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

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

September 20, 2003

Abstract

In a practical system, a message is often encrypted more than once by different encryptions, here called multiple encryption, to enhance its security. Additionally, new features may be achieved by multiple encrypting a message for a scheme, such as the key-insulated cryptosystems [13] and anonymous channels [8]. Intuitively, a multiple encryption should remain "secure", whenever there is one component cipher unbreakable in it. In NESSIE's latest Portfolio of recommended cryptographic primitives (Feb. 2003), it is suggested to use multiple encryption with component ciphers based on different assumptions to acquire long term security. However, in this paper we show this needs careful discussion. Especially, this may *not* be true according to (adaptive) chosen ciphertext attack (CCA), even with all component ciphers CCA secure. We define an extended version of CCA called *chosen ciphertext attack for multiple encryption* (ME-CCA) to emulate real world partial breaking of assumptions, and give constructions of multiple encryption satisfying ME-CCA security. Since CCA security seems so stringent, we further relax it by introducing *weak* ME-CCA (ME-wCCA), and prove IND-ME-wCCA secure multiple encryption can be acquired from IND-gCCA secure component ciphers. We also study the relation of various security notions for multiple encryption. We then apply these results to key-insulated cryptosystem. It is only previously known in [13] that a generic construction exists provably secure against CPA attack, however, we prove that this generic construction is in fact secure against ME-wCCA by choosing all components IND-CCA secure. We also give an efficient generic construction of key-insulated cryptosystem, which is so far the *first* generic construction provably secure against CCA (in the random oracle model).

key words: multiple encryption, CCA security, key-insulated cryptosystem

1 Introduction

A practical cryptosystem often encrypts a message several times under encryption schemes with independent secret keys or even distinct ciphers based on different assumptions to enhance the plaintext confidentiality. We call such cryptosystems multiple encryption, specifically double encryption and triple encryption for two times and three times multiple encryptions. In this paper, we investigate the security notion of multiple encryption against partial breaking of underlying assumptions.

WHY MULTIPLE ENCRYPTION. It is widely believed that multiple encryption provides better security because even if underlying assumptions of some component ciphers are broken or some of the 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 [27, 16] and on prime-field anomalous curves [33, 38, 32] have already reminded us the necessity to do

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

†Yokohama National University. Email: shikata@ynu.ac.jp

this. Especially, for example, it is suggested by NESSIE ([30], pp. 5, line 7-11) 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*". However, it seems that this needs more careful considerations.

On the other hand, multiple encryption can bring additional favorable features to a scheme. Combination of ordinary threshold encryptions may yield new threshold encryption with various access structures. Implementations achieving sender anonymity such as Mix-net [8], onion routing [8, 23], and the key-insulated cryptosystems [13] are all practical examples of multiple encryptions.

Contradiction to the Intuition. 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 CCA with partial component ciphers broken. This contradicts our intuition at the first sight, but many natural constructions of multiple encryption from combinations of IND-CCA secure components can be shown easily to lose the CCA security. Meanwhile, this result may imply CCA-security is too strong because practical schemes with "pretty good" security could be overkilled. Then we propose a generic construction of multiple encryption scheme achieving CCA security exactly. In emphasizing "natural" constructions' practical usability, we relax the CCA security. We then investigate the relations among security notions for multiple encryption. Finally as a byproduct, we give the first generic construction of CCA secure key-insulated cryptosystem.

1.1 Related work

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

Multiple Encryption and Related Primitives. Multiple encryption has been used in many practical schemes, for instance Triple DES. Recently, NESSIE [30] has also announced its recommendation to use (public key) multiple encryption under diverse assumptions to ensure long term security. Another example is the key-insulated cryptosystem, proposed by Dodis, Katz, Xu and Yung [13]. In such systems, with multiple encryption of messages under a number of keys from cover free family [25] and 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 channels in open network, such as Mix-net [23] and onion routing [8]. 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 Mix-net protocols [24, 21], which if translated in our language, have used insecure multiple encryption.

A similar notion to multiple encryption is the threshold cryptosystem [9, 10, 37], which maintains secrecy of the unique decryption key even if some shares of the secret key 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. Standard security definition of a public key encryption scheme is founded gradually in literature, e.g. [20, 29, 14, 31, 4, 15] and the strongest security notion turns to be indistinguishability against (adaptive) chosen-ciphertext attack (IND-CCA). *Semantic security*, first defined by Goldwasser and Micali [20], later refined by Goldreich [18, 19] and Watanabe, Shikata and Imai [39], captures the computational approximation of Shannon's information-theoretic security [34], 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 proven to be equivalent to semantic security in several attack models, namely chosen plaintext attack (CPA), (non-adaptive) chosen-ciphertext attack (CCA1) [29] and adaptive chosen-ciphertext attack (CCA2) [20, 18, 39, 19]. Another intricate notion, *non-malleability*, first defined by Dolev, Dwork and Naor [14, 15] and later refined by Bellare and Sahai [4, 5], formulates that the adversary should not be able to create a ciphertext of a different message that is meaningfully related to the original ciphertext. Non-malleability implies indistinguishability in all above three attack models. Independently in [4] and [15], indistinguishability and non-malleability are proven to be equivalent under (adaptive) chosen-ciphertext attack (hereafter CCA).

CCA security is crucial in analyzing security of protocols in the universal composability framework [6, 22, 7]. Mainly it allows the adversary can access the decryption oracle even after receiving a challenge ciphertext. However, Shoup first argues CCA security is too stringent for practical schemes and suggests "benign malleability" as a relaxation for CCA in the proposal for ISO public key encryption standard [36]. An, Dodis and Rabin [3] give similar discussion under the name "generalized-CCA" (gCCA). In these two relaxed definitions, a relation function checks and rejects "obvious" decryption queries decrypted to the target message. Canetti, Krawczyk and Nielsen [7] also propose another relaxation, namely "Replayable CCA", which is weaker than gCCA in most of cases.

Previous Work on Multiple Encryptions and Relations. Multiple encryption was addressed by Shannon as early as [34] under the name "product cipher", and in [11, 28, 2] in context of symmetric key cryptosystems. Massay and Maurer [26] have also studied the problem under the name "cascade cipher". However, all above work lacks considerations for CCA security and is not adequate for applying their underlying notions to public key setting straightforwardly, even only to the sequential case (see below).

In upcoming work of [12], Dodis and Katz, independently of our work, propose another generic construction of CCA secure multiple encryption. The security of their scheme can be proven in the standard model and they generate their scheme to various applications, such as key-insulated cryptosystem, threshold encryption and etc..[1]

1.2 Our contributions

Our contributions lie in following aspects:

Model and Security Definition of Multiple Encryption. We give the first formal model regarding public key multiple encryption. To the best of our knowledge, no previous work has strict formalization including CCA security, 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 ciphers to have a "secure" multiple encryption. As a theoretical extension of traditional security definitions, we give the corresponding security definition formulated by indistinguishability and non-malleability, especially against *chosen ciphertext attack for multiple encryption* (ME-CCA). We introduce a *Key Exposure Oracle* to emulate security of multiple encryption in the real world even when underlying assumptions are partially broken. Without loss of generality, breaking underlying assumptions of component ciphers can be esuriently modelled as the secret key is leaked to the adversary. Note that there should be at least one secret key hidden from the adversary, while underlying cryptosystems can be selected independently

[1] So far they only present their scheme in Rump Session in Crypto'03, Aug. 2003, while an earlier version of our work was publicly announced in [40], Jan. 2003.

(the keys can be independent). We note this security definition considers more than the key exposure problem. Choosing multiple encryption on different assumptions is the most generalized form of multiple encryption with more favorable confidentiality protection, guaranteeing maximum damage in case of partial breaking. Some analyses here can be applied to symmetric key schemes also.

VULNERABILITY OF NATURAL MULTIPLE ENCRYPTION. We demonstrate generic attacks against some natural construction of multiple encryption schemes with each component IND-CCA secure, by an adversary that breaks the indistinguishability of the scheme with only accesses to the Decryption Oracle and the Key Exposure Oracle. In fact, such adversary even breaks the onewayness of the scheme. This also explains that multiple encryption should be treated as a separate primitive from single-layered encryption.

SECURE CONSTRUCTION OF MULTIPLE ENCRYPTION. We exhibit a generic construction of secure multiple encryption from component ciphers satisfying only "weak" security, e.g., CPA. Though this can be achieved using general zero-knowledge proof techniques, considering efficiency and practicality, we adopt a scheme that is provably secure in the random oracle model.

RE-DEFINING SECURITY OF MULTIPLE ENCRYPTION. IND-CCA security has been treated as standard definition for encryption schemes, as this is convenient to have modular design on cryptographical protocols in the universal composability framework [6]. However, our analysis shows CCA security may be too stringent as even combining all IND-CCA secure component ciphers, it might result in a CCA insecure multiple encryption. As a reasonable relaxation, we give a new security definition named *weak chosen ciphertext attack for multiple encryption* (ME-wCCA) that is sufficient in most of interesting cases.

SECURITY NOTIONS OF MULTIPLE ENCRYPTION. We also study the relations between different security definitions for multiple encryption. We formulate the security definitions, namely indistinguishability and non-malleability, under different attack models. We show indistinguishability and non-malleability are still equivalent under ME-CCA and ME-wCCA, which corresponds to previous results (A multiple encryption degenerates to an ordinary public key cryptosystem, if there is only one component cipher in it.). We believe a good analysis of these relations will help protocol designer more than simply give a specific construction based on concrete mathematical assumptions.

APPLICATION TO KEY INSULATED ENCRYPTION. As an application, we reconsider the chosen ciphertext security for generic construction of key-insulated encryption proposed by Dodis, Katz, Xu and Yung [13]. It is only previously known in [13] that a generic construction exists provably secure against CPA attack. In this paper, we show that their scheme is in fact provably secure in the relaxed wCCA model, if each component cipher is selected IND-CCA secure. This result reasonably supports the correctness and practical usability of the scheme in [13]. We further give a generic construction meeting exact CCA security (in the random oracle model). We point out this is the first generic construction of CCA secure key-insulated cryptosystem so far.

2 The model

In this section, we give the model of a multiple encryption, basic construction methods and relative security definitions. Multiple encryption is a generalized form of public key encryption. Definitions for negligible function, public key encryption scheme, All-or-Nothing Transform and Cover-free family are given in Appendix A.

2.1 Multiple encryption scheme

Informally a multiple encryption is to encrypt a message by multiple cryptosystems. A multiple encryption scheme $\mathcal{ME}$ is generated by component ciphers. Naturally we have two basic combinations of these cryptosystems: parallel and sequential connection among different components.

2.1.1 Definition

Multiple encryption is a cryptosystem composed by distinct component ciphers. Suppose $\{\mathcal{E}_i\}_{1\leq i\leq n}$ is a set of *compatible* component ciphers, where for $\mathcal{E}_i$,

- $\mathsf{Enc\text{-}Gen}_i$ a probabilistic key-generation algorithm, with the input (1^k) and the internal coin flipping produces a public-secret key pair (pk_i, sk_i);
- Enc_i an encryption algorithm, with an input message $m_i \in \mathcal{M}_i$ and the public key pk_i, with the internal coin flipping, outputs a ciphertext $c_i \in \mathcal{C}_i$;
- Dec_i a decryption algorithm, which is a deterministic algorithm, with the input ciphertext c_i and the secret key sk_i, outputs a message m_i or "$\perp$".

A multiple encryption is a 3-tuple algorithm (MEnc-Gen,MEnc,MDec), where each algorithm may be combined from a number of public key cryptosystems with a unifilar connecting order. MEnc-Gen invokes every $\mathsf{Enc\text{-}Gen}_i$, and writes their outputs to a key list with public keys $PK = (pk_1, ..., pk_n)$ and secret keys $SK = (sk_1, ..., sk_n)$. MEnc with an input message M from message space $\mathcal{M}$ and PK, performs encryption MEnc on M by invoking a list of component encryption algorithms, also including AONT $\mathcal{T}$ if necessary, eventually outputs a ciphertext $C \in \mathcal{C}$. The decryption algorithm MDec takes (C, SK) as input and outputs M, or "$\perp$" if C is invalid. We also denote in brief the encryption algorithm as $\mathsf{MEnc}(M; \mathtt{COIN})$ (or $\mathsf{MEnc}(M)$), and the decryption algorithm as $\mathsf{MDec}(C)$ in clear context, where COIN stands for the randomness used the multiple encryption. Essentially, we have two basic constructions: parallel and sequential.

Parallel Construction. A parallel multiple encryption is an operation that messages are 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 lose immediately - 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) (Certainly a $(n-1, n)$ secret sharing also suffices.), which maps the desired message into several sub-messages so that only after all the sub-messages are decrypted and the plaintext can be recovered. Figure 1 depicts the construction in Appendix B.

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 [13] is an example of multiple parallel encryption.

Sequential Construction. Sequential multiple encryption is more straightforward, with the structure identical to cascade cipher mentioned in [26]. It should be clarified that there exists significant difference between multiple sequential encryption and the product cipher [34]: for multiple encryption, each component cipher scheme can be chosen independently. Initially the plaintext is encrypted by the innermost component cipher. Each output (ciphertext) of an component cipher will be passed on as the input of the next component cipher. Finally the output of the last component cipher is taken as the output of this multiple encryption. Figure 2 in Appendix B depicts it. 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 [8] is an example of multiple sequential encryption.

Hybrid Construction. If a multiple encryption contains both parallel encryption block and sequential encryption block, we call it a hybrid multiple encryption. We give another description that may help understand the structure. Consider a cipher cryptosystem with a tree structure. Fixing the root node as the first layer cipher, adding a parallel multiple encryption to a node just increases the sub-nodes of a node into e, where e is the number of component ciphers in this parallel block. Adding a sequential cipher cryptosystem to a node will increase the tree depth with a factor of f from that node, where f is the number of component ciphers in this sequential 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 sequential block. Then a multiple encryption ends with a parallel branch if there is one parallel encryption block in any branch, and ends with sequential branch if there is only one branch, with its all component ciphers forming a sequential encryption block.

2.1.2 Parallel construction vs. sequential construction

Parallel multiple encryption may serve as a secure data storage where a document is split into n pieces with (t, n) threshold 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, sequential multiple encryption has gain in the data size.

2.2 Chosen ciphertext security for multiple encryption

Partially breaking of underlying assumptions (key exposure) is usually not considered in the security of a normal public key encryption scheme, such as $\mathsf{IND\text{-}CCA}$, whereas a multiple encryption should remain secure even when most of the underlying assumptions are broken. Since this gap cannot merge sometimes, modifications should be performed to the standard CCA security definition in order to catch this act. We here introduce an additional oracle into standard CCA game to emulate this scenario: a Key Exposure Oracle that upon the adaptive request of the adversary, leaks secret keys of the component ciphers to the adversary. Note that more has been considered in our model than mere key exposure and the situations are more complicated.

Oracle Access Rules. There are three oracles in our model: An Encryption Oracle $\mathcal{EO}$, which upon calling with input (M_0, M_1), returns C_b, the encryption of M_b, where $b \in \{0, 1\}$ decided by internal coin flipping. A Decryption Oracle $\mathcal{DE}$, upon decryption query C, outputs $M = \mathsf{MDec}(C)$, if $C \neq C_b$; otherwise, "$\perp$". A Key Exposure Oracle, upon calling with i as one index of entire n component ciphers, $1 \leq i \leq n$, returns the corresponding secret key sk_i. The adversary can access three oracles in any order at any time of its choice, but it can only query $\mathcal{EO}$ once and $\mathcal{KE}$ at most $n - 1$ times.

Definition 1 ($\mathsf{IND\text{-}ME\text{-}CCA}$) *Assume any PPT adversary play the following game with a multiple encryption $\mathcal{ME}$. Key generation algorithm $\mathsf{MEnc\text{-}Gen}$ is run. 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 chooses to access the three oracles in any order and at any time. According to the timing of access to $\mathcal{EO}$, the adversary's strategy is divided into two algorithms $(\mathcal{A}_{\mathsf{find}}, \mathcal{A}_{\mathsf{guess}})$, where $\mathcal{A}_{\mathsf{find}}$ tries to find (M_0, M_1) to submit to $\mathcal{EO}$ which returns C_b, and $\mathcal{A}_{\mathsf{guess}}$ tries to output a guess 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:*

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

then we call this $\mathcal{ME}$ $\mathsf{IND\text{-}ME\text{-}CCA}$ secure.

Non-malleability of multiple encryption against CCA ($\mathsf{NM\text{-}ME\text{-}CCA}$) is similar to $\mathsf{IND\text{-}ME\text{-}CCA}$ except that the adversary succeeds by outputting a new ciphertext with is "meaningfully" related to the challenge ciphertext. That is, suppose R is a prescribed relation, then the adversary wins, if the adversary could output a different ciphertext C' from the challenge ciphertext C_b, with two plaintexts decrypted from C' and C_b satisfying R (R outputs TRUE).

Definition 2 (NM-ME-CCA) *Denote* $\mathbb{M}, \mathbb{C}$ *as sets of plaintexts and ciphertexts being empty initially, respectively. According to the above access rules for the three oracles, if any probabilistic polynomial time adversary in the following game has success probability negligibly close to 1/2, we call the multiple encryption scheme* NM-ME-CCA *secure.*

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

These definitions are also applicable to chosen plaintext attack CPA by letting $\mathcal{DO}$ always output an empty string on any decryption query, which results in the definition of *chosen plaintext attack for multiple encryption* ME-CPA. Analogously, we can define IND-ME-CPA, NM-ME-CPA. By fixing the number of component ciphers $n = 1$ in the dedition of IND-ME-CCA (or NM-ME-CCA), we obtain definition of the standard IND-CCA (or NM-CCA).

3 Insecurity of natural constructions

Given each component IND-CCA secure, let's consider the following problem: Is the 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 schemes 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 $\mathcal{DO}$. In this case $\mathcal{DO}$ can be very helpful 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 the CCA setting, the adversary cannot do this easily because the 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 $\mathsf{MDec}(C_0) = \mathsf{MDec}(C_1) = M$, where $M \in \mathcal{M}$ is any valid plaintext. Furthermore, we have the following theorem.

Theorem 1 There exists *insecure* multiple encryption in the sense of IND-ME-CCA, even if 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 select IND-CCA secure component ciphers $\mathcal{ME} = \{\mathsf{Enc}_i\}$, $i = 1, ..., n$, combine them according to the three constructions and generate public key $PK = (pk_1, ..., pk_n)$ and secret key $SK = (sk_1, ..., sk_n)$ (see section 2.1.1). We have two claims:

Claim 1 *If a multiple encryption has a branch that ends with a parallel block, we are then able to construct an adversary* $\mathcal{A}$ *that breaks 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 i-th component cipher. Let $\mathcal{EO}$'s challenge be $C_b = \mathsf{MEnc}(M_b)$ ($b \xleftarrow{R} \{0,1\}$). We can construct the following adversary:

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

Claim 2 *If a multiple encryption has a branch that ends with a sequential block, we may then be able to construct an adversary $\mathcal{A}$ that breaks 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 $\mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}$	Adversary $\mathcal{A}_{\mathsf{guess}}^{\mathcal{DO}}(M_0, M_1, \alpha, C_b)$
$(M_0, M_1, sk_n) \leftarrow \mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(PK, n)$	$c_{n-1} \leftarrow \mathsf{Dec}_{i,sk_n}(c_n)$ where $C_b = (c_1, ..., c_n)$
$\alpha \leftarrow sk_n$	For $c_n' = c_n$ do $c_i' = \mathsf{Enc}_n(c_{n-1})$
return (M_0, M_1, α)	$C_b' = c_n'$
	$M_b = \mathsf{MDec}(C_b')$ where $C_b' \neq C_b$
	return M_b

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

We can see in both case, M_b can be decrypted by querying $\mathcal{DO}$ with C_b', which enables the adversary to obtain b easily. Especially for some hybrid constructions, these two attacks can happen at the same time. ■

Discussion. The proof to this theorem shows only the case of indistinguishability under ME-CCA attack. We briefly explain the case of *onewayness* against chosen ciphertext attack for multiple encryption, denoted as OW-ME-CCA. Onewayness can be informally described as: given ciphertext C, output the plaintext M. It is a strictly weaker notion than indistinguishability. However, the proof of Theorem 1 tells us that not only IND-ME-CCA, but also onewayness may *not* be maintained in ME-CCA model, even if all the components are 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. NM-ME-CCA security better explains why the adversary can launch that attack: it actually has produced a ciphertext with relation that it contains the same plaintext to the challenge ciphertext. NM-ME-CCA security is not trivially obtainable in such situations, either.

4 A generic construction for secure multiple encryption

We have shown that the simple modular design without further treatment of multiple encryption is not sufficient to yield ME-CCA security. Then it is natural to consider the following questions: First, how to construct a ME-CCA secure multiple encryption. Second, whether a generic construction satisfying ME-CCA security can be achieved by component ciphers with weaker security, e.g., *onewayness against chosen plaintext attack* (OW-CPA) security. We answer both questions by giving a generic construction achieving ME-CCA security with component ciphers with weaker security.

For the "natural" constructions, 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 randomness used in encryption to ensure the validity of all parts of a ciphertext before outputting the plaintext. Suppose all randomness used in the encryption can be verified during decryption, then the Decryption Oracle in fact does not help the adversary: If the adversary can pass the randomness verification, with overwhelming probability, it has already known all the randomness used. This can be achieved by embedding all randomness into the plaintext. Consistence of all randomness can be verified in the decryption phase, i.e., to pass the test, the adversary must be forced to have known the corresponding plaintext when it submits a ciphertext query. Then a multiple encryption will be secure if an adversary cannot break all underlying component ciphers. Then what remains to be solved is how to combine a set of OW-CPA encryption schemes to have IND-ME-CCA secure multiple encryption.

Recall $\mathcal{E}_i$ is the i-th component cipher of the multiple encryption, $\mathsf{Enc}_i(m_i, pk_i; \mathtt{COIN}_i)$ and $\mathsf{Dec}_i(c_i, sk_i)$ are the encryption algorithm and decryption algorithm for $\mathcal{E}_i$ (in short $\mathsf{Enc}_i(m_i; \mathtt{COIN}_i)$ and $\mathsf{Dec}_i(c_i)$, respectively), where pk_i is the public key and sk_i is the secret key of $\mathcal{E}_i$ (see section 2.1.1).

4.1 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 $H_i : \{0,1\}^* \to \{0,1\}^{k_i}$ (k_i is the length of necessary random coin for $\mathcal{E}_i$) and $G_i : \{0,1\}^* \to \{0,1\}^{l_i}$ (l_i is the length of c_{i2}) as random functions.

Key-Generation $\mathsf{MGen\text{-}Enc}(1^k)$**:** $(pk_i, sk_i) \leftarrow \mathsf{Gen\text{-}Enc}_i$, for $1 \le i \le n$; $PK = (pk_1, ..., pk_n)$, $SK = (sk_1, ..., sk_n)$.

Encryption $\mathsf{MEnc}(M, PK)$**:** $(m_1, ..., m_n) \stackrel{\mathsf{AONT}}{\longleftarrow} \mathcal{T}(M)$. $r_i \in_R \{0,1\}^*$, for $1 \le i \le n$. For i-th 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 $\mathsf{MDec}(C, SK)$**:** $r_i \leftarrow \mathsf{Dec}_i(\bar{c}_{i1})$, $\bar{m}_i = G(\bar{r}_i) \oplus \bar{c}_{i2}$, $1 \le i \le n$. Outputs $\bar{M} \leftarrow \mathcal{I}(\bar{m}_1, ..., \bar{m}_n)$ as plaintext if $\bar{c}_{i1} = \mathsf{Enc}_i(\bar{r}_i; H_i(\bar{M}, \bar{r}_1, ..., \bar{r}_n))$, otherwise "$\perp$".

4.2 Secure sequential construction of multiple encryption

Sequential construction can be based on the same idea. In the following constructions, $H_i : \{0,1\}^* \to \{0,1\}^{k_i}$ (k_i is the length of necessary randomness for $\mathcal{E}_i$) and $G_i : \{0,1\}^* \to \{0,1\}^{l_i}$ (l_i is the length of c_{i2}) are random functions.

Key-Generation $\mathsf{MGen\text{-}Enc}(1^k)$**:** $(pk_i, sk_i) \leftarrow \mathsf{Gen\text{-}Enc}_i$, for $1 \le i \le n$; $PK = (pk_1, ..., pk_n)$, $SK = (sk_1, ..., sk_n)$.

Encryption $\mathsf{MEnc}(M, PK)$**:** Let $c_0 = M$, $r_i \in_R \{0,1\}^*$, for $1 \le i \le n$. For i-th component:
$c_{i1} \leftarrow \mathsf{Enc}_i(r_i; H_i(c_{i-1}, r_1, ..., r_n))$, $c_{i2} = G_1(r_i) \oplus c_0$, $c_i = (c_{i1}, c_{i2})$, for $1 \le i \le n$. Output $C = c_n$.

Decryption $\mathsf{MDec}(C, SK)$**:** Let $\bar{c}_n = C$, for $1 \le i \le n$, $\bar{c}_{n-1} \leftarrow \mathsf{Dec}_i(\bar{c}_n)$. 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 $1 \le i \le n$. Otherwise "$\perp$".

4.3 Security proof

The following theorem holds for our construction:

Theorem 2 *Multiple encryption consists of only parallel or sequential block from above construction is secure* IND-ME-CCA *secure in the random oracle model.*

The rest of this section will be dedicated to the proof of this theorem. We shall divide the proof into two parts: first part namely Lemma 1 proves the case of parallel construction and the second part namely Lemma 2 proves the case of sequential construction.

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

Lemma 1 *If there exists an adversary $\mathcal{B}$ that breaks a parallel multiple encryption $\mathcal{ME}$ with the construction given the section 4.1, then there is a probabilistic polynomial time adversary $\mathcal{A}$ breaks onewayness of any component cipher $\mathcal{E}_i$ with non-negligible advantage.*

Lemma 2 *If there exists an adversary that $\mathcal{B}$ breaks a sequential multiple encryption $\mathcal{ME}$ with the construction given the section 4.2, then there is an adversary $\mathcal{A}$ breaks onewayness of any component cipher $\mathcal{E}_i$ with non-negligible advantage.*

C.2.1: Proof of Lemma 1

CONSTRUCTION OF ADVERSARY. Suppose $\mathcal{B}$ breaks $\mathcal{ME}$ with probability $Succ_{\mathcal{B}}(k) = 1/2 + \varepsilon$ with adaptive queries on the 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)-AONT see section 7.3 and constructs $\mathcal{ME}$ as section 4.1. 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_j's. $\mathcal{A}$ 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 there exists an* $\mathsf{IND\text{-}ME\text{-}CPA}$ *adversary $\mathcal{B}$ that breaks parallel $\mathcal{ME}$ with advantage ε, there is $\mathcal{A}$ that breaks the indistinguishability of i-th component cipher with probability ϵ_1 or indistinguishability of (L, l, n)-*AONT *with advantage with advantage ϵ_2, such that $\varepsilon \leq \epsilon_1 + 2\epsilon_2$.*

Proof. Denote $Pr[\cdot]$ as the probability of events and define some events as:

$SucB$:	$\mathcal{B}$ gains advantage in the $\mathsf{IND\text{-}ME\text{-}CPA}$ game.
$E1$:	$\mathcal{B}$ breaks the indistinguishability of AONT, that is, $\mathcal{B}$ guesses b with (X_{-i}, M_0, M_1);
$E2$:	$\mathcal{B}$ outputs m_{i-b} from (m_{i-0}, m_{i-1}) and C_b.

Since $E1$ and $E2$ are independent, and $Pr[SucB|\neg E1 \wedge \neg E2]$ 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:

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

Completed. ■

Following section 4.1 and section 4.2, denote k_i is the length of necessary coin for $\mathcal{E}_i$ and l_i be the length of c_{i2}.

Claim 4 *Suppose $\mathcal{E}_i$ is γ-uniform (detailed discussion in [17]). If there is an* $\mathsf{IND\text{-}ME\text{-}CCA}$ *adversary $\mathcal{B}$ that breaks i-th 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 $G_i(M, r_1, ..., r_n) \oplus m_i$. $SucA_1$ 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 $Pr[fail1] \leq Pr[fail1|\neg AskA_0] \cdot Pr[\neg AskA_0] \leq \gamma + 2^{-l_i}$. So $Pr[SucA_1] = (1 - Pr[fail1])^{q_d} \geq (1 - \gamma - 2^{-l_i})^{q_d}$. 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 $SucB$, $SucA_1$ 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

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

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. $\mathcal{A}$ can then break onewayness of $\mathcal{E}_i$ with non-negligible advantage. Lemma 1 is thus proven.

Following section 4.1 and section 4.2, denote k_i is the length of necessary coin for $\mathcal{E}_i$ and l_i be the length of c_{i2}. Based on similar analysis of proof of Lemma 1, we can formulate the following:

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

The proof is quite similar to that of Claim 4, and is omitted here. Combine Lemma 1 and Lemma 2, theorem 2 is then proven.

DISCUSSION. One complementary remark should be addressed on the *uniformity* of underlying primitives [17]. What we have considered so far is mainly non-deterministic 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. 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+ [35], and form the whole multiple encryption with parallel construction with compatible input domain, or sequential connecting one after another. AONT can be even replaced by OAEP+. This construction should also be 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 forthcoming work.

5 New definition regarding multiple encryption

It seems contradictive to 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 that CCA security may be somehow excessively strong. In the real world, it is rare that $\mathcal{DO}$ helps even in such obvious attacks. For example a new cipher S' is constructed from a CCA-secure cipher S, where a harmless bit is appended to the ciphertext of S, and is discarded during decryption, then S' is no longer secure in the sense of CCA. It seems such attack to S' should be easily judged and have "no significant difference" in most of cases. In fact, when $\mathcal{DO}$ encounters such queries, it should easily determine whether this is really a "new" ciphertext, by just looking at the ciphertext.

5.1 Relaxing definition of CCA security

CCA security might be too strong and is not always necessary, as pointed out in [36, 3, 7], among which, Shoup's "benign malleability" [36] and An, Dodis and Rabin's "gCCA" [3] are basically equivalent: a relation function $\mathcal{RF}$ helps the Decryption Oracle against obvious attacks. In gCCA definition, the relation function performs as follows: if $\mathcal{RF}(c,c')=\mathsf{TRUE}\Rightarrow\mathsf{Dec}(c)=\mathsf{Dec}(c')$. 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 $\mathsf{Dec}(c)=\mathsf{Dec}(c')$ (refer [3] for more details). Canetti, Krawczyk and Nielsen [7] recently propose another relaxation, called "replayable chosen ciphertext attack" (RCCA), with most of cases strictly weaker than gCCA.

To rule out the definitional limitation of CCA security in multiple encryption setting, we also introduce a relaxed definition called "*weak chosen ciphertext attack for multiple encryption*" (ME-wCCA). In the definition of wCCA, there is a relation function $\mathcal{RF}^*$ is computed by invoking $\mathcal{RF}_i$ ($1\leq i\leq n$) during the decryption process inside $\mathcal{DO}$, with initial value of each $\mathcal{RF}_i$ set to FALSE, where $\mathcal{RF}_i$ is the relation function defined according to gCCA security for i-th component cipher $\mathcal{E}_i$. $\mathcal{RF}_i(c_i,c_i')=\mathsf{TRUE}\Rightarrow\mathsf{Dec}(c_i)=\mathsf{Dec}(c_i')$. Whenever $\mathcal{RF}_i=\mathsf{TRUE}$ for some i, $\mathcal{RF}^*$ halts and returns TRUE to $\mathcal{DO}$ immediately. Once receiving TRUE, $\mathcal{DO}$ outputs "$\perp$" to the adversary. Informally, if $\mathcal{RF}^*$ finds a part (may be the intermediate decryption result) of the query ciphertext looks "the same" as the corresponding part of the challenge ciphertext, it tells the Decryption Oracle to reject this decryption query. Since the rules for oracle access is the same, the definition of IND-ME-CCA only needs to be modified a little to adapt to IND-ME-wCCA.

We stress that ME-wCCA security is a reasonable relaxation for CCA security. This notion is basically an extension of gCCA security. By restricting a multiple encryption to only one component cipher, IND-ME-wCCA becomes IND-gCCA.

Definition 3 (IND-ME-wCCA) *In the beginning, 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. Then* MEnc-Gen *gives the public key PK to $\mathcal{EO}$ and the adversary, the secret key SK to an Key Exposure Oracle $\mathcal{KE}$ and Decryption Oracle $\mathcal{DO}$ with a Relation Function $\mathcal{RF}^*$ inside, which is computable in polynomial time. The adversary accesses at most $n-1$ time to $\mathcal{KE}$. The adversary access the $\mathcal{EO}$ with two messages $\{M_0, M_1\}$ as input. $\mathcal{EO}$ chooses $b \xleftarrow{R} \{0,1\}$ and encrypts M_b into C_b and returns C_b to the adversary. The adversary is allowed to access $\mathcal{DO}$ for arbitrary polynomial times, and $\mathcal{DO}$ responses with the corresponding plaintext as long as $\mathcal{RF}^*(C, C_b)$ does not output* TRUE. *The adversary may query the oracles adaptively, in any order it likes. 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.*

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

The following lemma shows that IND-ME-wCCA secure multiple encryption can be easily acquired from IND-gCCA secure component ciphers.

Lemma 3 *A multiple encryption scheme $\mathcal{ME}$ is* IND-ME-wCCA *secure w.r.t. $\mathcal{RF}^*$ by any of three basic constructions, if each component cipher $\mathcal{E}_i$ is* IND-gCCA *secure w.r.t relation function $\mathcal{RF}_i$, $1 \leq i \leq n$. $\mathcal{RF}^*$ is defined as $\mathcal{RF}^*(C, C') =$* TRUE, *such that $\mathcal{RF}_i(c_i, c_i') =$* TRUE *for some i, $1 \leq i \leq n$, where c_i, c_i' are two ciphertexts of $\mathcal{E}_i$, and C, C' are the corresponding ciphertexts for $\mathcal{ME}$.*

Proof. For simplicity, we assume AONT is secure according to the definition in Appendix A (It is easy to modify the proof to the case in which security of AONT is also strictly considered). 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-wCCA secure, then we can use the IND-ME-wCCA 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-wCCA 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 $(pk_1, ..., pk_i, ..., pk_n)$, and the secret key is $(sk_1, ..., sk_i, ..., sk_n)$. Only sk_i is unknown to $\mathcal{D}$. To simulate the decryption query Q_i made by $\mathcal{D}_i$, $\mathcal{D}$ completes Q_i for $\mathcal{E}_i$ into Q for $\mathcal{ME}$ by those secret keys in hand, checks that the respective Q is a valid query (otherwise it will outputs $\perp$) if relation function outputs FALSE, then make query Q to its Decryption Oracle to decrypt Q. Next $\mathcal{D}$ outputs a pair (M_0, M_1) and also generate the corresponding pair (m_{i_0}, m_{i_1}) for $\mathcal{E}_i$. 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 $\mathcal{RF}_i$ we know that $\mathcal{E}_i$ is forbidden to decrypt any $\mathcal{RF}_i(c_i, c_i') =$ TRUE, i.e., $\mathcal{RF}^*(C_1, C_2) = \mathsf{TRUE}$, but this is the only limit that $\mathcal{D}_i$ is forbidden to ask its Decryption Oracle. $\mathcal{D}$ 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 3 *If all component ciphers are* IND-CCA *secure and chosen independently according to above three constructions, then the resulting multiple encryption is* IND-ME-wCCA *secure.*

In fact, each attack per theorem 1 can construct a new ciphertext with the same plaintext. Since non-malleability is an arduous goal for multiple encryption, we define relaxed gNM-ME-CCA similar to IND-ME-wCCA. Informally, the definition says that the adversary does not win as long as it outputs with a new ciphertext with the same relation regulated by the relation function to the challenge ciphertext, where the relation function is defined analogously to that of IND-ME-wCCA.

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 with non-negligible probability, where $\mathcal{RF}^*$ is defined identical to* ME-wCCA. *Denote $\mathbb{M}, \mathbb{C}$ as sets of plaintexts and ciphertexts being empty initially, respectively.*

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

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

6 Relations among security definitions for multiple encryption

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

Theorem 4 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 $b \xleftarrow{R} \{0,1\}$, 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.

A NM-ME-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ utilizes another IND-ME-CCA adversary $\mathcal{B} = (\mathcal{B}_{\mathsf{find}}, \mathcal{B}_{\mathsf{guess}})$ to break the non-malleability of the scheme, by letting $\mathcal{B}_{\mathsf{find}}$ chooses a pair of messages M_0, M_1 where $M_0 = \bar{M}_1$ and passes on to $\mathcal{B}_{\mathsf{guess}}$ that correctly guesses b:

Adversary $\mathcal{A}_1^{\mathcal{KE},\mathcal{DO}}$	Adversary $\mathcal{A}_2^{\mathcal{DO}}(M_b, s')$ where $s' = (M_0, M_1, PK, s)$
$(M_0, M_1, s) \leftarrow \mathcal{B}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}(PK)$	$C_b \leftarrow \mathcal{B}_{\mathsf{guess}}^{\mathcal{DO}}(M_b, s)$
$b \stackrel{R}{\leftarrow} \{0,1\}$	$(C'_b, R) \leftarrow \mathsf{MEnc}(\bar{M}_b)$
$s' \leftarrow (M_0, M_1, PK, s)$	return C'_b, R
return M_b, s'	

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.

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 $\mathcal{A}_{\mathsf{find}}^{\mathcal{KE},\mathcal{DO}}$	Adversary $\mathcal{A}_{\mathsf{guess}}^{\mathcal{DO}}(M, s')$ where $s' = (M_0, M_1, PK, s)$
$(M_0, M_1, s) \leftarrow \mathcal{B}_1^{\mathcal{KE},\mathcal{DO}}(PK)$	$C'_b \leftarrow \mathcal{B}_2^{\mathcal{DO}}(M_0, M_1, s)$
$M \stackrel{R}{\leftarrow} \{0,1\}$	$(M'_b, R) \leftarrow \mathsf{MDec}(C'_b)$
$s' \leftarrow (M_0, M_1, PK, s)$	if $R(M'_b, M_0) = \mathsf{TRUE}$ then $d \leftarrow 0$
return M, s'	else $d \leftarrow 1$
	return d

Then $\mathcal{A}$ succeeds with exactly the probability of $\mathcal{B}$, which states any scheme meeting NM-ME-CCA security must also meet IND-ME-CCA security. Combining above two lemmas, we complete the proof. ■

Theorem 5 IND-ME-wCCA $\Leftrightarrow$ gNM-ME-CCA

Proof Idea. Since we have already proven 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-wCCA also meets gNM-ME-CCA while a scheme meet gNM-ME-CCA also meets IND-ME-wCCA security.

Proof. Denote two Relation Function in IND-ME-wCCA 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-wCCA 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-wCCA 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-wCCA 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-wCCA adversary. This completes the proof. ■

Theorem 6 IND-ME-wCCA $\Rightarrow$ IND-ME-CPA*, however,* IND-ME-CPA $\not\Rightarrow$ IND-ME-wCCA.

Proof Idea. It is trivial of the former part, for a ME-wCCA 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, the Decryption Oracle returns the the secret key. However, this scheme still remains ME-CPA secure.

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

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

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	MEnc(M)	MDec(C)
$(pk'_i, sk'_i) \leftarrow$MGen-Enc$'$, for $1 \le i \le n$;	$c' \leftarrow \mathsf{MEnc}'(M)$	$v\|\|\bar{c}' \leftarrow C$
$PK' \leftarrow (pk'_1, ..., pk'_n)$, $SK' \leftarrow (sk'_1, ..., sk'_n)$	$C = 0\|\|c'$	`if` $v = 0$
$u \leftarrow \{0, 1\}^k$	`Return` C	`Return` $\mathsf{MDec}'_{SK'}(\bar{c}')$
$PK = u\|\|PK'$, $SK = SK'$		`else if` $\bar{c}' = u$
`Return` (PK, SK)		`Return` SK

We can see $\mathcal{ME}$ is not ME-wCCA 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 $\mathcal{ME}$ 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, M_1), the adversary outputs its guess on b. Then denote the probability of following events as:

$$1 := [v = 0, (PK, SK) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0, 1\}, MEnc(M_b) \leftarrow MEnc(M_b) : b = \bar{b}];$$
$$2 := [v = 1, (PK, SK) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0, 1\}, MEnc(M_b) \leftarrow MEnc(M_b), c'_b \neq u : b = \bar{b}];$$
$$3 := [v = 1, (PK, SK) \leftarrow \mathsf{MGen} - \mathsf{Gen}, b \leftarrow \{0, 1\}, MEnc(M_b) \leftarrow MEnc(M_b), c'_b = u : b = \bar{b}]$$

Denote $SucB$ as the even that $\mathcal{B}$ outputs a successful guess on b with larger probability than 1/2. Let the advantage of an adversary $\mathcal{B}$ attacking $\mathcal{ME}'$ be p_0, denote $k = |c'|$ as the length of c', the following holds:

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

It is easy to see $AdvB$ is negligible. Proof completes. ■

7 Applications to key-insulated cryptosystem

7.1 Key-insulated cryptosystem

The key-insulated cryptosystem is proposed by [13] to protect cryptosystems against partial key exposure. In such system, computation is done in an insecure user device. Additionally, there is a physically secure server that 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 [13], a generic construction is proposed based on arbitrary semantically secure public key encryption against *chosen plaintext attack* and cover-free family.

Generic Construction of [13]. First the key generation algorithm is run and u public key/secret key pairs of underlying semantically secure cryptosystems are generated, where $S_1, ..., S_N \subset [u] \stackrel{\text{def}}{=} \{1, ..., u\}$ 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. 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: $M = \mathcal{I}(m_1, ..., m_n)$.

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 security of the system is defined as: 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 an Key Exposure Oracle and a Decryption Oracle. In the proof to this generic construction, it is shown that the whole system has indistinguishability of messages in any period that is not compromised with at most t other periods compromised under *chosen plaintext attack.*[2]

7.2 Chosen ciphertext security of generic construction in [13]

One may naturally think the generic construction in [13] is secure against chosen ciphertext attacks if the underlying cryptosystems are $\mathsf{IND\text{-}CCA}$ secure. However, it can be demonstrated that actually this generic construction is insecure against *chosen ciphertext attack*, although it is indeed an desirable property of a key-insulated cryptosystem. Recall that the authors of [13] do not claim their generic construction CCA secure.

At the first look, because of the property of cover-free family even if the secret keys are compromised in t periods, at most $t-1$ secret keys of a period other than these t are known to the adversary. 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. However, an adversary in fact can bypass the hard task and just needs to try to modify the challenge ciphertext using known secret keys in order to get help from the Decryption Oracle. In fact, it can obtain any secret key sk_j by sending adaptive query to the Key Exposure Oracle $\mathcal{KE}$ for sk_j in some period 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

[2]Also in [13], a concrete scheme is given based on DDH assumption, which is CCA secure.

can replace c_j with c'_j and submit this "new" ciphertext C' to the Decryption Oracle, which will return the corresponding message M. This attack works for any period i.

Though the original generic construction does not satisfy chosen ciphertext attack security, actually if every component cipher is chosen IND-CCA secure, this generic construction is actually IND-ME-wCCA secure (Theorem 3). It should be assessed that this scheme still provides very practical security.

7.3 A generic construction of the key-insulated cryptosystem with CCA security

In fact, the feasibility of constructing a CCA secure key-insulated cryptosystem (parallel multiple encryption) has already been shown in section 4. We are only fascinated at whether given IND-CCA secure ciphers and secure AONT as building blocks, a parallel construction can be transformed to a CCA secure key-insulated cryptosystem with minimum modification.

Observing the "natural" parallel construction (section 2.1.1) with IND-CCA secure components is already IND-ME-wCCA secure according to Theorem 3, we can further have more efficient construction by a simple modification to the scheme. As the gap between IND-ME-CCA and IND-ME-wCCA is just that for the former sometimes the adversary can lay a trap when asking the tailored decryption queries, this gap can be immediately merged once such attack is ruled out. For a secure multiple encryption must be probabilistic, there must be auxiliary randomness used in the encryption. If the Decryption Oracle can extract all the randomness and verify it before outputting the plaintext, then the Decryption Oracle should be able to immune itself from such partial re-encryption attacks. If a ciphertext passes such randomness check, then with overwhelming probability, the Decryption Oracle can make sure that the sender of this ciphertext knows the corresponding plaintext.

We add such transforms to the basic parallel construction: recall the notation $\mathtt{coin}_i$ is the auxiliary randomness input for encryption component $\mathcal{E}_i$. Let $\mathtt{coin}_i = h(\mathtt{r}||Index_i)$, where $\mathtt{r}$ is a random number, $Index_i$ is the description of i-th component and h is a random function. The Encryption is $C = \mathsf{MEnc}(M||r; (\mathtt{coin}_1, ..., \mathtt{coin}_n))$, especially for IND-CCA component $\mathcal{E}_i$, $\mathsf{Enc}_i(m_i; \mathtt{coin}_i)$ where m_i is generated from AONT with input $M||r$. Decryption process becomes: for a ciphertext C', $M'||\mathtt{r}' = \mathsf{MDec}(C')$, output M' only if $c'_i = \mathsf{Enc}_i(m_i; h(\mathtt{r}'||Index_i))$ is well formed, for every $1 \leq i \leq n$. Whenever it is detected that a ciphertext has used invalid randomness, the Decryption Oracle rejects this query immediately.

It is easy to see this scheme satisfies the security definition of [13] under CCA attack. The proof is easy and will be omitted here. We point out this is actually the *first* generic construction of key-insulated cryptosystem enjoying CCA security (Another generic construction for CCA secure key insulated cryptosystem will be given by Dodis and Katz in their upcoming work, whose security can be proven in the standard model.). In fact, this transform states the transform of turning IND-ME-CPA into IND-ME-CCA.

Acknowledgement

The authors would like to thank Masayuki Abe, Yevgeniy Dodis, Yumiko Hanaoka, Jonathan Katz and Kazukuni Kobara for invaluable comments and discussions.

References

[1] M. Abe and H. Imai. Flaws in some robust optimistic mix-nets. In *ACISP'03*, volume 2727 of *LNCS*, pages 39 – 50. Springer-Verlag, 2003.

[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] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Eurocrypt'02*, volume 2332 of *LNCS*, pages 83–107, Springer-Verlag, 2002.

[4] M. Bellare, A. Desai, D. Pointcheval, and P. Rogway. Relations among notions of security for public-key encryption schemes. In *Crypto'98*, volume 1462 of *LNCS*. Springer-Verlag, 1998.

[5] M. Bellare and A. Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Crypto'99*, volume 1666 of *LNCS*, pages 519–536. Springer-Verlag, 1999.

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

[7] R. Canetti, H. Krawczyk, and J. Nielsen. Relaxing chosen-ciphertext security. In *Crypto'03*. Full version available: `http://eprint.iacr.org/2003/174/`, 2003.

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

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

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

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

[12] Y. Dodis and J. Katz. Rump session talk. In *Crypto'03*, 2003.

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

[14] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23rd STOC*, pages 542–552. ACM, 1991.

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

[16] G. Frey and H.G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, 1994.

[17] 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.

[18] O. Goldreich. *Foundations of cryptography*, volume 1. Cambridge Unversity Press: New York, 2001.

[19] O. Goldreich. *Foundations of Cryptography: Volume II (third posted version)*. Aavailable at `http://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/enc.ps`, 2002.

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

[21] 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.

[22] D. Hofheinz, J. Mueller-Quade, and R. Steinwandt. On modeling IND-CCA security in cryptographic protocols, 2003. Available at: `http://eprint.iacr.org/2003/024/`.

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

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

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

[26] 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.

[27] 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.

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

[29] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *22nd STOC*, pages 427–437. ACM, 1990.

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

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

[32] T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Commentarii Mathematici Universitatis Sancti Pauli*, (47):81–92, 1998.

[33] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, (67):353–356, 1998.

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

[35] V. Shoup. OAEP reconsidered. In *Crypto'01*, volume 2139 of *LNCS*, pages 239–259, 2001.

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

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

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

[39] 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.

[40] R. Zhang, G. Hanaoka, J. Shikata, and H. Imai. On the security of multi-layered encryption or CCA-security+CCA-security=CCA-security? In *SCIS'03*, January, 2003.

Appendix A: Some definitions

A.1: Public key encryption scheme

A 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 ($\mathsf{IND\text{-}CCA}$), 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 the 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,(this can be 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. We denote this as:

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

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

A.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 $X \stackrel{\text{def}}{=} (m_1, ..., m_n)$, 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 $\mathsf{LR}_b(j, M_0, M_1) \stackrel{\text{def}}{=} \mathcal{T}_{-j}(M_b)$, for any PPT adversary $\mathcal{A}$ attacking AONT, define its advantage as $Adv_{\mathcal{A},\mathcal{T}} \stackrel{\text{def}}{=} Pr[b \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}^{\mathsf{LR}_b(\cdot,\cdot,\cdot)} : b' = b] - 1/2$. Then Adv is negligible.

A.3: Cover-free family

A family of subsets $S_1, ..., S_N$ over some universe 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 $|S_{i_0} \backslash \cup_{j=1}^{t} S_{i_j}| \geq \beta |S_{i_0}|$.

Appendix B: Figures

$$\boxed{\begin{array}{ccccccccc}
 & & & & M & & & & \\
 & & \swarrow & & \downarrow & & \searrow & \} & \mathsf{AONT} \\
m_1 & & & \cdots & \cdots & \cdots & & m_n & \\
 & & & & \Downarrow & & & & \\
m_1 & & & \cdots & \cdots & \cdots & & m_n & \\
\downarrow & & & & \downarrow & & & \downarrow & \\
\boxed{\mathcal{E}_1} & \leftarrow pk_1 & & & & & & \boxed{\mathcal{E}_n} & \leftarrow pk_n \\
\downarrow & & & & \downarrow & & & \downarrow & \\
c_1 & & & \cdots & \cdots & \cdots & & c_n &
\end{array}}$$

Figure 1: Parallel construction of multiple encryption

$$\boxed{\begin{array}{ccccccccccccc}
 & & pk_1 & & & & & & & & pk_n & & \\
 & & \downarrow & & & & & & & & \downarrow & & \\
M & \rightarrow & \boxed{\mathcal{E}_1} & \rightarrow & c_1 & \rightarrow & \cdots & \rightarrow & c_{n-1} & \rightarrow & \boxed{\mathcal{E}_n} & \rightarrow & c_n
\end{array}}$$

Figure 2: Sequential construction of multiple encryption