Injective Trapdoor Functions are Necessary and Sufficient for CCA2 Secure Public-Key Cryptosystems*

Rui Xue Dengguo Feng

The State Key Laboratory of Information Security
Institute of Software, Chinese Academy of Sciences
rxue@is.iscas.ac.cn feng@is.iscas.ac.cn

Abstract. We make the following contributions in this paper:

- We show that the existence of semantically secure public-key cryptosystems implies the existence of injective one-way trapdoor functions. This resolves one of long-standing open problems in cryptography. Moreover, the black-box way of construction to injective trapdoor functions from any semantically secure cryptosystem disproves a conclusion in [15] at FOCS'01.
- We further show that the injective trapdoor functions constructed are secure correlated products under uniform, repetitional distribution. This shows that the existence of semantically secure public-key cryptosystems implies the existence of CCA2 secure cryptosystems by a result of Rosen and Segev [33]. This settles another intensive investigated long-standing and fundamental open problem in cryptography. It also indicates that the secure correlated products under uniform, repetitional distribution exist if and only if injective trapdoor functions exist. That in turn answers the motivating question in [33].

Combining them with prior results, we achieve a somewhat surprising result: injective trapdoor functions exist if and only if CCA2 secure cryptosystems exist. Considering CCA2 security is the strongest among securities for public-key cryptosystems, this makes security hierarchy for public-key cryptosystems, in the sense of existence, collapses into one level.

The conclusions of this work have many consequences: for example, the trapdoor functions with poly-bounded pre-image size exist if and only if injective trapdoor functions exist; There exists a collection of efficient trapdoor functions from Ajtai-Dwork lattice based cryptosystems, and several others.

1 Introduction

1.1 Two Open Problems in Encryptions.

The introduction to the notion of public-key encryptions [10, 25] brings cryptography into a new era. It made several cryptographic applications possible, such as digital signature, oblivious transfer, key exchange, zero-knowledge proof and multi-party computation. The research in cryptographic community grows into two categories: one that focuses on theoretical constructions; the other on the efficient and practical constructions.

One of the main tasks in the theoretical category is to find the minimal necessary and sufficient assumptions for various cryptographic applications [18]. Another is to make it clear how two cryptographic objects or two secure properties for an object are related. For example, Rompel [32] (see also [23]) proves that one-way functions are necessary and sufficient for signature schemes. And one-way functions also imply pseudorandom generators and pseudorandom functions [20, 17].

This paper theoretically investigates the minimal necessary and sufficient cryptographic primitive for CCA2 secure public-key cryptosystems. It constitutes of two major problems: of first is

* This work is partially supported by the Fund of the China Scholarship Council, by the Grant from NSFC Grant No.60773029 and No. 60873260, and by National Basic Research Program of China （973 Program） under Grant No 2007CB311202.

about the primitive for semantically secure cryptosystems, and the second problem is about the relationship between semantically secure and CCA2 secure cryptosystems.

Loosely speaking, a public-key cryptosystem is semantically secure, with respect to passive adversaries, if ciphertexts do not leak any information about the messages encrypted. A CCA2 secure cryptosystem guarantees semantic security even if (active) adversaries are adaptively granted access to decryption oracles (See Section 2 for a formal definition).

Being aware of some weaknesses of deterministic encryptions, Goldwasser and Micali [19] introduced the notions of probabilistic encryptions and semantic security. Since then, semantic security becomes the least secure requirement to an encryption scheme. It is well known that a deterministic encryption scheme cannot be a semantically secure one. Hence the notion of semantic security strengthens previously existing notions of security.

Injective trapdoor functions are functions that is easy to evaluate for any input, and hard to invert on average. It associates each function with a trapdoor such that possession of trapdoor permits efficient inversion.

It has been shown that injective trapdoor functions imply semantically secure public-key cryptosystems [36, 19]. However the question, despite widespread belief, whether semantically secure public-key cryptosystems imply injective trapdoor functions bothers researchers remains unsolved. Although Gertner, Malkin and Reingold [15] claimed that injective trapdoor functions cannot be constructed in black-box way from semantically secure cryptosystems, one of our results in this paper indicates that this is not the case. In other words, we show the following:

(1) *Semantically secure cryptosystems imply injective one-way trapdoor functions.*

Our construction of injective one-way trapdoor functions from semantically secure cryptosystems is in a black-box way. It resolves a long standing open problem and disproves the claim in [15].

Morover, while semantic security fits for most cryptographic applications, it is not good enough in some scenarios where active adversaries exist, especially in case of malicious insiders. In this case, an adversary might get access to decryption oracle *before* producing her attack. Naor and Yung [26] modeled it with the term "security against chosen ciphertext attack"(CCA1), and constructed the first CCA1 secure cryptosystem based on semantically secure encryptions using non-interactive zero-knowledge (NIZK) proofs.

Later, Rackoff and Simon [29] and independently Dolev, Dwork and Naor [11] enhanced CCA1 into so called adaptive chosen ciphertext attack (CCA2), which in addition allows an adversary to get access decryption oracle *even after* the attacker knows the ciphertext she wishes to break (see Definition 3 for a formal definition). These kinds of attacks are practical in real-life scenarios (cf. for example, [5, 35]).

The CCA2 security currently has become *de facto* notion of security for public-key encryption schemes since then. Apart from the construction in [11], Sahai [34] enhanced the NIZK proofs used by Naor and Yung, and gained constructions of CCA2 secure cryptosystem from semantically secure cryptosystems under the enhanced NIZK proofs. The prototype of constructions in these papers is similar in that they encrypt the same message by many (at least two) public keys and then prove consistency of the resulting ciphertexts by NIZK.

The constructions above are based on the existence of enhanced permutations [24]. A natural question that has been significantly investigated in cryptography is: can CCA2 secure cryptosystems be constructed from semantically secure ones?

In practical construction, Cramer and Shoup [9] have cooked and proved the first CCA2 secure scheme based on the ElGamal encryption under the assumption of decisional Diffie-Hellman (DDH) problem. This to some extent builds some confidence in formulating CCA2 security based on semantic security. However, despite so many years intense investigations, this problem still remains an open problem. In this paper we settle this problem positively. That is, we show that

(2) *CCA2 secure cryptosystems exist if semantically secure cryptosystems exist.*

This resolves a fundamental problem in cryptography. Together with claim (1) and previously existing results, we obtain the conclusion that injective trapdoor functions are necessary and sufficient for CCA2 secure cryptosystems. This answers a major question in cryptography.

1.2 Previous Efforts and Related Work.

Since the invention of public-key encryption, the notion of injective trapdoor functions has been viewed as the synonymous to secure public-key encryption. Indeed, it is proved in [10] that the existence of secure public-key encryption cryptosystems is equivalent to the existence of injective trapdoor functions for *deterministic* encryptions. Goldwasser and Micali [19], and also Yao [36] have already formalized semantically secure encryptions from injective trapdoor functions at the very beginning of probabilistic encryptions.

However, as mentioned earlier, only randomized encryptions may possess semantic security. In order to construct injective trapdoor functions from a cryptosystems, or rather, to transform a randomized encryption into a injective trapdoor function, how to cope with randomness used in the encryption proves problematic. One issue is that encrypting one message using different random strings might produce the same result. Another obstacle is that the decryption algorithm might not retrieve randomness in ciphertext like ElGamal encryption scheme. Both hints us that the random strings and the messages encrypted should be moderately related if we try to naturally use secret key as trapdoor information.

Bellare, Halevi, Sahai and Vadhan [4] successfully constructed injective trapdoor functions from semantically secure encryptions *with the additional assumption* that truly random functions exist (so called the random oracle).

Their construction is a de-randomization process: the value for an input x is the encrypting to x using random string $G(x)$, where G is a random oracle to which adversaries also have access. The investigation there indicates that the attempt to replace $G(x)$ with random looking functions such as pseudorandom generator may be hard to succeed. They finally point out that: “In fact whether or not semantically secure public-key cryptosystems imply injective trapdoor functions is not only an open question, but seems a hard one.”

Theirs is the only result trying to base injective trapdoor functions on public-key encryptions appeared in the literature. Later, in [15], Gertner, Malkin and Reingold adopted the oracle separation approach invented by Impagliazzo and Rudich in [22] and made the claim that there is no black-box constructions possible from semantically secure encryptions to injective trapdoor functions. This largely limits the success possibility, and (to our knowledge) the investigation in this direction remains fruitless since then. Thus, the problem wether semantic secure public-key cryptosystems imply injective trapdoor functions still remains open.

The other problem we are concerned with, however, gains much attentions. Since the appearance of the notion of CCA2 security, the relationship between semantically and CCA2 security has been a major question.

As described above, Dolev, Dwork and Naor [11] and Sahai [34] followed the paradigm of multi-encrypting a message followed by NIZK proofs of consistency. Cramer and Shoup constructed the first practical scheme based on DDH assumption following the same paradigm. Elkind and Sahai [12] made a survey and formalization for the two keys paradigm.

The main issue in this paradigm is the way the proofs of consistency or "proofs of well-formness" of ciphertexts are constructed. It seems that constructing efficient NIZK proof for NP is not an easy task on a weaker assumption (e.g. assumption of injective trapdoor functions). Hence the forthcoming works in this line consider substituting NIZK proofs with other tools.

Pass, Shelat and Vaikuntanathan [27] replaced NIZK proofs with designated verifier proofs in the same paradigm and yielded non-malleable encryption schemes against chosen plaintext attack from semantically secure one. Choi, Dachman-Soled, Malkin and Wee [7] used encoding and decoding techniques to check the consistency of ciphertexts, instead of zero knowledge proofs. Cramer, Hanaoka, Hofheinz, Imai, Kiltz, Pass, Shelat and Vaikuntanathan [8] extended this technique construct a weaker CCA2 scheme, in which only a bounded number of queries are permitted. Unfortunately, the results in all these papers yield only schemes with weaker securities than CCA2 security, though stronger than semantic security.

Having noticed the hardness of basing injective functions on semantically secure public-key encryptions, Peikert and Waters [28] proposed a new and powerful cryptographic primitive named lossy trapdoor functions and proved that it implies injective trapdoor functions, semantically secure encryptions as well as CCA2 secure encryptions.

Rosen and Segev [33] recently proposed another primitive called secure correlated products. They showed that CCA2 secure encryptions can be constructed from secure correlated products under a natural distribution. They also show that correlated security is potentially weaker than lossy trapdoor functions.

A different approach was suggested by Boneh, Canetti, Halevi and Katz [6] who constructed a CCA2 secure public-key encryption scheme based on any identity-based encryption scheme. Their construction is out of the "proofs of well-formedness" prototype. How their construction in general relates with semantically secure encryptions is currently not clear.

Gertner, Malkin and Myers [14], again adopting the approaches by Impagliazzo and Rudich in [22], showed that there are no black-box constructions in which the decryption algorithm of the proposed CCA2 secure scheme does not query the encryption algorithm of the semantically secure one.

1.3 Our Resolutions and Contributions.

A natural way to base injective functions on encryption schemes will include a de-randomization process. This leads to dilemma due to seemingly contradictory requirements. On one side, we hope to make use of valid encryption(s) in the definition of injective trapdoor functions. For this purpose, the plaintext should be independent of the randomness used in the encrypting. On the other side, the randomness shall have some relation to the encrypted messages so that they can be retrieved with trapdoor information (usually the secret keys). This is because there is no guarantee that decryption algorithm in a cryptosystem will retrieve randomness in the ciphertext. ElGamal encryption scheme is one example. To make the functions injective, the randomness should be uniquely determined by the input to a function.

Bellare, Halevi, Sahai and Vadhan exploited their construction in [4] by encrypting input using randomness obtained from random oracle. The randomness is completely determined by the input.

Additionally they pointed out the difficulty of substituting random oracle there with pseudorandom generator.

We attack this problem with a new idea that do not try to encrypt an input to a function with some randomness as did in previous construction in [4]. Instead, we use the input directly as the randomness to encrypt some prescribed (randomly chosen) messages. Although a ciphertext may (or may not) leak some bits in the input, it will not leak all of them. Our goal is injective one-way (trapdoor) functions rather than perfectly hiding ones.

This alone will not get out of the dilemma. The key point is the way how to set the message to be encrypted. Recall that only this message will be *definitely* retrieved with trapdoor, and further, it should be independent of the random string (now the input) to make the encryption valid. The message should also be related to the input in such a way so that with a trapdoor one can precisely retrieve the input.

Our resolution to this seemingly paradox is to take two steps. First, in evaluation (i.e. encrypting) part, the randomness and messages in ciphertext are set independently such that even if it is decrypted with secret key, one cannot reveal any information about the random string from the plaintext alone. Second, to provide in explicit form with additional side information so that it allows precisely relating plaintext to the randomness. This helps uniquely deciding and retrieving the input. The side information is also chosen independent of any input. That is, the relation between randomness and the messages is described out of the encryption content.

In this way, we are able to solvethe contradictory problem. The semantically secure encryption completely hides the plaintext and keeps randomness hidden (though may not be perfectly hidden). Moreover, the ciphertext together with side information will not compromise the input to the function.

The way we implement the construction is to use a signal message to each bit of input: two messages are randomly chosen so that one of them indicating 1, the other 0. For any input of k bits, we use k pair of independently chosen messages to signal them. The value of a function at an input is to encrypt these ordered signal messages with the input as randomness. The k pairs of messages as a whole is the side information.

The realization of the construction is down so that it needs to encrypt many messages using the same randomness. Therefore, a random reusable multi-messages encryption scheme is naturally desired. Fortunately, this has been already exploited in the literature. Bellare, Boldyreva, Kurosawa and Staddon [3] recently gave out a clear exploration to randomness reusable multi-messages encryption. Loosely speaking, a randomness reusable multi-messages encryption is a public-key cryptosystem such that one may simultaneously encrypt a set of messages using the same randomness and still keep the encryption secure (as hard as with original scheme).

More formally, let $\mathit{\Pi} = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a randomness reusable multi-messages cryptosystem. Let $\ell(n)$ be a polynomial in n. Let $\mathit{\Pi}^\ell = (\mathtt{KGen}', \mathtt{Enc}', \mathtt{Dec}')$ be a new encryption system with messages space $\{0,1\}^{\ell\times k}$, where $\{0,1\}^k$ is the messages space for $\mathit{\Pi}$. The key generation algorithm $\mathtt{KGen}'$ will generate ℓ-pairs of keys (pk_i, sk_i) and take $\boldsymbol{pk} = (pk_1, \ldots, pk_\ell)$ and $\boldsymbol{sk} = (sk_1, \ldots, sk_\ell)$ as public and secret key, respectively. For any $m = m_1 \cdots m_\ell \in \{0,1\}^{\ell\times k}$, where $m_i \in \{0,1\}^k$, its ciphertext is

$$\mathtt{Enc}'_{\boldsymbol{pk}}(m, r) := (\mathtt{Enc}_{pk_1}(m_1, r), \ldots, \mathtt{Enc}_{pk_\ell}(m_\ell, r))$$

For any randomness reusable multi-messages encryption $\mathit{\Pi}$, if $\mathit{\Pi}$ is semantically secure, so is $\mathit{\Pi}^\ell$ for any polynomial ℓ.

Let $\mathit{\Pi}$ be a randomness reusable encryption system. The sampling function for trapdoor functions chooses $2k$ messages $\boldsymbol{m} = (m_{10}, m_{11}, \ldots, m_{k0}, m_{k1})$ which form k pairs, and it invokes the key generation algorithm $\mathtt{KGen}'$ of $\mathit{\Pi}^k$ to generate $\boldsymbol{pk}, \boldsymbol{sk}$, where $\boldsymbol{pk} = (pk_1, \ldots, pk_k)$, $\boldsymbol{sk} = (sk_1, \ldots, sk_k)$. The $\boldsymbol{m}, \boldsymbol{pk}$ as a whole is an index for a function $F_{\boldsymbol{m},\boldsymbol{pk}}$ and $\boldsymbol{sk}$ the trapdoor.

At any input $x = x_1 \cdots x_k \in \{0,1\}^k$, the value of $F_{\boldsymbol{m},\boldsymbol{pk}}$ is just the ciphertext $\mathtt{Enc}'_{\boldsymbol{pk}}(m, x)$, where $m = m_1 \cdots m_k \in \{0,1\}^{k \times k}$. Here m_i is m_{i0} iff x_i is 0, otherwise m_{i1}.

It is evident that the message m is independent of x in the "content". The only relation between input x and m is that the ith block of m is chosen from $\{m_{i0}, m_{i1}\}$ according to the ith bit in x. Hence the value $F_{\boldsymbol{m},\boldsymbol{pk}}(x)$ is a valid cipher under $\mathit{\Pi}^k$. Considering the semantic security, the security of the functions is not hard to see.

Bellare, Boldyreva, Kurosawa and Staddon [3] showed that if there is a semantically secure cryptosystem, there is also a semantically secure randomness reusable multi-messages cryptosystem. Combining the description from above, means that semantically secure cryptosystems implies injective one-way trapdoor functions.

This inference can be taken a step further. With careful investigation, we can show that the trapdoor functions defined as above are also correlated secure under a natural distribution on input domain. That leads us to the conclusion that semantically secure cryptosystems imply CCA2 secure cryptosystems by the result of [33] mentioned earlier.

A collection of injective trapdoor functions $\mathcal{F} = (G, F)$ is correlated products secure with respect to some distribution $\mathcal{C}_\ell$ over domains if ℓ Cartesian products is one-way with respect to input drawing according to $\mathcal{C}_\ell$. Specifically, For independently generated indexes $s_1, \ldots, s_\ell \leftarrow G(1^n)$, and $(x_1, \ldots, x_\ell) \leftarrow \mathcal{C}_\ell$, it is easy to evaluate $(F_{s_1}(x_1), \ldots, F_{s_\ell}(x_\ell))$ and hard to invert. The uniformly ℓ-repetitional distribution is the distribution that randomly samples x from the domain and repeats this ℓ times.

The trapdoor functions described above happens to be correlated secure under uniform, repetitional distribution. In fact, the ℓ products of our injective trapdoor functions is just ℓk-encryption and hence one-way.

To summarize, we make the following contributions in this paper:

- We show that the existence of semantically secure public-key cryptosystems implies the existence of injective one-way trapdoor functions. Additionally, since our construction of trapdoor functions is in a black-box way, it therefore disproves the claim in [15].
- We show further that the injective trapdoor functions constructed is correlated secure under uniform, repetitional distribution. This shows the existence of semantically secure public-key cryptosystems implies the existence of CCA2 secure cryptosystems. Tis result resolves another long-standing and fundamental open question in cryptography.

Combining these results together, we get the equivalence, in the sense of existence, between following three important objects in cryptography: injective trapdoor functions, semantically cryptosystems and CCA2 secure cryptosystems.

Our conclusions have many consequences: for example, Bellare, Halevi, Sahai and Staddon [4] proved that trapdoor functions with poly-bounded pre-image size imply semantically secure cryptosystems. It therefore holds that trapdoor functions with poly-bounded pre-image size imply injective trapdoor functions, a result proved previously only to be hold in random oracle model in [4].

Another example is that we can make the claim that there is a CCA2 secure cryptosystems from any lattice based encryption scheme. This includes the scheme by Ajtai and Dwork [2, 1] (not known before) as well as those by Regev [30, 31] (proved recently by Peikert and Waters [28, 13].

More importantly, this paper makes the hierarchical relations between security notions collapse into one level in the sense of existence. More explicitly, this is because the existence of CCA2 secure cryptosystems implies the existence of cryptosystems of various securities such as bounded CCA2 security, non-malleable security and so on.

The novelty of the construction may be of independent interest, and further applications are expected in cryptography.

1.4 The Organization of the Paper

In the next section we present some notions and notations. In Section 3, we present two properties of semantically secure encryptions, that will be used in the subsequent sections. In Section 4, we first provide an extension, with respect to pseudorandom generators, from any semantically secure encryption scheme into a random reusable multi-message encryption scheme. The encryption trapdoor functions are then defined based on the extended scheme.

In order to achieve the security of correlated products for our encryption trapdoor functions, in Section 5, we redefine the encryption trapdoor functions and base them on the extension with respect to pseudorandom functions proposed in [3]. It allows us to prove our main conclusion with great convenience. We make our conclusions and discussions in last section.

2 Preliminary and Notations

We denote by $n \in \mathbb{Z}$ the security parameter in this paper. For any $0 \leq \ell \in \mathbb{Z}$, we denote by $[\ell]$ the set $\{1, \ldots, \ell\}$. For a finite set S, we denote by $x \leftarrow S$ the experiment of choosing an element of S uniformly at random and $x_1, \ldots, x_k \overset{k}{\leftarrow} S$ the experiment of choosing k elements independently uniform from S. Similarly, for an algorithm $\mathcal{A}$, we denote by $x_1, \ldots, x_k \overset{k}{\leftarrow} \mathcal{A}(\cdot)$ the experiment that independently invokes $\mathcal{A}(\cdot)$ with input for k times and output x_i independently. We often write $x = x_1 x_2 \cdots x_k \in S$ to mean a string x in S such that $|x_1| = \cdots = |x_k|$. For example, $x = x_1 \cdots x_k \in \{0,1\}^k$ is a string of length k with the i-th bit x_i. A function $\mu(n)$ of n is negligible if for any $c > 0$, for sufficiently large n, it holds $\mu(n) < 1/n^c$. We denote by $\{0,1\}^{l \times k}$ the l times products of $\{0,1\}^k$. That is, $\underbrace{\{0,1\}^k \times \{0,1\}^k \cdots \times \{0,1\}^k}_{\ell\text{-times}}$

A collection of functions is said to be one-way if it has an efficient sampling function and for a sampled function f, given x from the domain it is easy to compute $f(x)$, and it is hard to find a pre-image of $f(x)$ with respect to the uniformly chosen x over the domain. The one-way trapdoor functions are one-way functions that, in addition, generates some trapdoor information by sampling function, with which one is able to efficiently invert functions. Formally

Definition 1 (Trapdoor Functions). *A collection of functions $\mathcal{F} = (G, F)$ satisfying the following conditions is said to be a collection of one-way trapdoor functions.*

`Efficient sampling` *: The probabilistic polynomial time (ppt) algorithm G accepts the security parameter n as input and outputs (s, tp), denoted by $(s, tp) \leftarrow G(1^n)$. Where s is an index of a function, and tp the trapdoor for the function.*

`Efficient evaluation` *: Given $x \in \mathcal{M}$ as input, $F(s, \cdot)$ can be evaluated as $F(s, x)$ in polynomial time.*

`Inverting functions` *: For any randomly sampled function $F(s, \cdot)$ and the value $F(s, x)$ of uniformly chosen $x \leftarrow \mathcal{M}$, it is computationally infeasible to find a pre-image. That is, for every ppt algorithm $\mathcal{A}$, the following probability is negligible.*

$$\mathbf{Pr}\left[s \leftarrow G(1^n), x \leftarrow \mathcal{M} \,:\, F(s, \mathcal{A}(1^n, s, F(s, x))) = F(s, x)\right]$$

However, with trapdoor information tp, it is easy to compute a pre-image of $F(s, tp)$. That is, there is a ppt algorithm $\mathcal{A}$ such that

$$\mathbf{Pr}\left[s \leftarrow G(1^n), x \leftarrow \mathcal{M} \,:\, F(s, \mathcal{A}(s, tp, F(s, x))) = F(s, x)\right] = 1 - \mu(n)$$

Where $\mu(n)$ is a negligible function.

We formally introduce the public-key cryptosystem as follows. Please note that we explicitly present the random string in encryption algorithm `Enc` so as to make it convenient for discussions.

Definition 2 (Public-Key Encryption System). *A public-key cryptosystem Π consists of three algorithms* (`KGen`, `Enc`, `Dec`) *such that*

1. *Probabilistic algorithm* `KGen`$(\cdot)$ *accepting security parameter n outputs a pair of keys (pk, sk). Where pk is a public key and sk the corresponding secret key.*
2. *The deterministic algorithm* `Enc`$(\cdot, \cdot)$ *takes a public key pk, a message $m \in \mathcal{M}$ and a random string $r \in \mathcal{R}$ and outputs $c = \mathtt{Enc}_{pk}(m, r)$ as the ciphertext. Where $\mathcal{M}$ is the messages space and $\mathcal{R}$ the random strings space.*
3. *The deterministic algorithm* `Dec`$(\cdot)$ *takes secret key sk and a ciphertext c as input and outputs $m = \mathtt{Dec}_{sk}(c)$.*

The *correctness* of a public-key cryptosystem requires that for any $m \in \mathcal{M}$, any $(pk, sk) \leftarrow \mathtt{KGen}(1^n)$, $r \leftarrow \mathcal{R}$, it holds that $\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(m, r)) = m$.

Definition 3 (Security of Cryptosystems). *Let* (`KGen`, `Enc`, `Dec`) *be an encryption scheme. For any ppt algorithm $\mathcal{A}$ if the following is negligible:*

$$\Pr\left[\begin{matrix}(pk, sk) \leftarrow \mathtt{KGen}(1^n), (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{D}_{sk}(\cdot)} \\ b \leftarrow \{0,1\}; r \leftarrow \{0,1\}^{k'} c^* \leftarrow \mathtt{Enc}_{pk}(m_b, r)\end{matrix} \,:\, A^{\mathcal{D}_{sk}(c^*, \cdot)}(pk, c^*) = b\right] - \frac{1}{2}$$

where $\mathcal{D}$ and $\mathcal{D}_{c^*}$ are two decryption oracles. And

1. *if both $\mathcal{D}_{sk}(\cdot)$ and $\mathcal{D}_{sk}(c^*, \cdot)$ are empty oracles, then the scheme is said secure against* chosen plaintext attack (CPA)*, or* semantically secure*. or*
2. *if oracle $\mathcal{D}_{sk}(\cdot)$ responds exactly as a decryption algorithm $\mathtt{Dec}_{sk}(\cdot)$ and $\mathcal{D}_{sk}(c^*, \cdot)$ is empty, then the scheme is said secure against* chosen ciphertext attack (CCA1)*. or*
3. *if both oracles $\mathcal{D}_{sk}(\cdot)$ and $\mathcal{D}_{sk}(c^*, \cdot)$ responds exactly as a decryption algorithm $\mathtt{Dec}_{sk}(\cdot)$, except the latter will output $\perp$ when c^* is queried, then the scheme is said secure against* adaptive chosen ciphertext attack (CCA2).

We will use notions of pseudorandom generators and pseudorandom functions in our constructions. The following definitions are cited from [16].

Definition 4 (Pseudorandom Generator). *A pseudorandom generator is a deterministic polynomial-time algorithm G satisfying the following two conditions:*

`Expansion`: *there exists a function $\ell : \mathbb{N} \to \mathbb{N}$ such that $\ell(n) > n$ for all $n \in \mathbb{N}$, and $|G(s)| = \ell(|s|)$ for all $s \in \{0,1\}^*$.*

`Pseudorandomness`: *the ensembles $\{G(U_m)\}_{m\in\mathbb{N}}$ and $\{U_{\ell(m)}\}_{m\in\mathbb{N}}$ are computationally indistinguishable in polynomial time. Where $\{U_{\ell(m)}\}_{m\in\mathbb{N}}$ is the uniform ensemble. That is, for any polynomial $p(\cdot)$ and polynomial time algorithm $\mathcal{D}$, for all sufficiently large $n \in \mathbb{N}$,*

$$|\mathbf{Pr}[\mathcal{D}(G(U_n), 1^n) = 1] - \mathbf{Pr}[\mathcal{D}(U_{\ell(n)}, 1^n) = 1]| < \frac{1}{p(n)}$$

Definition 5 (Pseudorandom Functions). *Let $r : \mathbb{N} \to \mathbb{N}$. The efficiently computable function ensemble*

$$\{f_s : \{0,1\}^* \to \{0,1\}^{r(|s|)}\}_{s\in\{0,1\}^*}$$

is a pseudorandom function ensemble *if following conditions hold:*

`Efficient Evaluation`: *there exists a polynomial-time algorithm that on input s and $x \in \{0,1\}^*$ returns $f_s(x)$.*

`Pseudorandomness`*: for every ppt oracle machine $\mathcal{D}$, Every polynomial $p(\cdot)$, and all sufficiently large n,*

$$|\mathbf{Pr}[\mathcal{D}^{F_n}(1^n) = 1] - \mathbf{Pr}[\mathcal{D}^{H_n}(1^n) = 1]| < \frac{1}{p(n)}$$

where F_n is a random variable uniformly distributed over the multi-set $\{f_s\}_{s\in\{0,1\}^n}$, and H_n is uniformly distributed among all functions mapping arbitrary long strings to $r(n)$-bit-long strings.

3 Properties for Semantically Secure Public-Key Encryptions

We use $\varPi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ to denote a public-key encryption system in this paper. The message space is $\{0,1\}^k$ and the random string space is $\{0,1\}^{k'}$. That is, the deterministic encryption algorithm $\mathtt{Enc}_{pk}(\cdot,\cdot)$ takes a message of length k and a random string of length k' as input.

Let's consider the possibility to extract a bit of the plaintext from a ciphertext. Let $\mathcal{P}_i$ be a ppt algorithm that tries to extract the i-th bit of plaintext.

Lemma 1. *Let $\varPi$ be a semantically secure cryptosystem, then for any $i \in [k]$ and any ppt bit extractor $\mathcal{P}_i$, the success probability of extracting the i-th bit of plaintex from ciphertext is negligibly better than guess. That is,*

$$\mathrm{Adv}_{\varPi}^{\mathrm{extract}}(\mathcal{P}_i) = \mathbf{Pr}\left[\begin{matrix}(pk, sk) \leftarrow \mathtt{KGen}(1^n); m \leftarrow \{0,1\}^k, r \leftarrow \{0,1\}^{k'} \\ c = \mathtt{Enc}_{pk}(m, r)\,; b' \leftarrow \mathcal{P}_i(1^n, pk, c)\end{matrix} : b' = b\right] - \frac{1}{2} \qquad (1)$$

is a negligible function in n. Where b is the i-th bit in m. The probability in (1) *is taken over the choice of keys, messages and coin tossing in $\mathcal{P}_i$.*

Proof. For any $i \in [k]$ and any ppt $\mathcal{P}_i$, denote $\varepsilon := \mathrm{Adv}_{\varPi}^{\mathrm{extract}}(\mathcal{P}_i)$. We construct the following IND-CPA attacker $\mathcal{A}$ to invoke $\mathcal{P}_i$ as a subroutine as depicted in Fig. 1.

```
       $\mathcal{O} : (pk, sk) \leftarrow G(1^n);$
  $\mathcal{A}(1^n, pk) : m_0, m_1 \leftarrow \{0,1\}^k;$
                 If the $i$-th bit of $m_0$ equals to the $i$-th bit of $m_1$ then choose again
       $\mathcal{O} : b \leftarrow \{0,1\}, r \leftarrow \{0,1\}^{k'}, c = \texttt{Enc}(pk, m_b, r);$
$\mathcal{A}(1^n, pk, c) : pk, c \rightarrow \mathcal{P}_i(1^n);$
                 $b' \leftarrow \mathcal{P}_i(1^n, pk, c)$
  output $b'$
```

Fig. 1. The IND-CPA attack using a bit extractor as a subroutine.

Whenever $\mathcal{A}$ receives the public key pk generated by challenger $\mathcal{O}$ using KGen, it chooses two messages m_0, m_1 from $\{0,1\}^k$ uniformly at random until the i-th bit in m_0 and m_1 differs. $\mathcal{A}$ submits them to $\mathcal{O}$. The latter then chooses $b \in \{0,1\}$ uniformly at random, and to encrypt m_b with a random string $r \leftarrow \{0,1\}^k$. The ciphertext $c = \texttt{Enc}_{pk}(m_b, r)$ is given to $\mathcal{A}$. Adversary $\mathcal{A}$ in turn invokes $\mathcal{P}_i$ with c, pk. $\mathcal{A}$ outputs the bit that $\mathcal{P}_i$ outputs.

The event success occurs when $b = b'$ in the game. It is easy to see that the success probability of $\mathcal{A}$ is no less than the success probability of $\mathcal{P}_i$ extracting the i-th bit for plaintext m_b. We obtain

$$\begin{aligned}\mathrm{Adv}_{\Pi}^{\mathrm{cpa}} &= \mathbf{Pr}[\texttt{success}] - \frac{1}{2} \\ &\geq \mathbf{Pr}\left[\begin{array}{c}(pk, sk) \leftarrow \texttt{KGen}(1^n); m \leftarrow \{0,1\}^k, r \leftarrow \{0,1\}^{k'} \\ c = \texttt{Enc}_{pk}(m, r)\,; b' \leftarrow \mathcal{P}_i(1^n, pk, c)\end{array} : b' = b\right] - \frac{1}{2} = \varepsilon \end{aligned} \tag{2}$$

The semantic security of Π implies that $\mathrm{Adv}_{\Pi}^{\mathrm{cpa}}$ is a negligible function in n. Hence ε is negligible. This ends the proof. □

Lemma 2. *For any semantically secure cryptosystem Π, it is infeasible for any adversary to extract random string used from a ciphertext. That is, the following probability λ is a negligible function in n for any ppt algorithm $\mathcal{A}$.*

$$\lambda := \mathbf{Pr}\left[\begin{array}{c}(pk, sk) \leftarrow \texttt{KGen}(1^n); m \leftarrow \{0,1\}^k, r \leftarrow \{0,1\}^{k'} \\ c = \texttt{Enc}_{pk}(m, r)\,; s \leftarrow \mathcal{A}(1^n, pk, c)\end{array} :\ s = r\right] \tag{3}$$

Proof. To show that λ is a negligible function in security parameter n, we construct a ppt algorithm $\mathcal{B}$ producing IND-CPA against Π, which includes $\mathcal{A}$ as a subroutine.

The challenger $\mathcal{O}$ invokes KGen with 1^n to obtain (pk, sk). After receiving public key pk , attacker $\mathcal{B}$ chooses $m_0, m_1 \in \{0,1\}^k$ uniformly at random and sends them to $\mathcal{O}$. The challenger $\mathcal{O}$ will choose, respectively, a bit $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^k$ uniformly at random. It further encrypts m_b using string r to get $c = \texttt{Enc}_{pk}(m_b, r)$ and delivers it to $\mathcal{B}$. Algorithm $\mathcal{B}$ invokes $\mathcal{A}$ with c to get s as feedback.

Algorithm $\mathcal{B}$ then checks if $c = \texttt{Enc}_{pk}(m_0, s)$ then let $b' = 0$; If $c = \texttt{Enc}_{pk}(m_1, s)$ then let $b' = 1$. Otherwise to choose $b' \leftarrow \{0,1\}$ uniformly at random. It finally outputs b' as the guess to b. This procedure is depicted in Fig. 2

It is easy to see from the construction of $\mathcal{B}$ that if $\mathcal{A}$ outputs the correct random string (i.e. $s = r$), $\mathcal{B}$ outputs b' satisfying $b' = b$. Otherwise it outputs b' such that $b = b'$ with probability $\frac{1}{2}$.

Let $\texttt{success}_{\mathcal{B}}$ denote the event that $\mathcal{B}$ returns b' such that $b' = b$. Let $\texttt{success}_{\mathcal{A}}$ denote the event that $\mathcal{A}$ extracts the correct random string from a ciphertext, which occurs with probability λ as denoted in (3).

```
             $\mathcal{O}$ : $(pk, sk) \leftarrow G(1^n)$;
    $\mathcal{B}(1^n, pk)$ : $m_0, m_1 \leftarrow \{0,1\}^k$;
             $\mathcal{O}$ : $b \leftarrow \{0,1\}, r \leftarrow \{0,1\}^{k'}, c = \texttt{Enc}(pk, m_b, r)$;
 $\mathcal{B}(1^n, pk, c)$ : $pk, c \rightarrow \mathcal{A}(1^n)$;
                             $s \leftarrow \mathcal{A}(1^n, pk, c) \in \{0,1\}^{k'}$
                             If $c = \texttt{Enc}_{pk}(m_0, s)$ then $b' = 0$
                             elseif $c = \texttt{Enc}_{pk}(m_1, s)$ then $b' = 1$
                             else $b' \leftarrow \{0,1\}$
        output $b'$
```

Fig. 2. The IND-CPA attack against Π using a random string extractor as a subroutine.

From the analysis above, we have

$$\begin{aligned}\mathbf{Pr}\left[\texttt{success}_{\mathcal{B}}\right] &= \mathbf{Pr}\left[\texttt{success}_{\mathcal{B}} \wedge \texttt{success}_{\mathcal{A}}\right] + \mathbf{Pr}\left[\texttt{success}_{\mathcal{B}} \wedge \neg\texttt{success}_{\mathcal{A}}\right] \\ &= \lambda + \frac{1}{2} \end{aligned} \tag{4}$$

This implies $\lambda = \mathbf{Pr}\left[\texttt{success}_{\mathcal{B}}\right] - \frac{1}{2} = \mathrm{Adv}_{\Pi}^{\mathrm{cpa}}(\mathcal{B})$. Which is negligible from the sematic security of cryptosystem Π . □

4 Encryption Trapdoor Functions

We will present two constructions of injective one-way trapdoor functions family based on semantically secure cryptosysterms. The first construction (presented in this section) is based on semantically secure cryptosystems by employing a pseudorandom generator. The second construction (presented in Section 5) is similar to the first one, but it employs a collection of pseudorandom functions instead. Though the second lacks simplicity in the presentation, it allows us to show the correlated products security under $\mathcal{C}_\ell$ (consult [33] for detailed definiton). Where $\mathcal{C}_\ell$ (defined in following context) is the uniform ℓ-repetition distribution on $\{0,1\}^{\ell \times k}$. That will lead us to a construction of CCA2 secure encryption scheme by the result of [33].

Since the second approach adopted is essentially what was used in [3] to transform any semantically secure encryption scheme into a randomness reusable multi-messages encryption scheme， we will cite the conclusion there directly.

4.1 Extensions of Semantically Secure Cryptosystems

In this subsection we first make an extension of a cryptosystem into a cryptosystem that allows encrypting multiple messages simultaneously with the same randomness. We then present the construction of encryption trapdoor functions and prove its security.

The extension is essentially to expand the length of messages encrypted. In other words, we construct a new cryptosystem from a given one, such that the length of encrypted messages is extended polynomial times and the random string space, however, remains the same as the original one. The extension preserves semantic security.

Let $\Pi = (\texttt{KGen}, \texttt{Enc}, \texttt{Dec})$ be a public-key encryption scheme. Let $k(n), k'(n)$ be two polynomials. The messages space is $\{0,1\}^k$ and the randomness space is $\{0,1\}^{k'}$. Let $\ell(n)$ be an integral

$\mathbf{Game_0}$: $(pk, sk) \leftarrow \texttt{KGen}'(1^n)$;
$pk = (pk_1, pk_2)$, $sk = (sk_1, sk_2)$;
$m_i = m_i^1 m_i^2 \leftarrow \mathcal{A}(1^n, pk) \in \{0,1\}^{2k}$, $i = 0,1$
$b \leftarrow \{0,1\}$, $r \leftarrow \{0,1\}^k$, $r_1 r_2 = \mathrm{PRG}(r) \in \{0,1\}^{2k'}$
$c = (\texttt{Enc}_{pk_1}(m_b^1, r_1), \texttt{Enc}_{pk_2}(m_b^2, r_2))$
$b' \rightarrow \mathcal{A}(1^n, pk, c)$

$\mathbf{Game_1}$: $(pk, sk) \leftarrow \texttt{KGen}'(1^n)$;
$pk = (pk_1, pk_2)$, $sk = (sk_1, sk_2)$;
$m_i = m_i^1 m_i^2 \leftarrow \mathcal{A}(1^n, pk) \in \{0,1\}^{2k}$, $i = 0,1$
$b \leftarrow \{0,1\}$, $\boxed{r_1, r_2 \leftarrow \{0,1\}^{k'}}$
$c = (\texttt{Enc}_{pk_1}(m_b^1, r_1), \texttt{Enc}_{pk_2}(m_b^2, r_2))$
$b' \rightarrow \mathcal{A}(1^n, pk, c)$

Fig. 3. The IND-CPA attacking games against Π^2_{PRG}.

polynomial in n, the ℓ-extension of a cryptosystem with respect to a pseudorandom generator is defined as follows.

Definition 6 (ℓ-Extension of Cryptosystems). *Let $\Pi = (\texttt{KGen}, \texttt{Enc}, \texttt{Dec})$ be a public-key encryption scheme with parameters as above. Let* PRG *be a pseudorandom generator with domain $\{0,1\}^k$ and range $\{0,1\}^{\ell k'}$. To define a new encryption scheme $\Pi^\ell_{\mathrm{PRG}} = (\texttt{KGen}^\ell, \texttt{Enc}^\ell, \texttt{Dec}^\ell)$ as follows:*

`Key Generation` $\texttt{KGen}^\ell(\cdot)$: *on input security parameter 1^n, it invokes* `KGen` *for ℓ times gaining key pair $(pk_i, sk_i) \leftarrow \texttt{KGen}(1^n)$ for $i = 1, \ldots, \ell$. Let public key be $\boldsymbol{pk} = (pk_1, \ldots, pk_\ell)$ and secret key $\boldsymbol{sk} = (sk_1, \ldots, sk_\ell)$.*

`Encryption` $\texttt{Enc}^\ell_{\boldsymbol{pk}}(\cdot,\cdot)$: *on input $m = m_1 \cdots m_\ell \in \{0,1\}^{\ell k}$ as input, algorithm $\texttt{Enc}^\ell_{\boldsymbol{pk}}(\cdot,\cdot)$ chooses $r \leftarrow \{0,1\}^k$ computes $\mathrm{PRG}(r) = r_1 \cdots r_\ell$ and $c_i = \texttt{Enc}_{pk_i}(m_i, r_i)$ for all $i \in [\ell]$. To output $\boldsymbol{c} = (c_1, \ldots, c_\ell)$. That is, $\texttt{Enc}^\ell_{\boldsymbol{pk}}(m, r) = \boldsymbol{c}$. Where $|r_i| = k'$, $|m_i| = k$ for each $i \in [\ell]$.*

`Decryption` $\texttt{Dec}^\ell_{\boldsymbol{sk}}(\cdot)$: *on input $\boldsymbol{c} = (c_1, \ldots, c_\ell)$ and $\boldsymbol{sk}$, it computes $m_i = \texttt{Dec}_{sk_i}(c_i)$ for all $i \in [\ell]$ and outputs $m = m_1 \cdots m_\ell$.*

It is evident that the scheme Π^ℓ_{PRG} in Definition 6 is indeed an encryption scheme. The correctness of encryption relies on the correctness of Π. We proceed to show it preserving semantic security.

Lemma 3 (Semantic Security Preservation). *For any $\ell(n)$ and a semantically secure cryptosystem Π with parameters as above, the encryption scheme Π^ℓ_{PRG} is also a semantically secure cryptosystem.*

Proof. Without lose of generality, we assume that $\ell = 2$ since negligibility is closed under operation "plus" of a (fixed) polynomial times. Suppose $\mathcal{A}$ is a ppt algorithm that produce IND-CPA attack against Π^2_{PRG}. **$\mathbf{Game_0}$** depicts (in Fig. 3) the attacking procedure.

The challenger $\mathcal{O}$ generates a pair of keys $pk = (pk_1, pk_2), sk = (sk_1, sk_2) \leftarrow \texttt{KGen}'(1^n)$. The public key pk is provided to adversary $\mathcal{A}$. Adversary then outputs and delivers to $\mathcal{O}$ a pair of messages m_0, m_1 such that $m_i = m_i^1 m_i^2$ for $i \in \{0,1\}$, where $|m_i^j| = k$ for $i \in \{0,1\}, j \in \{1,2\}$. Challenger $\mathcal{O}$ then chooses $r \leftarrow \{0,1\}^k$ and $b \leftarrow \{0,1\}$ uniformly at random. It encrypts m_b as $c = (\texttt{Enc}_{pk_1}(m_b^1, r_1), \texttt{Enc}_{pk_2}(m_b^2, r_2))$. Where $r_1 r_2 = \mathrm{PRG}(r)$ and $r_1, r_2 \in \{0,1\}^{k'}$. Ciphertext c is provided to $\mathcal{A}$. Upon receiving c, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

$\mathbf{Game_1}$ is the same as **$\mathbf{Game_0}$** but the random strings r_1, r_2 used in encryption are chosen uniformly at random from $\{0,1\}^{k'}$, rather than generated using PRG.

```
$\mathcal{B}(1^n, r)$ : $(pk, sk) \leftarrow \mathtt{KGen}^2(1^n)$;
          $m_0, m_1 \leftarrow \mathcal{A}(1^n, pk) \in \{0,1\}^{2k}$
          $b \leftarrow \{0,1\}$, $c = (\mathtt{Enc}_{pk_1}(m_b^1, r_1), \mathtt{Enc}_{pk_2}(m_b^2, r_2)$
          $d \leftarrow \mathcal{A}(1^n, pk, c)$
          If $d = b$ then output 1
          else output 0
```

Fig. 4. Pseudorandom generator distinguisher.

Let $\mathtt{success_i}$ be the event $b = b'$ in $\mathbf{Game_i}$ for $i = 0, 1$. We will construct a distinguisher $\mathcal{B}$ to tell pseudorandom strings generated by PRG from a random one, such that the success advantage $\mathrm{Adv}^{\mathrm{prg}}_{\mathrm{PRG},\mathcal{B}}$ satisfying

$$\mathrm{Adv}^{\mathrm{prg}}_{\mathrm{PRG},\mathcal{B}} = |\mathbf{Pr}[\mathtt{success_0}] - \mathbf{Pr}[\mathtt{success_1}]| \tag{5}$$

The algorithm $\mathcal{B}$ (depicted in Fig. 4) on input of security parameter 1^n and a string $r_0 r_1 \in \{0,1\}^{2k'}$ invokes $\mathtt{KGen}^2$ to obtain public key $pk = (pk_1, pk_2)$ and $sk = (sk_1, sk_2)$ and delivers pk to $\mathcal{A}$. Adversary returns with a pair of messages $m_0, m_1 \in \{0,1\}^{2k}$. One of uniformly chosen message m_b is then encrypted with $r_0 r_1, pk$. That is $c = (\mathtt{Enc}_{pk_1}(m_b^1, r_1), \mathtt{Enc}_{pk_2}(m_b^2, r_2)$. $\mathcal{A}$ provides with c outputs a bit d . If $d = b$ then outputs 1 else outputs 0 .

It is evident that if the input to $\mathcal{B}$ is a random string from $\{0,1\}^{2k'}$, the computation proceeds just as in $\mathbf{Game_1}$, hence

$$\mathbf{Pr}[r \leftarrow \{0,1\}^{2k'} : \mathcal{B}(1^n, r) = 1] = \mathbf{Pr}[\mathtt{success_1}]$$

If the input to $\mathcal{B}$ is a string from $\{\mathrm{PRG}(x) \mid x \in \{0,1\}^k\}$, the computation proceeds just as in $\mathbf{Game_0}$, therefore

$$\mathbf{Pr}[x \leftarrow \{0,1\}^k,\ r = \mathrm{PRG}(x) : \mathcal{B}(1^n, r) = 1] = \mathbf{Pr}[\mathtt{success_0}]$$

The equation (5) is from the difference of last two equations.

We next to construct an IND-CPA attacker $\mathcal{D}$ against $\mathit{\Pi}$ such that

$$\mathbf{Pr}[\mathtt{success}_{\mathcal{D}}] \geq \mathbf{Pr}[\mathtt{success_1}] \tag{6}$$

Where $\mathtt{success}_{\mathcal{D}}$ is the event success guess of adversary $\mathcal{D}$ in IND-CAP game. This together with (5) will show that

$$\begin{aligned} |\mathbf{Pr}[\mathtt{success_0}] - 1/2| &\leq |\mathbf{Pr}[\mathtt{success_0}] - \mathbf{Pr}[\mathtt{success_1}]| + |\mathbf{Pr}[\mathtt{success_1}] - 1/2| \\ &\leq \mathrm{Adv}^{\mathrm{prg}}_{\mathrm{PRG},\mathcal{B}} + |\mathbf{Pr}[\mathtt{success}_{\mathcal{D}}] - 1/2| \end{aligned} \tag{7}$$

From the definition of pseudorandom generator and the semantic security of $\mathit{\Pi}$, we know that the last two terms in (7) are both negligible functions in n . Which shows that the advantage of any ppt $\mathcal{A}$ against IND-CPA attack to $\mathit{\Pi}^2_{\mathrm{PRG}}$ is negligible and hence the semantic security of $\mathit{\Pi}^2_{\mathrm{PRG}}$ holds.

It remains to construct ppt algorithm $\mathcal{D}$ satisfying (6) . The algorithm $\mathcal{D}$ (depicted in Fig. 5) proceeds as follows: Challenger $\mathcal{O}$ will produce key pair $(pk_1, sk_1) \leftarrow \mathtt{KGen}(1^n)$ and delivers with pk_1 to $\mathcal{D}$. $\mathcal{D}$ then produces $(pk_2, sk_2) \leftarrow \mathtt{KGen}$ further and delivers $pk = (pk_1, pk_2)$ to $\mathcal{A}$ as public

$$\begin{aligned}
\mathcal{O}(1^n) &: (pk_1, sk_1) \leftarrow \mathtt{KGen}(1^n) \\
\mathcal{D}(1^n, pk_1) &: (pk_2, sk_2) \leftarrow \mathtt{KGen}(1^n); \\
&\quad pk = (pk_1, pk_2),\ sk = (*, sk_2); \\
&\quad m_i = m_i^1 m_i^2 \leftarrow \mathcal{A}(1^n, pk) \text{ for } i = 0, 1 \\
\mathcal{O}(m_0^1, m_1^1) &: b \leftarrow \{0,1\},\ r_0 \leftarrow \{0,1\}^{k'},\ c_0 = \mathtt{Enc}_{pk_1}(m_b^1, r_0) \\
\mathcal{D}(c_0) &: \hat{b} \leftarrow \{0,1\},\ r_1 \leftarrow \{0,1\}^{k'} \\
&\quad c_1 = \mathtt{Enc}_{pk_2}(m_{\hat{b}}^2, r_1),\ c = (c_0, c_1) \\
&\quad b' \rightarrow \mathcal{A}(1^n, pk, c) \\
\mathtt{output}\ b'
\end{aligned}$$

Fig. 5. IND-CPA game against Π with $\mathcal{A}$ as a subroutine.

key of Π_{PRG}^2 . After receiving the feedback of two challenge messages $m_0 = m_0^1 m_0^2, m_1 = m_1^1 m_1^2 \in \{0,1\}^{2k}$ from $\mathcal{A}$, algorithm $\mathcal{D}$ outputs m_0^1, m_1^1 as the challenge messages to challenger $\mathcal{O}$. A random message m_b among them is encrypted with pk_1 as $c_0 = \mathtt{Enc}_{pk_1}(m_b, t_0)$ for $r_0 \leftarrow \{0,1\}^{k'}$ and the ciphertext is delivered to $\mathcal{D}$. By chosen $\hat{b} \leftarrow \{0,1\}$ uniformly at random $\mathcal{D}$ computes ciphertext $c_1 = \mathtt{Enc}_{pk_2}(m_{\hat{b}}^2, r_1)$ for $r_1 \leftarrow \{0,1\}^{k'}$. The message (c_0, c_1) is provided to $\mathcal{A}$. $\mathcal{A}$ in turn outputs a bit d . This is also the output of $\mathcal{D}$.

It is evident to see that if $\hat{b} = b$ then the view of $\mathcal{A}$ here is identical to the view in $\mathbf{Game_1}$. Let $\mathtt{success}_{\mathcal{D}}$ denote the event $b = b'$ in the game. Hence

$$\begin{aligned}
\mathbf{Pr}[\mathtt{success}_{\mathcal{D}}] &= \mathbf{Pr}[\mathtt{success}_{\mathcal{D}} \mid \hat{b} = b] + \mathbf{Pr}[\mathtt{success}_{\mathcal{D}} \mid \hat{b} \neq b] \\
&= \mathbf{Pr}[\mathtt{success}_1] + \mathbf{Pr}[\mathtt{success}_{\mathcal{D}} \mid \hat{b} \neq b] \\
&\geq \mathbf{Pr}[\mathtt{success}_1]
\end{aligned}$$

This proves (6), and accomplishes the proof of Lemma 3. □

4.2 Encryption Trapdoor Functions

We will assume as before that the underlying cryptosystem $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ is with message space $\{0,1\}^k$ and the random string space $\{0,1\}^{k'}$, where $k(n), k'(n)$ are polynomials in security parameter n. A pseudo-random generator PRG is used in the construction, where PRG : $\{0,1\}^k \rightarrow \{0,1\}^{kk'}$.

The idea of the construction is: to fix k pairs of random messages uniformly at random from the messages space and k public keys that are randomly chosen, these two parts constitute the index of a function in the collection of functions. The corresponding secret keys as a whole is intended as trapdoor for the function. For any input $x = x_1 \cdots x_k \in \{0,1\}^k$, it produces a sequence of k encryptions like this: the plaintext for the i-th ciphertext is the first message of the ith pair if the ith bit in x is 0, or the second message otherwise. The random strings used in the encryption are generated by a pseudorandom generator with seed x. The result of the k ordered ciphertexts is the value of x , which is just a ciphertext under Π_{PRG}^k .

Here is a notation that will be used in subsequence contexts for conveniences.

Definition 7. *For any $k \in \mathbb{Z}$, let $\boldsymbol{m} = (m_{10}, m_{11}, \ldots, m_{k0}, m_{k1}) \in \{0,1\}^{2k \times k}$. For any $x = x_1 \cdots x_k \in \{0,1\}^k$. Define $\boldsymbol{m}_x := m_{1x_1} m_{2x_2} \cdots m_{kx_k} \in \{0,1\}^{k^2}$. The string $\boldsymbol{m}_x$ is called* the characteristic string of x with respect to $\boldsymbol{m}$.

Using the same notations as in last section, the construction of trapdoor functions is formally presented as follows:

Definition 8 (Encryption Trapdoor Functions, ETF). *Let $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a cryptosystem and* PRG *a pseudo-random generator with parameters as above. A collection of efficient computable functions $\mathcal{F}^{\Pi}_{\mathrm{PRG}} = (G, F)$ is defined as follows:*

`Sampling function` G: *on input security parameter 1^n, algorithm G invokes $\mathtt{KGen}^k(\cdot)$ with 1^n to obtain $\boldsymbol{pk}, \boldsymbol{sk}$, and then chooses messages $m_{ij} \leftarrow \{0,1\}^k$ uniformly at random for $i \in [k], j \in \{0,1\}$. It outputs $(\boldsymbol{m}, \boldsymbol{pk})$ and $\boldsymbol{sk}$. Where*

$$\boldsymbol{m} = (m_{10}, m_{11}, \ldots, m_{k0}, m_{k1}) , \qquad \boldsymbol{pk} = (pk_1, \ldots, pk_k) , \qquad \boldsymbol{sk} = (sk_1, \ldots, sk_k) .$$

The output $(\boldsymbol{m}, \boldsymbol{pk})$ is an index of a function and $\boldsymbol{sk}$ the trapdoor of the function.

`Evaluation of` $F_{\boldsymbol{m},\boldsymbol{pk}}$: *on input $x = x_1 \cdots x_k \in \{0,1\}^k$, it computes and outputs $\mathtt{Enc}^k_{\boldsymbol{pk}}(\boldsymbol{m}_x, x)$. Recall that $\boldsymbol{m}_x = m_{1x_1} \cdots m_{kx_k}$ is the characteristic string of x with respect to $\boldsymbol{m}$ in Definition 7.*

`Inverting` $F_{\boldsymbol{m},\boldsymbol{pk}}(x)$ `with trapdoor` $\boldsymbol{sk}$: *Receiving $\boldsymbol{c}$ and $\boldsymbol{sk}$ as input, it computes $m = \mathtt{Dec}^k_{\boldsymbol{sk}}(\boldsymbol{c})$. If there is some $x \in \{0,1\}^k$ such that $m = \boldsymbol{m}_x$ then output x, otherwise output $\perp$.*

We remark that since $m_{i0} = m_{i1}$ in $\boldsymbol{m}$ holds with only negligible probability for each $i \in [k]$, we assume $m_{i0} \neq m_{i1}$ in subsequent context[1].

It is easy to see that when Π and pseudo-random generator PRG is given, the collection of functions constructed above is indeed well-defined, since PRG is a deterministic algorithm. We show that it is a collection of injective one-way trapdoor functions.

Theorem 1. *Given Π as a semantically secure encryption system and* PRG *a pseudorandom generator with appropreated parameters, then $\mathcal{F}^{\Pi}_{\mathrm{PRG}}$ defined above is a collection of injective one-way trapdoor functions.*

Proof. Trapdoorness: it is almost straightforward to verify that for any tuple $(\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{sk})$ generated by G, the information $\boldsymbol{sk}$ is indeed a trapdoor for function $F_{\boldsymbol{m},\boldsymbol{pk}}$. But it is worthwhile to note that in the inverting part, when $m = \mathtt{Dec}^k_{\boldsymbol{sk}}(\boldsymbol{c})$ is computed, the $x \in \{0,1\}^k$ such that $m = \boldsymbol{m}_x$, if exists, is unique and can be defined as $x = x_1 \cdots x_k$, where $x_i = 0$ iff $m_i = m_{i0}$ and $x_i = 1$ iff $m_i = m_{i1}$ (recall $m_{i0} \neq m_{i1}$) for each $i \in [k]$. The correctness of inverting with trapdoor information relies on the correctness of extended encryption Π^k_{PRG}.

Injectivity: assume for some $x, y \in \{0,1\}^k$, there is a function index $\boldsymbol{m}, \boldsymbol{pk}$ such that $F_{\boldsymbol{m},\boldsymbol{pk}}(x) = \boldsymbol{c}$, $F_{\boldsymbol{m},\boldsymbol{pk}}(y) = \boldsymbol{d}$ and $\boldsymbol{c} = \boldsymbol{d}$. Denote $x = x_1 \cdots x_k$, $y = y_1 \cdots y_k$, $\mathrm{PRG}(x) = r_1 \cdots r_k$ and $\mathrm{PRG}(y) = s_1 \cdots s_k$. Where $x_i, y_i \in \{0,1\}$ and $r_i, s_i \in \{0,1\}^{k'}$ for $i = 1, \ldots, k$.

From the definition of extended encryption, $\boldsymbol{c} = (c_1, \ldots, c_k)$ and $\boldsymbol{d} = (d_1, \ldots, d_k)$, where $c_i = \mathtt{Enc}_{pk_i}(m_{ix_i}, r_i)$ and $d_i = \mathtt{Enc}_{pk_i}(m_{iy_i}, s_i)$. It holds that $c_i = d_i$ from $\boldsymbol{c} = \boldsymbol{d}$. The correctness of encryption implies $m_{ix_i} = m_{iy_i}$ for all $i \in [k]$. It further implies that $x_i = y_i$ for all $i \in [k]$ and hence $x = y$.

One-wayness: for any sampled function $F_{\boldsymbol{m},\boldsymbol{pk}}$, the value on input $x \in \{0,1\}^k$ under this function is $\boldsymbol{c} = (c_1, \ldots, c_k)$, which is just an encryption of k-extension for Π with respect to pseudorandom

[1] In fact, we may explicitly set $m_{i0} \neq m_{i1}$ during the indexes generation. Careful investigating on our approach shows that it is unnecessary to choose $2k$ messages here, rather, two different messages and k repetition of them will do.

generator PRG. Let $\mathcal{A}$ be any ppt algorithm intended as an inverter to $\mathcal{F}^{\Pi}_{\mathrm{PRG}}$. Denote

$$\varepsilon = \mathbf{Pr}\left[\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{sk} \leftarrow G(1^n),\ x \leftarrow \{0,1\}^k\ :\ \mathcal{A}(\boldsymbol{m}, \boldsymbol{pk}, F_{\boldsymbol{m},\boldsymbol{pk}}(x)) = x\right] . \tag{8}$$

From the definition of $F_{\boldsymbol{m},\boldsymbol{pk}}$, we have that

$$\varepsilon = \mathbf{Pr}\left[\begin{array}{c} \boldsymbol{m} = m_{10}, m_{11}, \ldots, m_{k0}, m_{k1} \xleftarrow{2k} \{0,1\}^k \\ (\boldsymbol{pk}, \boldsymbol{sk}) \leftarrow \mathtt{KGen}^k(1^n),\ x \leftarrow \{0,1\}^k,\ \boldsymbol{c} = \mathtt{Enc}^k_{\boldsymbol{pk}}(\boldsymbol{m}_x, x) \end{array} : \mathcal{A}(\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{c}) = x\right] \tag{9}$$

Where $\boldsymbol{pk} = (pk_1, \ldots, pk_k)$ and $\boldsymbol{c} = (c_1, \ldots, c_k)$.

Since messages m_{ij} is chosen uniformly independent at random from $\{0,1\}^k$, and x is a random string, $\boldsymbol{c} = (c_1, \ldots, c_k)$ is a valid ciphertext for a random plaintext. The semantic security of extension scheme implies that the following probability, denoted as ε_1, is with a negligible difference to ε.

$$\mathbf{Pr}\left[\begin{array}{c} \boldsymbol{m} = m_{10}, m_{11}, \ldots, m_{k0}, m_{k1} \xleftarrow{2k} \{0,1\}^k \\ (\boldsymbol{pk}, \boldsymbol{sk}) \leftarrow \mathtt{KGen}^k(1^n),\ x \leftarrow \{0,1\}^k,\ \boldsymbol{c} = \mathtt{Enc}^k_{\boldsymbol{pk}}(\boldsymbol{m}_x, x) \\ b_i \leftarrow \{0,1\},\ \text{if } b_i = 1 \text{ then } m'_{ij} = m_{ij} \text{ else } m'_{ij} = m_{i(1-j)} \\ \text{for } i \in [k], j \in \{0,1\}.\ \boldsymbol{m'} := m'_{10} m'_{11}, \ldots, m'_{k0}, m'_{k1}. \end{array} : \mathcal{A}(\boldsymbol{m'}, \boldsymbol{pk}, \boldsymbol{c}) = x\right] \tag{10}$$

Here $\boldsymbol{m'}$ is obtained by independently interchanging two messages in each of k pairs in $\boldsymbol{m}$ at random. The reason why $|\varepsilon - \varepsilon_1|$ is negligible is with the same argument as in Lemma 1: the semantically secure encryption hides every bit of encrypted message. If one can distinguish $\boldsymbol{m'}$ from $\boldsymbol{m}$ as appeared in experiments (9) and (10) with respect to $(\boldsymbol{pk}, \boldsymbol{c})$, it will break semantic security of encryption system Π^k_{PRG}.

This can be formally proved by hybrid argument: let $\boldsymbol{m}_0 = \boldsymbol{m}$, and for $i = 1, \ldots, k$, let $\boldsymbol{m}_i = \boldsymbol{m}_{i-1}\{m_{i0}/m'_{i0}, m_{i1}/m'_{i1}\}$ meaning that $\boldsymbol{m}_i$ is the same as $\boldsymbol{m}_{i-1}$ except replacing m_{i0} with m'_{i0}, and m_{i1} with m'_{i1}, respectively. We have $\boldsymbol{m}_k = \boldsymbol{m'}$. Since it is a semantically secure encryption, no ppt algorithm will be able to tell $\boldsymbol{m}_{i-1}, \boldsymbol{pk}, \boldsymbol{c}$ and $\boldsymbol{m}_i, \boldsymbol{pk}, \boldsymbol{c}$ apart, otherwise with the same argument as in proof for Lemma 1, it will break the semantic security of Π, and hence the semantic security of Π^k_{PRG}. That finally shows no ppt algorithm can distinguish $\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{c}$ and $\boldsymbol{m'}, \boldsymbol{pk}, \boldsymbol{c}$, except with negligible probability, since k is a polynomial in n. Therefore $|\varepsilon - \varepsilon_1|$ is negligible.

The probability ε_1 in (10) is evidently the same as probability ε_2 defined as follows:

$$\mathbf{Pr}\left[\begin{array}{c} m_1, \ldots, m_k \xleftarrow{k} \{0,1\}^k,\ (\boldsymbol{pk}, \boldsymbol{sk}) \leftarrow \mathtt{KGen}^k(1^n) \\ x \leftarrow \{0,1\}^k, \boldsymbol{c} = \mathtt{Enc}^k_{\boldsymbol{pk}}(m, x);\ n_i \leftarrow \{0,1\}^k, b_i \leftarrow \{0,1\} \\ \text{if } b_i = 1 \text{ then } m'_{i0} := m_i, m'_{i1} := n_i \text{ else } m'_{i0} := n_i, m'_{i1} := m_i \\ \text{for } i = 1, \ldots, k\ .\ \boldsymbol{m'} := m'_{10}, m'_{11}, \ldots, m'_{k0}, m'_{k1} \end{array} : \mathcal{A}(\boldsymbol{m'}, \boldsymbol{pk}, \boldsymbol{c}) = x\right] \tag{11}$$

Where $m = m_1 \cdots m_k$ and $\mathtt{Enc}^k(\cdot,\cdot)$ is the encryption algorithm in Π^k_{PRG} defined in Definition 8.

The experiment in equation (11) is interpreted as a random string extracting experiment as in equation (3) by an adversary $\mathcal{B}$ as follows:

Challenger chooses encryption keys $(\boldsymbol{pk}, \boldsymbol{sk}) \leftarrow \mathtt{KGen}'(1^n)$ and a random message $m = m_1 \ldots m_k \leftarrow \{0,1\}^{k \times k}$. It then encrypts m with a random string $x \leftarrow \{0,1\}^k$. As a result it has $\boldsymbol{c} = \mathtt{Enc}^k_{\boldsymbol{pk}}(m, x)$. The public key $\boldsymbol{pk}$ and ciphertext $\boldsymbol{c}$ is given to $\mathcal{B}$.

Algorithm $\mathcal{B}$ will choose $n_i \leftarrow \{0,1\}^k$ and $b_i \leftarrow \{0,1\}$ independently at random for all $i \in [k]$. It then sets $m'_{i0} := m_i, m'_{i1} := n_i$ if $b_i = 1$, else $m'_{i0} := n_i, m'_{i1} := m_i$. Let $\boldsymbol{m'} := m'_{10}, m'_{11}, \ldots, m'_{k0}, m'_{k1}$. $\mathcal{B}$ invokes $\mathcal{A}$ with $(\boldsymbol{m'}, \boldsymbol{pk}, \boldsymbol{c})$ to output whatever $\mathcal{A}$ outputs.

The success probability of $\mathcal{B}$ in extracting the random string x used in the encryption is the same as the probability ε_2 in equation (11). According to Lemma 2, ε_2 shall be a negligible function in security parameter n .

Therefore, ε_1 is negligible. From negligibility of $|\varepsilon - \varepsilon_1|$, we have ε is negligible, which shows the one-wayness of $\mathcal{F}^{\varPi}_{\mathrm{PRG}}$. □

From Theorem 1, we obtain one of our main conclusions.

Theorem 2. *The existence of semantically secure public-key cryptosystems is equivalent to the existence of the injective one-way trapdoor functions.*

Proof. It is well known [19, 36] that the existence of injective one-way trapdoor functions implies the existence of semantically secure encryptions.

On the other side, it follows from construction in Definition 8 and Theorem 1 that the functions $\mathcal{F}^{\varPi}_{\mathrm{PRG}}$ is a collection of injective one-way trapdoor functions given $\varPi$ a semantically secure cryptosystem and PRG a pseudorandom generator. The existence of semantically secure encryption implies the existence of one-way functions [21], which in turn implies the existence of pseudorandom generators [20, 16]. Therefore, the existence of semantically secure public-key cryptosystems implies the existence of injective one-way trapdoor functions. □

5 From Encryption Trapdoor Functions to CCA2 Secure Cryptosystems

In this section we review the notions of security under correlated products and its relation to CCA2 secure encryption scheme. We will redefine the encryption trapdoor functions based on another extension for any cryptosystem with respect to pseudorandom functions, rather than pseudorandom generator. The new encryption trapdoor functions can be proved secure in a similar way as before. It is further proved that they are secure correlated products under uniform, repetitional distribution (see the definitions in following subsection). Combining this with results in [33], it shows that the semantically secure encryptions imply CCA2 secure encryptions.

As pointed earlier, this approach is the same as transformation from a semantically secure encryption to randomness reusable multi-message encryption adopted in [3], we cite the result there directly without proof.

5.1 Review of Secure Correlated Products

Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions over domain $\{0,1\}^k$ and $\ell(n)$ a polynomial on $\mathbb{N}$. The ℓ-wise product functions is defined as the ℓ-wise Cartesian products of functions from $\mathcal{F}$ on domain $\{0,1\}^{\ell \times k}$. Formally,

Definition 9 (ℓ-Wise Product [33]). *Let $\mathcal{F} = (G, f)$ be a collection of efficiently computable functions and $\ell(n)$ a polynomial in security parameter n. The ℓ-wise product $\mathcal{F}_\ell = (G_\ell, F_\ell)$ is defined as follows:*

`Efficient sampling` *On input 1^n, G_ℓ invokes $G(1^n)$ for ℓ times and outputs $(s_1, s_2, \ldots, s_\ell)$ for the indexes of a function.*

`Efficient evaluation` *On input* $(s_1, s_2, \ldots, s_\ell\,; x_1, x_2, \ldots, x_\ell)$*, algorithm* $\mathcal{F}_\ell$ *evaluates each function* s_i *on* $x_i \in \{0,1\}^k$ *separately and outputs* $(F(s_1, x_1), F(s_2, x_2), \ldots, F(s_\ell, x_\ell))$*.*

For any distribution $\mathcal{C}_\ell$ on $\{0,1\}^{\ell \times k}$, with the inputs according to $\mathcal{C}_\ell$, product $\mathcal{F}_\ell$ needs not to be one-way even if $\mathcal{F}$ is a one-way function family. Hence the definition:

Definition 10 (Secure Correlated Products [33]). *Let* $\mathcal{F}$ *be a collection of efficiently computable functions and* $\mathcal{C}_\ell$ *a distribution over domain* $\{0,1\}^{\ell k}$ *for some polynomial* ℓ*. We say that* $\mathcal{F}$ *is secure under* $\mathcal{C}_\ell$*-correlated product if* $\mathcal{F}_\ell$ *is one-way with respect to the input distribution* $\mathcal{C}_\ell$ *.*

5.2 Rosen-Segev Scheme

We include Rosen-Segev [33] CCA2 secure encryption scheme in this section for completeness.

A collection of injective trapdoor functions $\mathcal{F}$ and one-time unforgeable signature scheme $(\mathtt{KGen_{sig}}, \mathtt{Sign}, \mathtt{Ver})$ is used in Rosen-Segev cryptosystem $\mathit{\Pi}_{RS} = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$. It is defined as follows.

`Key generation KGen` invokes $G(\cdot)$ for 2ℓ-times with input 1^n to obtain $(s_i^j, td_i^j) \leftarrow G_\ell(1^n)$ for $i \in [2\ell]$ and $j \in \{0,1\}$. To generate $(k^*_{\mathtt{Ver}}, k^*_{\mathtt{Sign}}) \leftarrow \mathtt{KGen_{sign}}(1^n)$ and denote $k^*_{\mathtt{Ver}} = v_1 v_2 \cdots v_\ell \in \{0,1\}^\ell$. The public key PK and secret key SK are

$$\begin{aligned} PK &= \big((s_1^0, s_1^1), \ldots, (s_\ell^0, s_\ell^1)\big) \\ SK &= (k^*_{\mathtt{Ver}}, td^{1-v_1}, \ldots, td^{1-v_k}) \end{aligned}$$

`Encryption` To encrypt bit $m \in \{0,1\}$, algorithm `Enc` generates $(k_{\mathtt{Ver}}, k_{\mathtt{Sign}}) \leftarrow \mathtt{KGen_{sign}}(1^n)$. Denote $k_{\mathtt{Ver}} = u_1 u_2 \cdots u_\ell \in \{0,1\}^\ell$. Choose $x \in \{0,1\}^n$ uniformly at random and compute

$$\begin{aligned} &y_i = F(s_i^{u_i}, x),\ \forall i \in [\ell] && c_1 = m \oplus h(s_1^{u_1}, \ldots, s_\ell^{u_\ell}, x) \\ &c_2 = \mathtt{Sign}\big(k_{\mathtt{Sign}}, (y_1, \ldots, y_k, c_1)\big) && C = \big(k_{\mathtt{Ver}}, (y_1, \ldots, y_k, c_1), c_2\big) \end{aligned}$$

Output C as ciphertext.

`Decryption` Algorithm `Dec` receives ciphertext C as input:

```
if   $k_{\mathtt{Ver}} = k^*_{\mathtt{Ver}}$ or $\mathtt{Ver}(k_{\mathtt{Ver}}, , y_1, \ldots, y_k, c_1, c_2) \neq 1$   then output $\perp$.
else   let $j \in [\ell]$ s.t. $v_j \neq u_j$ . Compute $x = F^{-1}(td_j^{u_j}, y_j)$
    if   $\exists i \in [\ell]$ s.t. $y_i \neq F(s_i^{u_i}, x)$   then output $\perp$
    else output $c_1 \oplus h(s_1^{u_1}, \ldots, s_\ell^{u_\ell}, x)$
```

Let $\mathcal{C}_\ell$ be the uniform ℓ-repetition distribution, Rosen and Segev prove the following.

Theorem 3 (Rosen-Segev [33]). *If the collection of injective trapdoor function* $\mathcal{F}$ *that is secure under* $\mathcal{C}_\ell$*-correlated products, and* $(\mathtt{KGen_{sig}}, \mathtt{Sign}, \mathtt{Ver})$ *is one-time unforgeable signature scheme, the encryption scheme* $\mathit{\Pi}_{RS}$ *is CCA2 secure.*

A proof of Theorem 3 can be found in [33].

5.3 Realization with Pseudorandom Functions

In order to construct a collection of correlated secure trapdoor functions, we will redefine the encryption trapdoor function in this section. In previous extension for the cryptosystems in definition 6, a pseudorandom generator is used. While this is good enough for our construction of a collection of injective trapdoor functions as already proved, it is not convenient in presentation when we consider the correlated security.

In the new construction, we use the same idea as before, but it adopts a collection of pseudorandom functions instead of a pseudorandom generator. In fact, we have a general construction similarly to the ℓ-extension in Definition 6. This construction is the same as the transformation in [3] and is presented formally as follows:

Definition 11 (ℓ-Extension of Cryptosystems, revised). *Suppose ℓ is an integral polynomial in n. Let Π be a public-key encryption scheme with parameters as in definition 12 and* PRF *a collection of pseudorandom functions from $\{0,1\}^{k''}$ to $\{0,1\}^{k'}$. Define a new encryption scheme $\Pi^{\ell}_{\mathrm{PRF}} = (\mathtt{KGen}^{\ell}, \mathtt{Enc}^{\ell}, \mathtt{Dec}^{\ell})$ as follows:*

`Key Generation` $\mathtt{KGen}^{\ell}(\cdot)$: *On input security parameter 1^n, it invokes $\mathtt{KGen}(1^n)$ for ℓ times to get key pair $(pk_i, sk_i) \leftarrow \mathtt{KGen}(1^n)$ for $i = 1, \ldots, \ell$. Let the public key be $pk = (pk_1, \ldots, pk_{\ell})$ and the secret key $sk = (sk_1, \ldots, sk_{\ell})$.*

`Encryption` $\mathtt{Enc}^{\ell}_{\boldsymbol{pk}}(\cdot,\cdot)$: *On input $m = m_1 \cdots m_{\ell} \in \{0,1\}^{\ell k}$ as input, algorithm $\mathtt{Enc}^{\ell}_{\boldsymbol{pk}}(\cdot,\cdot)$ chooses $r \leftarrow \{0,1\}^k$ and computes*

$$G_{\mathrm{prf}}(r) = r' \qquad r_i = F_{\mathrm{prf}}(r', pk_i) \qquad c_i = \mathtt{Enc}_{pk_i}(m_i, r_i)$$

where $|r_i| = k', |m_i| = k$ for all $i \in [\ell]$. The output is $\boldsymbol{c} = (c_1, \ldots, c_{\ell})$. That is, $\mathtt{Enc}^{\ell}_{\boldsymbol{pk}}(m, r) = \boldsymbol{c}$.

`Decryption` $\mathtt{Dec}^{\ell}_{\boldsymbol{sk}}(\cdot)$: *On input $\boldsymbol{c} = (c_1, \ldots, c_{\ell})$ and $\boldsymbol{sk}$, it computes $m_i = \mathtt{Dec}_{sk_i}(c_i)$ for each $i \in [\ell]$ and outputs $m = m_1 \cdots m_{\ell}$.*

The result of Theorem 4 belongs to Bellare, Boldtreva, Kurosawa and Staddon [3], where their main purpose is to construct a *random reusable multi-messages encryption system.* The cryptosystem $\Pi^{\ell}_{\mathrm{PRF}}$ is essentially to encrypt ℓ messages with the same random string r and thus the name.

Theorem 4 (BBKS [3]). *Given Π as a semantically secure encryption system and* PRF *a collection of pseudorandom functions with appropriate parameters as in Definition 11. For any integral polynomial $\ell(n)$, the scheme $\Pi^{\ell}_{\mathrm{PRF}}$ defined above is a semantically secure cryptosystem.*

With this new extension of encryption, our new construction is presented formally as follows.

Definition 12 (Encryption Trapdoor Functions, Revised). *Let $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a cryptosystem with message space $\{0,1\}^k$ and random string space $\{0,1\}^{k'}$. In addition, we assume that the keys in Π are in $\{0,1\}^{k''}$. Let* $\mathrm{PRF} = (G_{\mathrm{prf}}, F_{\mathrm{prf}})$ *be a collection of pseudo-random functions from $\{0,1\}^{k''}$ to $\{0,1\}^{k'}$. The notations in Definition 11 are inherited here. A collection of efficient computable functions $\mathcal{F}^{\Pi}_{\mathrm{PRF}} = (G, F)$ is defined in the same way as in Definition 8, except that the extended encryption scheme $\Pi^{\ell}_{\mathrm{PRG}}$ is here replaced with $\Pi^{\ell}_{\mathrm{PRF}}$ as defined in Definition 11. The detailed is omitted here.*

Theorem 4 allows us to restate Theorem 1 as follows.

Theorem 5. *Given Π as a semantically secure encryption system and* PRF *a collection of pseudorandom functions with appropriate parameters as in Definition 12, the functions $\mathcal{F}^{\Pi}_{\mathrm{PRF}}$ are a collection of injective trapdoor functions.*

The proof is similar to that of Theorem 1 and is omitted here. Here comes our key theorem:

Theorem 6 (Correlated Product Security). *Let $\ell(n)$ be any integral polynomial in n. Given Π as a semantically public-key cryptosystem and* PRF *as a collection of pseudorandom functions as in Definition 12, the function family $\mathcal{F}^{\Pi}_{\mathrm{PRF}}$ is secure correlated products under uniform, repetitional distribution $\mathcal{C}_\ell$.*

Proof. Given the collection of functions $\mathcal{F}^{\Pi}_{\mathrm{PRF}}$ and ℓ an integer, the ℓ-wise product $\mathcal{F}^{\Pi,\ell}_{\mathrm{PRF}} = (G_\ell, F_\ell)$ is defined as in Definition 9. Let $(s_1, s_2, \dots, s_\ell) \leftarrow G_\ell(1^n)$ be ℓ indexes, where $s_i = (\boldsymbol{m}^i, \boldsymbol{pk}_i)$, $\boldsymbol{m}^i = (m^i_{10}, m^i_{11}, \dots, m^i_{k0}, m^i_{k1})$, and $\boldsymbol{pk}_i = (pk_{i1}, \dots, pk_{ik})$ for all $i \in [\ell]$.

For any string $x \in \{0,1\}^k$, let $\boldsymbol{m}^i_x$ be the characteristic string of x with respect to $\boldsymbol{m}^i$. The evaluation of F_ℓ on input $(s_1, s_2, \dots, s_\ell;\ x, \dots, x)$ is

$$\begin{aligned}(F_{s_1}(x), \dots, F_{s_\ell}(x)) &= (\mathtt{Enc}^k_{\boldsymbol{pk}_1}(\boldsymbol{m}^1_x, x), \dots, \mathtt{Enc}^k_{\boldsymbol{pk}_\ell}(\boldsymbol{m}^\ell_x, x)) \\ &= ((c_{11}, \dots, c_{1k}), \dots, (c_{\ell 1}, \dots, c_{\ell k})) \qquad (12)\end{aligned}$$

where $r = G_{\mathrm{prf}}(x)$, $r_{ij} = F_{\mathrm{prf}}(r, pk_{ij})$ and $c_{ij} = \mathtt{Enc}_{pk_{ij}}(m^i_{jx_j}, r_{ij})$ for all $i \in [\ell]$ and $j \in [k]$. $\boldsymbol{pk} = (pk_{11}, pk_{12}, \dots, pk_{1k}, \dots, pk_{\ell 1}, pk_{\ell 2}, \dots, pk_{\ell k})$.

The last term in equation (12) is in the form of a ciphertext to a message encrypted under $\ell\, k$-extension of Π with respect to PRF by Definition 11.

To see that for any randomly chosen $x \in \{0,1\}^k$, it indeed is a valid ciphertext under $\Pi^{\ell k}_{\mathrm{PRF}}$. One notices that the index $(s_1, \dots, s_\ell)$ is obtained by running $\mathtt{KGen}^k(1^n)$ independently for ℓ times and choosing $\boldsymbol{m}^i$ uniformly at random according to the definition of ℓ-wise product. Which indicates that all the pk_{ij} in $\boldsymbol{pk}$ are generated independently by $\mathtt{KGen}(1^n)$ in Π, and all m_{i0} and m_{i1} are chosen uniformly independent. That means $\boldsymbol{pk}$ is generated in the same way as by $\mathtt{KGen}^{\ell k}(\cdot)$ from definition. For any random string $x \in \{0,1\}^k$, (12) is hence a valid ciphertext to $\boldsymbol{m}^1_x \cdots \boldsymbol{m}^\ell_x$ under $\Pi^{\ell k}_{\mathrm{PRF}}$. That is,

$$(F_{s_1}(x), \dots, F_{s_\ell}(x)) = \mathtt{Enc}^{\ell k}_{\boldsymbol{pk}}(\boldsymbol{m}^1_x \cdots \boldsymbol{m}^\ell_x, x) \qquad (13)$$

To show one-wayness of $\mathcal{F}^{\Pi,\ell}_{\mathrm{PRF}} = (G_\ell, F_\ell)$, one have to show the probability ε defined as follows is negligible in secure parameter n:

$$\varepsilon = \mathbf{Pr}\left[\begin{array}{c} \forall i \in [\ell],\ \boldsymbol{m}^i = m^i_{10}, m^i_{11}, \dots, m^i_{k0}, m^i_{k1} \xleftarrow{2k} \{0,1\}^k, \\ (\boldsymbol{pk}_i, \boldsymbol{sk}_i) \leftarrow \mathtt{KGen}^k(1^n),\ s_i = (\boldsymbol{m}^i, \boldsymbol{pk}_i),\ x \leftarrow \{0,1\}^k \\ \boldsymbol{m} := (\boldsymbol{m}^1, \dots, \boldsymbol{m}^\ell),\ \boldsymbol{c} := (F_{s_1}(x), \dots, F_{s_\ell}(x)) \end{array} : \mathcal{A}(\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{c}) = x\right] \qquad (14)$$

$$= \mathbf{Pr}\left[\begin{array}{c} \forall i \in [\ell],\ \boldsymbol{m}^i = m^i_{10}, m^i_{11}, \dots, m^i_{k0}, m^i_{k1} \xleftarrow{2k} \{0,1\}^k, \\ (\boldsymbol{pk}, \boldsymbol{sk}) \leftarrow \mathtt{KGen}^{\ell k}(1^n),\ x \leftarrow \{0,1\}^k, \\ \boldsymbol{m} := (\boldsymbol{m}^1, \dots, \boldsymbol{m}^\ell),\ \boldsymbol{c} = \mathtt{Enc}^{\ell k}_{\boldsymbol{pk}}(\boldsymbol{m}^1_x \cdots \boldsymbol{m}^\ell_x, x) \end{array} : \mathcal{A}(\boldsymbol{m}, \boldsymbol{pk}, \boldsymbol{c}) = x\right] \qquad (15)$$

Where equation (14) is from definition of secure correlated product, and equation (15) is from equation (13) and the discussion as above.

According to Theorem 4, $\Pi^{\ell k}_{\mathrm{PRF}}$ is a semantically secure encryption. Using the same arguments as in the proof of one-wayness property for Theorem 1, one can show ε is negligible. That shows the one-wayness of the products $\mathcal{F}^{\Pi,\ell}_{\mathrm{PRF}}$ under distribution $\mathcal{C}_\ell$. □

This leads to our next main conclusion in this paper.

Theorem 7. *The existence of semantically secure public-key cryptosystems is equivalent to the existence of CCA2 secure public-key cryptosystems.*

Proof. Since a CCA2 secure encryption system is already a semantically secure one. Hence one direction is obvious.

On the other side, the existence of semantically secure encryption implies the existence of one-way functions [21], which in turn implies the existence of pseudorandom functions [17, 16] and the existence of one-time unforgeable signature schemes [32].

Hence the existence of semantically secure encryption implies injective trapdoor functions by Theorem 5.

By Theorem 6, the existence of semantically secure encryption further implies the existence of injective one-way trapdoor functions that are secure under $\mathcal{C}_\ell$-correlated products. Where $\mathcal{C}_\ell$ is uniform ℓ-repetition distribution for any integral polynomial $\ell(n)$.

Recall that the CCA2 encryption scheme by Rosen-Segev (Section 5.2) is based on injective trapdoor functions secure under $\mathcal{C}_\ell$-correlated products and one-time unforgeable signature schemes. Both, however, exists if semantically secure public-key cryptosystems exist. □

Combining Theorem 1 (and Theorem 5) and Theorem 7, we conclude

Theorem 8. *The injective trapdoor functions exist if and only if CCA2 secure cryptosystems exist.*

6 Conclusions and Discussions

We conclude in this paper that the existence of injective trapdoor functions is equivalent to the existence of semantically secure public-key cryptosystems, and to the existence of CCA2 secure public-key cryptosystems. This settles two long-standing open problems in cryptography. The conclusions indicate that in the sense of existence, the notions of security for public-key cryptosystem like CCA1 secure, non-malleable against CPA, or CCA, bounded CCA secure collapse to one.

The main technique here is to employ the randomness reusable multi-message encryption, using the input to the function as randomness to encrypt its characteristic strings with respect to a prescribed set of messages. The security comes from the semantic security of underlying encryption scheme.

The way of constructing to trapdoor functions here is different from that adopted before. It may have separate significance itself. It leads new constructions of injective trapdoor functions under concrete assumptions like DDH or learning with errors (LWE) [31](resolved recently by Peikert and Waters in [28]). The constructions here are much more efficient than those in [28]. Especially, it gives the first injective trapdoor functions based on Ajtai-Dwork cryptosystem [1, 2].

Bellare et. al. [4] showed that trapdoor functions with poly-bounded pre-image size imply semantically secure encryptions. Combining their result with ours, it concludes that trapdoor functions with poly-bounded pre-image size exist if and only if injective trapdoor functions exist, which was known previously only to hold in random oracle model [4].

Our conclusions also imply, in the sense of existence, the equivalence between secure correlated products and injective trapdoor functions, which answer the motivating question of secure correlated products in [33].

Acknowledgements

We are grateful to Abhilasha Bhargav-Spantzel and Chuankun Wu for their helps in improving the presentation of this work and make it better in form.

References

1. Ajtai and Dwork. The first and fourth public-key cryptosystems with worst-case/average-case equivalence. In *Electronic Colloquium on Computational Complexity, technical reports*, 2007.
2. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc.* 29*th ACM Symp. on Theory of Computing*, pages 284–293, 1997.
3. M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon. Multirecipient encryption schemes: How to save on bandwidth and computation without sacrificing security. *IEEE Transactions on Information Theory*, 53(11):3927–3943, 2007.
4. M. Bellare, S. Halevi, A. Sahai, and S. Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In H. Krawczyk, editor, *Advances in Cryptology, Proc. CRYPTO' 98,* LNCS 1464, pages 283–298. Springer-Verlag, 1998.
5. D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS. In H. Krawczyk, editor, *Advances in Cryptology, Proc. CRYPTO' 98,* LNCS 1464, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, 1998.
6. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput*, 36(5):1301–1328, 2007.
7. S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In R. Canetti, editor, *Theory of Cryptography Conference, Proc. TCC'08*, volume 4948 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2008.
8. R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded CCA2-secure encryption. In *Advances in Cryptology–ASIACRYPT'07*, pages 502–518, 2007.
9. R. Cramer and V. Shoup. A practical public-key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology, Proc. CRYPTO' 98,* LNCS 1464, pages 13–25, 1998.
10. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, Nov. 1976.
11. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc.* 23*rd ACM Symp. on Theory of Computing*, pages 542–552. ACM, 1991.
12. E. Elkind and A. Sahai. A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/041, 2002.
13. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc.* 19*th ACM Symp. on Theory of Computing*, pages 197–206, 2008.
14. Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public-key encryption. In S. P. Vadhan, editor, *Theory of Cryptography Conference, Proc. TCC'07*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455. Springer, 2007.
15. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates (extended abstract). In *Proc.* 42*st IEEE Symp. on Foundations of Comp. Science*, pages 126–135, 2001.
16. O. Goldreich. *Foundations of Cryptography–Basic Tool*, volume I. Cambridge University Press, 2001.
17. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, Oct. 1986.
18. S. Goldwasser. New directions in cryptography: Twenty some years later. In *Proc.* 38*th IEEE Symp. on Foundations of Comp. Science*, pages 314–324, Los Alamitos-Washington-Brussels-Tokyo, 1997. IEEE Computer Society, IEEE Computer Society Press.
19. S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28(2):270–299, Apr. 1984.
20. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
21. R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In IEEE, editor, *30th annual Symposium on Foundations of Computer Science*, pages 230–235. IEEE Computer Society Press, 1989.

22. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In S. Goldwasser, editor, *Advances in Cryptology, Proc. CRYPTO' 88,* LNCS 403, pages 8–26. Springer-Verlag, 1988.
23. J. Katz and C.-Y. Koo. On constructing universal one-way hash functions from arbitrary one-way functions. Cryptology ePrint Archive, Report 2005/328, 2005. http://eprint.iacr.org/.
24. Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *Journal of Cryptology*, 19(3):359–377, 2006.
25. R. C. Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21:294–299, Apr. 1978.
26. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *Proc. of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, Maryland, 1990. ACM.
27. R. Pass, A. Shelat, and V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 271–289. Springer, 2006.
28. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc.* 19*th ACM Symp. on Theory of Computing*, pages 187–196, New York, NY, USA, 2008. ACM.
29. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology, Proc. CRYPTO' 91,* LNCS 576, pages 433–444. Springer, 1992.
30. O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.
31. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc.* 19*th ACM Symp. on Theory of Computing*, pages 84–93, 2005.
32. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc.* 22*nd ACM Symp. on Theory of Computing*, pages 387–394, Baltimore, Maryland, 1990. ACM.
33. A. Rosen and G. Segev. Chosen ciphertext security via correlated products. Cryptology ePrint Archive, Report 2008/116, 2008. http://eprint.iacr.org/.
34. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, 1999.
35. V. Shoup. *Why Chosen Ciphertext Security Matters.* IBM Research Report RZ 3076, November 1998.
36. A. C. Yao. Theory and application of trapdoor functions. In *Proc.* 23*rd IEEE Symp. on Foundations of Comp. Science*, pages 80–91, Chicago, 1982.