On the Security of Multiple Encryption or CCA-security+CCA-security=CCA-security?

Rui Zhang[*] Goichiro Hanaoka[*] Junji Shikata[†] Hideki Imai[*]

Abstract

In a practical system, a message is often encrypted more than once by different encryptions, which we call multiple encryption, to enhance its security. Other schemes use multiple encryption to achieve new features, such as anonymity of sender. Intuitively, whenever there is one component cipher unbreakable, such multiple encryption remains "secure". However, in this paper we show this is *not* always true. We introduce a new security definition called ME-CCA that is a natural extension of the CCA security in analyzing partial breaking of component ciphers in a multiple encryption. Our result shows even all component ciphers are IND-CCA secure, the whole multiple encryption may be not possibly ME-CCA secure. One the other hand, ME-gCCA security, the relaxation of ME-CCA security, is easily gained if the all component ciphers are gCCA secure. We study the relation of security notions of multiple encryption. We apply this analysis to key-insulated cryptosystem [10], showing the generic construction there is actually insecure against CCA attack. However, we point out their generic construction is in fact provably secure in the relaxed ME-gCCA model. We further add a patch to their scheme, which yields the first key-insulated cryptosystem provably secure against CCA attack.

1 Introduction

A practical cryptosystem often encrypts a message several times with independent secret keys or even encryption schemes based on different assumptions to enhance the confidentiality of message, (specifically double encryption and triple encryption for two times and three times multiple encryptions).

Why multiple encryption. This approach is widely believed to provide better security because even if underlying assumptions of some component ciphers are broken or some of secret keys are compromised, the confidentiality can still be maintained by the remaining encryptions. Historically, sudden emergence of efficient attacks against the elliptic curve cryptosystem on supersingular curves [22] and on prime-field anomalous curves [31] have already reminded us the necessity to do this. For example, it is suggested by NESSIE [24] on asymmetric encryption scheme to "use double encryption using ACE-KEM and RSA-KEM with different DEMs gives a good range of security, based on various different assumptions", "if very long term security is important". Furthermore, "Triple encryption that also uses a public-key scheme not based on number-theoretical assumptions might increase the security against future breakthrough".

On the other hand, multiple encryption may provide additional new features. Combination of ordinary threshold encryptions may yield new threshold encryption with various access structure. Many practical applications achieving sender anonymity via practical open network, Mix-net, Onion routing and so on, are all typical constructions of multiple encryption.

Multiple encryption has been studied in the literature such as [21, 2]. However, we argue in what sense this is correct, because in these work, only kind of passive attack on symmetric encryption is considered,

[*] University of Tokyo. Email: {zhang,hanaoka}@imailab.iis.u-tokyo.ac.jp, imai@iis.u-tokyo.ac.jp

[†] Yokohama National University. Email: shikata@mlab.jks.ynu.ac.jp

while the analysis under the standard security definition of public key encryption schemes, especially the chosen ciphertext security, does not seem to have been touched yet.

In this paper, we show that even if it consists of only *independently* selected IND-CCA secure components, a multiple encryption is not necessarily secure at all in the sense of chosen ciphertext attack. This seems contradictive to our intuition at the first sight. However, many natural constructions of multiple encryption from combinations of IND-CCA secure components can be shown easily to lose the CCA security. For example, one of our results shows that generic construction of key-insulated encryption [10] from cover free family can be broken under CCA. On the other hand, this result may imply CCA-security is too strong a notion because it overkills some schemes with practical use. We shall discuss relaxed security definition emphasizing practical usability based on the "natural" constructions. As theoretical interest, we also propose a construction of multiple encryption scheme achieving CCA security. However, our work is chiefly to show the existence of multiple encryption with various different assumptions can be CCA secure. It cannot compare to the performance of the key insulated cryptosystem [10] for practical resistance of key exposure.

1.1 Related work

In this section we review some previous work on multiple encryption and related primitives. Rather than simple repeat of ordinary public key encryption schemes, we regard multiple encryption as a separate primitive.

Multiple Encryption and Related Primitives. Multiple encryption has already been used in practical schemes against partial key exposures in distributed systems. One example is the key-insulated cryptosystem, recently proposed by Dodis et al. [10]. The lifetime of a (t, N) key-insulated secure cryptosystem is divided into N periods, where the user secret key is updated at the beginning of each period. The generic construction in [10] is formed by a multiple encryption: a message is first processed by AONT (All-Or-Nothing-Transform) [26, 4] into n shares, and encrypt these shares with n public keys, with corresponding secret keys generated from cover free family [20]. With physical assumption (separate physically secure device), it is guaranteed that secret key of period i cannot be compromised even if user secret keys are exposed to the adversary up to a number of t other periods.

Another important category of applications using multiple encryption are those practical implementations of anonymous channel in open network, for instance, the Mix-net [18] and onion routing [6]. In these settings, several agents are appointed to transmit data from the sender to the receiver without revealing identity of the sender. Typical design of such protocols is to encrypt data under multiple public keys of these agents, which decrypt the data one layer after another until eventually reach the destination. It is essential to perform these decryption correctly, e.g., [1] has shown some practical attacks against some carelessly designed Mix-net protocols [19, 17], which if translated in our language, are insecure multiple encryption.

A similar notion to multiple encryption is the threshold cryptosystem [7, 8, 30], which maintains secrecy of decryption key even if some of the secret key shares are compromised. However, all known constructions are based on particular number theoretic assumption and can be employed to only a restrictive range of applications.

Security Notions. To define the security of an encryption scheme, one needs to consider two aspects: the security goal and attack model, as has been suggested in [3]. Standard definition of a public key encryption scheme is founded with gradual progress in literature e.g. [16, 11, 25, 3, 12] and the strongest security notion is proved to be indistinguishability against (adaptive) chosen-ciphertext attack (IND-CCA) [3, 12]. *Semantic security*, first defined by Goldwasser and Micali [16], later refined by Goldreich [15] and Watanabe, Shikata and Imai [32], captures the computational approximation of Shannon's information-theoretic security, or perfect secrecy [27], regulating that it should be infeasible for any PPT (Probabilistic Polynomial Time) adversary to obtain any partial information about the plaintext of a given ciphertext. A similar definition, *indistinguishability*, defines that given a ciphertext an adversary cannot distinguish which plaintext is encrypted from two plaintexts. Indistinguishability is proved to be equivalent to semantic security in several attack models, namely chosen plaintext attack (CPA), (non-adaptive) chosen-ciphertext attack (CCA1) and adaptive chosen-ciphertext attack (CCA2) [16, 32]. Another intricate notion, *non-malleability*, defined by Dolev, Dwork and Naor [11, 12] formulates that the adversary should not be able to create a ciphertext of a different message that is meaningfully related to the original ciphertext and non-malleability implies indistinguishability in all above three attack models. Independently in [3] and [12], indistinguishability and non-malleability are proved to be equivalent under (adaptive) chosen-ciphertext attack (CCA).

CCA security is crucial in analyzing security of protocols. However, Shoup first argues CCA security is stringent for practical schemes and suggests "benign malleability" in the proposal for ISO public key encryption standard [29], as a relaxation for CCA model. An et al. give similar discussion under the name "generalized-CCA" (a.k.a. gCCA). In these two relaxations, a relation function checks and rejects "obvious" decryption queries decrypted to the target message. We note that such ciphertexts are publicly detectable.

Previous Work on Security of Multiple Encryption. Multiple encryption was addressed by Shannon as early as [27] under the name "product cipher", and in [9, 23] in context of symmetric key cryptosystems. Maurer et al. [21] have also studied similar problem under the name "cascade cipher". We point out that all above work is not adequate: largely symmetric key encryptions with sequential order are studied, while CCA security of public key multiple encryption has never been mentioned yet.

1.2 Our contributions

Our contributions are in following aspects:

Model and Security Definition of Multiple Encryption. We give a formal model regarding public key multiple encryption. To the best of our knowledge, no previous work has strict formalization on this, and actually our model can be extended to both public key and symmetric key based cryptosystems. Our model consorts the modular design: combining "secure" component cryptosystems to have a "secure" multiple encryption. Some analyses here can be applied to symmetric key schemes also. As theoretical extension of traditional security definitions, we give the corresponding security definition on multiple encryptions based on indistinguishability, especially chosen ciphertext attack. We introduce a Key Exposure Oracle ($\mathcal{KE}$) to model multiple encryption's ability. Without loss of generality, breaking underlying assumptions of component ciphers can be emulated as the secret key is leaked to the adversary.We allow underlying cryptosystems can be chosen independently as well as the secret keys. Of course, one can weaken this requirement by choosing all component on the same number theoretic assumption in practical schemes as [10]. It should be emphasized that choosing multiple encryption on different assumptions is the most generalized form of multiple encryption, with more favorable confidentiality protection to the plaintext, or provides maximum independence of multiple encrypter.

Vulnerability of Natural Multiple Encryption. We demonstrate attacks against multiple encryption schemes with each component IND-CCA secure. We show that there exists such adversary against such nature construction with access to the Decryption Oracle and Key Exposure Oracle, breaks the indistinguishability of the scheme, which consists of only IND-CCA secure components! As a matter of fact, such adversary even breaks the onewayness of the whole system.

Secure Construction of Multiple Encryption. We consider constructions of multiple encryption meeting CCA security. We exhibit the existence of transforms to enhance "weak" multiple encryption to satisfy "strong" security notion of indistinguishability against chosen ciphertext attack. However, in most cases, CCA security for multiple encryption seems rather literal and should be of little practical advantage over the straightforward combination of IND-CCA component ciphers, except that it meets CCA security exactly.

Re-defining Security of Multiple Encryption. IND-CCA security has been treated as standard definition for encryption schemes, as usually this is convenient to have modular design on cryptographical protocols in the universal composable framework [5]. However, our analysis shows CCA security may be too stringent as even IND-CCA secure components would result in a CCA insecure multiple encryption in most of the cases. We argue the CCA security definition is too strong for defining the multiple encryptions. As a reasonable relaxation, we give a new security definition named IND-ME-gCCA security that with aid of a relation function rules out some malleable ciphertext decryption queries. We prove that in the relaxed model the equivalence of indistinguishability and malleability still hold.

Security Analysis of Multiple Encryption. We also study the relation between different definitions in multiple encryption settings. We believe a good analysis of these relations will help protocol designer more than simply give a specific construction based on some concrete mathematical assumption. Actually we give analysis on the security definitions namely indistinguishability and non-malleability under a number of attack models. We show indistinguishability and non-malleability are equivalent under chosen ciphertext attack and generalized chosen ciphertext attack for multiple encryption, which has known in the extreme case: a multiple encryption degenerates to an ordinary public key cryptosystem, if there is only one component cipher in it.

Application to Key Insulated Encryption. As an application, we reconsider the security of key-insulated encryption proposed by Dodis, Katz, Xu and Yung [10]. In fact, their analysis on generic construction is insufficient: the security proof given there considers only chosen plaintext security, though their definition states also the chosen ciphertext security. In fact, their generic construction can be broken by a CCA adversary. However, we show that their scheme is in fact provably secure in the relaxed gCCA model, which reasonably supports the correctness and practical usability of their scheme. We further give a patch on their scheme to make it meet CCA security. We point out this is the first generic construction of provably secure key-insulated cryptosystem in CCA model.

2 Preliminary

We shall give the basic definitions for the rest part of the paper.

2.1 Public key encryption scheme

An public key encryption scheme $\mathcal{E}$ is a 3-tuple algorithm: $\mathcal{E} = (\mathsf{Enc\text{-}Gen}, \mathsf{Enc}, \mathsf{Dec})$. $\mathsf{Enc\text{-}Gen}(1^k)$ is a probabilistic algorithm, where k is the security parameter, with internal random coin flipping outputs a pair of keys (pk, sk). pk is the encryption key which is made public, and sk is the decryption which is kept secret. Enc may be a probabilistic algorithm that takes as input a key pk and a message m from associated message space $\mathcal{M}$, and internally flips some coins and outputs a ciphertext c, denoted by $c \leftarrow \mathsf{Enc}_{pk}(m)$, in short $c \leftarrow \mathsf{Enc}(m)$. Dec is a deterministic algorithm takes as input the ciphertext c and the secret key sk, and outputs some message $m \in \mathcal{M}$, or "$\perp$" in case c is "invalid". We denote it by $m \leftarrow \mathsf{Dec}_{sk}(c)$, in short $m \leftarrow \mathsf{Dec}(c)$.

A function $f : \mathsf{D} \rightarrow \mathbf{R}$ is called *negligible* if for every constant $l \geq 0$ there exists an integer k such that $f(k) \leq k_c^{-l}$ for all $k \geq k_c$, denoted by $\mathsf{neg}(k)$. Indistinguishability (semantic security) under chosen-ciphertext attack (IND-CCA), is defined as: if no PPT adversary $\mathcal{A}$ can distinguish encryptions of any two messages (M_0, M_1) of equal length chosen by it with negligible advantage than random guess. We require that $\mathcal{A}$ runs in two stages $\mathcal{A}_{\mathsf{find}}$ and $\mathcal{A}_{\mathsf{guess}}$, in which $\mathcal{A}_{\mathsf{find}}$ gets side information α from the queries and output a pair of challenge messages, and $\mathcal{A}_{\mathsf{guess}}$ outputs a guess $\tilde{b}$ on b according to the ciphertext C_b encrypted by Encryption Oracle with randomly chosen $b \in \{0, 1\}$. According to the ability of the adversary, $\mathcal{A}_{\mathsf{find}}$ and $\mathcal{A}_{\mathsf{guess}}$ can be assisted by an Decryption Oracle $\mathcal{DO}$ that for a decryption query other than the target ciphertext, returns the plaintext. Note that according to the adversary's ability, sometimes $\mathcal{DO}$ is unavailable (or equivalently denoted by $\mathcal{DO}$ outputting an empty string ϵ). In our analysis, it is sufficient to consider the case where $\mathcal{DO}$ is available, for one can easily construct an adversary $\mathcal{B}$ without $\mathcal{DO}$ achieving the same advantage of another adversary $\mathcal{A}$ with $\mathcal{DO}$ by using $\mathcal{A}$ as an oracle. We denote this as:

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{c} (pk, sk) \leftarrow \mathsf{Enc\text{-}Gen}(1^k), (M_0, M_1, \alpha) \leftarrow \mathcal{A}^{\mathcal{DO}}_{\mathsf{find}}(pk), \\ b \xleftarrow{R} \{0,1\}, C_b \leftarrow \mathsf{Enc}(M_b), \tilde{b} \leftarrow \mathcal{A}^{\mathcal{KE},\mathcal{DO}}_{\mathsf{guess}}(C_b, \alpha) \end{array} \right] \leq \frac{1}{2} + \mathsf{neg}(k)$$

If no such PPT adversary exists against $\mathcal{E}$, then we call $\mathcal{E}$ $\mathsf{IND\text{-}CCA}$ secure.

2.2 All-or-Nothing Transform

An AONT is a randomized transform $\mathcal{T}$ called an (L, l, n)-AONT if (1): on input $M \in \{0,1\}^L$, $\mathcal{T}$ outputs $X \overset{\text{def}}{=} (m_1, ..., m_n)$, where $m_j \in \{0,1\}^l$; (2) here exists an efficient inverse function $\mathcal{I}$ such that $\mathcal{I}(X) = M$; (3) $\mathcal{I}$ satisfies indistinguishability. Let $X_{-j} = (m_1, ..., m_{j-1}, m_{j+1}, ..., m_n)$ and $\mathcal{T}_{-j}(M) {=} X_{-j}$, where $X {\leftarrow} \mathcal{T}(M)$. Let left-or-right oracle $\mathsf{LR}_b(j, M_0, M_1) \overset{\text{def}}{=} \mathcal{T}_{-j}(M_b)$, for any PPT adversary $\mathcal{A}$ attacking AONT, define its advantage as $Adv_{\mathcal{A},\mathcal{T}} \overset{\text{def}}{=} Pr[b \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}^{\mathsf{LR}_b(\cdot,\cdot,\cdot)} : b' = b] - 1/2$. Then Adv is negligible.

2.3 Cover-free family

A family of subsets $S_1, ..., S_N$ over some unicerse U is said to be t-cover-free if no t subsets $S_{i_1}, ..., S_{i_t}$ contain a (different) subset S_{i_0}, that is, for all $\{i_0, ..., i_t\}$ with $i_0 \notin \{i_0, ..., i_t\}$, we have $S_{i_0} \nsubseteq \cup_{t=1}^{t} S_{i_j}$. A family is said to be (t, β)-cover-free, where $0 < \beta < 1$, if for all $\{i_0, ..., i_t\}$ with $i_0 \notin \{i_1, ..., i_t\}$, we have $|S_{i_0} \backslash \cup_{j=1}^{t} S_{i_j}| \geq \beta |S_{i_0}|$.

3 The model

In this section, we shall give the model of a multiple encryption, basic construction methods and relative security definitions. Multiple encryption is a generalized form of public key encryption.

3.1 Multiple encryption scheme

Informally a multiple encryption is a message encrypted by multiple cryptosystems. A multiple encryption scheme $\mathcal{ME}$ is generated by some component cipher systems. We shall give some intuitive descriptions before giving the formal definition. Naturally we have two basic combinations of these cryptosystems: parallel and serial connection among different components.

3.1.1 Basic specification

Multiple encryption is a complex system composed by distinct component ciphers, combined together to realize a certain functionality. Suppose $\{\mathcal{E}_i \,|\, i = 1, \ldots, n\}$ are a set of *compatible* component ciphers, where

$\mathsf{Enc\text{-}Gen}_i$ the probabilistic key-generation algorithm, with the input (1^k) and internal coin flipping produces a public-secret key pair (pk_i, sk_i);

Enc_i the encryption algorithm, with an input message $m_i \in \mathcal{M}_i$ and internal coin flipping, outputs a ciphertext $c_i \in \mathcal{C}_i$;

Dec_i the decryption algorithm, which is a deterministic algorithm, with the input ciphertext c_i outputs the message m_i or "$\perp$".

A multiple encryption is a 3-tuple algorithm ($\mathsf{MEnc\text{-}Gen}$,MEnc,MDec), each combined from a number of public key cryptosystems with a unifilar connecting order. $\mathsf{MEnc\text{-}Gen}$ invokes $\mathsf{Enc\text{-}Gen}_i$ and outputs a list of secret keys $(sk_1, ..., sk_n)$. MEnc with an input message M chosen from the message space $\mathcal{M}$, performs encryption MEnc on M by invoking a list of component ciphers $\{\mathcal{E}_i\}$ also including AONT $\mathcal{T}$ if necessary, eventually outputs a ciphertext $C \in \mathcal{C}$. The decryption algorithm MDec takes C as input and outputs M, or "$\perp$" if C is invalid. Essentially, we have two basic constructions:

Parallel construction. A parallel multiple encryption is an operation that messages is encrypted in parallel by cryptosystems $\mathcal{E}_1, \ldots, \mathcal{E}_n$. If a message m is chosen from the message space $\mathcal{M}$ and is directly processed by $\mathcal{E}_1, \ldots, \mathcal{E}_n$, the merit of multiple encryption will never exist – if the adversary breaks one component cipher, it succeeds. The right way is to pre-process the plaintext before encrypting it. Such pre-procession can be an All-Or-Nothing Transform (AONT) (intuitively one can consider a $(n-1, n)$ secret sharing), which maps the desired message into several sub-messages, so that only after all the sub-messages are decrypted can the plaintext be recovered. Figure 1 depicts the construction in Appendix A.

To decrypt the ciphertext $c = (c_1, \ldots, c_n)$, one uses every sk_i in the underlying $\mathcal{E}_i$ to decrypt every c_i and gets m_i $(1 \leq i \leq n)$. The plaintext m can then be reconstructed from $m_1, \ldots, m_n$. For an adversary attacking AONT, it can never obtain any information of the plaintext unless it gets all m_i's. The generic construction of the key-insulated cryptosystem [10] is an example of multiple parallel encryption.

Serial construction. Serial multiple encryption is more straightforward, is identical to cascade cipher in [21]. It should be clarified that there exists significant difference between multiple serial encryption and the product cipher [27]: for multiple encryption, each component cipher scheme can be chosen independently. Initially the plaintext is encrypted by the most inner component cipher. Each output (ciphertext) of an component cipher system will be passed on as the input of the next component cipher system. Finally the output of the last component cipher is taken as the output of this multiple encryption system. An illustration can be found in Figure 2 in Appendix A. Since the operation is done sequentially, by observing $C = c_n$, the decryption algorithm takes c_n and sk_i, $i = 1, \ldots, n$ as input and eventually outputs m. The construction of onion routing [6] is an example of multiple serial encryption.

Hybrid construction. If a multiple encryption contains both parallel encryption block and serial encryption block, we call it a hybrid multiple encryption. We give another description that may help readers to understand the structure. Consider a cipher cryptosystem with a tree structure. Fixing the root as the first layer cipher system, then adding a parallel multiple encryption to a node just encreases the sub-nodes of a node into m, where m is the number of parallelled cipher cryptosystems. Adding a serial cipher cryptosystem to a node will increase the tree depth with a factor of n from that node, where n is the number in this serial multiple encryption block. Then the output of the whole multiple encryption is the output of all nodes that don't have sub-nodes. We call the set of a node of a certain level and its sub-nodes a branch. If there is more than one end node in the branch, we say the branch ends with parallel block. Otherwise, ends with serial block. Then a multiple encryption ends with a parallel block if there is one parallel encryption block in any branch, and ends with serial encryption block if there is only one branch, with its all component cipher forming a serial encryption block.

3.1.2 Parallel construction vs serial construction

Parallel multiple encryption may act as a secure data storage where a document is split into n pieces with (t, n) threshold (computational) secret sharing other than AONT and stored in several not necessarily secure servers. As long as no more than t secret keys are not compromised, the secret is still secure. Compared to parallel multiple encryption, serial multiple encryption has gain in the data size while enhancing the security. However, for careless users who do not tend to manage their keys well, it is better to use parallel multiple encryption to have a secure backup.

3.2 Security definition

Here two things should be specified: the security goal and attack model. We should not apply $\mathsf{IND\text{-}CCA}$ directly to the multiple encryption because partially breaking of underlying assumptions (key exposure) is usually not considered in $\mathsf{IND\text{-}CCA}$. As discussed previously, a multiple encryption is aimed at improving the security in longer term. We believe breaking of assumptions of underlying component ciphers should be considered in its definition[1]. We capture this real world behavior by introducing a Key Exposure Oracle $\mathcal{KE}$, that according to the key exposure queries from the adversary, returns the secret keys of concrete component cipher systems. We shall focus the more stronger attack model CCA attack and briefly discuss the weaker attack CPA later. These definitions are natural extensions from those of traditional public key encryptions: by fixing the number of component ciphers $n = 1$, we get normal $\mathsf{IND\text{-}CCA}$ or $\mathsf{NM\text{-}CCA}$ definition.

Definition 1 ($\mathsf{IND\text{-}ME\text{-}CCA}$) *Assume any PPT adversary play the following game with a multiple encryption $\mathcal{ME}$: First key generation algorithm* $\mathsf{MEnc\text{-}Gen}$ *is run, by calling each* $\mathsf{Enc\text{-}Gen}_i$ *independently to produce distinct public key and secret key pairs. The public key $pk = \{pk_i \,|\, i = 1, \ldots, n\}$ is then given to an Encryption Oracle $\mathcal{EO}$ and the adversary. The secret key $sk = \{sk_i | i = 1, \ldots, n\}$ is given to a Decryption Oracle $\mathcal{DO}$ and a Key Exposure Oracle $\mathcal{KE}$. The adversary adaptively chooses arbitrarily a part of the component ciphers $\{\mathcal{E}_{i_t} | 1 \leq t \leq n-1\}$ at his will, informs $\mathcal{KE}$, who will hand all requested secret keys to the adversary. The adversary chooses some ciphertexts C according to his choice and submits to the Decryption Oracle $\mathcal{DO}$. The adversary sends $\{M_0, M_1\}$ to the Encryption Oracle $\mathcal{EO}$, which chooses $b \in_R \{0,1\}$, encrypts M_b and returns the corresponding ciphertext C_b to the adversary. The adversary continues to submit decryption queries. The only limit on these queries is that the adversary can't query $\mathcal{DO}$ with the target ciphertext. Finally the adversary outputs a guess $\tilde{b}$ on b. If the difference of the success probability of the adversary $\mathcal{A}$ compared to random guess in the* $\mathsf{IND\text{-}ME\text{-}CCA}$ *game is negligible:*

$$\Pr\left[b = \tilde{b} \left| \begin{array}{c} (pk, sk) \leftarrow \mathsf{MEnc\text{-}Gen}(1^k), (M_0, M_1, \alpha) \leftarrow \mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(pk), \\ b \xleftarrow{R} \{0,1\}, C_b \leftarrow \mathsf{MEnc}(M_b), \tilde{b} \leftarrow \mathcal{A}_{\mathsf{guess}}^{\mathcal{KE},\mathcal{DO}}(C_b, \alpha) \end{array} \right. \right] \leq \frac{1}{2} + \mathsf{neg}(k)$$

then we call this $\mathcal{ME}$ $\mathsf{IND\text{-}ME\text{-}CCA}$ *secure, furthermore, adversary playing this game is called* $\mathsf{ME\text{-}CCA}$ *adversary.*

in ordinary public key cryptosystem, non-malleability can be also defined. The formulation of non-malleability is that the adversary cannot produce a valid "new" ciphertext meaningfully related to the original one. We can define non-malleability under the chosen ciphertext attack of a multiple encryption.

Definition 2 ($\mathsf{NM\text{-}ME\text{-}CCA}$) *A PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $\mathcal{ME}$ schedules its attack in two stages: in the first stages, it adaptively interacts with a Decryption Oracle $\mathcal{DO}$, which on requests of ciphertext returns the plaintexts, and a Key Exposure Oracle $\mathcal{KE}$, which on requests of index of component ciphers returns the corresponding secret key. The adversary finally outputs a pair of message (M_0, M_1). In the second stage, an Encryption Oracle $\mathcal{EO}$ chooses one message randomly from the two the adversary and give the adversary a challenge ciphertext C_b. Denote $\mathbb{M}, \mathbb{C}$ as sets of plaintexts and ciphertext respectively. If the adversary cannot output a valid "new" set of ciphertext with $C_b \notin \mathbb{C}$ and a relation R with which states the relation between $M_b \leftarrow$* $\mathsf{MDec}(C_b)$ *and $\mathbb{M} \leftarrow$* $\mathsf{MDec}(\mathbb{C})$. *In both stages, the adversary may query the oracles adaptively, in any order he wants, subject to the restriction that he makes only one query to the Encryption Oracle. Such scheme is secure if any probabilistic polynomial time adversary has success probability negligibly close to 1/2.*

$$\Pr\left[b = 1 \left| \begin{array}{c} (pk, sk) \leftarrow \mathsf{MEnc\text{-}Gen}(1^k), (M_0, M_1, \alpha) \leftarrow \mathcal{A}_1^{\mathcal{KE},\mathcal{DO}}(pk), C_b \leftarrow \mathsf{MEnc}(M_1), \\ (R, \mathbb{C}) \leftarrow \mathcal{A}_2^{\mathcal{KE},\mathcal{DO}}(C_b, \alpha, M_0, M_1), \mathbb{M} \leftarrow \mathsf{MDec}(\mathbb{C}), (C_b \notin \mathbb{C}) \wedge (\perp \notin \mathbb{M}) \wedge R(M_b, \mathbb{M}) \end{array} \right. \right] \leq \frac{1}{2} + \mathsf{neg}(k)$$

[1] If all component ciphers are unbreakable, then there is no need to do multiple encryption.

4 Insecurity of natural constructions of multiple encryption

Given each component IND-CCA secure, let's consider the following problem, is above "natural" construction IND-ME-CCA secure? Rather disappointing, the answer is negative. There does exit insecure constructions.

Basic Idea. At the first glance, one may think all multiple encryption from such construction should be secure, since each component is chosen independently from each other and satisfies strong security notion IND-CCA, then all outputs will be indistinguishable from random sequence. However, this reasoning is fallacious. The flaw is in that this does not consider the case that the adversary can make use of the Decryption Oracle $\mathcal{DO}$. The Decryption Oracle $\mathcal{DO}$ can be very helpful in this case because every ciphertext different from the original can be decrypted and returned according to the definition of CCA attack. Then all the adversary needs to do is to modify the challenge ciphertext to a "new" one but decrypt to the same message, and submit it to the Decryption Oracle $\mathcal{DO}$. In an IND-CCA setting, the adversary cannot do this easily because the (whole) secret key is kept privately. However, in ME-CCA setting, partial key can be exposed by the Key Exposure Oracle $\mathcal{KE}$, moreover, since every component is semantically secure, as it must be probabilistic, where there exist at least two valid ciphertexts $C_0, C_1 \in \mathcal{C}$ with $Dec(C_0) = Dec(C_1) = M$, where $M \in \mathcal{M}$ is any valid plaintext. Furthermore, we have the following theorem.

Theorem 4.1 Multiple encryption may be totally *insecure* in ME-CCA model, even it is combined from independently chosen IND-CCA secure component ciphers and secure AONT.

Proof. Given a multiple encryption scheme $\mathcal{ME}$ constructed in the following way: independently take IND-CCA secure component ciphers $\mathcal{ME} = \{\mathsf{Enc}_i\}$, $i = 1, ..., n$, and connect them according to parallel or serial order, we have two claims:

Claim 1 *If a multiple encryption has a branch that ends with a parallel multiple encryption block, we are then able to construct an adversary $\mathcal{A}$ that breaks the it with only one key exposure query and one decryption query.*

Suppose $\mathcal{A} = (\mathcal{A}_{\mathsf{find}}, \mathcal{A}_{\mathsf{guess}})$ that chooses i, $1 \leq i \leq n$, and submits $\mathcal{E}_i$ to $\mathcal{KE}$. Denote (m_i, c_i) as the input and output of ith component cipher. Let $\mathcal{EO}$'s challenge be $C_b = \mathsf{MEnc}(M_b)$ $(b \overset{R}{\leftarrow} \{0,1\})$. We can construct the following adversary:

```
Adversary $\mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}$
    $(M_0, M_1, sk_i) \leftarrow \mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(pk, i)$
    $\alpha \leftarrow sk_i$
    return $(M_0, M_1, \alpha)$
```

```
Adversary $\mathcal{A}_{\mathsf{guess}}^{\mathcal{DO}}(M_0, M_1, \alpha, C_b)$
    $m_i \leftarrow \mathsf{Dec}_{i,sk_i}(c_i)$ where $C_b = (c_1, ..., c_i, ..., c_n)$
    For $c'_i = c_i$ do $c'_i = \mathsf{Enc}_i(m_i)$
    $C'_b = (c_1, ..., c'_i, ..., c_n)$
    $M_b = \mathsf{MDec}(C'_b)$ where $C'_b \neq C_b$
  return $M_b$
```

Claim 2 *If a multiple encryption has a branch that ends with a serial encryption block, we may then able to construct an adversary $\mathcal{A}$ that breaks the it with only one key exposure query on the last component and one decryption query.*

Observing that $\mathsf{Dec}_n(c_n) = c_{n-1}$ and $C = c_n$, we can build the adversary as follows:

```
Adversary $\mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}$
    $(M_0, M_1, sk_n) \leftarrow \mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(pk, n)$
    $\alpha \leftarrow sk_n$
    return $(M_0, M_1, \alpha)$
```

```
Adversary $\mathcal{A}_{\mathsf{guess}}^{\mathcal{DO}}(M_0, M_1, \alpha, C_b)$
    $c_{n-1} \leftarrow \mathsf{Dec}_{i,sk_n}(c_n)$ where $C_b = (c_1, ..., c_n)$
    For $c'_n = c_n$ do $c'_i = \mathsf{Enc}_n(c_{n-1})$
    $C'_b = c'_n$
    $M_b = \mathsf{MDec}(C'_b)$ where $C'_b \neq C_b$
  return $M_b$
```

where $\mathcal{EO}$'s challenge is $C_b = \mathsf{MEnc}(M_b)$ ($b \overset{R}{\leftarrow} \{0,1\}$).

We can see in both case, M_b can be returned by $\mathcal{DO}$, which enables the adversary to obtain b easily. Especially for some hybrid constructions, these two attacks can happen at the same time, where there is with a serial encryption block as a branch in its all branches. ■

Above theorem only shows the case of indistinguishability under ME-CCA attack, we also briefly explain the case of *onewayness* against ME-CCA, denoted as OW-ME-CCA. Onewayness is a strictly weaker notion than indistinguishability. However, theorem 4.1 tells us that not only IND-ME-CCA, but also OW-ME-CCA security *cannot* always be maintained in ME-CCA model, even if all the components are IND-CCA secure. On the other hand, we can see such natural schemes are malleable because the adversary can easily produce a "new" ciphertext with a proper key exposure query and simulates the encryption oracle. This result from another aspect implies that non-malleability is also essential in analyzing security of encryption schemes. NM-ME-CCA security better explains why the adversary in proving theorem 4.1 can launch that attack: it actually has produced a ciphertext with relation that it contains the same plaintext to the challenge ciphertext. It should be noted that NM-ME-CCA security is not trivially obtainable in such situations.

5 Securing multiple encryption in random oracle model

We have shown that the simple modular design without further treatment of multiple encryption is not sufficient to yield ME-CCA security. Another question is if multiple encryption satisfying IND-ME-CCA security can be achieved component ciphers with weaker security, e.g., OW-CPA security. In fact, this is possible, at least in the random oracle model.

In multiple encryption, ME-CCA security is hard to achieve with simple connections of component ciphers because partial exposure of the secret keys will always cause malleability of ciphertexts. This prompts us the necessity to check the redundancy used in encryption to ensure the validity of all parts of a ciphertext. Suppose all randomness used in the encryption can be verified during decryption, then the Decryption Oracle in fact will not help in a chosen ciphertext attack scenario, which is also the pith of several generic transforms from OW-CPA secure public key encryption schemes to IND-CCA secure schemes. Then what remains unsolved is how to combine a set of OW-CPA encryption schemes to have IND-ME-CCA secure multiple encryption. We notice that if consistence of randomness and the plaintext can be verified in the decryption, that is, the adversary must be forced to know the plaintext when it submits a ciphertext to the Decryption Oracle, then no ciphertext can be forged without breaking all underlying component ciphers.

5.1 Secure parallel construction of multiple encryption

We can build constructions based on any public key encryption components with OW-CPA security. Most of the practical public key encryption schemes satisfy this. Denote H_i and G_i as hash functions, which are regarded here as random oracles.

Key-Generation $\mathsf{MGen\text{-}Enc}(1^k)$**:** $(pk_i, sk_i) \leftarrow \mathsf{Gen\text{-}Enc}_i$; $pk = (pk_1, ..., pk_n)$, $sk = (sk_1, ..., sk_n)$.

Encryption $\mathsf{MEnc}(M)$**:** $m_1, ..., m_n \overset{\mathsf{AONT}}{\longleftarrow} \mathcal{T}(M)$, $r_i \in_R \{0,1\}^*$, $i = 1, ..., n$. For ith component cipher: $c_{i1} \leftarrow (\mathsf{Enc}_i(r_i; H_i(M, r_1, ..., r_n))$, $c_{i2} \leftarrow G_i(r_i) \oplus m_i$, $c_i = (c_{i1}, c_{i2})$, $1 \le i \le n$. Outputs $C = (c_1, ..., c_n)$ as ciphertext.

Decryption $\mathsf{MDec}(C)$**:** $r_i \leftarrow \mathsf{Dec}_i(\bar{c}_{i1})$, $\bar{m}_i = G(\bar{r}_i) \oplus \bar{c}_{i2}$, $1 \le i \le n$. Outputs $\bar{M} \leftarrow \mathcal{I}(\bar{m}_1, ..., \bar{m}_n)$ as plaintext if $\bar{c}_{i1} == \mathsf{Enc}(\bar{r}_i; H_i(\bar{M}, \bar{r}_1, ..., \bar{r}_n))$, otherwise "$\perp$".

5.2 Secure serial construction of multiple encryption

Serial construction can be based on the same idea. In the following constructions, $H_i : \{0,1\}^* \to \{0,1\}^{k_i}$ and $G_i : \{0,1\}^* \to \{0,1\}^{|c_{i2}|}$ are random oracles.

Key-Generation $\mathsf{MGen\text{-}Enc}(1^k)$**:** $(pk_i, sk_i) \leftarrow \mathsf{Gen\text{-}Enc}_i$; $pk = (pk_1, ..., pk_n)$, $sk = (sk_1, ..., sk_n)$.

Encryption $\mathsf{MEnc}(M)$**:** Let $c_0 = M$, $r_i \in_R \{0,1\}^*$, $i = 1, ..., n$. For ith component:
$c_{i1} \leftarrow (\mathsf{Enc}_i(r_i; H_i(c_{i-1}, r_1, ..., r_n))$, $c_{i2} = G_1(r_i) \oplus c_0$,...,
$r_n \in_R \{0,1\}^*$, $c_{n1} \leftarrow (\mathsf{Enc}_n(r_n; H_n(c_{n-1}, r_1, ..., r_n)), c_{n2} = G_n \oplus c_{n-1}$. Output $C = c_n$.

Decryption $\mathsf{MDec}(C)$**:** Let $\bar{c}_n = C$, for $1 \leq i \leq n$, $\bar{c}_{n-1} \leftarrow \mathsf{Dec}_i(\bar{c}_n)$. Outputs $\bar{M} = \bar{c_0}$ as output, if $\bar{c}_{i1} == \mathsf{Enc}(\bar{r}_i; H_i(\bar{M}, \bar{r}_1, ..., \bar{r_n}))$ for $1 \leq i \leq n$. Otherwise "$\perp$".

5.3 Security proof

We claim the following theorem holds for our construction:

Theorem 5.1 *Multiple encryption consists of parallel or serial block with above construction is secure* IND-ME-CCA *secure in random oracle model.*

Assume each component cipher is chosen independently. We claim the following lemmas:

Lemma 1 *If an adversary $\mathcal{B}$ breaks a parallel multiple encryption $\mathcal{ME}$ with the construction given the section 5.1, then there is an adversary $\mathcal{A}$ breaks onewayness of any component cipher $\mathcal{E}_i$ or* AONT *with non-negligible advantage.*

$\mathcal{B}$ breaks $\mathcal{ME}$ with probability $Succ_{\mathcal{B}}(k) = 1/2 + \varepsilon$ with adaptive Key Exposure Oracle that leaves at most $n-1$ keys to $\mathcal{B}$. Construct $\mathcal{A}$ as follows: $\mathcal{A}$ picks arbitrary encryption scheme $\mathcal{E}_i$ and a secure $(L, l, n) - \mathsf{AONT}$ and constructs $\mathcal{ME}$ as section 5.1. The adaptive key exposure is simulated as $\mathcal{A}$ chooses arbitrary $\mathcal{E}_j$ for $j \neq i$ and hand the secret keys to $\mathcal{B}$. This time since $\mathcal{B}$ knows all the secret keys, then there is no barrier for $\mathcal{B}$ to make decryption on c_js. $\mathcal{A}$ can simulate all this by itself.

When $\mathcal{B}$ asks encryption queries on a message M, $\mathcal{A}$ first transforms M with $(m_1, ..., m_n) \leftarrow \mathcal{T}(M)$ with AONT, specially $\mathcal{A}$ will take m_i as input for $\mathcal{E}_i$. $\mathcal{A}$ simulates random oracle H_i and G_i as two tables T_{H_i}, T_{G_i} by itself: if when $\mathcal{B}$ has a query σ_{count} on H_i, if it has not been entered as an entry in T_{H_i} it flips coins to get a random number increases the counter *count* (initially set 0) by 1, put the query and answer $(\sigma_{i,count}, h_{i,count})$ in the table and proceeds. It does the same for G_i where it instead puts the query $\sigma_{i,count}, m_{i,count}$ and the answer is $g_{i,count}$ in T_{G_i}. Then $\mathcal{A}$ simulates other random oracle H_j and G_j and gets output of $\mathcal{E}_j$ as $c_j = (c_{j1}, c_{j2})$.

When $\mathcal{B}$ makes decryption query on $C = (c_1, ..., c_n)$, $\mathcal{A}$ decrypts c_j such that $j \neq i$ to get $X{-}i = (m_1, ..., m_{i-1}, m_{i+1}, ..., m_n)$. Especially it runs the following program to get m_i and inverses $X = (m_0, ..., m_n)$ to get $M \leftarrow \mathcal{I}(M)$ and hand M to $\mathcal{B}$. Here, the program $K(\mathcal{T}_{H_i}, \mathcal{T}_{G_i}, c_i, pk_i)$ for $\mathcal{E}_i$, where on random oracle queries $\mathcal{T}_{H_i}, \mathcal{T}_{G_i}$, input ciphertext $c_i = (c_{i1}, c_{i2})$ and public key pk_i outputs the plaintext m_i if there is an entry in $\mathcal{T}_{H_i}$ satisfying $c_{i1} \leftarrow (\mathsf{Enc}_i(r_i; H_i(M, r_i))$, and an entry in $\mathcal{T}_{G_i}$ satisfying $c_{i2} \leftarrow G_i(r_i) \oplus m_i$.

First $\mathcal{A}$ runs $\mathcal{B}$ in the find model. When $\mathcal{B}$ makes encryption or decryption queries, $\mathcal{A}$ answers as described above. Finally, $\mathcal{B}$ halts automatically, outputs (M_0, M_1, s). Otherwise, if $\mathcal{B}$ cannot finish within $couter = q_{H_i} + q_{H_i}$ queries on H_i and G_i stop $\mathcal{B}$.

Let $b \leftarrow_R \{0,1\}$, an challenge ciphertext c_{i-b} is generated by an Encryption Oracle $\mathcal{EO}_i$ outside $\mathcal{A}$. Using the same b, $\mathcal{E}_i$ also generates $X{-}i = (m_1, ..., m_{i-1}, m_{i+1}, ..., m_n)$. Now $\mathcal{A}$ runs $\mathcal{B}$ in the guess mode taking $(m_{i-0}, m_{i-1}, s, X_{-i})$ as input. If $\mathcal{B}$ asks encryption or decryption queries, follow above specifications. At last, $\mathcal{B}$ outputs a guess bit $\tilde{b}$ on M_b. $\mathcal{A}$ also outputs b as its guess.

Claim 3 *If an* $\mathsf{IND\text{-}ME\text{-}CPA}$ *adversary* $\mathcal{B}$ *breaks parallel* $\mathcal{ME}$ *with advantage* ε*, there is* $\mathcal{A}$ *that breaks the indistinguishability of ith component cipher with probability epsilon*$_1$ *or indistinguishability of* $(L,l,n)-$ AONT *with advantage with advantage* ϵ_2*. Then* $\varepsilon \le \epsilon_1 + 2\epsilon_2$*.*

Proof. Denote some events as:

$AdvB$:	$\mathcal{B}$'s advantage on guessing b.
$E1$:	$\mathcal{B}$'s advantage to break the indistinguishability of AONT, that is, $\mathcal{B}$ guess b by only looking at (X_{-i}, M_0, M_1);
$E2$:	$\mathcal{B}$'s advantage to output m_{i-b} from (m_{i-0}, m_{i-1}) and C_b.

Since $E1$ and $E2$ are independent, and $Pr[AdvB|\neg E1 \wedge \neg E2]$ must be 0 from the assumption, let the advantage of $\mathcal{B}$ inverting c_{i-b} to get m_{i-b} be ϵ_1 and breaks AONT as ϵ_2, we have:

$$\begin{aligned}
Pr[AdvB] = \varepsilon &= Pr[Adv|E1 \wedge E2] \cdot Pr[E1 \wedge E2] + Pr[AdvB|\neg E1 \wedge E2] \cdot Pr[\neg E1 \wedge E2] \\
&\quad Pr[AdvB|E1 \wedge \neg E2] \cdot Pr[E1 \wedge \neg E2] + Pr[AdvB|\neg E1 \wedge \neg E2] \cdot Pr[\neg E1 \wedge \neg E2] \\
&\le Pr[E1 \wedge \neg E2] + Pr[\neg E1 \wedge E2] + Pr[\neg E1 \wedge \neg E2] \\
&\le \epsilon_1 + 2\epsilon_2
\end{aligned}$$

Completed. ■

Claim 4 *Suppose* $\mathcal{E}_i$ *is* γ*-uniform (detailed discussion in [14]). Let* $l_i = |c_{i2}|$*. If there is an adversary* $\mathcal{B}$ *breaks ith component cipher* $c_{i1} \leftarrow (\mathsf{Enc}_i(r_i; H_i(M, r_i))$*,* $c_{i2} \leftarrow G_i(r_i) \oplus m_i$*,* $c_i = (c_{i1}, c_{i2})$ *with* $(q_{H_i}, q_{G_i}, q_{d_i})$ *of* H_i, G_i *and decryption queries of advantage* ϵ_1*, then* $\mathcal{A}$ *breaks onewayness of* $\mathcal{E}_i$ *with advantage at least* $\varepsilon(1-2^{-k_i})^{q_{H_i}}(1-\gamma-2^{-l_i})^{q_d}$*.*

Proof. Denote the event $AskH_i$ is true if there is an entry in $\mathcal{T}_{H_i}$ satisfying $\mathsf{Enc}_i(r_i; H_i(M, r_1, ..., r_n)$, and $AskG_i$ is there is an entry in $\mathcal{T}_{G_i}$ satisfying $G_i(M, r_1, ..., r_n) \oplus m_i$. $SucA_0$ is true if $\mathcal{A}$ correctly guess b. $SucA_1$ is true if $\mathcal{A}$ simulates at most q_d decryption queries correctly. $SucA_2$ is true if on input unknown plaintext m_i, $\mathcal{A}$ outputs a correct ciphertext c_i. $fail1$ is true if $\mathcal{A}$ fails to simulate a specific $\mathcal{B}$'s decryption query.

From above specification, we know that $\mathcal{A}$ can simulate decryption queries for $\mathcal{B}$, for c_{i2} part is in fact one-time pad, the probability of $\mathcal{A}$ fails to simulate one decryption query of $\mathcal{B}$, since $AskH_i$ and $AskG_i$ is independent,

$$\begin{aligned}
Pr[fail1] &= Pr[fail1|AskH_i \wedge AskG_i] \cdot Pr[AskH_i \wedge AskG_i] \\
&\quad + Pr[fail1|\neg AskH_i \wedge AskG_i] \cdot Pr[\neg AskH_i \wedge AskG_i] \\
&\quad + Pr[fail1|AskH_i \wedge \neg AskG_i] \cdot Pr[AskH_i \wedge \neg AskG_i] \\
&\quad + Pr[fail1|\neg AskH_i \wedge \neg AskG_i] \cdot Pr[\neg AskH_i \wedge \neg AskG_i]
\end{aligned}$$

Since $Pr[fail1|AskH_i \wedge AskG_i]$ must be 0, $Pr[fail1|\neg AskH_i \wedge \neg AskG_i]$ must be 1, we have $Pr[fail1] \le Pr[fail1|\neg AskA_0] \cdot Pr[\neg AskA_0] \le \gamma + 2^{-l_i}$. So $Pr[SucA_1] = (1 - Pr[fail1])^{q_d} \ge (1 - \gamma - 2^{-l_i})^{q_d}$. On the other hand, $SucA_2$ fails when $\mathcal{B}$ make exactly query on r_i, denote the length of r_i to be $k_i = |r_i|$,

$$Pr[SucA_2] = (1 - 2^{-k_i})^{q_{H_i}}$$

Finally, from above specification of $\mathcal{A}$ we know $AdvB$, $SucA_1$ and $SucA_2$ are independent events. So the advantage $AdvA$ of $\mathcal{A}$ breaking onewayness of $\mathcal{E}_i$ using $\mathcal{B}$ as oracle is

$$\begin{aligned}
AdvA_{\mathcal{B}} &= Pr[AdvB \wedge SucA_1 \wedge SucA_2] = AdvB \cdot Pr[SucA_1] \cdot Pr[SucA_2] \\
&= \epsilon_1(1 - 2^{-k_i})^{q_{H_i}}(1 - \gamma - 2^{-l_i})^{q_d}
\end{aligned}$$

Proof completes. ■

Combining above two claims, we have $\mathcal{A}$ breaks onewayness of $\mathcal{E}_i$ with advantage at least:

$$AdvA \geq \min_{1 \leq i \leq n} \{(\varepsilon - 2\epsilon_2)(1 - 2^{-k_i})^{q_{H_i}}(1 - \gamma - 2^{-l_i})^{q_d}\}$$

Apparently both $\mathcal{A}$ and $\mathcal{B}$ can finish in polynomial time. By requirement of secure AONT, ϵ_2 is negligible. On the other hand we have assumed that no PPT adversary can break onewayness of $\mathcal{E}_i$, i.e., $AdvA$ is negligible. Lemma 1 is thus proved.

Lemma 2 *If an adversary $\mathcal{B}$ breaks a parallel multiple encryption $\mathcal{ME}$ with the construction given the section 5.2, then there is an adversary $\mathcal{A}$ breaks onewayness of any component cipher $\mathcal{E}_i$ with non-negligible advantage.*

Based on similar analysis, we can formulate the following:

Claim 5 *an adversary $\mathcal{A}$ can use an adversary $\mathcal{B}$ attacking* ME-CCA *with advantage ε to break the onewayness of a certain component cipher $\mathcal{E}_i$ with advantage at least* $\min_{1 \leq i \leq n}\{\varepsilon(1 - \cdot q_{H_i} \cdot 2^{-k_i})(1 - \gamma - 2^{-l_i})^{q_d}\}$.

The proof is quite similar to that of theorem 5, and is left to the readers. From above two lemmas, theorem 5.1 is proved.

One complementary remark should be addressed on the *uniformity* of underlying primitives [14]. What we have considered so far is mainly IND-CCA or IND-CPA encryption serving as component ciphers. For deterministic primitive public key encryption, e.g., RSA, above construction is not sufficient, however, it can be modified to fit this transform by using the same technique in [14]. Furthermore, if all the component ciphers are deterministic, the task is easier: just connect them together and set proper padding schemes as pre-procession of the message, like OAEP+ [28], and form the whole multiple encryption with parallel construction with compatible input domain, or serial connecting one after another. The AONT is even unnecessary because of OAEP+. This construction is also secure because if the encryption primitive is deterministic, an adversary cannot re-encrypt the corresponding parts of a ciphertext into valid new part to produce another ciphertext even if it seizes corresponding secret keys. We shall give formal analysis regarding the deterministic encryption primitive in the full version of this paper.

6 New definition regarding multiple encryption

It seems incredibly contradictive with our intuition that though component ciphers are independent, even onewayness may lose with just simple connection of independently chosen ciphers. However, if we follow the CCA security, it is doomed to appear completely insecure. From another aspect, it suggests ME-CCA is somehow excessively strong, though it is such a natural extension of CCA security. In real world, it is rare that a Decryption Oracle is available that provides such favors to an adversary. For example if a harmless bit is appended, a CCA-secure cipher S is no longer secure in the sense of CCA. It seems this should be easily judged and have "no significant difference" in most of cases. In fact, when the Decryption Oracle encounters such queries, it should easily determine whether this is really a "new" ciphertext, by just looking at the ciphertext.

6.1 A relaxed definition

We introduce a relation function $\mathcal{RF}$ to eliminate this definitional blemish. A relation function is such a function that if $\mathcal{RF}(C, C') = \mathsf{TRUE} \Rightarrow \mathsf{Dec}(C) = \mathsf{Dec}(C')$. We call (C', C) a qualified pair according to $\mathcal{RF}$. $\mathcal{RF}$ is called decryption respective, which will be part of the public key. The opposite direction does not hold otherwise the relation function can be used as an oracle breaking the indistinguishability. There must be $\exists$ (C, C'), such that $\mathcal{RF}(C, C') = \mathsf{FALSE}$, with $\mathsf{Dec}(C) = \mathsf{Dec}(C')$.

Definition 3 (IND-ME-gCCA) *In the beginning the of setup, the key generation algorithm* MEnc-Gen *is run, and with the input $\{1^k\}$, generating every underlying encryption scheme's public-secret key pair (pk_i, sk_i), n pairs in total. $pk = (pk_1, \ldots, pk_n)$ is the public key and $sk = (sk_1, \ldots, sk_n)$ is the secret key of $\mathcal{ME}$. Then* MEnc-Gen *gives the public key pk to the Encryption Oracle and the adversary, the secret key sk to Key Exposure Oracle $\mathcal{KE}$ and Decryption Oracle $\mathcal{DO}$ with a Relation Function $\mathcal{RF}$ inside, which is computable in polynomial time. The Exposure Oracle with the choice of the adversary, gives at most $n-1$ secret keys to the adversary. The adversary chooses two messages $\{M_0, M_1\}$ and sends them to the Encryption Oracle. The Encryption Oracle chooses $b \overset{R}{\leftarrow} \{0,1\}$ and encrypts $M_b \in \{M_0, M_1\}$ with some internal coin flipping using pk to get C_b. The adversary is allowed to submit any queries C to the Decryption Oracle. The Decryption Oracle returns the decryption result unless $\mathcal{RF}(C, C_b)$ does not output* TRUE*. The adversary may query the oracles adaptively, in any order he wants, subject to the restriction that he makes only one query to the Encryption Oracle. The adversary succeeds by guessing the value b, and a scheme is secure if any probabilistic polynomial time adversary has success negligibly close to 1/2.*

$$\Pr\left[b = \tilde{b} \,\middle|\, \begin{array}{c} (pk, sk) \leftarrow \mathsf{MEnc\text{-}Gen}(1^k), (M_0, M_1, \alpha) \leftarrow \mathcal{A}_{\mathsf{find}}^{\mathcal{KE}, \mathcal{DO}\neg\mathcal{RF}}(pk), \\ b \overset{R}{\leftarrow} \{0,1\}, C_b \leftarrow \mathsf{Enc}(M_b), \tilde{b} \leftarrow \mathcal{A}_{\mathsf{guess}}^{\mathcal{KE}, \mathcal{DO}\neg\mathcal{RF}}(C_b, \alpha) \end{array} \right] \leq \frac{1}{2} + \mathsf{neg}(k)$$

IND-ME-gCCA scheme can be easily acquired from IND-gCCA component ciphers. Generally, we have the following lemma:

Lemma 3 *A multiple encryption scheme is* IND-ME-gCCA *secure with respect to $\mathcal{RF}$ if all n component ciphers are* IND-gCCA *secure, where letting c_i and c'_i $1 \leq i \leq n$ denote the ciphertexts corresponding to component cryptosystem $\mathcal{E}_i$ for cihpertests C and C', respectively, $\mathcal{RF}$ is defined as $\mathcal{RF}(C, C') =$* TRUE *once $\exists$ $\mathcal{RF}_i(c_i, c'_i) =$* TRUE *for some i $1 \leq i \leq n$, and $\mathcal{RF}_i$ is done gradually inside $\mathcal{DO}$.*

Proof. Without loss of generality, we assume AONT is secure. It is easy to see within our definition of relation function, $\mathcal{RF}$ and $\mathcal{RF}_i$ are computable in polynomial time. If a $\mathcal{ME}$ scheme constructed from IND-gCCA components by above three construction methods is not IND-ME-gCCA secure, then we can use the IND-ME-gCCA adversary as an oracle to break the underlying IND-gCCA secure encryption schemes. For multiple encryption scheme, we denote "$\mathcal{RF}_i$" as equivalence relation w.r.t. any internal IND-gCCA secure component cipher $\mathcal{E}_i$. Now assume that $\mathcal{ME}$ is not IND-ME-gCCA secure w.r.t. $\mathcal{RF}$, we show that the same holds for $\mathcal{E}_i$ is not secure w.r.t. $\mathcal{RF}_i$, either. To do this, we take any adversary $\mathcal{D}$ for $\mathcal{ME}$ which contains $\mathcal{E}_i$ as internal component cipher and construct adversary $\mathcal{D}_i$ for $\mathcal{E}_i$.

When $\mathcal{D}_i$ views the public key pk_i of $\mathcal{E}_i$, it generates some key pairs $(pk_j, sk_j) \leftarrow \mathsf{Enc\text{-}Gen}_j(1^k)$ $(j \neq i)$ by itself, so that the inputs and outs are compatible. Without loss of generality, we denote the resulting cryptosystem as $\mathcal{ME}$ with $\mathcal{E}_i$ as one component cipher. The public key of $\mathcal{ME}$ is $(pk_1, \cdots, pk_i, \cdots, pk_n)$, and the secret key is $(sk_1, \cdots, sk_i, \cdots, sk_n)$. Only sk_i is unknown to $\mathcal{D}$. To simulate the decryption query Q_i made by $\mathcal{D}_i$, $\mathcal{D}$ checks that the respective Q is a valid query (or it will outputs $\perp$), and relation function outputs FALSE, then make query Q to his Decryption Oracle to decrypt Q. Next $\mathcal{D}_i$ outputs a pair (M_0, M_1) and also generate the corresponding pair (m_{i_0}, m_{i_1}) of proper intermediate output by those secret keys in hand. Then when $\mathcal{EO}_i$ generates a random challenge $c_{i_b} = \mathsf{Enc}_i(m_{i_b})$ for $b \in_R \{0,1\}$, $\mathcal{D}_i$ hands c_{i_b} to $\mathcal{D}$, who by itself complete a ciphertext C_b corresponding to the public key $(pk_1, ..., pk_n)$. By definition of the $\mathcal{RF}$ we know that $\mathcal{E}_i$ is forbidden to decrypt any $\mathcal{RF}_i(c_i, c'_i) =$ TRUE, i.e., $\mathcal{RF}(c_1, c_2) =$ TRUE, but this is the only limit that $\mathcal{D}_i$ is forbidden to ask its Decryption Oracle. So $\mathcal{D}$ can still feed the Decryption Oracle every single legal query. Finally, $\mathcal{D}_i$ outputs the same guess as $\mathcal{D}$ outputs, which enables $\mathcal{D}_i$ to succeed exactly with the same advantage as $\mathcal{D}$. ■

Since IND-CCA implies IND-gCCA, we further have the following theorem:

Theorem 6.1 *If all component ciphers are* IND-CCA *secure and chosen independently, then the resulting multiple encryption is* IND-ME-gCCA *secure.*

In fact, each attack per theorem 4.1 can construct an attack of malleable adversary to produce a new ciphertext with the same plaintext. Non-malleability is an arduous goal for all schemes of such kind, although the single component each is NM-CCA secure. For practical scheme usability, we also define gNM-ME-CCA analogously:

Definition 4 (gNM-ME-CCA) *A multiple encryption scheme is generalized-non-malleable against* ME-CCA *attack if for any PPT adversary, which is assisted by Decryption Oracle $\mathcal{DO}$, and a Key Exposure Oracle $\mathcal{KE}$, it cannot produce a new ciphertext with relation other than what the Relation Function $\mathcal{RF}$ specifies to the random challenge ciphertext generated by $\mathcal{EO}$ with non-negligible probability. Denote* $\mathbb{M}, \mathbb{C}$ *as sets of plaintexts and ciphertexts respectively.*

$$\Pr\left[b = 1 \left| \begin{array}{c} (pk, sk) \leftarrow \mathsf{MEnc\text{-}Gen}(1^k), (M_0, M_1, \alpha) \leftarrow \mathcal{A}_1^{\mathcal{KE},\mathcal{DO}}(pk), \\ C_b \leftarrow \mathsf{MEnc}(M_1), (R, \mathbb{C}) \leftarrow \mathcal{A}_2^{\mathcal{KE},\mathcal{DO}}(C_b, \alpha, M_0, M_1), \\ \mathbb{M} \leftarrow \mathsf{MDec}(\mathbb{C}), (C_b \notin \mathbb{C}) \wedge (\perp \notin \mathbb{M}) \wedge R(M_b, \mathbb{M}) \wedge (R \neq \mathcal{RF}) \end{array} \right.\right] \leq \frac{1}{2} + \mathsf{neg}(k)$$

gNM-ME-CCA is a relaxed notion to NM-ME-CCA security (cf. IND-ME-gCCA to IND-ME-CCA). Such We shall continue to discuss the relation between these security notions in next section.

7 Relations among security definitions for multiple encryption

In this section, we discuss the relation among security definitions of multiple encryptions. The good news is in multiple encryption scenario indistinguishability and non-malleability are still equivalent in most of the cases, namely under ME-CCA and ME-gCCA attacks.

Theorem 7.1 IND-ME-CCA $\Leftrightarrow$ NM-ME-CCA

Proof Idea. The idea is that one can construct an IND-ME-CCA adversary $\mathcal{A}$ who upon a challenge ciphertext C chosen randomly from two possible messages by using a NM-ME-CCA adversary $\mathcal{B}$ as an oracle to output another ciphertext C' and a relation of plaintexts of C' and C. Since $\mathcal{A}$ is executed in a CCA mode, then the new ciphertext can be submitted to the Decryption Oracle, who will return to $\mathcal{A}$ the corresponding plaintext M', with which and the relation $\mathcal{A}$ can recover the plaintext, and get correct guess on b. Denote $\bar{x}$ as bit-wise complement of x. On the other hand, if an IND-ME-CCA adversary can distinguish two chosen messages (M_0, M_1) with $M_1 = \bar{M_1}$, then we can always have the NM-ME-CCA adversary outputs a new ciphertext C_b' given $C_b = \mathsf{MEnc}(M_b)$ where $b \overset{R}{\leftarrow} \{0,1\}$, then it can output with $M_{\bar{b}} = \bar{M_b} = \mathsf{MDec}(C_b')$ satisfying relation complement R.
Proof. Without loss of generality, we assume the two challenge messages $M_0 \neq M_1$.

Lemma 4 NM-ME-CCA $\Rightarrow$ IND-ME-CCA.

Consider a NM-ME-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and IND-ME-CCA adversary $\mathcal{B} = (\mathcal{B}_{\mathsf{find}}, \mathcal{B}_{\mathsf{guess}})$, which $\mathcal{B}_{\mathsf{find}}$ chooses a pair of messages M_0, M_1 where $M_0 = \bar{M_1}$ and passes on to $\mathcal{B}_{\mathsf{guess}}$:

```
Adversary $\mathcal{A}_1^{\mathcal{KE},\mathcal{DO}}$
    $(M_0, M_1, s) \leftarrow \mathcal{B}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(pk)$
    $b \overset{R}{\leftarrow} \{0,1\}$
    $s' \leftarrow (M_0, M_1, pk, s)$
    return $M_b, s'$
```

```
Adversary $\mathcal{A}_2^{\mathcal{DO}}(M_b, s')$ where $s' = (M_0, M_1, pk, s)$
    $C_b \leftarrow \mathcal{B}_{\mathsf{guess}}^{\mathcal{DO}}(M_b, s)$
    $(C_b', R) \leftarrow \mathsf{MEnc}(\bar{M_b})$
    return $C_b', R$
```

It is obvious such adversary $\mathcal{A}$ succeeds in attacking IND-ME-CCA schemes at least the probability of an adversary $\mathcal{B}$ attacking NM-ME-CCA schemes, which contradicts our assumption.

Lemma 5 IND-ME-CCA $\Rightarrow$ NM-ME-CCA.

Consider a NM-ME-CCA adversary $\mathcal{A}$ and an IND-ME-CCA adversary $\mathcal{B}$:

```
Adversary $\mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}$
    $(M_0, M_1, s) \leftarrow \mathcal{B}_1^{\mathcal{KE},\mathcal{DO}}(pk)$
    $M \overset{R}{\leftarrow} \{0,1\}$
    $s' \leftarrow (M_0, M_1, pk, s)$
    return $M, s'$
```

```
Adversary $\mathcal{A}_{\mathsf{guess}}^{\mathcal{DO}}(M, s')$ where $s' = (M_0, M_1, pk, s)$
    $C'_b \leftarrow \mathcal{B}_2^{\mathcal{DO}}(M_0, M_1, s)$
    $(M'_b, R) \leftarrow \mathsf{MDec}(C'_b)$
    if $R(M'_b, M_0) = \mathsf{TRUE}$ then $d \leftarrow 0$
      else $d \leftarrow 1$
    return $d$
```

Then $\mathcal{A}$ succeeds with exactly the probability of $\mathcal{B}$, that means any scheme satisfying NM-ME-CCA security will satisfy IND-ME-CCA security. Combining above two lemmas, we complete the proof. ∎

Theorem 7.2 IND-ME-gCCA $\Rightarrow$ IND-ME-CPA*, however,* IND-ME-CPA $\not\Rightarrow$ IND-ME-gCCA.

Proof Idea. It is trivial of the former part, for a ME-CCA adversary is strictly stronger. On proof of the latter part, we just need to construct a counterexample. Suppose we have a multiple encryption scheme from a IND-ME-CCA secure multiple encryption schemes. If we append a special string to the public key. If special string is queried, Decryption Oracle returns the the secret key. However, this scheme still remains ME-CPA secure.

Proof. It is trivial to have: IND-ME-gCCA $\Rightarrow$ IND-ME-CPA. What left is to prove the following lemma:

Lemma 6 IND-ME-CPA $\not\Rightarrow$ IND-ME-gCCA.

Suppose $\mathcal{ME}' = (\mathsf{MEnc\text{-}Gen}', \mathsf{MEnc}', \mathsf{MDec}')$ is a IND-ME-CCA encryption scheme, we can modify it and build an new multiple encryption $\mathcal{ME}$ as follows:

```
MGen-Enc
    $(pk'_i, sk'_i) \leftarrow$MGen-Enc$'$, for $1 \le i \le n$;
    $pk' \leftarrow (pk'_1, ..., pk'_n)$, $sk' \leftarrow (sk'_1, ..., sk'_n)$
    $u \leftarrow \{0,1\}^k$
    $pk = u||pk'$, $sk = sk'$
    Return $(pk, sk)$
```

```
MEnc$(M)$
$c' \leftarrow \mathsf{MEnc}'(M)$
$C = 0||c'$
Return $C$
```

```
MDec$(C)$
    $v||\bar{c}' \leftarrow C$
    if $v = 0$
        Return $\mathsf{MDec}'_{sk'}(\bar{c}')$
    else if $\bar{c}' = u$
        Return $sk$
```

We can see $\mathcal{ME}$ is not ME-gCCA secure. For a challenge ciphertext C, the adversary can query the Decryption Oracle at $1||u$ to get sk then it can decrypt the challenge ciphertext by itself. Note that the relation function will fail to check this malicious query for $\mathcal{RF}(c', u) = \mathsf{FALSE}$ with overwhelming probability.

Claim 6 *Above encryption scheme ME is secure in the sense of* IND-ME-CPA.

Let C_b be the challenge ciphertext generated outside the adversary by an Encryption Oracle from one of a pair of messages $(M_0, M1)$, the adversary outputs its guess on b. Then denote the probability of following events as:

$$
\begin{aligned}
1 &:= [v = 0, (pk, sk) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0,1\}, MEnc(M_b) \leftarrow MEnc(M_b) : b = \bar{b}];\\
2 &:= [v = 1, (pk, sk) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0,1\}, MEnc(M_b) \leftarrow MEnc(M_b), c'_b \neq u : b = \bar{b}];\\
3 &:= [v = 1, (pk, sk) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0,1\}, MEnc(M_b) \leftarrow MEnc(M_b), c'_b = u : b = \bar{b}]
\end{aligned}
$$

Let the advantage of $\mathcal{B}$ attacking $\mathcal{ME}'$ be p_0, denote $k = |c'|$ as the length of c', the following holds:

$$
\begin{aligned}
Pr[AdvB] &= Pr[AdvB|1] \cdot Pr[1] + Pr[AdvB|2] \cdot Pr[2] + Pr[AdvB|3] \cdot Pr[3]\\
&\le Pr[AdvB|1] + Pr[AdvB|2] + Pr[AdvB|3]\\
&\le p_0 + p_0 + 2^{-k}
\end{aligned}
$$

It is easy to see $Pr[AdvB]$ is negligible. Proof completes. ∎

Theorem 7.3 ME-CCA $\Rightarrow$ ME-gCCA*, however,* ME-gCCA $\not\Rightarrow$ ME-CCA

Theorem 7.4 IND-ME-gCCA $\Leftrightarrow$ gNM-ME-CCA

Proof Idea. Since we have already proved IND-ME-CCA $\Leftrightarrow$ NM-ME-CCA, with the fact that the relation function in defining these two notions are the same, it is sufficient to show that a scheme meeting IND-ME-gCCA also meets gNM-ME-CCA while a scheme meet gNM-ME-CCA also meets IND-ME-gCCA security.

Proof. Denote two Relation Function in IND-ME-gCCA definition and gNM-ME-CCA definition as $\mathcal{RF}_{\mathsf{gIND}}$ and $\mathcal{RF}_{\mathsf{gNM}}$ respectively. S_{IND} and S_{NM} are the sets of schemes satisfy IND-ME-CCA and NM-ME-CCA respectively. Then if any scheme $s_i \in S_{\mathsf{IND}}$ then $s_i \in S_{\mathsf{NM}}$. Denote S_{gIND} and S_{gNM} as the sets of schemes satisfying IND-ME-gCCA and gNM-ME-CCA security respectively. Then it suffices$s_j \in S_{\mathsf{gNM}} \backslash S_{\mathsf{NM}}$, if $\forall s_j \in S_{\mathsf{gIND}} \backslash S_{\mathsf{IND}}$, and at the same time, $s'_j \in S_{\mathsf{gIND}} \backslash S_{\mathsf{IND}}$, if $\forall s'_j \in S_{\mathsf{gNM}} \backslash S_{\mathsf{NM}}$. We claim in these conditions, the adversary's power doesn't increase, that is, $\forall$ s_j and s'_j, we have an adversary that succeeds in attacking s_j will always succeeds in attacking s'_j and vice versa. Then denote adversary's query ciphertexts c_j and c'_j in gIND and gNM attacks respectively. Let c_i be the challenge ciphertext. $\mathcal{RF}_{\mathsf{gIND}}(c_i, c_j) = \mathsf{FALSE} \Rightarrow \mathcal{RF}_{\mathsf{gNM}}(c_i, c'_j) = \mathsf{FALSE}$ and vice versa. All left is then the same as proving equivalence of this pair of notions in ME-CCA model, we can easily have: if $\exists$ $s_j \in S_{\mathsf{gIND}} \backslash S_{\mathsf{IND}}$, there is always $s_j \in S_{\mathsf{gIND}} \backslash S_{\mathsf{IND}}$ and if $\exists$ $s'_j \in S_{\mathsf{gNM}} \backslash S_{\mathsf{NM}}$ there is always $s'_j \in S_{\mathsf{gIND}} \backslash S_{\mathsf{IND}}$.

Let's make the proof more easier to understand. Suppose an adversary $\mathcal{B}$ attacking scheme s_j in the sense of IND-ME-gCCA succeed with non-negligible advantage, then we can create an adversary $\mathcal{A}$ using $\mathcal{B}$ as oracle to attack the s_j with non-negligible advantage. Defining the generalized relation R is the same as the relation function $\mathcal{RF}$ in the ME-gCCA model. Now, let $\mathcal{A}$ run $\mathcal{B}$ in the first stage. If $\mathcal{B}$ asks for any decryption query, $\mathcal{A}$ passes it on to its Decryption Oracle. If there is any key exposure query questioned by $\mathcal{B}$, $\mathcal{A}$ also passes it to its Key Exposure Oracle. Specially, $\mathcal{A}$ can simulate the Encryption Oracle when $\mathcal{B}$ asks for encryption queries. After some steps $\mathcal{B}$ ends with side information and a pair of message. $\mathcal{A}$ outputs the same pair. Then outsides $\mathcal{A}$ a random bit b is chosen from $\{0, 1\}$ and M_b is encrypted by the Encryption Oracle. At the second stage, $\mathcal{A}$ runs $\mathcal{B}$ to get a new ciphertext C'_b with relation other than the relation specified in $\mathcal{RF}$ which is s_j's relation function. $\mathcal{B}$ may continue to ask encryption, decryption or key exposure queries according to the basic rule of a gNM-ME-CCA game. At last $\mathcal{B}$ outputs C'_b, $\mathcal{A}$ submit it to its Decryption Oracle, at the same advantage as $\mathcal{B}$, the Decryption Oracle will return it the plaintext. Thus it can get to know M_b.

From analogous discussion, we can also construct a gNM-ME-CCA adversary with exactly the same advantage as an IND-ME-gCCA adversary. This completes the proof. ■

8 Applications to key-insulated cryptosystem

8.1 Key-insulated cryptosystem

Key-insulated cryptosystem is recently proposed by [10] against partial key exposure. Computation is done in an insecure user device assuming the existence of physically secure server, which stores a master key. With the help of this server, user keys are updated periodically so that compromise of user keys in some periods does not affect the system in other periods. In [10], a generic construction based on arbitrary semantically secure public key encryption against chosen plaintext attack and cover-free family is proposed.

Generic Construction of [10]. The definition of cover-free family is given in Appendix A. First the key generation algorithm is run and u public key/secret key pairs of underlying semantically secure cryptosystems are generated, where $S_1, ..., S_N \subset [u] \overset{\text{def}}{=} \{1, ..., u\}$ is $\{t, 1/2\}$-cover-free family of n element sets. Any t subsets of secret keys do not contain other subsets. The underlying encryption scheme is semantic secure against chosen plaintext attack. The lifetime of the whole system is divided into N periods. Then the public key is $pk = (pk_1, ..., pk_u)$, and secret key of period i is $sk_i = \{sk_r : r \in S_i\}$, where $S_i = \{r_1, ..., r_n\}$. Specially the master key stored in a physically secure device will be $sk^* = \{sk_1, ..., sk_u\}$. We define the encryption of $M \in \{0,1\}^L$ at time period i as $C = \mathcal{E}_{pk}(i, M) = (i, \mathsf{Enc}_{pk_{r_1}}(m_1), ..., \mathsf{Enc}_{pk_{r_n}}(m_n))$ where, $(m_1, ..., m_n) \leftarrow \mathcal{T}(M)$ is generated from the real message M by a AONT $\mathcal{T}$. Decryption is done as: decrypt all the sub-messages $(m_1, ..., m_n)$ by $sk_{r_1}, ..., sk_{r_n}$ and synthesize the messages: $M = \mathcal{I}(m_1, ..., m_n)$.

The authors then claim that such system has key-insulated security with assumption of physically secure device holding sk^* and an adversary can at most obtain secret keys of t distinct periods. The authors define the security of the system in such a way that if no PPT adversary can break the indistinguishability of the any period i that is not compromised if it cannot obtain user secret keys for no more than t other periods even with the help of Key Exposure Oracle. In their proof, they show the whole system has indistinguishability of message in any period that is not compromised with at most t other periods compromised under chosen plaintext attack. However, their definition states the chosen ciphertext security of each period, the construction and analysis do not follow *adaptive* chosen ciphertext attack.

8.2 Chosen ciphertext attack on generic construction in [10]

One then may naturally think their construction is also secure against chosen ciphertext attacks if the underlying cryptosystems are $\mathsf{IND\text{-}CCA}$ secure. However, actually, we can show that their construction is *insecure* against chosen ciphertext attack which is defined by authors of [10].

At the fist look, because of the property of cover-free family even if adversary get t periods compromised, it can at most get $t-1$ secret keys of a new period that it doesn't compromise. Since the message is split into shares by AONT, we know it is still computationally infeasible to break the indistinguishability even after viewing part of the sub-messages generated by AONT. An adversary in fact can bypass the hard task of attacking the encryption and just needs to get the help of the Decryption Oracle. Typically an adversary is able to have any secret key sk_j by sending adaptive query to Key Exposure Oracle $\mathcal{KE}$ for sk_j other than i with $j \in S_i$. Then it can decrypt $c_j = \mathsf{Enc}_j(m_j)$, and re-encrypt it. It can always succeed to produce $c'_j = \mathsf{Enc}_j(m_j)$ with $c'_j \neq c_j$, since according to the system settings, since all component ciphers are semantically secure. Now the adversary can replace c_j with c'_j and submit this "new" ciphertext C' to the Decryption Oracle, which will return the corresponding message M. We can see their construction is no longer secure then!

Though their original generic construction does not satisfy chosen ciphertext attack security, actually by choosing every component $\mathsf{IND\text{-}CCA}$ secure, their generic construction is $\mathsf{IND\text{-}ME\text{-}gCCA}$ secure, which we believe is a very practical security definition. It can be proven easily using our analysis above: since natural construction of multiple encryption combined from any $\mathsf{IND\text{-}gCCA}$ components must be $\mathsf{IND\text{-}ME\text{-}gCCA}$ secure, and $\mathsf{IND\text{-}CCA}$ implies $\mathsf{IND\text{-}gCCA}$ unconditionally. We have all multiple encryption schemes combined from $\mathsf{IND\text{-}CCA}$ components are in fact $\mathsf{IND\text{-}ME\text{-}gCCA}$ secure (theorem 3). We evaluate that their scheme is still of practical meaning for conciseness.

8.3 A patch for generic construction of [10]

We are also fascinated at whether the construction in [10] can be turned $\mathsf{IND\text{-}ME\text{-}CCA}$ secure with minimum cost. In fact, using the technique given in section 5, it is possible. However, since their original construction requires $\mathsf{IND\text{-}CCA}$ components as building blocks versus CCA adversary, then their scheme is already $\mathsf{IND\text{-}ME\text{-}gCCA}$ secure referring to corollary 3, we can further have more efficient construction by applying a patch to their scheme. As the gap between $\mathsf{IND\text{-}ME\text{-}CCA}$ and $\mathsf{IND\text{-}ME\text{-}gCCA}$ is just that the adversary can lay a trap when asking the decryption queries, this can be immediately bridged once such attack is ruled out. For a secure multiple encryption must be probabilistic, i.e., there must be auxiliary randomness used in the encryption, to get multiple valid ciphertexts. If the Decryption Oracle can extract all randomness and verify it before out put a ciphertext, then the Decryption Oracle should be able to immune itself from such partial re-encryption attacks. We call such procession *randomness check*. If a ciphertext passes randomness check, then with overwhelming probability, the Decryption Oracle can assure that the sender of this ciphertext knows the corresponding plaintext, because for a public key multiple encryption $\mathsf{MEnc}_{pk}(M;\mathtt{COIN})$ scheme, where $\mathtt{COIN}$ stands for internal random coins, it should hold with probability 1 that $\mathsf{MEnc}_{pk}(M_1;\mathtt{coin}_1) \neq \mathsf{MEnc}_{pk}(M_2;\mathtt{coin}_1)$, when $M_1 \neq M_2$.

Again we consider the random oracle model, where a hash function is treated like a real random function. While keeping the unique mapping of input to the output, the output can also be regarded having uniform distribution. We are then considering adding such transforms into their scheme [10]: suppose the $\mathtt{coin}_i$ is the auxiliary random input by internal coin flipping for encryption component $\mathcal{E}i$, let $\mathtt{coin}_i = h(\mathtt{COIN}||Index_i)$, where $\mathtt{COIN}$ is a random number, $Index_i$ is the description of ith component and h is random oracle. The Encryption is $C = \mathsf{MEnc}(M||\mathtt{COIN};\mathtt{coin}_1,...,\mathtt{coin}_n)$, especially for IND-CCA component $\mathcal{E}_i$, $\mathsf{Enc}_i(m_i||\mathtt{coin}_i;\mathtt{coin}_i)$ where m_i is generated from AONT. Decryption process becomes: for a ciphertext C', $M'||\mathtt{COIN}' = \mathsf{MDec}(C')$, output M' if $C' = \mathsf{MEnc}(M'||\mathtt{COIN}';\mathtt{coin'}_1,...,\mathtt{coin'}_n)$, where $(\mathtt{coin'}_1,...,\mathtt{coin'}_n)$ are generated from $\mathtt{COIN}'$ as defined.

We shall only give the sketch of proof here. From above discussions, it is easy to see the modified scheme satisfying security definition of [10] under CCA attack. We point out this is actually the *first* generic construction of key-insulated cryptosystem enjoying CCA security. As pointed out, [10] contains no real proof on CCA security and in fact insecure for CCA. We also indicate that in [13], similar technique has been used to transform an IND-CPA encryption scheme into an IND-CCA scheme, with just $\mathsf{Enc}(M||\mathtt{COIN};h(M||\mathtt{COIN}))$. In fact, our transform states the transform of turning IND-ME-CPA to IND-ME-CCA.

References

[1] M. Abe and H. Imai. Flaws in some robust optimistic mix-nets. In *ACISP'03*, 2003, to appear.

[2] B. Aiello, M. Bellare, G. Di Crescenzo, and R. Venkatesan. Security amplification by composition: the case of doubly-iterated, ideal ciphers. In *Crypto'98*, volume 1462 of *LNCS*, pages 390–407. Springer-Verlag, 1998.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogway. Relations among notions of security for public-key encryption schemes. In *Advances in cryptology - Crypto'98*. Springer-Verlag, 1998.

[4] V. Boyko. On the security properties of the oaep as an all-or-nothing transform. In *Advances in Cryptology - Crypto'99*, volume 1666 of *LNCS*, pages 503–518. Springer-Verlag, 1999.

[5] Ran Canetti. Composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145, 2001.

[6] D. Chaum. Untraceable electronic mail, return address, and digitalpseudonyms. *Comm. of the ACM*, 24:84–88, 1981.

[7] Y. Desmedt. Society and group oriented cryptography: a new concept. In *Advances in Cryptology-Crypto'87*, volume 293 of *LNCS*, pages 120–127. Springer-Verlag, 1987.

[8] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology-Crypto'89*, volume 435 of *LNCS*, pages 307–315. Springer-Verlag, 1989.

[9] W. Diffie and M.E. Hellman. Exhaustive cryptananlysis of NBS data encryption standard. *IEEE Computer Magazine*, 10(6):74–84, June 1977.

[10] Y. Dodis, J. Katz, S. Xu, and M. Yung. Key-insulated public key cryptosystems. In *Advances in Cryptology - Eurocrypt'02*, volume 2332 of *LNCS*, pages 65–82. Springer-Verlag, 2002.

[11] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proceedings of the 23rd STOC*. ACM, 1991.

[12] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *SIAM Journal of Computing*, volume 30(2). ACM, 2000.

[13] E. Fujisaki and T. Okamoto. How to enchance the security of public-key encryption at minimum cost. In *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer-Verlag, 1999.

[14] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Crypto'99*, volume 1666 of *LNCS*, pages 537–554. Springer-Verlag, 1999.

[15] O. Goldreich. *Foundations of Cryptography: basic tools*, volume 1. New York, Cambridge University Press, ISBN 0-521-79172-3, 2001.

[16] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Science*, (28):270–299, 1984.

[17] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, and A. Juels. Optimistic mixing for exit-polls. In *Asiacrypt'02*, volume 2501, pages 451–465. Springer-Verlag, 2002.

[18] M. Jakobsson. A practical mix. In *Eurocrypt'98*, volume 1403 of *LNCS*, pages 448–461. Springer-Verlag, 1998.

[19] M. Juels and M. Jakobsson. An optimally robust hybrid mix network. In *Proc. of 20th annual ACM Symposium on Principles of Distributed Computation*, 2001.

[20] R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems. In *Crypto'99*, volume 1666 of *Springer-Verlag*, pages 609–623, 1999.

[21] U.M. Maurer and J.L. Massey. Cascade ciphers: The importance of being first. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 6(1):55–61, 1993.

[22] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to lgarithms in a finite field. *IEEE Trans. on Information Theory*, 39:1639–1646, 1993.

[23] R. Merkle and M. Hellman. On the security of multiple encryption. *Communications of the ACM*, 24(7):465–467, July 1981.

[24] NESSIE. NESSIE Portfolio of recommended cryptographic primitives (Latest version: Feb. 2003). Available at: `https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/decision-fin%al.pdf`.

[25] C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology-Crypto'91*, volume 576 of *LNCS*, pages 433–444. Springer-Verlag, 1991.

[26] R. Rivest. All-or-nothing encryption and the package transform. In *Proceedings of the 1997 Fast Software Encryption Conference*, volume 1267 of *LNCS*, pages 210–218. Springer-Verlag, 1997.

[27] C. Shannon. Communication theory of secrecy systems. In *Bell System Technical Journal*, volume 28, 1949.

[28] V. Shoup. OAEP reconsidered. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 239–259, 2001.

[29] V. Shoup. A proposal for an iso standard for public key encryption (version 2.1). Manuscript, 2001.

[30] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96, 2002.

[31] N. Smart. The discrete logarithm problems on elliptic curves of trace one. *Journal of Cryptology*, 12:193–196, 1999.

[32] Y. Watanabe, J. Shikata, and H. Imai. Equivalence between semantic security and indistinguishability against chosen ciphertext attacks. In *PKC 2003*, volume 2567 of *LNCS*, pages 71–84, 2003.

Appendix A: Figures

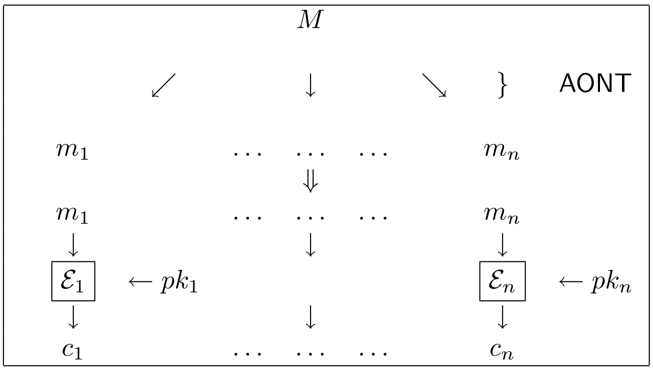

Figure 1: Parallel construction of multiple encryption

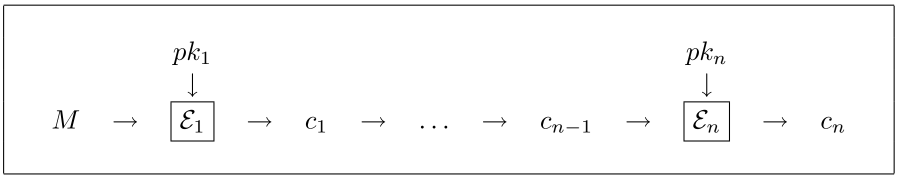

Figure 2: Serial construction of multiple encryption