Anonymous Fuzzy Identity-Based Encryption for Similarity Search*

David W. Cheung, Nikos Mamoulis, W.K. Wong, S.M. Yiu, and Ye Zhang

Department of Computer Science, University of Hong Kong, Hong Kong
{dcheung,nikos,wkwong2,smyiu,yzhang4}@cs.hku.hk

Abstract. In this paper, we consider the problem of predicate encryption and focus on the predicate for testing whether the Hamming distance between the attribute X of a data item and a target V is equal to (or less than) a threshold t where X and V are of length m. Existing solutions either do not provide attribute protection or produce a big ciphertext of size $O(2^m)$. For the equality version of the problem, we provide a scheme which is match-concealing (MC) secure and the sizes of the ciphertext and token are both $O(m)$. For the inequality version of the problem, we give a practical scheme, also achieving MC security, which produces a ciphertext with size $O(m^{t_{max}})$ if the maximum value of t, t_{max}, is known in advance and is a constant. We also show how to update the ciphertext if the user wants to increase t_{max} without constructing the ciphertext from scratch.

Keywords: predicate encryption, anonymous fuzzy IBE, inner-product encryption.

1 Introduction

It is getting more popular for a data owner to take advantage of the storage and computing resources of a data center to hold the data in encrypted form. Depending on the access right of a user, only authorized records can be retrieved. Due to privacy and security concerns, the data should not be decrypted at the data center and checked against the criteria. Thus computation should be carried out on encrypted data directly. Usually, users are given a token (by the owner) and based on this token, only authorized records are selected and later decrypted on the user site. Examples of outsourcing applications which are based on this model are retrieval of encrypted documents by keyword matching, selection of encrypted audit logs using multi-dimensional range queries on authorized IP addresses or port numbers, and Hamming distance based similarity search on encrypted DNA sequence data. The problem, in fact, has received much attention from both database [8,17] and cryptography [15,2,14,6,10] communities.

In general, the problem can be stated as follows. For each data item M, there is an associated predicate attribute value X (X may or may not be part of the

* This work was supported by Grant HKU 715509E from Hong Kong RGC.

O. Cheong, K.-Y. Chwa, and K. Park (Eds.): ISAAC 2010, Part I, LNCS 6506, pp. 61–72, 2010.

record M). Let f be a predicate on X representing the computation we want to carry out so that the data item M can be successfully decrypted if and only if $f(X) = 1$. Authorized users will obtain a token generated by the owner in order to perform the predicate evaluation. A different token can be generated for different users with different access power. For example, consider a database of medical records. Each record (M) can be encrypted based on a selected region of the DNA sequence (X) of the person. Note that X is the associated predicate attribute of M and needs not be part of the record M. When a research team is authorized to investigate the relationship between a certain DNA sequence V with diseases, this team would acquire a token which corresponds to the predicate f such that $f(X) = 1$ if and only if $HammingDist(X, V) \leq t$, say $t = 5$. By using the token, the research team would decrypt all medical records for which the corresponding DNA sequence is similar to V. In the above motivating example, it is obvious that the research team should not infer any information on records for which the corresponding attribute X is far away from V (i.e. $HammingDist(X, V) > 5$) since they are not authorized to do so. In addition, it is desirable that the ciphertext $E(pk, I, M)$, where pk is the public key of the data owner and E is the encryption algorithm, is the same for different V and t such that the encryption of data items needs only to be done once. This emerging branch of encryption schemes are referred to as *predicate encryption*.

Here we focus on the predicate f that tests whether the Hamming distance between V and X is equal to (or less than) a certain threshold t, where V and X are assumed to be bit vectors of equal length $m \in \mathbb{N}$. Hamming distance is an important searching criterion for record retrieval, with many interesting applications in databases, bioinformatics, and other areas. Note that V and t can vary and will be given to the owner for the generation of a token independent of the ciphertext $E(pk, I, M)$.

The security of predicate encryption [10] can be classified into (1) protecting the data items only; and (2) protecting both the data items and the associated predicate attributes. Attribute protection is usually referred to as *anonymous* in general and can be further classified into two levels: *match-revealing (MR)* [14] and *match-concealing (MC)* [6] (or say *attribute-hiding* in [10]). The difference between MR and MC is that predicate attributes will remain hidden in MC level even if they satisfy the predicate. While in MR level, if attribute X satisfies the predicate f (i.e. $f(X) = 1$), some more information on X (or even the whole X) other than the information of $f(X) = 1$ may be known. In our "medical record" example, we sometimes require the encryption scheme to be anonymous in order for the DNA sequence to be protected since the DNA sequence may contain genetic disorder information which should be kept private for individuals. It depends on applications whether we require MC or MR level of security. For example, if attribute X is part of data item M, when X satisfies the predicate, data item M will be properly decrypted and therefore people can see the entire X anyway. In such case, MR security seems to be a proper choice. So far, the predicate encryption scheme supporting this Hamming distance predicate is the one in [12], called "Fuzzy Identity-Based Encryption". However, this scheme

does not provide the property of anonymity (i.e., attribute protection). In this paper, we propose "Anonymous Fuzzy Identity-Based Encryption" schemes to handle both the equality (i.e., $HammingDist(X, V) = t$) and inequality (i.e., $HammingDist(X, V) \leq t$) threshold versions of the predicate.

It is not trivial how to make the scheme in [12] anonymous. On the other hand, there is a generic solution [6] (see Appendix A) that can support the predicate we study with the property of anonymity and it is MC secure. This general construction supports for any polynomially computable predicate. However, it embeds (pre-computes for) every possible value of $V \in \{0,1\}^m$ and t in the ciphertext even for the equality threshold version of the problem (the same applies to the inequality version), thus the size of each ciphertext is $O(t2^m)$ which is impractical even for moderate m although the token size is constant.

1.1 Our Contributions

For the equality threshold version, we provide an anonymous fuzzy identity-based encryption scheme achieving the MC level of security with both the size of ciphertext and token equal to $O(m)$. The core idea of our scheme comes from [10] which provides an inner-product encryption scheme. We represent the Hamming distance computation as an inner product such that X and V can be separated into the ciphertext and the token, respectively, so that V can be given only when the token is needed to be generated.

For the inequality threshold version, we provide a practical scheme to solve the problem. In many applications (e.g., in bioinformatics applications), $t << m$. Even assuming that we know the maximum value of t (t_{max}) in advance and is a constant, the size of the ciphertext produced by the solution based on [6] is still $O(2^m)$. In our scheme, also achieving the MC security level, the size of ciphertext is only $O(m^{t_{max}})$ (precisely, $\sum_{i=0}^{t_{max}+1} \binom{m}{i}$) which is much smaller than $O(2^m)$ if $t_{max} << m$. The core of this scheme is to come up with an inner product expression with a total number of $\sum_{i=0}^{t+1} \binom{m}{i}$ terms to express whether $HammingDist(X, V) \leq t$ and modifying the scheme in [10] to a new primitive to support our encryption scheme. We also show how to update the ciphertext to increase the value of t_{max} without recomputing the ciphertext from scratch.

1.2 Related Work

The predicate that was studied in the beginning is "exact keyword matching". That is, whether the value associated with the token is equal to the attribute value hidden in the ciphertext. Schemes that only provide data item security are basically "Identity-Based Encryption" [3]. Schemes protecting both the data item and the attributes were initiated by Song *et al.* [15] in the private-key setting and by Boneh *et al.* [2] in the public-key setting. The relationship between [2] and "Anonymous Identity-Based Encryption" [7] was revisited in [1].

Later, range predicates were also considered. Boneh *et al.* devised an Augmented Broadcast Encryption [5] which allows checking if the attribute value falls within a range on encrypted data. Their scheme also provides attribute

protection. Then, Boneh and Waters [6] extended it for multi-dimensional range queries. Shi *et al.* [14] devised a more efficient scheme, but it is MR secure.

The predicate investigated in this paper was initiated by [12] where their scheme only protects the data item. However, there is no practical scheme supporting this predicate with attribute protection in a public-key setting. Park *et al.* [11] investigated this problem in the private-key setting and their solution is IND2-CKA secure. Liesdonk [16] proposed a public-key setting for this problem. However, the scheme requires the threshold value t to be fixed in the setup time.

Our work is using [10] as a framework. [10] provided schemes for handling predicates represented as inner products. We show how to represent the Hamming distance computation as inner products, and then derive a slightly different encryption scheme for better performance when considering the inequality case. In our work, we consider the problem of attribute protection in a public-key setting. In some applications, people may also want to provide protection to predicate ("the token"), which is inherently unachievable in public-key setting. A predicate encryption supporting inner product in private-key setting has been devised in [13] which can provide predicate privacy.

In this paper, we only consider a non-interactive solution (i.e., an encryption scheme). We should note that there are interactive solutions (e.g., [9]) for the same problem.

1.3 Paper Organization

The rest of this paper is organized as follows. Section 2 introduces the framework of the encryption scheme and the MC security model. Section 3 presents the scheme for the equality threshold version (i.e., $HammingDist(V, X) = t$) of the problem and Section 4 deals with the inequality threshold version (i.e., $HammingDist(V, X) \leq t$) of the problem.

2 Preliminaries

We assume that the attribute X is represented as a bit vector of length $m \in \mathbb{N}$. The attribute V (referred to as the *target attribute*) provided by the user to generate the token is also a bit vector of the same length as X. In the rest of the paper, for simplicity, we focus on predicate-only encryption, that is, we assume that we only have X without the data item M. So, the scheme will output "1" to indicate the decryption is successful ($f(X) = 1$) and "0" otherwise. Note that extending solutions for predicate-only encryption to include the data item M can be done easily (e.g., [10]). Also, there exist applications that we only need to encrypt the attribute X and based on the decryption result to retrieve the corresponding records separately.

Let $\mathcal{G}$ be a group generator which takes security parameter $n \in \mathbb{N}$ as input and (randomly) outputs $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map which can be computed efficiently. $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic with the same composite order $N = pqr$ where p, q and r are three distinct large primes. Let $\mathbb{G}_p$, $\mathbb{G}_q$ and $\mathbb{G}_r$ be the cyclic subgroups of $\mathbb{G}$ with order of p, q and r separately.

2.1 Framework

An anonymous fuzzy identity-based encryption scheme Π consists of the following four probabilistic polynomial-time (PPT) algorithms.

- $\texttt{Setup}(1^n)$: On an unary string input 1^n where $n \in \mathbb{N}$ is a security parameter, it produces the public-private key pair (pk, sk).
- $\texttt{Encrypt}(pk, X)$: On the public key pk and attribute vector X, it outputs the ciphertext C.
- $\texttt{GenTK}(pk, sk, V, t)$: The token generation algorithm takes the public key pk, private key sk, outputs the token TK for the vector V and threshold t.
- $\texttt{Test}(pk, TK, C)$: On the ciphertext C, the token TK and the public key pk, it outputs "1" if the Hamming distance between the vector X associated with C and the vector V associated with TK is equal to t associated with TK (is less than or equal to t for the inequality version); "0" otherwise.

2.2 The MC Security Model

We define the MC security in the selective model [6,14,10] as follows.

Definition 1. *An anonymous fuzzy identity-based encryption scheme Π is selectively MC secure if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible.*

Setup: The adversary $\mathcal{A}(1^n)$ outputs two possible equal-length vectors X_0 and X_1 to the challenger $\mathcal{C}$. $\mathcal{C}$ runs $\texttt{Setup}(1^n)$ and gives pk to $\mathcal{A}$.

Challenge: The challenger $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and encrypts X_b under pk. The ciphertext C^* is given to $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ may adaptively request a polynomially bounded number of tokens for any (V_i, t_i), with the restriction that $t_i = HammingDist(V_i, X_j)$ for both $j = 0, 1$ or $t_i \neq HammingDist(V_i, X_j)$ for both $j = 0, 1$. (For inequality threshold: $t_i < HammingDist(V_i, X_j)$ for both $j = 0, 1$ or $t_i \geq HammingDist(V_i, X_j)$ for both $j = 0, 1$)

Guess: $\mathcal{A}$ outputs a guess bit b'. The advantage of $\mathcal{A}$ is defined as $\left|\Pr[b' = b] - \frac{1}{2}\right|$.

The intuition behind Definition 1 is that if the encryption scheme is MC secure that C^* leaks no information on the attribute X_0 and X_1, then the adversary $\mathcal{A}$ cannot distinguish X_0 from X_1 to output a proper guess bit $b' = b$. To restrict $t_i = HammingDist(V_i, X_j)$ for both $j = 0, 1$ or $t_i \neq HammingDist(V_i, X_j)$ for both $j = 0, 1$ prevents $\mathcal{A}$ trivially distinguishes X_0 from X_1 because the only information allowed to leak in the MC security is whether t_i is equal to $HammingDist(V_i, X_j)$ or not. A similar restriction is applied to the inequality threshold case as well.

3 Scheme for Equality Threshold

In this section, we describe our scheme for handling the equality threshold version of the Hamming distance predicate. Recall that both the target attribute V and

threshold t will only be known when the user wants to obtain a token. We need to produce a ciphertext based on attribute X; a token based on V and t even after X is encrypted. The $\texttt{Test}()$ combines the ciphertext and token together to compute Hamming distance $HammingDist(X, V)$. To the best of our knowledge, we are aware that only bilinear map can provide such ability while not being too powerful to break the security. Intuitively, given g^a and g^b, bilinear map combines a and b by computing $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$. More specifically, if we encrypt attribute X as ciphertext $C = g^{f(X)}$ and generate token $TK = g^{y(V,t)}$ for target attribute V and threshold t, we can construct $\texttt{Test}(C, TK)$ as $\hat{e}(C, TK) = \hat{e}(g^{f(X)}, g^{y(V,t)}) = \hat{e}(g, g)^{f(X)\cdot y(V,t)}$. If we can find $f(X)$ and $y(V, t)$ such that $f(X)y(V, t) = HammingDist(X, V)$, $\texttt{Test}(C, TK)$ will function correctly. More generally, $f(X)$ and $y(V, t)$ would output a vector. This is because given two vector $(g^{a_1}, \ldots, g^{a_m})$ and $(g^{b_1}, \ldots, g^{b_m})$, we would combine $\boldsymbol{a} = (a_1, \ldots, a_m)$ and $\boldsymbol{b} = (b_1, \ldots, b_m)$ by computing $\prod_{i=1}^{m} \hat{e}(g^{a_i}, g^{b_i}) = \hat{e}(g, g)^{\sum_{i=1}^{m} a_i, b_i} = \hat{e}(g, g)^{\boldsymbol{a}\cdot\boldsymbol{b}}$ where $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the inner product [10,4] of $\boldsymbol{a}$ and $\boldsymbol{b}$.

Lemma 1. *Given two bit vectors X and V of equal length m, $HammingDist$ (X, V) equals $\sum_{i=1}^{m} x_i(1 - 2v_i) + 1 \times \sum_{i=1}^{m} v_i$, where $X = x_1 \ldots x_m$ and $V = v_1 \ldots v_m$.*

The encryption scheme [10] allows us to generate a ciphertext C based on $\boldsymbol{a} = (a_1, \ldots, a_n)$ and a token TK based on $\boldsymbol{b} = (b_1, \ldots, b_n)$ such that given C and TK, we can compute $e(g, g)^{s[\sum_{i=1}^{n} a_i b_i]}$, where s is a random number, which gives $1_{\mathbb{G}_T}$ only when the inner product $\sum_{i=1}^{n} a_i b_i = 0$, or a random number otherwise. [10] is MC secure for the above inner product predicate which allows us devising encryption schemes based on the inner product expression which will be also MC secure. To evaluate whether $HammingDist(X, V) = t$, according to Lemma 1, we can check whether $e(g, g)^{s[\sum x_i(1-2v_i)+1\times(\sum v_i - t)]}$ equals $1_{\mathbb{G}_T}$ or not. Equivalently, we construct $f(X) = \boldsymbol{a} = (x_1, \ldots, x_m, 1)$ and $y(V, t) = \boldsymbol{b} = (1 - 2v_1, \ldots, 1 - 2v_m, \sum v_i - t)$.

The sizes of both ciphertext and token are $O(n)$ in [10] provided that $\boldsymbol{a}$ and $\boldsymbol{b}$ are n-length vectors, meaning that the sizes of both ciphertext and token in the above scheme are $O(m)$.

Security analysis: Our encryption scheme is MC secure. The proof is based on a reduction. Assume that there exits an adversary $\mathcal{A}_1$ that can win the MC game of our scheme with non-negligible advantage, we can use $\mathcal{A}_1$ as a subroutine to construct an adversary $\mathcal{A}_2$ that can win the MC game of the scheme in [10] with non-negligible advantage: When $\mathcal{A}_1$ outputs two vectors X_0 and X_1 to be challenged, $\mathcal{A}_2$ forwards $(X_0, 1)$ and $(X_1, 1)$ to the challenger. When $\mathcal{A}_1$ asks for a token query for (V, t) to $\mathcal{A}_2$, since $HammingDist(X, V) = t$ (or $\neq t$) corresponds to $\sum x_i(1 - 2v_i) + 1 \times (\sum v_i - t) = 0$ (or $\neq 0$), $\mathcal{A}_2$ is able to answer the query by asking the challenger a token query for $(1 - 2v_1, \ldots, 1 - 2v_m, \sum v_i - t)$. The detailed proof is omitted in this paper.

4 Scheme for Inequality Threshold

We borrow an idea from [6] to construct a generic solution for the case of having an inequality threshold. This solution is MC secure. The details of this generic solution are given in Appendix A. The ciphertext size of this solution is $O(t2^m)$ although the token size is constant which is not practical. In the following, we provide a practical scheme to handle the inequality threshold version.

If we can know the maximum value for the threshold t, t_{max}, in advance, we can have a scheme which is better than the generic solution. The size of the ciphertext can be reduced to $O(\sum_{i=0}^{t_{max}+1} \binom{m}{i})$. In some applications, $t_{max} << m$ and is a constant. In that case, the size becomes $O(m^{t_{max}})$. The restriction on setting t_{max} seems to be quite stringent. At the end of this section, we show how one can update the ciphertext if the user decides to increase t_{max} without computing ciphertext from scratch. We first present the scheme for known t_{max}.

The idea behind our construction is based on the observation that for Hamming distance H, $H \leq t$ if and only if $H(H-1)\dots(H-t) = 0$. Then, if we evaluate $\hat{e}(g,g)^{sH(H-1)\dots(H-t)}$ as `Test`() result where s is random, when $H \leq t$, `Test`() will be $1_{\mathbb{G}_T}$ (no information is leaked except from the fact that $H \leq t$); when $H > t$, $H(H-1)\dots(H-t) \neq 0$, `Test`() will output a random number (still no information is leaked except from the fact $H > t$ since `Test`() $\neq 1_{\mathbb{G}_T}$ computationally). Note that although evaluating $H(H-1)\dots(H-t)$ seems trivial in performance, it helps to ensure no information can be leaked which is required in the MC security.

Since the formula $H(H-1)\dots(H-t)$ where $H = \sum x_i(1-2v_i)+\sum v_i$ contains both information from ciphertext (i.e. knowledge of x_i) and token (i.e. knowledge of v_i and t) which cannot be available at the same time, we need to split the formula to these two parts (ciphertext and token). Recall that as we discussed in Section 3, we can split the formula to $f(X)$ and $y(V,t)$ whose inner product $f(X)\cdot y(V,t)$ provides the result for $H(H-1)\cdot\ldots\cdot(H-t)$. The following lemma expands $H(H-1)\cdot\ldots\cdot(H-t)$ so that we can find $f(X)$ and $y(V,t)$. We let

$$H(H-1)\cdot\ldots\cdot(H-t) = a_{t+1}H^{t+1} + a_tH^t + \ldots + a_1H \tag{1}$$

where we assume a_k $(k = 1,\dots,t+1)$ can be efficiently determined.

Lemma 2. *Given attribute $X = (x_1,\dots,x_m)$, target attribute $V = (v_1,\dots,v_m)$ and threshold t, we denote H as the Hamming distance $HammingDist(X,V)$ and define $a_0 = 0$ and b_j $(j = 0,\dots,t+1)$ as*

$$b_j = a_{t+1}\binom{t+1}{t+1-j}(\sum v_i)^{t+1-j} + a_t\binom{t}{t-j}(\sum v_i)^{t-j} + \ldots + a_j\binom{j}{0}. \tag{2}$$

Then, we have $H(H-1)\cdot\ldots\cdot(H-t)$

$$= \sum_{j=0}^{t+1} b_j\Big(\sum_{k_1+\ldots+k_m=j} \frac{j!}{k_1!\cdot\ldots\cdot k_m!}(1-2v_1)^{k_1}\cdot\ldots\cdot(1-2v_m)^{k_m}x_1^{k_1}\cdot\ldots\cdot x_m^{k_m}\Big). \tag{3}$$

Now, $H(H-1)\cdot\ldots\cdot(H-t)$ can be represented as inner product of $f(X)\cdot y(V,t)=\sum f_i(X)\cdot y_i(V,t)$. This is the key idea to our construction for inequality threshold. However, notice that $x_i\in\{0,1\}$, we would future reduce the number of items in Eq. (3) based on the observation that $x_1^{k_1}\cdot\ldots\cdot x_m^{k_m}=\prod_{\{i:k_i>0\}}x_i$ if $x_i\in\{0,1\}$. Then, Eq. (3) can be refined as: $H(H-1)\cdot\ldots\cdot(H-t)$

$$
\begin{aligned}
&= b_0+\sum_{1\le j\le m}\Big[\sum_{1\le k_1\le t+1} b_{k_1}(1-2v_j)^{k_1}\Big]x_j\\
&+\sum_{1\le j_1<j_2\le m}\Big[\sum_{k_1+k_2\le t+1;k_i\ge 1}\frac{(k_1+k_2)!}{k_1!k_2!}b_{k_1+k_2}(1-2v_{j_1})^{k_1}(1-2v_{j_2})^{k_2}\Big]x_{j_1}x_{j_2}\\
&+\ldots\\
&+\sum_{1\le j_1<\ldots<j_l\le m}\Big[\sum_{\substack{k_1+\ldots+k_l\le t+1;\\ k_i\ge 1}}\frac{(k_1+\ldots+k_l)!}{k_1!\ldots k_l!}b_{k_1+\ldots+k_l}(1-2v_{j_1})^{k_1}\ldots(1-2_{j_l})^{k_l}\Big]x_{j_1}\ldots x_{j_l}\\
&+\ldots\\
&+\sum_{1\le j_1<\ldots<j_{t+1}\le m}[(t+1)!b_{t+1}(1-2v_{j_1})\ldots(1-2v_{j_{t+1}})]x_{j_1}\ldots x_{j_{t+1}}.
\end{aligned}
\tag{4}
$$

For simplicity, we can denote Eq. (4) as below $H(H-1)\cdot\ldots\cdot(H-t)$:

$$
\begin{aligned}
&= B_0+B_1x_1+B_2x_2+\ldots+B_mx_m\\
&+B_{m+1}x_1x_2+\ldots+B_{m+\binom{m}{2}}x_{m-1}x_m\\
&+\ldots\\
&+B_{m+\binom{m}{2}+\ldots+\binom{m}{t}+1}x_1x_2\ldots x_{t+1}+\ldots+B_{m+\binom{m}{2}+\ldots+\binom{m}{t+1}}x_{m-t}\ldots x_m.
\end{aligned}
\tag{5}
$$

The number of items in Eq. (5) is $1+\binom{m}{1}+\ldots+\binom{m}{t+1}=\sum_{i=0}^{t+1}\binom{m}{i}$. We now describe a construction based on [10] and Eq. (5) whose ciphertext and token size are both $O(\sum_{i=0}^{t_{max}+1}\binom{m}{i})$.

Recall $(\texttt{Setup},\texttt{Enc},\texttt{GenKey},\texttt{Dec})$ in [10] can support n-dimension vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $C\xleftarrow{\$}\texttt{Enc}(\boldsymbol{a})$ and $TK\xleftarrow{\$}\texttt{GenKey}(\boldsymbol{b})$ where $\texttt{Dec}(C,TK)=1$ if and only if the inner product $\boldsymbol{a}\cdot\boldsymbol{b}=0$. In our construction, we let n be $\sum_{i=0}^{t_{max}+1}\binom{m}{i}$. Encryption algorithm $\texttt{Encrypt}(X=x_1,\ldots,x_m)$ in our construction calls $\texttt{Enc}()$ with input vector[1]:

$$\boldsymbol{a}=(1,x_1,\ldots,x_m,x_1x_2,\ldots,x_{m-1}x_m,\ldots,x_{m-t_{max}}\cdot\ldots\cdot x_m). \tag{6}$$

Token for V and t is generated by calling $\texttt{GenKey}()$ with input vector:

$$\boldsymbol{b}=(B_0,\ldots,B_{m+\ldots+\binom{m}{t}+1},\ldots,B_{m+\ldots+\binom{m}{t+1}},0\ldots,0). \tag{7}$$

[1] Note that although there exists x_1,x_2 and x_1x_2 in $\boldsymbol{a}$, given ciphertext for x_1 and x_2, we cannot reuse x_1 and x_2 to compute x_1x_2. This is because bilinear map is able to do only one multiplication (on encrypted data), however, we have used this ability to combine ciphertext and token, therefore, such redundancy in $\boldsymbol{a}$ seems to be necessary.

Note that $\boldsymbol{a}$ and $\boldsymbol{b}$ are constructed according to Eq. (5) and therefore, the inner product of $\boldsymbol{a} \cdot \boldsymbol{b} = H(H-1) \cdot \ldots \cdot (H-t)$.

This construction has ciphertext and token both of size $O(n) = O(\sum_{i=0}^{t_{max}+1} \binom{m}{i})$, however, some items in the token are in fact "0" since t may be less than t_{max}; more specifically, $H(H-1)\ldots(H-t)$ is $t+1$ degree and items in $\boldsymbol{a}$ whose degree larger than $t+1$ (i.e. $x_1x_2\ldots x_{t+2}, \ldots, x_{m-t_{max}}\ldots x_m$) will have coefficient "0" in $\boldsymbol{b}$ (Eq. (7)). This allows us to further reduce the token size. To do so, we devise an encryption scheme slightly different from [10] such that $\boldsymbol{a}$ is still n-dimensional while $\boldsymbol{b}$ can be any n'-dimensional ($n' \leq n$) and decryption will output "1" if and only if the inner product $\sum_{i=1}^{n'} a_i b_i = 0$. We describe this construction as follows:

- $\texttt{Setup}(1^n)$. This algorithm is the same as $\texttt{Setup}(1^n)$ in [10].
- $\texttt{Encrypt}(pk, X = x_1 \ldots x_n)$. It is the same as $\texttt{Enc}(pk, X = x_1 \ldots x_n)$ in [10].
- $\texttt{GenTK}(pk, sk, V = v_1 \ldots v_{n'})$. Note that $n' \leq n$. It randomly selects $\{r_{1,i}, r_{2,i}\}_{i\in[1,n']}$ and f_1, f_2 from $\mathbb{Z}_N$. Then, it randomly selects Q'' from $\mathbb{G}_q$ and R'' from $\mathbb{G}_r$. It outputs token TK:

$$\left\{ \begin{array}{c} K_0 = Q''R'' \prod_{i=1}^{\boxed{n'}} h_{1,i}^{-r_{1,i}} h_{2,i}^{-r_{2,i}} \\ \left[K_{1,i} = g_p^{r_{1,i}} g_q^{f_1 v_i}, K_{2,i} = g_p^{r_{2,i}} g_q^{f_2 v_i}\right]_{i\in\boxed{[1,n']}} \end{array} \right\}.$$

- $\texttt{Test}(pk, TK, C)$. It computes $r = \hat{e}(C_0, K_0) \prod_{i=1}^{\boxed{n'}} \hat{e}(C_{1,i}, K_{1,i}) \hat{e}(C_{2,i}, K_{2,i})$. If $r = 1_{\mathbb{G}_T}$, it will output "1"; otherwise it outputs "0".

The above encryption scheme is MC secure provided that Assumption 1 in [10] holds. Although we cannot directly reduce (i.e., by a black-box way) the security of [10] to the security of the above scheme because K_0 in [10] is fixed for n-length (rather than for any $n' < n$ in our case), we are able to prove the security of the above scheme from scratch, following a similar idea as [10].

Applying the above encryption scheme instead of the original scheme of [10] to Eq. (6) and (7), we obtain the final construction Π_1. Note that the above scheme also makes our security analysis of the final construction much easier (see the full paper). Π_1 is described as follows.

- $\texttt{Setup}(1^n)$: The algorithm first runs $\mathcal{G}(1^n)$ to obtain $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then, it randomly selects g_p from $\mathbb{G}_p$, g_q from $\mathbb{G}_q$ and g_r from $\mathbb{G}_r$. It also randomly selects $\{h_{1,l,i}, h_{2,l,i}\}_{l\in[1,t_{max}+1], i\in[1,\binom{m}{l}]}$ from $\mathbb{G}_p$. Then it randomly selects h_3, h_4 from $\mathbb{G}_p$. It also randomly selects R, R_3, R_4 and $\{R_{1,l,i}, R_{2,l,i}\}_{l\in[1,t_{max}+1], i\in[1,\binom{m}{l}]}$ from $\mathbb{G}_r$. It outputs

$$pk = \left\{ \begin{array}{c} g_p, g_r, Q = g_q R, \\ \left[H_{1,l,i} = h_{1,l,i} R_{1,l,i}, H_{2,l,i} = h_{2,l,i} R_{2,l,i}\right]_{l\in[1,t_{max}+1], i\in[1,\binom{m}{l}]}, \\ H_3 = h_3 R_3, H_4 = h_4 R_4 \end{array} \right\}$$

and

$$sk = \left\{ \begin{array}{c} p, q, r, g_q, \\ \left[h_{1,l,i}, h_{2,l,i}\right]_{l\in[1,t_{max}+1], i\in[1,\binom{m}{l}]}, \\ h_3, h_4 \end{array} \right\}.$$

– $\texttt{Encrypt}(pk, X = x_1 \dots x_m)$: Encryption algorithm first randomly selects s, α, β from $\mathbb{Z}_N$ and $\{R'_{1,l,i}, R'_{2,l,i}\}_{l\in[1,t_{max}+1], i\in[1,\binom{m}{l}]}$, R'_3, R'_4 from $\mathbb{G}_r$. Then it outputs ciphertext C:

$$\left\{ \begin{array}{c} C_0 = g_p^s, \\ \left[C_{1,l,i} = H_{1,l,i}^s Q^{\alpha x_{j_1}\cdots x_{j_l}} R'_{1,l,i}, C_{2,l,i} = H_{2,l,i}^s Q^{\beta x_{j_1}\cdots x_{j_l}} R'_{2,l,i}\right]_{1\le j_1<\ldots<j_l\le m}^{l\in[1,t_{max}+1];}, \\ C_3 = H_3^s Q^\alpha R'_3, C_4 = H_4^s Q^\beta R'_4 \end{array} \right\}.$$

– $\texttt{GenTK}(pk, sk, V = v_1 \dots v_m, t)$: It randomly selects $\{r_{1,l,i}, r_{2,l,i}\}_{l\in[1,t+1], i\in[1,\binom{m}{l}]}$, r_3, r_4 and f_1, f_2 from $\mathbb{Z}_N$. Then, it randomly selects Q'' from $\mathbb{G}_q$ and R'' from $\mathbb{G}_r$. It outputs token TK:

$$\left\{ \begin{array}{cc} K_0 = Q''R''h_3^{-r_3}h_4^{-r_4}\prod_{l=1}^{t+1}\prod_{i=1}^{\binom{m}{l}} h_{1,l,i}^{-r_{1,l,i}} h_{2,l,i}^{-r_{2,l,i}}, & \\ K_{1,1,1} = g_p^{r_{1,1,1}} g_q^{f_1 B_1}, & K_{2,1,1} = g_p^{r_{2,1,1}} g_q^{f_2 B_1} \\ \dots & \dots \\ K_{1,1,m} = g_p^{r_{1,1,m}} g_q^{f_1 B_m}, & K_{2,1,m} = g_p^{r_{2,1,m}} g_q^{f_2 B_m} \\ K_{1,2,1} = g_p^{r_{1,2,1}} g_q^{f_1 B_{m+1}}, & K_{2,2,1} = g_p^{r_{2,2,1}} g_q^{f_2 B_{m+1}} \\ \dots & \dots \\ K_{1,2,\binom{m}{2}} = g_p^{r_{1,2,\binom{m}{2}}} g_q^{f_1 B_{m+\binom{m}{2}}}, & K_{2,2,\binom{m}{2}} = g_p^{r_{2,2,\binom{m}{2}}} g_q^{f_2 B_{m+\binom{m}{2}}} \\ \dots & \dots \\ K_{1,t+1,1} = g_p^{r_{1,t+1,1}} g_q^{f_1 B_{m+\ldots+\binom{m}{t}+1}}, & K_{2,t+1,1} = g_p^{r_{2,t+1,1}} g_q^{f_2 B_{m+\ldots+\binom{m}{t}+1}} \\ \dots & \dots \\ K_{1,t+1,\binom{m}{t+1}} = g_p^{r_{1,t+1,\binom{m}{t+1}}} g_q^{f_1 B_{m+\ldots+\binom{m}{t+1}}}, & K_{2,t+1,\binom{m}{t+1}} = g_p^{r_{2,t+1,\binom{m}{t+1}}} g_q^{f_2 B_{m+\ldots+\binom{m}{t+1}}} \\ K_3 = g_p^{r_3} g_q^{f_1 B_0}, K_4 = g_p^{r_4} g_q^{f_2 B_0} & \end{array} \right\}$$

– $\texttt{Test}(pk, sk, TK, C)$: It outputs "1" if $r = 1_{\mathbb{G}_T}$ and "0" otherwise, where

$$r = \hat{e}(K_0, C_0)\hat{e}(K_3, C_3)\hat{e}(K_4, C_4)\prod_{l=1}^{t+1}\prod_{i=1}^{\binom{m}{l}} \hat{e}(K_{1,l,i}, C_{1,l,i})\hat{e}(K_{2,l,i}, C_{2,l,i}).$$

The size of ciphertext is still $O(\sum_{i=0}^{t_{max}+1}\binom{m}{i})$ but the size of token is now $O(\sum_{i=0}^{t+1}\binom{m}{i})$ for threshold t. The security of the scheme is stated in Theorem 1 and proved in the full paper.

Theorem 1. *Our construction Π_1 in Section 4 is Selectively MC secure provided that Assumption 1 in [10] holds.*

Lastly, to show that it is feasible to compute the coefficients a_k $(k = 1, \dots, t+1)$ in Eq. (1), we have implemented an algorithm which, in fact, can calculate all elements of vector $\boldsymbol{b}$ in Eq. (7). For example, with input $m = 100$ and $t = 3$, it took about 16 seconds to calculate all elements on an Intel Core 2 Due E6750 2.66GHz CPU platform.

Increasing t_{max}: If the value α, β and s generated in $\texttt{Encrypt}()$ are kept by that user, the user can update the ciphertext to increase t_{max} without producing the ciphertext from scratch. When the maximum threshold is updated from t_{max} to T', the corresponding vector $\boldsymbol{a}$ in Eq. (6) also needs to be updated as:

$$\boldsymbol{a'} = (\boldsymbol{a}, x_1 \cdot \ldots \cdot x_{t_{max}+2}, \dots, x_{m-T'} \cdot \ldots \cdot x_m). \tag{8}$$

Recall that when the maximum threshold is t_{max}, $\boldsymbol{a}$ contains all items whose degree $l \leq t_{max}+1$. Thus, when the maximum threshold becomes T', we need to produce those items whose degree is within $t_{max}+2$ to $T'+1$, namely $x_{j_1}...x_{j_l}$ where $t_{max}+2 \leq l \leq T'+1$ and $1 \leq j_1 < \ldots < j_l \leq m$ in Eq. (8). This can be done without from scratch by calculating $C_{1,i} = H_{1,i}^s Q^{\alpha a_i''} R_{1,i}$ and $C_{2,i} = H_{2,i}^s Q^{\beta a_i''} R_{2,i}$ for $i = 1, \ldots, k$, given α, β and s. Where we denote vector $\boldsymbol{a''} = (a_1'', \ldots, a_k'')$ such that $\boldsymbol{a'} = (\boldsymbol{a}, \boldsymbol{a''})$.

The above update procedure can be shown to be MC secure. Intuitively, when $x_1, \ldots, x_m$ in $\boldsymbol{a}$ are determined, all items in $\boldsymbol{a}$ (also in $\boldsymbol{a'}$) have been determined since they are the multiplication of two or more items in $\{x_1, \ldots, x_m\}$. For any possible $t_{max} \geq 0$, $\boldsymbol{a}$ (and therefore $\boldsymbol{a'}$) contains $(x_1, \ldots, x_m)$ for sure. That means all terms including the one to be generated due to the increase in t_{max} has been fixed although they are not computed yet. Therefore, an adaptive attack will not work since it has no way to adaptively modify how the missing items are generated. The detailed proof is omitted in this paper.

In the worst case, $t_{max} = m$, the size of the ciphertext (and token) becomes $O(2^m)$. Although it is better than $O(m2^m)$ (since $t = m$) for the generic solution in Appendix A, it is not practical. So, this scheme should be used when t_{max} is small.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
3. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
4. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
5. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: CCS (2006)
6. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
7. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
8. Hacigümüş, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL over encrypted data in the database-service-provider model. In: SIGMOD (2002)
9. Jarrous, A., Pinkas, B.: Secure hamming distance based computation and its applications. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 107–124. Springer, Heidelberg (2009)

10. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
11. Park, H.-A., Kim, B.H., Lee, D.H., Chung, Y.D., Zhan, J.: Secure similarity search. In: GRC (2007)
12. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
13. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
14. Shi, E., Bethencourt, J., Chan, T.-H.H., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy (2007)
15. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE Symposium on Security and Privacy (2000)
16. van Liesdonk, P.: Anonymous and fuzzy identity-based encryption. Master's thesis, Eindhoven University (2007)
17. Wong, W.K., Cheung, D.W., Kao, B., Mamoulis, N.: Secure kNN computation on encrypted databases. In: SIGMOD (2009)

A A Generic Construction from [6]

The main idea is that we can generate a ciphertext vector $(C_0, C_1, \ldots, C_{t_{max}})$ for each possible $V \in \{0,1\}^m$. Given $j \in [0, t_{max}]$, if $HammingDist(X, V) \leq j$, C_j is an encryption of the message "1"; otherwise, C_j will be an encryption of "0". When we $\texttt{Test}()$ for a certain (V, t), we find the ciphertext vector of V and then decrypt the t-th element C_t in the vector. If $HammingDist(X, V) \leq t$, the decryption result should be "1". More specifically, Let $(\texttt{G}, \texttt{E}, \texttt{D})$ be an IND-CPA secure encryption scheme.

- $\texttt{Setup}(1^n)$: Run $\texttt{G}(1^n)$ to generate $\{(pk_{l,j}, sk_{l,j})\}_{l \in \{0,1\}^m, j \in [0,t_{max}]}$ for $(t_{max}+1)2^m$ times. Return the public-key pk as $\{pk_{l,j}\}_{l \in \{0,1\}^m, j \in [0,t_{max}]}$ and the secret key sk as $\{sk_{l,j}\}_{l \in \{0,1\}^m, j \in [0,t_{max}]}$.
- $\texttt{Encrypt}(pk, X = x_1 \ldots x_m)$: For each $l \in \{0,1\}^m$, return $(C_0, C_1, \ldots, C_{t_{max}})_l$ where

$$C_j = \begin{cases} \texttt{E}_{pk_{l,j}}(\text{"1"}) \text{ if } HammingDist(X, l) \leq j; \\ \texttt{E}_{pk_{l,j}}(\text{"0"}) \text{ otherwise.} \end{cases}$$

- $\texttt{GenTK}(pk, sk, V, t)$: Return $sk_{V,t}$ as the token TK.
- $\texttt{Test}(pk, TK, C)$: It first finds $(C_0, C_1, \ldots, C_m)_V$ and returns $\texttt{D}_{TK}(C_t)$.

The security of the above solution comes from the IND-CPA secure encryption scheme, see appendix A of [6] for more details.