Privacy-Preserving Noisy Keyword Search in Cloud Computing*

Xiaoqiong Pang[1,2], Bo Yang[3], and Qiong Huang[1]

[1] South China Agricultural University, Guangzhou 510642, China
[2] North University of China, Taiyuan 030051, China
[3] Shaanxi Normal University, Xi'an 710062, China
pxqiong@126.com, byang@snnu.edu.cn, csqhuang@alumni.cityu.edu.hk

Abstract. We consider the following problem: a user with either limited resources or limited expertise wants to outsource its private documens, which are associated with noisy keywords, to an untrusted cloud server in a private manner, while maintaining the ability to retrieve the stored data in a fault-tolerant manner. For example, the organization of homeland security wishes to outsource its private criminal database comprised of a set of criminal dossiers to the cloud server in encrypted form and hopes to retrieve encrypted dossiers by biometrics.

In this paper, we first present a general framework for searching on private-key encrypted data by noisy keywords in a fault-tolerant manner. Then we propose a concrete scheme which is proved secure against an adaptive adversary under well-defined security definition. It achieves search in two rounds of communication, and requires an amount of work from the server that is linear in the number of noisy keywords.

Keywords: Noisy Keyword, Searchable Encryption, Storage Outsourcing.

1 Introduction

Consider a cloud data system: a user with either limited resources or limited expertise wishes to outsource its private data to an untrusted cloud server in a private manner, while maintaining the ability to retrieve the stored data. The informal security requirement for this system claims that the cloud server should not learn any useful information about the data that it stores for the user and the queried words. A feasible solution to preserve privacy is to design a searchable encryption scheme such that: 1) the records are stored in disguised/encrypted form. 2) the key employed to encrypt the data is kept secret from the cloud server. 3) the records can be searched for securely and efficiently[21].

Motivated by the practical relevance of this problem, the research community has mainly considered two models for searching with privacy (see the related work for a detailed description) quite extensively and presented a number of techniques

* This work is supported by the National Natural Science Foundation of China under Grants 60973134, 61173164, 61272436 and 61103232.

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 154–166, 2012.

that vary in costs and the levels of security guarantees, e.g. [1–7, 14, 15, 17, 19–21]. Unfortunately, except[14, 5] all of solutions above that only support exact keyword search do not apply to the situation where the keywords associated with the records are noisy data. For example, the organization of homeland security wishes to outsource its private criminal database comprised of a set of criminal dossiers to the cloud server in encrypted form and hopes to retrieve encrypted dossiers by biometrics whenever. As we know, biometric traits are the deciding factors to recognize a person's identity by using special features such as face, finger-prints, iris, voice, DNA, and so on. However, biometric data are noisy, even two readings of the same biometric source are rarely identical. Therefore, exact-match search over biometric data does not work. It seems that searching on encrypted fuzzy data in a private manner is much more difficult. To address this problem we present a solution for privacy-preserving noisy-keyword-based search on remote encrypted data in a fault-tolerant manner.

1.1 Related Work

Various methods have been proposed for searching on encrypted data. The existing solutions can be divided into two classes according to the models of searching on encrypted data with privacy.

Searching on private-key encrypted data. In this setting, the user itself encrypts the database and uploads it to the server so that only somebody holding the private key can obtain the records it retrieves. Solution in this model can be realized with optimal security by using the work of Goldreich and Ostrovsky about oblivious RAM in [12], which hides the identities of the data items being accessed from a remote server. Because of the overwhelming cost of the oblivious RAM protocol (see the analyse in [18]), it was always quoted as a theoretical solution that was clearly infeasible in practice. Compared with the method of [12], Song et al.[20] presented a solution with little communication in one round of interaction. However, their construction is not secure against statistical analysis. In [6], Chang and Mitzenmacher proposed a scheme based on index. In their scheme, the overhead required for a query is proportional to the number of files. In [7], Curtmola et al. presented two solutions, in which the server's search time is optimal but updates to the index are inefficient. The authors also considered adaptive security of their schemes. Li et al. [14] provided a solution for privacy-preserving fuzzy search on encrypted data. To achieve fuzzy search, the user constructs a wildcard-based fuzzy keyword set for each keyword according to some edit distance before outsourcing data, which incurs extra large storage. There also exists other solutions in this model, however, almost all of existing solutions except [14] do not apply to the situation where the keywords associated with the records are noisy data. Our goal is to design an efficient and secure noisy-keyword-based searchable encryption scheme in a fault-tolerant manner under this model.

Searching on public-key encrypted data. In this setting, the party who searches over the data can be different from the party that generates it. In other words, anyone with access to a party's public key can add encrypted data to the

database, but only the party holding the decryption key can decrypt and retrieve. As far as we know, solution in this model was proposed for the first time by Boneh et al. in [2]. The scheme in [2] reveals the users access pattern. In [3], Boneh et al. presented a solution that guarantees the complete privacy of queries by sacrificing efficiency. Bellare et al.[1] proposed fast search schemes in the random oracle model. Since the encryption algorithm they utilized is deterministic, their constructions provide weak security guarantees. Bringer et al.[5] described a primitive called Public Key Error-Tolerant Searchable Encryption and applied it to biometric identification. In addition, some works about more complex search have been studied such as range queries[4, 19], join queries[15] and conjunctive searches[17, 4].

1.2 Our Contribution

We propose a general framework for noisy-keyword-based searchable private-key encryption in a fault-tolerant manner. This general scheme allows to search on encrypted/disguised data with a noisy keyword. In addition, different from all the previous schemes, which use the same private key to encrypt all documents, our scheme utilizes the fuzzy extractor, which can extract a uniformly random string k_i from each noisy keywords w_i in a noise-tolerant way. The extracted k_i is used to encrypt the documents which are associated with noisy keyword w_i. The advantage follows the fact that our scheme does not store the extracted k_i. Instead, the noisy keyword itself effectively acts as the key, and only when the "correct keyword" is presented will the documents be decrypted. Therefore, if a k_i is leaked for some reason, the privacy of the documents that are not labeled with w_i is still preserved. Another important notion we employ is secure sketch, which can be used to construct the fuzzy extractor.

In accordance with the simulation-based security definition presented in [7] and the real condition in noisy-keyword-based search, we present a notion named match pattern, and use it in security proof. Compared with the notion of search pattern, which refers to the information whether searches are for the same word or not[7], match pattern indicates how well the queried words match the noisy keywords.

We present a concrete scheme which is proved secure against an adaptive adversary under the simulation-based security definition presented in [7]. It achieves search in two rounds of interaction, and requires an amount of work from the server that is linear in the number of noisy keywords. Our idea comes from the **SSO/Approx/Squ** protocol proposed by Du and Atallah in [10], where they address the problem that a user, who outsources its database comprised of N keywords to the server, wants to know which keyword in the database is closest to the query string x. We use this protocol as a building block in our concrete scheme.

Organization. The remainder of this paper is structured as follows. In Section 2 we introduce the notions of secure sketch and fuzzy extractor. In Section 3 we present a general framework for noisy-keyword-based searchable private-key

encryption and describe its security requirement. Section 4 describes a concrete scheme, argues its correctness and security. We consider performance issues in Section 5. Finally, Section 6 concludes the paper.

2 Useful Tools

2.1 Preliminaries

If X is a random variable, we denote the probability distribution over the range of the variable by X for simplicity. We denote the uniform distribution over binary strings of length l by U_l. For a set A, the notation $a \leftarrow_r A$ means a is chosen uniformly at random from A. For a matrix $\mathbf{B}$, the rank of $\mathbf{B}$ is denoted by $\mathsf{Rank}(\mathbf{B})$.

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. The edit distance between two strings is the number of operations required to transform one string into the other. The set difference between two sets is the size of the symmetric difference of the two sets.

The *min-entropy* $\mathbf{H}_\infty(X)$ of a random variable X is $-\log(\max_x \Pr(X = x))$. A random variable with min-entropy at least m is called an m-source. We define *average min-entropy* of X given Y to be the logarithm of the average probability of the most likely value of X given Y, namely $\widetilde{\mathbf{H}}_\infty(X|Y) = -\log(\mathrm{E}_{y \leftarrow Y}[2^{-\mathbf{H}_\infty(X|Y)}])$. The statistical distance between two probability distributions X and Y is defined as $\mathsf{SD}[X, Y] = \frac{1}{2}\sum_v |\mathrm{P}[X = v] - \mathrm{P}[Y = v]|$. We use $X \approx_\varepsilon Y$ to denote that X and Y are at distance at most ε.[9]

2.2 Secure Sketch and Fuzzy Extractor

The following descriptions and definitions are based on [9, 8]. Let $\mathcal{M}$ be a metric space with distance function dis. t is a threshold value. Informally, a secure sketch allows precise reconstruction of a noisy input $w \in \mathcal{M}$ from any w' close to w without revealing much about w.

Definition 1. *An* (m, m', t)-secure sketch *is a pair of efficient randomized procedures* $(\mathsf{SS}, \mathsf{Rec})$ *such that the following hold:*

1. The sketching procedure SS *on input* $w \in \mathcal{M}$ *returns a string* $s \in \{0,1\}^*$. *The recovery procedure* Rec *takes as input an element* $w' \in \mathcal{M}$ *and* $s \in \{0,1\}^*$.

2. Correctness: *If* $\mathsf{dis}(w, w') \le t$, *then* $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$.

3. Security: *For any m-source over* $\mathcal{M}$, *the min-entropy of W given s is high: For any (W,E), if* $\widetilde{\mathbf{H}}_\infty(W|E) \ge m$, *then* $\widetilde{\mathbf{H}}_\infty(W|\mathsf{SS}(W), E) \ge m'$.

Next we present the definition of fuzzy extractor. As opposed to secure sketch, the goal of fuzzy extractor is not to recover the original noisy input, but to extract a close-to-uniform string R from w and then reproduce R exatcly from any w' that is close to w. The reproduction is done with the help of the helper string P that is generated during the initial extraction procedure.

Definition 2. *An* (m, l, t, ε)-fuzzy extractor *is a pair of efficient randomized procedures* (Gen,Rep) *such that the following hold:*

1. Given $w \in \mathcal{M}$, Gen *outputs an extracted string* $R \in \{0,1\}^l$ *and a helper string* $P \in \{0,1\}^*$. Rep *takes as input an element* $w' \in \mathcal{M}$ *and a string* $P \in \{0,1\}^*$.

2. Correctness: *If* $\mathsf{dis}(w, w') \leq t$ *and* $(R, P) \leftarrow \mathsf{Gen}(w)$, *then* $\mathsf{Rep}(w', P) = R$.

3. Security: *For all m-sources* W *over* $\mathcal{M}$, *the string* R *is nearly uniform even given* P; *that is, if* $\widetilde{\mathbf{H}}_\infty(W|E) \geq m$, *then* $(R, P, E) \approx_\varepsilon (U_l, P, E)$.

A secure sketch can be used to construct a fuzzy extractor that extracts a key by combining the sketch with a strong randomness extractor, such as the universal hash functions which extract well even from conditional min-entropy.

Lemma 1. *Suppose we compose an* (m, m', t)*-secure sketch* (SS,Rec) *for a space* $\mathcal{M}$ *and a universal hash function* $\mathsf{Ext}\colon \mathcal{M} \to \{0,1\}^l$ *as follows: In* Gen,*choose a random* i *and let* $P = (\mathsf{SS}(w), i)$ *and* $R = \mathsf{Ext}(w; i)$; *let* $\mathsf{Rep}(w', (s, i)) = \mathsf{Ext}(\mathsf{Rec}(w', s), i)$. *The result is an* (m, l, t, ε)*-fuzzy extractor with* $l = m' + 2 - 2\log(1/\varepsilon)$.

Remark. It is noting that: 1)There exist many concrete constructions of secure sketch and fuzzy extractor for the Hamming distance, set difference, edit distance and other notions of distance, e.g. [9, 16]. 2) The concrete constructions of secure sketch and fuzzy extractor are not our goal, we use them as building blocks to construct our scheme. 3)In our general framework we do not assume any particular metrics to measure the closeness between w' and w.

3 General Framework

3.1 Definition for Noisy-Keyword-Based Searchable Private-Key Encryption

We begin by defining a general framework for searching on encrypted data by noisy keywords in a fault-tolerant manner. A user owns a private document set $\mathbf{D} = \{D_1, \cdots, D_N\}$, each document is associated with corresponding noisy keyword. We consider an honest-but-curious server in the sense that it correctly follows the protocol specification while it attempts to derive as much information as possible from user's queries and access patterns.

In the following definitions, we use $\mathsf{id}(D_i)$ to denote the identifier of the document D_i, and *threshold* to denote the predetermined threshold value. Let $\delta(\mathbf{D})$ denote the set of all noisy keywords in $\mathbf{D}$. If $\mathsf{dis}(x, w_i) \leq threshold$, we refer to x and w_i as synonyms. W.l.o.g, we assume that for a noisy keyword $w_i \in \delta(\mathbf{D})$, if x is a synonym of w_i, then x is not synonymous with all the other noisy keywords in $\delta(\mathbf{D})$, furthermore, it is evident that $\mathsf{dis}(x, w_i) = \mathbf{min}_{j=1}^{|\delta(\mathbf{D})|}\mathsf{dis}(x, w_j)$. We use $\mu(w_i)$ to denote the set of all the synonyms of w_i, and $\delta'(\mathbf{D})$ to denote $\bigcup_{i=1}^{|\delta(\mathbf{D})|}\mu(w_i)$. Let $\mathbf{D}(w_i)$ be the set of identifiers of documents in $\mathbf{D}$ that are

labeled with noisy keyword w_i. For some queried word $x_i \in \delta'(\mathbf{D})$, if x_i is synonymous with w_j, then $\mathbf{D}(x_i) = \mathbf{D}(w_j)$. We denote the set of ciphertexts of all documents in $\mathbf{D}$ by $\mathbf{C}=(c_1, \cdots, c_N)$ and the set of ciphertexts of documents in $\mathbf{D}$ that are associated with noisy keyword w_i by $\mathbf{C}_{w_i}$.

Definition 3. *A Noisy-Keyword-based Searchable Private-key Encryption scheme* NKSPE= (KeyGen, Document-Storage, Search) *consists of three phases:*

1. KeyGen(1^k): *given a security parameter k as input, output a secret key K.*
2. Document-Storage$(K, \mathbf{D})$: *given a document set* $\mathbf{D} = \{D_1, \cdots, D_N\}$ *and a secret key K as input, output a secure index I and a series of ciphertexts* $\mathbf{C}=(c_1, \cdots, c_N)$. *For each* D_j *associated with noisy keyword* w_i, *the ciphertexts are produced by employing a fuzzy extractor and a private-key encryption scheme as follows:*
   Gen(w_i) : *given* $w_i \in \delta(\mathbf{D})$ *as input, output an extracted string* k_i *and a helper string* P_i.
   Enc(D_j, k_i) : *given* D_j *and* k_i *(extracted from* w_i*) as input, output ciphertext* c_j *under private key* k_i .
3. Search(I, x) *is an interactive two party protocol between the user and the server. For any query* $x \in \delta'(\mathbf{D})$, *the user generates corresponding* ***trapdoor*** *t under secret key K. Given the trapdoor t and index I, the server can find the* w_i *which is synonymous with x and the user will get encrypted documents* $\mathbf{C}_{w_i}$. *Then the decryption procedure utilizing fuzzy extractor is as follows:*
   Rep(x, P_i) : *given x and* P_i *as input, reproduce the decryption key* k_i.
   Dec$(\mathbf{C}_{w_i}, k_i)$: *given* $\mathbf{C}_{w_i}$ *and* k_i *as input, output corresponding documents.*

We now define correctness for such an NKSPE scheme.

Definition 4. *For all* $k \in \mathbb{N}$, *for all K output by* KeyGen(1^k), *for all* $(I, \mathbf{C})$ *output by* Document-Storage$(K, \mathbf{D})$, *for all* $w_i \in \delta(\mathbf{D})$, *for all* $x \in \delta'(\mathbf{D})$, *if* $\mathsf{dis}(x, w_i) \leq threshold$ *and* $(k_i, P_i) \leftarrow \mathsf{Gen}(w_i)$, *then*

$$\mathbf{C}_{w_i} \leftarrow \mathsf{Search}(I, x) \wedge k_i \leftarrow \mathsf{Rep}(x, P_i) \wedge \mathsf{Dec}(k_i, c_j) = D_j, \ for\ all\ c_j \in \mathbf{C}_{w_i}.$$

We say the NKSPE *scheme is correct.*

3.2 Security Definition

In this paper, we will follow the security definitions presented in [7]. The security requirement for searchable encryption is typically characterized as one that nothing should be leaked except the result of a search, which is referred to as access pattern. However, except for oblivious RAMs, there exists no practical construction that satisfies this requirement, all existing exact-match schemes also disclose whether queries are for the same keyword or not, which is referred to as search pattern. Curtmola et al.[7] analyzed two existing security definitions that had been used for searching on private-key encrypted data: IND2-CKA(Indistinguishability against Chosen-Keyword Attacks) in [11] and

a simulation-based definition in [6]. They pointed out that IND2-CKA was not strong enough to ensure that an index could be safely employed to construct a searchable private-key encryption scheme. For the simulation-based definition in [6], they pointed out that even an insecure scheme would satisfy this definition and this definition was inherently non-adaptive. In [7], Curtmola et al. proposed more accurate security definitions for searchable private-key encryption scheme under non-adaptive and adaptive adversarial models [1]respectively. In next section we will present an adaptively secure NKSPE scheme under security definitions in [7].

Next we introduce four auxiliary notions we will use in security definition. Except Definition 7, the following descriptions and definitions are based on [7].

Definition 5. (History) *Let* $\mathbf{D}$ *be a set of N documents. A q-query* history *over* $\mathbf{D}$ *is a tuple* $H = (\mathbf{D}, \boldsymbol{x})$ *that consists of the document set* $\mathbf{D}$ *and a vector of q query keywords* $\boldsymbol{x} = (x_1, \cdots, x_q)$*, where* $x_i \in \delta'(\mathbf{D})$*, for all* $i \in [1, q]$*.*

Definition 6. (Access pattern) *The access pattern induced by a q-query history* $H = (\mathbf{D}, \boldsymbol{x})$ *is a tuple* $\alpha(H) = (\mathbf{D}(x_1), \cdots, \mathbf{D}(x_q))$*.*

Compared with the notion of search pattern in exact-match scenario, we present a notion termed match pattern in noisy-keyword-based search scenario, which indicates how well the queried words match the noisy keywords.

Definition 7. (Match pattern) *The match pattern induced by a q-query history* $H = (\mathbf{D}, \boldsymbol{x})$ *is a matrix* $\beta(H) = (\mathbf{b}_1, \cdots, \mathbf{b}_q)$*, where* $\mathbf{b}_i = (b_i^1, \cdots, b_i^{|\delta(\mathbf{D})|})^\top$ *and* b_i^j *indicates the closeness degree*[2] *between queried word* x_i *and noisy keyword* w_j*, where* $i \in [1, q]$ *and* $j \in [1, |\delta(\mathbf{D})|]$*.*

Next, we present the definition of the trace of a history, which can be considered as the information the user would like to leak about the history to the server. This should include document size and identifier, the number of keywords used in all documents , access pattern and match pattern.

Definition 8. (Trace) *Let* $\mathbf{D}$ *be a set of N documents. The trace induced by a q-query history* $H = (\mathbf{D}, \boldsymbol{x})$ *is a sequence* $\tau(H) = (\mathsf{id}(D_1), \cdots, \mathsf{id}(D_N), |D_1|, \cdots, |D_N|, |\delta(\mathbf{D})|, \alpha(H), \beta(H))$ *consisting of sizes and identifiers of the documents in* $\mathbf{D}$*, the number of noisy keywords, the access and match patterns induced by H.*

We now present the adaptive simulation-based security definition, where we require that the view of an adversary(including the index, the trapdoors and the ciphertexts) generated from an adversarially and adaptively chosen history be simulatable given only the trace.

[1] Non-adaptive adversaries make queries without considering previous trapdoors and search outcomes while adaptive ones make queries according to previous trapdoors and search results.

[2] The actual closeness degree depends on the particular metric in concrete scheme.

Definition 9. (Adaptive semantic security) *Let* NKSPE=(KeyGen, Document-Storage, Search) *be a Noisy-Keyword-based Searchable Private-key Encryption scheme, $k \in \mathbb{N}$ be the security parameter, $\mathcal{A} = (\mathcal{A}_0, \cdots, \mathcal{A}_q)$ be an adversary such that $q \in \mathbb{N}$ and $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_q)$ be a simulator and consider the following probabilistic experiments* $\mathbf{Real}_{\mathsf{NKSPE},\mathcal{A}}(k)$ *and* $\mathbf{Sim}_{\mathsf{NKSPE},\mathcal{A},\mathcal{S}}(k)$*:*

$\mathbf{Real}_{\mathsf{NKSPE},\mathcal{A}}(k)$	$\mathbf{Sim}_{\mathsf{NKSPE},\mathcal{A},\mathcal{S}}(k)$
$K \leftarrow \mathsf{KeyGen}(1^k)$	$(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(1^k)$
$(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(1^k)$	$(I, \mathbf{C}, st_{\mathcal{S}}) \leftarrow \mathcal{S}_0(\tau(\mathbf{D}))$
$(I, \mathbf{C}) \leftarrow \mathsf{Document\text{-}Storage}(K, \mathbf{D})$	$(x_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$
$(x_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$	$(t_1, st_{\mathcal{S}}) \leftarrow \mathcal{S}_1(st_{\mathcal{S}}, \tau(\mathbf{D}, x_1))$
generate trapdoor t_1 *from* K *and* x_1	*for* $2 \le i \le q$
for $2 \le i \le q$	$\quad(x_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, t_1, \cdots, t_{i-1})$
$\quad(x_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, t_1, \cdots, t_{i-1})$	$\quad(t_i, st_{\mathcal{S}}) \leftarrow \mathcal{S}_i(st_{\mathcal{S}}, \tau(\mathbf{D}, x_1, \cdots, x_i))$
$\quad$*generate trapdoor* t_i *from* K *and* x_i	*let* $\boldsymbol{t} = (t_1, \cdots, t_q)$
let $\boldsymbol{t} = (t_1, \cdots, t_q)$	*output* $\mathbf{v}= (I, \mathbf{C}, \boldsymbol{t})$ *and* $st_{\mathcal{A}}$
output $\mathbf{v}= (I, \mathbf{C}, \boldsymbol{t})$ *and* $st_{\mathcal{A}}$	

We say that NKSPE *is adaptively semantically secure if for all polynomial-size adversaries $\mathcal{A} = (\mathcal{A}_0, \cdots, \mathcal{A}_q)$ such that $q = \mathsf{poly}(k)$, there exists a non-uniform polynomial-size simulator $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_q)$, such that for all polynomial-size distinguisher $\mathcal{D}$,*

$$|\Pr[\mathcal{D}(\mathbf{v}, st_{\mathcal{A}}) = 1 : (\mathbf{v}, st_{\mathcal{A}}) \leftarrow \mathbf{Real}_{\mathsf{NKSPE},\mathcal{A}}(k)]$$
$$-\Pr[\mathcal{D}(\mathbf{v}, st_{\mathcal{A}}) = 1 : (\mathbf{v}, st_{\mathcal{A}}) \leftarrow \mathbf{Sim}_{\mathsf{NKSPE},\mathcal{A},\mathcal{S}}(k)]| \le \mathsf{negl}(k)$$

where $st_{\mathcal{A}}$ is a string that captures $\mathcal{A}$'s state, and the probabilities are taken over the random coins of KeyGen, Document-Storage *and trapdoor.*

4 Efficient and Secure Noisy-Keyword-Based Searchable Private-Key Encryption

In this section we present our concrete NKSPE construction, and argue its correctness and security according to the definitions in section 3. The concrete scheme is described in Fig.1.

Different noisy data, such as different biometric information, has different error patterns[3]. In our concrete construction, we consider $\sum_{k=1}^{n}(w_{ik} - x_k)^2$ as metric to measure the closeness between binary strings $x = x_1 \cdots x_n$ and $w_i = w_{i,1} \cdots w_{i,n}$. In fact, the value $\sum_{k=1}^{n}(w_{ik} - x_k)^2$ is the same with the Hamming distance between $x = x_1 \cdots x_n$ and $w_i = w_{i,1} \cdots w_{i,n}$.

[3] Biometric measurements usually have to be processed before they fall under a suitable metric space; For example, techniques such as IrisCode transform images of irises into strings in the Hamming space. This procedure is itself a research area. Transformations of biometric feature vectors into binary strings have been discussed in [13].

1. $\mathsf{KeyGen}(1^k)$: sample $K_1 \leftarrow_r \{0,1\}^k$ and invertible matrix $\mathbf{Q} \leftarrow_r \mathbf{M}_{n+3,n+3}(\mathbb{F})$, where $\mathbf{M}_{n+3,n+3}(\mathbb{F})$ is a predetermined finite integral matrix group consisting of invertible $(n+3) \times (n+3)$ matrices over field $\mathbb{F}$. output $K = (K_1, \mathbf{Q})$.
2. $\mathsf{Document\text{-}Storage}(K, \mathbf{D})$:
   - $\mathsf{Init}(\mathbf{D})$:
     1) scan $(\mathbf{D})$ and generate the set $\delta(\mathbf{D})$
     2) for all $w_i \in \delta(\mathbf{D})$, output $\mathbf{D}(w_i)$
   - $\mathsf{BuildIndex}(K, \delta(\mathbf{D}), \{\mathbf{D}(w_i) | w_i \in \delta(\mathbf{D})\})$:
     3) $R \leftarrow_r \mathbb{F}$
     4) for $1 \le i \le |\delta(\mathbf{D})|$, creat index $I_{w_i} = (\mathbf{A}_{w_i}, B_{w_i})$
     4.1) for each $w_i = w_{i,1} \cdots w_{i,n} \in \delta(\mathbf{D})$, $R_i \leftarrow_r \mathbb{F}$, let $\mathbf{w}_i = (\sum_{k=1}^{n} w_{ik}^2 + R - R_i, w_{i1}, \cdots, w_{in}, 1, R_i)$, then compute $\mathbf{A}_{w_i} = \mathbf{Q}\mathbf{w}_i^\top$
     4.2) compute $\mathsf{EXT.Gen}(w_i)$, output an extracted string k_i and a helper P_i
     4.3) compute $B_{w_i} = \pi_{K_1}(\mathbf{D}(w_i) || P_i)$
   - $\mathsf{Data\text{-}Storage}(\mathbf{D})$:
     5) for each D_j associated with noisy keyword $w_i (1 \le j \le N, 1 \le i \le |\delta(\mathbf{D})|)$, compute $\mathsf{SKE.Enc}(D_j, k_i)$, output ciphertext c_j under private key k_i
3. $\mathsf{Search}(I, x)$:
   - for any query $x = x_1 \cdots x_n$, $R_A \leftarrow_r \mathbb{F}$, the user constructs a vector $\mathbf{x} = (1, -2x_1, \cdots, -2x_n, R_A, 1)$, then generates trapdoor $\mathbf{t} = \mathbf{x}\mathbf{Q}^{-1}$ under secret key $\mathbf{Q}$. The user sends $\mathbf{t}$ to the server.
   - for $1 \le i \le |\delta(\mathbf{D})|$, the server computes $\mathbf{t} \times \mathbf{A}_{w_i} = \mathbf{x}\mathbf{Q}^{-1}\mathbf{Q}\mathbf{w}_i^\top = \mathbf{x}\mathbf{w}_i^\top$, gets $dis' = \mathbf{min}_{i=1}^{|\delta(\mathbf{D})|} \mathbf{x}\mathbf{w}_i^\top$ and the corresponding i. The server then returns (dis', B_{w_i}) to the user.
   - the user computes $\mathsf{dis}(x, w_i) = dis' + \sum_{k=1}^{n} x_k^2 - R - R_A$. If $\mathsf{dis}(x, w_i) > threshold$, it implies that there exists no matched record with overwhelming probability. Otherwise, the user computes $\pi^{-1}(K_1, B_{w_i})$, gets $\mathbf{D}(w_i)$ and P_i. The user then sends search result, i.e. $\mathbf{D}(w_i)$, to the server.
   - the server returns $\mathbf{C}_{w_i}$ to the user.
   - the user computes $\mathsf{EXT.Rep}(x, P_i)$ to reproduce the decryption key k_i, then outputs $\mathsf{SKE.Dec}(\mathbf{C}_{w_i}, k_i)$.

Fig. 1. An adaptively secure NKSPE scheme

We assume that each helper string is a l-bit string. Each $w_i \in \delta(\mathbf{D})$ is associated with at most m documents, The identifier of each document can be represented as an h-bit string. $\pi : \{0,1\}^k \times \{0,1\}^{l+m*h} \rightarrow \{0,1\}^{l+m*h}$ is a pseudorandom permutation. Let SKE=(Enc,Dec) be a PCPA-secure symmetric encryption scheme (refer to [7], Appendix A). Let EXT=(Gen,Rep) be a fuzzy extractor.

We observe that, for a queried word x, the match pattern is $\mathbf{b}$= $(b^1, \cdots, b^{|\delta(\boldsymbol{D})|})^\top$, where $b^j = \mathbf{t} \times \mathbf{A}_{w_j}$. Next, we analyze the correctness of our concrete scheme.

Theorem 1. *The* NKSPE *scheme described in Fig.1 is correct (i.e. satisfies Definition 4).*

Proof. For query string $x = x_1 \cdots x_n \in \delta'(\mathbf{D})$, let $\mathbf{x} = (1, -2x_1, \cdots, -2x_n, R_A, 1)$. For each $w_i = w_{i,1} \cdots w_{i,n} \in \delta(\mathbf{D})$, let $\mathbf{w}_i = (\sum_{k=1}^{n} w_{ik}^2 + R - R_i, w_{i1}, \cdots, w_{in}, 1, R_i)$. Thus,

$$\mathbf{x}\mathbf{Q}^{-1}\mathbf{Q}\mathbf{w}_i^\top = \mathbf{x}\mathbf{w}_i^\top = (\sum_{k=1}^{n} w_{ik}^2 + R - R_i) - 2(x_1 w_{i1} + \cdots + x_n w_{in}) + R_A + R_i$$

$$= \sum_{k=1}^{n}(w_{ik} - x_k)^2 - \sum_{k=1}^{n} x_k^2 + R_A + R$$

Therefore,

$$\sum_{k=1}^{n}(w_{ik} - x_k)^2 = \mathbf{x}\mathbf{w}_i^\top + \sum_{k=1}^{n} x_k^2 - R_A - R$$

Since $\sum_{k=1}^{n} x_k^2 - R_A - R$ is a constant, the server can use $\mathbf{x}\mathbf{w}_i^\top$ to compute the closest match and return corresponding $(dis', \pi_{K_1}(\mathbf{D}(w_i)||P_i))$. It is evident that $\mathsf{dis}(x, w_i) = dis' + \sum_{k=1}^{n} x_k^2 - R - R_A \leq threshold$ (x is a synonym of w_i). After receiving $\mathbf{D}(w_i)$, the server returns $\mathbf{C}_{w_i}$ to the user. The user then computes $\mathsf{Rep}(x, P_i)$ to reproduce the decryption key k_i. Therefore, the user can decrypt $\mathbf{C}_{w_i}$ correctly. This completes the proof.

4.1 The Proof of Security

In this subsection, we analyze the security of our concrete scheme.

Theorem 2. *The proposed scheme is adaptively secure (i.e. satisfies Definition 9) assuming that the private-key encryption scheme* SKE *is PCPA-secure and* π *is a pseudorandom permutation.*

Proof. What we need to do is to construct a simulator $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_q)$ such that for the adversary $\mathcal{A} = (\mathcal{A}_0, \cdots, \mathcal{A}_q)$, the outputs of $\mathbf{Real}_{\mathsf{NKSPE},\mathcal{A}}(k)$ and $\mathbf{Sim}_{\mathsf{NKSPE},\mathcal{A},\mathcal{S}}(k)$ are computationally indistinguishable. We construct a simulator $\mathcal{S} = (\mathcal{S}_0, \cdots, \mathcal{S}_q)$ that adaptively produces a string $\mathbf{v}' = (I', \mathbf{C}', \boldsymbol{t}') = (I', c_1', \cdots, c_N', \mathbf{t}_1', \cdots, \mathbf{t}_N')$ as follows:

1. $\mathcal{S}_0(1^k, \tau(\mathbf{D}))$: it constructs a simulated index I' by making a table comprised of entries $(\mathbf{A}_i, B_i)$, for $i = 1, \cdots, |\delta(\mathbf{D})|$, where $\mathbf{A}_i \leftarrow_r \mathbf{M}_{n+3,1}(\mathbb{F})$ and $B_i \leftarrow_r \{0,1\}^{l+m*h}$, such that for the matix $\mathbf{A}_{|\delta(\mathbf{D})|\times(n+3)} = (\mathbf{A}_1^\top, \cdots, \mathbf{A}_{|\delta(\mathbf{D})|}^\top)^\top$, $\mathsf{Rank}(\mathbf{A}) = \mathbf{min}(|\delta(\mathbf{D})|, n+3)$. $\mathcal{S}_0$ then includes I' in $st_{\mathcal{S}}$ and outputs $(I', \mathbf{C}', st_{\mathcal{S}})$, where $c_i' \leftarrow_r \{0,1\}^{|D_i|}$.
   We now claim that I' is indistinguishable from a real index, i.e. the tuples $(\mathbf{A}_i, B_i)$ are indistinguishable from tuples $(\mathbf{A}_{w_i}, B_{w_i})$. It is evident that the distributions over $\mathbf{A}_i$ and $\mathbf{A}_{w_i}$ are identical, and that B_i is indistinguishable from $B_{w_i} = \pi(\mathbf{D}(w_i)||P_i)$ since π is a pseudorandom permutation. Furthermore, since the private-key encryption scheme is PCPA-secure, each c_i' is indistinguishable from a real ciphertext.

2. $\mathcal{S}_1(st_{\mathcal{S}}, \tau(\mathbf{D}, x_1))$: it solves system of linear equations $\mathbf{Ax} = \mathbf{b}_1$. Note that it knows $\mathbf{b}_1$ from the trace of $(\mathbf{D}, x_1)$. We denote a solution of $\mathbf{Ax} = \mathbf{b}_1$ by $\mathbf{t}^*$ (if there exists solution). Let $\mathbf{t}'_1 = \mathbf{t}^{*\top}$ that is indistinguishable from a real trapdoor $\mathbf{t}_1$, since $\mathbf{t}_1 \times \mathbf{A}_{w_i} = \mathbf{t}'_1 \times \mathbf{A}_i$ holds for all $i \in [1, |\delta(\mathbf{D})|]$. $\mathcal{S}_1$ then includes $\mathbf{t}'_1$ in $st_{\mathcal{S}}$ and outputs $(\mathbf{t}'_1, st_{\mathcal{S}})$.
3. $\mathcal{S}_i(st_{\mathcal{S}}, \tau(\mathbf{D}, x_1, \cdots, x_i))$: $\mathcal{S}_i$ generates a trapdoor $\mathbf{t}'_i$ in the same way that $\mathcal{S}_1$ does, i.e. by solving the system of linear equations $\mathbf{Ax} = \mathbf{b}_i$. $\mathcal{S}_i$ then includes $\mathbf{t}'_i$ in $st_{\mathcal{S}}$ and outputs $(\mathbf{t}'_i, st_{\mathcal{S}})$. It is evident that $\mathbf{t}'_i$ is indistinguishable from a real trapdoor $\mathbf{t}_i$. This completes the proof.

Remark. Note that in our security proof if $|\delta(\mathbf{D})| \leq n + 3$, then $\mathsf{Rank}(\mathbf{A}) = \mathsf{Rank}(\mathbf{A}, \mathbf{b}_i) = |\delta(\mathbf{D})| \leq n+3$ which guarantees that $\mathbf{Ax} = \mathbf{b}_i$ must have solution. Otherwise, for a randomly selected matrix $\mathbf{A}$ in advance, it is hard to guarantee $\mathsf{Rank}(\mathbf{A}) = \mathsf{Rank}(\mathbf{A}, \mathbf{b}_i) \leq n + 3$. A solution to this problem is to divide $|\delta(\mathbf{D})|$ noisy keywords into $\lceil \frac{|\delta(\mathbf{D})|}{n+3} \rceil$ groups. There are at most $n + 3$ noisy keywords in each group. The user selects and stores different secret keys for each group.

5 Performance

5.1 Exact Efficiency of the Proposed Scheme

For each query, the number of rounds of communication is exactly 2. The user computes 1 matrix multiplication, the server performs $|\delta(\mathbf{D})|$ vector inner product operations. Regarding storage, the user must have a long-term storage for private key $K = (K_1, \mathbf{Q})$ and a random number R. The server stores an index I comprised of $|\delta(\mathbf{D})|$ entries (each entry includes a matrix of dimensions $(n + 3) \times 1$ and a $(l + m * h)$-bit string) and N encrypted documents.

5.2 Comparison

In this subsection, we highlight the differences between the fuzzy keyword search scheme in [14] and ours in Table1.

Metric. In [14], edit distance is used to measure the similarity, while Hamming distance is considered when the input can be represented as a binary string in our concrete scheme.

Security. The scheme in [14] was proved secure against IND2-CKA, however, it is not adaptively secure. The reason is that for a query (w, k), the simulator has no idea how many simulated trapdoors it should compute even if given the trace of $(\mathbf{D}, (w, k))$[4]. While our scheme is proven secure against adaptive adversary.

Efficiency. The scheme in [14] requires two rounds of communication for each query. Assume that the length of all keywords is polynomial in n. For a query (w, k), the user computes the trapdoor set of size $O(n^k)$, the server compares this trapdoor set with the index table. Regarding the server's storage, except for N encrypted documents, the size of index is $O(Mn^k)$, where M is the number of distinct keywords in document set and k is the edit distance.

[4] To search with (w, k), the user computes a trapdoor set T of size $O(n^k)$, where w is a fuzzy query with edit distance k and n is the length of w.

Table 1. Performance (per query) and properties comparison between the scheme in [14] and ours

Performance and Properties	[14]	ours
metric	edit distance	Hamming distance
security	IND2-CKA	adaptively secure
server computation	$O(Mn^{2k})$	$O(M)$
server's extra storage (index)	$O(Mn^{k})$	$O(M)$
user computation	$O(n^{k})$	$O(1)$
user storage	$O(1)$	$O(1)$
number of rounds	2	2

6 Conclusion and Future Work

In this paper, we presented a general framework for noisy-keyword-based searchable private-key encryption in a fault-tolerant manner. Under this framework, we proposed a concrete scheme which is proved adaptively secure according to the simulation-based security definition presented in [7]. The keyword search finishes in two rounds of interaction bewteen the user and the server, and requires an amount of work from the server that is linear in the number of noisy keywords. Finally, we compared the scheme in [14] and ours both in properties and efficiency.

In our model, only the owner of the document set can search on encrypted documents by noisy keywords. The next problem we will address is a natural extension where a group of authorized parties other than the owner can submit search queries. On the other hand, we will consider other notions of distance such as set difference, edit distance and so on in concrete NKSPE schemes.

References

1. Bellare, M., Boldyreva, A., ONeill, A.: Efficiently-searchable and deterministic asymmetric encryption. Tech. rep., Citeseer (2006)
2. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
3. Boneh, D., Kushilevitz, E., Ostrovsky, R., Skeith, W.: Public Key Encryption That Allows PIR Queries. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 50–67. Springer, Heidelberg (2007)
4. Boneh, D., Waters, B.: Conjunctive, Subset, and Range Queries on Encrypted Data. In: Vadhan, S. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
5. Bringer, J., Chabanne, H., Kindarji, B.: Error-tolerant searchable encryption. In: IEEE International Conference on Communications, ICC 2009, pp. 1–6. IEEE (2009)
6. Chang, Y., Mitzenmacher, M.: Privacy Preserving Keyword Searches on Remote Encrypted Data. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005)

7. Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: Improved definitions and efficient constructions. Journal of Computer Security 19(5), 895–934 (2011)
8. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
9. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors. Security with Noisy Data, 79–99 (2007)
10. Du, W., Atallah, M.: Protocols for secure remote database access with approximate matching. In: E-Commerce Security and Privacy, pp. 87–111. Springer (2001)
11. Goh, E.: Secure indexes. An early version of this paper first appeared on the Cryptology ePrint Archive, pp. 1–18 (October 7, 2003)
12. Goldreich, O., Ostrovsky, R.: Software protection and simulation on oblivious rams. Journal of the ACM (JACM) 43(3), 431–473 (1996)
13. Kevenaar, T.: Protection of biometric information. Security with Noisy Data, 169–193 (2007)
14. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Fuzzy keyword search over encrypted data in cloud computing. In: 2010 Proceedings IEEE INFOCOM, pp. 1–5. IEEE (2010)
15. Ma, S., Yang, B., Li, K., Xia, F.: A Privacy-Preserving Join on Outsourced Database. In: Lai, X., Zhou, J., Li, H. (eds.) ISC 2011. LNCS, vol. 7001, pp. 278–292. Springer, Heidelberg (2011)
16. Monrose, F., Reiter, M., Li, Q., Wetzel, S.: Cryptographic key generation from voice. In: 2001 IEEE Symposium on Security and Privacy, S&P 2001. Proceedings, pp. 202–213. IEEE (2001)
17. Park, D., Kim, K., Lee, P.: Public Key Encryption with Conjunctive Field Keyword Search. In: Lim, C., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 73–86. Springer, Heidelberg (2005)
18. Pinkas, B., Reinman, T.: Oblivious RAM Revisited. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 502–519. Springer, Heidelberg (2010)
19. Shi, E., Bethencourt, J., Chan, T., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 350–364. IEEE (2007)
20. Song, D., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceedings of IEEE Symposium on Security and Privacy, S&P 2000, pp. 44–55. IEEE (2000)
21. Van Liesdonk, P., Sedghi, S., Doumen, J., Hartel, P., Jonker, W.: Computationally Efficient Searchable Symmetric Encryption. In: Jonker, W., Petkovic, M. (eds.) SDM 2010. LNCS, vol. 6358, pp. 87–100. Springer, Heidelberg (2010)