Chosen-Ciphertext Secure Efficiently Searchable Encryption in the Standard Model

Yang Cui[1] and Kirill Morozov[2]

[1] Beijing Research Institute, Huawei Technologies Co Ltd, China
[2] Institute of Mathematics for Industry, Kyushu University, Japan
E-mail: morozov@imi.kyushu-u.ac.jp

Abstract. In the standard model, deterministic public-key encryption (PKE) secure against chosen-ciphertext attacks by privacy adversary (PRIV-CCA) is known to be built only from lossy trapdoor functions as demonstrated by Boldyreva et al at Crypto 2008. We show that the method of achieving IND-CCA security via correlated products, recently introduced by Rosen and Segev at TCC 2009, can be used to achieve PRIV-CCA secure PKE of uniform messages from any trapdoor permutation (TDP) in the standard model. Our schemes are *not* deterministic as a whole, however randomness is only applied to a particular part of the ciphertext - an one-time signature used for validity check. This allows efficient (logarithmic in the database size) search on encrypted data. In a nutshell, our first construction (which is generic) departs from any IND-CPA secure PKE (implied by TDP), builds its k-correlated version, transforms it into the k-correlated PRIV-CPA encryption, and finally lifts it up to PRIV-CCA security. In contrast to Rosen and Segev's correlated products method, we do not assume one-wayness under correlated inputs, thus any IND-CPA secure PKE can be used in our construction. In addition, we present the second construction – which is more efficient, than the first one – based on assumptions from coding theory and any TDP. Note that for the price of allowing some limited use of randomness, we achieve PRIV security for multiple messages, which is strictly stronger than the single-message notion PRIV1 achieved by the scheme of Boldyreva et al at Crypto 2008.

1 Introduction

Background. Bellare et al [4] introduced the notion of security against *privacy adversary* (denoted as PRIV). This notion requires that the ciphertext does not leak any predicate on the corresponding plaintext. In order to satisfy this notion, the plaintext must: 1) Come from a large domain and have a smooth (i.e. high min-entropy) distribution; 2) Be – along with the target predicate – independent of the public key. PRIV-security was defined for deterministic encryption (DE) and (not necessarily deterministic) efficiently searchable encryption, featuring an upgrade from the basic notion of one-wayness.

Two flavors of PRIV-security were presented in [4], similarly to the standard indistinguishability notions: against chosen-plaintext attacks (CPA) and against chosen-ciphertext attacks (CCA). Constructions in the random oracle (RO) model were presented in [4, 12], while schemes in the standard model as well as new security notions were treated in [6, 10].

CCA security guarantees protection even against an adversary who has access to a decryption oracle. In other words, the adversary is allowed to decrypt a polynomial number of (arbitrary) ciphertexts, except for the target one. CCA security is a *de facto* standard for modern public-key encryption. It guarantees protection against a lot of attacks, which may not be covered by weaker security notions – perhaps, the most famous example of the latter is Bleichenbacher's attack [2].

Application. Our schemes allow for efficient search on encrypted data in database applications, since a (known) part of the ciphertext is deterministic. By "efficient" we refer to the search which is logarithmic in the database size. This is in sharp contrast with the (randomized) public-key encryption with keyword search (PEKS) – the line of works initiated in [8] – where the search time is linear.

We employ an approach similar to *correlated products*, recently introduced by Rosen and Segev [24] in TCC 2009, in order to construct PRIV-CCA secure PKE in the standard model. Security under correlated products means that a product of one-way functions remains one-way, even though their inputs might be correlated. If an injective trapdoor function is used, it was shown in [24] that the correlated products construction yields IND-CCA secure PKE in a natural manner. To the best of our knowledge, trapdoor functions secure under correlated products can only be built from lossy trapdoor functions [23].

Motivation. Our main motivation is to build PRIV-CCA secure efficiently searchable PKE from generic assumptions *in the standard model.* Constructions in the standard model may be preferable, since there exist cryptographic schemes which can be proven secure in the RO model, but become insecure when idealized RO's are substituted by any implementation. This was first shown by Canetti et al [11] as a proof-of-concept, and later extended to practical scenarios such as hybrid encryption by Bellare et al [5] (for other examples, see references therein).

From now on, we focus on the constructions in the standard model, unless stated otherwise.

Our Contribution. We present two efficiently searchable PKE [4] schemes achieving PRIV-CCA security. The efficiently searchable property comes from the fact, that the ciphertext, except for an attached one-time signature, used for validity check, is deterministic. The first one enjoys generality as it can be based on any trapdoor permutation. The second one is more computationally efficient as compared to the first scheme, but it uses additional assumptions from coding theory.

Both of our constructions require the messages to be distributed uniformly and independently. It is indeed a strict requirement, however these schemes can be viewed as the first step towards achieving efficiently searchable PRIV-CCA secure PKE from generic assumptions.

It is worth noting that the above requirement is inherited by our construction from PRIV-CPA secure TDP-based deterministic encryption, which – being a part of schemes – needs uniform plaintexts [6] to be secure. In fact, this requirement can be dropped [7, Sec. 6] to the (standard) high entropy requirement as long as the assumption is strengthened to TDP being one-way for high-min entropy distributions. We believe, but do not prove formally here, that our constructions generalize in a similar manner.

Our generic scheme consists of the following three main steps:

1. Transform the IND-CPA secure PKE $\overline{\mathit{\Pi}}$ into k-correlated IND-CPA secure PKE $\overline{\mathit{\Pi}}_k$ as in the work of multiple user setting by Bellare et al [3].
2. Convert the latter into k-correlated PRIV-CPA secure deterministic encryption $\mathit{\Pi}_k$, with the help of trapdoor permutations as in PRIV-CPA secure scheme by Bellare et al [6].
3. Lift $\mathit{\Pi}_k$ up to PRIV-CCA secure $\mathit{\Pi}_{cca}$ using the method of correlated products [24].

The merits of our method are two-fold: 1) k-correlated one-wayness is implicitly achieved by IND-CPA security, thus do not need the assumption on one-wayness under correlated inputs as it appears in Rosen-Segev's construction; 2) PRIV security under multiple user (public-key) setting is obtained. Note that there is in general no guarantee that PRIV security in the single user setting also holds in the multiple user setting, as observed in [4].

In order to obtain pseudorandomness with help of trapdoor permutation, we make use of Goldreich-Levin [17], Blum-Micali-Yao [9, 26] pseudorandom generator (PRG). As the correlated products construction requires k instances of encryption, it will be costly to adopt the

PRG in each instance. Our second scheme based on coding-theoretic assumptions (code-based) is designed to avoids the above mentioned overhead. It is more efficient compared to the generic one, since we use the (relatively heavy) scheme with pseudorandom coins from [6] only for the first of k-Correlated Products. The security of this scheme is based on syndrome decoding, indistinguishability of the Randomized McEliece encryption [21], and any TDP.

Related Work. To the best of our knowledge, only the work by Boldyreva et al [10] achieves PRIV-CCA secure PKE in the standard model. Their generic scheme is based on lossy trapdoor functions, while our generic scheme only uses trapdoor permutations. Efficiency of these schemes is comparable. At the same time, any of instantiations in [10] is more efficient than our code-based scheme. We achieve the (standard) PRIV-security [4] for multiple messages, while the schemes of [10] are PRIV-secure for single message (PRIV1), or – equivalently [10, Sec. 4] – for block-sources i.e. each message must have high min-entropy conditioned on the values of the other messages. In this respect, these schemes are not easily comparable, as we require the messages to be uniform. At the same time, our schemes use one-time signatures for validity checks, which is similarly as the "encrypt-then-sign fashion" to obtain PRIV-CCA security in the random oracle model [4, Sec.7]. While on a different approach, it is not the case in [10] – target collision resistant hash functions is proven to work in their case.

It is worth noting that our second scheme is inspired by [15] in that we use k-Correlated Products of [24], substituting the injective one-way functions with another primitive of our convenience (i.e. IND-CPA secure encryption).

Organization. The paper will be organized in the following way: Sec. 2 presents basic notation and definitions. Sec. 3 is devoted to the generic construction of PRIV-CCA PKE from trapdoor permutations. The code-based construction is presented in Sec. 4.

2 Preliminaries

Denote by $\boldsymbol{x}$ the vector, and by "$|\boldsymbol{x}|$" the cardinality of $\boldsymbol{x}$. Denote by $\boldsymbol{x}[i]$ the i-th component of $\boldsymbol{x}$ ($1 \leq i \leq |\boldsymbol{x}|$), and by "$|x|$" the bit length of x. Let $x \overset{\$}{\leftarrow} X$ denote the operation of selecting x from the set X uniformly at random. Denote by $z \overset{\$}{\leftarrow} A(x, y, ...)$ the operation of running algorithm A with input $(x, y, ...)$ and fresh random coins, to output a result z. We also write $\Pr[A(x) = y : x \overset{\$}{\leftarrow} X]$ the probability that A outputs y corresponding to input x, which is sampled from X. We say a function $\epsilon(k)$ is negligible, if for any constant c, there exists $k_0 \in \mathbb{N}$, such that $\epsilon < (1/k)^c$ for any $k > k_0$.

A public key encryption scheme $\mathit{\Pi}$ consists of a triple of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key generation algorithm $\mathcal{K}$ outputs a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$ taking on input 1^λ, a security parameter λ in unitary notation. The encryption algorithm $\mathcal{E}$ on input pk and a plaintext x outputs a ciphertext c. The decryption algorithm $\mathcal{D}$ takes sk and c as input and outputs the plaintext message x. We require that for any key pair $(\mathsf{pk}, \mathsf{sk})$ obtained from $\mathcal{K}$, and any plaintext x from the plaintext space of $\mathit{\Pi}$, $x \leftarrow \mathcal{D}(\mathsf{sk}, \mathcal{E}(\mathsf{pk}, x))$.

2.1 Security Definitions

Definition 1 (PRIV-CPA, PRIV-CCA [4]). *Let a probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{DE}$ against the privacy of the public-key encryption $\mathit{\Pi} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, be a pair of algorithms $\mathcal{A}_{DE} = (\mathcal{A}_m, \mathcal{A}_g)$, where $\mathcal{A}_m, \mathcal{A}_g$ do not share any random coins or state. The advantage of adversary is defined as follows,*

$$\mathbf{Adv}^{\textit{priv-atk}}_{\mathit{\Pi},\mathcal{A}_{DE}}(\lambda) = \Pr[\mathbf{Exp}^{\textit{priv-1}}_{\mathit{\Pi},\mathcal{A}_{DE}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\textit{priv-0}}_{\mathit{\Pi},\mathcal{A}_{DE}}(\lambda) = 1]$$

where experiments are described as:

$$\begin{array}{l|l}
\textbf{Experiment } \mathbf{Exp}^{\textit{priv-1}}_{\Pi,\mathcal{A}_{DE}}(\lambda): & \textbf{Experiment } \mathbf{Exp}^{\textit{priv-0}}_{\Pi,\mathcal{A}_{DE}}(\lambda): \\
 & \\
(\mathsf{pk},\mathsf{sk}) \overset{\$}{\leftarrow} \mathcal{K}(1^\lambda), & (\mathsf{pk},\mathsf{sk}) \overset{\$}{\leftarrow} \mathcal{K}(1^\lambda), \\
(\boldsymbol{x}_1, t_1) \overset{\$}{\leftarrow} \mathcal{A}_m(1^\lambda), & (\boldsymbol{x}_0, t_0) \overset{\$}{\leftarrow} \mathcal{A}_m(1^\lambda), (\boldsymbol{x}_1, t_1) \overset{\$}{\leftarrow} \mathcal{A}_m(1^\lambda), \\
\boldsymbol{c} \overset{\$}{\leftarrow} \mathcal{E}(1^\lambda, \mathsf{pk}, \boldsymbol{x}_1), & \boldsymbol{c} \overset{\$}{\leftarrow} \mathcal{E}(1^\lambda, \mathsf{pk}, \boldsymbol{x}_0), \\
g \overset{\$}{\leftarrow} \mathcal{A}_g^{\mathcal{DO}(\cdot)}(1^\lambda, \mathsf{pk}, \boldsymbol{c}); & g \overset{\$}{\leftarrow} \mathcal{A}_g^{\mathcal{DO}(\cdot)}(1^\lambda, \mathsf{pk}, \boldsymbol{c}); \\
\textit{return 1 if } g = t_1, \textit{ else return 0} & \textit{return 1 if } g = t_1, \textit{ else return 0}
\end{array}$$

We say that Π is PRIV secure*, if $\mathbf{Adv}^{\textit{priv-atk}}_{\Pi,\mathcal{A}_{DE}}(\lambda)$ is negligible, for any PPT $\mathcal{A}_{DE}$ with high min-entropy, where $\mathcal{A}_{DE}$ has a high min-entropy $\mu(\lambda)$ means that $\mu(\lambda) \in \omega(log(\lambda))$, and* $\Pr[\boldsymbol{x}[i] = x : (\boldsymbol{x}, t) \overset{\$}{\leftarrow} \mathcal{A}_m(1^\lambda)] \leq 2^{-\mu(\lambda)}$ *for all λ, all $1 \leq i \leq |\boldsymbol{x}|$, and any $x \in \{0,1\}^*$. The encryption is done in a component-wise way, i.e. $\boldsymbol{c}[i] \overset{\$}{\leftarrow} \mathcal{E}(1^\lambda, \mathsf{pk}, \boldsymbol{x}[i])$, $1 \leq i \leq |\boldsymbol{x}|$.*

If $atk = cpa$, then $\mathcal{DO}(\cdot) = \epsilon$,
If $atk = cca$, then $\mathcal{DO}(\cdot) = \mathcal{D}(1^\lambda, \mathsf{pk}, \mathsf{sk}, \cdot)$,
where $\mathcal{DO}(\cdot) = \epsilon$ means that $\mathcal{DO}$ is a function on any input returning an empty string ϵ; $\mathcal{DO}(\cdot) = \mathcal{D}(1^\lambda, \mathsf{pk}, \mathsf{sk}, \cdot)$ means $\mathcal{DO}$ is a function on input of a ciphertext returning a message or $\perp$. Any string can be queried to decryption oracle, except what have appeared as a component of $\boldsymbol{c}$.

Another way to define the advantage of privacy adversary, in the underlying definition, is written as follows:

$$\mathbf{Adv}^{\text{priv-atk}}_{\Pi,\mathcal{A}_{DE}}(\lambda) = 2\Pr[\mathbf{Exp}^{\text{priv-b}}_{\Pi,\mathcal{A}_{DE}}(\lambda) = b] - 1$$

where $b \in \{0,1\}$ and probability is taken over the choice of all of the random coins in the experiments.

It has been proven that PRIV security can be built from non-deterministic encryption algorithms such as, indistinguishability (IND) secure PKE, as observed in [4]. In the following, we recall the notion of IND security.

Definition 2 (IND-CPA, IND-CCA). *We say a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-secure, if the advantage $\mathbf{Adv}^{\textit{ind-atk}}_{\Pi,\mathcal{A}}$ of any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is negligible, (let st be the state information of $\mathcal{A}_1$, and $\hat{b} \in \{0,1\}$):*

$$\mathbf{Adv}^{\textit{ind-atk}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr\left[\begin{array}{l} \hat{b} = b : (\mathsf{pk},\mathsf{sk}) \overset{\$}{\leftarrow} \mathcal{K}(1^\lambda), \\ (x_0, x_1, st) \overset{\$}{\leftarrow} \mathcal{A}_1^{\mathcal{DO}(\cdot)}(1^\lambda, \mathsf{pk}), \\ b \overset{\$}{\leftarrow} \{0,1\}, c \overset{\$}{\leftarrow} \mathcal{E}(1^\lambda, \mathsf{pk}, x_b), \\ \hat{b} \overset{\$}{\leftarrow} \mathcal{A}_2^{\mathcal{DO}(\cdot)}(1^\lambda, c, st) \end{array}\right] - 1$$

and if $atk = cpa$, then $\mathcal{DO}(\cdot) = \epsilon$; if $atk = cca$, then $\mathcal{DO}(\cdot) = \mathcal{D}(1^\lambda, \mathsf{pk}, \mathsf{sk}, \cdot)$, and $\mathcal{DO}(\cdot)$ is the same as in Def. 1. Any ciphertext except the target ciphertext c output by encryption oracle, can be queried to decryption oracle.

Definition 3 (Strongly unforgeable one-time signature). *$\mathcal{SIG} = (\mathcal{SG}, \mathcal{SS}, \mathcal{SV})$: Key generation algorithm $\mathcal{SG}(1^\lambda)$ outputs a verification key* vk *and signing key* sigk *randomly. Signing algorithm $\mathcal{SS}(1^\lambda, \mathsf{sigk}, m)$ takes as input a signing key* sigk *and a message m from message* *space, and outputs a signature σ. Verification algorithm $\mathcal{SV}(1^\lambda, \mathsf{vk}, m, \sigma)$ takes as input a verification key* vk*, a message m and a signature σ, and output 0 for invalid or 1 for valid.*

$$\mathbf{Adv}^{\textit{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda) = \Pr\begin{bmatrix}\mathcal{SV}(1^\lambda, \mathsf{vk}, m^*, \sigma^*) = 1 \wedge (m^*, \sigma^*) \neq (m, \sigma) : \\ (\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathcal{SG}(1^\lambda), (m, \sigma) \leftarrow \mathcal{F}^{\mathcal{SIG}(\mathsf{sigk}, \cdot)}(1^\lambda, \mathsf{vk}), \\ st \leftarrow (m, \sigma), (m^*, \sigma^*) \leftarrow \mathcal{F}(1^\lambda, \mathsf{vk}, st)\end{bmatrix}$$

Forger $\mathcal{F}$ outputs (m^*, σ^*) after invoking signing oracle $\mathcal{SIG}(\mathsf{sigk}, \cdot)$ only once, with the restriction that $(m^*, \sigma^*) \neq (m, \sigma)$. We say $\mathcal{SIG}$ is strongly unforgeable one-time signature, if the $\mathbf{Adv}^{\textit{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda)$ *is negligible.*

For generic construction of CCA security, a strongly unforgeable one-time signature [1] $\mathcal{SIG} = (\mathcal{SG}, \mathcal{SS}, \mathcal{SV})$ is often employed, which could in principle be built from the existence of one-way function. Since we already make use of secure PKE scheme which implies the existence of one-way function, this will not introduce additional assumptions.

2.2 Correlated Products

Rosen and Segev [24] proposed a simple and general construction in order to obtain CCA-secure PKE, namely Correlated Products (CP), from injective trapdoor functions. Their scheme is built in a black-box way and achieves a direct proof of security, widely used by recent proposals [22, 15, 16].

Definition 4 (k-wise product). *For integer k and a collection of one-way functions $\mathcal{F}_n = (\mathsf{G}, \mathsf{F})$ which is efficiently computable, define the k-wise product $\mathcal{F}_k = (\mathsf{G}_k, \mathsf{F}_k)$ as follows.*

- *On input 1^λ, the key generation algorithm $\mathsf{G}_k(1^\lambda)$ invokes k independent instances of $\mathsf{G}(1^\lambda)$, and outputs: $(f_1, \ldots, f_k)$.*
- *On input $(f_1, \ldots, f_k, x_1, \ldots, x_k)$, the evaluation function F_k invokes function F to evaluate each function f_i to x_i $(1 \leq i \leq k)$, i.e. $\mathsf{F}_k(f_1, \ldots, f_k, x_1, \ldots, x_k) = (\mathsf{F}(f_1, x_1), \ldots, \mathsf{F}(f_k, x_k))$.*

If the inputs $(x_1, \ldots, x_k)$ are randomly and independently chosen from a sufficiently large domain, then it is easy to see that one-wayness of k-wise product holds if $\mathcal{F}_n$ is a family of one-way functions. However, when the inputs are correlated, it is known that the above result is in general unavailable, as many examples have shown including Håstad's attack [19] on RSA broadcast encryption. As a consequence, to secure the one-wayness of k-wise product, the following definition is required especially when the inputs are correlated.

Definition 5 (One-wayness under correlated inputs [24]). *Let $\mathcal{F}_n = (\mathsf{G}, \mathsf{F})$ be a collection of efficiently computable one-way functions with domain of $\{D_\lambda\}$. Distribution $\mathcal{C}(1^\lambda)$ is over $D^k_\lambda = D_\lambda \times \cdots \times D_\lambda$ for some integer $k = poly(\lambda)$. We say that $\mathcal{F}_n$ is one-way under $\mathcal{C}$-correlated inputs, if $\mathcal{F}_k$ is one-way on input distribution $\mathcal{C}$.*

Similar as Rosen-Segev's paper [24] and others [22, 15, 16] did, we also focus on the distribution represented by k copies of x, where x is chosen uniformly from the domain, $x \overset{\$}{\leftarrow} D_\lambda$. We define it as *$k$-correlated inputs.*

Remark. We refer to Appendix A for the concrete construction of Correlated Products. That construction requires the assumption that one-wayness under k-correlated inputs is difficult to invert for any PPT adversary $\mathcal{A}$, i.e.

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{F}(f_1, x), \ldots, \mathsf{F}(f_k, x); f_1, \ldots, f_k) = x] < \epsilon(\lambda)$$

Note that following [24], we describe a simplified version of the Correlated Products primitive, omitting application of the universal one-way hash function to the verification keys (as it is done in [14]). We note that if this function is applied, then we only need $k(\lambda) = \lambda^{\epsilon}$ for a constant $0 < \epsilon < 1$.

It worth nothing that the Correlated Products primitive has so far been instantiated only from lossy trapdoor functions and specific algebraic assumptions, and it is not known to be obtained from generic assumptions. In the following section, we propose our scheme which uses the basic idea of the Correlated Products construction, and yet it is based on trapdoor permutations.

3 Generic Construction of PRIV-CCA via Correlated Products

In this section, we provide our generic construction of PRIV-CCA secure encryption from trapdoor permutations. In fact, we depart from IND-CPA secure encryption, but it is implied by the latter.

Our scheme is inspired by Rosen-Segev's Correlated Products [24], but it takes a shot at PRIV-CCA security rather than IND-CCA security, so that we are able to rely on more general assumptions and adapt to various concrete schemes. In fact, any IND-CPA PKE, which is secure when the input comes from a high min-entropy distribution, is admissible for our scheme. Compared to [24], we do not need the one-wayness under correlated inputs (Def. 5).

Our scheme consists of three main steps. These steps are described in details in the corresponding subsections as shown in the following figure:

$$\text{IND-CPA} \overset{\text{Sec.3.1}}{\Longrightarrow} \text{k-IND-CPA} \overset{\text{Sec.3.2}}{\Longrightarrow} \text{k-PRIV-CPA} \overset{\text{Sec.3.3}}{\Longrightarrow} \text{PRIV-CCA}$$

3.1 *k*-correlated IND-CPA PKE from IND-CPA PKE

It has been proven by Bellare et al. [3] that for indistinguishability, the security in a single user setting implies that in a multiple user setting. Hence, it immediately leads to the following construction in Table 1, which is secure even for k-correlated input. Given an IND-CPA secure PKE scheme $\overline{\mathit{\Pi}} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$, it builds k-correlated IND-CPA secure PKE scheme $\overline{\mathit{\Pi}}_k = (\overline{\mathcal{K}}_k, \overline{\mathcal{E}}_k, \overline{\mathcal{D}}_k)$. In the following, define $k = k(\lambda)$ where $k(\cdot)$ is a polynomial. At first, invoke the key generation algorithm k times independently, and output $(\overline{\mathbf{PK}}, \overline{\mathbf{SK}})$. Next, use the encryption algorithm $\overline{\mathcal{E}}$ to encrypt the uniformly chosen x with k different public-keys $\{\mathsf{pk}_i\}_{(1 \le i \le k)}$. Decryption $\overline{\mathcal{D}}$ is done with respect to corresponding secret key.

```
Algorithm $\overline{\mathcal{K}}_k(1^\lambda)$ :
    $(\overline{\mathsf{pk}}_1, \overline{\mathsf{sk}}_1) \overset{\$}{\leftarrow} \overline{\mathcal{K}}(1^\lambda)$
        $\vdots$            $\vdots$
    $(\overline{\mathsf{pk}}_k, \overline{\mathsf{sk}}_k) \overset{\$}{\leftarrow} \overline{\mathcal{K}}(1^\lambda)$
    $\overline{\mathbf{PK}} \leftarrow \{\overline{\mathsf{pk}}_1, \ldots, \overline{\mathsf{pk}}_k\}$
    $\overline{\mathbf{SK}} \leftarrow \{\overline{\mathsf{sk}}_1, \ldots, \overline{\mathsf{sk}}_k\}$
    Return $(\overline{\mathbf{PK}}, \overline{\mathbf{SK}})$
```

```
Algorithm $\overline{\mathcal{E}}_k(1^\lambda, \overline{\mathbf{PK}}, x)$ :
    $\{\overline{\mathsf{pk}}_1, \ldots, \overline{\mathsf{pk}}_k\} \leftarrow \overline{\mathbf{PK}}$
    $c_i \overset{\$}{\leftarrow} \overline{\mathcal{E}}(1^\lambda, \overline{\mathsf{pk}}_i, x)$, $(1 \le i \le k)$
    $c \leftarrow (c_1, \ldots, c_k)$
    Return $c$
```

```
Algorithm $\overline{\mathcal{D}}_k(1^\lambda, \overline{\mathbf{SK}}, c)$ :
    $\{\overline{\mathsf{sk}}_1, \ldots, \overline{\mathsf{sk}}_k\} \leftarrow \overline{\mathbf{SK}}$
    $(c_1, \ldots, c_k) \leftarrow c$
    $x_i \leftarrow \overline{\mathcal{D}}(1^\lambda, \mathsf{sk}_i, c_i)$, $(1 \le i \le k)$
    If $x_1 = \cdots = x_k$, return $x$
    Otherwise, return $\perp$
```

Table 1. k-correlated IND-CPA Construction from IND-CPA.

Theorem 1. *[3] Let $\mathcal{A}_k$ be any PPT adversary against k-correlated IND-CPA scheme $\overline{\mathit{\Pi}}_k$, then there exists a PPT adversary $\mathcal{A}$ who can break the IND-CPA secure $\overline{\mathit{\Pi}}$, with the following* *probability,*

$$\mathbf{Adv}^{k\text{-}ind\text{-}cpa}_{\overline{\Pi}_k,\mathcal{A}_k}(\lambda) \le q_e k \cdot \mathbf{Adv}^{ind\text{-}cpa}_{\overline{\Pi},\mathcal{A}}(\lambda)$$

where $\mathcal{A}$ asks at most q_e queries to any of its k encryption oracles.

Proof. Refer to [3] for details.

The above result can be proven to work for CCA security as well. But it is only effective to indistinguishability notion of PKE, and can hardly be extended to the case of deterministic encryption in general, as [4] observed. More precisely, unlike conventional IND-CPA/CCA PKE scheme, PRIV security in single public-key (message, respectively) case does not imply security in multiple public-keys (messages, respectively) case (see the discussion on the difference between PRIV and PRIV1 security in [4, Sec. 3] and [10, Sec. 3]). This fact provides an evidence that CCA security for the PRIV security notion is difficult to obtain in the standard model.

3.2 k-correlated PRIV-CPA DE from k-correlated IND-CPA PKE

In the paper by Bellare et al [6], a standard model construction of PRIV-CPA deterministic encryption has been proposed. The main idea of their scheme is to replace the random coins of encryption algorithm, with a "pseudorandom coin" output by pseudorandom generators (PRG), such as Blum-Micali-Yao [9, 26], Goldreich-Levin [17]. The next construction for PRIV secure deterministic encryption is inspired by Bellare et al's scheme [6], but works in a more general setting, where k independent instances of encryption are employed with the correlated input.

Given a k-correlated IND-CPA secure PKE scheme $\overline{\Pi}_k = (\overline{\mathcal{K}}_k, \overline{\mathcal{E}}_k, \overline{\mathcal{D}}_k)$, and a family of trapdoor permutations $\mathcal{P} = (\mathsf{G}, \mathsf{F}, \mathsf{D})$, it is possible to build the following scheme $\Pi_k = (\mathcal{K}_k, \mathcal{E}_k, \mathcal{D}_k)$ that is composed of $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where Π_k is k-correlated PRIV-CPA secure deterministic encryption, for uniformly chosen $x \in \{0,1\}^\lambda$. Before explaining Π_k, let us look at each component of $\overline{\Pi}_k$ and Π_k. In Table 2, each component $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of Π_k is built from $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. For input of vector $\boldsymbol{x}$, $\mathsf{F}(\boldsymbol{x})$ denotes the vector whose i-th component is $\mathsf{F}(\boldsymbol{x}[i])$. Both $\mathcal{PG}(1^\lambda, 1^{n(\lambda)}, \phi, \boldsymbol{x}, s)$ and $\mathsf{F}^{n(\lambda)}_\phi(\boldsymbol{x})$ described below are component-wise operations as well.

Algorithm $\mathcal{K}(1^\lambda)$:	**Algorithm** $\mathcal{E}(1^\lambda, \mathsf{pk}, x)$:	**Algorithm** $\mathcal{D}(1^\lambda, \mathsf{sk}, c)$:
$(\phi, \tau) \xleftarrow{\$} \mathsf{G}(1^\lambda)$ $s \xleftarrow{\$} \{0,1\}^\lambda$ $(\overline{\mathsf{pk}}, \overline{\mathsf{sk}}) \xleftarrow{\$} \overline{\mathcal{K}}(1^\lambda)$ $\mathsf{pk} \leftarrow (\phi, \overline{\mathsf{pk}}, s)$ $\mathsf{sk} \leftarrow (\tau, \overline{\mathsf{sk}})$ Return $(\mathsf{pk}, \mathsf{sk})$	$(\phi, \overline{\mathsf{pk}}, s) \leftarrow \mathsf{pk}$ $y \leftarrow \mathsf{F}^{n(\lambda)}_\phi(x)$ $\omega \leftarrow \mathcal{PG}(1^\lambda, 1^{n(\lambda)}, \phi, x, s)$ $c \leftarrow \overline{\mathcal{E}}(1^\lambda, \overline{\mathsf{pk}}, y; \omega)$ Return c	$(\tau, \overline{\mathsf{sk}}) \leftarrow \mathsf{sk}$ $y \leftarrow \overline{\mathcal{D}}(1^\lambda, \mathsf{sk}, c)$ $x \leftarrow \mathsf{D}^{n(\lambda)}_\tau(y)$ Return x

Table 2. PRIV-CPA Construction from IND-CPA

In the above, the random coin ω of the IND-CPA scheme $\overline{\Pi}$ is not directly generated at random but built from a random string s and uniform input x of high min-entropy μ. More precisely, on input 1^λ, the key generation algorithm G randomly generates an index (ϕ, τ) of trapdoor permutation family $\mathcal{P}$, where $\mathsf{F}_\phi(\cdot) = \mathsf{F}(\phi, \cdot)$, $\mathsf{D}_\tau(\cdot) = \mathsf{D}(\tau, \cdot)$ and $\mathsf{D}_\tau(\mathsf{F}_\phi(x)) = x$. Define a permutation f^i: $\{0,1\}^\lambda \to \{0,1\}^\lambda$ inductively from f by $f^0(x) = x$ and $f^{i+1}(x) = f(f^i(x))$ for $i \ge 0$ and $x \in \{0,1\}^\lambda$. Then, a pseudorandom coin ω is generated by Blum-Micali-Yao [9, 26], and Goldreich-Levin [17] generator $\mathcal{PG}$, s.t. on input $(1^\lambda, 1^{n(\lambda)}, \phi, x \in \{0,1\}^\lambda, s \in \{0,1\}^\lambda)$, where h denotes the hard-core predicate,

$$\begin{aligned}\omega &= \mathcal{PG}(1^\lambda, 1^{n(\lambda)}, \phi, x, s)\\ &= h(\mathsf{F}^0_\phi(x), s) \circ h(\mathsf{F}^1_\phi(x), s) \circ \ldots \circ h(\mathsf{F}^{n-1}_\phi(x), s).\end{aligned}$$

Lemma 1. *[6, Theorem.5.3] Given an IND-CPA encryption $\overline{\Pi}$, and a family of trapdoor permutations $\mathcal{P} = (\mathsf{G}, \mathsf{F}, \mathsf{D})$, let $\mathcal{A}$ be an adversary against the associated deterministic encryption $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with advantage $\mathbf{Adv}^{\textit{priv-cpa}}_{\Pi,\mathcal{A}}$ and number of messages $v(\cdot)$, then there exists an IND-CPA adversary $\mathcal{B}$ and an inversion adversary $\mathcal{J}$ against trapdoor permutations $\mathcal{P}$, s.t. for any $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}^{\textit{priv-cpa}}_{\Pi,\mathcal{A}}(\lambda) \leq 2 \cdot \mathbf{Adv}^{\textit{ind-cpa}}_{\overline{\Pi},\mathcal{B}}(\lambda) + 16n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$$

where, $n(\lambda)$ is the length of the randomness of encryption algorithm $\overline{\mathcal{E}}$.

In our construction, k-correlated PRIV-CPA secure deterministic encryption is essential for obtaining CCA security via Correlated Products. Hence, we need the following Π_k to be PRIV-CPA secure as well. The whole scheme $\Pi_k = (\mathcal{K}_k, \mathcal{E}_k, \mathcal{D}_k)$ is composed of Π as Table 3.

Algorithm $\mathcal{K}_k(1^\lambda)$:	**Algorithm** $\mathcal{E}_k(1^\lambda, \mathbf{PK}, x)$:	**Algorithm** $\mathcal{D}_k(1^\lambda, \mathbf{SK}, c)$:
$(\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \mathcal{K}(1^\lambda)$ $\vdots \qquad \vdots$ $(\mathsf{pk}_k, \mathsf{sk}_k) \xleftarrow{\$} \mathcal{K}(1^\lambda)$ $\mathbf{PK} \leftarrow \{\mathsf{pk}_1, \ldots, \mathsf{pk}_k\}$ $\mathbf{SK} \leftarrow \{\mathsf{sk}_1, \ldots, \mathsf{sk}_k\}$ Return $(\mathbf{PK}, \mathbf{SK})$	$\{\mathsf{pk}_1, \ldots, \mathsf{pk}_k\} \leftarrow \mathbf{PK}$ $c_i \xleftarrow{\$} \mathcal{E}(1^\lambda, \mathsf{pk}_i, x)$, $(1 \leq i \leq k)$ $c \leftarrow (c_1, \ldots, c_k)$ Return c	$\{\mathsf{sk}_1, \ldots, \mathsf{sk}_k\} \leftarrow \mathbf{SK}$ $(c_1, \ldots, c_k) \leftarrow c$ $x_i \leftarrow \mathcal{D}(1^\lambda, \mathsf{sk}_i, c_i)$, $(1 \leq i \leq k)$ If $x_1 = \cdots = x_k$, return x Otherwise, return $\perp$

Table 3. k-correlated PRIV-CPA Construction.

Since the algorithms (F, D) of $\mathcal{P}$ are both deterministic, it is easy to see that given the input of pk and x, the output ciphertext is uniquely determined. Next, we show that the pseudorandom coin is sufficient to transform IND-CPA PKE scheme $\overline{\Pi}_k = (\overline{\mathcal{K}}_k, \overline{\mathcal{E}}_k, \overline{\mathcal{D}}_k)$ to PRIV-CPA deterministic encryption scheme $\Pi_k = (\mathcal{K}_k, \mathcal{E}_k, \mathcal{D}_k)$.

Theorem 2. *Given a family of trapdoor permutations $\mathcal{P} = (\mathsf{G}, \mathsf{F}, \mathsf{D})$ and IND-CPA secure PKE scheme $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. Let $\mathcal{A}$ be an adversary against the associated scheme of $\Pi_k = (\mathcal{K}_k, \mathcal{E}_k, \mathcal{D}_k)$ with advantage $\mathbf{Adv}^{\textit{priv-cpa}}_{\Pi_k,\mathcal{A}}$ and number of correlated messages $v(\cdot)$, then there is an IND-CPA adversary $\mathcal{B}$ and an inversion adversary $\mathcal{J}$ against trapdoor permutations $\mathcal{P}$, s.t. for any $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}^{\textit{priv-cpa}}_{\Pi_k,\mathcal{A}}(\lambda) \leq 2q_e k(\lambda) \cdot \mathbf{Adv}^{\textit{ind-cpa}}_{\overline{\Pi},\mathcal{B}}(\lambda) + 16k(\lambda)n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$$

Proof. We refer to Appendix B for the proof of Theorem 2.

3.3 PRIV-CCA Security via Correlated Products

In the following, we finally lift up the underlying k-correlated CPA-secure deterministic encryption scheme Π_k to PRIV-CCA secure efficiently searchable (but not deterministic) scheme Π_{cca}. The following scheme $\Pi_{cca} = (\mathcal{K}_{cca}, \mathcal{E}_{cca}, \mathcal{D}_{cca})$ is obtained from k-correlated scheme Π_k, which is built from PRIV-CPA secure scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The encryption/decryption algorithms are component-wise operation, as well.

In Table 4, Π is built from IND-CPA secure $\overline{\Pi}$ with the help of trapdoor permutations to fix the random coin with respect to pk and input x. Therefore, each encryption algorithm

```
Algorithm $\mathcal{K}_{cca}(1^\lambda)$ :
    $(\mathsf{pk}_1^0, \mathsf{sk}_1^0) \xleftarrow{\$} \mathcal{K}(1^\lambda)$
    $\vdots \quad \vdots$
    $(\mathsf{pk}_k^0, \mathsf{sk}_k^0) \xleftarrow{\$} \mathcal{K}(1^\lambda)$
    $(\mathsf{pk}_1^1, \mathsf{sk}_1^1) \xleftarrow{\$} \mathcal{K}(1^\lambda)$
    $\vdots \quad \vdots$
    $(\mathsf{pk}_k^1, \mathsf{sk}_k^1) \xleftarrow{\$} \mathcal{K}(1^\lambda)$
    $\mathbf{PK} \leftarrow \{\mathsf{pk}_1^0, .., \mathsf{pk}_k^0;$
              $\mathsf{pk}_1^1, .., \mathsf{pk}_k^1\}$
    $\mathbf{SK} \leftarrow \{\mathsf{sk}_1^0, .., \mathsf{sk}_k^0;$
              $\mathsf{sk}_1^1, .., \mathsf{sk}_k^1\}$
    Return $(\mathbf{PK}, \mathbf{SK})$
```

```
Algorithm $\mathcal{E}_{cca}(1^\lambda, \mathbf{PK}, x)$ :
    $(\mathsf{vk}, \mathsf{sigk}) \xleftarrow{\$} \mathcal{SG}(1^\lambda)$
    wlog, set $|\mathsf{vk}| = k$,
    $\mathsf{vk} = \mathsf{vk}_1 \circ \mathsf{vk}_2 \circ \cdots \circ \mathsf{vk}_k$
    $\{\mathsf{pk}_1^{\mathsf{vk}_1}, \ldots, \mathsf{pk}_k^{\mathsf{vk}_k}\} \leftarrow \mathbf{PK}$
    $c_i \xleftarrow{\$} \mathcal{E}(1^\lambda, \mathsf{pk}_i^{\mathsf{vk}_i}, x)$,
              $(1 \leq i \leq k)$
    $c \leftarrow (c_1, \ldots, c_k)$
    $\sigma \xleftarrow{\$} \mathcal{SS}(1^\lambda, \mathsf{sigk}, c)$
    $C \leftarrow (\mathsf{vk}, c, \sigma)$
    Return $C$
```

```
Algorithm $\mathcal{D}_{cca}(1^\lambda, \mathbf{SK}, C)$ :
    $(\mathsf{vk}, c, \sigma) \leftarrow C$
    If $0 \leftarrow \mathcal{SV}(1^\lambda, \mathsf{vk}, c, \sigma)$
    Return $\perp$ and halt.
    Otherwise,
    $(c_1, \ldots, c_k) \leftarrow c$
    $\{\mathsf{sk}_1^{\mathsf{vk}_1}, \ldots, \mathsf{sk}_k^{\mathsf{vk}_k}\} \leftarrow \mathbf{SK}$
    $x_i \leftarrow \mathcal{D}(1^\lambda, \mathsf{sk}_i^{\mathsf{vk}_i}, c_i)$,
              $(1 \leq i \leq k)$
    If $x_1 = \cdots = x_k$,
    Return $x = x_1$
    Otherwise, return $\perp$
```

Table 4. PRIV-CCA Construction from k-correlated PRIV-CPA

$\mathcal{E}(1^\lambda, \mathsf{pk}_i, \cdot)$ in our proposed construction is an injective mapping under certain pk_i, which can be re-encrypted to have a validity check as in Rosen-Segev's construction [24]. Since its one-wayness under k-correlated inputs has been naturally inherited in our transformation, we do not need to an extra assumption that all the correlated inputs are one-way. This helps us to encrypt a message which is longer message than one bit. A strongly unforgeable one-time signature (vk, σ) is generated as well in order to thwart CCA attack, which reminiscently follows the seminal construction of DDN [14].

To provide some intuition on the security of this scheme, we note even though all the ciphertexts contain the same input message $x \in \{0,1\}^\lambda$, the randomness used by each encryption algorithm $\mathcal{E}$ could still be considered fresh because it is obtained via Blum-Micali-Yao, Goldreich-Levin PRG from each pk and x. Note that in the scheme $\varPi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ described in Table 2, $\mathsf{pk} = (\phi, \overline{\mathsf{pk}}, s)$ has sufficient high min-entropy, where $\overline{\mathsf{pk}}$ is the encryption key of IND-CPA scheme $\overline{\varPi}$, ϕ and s are randomly generated. Next, we look at the CCA security proof of the proposal scheme $\varPi_{cca}$ in Table 4.

Theorem 3. *Given a k-correlated PRIV-CPA secure deterministic encryption $\varPi_k$, and a strongly unforgeable one-time signature $\mathcal{SIG}$, the associated encryption scheme $\varPi_{cca}$ is PRIV-CCA secure. More precisely, if there exists any PPT adversary $\mathcal{A}$ against the encryption scheme $\varPi_{cca}$ with advantage of $\mathbf{Adv}^{priv\text{-}cca}_{\varPi_{cca},\mathcal{A}}$, then there is a PRIV-CPA adversary $\mathcal{B}$ and a forger $\mathcal{F}$ against one-time signature $\mathcal{SIG}$, s.t. for any $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}^{priv\text{-}cca}_{\varPi_{cca},\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{priv\text{-}cpa}_{\varPi_k,\mathcal{B}}(\lambda) + 2\mathbf{Adv}^{suf\text{-}cma}_{\mathcal{SIG},\mathcal{F}}(\lambda) + \frac{1}{2^{k(\lambda)-1}}$$

Proof. We refer to Appendix C for proof of Theorem of 3.

Corollary 1. *Given IND-CPA secure PKE with input from a high min-entropy domain, and trapdoor permutations, it is possible to obtain PRIV-CCA security.*

Proof. It is clear to see that from Theorem 1,2,3, IND-CPA security can be converted to PRIV-CCA security. □

Corollary 2. *Given trapdoor permutations solely, it is possible to achieve PRIV-CCA security.*

Proof. According to Goldwasser-Micali [18] and Yao [26], since trapdoor permutations imply semantic security (IND-CPA), the condition of Corollary 1 can be re-written as "trapdoor permutations" instead of "IND-CPA security". □

Remark. We have proposed the general construction of PRIV-CCA security, from IND-CPA security, by first building k-correlate PRIV-CPA and then lifting up to CCA. A question may be raised whether it is possible to change the order of our transformation, such that, first building IND-CCA via Correlated Products and then derandomizing to PRIV-CCA? The answer seems to be negative, because Correlated Products work well only with injective trapdoor function as a building block and it is not known how to adapt IND-CPA PKE to Correlated Products setting, in general. This fact also indicates that our result is not trivial to obtain from the Correlated Products, although our security goal is different from the one in [24].

4 Efficiently Searchable Encryption from Coding-Theoretic Assumptions

Our proposal achieves PRIV-CCA security from IND-CPA PKE, in a general way. However, since it employs Blum-Micali-Yao, Goldreich-Levin PRGs repetitively, the general construction is not so efficient. Note that PRIV security is developed mainly for the practical purpose of searchable encryption, it is highly desired to get more efficient PRIV-CCA secure scheme. For most of the applications, it is good to search a certain field of the data with a universal format, instead of the whole data. In other words, there is a way to obtain more efficient searchable encryption by setting the encrypted data contain a certain field for searching particularly, rather than generate the whole ciphertext in a deterministic way.

In this section, we propose a more efficient PRIV-CCA secure searchable encryption, by (wlog.) setting the searching tag as the first component of the ciphertext. Our construction is more efficient than the scheme in Sec.3.3 but relies on a specific assumption of syndrome decoding, thus is not generally achievable anymore.

The following scheme $\mathit{\Pi}_s = (\mathcal{K}_s, \mathcal{E}_s, \mathcal{D}_s)$ in Table 5 is obtained by trapdoor permutations $\mathcal{P} = (\mathsf{G}, \mathsf{F}, \mathsf{D})$ and k instances of IND-CPA PKE scheme $\overline{\mathit{\Pi}} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ instantiated by McEliece encryption [21, 15], in addition to a strongly unforgeable one-time signature $\mathcal{SIG} = (\mathcal{SG}, \mathcal{SS}, \mathcal{SV})$. According to Sec.3.2, IND-CPA PKE and trapdoor permutations generate PRIV-CPA secure deterministic encryption $\mathit{\Pi} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. We set the first component of the ciphertext c as the searching tag, which is generated by a PRIV-CPA secure scheme $\mathit{\Pi}$, indicated in the boxed statement in Table 5. Let c_1 be generated by PRIV-CPA encryption, i.e., $y \leftarrow \mathsf{F}_{\phi}^{n(\lambda)}(x)$, $\omega \leftarrow \mathcal{PG}(1^{\lambda}, 1^{n(\lambda)}, \phi, x, s)$, $c \leftarrow \overline{\mathcal{E}}(1^{\lambda}, \overline{\mathsf{pk}}, y; \omega)$; and $(c_2, \ldots, c_k)$ are normally output by the encryption algorithm of IND-CPA scheme.

Since repetitive operations to generate pseudorandom coin by PRG are costly, we simply employ one PRIV-CPA encryption for searching, and make use of other $k-1$ IND-CPA encryptions instead of PRIV-CPA encryptions. Unfortunately, this replacement, in general, will fail the security proof because the $k-1$ instances of (probabilistic) IND-CPA encryptions are no longer injective mapping, thus can not be verified by re-encrypting the message with corresponding public-key. Thanks to the bounded distance decoding property of code-based PKE we are able to verify the ciphertext, without knowing all randomness in encryption algorithm.

Randomized McEliece PKE. To obtain IND-CPA security, a randomized McEliece encryption in the standard model has been proposed [21, 15], which requires that input to be m padded with random r. In our construction, the input is uniformly chosen from a high min-entropy domain, so it actually satisfies the above requirement, for example, $x = x_m \circ x_r$, where x_r could be built from a bijective encoding from x_m. And the ciphertext validity could be checked as that the Hamming weight of (possible) ciphertexts of the same input must be in a Hamming ball with radius of t, s.t. t is the error-correcting capability of error-correcting code that McEliece encryption uses, say, binary Goppa code.

More precisely, look at the McEliece encryption: $xG \oplus e$, where G is the public key, e with Hamming weight at most t is the random coin in encryption algorithm, and input $x = x_m \circ x_r$ are padded and uniformly chosen. Then, with the knowledge of x, it is possible to check in $c = x'G \oplus e$ whether $x' = x$ by computing the Hamming distance of $(x'G \oplus e)$ to xG. If the distance is no more than t, then according to bounded distance decoding, they are decoded to the same x. Otherwise, $x \neq x'$. This fact has also been used by [15] to obtain the first CCA-secure McEliece encryption in the standard model.

```
Algorithm $\mathcal{K}_s(1^\lambda)$ :
  $(\overline{\mathsf{pk}}_1^0, \overline{\mathsf{sk}}_1^0) \xleftarrow{\$} \overline{\mathcal{K}}(1^\lambda)$
      $\vdots$          $\vdots$
  $(\overline{\mathsf{pk}}_k^0, \overline{\mathsf{sk}}_k^0) \xleftarrow{\$} \overline{\mathcal{K}}(1^\lambda)$
  $(\overline{\mathsf{pk}}_1^1, \overline{\mathsf{sk}}_1^1) \xleftarrow{\$} \overline{\mathcal{K}}(1^\lambda)$
      $\vdots$          $\vdots$
  $(\overline{\mathsf{pk}}_k^1, \overline{\mathsf{sk}}_k^1) \xleftarrow{\$} \overline{\mathcal{K}}(1^\lambda)$
  $(\phi, \tau) \xleftarrow{\$} \mathsf{G}(1^\lambda)$, $s \xleftarrow{\$} \{0,1\}^\lambda$
  $\boxed{\mathsf{pk}_1^0 \leftarrow (\overline{\mathsf{pk}}_1^0, \phi, s),\ \mathsf{sk}_1^0 \leftarrow (\overline{\mathsf{sk}}_1^0, \tau)}$
  $\boxed{\mathsf{pk}_1^1 \leftarrow (\overline{\mathsf{pk}}_1^1, \phi, s),\ \mathsf{sk}_1^1 \leftarrow (\overline{\mathsf{sk}}_1^1, \tau)}$
  $\mathbf{PK} \leftarrow \{\mathsf{pk}_1^0, \overline{\mathsf{pk}}_2^0, \ldots, \overline{\mathsf{pk}}_k^0;$
            $\mathsf{pk}_1^1, \overline{\mathsf{pk}}_2^1, \ldots, \overline{\mathsf{pk}}_k^1\}$
  $\mathbf{SK} \leftarrow \{\mathsf{sk}_1^0, \overline{\mathsf{sk}}_2^0, \ldots, \overline{\mathsf{sk}}_k^0;$
            $\mathsf{sk}_1^1, \overline{\mathsf{sk}}_2^1, \ldots, \overline{\mathsf{sk}}_k^1\}$
  Return $(\mathbf{PK}, \mathbf{SK})$
```

```
Algorithm $\mathcal{E}_s(1^\lambda, \mathbf{PK}, x)$ :
  $(\mathsf{vk}, \mathsf{sigk}) \xleftarrow{\$} \mathcal{SG}(1^\lambda)$
  wlog, set $|\mathsf{vk}| = k$,
  $\mathsf{vk} = \mathsf{vk}_1 \circ \mathsf{vk}_2 \circ \cdots \circ \mathsf{vk}_k$
  $\{\mathsf{pk}_1^{\mathsf{vk}_1}, \ldots, \overline{\mathsf{pk}}_k^{\mathsf{vk}_k}\} \leftarrow \mathbf{PK}$
  $\boxed{c_1 \xleftarrow{\$} \mathcal{E}(1^\lambda, \mathsf{pk}_1^{\mathsf{vk}_1}, x),}$
  $c_i \xleftarrow{\$} \overline{\mathcal{E}}(1^\lambda, \overline{\mathsf{pk}}_i^{\mathsf{vk}_i}, x),$
          $(2 \leq i \leq k)$
  $c \leftarrow (c_1, \ldots, c_k)$
  $\sigma \xleftarrow{\$} \mathcal{SS}(1^\lambda, \mathsf{sigk}, c)$
  $C \leftarrow (\mathsf{vk}, c, \sigma)$
  Return $C$
```

```
Algorithm $\mathcal{D}_s(1^\lambda, \mathbf{SK}, C)$ :
  $(\mathsf{vk}, c, \sigma) \leftarrow C$
  If $0 \leftarrow \mathcal{SV}(1^\lambda, \mathsf{vk}, c, \sigma)$
  Return $\perp$ and halt.
  Otherwise,
  $(c_1, \ldots, c_k) \leftarrow c$
  $\{\mathsf{sk}_1^{\mathsf{vk}_1}, \ldots, \overline{\mathsf{sk}}_k^{\mathsf{vk}_k}\} \leftarrow \mathbf{SK}$
  $\boxed{x_1 \leftarrow \mathcal{D}(1^\lambda, \mathsf{sk}_1^{\mathsf{vk}_1}, c_1),}$
  $x_i \leftarrow \overline{\mathcal{D}}(1^\lambda, \overline{\mathsf{sk}}_i^{\mathsf{vk}_i}, c_i),$
          $(2 \leq i \leq k)$
  If $x_1 = \cdots = x_k$,
  Return $x = x_1$
  Otherwise, return $\perp$
```

Table 5. One PRIV-CPA + $(k-1)$ IND-CPA Construction

In the construction of Table 5, Randomized McEliece encryption $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ is IND-CPA secure for input x_m proven by [21], but could be re-encrypted to verify the pair of plaintext and ciphertext given $x = x_m \circ x_r$. Thanks to this property, we can use one instance of PRIV-CPA encryption (boxed statement) as described in Sec. 3.2, and leave the $k-1$ IND-CPA secure encryption as it is. For efficient searching over encrypted data, it is sufficient to find and compare c_1 part. And for the security of the scheme, we prove in the following theorem.

Theorem 4. *Given IND-CPA secure Randomized McEliece encryption $\overline{\Pi}$, a collection of trapdoor permutations $\mathcal{P}$ and a strongly unforgeable one-time signature $\mathcal{SIG}$, the associated scheme Π_s is PRIV-CCA secure. More precisely, if there exists any PPT adversary $\mathcal{A}$ against Π_s with advantage of $\mathbf{Adv}_{\Pi_s,\mathcal{A}}^{\textit{priv-cca}}$, then there is an IND-CPA adversary $\mathcal{B}$, an inverter $\mathcal{J}$ against trapdoor permutations $\mathcal{P}$ and a forger $\mathcal{F}$ against signature $\mathcal{SIG}$, s.t. for any $\lambda \in \mathbb{N}$,*

$$\mathbf{Adv}_{\Pi_s,\mathcal{A}}^{\textit{priv-cca}}(\lambda) \leq 2q_e k(\lambda) \cdot \mathbf{Adv}_{\overline{\Pi},\mathcal{B}}^{\textit{ind-cpa}}(\lambda) + 16n(\lambda)v(\lambda) \cdot \mathbf{Adv}_{\mathcal{P},\mathcal{J}}^{\textit{ow}}(\lambda) + 2\mathbf{Adv}_{\mathcal{SIG},\mathcal{F}}^{\textit{suf-cma}}(\lambda) + \frac{1}{2^{k(\lambda)-1}}$$

Proof. We refer to Appendix D for the proof of Theorem 4.

5 Conclusion

In this paper, we have proposed a general construction to build PRIV-CCA secure efficiently searchable encryption from IND-CPA secure public-key encryption (also say, from trapdoor permutations). Our approach is employing Rosen-Segev's Correlated Products, but has removed its strong assumption of one-wayness under correlated inputs, thus has capability in general use. In addition, we have provided a more efficient searchable encryption, which satisfies PRIV-CCA security, by using Randomized McEliece encryption.

References

1. J. An, Y. Dodis and T. Rabin. "On the Security of Joint Signature and Encryption". EUROCRYPT'02, LNCS 2332, pp. 83-107, 2002.
2. D. Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," CRYPTO'98, LNCS 1462, pp.1-12, 1998.
3. M. Bellare, A. Boldyreva and S. Micali,"Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements," EUROCRYPT'00, LNCS 1807, pp. 259-274, 2000.
4. M. Bellare, A. Boldyreva and A. O'Neill, "Deterministic and Efficiently Searchable Encryption," CRYPTO'07, LNCS 4622, pp. 535-552, 2007.
5. M. Bellare, A. Boldyreva and A. Palacio, "An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem," EUROCRYPT'04, LNCS 3027, pp. 171-188, 2004.
6. M. Bellare, M. Fischlin, A. O'Neill and T. Ristenpart, "Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles," CRYPTO'08, LNCS 5157, pp. 360-378, 2008.
7. M. Bellare, M. Fischlin, A. O'Neill and T. Ristenpart, "Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles," Cryptology ePrint Archive: Report 2008/267. Available at: `http://eprint.iacr.org/2008/267`.
8. D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public Key Encryption with Keyword Search," EUROCRYPT'04, LNCS 3027, pp. 506-522, 2004.
9. M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudorandom bits," SIAM Journal on Computing, vol. 13, pp. 850-864, 1984.
10. A. Boldyreva, S. Fehr and A. O'Neill, "On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles," CRYPTO'08, LNCS 5157, pp. 335-359, 2008.
11. R. Canetti, O. Goldreich, S. Halevi: The random oracle methodology, revisited. J. ACM 51(4): 557-594, 2004.
12. Y. Cui, K. Morozov, K. Kobara and H. Imai, "Efficient Constructions of Deterministic Encryption from Hybrid Encryption and Code-Based PKE," AAECC'09, LNCS 5527, pp. 159-168, 2009.
13. R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," SIAM Journal on Computing, Volume 33, Number 1, pp. 167-226 (2003).
14. D. Dolev, C. Dwork and M. Naor, "Non-Malleable Cryptography (Extended Abstract)," ACM STOC'91, pp.542-552, 1991. Journal version in SIAM Journal on Computing, 30(2), pp.391-437, 2000.
15. R. Dowsley, J. Muller-Quade, A. Nascimento, "A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model," CT-RSA'09, LNCS 5473, pp. 240-251, 2009.
16. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen and G. Segev, "More Constructions of Lossy and Correlation-Secure Trapdoor Functions," PKC'10, LNCS 6056, pp. 279-295, 2010.
17. O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions," ACM STOC'89, pp. 25-32, 1989.
18. S. Goldwasser and S. Micali. "Probabilistic encryption," Journal of Computer and System Sciences, 28(2), pp. 270-299, 1984.
19. J. Håstad, "Solving simultaneous modular equations of low degree," SIAM Journal on Computing, 17(2), pp. 336-341, 1988.
20. M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," ACM STOC'90, pp.427-437, 1990.
21. R. Nojima, H. Imai, K. Kobara and K. Morozov, "Semantic Security for the McEliece Cryptosystem without Random Oracles," Designs, Codes and Cryptography, vol. 49, no. 1-3, pp. 289-305, 2008.
22. C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," ACM STOC'09, pp. 333-342, 2009.
23. C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," ACM STOC'08, pp. 187-196, 2008.
24. A. Rosen and G. Segev, "Chosen-Ciphertext Security via Correlated Products," TCC'09, LNCS 5444, pp. 419-436, 2009.
25. V. Shoup, "Sequences of Games: A Tool for Taming Complexity in Security Proofs". Available from `http://www.shoup.net` .
26. A. Yao, "Theory and applications of trapdoor functions," IEEE FOCS'82, pp. 80-91, 1982.

A Rosen-Segev's Correlated Products

Now we recall the Rosen-Segev's construction for building CCA security, s.t. given a family of injective trapdoor function $\mathcal{F}_n = (\mathsf{G}, \mathsf{F}, \mathsf{D})$, the following scheme provides CCA security.

- Key Generation: On input 1^λ the algorithm invokes G independently for $2k$ times to generate indexes of functions $(f_1^0, f_1^1), \ldots, (f_k^0, f_k^1)$, with corresponding trapdoors $(d_1^0, d_1^1), \ldots, (d_k^0, d_k^1)$. The public and secret key pair $(\mathbf{PK}, \mathbf{SK})$ is output,
$$\mathbf{PK} = (f_1^0, f_1^1), \ldots, (f_k^0, f_k^1) \qquad \mathbf{SK} = (d_1^0, d_1^1), \ldots, (d_k^0, d_k^1)$$
- Encryption: On input security parameter λ and a single-bit message $m \in \{0,1\}$ and a public-key $\mathbf{PK}$, sample a strongly unforgeable one-time signature $(\mathsf{vk}, \mathsf{sigk}) \xleftarrow{\$} \mathcal{SG}(1^\lambda)$. Define $\circ$ as bit concatenation, wlog, $|\mathsf{vk}| = k$ and $\mathsf{vk} = \mathsf{vk}_1 \circ \cdots \circ \mathsf{vk}_k$. Choose a uniformly distributed $x \in \{0,1\}^\lambda$, output $(\mathsf{vk}, y_1, \ldots, y_k, c_1, c_2)$, where h is a hard-core predicate of F.
$$\begin{aligned} y_i &= \mathsf{F}(f_i^{\mathsf{vk}_i}, x) \ \text{ for } (1 \le i \le k) \\ c_1 &= m \oplus h(f_1^{\mathsf{vk}_1}, \ldots, f_k^{\mathsf{vk}_k}, x) \\ c_2 &= \mathcal{SS}(\mathsf{sigk}, (y_1, \ldots, y_k, c_1)) \end{aligned}$$
- Decryption: On input a ciphertext $(\mathsf{vk}, y_1, \ldots, y_k, c_1, c_2)$, decryption algorithm first checks the validity of the ciphertext and then outputs message m: If $\mathcal{SV}(1^\lambda, \mathsf{vk}, (y_1, \ldots, y_k, c_1), c_2) = 0$, then output $\perp$. Otherwise, output $x_i = \mathsf{D}(d_i^{\mathsf{vk}_i}, y_i)$ for $(1 \le i \le k)$. If $x_1 = \cdots = x_k$, then set $x = x_1$ and output $m = c_1 \oplus h(f_1^{\mathsf{vk}_1}, \ldots, f_k^{\mathsf{vk}_k}, x)$, otherwise output $\perp$.

B Proof of Theorem 2

It is possible to take the total k-correlated IND-CPA scheme as one IND-CPA scheme, and employ the trapdoor permutations to generate the pseudorandom coins for that encryption algorithms need. Next, we build pseudorandom coins and show their security requirements.

Note that indexes of trapdoor permutations are independently and randomly generated $(\phi_1, \tau_1) \xleftarrow{\$} \mathsf{G}(1^\lambda)$, $\ldots$, $(\phi_k, \tau_k) \xleftarrow{\$} \mathsf{G}(1^\lambda)$, and random seed is independently generated $s_1 \xleftarrow{\$} \{0,1\}^\lambda, \ldots, s_k \xleftarrow{\$} \{0,1\}^\lambda$, as well. Change the format to vectors $(\boldsymbol{\phi}, \boldsymbol{\tau}) \xleftarrow{\$} \mathsf{G}(1^\lambda)$, $\boldsymbol{s}$, it is obvious to see that the scheme $\mathit{\Pi}_k$ is PRIV-CPA secure, according to Lemma 1. But for the k-correlated security, there is a reduction loss. Because in the following transformation, each row on the right side constitutes a PRIV-CPA secure deterministic encryption as proven in [10], for one row of the pseudorandoness employed, the advantage of distinguisher is $4n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$(based on the analysis of Lemma 1). Using the same x with k instances of encryption, but with distinct and independent public keys and random seeds, it takes $k(\lambda)$ times more than the single case. Hence, the advantage bound gets worse to $4k(\lambda)n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$.

$$\begin{pmatrix} \mathcal{E}_1(\overline{pk}_1, x; \omega_1) \\ \vdots \\ \mathcal{E}_k(\overline{pk}_k, x; \omega_k) \end{pmatrix} \Rightarrow \begin{pmatrix} \overline{\mathcal{E}}_1(\overline{pk}_1, \mathsf{F}^{n(\lambda)}_{\phi_1}(x); h(\mathsf{F}^0_{\phi_1}(x), s_1) \circ h(\mathsf{F}^1_{\phi_1}(x), s_1) \circ \ldots \circ h(\mathsf{F}^{n-1}_{\phi_1}(x), s_1)) \\ \vdots \\ \overline{\mathcal{E}}_k(\overline{pk}_k, \mathsf{F}^{n(\lambda)}_{\phi_k}(x); h(\mathsf{F}^0_{\phi_k}(x), s_k) \circ h(\mathsf{F}^1_{\phi_k}(x), s_k) \circ \ldots \circ h(\mathsf{F}^{n-1}_{\phi_k}(x), s_k)) \end{pmatrix}$$

In the transformed scheme, if there exists any PPT adversary $\mathcal{A}$ against $\mathit{\Pi}_k$ with public key $\{\mathsf{pk}_i\} = \{(\overline{\mathsf{pk}}_i, \phi_i, s_i)\}$, then there must be a $\mathcal{A}_k$ against any one of k scheme $\overline{\mathit{\Pi}}_k$, whose security can be reduced to IND-CPA security of $\overline{\mathcal{E}}$ with public key $\overline{\mathsf{pk}}_i$. Meanwhile, for the k-correlated IND-CPA security, there is a security reduction loss.

$$\mathbf{Adv}^{\text{priv-cpa}}_{\Pi_k,\mathcal{A}}(\lambda) \leq 2 \cdot \mathbf{Adv}^{\text{k-ind-cpa}}_{\overline{\Pi}_k,\mathcal{A}_k}(\lambda) + 16k(\lambda)n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$$

Substitute the advantage of IND-CPA security to k-IND-CPA security, we can get the reduction and complete the proof. □

C Proof of Theorem 3

The proof is provided by a sequence of games G_i, which method is formalized by Shoup [25]. According to the "Difference Lemma" [25], suppose that the events A, B, and F are events defined over some probability distribution, s.t. $A|\neg F \Leftrightarrow B|\neg F$, then there is $|\Pr[A] - \Pr[B]| \leq \Pr[F]$. This fact can be used to prove that $\Pr[A]$ and $\Pr[B]$ are negligibly close if $\Pr[F]$ is negligible.

Define a simulator who simulates the PRIV-CCA interaction to privacy adversary $\mathcal{A}$, with the help of the encryption oracle of Π_k scheme. On input $(1^\lambda, \mathsf{pk}_1, \ldots, \mathsf{pk}_k, c_1, \ldots, c_k)$, the simulator first sets the public and secret key pair $(\mathbf{PK}, \mathbf{SK})$ as follows: generate a strongly unforgeable one-time signature at random, $(\mathsf{vk}^*, \mathsf{sigk}^*) \xleftarrow{\$} \mathcal{SG}(1^\lambda)$, where $\mathsf{vk}^* = \mathsf{vk}_1^* \circ \mathsf{vk}_2^* \circ \cdots \circ \mathsf{vk}_k^*$. Set the $\mathsf{pk}_i^{\mathsf{vk}_i^*} = \mathsf{pk}_i$, and sample another k pairs of $(\mathsf{pk}_i^{1-\mathsf{vk}_i^*}, \mathsf{sk}_i^{1-\mathsf{vk}_i^*}) \xleftarrow{\$} \mathcal{K}(1^\lambda)$, independently for each $1 \leq i \leq k$. In this way, the simulator outputs $\mathbf{PK}$,

$$\mathbf{PK} = (\mathsf{pk}_1^0, \mathsf{pk}_1^1), \ldots, (\mathsf{pk}_k^0, \mathsf{pk}_k^1)$$

in which, for each $(\mathsf{pk}_i^0, \mathsf{pk}_i^1)$, simulator knows one corresponding secret key, but has no idea of the other because it comes from its challenge.

Denote by Forge the event that for one of $\mathcal{A}$'s decryption queries, s.t. $(\mathsf{vk}, (c_1, \ldots, c_k), \sigma)$, there is $\mathcal{SV}(1^\lambda, \mathsf{vk}, (c_1, \ldots, c_k), \sigma) = 1$ and $\mathsf{vk} = \mathsf{vk}^*$. Denote by Succ the event that privacy adversary $\mathcal{A}$ successfully guess b in the privacy experiment $\mathbf{Exp}^{\text{priv-cca-b}}_{\Pi_{cca},\mathcal{A}}(\lambda)$(see Definition 1.)

In the following, denote by $\Pr[\mathsf{G}_i \Rightarrow 1]$ the event that G_i outputs 1. The game G_0 is defined as the original PRIV-CCA attack, and suppose PPT adversary $\mathcal{A}$ can break it with non-negligible probability $\Pr[\mathbf{Exp}^{\text{priv-cca-b}}_{\Pi_{cca},\mathcal{A}}(\lambda) = b]$. Thus, $\Pr[\mathsf{Succ}] = \Pr[\mathsf{G}_0 \Rightarrow 1]$. We have the following inequality, s.t.

$$\begin{aligned}
\frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{priv-cca}}_{\Pi_{cca},\mathcal{A}}(\lambda) &= \Pr[\mathbf{Exp}^{\text{priv-cca-b}}_{\Pi_{cca},\mathcal{A}}(\lambda) = b] \\
&= \Pr[\mathsf{Succ}] \\
&= \Pr[\mathsf{G}_0 \Rightarrow 1] \\
&\leq \Pr[\mathsf{G}_1 \Rightarrow 1] + \Pr[\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]] \\
&\leq \Pr[\mathsf{G}_2 \Rightarrow 1] + \Pr[\mathsf{Forge}] + \Pr[\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]] \\
&= \Pr[\mathsf{G}_3 \Rightarrow 1] + \Pr[\mathsf{Forge}] + \Pr[\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]] \\
&\leq \Pr[\mathbf{Exp}^{\text{priv-cpa-b}}_{\Pi_k,\mathcal{B}}(\lambda) = b] + \Pr[\mathsf{Forge}] + \Pr[\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]] \\
&= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{priv-cpa}}_{\Pi_k,\mathcal{B}}(\lambda) + \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda) + 2^{-k(\lambda)}
\end{aligned}$$

The game G_1 is modified from the game G_0 by replacing real $(\mathbf{PK}, \mathbf{SK})$ by those output by the simulator. Thus, G_1 and G_0 are the same except that the distribution of vk and can be distinguished by $\mathcal{A}$. Denote by $\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]$ the event that adversary manages to distinguish the distributions of $\{\mathsf{vk}\}$ and $\{\mathsf{vk}^*\}$. Note that vk^* is randomly generated and $|\mathsf{vk}| = k$, where

$k = k(\lambda)$ and $k(\cdot)$ is a polynomial, it is easy to see this probability $\Pr[\mathsf{Dist}[\{\mathsf{vk}\}, \{\mathsf{vk}^*\}]]$ is at most $2^{-k(\lambda)}$.

The game G_2 is defined as the same as the game G_1, except that event Forge occurs. Thus there is $|\Pr[\mathsf{G}_1 \Rightarrow 1] - \Pr[\mathsf{G}_2 \Rightarrow 1]| \leq \Pr[\mathsf{Forge}]$. Next we will show that $\Pr[\mathsf{Forge}]$ is negligible. Note that at the beginning of the game, the simulator is given a randomly generated vk^* from the strongly unforgeable one-time signature scheme $\mathcal{SIG}$ and accordingly sets the $(\mathbf{PK}, \mathbf{SK})$. Simulator later invokes the signing oracle to build the challenge ciphertext $(\mathsf{vk}^*, (\mathsf{pk}_1^{\mathsf{vk}_1^*}, \ldots, \mathsf{pk}_k^{\mathsf{vk}_k^*}, c_1^*, \ldots, c_k^*), \sigma^*)$, using the input from Π_k's oracle. In the decryption query phase, $\mathcal{A}$ submits a query s.t. $(\mathsf{vk}, (c_1, \ldots, c_k), \sigma)$ to the simulator. If Forge happens, i.e. $\mathcal{SV}(1^\lambda, \mathsf{vk}, (c_1, \ldots, c_k), \sigma) = 1$ and $\mathsf{vk} = \mathsf{vk}^*$, then the simulator halts the game and outputs $(\mathsf{vk}^*, (c_1, \ldots, c_k), \sigma)$ as the forgery of $\mathcal{SIG}$. Because according to CCA game, the decryption query cannot be equal to the challenge ciphertext, i.e. $\mathsf{vk} = \mathsf{vk}^*$ and $\{(c_1^*, \ldots, c_k^*), \sigma^*\} \neq \{(c_1, \ldots, c_k), \sigma\}$, then $\{(c_1, \ldots, c_k), \sigma\}$ must be a new forgery against the $\mathcal{SIG}$. It is easy to see that $\Pr[\mathsf{Forge}] = \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda)$, and due to the security definition of $\mathcal{SIG}$, this probability is negligible.

The game G_3 simulates the decryption oracle in the following way. When $\mathcal{A}$ submits a query $(\mathsf{vk}, (c_1, \ldots, c_k), \sigma)$, if $\mathsf{vk} = \mathsf{vk}^*$ or $\mathcal{SV}(1^\lambda, \mathsf{vk}, (c_1, \ldots, c_k), \sigma) = 0$, it outputs $\perp$. Otherwise, then $\mathsf{vk} \neq \mathsf{vk}^*$, there is at least one different bit in vk and vk^*. Wlog, set the index j of this position $(1 \leq j \leq k)$, there is $\mathsf{pk}_j^{\mathsf{vk}_j} = \mathsf{pk}_j^{1-\mathsf{vk}_j^*}$. Note that in the beginning, $(\mathsf{pk}_j^{1-\mathsf{vk}_j^*}, \mathsf{sk}_j^{1-\mathsf{vk}_j^*})$ has been generated by the simulator, thus he can make use of $\mathsf{sk}_j^{1-\mathsf{vk}_j^*}$ to decrypt c_j and recover x_j. For $\{c_i\}_{i \neq j}$, it is convenient to compute $c_i' = \mathcal{E}(1^\lambda, \mathsf{pk}_i^{\mathsf{vk}_i}, x_j)$, because the encryption algorithm of Π is an injective mapping under certain public key and input, due to the property of deterministic encryption. For any $1 \leq i \leq k$, if there is $c_i' \neq c_i$, simulator outputs $\perp$. Otherwise, send $\mathcal{A}$ $x = x_j$ as the answer to decryption query. From the above explanation, it is obvious to see that if Forge does not happen, the simulation of decryption oracle is perfect. Then, G_3 is the same as G_2.

Since from the game G_3, the adversary $\mathcal{A}$ cannot get help from the decryption oracle, the privacy attack can be bounded by the success probability of PRIV adversary without decryption query, who is the PRIV-CPA adversary $\mathcal{B}$. Thus, $\Pr[\mathsf{G}_3 \Rightarrow 1] \leq \Pr[\mathbf{Exp}^{\text{priv-cpa-b}}_{\Pi_k,\mathcal{B}}(\lambda) = b]$.

Summarizing all above, we get the underlying inequality, and simplify it to the following,

$$\frac{1}{2} \cdot \mathbf{Adv}^{\text{priv-cca}}_{\Pi_{cca},\mathcal{A}}(\lambda) \leq \frac{1}{2} \cdot \mathbf{Adv}^{\text{priv-cpa}}_{\Pi_k,\mathcal{B}}(\lambda) + \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda) + 2^{-k(\lambda)}$$

which completes the proof. □

D Proof of Theorem 4

Following Theorem 3, we can use the same setting and simulate the CCA decryption interaction in a similar way. The game $\mathsf{G}_1, \mathsf{G}_2$ keep unchanged. The game G_3 changes a little, since the simulation of decryption oracle needs a coding-theoretic property to have the plaintext-ciphertext verification, where the simulator takes advantage of the $\mathsf{sk}_j^{1-\mathsf{vk}_j^*}$ to recover not only message but also randomness: $x_j = x_{j_m} \circ x_{j_r}$, where both of x_{j_m} and x_{j_r} are uniformly distributed. Thanks to the bounded distance decoding of linear error-correcting codes, simulator can use $x_j = x_{j_m} \circ x_{j_r}$ to re-encrypt with all public keys $\{\mathsf{pk}_i\}$. If ciphertexts are valid, which must have $\forall i$, $\mathsf{Hw}(x_i G \oplus c_i) \leq t$, where $\mathsf{Hw}(\cdot)$ means the Hamming weight. Otherwise, output $\perp$ and halt. In this way, even without using Blum-Micali-Yao, Goldreich-Levin PRGs, we can still check the validity of the ciphertext and run the simulation. Thus, we get the following as in

Theorem 3,

$$\mathbf{Adv}^{\text{priv-cca}}_{\mathit{\Pi}_s,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\text{priv-cpa}}_{\mathit{\Pi}_k,\mathcal{A}_k}(\lambda) + 2\mathbf{Adv}^{\text{suf-cma}}_{\mathcal{SIG},\mathcal{F}}(\lambda) + \frac{1}{2^{k(\lambda)-1}}$$

In addition, we can see that using $(k-1)$ real IND-CPA secure $\overline{\mathit{\Pi}}$ to replace PRIV-CPA secure $\mathit{\Pi}$ will not decrease the security, because the random coin used in encryption algorithm changes to real randomness. Thus, for $k-1$ instances of $\mathcal{E}$, PRIV adversary cannot get advantage with non-deterministic encryption. According to Lemma 1 and Theorem 2, we can get the following,

$$\mathbf{Adv}^{\text{priv-cpa}}_{\mathit{\Pi}_k,\mathcal{A}_k}(\lambda) \leq 2q_e k(\lambda) \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\overline{\mathit{\Pi}},\mathcal{B}}(\lambda) + 16n(\lambda)v(\lambda) \cdot \mathbf{Adv}^{ow}_{\mathcal{P},\mathcal{J}}(\lambda)$$

Note that the first part of right side has $k(\lambda)$ because the scheme starts from the k IND-CPA secure $\overline{\mathit{\Pi}}$, and the second part does not have $k(\lambda)$ because for $k-1$ IND-CPA encryption, the advantage of distinguisher of pseudorandomness is zero. This finishes the proof. □