I. J. Computer Network and Information Security, **2019, 5, 1-10**
Published Online May 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2019.05.01

Construction for Searchable Encryption with Strong Security Guarantees

István Vajda
Technical University of Budapest, Department of Informatics, Budapest, 1117, Hungary
E-mail: vajda@frogstar.hit.bme.hu

Received: 15 February 2019; Accepted: 20 March 2019; Published: 08 May 2019

***Abstract*—**We present a construction for searchable symmetric encryption (SSE). We consider a wide range of attacks and hardness assumptions and fulfill the strongest security requirements.

The "standard" privacy requirement against searchable encryption is message indistinguishability under an adaptively chosen keyword attack (IND-CKA2). We consider to protect the data and the keyword(s) together, i.e. privacy of the data is not considered as a separate problem (as the latter is typical in research papers). Beside the CKA model, we consider also the adaptively chosen trapdoor attack (CTA). Against active attacks (such as swapping attack) we add integrity protection for the (data, keyword) pair. By guaranteeing existential unforgeability (EU) for trapdoor keys we give protection against Keyword Guessing Attack (KGA). Attacks via searching for patterns in the database is prevented by randomized keyword encryption and trapdoor generation. Our construction is secure in the standard model of computation assuming bilinear groups with the widely used Symmetric eXternal Diffie Hellmann (SXDH) assumption.

***Index Terms*—**Searchable encryption, cryptanalysis, pairings, cloud computation.

I. Introduction

In general, the searchable encryption scheme involves four roles, the contributor or data owner, the server, the group of users and the issuer of trapdoors. Systems may have also a trusted authority if key generation is not done individually.

A corresponding codeword consists of two encryption parts, the data ciphertext, and the keyword ciphertext. The contributor generates codewords. A codeword may contain multiple keywords. In the next step, the contributor sends the codeword to the server. The user asks the issuer of trapdoor for a trapdoor matching an aimed keyword. Then the user sends the trapdoor to the server to carry out the search. Finally, the server allows the user to download the matching codeword and the user may decrypt the payload.

There are two main approaches within the domain of searchable encryption, the Searchable Symmetric Encryption (SSE) and the Public Key Encryption with Keyword Search (PEKS).

The searchable symmetric encryption allows Alice to outsource the storage of her data to another party (a server) in a private manner while maintaining the ability to search over it. The searching for encrypted data entries is done by the server using appropriate trapdoors generated by Alice. The straightforward application of SSE is a privately searchable cloud storage system providing end-to-end security without sacrificing utility from the side of Alice. In contrast, in case of PEKS ([5], [8]), the entries of Alice's database (DB) are generated by parties usually different from Alice, where the entries are encrypted with the public key of Alice (e.g. emails sent to Alice with sensitive content). Alice, using her secret key generates trapdoors to different keywords and sends to the server running the database.

The two approaches for searching in SSE encrypted database are called "forward" and "inverted index". In the first approach, the encrypted data together with attached (or included) encoded keywords are stored in the DB sequentially in time order. The update of the DB with a new record is straightforward. The server sequentially scans the DB for finding all the matches to a trapdoor. This leads to search complexity linear in the number of data entries in the DB. With respect to the search complexity, the inverted index approach is more efficient. In this approach, the data entries are sorted based on encoded keywords which are related to them. This arrangement allows logarithmic search complexity in the number of keywords.

In this work, we follow the forward index approach. We offer our solution for specific application scenarios with two characteristic features well suited to the forward index approach. The first is that we search for keywords only in a part of the DB (e.g. stored within a given time interval). The second feature is that new entries are frequently added to the DB. We show corresponding novel application scenarios subsequently.

The capability of fast searching of the database by the inverted index approach is an obvious advantage and it can operate smoothly on big static databases. However, the case of frequent updates is troublesome as the underlying data structure should be updated without information leakage.

For a fair comparison of SSE solutions we consider the following characteristical features: types of security guarantees and their level, the existence of formal proofs for security guarantees in the standard model of computation, forward/backward privacy, update complexity, update privacy and the above-mentioned search complexity.

"Forward privacy" guarantees that trapdoors can only be used to test keywords of documents which were already part of the DB at the time of issuing the trapdoor. The "complexity of updates" is determined by the computational effort of the server when it has to "find the place" of the new entry in the index. The process of insertion of a new entry into the database may harm the "privacy of updates" by leaking information about the added keywords (e.g., identification of entries with common keywords).

Examples of novel potential applications are the following. A set of sensors, autonomously running physically secure devices collect data from a functioning entity. For example, medical data of a human body, emergency/energy consumption data related to a building or functional parameters of an autonomous vehicle in the move are collected. The data is encrypted and uploaded to the server. The trapdoor issuer generates the necessary keys for encryption and loads it into the devices. Different data elements are expected to be processed by different end users, and the keywords are appended accordingly. For instance, different events in case of a vehicle such as speeding, failures of the engine (e.g. over consumption), in case of a building such as signals from smoke detector, leakage of gas, detection of breaking-in, last month's gas consumption or in case of a human such as body high temperature, cardiac arrhythmia may be interested to different "users" (like police, car manufacturer / fire station, police, energy service provider / family doctor, emergency medicine). Note, there are "users" who are authorized only to scan the database for keywords without seeing the complete data file (e.g. police scans for "speeding…" keyword without the need to access to other measured details like the number of persons, the planned route of the vehicle, or in the actual value of some engine parameter).

A. *Key Security Issues and Guarantees*

The key security issues for the encryption scheme we consider in this works are the following:

- Privacy of the data and the keyword,
- Eexistential unforgeability of trapdoors,
- Hiding of search patterns,
- Integrity protection of the (data, keyword) pair.

It is a challenge to provide all these guarantees in the standard model of computation (i.e. avoiding oracle models). Furthermore, the construction should be integral and uniform in its elements (e.g. the integrity protection is built in the codeword (and not an outer service), keywords and trapdoors are generated by the same algorithm).

By standard, the (external) adversary sees all messages sent between entities. The server is modeled as honest-but-curious, meaning it follows the rules of the protocol but might try to learn information that it is not authorized to know. The server is allowed to see the encrypted data items found by a keyword search action. However, information regarding the searched keyword should be hidden from the server. Accordingly, we want to guarantee the indistinguishability of (data, keyword) pair under adaptive encrypted keyword attack (IND-CKA), where the adversary has access to codeword oracle queried by (data, keyword) pairs.

There may be some leakage of information via trapdoors. An adversary may have access to trapdoors in two different ways. One of them is when (besides a CKA oracle) she has access also to a CTA oracle (Chosen Trapdoor Attack) (IND-CKA/CTA model). Obviously, in the security game, the adversary is allowed to query the CTA oracle with a keyword different from those appearing in the (data, keyword) pairs to be distinguished.

The other possibility is when an adversary fabricates trapdoor to the chosen keyword. If the adversary is successful in forging trapdoor to a chosen keyword w' (with non-negligible probability), then she could challenge the message indistinguishability game by the generation of a keyword pair $(w' \neq w)$ in the left-or-right indistinguishability game, where w is arbitrary and w ≠ w'. By knowing trapdoor Tw' the adversary could easily win the game. Equivalently, this means that keyword encryption cannot give semantic security for keywords. By this reason, EU-property of trapdoors is a necessary requirement for keyword privacy.

The retrieval of the data items is often claimed to be beyond the scope of research papers and they focus on the task of keywords search. However, in the general case, these two pieces of information can be dependent as the keyword can reveal partial information about the payload. Therefore, security guarantees, in particular, semantic security of these two ciphertexts should be considered together.

In general, the sensitivity of the data is not homogenous. The sensitivity of different data items or even of different parts of a data item may be different. For example, in a database of customer details, only the very sensitive portions of the data such as social security numbers might need protection. Data items related to the tactical or strategic actions of a corporation might need a different level of protection.

Accordingly, we classify data by the level of sensitivity as non-sensitive, sensitive or secret. We encrypt only sensitive and secret data. Best of practice, but ad hoc (i.e. with no formal proof) fast algorithms could be used to protect sensitive data, which may have bulk size. The secret class data is expected to be of short or moderate size should get provably secure protection.

Obviously, such a model with multilevel protection does not fit to an application environment of fully automated information processing. It assumes human controlled rules, in particular in the identification/designation of secret data items.

For the sensitive class hybrid encryption with an ad-hoc DEM scheme could be used. For the secret class, we assume semantically secure block encryption for data items with a typical size of one (or few) "public key size" blocks (a few thousand bits of each).

The space of keywords is considered "guessable" when the probability distribution over this set assigns non-negligible probabilities to at least one of keywords. If the adversary (say a dishonest server) is successful in forging trapdoor to chosen (guessed) keyword, she is able to scan the database for the occurrence of the corresponding keywords. Note, this is leakage of information, a breakage of the privacy of (data, keyword) pairs.

Conversely, if we can generate the ciphertext to a guessed keyword we can test a target trapdoor against this keyword. Obviously, in case of public key encryption for keywords, the latter attack can easily be carried out (e.g. by a curious server).

Hiding of search patterns is implied by randomized keyword encryption and trapdoor generation.

The codewords containing the encrypted data and encrypted keyword(s) may come under active attack (an example, is the well-known swapping attack [2]). Recall, we use symmetric key solution with respect the (encrypted keyword, trapdoor pair), for integrity protection we again consider the symmetric key approach, Message Authentication Code (MAC) for the integrity protection.

B. *Related Works*

There are no standardized algorithms and protocols relating to the practical task of searchable encryption. Therefore new research results, protocols in this field may contribute to a more complete understanding of the security requirements and needed tools of implementation.

The closest results to ours can be found in works [7], [11] and [12].

In construction [12] the data owner uses a public key algorithm for the generation of keyword ciphertexts, in contrast to our symmetric key approach. We admit that the public key approach (PEKS) offers a wider scale for potential application scenarios with finer control of users in the searching phase. For instance, in TBEKS approach ([12]) "the data search process can be carried out by the server only if the number of authorized users reaches the threshold value defined in the ciphertext". Note, in principle threshold could technically be introduced even in case of secret key encryption, where the data owner (or equivalent secret key keeper) distributes shares of the key among the members of a group. However, it would need dedicated secret keys per keyword.

It is well-known [9] that PEKS, in general, is insecure under Keyword Guessing Attack (KGA): observing a trapdoor an adversary (even the server) can test it against different guessed keywords encrypted under public key, so it is the case also with the constructions in [12]. In contrast, we give provably secure construction against such an attack.

In works [12] and [11] the authors consider the privacy of the data and the keyword as an independent task. If these two ciphertexts are generated by independent invocations of corresponding encryption algorithms, it can provide privacy for the pair of (keyword, data). However, if the two invocations share common variables such implication from component-wise guaranties may not be true or at least should be proved. Here, sharing variables may come from "optimization" of some related complexity, say the length of the codeword.

In [12] proofs are given in the non-standard model of computation (in the random oracle model), while we work in the standard model. Finally, we provide also integrity protection per ciphertext base. All in one, result [12] provides a more flexible construction as for versatility of access control scenarios, while we can give a superior set of security guarantees.

In work [11] the authors proposed the KEM/DEM technique [1] which gives the opportunity for embedded, efficient symmetric key encryption for bulk data. By our model for the sensitivity of data, our provably secure construction aims short/moderate size pieces of data. Note, if the KEM/DEM technique would be realized in a fully provably secure way then provably secure symmetric key encryption should be applied which could eliminate the hoped efficiency advantage in scenarios with large size data.

Work [11] presents generic construction theorems for PEKS/PKE approach with a goal of modular design, where the public key keyword search (PEKS) and public key data encryption subtasks (PKE) are separated. We present an explicit construction, where we consider the (data, keyword) pair as the information to be protected against passive and active attacks, and this way we can set the aim of minimization of the size of the total codeword and analyze the possibilities for weakening the underlying hardness assumptions.

II. Results

We show a construction for symmetric key searchable encryption. For the encryption of data and the keywords, we use the public key and secret (symmetric) key algorithm, respectively.

First, we give our claims about the cryptographic quality of our solution and then we add an explanation, discussion.

Our solution guarantees

- IND-CKA/CTA security for the (data, keyword) pair (Claim 1),
- Existential unforgeability (EU-CTA) for the trapdoor key (Claim 2),
- Integrity protection for the (data, keyword) pair and IND-CCA2 level of privacy for data (Claim 4).

We can prove these properties in the standard model of computation. assuming bilinear groups with the widely used Symmetric eXternal Diffie Hellmann (SXDH) assumption.

Our security goals include privacy and integrity protection for the (data, keyword) pair as well as protection against keyword guessing attack.

Our construction gives security by message indistinguishability under adaptive encrypted keywords attack (IND-CKA) and adaptive trapdoor key attack (IND-CTA), where the "message" is the (data, keyword) pair. Protection against Keyword Guessing Attack (KGA) is guaranteed by the existential unforgeability of trapdoors under adaptively chosen trapdoor attack (EU-CTA).

The construction is based on a construction of ours [7] by upgrading it in two essential steps. In the first step, we omit an assumption on the availability of a secret key MAC function with EU-CMA property used in the generation of keywords and trapdoors. Instead, we assume just the availability of a public collision-resistant function. For practical (small) sizes of keywords, we can reduce even this latter assumption to arbitrary one-to-one mapping with appropriate input-output dimensions. For an easier reference, we will call the construction as Construction 1 after step one. In the second step, we extend the security guarantees by adding integrity protection for the (data, keyword) pair (the final construct is called Construction 2). The second step uses ideas from the Cramer-Shoup IND-CCA2 secure public key encryption algorithm [6].

In our construction the algorithm used for the generation of the encrypted keywords and the trapdoors is the same, therefore a proof for the EU-property of trapdoors gives a proof also for the security of keyword encryption against fabrication attacks.

The schemes are formally proven secure in their respective security models, where we use standard and non-standard models of computation. In the standard model of computation, we assume the hardness of SXDH problem, which states that no efficient algorithm can solve the DDH problem in either of the component groups (Claim 1, Claim 2, Claim 4). We guess that the assumption of the weaker problem of Computational DH (CDH) is sufficient (Guess 1, Guess 2). We give detailed arguments to our guesses in the Appendix. Our non-standard model of computation is the generic group model. We prove the EU-CTA property of trapdoors also in this model (Claim 3).

Construction 1 gives some level of integrity protection via linking the encryption of the data and keywords(s) within codeword by using a common one-time random element r. This means that if an adversary in a swapping attack would substitute the keyword ciphertext by another one with different keyword and a random element $r'(\neq r)$ such a way that the data encryption remains untouched, it would result in failure when the keyword is tested.

However, when the encryption of the data is non-malleable (like in the case of ElGamal encryption) such weak integrity protection fails to provide detection capability against data modification attacks. In contrast, Construction 2 provides explicit integrity protection for all information (data and keyword(s)). For achieving this goal we rely on the authentication technique within the IND-CCA2 secure underlying CS-encryption algorithm. In particular, Construction 2 provides integrity for all information (data and keyword(s)) as well as IND-CCA2 security for the data ciphertext:

Our solution is a piggybacking of an IND-CCA2 secure data encryption with the goal of achieving additional integrity protection for the (data, keyword) pair. We bind the data and keyword together via putting (just) the one-time random element of the keyword ciphertext under the protection of the authentication tag of the data ciphertext. This way we leave freedom for the construction of the keyword encryption. By our protection of integrity, if the (main) part of the keyword ciphertext (left outside the protected area of the data encryption algorithm) gets attacked (e.g. by a swapping attack) the adversary fails to reach her goal as the changed part of the encrypted keyword will not be consistent with the part under protection and cannot be searched.

We make considerations about the possibilities for weakening the assumptions. We guess Claim 1 and Claim 2 remain valid even if we weaken the hardness assumption from the DDH to the CDH problem (the supporting argument is provided in the Appendix).

The discrete logarithm problem on general elliptic curves, exploiting the representation is not known to be of any help and hence generic algorithms are the best known. This implies that assuming the oracle for group operation is reasonable for our construction even from a practical viewpoint. Our point here is if we can assume generic group model then we can swap even the guessed weaker hardness assumption (CDH) to a milder one assuming just that function f is collision resistant. We show the EU-property for trapdoor keys also in the generic group model, without assuming even the hardness of the CDH problem (Claim 3).

III. Definition of the Construction

A. Construction 1

We improve the recent construction of [7] by weakening its assumptions.

Let $G = \{p, G_1, G_2, G_T, g_1, g_2, e\}$ be an instance of asymmetric bilinear pairing groups, where G_1, G_2 and G_T are three cyclic groups of prime order p, ($G_1 \neq G_2$). Let g_1 and g_2 be generators of G_1 and G_2, respectively. Let $e: G_1 x G_2 \to G_T$ be a bilinear mapping (pairing).

The mappings for encryption (C) and trapdoor generation (T) are the following:

$$C = (c_1 = g_1^r;\ c_2 = g_1^{rk_1} m;\ c_3 = g_1^{r(k_2 + k_3 f(w))}),$$
$$T = (t_1 = g_2^{r'};\ t_2 = g_2^{r'(k_2 + k_3 f(w'))}),$$

where g^{k_1} is the public key for the encryption of data, k_1, k_2 are secret keys used for encryption of keywords and generation of trapdoors, r, r' are one-time random elements $r, r', k_1, k_2, k_3 \leftarrow_r Z_p^*$. Furthermore, $f: W \to Z_p^*$ is a collision resistant mapping, where W denotes the space of codewords.

This construction is a modification of the one published in [7]. In [7]

$$c_3 = g_1^{rk_2 F(sk_{MAC}, w)},$$

$$t_2 = g_2^{r' k_2 F(sk_{MAC}, \hat{w})},$$

where $F(sk_{MAC}, x)$ is a Message Authentication Code (MAC) for message x under secret MAC-key sk_{MAC}, with the standard property of EU-CMA.

Comment: Note, if $|W| \leq p-1$ then mapping f is an arbitrary one-to-one mapping with appropriate dimensions. For instance, for a standard parameter value p=384 bit we get this simpler case with keywords not longer than 54 ASCII characters, which can be a very practical size limit.

B. Construction 2

In this step, we add integrity protection for the (data, keyword) pair. Here we use a corresponding technique from the construction of the IND-CCA2 secure Cramer-Shoup (CS-) encryption ([6]).

Let's consider a CS-ciphertext $C' = (c_1, c_2, c_3, c_4)$ over the group G_1 for a message m, where

$$c_1 = g_1^r;\ c_2 = g_1^{qr};\ c_3 = g_1^{rk_1} m;\ c_4 = g_1^{r(k_2 + qk_3 + k_4 H(all) + qk_5 H(all))}$$

Here $q \leftarrow_r Z_p^*$, furthermore H is a universal one-way hash function (UOWHF). Term $H(all)$ in authentication tag c_4 denotes the evaluation of the hash function for input (c_1, c_2, c_3). Furthermore, g^{k_1} is the public key for the encryption of data. All other keys (i.e. k_2, k_3, k_4, k_5) are secret.

For encryption of keywords and generation of trapdoors own set of secret keys are used (keys $k_2^{'}, k_3^{'}$ in Construction 1). We bind the keyword to the data via putting the first element of keyword ciphertext $(c_{11} (= g_1^{r'}), c_{41})$ (carrying the one-time random element) under the protection the authenticator tag c_4 of codeword *C'*. In other words, we recompute authenticator tag c_4 over the extended set (c_1, c_{11}, c_2, c_3) getting intermediate ciphertext $C^* = (c_1, c_{11}, c_2, c_3, c_4)$. We append the keyword tag c_{41} (the second element of the keyword ciphertext) and we get the final codeword

$$C = (c_1, c_2, c_{11}, c_3, c_4, c_{41}).$$

IV. Analysis

In this chapter, we present our claims on the security of our construction. Here we also show our guessed claims with detailed supporting arguments.

A. Analysis of Construction 1 (standard model)

Our construction is based on bilinear groups, where the encryption of data and keyword is done is one of the underlying groups, while the trapdoor keys are generated in the other. It is implicitly assumed that the only operation between the cryptographic elements over these groups is pairing by mapping *e*. It follows that information leaking about (data, keyword) pair via trapdoors might come from breaking the trapdoors (i.e. finding the corresponding keyword or secret keys used as inputs to trapdoor generation) or from accessing to trapdoors to chosen keywords (via a CTA oracle or via successful fabrication). Note, parties who are authorized accessing trapdoors to chosen keywords (equivalently, accessing a CTA oracle) are authorized to use them for scanning the database. Accordingly, information leakage through keywords via adversarial attacks may come from fabrication attack (CTA oracle can be queried with keywords different from the target ones). Recall, if the space of keywords is "guessable" and the adversary is able to fabricate trapdoors to chosen keywords, the scheme will leak information on the target (data, keyword) pair with non-negligible probability. Because we do not want to make restrictions on the space of keywords, therefore, we have to guarantee that trapdoors are unforgeable.

Let's introduce event B ={adversary cannot forge valid trapdoor}. Using this notation, the analysis of the indistinguishability game proceeds as

$$P(b = b') \leq P(b = b' \mid B) + P(B^c)$$

where $P(b = b')$ denotes the probability of correct decision in the indistinguishability (IND) game, where the oracle chooses a bit b and the challenger decides on the bit b'. Accordingly, we will require that $P(B^c)$ is negligible. In the analysis of privacy, we will restrict our attention to the conditional probability $P(b = b' \mid B)$.

In other words, the analysis is divided into the analysis of forging the trapdoors and the analysis of IND-CKA/CTA security conditioned on the event B (with associated probabilities $P(B^c)$ and $P(b = b' \mid B)$, respectively).

IND-CKA/CTA property: We assume the hardness of the so-called Symmetric Xternal Diffie-Hellman (SXDH) problem. The SXDH assumption states that no efficient algorithm can solve the DDH problem in either of groups G_1 and G_2 of a bilinear instance. As a consequence, the ElGamal encryption of data is in Construction 1 is IND-CPA secure.

Claim 1: Construction 1 guaranties IND-CKA/CTA security for the (data, keyword) pair.

Proof: We assume that $P\left(B^c\right)$ is negligible (see Claim 2). We use a result in [4] on multi-recipient encryption technique in case of the ElGamal encryption. The scenario of multi-recipient encryption is the following. We want to send messages $m_1,...,m_n$ to recipients $A_1,...,A_n$, respectively, where different recipients have different public keys. The idea is that we can save one-time random elements, and instead of n different elements $r_1,...,r_n$ we can use the same random element r for each recipient. If the underlying encryption (like ElGamal) is IND-CPA secure, the multi-recipient encryption will keep this level of security for the series of messages $m_1,...,m_n$. (The same is true for IND-CCA2 secure encryption, like Cramer-Shoup, [4]).

Let's write codeword (*C*) and a trapdoor (*T*) in Construction 1 into the following equivalent form:

$$C=(c_1=g_1^r;\ c_2=g_1^{rk_1}m;\ c_3=g_1^{rk_2}m')$$
$$T*=(t_1=g_2^{r'};t*=g_2^{r'k_1}0;\ t_2=g_2^{r'k_2}m'')$$

where $m'=g_1^{rk_3f(w)}$, $m''=g_2^{r'k_3f(w')}$, furthermore $t*(=0)$ is a dummy element. Note, C and $T*$ have similar formulae. Note, these "codewords" can be considered as 2-recipient encryptions (in our construction with hidden "public" keys g^{k_1} , g^{k_2}). Having access to more encryptions of type C or $T*$ is equivalent to more 2-recipient encryptions. Recall, IND-CPA security assumes an adaptively chosen plaintext attack. As the underlying encryption is IND-CPA secure and the multi-recipient encryption maintains this level of security against adaptively chosen plaintexts, IND-security of (m,w) (data, keyword) pair will be kept when accessing KA and CTA oracles, where CTA oracles can be queried only with keywords different from the ones in the indistinguishability game.

We have the following guess on weakening the hardness assumption in Claim 1 (see the detailed argument in the Appendix).

Guess 1: Claim 1 is valid even if we assume a weaker problem, the hardness of the CDH problem in the component groups.

If this guess is true then in this paper we can assume just the hardness of the CDH problem in the component groups, as our proof for the existential unforgeability of trapdoor keys assumes just the hardness of this weaker problem (shown subsequently).

Existential unforgeability of trapdoors: Assume function f is collision resistant.

Claim 2: Construction 1 guaranties existential unforgeability (EU-CTA) for the trapdoor key.

Comment: as the computation of trapdoors and keyword encryptions uses the same algorithm the EU-guarantee claimed above is valid also for keyword encryptions.

Proof: Let q denote the number of requests to the trapdoor oracle.

Case a): adversary tries to fabricate valid t_2 from scratch ($q=0$):

Key variables k_2,k_3 are uniformly distributed random variables over Z_p^* in the view of the adversary. First consider the case, when the adversary computes $t_1=g^{r'}$ for a randomly chosen element r' . Assume she is able to fabricate $t_2=g^{r'(k_2+k_3f(w'))}$. Obviously, the adversary is aware also of $t_2^{'}=g^{k_2+k_3f(w')}(=t_2^{r'^{-1}})$ (with non-negligible probability). However, the latter contradicts to the fact that exponent $k_2+k_3f(w)$ is a uniformly distributed random variable over Z_p^* in the view of the adversary. (In the case when the adversary directly chooses a random group element t_1 (i.e. without seeing r'), a similar probabilistic argument can be used.)

Case b): ($q\geq 1$)

The adversary tries to modify the trapdoor component $t_2(w)$ in its keyword value to get a trapdoor $t_2(w')$ to a new keyword w' known by her. In this attack, the adversary reuses an existing random element r unknown for her.

Assume the adversary is successful with non-negligible probability. By introducing notation $t_2\left(w'\right)=X\ t_2\left(w\right)$ where $X =^{rk_3(f(w')-f(w))}$ we can observe that the adversary should also be able to compute group element X . Knowing X she could compute group element g^{rk_3} as difference $f(w')-f(w)$ is not zero with overwhelming probability and is (assumed to be) known by the adversary. It follows that she would know also group element g^{rk_2} (with non-negligible probability). However, the latter would lead to a contradiction: similarly, as in the proof of Claim 1 we consider the trapdoor associated with component $t_2(w)$ in the following ElGamal-encryption-like form

$$T=(t_1=g_2^r;t_2=g_2^{rk_2}m)\ ,$$

where $m=g_2^{rk_3f(w)}$ and $z=g^{rk_2}$ correspond to the message and to the one-time secret symmetric key, respectively. Note, the message $m=g_2^{rk_3f(w)}$ is independently chosen from the key z. If the adversary could compute z it would contradict to the semantic security of ciphertext T (based on DDH assumption). Note, this conclusion is independent of the number of requests to the trapdoor oracle.

Case c): ($q \geq 1$)

Now we consider the case when the adversary fabricates trapdoor to a new keyword w'

$$T' = (t_1 = g_2^{r'}; t_2 = g_2^{r'k_2} m'),\ m' = g_2^{r'k_3 f(w')}$$

with a fresh random element r', where the trapdoor is represented as an ElGamal ciphertext as above. As the adversary is not aware of "public key" g^{k_2} therefore she should not be able to compute a ciphertext even to a known "message" (the case for the adversary is worse as she knows r' and w' ($f(w')$) but does not know g^{k_3}).

We have the following guess on weakening the hardness assumption in Claim 2 (see the detailed argument in the Appendix).

Guess 2: Claim 2 is valid even if we assume a weaker problem, the hardness of the CDH problem in the component groups.

B. Analysis of Construction 1 (generic group model)

The generic group model is an idealized cryptographic model, where the adversary is given access only to a randomly chosen encoding of a group, instead of specific encodings as we assume that the properties of the representation of the elements of the algebraic structure under consideration cannot be exploited. The formal model includes an oracle that executes the group operation. This oracle takes two encodings of group elements as input and outputs an encoding of a third element. If the group should allow for a pairing operation this operation is modeled by an additional oracle.

Claim 3: Assume generic group model of computation and that function f is collision resistant. Construction 1 guaranties existential unforgeability for the trapdoor key for adaptively chosen trapdoor attack.

Proof: Assuming a cyclic group, we can add, multiply by a constant or negate (unseen) exponents of group elements by multiplication, exponentiation or inversion operations offered by the oracle. Equivalently, the adversary has access to any group element with an exponent equal to any linear combinations of the exponents of any group elements accessible for the adversary. We prove that no linear combination of such exponents can lead to a new valid exponent of a trapdoor key.

Exponents from trapdoor requests are the following:

$$1)\ r_1, r_1(k_2 + k_3 f(w_1))$$

$$2)\ r_2, r_2(k_2 + k_3 f(w_2))$$

$$\dots$$

$$n)\ r_n, r_n(k_2 + k_3 f(w_n))$$

The goal of the adversary is to generate a pair $[r',\ r'(k_2 + k_3 f(w'))]$ for arbitrary r' and a new keyword w' by using (just) linear combinations of the exponents above. The output of the adversarial computation has the following form:

$$r' = \sum_{i=1}^{m} a_i r_{j(i)};\quad k_2 r' + k_3 (\sum_{i=1}^{m} a_i r_{j(i)} f(w_{j(i)})) \qquad (1)$$

with the following restriction

$$r' = \sum_{i=1}^{m} a_i r_{j(i)} f(w_{j(i)})\ f(w')^{-1} \qquad (2)$$

where the adversary controls combining coefficients $a_i,\ i = 1, \dots, m$. Note, restriction (2) can be kept (with overwhelming probability) only if $f(w_{j(i)}) = f(w')$ for all i, which would imply finding f -collisions.

Note, adaptive choosing of keywords in trapdoor requests does not affect the above argument.

Comment: It is not hard to see, that if the adversary were able to do any (modular) computations directly with the exponents (in particular, if she could compute multiplicative inverses in addition to linear combinations), then she could compute even the keys. This would be the case of an unconditional adversary having direct access to the exponents. Note, even such a powerful attack could be prevented by ensuring that $f(w_i)$ is unknown to the adversary, which is equivalent to an (impractical) assumption that $f()$ is an (unconditionally) secret random function.

C. Analysis of Construction 2 (standard model)

The main goal with this step is to add integrity protection for the (data, keyword) pair.

Claim 4: Construction 2 provides integrity protection for the (data, keyword) pair and IND-CCA2 level of privacy for data.

Proof:

Integrity protection: We can observe, that intermediate ciphertext $C^* = (c_1, c_{11}, c_2, c_3, c_4)$ inherits the security property of the underlying CS-construction. Indeed, as it can be verified by checking the original proof, the inclusion of an independent random element (c_{11}) into the CS-ciphertext, does not weaken the IND-CCA2 level of security.

Accordingly, an adversary is not able to modify element $c_{11}(g_1, r')$. If she modifies it to a group element ρ, then it is equivalent to changing random element r' to $r'' \neq r'$, where $\rho = c_{11}(g_1, r'')$. This would lead to a failure in testing the keyword, and equivalently it would result in a DoS-like attack, which cannot be the goal of an adversary by the usual adversarial models. The same argument applies for the case of several keywords tags (each with own random element).

Privacy: Instances of two encryptions are invoked independently, where the one-time random elements and also the key sets are independent. One of them encrypts the data with an IND-CCA2 level under CS-encryption. Keywords are encrypted under IND-CKA/CTA secure encryption. The level of privacy of data will not be weakened (from IND-CCA2) if we allow the adversary to access also CKA and CTA oracles.

Existential unforgeability: It is obvious from the construction, that the property of existential unforgeability of the trapdoor key is not affected by adding the property of integrity protection.

There is another way of construction, which is fully based on CS-construction. Instead of a single, we start from two CS- ciphertexts, $C = (c_1, c_2, c_3, c_4)$ and $C_1 = (c_{11}, c_{21}, c_{31}, c_{41})$, where the first and the second ciphertext encrypts message m and keyword w, resp. Now, we drop terms c_{21}, c_{31} and use just remaining terms c_{11}, c_{41}, i.e. we use the CS-authentication tag in the role of a keyword tag. Note, the EU-property of the CS-authentication tag will be inherited by the keyword tag. Furthermore, as terms c_{11}, c_{41} were part of a CS-ciphertext for plaintext w, the semantic security of it is also inherited in the new role. We can observe on pro side that no separate key set is needed for keyword encryption, while on against side we can see an increase in the computational complexity of keyword encryption.

V. Conclusion

This work is an upgrade and extension of our solution [7] with respect to the offered security guarantees. For a fair comparison of our solution to existing dynamic SSE constructions we consider the following characteristical features: types of security guarantees and their level, existence of formal proofs for security guarantees, forward/backward privacy, update complexity, update privacy and search complexity.

If we consider just the dynamic SSE constructions ([14],[15]) which are able to offer IND-CKA2 security in the standard model of computation our construction has the following advantages:

Security: resistance against adaptive chosen-keyword attacks, adaptively chosen trapdoor attack, keyword guessing attack as well integrity protection of the (data, keyword) pair at record level (e.g. swapping attack). These guarantees are proved in the standard model of computation.

The complexity of encoding: our ciphertexts with keyword tags and trapdoors are short (in the number of group elements they consist of).

Update complexity: the update of records with new keyword tags is non-interactive with low complexity, depending only on the number of keywords.

Update privacy: when the DB is updated with a new record the data ciphertext and the added encrypted keywords are stored, and leaks only the number of keyword tags. In contrast, other constructions leak information about the keywords.

Forward/backward privacy: the construction can provide forward/backward privacy by using time-dependent keywords. Note, time-specific keywords are not suitable in inverted index solutions as they cause an infinitely growing inverted index.

Search complexity: our solution is competitive to the inverted index approach in the mentioned application scenarios, concretely, when we search for keywords only in a part of the DB (e.g. stored within the last three days) and when new entries are frequently added to the DB.

At first glance, it seems not meaningful to consider constructions for integrity protection, especially at so granular level as codewords, when a standard technology is available, a single digital signature for the whole (or segments) of the database. Note, however, if it is done securely, the digital signature has to be updated each time when a new record (codeword) is stored in the database. This means that integrity protection per ciphertext does not necessarily need more computation then protection by using digital signatures. Furthermore, computation of an authentication tag (MAC) may have smaller computational complexity than a digital signature. On the other hand, the main advantage of the digital signature technique is its public verifiability.

Recall the procedure of transmission of data from the data owner to the final recipient is the following: first, the data owner uploads the encrypted data and the attached encrypted keywords to the server, next users scan the database searching for particular keywords and finally (having appropriate decryption keys) decrypt corresponding data. But what if the computational complexity of the decryption of the data item is lower than that of the testing a keyword? In such cases, it might seem that a data user may be better off skipping the step of keyword testing, and jumping directly to the decryption of the data as it may happen that by looking into the payload he can easily identify also the keyword.

Recall, however, that by the underlying application scenario the remote database (cloud) provides the computational power for scanning the database for keywords. Furthermore, there are parties in this scenario users who are allowed only testing the keywords, i.e. not permitted to see the plaintext data.

Nonetheless, if we are purely interested in complexity comparison, for our construction we should compare the complexity of the evaluation of a bilinear mapping to the complexity of the decryption of an ElGamal-type ciphertext. Recall, pairing computation is the heaviest operation in pairing-based cryptosystems, although the concrete values of complexity strongly depend on the actual parameter values and the details of the implementation. Efficient software implementations of pairings exist for different ARM-based platforms and x86-64 PCs for 128-, 164-, and 192-bit security levels. The ARM-based platforms are used in handheld smartphones and tablets. These platforms are predicted to become a dominant computing platform [13].

Appendix: Guesses on Weakened Hardness Assumptions

Guess 1: Claim 1 is valid even if we assume a weaker problem, the hardness of the CDH problem in the component groups.

An argument to Guess 1: Consider the following pair of triplets of random variables

$$E=(c_1=g^r;\, c_2=g^{rk_1};\, c_3=g^{rk_2})$$

and

$$D=(d_1=g^{R_1};\, d_2=g^{R_2};\, d_3=g^{R_3})$$

where all parameters in the exponents are independent uniformly chosen random variables (over the range of exponents). Note, triplets E and D have the same distribution, consequently, they are also algorithmically indistinguishable. The question is that if these triplets remain indistinguishable when we can observe further samples of E and D type where the one-time random elements are freshly chosen, however random variables k_1, k_2 are kept fixed. Intuitively, the wish of a distinguisher (with the task to decide if the samples come from distribution E or D) is to decide if the variable $\log_g(c_1)^{-1}\log_g(c_2)$ is constant or varies over different samples.

By the Divisible Computation Diffie-Hellman problem (DCDH) (equivalent to the CDH problem, [5]) from knowing group elements (g, g^x, g^y), it is hard to compute element $g^{y/x}$. Applying this to our problem, from knowing group elements $(g, g^r, g^{rk_1}, g^{rk_2})$ it is hard to compute element g^{k_1} or g^{k_2} .

Guess 2: Claim 2 is valid even if we assume a weaker problem, the hardness of the CDH problem in the component groups.

An argument to Guess 2: Let's reconsider the proof of Claim 2. In case a) we needed no hardness assumption.

In the case of b) the adversary tries to modify a given trapdoor key $t_2(w)$ in its keyword value to get a trapdoor key $t_2(w')$ for a new keyword w'. Note, in this attack, the adversary reuses an existing random element r unknown for her. We have seen that the adversary could compute group element g^{rk_3} (with non-negligible probability). However, the latter would contradict to the CDH assumption:

By the equivalent Divisible Computation Diffie-Hellman problem (DCDH problem, [5]) from knowing a pair of group elements (g^r, t_2) an efficient adversary is not able to compute element $g^{(k_2+k_3f(w))}$. Because she is not aware of the product $g^{k_2}g^{k_3f(w)}$, she is not aware of either of its factors (g^{k_2}, $g^{k_3f(w)}$). As she does not know $g^{k_3f(w)}$ she cannot find out g^{k_3} .

It follows that the success probability of an adversarial guess on the value of g^{k_3} can only be negligible. On the other hand, the adversary is aware of g^r with certainty. Even if at this step the adversary could send the guessed value of g^{k_3} together with g^r as input to an oracle with unconditional computational complexity, she would not be able to get a correct guess on g^{rk_3} with non-negligible probability.

References

[1] M. Abe, R. Gennaro, K. Kurosawa, V. Shoup, "Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM", *In Advances in Cryptology—EUROCRYPT 2005, ed. by R. Cramer. Lecture Notes in Computer Science*, vol. 3494 (Springer, Berlin, 2005), pp. 128–146.

[2] J. Baek, R. Safavi-Naini, and W. Susilo, "On the Integration of Public Key Data Encryption and Public Key Encryption with Keyword Search", *In ISC'06, volume 4176 of LNCS*, pages 217–232. Springer, 2006.

[3] D. Boneh, G.D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search", *In EUROCRYPT 2004, Proceedings*, pages 506–522.

[4] M. Bellare, A. Boldyreva, K. Kurosawa and J. Staddon, "Multirecipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security", *IEEE TRANSACTIONS ON INFORMATION THEORY*, VOL. 53, NO. 11, NOVEMBER 2007

[5] F. Bao, R. H. Deng, H. Zhu, "Variations of Diffie-Hellman Problem", *International Conference on Information and Communications Security, ICICS 2003: Information and Communications Security* pp 301-312

[6] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack", *In proceedings of Crypto 1998, LNCS 1462*, p. 13-25.

[7] M. Horvath and I. Vajda, "Searchable Symmetric Encryption for Restricted Search", *Journal on Communications Software and Systems*. 2017

[8] S. L. Renwick and K. M. Martin, "Practical Architectures for Deployment of Searchable Encryption in a Cloud

Environment", *Information Security Group, Royal Holloway, University of London*, London TW20 0EX, UK; Cryptography 2017,1, 19; 15 November 2017

[9] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack", *IEEE Transactions on Computers*, 62(11):2266-2277, Nov 2013.

[10] G. S. Poh, J. Chin, W. Yau, K. R. Choo and M. S. Mohamad, "Searchable Symmetric Encryption: Designs and Challenges", *ACM Comput. Surv.* 50, 3, Article 40 (May 2017), 37 pages.

[11] R. Zhang and H. Imai, "Generic Combination of Public Key Encryption with Keyword Search and Public Key Encryption ", *Cryptology and Network Security, 6th International Conference, CANS 2007*, Singapore, December 8-10, 2007, Proceedings (pp.159-174)

[12] S. Zhang, G. Yang, and Y. Mu, "Linear encryption with keyword search", *In Joseph K. Liu and Ron Steinfeld, editors, Information Security and Privacy: 21st Australasian Conference, ACISP 2016*, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II, pages187-203, Cham, 2016. Springer International Publishing.

[13] R. Azarderakhsh et.al., "Fast Software Implementations of Bilinear Pairings", *IEEE Transactions on Dependable and Secure Computing*, Vol. 14 , No. 6, Nov.-Dec. 1, 2017.

[14] Van Liesdonk, P., Sedghi, S., Doumen, J., Hartel, P. H., and W. Jonker, "Computationally efficient searchable symmetric encryption", *In Secure Data Management, 7th VLDB Workshop, SDM 2010*, Proceedings, pages 87–100.

[15] S. Gajek, "Dynamic symmetric searchable encryption from constrained functional encryption", *In Topics in Cryptology – CT-RSA 2016*, pp. 75–89.

Author Profile

István Vajda graduated from the Telecommunication Department at the Technical University of Budapest. He received the PhD and DSc degrees in 1985 and 1997, respectively. Since 1998, he has been a Professor at the Department of Informatics. He is the co-founder of the Laboratory of Cryptography and Systems Security (CrySyS). During 1990's his research interest was in algebraic code designs for secure multiple access channels. Recently, his research interests are in design and analysis of secure systems, with a special emphasis on provably secure cryptographic primitives and protocols. His application expertise covers secure wireless communication, secure routing and sensor networks.

How to cite this paper: István Vajda,"Construction for Searchable Encryption with Strong Security Guarantees", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.5, pp.1-10, 2019.DOI: 10.5815/ijcnis.2019.05.01