Journal of Communications and Information Networks
Vol.1, No.4, Dec. 2016
DOI: 10.11959/j.issn.2096-1081.2016.043

Review paper

Secure searchable encryption: a survey

WANG Yunling[1], WANG Jianfeng[1,2], CHEN Xiaofeng[1]

1. State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xian 710071, China
2. Guangxi Cooperative Innovation Center of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China

Abstract: Cloud computing facilitates convenient and on-demand network access to a centralized pool of resources. Currently, many users prefer to outsource data to the cloud in order to mitigate the burden of local storage. However, storing sensitive data on remote servers poses privacy challenges and is currently a source of concern. SE (Searchable Encryption) is a positive way to protect users sensitive data, while preserving search ability on the server side. SE allows the server to search encrypted data without leaking information in plaintext data. The two main branches of SE are SSE (Searchable Symmetric Encryption) and PEKS (Public key Encryption with Keyword Search). SSE allows only private key holders to produce ciphertexts and to create trapdoors for search, whereas PEKS enables a number of users who know the public key to produce ciphertexts but allows only the private key holder to create trapdoors. This article surveys the two main techniques of SE: SSE and PEKS. Different SE schemes are categorized and compared in terms of functionality, efficiency, and security. Moreover, we point out some valuable directions for future work on SE schemes.

Key words: cloud storage, encrypted data, searchable encryption, searchable symmetric encryption, public key encryption with keyword search.

Citation: WANG Y L, WANG J F, CHEN X F. Secure searchable encryption: a survey[J]. Journal of communications and information networks, 2016, 1(4): 52-65.

1 Introduction

With the rapid development of cloud computing, cloud storage has enabled the provision of high data availability, easy access to data, and reduced infrastructure costs from outsourcing of data to remote servers. Many users prefer cloud storage services to relieve the burden of maintenance costs as well as the overhead of storing data locally. Moreover, users are able to access their data from anywhere and at any time instead of having to use dedicated machines.

Although cloud storage offers many advantages to users, there are still various security concerns. A remote server cannot be fully trusted because it may not only be curious about the users data but also abuse the data. When users outsource their data to a remote server, the physical access to the data is actually lost and the administration of the data is delegated to the server as well. Thus, it is necessary to guarantee the privacy of users sensitive data. The most common way of achieving privacy is to encrypt the data before outsourcing them. This approach provides end-to-end data privacy as soon as the data leave the users

Manuscript received Aug. 29, 2016; accepted Nov. 25, 2016
This work is supported by Guangxi Cooperative Innovation Center of Cloud Computing and Big Data (No.YD16506)

possession. While such a solution guarantees the privacy of sensitive data, it also brings difficulties for the server to perform any meaningful function, especially search functions, on the encrypted data.

Consider a search function on plaintexts. A user sends query keywords to the server in order to retrieve corresponding documents. After searching, the server will return the search results to the user. However, during the search process, both the knowledge of the contents stored on the server and the query keywords are exposed to the semi-trusted server. Fortunately, encryption is a positive way to protect the privacy of users data, but at the same time it disrupts search functionality. A trivial way to search is to download all the ciphertexts, decrypt them, and then search on the plaintexts. However, this is impractical. Consequently, a method that provides data confidentiality and preserves search functionality simultaneously is needed as this is an open problem.

SE has been proposed. It is not only an encryption scheme but also supports keyword search on encrypted data. In SE schemes, a user can outsource a collection of encrypted data to the server while maintaining the ability to search them. From the aspect of security, the privacy of documents and keywords is maintained. The two main branches of SE are SSE and PEKS. SSE is related to the private key primitive. It allows only the private key holder to produce ciphertexts and to create trapdoors for search. PEKS, on the other hand, is related to the public key primitive. It enables a number of users who know the public key to produce ciphertexts but only allows the private key holder to create trapdoors for search.

This article surveys the practical techniques of SE. The main contributions of this paper are (1) a review of the most meaningful SE approaches, mainly focusing on SSE and PEKS, and (2) analysis and classification of these approaches. The similarities and differences of these schemes are also examined and the outstanding issues for further studies discussed.

The remainder of this paper is organized as follows: the model and security requirements for SE are given in Section 2. A review of SSE techniques is given in Section 3. A review of PEKS techniques is given in Section 4. Finally, conclusions and valuable issues for further work are given in Section 5.

2 Model and security requirements

2.1 Model of searchable encryption

A searchable encryption scheme includes three parties: a trusted data owner O, a semi-trusted server S, and a collection of users who are authorized to search. The task for each party is as follows:

- Data owner: A data owner would like to outsource a collection of documents $\mathcal{D}=\{D_1, D_2,\cdots,D_n\}$ together with some keywords. The data owner needs to encrypt the documents and keywords in a particular manner, in order to easily search them afterward, then sends the ciphertexts to the server.
- Data user: If an authorized user wants to search the documents that contain a particular keyword, she/he has to submit the trapdoor of this query keyword to the server. After searching, the server returns the documents that contain this keyword to the user.
- Server: A server performs search tasks. When the server receives a trapdoor of a query keyword from a user, it searches over ciphertexts and then returns related documents to the user. We assume that the server is honest-but-curious. This means the server will follow the protocol correctly, but it may analyze the data received and attempt to obtain some additional information.

2.2 Security requirement of searchable encryption

In a searchable encryption scheme, the security of the documents and keywords stored on the server should be guaranteed. In addition, the security of query keywords should also be assured. Moreover, the following two security items should also be protected[1]:

- Search pattern: Search pattern is defined as any information that can be derived from knowledge of whether two search results are from the same keyword.
- Access pattern: Access pattern is defined as a sequence of search results $(\mathcal{D}(w_1),\cdots,\mathcal{D}(w_n))$, where $\mathcal{D}(w_i)$ is the search results of wi. In other words, $\mathcal{D}(w_i)$ is a collection of documents in $\mathcal{D}$ that contains the keyword w.

3 Searchable symmetric encryp-tion scheme

Consider the following scenario: A user, Alice, wants to store a set of documents on a server because of limited storage resources. As the server is semi-trusted, Alice has to encrypt the documents before outsourcing them. If Alice needs some documents containing a particular keyword, she will need to submit some information in terms of query keywords to the server. Then, the server will search the ciphertext to determine which document contains the query keyword. Fig.1 shows the model of SSE schemes.

3.1 Algorithms description

A general searchable symmetric encryption scheme includes four polynomial-time algorithms:

- Keygen(1^k): a key generation algorithm run by the data owner. It takes a security parameter k as input, and outputs a secret key K.

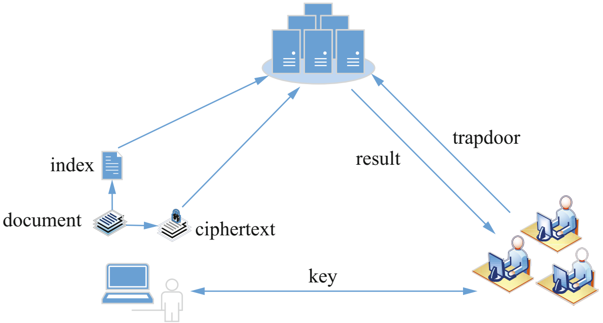

Figure 1 Model of SSE schemes

- BuildIndex(K, $\mathcal{D}$): a keyword index generation algorithm run by the data owner. It takes a secret key K and a set of documents $\mathcal{D}$ as inputs, and outputs a keyword index $\mathcal{I}$.
- Trapdoor(K, w): a keyword trapdoor generation algorithm run by the user. It takes a secret key K and a query keyword w as inputs, and outputs the trapdoor T_w for the keyword w.
- Search($\mathcal{I}$, T_w): a keyword search algorithm run by the server. It takes a keyword index $\mathcal{I}$ and a trapdoor T_w as inputs, and outputs a set of documents $\mathcal{D}(w)$ that contains query keyword w.

3.2 Security instructions

A searchable symmetric encryption scheme should satisfy various security requirements. The privacy of documents, search index, and query keywords should be protected, as well as the search pattern and access pattern. Song, et al.[2] argued that their scheme is provably secure because the server cannot learn any information about the plaintext by knowing the ciphertext. However, this kind of security is not strong enough in the context of SSE. IND1-CKA and the stronger IND2-CKA security model, which address the security of keyword indexes, have been proposed by Goh[3]. In both security models, an adversary $\mathcal{A}$ cannot learn the contents of a document from its index. In the IND1-CKA model, it is assumed that the indexes are built from documents with the same number of keywords. Conversely, in the IND2-CKA model, this assumption is not necessary. However these two models do not consider the security of trapdoors. Curtmola, et al.[1] introduced new adversarial models that consider the security of trapdoors. The first is a non-adaptive model, IND-CKA1, in which the adversary does not consider the trapdoors and search results of previous searches when he or she chooses challenge search queries. The other is an adaptive model, IND-CKA2, in which the adversary chooses their challenge search queries with the knowledge of trapdoors and search results previously obtained. In this article, we primarily focus on the IND-CKA1 and IND-CKA2 security models to compare various schemes.

3.3 SSE schemes

3.3.1 Single keyword search

1) SSE schemes with sequential scan. Song, et al.[2] proposed the first SSE scheme. In their solution, search is performed by sequentially scanning the entire ciphertext. The underlying idea of this scheme is that the ciphertext is obtained by XORing each of the keywords in the plaintext with a sequence of pseudorandom bits. Thus, it is allowed to directly search on the ciphertext.

The scheme comprises three steps: encryption, search, and decryption.

Firstly, encryption is performed by the user, Alice. Suppose Alice wants to encrypt a document containing a sequence of keywords $W_1,\cdots,W_l$. The encryption for each keyword W_i is as follows. First, Alice encrypts W_i by using function E with key k'' and obtains the ciphertext X_i which is n bits. That is, $X_i=E_{k''}(W_i)$. Then X_i is split into left part L_i and right part R_i, where L_i is the first $(n-m)$ bits and R_i is the latter m bits of X_i. Then Alice generates a sequence of values $S_1,\cdots,S_i$, where S_i is $(n-m)$ bits long. To encrypt n-bits X_i, Alice takes value S_i, sets $T_i=\langle S_i, F_{k_i}(S_i)\rangle$, and outputs the ciphertext $C_i = X_i \oplus T_i$, where $k_i = f_{k'}(L_i)$. Finally, Alice outputs all the ciphertext C_i to the server. Fig.2 shows the procedure utilized for encryption in this scheme.

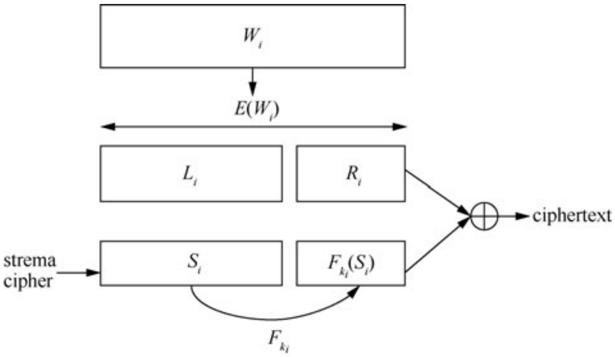

Figure 2 Encryption procedure

Secondly, search is performed by the server. When Alice wants to search documents containing the keyword W, she computes $X=E_{k''}(W)$ and $k = f_{k'}(L)$, and sends $\langle X, k\rangle$ to the server. Then, the server searches for X in the ciphertext by checking whether $C \oplus X$ is of the form $\langle s, F_k(s)\rangle$ for some s. If this C exists, the server sends the entire ciphertext for the document containing the query keyword to Alice.

Finally, Alice decrypts the ciphertext. For each C_i in the ciphertext, Alice generates S_i using pseudorandom generator, then she XORs S_i and the first $(n-m)$ bits of C_i to obtain the L_i. With the knowledge of L_i, Alice can compute k_i and eventually recover W_i.

In this scheme, the privacy of plaintext and the query keyword is maintained. However, it has a low search efficiency and the search time is linear in the length of the document collection, because the server needs to scan the entire ciphertext of a document when it determines whether a certain keyword is contained in the document. In addition, the plaintext is vulnerable to statistical attack according to the frequency of the query keyword occurring in the document.

2) SSE schemes with secure index. Document-based Index: To improve search efficiency, Goh[3] proposed a secure index construction using pseudorandom functions and Bloom filters[4]. With a Bloom filter, one can quickly determine whether an element belongs to a set. In this scheme, the pseudorandom function is applied twice to each keyword in the document. Then, the outputs are mapped to the Bloom filter. With the knowledge of the Bloom filter, it is much easier to determine whether a document contains a certain keyword.

Unlike the scheme proposed by Song, et al.[2], this scheme is based on a secure index. The advantage of this scheme is that the server only needs to search over the search indexes instead of scanning the entire ciphertext. As a result, the search efficiency is improved. However, the server also has to search each index and the search work for a query is linear in the number of documents, even if only one document contains the query keyword. We denote this kind of secure index as document-based index, in which the index corresponds with the documents.

Keyword-based Index: Another kind of secure index, called a keyword-based secure index, was proposed by Curtmola, et al.[1]. In this keyword-based secure index, one keyword corresponds to many document identifiers. In this scheme, the search time for a query keyword is linear in the number of documents containing the query keyword. Consequently, compared with the document-based index, keyword-based index is more efficient in searching a query. However, updating a keyword-based index when documents are added, deleted, or modified in the collection is difficult.

3) Dynamic SSE scheme. Van Liesdonk, et al.[5] proposed a dynamic SSE scheme that can deal with document updates. In their scheme, the search time is logarithmic in the keywords stored in the server. The basic scheme extends to two schemes, both of which support document updates. The first scheme is interactive, whereas the second is no interactive. Kamara, et al.[6] proposed an extension of Curtmola, et al.'s scheme to support updates, which is based on PRFs and XORs. However, updates would leak some information about the trapdoors. Subsequently, Kamara and Papamanthou[7] proposed a new dynamic tree based SSE scheme. In their proposed scheme, no information is leaked after the updating operation. Stefanov, et al.[8] proposed an efficient dynamic SSE scheme with a small information leakage. Various researchers have also focused on the dynamic property[9-12]. A comparison of several classic SSE schemes is given in Tab.1. In the table, n denotes the size of the documents set, r denotes the number of documents containing query keyword ω, m denotes the size of the keywords space and p denotes the number of cores.

Cash, et al.[13] and Zhang, et al.[14] recently focused on attacks on the SSE scheme, whereas Ishai, et al.[15] focused on improving its security. Further, Kamara and Moataz[16] focused on improving its functionality, while Asharov, et al.[17] focused on improving its performance.

3.3.2 Fuzzy keyword search

In the SSE scheme, a user submits the trapdoor of a query keyword to the server, and the server returns the documents containing the query keyword. However, if the query keyword does not match a preset keyword, such as “campus” and “compus”, the keyword search will fail. Fortunately, fuzzy keyword search can deal with this problem as it can tolerate minor typos and formatting inconsistencies. Li, et al.[18] constructed a fuzzy keywords collection using “edit distance” to quantify keywords similarity. Kuzu, et al.[19] used gram to construct fuzzy sets and LSH and Bloom filter to construct a ranking search scheme. Because a semi-honest server may only

Table 1 Comparison of several SSE schemes

scheme	search time	index size	security	dynamism
Song, et al.[2]	$O(n/p)$	N/A	CPA	static
Goh[3]	$O(n/p)$	$O(n)$	IND1-CKA	dynamic
Curtmola, et al.[1](SSE-1)	$O(r)$	$O(m+n)$	CKA1	static
Curtmola, et al.[1](SSE-2)	$O(r)$	$O(mn)$	CKA2	static
Van Liesdonk, et al.[5]	$O(r)$	$O(mn)$	CKA2	dynamic
Kamara, et al.[6]	$O(r)$	$O(m+n)$	CKA2	dynamic
Kamara, et al.[7]	$O((r/p)\log n)$	$O(mn)$	CKA2	dynamic

return a fraction of the results, Wang[20] proposed a verifiable fuzzy keyword search scheme that not only supports fuzzy keyword search, but also provides proof to verify whether the server returns all the search results. Several other proposed schemes also support fuzzy keyword search[21, 22].

3.3.3 Conjunctive keyword search

Conjunctive keyword search allows a user to obtain documents containing several keywords during a single query. It is more efficient and suitable for real applications than single keyword search. A trivial procedure is to perform single keyword search for each keyword separately and then deal with the results. However, it is inefficient and leaks some information to the server. Golle, et al.[23] proposed the first two conjunctive keyword search schemes. The communication cost of their first scheme is linear in the number of documents, but the major job can be done offline. Their second scheme requires only constant communication and there is no need to do anything offline. Ballard, et al.[24] presented two conjunctive keyword search constructions, one based on the nonstandard shamir secret sharing technique and the other on bilinear pairings. However, in their case, the trapdoor size is linear in the number of documents being searched. Cash, et al.[25] extended conjunctive query to Boolean query. Faber, et al.[26] extended Cash, et al.'s scheme[25] to support range, substring, wildcard, and phrase queries. In their system, the least frequent keyword is queried first, and the search results are then applied to other keywords. Theirs is the first sublinear SSE construction supporting Boolean query. Several other studies have also been conducted on this topic[27, 28].

3.3.4 Ranked and verifiable keyword search

Ranked keyword search can optimize search results by returning the most relevant documents. This can reduce network traffic and enhance system usability. Swaminathan, et al.[29], Zerr, et al.[30], and Wang, et al.[31, 32] achieved ranked search in the single keyword search paradigm using an order-preserving function. Cao, et al.[33] were the first to present a multi-keyword ranked search scheme with"coordinate matching" measurement. However, their search results are ranked based on the number of matching keywords without considering the importance of Different keywords. Consequently, their results are not very accurate. Sun, et al.[34] proposed a multi-keyword ranked search scheme using"cosine measure" techniques. Their scheme achieves better-than-linear search efficiency at the expense of search accuracy.

Xia, et al.[35] recently proposed a dynamic multi-keywords ranked search scheme that uses a secure tree. Chen, et al.[36] proposed a multi-keyword ranked search scheme based on hierarchical clustering index to improve search efficiency. In their scheme, the search time has a linear growth when the size of the data collection has an exponential growth. Other studies have also been conducted on ranked search[37-39].

Verifiable keyword search can detect whether the search results are complete and correct. This can verify inaccurate search results caused by software or hardware failure, storage corruption, or even malicious behavior by a semi-honest server trying to save computation resources. Studies have also been conducted on verifiable keyword search[40-43]. However, these studies are based on MHT (Merkle Hash Tree) and signature techniques, which have expensive communication and computation overhead. Chai, et al.[44] proposed a verifiable searchable encryption scheme. Wang, et al.[45] focused on verification of an empty set as a search result.

4 Public key encryption with keyword search

Consider the following scenario: Bob sends an email with corresponding keywords to Alice. In order to protect the contents of the email and keywords, both are encrypted with Alices public key. However, in this case the email server cannot make a routing decision according to the keywords. Therefore, it is necessary to give the email server the ability to decide whether a certain keyword is contained in an email or not. Meanwhile, the email server cannot learn anything about the contents of the email and keywords. To achieve this goal, Boneh, et al.[46] proposed the first scheme supporting keyword search in a public key system. Fig.3 shows the model of PEKS schemes.

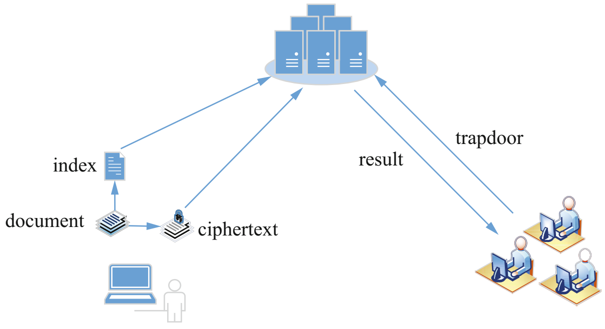

Figure 3 Model of PEKS schemes

4.1 Algorithms description

When user Bob wants to send Alice an email with a number of keywords, $W_1,\cdots,W_k$, Bob sends the following ciphertext: $E_{A_{\text{pub}}}(M)$, $PEKS(A_{\text{pub}}, W_1),\cdots$, $PEKS(A_{\text{pub}}, W_k)$, where M is the content of the email, Apub is Alices public key, and PEKS is an algorithm supporting keyword search. Then, Alice produces a trapdoor T_ω of keyword W and sends T_ω to the gateway. After searching, the gateway returns the emails containing W to Alice. A general PKES scheme contains four polynomial-time algorithms:

- KeyGen(1^k): a key generation algorithm run by Alice. It takes a security parameter k as input, and outputs a public/private key pair A_{pub}, A_{priv}.
- $PEKS(A_{\text{pub}},W)$: a public key encryption algorithm preserving search ability that is run by Bob. It takes the public key Apub of Alice and a keyword W as input, and outputs ciphertext S of W.
- Trapdoor(A_{priv},W): a keyword trapdoor generation algorithm run by Alice. It takes Alices private key A_{priv} and a query keyword W as input, and outputs the trapdoor T_ω of query keyword W.
- Test(A_{pub}, S, T_ω): a test algorithm run by the mail server. It takes Alices public key Apub, a ciphertext S of keyword W' and a trapdoor of query keyword W. If W=W', this algorithm outputs “yes”; otherwise, it outputs “no”.

4.2 Security instructions

In PEKS schemes, the security of the ciphertexts of a keyword (the output of the PEKS algorithm) should be guaranteed. We argue that the ciphertexts should not leak any information about a keyword even under an adaptive chosen keyword attack. In such an attack model, an active attacker has the ability to obtain trapdoors T_W for any keyword W except the challenge keywords. The attacker chooses two challenge keywords W_0 and W_1 for a challenger. The challenger randomly chooses $b \in 0$; 1 and sends the ciphertext of W_b to the attacker. The attacker needs to determine the number b with the knowledge of trapdoors for other keywords. We refer to this kind of security model as PK-CKA2 security, in which the attacker cannot determine whether the ciphertext is from W_0 or W_1. For a detailed definition of this PK-CKA2 security, please see Ref.[46]. This security definition is predominantly used in the remainder of this paper.

4.3 PEKS schemes

4.3.1 Single keyword search

The first PEKS scheme was proposed by Boneh, et al.[46]. It was based on IBE (Identity Based Encryption)[47,48] and consisted of three stages. First, the message sender encrypts her/his message and keywords with the receivers public key in a particular way. The ciphertext is $E_{A_{\text{pub}}}(M)$, $PEKS(A_{\text{pub}},W_1),\cdots$, $PEKS(A_{\text{pub}}, W_k)$, where M is the message content, and Apub is the public key of the receiver. Next, the receiver sends the trapdoor T_ω of query keyword W to the server. Finally, the server searches over the ciphertexts and determine whether a certain keyword is in a particular ciphertext.

The scheme requires two groups, G_1, G_2, of prime order p, which is determined by a security parameter and a bilinear map e: $G_1 \times G_1 \to G_2$. In addition, the scheme requires two hash functions H_1: $\{0, g\}^* \to G_1$ and H_2: $G_2 \to \{0, 1\}^{\log p}$. The detailed algorithm is as follows:

- KeyGen(1^k): The input is a security parameter k that is used to determine the size, p, of the two groups G_1 and G_2. In addition, it needs to pick a random element $\alpha \in \mathbb{Z}_p^*$ and a generator g of G_1. Finally, it outputs a public key $A_{\text{pub}}=[g, h=g^\alpha]$ and a private key $A_{\text{priv}}=\alpha$.
- $PEKS(A_{\text{pub}}, W)$: It first computes $t=e(H_1(W), h^r) \in G_2$, where r is a random element in group Z_p^*. Then, it outputs $PEKS(A_{\text{pub}}, W)=[g^r, H_2(t)]$.
- $Trapdoor(A_{\text{priv}}, W)$: It outputs a trapdoor T_W of certain keyword W, $T_W = H_1(W)^\alpha \in G_1$.
- $Test(A_{\text{pub}}, S, T_W)$: It tests whether $H_2(e(T_W, g^r)) =H_2(t)$. If so, it outputs "yes"; otherwise, it outputs "no".

On one hand, this scheme has been proven PK-CKA2 secure in the Random Oracle model under the difficulty of the Bilinear Diffie-Hellman problem. However, trapdoors should be transmitted over a secure channel, to ensure only the server receives them. Furthermore, trapdoors are produced by using a deterministic encryption, thus the server can store them for further use. On the other hand, the efficiency is not sufficiently high as the PEKS algorithm requires one pairing and two exponentiations. In addition, the test algorithm requires one mapping and the search complexity is linear in the number of keywords per document.

Many studies have been conducted on methods of constructing PEKS schemes. Some typical methods are introduced and classified into three categories based on their security below.

1) Traditional PEKS: Abdalla, et al.[49] proposed a generic solution for transforming an anonymous IBE scheme into a PEKS scheme. They also constructed a PEKS scheme based on temporary keyword from hierarchical IBE. In their scheme,

a trapdoor is produced along with a time interval [s, e], and the mail server is only allowed to test whether a certain keyword w is in the ciphertext during this time interval. This method effectively prevents the server from searching a keyword in the past or future.

Di Crescenzo and Saraswat[50] introduced the first PEKS scheme without bilinear maps. Their scheme, transformed from Cocks IBE scheme[51], is based on a variant of the quadratic residuosity problem. The scheme has been proven PK-CKA2 secure in the RO model. However, it has to calculate 4k Jacobi symbols in order to test whether a certain keyword is in a document, where k is a security parameter. Further, the search time is linear in the number of ciphertexts. Moreover, it has a high storage and communication overhead.

Khader[52] proposed a PEKS scheme based on k-resilient IBE. The scheme is PK-CKA2 secure without an RO model. However, PEKS algorithm is inefficient and involves the calculation of four exponentiations to test whether a certain keyword is in a ciphertext. The scheme was also used to construct another two schemes: one supporting conjunctive keyword search; the other requiring no secure channel to transmit the trapdoors.

2) Secure Channel Free PEKS: Boneh, et al.'s scheme[46] has the limitation that a secure channel is required to transmit the trapdoors; thus, only the server can learn the trapdoor. However, it is not practical as building a secure channel is expensive. To overcome this problem, Baek, et al.[53] proposed SCF-PEKS (Secure Channel Free PEKS), which does not require a secure channel. This new PEKS scheme adds the servers public/private key pair; hence, only the server can search over the ciphertext. Rhee, et al.[54] subsequently enhanced the security of Beak, et al.'s model[53]. In their security enhanced model, an attacker can obtain the relationship between ciphertexts and a trapdoor. Recently, Emura, et al.[55] extended the security of the SCF-PEKS scheme to an adaptive SCF-PEKS scheme based on the anonymous IBE. This model allows an attacker to test query keywords adaptively.

3) Against Keyword Guessing Attack: Byun, et al.[56] first raised an off-line KGA (Keyword Guessing Attack) because of the small space for keywords. They also stated that Boneh, et al.'s scheme is vulnerable to off-line keyword guessing attacks. Yau, et al.[57] asserted that the SCFPEKS[53] and PKE/PEKS[58] schemes are also vulnerable to this attack. They showed that an outside adversary can capture the trapdoor from a public channel (outside KGA), while an inside adversary, such as a malicious server, can capture the trapdoors from either a public or secure channel. Rhee, et al.[59] proposed a scheme against outside KGA that introduces a random variable in the trapdoor computation to make the trapdoors indistinguishable. Fang, et al.[60] proposed a concrete SCF-PEKS against outside KGA.

For inside KGA, Jeong, et al.[61] showed that constructing a secure and consistent PEKS scheme against KGA is impossible when the number of possible keywords is bounded by some polynomial. Xu, et al.[62] presented a PEKS scheme that supports fuzzy keyword search. In their scheme, more than one keywords share the same fuzzy keyword trapdoor such that the server cannot learn the exact keyword. However, their scheme has limitations in terms of security and efficiency. Chen, et al.[63] proposed a new PEKS framework, called dual-server PEKS, which is secure against inside KGA.

Tab.2 compares several classic PEKS schemes. In the table, n denotes the size of the documents set, v denotes the number of distinct keywords per document, p denotes the symmetric prime order pairing, l denotes the length of the keyword in characters, J denotes the Jacobi symbol, and e denotes the exponentiation.

Table 2 Comparison of several PEKS schemes

scheme	search time	index	size security	assumption
Boneh, et al.[46]	nvp	$nv(2e+p)$	PK-CKA2	BDH
Baek, et al.[53] (PEKS-1)	nvp	$nv(3e+p)$	PK-CKA2	CDH
Baek, et al.[53] (PEKS-2)	nvp	$nv(e+2p)$	PK-CKA2	BDH
Crescenzo and Saraswat[50]	$4nlJ$	$4nvlJ$	PK-CKA2	QIP
Khader[52]	$4nve$	$5+3nve$	PK-CKA2	DDH
Rhee, et al.[54]	$nv(e+p)$	$2nve+(7+nv)p$	PK-CKA2	BDH, 1-BDHI
Rhee, et al.[59]	$nv(2e+p)$	$2nve+nvp$	PK-CKA2	BDH, 1-BDHI

4.3.2 Conjunctive keyword search

Park, et al.[64] constructed two schemes supporting conjunctive keyword search in public key systems, in which the computation overhead is efficient and trapdoor size is constant. However, the first scheme requires a number of bilinear paring mappings and the number of private keywords is linear in the size of keyword fields. Boneh and Waters[65] presented a public key scheme based on hidden vector encryption that supports comparison queries, subset queries, and arbitrary conjunctive queries. The attribute values cannot be leaked after decryption. However, the ciphertext size is large because of the use of composite order bilinear groups. In addition, it has a high cost in terms of public key size and encryption operation. Fortunately, the decryption key size and decryption cost are minimized. Shi, et al.[66] proposed a scheme supporting multidimensional range query over encrypted data that can be used to share network audit logs. However, it leaks attribute values after decryption. Hwang and Lee[67] improved the size of ciphertext and private key. Kaze, et al.[68] constructed a PEKS scheme supporting disjunctive keyword search that is based on inner-product predicate encryption. However, the ciphertext size and private key size is bounded by some superpolynomial. Lai, et al.[69] introduced an efficient PEKS scheme supporting arbitrary monotone Boolean predicates that is based on key-policy attribute-based encryption (KP-ABE)[70].

4.3.3 Fuzzy keyword search

Bringer, et al.[71] proposed an error-tolerant searchable encryption in the context of public key that is based on functions of LSH (Locally Sensitive Hashing)[72] and BFS (Bloom Filter with Storage)[73]. An LSH function can be used to reduce the difference among similar items and a BFS function is used to answer set membership queries. If two keywords are sufficiently similar, the hash values of LSHs output is the same. Inputting the LSH values into BFS achieves error tolerant search. Their scheme has been proven PK-CKA2 secure, and protects the search pattern as well using the PIR technique.

4.3.4 Verifiable keyword search

The server is assumed semi-honest such that it may just return a part of the search results or even inaccurate results. Zheng, et al.[74] dealt with this problem with an attribute based encryption scheme called VABKS (Verifiable Attribute-Based Keyword Search). In this scheme, only the user who satisfies the data owners access control policy is allowed to perform search and verify operations. However, verifying the correctness of search results is expensive. Furthermore, the scheme is vulnerable to off-line attacks because the keyword ciphertext and the search token can be easily obtained by an adversary. Liu, et al.[75] proposed a new VABKS scheme, in which a secure channel is not necessary. Their proposed scheme is also relatively efficient in verifying the correctness and integrity of search results.

5 Conclusion and future work

Since the proposal of SSE and PEKS in 2000 and 2004, respectively, the searchable encryption research field has received significant attention. Progress has been made in the following three main directions.

1) Query Expressiveness. Much research has been conducted on extension of query expressiveness. To make schemes more practical, not only exact single keyword search, but also fuzzy keyword search, range search, and subset search are supported. Query results have also been optimized. For example, ranked keyword search finds the most closely related results and verifiable keyword search verifies the correctness and completeness of the results. However, many schemes improve query expressiveness at the expense of efficiency or security. Therefore, future research should pay attention to the tradeoff between query expressiveness and efficiency or security.

2) Efficiency. From the aspect of SSE, the search complexity in some schemes is linear in the number of documents stored on the server. Further, some schemes achieve sublinear search times, in which the search complexity is logarithmic in the number of keywords in all documents. In addition, some schemes achieve optimal search time, in which the search complexity is linear in the number of documents containing the query keywords. With the advent of the big data era, large scale data now need to be stored on servers. Thus, the question of how to deal with large-scale data efficiently is a direction for further work. Moreover, the documents cannot be flexibly updated because the search index is related to the keywords. Hence, the question of how to construct an efficient dynamic SSE scheme is another direction for future work. From the aspect of PKES, a large number of schemes are based on pairing maps. As a result, these schemes are inefficient because pairing maps are inefficient algorithms. Thus, construction of practical PEKS schemes is also a direction for future work.

3) Security. On one hand, although virtually all SE schemes achieve provable secure, they do not use a common security model. That is, different schemes use different security models under different assumptions. Hence, it is always difficult to compare their security. Thus, proposal of a standard security model for SE schemes is a direction for future work. Further, most schemes compromise on search pattern and access pattern. Thus, construction of an efficient scheme that does not leak search pattern and access pattern is another direction for future work.

Future work should also focus on the question of how to apply the ideas underlying SE to deal with other kinds of data. For example, how to search encrypted media data containing image data or video data; how to search an encrypted database containing relational database or non-relational database; and how to search structured social network data.

References

[1] CURTMOLA R, GARAY J, KAMARA S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[C]//Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, USA, 2016: 79-88.

[2] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, 2000: 44-55.

[3] GOH E J. Secure indexes[J]. IACR cryptology eprint archive, 2003: 216.

[4] BLOOM, B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 131(5): 451-459.

[5] VAN LIESDONK P, SEDGHI S, DOUMEN J, et al.

Computationally efficient searchable symmetric encryption[C]//Proceedings of the 7th VLDB Workshop on Secure Data Management, Singapore, 2010: 87-100.

[6] KAMARA S, PAPAMANTHOU C, ROEDER T. Dynamic searchable symmetric encryption[C]//The ACM Conference on Computer and Communications Security, Raleigh, USA, 2012: 965-976.

[7] KAMARA S, PAPAMANTHOU C. Parallel and dynamic searchable symmetric encryption[C]//The 17th International Conference on Financial Cryptography and Data Security, Okinawa, Japan, 2013: 258-274.

[8] STEFANOV E, PAPAMANTHOU C, SHI E. Practical dynamic searchable encryption with small leakage[C]//The 21st Annual Network and Distributed System Security Symposium, California, USA, 2014: 23-26.

[9] CASH D, JAEGER J, JARECKI S, et al. Dynamic searchable encryption in very-large databases: data structures and implementation[J]. IACR cryptology eprint archive, 2014: 853.

[10] NAVEED M, PRABHAKARAN M, GUNTER C A. Dynamic searchable encryption via blind storage[C]//2014 IEEE Symposium on Security and Privacy, Berkeley, USA, 2014: 639-654.

[11] HAHN F, KERSCHBAUM F. Searchable encryption with secure and efficient updates[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, USA, 2014: 310-320.

[12] GAJEK S. Dynamic symmetric searchable encryption from constrained functional encryption[C]//The Cryptographers Track at the RSA Conference, San Francisco, USA, 2016: 75-89.

[13] CASH D, GRUBBS P, PERRY J, et al. Leakage-abuse attacks against searchable encryption[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, USA, 2015: 668-679.

[14] ZHANG Y, KATZ J, PAPAMANTHOU C. All your queries are belong to us: the power of file-injection attacks on searchable encryption[C]//The 25th USENIX Security Symposium, Austin, USA, 2016: 707- 720.

[15] ISHAI Y, KUSHILEVITZ E, LU S, et al. Private large-scale databases with distributed searchable symmetric encryption[C]//Cryptographers Track at the RSA Conference, San Francisco, USA, 2016: 90-107.

[16] KAMARA S, MOATAZ T. SQL on structurally-encrypted databases[R]. Cryptology ePrint Archive, Report 2016/453, 2016.

[17] ASHAROV G, NAOR M, SEGEV G, et al. Searchable symmetric encryption: optimal locality in linear space via two-dimensional balanced allocations[C]//Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, Cambridge, USA, 2016: 1101-1114.

[18] LI J, WANG Q, WANG C, et al. Fuzzy keyword search over encrypted data in cloud computing[C]//The 29th IEEE International Conference on Computer, San Diego, USA, 2010: 441-445.

[19] KUZU M, ISLAM M S, KANTARCIOGLU M. Efficient similarity search over encrypted data[C]//The 28th IEEE International Conference on Data Engineering, Washington, USA, 2012: 1156-1167.

[20] WANG J, MA H, TANG Q, et al. Efficient verifiable fuzzy keyword search over encrypted data in cloud computing[J]. Computer science and information systems, 2013, 10(2): 667-684.

[21] ADJEDJ M, BRINGER J, CHABANNE H, et al. Biometric identification over encrypted data made feasible[C]//The 5th International Conference on Information Systems Security, Kolkata, India, 2009: 86-100.

[22] WANG C, REN K, YU S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data[C]//Proceedings of the IEEE INFOCOM, Orlando, USA, 2012: 451-459.

[23] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[C]//The 2nd International Conference on Applied Cryptography and Network Security, Yellow Mountain, China, 2004: 31-45.

[24] BALLARD L, KAMARA S, MONROSE F. Achieving e_cient conjunctive keyword searches over encrypted data[C]//The 7th International Conference on Information and Communications Security, Beijing, China, 2005: 414-426.

[25] CASH D, JARECKI S, JUTLA C, et al. Highly-scalable searchable symmetric encryption with support for Boolean queries[C]//The 33rd Annual Cryptology Conference (CRYPTO), Santa Barbara, USA, 2013: 353-373.

[26] FABER S, JARECKI S, KRAWCZYK H, et al. Rich queries on encrypted data: beyond exact matches[C]//The 20th European Symposium on Research in Computer Security, Vienna, Austria, 2015: 123-145.

[27] BYUN J W, LEE D H, LIM J. Efficient conjunctive keyword search on encrypted data storage system[C]//The 3rd European Public Key Infrastructure Workshop on Theory and Practice, Turin, Italy, 2006: 184-196.

[28] WANG P, WANG H, PIEPRAYK J. Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups[C]//The 7th International Conference on Cryptology and Network Security, Hong-Kong, China, 2008: 178-195.

[29] SWAMINATHAN A, MAO Y, SU G M, et al. Confidentiality-preserving rank-ordered search[C]//Proceedings of the 2007 ACM Workshop on Storage Security and Survivability, Alexandria, USA, 2007: 7-12.

[30] ZERR S, OLMEDILLA D, NEJDL W, et al. Zerber+ r: Top-k retrieval from a confidential index[C]//Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Saint Petersburg, Russia, 2009: 439-449.

[31] WANG C, CAO N, LI J, et al. Secure ranked keyword search over encrypted cloud data[C]//2010 International Conference on Distributed Computing Systems, Genova, Italy, 2010: 253-262.

[32] WANG C, CAO N, REN K, et al. Enabling secure and efficient ranked keyword search over outsourced cloud data[J]. IEEE transactions on parallel and distributed systems, 2012, 23(8): 1467-1479.

[33] CAO N, WANG C, LI M, et al. Privacy preserving multi-keyword ranked search over encrypted cloud data[C]//The 30th International

Conference on Computer Communications, Shanghai, China, 2011: 829-837.

[34] SUN W, WANG B, CAO N, et al. Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity based ranking[J]. IEEE transactions on parallel and distributed systems, 2014, 25(11): 3025-3035.

[35] XIA Z, WANG X, SUN X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data[J]. IEEE transactions on parallel and distributed systems, 2016, 27(2): 340-352.

[36] CHEN C, ZHU X, SHEN P, et al. An efficient privacy-preserving ranked keyword search method[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4): 951-963.

[37] ORENCIK C, KANTARCIOGLU M, SAVAS E. A practical and secure multi-keyword search method over encrypted cloud data[C]// The 6th IEEE International Conference on Cloud Computing, Santa Clara, USA, 2013: 390-397.

[38] ZHANG W, XIAO S, LIN Y, et al. Secure ranked multi-keyword search for multiple data owners in cloud computing[C]//The 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, USA, 2014: 276-286.

[39] BOUABANA-TEBIBEL T, KACI A. Parallel search over encrypted data under attribute based encryption on the Cloud Computing[J]. Computers & security, 2015, 54:77-91.

[40] PANG H H, TAN K L. Authenticating query results in edge computing[C]//Proceedings of the 20th International Conference on Data Engineering, Boston, USA, 2004: 560-571.

[41] LI F, HADJILEFTHERIOU M, KOLLIOS G, et al. Authenticated index structures for outsourced databases[M]. Handbook of database security-applications and trends, Springer US, 2008: 115-136.

[42] MOURATIDIS K, SACHARIDIS D, PANG H H. Partially materialized digest scheme: an efficient verification method for outsourced databases[J]. The International journal on very large data bases, 2009, 18(1): 363-381.

[43] KUROSAWA K, OHTAKI Y. UCsecure searchable symmetric encryption[C]//International Conference on Financial Cryptography and Data Security, Kralendijk, Bonaire, 2012: 285-298.

[44] CHAI Q, GONG G. Verifiable symmetric searchable encryption for semi-honest but-curious cloud servers[C]//Proceedings of IEEE International Conference on Communications, Ottawa, Canada, 2012: 917-922.

[45] WANG J, CHEN X, HUANG X, et al. Verifiable auditing for outsourced database in cloud computing[J]. IEEE transactions on computers, 2015, 64(11): 3293-3303.

[46] BONEH D, DI CRESCENZO G, OSTROVSKY R, et al. Public key encryption with keyword search[C]//International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2004: 506-522.

[47] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[J]. SIAM journal on computing, 2003, 32(3): 586-615.

[48] SHAMIR A. Identity-based cryptosystems and signature schemes[C]//Workshop on the Theory and Application of Cryptographic Techniques Santa, Barbara, California, USA, 1984: 47-53.

[49] ABDALL M, BELLARE M, CATALANO D, et al. Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions[J]. Journal of cryptology, 2008, 21(3): 350-391.

[50] DI CRESCENZO G, SARASWAT V. Public key encryption with searchable keywords based on Jacobi symbols[C]//International Conference on Cryptology in India, Chennai, India, 2007: 282-296.

[51] COCKS C. An identity based encryption scheme based on quadratic residues[C]//The 8th IMA International Conference on Cryptography and Coding, Cirencester, UK, 2001: 360-363.

[52] KHADER D. Public key encryption with keyword search based on k-resilient IBE[C]//International Conference on Computational Science and Its Applications, Kuala Lumpur, Malaysia, 2007: 1086-1095.

[53] BAEK J, SAFAVI-NAINI R, SUSILO W. Public key encryption with keyword search revisited[C]//International conference on Computational Science and Its Applications, Perugia, Italy, 2008: 1249-1259.

[54] RHEE H S, PARK J H, AUSILO W, et al. Improved searchable public key encryption with designated tester[C]//Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, Sydney, Australia, 2009: 376-379.

[55] EMURA K, MIYAJI A, RAHMAN M S, et al. Generic constructions of secure channel free searchable encryption with adaptive security[J]. Security and communication networks, 2015, 8(8): 1547-1560.

[56] BYUN J W, RHEE H S, PARK H A, et al. O_-line keyword guessing attacks on recent keyword search schemes over encrypted data[C]//The 3rd VLDB Workshop on Secure Data Management. Springer Berlin Heidelberg, 2006: 75-83.

[57] YAU W C, HENG S H, GOI B M. O_-line keyword guessing attacks on recent public key encryption with keyword search schemes[C]// International Conference on Autonomic and Trusted Computing, Seoul, Korea, 2008: 100-105.

[58] BAEK J, SAFAVI-NAINI R, SUSILO W. On the integration of public key data en cryption and public key encryption with keyword search[C]//The 9th International Conference on Information Security, Samos Island, Greece, 2006: 217-232.

[59] RHEE H S, PARK J H, SUSILO W, et al. Trapdoor security in a searchable publickey encryption scheme with a designated tester[J]. Journal of systems and software, 2010, 83(5): 763-771.

[60] FANG L, SUSILO W, GE C, et al. Public key encryption with keyword search secure against keyword guessing attacks without random oracle[J]. Information sciences, 2013, 238: 221-241.

[61] JEONG I R, KWON J O, HONG D, et al. Constructing PEKS schemes secure against keyword guessing attacks is possible?[J]. Computer communications, 2009,32(2): 394-396.

[62] XU P, JIN H, WU Q, et al. Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack[J]. IEEE transactions on computers, 2013, 62(11): 2266-2277.

[63] CHEN R, MU Y, YANG G, et al. Dual-server public-key encryption with keyword search for secure cloud storage[J]. IEEE transactions on information forensics and security, 2016, 11(4): 789-798.

[64] PARK D J, KIM K, LEE P J. Public key encryption with conjunctive field keyword search[C]//The 5th International Workshop on Information Security Applications, Jeju Island, Korea, 2004: 73-86.

[65] BONEH D, WATERS B. Conjunctive, subset, and range queries on encrypted data[C]//The 4th Theory of Cryptography Conference, Amsterdam, Netherlands, 2007: 535-554.

[66] SHI E, BETHENCOURT J, CHAN T H H, et al. Multi-dimensional range query over encrypted data[C]//2007 IEEE Symposium on Security and Privacy (SP'07). Oakland, California, USA, 2007: 350-364.

[67] HWANG Y H, LEE P J. Public key encryption with conjunctive keyword search and its extension to a multi-user system[C]//The 1st International Conference on Pairing- Based Cryptography, Tokyo, Japan, 2007: 2-22.

[68] KATZ J, SAHAI A, WATERS B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]//The 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, 2008: 146-162.

[69] LAI J, ZHOU X, DENG R H, et al. Expressive search on encrypted data[C]//The 8th ACM symposium on Information, computer and communications security, Hangzhou, China, 2013: 243-252.

[70] LEWKO A, OKAMOTO T, SAHAI A, et al. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption[C]//The 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, 2010: 62-91.

[71] BRINGER J, CHABANNE H, KINDARJI B. Error-tolerant searchable encryption[C]//2009 IEEE International Conference on Communications, Dresden, Germany, 2009: 1-6.

[72] INDYK P, MOTWANI R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]//Proceedings of the 30th annual ACM symposium on Theory of computing, Dallas, Texas, USA, 1998: 604-613.

[73] BONEH D, KUSHILEVITZ E, OSTROVSKY R, et al. Public key encryption that allows PIR queries[C]//The 27th Annual International Cryptology Conference, Santa Barbara, USA, 2007: 50-67.

[74] ZHENG Q, XU S, ATENIESE G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data[C]//2014 IEEE Conference on Computer Communications, Toronto, Canada, 2014: 522-530.

[75] LIU P, WANG J, MA H, et al. Efficient verifiable public key encryption with keyword search based on KP-ABE[C]//The 9th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Guangdong, China, 2014: 584-589.

About the authors

WANG Yunling [corresponding author] received a masters degree in electronics and communication engineering from Xidian University, China, in 2015. She is now a Ph.D. candidate in the area of cryptography at Xidian University, China. Her research interests include cloud computing and applied cryptography. (Email: ylwang0304@ 163.com)

WANG Jianfeng received an M.S. degree in mathematics and a Ph.D. degree in cryptography from Xidian University, in 2013 and 2016, respectively. He currently works at Xidian University. His research interests include applied cryptography and secure outsourced storage. (E-mail: jfw01@163.com)

CHEN Xiaofeng received B.S. and M.S. degrees in mathematics from Northwest University, Xi’an, China, in 1998 and 2000, respectively, and a Ph.D. degree in cryptography from Xidian University, Xi’an, in 2003, where he is currently a Professor. His research interests include applied cryptography and cloud computing security. He has authored over 100 research papers in refereed international conferences and journals. His work has been cited over 5 300 times in Google scholar. He is on the editorial board of IEEE Transactions on dependable and secure computing security and communication networks, telecommunication systems, etc. He has served as the program/general chair or a program committee member for over 30 international conferences. (Email: xfchen@xidian.edu.cn)