Malaysian Journal of Mathematical Sciences 13(S) August: 31–47 (2019)
Special Issue: The 6th International Cryptology and Information Security Conference (CRYPTOLOGY2018)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Searchable Symmetric Encryption Security Definitions

Mohamad, M.S. *[1,2], Tan, S.Y.[3], and Chin, J.J.[4]

[1]*Faculty of Computing and Informatics, Multimedia University, Cyberjaya*
[2]*Information Security Lab, Mimos Berhad, Kuala Lumpur*
[3]*Faculty of Information Science and Technology, Multimedia University, Melaka*
[4]*Faculty of Engineering, Multimedia University, Cyberjaya*

E-mail: soeheila.mohamad@mimos.my
* *Corresponding author*

ABSTRACT

After many searchable symmetric schemes have been proposed and proven secure, a few published work show empirical evidence of successful attacks on all published schemes. The attacks indicate a gap between the current security models and the practical attackers. This work reviews indistinguishability and semantic security games for SSE. Finally, a new security definition against the practical attacks is proposed and proven to imply current security definitions.

Keywords: searchable encryption, SSE, security model.

1. Introduction

Searchable symmetric encryption (SSE) is a category of schemes with ciphertext searching function utilising symmetric cryptographic algorithms. The purpose of SSE is for a user to store a document collection in a storage facility without exposing the documents to the storage owner or co-resident. In addition, the data owner or authorized users may perform searches to find documents of interest, while maintaining security.

In order to achieve sublinear search complexity, SSE scheme designs deploys inverted index where the index key is encoded keywords and the index values are document identifiers. Examples of such design includes Curtmola et al. (2006), Chase and Kamara (2010), Naveed et al. (2014), Cash et al. (2014) and Kamara and Moataz (2017).

The trade off in enabling search on ciphertexts is the disclosure of some information regarding the documents, called leakage. Despite the leakage, an SSE scheme aims to protect the confidentiality of the stored documents and the queried keyword using symmetric cryptographic schemes. Currently, the $\mathcal{L}$-security (Chase and Kamara, 2010) is the definition accepted for SSE schemes security.

However, attacks by Islam et al. (2012), Zhang et al. (2016) and Cash et al. (2015) were successful in recovering query keywords in published index-based SSE schemes. Wright and Pouloit (2017) generalized the attacks as statistic inference attack and produced a statistical method framework to detect such vulnerabilities in an SSE scheme. Cash et al. (2015) studied the practical attacks and defined the attack goals and adversary capabilities. In addition, they categorize SSE leakage profiles to identify the extent of vulnerability of an SSE design to the different attacks. Here we are only concerned with leakage profile L1 which reveals keyword occurrence pattern only after search queries are performed.

Our Contribution This work takes the provable security perspective by comparing the indistinguishability and semantic security definitions for SSE to the attacks. Then, a security game is proposed to define strength against the distribution-based query recovery attacks.

2. Preliminaries

2.1 Searchable Symmetric Encryption Scheme

There are two types of SSE, static and dynamic. A static SSE is where the data is prepared and uploaded to the storage server once and after that only search queries are made. A dynamic SSE allows for data adding, removing or modified after the first uploading. A static SSE scheme consists of six algorithms.

KeyGen This is a probabilistic algorithm run by the client. From a security parameter 1^k, this algorithm generates a set of symmetric keys, K, including an encryption key.

BuildIndex This algorithm is run by the client, taking the document-keywords mapping DB and keys K and output the index I.

Encrypt This a symmetric encryption algorithm is run by the client and is usually probabilistic. For input a set of document $\mathbf{D}$ and keys K, this algorithm outputs a set of ciphertexts $\mathbf{c}$ of the documents.

Trapdoor This algorithm is run by the client and is usually deterministic. It takes as input the keyword w and the key K and outputs a trapdoor t_w.

Search This is a deterministic interactive algorithm run by the server. Inputs are t_w sent by the client and the index I stored on the server. This algorithm finds the set of document identifiers corresponding to documents containing the keyword w. The set of document identifiers being output is returned to the client.

Decrypt This the corresponding symmetric decryption algorithm which runs on the client. Taking input ciphertexts $c_1, \ldots, c_n$ and key K to output documents $d_1, \ldots, d_n$.

There are schemes presented with four algorithms KeyGen, Setup, Search and Decrypt where Setup consists of BuildIndex, Trapdoor and Encrypt, and Search is interactive and includes Trapdoor and Search. Dynamic SSE schemes include another algorithm, Update which takes as input a document, the list of keywords and an operation name such as add, remove and modify. The algorithm outputs a new index and ciphertext.

Let $\mathbf{D}(w)$ denote the set of documents containing keyword w and $\mathsf{id}(d)$ denote the identifier for document $d \in \mathbf{D}$. An SSE scheme is correct if the symmetric encryption scheme deployed is correct and for all keywords w,

$$\mathsf{Search}(\mathsf{Trapdoor}_K(w), I) = \{\mathsf{id}(d) | d \in \mathbf{D}(w)\}.$$

2.2 SSE Scheme Leakage

An SSE scheme leakage is the information revealed to the storage server by the data submitted by the client. The leakage is defined as a function of *history* which is the documents $\mathbf{D}$, keyword-document mapping DB and the sequence of keywords for search queries $w_1, w_2, \ldots, w_q$. The leakage function varies from scheme to scheme. The leakage function is denoted as

$$\mathcal{L}(\mathbf{D}, \mathsf{DB}, w_1, w_2, \ldots, w_q) = (\mathcal{L}^{setup}(\mathbf{D}, \mathsf{DB}), \mathcal{L}^{query}(w_1, w_2, \ldots, w_q)).$$

Setup leakage $\mathcal{L}^{setup}(\mathbf{D}, \mathsf{DB})$ results from the index $I \leftarrow \mathsf{BuildIndex}(\mathbf{D}, \mathsf{DB})$ and ciphertexts $\mathbf{c} \leftarrow \mathsf{Encrypt}_K(\mathbf{D})$ where $K \leftarrow \mathsf{KeyGen}(1^k)$. Clearly, $\mathcal{L}^{setup}$ contains at least the number of ciphertexts $|\mathbf{D}|$ and the length of ciphertexts. However, Naveed et al. (2014) SSE design hides the document lengths until the documents occurs in search results. Depending on the design of index, I may reveal the number of keywords of the particular document set. For example, the index in (Chase and Kamara, 2010) gives the number of keywords but in (Cash et al., 2014) the number is hidden.

The query leakage increases as more search queries are made. The leakage up to the q-th query $\mathcal{L}^{query}(w_1, w_2, \ldots, w_q)$, is leakage from the keyword trapdoors $t_i \leftarrow \mathsf{Trapdoor}(w_i)$ and the set of document identifiers in the search results $\{\mathsf{id}(d) | d \in \mathbf{D}(w_i)\} \leftarrow \mathsf{Search}(I, t_i)$. $\mathcal{L}^{query}$ contains at least the Access Pattern(AP) from the search results and the Query Pattern(QP) from the trapdoors. SP is a record of repeated (and unrepeated) queries in the query sequence in the history. AP is a record of the document identifiers and the ciphertexts returned in each of the queries in the history. Clearly, AP reveals the number of documents associated to each of the trapdoor too, which incrementally provide the server with the keyword distribution information of $\mathbf{D}$. Intersection pattern (IP) which indicates which documents contains two or more keywords can be extracted from AP, and IP reveals keyword co-occurrence distribution. From AP one can also extract the number of trapdoors associated to each documents.

For dynamic SSE schemes there is update leakage, $\mathcal{L}^{update}$. Since this work focuses on static SSE, we do not discuss update leakage further. The leakage functions are usually written without the inputs because it is always the same as defined in this section.

2.3 Adversary Model

Cash et al. (2015) defined the attack goals and adversary capabilities as shown in

Table 1. This model is used to study SSE security definitions in the next section.

Attack Mode	**Adversary Knowledge**	**Attack Goals**
Passive[1] Chosen query attack Chosen document attack	Query distribution Known queries Document distribution Known document	Query recovery Plaintext recovery[2]

Table 1: Adversary model proposed by Cash et al. (2015). [1]Passive adversary includes honest-but-curious server. [2]The plaintext recovery goal includes partial plaintext recovery.

In the definition of the security games, there are non-adaptive and adaptive adversaries. In SSE, these adversaries differ at the stage of choosing keywords for trapdoor queries. For a non-adaptive adversary, after obtaining the index and ciphertexts from the challenger, the adversary has to generate the keyword sequence and submit it altogether. On the other hand, an adaptive adversary submits one keyword at a time, and gets a reply before the next keyword is queried. As such the adaptive adversary may use the information gained from a trapdoor to choose the next keyword with a strategy to gain as much information as it can.

Besides the adversary model, (Cash et al., 2015) presents categories of SSE leakage profiles to identify the extent of vulnerability of an SSE design to the different attacks.

L4 Full plaintext under deterministic word-substitution cipher

L3 Fully revealed occurrence pattern with keyword order

L2 Fully revealed occurrence pattern

L1 Query revealed occurrence pattern

Category L3 and L2 are SSE schemes in which the index reveals the keyword distribution. Such designs has an index with entries being a list of document identifiers without any padding or unencrypted list. Most published SSE schemes fall under L1. The leakage abuse attacks are targeted towards the L1 schemes.

3. Security Definitions Review

For encryption schemes the preferred security is indistinguishability (IND) and semantic security. Here we review the games which define IND and semantic security for SSE schemes.

3.1 Indistinguishability

The IND game for encryption scheme involves the adversary providing two plaintexts of the same length and then when given the encryption of one of them, having to identify the corresponding plaintext. This definition implies that information regarding the plaintext is fully hidden in the ciphertext, except for its length.

The first security definition for SSE is by Goh (2003). In that work, every document ciphertext is appended with the list of encoded keyword contained in the documents. For this design the indistinguishability game is where the adversary chooses two list of keywords associated to some documents and then is given as challenge the encoded list of one of them. The adversary is asked to identify which keyword list has been encoded to be the challenge. By this, the SSE is secure when the encoded keyword list does not reveal which document it belongs to, up to the subset of documents having the same number of keywords.

Curtmola et al. (2006) adopted the game by Goh (2003) for index-based SSE design. That is for all documents in the storage, there is one index whose key is the encoded keyword and the entry lists identifiers of documents containing the keyword. The adversary is given the power to choose two sets of documents with the condition their history (include search sequence) produce equal leakage. Then, the adversary is given the corresponding two sets of index and ciphertexts and allowed to make trapdoor queries. However, instead of two keyword lists, the adversary chooses two keywords, one from each document set. When given the encoded keyword challenge, the adversary have to guess which document set the keyword belong to. An SSE scheme proven secure by rendering the adversary's advantage in this game insignificant, means the scheme produces trapdoors, index and ciphertexts which do not reveal any content of the documents.

The IND game for SSE as defined by (Curtmola et al., 2006) is presented here.

Definition 3.1. *Let Σ=(*KeyGen, Encrypt, BuildIndex, Trapdoor, Search, Decrypt*) be an index-based SSE, $k \in \mathbb{N}$ be a security parameter and $\mathcal{A}$=$(\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{q+1})$ be such that $q \in \mathbb{N}$. Consider the following probabilistic experiment $\textbf{\textit{Ind}}_{\mathcal{A},\Sigma}(k)$: The challenger begins with generating $K \leftarrow$*KeyGen(1^k) *and then randomly selecting $b \leftarrow \{0,1\}$. The adversary $\mathcal{A}_0$ generates two sets of documents $\mathbf{D}_0, \mathbf{D}_1$ with the restriction that $\mathcal{L}^{setup}(\mathbf{D}_0)$=$\mathcal{L}^{setup}(\mathbf{D}_1)$. The challenger returns $\mathbf{c}_b \leftarrow$*Encrypt$_K(\mathbf{D}_b)$ *and $I_b \leftarrow$*BuildIndex$_K(\mathbf{D}_b)$. *Next, for $i = 1, \ldots, q$the adversary $\mathcal{A}_i$ chooses a keyword $w_{0,i}$ from $\mathbf{D}_0$ and $w_{1,i}$ from $\mathbf{D}_1$ such that $\mathcal{L}^{query}(w_{0,1}, \ldots, w_{0,q})$=$\mathcal{L}^{query}(w_{1,1}, \ldots, w_{1,q})$. For every query $\mathcal{A}_i$ waits for the reply, $t_{b,i} \leftarrow$*Trapdoor$_K(w_{b,i})$ *before making the next query. Finally, $\mathcal{A}_{q+1}$ outputs b'. If $b' = b$, the experiment outputs 1, otherwise outputs 0. We say the that Σ achieves adaptive indistinguishability if for all polynomial size adversaries $\mathcal{A}$=$(\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{q+1})$ such that q is polynomial in terms of k,*

$$\Pr\left[\mathbf{Ind}_{\Sigma,\mathcal{A}}(k) = 1\right] \leqslant \frac{1}{2} + \mathsf{negl}(k)$$

where the probability is over the choice of b and the coins of KeyGen *and* Encrypt.

3.2 Semantic Security

Semantic security of an encryption scheme carries the meaning that no adversary can recover partial information about the plaintexts from the ciphertexts. In the context of SSE, semantic security means the same with the exception of the declared leakage.

The first simulation based security definition is by Chang and Mitzenmacher (2005). In their game, the adversary $\mathcal{A}$ is given the actual communications between server and user, C_q, while another adversary $\mathcal{A}^*$ is given information learned (leakage) from the communication, C_q^*. The defining statement

$$\left|\mathbf{Pr}\left[\mathcal{A}\left(C_q, 1^k\right) = f\left(H_q\right)\right] - \mathbf{Pr}\left[\mathcal{A}^*\left(C_q^*, 1^k\right) = f\left(H_q\right)\right]\right| \leqslant \mathsf{negl}(k).$$

expresses that the adversaries $\mathcal{A}$ and $\mathcal{A}^*$ has the same amount of information regarding the documents and queried keywords. Effectively, this semantic security definition implies that the communications C_q leaks exactly the defined leakage C_q^* which contain only the relation of keyword trapdoor to document identifiers.

In this definition the history (document set and query sequence) is 'given' in the environment. This means the adversaries $\mathcal{A}$ and $\mathcal{A}^*$ are passive. The leakage C_q^* includes the ciphertexts and the access pattern.

The game is put into the Real-Ideal game framework by Curtmola et al. (2006). In the Real environment, the challenger runs the SSE scheme to reply the adversary's queries, while in the Ideal environment a simulator creates the replies based on the given leakages. The adversary's task is to identify whether it is playing in a Real environment or in the Ideal environment. To proof the security of an SSE scheme, one has to describe a simulator which will produce ciphertexts, index and trapdoors that the adversary cannot distinguish from the SSE scheme outputs.

Besides that change, the adversary is given the capability to choose the documents and keywords for the search query sequence. Also, the leakages now includes more items, namely access pattern and query patterns.

A major change on the leakage function was proposed by Chase and Kamara (2010). The function $\mathcal{L}$ is defined as a tuple of Setup and Search leakage functions, $(\mathcal{L}^{setup}, \mathcal{L}^{query})$. For dynamic SSE, another element in the tuple is the Update leakage, $\mathcal{L}^{update}$. The leakage function is part of the SSE scheme specification and is declared explicitly in the security statement. This is in accord to the meaning of semantic security of SSE: no partial information regarding the documents and keywords can be recovered except for the declared leakage. Following this the semantic security of SSE is renamed as $\mathcal{L}$-security.

Prior to $\mathcal{L}$-security definition, SSE leakage is set at the SSE security definition. Now, every SSE scheme is allowed to define its leakage. Nevertheless, previous defined leakage is the minimum leakage for any SSE scheme. Some schemes leak more to accommodate their design or data type. In fact this was proposed in (Chase and Kamara, 2010) because they are applying searchable encryption on structured data such such as matrix and graphs. In their SSE constructions some structure information is leaked to complete the search result.

The widely used SSE security definition is the $\mathcal{L}$-security game as presented below.

Definition 3.2 ($\mathcal{L}$-security CKA). *Let* $\Sigma=\big(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{BuildIndex}, \mathsf{Trapdoor}, \mathsf{Search}, \mathsf{Decrypt}\big)$ *be an SSE scheme. Consider the following probabilistic experiments where $\mathcal{A}$ is an adversary, $\mathcal{S}$ is a simulator and $\mathcal{L} = \big(\mathcal{L}^{setup}, \mathcal{L}^{query}\big)$ is a tuple of stateful leakage algorithms:*

$\mathbf{Real}_{\Sigma,\mathcal{A}}(k)$ *: the challenger begins by running $\mathsf{KeyGen}(1^k)$ to generate a key K. $\mathcal{A}$ outputs a document set and a document-keyword mapping ($\boldsymbol{M}$,DB) and receives $(\gamma, \mathbf{c})$ from the challenger where $\gamma \leftarrow \mathsf{BuildIndex}_K(DB)$ and $\mathbf{c} \leftarrow \mathsf{Encrypt}_K(\mathbf{M})$. The adversary makes a polynomial number of adaptive queries and, for each query q, receives a trapdoor $t \leftarrow \mathsf{Trapdoor}_K(q)$ from the challenger. Finally $\mathcal{A}$ returns a bit b that is output by the experiment.*

$\mathbf{Ideal}_{\Sigma,\mathcal{A},\mathcal{S}}(k)$ *: $\mathcal{A}$ outputs a document set and a document-keyword mapping ($\boldsymbol{M}$,DB). Given $\mathcal{L}^{setup}(\mathbf{M}, \mathrm{DB})$, $\mathcal{S}$ generates and sends a pair $(\gamma, \mathbf{c})$ to $\mathcal{A}$. The adversary makes a polynomial number of adaptive queries and for each query q the simulator $\mathcal{S}$ is given $\mathcal{L}^{query}(DB, q)$. Then $\mathcal{S}$ returns a trapdoor t to $\mathcal{A}$. Finally, $\mathcal{A}$ returns a bit b that is output by the experiment.*

We say that Σ is $\mathcal{L}$-secure against adaptive chosen keyword attacks(CKA2) if for all PPT adversaries $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that

$$|\Pr\left[\mathbf{Real}_{\Sigma,\mathcal{A}}(k) = 1\right] - \Pr\left[\mathbf{Ideal}_{\Sigma,\mathcal{A},\mathcal{S}}(k) = 1\right]| \leqslant \mathsf{negl}(k).$$

3.3 Summary

Relation between IND and semantic security has been proven by Curtmola et al. (2006). Refer to their work for the proofs.

Theorem 3.1. *(Curtmola et al., 2006)[Theorem 4.9] Non-adaptive indistinguishability security of SSE is equivalent to non-adaptive semantic security of SSE.*

Theorem 3.2. *(Curtmola et al., 2006)[Theorem 4.12] Adaptive semantic security of SSE implies adaptive indistinguishability of SSE.*

Since $\mathcal{L}$-security in Definition 3.2 is the most accurate semantic security definition, the theorems implies it is the strongest security definition for SSE. Consequently, published SSE schemes opt for the adaptive $\mathcal{L}$-security as the definition for its scheme security.

Security Definition	**Security Goal**	**Attack Mode**	**Leakage Profile**
Goh (2003)	IND of documents w.r.t. index	Chosen keyword & document	L4
Chang and Mitzenmacher (2005)	Semantic security of documents and queries	Passive	L2
Curtmola et al. (2006) semantic security	Semantic security of document set	Chosen keyword & document	L1
Curtmola et al. (2006) indistinguishability	IND of document sets	Chosen keyword & document	L1
Chase and Kamara (2010) $\mathcal{L}$-security	Semantic security of document set	Chosen query & document	L1

Table 2: Comparison of security definitions and the best leakage profile which achieves it according to Cash et al. (2015) model as in Table 2.3. In all of the definitions, the adversary has no prior knowledge about the document set.

3.4 Practical Attacks

The practical attacks aims for query recovery and exploit keyword distribution revealed by AP. They are called Leakage Abuse Attacks (LAA).

Islam et al. (2012) creates the first practical attack and is known as the IKK attack. In this attack, the attacker has the whole document set and some keyword-trapdoor pairs. From the document set, the attacker computed the keyword co-occurrence matrix. Then, by observing a number of search queries, the observed co-occurrence, inferred from AP in $\mathcal{L}^{\text{query}}$ and the estimated co-occurrence are put through simulated annealing to guess the keyword corresponding to the trapdoor. Then, Cash et al. (2015) introduced count attack which identifies unique keyword frequency in the known document and finds a count match in AP in $\mathcal{L}^{\text{query}}$, before applying the IKK attack.

File injection attack was created by Zhang et al. (2016). In this attack, the attacker insert documents into the SSE to make its keyword distribution estimation more accurate. The attacker create documents containing intersecting subsets of keywords. If the inserted document appears in a search result disclosed by AP in $\mathcal{L}^{\text{query}}$, the attacker can infer the keyword of the search trapdoor. Less injected files are required in the attack if the attacker has partial knowledge of the stored documents.

3.5 Comparing Security Model to Attacks

Table 3: Comparing security/attack goals and adversary capabilities of $\mathcal{L}$-security to count attack and file injection attack.

		$\mathcal{L}$-security	**Count Attack**	**File Injection Attack**
Security& Attack Goals	Documents	Semantic security	Know document keywords	
	Queries	Semantic security	Query keyword recovery	
Adversary Capability	Documents	Chosen document	Known document	Chosen document
	Queries	Adaptively chosen query	Observed queries	

By comparing the $\mathcal{L}$-security game to the attacks, as in Table 3, the attackers are at most as powerful as the adversary in the game. However, the attackers are able to break the semantic security of both the documents and the search queries. One gap immediately identified is the usage of the complete knowledge about the keyword distribution from the generated document.

We illustrate the gap by an example attack where the adversary can perform a perfect count attack: an adversary can perform 100% query recovery with probability 1. When allowed to choose the documents in the game, the adversary can generate a document set such that each keyword has unique number of documents containing it. As such, by the number of document identifiers in $\mathcal{L}^{\text{query}}$, the attacker can immediately identify the correct keyword.

However, in the current Real-Ideal game the ability to choose the document set and keywords is useless. The power is also useless for the **Ind** game in (Curtmola et al., 2006) because the choice of two distinct histories is conditioned to produce the same leakage.

Therefore, the gap between the security definitions and the attacks is query security. The Real-Ideal security game do not signify the vulnerability due to the stored documents keyword distribution revealed in $\mathcal{L}^{\text{query}}$. Although SSE is initiated with the allowance for access pattern to be revealed, the practical attacks now have shown that the disclosure of access pattern is a threat to semantic security of queries and documents under chosen document and known distribution attacks.

3.6 Update in Security Definitions

The main gap in adversary model and the attackers is the ability to choose the document. The attack has shown this is a very powerful adversary. Hence the first changes noticed in the games is the source of the document set. The document set is given in the environment in (Kamara and Moataz, 2017) and (Pouloit et al., 2017).

A more formal change made by Bost and Fouque (2017) defines a structure called Constraint to declare the adversary's knowledge. This is deployed in the IND game as in Definition 3.1. The documents sets have to fulfill the Constraint as set by the adversary.

Despite the changes, the security games maintain the status where the leakage is allowed and does not identify whether the scheme secure against LAA. In the next section we define a game which if a scheme can be proven to allow only negligible advantage for the adversary to win implies that the scheme is secure against query recovery under LAA.

4. New Security Definition

Here, a security game is defined to discriminate SSE schemes whose leakage enable query recovery attacks. The strength against query recovery is defined by providing assurance that even knowing the set of all keywords and access to the trapdoor oracle, the adversary would not be able to identify keywords of other trapdoors.

The adversary's prior knowledge determines the source of documents during the game initiation as follows.

- Without prior knowledge: Document set $\mathbf{D}$ is in the environment.
- Known document distribution: $\mathbf{D}$ is set by environment. Challenger gives adversary distribution information.
- Known document: $\mathbf{D}$ is set by environment. Challenger gives adversary all or some document.

Definition 4.1 (Query indistinguishability(Q-IND)). *Let* SSE *be an index-based SSE scheme consisting of* (KeyGen, $\mathsf{Encrypt}$, $\mathsf{Trapdoor}$, Search, $\mathsf{Decrypt}$)*,* $k \in \mathbb{N}$ *be the security parameter and* $\mathcal{A}$ *be an adversary.*

Initiation *The document set* $\mathbf{D}$ *is generated according to the adversary knowledge above. The challenger* $\mathcal{C}$ *generate the secret keys* $\mathbf{K}=\mathsf{KeyGen}(1^k)$ *and index* I *on the document set* $\mathbf{D}$*.* $\mathcal{C}$ *sends* I*, ciphertexts* $\mathbf{c}$*, set of all keywords* $\mathbf{W}$ *to* $\mathcal{A}$*.*

Queries $\mathcal{A}$ *is allowed to make adaptive trapdoor queries by keyword* $w_i \in \mathbf{W}$ *to obtain* $t_i=\mathsf{Trapdoor}(K,w_i)$*.*

Challenge *Next, $\mathcal{A}$ chooses two keywords $w_0, w_1 \in \mathbf{W}$ which has not been queried and submit to $\mathcal{C}$. $\mathcal{C}$ randomly choose $b \in \{0,1\}$ and give $\mathcal{A}$ the corresponding trapdoor t_b=$\mathsf{Trapdoor}(K,w_b)$. After the challenge is issued, $\mathcal{A}$ can make more trapdoor queries except for w_0, w_1.*

Response *Finally, $\mathcal{A}$ outputs b' as a guess of b. The adversary $\mathcal{A}$ wins if $b' = b$.*

The advantage of $\mathcal{A}$ is defined as the probability of winning this game beyond guessing, $\mathrm{Adv}_{\mathcal{A}}(k) = \left|\Pr[b = b'] - \frac{1}{2}\right|$ where the probability is over $\mathcal{A}$ and $\mathcal{C}$'s coin tosses. An SSE *scheme is said to achieve query indistinguishability if for any $\mathbf{D}$, $\mathrm{Adv}_{\mathcal{A}}(k) \leqslant \mathsf{negl}(k)$.*

The schemes on which the count attack applies, would not achieve Q-IND because the distinguisher can use the attack to identify b correctly. On the other hand, schemes with less leakage, especially those which obfuscate the keyword distribution, will be able to achieve this.

Since Curtmola et al. (2006) has proven that IND implies semantic security under adaptive attacks, the soundness of the new security definition is demonstrated by proving that Q-IND implies IND and $\mathcal{L}$-security.

Theorem 4.1. *Adaptive Q-IND under chosen document and keyword attack implies* ***Ind*** *under adaptive chosen document and keyword attack as in Definition 3.1.*

Proof. Assume there exists an adversary $\mathcal{A}$ who has non-negligible advantage in the **Ind** game as defined by Curtmola et al. (2006). We show that there exists an adversary $\mathcal{B}$ who has non-negligible advantage in guessing b correctly in the Q-IND game. Consider the adversary $\mathcal{B}$ who works as below.

Setup

1. $\mathcal{B}$ initiates **Ind** game and receives $\mathbf{D}_0$ and $\mathbf{D}_1$ from $\mathcal{A}$.
2. $\mathcal{B}$ submits $\mathbf{D}$=$\mathbf{D}_0 \cup \mathbf{D}_1$ to the challenger $\mathcal{C}$ who returns $\mathbf{W}$, I and $\mathbf{c}$.
3. $\mathcal{B}$ create ciphertext set $\mathbf{c}'$ by including in $\mathbf{c}'$ exactly half of every set of equal length ciphertexts in $\mathbf{c}$. Next, index I' is created by generating a random entry such that $|I'[t_i]| = \frac{1}{2}|I[t_i]|$ for every key t_i of I.
4. $\mathcal{B}$ gives $\mathcal{A}$ I' and $\mathbf{c}'$.

Queries: For $i = 1$ to $q - 1$,

1. $\mathcal{A}$ submits $(w_{0,i}, w_{1,i})$ to $\mathcal{B}$.
2. $\mathcal{B}$ passes $w_{0,i}$ to $\mathcal{C}$ and obtains trapdoor t_i.
3. $\mathcal{B}$ gives t_i to $\mathcal{A}$.

Challenge: When $\mathcal{A}$ submits $(w_{0,q}, w_{1,q})$, $\mathcal{B}$ forwards $(w_{0,q}, w_{1,q})$ to $\mathcal{C}$. $\mathcal{C}$ returns t_b=$\mathsf{Trapdoor}(K, w_{b,q})$ where $b \xleftarrow{R} \{0,1\}$ as the challenge for $\mathcal{B}$.

Response: $\mathcal{B}$ passes t_b to $\mathcal{A}$ and obtain a reply b'. If $b' = 0$ then $\mathcal{B}$ submits 0 to $\mathcal{C}$, otherwise submits 1.

First, we argue that I' and $\mathbf{c}'$ is indistinguishable from the index and ciphertexts for $\mathbf{D}_b$ to $\mathcal{A}$. Since $\tau(\mathbf{D}_0)$=$\tau(\mathbf{D}_1)$, the trace of the index and ciphertexts for $\mathcal{A}$ are exactly one half of the trace $\tau(\mathbf{D})$.

Denote the ciphertext size $|d_{i,j}|$ as $\ell_{i,j}$. Then $\tau(\mathbf{D}_0)$=$(\ell_{0,1}, \ell_{0,2}, \ldots, \ell_{0,n})$ and $\tau(\mathbf{D}_1)$=$(\ell_{1,1}, \ell_{1,2}, \ldots, \ell_{1,n})$. Since $\tau(\mathbf{D}_0)$=$\tau(\mathbf{D}_1)$, $\ell_{0,j} = \ell_{1,j}$ for all $j = 1, \ldots, n$, let $\ell_j = \ell_{0,j} = \ell_{1,j}$, and hence $\tau(\mathbf{D})$=$(\ell_1, \ell_1, \ell_2, \ell_2, \ldots, \ell_n, \ell_n)$ because $\mathbf{D}$=$\mathbf{D}_0 \cup \mathbf{D}_1$. The constructed $\mathbf{c}'$ consists of one ciphertext for each ℓ_j, and hence indistinguishable from the ciphertext sets for either $\mathbf{D}_0$ or $\mathbf{D}_1$ because it produce the same trace as $\tau(\mathbf{D}_0)$ or $\tau(\mathbf{D}_1)$. Similar argument applies to the indistinguishability of I' from I_{D_0} and I_{D_1}.

At the query stage, from $\mathcal{A}$'s perspective, it is playing the **Ind** game when its challenger chooses $b = 0$. The keywords $w_{0,i}$ passed to $\mathcal{C}$ is input to the $\mathsf{Trapdoor}$ algorithm and hence t_i returned to $\mathcal{A}$ is exactly what it will receive in the **Ind** game because it is produced using the correct key. By the construction of I', $I'[t_i]$ will produce $\tau(w_i) = (\alpha(w_i), \sigma(w_i))$ such that $|\alpha_D(w_i)| = \frac{1}{2}|\alpha_D(w_i)|$ because $\tau(w_{0,i})$=$\tau(w_{1,i})$. For the same reason, $\sigma(w_{0,i})$=$\sigma(w_{1,i})$.

Secondly, we show that the probability of $\mathbf{B}$'s response to $\mathcal{C}$ is correct with non-negligible probability. If the challenge given to $\mathcal{B}$ is $\mathsf{Trapdoor}(w_{0,q})$ then in $\mathcal{A}$'s perspective it has been receiving $(t_{0,1}, t_{0,1}, \ldots, t_{0,q-1}, t_{0,q})$ and hence replies $b' = 0$ with probability $\frac{1}{2} + \mathrm{Adv}_{\mathcal{A}}(k)$. On the other hand, if the challenge for $\mathcal{B}$ is $\mathsf{Trapdoor}(w_{0,q})$ then in $\mathcal{A}$'s perspective it has received $(t_{0,1}, t_{0,1}, \ldots, t_{0,q-1}, t_{1,q})$ which is not consistent with the **Ind** challenger choosing $b = 0$ or $b = 1$. We assume here that $\mathcal{A}$ will make a random guess. Therefore, we have that

$$\begin{aligned}
\Pr[\mathcal{B} \text{ wins}] &= \Pr[b=0]\cdot\Pr[0 \leftarrow \mathcal{B} | b=0] + \Pr[b=1]\cdot\Pr[1 \leftarrow \mathcal{B} | b=1] \\
&= \frac{1}{2}\Pr[b'=0|b=0] + \frac{1}{2}\Pr[b'=1|b=1] \\
&= \frac{1}{2}\left(1 + \mathrm{Adv}_{\mathcal{A}}(k)\right) \\
&= \frac{1}{2} + \frac{1}{2}\mathrm{Adv}_{\mathcal{A}}(k)
\end{aligned}$$

which means $\mathrm{Adv}_{\mathcal{B}}(k)$=$\frac{1}{2}\mathrm{Adv}_{\mathcal{A}}(k)$ and hence non-negligible.

Therefore, if an adversary which can distinguish between document sets exists, then an adversary who can distinguish queries exists. In conclusion, if an SSE scheme achieves adaptive query indistinguishability, then it also achieves adaptive (document set) indistinguishability (**Ind**). □

Theorem 4.2. *Adaptive query indistinguishability under chosen document and keyword attack implies adaptive $\mathcal{L}$-security under chosen keyword attack as in Definition 3.2.*

Proof. Assume for any polynomial-sized simulator there exists an adversary $\mathcal{A}$ and a distinguisher $\mathcal{D}$ such that after $\mathcal{A}$ makes q adaptive trapdoor queries, $\mathcal{D}$ can distinguish the **Real** environment from **Ideal** with non-negligible advantage. We show that there exists an adversary $\mathcal{B}$ who has non-negligible advantage in distinguishing queries. Consider $\mathcal{B}$ below.

Setup

1. $\mathcal{B}$ initiate the $\mathcal{L}$-security game.
2. $\mathcal{A}$ submits a document set $\mathbf{D}$ and a keyword-documents mapping DB to $\mathcal{B}$.
3. $\mathcal{B}$ submits $\mathbf{D}$ to $\mathcal{C}$ and obtain $(I, \mathbf{c}, \mathbf{W})$.
4. $\mathcal{B}$ gives $(I, \mathbf{c})$ to $\mathcal{A}$.

Queries: For $i = 1$ to $q - 1$

1. $\mathcal{A}$ submits query w_i to $\mathcal{B}$.
2. $\mathcal{B}$ submits w_i to $\mathcal{C}$ and obtain trapdoor t_i.
3. $\mathcal{B}$ gives t_i to $\mathcal{A}$.

Challenge

1. When $\mathcal{A}$ submits the last query w_q.
2. $\mathcal{B}$ chooses a keyword $\tilde{w} \in \mathbf{W}$ which has not been queried by $\mathcal{A}$.
3. $\mathcal{B}$ submits $(w_0 = \tilde{w}, w_1 = w_q)$ to $\mathcal{C}$.
4. $\mathcal{C}$ returns the challenge $t^*{=}\mathsf{Trapdoor}(w_b)$ where $b \overset{R}{\longleftarrow} \{0, 1\}$ to $\mathcal{B}$.
5. $\mathcal{B}$ passes t^* to $\mathcal{A}$.

Response: Finally, $\mathcal{A}$ replies b' to $\mathcal{B}$ which is forwarded to $\mathcal{C}$ as response from $\mathcal{B}$.

The game is played by $\mathcal{B}$ in a way that $\mathcal{A}$ is playing in the **Real** environment because $(I, \mathbf{c}, t_1, t_2, \ldots, t_{q-1})$ is computed by $\mathcal{C}$ using the SSE scheme. Hence, $\mathcal{A}$ is receiving expected replies from $\mathcal{B}$ except for the last query.

If $\mathcal{C}$ chooses $b = 1$, $\mathcal{A}$ would have $(I, \mathbf{c}, t_1, t_2, \ldots, t_{q-1}$, $\mathsf{Trapdoor}(w_q))$ which is a consistent **Real** environment replies. Hence, $\mathcal{A}$ would output $b' = 1$ with probability $\frac{1}{2}{+}\mathbf{Adv}_{\mathcal{A}}$. Otherwise, if $\mathcal{C}$ chooses $b = 0$, $\mathcal{A}$ would have $(I, \mathbf{c}, t_1, t_2, \ldots, t_{q-1}, \mathsf{Trapdoor}(\tilde{w}))$

which is not consistent with both **Real** and **Ideal**. In this case, $\mathcal{A}$ may output $b' = 0$ or $b' = 1$ with equal probability. Thus,

$$\begin{aligned}
\Pr[\mathcal{B} \text{ wins}] &= \Pr[b = 0]\cdot\Pr[0 \leftarrow \mathcal{B}|b = 0] + \Pr[b = 1]\cdot\Pr[1 \leftarrow \mathcal{B}|b = 1] \\
&= \frac{1}{2}\Pr[b' = 0|b = 0] + \frac{1}{2}\Pr[b' = 1|b = 1] \\
&= \frac{1}{2}\left(1 + \mathrm{Adv}_{\mathcal{A}}(k)\right) \\
&= \frac{1}{2} + \frac{1}{2}\mathrm{Adv}_{\mathcal{A}}(k)
\end{aligned}$$

That implies $\mathrm{Adv}_{\mathcal{B}}(k)=\frac{1}{2}\mathrm{Adv}_{\mathcal{A}}(k)$ which is non-negligible.

This contradicts the assumption that Q-IND holds. This means that adversary $\mathcal{A}$ cannot exists. Therefore, adaptive Q-IND implies adaptive $\mathcal{L}$-security under chosen keyword attack. □

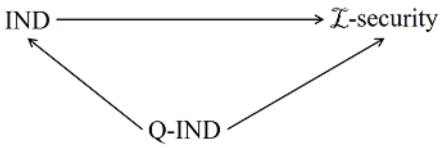

Figure 1: The implication relation of the Q-IND definition to the current IND-CKA and $\mathcal{L}$-security definitions.

By these theorem, we conclude that the Q-IND security is consistent with the existing SSE security definitions.

5. Conclusion

The review of both indistinguishability and semantic security game for SSE has shown that the gap between the SSE security definitions and the practical attacks is the adversary capability power of choosing the documents and the significance of the keyword distribution in an SSE scheme search leakage to recover the query keywords. The query indistinguishability game is proposed to identify schemes which obfuscate keyword distribution information. Query indistinguishability implies both the IND and $\mathcal{L}$-security games. Nevertheless, the defining game for semantic security of both the documents and the queries which manifest safe leakage is still an open question.

References

Bost, R. and Fouque, P.-A. (2017). Thwarting leakage abuse attacks against searchable encryption a formal approach and applications to database padding. Cryptology ePrint Archive, Report 2017/1060. http://eprint.iacr.org/2017/1060/.

Cash, D., Grubbs, P., Perry, J., and Ristenpart, T. (2015). Leakage-Abuse Attacks Against Searchable Encryption. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 668–679. ACM.

Cash, D., Jaeger, J., Jarecki, S., Jutla, C., Krawcyzk, H., Rosu, M.-C., and Steiner, M. (2014). Dynamic Searchable Encryption in Very Large Databases:Data Structures and Implementation. Cryptology ePrint Archive, Report 2014/853. http://eprint.iacr.org/2014/853.

Chang, Y. and Mitzenmacher, M. (2005). Privacy Preserving Keyword Searches on Remote Encrypted Data. In Ioannidis, J., Keromytis, A. D., and Yung, M., editors, *ACNS 2005*, volume 3531 of *LNCS*, pages 442–455. Springer.

Chase, M. and Kamara, S. (2010). Structured Encryption and Controlled Disclosure. In Abe, M., editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 577–594. Springer.

Curtmola, R., Garay, J. A., Kamara, S., and Ostrovsky, R. (2006). Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. In Juels, A., Wright, R. N., and di Vimercati, S. D. C., editors, *ACM Conference on Computer and Communications Security, CCS 2006*, pages 79–88. ACM.

Goh, E.-J. (2003). Secure indexes. Cryptology ePrint Archive, Report 2003/216. http://eprint.iacr.org/2003/216/.

Islam, M. S., Kuzu, M., and Kantarcioglu, M. (2012). Access Pattern Disclosure on Searchable Encryption: Ramification, Attack and Mitigation. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012*. The Internet Society.

Kamara, S. and Moataz, T. (2017). Boolean searchable symmetric encryption with worst-case sub-linear complexity. Cryptology ePrint Archive, Report 2017/126. http://eprint.iacr.org/2017/126/.

Naveed, M., Prabhakaran, M., and Gunter, C. A. (2014). Dynamic Searchable Encryption via Blind Storage. In *2014 IEEE Symposium on Security and Privacy, SP 2014*, pages 639–654. IEEE Computer Society.

Pouloit, D., Griffy, S., and Wirght, C. V. (2017). The strength of weak randomization: Efficiently searchable encryption with minimal leakage. Cryptology ePrint Archive, Report 2017/1098. http://eprint.iacr.org/2017/1098/.

Wright, C. V. and Pouloit, D. (2017). Early detection and analysis of leakage abuse vulnerabilities. Cryptology ePrint Archive, Report 2017/1052. `http://eprint.iacr.org/2017/1052/`.

Zhang, Y., Katz, J., and Papamanthou, C. (2016). All Your Queries Are Belong To Us: The Power of File-Injection Attacks on Searchable Encryption. Cryptology ePrint Archive, Report 2016/172. `http://eprint.iacr.org/2016/172/`.