DE GRUYTER
OPEN

Int. J. Appl. Math. Comput. Sci., 2015, Vol. 25, No. 2, 415–430
DOI: 10.1515/amcs-2015-0032

amcs

SENDER–EQUIVOCABLE ENCRYPTION SCHEMES SECURE AGAINST CHOSEN–CIPHERTEXT ATTACKS REVISITED

ZHENGAN HUANG [a], SHENGLI LIU [a,*], BAODONG QIN [a,b], KEFEI CHEN [c,d]

[a]Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai, 200240, China
e-mail: zhahuang.sjtu@gmail.com,slliu@sjtu.edu.cn

[b]College of Computer Science and Technology, Southwest University of Science and Technology
59 Qinglong Road, Mianyang, Sichuan, 621010, China
e-mail: qinbaodong@sjtu.edu.cn

[c]School of Science, Hangzhou Normal University
16 Xuelin Street, Xisha Higher Education Zone, Hangzhou, 310036, China

[d]State Key Laboratory of Mathematical Engineering and Advanced Computing
30 Lianze Road, Building #18, Science and Education Industry Park, Binhu District, Wuxi, 214000, China
e-mail: kfchen@sjtu.edu.cn

Fehr *et al.* (2010) proposed the first sender-equivocable encryption scheme secure against chosen-ciphertext attacks (NC-CCA) and proved that NC-CCA security implies security against selective opening chosen-ciphertext attacks (SO-CCA). The NC-CCA security proof of the scheme relies on security against substitution attacks of a new primitive, the “cross-authentication code”. However, the security of the cross-authentication code cannot be guaranteed when all the keys used in the code are exposed. Our key observation is that, in the NC-CCA security game, the randomness used in the generation of the challenge ciphertext is exposed to the adversary. Based on this observation, we provide a security analysis of Fehr *et al.*'s scheme, showing that its NC-CCA security proof is flawed. We also point out that the scheme of Fehr *et al.* encrypting a single-bit plaintext can be refined to achieve NC-CCA security, free of the cross-authentication code. Furthermore, we propose the notion of “strong cross-authentication code”, apply it to Fehr *et al.*'s scheme, and show that the new version of the latter achieves NC-CCA security for multi-bit plaintexts.

Keywords: sender-equivocable encryption, chosen-ciphertext attack, cross-authentication code.

1. Introduction

The notion of sender equivocability for a public-key encryption (PKE) scheme was formalized by Fehr *et al.* (2010). It is an important tool to construct PKE schemes secure against chosen-plaintext/ciphertext selective opening attacks (SO-CPA/CCA). Sender equivocability focuses on the ability of a PKE scheme to generate some “equivocable” ciphertexts which can be efficiently opened arbitrarily. More specifically, a PKE scheme is called sender-equivocable if there is a simulator which can generate non-committing ciphertexts and later open them to any requested plaintexts by releasing some randomness, such that the simulation and real encryption are indistinguishable. This notion is similar to non-committing encryption (Canetti *et al.*, 1996). In fact, Fehr *et al.* (2010) pointed out that sender-equivocable encryption secure under chosen-plaintext attacks (CPAs) is a variant of non-committing encryption defined by Canetti *et al.* (1996). Following the notation in the work of Fehr *et al.* (2010), the security of a sender-equivocable encryption scheme against chosen-plaintext/ciphertext attacks is denoted by *NC-CPA/CCA security*.

As proved by Fehr *et al.* (2010), NC-CPA/CCA security implies simulation-based selective opening

[*]Corresponding author

security against chosen-plaintext/ciphertext attacks (SIM-SO-CPA/CCA security). This fact suggests an alternative way of constructing PKE secure against selective opening attacks, besides the construction from the lossy encryption proposed by Bellare *et al*. (2009).

1.1. Discussion and related work.

Bellare *et al*. (2009) formalized the notion of security against selective opening attacks (SOA security) for sender corruptions. This security notion captures a situation that n senders encrypt their own messages and send the ciphertexts to a single receiver. Some subset of the senders can be corrupted by an adversary, exposing their messages and randomness to the latter. SOA security requires that the unopened ciphertexts remain secure.

Bellare *et al*. (2009) proposed two kinds of SOA security: simulation-based selective opening (SIM-SO) security and indistinguishability-based selective opening (IND-SO) security. The relations between the two notions are figured out by Böhl *et al*. (2012). Bellare *et al*. (2012) showed that the standard security of PKE does not imply SIM-SO security. Bellare *et al*. (2009) proposed that IND-SO-CPA security and SIM-SO-CPA security can be achieved through a special class of encryption named lossy encryption, which can be constructed from lossy trapdoor functions (Peikert and Waters, 2011). Hemenway *et al*. (2011) showed more constructions of lossy encryption, which achieved IND-SO-CCA security with an *a-priori* bounded number of challenge ciphertexts. Hofheinz (2012) proposed a new primitive called all-but-many lossy trapdoor functions, which were employed to construct IND-SO-CCA secure and SIM-SO-CCA secure PKE with an unbounded number of challenge ciphertexts. Bellare *et al*. (2011) extended SOA security from PKE to IBE.

Fehr *et al*. (2010) presented a totally different way of achieving SIM-SO-CCA security, also with an unbounded number of challenge ciphertexts. They formalized the security notion of sender equivocability under chosen-plaintext/ciphertext attacks (NC-CPA/CCA security), and proved that NC-CPA (resp. NC-CCA) security implies SIM-SO-CPA (resp. SIM-SO-CCA) security. In the work of Fehr *et al*. (2010), two PKE schemes were proposed. The first one, constructed from trapdoor one-way permutations, is NC-CPA secure, so it is SIM-SO-CPA secure. The second one (denoted as the FHKW scheme) is constructed from an extended hash proof system (Cramer and Shoup, 2002) and a new primitive, the "cross-authentication code". They proved that the FHKW scheme is NC-CCA secure.

With the help of similar techniques as those in the FHKW scheme, Gao *et al*. (2012) presented a deniable encryption scheme. The CCA security of their scheme was guaranteed mainly by an extended hash proof system (Cramer and Shoup, 2002) and a cross-authentication code (Fehr *et al*., 2010).

In this paper, we will analyze the security proof of the FHKW scheme and show that its NC-CCA security cannot be guaranteed by their proof. The GXW scheme suffers from a similar security problem. Then, we will offer a refined version of the FHKW scheme for a single bit with NC-CCA security. To completely fix the problem, we will introduce the strong notion of cross-authentication code, apply it to the FHKW scheme, and show that the new version of the FHKW scheme achieves NC-CCA security for multi-bit plaintexts.

1.2. Our contribution.

In this paper, we focus on NC-CCA security. First, we provide an analysis of the security proof of the FHKW scheme (Fehr *et al*., 2010), and show that the proof of NC-CCA security (Fehr *et al*., 2010) is flawed by showing an attack. The key observation is that, in the definition of NC-CCA security, the randomness used in the generation of the challenge ciphertext C^* is offered to the adversary. The adversary is able to use the randomness to forge a ciphertext and obtain useful information by querying the forged ciphertext to the decryption oracle. Assume that the plaintext consists of L bits. We present a PPT adversary who can always distinguish the real experiment and the simulated experiment for $L > 1$. We also show that the security requirement of "L-cross-authentication codes" is not enough for the NC-CCA security proof in the work of Fehr *et al*. (2010) for any positive integer L.

Second, we refine the FHKW scheme encrypting one bit. Although we showed that "L-cross-authentication codes" are generally not sufficient to prove NC-CCA security, some specific instances of "1-cross-authentication codes" are helpful to finish the proof of NC-CCA security of the FHKW scheme (Fehr *et al*., 2010), but limited to encryption of a single bit. We provide a simpler encryption scheme for single-bit plaintexts, free of any cross-authentication code.

Third, we fix the security proof of the FHKW scheme by introducing the strong notion of an L-cross-authentication code and using it to construct the FHKW scheme instead of the original one. Informally, a strong L-cross-authentication code requires the existence of a PPT algorithm to generate another key indistinguishable from the original one. With this property, the randomness in the simulated experiment is different but indistinguishable from that in the real experiment, which helps the L-cross-authentication code's security against substitution attacks work again.

Organization. We start with the notation and definitions in Section 2. We recall the FHKW scheme in Section 3, and then provide its security analysis in Section 4. We present a refined version of the FHKW scheme for single-bit plaintexts in Section 5 and leave the proof for Appendices. We introduce the notion of a strong cross-authentication code in Section 6, and use it to fix the security proof in Section 7. Finally, we give a summary of our work in Section 8.

2. Preliminaries

2.1. Notation. Let $\mathbb{N}$ denote the set of natural numbers. We use $k \in \mathbb{N}$ as the security parameter throughout the paper. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$ and $\{0,1\}^n$ the set of bitstrings of length n. For a finite set S, let $s \leftarrow S$ denote the process of sampling s uniformly at random from S. If A is a probabilistic algorithm, we denote by $\mathcal{R}_A$ the randomness set of A. Let $y \leftarrow A(x_1, x_2, \ldots, x_t)$ denote the process of running A on inputs $\{x_1, x_2, \ldots, x_t\}$ and inner randomness $R \leftarrow \mathcal{R}_A$, and outputting y. If the running time of probabilistic algorithm A is polynomial in k, then A is a probabilistic polynomial time (PPT) algorithm.

2.2. Sender-equivocable encryption schemes. The notion of sender equivocability was formalized by Fehr *et al.* (2010). For a public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, let $A = (A_1, A_2)$ denote a stateful adversary, $S = (S_1, S_2)$ denote a stateful simulator, and M denote a plaintext. Let *state* denote some state information output by A_1 and then passed to A_2. Sender equivocability under adaptive chosen-ciphertext attacks is defined through the following two experiments.

```
Experiment $\mathbf{Exp}_{\Pi,A}^{\text{NC-CCA-Real}}(k)$:
    $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
    $(M, state) \leftarrow A_1^{\mathsf{Dec}_{sk}(\cdot)}(pk)$
    $R \leftarrow \mathcal{R}_{\mathsf{Enc}}$
    $C \leftarrow \mathsf{Enc}_{pk}(M; R)$
    return $A_2^{\mathsf{Dec}_{sk}(\cdot)}(M, C, R, state)$
```

```
Experiment $\mathbf{Exp}_{\Pi,A}^{\text{NC-CCA-Sim}}(k)$:
    $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
    $(M, state) \leftarrow A_1^{\mathsf{Dec}_{sk}(\cdot)}(pk)$
    $C \leftarrow S_1(pk, 1^{|M|})$
    $R \leftarrow S_2(M)$
    return $A_2^{\mathsf{Dec}_{sk}(\cdot)}(M, C, R, state)$.
```

In both experiments, $A = (A_1, A_2)$ is allowed to access a decryption oracle $\mathsf{Dec}_{sk}(\cdot)$ with the constraint that A_2 is not allowed to query C.

The advantage of adversary A is defined as follows:

$$\mathbf{Adv}_{\Pi,A,S}^{\text{NC-CCA}}(k) := \Big|\Pr\Big[\mathsf{Exp}_{\Pi,A}^{\text{NC-CCA-Real}}(k) = 1\Big] - \Pr\Big[\mathsf{Exp}_{\Pi,A}^{\text{NC-CCA-Sim}}(k) = 1\Big]\Big|.$$

Definition 1. A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *sender-equivocable* under adaptive chosen-ciphertext attacks (*NC-CCA secure*) if there is a stateful PPT algorithm S (the simulator), such that for any PPT algorithm A (the adversary) the advantage $\mathbf{Adv}_{\Pi,A,S}^{\text{NC-CCA}}(k)$ is negligible.

2.3. Building blocks of the FHKW scheme. Fehr *et al.* (2010) presented a construction of PKE with NC-CCA security. We will call their scheme FHKW. It was built using the following cryptographic primitives: a collision-resistant hash function, a subset membership problem, an extended version of the hash proof system (Cramer and Shoup, 2002), and a cross-authentication code (Fehr *et al.*, 2010).

Definition 2. A family of *collision-resistant hash functions* $\mathcal{H} : \mathcal{D} \to \mathcal{R}$ consists of two PPT algorithms $(\mathsf{HGen}, \mathsf{HEval})$. Algorithm $\mathsf{HGen}(1^k)$ randomly chooses a hash function from the family and outputs the description of the hash function H. Algorithm $\mathsf{HEval}(\mathsf{H}, x)$ produces the hash value $\mathsf{H}(x)$ for all $x \in \mathcal{D}$. Furthermore, for any PPT algorithm A, the following function is negligible in k:

$$\mathbf{Adv}_{\mathcal{H},A}^{cr}(k) := \Pr\left[\begin{array}{l}\mathsf{H} \leftarrow \mathsf{HGen}(1^k)\\ (x, x') \leftarrow A(\mathsf{H})\end{array} : \begin{array}{c}x \neq x' \wedge\\ \mathsf{H}(x) = \mathsf{H}(x')\end{array}\right].$$

Here we do not distinguish a function H from its description output by HGen.

Definition 3. A *subset membership problem* consists of the following PPT algorithms:

- $\mathsf{SmpGen}(1^k)$: On input 1^k, Algorithm SmpGen outputs a parameter Λ, which specifies a set $\mathcal{X}_\Lambda$ and its subset $\mathcal{L}_\Lambda \subseteq \mathcal{X}_\Lambda$. Set $\mathcal{X}_\Lambda$ is required to be easily recognizable with Λ.
- $\mathsf{SampleL}(\mathcal{L}_\Lambda; W)$: Algorithm $\mathsf{SampleL}$ samples $X \in \mathcal{L}_\Lambda$ using randomness $W \in \mathcal{R}_{\mathsf{SampleL}}$.

A subset membership problem SMP is *hard* if, for any PPT distinguisher D, D's advantage

$$\mathbf{Adv}_{\mathsf{SMP},D}(k) := \left|\Pr\left[\begin{array}{l}\Lambda \leftarrow \mathsf{SmpGen}(1^k)\\ X \leftarrow \mathcal{L}_\Lambda\end{array} : D(X) = 1\right] - \Pr\left[\begin{array}{l}\Lambda \leftarrow \mathsf{SmpGen}(1^k)\\ X \leftarrow \mathcal{X}_\Lambda\end{array} : D(X) = 1\right]\right|$$

is negligible.

Definition 4. A subset membership problem SMP has the property of *subset sparseness* if the probability $\Pr[\Lambda \leftarrow \mathsf{SmpGen}(1^k), X \leftarrow \mathcal{X}_\Lambda : X \in \mathcal{L}_\Lambda]$ is negligible.

Definition 5. A *hash proof system* HPS for a subset membership problem SMP associates each $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$ with an efficiently recognizable key space $\mathcal{K}_\Lambda$ and the following PPT algorithms:

- $\mathsf{HashGen}(\Lambda)$: On input Λ, $\mathsf{HashGen}$ outputs a public key hpk and a secret key hsk, both containing the parameter Λ.
- $\mathsf{SecEvl}(hsk, X)$: It is a deterministic algorithm. On input a secret key hsk and an element $X \in \mathcal{X}_\Lambda$, SecEvl outputs a key $K \in \mathcal{K}_\Lambda$.
- $\mathsf{PubEvl}(hpk, X, W)$: It is a deterministic algorithm. On input a public key hpk, an element $X \in \mathcal{X}_\Lambda$ and a witness W for $X \in \mathcal{L}_\Lambda$, PubEvl outputs a key $K \in \mathcal{K}_\Lambda$. The correctness requires that $\mathsf{PubEvl}(hpk, X, W) = \mathsf{SecEvl}(hsk, X)$ for all $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, $(hpk, hsk) \leftarrow \mathsf{HashGen}(\Lambda)$ and $X \leftarrow \mathsf{SampleL}(\mathcal{L}_\Lambda; W)$.

An *extended hash proof system* EHPS is a variation of a hash proof system HPS, extending the sets $\mathcal{X}_\Lambda$ and $\mathcal{L}_\Lambda$ by taking the Cartesian product of these sets with an efficiently recognizable tag space $\mathcal{T}_\Lambda$. Hence, the tuple of the three algorithms ($\mathsf{HashGen}$, SecEvl, PubEvl) of EHPS is changed to $(hpk, hsk) \leftarrow \mathsf{HashGen}(\Lambda)$, $K \leftarrow \mathsf{SecEvl}(hsk, X, t)$ and $K \leftarrow \mathsf{PubEvl}(hpk, X, W, t)$, with $t \in \mathcal{T}_\Lambda$.

The public key hpk in a hash proof system HPS uniquely determines the action of algorithm SecEvl for all $X \in \mathcal{L}_\Lambda$. However, the action of SecEvl for $X \in \mathcal{X}_\Lambda \setminus \mathcal{L}_\Lambda$ is still undetermined by hpk. This is defined by a *perfectly* 2*-universal* property.

Definition 6. A hash proof system HPS for SMP is *perfectly 2-universal* if, for any $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, any hpk from $\mathsf{HashGen}(\Lambda)$, any distinct $X_1, X_2 \in \mathcal{X}_\Lambda \setminus \mathcal{L}_\Lambda$, and any $K_1, K_2 \in \mathcal{K}_\Lambda$,

$$\begin{aligned}&\Pr[\mathsf{SecEvl}(hsk, X_2)\\ &\quad = K_2 \mid \mathsf{SecEvl}(hsk, X_1) = K_1]\\ &\quad = \frac{1}{|\mathcal{K}_\Lambda|},\end{aligned}$$

where the probability is taken over all possible hsk with $(hpk, hsk) \leftarrow \mathsf{HashGen}(\Lambda)$.

Definition 7. A domain $\mathcal{D}$ is *efficiently samplable and explainable* if there exists two PPT algorithms:

- $\mathsf{Sample}(\mathcal{D}; R)$: On input a random coin $R \leftarrow \mathcal{R}_{\mathsf{Sample}}$ and a domain $\mathcal{D}$, it outputs an element uniformly distributed over $\mathcal{D}$.
- $\mathsf{Explain}(\mathcal{D}, x)$: On input $\mathcal{D}$ and $x \in \mathcal{D}$, this algorithm outputs R that is uniformly distributed over the set $\{R \in \mathcal{R}_{\mathsf{Sample}} \mid \mathsf{Sample}(\mathcal{D}; R) = x\}$.

Definition 8. (*Fehr et al., 2010*) For any $L \in \mathbb{N}$, an *L-cross-authentication code* XAC, associated with a key space $\mathcal{XK}$ and a tag space $\mathcal{XT}$, consists of three PPT algorithms (XGen, XAuth, XVer). Algorithm $\mathsf{XGen}(1^k)$ generates a uniformly random key $K \in \mathcal{XK}$, $\mathsf{XAuth}(K_1, \ldots, K_L)$ produces a tag $T \in \mathcal{XT}$, and $\mathsf{XVer}(K, i, T)$ outputs $b \in \{0, 1\}$. The following properties are required.

Correctness. The function

$$\begin{aligned}&\mathsf{fail}^{\mathsf{correct}}_{\mathsf{XAC}}(k)\\ &\quad := \max_{i\in[L]} \Pr[\mathsf{XVer}(K_i, i, \mathsf{XAuth}((K_j)_{j\in[L]})) \neq 1]\end{aligned}$$

is negligible in k, where the maximum is over all $i \in [L]$ and the probability is taken over all possible $K_1, \cdots, K_L \leftarrow \mathsf{XGen}(1^k)$.

Security against impersonation and substitution attacks. The advantages $\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)$ and $\mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$, defined as follows, are both negligible:

$$\begin{aligned}&\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)\\ &\quad := \max_{i,T'} \Pr[K \leftarrow \mathsf{XGen}(1^k) : \mathsf{XVer}(K, i, T') = 1]\ ,\end{aligned}$$

where the maximum is over all $i \in [L]$ and $T' \in \mathcal{XT}$.

$$\begin{aligned}&\mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)\\ &:= \max_{i,K_{\neq i},\mathsf{Func}}\\ &\Pr\left[\begin{array}{l} K_i \leftarrow \mathsf{XGen}(1^k)\\ T = \mathsf{XAuth}((K_j)_{j\in[L]})\\ T' \leftarrow \mathsf{Func}(T)\end{array} : \begin{array}{c} T' \neq T \wedge \\ \mathsf{XVer}(K_i, i, T') = 1\end{array}\right]\end{aligned}$$

where the maximum is over all $i \in [L]$, all $K_{\neq i} := (K_j)_{j\neq i} \in \mathcal{XK}^{L-1}$ and all possibly randomized functions $\mathsf{Func} : \mathcal{XT} \to \mathcal{XT}$.

3. Review of the FHKW scheme

With the above cryptographic primitives, we now present the FHKW scheme (Fehr *et al.*, 2010).

Let SMP be a hard subset membership problem that has the property of subset sparseness. Let $\mathcal{X}_\Lambda$, with $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, be efficiently samplable and explainable. Let EHPS be a perfectly 2-universal extended hash proof system for SMP with tag space $\mathcal{T}_\Lambda$ and key space (range) $\mathcal{K}_\Lambda$, which is efficiently samplable and explainable as well. Let $\mathcal{H} : (\mathcal{X}_\Lambda)^L \to \mathcal{T}_\Lambda$ be a family of collision-resistant hash functions, and XAC be an L-cross-authentication code with key space $\mathcal{XK} = \mathcal{K}_\Lambda$ and tag space $\mathcal{XT}$.

FHKW scheme:

$\mathsf{Gen}(1^k)$: On input 1^k, algorithm Gen runs $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, $(hpk, hsk) \leftarrow \mathsf{HashGen}(\Lambda)$, $\mathsf{H} \leftarrow \mathcal{H}$, and outputs (pk, sk), where $pk = (hpk, \mathsf{H})$ and $sk = (hsk, \mathsf{H})$.

$\mathsf{Enc}(pk, M; R)$: To encrypt a plaintext

$$M = (M_1, \dots, M_L) \in \{0,1\}^L$$

under a public key $pk = (hpk, \mathsf{H})$ with randomness

$$R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}$$

$\in (\mathcal{R}_{\mathsf{SampleL}} \times \mathcal{R}_{\mathsf{Sample}} \times \mathcal{R}_{\mathsf{Sample}})^L$.

Algorithm Enc runs as follows:
For $i \in [L]$, set

$$X_i := \begin{cases} \mathsf{Sample}(\mathcal{X}_\Lambda; R_i^{\mathcal{X}_\Lambda}) & \text{if } M_i = 0, \\ \mathsf{SampleL}(\mathcal{L}_\Lambda; W_i) & \text{if } M_i = 1, \end{cases}$$

and $t := \mathsf{H}(X_1, \dots, X_L)$. Then for $i \in [L]$, set the keys

$$K_i := \begin{cases} \mathsf{Sample}(\mathcal{K}_\Lambda; R_i^{\mathcal{K}_\Lambda}) & \text{if } M_i = 0, \\ \mathsf{PubEvl}(hpk, X_i, W_i, t) & \text{if } M_i = 1, \end{cases}$$

and the tag $T := \mathsf{XAuth}(K_1, \dots, K_L)$. Finally, return $C = (X_1, \cdots, X_L, T)$ as the ciphertext.

$\mathsf{Dec}(sk, C)$: To decrypt a ciphertext

$$C = (X_1, \dots, X_L, T) \in \mathcal{X}_\Lambda^L \times \mathcal{XT}$$

under a secret key $sk = (hsk, \mathsf{H})$, Algorithm Dec computes $t = \mathsf{H}(X_1, \cdots, X_L)$, for $i \in [L]$ sets $\overline{K_i} := \mathsf{SecEvl}(hsk, X_i, t)$ and $M_i = \mathsf{XVer}(\overline{K_i}, i, T)$, and returns $M = (M_1, \dots, M_L)$ as the plaintext.

The correctness of the FHKW scheme is proved by Fehr *et al.* (2010), and omitted here.

4. Security analysis of the FHKW scheme

According to the definition of NC-CCA security, the FHKW scheme is NC-CCA secure, if and only if there exists a simulator S such that for any PPT algorithm A, the two experiments $\mathsf{Exp}_{\text{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\text{FHKW},A}^{\text{NC-CCA-Sim}}(k)$, defined in Section 2, are indistinguishable.

In order to prove NC-CCA security of the FHKW scheme, Fehr *et al.* (2010) constructed the following simulator $S = (S_1, S_2)$.

Simulator S:

- $S_1(pk, 1^{|M|})$: Parse $pk = (hpk, \mathsf{H})$. For $i \in [L]$, choose $\widetilde{W_i} \leftarrow \mathcal{R}_{\mathsf{SampleL}}$ and set $X_i := \mathsf{SampleL}(\mathcal{L}_\Lambda; \widetilde{W_i})$. Compute $t := \mathsf{H}(X_1, \dots, X_L)$. For $i \in [L]$, set $K_i := \mathsf{PubEvl}(hpk, X_i, \widetilde{W_i}, t)$. Set $T \leftarrow \mathsf{XAuth}(K_1, \dots, K_L)$. Return the ciphertext $C = (X_1, \dots, X_L, T)$.
- $S_2(M)$: Parse $M = (M_1, \dots, M_L)$. For $i \in [L]$, if $M_i = 1$, set $W_i := \widetilde{W_i}$, and choose $R_i^{\mathcal{X}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$, $R_i^{\mathcal{K}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$; otherwise, choose $W_i \leftarrow \mathcal{R}_{\mathsf{SampleL}}$, and set $R_i^{\mathcal{X}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{X}_\Lambda, X_i)$, $R_i^{\mathcal{K}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{K}_\Lambda, K_i)$. Return the randomness

$$R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}.$$

With the simulator S, Fehr *et al.* (2010) proved that the FHKW scheme is NC-CCA secure. However, we will show that this specific simulator S does not guarantee NC-CCA security of the FHKW scheme for any positive integer L.

4.1. Security proof problem. To prove NC-CCA security, it is essential to show that the decryption oracle will not leak any useful information to any PPT adversary. As to the FHKW scheme, given a challenge ciphertext $C = (X_1, \dots, X_L, T)$, an adversary A comes up with a decryption query $C' = (X_1, \dots, X_L, T')$, where $T' \neq T$. NC-CCA security expects the decryption of C' by the oracle will not help the adversary to distinguish the two experiments $\mathsf{Exp}_{\text{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\text{FHKW},A}^{\text{NC-CCA-Sim}}(k)$ (see the proof of Lemma 5 in the work of Fehr *et al.* (2010)). This strongly relies on the security against substitution attacks of the cross-authentication code, which requires that "given T and $K_{\neq i}$, it is difficult to output a $T' \neq T$ such that $\mathsf{XVer}(K_i, i, T') = 1$, where K_i is uniformly distributed".

However, in the NC-CCA game, adversary A KNOWs K_i for any $i \in [L]$! The reason is as follows. Upon returning a plaintext M, adversary A receives not only a challenge ciphertext C, but also some related random coins R which are supposed to have been consumed in the challenge ciphertext generation. With R and M, adversary A can recover K_i for any $i \in [L]$. Then, it is possible for A to output a $T' \neq T$ such that $\mathsf{XVer}(K_i, i, T') = 1$. Hence, XAC's security against substitution attacks is not sufficient to guarantee the aforementioned property. That is why the security proof proposed by Fehr *et al.* (2010) fails (more precisely, the proof of Lemma 5 in the work of Fehr *et al.* (2010) does).

In fact, this kind of adversary, which can output a $T' \neq T$ such that $\mathsf{XVer}(K_i, i, T') = 1$ given T and K_i for any $i \in [L]$, does exist. In Section 4.2, we will present such an adversary A to destroy the security proof of the FHKW scheme for $L > 1$.

Deniable scheme. Gao *et al.* (2012) utilized exactly the same technique as that in the FHKW scheme to construct a deniable encryption scheme and "proved" the CCA security. A similar problem we pointed out above also exists in their security proof (more specifically, the proof of Claim 1 in the work of Gao *et al.* (2012)). As a result, our attack in Section 4.2 applies to their scheme and ruins their proof, too.

4.2. Security analysis of the FHKW scheme: $L > 1$.

Before going into a formal statement and its proof, we briefly give a high-level description of our security analysis for $L > 1$.

With the aforementioned simulator S, for any $L > 1$, our aim is to construct an adversary $A = (A_1, A_2)$ to distinguish the two experiments $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$. The construction of adversary A is as follows.

In an experiment environment (either $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ or $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$), upon receiving pk, A_1 returns $M = (0, \dots, 0)$. Then, upon receiving a ciphertext $C = (X_1, \dots, X_L, T)$ and randomness R, A_2 returns $C' = (X_1, \dots, X_L, T')$ as his decryption query, where $T' \leftarrow \mathsf{XAuth}(K_1', K_2, \dots, K_L)$, K_1' is uniformly random chosen from $\mathcal{K}_\Lambda$ and $K_2, \dots, K_L$ are all recovered from R. Finally, if the decryption oracle returns $M' = (0, \dots, 0)$, A_2 will output $b = 1$, and otherwise, A_2 will output $b = 0$.

Now, we consider the probabilities that A outputs 1 in the two experiments. In $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$, for $i \in [L]$, X_i (resp. K_i) is chosen uniformly random from $\mathcal{X}_\Lambda$ (resp. $\mathcal{K}_\Lambda$), so the subset sparseness of SMP and the perfect 2-universality of HPS guarantee that for $i \in [L]$, $\overline{K_i'} = \mathsf{SecEvl}(hsk, X_i, t)$ is uniformly random in $\mathcal{K}_\Lambda$ from A's point of view. Due to the security of XAC, the decryption oracle returns $M' = (0, 0, \dots, 0)$ for the queried ciphertext C'. Consequently, A outputs $b = 1$ with an overwhelming probability in $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$.

On the other hand, in $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$, for $i \in [L]$, X_i is chosen uniformly random from $\mathcal{L}_\Lambda$ and $K_i = \mathsf{PubEvl}(hpk, X_i, W_i, t)$, so the property of HPS guarantees that, for $i \in [L]$, $\overline{K_i'} = \mathsf{SecEvl}(hsk, X_i, t) = K_i$. Due to the correctness of XAC and the facts that $T' \leftarrow \mathsf{XAuth}(K_1', K_2, \dots, K_L)$ and $M_i' = \mathsf{XVer}(\overline{K_i'}, i, T') = 1$ for $i \in \{2, 3, \dots, L\}$, the decryption oracle returns $M' = (0, 1, \dots, 1)$ with an overwhelming probability. As a result, A outputs $b = 1$ with negligible probability in $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$. The two experiments $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$ have been distinguished by A with an overwhelming advantage.

A formal statement of the result and its corresponding proof are as follows.

Theorem 1. *With the aforementioned simulator S, the* FHKW *scheme cannot be proved to be NC-CCA secure for any $L > 1$. More specifically, there exists an adversary A distinguishing the real and the simulated NC-CCA experiments, with the advantage*

$$\begin{aligned}&\mathbf{Adv}^{\text{NC-CCA}}_{\text{FHKW},A,S}(k)\\&\qquad \geq 1 - 2\mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k) - \mathsf{fail}^{\text{correct}}_{\mathsf{XAC}}(k).\end{aligned}$$

Proof. For simplicity, we consider the case of $L = 2$. We note that this attack is applicable to any $L > 1$.

Our aim is to construct a specific adversary $A = (A_1, A_2)$ to distinguish the two experiments $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$ with a non-negligible advantage.

Specifically, given an experiment environment (either $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ or $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$), the adversary $A = (A_1, A_2)$ behaves as follows.

- Upon receiving $pk = (hpk, \mathsf{H})$, A_1 returns $M = (0, 0)$, i.e., $M_1 = M_2 = 0$.

- Upon receiving a ciphertext

  $$C = (X_1, X_2, T)$$

  and randomness

  $$R = ((W_1, R_1^{\mathcal{X}_\Lambda}, R_1^{\mathcal{K}_\Lambda}),$$

  $(W_2, R_2^{\mathcal{X}_\Lambda}, R_2^{\mathcal{K}_\Lambda}))$, A_2 creates a new ciphertext C' according to C:

  - Set $X_1' := X_1$, $X_2' := X_2$.
  - Set $K_1' \leftarrow \mathcal{K}_\Lambda$, $K_2' \leftarrow \mathsf{Sample}(\mathcal{K}_\Lambda; R_2^{\mathcal{K}_\Lambda})$.
  - Compute $T' \leftarrow \mathsf{XAuth}(K_1', K_2')$.
  - Check that $T' \neq T$. If $T' = T$, choose another random value for K_1' and repeat the above steps, until $T' \neq T$.
  - Set $C' := (X_1', X_2', T')$.

  Then A_2 submits C' to the decryption oracle.

- Let $M' \leftarrow \mathsf{Dec}(sk, C')$. A_2 outputs b, where

  $$b = \begin{cases} 1 & \text{if} \quad M' = (0,0),\\ 0 & \text{if} \quad M' \neq (0,0).\end{cases}$$

Now we analyze the probabilities that A_2 outputs $b = 1$ in the real and the simulated experiment,

In both experiments, A_2 receives a ciphertext $C = (X_1, X_2, T)$ and randomness $R = ((W_1, R_1^{\mathcal{X}_\Lambda}, R_1^{\mathcal{K}_\Lambda}), (W_2, R_2^{\mathcal{X}_\Lambda}, R_2^{\mathcal{K}_\Lambda}))$. The ciphertext created and submitted to the decryption oracle by A_2 is $C' = (X_1', X_2', T') = (X_1, X_2, T')$, where $T' = \mathsf{XAuth}(K_1', K_2') = \mathsf{XAuth}(K_1', K_2)$ (due to $K_2' = K_2$) and $T' \neq T$.

Real experiment. The challenge ciphertext $C = (X_1, X_2, T)$ satisfies $X_1 \leftarrow \mathsf{Sample}(\mathcal{X}_\Lambda; R_1^{\mathcal{X}_\Lambda})$, $X_2 \leftarrow \mathsf{Sample}(\mathcal{X}_\Lambda; R_2^{\mathcal{X}_\Lambda})$, and $T = \mathsf{XAuth}(K_1, K_2)$, where $K_1 \leftarrow \mathsf{Sample}(\mathcal{K}_\Lambda; R_1^{\mathcal{K}_\Lambda})$ and $K_2 \leftarrow \mathsf{Sample}(\mathcal{K}_\Lambda; R_2^{\mathcal{K}_\Lambda})$.

The decryption of C' by the decryption oracle $\mathsf{Dec}(sk,\cdot)$ involves the computation of $t' := \mathsf{H}(X_1', X_2') = \mathsf{H}(X_1, X_2) = t$ and $\overline{K_i'} := \mathsf{SecEvl}(hsk, X_i', t') = \mathsf{SecEvl}(hsk, X_i, t)$, for $i \in \{1,2\}$.

Due to the perfect 2-universality of EHPS, $\overline{K_i'}$ is uniformly random distributed in $\mathcal{K}_\Lambda$. Hence, for $i \in \{1,2\}$,

$$\Pr\left[\mathsf{XVer}(\overline{K_i'}, i, T') = 1 | \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right] \leq \mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k). \quad (1)$$

Let $M' = (M_1', M_2')$ denote the decryption result of C' by the decryption oracle $\mathsf{Dec}(sk,\cdot)$. Then for $i \in \{1,2\}$,

$$\begin{aligned}\Pr&\left[M_i' = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right]\\ &= \Pr\left[\mathsf{XVer}(\overline{K_i'}, i, T') = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right]\\ &\leq \mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k).\end{aligned} \quad (2)$$

The probability that A_2 outputs $b = 1$ in the real experiment is given by

$$\begin{aligned}\Pr&\left[\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k) = 1\right]\\ &= \Pr\left[M' = (0,0) \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right]\\ &= 1 - \Pr\left[M' \neq (0,0) \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right]\\ &= 1 - \Pr\left[M_1' = 1 \vee M_2' = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)\right]\\ &\geq 1 - 2\mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k).\end{aligned} \quad (3)$$

Simulated experiment. The ciphertext $C = (X_1, X_2, T)$ satisfies $X_1 \leftarrow \mathsf{SampleL}(\mathcal{L}_\Lambda; \widetilde{W_1})$, $X_2 \leftarrow \mathsf{SampleL}(\mathcal{L}_\Lambda; \widetilde{W_2})$, and $T = \mathsf{XAuth}(K_1, K_2)$, where, for $i \in \{1,2\}$, $\widetilde{W_i} \leftarrow \mathcal{R}_{\mathsf{SampleL}}$ and $K_i = \mathsf{PubEvl}(hpk, X_i, \widetilde{W_i}, t)$ with $t = \mathsf{H}(X_1, X_2)$.

The decryption of C' by the decryption oracle $\mathsf{Dec}(sk,\cdot)$ involves the computation of $t' = \mathsf{H}(X_1', X_2') = \mathsf{H}(X_1, X_2) = t$ and $\overline{K_i'} = \mathsf{SecEvl}(hsk, X_i', t') = \mathsf{SecEvl}(hsk, X_i, t)$, for $i \in \{1,2\}$. On the other hand, we know that $K_2' = K_2$ and $K_2 = \mathsf{PubEvl}(hpk, X_2, W_2, t)$. Since $X_2 \in \mathcal{L}_\Lambda$, the property of EHPS guarantees that $\mathsf{SecEvl}(hsk, X_2, t) = \mathsf{PubEvl}(hpk, X_2, W_2, t)$, which means that $\overline{K_2'} = K_2 = K_2'$. Note that $M_2' = \mathsf{XVer}(\overline{K_2'}, 2, T')$. Hence, we have

$$\begin{aligned}\Pr&\left[M_2' = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &= \Pr\left[\mathsf{XVer}(\overline{K_2'}, 2, T') = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &= \Pr\left[\mathsf{XVer}(K_2', 2, T') = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &\geq 1 - \mathsf{fail}^{\text{correct}}_{\mathsf{XAC}}(k).\end{aligned} \quad (4)$$

The probability that A_2 outputs $b = 1$ in the simulated experiment is given by

$$\begin{aligned}\Pr&\left[\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k) = 1\right]\\ &= \Pr\left[M' = (0,0) \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &= 1 - \Pr\left[M' \neq (0,0) \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &\leq 1 - \Pr\left[M_2' = 1 \mid \text{in } \mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)\right]\\ &\leq \mathsf{fail}^{\text{correct}}_{\mathsf{XAC}}(k).\end{aligned} \quad (5)$$

The advantage of adversary A is given by

$$\begin{aligned}&\mathbf{Adv}^{\text{NC-CCA}}_{\text{FHKW},A,S}(k)\\ &= \left|\Pr\left[\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k) = 1\right]\right.\\ &\qquad \left. - \Pr\left[\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k) = 1\right]\right|\\ &\geq 1 - 2\mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k) - \mathsf{fail}^{\text{correct}}_{\mathsf{XAC}}(k).\end{aligned} \quad (6)$$

Note that both $\mathbf{Adv}^{\text{imp}}_{\mathsf{XAC}}(k)$ and $\mathsf{fail}^{\text{correct}}_{\mathsf{XAC}}(k)$ are negligible. So A's advantage $\mathbf{Adv}^{\text{NC-CCA}}_{\text{FHKW},A,S}(k)$ is non-negligible (in fact, it is overwhelming), i.e., the security proof of the FHKW scheme (Fehr *et al.*, 2010) is incorrect. ■

4.3. Security analysis of the FHKW scheme: $L = 1$.

Note that our attack in the previous section does not apply to the case $L = 1$. There upon receiving the ciphertext C and randomness R, the adversary A recovers K and switches the first element of K with a random one. If $L = 1$, A will get a new $K' = K_1'$ and then $T' = \mathsf{XAuth}(K_1')$. Afterwards, A will return $C' = (X_1, T')$ as his decryption query. Then, A will receive $M' = 0$ with overwhelming probability in both $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$. Hence, the two experiments are still indistinguishable for A.

As we have pointed out earlier, the security of the L-cross-authentication code against substitution attacks is not sufficient for the security proof of the FHKW scheme for any value of L. But our attack above only works for $L > 1$. Therefore, the remaining problem is whether it is possible for the FHKW scheme to achieve

NC-CCA security for $L = 1$, still with the aforementioned simulator S.

Before solving the problem, we claim that algorithm XAuth of XAC in the FHKW scheme is deterministic (this is not explicitly expressed in the work of Fehr *et al.* (2010)). That is because $R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i \in [L]}$ is the only randomness used in the encryption process. In other words, if XAuth is probabilistic, the inner random number used by XAuth should be contained in the randomness R (and then passed to the adversary, according to the definition of NC-CCA security). On the other hand, if algorithm XAuth of XAC in the FHKW scheme is probabilistic, with the aforementioned simulator S, the FHKW scheme *cannot* be proved secure in the sense of NC-CCA for any positive integer L. (See Appendix A for the proof.)

In fact, the security proof of the FHKW scheme expected such a property from the L-cross-authentication code: "given $(K_1, K_2, \ldots, K_L)$ and $T = \mathsf{XAuth}(K_1, \ldots, K_L)$, it is difficult to output a $T' \neq T$ such that $\mathsf{XVer}(K_i, i, T') = 1$ for some $i \in [L]$". This property generally does not hold for the L-cross-authentication code. However, it is true for some special 1-cross-authentication code, for example, the instance of an L-cross-authentication code given by Fehr *et al.* (2010) when constricted to $L = 1$. For that special instance, when $L = 1$, given $K = K_1$ and $T = \mathsf{XAuth}(K_1)$ (note that XAuth is deterministic), it is *impossible* to find a $T' \neq T$ such that $\mathsf{XVer}(K_1, 1, T') = 1$, since only $T = \mathsf{XAuth}(K_1)$ itself could pass the verification. Therefore, with the special 1-cross-authentication code instance (or other instance with a similar property) as the ingredient, the FHKW scheme is NC-CCA secure for $L = 1$.

5. Sender-equivocable encryption scheme for a single bit

In this section, we will refine the FHKW scheme for $L = 1$. Specifically, we will present a PKE scheme with NC-CCA security for $L = 1$ without any L-cross-authentication code.

Our scheme can be seen as a simplified version of the FHKW scheme instantiated with a special 1-cross-authentication code. As we have pointed earlier, the special property of a 1-cross-authentication code requires that each K determine a unique tag T satisfying $\mathsf{XVer}(K, T) = 1$. In our scheme, the encryption algorithm replaces the tag T by the key K directly. In the decryption, whether the plaintext is 1 or 0 depends on the equality of K in the ciphertext and $\overline{K}$ computed by $\mathsf{SecEvl}(hsk, X)$, while in the FHKW scheme the plaintext bit is determined by whether $\mathsf{XVer}(K, T') = 1$ or not.

Below we describe our scheme $\mathcal{E} = (\mathsf{Gen}_{\mathcal{E}}, \mathsf{Enc}_{\mathcal{E}}, \mathsf{Dec}_{\mathcal{E}})$. It consists of a hard subset membership problem SMP, with subset sparseness, and its corresponding perfectly 2-universal hash proof system HPS. We require that for any $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, both $\mathcal{X}_\Lambda$ (with respect to SMP) and $\mathcal{K}_\Lambda$ (with respect to HPS) be efficiently explainable. As suggested by Fehr *et al.* (2010), the requirement of efficient samplability and explainability on $\mathcal{K}_\Lambda$ imposes no real restriction, and it was shown in the work of Cramer and Shoup (2002) that both of the above ingredients can be constructed based on some standard number-theoretic assumptions, such as the DDH, DCR and QR assumptions.

Scheme $\mathcal{E} = (\mathsf{Gen}_{\mathcal{E}}, \mathsf{Enc}_{\mathcal{E}}, \mathsf{Dec}_{\mathcal{E}})$:

$\mathsf{Gen}_{\mathcal{E}}(1^k)$: On input 1^k, algorithm $\mathsf{Gen}_{\mathcal{E}}$ runs $\Lambda \leftarrow \mathsf{SmpGen}(1^k)$, $(hpk, hsk) \leftarrow \mathsf{HashGen}(\Lambda)$, and outputs (pk, sk), where $pk = hpk$ and $sk = hsk$.

$\mathsf{Enc}_{\mathcal{E}}(pk, M; R)$: To encrypt a plaintext $M \in \{0, 1\}$ under a public key $pk = hpk$ with randomness $R = (W, R^{\mathcal{X}_\Lambda}, R^{\mathcal{K}_\Lambda}) \in \mathcal{R}_{\mathsf{SampleL}} \times \mathcal{R}_{\mathsf{Sample}} \times \mathcal{R}_{\mathsf{Sample}}$, algorithm $\mathsf{Enc}_{\mathcal{E}}$ sets

$$X := \begin{cases} \mathsf{Sample}(\mathcal{X}_\Lambda; R^{\mathcal{X}_\Lambda}) & \text{if } M = 0, \\ \mathsf{SampleL}(\mathcal{L}_\Lambda; W) & \text{if } M = 1, \end{cases}$$

and

$$K := \begin{cases} \mathsf{Sample}(\mathcal{K}_\Lambda; R^{\mathcal{K}_\Lambda}) & \text{if } M = 0, \\ \mathsf{PubEvl}(hpk, X, W) & \text{if } M = 1, \end{cases}$$

then returns ciphertext $C = (X, K)$.

$\mathsf{Dec}_{\mathcal{E}}(sk, C)$: To decrypt a ciphertext $C = (X, K) \in \mathcal{X}_\Lambda \times \mathcal{K}_\Lambda$ under a secret key $sk = hsk$, algorithm $\mathsf{Dec}_{\mathcal{E}}$ sets $\overline{K} := \mathsf{SecEvl}(hsk, X)$. If $\overline{K} = K$, return $M = 1$; otherwise, return $M = 0$.

Correctness. On the one hand, if $C = (X, K)$ is a ciphertext of $M = 1$, then $\overline{K} = \mathsf{SecEvl}(hsk, X) = \mathsf{PubEvl}(hpk, X, W) = K$ due to the property of HPS. So $\mathsf{Dec}_{\mathcal{E}}(sk, C)$ returns $M = 1$. On the other hand, if $C = (X, K)$ is a ciphertext of $M = 0$, then $X \leftarrow \mathcal{X}_\Lambda$, $K \leftarrow \mathcal{K}_\Lambda$ and $\overline{K} = \mathsf{SecEvl}(hsk, X)$. So $\Pr[\overline{K} = K] = 1/|\mathcal{K}_\Lambda|$. Hence, with probability $1 - 1/|\mathcal{K}_\Lambda|$, $\mathsf{Dec}_{\mathcal{E}}(sk, C)$ returns $M = 0$.

Security. As for the security of scheme $\mathcal{E}$, we have the following theorem. The proof is similar to that of the FHKW scheme (Fehr *et al.*, 2010). But the key observation is: Given $C = (X, K)$, it is impossible to create $C' = (X, K')$, $K \neq K'$, such that $K' = \overline{K'}$. Note that the security proof of our scheme does not involve any cross-authentication code. Details of the proof are in Appendix B.

Theorem 2. *Assuming that SMP is a hard subset membership problem with subset sparseness, and HPS is its corresponding perfectly 2-universal hash proof system, scheme $\mathcal{E} = (\mathsf{Gen}_{\mathcal{E}}, \mathsf{Enc}_{\mathcal{E}}, \mathsf{Dec}_{\mathcal{E}})$ is NC-CCA secure.*

6. Strong L-cross-authentication codes

In this section, we will introduce a strong version of L-cross-authentication codes, which will be used to construct a new version of the FHKW scheme achieving NC-CCA security. This primitive may find other cryptographic applications.

The formal definition of a strong L-cross-authentication code is as follows.

Definition 9. For $L \in \mathbb{N}$, an *L-cross-authentication code* XAC is strong if there exists a PPT algorithm ReSamp satisfying the following property: Given $K_1, \ldots, K_L \leftarrow \mathsf{XGen}(1^k)$ and $T = \mathsf{XAuth}((K_j)_{j\in[L]})$ such that $\mathsf{XVer}(K_j, j, T) = 1$, $j \in [L]$, algorithm ReSamp takes as input $i \in [L]$, $K_{\neq i} := (K_j)_{j\neq i}$ and T, and outputs K'_i, which is statistically indistinguishable from K_i, i.e.,

$$\begin{aligned}&\mathrm{Dist}(k)\\ &\quad := \frac{1}{2}\sum_{K\in\mathcal{XK}} |\Pr[K'_i = K|(K_{\neq i}, T)]\\ &\qquad - \Pr[K_i = K|(K_{\neq i}, T)]|\end{aligned}$$

is negligible, where $K'_i \leftarrow \mathsf{ReSamp}(i, K_{\neq i}, T)$ and the probabilities are taken over all possible $K_i \leftarrow \mathsf{XGen}(1^k)$ such that $T = \mathsf{XAuth}((K_j)_{j\in[L]})$, and the randomness of ReSamp.

Remark 1. Recalling the discussion in Section 4.3, algorithm XAuth is deterministic. The indistinguishability of ReSamp implies that

$$\begin{aligned}&\mathsf{XAuth}(K_1, \ldots, K_i, \ldots, K_L)\\ &\quad = \mathsf{XAuth}(K_1, \ldots, K'_i, \ldots, K_L)\\ &\quad = T,\end{aligned} \tag{7}$$

with overwhelming probability, where $K'_i \leftarrow \mathsf{ReSamp}(i, K_{\neq i}, T)$.

Remark 2. The requirement that ReSamp is efficient is very important. Because this algorithm will be used to construct a simulator S in the next section, and NC-CCA security requires that the simulator should be a PPT algorithm.

Remark 3. This "efficient resampling" property is just a missing element in the security proof of the FHKW scheme. With this particular property, the strong cross-authentication code is able to resist the attack proposed in Section 4, and fill the gap in the security proof of the FHKW scheme.

Example of a strong L-cross-authentication code. Quite interestingly, the instance of an L-cross-authentication code XAC (Fehr *et al.*, 2010) is also strong. Now we recall the instance XAC=(XGen,XAuth,XVer) proposed by Fehr *et al.* (2010).

Let $\mathbb{F}_q$ be a finite field, where q is determined by the security parameter k. Define $\mathcal{XK} = \mathbb{F}_q^2$ and $\mathcal{XT} = \mathbb{F}_q^L \cup \{\perp\}$.

- $\mathsf{XGen}(1^k)$: Generate a random key $(a, b) \leftarrow \mathbb{F}_q^2$.
- $\mathsf{XAuth}(K_1, \ldots, K_L)$: For
$$K_1 = (a_1, b_1), \ldots, K_L = (a_L, b_L) \in \mathbb{F}_q^2,$$
XAuth computes a tag $T = (T_0, \ldots, T_{L-1})$ satisfying that for $i \in [L]$, $\mathrm{poly}_T(a_i) = b_i$, where $\mathrm{poly}_T(x) = T_0 + T_1x + \cdots + T_{L-1}x^{L-1} \in \mathbb{F}_q[x]$. Note that T can be computed efficiently by solving a linear equation system $\mathbf{A}T = \mathbf{B}$, where $\mathbf{A} \in \mathbb{F}_q^{L\times L}$ is a Vandermonde matrix and its i-th row is $(1, a_i, a_i^2, \cdots, a_i^{L-1})$ for $i \in [L]$, and $\mathbf{B} \in \mathbb{F}_q^L$ is a column vector with elements $b_1, \cdots, b_L$. If there are more than one or no solution for $\mathbf{A}T = \mathbf{B}$, XAuth will output $T = \perp$.
- $\mathsf{XVer}(K, i, T)$: For any $K = (a, b) \in \mathcal{XK}$, $i \in [L]$ and $T \in \mathcal{XT}$, XVer outputs 1 if and only if $T \neq\perp$ and $\mathrm{poly}_T(a) = b$.

The code XAC has been proved to be correct and secure against impersonation and substitution attacks (Fehr *et al.*, 2010). Here we only show that XAC is strong as well.

Lemma 1. *For any $L \in \mathbb{N}$, the L-cross-authentication code* XAC *is strong.*

Proof. A PPT algorithm ReSamp is constructed as follows. The input of ReSamp is $(i, K_{\neq i}, T)$, where $K_j = (a_j, b_j)$ for $j \in [L]\backslash\{i\}$, and T satisfies $\mathsf{XVer}(K_l, l, T) = 1$ for $l \in [L]$. This implies that $\mathbf{A}$ is non-singular. On input $(i, K_{\neq i}, T)$, ReSamp chooses $a'_i \leftarrow \mathbb{F}\backslash\{a_{\neq i}\}$, computes $b'_i = \mathrm{poly}_T(a'_i)$ and returns $K'_i = (a'_i, b'_i)$ as its output. As a result, $\Pr[K'_i = (a'_i, b'_i)] = 1/(q - L + 1)$.

On the other hand, conditioned on $K_{\neq i}$ and $T \neq \perp$, the solution space of $K_i = (a_i, b_i)$ is given by the set $\{(a, b) \in \mathbb{F}_q^2 \mid \mathrm{poly}_T(a) = b, a \neq a_j, j \in [L]\backslash\{i\}\}$. Hence

$$\Pr[K_i = (a, b) \mid (K_{\neq i}, T)] = \frac{1}{q - L + 1},$$

which has identical probability distribution with K'_i. ■

Relations between the strong and the normal version of cross-authentication codes. Although the instance XAC proposed by Fehr *et al.* (2010) is strong, we cannot conclude that every cross-authentication code is such. On the other hand, unfortunately, we cannot provide a counterexample either, i.e., a cross-authentication code example that is not strong. Whether the strong and the normal version are equivalent is still an open question.

7. Fixing the security proof of the FHKW scheme with strong L-cross-authentication codes

Replacing XAC with a strong one, we get a new version of the FHKW scheme, called the new FHKW scheme. In other words, the new FHKW scheme is identical with the original one, except that its building block XAC has one more algorithm ReSamp which does not appear in neither of the two versions of the FHKW scheme. The description of the new FHKW scheme is the same as that in Section 3, so we will not repeat it again.

Although algorithm ReSamp does not appear in the new FHKW scheme, it is essential for the strongness of XAC (and will be needed in the security proof). The strongness of the cross-authentication code helps its security against substitution attacks work in the security proof of the FHKW scheme (see the proof of Lemma 3). Roughly speaking, when the randomness of a ciphertext is disclosed to an adversary, all $K_1, K_2, \ldots, K_L$ are known to the adversary. In this case, security against substitution attacks does not hold. However, if we replace the output of $\mathsf{ReSamp}(i, K_{\neq i}, T)$ for K_i and open the corresponding randomness, the adversary can not tell the difference due to the strongness of the cross-authentication code. Consequently, security against substitution attacks works: given $K_{\neq i}$ and T, the adversary can not forge a T' such that $T \neq T'$ and $\mathsf{XVer}(K_i, i, T') = 1$ with non-negligible probability.

Details are as follows. With the help of algorithm ReSamp of strong L-cross-authentication code XAC, we construct an NC-CCA simulator S' as follows.

Simulator S':

- $S'_1(pk, 1^{|M|})$: Parse $pk = (hpk, \mathsf{H})$. For $i \in [L]$, choose $\widetilde{W_i} \leftarrow \mathcal{R}_{\mathsf{SampleL}}$ and set $X_i := \mathsf{SampleL}(\mathcal{L}_\Lambda; \widetilde{W_i})$. Compute $t := \mathsf{H}(X_1, \ldots, X_L)$. For $i \in [L]$, set $K_i := \mathsf{PubEvl}(hpk, X_i, \widetilde{W_i}, t)$. Set $T = \mathsf{XAuth}(K_1, \ldots, K_L)$. Return the ciphertext $C = (X_1, \ldots, X_L, T)$.
- $S'_2(M)$: Parse $M = (M_1, \ldots, M_L)$. For $i \in [L]$, if $M_i = 1$, set $W_i := \widetilde{W_i}$, $R_i^{\mathcal{X}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$ and $R_i^{\mathcal{K}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$; if $M_i = 0$, generate $(W_i, R_i^{\mathcal{X}_\Lambda})$ by $W_i \leftarrow \mathcal{R}_{\mathsf{SampleL}}$ and $R_i^{\mathcal{X}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{X}_\Lambda, X_i)$, and generate $R_i^{\mathcal{K}_\Lambda}$ with the following method: Run $K'_i \leftarrow \mathsf{ReSamp}(i, K_{\neq i}, T)$, set $R_i^{\mathcal{K}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{K}_\Lambda, K'_i)$ and update $K_i := K'_i$. Finally, return the randomness $R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i \in [L]}$.

With the help of simulator S', we have the following result.

Theorem 3. *Let SMP be a hard subset membership problem with subset sparseness, and EHPS be its corresponding perfectly 2-universal extended hash proof system. For any $L > 1$, assuming that XAC is a strong L-cross-authentication code, the new FHKW scheme is NC-CCA secure.*

Before going into the formal proof, we briefly give a high-level description of the following game-based security proof. This proof is similar to that proposed by Fehr *et al.* (2010), but we utilize the strongness of XAC to help guarantee NC-CCA security, avoiding the problem pointed out in Section 4.

We start with the real experiment $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$, for any PPT adversary A, and let Game -2 denote $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$. First of all, as in the proof in the work of Fehr *et al.* (2010), we exclude some collisions from Game -2 to Game 0. It is easy to see that Game -2 and Game 0 are indistinguishable. Then, from Game 0 to Game L, we stepwise replace the challenge ciphertexts $C^* = (X_1^*, \ldots, X_L^*, T^*)$ and randomness $R^* = (R_1^*, \ldots, R_L^*)$ with those generated by simulator S', where $R_i^* = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})$ for $i \in [L]$. More specifically, for $0 \leq m \leq L$, Game m coincides with Game 0 except that X_i^*, K_i^* and R_i^*, for all $i \leq m$, are all generated by S'. Note that Game L is identical to the simulated experiment $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. Therefore, what remains is to prove that, for $m \in \{0, 1, 2, \ldots, L-1\}$, Game m and Game $m+1$ are indistinguishable. We will show that the strongness of XAC is essential to this indistinguishability.

Note that the differences between Game m and Game $m+1$ lie in X_{m+1}^*, K_{m+1}^* and R_{m+1}^*. Similar to the proof of Theorem 3 in the work of Fehr *et al.* (2010), we proceed with the proof in a series of games. Let Game $m.1$ denote Game m. In Game $m.2$, we modify the decryption oracle $\mathsf{Dec}(sk, \cdot)$ such that it does not make any use of hsk, i.e., for a decryption query C, rather than verifying tag T, $\mathsf{Dec}(sk, \cdot)$ returns $M_i = 0$ directly if $X_i \notin \mathcal{L}_\Lambda$. Two properties, the perfect 2-universality of EHPS and the security of XAC against impersonation attacks, guarantee that Game $m.2$ and Game $m.1$ are statistically indistinguishable. Note that Game $m.2$ is inefficient. In Game $m.3$, if $M_{m+1}^* = 0$, instead of uniformly choosing, set $K_{m+1}^* = \mathsf{SecEvl}(hsk, X_{m+1}^*, t^*)$. The subset sparseness of SMP and the perfect 2-universality of EHPS guarantee that Game $m.3$ and Game $m.2$ are statistically indistinguishable. In Game $m.4$, we modify the way of computing K_{m+1}^* again, i.e., if $M_{m+1}^* = 0$, compute $K_{m+1}^* \leftarrow \mathsf{ReSamp}(m+1, K_{\neq m+1}^*, T^*)$.

The strongness of XAC guarantees that Game $m.4$ and Game $m.3$ are statistically indistinguishable. In Game $m.5$, we modify the decryption oracle $\mathsf{Dec}(sk, \cdot)$ such that it works with the original decryption rule. The perfect 2-universality of EHPS and the security of XAC against impersonation attacks and substitution attacks of XAC guarantee that Game $m.5$ and Game $m.4$ are statistically indistinguishable. Note that Game $m.5$ is efficient. In Game $m.6$, we modify the way of generating X^*_{m+1}, i.e., choose X^*_{m+1} uniformly random from $\mathcal{L}_\Lambda$ no matter whether M^*_{m+1} is 0 or 1. The hardness of SMP guarantees that Game $m.6$ and Game $m.5$ are computationally indistinguishable. Game $m.6$ is identical to Game $m+1$. Hence, we have the conclusion that Game m is indistinguishable from Game $m+1$.

The formal proof is as follows.

Proof. Our aim is to prove that, for any PPT adversary A, the simulated experiment $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$ is computationally indistinguishable from the real experiment $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$. Technically, we denote the challenge ciphertext and its related plaintext by C^* and M^*, and write $C^* := (X^*_1, \dots, X^*_L, T^*)$ and $M^* := (M^*_1, \dots, M^*_L)$. Denote A's j-th decryption query by $C^j := (X^j_1, \dots, X^j_L, T^j)$, the corresponding plaintext by $M^j = (M^j_1, \dots, M^j_L)$, and define t^*, t^j, K^*_i and K^j_i similarly. Define $\overline{K^*_i} := \mathsf{SecEvl}(hsk, X^*_i, t^*)$, $\overline{K_i}^j := \mathsf{SecEvl}(hsk, X^j_i, t^j)$ and denote the final output of A in Game i by $output_{A,i}$. Without loss of generality, we assume that A always makes q decryption queries, where $q = \text{poly}(k)$.

Game -2: Game -2 is the real experiment $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$. Hence

$$\Pr\left[output_{A,-2} = 1\right] = \Pr\left[\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k) = 1\right]. \tag{8}$$

Game -1: Game -1 is the same as Game -2, except that, in the challenge ciphertext generation, the experiment aborts (with A outputting 1) if there exist some distinct $i, i' \in [L]$ such that $X^*_i = X^*_{i'}$. By a union bound, we have that

$$\begin{aligned}|\Pr\left[output_{A,-1} = 1\right] - \Pr\left[output_{A,-2} = 1\right]|& \\ \leq \frac{L(L-1)}{2|\mathcal{L}_\Lambda|}.&\end{aligned} \tag{9}$$

Game 0: Game 0 is the same as Game -1, except for the decryption oracle. In Game 0, if A makes a decryption query C^j with $(X^j_1, \dots, X^j_L) \neq (X^*_1, \dots, X^*_L)$ and $t^j = \mathsf{H}(X^j_1, \dots, X^j_L) = \mathsf{H}(X^*_1, \dots, X^*_L) = t^*$, the experiment aborts (without loss of generality, A outputs 1). Since H is a collision-resistant hash function, we have that

$$\begin{aligned}|\Pr\left[output_{A,0} = 1\right] - \Pr\left[output_{A,-1} = 1\right]|& \\ \leq \mathbf{Adv}^{cr}_{\mathcal{H},A'}(k)&\end{aligned} \tag{10}$$

for a suitable PPT algorithm A'.

■

In the remainder, we will use a hybrid argument to finish this proof. From Game 0 to Game L, we will replace the challenge ciphertext C^* and its related randomness R^* with those generated by simulator S' step by step. Specifically, for any $0 \leq m \leq L$, Game m is identical to Game 0, except that, for any $i \leq m$, X^*_i, K^*_i and their related randomness are all generated by simulator S'. Note that, in Game L, the whole challenge ciphertext C^* and the whole randomness R^* are both generated by simulator S'.

Looking ahead, if we can prove that, for any $0 \leq m \leq L-1$, Game m and Game $m+1$ are indistinguishable, we will have that Game 0 and Game L are indistinguishable. So Game -2, which is $\mathsf{Exp}^{\text{NC-CCA-real}}_{\text{FHKW},A}(k)$, and Game L are indistinguishable. Note that Game L is indistinguishable from $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$. That is because if, in Game L, we reverse the changes from Game 0 and Game -1, we will get $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$. This finishes the whole proof.

Now we prove that, for any $0 \leq m \leq L-1$, Game m and Game $m+1$ are indistinguishable. This is through a series of indistinguishable games as well.

Game $m.1$: Game $m.1$ is identical with Game m.

Game $m.2$: Game $m.2$ is the same as Game $m.1$, except for the decryption oracle. In Game $m.2$, for any decryption query $C^j = (X^j_1, \dots, X^j_L, T^j)$ and for any $i \in [L]$, the challenger will return $M^j_i = 0$ directly if $X^j_i \notin \mathcal{L}_\Lambda$, and behave just as in Game $m.1$, otherwise compute $\overline{K_i}^j = \mathsf{SecEvl}(hsk, X^j_i, t^j)$, and return $M^j_i = \mathsf{XVer}(\overline{K_i}^j, i, T^j)$. Note that the decryption oracle in Game $m.2$ is inefficient and it does not leak any information on hsk beyond hpk.

Let $\mathsf{bad}_{m.2}$ (resp. $\mathsf{bad}_{m.1}$) denote the event that, in Game $m.2$ (resp. Game $m.1$), A makes some decryption query C^j such that there is an $X^j_i \notin \mathcal{L}_\Lambda$ but $\mathsf{XVer}(\overline{K_i}^j, i, T^j) = 1$. Note that $\Pr[\mathsf{bad}_{m.2}] = \Pr[\mathsf{bad}_{m.1}]$ and that Game $m.2$ and Game $m.1$ are identical unless $\mathsf{bad}_{m.2}$ or $\mathsf{bad}_{m.1}$ occurs. We present the following lemma with a postponed proof.

Lemma 2. $\Pr[\mathsf{bad}_{m.2}] \leq qL \cdot \mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)$.

With the lemma, we have that

$$\begin{aligned}&|\Pr\left[output_{A,m.2} = 1\right] - \Pr\left[output_{A,m.1} = 1\right]| \\ &\quad \leq \Pr\left[\mathsf{bad}_{m.2}\right] \\ &\quad \leq qL \cdot \mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k).\end{aligned} \tag{11}$$

Game $m.3$: Game $m.3$ is the same as Game $m.2$, except for the generation of K^*_{m+1} in the challenge ciphertext. In this game, set $K^*_{m+1} := \mathsf{SecEvl}(hsk, X^*_{m+1}, t^*)$ if $M^*_{m+1} = 0$, and the randomness of K^*_{m+1} is opened as $\mathsf{Explain}(\mathcal{K}_\Lambda, K^*_{m+1})$. When $M^*_{m+1} = 0$, X^*_{m+1} is chosen from $\mathcal{X}_\Lambda$. If $X^*_{m+1} \notin \mathcal{L}_\Lambda$, the perfect 2-universality of EHPS implies K^*_{m+1} is uniformly distributed over $\mathcal{K}_\Lambda$, which is exactly like Game $m.2$. Let $\mathsf{sub}_{m.3}$ (resp. $\mathsf{sub}_{m.2}$) denote the event that $X^*_{m+1} \in \mathcal{L}_\Lambda$ given $M^*_{m+1} = 0$ in Game $m.3$ (resp. Game $m.2$). Note that $\Pr[\mathsf{sub}_{m.3}] = \Pr[\mathsf{sub}_{m.2}]$ and that Game $m.3$ and Game $m.2$ are the same unless $\mathsf{sub}_{m.3}$ or $\mathsf{sub}_{m.2}$ occurs. So we have that

$$\begin{aligned}&|\Pr\left[output_{A,m.3} = 1\right] - \Pr\left[output_{A,m.2} = 1\right]|\\ &\quad\le \Pr\left[\mathsf{sub}_{m.2}\right]\\ &\quad= \frac{|\mathcal{L}_\Lambda|}{|\mathcal{X}_\Lambda|}.\end{aligned} \tag{12}$$

Game $m.4$: Game $m.4$ is the same as Game $m.3$, except for the generation of K^*_{m+1} in the challenge ciphertext. In this game, the way of computing K^*_{m+1} is modified again. If $M^*_{m+1} = 0$, compute $K^*_{m+1} \leftarrow \mathsf{ReSamp}(m+1, K^*_{\neq m+1}, T^*)$. The randomness of K^*_{m+1} is still opened as $\mathsf{Explain}(\mathcal{K}_\Lambda, K^*_{m+1})$. The strongness of XAC guarantees that K^*_{m+1} in Game $m.4$ and K^*_{m+1} in Game $m.3$ are statistically indistinguishable. Hence,

$$\begin{aligned}&|\Pr\left[output_{A,m.4} = 1\right] - \Pr\left[output_{A,m.3} = 1\right]|\\ &\qquad\le \mathrm{Dist}(k),\end{aligned} \tag{13}$$

where $\mathrm{Dist}(k)$ is the statistical distance between K^*_{m+1} in Game $m.4$ and K^*_{m+1} in Game $m.3$.

Game $m.5$: Game $m.5$ is the same as Game $m.4$, except that the decryption oracle works with the original decryption rule. In Game $m.5$, for any decryption query $C^j = (X^j_1, \ldots, X^j_L, T^j)$, the challenger computes $\overline{K_i}^j = \mathsf{SecEvl}(hsk, X^j_i, t^j)$, and returns $M^j_i = \mathsf{XVer}(\overline{K_i}^j, i, T^j)$. Note that the decryption oracle in Game $m.5$ is efficient again. Similarly, let $\mathsf{bad}_{m.5}$ (resp. $\mathsf{bad}_{m.4}$) denote the event that, in Game $m.5$ (resp. Game $m.4$), A makes some decryption query C^j such that there is an $X^j_i \notin \mathcal{L}_\Lambda$ but $\mathsf{XVer}(\overline{K_i}^j, i, T^j) = 1$. Note that $\Pr[\mathsf{bad}_{m.5}] = \Pr[\mathsf{bad}_{m.4}]$ and that Game $m.5$ and Game $m.4$ are identical unless $\mathsf{bad}_{m.5}$ or $\mathsf{bad}_{m.4}$ occurs. We present the following lemma with a postponed proof.

Lemma 3. *We have*

$$\begin{aligned}&\Pr[\mathsf{bad}_{m.4}]\\ &\quad\le qL \cdot \max\{\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k), \mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)\}.\end{aligned} \tag{14}$$

With this lemma, we have that

$$\begin{aligned}&|\Pr\left[output_{A,m.5} = 1\right] - \Pr\left[output_{A,m.4} = 1\right]|\\ &\quad\le \Pr\left[\mathsf{bad}_{m.4}\right]\\ &\quad\le qL \cdot \max\{\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k), \mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)\}.\end{aligned} \tag{15}$$

Game $m.6$: Game $m.6$ is the same as Game $m.5$, except that, in the challenge ciphertext generation, the challenger chooses $X^*_{m+1} \leftarrow \mathcal{L}_\Lambda$ no matter whether M^*_{m+1} is 0 or 1, and X^*_{m+1} is opened as $\mathsf{Explain}(\mathcal{X}_\Lambda, X^*_{m+1})$, if $M^*_{m+1} = 0$. Now the subset membership problem SMP can be reduced to the problem of efficiently distinguishing Game $m.6$ from Game $m.5$. We have that

$$\begin{aligned}&|\Pr\left[output_{A,m.6} = 1\right] - \Pr\left[output_{A,m.5} = 1\right]|\\ &\qquad\le \mathbf{Adv}_{\mathsf{SMP},A''}(k)\end{aligned} \tag{16}$$

for a suitable PPT algorithm A''.

Combining the above results, we have that Game $m.1$ and Game $m.6$ are indistinguishable. Now that Game $m.6$ is identical to Game $m+1$, we have that Game m and Game $m+1$ are indistinguishable. What remains is to prove Lemmas 2 and 3.

Proof. (Lemma 2) Let $\mathsf{bad}^j_{m.2.i}$ denote the event that A's j-th decryption query

$$C^j = (X^j_1, \ldots, X^j_L, T^j)$$

satisfies that $X^j_i \notin \mathcal{L}_\Lambda$ but $\mathsf{XVer}(\overline{K_i}^j, i, T^j) = 1$ in Game $m.2$. In Game $m.2$, A has no information on hsk beyond hpk. For arbitrary $(j,i) \in [q] \times [L]$ and $X^j_i \notin \mathcal{L}_\Lambda$, the perfect 2-universality of EHPS implies that $\overline{K_i}^j = \mathsf{SecEvl}(hsk, X^j_i, t^j)$ is uniformly random in $\mathcal{K}_\Lambda$ from A's point of view. Therefore,

$$\Pr\left[\mathsf{bad}^j_{m.2.i}\right] \le \mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k).$$

Note that

$$\mathsf{bad}_{m.2} = \bigvee_{(j,i)\in[q]\times[L]} \mathsf{bad}^j_{m.2.i}.$$

By a union bound, we have that

$$\begin{aligned}\Pr\left[\mathsf{bad}_{m.2}\right] &\le \sum_{(j,i)\in[q]\times[L]} \Pr\left[\mathsf{bad}^j_{m.2.i}\right]\\ &\le qL \cdot \mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k).\end{aligned} \tag{17}$$

■

Proof. (Lemma 3) Let $\mathsf{bad}^j_{m.4.i}$ denote the event that A's j-th decryption query $C^j = (X^j_1, \dots, X^j_L, T^j)$ satisfies that $X^j_i \notin \mathcal{L}_\Lambda$ but $\mathsf{XVer}(\overline{K_i}^j, i, T^j) = 1$ in Game $m.4$. Let K^{hsk}_{m+1} denote the random variable $\mathsf{SecEvl}(hsk, X^*_{m+1}, t^*)$.

For arbitrary fixed $(j, i) \in [q] \times [L]$, we only consider $X^j_i \notin \mathcal{L}_\Lambda$ (otherwise there is nothing to prove). If $(X^j_i, t^j) \neq (X^*_{m+1}, t^*)$, the perfect 2-universality of EHPS implies that $\overline{K_i}^j = \mathsf{SecEvl}(hsk, X^j_i, t^j)$ is uniformly random in $\mathcal{K}_\Lambda$ from A's point of view. Hence,

$$\Pr\left[\mathsf{bad}^j_{m.4.i} \mid (X^j_i, t^j) \neq (X^*_{m+1}, t^*)\right] \leq \mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k). \quad (18)$$

If $(X^j_i, t^j) = (X^*_{m+1}, t^*)$, then $(X^j_1, \dots, X^j_L) = (X^*_1, \dots, X^*_L)$, since Game 0 excludes hash collisions. The decryption query C^j is not equal to the challenge ciphertext, so $T^j \neq T^*$. Note that, in this case, $\overline{K_i}^j = K^{hsk}_{m+1}$. What the adversary knows is given by $(K^*_1, \dots, K^*_m, K^*_{m+1}, K^*_{m+2}, \dots, K^*_L)$ and T^*.

However, $K^*_{m+1} = \mathsf{ReSamp}(m+1, K^*_{\neq m+1}, T^*)$, which means that A's information can be characterized by $K^*_{\neq m+1}$ and T^*. The security against substitution attack of XAC guarantees that, given $K^*_{\neq m+1}$ and T^*, A produces a $T^j \neq T^*$ such that

$$\mathsf{XVer}(K^{hsk}_{m+1}, i, T^j) = \mathsf{XVer}(\overline{K_i}^j, i, T^j) = 1$$

with probability at most $\mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$, i.e.,

$$\Pr\left[\mathsf{bad}^j_{m.4.i} \mid (X^j_i, t^j) = (X^*_{m+1}, t^*)\right] \leq \mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k).$$

Therefore, we have that

$$\Pr\left[\mathsf{bad}^j_{m.4.i}\right] \leq \max\{\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k), \mathbf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)\}.$$

Lemma 3 follows from a union bound. ■

Remark 4. Recall that Game $m.4$ is missing in the original security proof of the FHKW scheme (Fehr *et al.*, 2010). Without the employment of algorithm ReSamp in Game $m.4$, we will have $K^*_{m+1} = \mathsf{SecEvl}(hsk, X^*_{m+1}, t^*)$. Then the simulator has to present the adversary the randomness corresponding to K^*_{m+1}. Consequently, the adversary is able to recover $K^*_{m+1} = \mathsf{SecEvl}(hsk, X^*_{m+1}, t^*)$ from the randomness. But security against substitution attacks of the L-cross-authentication code assumes that the adversary knows nothing about K^*_{m+1} except for $(K^*_{\neq m+1}, T^*)$. That is why the original security proof (Fehr *et al.*, 2010) fails, and why ours can go through.

8. Conclusion

We provided a security analysis of the FHKW scheme (Fehr *et al.*, 2010), and showed that the original simulator constructed by Fehr *et al.* (2010) is not sufficient to prove NC-CCA security. We provided a refined version of the FHKW scheme for a single bit and proved its NC-CCA security. Our scheme does not involve any cross-authentication code, avoiding the security problem that annoys the FHKW scheme. To fix the security proof of the FHKW scheme, we introduced the notion of strong cross-authentication code, applied it to the FHKW scheme, and proved that the new version of the FHKW scheme is NC-CCA secure for multi-bit plaintexts.

Open questions:

(i) The failure of the simulator proposed by Fehr *et al.* (2010) does not rule out the existence of other simulators working properly for the NC-CCA security proof of the FHKW scheme. Therefore, it is still open whether the original version of the FHKW scheme is NC-CCA secure or not.

(ii) Even if the original version of the FHKW scheme is not NC-CCA secure, it might still possess SIM-SO-CCA security. Hence, another question is whether it is SIM-SO-CCA secure or not.

(iii) It can be interesting to construct an NC-CCA secure PKE encrypting multiple bits from an NC-CCA secure PKE encrypting single bits. This question in the relaxed setting of IND-CCA2 has been answered by Myers and Shelat (2009). But the selective opening scenario is much more complicated, and we believe that the problem is much harder.

(iv) The last open question is whether every cross-authentication code is also a strong one, as discussed in Section 6.

Acknowledgment

This work is supported by NSFC (Nos. 61170229, 61133014, 61373153), Innovation Project (No.12ZZ021) of the Shanghai Municipal Education Commission, Specialized Research Fund (No. 20110073110016) for the Doctoral Program of Higher Education, and the Open Project Program (No.2013A01) of the State Key Laboratory of MEAC.

This is a full paper combining two conference papers, one featured in *Public-Key Cryptography (PKC 2013)* and the other in *Intelligent Networking and Collaborative Systems (INCoS 2013)*.

References

Bellare, M., Dowsley, R., Waters, B. and Yilek, S. (2012). Standard security does not imply security against selective-opening, *in* D. Pointcheval and T. Johansson (Eds.), *Advances in Cryptology—EUROCRYPT 2012*, Springer, Berlin/Heidelberg, pp. 645–662.

Bellare, M., Hofheinz, D. and Yilek, S. (2009). Possibility and impossibility results for encryption and commitment secure under selective opening, *in* A. Joux (Ed.), *Advances in Cryptology—EUROCRYPT 2009*, Springer, Berlin/Heidelberg, pp. 1–35.

Bellare, M., Waters, B. and Yilek, S. (2011). Identity-based encryption secure against selective opening attack, *in* Y. Ishai (Ed.), *Theory of Cryptography*, Springer, Berlin/Heidelberg, pp. 235–252.

Böhl, F., Hofheinz, D. and Kraschewski, D. (2012). On definitions of selective opening security, *in* M. Fischlin, J. Buchmann and M. Manulis (Eds.), *Public Key Cryptography—PKC 2012*, Springer, Berlin/Heidelberg, pp. 522–539.

Canetti, R., Friege, U., Goldreich, O. and Naor, M. (1996). Adaptively secure multi-party computation, *Technical report*, Massachusetts Institute of Technology, Cambridge, MA.

Cramer, R. and Shoup, V. (2002). Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, *in* L.R. Knudsen (Ed.), *Advances in Cryptology—EUROCRYPT 2002*, Springer, Berlin/Heidelberg, pp. 45–64.

Fehr, S., Hofheinz, D., Kiltz, E. and Wee, H. (2010). Encryption schemes secure against chosen-ciphertext selective opening attacks, *in* H. Gilbert (Ed.), *Advances in Cryptology—EUROCRYPT 2010*, Berlin/Heidelberg, Springer, pp. 381–402.

Gao, C.-z., Xie, D. and Wei, B. (2012). Deniable encryptions secure against adaptive chosen ciphertext attack, *in* M.D. Ryan, B. Smyth and G. Wang (Eds.), *Information Security Practice and Experience*, Springer, Berlin/Heidelberg, pp. 46–62.

Hemenway, B., Libert, B., Ostrovsky, R. and Vergnaud, D. (2011). Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security, *in* D.H. Lee and X. Wang (Eds.), *Advances in Cryptology—ASIACRYPT 2011*, Springer, Berlin/Heidelberg, pp. 70–88.

Hofheinz, D. (2012). All-but-many lossy trapdoor functions, *in* D. Pointcheval and T. Johansson (Eds.), *Advances in Cryptology—EUROCRYPT 2012*, Springer, Berlin/Heidelberg, pp. 209–227.

Myers, S. and Shelat, A. (2009). Bit encryption is complete, *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS'09, Atlanta, GA, USA*, pp. 607–616.

Peikert, C. and Waters, B. (2011). Lossy trapdoor functions and their applications, *SIAM Journal on Computing* **40**(6): 1803–1844.

Zhengan Huang received his B.Sc. and M.Sc. degrees from the Department of Mathematics, Sun Yat-sen University, in 2009 and 2011, respectively. Currently, he is a Ph.D. candidate at Shanghai Jiao Tong University, Shanghai, China. His research interests include public-key cryptography and information security.

Shengli Liu obtained her Bachelor's, Master's and Ph.D. degrees from Xidian University in 1995, 1998 and 2000, respectively. From 2000 till 2002, she continued her research on cryptography and received another Ph.D. degree at Technische Universiteit Eindhoven, the Netherlands. In 2002, she joined the Department of Computer Science and Engineering, Shanghai Jiao Tong University. She is now a professor and her research interests include ID-based cryptography, pairing-based cryptosystems, and information-theoretic security.

Baodong Qin received the M.Sc. degree in 2007 from Shandong University, China. He now is a Ph.D. student at Shanghai Jiao Tong University, China. His research interests include theoretic cryptography and information security, particularly the construction of provably secure public-key cryptosystems.

Kefei Chen received the B.Sc. and M.Sc. degrees in applied mathematics from Xidian University, Xi'an, in 1982 and 1985, respectively, and the Ph.D. degree from Justus-Liebig University, Giessen, Germany, in 1994. From 1996 to 2013, he served as a professor at Shanghai Jiao Tong University. He is now a professor of Hangzhou Normal University. His fields of interest are public key cryptography, cryptographic protocol analysis and automatic verifying, as well as network security.

Appendix A
When algorithm XAuth is probabilistic

In Section 4.3, we claimed that, if algorithm XAuth of XAC in the FHKW scheme is probabilistic, with the aforementioned simulator S in Section 4, the FHKW scheme cannot be proved NC-CCA secure for any positive integer L. Now we show the reason.

Firstly, a slight modification to XAuth is needed. Because XAuth is probabilistic, there exists an inner random number R^{XAuth} used by XAuth during the encryption process (i.e., $T \leftarrow \mathsf{XAuth}(K_1, \ldots, K_L; R^{\mathsf{XAuth}})$). Note that the aforementioned simulator S should output randomness $R = ((W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}, R^{\mathsf{XAuth}})$ according to the ciphertext C and its related plaintext M. In the mean time, the original simulator S can recover $(W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}$. Therefore, S should generate R^{XAuth} according to T and $(K_1,\dots,K_L)$, which can be recovered from $R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}$. Now we make a modification to XAuth: we require that XAuth be efficiently "explainable", which means that there is an efficient algorithm $\mathsf{Explain}_{\mathsf{XAuth}}$ such that $R^{\mathsf{XAuth}} \leftarrow \mathsf{Explain}_{\mathsf{XAuth}}((K_1,\dots,K_L),T)$. For simplicity, we still use the original notation S and XAuth after this modification.

Secondly, with the above modification, consider our main conclusion of this appendix. As the proof of Theorem 1, our aim is to construct an adversary $A = (A_1, A_2)$ to distinguish the two experiments $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. The adversary A is the same as the one in the proof of Theorem 1, except that, in the decryption query stage, instead of choosing a random K_1', the adversary A uses the original K_1, which can be recovered from randomness $R = ((W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}, R^{\mathsf{XAuth}})$. More specifically, in the first stage, A_1 returns $M = (0,\cdots,0)$ to the challenger, and in the second stage, upon receiving the ciphertext $C = (X_1,\dots,X_L,T)$ and randomness R, A_2 recovers $(K_1,\dots,K_L)$ from R, computes $T' \leftarrow \mathsf{XAuth}(K_1,\dots,K_L;\widetilde{R}^{\mathsf{XAuth}})$, where $\widetilde{R}^{\mathsf{XAuth}}$ is uniformly random chosen from $\mathcal{R}_{\mathsf{XAuth}}$, and returns $C' = (X_1,\dots,X_L,T')$ as his decryption query. Because XAuth is probabilistic, it is very easy for A to get a $T' \neq T$ with the above method. As a result, with an overwhelming probability, A_2 will receive $M' = (0,\cdots,0)$ as the decryption result of C' in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$, and receive $M' = (1,\cdots,1)$ in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. Hence, A can distinguish $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$.

Appendix B
Proof of Theorem 2

Proof. First, we construct a simulator $S_{\mathcal{E}}$ for scheme $\mathcal{E} = (\mathsf{Gen}_{\mathcal{E}}, \mathsf{Enc}_{\mathcal{E}}, \mathsf{Dec}_{\mathcal{E}})$.

Simulator $S_{\mathcal{E}}$:

- $S_{\mathcal{E}1}(pk, 1)$: With $pk = hpk$, choose $\widetilde{W} \leftarrow \mathcal{R}_{\mathsf{SampleL}}$ and set $X := \mathsf{SampleL}(\mathcal{L}_\Lambda;\widetilde{W})$. Then set $K := \mathsf{PubEvl}(hpk, X, \widetilde{W})$. Return the ciphertext $C = (X, K)$.
- $S_{\mathcal{E}2}(M)$: If $M = 1$, set $W := \widetilde{W}$ and choose $R^{\mathcal{X}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$, $R^{\mathcal{K}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$; otherwise choose $W \leftarrow \mathcal{R}_{\mathsf{SampleL}}$, and set $R^{\mathcal{X}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{X}_\Lambda, X)$, $R^{\mathcal{K}_\Lambda} \leftarrow \mathsf{Explain}(\mathcal{K}_\Lambda, K)$. Return the randomness $R = (W, R^{\mathcal{X}_\Lambda}, R^{\mathcal{K}_\Lambda})$.

With simulator $S_{\mathcal{E}}$, we will show that, for any PPT adversary A, the two experiments $\mathsf{Exp}_{\mathcal{E},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathcal{E},A}^{\text{NC-CCA-Sim}}(k)$ are computationally indistinguishable through a series of indistinguishable games. Technically, we denote the challenge ciphertext and its corresponding plaintext by C^* and M^*, and write $C^* := (X^*, K^*)$. Without loss of generality, we assume that A always makes q decryption queries, where $q = \mathrm{poly}(k)$. For $j \in [q]$, denote A's j-th decryption query by $C^j := (X^j, K^j)$ and let its corresponding plaintext be M^j. At the same time, we define $\overline{K^*} := \mathsf{SecEvl}(hsk, X^*)$, $\overline{K^j} := \mathsf{SecEvl}(hsk, X^j)$ for $j \in [q]$, and denote the final output of A in Game i by $output_{A,i}$.

Game 0: Game 0 is the real experiment $\mathsf{Exp}_{\mathcal{E},A}^{\text{NC-CCA-Real}}(k)$. By our notation above,

$$\Pr\left[output_{A,0} = 1\right] = \Pr\left[\mathsf{Exp}_{\mathcal{E},A}^{\text{NC-CCA-Real}}(k) = 1\right]. \tag{B1}$$

Game 1: Game 1 is the same as Game 0, except for the decryption oracle. In Game 1, for any decryption query $C^j = (X^j, K^j)$ made by A, if $X^j \notin \mathcal{L}_\Lambda$, the challenger will return $M^j = 0$ directly, and if $X^j \in \mathcal{L}_\Lambda$, the challenger will answer the query as in Game 0: compute $\overline{K^j} = \mathsf{SecEvl}(hsk, X^j)$, and if $\overline{K^j} = K^j$, return $M^j = 1$, otherwise return $M^j = 0$. Note that the decryption oracle in Game 1 is inefficient and it doesn't leak any information on hsk beyond hpk. Let bad_i denote the event that in Game i, A makes some decryption query $C^j = (X^j, K^j)$ such that $X^j \notin \mathcal{L}_\Lambda$ and $K^j = \overline{K^j}$. Note that $\Pr[\mathsf{bad}_1] = \Pr[\mathsf{bad}_0]$ and that Game 1 and Game 0 are identical unless events bad_1 or bad_0 occurs. By the perfect 2-universality of HPS and a union bound, $\Pr[\mathsf{bad}_1] = \Pr[\mathsf{bad}_0] \le q/|\mathcal{K}_\Lambda|$. So we have

$$\begin{aligned} &|\Pr\left[output_{A,1} = 1\right] - \Pr\left[output_{A,0} = 1\right]| \\ &\quad \le \Pr\left[\mathsf{bad}_1\right] \\ &\quad = \frac{q}{|\mathcal{K}_\Lambda|}. \end{aligned} \tag{B2}$$

Game 2: Game 2 is the same as Game 1, except that, in the challenge ciphertext generation, set $K^* = \mathsf{SecEvl}(hsk, X^*)$ for $M^* = 0$, and then the randomness of K^* is opened as $\mathsf{Explain}(\mathcal{K}_\Lambda, K^*)$. In Game 1, if $M^* = 0$, K^* also can be seen as being opened by $\mathsf{Explain}(\mathcal{K}_\Lambda, K^*)$. In Game 2, since the only information on hsk beyond hpk is released in the computation of K^*, the perfect 2-universality of HPS implies that, if $X^* \notin \mathcal{L}_\Lambda$, K^* is uniformly distributed in $\mathcal{K}_\Lambda$. Let sub_i denote the event that in Game i, when $M^* = 0$, $X^* \in \mathcal{L}_\Lambda$. Note that $\Pr[\mathsf{sub}_2] = \Pr[\mathsf{sub}_1]$ and that Game 2 and Game 1 are the same unless events sub_2 or sub_1 occurs. So we have

$$\begin{aligned}|\Pr\left[output_{A,2}=1\right]-\Pr\left[output_{A,1}=1\right]| &\\ \leq \Pr\left[\mathsf{sub}_2\right] &\\ =\frac{|\mathcal{L}_\Lambda|}{|\mathcal{X}_\Lambda|}. &\end{aligned}\tag{B3}$$

Game 3: Game 3 is the same as Game 2, except that the decryption oracle works with the original decryption rule. In Game 3, for any decryption query $C^j=(X^j,K^j)$, the challenger sets $\overline{K^j}=\mathsf{SecEvl}(hsk,X^j)$, then returns $M^j=1$ if $\overline{K^j}=K^j$, or returns $M^j=0$ if $\overline{K^j}\neq K^j$. Note that the decryption oracle in Game 3 is efficient. Similarly, bad_i denotes the event that in Game i, A makes some decryption query $C^j=(X^j,K^j)$ such that $X^j\notin\mathcal{L}_\Lambda$ and $K^j=\overline{K^j}$. Note that $\Pr[\mathsf{bad}_3]=\Pr[\mathsf{bad}_2]$ and that Game 3 and Game 2 are identical unless events bad_3 or bad_2 occurs. Since the only information on hsk beyond hpk is released in the computation of K^*, by the perfect 2-universality of HPS and a union bound, $\Pr[\mathsf{bad}_3]=\Pr[\mathsf{bad}_2]=q/|\mathcal{K}_\Lambda|$. So

$$\begin{aligned}|\Pr\left[output_{A,3}=1\right]-\Pr\left[output_{A,2}=1\right]| &\\ \leq \Pr\left[\mathsf{bad}_3\right] &\\ =\frac{q}{|\mathcal{K}_\Lambda|}. &\end{aligned}\tag{B4}$$

Game 4: Game 4 is the same as Game 3, except that, in the challenge ciphertext generation, the challenger chooses $X^*\leftarrow\mathcal{L}_\Lambda$ if $M^*=0$. That is to say, choose $X^*\leftarrow\mathcal{L}_\Lambda$ no matter whether M^* is 0 or 1, and X^* is opened as $\mathsf{Explain}(\mathcal{X}_\Lambda,X^*)$ if $M^*=0$. Since SMP is hard,

$$\begin{aligned}|\Pr\left[output_{A,4}=1\right]-\Pr\left[output_{A,3}=1\right]| &\\ \leq \mathbf{Adv}_{\mathsf{SMP},A}(k). &\end{aligned}\tag{B5}$$

Combining all the above results, we have

$$\begin{aligned}|\Pr\left[output_{A,0}=1\right]-\Pr\left[output_{A,4}=1\right]| &\\ \leq \frac{2q}{|\mathcal{K}_\Lambda|}+\frac{|\mathcal{L}_\Lambda|}{|\mathcal{X}_\Lambda|}+\mathbf{Adv}_{\mathsf{SMP},A}(k). &\end{aligned}\tag{B6}$$

Note that Game 4 is just the experiment $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\mathcal{E},A}(k)$. So we have

$$\begin{aligned}&\mathbf{Adv}^{\text{NC-CCA}}_{\mathcal{E},A,S}(k)\\ &=|\Pr\left[\mathsf{Exp}^{\text{NC-CCA-Real}}_{\mathcal{E},A}(k)=1\right]\\ &\quad-\Pr\left[\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\mathcal{E},A}(k)=1\right]|\\ &\leq\frac{2q}{|\mathcal{K}_\Lambda|}+\frac{|\mathcal{L}_\Lambda|}{|\mathcal{X}_\Lambda|}+\mathbf{Adv}_{\mathsf{SMP},A}(k).\end{aligned}\tag{B7}$$

■

Received: 3 January 2014
Revised: 10 June 2014