure channel established by PKE.

1.3. Related works on PEKS

PEKS was first proposed by Boneh et al. in 2004 [1], and realized based on anonymous identity-based encryption [6–9]. Abdalla et al. [10] perfected the foundations of PEKS, and proposed a more secure transformation from anonymous IBE to PEKS and an extended PEKS. Baek et al. [11] freed the secure channel between the storage and users by employing public-key encryption. To achieve conjunctive keyword search, two schemes on public-key encryption with conjunctive keyword search (PECKS) [12,13] were respectively proposed. Furthermore, Bethencourt et al. succeeded in public-key encryption with conjunctive keyword range search [14] by anonymous hierarchical IBE (HIBE) [7] in 2006, and updated their work in 2007 [15]. Boneh et al. proposed a novel technique called hidden vector encryption (HVE) to achieve conjunctive, range and subset searches [16]. Camenisch et al. [17] employed the committed two-part computation protocol to achieve the oblivious keyword in the generation of keyword trapdoor. Refs. [18,19] proposed several efficient PECKS by sharing a secret between senders and receivers. And there is also a work proposed in [20] to build proxy re-encryption with PEKS. Most of related works about PEKS are focused on the versatile searchability, several care about the secure search anti guessing attacks, like [21] and our previous work [22]. About searching efficiency [3,23] gave ways to accelerate the retrieval speed by providing indexes with symmetric encryption. So far as we known, our paper is the first one to improve the search complexity of PEKS.

1.4. Organization

The organization of this paper is as follows. In Section 2, we introduce the definitions of PEKS and DE. In Section 3, we model SHIS and propose the universal transformation from PEKS and DE to SHIS. Then we prove the security of SHIS in Section 4. Section 5 analyzes the search complexity of SHIS with both theoretic and numerical results. Section 6 extends SHIS to the application of