Article ID: 0465-7942(2020)01-0030-11

Smooth Projective Hashing from DDH Asumption and d-linear Assumption

Zhai Jiaqi[1], Liu Jian[2], Chen Lusheng[1]

(1. *School of Mathematical Sciences, Nankai University, Tianjin* 300071, *China*;
2. *School of Cybersecurity, College of Intelligence and Computing, Tianjin University, Tianjin* 300350, *China*)

Abstract: Two new constructions of smooth projective hashing are presented over the languages defined by CCA2 secure public-key encryption schemes based on lossy trapdoor functions. One lossy trapdoor function is based on DDH assumption and the other is based on d-linear assumption.

Keywords: smooth projective hashing; lossy trapdoor functions; CCA2 secure
CLC number: TN918.1 **Document code:** A

0 Introduction

Cramer and Shoup[1] introduced a primitive called universal hash proof system. With this notion they gave a framework to construct adaptive chosen ciphertext secure public-key encryption scheme, which is based on their earlier works[2-3]. Some following works in this paradigm include[4]. A universal hash proof system can be viewed as a kind of non-interactive zero-knowledge proof system for a language. More precisely, there is a keyed function F_k defined over a set X and a projection function α defined over the key space. When an element x belongs to the language $L\subset X$, $F_k(x)$ can be efficiently computed using $\alpha(k)$ and a witness w proving that $x\in L$. While $x\in X\backslash L$, $\alpha(k)$ tells nothing about $F_k(x)$. The verifier chooses a random key k and sends $\alpha(k)$ to the prover. To prove that $x\in L$, the prover who holds (x,w) computes $F_k(x)$ via (x,w) and $\alpha(k)$, then sends $F_k(x)$ to the verifier. The verifier computes $F_k(x)$ via x, and k and checks whether it is equal to the value sent by the prover. Here the function value $F_k(x)$ is exactly the proof. F_k is called a smooth projective hash function. Smooth projective hashing(SPH) can also be applied to construct oblivious transfer protocol[5] and public-key cryptosystem resilient to key leakage[6]. In reference [7], Wee introduced the notion of dual projective hashing, and gave a construction of lossy trapdoor functions by using this notion.

Another application of the smooth projective hashing is to construct password-based authenticated key exchange(PAKE) protocol. A framework to construct such a protocol by using SPH is given in reference [8]. The aim of PAKE is to enable the participants in the protocol who share a low-entropy password to generate a common cryptographically-strong secret key. The PAKE protocol proposed in reference [8] is based on a special kind of smooth projective hashing. In detail, let $\mathcal{C}$ be a non-interactive non-malleable perfectly-binding commitment scheme, M be the message space, C

Received date: 2019-02-22
Foundation item: Supported by National Key Basic Research Program of China under Grand(2013CB834204); National Natural Science Foundation of China under Grand(61902276)
Biography: Zhai Jiaqi(1990-), male, native place: Tianjin, Ph D. E-mail: jqzhai@mail.nankai.edu.cn

be the set of all possible commitments, $X=C\times M$. The language L consists of all the pairs (c,m) such that c is a commitment of m, then a SPH defined over $X=C\times M$ with language L can yield a PAKE protocol.

Since any adaptive chosen ciphertext secure(CCA2 secure) public-key encryption scheme can be used as a non-interactive non-malleable perfectly-binding commitment scheme, the commitment scheme above can be replaced by a CCA2 secure encryption scheme. So the problem is to construct smooth projective hashing over the set with form $C\times M$, where M and C are the plaintext space and ciphertext space of a CCA2 secure encryption scheme respectively. In reference [8], two CCA2 secure schemes are considered and the smooth projective hash functions based on the two schemes are constructed. One is the Cramer-Shoup CCA2 secure encryption scheme based on the decisional Diffie-Hellman(DDH) assumption[2], the other is based on the quadratic residuosity assumption. Katz and Vaikuntanathan[9] described a public-key encryption scheme based on the hardness of learning with errors problem[10] which is CCA2 secure, and they gave a smooth projective hashing based on their encryption scheme. Combining this SPH with the paradigm in reference [8], they obtained the first PAKE protocol based on lattices. It is not only an application problem but also an interesting theoretical question to find out which kind of CCA2 secure encryption scheme can be used to construct smooth projective hashing.

Peikert and Waters[11] proposed a cryptographic primitive called lossy trapdoor functions(lossy TDFs) and used it to construct CCA2 secure encryption schemes. They also realized lossy TDFs based on the DDH assumption and learning with errors problem respectively, which lead to two CCA2 secure encryption schemes. Other hard computational problems employed to construct lossy TDFs include the quadratic residuosity assumption, the composite residue assumption, and the d-linear assumption[12]. CCA2 secure encryption schemes can be obtained by using these lossy TDFs in the paradigm of reference [11].

Here we present two new SPH over the languages defined by CCA2 secure public-key encryption schemes based on lossy trapdoor functions. One lossy trapdoor function we used is based on DDH assumption and the other is based on d-linear assumption. We note that because of the special structure of SPH, the main obstacle in construction is giving two different ways to compute the value of the SPH. One is the direct way which is the definition of the SPH. The other way is via the projective of the key and a witness. It is required that the projective reveals nothing about the SPH value. We have to introduce techniques to overcome this problem. One of the techniques we used is based on the idea of associative laws of multiplication between matrices, that is,

$$(AB)C=A(BC).$$

It works well in the matrix-based constructions of lossy TDF such as the ones based on DDH and d-linear assumptions.

In the following, we firstly give the definitions of SPH, lossy and all-but-one trapdoor functions. The construction of CCA2 secure public-key encryption scheme based on lossy TDFs[11] is necessary for our construction, we also describe it. Our main constructions of SPH are proposed. The first SPH is over the encryption scheme using the lossy TDFs[11] based on DDH assumption, and the other one utilizes the lossy TDFs in reference [12] based on d-linear assumption[4,13]. Then we make a conclusion.

1 Preliminaries

1.1 Smooth projective hashing

Before defining smooth projective hashing, we first give the definition of subset membership problem.

Let $X_n, L_n \subseteq \{0,1\}^{poly(n)}$ be such that $L_n \subset X_n$, and distributions $D(L_n)$ be over L_n, $D(X_n \backslash L_n)$ be over $X_n \backslash L_n$. Here X_n will be the domain of the SPH function. Let $W_n \subseteq \{0,1\}^{poly(n)}$ is a witness set. $R_n \subseteq X_n \times W_n$ is an NP-relation. $x \in L_n$ if and only if there exists $w \in W_n$ such that $(x, w) \in R_n$. There is a probabilistic polynomial-time algorithm such that on input 1^n it outputs an instance

$$\Lambda = (X_n, L_n, D(L_n), D(X_n \backslash L_n), W_n, R_n)$$

from a distribution I_n. There is a probabilistic polynomial-time algorithm which outputs $x \leftarrow D(L_n)$ along with a witness w such that $(x, w) \in R_n$. There is a probabilistic polynomial-time algorithm that samples $x \in X_n \backslash L_n$ according to $D(X_n \backslash L_n)$.

The subset membership problem $\boldsymbol{I} = \{I_n\}$ defined above is called hard if the random variables from $D(L_n)$ and $D(X_n \backslash L_n)$ are computationally indistinguishable. For simplicity we omit the subscribe "n" in the following and suppose the subset membership problem is hard.

Now we begin to define the SPH. Let G be a non-empty set and $\mathcal{F} = \{F_k\}_{k \in K}$ be a collection of hash functions from X to G. There is a key projection function $\alpha: K \times X \rightarrow S$, where K is the key space of the function family and S is the space of key projection. The tuple $(\mathcal{F}, K, X, L, G, S, \alpha)$ is a collection of smooth projective hash functions if the following properties are satisfied.

1) The uniform distribution over the key space K can be efficiently sampled. The mapping α and F_k can be efficiently computed.

2) If $x \in L$, the value $F_k(x)$ is uniquely determined by $\alpha(k, x)$ and x. Moreover, given x, $\alpha(k, x)$ and a witness w of x such that $(w, x) \in R$, then $F_k(x)$ can be efficiently computed.

3) If $x \in X \backslash L$, then the random variables $(x, \alpha(k, x), F_k(x))$ and $(x, \alpha(k, x), g)$ are statistically indistinguishable, where $x \leftarrow D(X \backslash L)$ and $k \leftarrow K, g \leftarrow G$ are sampled according to uniform distributions. This property is called smoothness.

In the following we describe a more concrete realization of hard subset membership problem where a CCA2 secure encryption scheme is used. Suppose $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a CCA2 secure encryption scheme, $\mathbf{M}$ is the plaintext space and $\mathbf{C}$ is the ciphertext space.

Let $X = \mathbf{C} \times \mathbf{M}$, $L = \{(c, m) \in X | \exists r, \text{ s.t. } c = \mathcal{E}(m; r)\}$ which means that $(c, m) \in L$ if and only if c is an encryption of m. Let W be all the random coins used in the encryption algorithm, and $R = \{((c, m), r) \in L \times W | r$ is the random coins used in the encryption$\}$. Let $D(L)$ and $D(X \backslash L)$ be the uniform distribution. Then $(X, L, D(L), D(X \backslash L), W, R)$ is a hard subset membership problem[8].

In this paper, we will also construct the SPH functions over such a domain. The CCA2 secure schemes we used are based on the paradigm in reference [11], where a new primitive called lossy trapdoor functions is used.

1.2 Lossy and all-but-one trapdoor functions

Lossy trapdoor functions and all-but-one trapdoor functions (ABO TDFs in short) were introduced by Peikert and Waters[11]. They are defined as follows.

A collection of (n, k)-lossy trapdoor functions is defined by a tuple of probabilistic polynomial-

time algorithms $(S_{inj}, S_{loss}, F_{ltdf}, F_{ltdf}^{-1})$ satisfying:

1) S_{inj} outputs (s,t) where s is a function index and t is the trapdoor. $F_{ltdf}(s,\cdot)$ computes the injective function $f_s(\cdot)$ over the domain $\{0,1\}^n$. $F_{ltdf}^{-1}(t,\cdot)$ computes $f^{-1}(\cdot)$.

2) S_{loss} outputs $(s,\perp)$ where s is a function index. $F_{ltdf}(s,\cdot)$ computes the function $f_s(\cdot)$ over the domain $\{0,1\}^n$ and the size of the image set is at most $2^r=2^{n-k}$.

3) The function indices output by S_{inj} and S_{loss} are computationally indistinguishable.

A collection of (n,k)-all-but-one trapdoor functions with branch collection $\mathfrak{B}=\{B_n\}_{n\in\mathbb{N}}$ is defined by a tuple of probabilistic polynomial-time algorithms $(S_{abo}, G_{abo}, G_{abo}^{-1})$ satisfying:

1) For any $b^*\in B_n$, $S_{abo}(b^*)$ outputs (s,t), where s is a function index and t is the trapdoor;

2) For any $b\in B_n$ and $b\neq b^*$, $G_{abo}(s,b,\cdot)$ computes an injective function $g_{s,b}(\cdot)$ over the domain $\{0,1\}^n$, and $G_{abo}^{-1}(t,b,\cdot)$ computes $g_{s,b}^{-1}(\cdot)$. $G_{abo}(s,b^*,\cdot)$ computes the function $g_{s,b^*}(\cdot)$ over the domain $\{0,1\}^n$ and the size of the image set is at most $2^r=2^{n-k}$.

3) For any probabilistic polynomial-time algorithm $A=(A_1,A_2)$ the following probability is negligible:

$$\mathbf{P}\{i\leftarrow\{0,1\},(b_0,b_1,\text{state})\leftarrow A_1,\ s\leftarrow S_{abo}(b_i),\ i'\leftarrow A_2(s,b_0,b_1,\text{state}):i'=i\}.$$

1.3 CCA2 secure encryption scheme from lossy and ABO TDFs

In reference [11], Peikert and Waters give a construction of CCA2 secure encryption scheme from lossy and ABO TDFs. Their construction involves a signature scheme and a family of universal hash functions. Universal hash function is a family of functions $\mathfrak{F}$ from X_F to Y_F such that for any $x_1,x_2\in X_F, x_2\neq x_2$, the probability that a uniform random $f\leftarrow\mathfrak{F}$ satisfying $f(x_1)=f(x_2)$ is $1/|Y_F|$.

Let $(Gen, Sign, Ver)$ be a strongly unforgeable one-time signature scheme and the verification keys are in $\{0,1\}^v$, $(S_{inj}, S_{loss}, F_{ltdf}, F_{ltdf}^{-1})$ be a collection of (n,k)-lossy trapoor functions, $(S_{abo}, G_{abo}, G_{abo}^{-1})$ be a collection of (n,k')-all-but-one trapdoor functions with branch collection $B=\{0,1\}^v$. Let

$$(n-k)+(n-k')=r+r'=n-\kappa$$

for some $\kappa(n)=\omega(n)$ and $\mathfrak{K}$ be a family of universal hash functions from $\{0,1\}^n$ to $\{0,1\}^l$ where $l\leqslant\kappa-2\log(1/\varepsilon)$ for some negligible ε.

Key Generation. $\mathcal{G}$ generates $(s,t)\leftarrow S_{inj}$, $(s',t')\leftarrow S_{abo}(0^v)$, and chooses a hash function $h\leftarrow\mathfrak{K}$ uniformly. Then the public key is $pk=(s,s',h)$ and secret is $sk=(t,t',pk)$.

Encryption. On input the public key pk and a message $m\in\{0,1\}^l$, the encryption algorithm $\mathcal{E}$ generates $(vk, sk_\sigma)\leftarrow Gen$ and $x\leftarrow\{0,1\}^n$ uniformly. The ciphertext is $c=(vk,c_1,c_2,c_3,\sigma)$, where

$$c_1=F_{ltdf}(s,x),\ c_2=G_{abo}(s',vk,x),\ c_3=m\oplus h(x),$$
$$\sigma\leftarrow Sign(sk_\sigma, c_1\|c_2\|c_3).$$

Decryption. On input a secret key $sk=(t,t',pk=(s,s',h))$ and a ciphertext $c=(vk,c_1,c_2,c_3,\sigma)$, the decryption algorithm $\mathcal{D}$ outputs $\perp$ if $Ver(vk,c_1\|c_2\|c_3,\sigma)$ fails. If not, it computes $x=F_{ltdf}^{-1}(t,c_1)$ and checks that

$$c_1=F_{ltdf}(s,x)\text{ and }c_2=G_{abo}(s',vk,x).$$

If not, it outputs $\perp$. Otherwise, it outputs $m=c_3\oplus h(x)$.

The above encryption scheme $(\mathcal{G},\mathcal{E},\mathcal{D})$ is CCA2 secure. With realizations of lossy-TDF and ABO TDF, we can obtain a CCA2 secure scheme under the above paradigm. We will construct SPH functions from such CCA2 secure schemes in the following. Firstly, we describe the realizations of lossy-TDFs and ABO TDFs.

2 Main constructions

2.1 Construction from DDH assumption

In this subsection, we give a construction of SPH from DDH assumption.

First we describe the DDH-based lossy and all-but-one trapdoor functions. Let $\mathbf{G}=(p,\langle G\rangle,g)$ be a multiplicative group in which the DDH problem is hard, where the order p is a prime, $\langle G\rangle$ is a description of an efficient algorithm of the multiplication in G and g is a generator of G. We assume the tuple $\mathbf{G}=(p,\langle G\rangle,g)$ is fixed and public.

S_{inj} and S_{loss}: Choose $r=(r_1,\cdots,r_n)\in\mathbf{Z}_n^p$ and $s=(s_1,\cdots,s_n,1)\in\mathbf{Z}_p^n\times\{1\}$ uniformly random, and let $V=r^ts\in\mathbf{Z}_p^{n\times(n+1)}$. Let $V=\{v_{i,j}\}_{n\times(n+1)}$, g^{V} denote $\{g^{v_{i,j}}\}_{n\times(n+1)}\in G^{n\times(n+1)}$. The output of S_{loss} is g^{V} and the output of S_{inj} is $(g^{\mathrm{V}+\mathrm{I}'},s)$, where $\mathbf{I}'\in\mathbf{Z}_p^{n\times(n+1)}$ is the identity matrix extenden by a zero column.

F_{ltdf}: On input an index $g^{\mathrm{Y}}\in G^{n\times(n+1)}$ and $x\in\{0,1\}^n$, it outputs $z=g^{x\mathrm{Y}}\in G^{n+1}$.

F_{ltdf}^{-1}: On input a function value z and a trapdoor s, F_{ltdf}^{-1} computes $a_j=z_j/z_{n+1}^{s_j}$. Let $x_j\in\{0,1\}$ be such that $a_j=g^{x_j}$. Then it outputs x.

The tuple $(S_{inj},S_{loss},F_{ltdf},F_{ltdf}^{-1})$ described above defines a collection of $(n,n-\log p)$-lossy TDFs. The following is the construction of the ABO TDFs. The branch collection is $B=\mathbf{Z}_p$.

S_{abo}: On input $b^*\in\mathbf{Z}_p$, S_{abo} generates $V=r^ts\in\mathbf{Z}_p^{n\times(n+1)}$ as S_{inj} does. Then it outputs the function index $g^{\mathrm{Y}}=g^{\mathrm{V}-b^*\mathrm{I}'}$ and the trapdoor $t=(s,b^*)$.

G_{abo}: On input (g^{Y},b,x) where $x\in\{0,1\}^n$, it outputs $z=g^{\mathbf{x}(\mathrm{Y}+b\mathrm{I}')}\in G^{n+1}$.

G_{abo}^{-1}: On input (t,b,y) where $t=(s,b^*)$ is the trapdoor, $b\in\mathbf{Z}_p$ and $b\neq b^*$, G_{abo}^{-1} computes $a_j=z_j/z_{n+1}^{s_j}$. Let $x_j\in\{0,1\}$ be such that $a_j=g^{(b-b^*)x_j}$. It outputs x.

The tuple $(S_{abo},G_{abo},G_{abo}^{-1})$ described above defines a collection of $(n,n-\log p)$-ABO TDFs.

The above construction of lossy and ABO TDFs give a CCA2 scheme, which implies a hard subset membership problem. We will construct a SPH over this subset membership problem.

Parameters. We use the same letters as above for the parameters. Let n be the security parameter, $\mathbf{G}=(p,\langle G\rangle,g)$ be a multiplicative group in which the DDH problem is hard. The vectors used in the construction of lossy TDF is denoted by r,s and r',s' for ABO TDF. Let $V=r^ts$, $V'=r''s'$. Note that r,s and V,V' are not the public parameters for the SPH function and nobody needs to know their values. The function indices output by S_{inj}, S_{abo} are denoted by g^{Y}, $g^{\mathrm{Y}'}$. Note that g^{Y} and $g^{\mathrm{Y}'}$ are fixed public parameters and may be used when computing the SPH value or projected key. Let $(Gen,Sign,Ver)$ be a strongly unforgeable one-time signature scheme and the verification keys are in $\mathbf{Z}_p$. Let $\mathcal{H}$ be a family of universal hash functions from $\{0,1\}^n$ to $\mathbf{Z}_p$ and $p=2^{n/3-\omega\log n}$.

Construction. The universal hashing we use is simply $\mathcal{H}=\{h_a\}_{a\in\mathbf{Z}_p^n}$: for any $x\in\{0,1\}^n$, $h_a(x)=xa^t$. It is easy to check that $\mathcal{H}$ is universal. A random $a\leftarrow\mathbf{Z}_p^n$ is chosen. Let $h_{\mathbf{a}}$ be the universal hash function used in the encryption scheme. So a is also a fixed public parameter of the SPH function.

The key space for the SPH family is $K=\mathbf{Z}_p^{2n+3}$. The projection set is $S=G^n$. For a key $k=(u_1,\cdots,u_{n+1},v_1,\cdots,v_{n+1},w)\in K$ and $(c,m)\in X$, the projection is

$$\alpha(k,c)=g^{Y\mathbf{u}^{\mathrm{t}}+(\mathbf{Y}'+v\mathbf{I})\mathbf{v}^{\mathrm{t}}+\mathbf{a}^{\mathrm{t}}w}\in G^n, \tag{1}$$

where $u=(u_1,\cdots,u_{n+1})\in\mathbf{Z}_p^{n+1}$, $v=(v_1,\cdots,v_{n+1})\in\mathbf{Z}_p^{n+1}$, vk is the first part of the ciphertext and $a\in\mathbf{Z}_p^n$ is the index of the universal hash function used in encryption which is selected during the public-key generation. We note that given g^{Y} and u then $z=g^{\mathrm{Y}u^t}$ can be calculated in the following way:

$$z_j=\prod_{1\leqslant j\leqslant n+1}(g^{y_{i,j}})^{u_j}.$$

To get $g^{\mathrm{Y}u^t+\mathrm{Y}'v^t+a^tw}$, just compute $g^{\mathrm{Y}u^t}$, $g^{\mathrm{Y}'v^t}$ and g^{xa^t}, the result is the entry-wise multiplication of the three parts. The input of the function does not include the plaintext part m, this is required by the proof of the security of the PAKE protocol in reference [8].

We now define the smooth projective hash functions $\{F_k\}_{k\in K}$. On input a key $(u_1,\cdots,u_{n+1},v_1,\cdots,v_{n+1},w)\in K$ and (c,m), the hash function is computed as follows.

We note that the hash function is only defined over the domain $\mathbf{C}\times\mathbf{M}$ where $\mathbf{C}$ is the ciphertext space. That is $c=(vk,c_1,c_2,c_3,\sigma)\in\mathbf{C}$ implies that $Ver(vk,c_1\|c_2\|c_3,\sigma)$ succeeds. If not the hash function outputs $\perp$. For simplicity, suppose that all the ciphertexts in the follow are verified by Ver. Here $c_1,c_2\in G^{n+1}$ are vectors, let $c_1=(c_1^{(1)},\cdots,c_1^{(n+1)})$, $c_2=(c_2^{(1)},\cdots,c_2^{(n+1)})$. After checking $Ver(vk,c_1\|c_2\|c_3,\sigma)$, the SPH function outputs

$$\Big(\prod_{1\leqslant j\leqslant n+1}(c_1^{(j)})^{u_j}\Big)\cdot\Big(\prod_{1\leqslant j\leqslant n+1}(c_2^{(j)})^{v_j}\Big)\cdot g^{(c_3\oplus m)w}. \tag{2}$$

The other way (via the projected key and a witness) to compute the hash function is as follows. On input the projection $p=(p_1,\cdots,p_n)\in G^n$ of a key, (c,m) and the random coins used in the encryption which is $x\in\{0,1\}^n$, the output is

$$\prod_{1\leqslant i\leqslant n}p_i^{x_i} \tag{3}$$

Here the set of the random coins is a witness proving that $(c,m)\in L$. Note that we do not use the value (c,m) here, but in general (c,m) is used to compute the output of hash function.

To prove the smoothness, we need the following lemma.

Lemma 1 Let $\mathbf{F}$ be a finite field, $A\in\mathbf{F}^{m\times n}$, $a\in\mathbf{F}^m$ be a column vector and be some linear combination of the column vectors in A, $b\in\mathbf{F}^n$ be a row vector and independent of the row vectors in A. Then for any fixed $y\in\mathbf{F}$,

$$\mathbf{P}_{x\leftarrow\mathbf{F}^n}\Big\{\sum_{1\leqslant j\leqslant n}b_jx_j=y\,\Big|\,Ax=a\Big\}=\frac{1}{|F|}.$$

In other words, the conditional distribution above is the uniform distribution over $\mathbf{F}$.

Proof First, for a fix $y\in\mathbf{F}$, we find a vector $x\in\mathbf{F}^n$ such that $Ax=a$ and $bx^t=\langle b,x\rangle=y$.

Let $L(A)$ be the vector space spanned by the row vectors of A, $L^{\perp}(A)$ be the orthogonal complement space of $L(A)$. Then $L(A)$ is also the orthogonal complement space of $L^{\perp}(A)$. So $b\notin L(A)$ implies that there exists $x_0\in L^{\perp}(A)$ such that $\langle b,x_0\rangle\neq 0$.

Suppose that $x_1\in\mathbf{F}^n$ is such that $Ax_1=a$. Then let

$$x=x_1+\frac{y-\langle b,x_1\rangle}{\langle b,x_0\rangle}x_0.$$

It is easy to check that $Ax=a$ and $bx^t=\langle b,x\rangle=y$.

Let $A'=\begin{pmatrix}A\\b\end{pmatrix}$ be fixed, and $z\in\mathbf{F}^{m+1}$ is a vector. By the property of the linear equation, if

there exists a solution to the equation $A'x=z$, then the number of the solutions is a constant.

Therefore, we conclude that for any $y\in\mathbf{F}$, on the condition $Ax=a$, the probability

$$\mathbf{P}_{x\leftarrow\mathbf{F}^n}\{\sum_{1\leqslant j\leqslant n}b_jx_j=y\mid Ax=a\}=\frac{\mathbf{P}_{x\leftarrow\mathbf{F}^n}\{A'x=\begin{pmatrix}a\\y\end{pmatrix}\}}{\mathbf{P}_{x\leftarrow\mathbf{F}^n}\{Ax=a\}}$$

is a constant, which is obviously $1/|\mathbf{F}|$.

Theorem 1 The above construction $\{F_k\}_{k\in K}$ is a collection of smooth projective hash functions.

Proof We can see that the uniform distribution over the key space K can be efficiently sampled and the mapping $\alpha, \mathbf{F}_k$ can be efficiently computed. The residual leakage of the lossy and all-but-one collection is

$$r+r'=(n-(n-\log p))+(n-(n-\log p))=2\log p,$$

and

$$\kappa=n-2-\log p=n-2(n/3-\omega(\log n))=n/3+\omega(\log n)\geqslant\omega\log n.$$

In our construction, $l=\log q$, so,

$$\kappa-l=n-2\log p-\log p=n-3(n/3-\omega\log n)=\omega\log n.$$

And

$$\omega\log n=2\log 2^{\omega\log n}=2\log(1/2^{-\omega\log n}),$$

where $2^{-\omega\log n}$ is some negligible function. So the CCA-security requirement is satisfied.

Correctness. We need to show that when $(c,m)\in L$ which means c is an encryption of m, the two ways to compute the smooth projective hash function give the same result. As the construction of the lossy and all-but-one TDF from DDH-hard groups,

$$c_1=g^{x\mathbf{Y}},\ c_2=g^{x(\mathbf{Y}'+vk\mathbf{I})}.$$

And $c_3=xa^t\oplus m$. So the output of the smooth projective hash function in (2) is

$$g^{x\mathbf{Y}u^t}g^{x\mathbf{Y}'(\mathbf{Y}'+vk\mathbf{I})v^t}g^{(xa^t\oplus m\oplus m)w}=g^{x\mathbf{Y}u^t+x(\mathbf{Y}'+vk\mathbf{I})v^t+xa^tw}. \tag{4}$$

On the other hand, consider the way computing SPH value using the projected key and the witness. By (1) we have

$$p=(p_1,\cdots,p_n)=g^{\mathbf{Y}u^t+(\mathbf{Y}'+vk\mathbf{I})v^t+a^tw}\in G^n, \tag{5}$$

taking (5) into (3) we get

$$\prod_{1\leqslant i\leqslant n}p_i^{x_i}=g^{x(\mathbf{Y}u^t+(\mathbf{Y}'+vk\mathbf{I})v^t+a^tw)}=g^{x\mathbf{Y}u^t+x(\mathbf{Y}'+vk\mathbf{I})v^t+xa^tw}. \tag{6}$$

It follows that when the input $(c,m)\in L$ the two ways to compute the smooth projective hash function give the same result.

Smoothness. Fixing $(c,m)\in X\backslash L$, the random variable $(\alpha(k),F_k(c,m))$ and $(\alpha(k),r)$ are statistically indistinguishable, where k and r are chosen uniformly random from K and G respectively. Let

$$A=(\mathbf{Y},\mathbf{Y}'+vk\mathbf{I}',a^t),\ b=(\log_g c_1,\log_g c_2,c_3\oplus m),$$

where $\log_g c_i\in\mathbf{Z}_p^{n+1}$ is the vector whose entries are the logarithms of the entries of $c_i, i=1,2$.

As the ciphertext has the form

$$(vk,g^{x\mathbf{Y}},g^{x(\mathbf{Y}'+vk\mathbf{I}')},xa^t\oplus m,\sigma),$$

we can see that $(c=(vk,c_1,c_2,c_3,\sigma),m)\in L$ if and only if there exists $x\in\{0,1\}^n$ such that $b=xA$, where the signature σ has been checked to be valid as X is defined.

Let $V=\{xA|x\in\{0,1\}^n\}$. It is easy to check that A is full-rank, $|V|=2^n$, $|L(A)|=p^n$.

$$\mathbf{P}\{b\in L(A)\mid (c,m)\notin L\}=\frac{\mathbf{P}\{b\in L(A),b\notin V\}}{\mathbf{P}\{b\notin V\}}=\frac{p^n-2^n}{p^{2n+3}-2^n}$$
$$<\frac{p^n}{p^{2n+3}}=p^{-(n+3)},$$

which is negligible. That is, $(c,m)\notin L$ is such that b is independent of the rows in A except for a negligible fraction of $p^{-(n+3)}$. We have the following two equations,

$$Ak^t=\log_g\alpha(k,c),\ \ bk^t=\log_g H_k(c,m),$$

where all operations are in the field $\mathbf{Z}_p$. On the condition that b is independent of the rows in A, by lemma 1, we know that $\log_g H_k(c,m)$ is uniformly distributed over $\mathbf{Z}_p$ given $\alpha(k,c)$. It follows that $H_k(c,m)$ is uniformly distributed over G.

2.2 Construction from *d*-linear assumption

In this subsection, we present another construction of SPH from d–linear assumption.

First, we describe a construction of lossy TDFs and ABO TDFs from which is based on the d–linear assumption. This construction is similar to the one in the above subsection, so the smooth projective function can be obtained in a similar way.

Let $(p,\langle G\rangle,g)$ be a multiplicative group in which the d–linear assumption holds, where the order p is a prime, $\langle G\rangle$ is a description of an efficient algorithm of the operation in G and g is a generator. $Rk_d(\mathbf{Z}_p^{n\times n})$ denotes the set of all $n\times n$ matrices over $\mathbf{Z}_p$ of rank d. We assume the $(p,\langle G\rangle,g)$ is fixed and public in the following. Here are some notations. If $g=(g_1,\cdots,g_n)^t\in G^n$ is a column vector and $M=(a_{ij})_{n\times n}\in\mathbf{Z}_p^{n\times n}$, then g^M denotes the column vector

$$g^M=(\prod_{1\leqslant j\leqslant n}g_j^{a_{1j}},\cdots,\prod_{1\leqslant j\leqslant n}g_j^{a_{nj}})^t.$$

If $S=(g_{ij})_{n\times n}\in G^{n\times n}$ and $e=(e_1,\cdots,e_n)\in\mathbf{Z}_p^n$ is a column vector, then S^e denotes the column vector

$$S^e=(\prod_{1\leqslant j\leqslant n}g_{1j}^{e_j},\cdots,\prod_{1\leqslant j\leqslant n}g_{nj}^{e_j})^t.$$

With the above definitions, it holds that $(g^M)^e=(g^e)^M=g^{Me}$.

Construction of lossy TDFs:

S_{inj} and S_{loss}: The algorithm S_{inj} chooses a matrix $M\leftarrow Rk_d(\mathbf{Z}_p^{n\times n})$ uniformly random and computes $S=g^M\in G^{n\times n}$. It outputs S as the function index and M as the trapdoor. Similarly, S_{loss} chooses a matrix $M\leftarrow Rk_d(Z_p^{n\times n})$ uniformly random and computes $S=g^M\in G^{n\times n}$. It outputs S as the function index.

F_{ltdf}: On input an index S and $x\in\{0,1\}^n$, it outputs $z=S^x$.

F_{ltdf}^{-1}: On input a function value z and a trapdoor M, it computes $h=z^{M^{-1}}$. Let $x_i\in\{0,1\}$ be such that $g^{x_i}=h_i,i=1,\cdots,n$. It outputs $x=(x_1,\cdots,x_n)^t$.

The tuple $(S_{inj},S_{loss},F_{ltdf},F_{ltdf}^{-1})$ described above defines a collection of $(n,n-d\log p)$–lossy TDFs.

The branch of the ABO TDFs is $\mathbf{Z}_p$. The construction is described as follows:

S_{abo}: On input $b^*\in\mathbf{Z}_p$, choose a matrix $A\leftarrow Rk_1(\mathbf{Z}_p^{n\times n})$. Let $M=A-b^*I_n\in\mathbf{Z}_p^{n\times n}$ and $S=g^M\in G^{n\times n}$. Then S_{abo} outputs the function index S and the trapdoor M.

G_{abo}: On input the function index S, the branch b and $x\in\{0,1\}^n$, output

$$g^{(M+bI_n)x}=S^x*g^{bx},$$

where "$*$" indicates the componentwise product.

G_{abo}^{-1}: On input the function value $z \in G^n$, the branch b and the trapdoor M, G_{abo}^{-1} computes $M + bI_n$. If $M + bI_n$ is not invertible, it output $\perp$. Otherwise it computes $h = (h_1, \cdots, h_n) = z^{(M+bI_n)^{-1}}$. Let $x_i \in \{0,1\}$ be such that $g^{x_i} = h_i, i = 1, \cdots, n$. It outputs $x = (x_1, \cdots, x_n) \in \{0,1\}^n$.

The tuple $(S_{abo}, G_{abo}, G_{abo}^{-1})$ described above defines a collection of $(n, n - \log p)$–ABO TDFs.

As the authors mentioned in reference [12], this class of ABO TDFs has more than one lossy branch, that is to say, the other lossy branch in the above setting is $b^* - \mathrm{Tr}(A)$. In their definition, the algorithm S_{abo} also outputs the lossy branch set. They also added the extra property of the ABO TDFs to guarantee the security proof in reference [11]. This method can also be adapted to this more general context. The property is that the adversary can not output another lossy branch given (b^*, σ), where b^* is sampled from the branch set and σ is a function index output by $S_{abo}(b^*)$. The ABO TDF we used in the above construction satisfies this extra property, the implicit encryption scheme is secure. We refer to reference [12] for more details.

With the above lossy and ABO TDFs, we get a CCA2 secure encryption scheme. So a hard subset membership problem is obtained and we give a construction of SPH over this subset membership problem.

Parameters. We use the same letters as above for the parameters. Let n be the security parameter and the group size $p = 2^{\frac{n - \omega \log n}{d+2}}$. We use $S = g^M \in G^{n \times n}$ to denote the function index of the lossy TDF and $S' = g^{M'}$ to denote the function index of the ABO TDF. $(Gen, Sign, Ver)$ is a strongly unforgeable one–time signature scheme and the verification keys are in the branch set $\mathbf{Z}_p$. The universal hash function used is also $\mathcal{H} = \{h_a\}_{a \in Z_p^n}$, where

$$h_a: \{0,1\}^n \to \mathbf{Z}_p, x \mapsto \langle a, x \rangle.$$

And a uniform random $a \leftarrow \mathbf{Z}_p^n$ is chosen. The parameters S, S', a are all fixed and public parameters of the SPH function.

Construction. We construct smooth projective hash functions based on the hard subset membership problem implied by the above lossy and ABO TDFs.

The key space for the SPH family is $K = \mathbf{Z}_p^{2n+1}$ and the projection set is $S = G^n$. For a key $k = (u_1, \cdots, u_n, v_1, \cdots, v_n, w) \in K$ and $(c, m) \in X$, the projection is

$$\alpha(k, c) = g^{uM + v(M' + vkI) + wa},$$

where $u = (u_1, \cdots, u_n) \in \mathbf{Z}_p^n$, $v = (v_1, \cdots, v_n) \in \mathbf{Z}_p^n$, vk is the first part of the ciphertext and a is the index of the universal hash function used in the encryption. It is clear that the mapping can be computed efficiently. Here is the construction of the smooth hash functions $\{F_k\}_{k \in K}$. Also the functions are only defined over the set $X = C \times M$, where

$$c = (vk, c_1, c_2, c_3, \sigma) \in C$$

implies $Ver(vk, c_1 \| c_2 \| c_3, \sigma)$ succeeds. Let $c_1 = (c_1^{(1)}, \cdots, c_1^{(n)})$ and $c_2 = (c_2^{(1)}, \cdots, c_2^{(n)})$. Then the value of $F_k(c, m)$ is

$$\Big(\prod_{1 \leqslant i \leqslant n} (c_1^{(i)})^{u_i}\Big) \cdot \Big(\prod_{1 \leqslant i \leqslant n} (c_2^{(i)})^{v_i}\Big) \cdot g^{w(c_3 \oplus m)}.$$

Given the projection $p = (p_1, \cdots, p_n) \in G^n$, (c, m) and a witness $x \in \{0,1\}^n$, the function value can be computed as $\prod_{1 \leqslant j \leqslant n} p_j^{x_j}$.

The proofs of the correctness and smoothness of the SPH family constructed above are similar

to that in section 2.1, so we omit it here.

2.3 Adjustment of the encryption scheme

The lossy and ABO TDFs we used in section 2.1 and 2.2 are not exactly the original ones in references [11-12]. In the original constructions, the group $(p,\langle G\rangle,g)$ is output by the algorithm S_{inj}, S_{loss} and S_{abo}, which means that S_{inj}, S_{loss} or S_{abo} outputs the group and a function index (and the trapdoor if there is one). However, in our construction, we need the entries in the two function indices of the encryption scheme be in the same group such that the multiplication between them can be operated. Hence, we adjust the key generation phase of the encryption scheme by first generating a group and then sampling the two function indices of the lossy and ABO TDFs.

We take the first construction for example and the other is similar. The key generation algorithm $\mathcal{G}$ works as follows. On input 1^n, $\mathcal{G}$ generates a group $(p,\langle G\rangle,g)$ satisfying the security requirements. Then it runs the algorithms S_{inj} and S_{abo} we described in section 2.1 to obtain the indices and trapdoors. We note that this adjustment does not influence the CCA2 security of the encryption scheme. The complete proof is identical to that in reference [11], so we just give a simple clarification about the different part. The two critical conditions in the proof of security from reference [11] are: 1) Property of the lossy TDF: given a function index, it is hard to tell whether it is a lossy function or an injective function; 2) Property of the ABO TDF: given a function index $\sigma\leftarrow S_{abo}(b^*)$, where $b^*\in\{b_0,b_1\}$ and $\{b_0,b_1\}$ is generated by the adversary, the adversary can not tell b^*. The two properties still hold in our construction. This mean that the proof in reference [11] still works here. We denote by G the group generated in the key generation phase, σ,σ' the lossy and ABO function indices respectively. To distinguish σ from a lossy one, the extra information can be used is σ'. Note that σ' is generated independently of σ which can be generated by the adversary himself. So the lossy index and injective index remain indistinguishable. Thus the first property holds. The second one can be illustrated similarly.

3 Concluding remarks

In this paper, we present a new application of some specific lossy TDFs. We show that the CCA2 secure scheme based on the lossy TDFs constructed from DDH and d-linear assumptions can be used to defined a domain language of SPH. In the constructions of SPH, the main obstacle is finding two different ways to compute the value of the SPH. One is the direct way which is the definition of the SPH. The other way is via the projective of the key and a witness. It is required that the "projective" reveals nothing about the SPH value. One of the techniques we used is based on the idea of associative laws of multiplication between matrices, that is, $(AB)C=A(BC)$. It works well in the matrix-based constructions of lossy TDF such as the ones based on DDH and d-linear assumptions.

Works can be done in improving this technique. As not all languages given by the CCA2 secure scheme based on lossy TDF can induce SPH in this way. For example, we find that the same method is not adapted for the second lossy TDFs given in reference [11] directly. Therefore, the encryption schemes that can be used to construct SPH need to have some special structures of the ciphertext as well as the plaintext. Portraying suitable CCA2 secure encryption schemes for SPH and exploring the relationships between these two notions are significant in the further work.

References

1 Cramer R, Shoup V. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-key Encryption (Lecture Notes in Computer Science, Vol 2 332)[M]. Berlin: Springer-Verlag, 2002: 45-64.

2 Cramer R, Shoup V. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack (Lecture Notes in Computer Science, Vol 1 462)[M]. Berlin: Springer-Verlag, 1998.

3 Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack[J]. SIAM Journal on Computing, 2004, 33(1): 167-226.

4 Shacham H. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants, IACR Cryptology ePrint Archive, 2007/074, 2007.

5 Kalai Y T. Smooth Projective Hashing and Two-message Oblivious Transfer (Lecture Notes in Computer Science, Vol 3 494)[M]. Berlin: Springer-Verlag, 2005: 78-95.

6 Naor M, Segev G. Public-key Cryptosystems Resilient to Key Leakage (Lecture Notes in Computer Science, Vol 5 677)[M]. Berlin: Springer-Verlag, 2009: 18-35.

7 Wee H. Dual Projective Hashing and Its Applications Lossy Trapdoor Functions and More (Lecture Notes in Computer Science, Vol 7 237)[M], Berlin: Springer-Verlag, 2012: 246-262.

8 Gennaro R, Lindell Y. A Framework for Password-based Authenticated Key Exchange (Lecture Notes in Computer Science, Vol 2 656)[M]. Berlin: Springer-Verlag, 2003: 524-543.

9 Katz J, Vaikuntanathan V. Smooth Projective Hashing and Password-based Authenticated Key Exchange from Lattices (Lecture Notes in Computer Science, Vol 5 912)[M]. Berlin: Springer-Verlag, 2009: 636-652.

10 Regev O. On lattices, learning with errors, random linear codes, and cryptography[J]. Journal of the ACM, 2009, 56(6): 1-40.

11 Peikert C, Waters B. Lossy trapdoor functions and their applications[J]. Proceeding of the 40th Annual ACM Symposium on Theory of Computing, 2008, 40: 187-196.

12 Freeman D M, Goldreich O, Kiltz E, et al. More constructions of lossy and correlation secure trapdoor functions [J]. Journal of Cryptology, 2013, 26(1): 39-74.

13 Hofheinz D, Kiltz E. Secure Hybrid Encryption from Weakened Key Encapsulation (Lecture Notes in Computer Science, Vol 4 622)[M]. Berlin: Springer-Verlag, 2007: 553-571.

基于DDH假设和*d*-线性假设的平滑投影散列

翟嘉祺[1], 刘 健[2], 陈鲁生[1]

(1. 南开大学 数学科学学院, 天津 300071; 2. 天津大学 智能与计算学部 网络安全学院, 天津 300350)

摘要: 给出了两个新的平滑投影散列的构造,这两个平滑投影散列输入语言是由两个基于有损陷门函数的CCA2安全公钥加密体制定义的,其中一个有损陷门函数是基于DDH假设构造的,另一个是基于*d*-线性假设构造的.

关键词: 平滑投影散列; 有损陷门函数; CCA2安全