Sender Equivocable Encryption Schemes Secure against Chosen-Ciphertext Attacks Revisited

Zhengan Huang[1], Shengli Liu[1], and Baodong Qin[1,2]

1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2. College of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621000, China
{hzayusuo5288, slliu, qinbaodong}@sjtu.edu.cn *

Abstract. In Eurocrypt 2010, Fehr et al. proposed the first sender equivocable encryption scheme secure against chosen-ciphertext attack (NC-CCA) and proved that NC-CCA security implies security against selective opening chosen-ciphertext attack (SO-CCA). The NC-CCA security proof of the scheme relies on security against substitution attack of a new primitive, "cross-authentication code". However, the security of cross-authentication code can not 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. This random information can be used to recover all the keys involved in cross-authentication code, and forge a ciphertext (like a substitution attack of cross-authentication code) that is different from but related to the challenge ciphertext. And the response of decryption oracle, with respect to the forged ciphertext, leaks information. This leaked information can be employed by an adversary to spoil the NC-CCA security proof of Fehr et al.'s scheme encrypting multi-bit plaintext.

In this paper, we provide a security analysis of Fehr et al.'s scheme, showing that its NC-CCA security proof is flawed by presenting an attack. We point out that Fehr et al.'s scheme encrypting single-bit plaintext can be refined to achieve NC-CCA security, free of cross-authentication code. We introduce the strong notion of cross-authentication code, apply it to Fehr et al.'s scheme, and show that the new version of Fehr et al.'s scheme achieves NC-CCA security for multi-bit plaintext.

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

1 Introduction

The notion of sender equivocability for a public-key encryption (PKE) scheme is formalized by Fehr et al.[5] in Eurocrypt 2010. It is an important tool to construct PKE schemes secure against chosen-plaintext/ciphertext selective opening attacks (SO-CPA/CCA secure). 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[3]. In fact, in [5], Fehr et al. have pointed out that sender equivocable encryption secure under chosen-plaintext attack (CPA secure) is a variant of non-committing encryption in [3]. Following the notations in [5], security of a sender equivocable encryption scheme against chosen-plaintext/ciphertext attack is denoted by *NC-CPA/CCA security*.

* Funded by NSFC (No. 61170229), Innovation Project (No.12ZZ021) of Shanghai Municipal Education Commission, and Specialized Research Fund (No. 20110073110016) for the Doctoral Program of Higher Education.

As proved in [5], NC-CPA/CCA security implies simulation-based selective opening security against chosen-plaintext/ciphertext attack (SIM-SO-CPA/CCA security). This fact suggests an alternative way of constructing PKE secure against selective opening attacks, besides the construction from lossy encryption proposed in [2].

Discussion and related work. In Eurocrypt 2009, Bellare et al.[2] formalized the notion of security against selective opening attack (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 adversary. SOA security requires that the unopened ciphertexts remain secure.

In [2], Bellare et al. 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.[1]. Bellare et al.[2] proposed that IND-SO-CPA security and SIM-SO-CPA security can be achieved through a special class of encryption named lossy encryption, and lossy encryption can be constructed from lossy trapdoor functions [9]. Hemenway et al.[8] showed more constructions of lossy encryption. In Eurocrypt 2012, Hofheinz[7] proposed a new primitive called all-but-many lossy trapdoor functions, and achieved IND-SO-CCA security and SIM-SO-CCA security from the new primitive.

Fehr et al.[5] presented a totally different way of achieving SIM-SO-CCA security. They formalized the notion of sender equivocability under chosen-plaintext/ciphertext attack (NC-CPA/CCA security), and proved that NC-CPA (resp. NC-CCA) security implies SIM-SO-CPA (resp. SIM-SO-CCA) security. In [5], two PKE schemes were proposed. The first one, constructed from trapdoor one-way permutations, is NC-CPA secure, so it achieves SIM-SO-CPA security. The second one (denoted by the FHKW scheme) is constructed from an extended hash proof system [4] and a new building block proposed by themselves, "cross-authentication code". They proved that the FHKW scheme is NC-CCA secure.

In 2012, Gao et al.[6] presented a deniable encryption construction (denoted by the GXW scheme) utilizing an extended hash proof system of [4] and a cross-authentication code of [5] as ingredients. They utilized similar techniques as those in the FHKW scheme to guarantee the CCA security of their scheme.

However, as we will show in this paper, there is some problem in the security proof of the FHKW scheme. We will present a security analysis of the FHKW scheme and show that NC-CCA security can not be guaranteed. The GXW scheme suffers from the similar security problem. We will offer a refined version of the FHKW scheme for single bit with NC-CCA security. 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 plaintext.

Our contribution. In this paper, we focus on NC-CCA security.

- We provide a security analysis of the FHKW scheme in [5], and show the proof of NC-CCA security in [5] is flawed by showing an attack. The key observation is: 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 in the proof of NC-CCA security in [5] for any positive integer L.

- 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 [5], but only encrypting 1 bit. We provide a simpler encryption scheme for single bit, free of any cross-authentication code.
- We fix the security proof of the FHKW scheme, by introducing the strong notion of L-cross-authentication code. Informally, 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 by notations and definitions in Section 2. We recall the FHKW scheme of [5] in Section 3, and then provide a security analysis of it in Section 4. We present a refined version of the FHKW scheme for single bit in Section 5 and leave the proof in the Appendix. We fix the security proof of the FHKW scheme in Section 6. Finally, we give a summary of our work in Section 7.

2 Preliminaries

2.1 Notations

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 $\{1, 2, \cdots, n\}$ 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. And let $y \leftarrow A(x_1, x_2, \cdots, x_t)$ denote the process of running A on inputs $\{x_1, x_2, \cdots, x_t\}$ and inner randomness $R \leftarrow \mathcal{R}_A$, and outputting y. If the running time of the 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 is formalized by Fehr et al.[5] in 2010. For a public-key encryption scheme $\prod = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, 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 is passed to A_2. Sender-equivocability under adaptive chosen-ciphertext attack is defined through the following two experiments.

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

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

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

The advantage of adversary A is defined as follows.

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

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

2.3 Building Blocks of the FHKW Scheme

In [5], Fehr et al. presented a construction of PKE with NC-CCA security. We will call their scheme the FHKW scheme. The FHKW scheme was built from the following cryptographic primitives: collision-resistant hash function, subset membership problem, extended version of hash proof systems[4], and cross-authentication codes[5].

Definition 2 (Collision-resistant hash function). *A family of collision-resistant hash function* $\mathcal{H}$*, associated a domain* $\mathcal{D}$ *and a range* $\mathcal{R}$*, consists of two PPT algorithms* $(\mathsf{HGen}, \mathsf{HEval})$. $\mathsf{HGen}(1^k)$ *generates a description* des_{H} *of a uniformly random function* $\mathsf{H} : \mathcal{D} \rightarrow \mathcal{R}$. $\mathsf{HEval}(des_{\mathsf{H}}, x)$ *produces the value* $\mathsf{H}(x)$ *for all* $x \in \mathcal{D}$*. Further more, for any PPT algorithm* A*, the following function is negligible in* k*:*

$$\mathbf{Adv}^{cr}_{\mathcal{H},A}(k) := Pr\left[x \neq x' \bigwedge \mathsf{H}(x) = \mathsf{H}(x') \mid des_{\mathsf{H}} \leftarrow \mathsf{HGen}(1^k), (x, x') \leftarrow A(des_{\mathsf{H}})\right] .$$

For simplicity, we do not distinguish a function H from its description des_{H} output by HGen. So in the rest of this paper, we will write $\mathsf{H} \leftarrow \mathcal{H}$ instead of $des_{\mathsf{H}} \leftarrow \mathsf{HGen}(1^k)$.

Definition 3 (Subset membership problem). *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*

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

is negligible.

Definition 4 (Subset sparseness). *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 (Hash Proof System and Extended Hash Proof System). *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:*

- $\mathsf{HashGen}(\Lambda)$: *It is a PPT algorithm. 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 $X \in \mathcal{X}_\Lambda \backslash \mathcal{L}_\Lambda$ is still undetermined by *hpk*. This is defined by a *perfectly* 2*-universal* property.

Definition 6 (perfectly 2-universal). *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* $X_1, X_2 \in \mathcal{X}_\Lambda \backslash \mathcal{L}_\Lambda$, *and any* $K_1, K_2 \in \mathcal{K}_\Lambda$,

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

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

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

- $\mathsf{Sample}(\mathcal{D}; R)$: *On input a randomness* $R \leftarrow \mathcal{R}_{\mathsf{Sample}}$ *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 (L-Cross-Authentication Code [5]). *For any* $L \in \mathbb{N}$, *an* L-cross-authentication code XAC, *associated 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* $K \in \mathcal{XK}$, $\mathsf{XAuth}(K_1, \cdots, K_L)$ *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 of k*

$$\mathsf{fail}_{\mathsf{XAC}}^{correct}(k) := \max_{i\in[L]} \Pr[\mathsf{XVer}(K_i, i, \mathsf{XAuth}(K_1, \cdots, K_L)) \neq 1]$$

is negligible, where the max *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 $\mathbf{Adv}_{\mathsf{XAC}}^{imp}(k)$ and $\mathbf{Adv}_{\mathsf{XAC}}^{sub}(k)$, defined as follows, are both negligible.*

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

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

$$\mathbf{Adv}_{\mathsf{XAC}}^{sub}(k) := \max_{i, K_{\neq i}, \mathsf{Func}} \Pr\begin{bmatrix} K_i \leftarrow \mathsf{XGen}(1^k), T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L), T' \leftarrow \mathsf{Func}(T) : \\ T' \neq T \wedge \mathsf{XVer}(K_i, i, T') = 1 \end{bmatrix}$$

where the max *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* $\mathsf{Func} : \mathcal{XT} \rightarrow \mathcal{XT}$.

3 Review on the FHKW Scheme in [5]

With the above cryptographic primitives, we now present the FHKW scheme[5].

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 $\mathcal{H} : (\mathcal{X}_\Lambda)^L \rightarrow \mathcal{T}_\Lambda$ 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 $\mathcal{XT}$.

The 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 $M = (M_1, \cdots, M_L) \in \{0, 1\}^L$ 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 $t := \mathsf{H}(X_1, \cdots, X_L)$. 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 $T := \mathsf{XAuth}(K_1, \cdots, K_L)$. Finally, return $C = (X_1, \cdots, X_L, T)$ as the ciphertext.

$\mathsf{Dec}(sk, C)$: To decrypt a ciphertext $C = (X_1, \cdots, X_L, T) \in \mathcal{X}_\Lambda^L \times \mathcal{XT}$ 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 $M = (M_1, \cdots, M_L)$ as the plaintext.

The correctness of the FHKW scheme is proved by [5], which we omit 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 $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$, defined in Section 2, are indistinguishable.

In order to prove NC-CCA security of the FHKW scheme, Fehr et al.[5] 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 $t := \mathsf{H}(X_1, \cdots, X_L)$. For $i \in [L]$, set $K_i := \mathsf{PubEvl}(hpk, X_i, \widetilde{W_i}, t)$. Set $T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L)$. Return the ciphertext $C = (X_1, \cdots, X_L, T)$.
- $S_2(M)$: Parse $M = (M_1, \cdots, M_L)$. For $i \in [L]$, if $M_i = 1$, set $W_i := \widetilde{W_i}$, and choose $R_i^{\mathcal{X}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$, $R_i^{\mathcal{K}_\Lambda} \leftarrow \mathcal{R}_{\mathsf{Sample}}$; else, 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 simulator S, Fehr et al.[5] 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 The Problem of Security Proof in [5]

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 $C = (X_1, \cdots, X_L, T)$, the adversary comes up with a decryption query $C' = (X_1, \cdots, X_L, T')$ 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 $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$ and $\mathsf{Exp}^{\text{NC-CCA-Sim}}_{\text{FHKW},A}(k)$(see the proof of [5, Lemma 5]). This strongly relies on the security against substitution attack of cross-authentication code, which requires that "given T and $K_{\neq i}$, it is difficult to output a $T' \neq T$ such that $\mathsf{XVer}(K_i, i, T') = 1$." However, in the NC-CCA game, the adversary A KNOWs K_i for any $i \in [L]$! The reason is as follows. Upon returning a plaintext M, the 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, the 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, the XAC's security against substitution attack is not sufficient to guarantee the aforementioned property. That is why the security proof of [5] fails (more precisely, the proof of [5, Lemma 5] fails).

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

Gao et al.'s deniable scheme in [6]. In [6], Gao et al. utilized exactly the same technique as that in the FHKW scheme to construct a deniable encryption scheme and "proved" the CCA security. The similar problem we pointed out above also exists in their security proof (more specifically, the proof of [6, Claim 1]). Besides, our following 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 $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. The construction of adversary A is as follows.

In an experiment environment (either $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ or $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$), upon receiving pk, A_1 returns $M = (0, \cdots, 0)$. Then, upon receiving a ciphertext $C = (X_1, \cdots, X_L, T)$ and randomness R, A_2 returns $C' = (X_1, \cdots, X_L, T')$ as his decryption query, where $T' \leftarrow \mathsf{XAuth}(K'_1, K_2, \cdots, K_L)$, K'_1 is uniformly random chosen from $\mathcal{K}_\Lambda$ and $K_2, \cdots, K_L$ are all recovered from R. Finally, if the decryption oracle returns $M' = (0, \cdots, 0)$, 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, respectively. In $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$, 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 $M' = (0, 0, ..., 0)$ for the queried ciphertext C' and then A outputs $b = 1$, with overwhelming probability in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$. On the other hand, in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$, 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 $T' \leftarrow \mathsf{XAuth}(K'_1, K_2, \cdots, K_L)$ and $M'_i = \mathsf{XVer}(\overline{K'_i}, i, T') = 1$ for $i \in \{2, 3, \cdots, L\}$, the decryption oracle returns $M' = (0, 1, \cdots, 1)$ with overwhelming probability. As a result, A outputs $b = 1$ with negligible probability in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. The two experiments $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$ have been distinguished by A with overwhelming probability.

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

Theorem 1. *With the aforementioned simulator S, the* FHKW *scheme is* insecure *in the sense of NC-CCA for any $L > 1$.*

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

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

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

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

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

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 $T' \neq T$.

The Real Experiment. The 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 over $\mathcal{K}_\Lambda$. Hence, for $i \in \{1, 2\}$,

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

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

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

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

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

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

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

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

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

The advantage of adversary A is given by

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

Note that both $\mathbf{Adv}^{imp}_{\mathsf{XAC}}(k)$ and $\mathsf{fail}^{correct}_{\mathsf{XAC}}(k)$ 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 in [5] is incorrect. QED.

4.3 Security Analysis of the FHKW Scheme - $L = 1$

Note that our attack in the previous section does not apply to the case $L = 1$. In the previous section, 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 L-cross-authentication code against substitution attack is not sufficient for the security proof of the FHKW scheme for any value of L. But our above attack 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 [5]). That's because $R = (W_i, R_i^{\mathcal{X}_A}, R_i^{\mathcal{K}_A})_{i\in[L]}$ 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, in the sense of NC-CCA). On the other hand, if algorithm XAuth of XAC in the FHKW scheme is probabilistic, with the aforementioned simulator S, the FHKW scheme is *insecure* 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 L-cross-authentication code: "given $(K_1, K_2, \cdots, K_L)$ and $T = \mathsf{XAuth}(K_1, \cdots, K_L)$, 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 L-cross-authentication code. However, it is true for some special 1-cross-authentication code, for example, the instance of L-cross-authentication code given by Fehr et al.[5] 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 some similar property) as ingredient, the FHKW scheme is NC-CCA secure for $L = 1$.

5 A Sender Equivocable Encryption Scheme for Single-bit Plaintext

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 pointed earlier, the special property of 1-cross-authentication code requires each K determines 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. As a result, whether the paintext is 1 or 0 depending on the equality of K' and K in the decryption, while in the FHKW scheme the plaintext bit is determined by whether $\mathsf{XVer}(K, T') = 1$ or not.

Below describes our scheme $\mathcal{E} = (\mathsf{Gen}_{\mathcal{E}}, \mathsf{Enc}_{\mathcal{E}}, \mathsf{Dec}_{\mathcal{E}})$. The scheme consists of a hard subset membership problem SMP, with subset sparseness, and its related 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) are efficiently explainable. As suggested in [5], the requirement of efficient samplability and explainability on $\mathcal{K}_\Lambda$ imposes no real restriction, and it has shown in [4] that both the above ingredients can be constructed based on some standard number-theoretic assumptions, such as DDH assumption, DCR assumption and QR assumption.

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 $M = 1$; else, return $M = 0$.

Correctness: On 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 $\Pr[\overline{K} = K] = \frac{1}{|\mathcal{K}_\Lambda|}$. Hence, with probability $1 - \frac{1}{|\mathcal{K}_\Lambda|}$, $\mathsf{Dec}_\mathcal{E}(sk, C)$ returns $M = 0$.

Security: As to the security of scheme $\mathcal{E}$, we have the following Theorem 3. The proof is similar to that of the FHKW scheme in [5]. 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 doesn't involve any cross-authentication code. Details of the proof are in Appendix B.

Theorem 2. *Scheme* $\mathcal{E} = (\mathsf{Gen}_\mathcal{E}, \mathsf{Enc}_\mathcal{E}, \mathsf{Dec}_\mathcal{E})$ *is NC-CCA secure.*

6 Fixing the Security Proof of the FHKW Scheme

In this section, we will present a strong version of cross-authentication code, and fix the security proof of the FHKW scheme with it.

6.1 Strong L-Cross-Authentication Codes

The notion of strong L-cross-authentication code is as follows.

Definition 9 (Strong L-Cross-Authentication Code). *An L-cross-authentication code* $\mathsf{X\text{-}AC}$ *is* strong, *if there exists another PPT algorithm* ReSamp *satisfying the following property: Given* $K_1, \cdots, K_L \leftarrow \mathsf{XGen}(1^k)$ *and* $T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L)$ *such that* $\mathsf{XVer}(K_l, l, T) = 1$, $l \in [L]$, *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 with* K_i, *i.e.,*

$$Dist(k) := \frac{1}{2} \cdot \sum_{K \in \mathcal{XK}} |\Pr[K'_i = K | (K_{\neq i}, T)] - \Pr[K_i = K | (K_{\neq i}, T)]|$$

is negligible, where $K'_i \leftarrow \mathsf{ReSamp}(i, K_{\neq i}, T)$ *and the probabilities are taken over all possible* $K_i \leftarrow \mathsf{XGen}(1^k)$ *and the randomness of* ReSamp.

Example of a strong L-cross-authentication code. In [5], Fehr et al. proposed an efficient construction of L-cross-authentication code, $\mathsf{XAC}_{\mathrm{FHKW}}$=($\mathsf{XGen}$,$\mathsf{XAuth}$,$\mathsf{XVer}$), as follows.

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 $\mathcal{XT} = \mathbb{F}_q^L \cup \{\perp\}$. $\mathsf{XGen}(1^k)$ generates a uniformly random key $K \in \mathcal{XK}$. For $K_1 = (a_1, b_1), \cdots, K_L = (a_L, b_L) \in \mathcal{XK}$, $\mathsf{XAuth}(K_1, \cdots, K_L)$ computes a tag $T = (T_0, \cdots, T_{L-1})$ satisfying that for $i \in [L]$, $poly_T(a_i) = b_i$, where $poly_T(x) = T_0 + T_1 x + \cdots + T_{L-1} x^{L-1} \in \mathbb{F}_q[x]$. 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 $T = \perp$. For any $K = (a, b) \in \mathcal{XK}$, $i \in [L]$ and $T \in \mathcal{XT}$, $\mathsf{XVer}(K, i, T)$ outputs 1 if and only if $T \neq \perp$ and $poly_T(a) = b$.

We will show that $\mathsf{XAC}_{\mathrm{FHKW}}$ is strong as well.

Lemma 1. *For any* $L \in \mathbb{N}$, *L-cross-authentication code* $\mathsf{XAC}_{\mathrm{FHKW}}$ *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 satisfying that $\mathsf{XVer}(K_l, l, T) = 1$ 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 $b_i' = poly_T(a_i')$ and returns $K_i' = (a_i', b_i')$ as its output. As a result, $\Pr\left[K_i' = (a_i', b_i')\right] = \frac{1}{q-L+1}$.

On the other hand, $K_i = (a_i, b_i) \leftarrow \mathsf{XGen}(1^k)$ and $\mathbf{A}$ is non-singular, so a_i is chosen uniformly random from $\mathbb{F}_q$ under the constraint that $a_i \neq a_j$ for $j \in [L]\backslash\{i\}$. We know that $b_i = poly_T(a_i)$. Hence $\Pr\left[K_i = (a_i, b_i)\right] = \frac{1}{q-L+1}$, which has identical probability distribution with K_i'.

6.2 Fixing the Security Proof of the FHKW Scheme with Strong XAC

Replacing XAC with a strong one, we get a new version of the FHKW scheme. 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 $K_1, K_2, \cdots, K_L$ 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.

Theorem 3. *For any $L > 1$, assuming that* XAC *is a strong L-cross-authentication code, the* FHKW *scheme is NC-CCA secure.*

Proof. The main idea of this proof is similar to that of the proof of [5, Theorem 3]. First, we construct a simulator $S' = (S_1', S_2')$ for the FHKW scheme.

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 $t := \mathsf{H}(X_1, \cdots, X_L)$. For $i \in [L]$, set $K_i := \mathsf{PubEvl}(hpk, X_i, \widetilde{W_i}, t)$. Set $T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L)$. Return the ciphertext $C = (X_1, \cdots, X_L, T)$.
- $S_2'(M)$: Parse $M = (M_1, \cdots, M_L)$. 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 $K_i' \leftarrow \mathsf{ReSamp}(i, K_{\neq i}, T)$, where ReSamp is from the strong L-cross-authentication code XAC, 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 simulator S', we will show that for any PPT adversary A, the two experiments $\mathsf{Exp}_{\mathrm{FHKW},A}^{\mathrm{NC\text{-}CCA\text{-}Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\mathrm{NC\text{-}CCA\text{-}Sim}}(k)$ are computationally indistinguishable through a series of indistinguishable games. Technically, we denote the challenge ciphertext and its related plaintext by C^* and M^*, and write $C^* := (X_1^*, \cdots, X_L^*, T^*)$ and $M^* := (M_1^*, \cdots, M_L^*)$. Denote A's j-th decryption query by $C^j := (X_1^j, \cdots, X_L^j, T^j)$, the corresponding plaintext by $M^j = (M_1^j, \cdots, M_L^j)$, and define t^*, t^j, K_i^* and K_i^j similarly. Define $\overline{K_i^*} := \mathsf{SecEvl}(hsk, X_i^*, t^*)$, $\overline{K_i}^j := \mathsf{SecEvl}(hsk, X_i^j, t^j)$ 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 $q = poly(k)$.

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

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

Game -1: Game -1 is the same as Game -2, except that in the challenge ciphertext generation, we abort the experiment (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

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

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 $(X_1^j, \cdots, X_L^j) \neq (X_1^*, \cdots, X_L^*)$ and $t^j = \mathsf{H}(X_1^j, \cdots, X_L^j) = \mathsf{H}(X_1^*, \cdots, X_L^*) = t^*$, we abort the experiment (without loss of generality, with A outputting 1). Since H is a collision-resistant hash function, we have that

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

for a suitable PPT algorithm A'.

In the rest, 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 and Game L are indistinguishable. Note that Game L is identical to $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$. Hence, we can finish 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 $C^j = (X_1^j, \cdots, X_L^j, T^j)$ and for any $i \in [L]$, the challenger will return $M_i^j = 0$ directly if $X_i^j \notin \mathcal{L}_\Lambda$, and behave just as in Game $m.1$ otherwise: compute $\overline{K_i}^j = \mathsf{SecEvl}(hsk, X_i^j, t^j)$, and return $M_i^j = \mathsf{XVer}(\overline{K_i}^j, i, T^j)$. Note that the decryption oracle in Game $m.2$ is inefficient and it doesn't 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 $X_i^j \notin \mathcal{L}_\Lambda$ 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. $\Pr[\mathsf{bad}_{m.2}] \leq qL \cdot \mathbf{Adv}_{\mathsf{XAC}}^{imp}(k)$.

With the lemma, we have that

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

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 HPS 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 events $\mathsf{sub}_{m.3}$ or $\mathsf{sub}_{m.2}$ occurs. So we have that

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

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,

$$|\Pr\left[output_{A,m.4} = 1\right] - \Pr\left[output_{A,m.3} = 1\right]| \le Dist(k),$$

where $Dist(k)$ 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 $C^j = (X^j_1, \cdots, X^j_L, T^j)$, 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. $\Pr[\mathsf{bad}_{m.4}] \le qL \cdot \max\{\mathbf{Adv}^{imp}_{\mathsf{XAC}}(k), \mathbf{Adv}^{sub}_{\mathsf{XAC}}(k)\}$.

With the lemma, we have that

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

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

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

for a suitable PPT algorithm A''. (If $m+1$ is not known to A'', A'' can guess it with probability $\frac{1}{L}$.)

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 Lemma 2 and Lemma 3.

Proof (of Lemma 2). Let $\mathsf{bad}^j_{m.2.i}$ denote the event that A's j-th decryption query $C^j = (X^j_1, \cdots, X^j_L, T^j)$ 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, $\Pr\left[\mathsf{bad}^j_{m.2.i}\right] \leq \mathbf{Adv}^{imp}_{\mathsf{XAC}}(k)$. 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

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

Proof (of Lemma 3). Let $\mathsf{bad}^j_{m.4.i}$ denote the event that A's j-th decryption query $C^j = (X^j_1, \cdots, X^j_L, T^j)$ 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, since the only possible information A has on hsk beyond hpk is K^*_{m+1}, and K^*_{m+1} is not equal but related to $K^{hsk}_{m+1} = \mathsf{SecEvl}(hsk, X^*_{m+1}, t^*)$ in Game $m.4$. Hence, $\Pr\left[\mathsf{bad}^j_{m.4.i} \mid (X^j_i, t^j) \neq (X^*_{m+1}, t^*)\right] \leq \mathbf{Adv}^{imp}_{\mathsf{XAC}}(k)$.

If $(X^j_i, t^j) = (X^*_{m+1}, t^*)$ then $(X^j_1, \cdots, X^j_L) = (X^*_1, \cdots, X^*_L)$, since Game 0 excludes hash collisions. The decryption query C^j has to be valid, 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 $(K^*_1, \cdots, K^*_m, K^*_{m+1}, K^*_{m+2}, \cdots, K^*_L)$ 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 $\mathbf{Adv}^{sub}_{\mathsf{XAC}}(k)$, i.e., $\Pr\left[\mathsf{bad}^j_{m.4.i} \mid (X^j_i, t^j) = (X^*_{m+1}, t^*)\right] \leq \mathbf{Adv}^{sub}_{\mathsf{XAC}}(k)$.

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

Lemma 3 follows from a union bound.

So the whole proof of Theorem 3 is finished. QED.

7 Conclusion

We provided a security analysis of the FHKW scheme of [5] and showed that the original simulator of [5] is not sufficient to prove the NC-CCA security. We provided a refined version of the FHKW scheme for 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.

Open questions. There are two questions to be solved: 1. Whether every cross-authentication code is also a strong one; 2. How to construct an NC-CCA secure PKE encrypting multi bits from an NC-CCA secure PKE encrypting single bit.

References

1. F. Böhl, D. Hofheinz and D. Kraschewski. On definitions of selective opening security. In: Cryptology ePrint Archive, Report 2011/678 (2011)
2. M. Bellare, D. Hofheinz and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In: Eurocrypt 2009. LNCS, vol. 5479, pp. 1-35. Springer, Heidelberg (2009)
3. R. Canetti, U. Friege, O. Goldreich and M. Naor. Adaptively secure multi-party computation. In: 28th ACM STOC, pp. 639-648. ACM Press, New York (1996)
4. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Eurocrypt 2002. LNCS, vol. 2332, pp. 45-64. Springer, Heidelberg (2002)
5. S. Fehr, D. Hofheinz, E. Kiltz and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Eurocrypt 2010. LNCS, vol. 6110, pp. 381-402. Springer, Heidelberg (2010)
6. C. Gao, D. Xie and B. Wei. Deniable encryptions secure against adaptive chosen ciphertext attack. In: ISPEC 2012. LNCS, vol. 7232, pp. 46-62. Springer, Heidelberg (2012)
7. D. Hofheinz. All-but-many lossy trapdoor functions. In: Eurocrypt 2012. LNCS, vol. 7237, pp. 209-227. Springer, Heidelberg (2012)
8. B. Hemenway, B. Libert, R. Ostrovsky and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Asiacrypt 2011. LNCS. Springer (2011)
9. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In: STOC 2008. pp. 187-196. ACM, New York (2008)

A In case algorithm XAuth is probabilistic.

In Section 4.3, we have claimed that if algorithm XAuth of XAC in the FHKW scheme is probabilistic, with the aforementioned simulator S in Section 4, the FHKW scheme will be *insecure* in the sense of NC-CCA for any positive integer L.

Now we show how to reach this conclusion.

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., $T \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L; R^{\mathsf{XAuth}})$). 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, and that $(W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}$ have been able to be recovered by the original S, i.e., S should generate R^{XAuth} according to T and $(K_1, \cdots, K_L)$, which can be recovered from $R = (W_i, R_i^{\mathcal{X}_\Lambda}, R_i^{\mathcal{K}_\Lambda})_{i\in[L]}$. Therefore, we make a modification to XAuth: we require that XAuth is efficiently "explainable", which means that there is an efficient algorithm $\mathsf{Explain}_{\mathsf{XAuth}}$ such that $R^{\mathsf{XAuth}} \leftarrow \mathsf{Explain}_{\mathsf{XAuth}}((K_1, \cdots, K_L), T)$. For simplicity, we still use the original notations S and XAuth after this modification.

Secondly, with the above modification, consider our main conclusion of this Appendix. As the proof of Theorem 2, 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 adversary A is the same as the one in the proof of Theorem 2, 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 $C = (X_1, \cdots, X_L, T)$ and randomness R, A_2 recovers $(K_1, \cdots, K_L)$ from R, computes $T' \leftarrow \mathsf{XAuth}(K_1, \cdots, K_L; \widetilde{R}^{\mathsf{XAuth}})$, where $\widetilde{R}^{\mathsf{XAuth}}$ is uniformly random chosen from $\mathcal{R}_{\mathsf{XAuth}}$, and returns $C' = (X_1, \cdots, X_L, T')$ 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 overwhelming probability, if in $\mathsf{Exp}^{\text{NC-CCA-Real}}_{\text{FHKW},A}(k)$, A_2 will receive $M' = (0, \cdots, 0)$ as the decryption result of C', and if in $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$, A will receive $M' = (1, \cdots, 1)$. Hence, A can distinguish $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Real}}(k)$ and $\mathsf{Exp}_{\mathrm{FHKW},A}^{\text{NC-CCA-Sim}}(k)$.

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 related plaintext by C^* and M^*, and write $C^* := (X^*, K^*)$. Denote A's decryption query by $C' := (X', K')$ and let its corresponding plaintext be M'. At the same time, we define $\overline{K^*} := \mathsf{SecEvl}(hsk, X^*)$, $\overline{K'} := \mathsf{SecEvl}(hsk, X')$ and 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 above notations,

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

Game 1: Game 1 is the same as Game 0, except for the decryption oracle. In Game 1, if A makes a decryption query $C' = (X', K')$ such that $X' \notin \mathcal{L}_\Lambda$, the challenger will return $M' = 0$ directly, and if $X' \in \mathcal{L}_\Lambda$, the challenger will answer the query as in Game 0: compute $\overline{K'} = \mathsf{SecEvl}(hsk, X')$, and if $\overline{K'} = K'$, return $M' = 1$, else return $M' = 0$. Note that the decryption oracle in Game 1 is inefficient and it doesn't leak any information of hsk beyond hpk. Let bad_i denote the event that in Game i, A makes a decryption query $C' = (X', K')$ such that $X' \notin \mathcal{L}_\Lambda$ and $K' = \overline{K'}$. Note that $\Pr[\mathsf{bad}_1] = \Pr[\mathsf{bad}_0]$ and that Game 1 and Game 0 are identical unless the respective bad_1 and bad_0 occur. The perfect 2-universality of HPS implies $\Pr[\mathsf{bad}_1] = \Pr[\mathsf{bad}_0] = \frac{1}{|\mathcal{K}_\Lambda|}$. So we have

$$|\Pr\left[output_{A,1} = 1\right] - \Pr\left[output_{A,0} = 1\right]| \leq \Pr\left[\mathsf{bad}_1\right] = \frac{1}{|\mathcal{K}_\Lambda|}.$$

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 the way $\mathsf{Explain}(\mathcal{K}_\Lambda, K^*)$. In Game 2, since the only information of 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 over $\mathcal{K}_\Lambda$. 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 the respective events sub_2 and sub_1 occur. So we have

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

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

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

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 $M^* = 0$. (I.e., in Game 4, choose $X^* \leftarrow \mathcal{L}_\Lambda$ whatever M^* is.) Then X^* is opened as $\mathsf{Explain}(\mathcal{X}_\Lambda, X^*)$ in this case. Note that in Game 3, if $M^* = 0$, X^* also can be seen as being opened by the way $\mathsf{Explain}(\mathcal{X}_\Lambda, X^*)$. Since SMP is hard,

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

Combining all the above results, we have

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

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

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

QED.