SELECTIVE OPENING SECURE FUNCTIONAL ENCRYPTION

Yuanyuan Ji[1], Haixia Xu[2] and Peili Li[1]

[1]Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Information Security,
Institute of Information Engineering, CAS, Beijing, China
jiyuanyuan@iie.ac.cn, xuhaixia@iie.ac.cn, lipeili@iie.ac.cn

ABSTRACT

Functional encryption (FE) has more fine-grained control to encrypted data than traditional encryption schemes. The well-accepted security of FE is indistinguishability-based security (IND-FE) and simulation-based security (SIMFE), but the security is not sufficient. For example, if an adversary has the ability to access a vector of ciphertexts and can ask to open some information of the messages, such as coins used in the encryption or secret key in multi-key setting, whether the privacy of the unopened messages is guaranteed. This is called selective opening attack (SOA).

In this paper, we propose a stronger security of FE which is secure against SOA (we call SO-FE) and propose a concrete construction of SO-FE scheme in the standard model. Our scheme is a non-adaptive IND-FE which satisfies selective opening secure in the simulation sense. In addition, the scheme can encrypt messages of any bit length other than bitwise and it is secure against SOA-C and SOAK simultaneously while the two attacks were appeared in different model before. According to the different functionality f, our scheme can specialize as IBE, ABE and even PE schemes secure against SOA.

KEYWORDS

Functional encryption, Selective opening attack, Indistinguishability obfuscation, Deniable encryption

1. INTRODUCTION

Traditional encryption schemes provide rather coarse-grained access to encrypted data, because the receiver can get the message in its entirety if he possesses the right key or he can learn nothing without the secret key. Thus a new encryption scheme — functional encryption (FE), with much more fine-grained control, has been extensively studied. FE was introduced by Boneh, Sahai and Waters [13]. A FE scheme means one who owns SKf can decrypt the cipher of m to get the value of f(m). It requires that the user learns nothing other than f(m). There are two well-accepted security notions for FE: indistinguishable based security definition (IND-FE) and simulation based definition (SIM-FE) [13]. But the security can't

David C. Wyld et al. (Eds) : NETCOM, NCS, WiMoNe, CSEIT, SPM - 2015
pp. 115–130, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51610

satisfy people's needs because of the different modes of attack, here we consider selective opening attack.

Selective opening security had been first investigated to the traditional public key encryption field by Bellare, Hofheinz and Yilek [10] in 2009. In the public key encryption system, there are two kinds of selective opening attack (SOA). One is coin-revealing SOA (SOA-C), that is to say, if an adversary obtains a number of ciphertexts and then corrupts a subset of the senders, obtaining not only the corresponding messages but also the coins under which they were encrypted, then the unopened messages still remain privacy. The other is key-revealing SOA (SOA-K), which means if an adversary obtains a number of ciphertexts encrypted under different public keys, then the senders are asked to reveal a subset of the corresponding decryption keys, in this case it remains secure for the rest of the messages. Creating an encryption scheme secure against SOA has important practical meaning. Under the complex environment of cloud computing, distributed shares in a distributed file-system are allotted to different servers to perform a task, if a subset of the distributed servers are corrupted by an adversary who may get the encrypted messages as well as the randomness, then can messages under the other uncorrupted severs remain secure?

Achieving security against SOA is challenging but even so there has been some works to achieve the security goal ([5], [6], [8], [4], [9], [7]). There are two flavors of definitions to capture security under selective opening attacks: simulation-based selective opening security (SIM-SO) and indistinguishability-based selective opening security (IND-SO) [5]. Because IND-SO security notion requires that the joint plaintext distribution should be conditionally effective re-sampled, which restricts SOA security to limited setting, so we just concern SIM-SO security. SO secure PKE scheme had been investigated by Bellare et al. [5] in 2009. Bellare showed that any lossy encryption is able to achieve SO security. Later on, several other SOA secure PKE schemes had been constructed ([6],[9],[8]). In 2011, with the development of IBE, Bellare, Waters and Yilek [11] introduced SOA to IBE. In IBE, ciphertexts and secret keys SKID are generated according to the corresponding target identity ID, only the right SKID can open the ciphertexts and an adversary can make many key queries using the ID (different from the challenge ID) as input. Later, Junzuo Lai et al. [12] proposed a concrete CCA2 secure SO-IBE scheme. However, almost known SO-IBE schemes utilize the technology of one-side public openability which means these schemes have to encrypt bit by bit which is comparatively inefficient, and it is challenging to construct a SOA secure IBE scheme which is not bitwise.

FE schemes seems to be different from PKE or IBE, but it aims to keep the encrypted message secret even though the adversary can get some special information SKf. But if the adversary has more ability to open a part of the message and get the randomness used in the encryption, can the security of the unopened messages be kept?

[13] and [15] proved that the simulation secure FE can not be achieved in the standard model. So in this paper, we focus on the construction of IND-FE and simulation-based secure against SOA

1.1 Related Works

With the development of indistinguishability obfuscation (io), many difficult cryptography tasks can be achieved. In 2013, [16] proposed a concrete construction of functional encryption for all circuits. In their scheme, the SKf is generated by using indistinguishability obfuscation, at the same time, it uses double encryption of the same message as the ciphertext and statistical simulation soundness NIZK (SSS-NIZK) to get well-formed ciphertexts. With the help of io, their scheme can hide important process (decryption and compution) in the SKf. In 2014, Sahai and Waters [3] introduced a new technique: puncture programs. They proposed an effective method to transform the private key encryption to the public key encryption and they designed a deniable encryption scheme which had opened for 16 years [2]. In deniable encryption, if a sender is forced to reveal to an adversary both his message and the randomness under encryption, he should be able to provide a fake randomness and a fake message that will make the adversary believe the ciphertext is encryption of the fake message.

1.2 Our Contributions

The contribution of this work consists of the following two steps. We first propose a new security model of functional encryption secure against selective opening attacks (including coins and private keys), which we call SO-FE, and then propose a concrete construction of SO-FE scheme for general function without random oracle. In view of the impossiblility result of the SIM-FE in the standard model and the limitation of the IND-SO, the security of our scheme is indistinguishable based secure FE and simulation based secure against SOA.

In our scheme, we combine the coin-revealing selective opening security and key-revealing selective opening security owing to the special property of KeyGen process of FE. Before, SOA-C and SOA-K are mentioned in different scenes, specially, SOA-K is only used in the multi-key encryption, the feature of FE can make sure the key query even though ciphertexts are encrypted under the same public key.

The SO-FE scheme can be applied to the special situation, such as SO-IBE scheme, SO-ABE scheme, SO-PE scheme. Thus using io, we can get many encryption schemes secure against selective opening attacks. So far there are only SO-IBE schemes (ABE or PE scheme secure against SOA haven't be proposed). Moreover, all known SO-IBE schemes are bitwise, while our scheme can encrypt the message with any bit.

1.3 Our Technique

There are two difficult challenges in achieving this goal. The first is the corrupt query of coins in SOA-C process: when the adversary chooses a set I and asks to open the corresponding messages and randomness, how can the simulator provide the eligible randomness which is indistinguishable from the real one. The second is key queries in SOA-K process — a feature of FE security formalizations since [13], that allows the adversary to obtain the decryption key of any reasonable functionality f of his choice, but how to define reasonablity in SOA-based security model.

To solve the first problem, we adopt deniable encryption (DE, refer to section 2.2) which can output a fake random r_0 (satisfies DE_{Enc} (pk_{DE}, m_0, r_0) = C). The special property of DE can make sure the simulator generates a fake randomness to cheat the adversary that the opened coins match the opened ciphers and the opened messages.

To solve the second problem, we impose restrictions on the adversary's choice of functions that can be queried to the key generation. Here we define reasonable function.

Intuition. We start by giving an overview of the main ideas behind our SOA-based security definition. To convey the core ideas, it suffices to consider the simple case of X = m_1,m_2,f(m_1,m_2), ($m_i \in \{0,1\}$). Suppose that the adversary queries secret keys for function f. Now, recall that the IND-security definition guarantees that an adversary cannot differentiate between encryption of x_0 and x_1 as long as $f(x_0) = f(x_1)$ for every f. It is the only restriction of IND-security definition, in SOA security model, the above restricting of f is not enough since an adversary can learn part information of message by making corrupt query of I. For example, an adversary can make I = {1} query and know m_1, by using key query to f, it can learn f(m_1,m_2). In particular, if $f(m_1,0) \neq f(m_1,1)$, it is easy to guess the unopened message m_2. Obviously, it makes no sense in SOA-based security definition. So we make the limitation of f: if the input of f contains the element of set m[I], which is opened in the corrupt query phase, thus except those messages in m[I], no matter what other input it is, the value of f is equal. That is to say, if $\exists$ i subject to $x_i \in m[I]$, the value of $f(\cdots,x_i,\cdots)$ are equal ($\cdots$ can be any value). Bellow, we present a unified definition of reasonable function.

Reasonable Function. Let $M = \{m_1,\cdots,m_l\}$ and $X = \{x_1,\cdots,x_l\}$ be any message of message space M, M is the challenge message, $I = \{i_1,\cdots,i_t\} \subseteq \{1,\cdots,l\}$ is the query in the SOA-C process. Define:

$$m[I] = \{m_{i_1},\cdots,m_{i_t}\};\ X|_I = \{x_i \in (m[I] \cap X)\};\ X|_{\bar{I}} = X \diagup X|_I;$$

$< y_1,y_2,\cdots,y_l >$ denotes a permutation of the values $y_1,\cdots,y_l$ such that the value y_i is mapped to the k 'th location if y_i is the k' th input to f. Thus, $X = < X|_{\bar{I}}, X|_I >$.

Definition 1. (Reasonability). Let {f} be a set of functions $f \in F$. We say f is reasonable if $f < X|_{\bar{I}}, X|_I > = f < X'|_{\bar{I}}, X|_I >$ for $\forall$ X, X' $\in$ M.

What we want to emphasize is that the key query and the corrupt query influence each other. The query of keys can increase the knowledge of the adversary, which can affect the choice of I; the corrupt query of I can make the adversary learn more about the message and can affect the choice of functionality f. In our scheme, we impose restrictions on the sequence of queries (the key queries of f must be made after the corrupt query of I) to remove the affect of the key queries, at the same time, on the KeyGen phase we limit the choice of f to remove the affect of the corrupt query on the basis of the opened messages in m[I], because an adversary may choose some special f in view of m[I] which can leak the information of unopened messages.

2. PRELIMINARIES

2.1 Functional encryption

A functional encryption scheme for a functionality f is a tuple of four algorithms: Setup. This is a PPT algorithm that takes the security parameter as input. It outputs a public and master secret key pair (PK,MSK).

Key Generation. This is a PPT algorithm that takes the functionality f as input, master secret key MSK. It outputs a decryption key SKf.

Encryption. This is a PPT algorithm that takes as input a message m and the public parameter PK. It outputs the ciphertext C.

Decryption. This algorithm takes the ciphertext C and the decryption key SKf as input, and outputs f(m).

We utilize Garg et al.[16]'s construction of FE (dual system encryption):

Setup. Generate (PKa,SKa) ← SetupPKE, (PKb,SKb) ← Setup $_{PKE}$, crs ← Setup $_{NIZK}$ Key Generation(MSK,f). SKf = io(Pf) (refer to the following table).

Encryption(m). c = (c_1,c_2,π), where c_1 = Enc(PKa;m,r_1), c_2 = Enc(PKb;m,r_2), π is a NIZK proof of the fact that : ∃m,r_1,r_2 : c_1 = Enc(PKa;m,r_1) ∧ c_2 = Enc(PK_b;m,r_2).

Decryption. Compute SK_f(c).

$\mathcal{P}_f$: (SK_a, crs)
Input: $c = (c_1, c_2, \pi)$
a. Check π is valid NIZK proof: $Ver_{NIZK}(crs, c_1, c_2, \pi)$=1. If yes , countine; if not, $\perp$.
b. Compute $m = Dec(SK_a, c_1)$.
c. $f(m)$.

Table 1. Program $\mathcal{P}_f$

2.2 Deniable Encryption

An encryption scheme is deniable if the sender can generate fake randomness that will make the ciphertext looks like an encryption of a different plain message, thus to keep the real message private. A deniable encryption scheme contains the following algorithms:

Setup$_{DE}$. This is a PPT algorithm that takes the security parameter as input. It outputs a public and master secret key pair (pk_{DE}, sk_{DE}).

Enc$_{DE}$. This is a PPT algorithm that takes as input a message m and the public parameter pkDE, and outputs the ciphertext C.

Dec_{DE}. This algorithm takes C and the decryption key skDE as input, and outputs m. **Exp_{DE}.** This is a PPT algorithm that takes C,m0 as input. Output a fake random r_0 which satisfies EncDE(pk_{DE}, m_0, r_0) = C.

We utilize SW's [3] construction of DE:

Bellare et al. [4] had proved no binding encryption scheme is simulator-based SOA security. That is why we use deniable encryption to realize our scheme. Specially, we use Sahai and Waters' scheme [3] which proposed a construction of deniable encryption. The scheme is proved to be IND-CPA secure and one-bit message encryption by using the technology of puncture, but it is not hard to generalize one-bit to a message string.

$Setup_{DE}$. (pk $_{PKE}$, sk $_{PKE}$) ← Setup $_{PKE}$. F1 is a puncturable extracting PRF, F2 is a puncturable statistically injective PRF, F3 is a puncturable PRF and (K1,K2,K3) is the corresponding puncturable PRFs' keys. pk_{DE} = (io(P_{Enc}),io(P_{Exp})), sk_{DE} = sk $_{PKE}$.

Enc_{DE}. c = io(P_{Enc}) (m,r)

Dec_{DE}. m = Dec $_{PKE}$ (sk $_{DE}$,c).

Exp_{DE}. r_0 ← io(P_{Exp}) (c, m_0, s): Enc_{DE} (pk_{DE}, m_0, r_0) = c. (s is a randomness.)

$\mathcal{P}_{Enc}$:(K_1, K_2, K_3)	$\mathcal{P}_{Exp}$:(K_2, K_3)
Input:$(m, r = (r_1, r_2))$ **a.** Let $F_3(K_3, r_1) \oplus r_2 = (m', c', r')$ for proper strings m',c',r'. Then check wether m'=m and $r_1 = F_2(K_2, (m', c', r'))$. If yes, output c=c' and $\perp$; if no, runs **b**. **b.**let $x = F_1(K_1, (m, r))$. Output $c = Enc_{PKE}(pk, m, x)$	**Input**:(c, m, s) **a.** $\alpha = F_2(K_2, (m, c, PRG(s)))$ $\beta = F_3(K_3, \alpha) \oplus (c, m, PRG(s))$. **b.** Output $r = (\alpha, \beta)$

Table 2. Program $\mathcal{P}_{Enc}$ and $\mathcal{P}_{Exp}$

3. THE DEFINITION OF SO-FE

We now propose the security model of a functional encryption secure against selective opening attacks, we call SO-FE.

Definition 2. We define two games GameREAL and GameSIM (refer to the following table).

GameREAL:

Setup. The challenger runs the Setup algorithm of FE, generates (PK,MSK) and gives the public parameters to the adversary.

Challenge. The adversary chooses a meessage distribution. The challenger chooses a message M from the distribution, and encrypts M . The ciphertext C is sent to the adversary. Corrupt query. The adversary makes one query to corrupt over a set of I (I ⊂ {1,2,⋯ ,l}), the challenger returns the messages m[I] and randomness r[I] used in challenge phase corresponding to I.

Key Query. The adversary is allowed to issue Key generation queries. That is to say the adversary outputs the function f to the challenger (f is reasonable), then the challenger runs KeyGen on f to generate the corresponding private key SKf and sends SKf to the adversary.

Final. The adversary guesses M.

GameSIM:

Setup. The simulator generates (PK,MSK) and sends PK to the adversary.

Challenge. The simulator chooses a message M0 from the distribution, and encrypts M0 . The ciphertext C' is sent to the adversary which is indistinguishable with C in GameREAL. Corrupt query. The adversary makes one query to corrupt over a set of I, the simulator runs Oracle to get the messages m[I] ⊆ M in GameREAL and generates fake randomness r^*[I] which satisfy C ' [I] = EncFE(m[I],r^*[I]).

Key Query. The simulator runs KeyGen on f to generate SKf and sends SKf to the adversary.

Final. The adversary guesses M.

proc.$Game_{REAL}$	**proc.**$Game_{SIM}$
$(PK, MSK) \leftarrow \mathbb{K}(1^\lambda)$; $M \leftarrow \mathbb{M}(1^\lambda)$	
For$i = 1, \cdots, l$ do	
$r[i] \leftarrow R$	$M' \leftarrow \mathbb{M}(1^\lambda)$
$c[i] \leftarrow Enc(m[i], r[i])$	$I \leftarrow S_1(1^\lambda, pk, C')$
$I \leftarrow A_1(1^\lambda, pk, C)$	$\omega \leftarrow S_2(m[I])$
$\omega \leftarrow A_2(r[I], m[I])$	$f \leftarrow S_3(1^\lambda)$
$f \leftarrow A_3(1^\lambda, pk)$	return M
$SK_f \leftarrow KeyGen(f, MSK)$	
return M	

Table 3. The security Games for SO-FE

We define the advantage of the adversary in this SO-FE Game:

$$\mathrm{AdvSO{-}FE(A)} = |\Pr[\mathrm{Gamereal} \Rightarrow \mathrm{true}] - \Pr[\mathrm{GameSIM} \Rightarrow \mathrm{true}]|$$

A functional encryption scheme is secure against SOA if all polynomial time adversaries A have at most a negligible advantage in the Game.

Our scheme is post SO-FE, that is to say, the KeyGen queries of f must be made after the corrupt query of I. There are two reasons to explain why our scheme is asked to be post secure: one is to make sure the adversary choose the set of I without the help of the KeyGen queries. In the proof of the security, the simulator hope to run the adversary and utilize the rewind technology after the corrupt query hIi until the challenge cipher is not contain in I. The other is to make sure there is no leak about information of the challenge plaintext after the adversary receives SKf, because we restricy the choices of functions that can be queried based on I. The Specific reasons can refer to the proof of the security in section 5.

4. A CONSTRUCTION OF SO-FE

We now give our construction of SO-FE scheme. In fact, our construction is based on that of Garg et al.' FE scheme, mixed with SW' DE scheme. The dual public key encryption in FE is replaced with a dual DE.

Let $M = m_1, m_2, \cdots, m_l$ ($m_i \in \{0,1\}^n$), we have

$\text{Setup}_{\text{SO-FE}}$: The Setup algorithm first runs $\text{Setup}_{\text{NIZK}}$ to get crs and runs Setup_{DE} twice to get $(pk^{\alpha}_{DE}, sk^{\alpha}_{DE})$ $(\alpha = a, b)$, where $pk^{\alpha}_{DE} = (io(P^{\alpha}_{Enc}), io(P^{\alpha}_{Exp}))$.

$$PK = (pk^{a}_{DE}, pk^{b}_{DE}, crs)\ ;\ MSK = (sk^{a}_{DE}, sk^{b}_{DE}, K^{a}_{1}, K^{b}_{1}, K^{a}_{2}, K^{b}_{2}, K^{a}_{3}, K^{b}_{3}).$$

(We utilize the SW's DE scheme introduced in section 2, K_i^{α} (i = 1, 2, 3; α = a, b) are keys of F_1, F_2, F_3 in DE.)

$\text{Enc}_{\text{SO-FE}}$: ∀i = 1,⋯ ,l, α ∈ {a,b}, choose randomness $r^{\alpha}_{i} = (r^{\alpha}_{i,1}, r^{\alpha}_{i,2}) \leftarrow R$

Check if $r^{\alpha}_{i,1} = F_2(K^{\alpha}_{2}, F_3(K^{\alpha}_{3}, r^{\alpha}_{i,1}) \oplus r^{\alpha}_{i,2})$. If yes, choose randomness once again until the random does not satisfy the above condition.

$$c^{(a)}_{i} = io(P^{a}_{Enc})(m_i, r^{a}_{i})$$
$$c^{(b)}_{i} = io(P^{b}_{Enc})(m_i, r^{b}_{i})$$

Creat a NIZK proof $\pi_i \leftarrow \Pr ove_{NIZK}(crs, (c^{(a)}_{i}, c^{(b)}_{i}), (r^{a}_{i}, r^{b}_{i}, m_i))$ to prove the fact that:

$$\exists m_i, r^{a}_{i}, r^{b}_{i} : c^{(a)}_{i} = Enc(PK_a; m_i, r^{a}_{i}) \wedge c^{(\ )}_{i} = Enc(PK_b; m_i, r^{b}_{i}).$$

$C_i = (c^{(a)}_{i}, c^{(b)}_{i}, \pi_i)$. Let the encryption of M is C = (C1,⋯ ,Cl).

$\text{KeyGen}_{\text{SO-FE}}$: Create an obfuscation of the program like the following Table 3, and output SKf = io(PKeyGen). DecSO−FE: Compute SKf (C).

$\mathcal{P}_{KeyGen}$: $(K^{a}_{1}, K^{b}_{1}, crs)$
Input: C
a. For $\forall i = 1, \cdots, l$, check wether $Ver_{NIZK}(crs, c^{(a)}_{i}, c^{(b)}_{i}, \pi_i)$=1. If yes , countine; if not, $\perp$.
b. $\forall i = 1, \cdots, l$, compute $m'_i = Dec_{PK}(sk^{a}_{DE}, c^{(a)}_{i})$.
c. $f(m'_1, \cdots, m'_l)$.

Table 4. Program $\mathcal{P}_{KeyGen}$

5. THE SECURITY OF SO-FE

The SO-FE scheme in section 4 is a SIM-SO FE scheme, the security model is given in section 3. Now we will give the security proof.

Theorem 1. If io is an indistinguishability obfuscator, DE is IND-CPA security and the NIZK is statistically simulation sound, the scheme is a no-adaptive secure SO-FE.

Proof. In order to prove the FE scheme is SIM-SO security, we need to construct a simulator which can run in the GameSIM to simulate all the possibility in the GameREAL. That is to say,

$$|\Pr(\text{GameREAL} \Rightarrow \text{true}) - \Pr(\text{GameSIM} \Rightarrow \text{true})| \leq \text{neg}(\cdot).$$

In short, the simulator needs to create equivocable ciphertexts as the challenge ciphertexts, then open them accordingly. Here, we must make sure the equivocable ciphertexts are indistinguishable from the real encryption of the messages in the REAL setting. In order to provide the environment of the adversary in GameREAL, on the corrupt phase, the simulator first gets the corrupt messages from the Oracle in the GameSIM and then outputs the fake randomness which is indistinguishable from the real random used in the encryption to the adversary (here we use the technology of DE).

we proof the theorem through a series of Hybrids:

Hybrid 0: Let A be an arbitrary adversary in GameREAL of the SO-FE security model. The challenger first generates (PK, MSK) and send the public key to to the adversary. Then the challenger chooses the message M from the message space M and encrypt the message running $\text{Enc}_{\text{SO–FE}}$. Later the adversary makes a corrupt query and some key generation queries, the challenger sends m[I],r[I] to A (r[I] is the real random used in encryption of m[I]). Finally, A give its guess of the message.

We can see Pr(Hybrid0 ⇒ true) = Pr(GameREAL ⇒ true)

proc. Initialize	**proc.**Ch(M)	**proc.**KeyGen(f)	**proc.**Corrupt(I)	**proc.**Finalize
$(PK, MSK) \leftarrow Setup$ return PK	$M \leftarrow \mathcal{M}$, for $i = 1, \cdots, l$ $r_i \leftarrow R$ $c_i = Enc(m_i)$ returen c	return SK_f	return m[I],r[I]	return M'

Table 5. Hybrid $0/Game_{REAL}$

Hybrid 1: We define Hybrid 1 to be the same as Hybrid 0, except that on the corrupt phase, the challenger first runs the Oracle in GameSIM to get the message m[I], for i ∈ |I|,α = {a,b}, set sαi ← R, riα = io(PExpα)(mi,ciα ,sαi). Output r[i] = (ria,rib). (cαi is the cipher generated by simulator, mi is the output of Oracle).

proc. Initialize	**proc.**Ch(M)	**proc.**KeyGen(f)	**proc.**Corrupt(I)	**proc.**Finalize
$(PK, MSK) \leftarrow Setup$ return PK	$M \leftarrow \mathcal{M}$, for $i = 1, \cdots, l$ $r_i \leftarrow R$ $c_i = Enc(m_i)$ returen c	return SK_f	$m[I] \leftarrow S^{Oracle}$ for $i \in \|I\|, \alpha = \{a, b\}$ $s_i^\alpha \leftarrow R$ $r_i^\alpha = io(P_{Exp})(m_i, c_i^\alpha, s_i^\alpha)$ return $m_i, r_i = (r_i^a, r_i^b)$	return M'

Table 6. Hybrid 1

We now say |Pr(Hybrid0 ⇒ true) – Pr(Hybrid1 ⇒ true)| ≤ neg(·), because the random returned in Hybrid 1 and Hybrid 0 are almost identically distributed in the view of A. The indistinguishability between Hybrid0 and Hybrid1 can reduce to the explainability of DE scheme.

In [3], Sahai and Waters had proved the explainability of deniable encryption: if the io is indistinguishable and F1 is a puncturable extracting PRF, F2 is a puncturable statistically injective PRF, F3 is a general puncturable PRF, then the generated pseudo-randomness is indistinguishable with the real random. While in Hybrid 0, the encrypted randomness is chosen from set {0,1}|r|/S,(S = {(a,b)|a = $F_2(K_2, F_3(K_3,a) \oplus b)$,a = {0,1}|$r_1$|,b = {0,1}|$r_2$|}). Now we can see the size of S: for any fixed a, there exist at most one preimage a0 because of F2 is a puncturable statistically injective PRF, thus b = $a_0 \oplus F3(K_3,a)$ is well-determined. So |S| = 2|r_1| and choose a random from S is negligible if r is large enough.

Hybrid 2: We define Hybrid 2 is the same with Hybrid 1 except that on the KeyGen query phase, the challenger returns $\widehat{SK_f}(\widehat{SK_f} = io(\widehat{P}_{KeyGen})$ is defined as follows). Our scheme is no-adaptive security, the KeyGen query is made after the challenge phase. It's easy to see SK[f and SKf is indistinguishable . So |Pr(Hybrid1 ⇒ true)−Pr(Hybrid2 ⇒ true)| ≤ neg(·).

The indistinguishability between Hybrid1 and Hybrid2 can reduce to the indistinguishability of io.

$\widehat{\mathcal{P}}_{KeyGen}$: (K_1^a, K_1^b, crs, C^*)
Input:$C = c_1, \cdots, c_l$
a. For $\forall i = 1, \cdots, l$, check wether $Ver_{NIZK}(crs, c_i^{(a)}, c_i^{(b)}, \pi_i)$=1. If yes , countine; if not, $\perp$.
b. If $c_i \in C^*$, $m_i' \leftarrow Oracle(i)$; if not, $m_i' = Dec_{DE}(c_i^{(a)}, sk_{DE}^a)$.
c. $f(m_1', \cdots, m_l')$.

Table 7. Program $\widehat{\mathcal{P}}_{KeyGen}$

Hybrid 3−p:(0 ≤ p ≤ q) We define Hybrid 3−p is the same with Hybrid 2 except that on the challenge phase, if i ≤ p, we replace the real challenge cipher to new ones which are generate by simulater, (here specially the simulator choose messages mi = 1n and send the ciphers to A); If p < i ≤ q, the simulate sends the real challenge cipher to A.

We can see Pr(Hybrid3−0 ⇒true) =Pr(Hybrid2 ⇒true) and Pr(Hybrid3−q ⇒true) =Pr(GameSIM ⇒ true). So our aim is to prove |Pr(Hybrid3−0 ⇒ true) – Pr(Hybrid3−q ⇒ true)| ≤ neg(·). We define the Hybrid3−p is like the following table 7.

proc. Initialize	**proc.**Ch(M)	**proc.**KeyGen(f)	**proc.**Corrupt(I)	**proc.**Finalize
$(PK, MSK) \leftarrow Setup$ return PK	$M \leftarrow \mathcal{M}$, for $i = 1, \cdots, l$ if $i < p, c_i = Enc(1^n)$ if $i = p, c_i = c^*$ if $i > p, c_i = Enc(m_i)$ returen c	return $\widehat{SK_f}$	$m[I] \leftarrow S^{Oracle}$ for $i \in \|I\|, \alpha = \{a, b\}$ $s_i^\alpha \leftarrow R$ $r_i^\alpha = io(P_{Exp})(c_i^\alpha, m_i, s_i^\alpha)$ return $m_i, r_i = (r_i^a, r_i^b)$	return M'

Table 8. Hybrid $3_{-p}(0 \le p \le q$ and Hybrid $3_{-q}/Game_{SIM})$

Now we begin to explain the indistinguishability between Hybrid3−p and Hybrid3−(p−1). To prove the above problem, we first define the following hybrids and then reduce the indistinguishability to security of IND-CPA DE.

Hybrid3−(p−1)−(0): This hybrid is the same with Hybrid3−(p−1).

Hybrid3−(p−1)−(1): This hybrid uses the trapdoor in NIZK to generate an fake proof to make sure that the adversary can believe two ciphertexts in double system encryption is an encryption of the same message.

$$\pi^* \leftarrow Sim_{NIZK}(1^\lambda, \exists m, r_a, r_b : c_a^* = Enc(pk_{DE}^a, m, r_a) \wedge c_b^* = Enc(pk_{DE}^b, m, r_b)).$$

Hybrid3−(p−1)−(2): This hybrid change the pth ciphertext to $(c_p^a, c_p^b, \pi_p)^*$, where

$c_p^a* = Enc(pk_{DE}^a, m_p, r_a)$, $c_p^b* = Enc(pk_{DE}^a, 1^n, r_b)$, π_p* is a fake proof generated by

$Sim_{NIZK.}$

Hybrid3−(p−1)−(3): This hybrid is the same with Hybrid3−p−(2) except that the pth ciphertext is $(c_p^a, c_p^b, \pi_p)^*$, where $c_p^a* = Enc(pk_{DE}^a, 1^n, r_a)$, $c_p^b* = Enc(pk_{DE}^a, 1^n, r_b)$, and on the io of KeyGen query phase, we replace K_1^a to K_1^b and make sure we can use the key in the second part of the double encryption system. It's not hard to see Hybrid3−(p−1)−(3) ≈ Hybrid3−p.

If SSS-NIZK is computationally zero knowledge, then Hybrid3−(p−1)−(0), Hybrid3−(p−1)−(1) is indistinguish. For the indistinguishability between (1) and (2) or (2) and (3), we hope to reduce the problem to the IND-CPA secure DE. That is to say we hope to structure a simulator B who can run A, if there is an A who can distinguish (1) and (2) or (2) and (3), there is an adversary B who can distinguish the challenge cipher c□ in Game of IND-CPA DE. The reduction can refer to appendix. So

$$\left|Pr(Hybrid3_{-p} \Rightarrow true) - Pr(Hybrid3_{-(p-1)} \Rightarrow true)\right| \leq neg(\cdot)$$

$$\begin{aligned}
&|Pr(Hybrid3_{-0} \Rightarrow true) - Pr(Hybrid3_{-q} \Rightarrow true)| \\
&\leq |Pr(Hybrid3_{-0} \Rightarrow true) - Pr(Hybrid3_{-1} \Rightarrow true)| + \cdots \\
&\qquad + \left|Pr(Hybrid3_{-(q-1)} \Rightarrow true) - Pr(Hybrid3_{-q} \Rightarrow true)\right| \\
&\leq neg(\cdot).
\end{aligned}$$

6. CONCLUSION

Our paper proposed a stronger security of FE which is secure against SOA and proposed a concrete construction of SO-FE scheme. A lot of work is worth doing in the future, for example, how to concrete a SO-FE without indistinguishability obfuscation.

ACKNOWLEDGEMENTS

We would like to thank all workers who have helped us to make the paper better.

REFERENCES

[1] Mihir Bellare, Dennis Hofheinz, Scott Yilek: Possibility and impossibility results for encryption and commitment secure under selective opening. EUROCRYPT 2009. LNCS, vol. 5479, pp. 1-35. Springer, Heidelberg (2009)

[2] Ran Canetti, Cynthia Dwork, Moni Naor and Rafi Ostrovsky: Deniable Encryption. CRYPTO. Cryptology ePrint Archive, Report 1996/002. pp 90-104. (1997)

[3] Amit Sahai and Brent Waters: How to Use Indistinguishability Obfuscation: Deniable Encryption, and More. STOC 2014. Cryptology ePrint Archive, Report 2013/454. pp 475-484, (2014)

[4] Mihir Bellare, Rafael Dowsley, Brent Waters, Scott Yilek: Standard security does not imply security against selective-opening. EUROCRYPT 2012. LNCS, vol. 7237, pp. 645-662. Springer, Heidelberg (2012)

[5] Mihir Bellare, Dennis Hofheinz, Scott Yilek: Possibility and impossibility results for encryption and commitment secure under selective opening. EUROCRYPT 2009. LNCS, vol. 5479, pp. 1-35. Springer, Heidelberg (2009)

[6] Serge Fehr, Dennis Hofheinz, Eike Kiltz, Hoeteck Wee: Encryption schemes secure against chosen-ciphertext selective opening attacks. EUROCRYPT 2010. LNCS, vol. 6110, pp. 381-402. Springer, Heidelberg (2010)

[7] Zhengan Huang, Shengli Liu, Baodong Qin: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. PKC2013. LNCS, vol. 7778, pp. 369-385. Springer, Heidelberg (2013)

[8] Brett Hemenway, Benoit Libert, Rafail Ostrovsky, Damien Vergnaud: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. ASIACRYPT 2011. LNCS, vol. 7073, pp. 70-88. Springer, Heidelberg (2011)

[9] Dennis Hofheinz: All-but-many lossy trapdoor functions. EUROCRYPT 2012. LNCS, vol. 7237, pp. 209-227. Springer, Heidelberg (2012)

[10] Mihir Bellare, Scott Yilek: Encryption schemes secure under selective opening attack. IACR Cryptology ePrint Archive, 2009:101 (2009)

[11] Mihir Bellare, Brent Waters, Scott Yilek: Identity-based encryption secure against selective opening attack. TCC 2011. LNCS, vol. 6597, pp. 235-252.Springer, Heidelberg (2011)

[12] Junzuo Lai, Robert H. Deng, Shengli Liu,Jian Weng, Yunlei Zhao : Identity-Based Encryption Secure against Selective Opening Chosen-Ciphertext Attack. EUROCRYPT 2014. LNCS, vol. 8441, pp 77-92. Springer, Heidelberg (2014)

[13] Dan Boneh, Amit Sahai, Brent Waters: Functional Encryption: Definitions and Challenges. LNCS, vol. 6597, pp 253-27 (2011)

[14] Florian B\ddot{o}hl, Dennis Hofheinz, Daniel Kraschewski: On definitions of selective opening security. PKC 2012. LNCS, vol. 7293, pp. 522-539. Springer, Heidelberg (2012)

[15] Mihir Bellare, Adam O'Neill: Semantically - secure functional encryption: Possibility results, impossibility results and the quest for a general definition. Cryptology ePrint Archive, Report 2012:515 (2012)

[16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai and Brent Waters: Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits. FOCS 2013, IEEE Computer Society. pp 40-49 (2013)

[17] Dan Boneh and Brent Waters: Constrained pseudorandom functions and their applications. IACR Cryptology ePrint Archive, 2013:352. (2013)

APPENDIX

A. Puncturable PRF

A puncturable family of PRFs F mapping ($\{0,1\}^{n(\cdot)} \rightarrow \{0,1\}^{m(\cdot)}$) is given by a triple of Turing Machines (Key_F,$Puncture_F$,$Eval_F$) satisfying the following conditions:

Functionality preserved. For every PPT adversary A such that A(1λ) outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$, then we have

$$Pr\left[Eval_F(K,x) = Eval_F(K_S,x) \middle| \begin{array}{l} K \leftarrow Key_F(1^\lambda) \\ K_S = Puncture_F(K,S) \\ \forall x \in \{0,1\}^{n(\lambda)} \wedge x \notin S \end{array}\right] = 1$$

Pseudorandom at punctured points.For every PPT adversary A such that $A(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$ and state σ, consider an experiment where K ← KeyF(1^λ) and KS = PunctureF(K,S), for any PPT distinguisher D, we have

$$|Pr[D(\sigma,KS,S,EvalF(K,S)) = 1] - Pr[D(\sigma,KS,S,Um(\lambda)\cdot|S|) = 1]| \leq neg(\lambda)$$

Definition 3. A puncturable statistically injective PRF family with failure probability ε(·) is a family of PRFs F such that with probability 1 − ε(λ) over the random choice of key K ← KeyF(1λ), we have that F(K,) is injective.

Definition 4. A puncturable extracting PRF family with error ε(·) for min-entropy k(·) is a family of PRFs F mapping {0,1}n(λ) → {0,1}m(λ) such that for all λ, if X is any distribution over {0,1}m(λ) with min-entropy greater than k(λ), then the statistical distance between (K ← KeyF(1λ),F(K,X)) and (K ← KeyF(1λ),Um(λ)) is at most ε(λ).

B. Indistinguishability Obfuscator

A uniform PPT machine io is called an indistinguishability obfuscator (io) for a circuit family {Cλ} if the following conditions are satisfied:

Functionality preserved. For all security parameters $\lambda \in N$, for all $C \in \{C_\lambda\}$, for all input x, we have

Pr[C0(x) = C(x) : C0 ← io(λ,C)] = 1

Indistinguishability. For any PPT distinguisher D, for all security parameters $\lambda \in N$, for all pairs of circuits C0,C1 $\in \{C_\lambda\}$ which satisfies $Pr[\forall x, C_0(x) = C_1(x)] > 1-neg(\cdot)$, then

$|Pr[D(io(\lambda, C_0)) = 1] - Pr[D(io(\lambda, C_1)) = 1]| \leq neg(\lambda)$

C.NIZK

A non-interactive zero-knowledge proof system (NIZK) contains three algorithms NIZK = (Setup,Prove,V er): crs ← Setup(1^k);π ← Prove(crs, stmt, ω);b ← V er(crs, stmt, π), where k is the security parameter, crs is the common reference string, stmt is the statement information, ω is a witness and π is the proof, moreover b is 0/1 means rejection or acceptance.

Completeness. Pr[crs ← Setup,π ← Prove(crs,stmt,ω): V er(crs,stmt,π) = 1] = 1

Soundness. Pr[crs ← Setup,∃(stmt,π) : (stmt /∈ L) ∧ V er(crs,stmt,π) = 1] ≤ neg(·)

Zero-knowledge. If there exists a simulator S=(SimSetup,SimProve),such that for all PPT adversary A, it holds that

$$\left| Pr\begin{bmatrix} crs \leftarrow Setup(1^k, m) \\ \forall i \in [m] : \pi_i \leftarrow Prove(crs, stmt_i, \omega_i) \\ \mathcal{A}(crs, \{stmt_i, \pi_i\}_{i\in[m]}) = 1 \end{bmatrix} - Pr\begin{bmatrix} (\widetilde{crs}, trap) \leftarrow SimSetup(1^k, \{stmt_i\}_{i\in[m]}) \\ \forall i \in [m] : \pi_i \leftarrow SimProve(crs, stmt_i, trap) \\ \mathcal{A}(\widetilde{crs}, \{stmt_i, \pi_i\}_{i\in[m]}) = 1 \end{bmatrix} \right|$$

is negligible.

In [16], the FE scheme used statistically simulation sound NIZK, which they called SSS-NIZK, and Garg et al. proposed a concrete construction of SSS-NIKZ. Informally, a NIZK system is statistically simulation sound, if under a simulated crs, there is no valid

proof for any false statement, except for the simulated proofs for statements fed into the SimSetup algorithm to generate crs. That is to say, f

$$Pr\begin{bmatrix} (\widetilde{crs}, trap) \leftarrow SimSetup(1^k, \{stmt_i\}_{i\in[m]}) \\ \forall i \in [m] : \pi_i \leftarrow SimProve(crs, stmt_i, trap) \\ \exists (stmt', \pi') s.t. (stmt' \notin \{stmt_i\}_{i\in[m]}) \wedge Ver(crs, stmt, \pi) = 1 \end{bmatrix} \leq neg(\cdot)$$

D. Reduct to IND-CPA DE

Here we will explain the indistinguishability between Hybrid3−(p−1)−(1) and Hybrid3−(p−1)−(2) or Hybrid3−(p−1)−(2) and Hybrid3−(p−1)−(3). We hope to structure a simulator B who can run A, if there is an A who can distinguish (1) and (2) or (2) and (3), there is an adversary B who can distinguish the challenge cipher c^* in Game of IND-CPA DE (refer to the following figures).

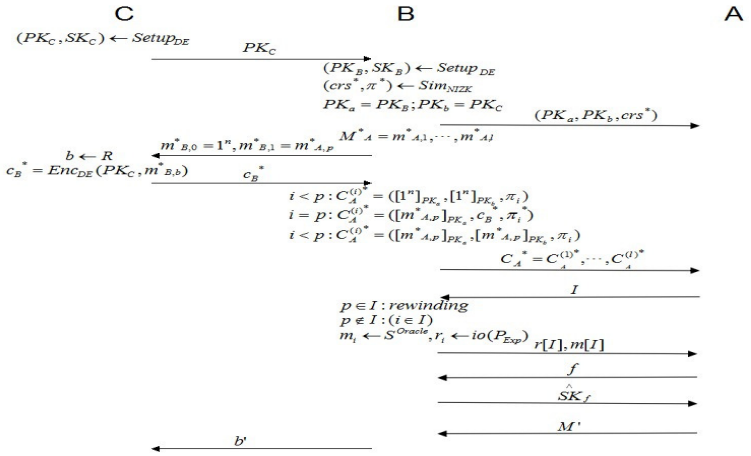

Fig.1.The reduction process: the indistinguishability between Hybrid3−(p−1)−(1) and Hybrid3−(p−1)−(2). [m]PK means encryption of m with public key PK.

Take Hybrid3−(p−1)−(2) and Hybrid3−(p−1)−(3) for example:

B gets PKC from the challenger from IND-CPA game of DE, then sets PKa = PKC and generates a pair of key (PK_b,SK_b). B sends (PK_a,PK_b) to adversary A in the SOAGame FE. B chooses message $M^*A = m^*A,1,\cdots ,m^*A,l$ from message space and makes the challenger's challenge message $m^*_{\mathcal{B},0} = 1^n, m^*_{\mathcal{B},1} = m^*_{\mathcal{A},\mathbb{P}}$. The challenger will return a

challenge cipher c^*B. Then B hides the c^*B into the challenge cipher $C^*_{\mathcal{A}} = C^{(1)*}_{\mathcal{A}}, \cdots, C^{(l)*}_{\mathcal{A}}$ of A in the following way:

$$\text{if } i < p, C^{(i)*}_{\mathcal{A}} = (Enc^a_{DE}(1^n), Enc^b_{DE}(1^n), \pi_i);$$

$$\text{if } i = p, C^{(i)*}_{\mathcal{A}} = (c^*_{\mathcal{B}}, Enc^b_{DE}(1^n), \pi^*_i);$$

$$\text{if } i > p, C^{(i)*}_{\mathcal{A}} = (Enc^a_{DE}(m_p), Enc^b_{DE}(m_p), \pi_i)$$

When A makes corrupt query hIi: B first check whether p ∈ I, if yes, B uses the rewind technology to repeatedly run A until p ∉ I; if not, B makes pseudo randomness using $io(P_{Exp})$ after knowing the message m[I].

When A make key generate queries hfi(q-bounded): B replaces K^a_1 to K^b_1 in the SK[f to decrypt and make sure we can use the key in the second part of the double encryption system. Then send it to A.

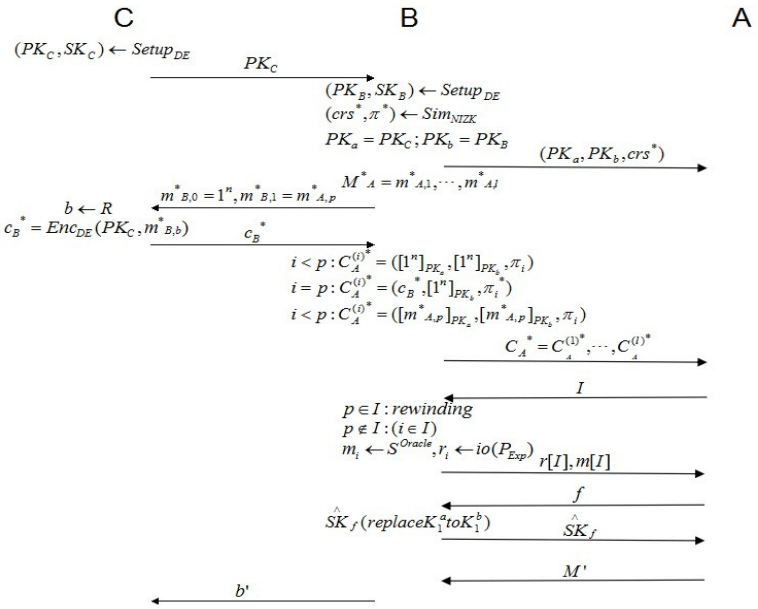

Fig.2. The reduction process: the indistinguishability betwee n Hybrid3−(p−1)−(2) and Hybrid3−(p−1)−(3)

Finally A will output its guess M0, then the B can utilize the pth guess to reply the challenger in Game DE. So if A can guess the message rightly, thus B can distinguish between the cipher of m0 or m1 with non-negligible advantage, which will break the INDCPA property of DE.

AUTHORS

Yuanyuan Ji, was born in henan, China, on Nov. 10, 1989. She is studying for a master's degree at the university of Chinese academy of sciences, Beijing, China