Hindawi
Security and Communication Networks
Volume 2019, Article ID 1042649, 21 pages
https://doi.org/10.1155/2019/1042649

Research Article

Multidivisible Online/Offline Cryptography and Its Application to Signcryptions

Dan Yamamoto [1] **and Wakaha Ogata** [2]

[1]*Hitachi, Ltd., Yokohama, Japan*
[2]*Tokyo Institute of Technology, Tokyo, Japan*

Correspondence should be addressed to Dan Yamamoto; dan.yamamoto.vx@hitachi.com

Received 17 January 2019; Accepted 17 July 2019; Published 8 October 2019

Academic Editor: Bruce M. Kapron

We introduce a general concept of *multidivisible online/offline (MDO) cryptography*, which covers the previous works including online/offline cryptographic schemes, divisible online/offline signatures, incrementally executable signcryptions, and multi-divisible online/offline encryptions. We then present the notion of *multidivisible online/offline signcryptions* (MDOSCs) as novel application of MDO cryptography. We define several security notions for MDOSCs and show implications and separations between these security notions. We also present a generic construction of MDOSC that achieves the strongest security notions with regard to confidentiality and unforgeability. Using MDOSC schemes, the computationally restricted and/or bandwidth-restricted devices can transmit messages in both confidential and authenticated way with low computational overhead and/or low-bandwidth network.

1. Introduction

In the emerging network environment, e.g., Internet of Things (IoT), more and more computationally restricted devices as well as bandwidth-restricted devices are connected to each other at any time in any place. Securing their communication is urgently required to make these environments sustainable and scalable. We can use cryptographic schemes as building blocks to achieve security for them, only if it is taken into account that the devices here have restricted resources.

Online/offline cryptography, a fundamental concept for many cryptographic systems (e.g., signatures [1–4], encryptions [5–8], signcryptions [9–13], and so forth), is a key to reduce computational costs of the devices sending their messages in confidential and/or authenticated way.

The history of online/offline cryptography began with online/offline signatures proposed by Even et al. [2, 3]. The signing procedure of online/offline signature scheme is split into the offline phase and the online phase. The signer can perform all the computationally expensive operations in the offline phase, i.e., in any idle time before the sender decides the message to be signed. Then, in the online phase, i.e., after the message is determined, only lightweight operations are required to sign the message so that even low-power devices can handle the signing process.

As an extension of online/offline signature scheme in the context of signcryption, the notion of incrementally executable signcryptions (IESCs) was proposed in [12]. Here, a signcryption process is split into three subprocesses, where the sender can activate each subprocess incrementally by its given sequential input: the sender's own key pair, a recipient's public key, and a plaintext message to be sent to the recipient. We can utilize significant intervals between these three subprocesses to perform as much precomputation as possible.

To save the transmission bandwidth as well as computational cost, Gao et al. [4] proposed the notion of *divisible online/offline signatures* (DOSs). Using DOS, intermediate outputs can be securely sent to the receiver in the offline phase so that the senders can save not only computational resources but also transmission bandwidths in the online phase.

Applying the above attractive features of DOS as well as IESC to public-key encryption schemes, the notion of multidivisible online/offline encryptions (MDOEs) was proposed in [14]. In MDOEs, encryption process can be executed in at most three steps and at most three partial ciphertexts can be transmitted one-by-one.

Our contributions: in this paper, we propose a general concept of *multidivisible online/offline cryptography* (MDO cryptography, for short), which covers the online/offline cryptography, divisible online/offline signatures, incrementally executable signcryptions, and multidivisible online/offline encryptions.

In general, MDO cryptographic schemes have the following features:

Incremental processing: a sender's process can be divided into two or more subprocesses

Incremental sending: outputs of intermediate subprocesses can be sent prior to all the subsequent subprocesses

These features enable us to save both computational overhead and transmission bandwidth, on which we can construct secure communication platform for IoT-like environment.

As instances taking full advantage of these two features of MDO cryptography, we introduce notions of multidivisible online/offline signcryptions (MDOSCs) and divisible online/offline tag-based KEMs.

We define parameterized security notions of MDOSCs, denoted as (k, n, q)-dM-IND-iCCA for confidentiality and (k, n)-dM-UF-iCMA for unforgeability, with regard to three parameters: the level of divisibility k, the number of users n, and the number of queries per user q. Then, we analyze all the relationships, i.e., the implications and separations, among these security notions.

We also present a generic construction of MDOSC that achieves the strongest security notions of both confidentiality and unforgeability.

Expected application scenario: Figure 1 shows a usage example of our MDOSCs, where a signcryption algorithm is split into three subalgorithms, i.e., sc_1, sc_2, and sc_3, each of which can be processed incrementally and can send outputs to recipient incrementally. The sender in our example system consists of the following three components: a computationally powerful cloud with sc_1, an energy-restricted mobile device with sc_2, and the most computationally restricted as well as bandwidth-restricted IoT device such as a sensor node with sc_3. Both the sender and the receiver in our system can access a shared storage. We assume the receiver has enough computational resource to decrypt signcrypted messages received from the sender. Firstly, the sender uses cloud computing to execute the most computationally expensive algorithm sc_1 with the sender's own key pair $(\mathrm{sk}_S, \mathrm{pk}_S)$ to generate partial ciphertext Σ_1, store it into the shared storage, and also store state information φ_1 to the storage of the mobile device. These procedures can be repeatedly executed depending on the storage capacity of the mobile device and the costs for using cloud computing as well as shared storage. Next, when the sender recognizes the target recipient to whom she might send some messages and obtains the receiver's public key pk_R, the sender's mobile device executes less-expensive algorithm sc_2 with φ_1 and pk_R to generate partial ciphertext Σ_2, transmit it to the receiver, and store state information φ_2 to the storage of the IoT device. These procedures can also be repeatedly executed depending on the storage capacities of the IoT device and the receiver. After that, when the IoT device is ready to send a message m (e.g., typically very short numeric value sensed by sensor), it executes the least expensive algorithm sc_3 with φ_2 and m to generate partial ciphertext Σ_3, transmit it to the receiver. Note that Σ_3 is reasonably short since the cryptographic information for decryption (unsigncryption) has been already transmitted as Σ_1 and Σ_2. Finally, the receiver securely obtains the message m by unsigncrypting Σ_2 and Σ_3 with separately downloaded Σ_1 from the shared storage as well as pk_S and sk_R using the unsc algorithm.

Organization of this paper: this paper is organized as follows. Section 2 introduces basic notations used in this paper. In Section 3, we introduce the concept of MDO cryptography and discuss its instances including MDO signatures, MDO encryptions, and MDO signcryptions. Section 4 formally defines the notion of MDO signcryptions and its security notions. The relations between the security notions of MDOSC are discussed in Section 5. Section 6 shows a generic construction of MDOSC and comparison with previous signcryption schemes as well as usage example of our MDOSC construction. Concluding remarks are given in Section 7.

2. Notations

For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. We write $x \longleftarrow y$ or $y \longrightarrow x$ to indicate that x is the output from a function or an algorithm y, or the assignment of the value y to x. $x \longleftarrow_{\$} \mathcal{X}$ denotes the operation of selecting a random element x from a set $\mathcal{X}$. We write A.x to indicate explicitly that the algorithm x belongs to the scheme A. We write $z \longleftarrow \mathcal{A}(x, y, \ldots; \mathrm{oracle}_1, \mathrm{oracle}_2, \ldots)$ to indicate the operation of a turing machine $\mathcal{A}$ with inputs $x, y, \ldots$ and access to oracles $\mathrm{oracle}_1, \mathrm{oracle}_2, \ldots$, and letting z be the output. Unless otherwise indicated, algorithms are randomized, i.e., it takes a source of randomness to make random choices during execution. In all the experiments (games), every number, set, and bit string is implicitly initialized by 0, empty set $\varnothing$, and empty string ε, respectively. We write $\Pr[G_X^Y(\mathcal{A}) = b]$ to indicate the probability of the event that adversary $\mathcal{A}$ outputs b in attack game G_X^Y.

3. Multidivisible Online/Offline Cryptography

In this section, we propose a concept of (multi)-divisible online/offline (MDO) cryptography as a generalization of online/offline cryptography. This concept can be applied to various cryptographic primitives, such as encryption schemes and signature schemes. MDO cryptographic

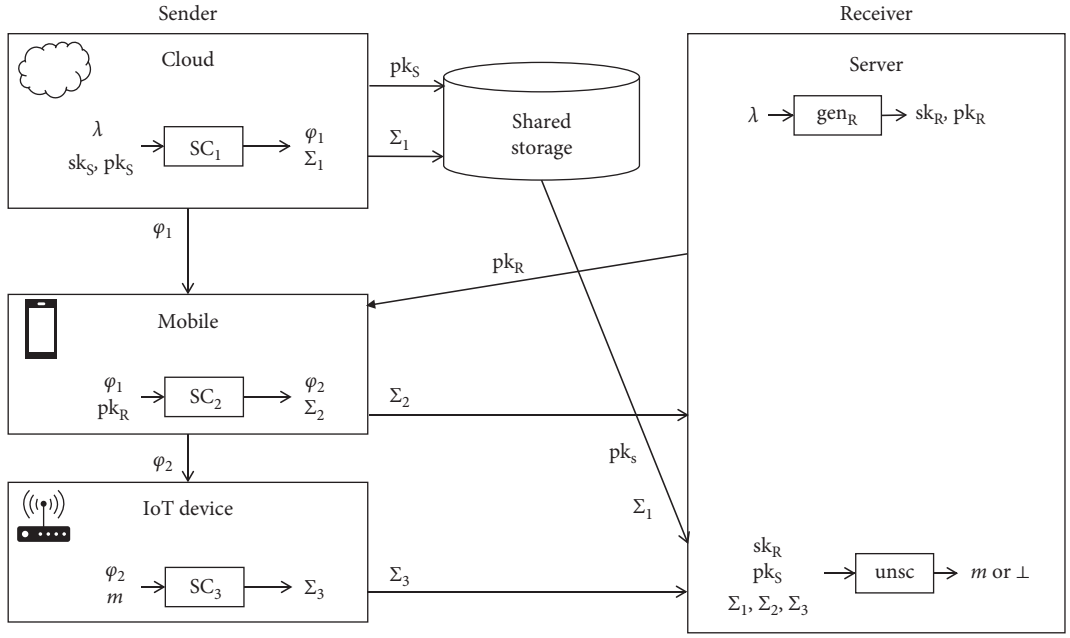

Figure 1: Usage example of our proposed signcryption construction MDOSC.

schemes have the two features: incremental processing and incremental sending.

Incremental Processing. In a cryptographic primitive, a list of several inputs $(x_1, x_2, \ldots, x_k)$ is processed by an algorithm ALG to compute $y = \mathrm{ALG}(x_1, x_2, \ldots, x_k)$, which is sent via a (insecure) channel.

In contrast, ALG in MDO cryptography is split into $k(\geq 1)$ subalgorithms $\mathrm{ALG}_1, \ldots, \mathrm{ALG}_k$, and

$$\begin{aligned}(\varphi_1, y_1) &\longleftarrow \mathrm{ALG}_1(x_1)\\ (\varphi_2, y_2) &\longleftarrow \mathrm{ALG}_2(x_2, \varphi_1)\\ &\vdots\\ y_k &\longleftarrow \mathrm{ALG}_k(x_k, \varphi_{k\text{-}1}),\end{aligned} \tag{1}$$

are computed instead of $y = \mathrm{ALG}(x_1, x_2, \ldots, x_k)$. $y = (y_1, \ldots, y_k)$ is treated as the output of ALG. $\varphi_i\,(i = 1, \ldots, k\text{-}1)$, state information, is assumed to be secretly stored and will be consumed *only once* by subalgorithm ALG_{i+1}.

For a more general model, k inputs are grouped into $k'\,(\leq k)$ input groups, $(x'_1, \ldots, x'_{k'})$. In this case, ALG is split into k' subalgorithms. By this definition, our concept enables us to model wide variety of systems in a unified way. For example, an online/offline scheme is a special case where inputs are divided into two groups, a message and the others. Moreover, an ordinary cryptographic scheme can be treated as a special case where all inputs are grouped into only one input group, i.e., $k' = 1$.

Grouping of inputs should depend on the timing when each input is given to the sender.

Incremental Sending. To save transmission bandwidth, it is desired to send each partial output y_i one-by-one. However, it is known that a scheme could be insecure if each y_i is allowed to be sent one-by-one, even when the scheme is secure if all the y_i's are sent at the same time [14]. Therefore, we need extended security notions that capture new types of adversary exploiting partial output y_i's for attacks.

In the following, we show several MDO cryptographic primitives including: signatures, public-key encryptions, key encapsulation mechanisms (KEMs), tag-based key encapsulation mechanisms (TBKEMs), and signcryptions.

3.1. Signatures. We recall the notion of *divisible online/offline signatures (DOSs)* [4]. A divisible online/offline signature scheme DOS consists of the following algorithms:

$\mathrm{init}(\,) \longrightarrow \lambda$ (the *public parameter generation* algorithm): it outputs public parameters λ to be used by all parties

$\mathrm{gen}(\lambda) \longrightarrow (\mathrm{sk}, \mathrm{pk})$ (the *key generation* algorithm): it generates a key pair $(\mathrm{sk}, \mathrm{pk})$ for signing and verification

$\mathrm{sign}_1(\lambda, \mathrm{sk}) \longrightarrow (\varphi, \sigma_1)$ (the *offline signing* algorithm): given public parameters λ and a secret key sk, it outputs state information φ and an offline signature token σ_1. We require that $\sigma_1 \neq \varepsilon$

$\mathrm{sign}_2(\varphi, m) \longrightarrow \sigma_2$ (the *online signing* algorithm): given state information φ and a message m, it outputs online signature token σ_2

$\mathrm{ver}(\lambda, \mathrm{pk}, m, \sigma_1, \sigma_2) \longrightarrow v$ (the deterministic *verification* algorithm): given a public key pk, a message m, and signature tokens (C_1, C_2), it outputs 1 (accept) or 0 (reject)

We require the correctness for DOS, namely, for any $\lambda \longleftarrow \mathrm{DOS.init}(\,)$, any $(\mathrm{sk}, \mathrm{pk}) \longleftarrow \mathrm{DOS.gen}(\lambda)$, any $m \in$

$\{0,1\}^*$, any $(\varphi, \sigma_1) \longleftarrow \mathrm{DOS.sign}_1(\lambda, \mathrm{sk})$, and any $\sigma_2 \longleftarrow \mathrm{DOS.sign}_2(\varphi, m)$, we have $\mathrm{DOS.ver}(\lambda, \mathrm{pk}, m, \sigma_1, \sigma_2) = 1$.

As an attempt, we might try to extend DOS to MDOS, *multi*divisible online/offline signatures, based on our framework. More specifically, we could further split $\mathrm{sign}_1(\lambda, \mathrm{sk})$ into $\mathrm{sign}_{1-}(\lambda)$ and $\mathrm{sign}_{1+}(\mathrm{sk})$. While this splitting is possible theoretically, it does not provide significant benefits to senders in practical. This is because public parameters λ and secret key sk are obtained by senders almost at the same time in most applications so that senders cannot take advantage of the precomputation of sign_{1-}. We therefore do not try to formalize the syntax of *multi*divisible online/offline signatures in this paper. Nevertheless, we can provide the security notions for it based on multidivisible online/offline style to capture the unforgeability notion for ordinary signatures [15] as well as the notion for divisible online/offline signatures [4] in a unified way.

We define the security notion of (strong) unforgeability against adaptive chosen message attacks ((k,n)-UF-CMA and (k,n)-sUF-CMA) as follows.

Definition 1. Let $k \in \{1,2\}$ and $n \in \mathbb{N}$. Let $X \in \{\mathrm{UF}, \mathrm{sUF}\}$. Let DOS be a DOS scheme, and let $\mathscr{A}$ be an adversary. The (k,n)-X-CMA-advantage of $\mathscr{A}$ against DOS is defined as

$$\mathrm{Adv}_{\mathrm{DOS}}^{(k,n)\text{-X-CMA}}(\mathscr{A}) = \Pr\left[G_{\mathrm{DOS}}^{(k,n)\text{-X-CMA}}(\mathscr{A}) = 1\right], \quad (2)$$

where the attack game $G_{\mathrm{DOS}}^{(k,n)\text{-X-CMA}}$ is defined in Figure 2.

We say that DOS is (t, q, ϵ)-(k,n)-X-CMA-secure if $\mathrm{Adv}_{\mathrm{DOS}}^{(k,n)\text{-X-CMA}}(\mathscr{A}) \leq \epsilon$ holds for any adversary $\mathscr{A}$ that runs in time t and makes at most q queries to oracle sign.

Note that the notion of $(1,1)$-UF-CMA is equivalent to the standard unforgeability notion for ordinary signatures [15], whereas $(2,1)$-UF-CMA is identical to the notion for divisible online/offline signatures [4].

We also define the notion of *smooth* divisible online/offline signature scheme in a similar fashion with a smooth KEM [16].

Definition 2. Smoothness $\mathrm{Smth}_{\mathrm{DOS}}$ for DOS is defined as follows:

$$\mathrm{Smth}_{\mathrm{DOS}} = E\left[\max_{\sigma_1 \in \{0,1\}^*} \Pr_{(\varphi, \sigma_1') \longleftarrow \mathrm{DOS.sign}_1(\lambda, \mathrm{sk})}\left[\sigma_1' = \sigma_1\right]\right], \quad (3)$$

where the expected value is taken over $\lambda \longleftarrow \mathrm{DOS.init}()$; $(\mathrm{sk}, \mathrm{pk}) \longleftarrow \mathrm{DOS.gen}(\lambda)$. We say DOS is ϵ-smooth if $\mathrm{Smth}_{\mathrm{DOS}} \leq \epsilon$ holds.

3.2. Public-Key Encryptions and KEMs. The notion of multidivisible online/offline encryptions (MDOEs) was proposed in [14]. It captures both the desirable features of identity-based online/offline encryptions [5–8] and divisible online/offline signatures [4], that is, a part of encryption can be executed even when the public key to the receiver is unknown and partial ciphertexts can be sent to the receiver even when the sender's inputs (e.g., a public key or a plaintext) are not fully determined. Concrete constructions are also proposed in [14], which allow the computationally restricted and/or bandwidth-restricted devices to transmit ciphertexts with low computational overhead and/or low-bandwidth network.

Their concrete constructions are built on *partitioned* KEM, which can be regarded as an instance of divisible online/offline KEM. As with DOS and MDOE, partitioned KEMs have divided encapsulation algorithms, $\mathrm{enc}_1(\lambda)$ and $\mathrm{enc}_2(\mathrm{pk})$, both of which output partial encapsulations C_1 and C_2.

3.3. Tag-Based Key Encapsulation Mechanism. We introduce the notion of *divisible online/offline tag-based KEM (DOTK)*, which is a tag-based analogue of divisible online/offline KEM, i.e., partitioned KEM.

A divisible online/offline tag-based key encapsulation mechanism DOTK consists of the following algorithms:

$\mathrm{init}() \longrightarrow \lambda$ (the *public parameter generation* algorithm): it outputs public parameters λ to be used by all parties. Public parameters λ contains a description of the session key space, $\mathrm{DOTK}.\mathscr{K}$.

$\mathrm{gen}(\lambda) \longrightarrow (\mathrm{sk}, \mathrm{pk})$ (the *key generation* algorithm): it generates a key pair $(\mathrm{sk}, \mathrm{pk})$ for decapsulation and encapsulation.

$\mathrm{enc}_1(\lambda, \tau) \longrightarrow (\varphi, C_1)$ (the *first key encapsulation* algorithm): given public parameters λ and a tag $\tau \in \{0,1\}^*$, it outputs state information φ and a partial encapsulation C_1. We require that $C_1 \neq \varepsilon$.

$\mathrm{enc}_2(\varphi, \mathrm{pk}) \longrightarrow (K, C_2)$ (the *second key encapsulation* algorithm): given state information φ and a public key pk, it outputs a pair (K, C_2), where $K \in \mathrm{DOTK}.\mathscr{K}$ is a generated session key and C_2 is a partial encapsulation of K.

$\mathrm{dec}(\lambda, \mathrm{sk}, \tau, C_1, C_2) \longrightarrow K$ (the deterministic *decapsulation* algorithm): given a secret key sk, a tag τ, and encapsulations (C_1, C_2), it outputs either a session key K or an error symbol $\perp$.

We require the correctness for DOTK, namely, for any $\lambda \longleftarrow \mathrm{DOTK.init}()$, any $(\mathrm{sk}, \mathrm{pk}) \longleftarrow \mathrm{DOTK.gen}(\lambda)$, any $\tau \in \{0,1\}^*$, any $(\varphi, C_1) \longleftarrow \mathrm{DOTK.enc}_1(\lambda, \tau)$, and any $(K, C_2) \longleftarrow \mathrm{DOTK.enc}_2(\varphi, \mathrm{pk})$, we have $\mathrm{DOTK.dec}(\lambda, \mathrm{sk}, \tau, C_1, C_2) = K$.

We define the security notion of indistinguishability against adaptive tag and adaptive chosen ciphertext attacks ((k,n,q)-IND-tag-CCA).

Definition 3. Let $k \in \{1,2\}$ and $n, q \in \mathbb{N}$. Let DOTK be a DOTK scheme, and let $\mathscr{A}$ be an adversary. The (k,n,q)-IND-tag-CCA-advantage of $\mathscr{A}$ against DOTK is defined as

$$\mathrm{Adv}_{\mathrm{DOTK}}^{(k,n,q)\text{-IND-tag-CCA}}(\mathscr{A}) = \left|2\Pr\left[G_{\mathrm{DOTK}}^{(k,n,q)\text{-IND-tag-CCA}}(\mathscr{A}) = 1\right] - 1\right|, \quad (4)$$

where the attack game $G_{\mathrm{DOTK}}^{(k,n,q)\text{-IND-tag-CCA}}$ is defined in Figure 3.

Game $\mathsf{G}_{\mathsf{DOS}}^{(k,n)\text{-X-CMA}}(\mathcal{A})$**:**
$\lambda \leftarrow \mathsf{DOS.init}()$
for $i \in [n]$**:** $(sk_i, pk_i) \leftarrow \mathsf{DOS.gen}(\lambda)$
$\mathcal{O}_1 \leftarrow (\mathtt{sign12})$; $\mathcal{O}_2 \leftarrow (\mathtt{sign1}, \mathtt{sign2})$
$(i^*, m^*, \sigma_1^*, \sigma_2^*) \leftarrow \mathcal{A}(\lambda, \{pk_i\}_{i\in[n]};\ \mathcal{O}_k)$
$v_1 \leftarrow \mathsf{DOS.ver}(\lambda, pk_{i^*}, m^*, \sigma_1^*, \sigma_2^*)$
$v_2 \leftarrow \forall j.(i^*, m^* \boxed{, \sigma_1^*, \sigma_2^*}) \neq (i^{(j)}, m^{(j)} \boxed{, \sigma_1^{(j)}, \sigma_2^{(j)}})$
return $v_1 \wedge v_2$
Oracle $\mathtt{sign1}(i)$**:**
// $i \in [n]$; queried at most q times for each i
$j \leftarrow j+1$; $i^{(j)} \leftarrow i$; $(\varphi^{(j)}, \sigma_1^{(j)}) \leftarrow \mathsf{DOS.sign}_1(\lambda, sk_i)$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{j\}$; **return** $(j, \sigma_1^{(j)})$
Oracle $\mathtt{sign2}(j, m)$**:**
if $(j \notin \mathcal{Q}_1) \vee (j \in \mathcal{Q}_2)$**: return** $\perp$
$m^{(j)} \leftarrow m$; $\sigma_2^{(j)} \leftarrow \mathsf{DOS.sign}_2(\varphi^{(j)}, m^{(j)})$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{j\}$; **return** $\sigma_2^{(j)}$
Oracle $\mathtt{sign12}(i, m)$**:**
// $i \in [n]$; queried at most q times for each i
$(j, \sigma_1^{(j)}) \leftarrow \mathtt{sign1}(i)$; $\sigma_2^{(j)} \leftarrow \mathtt{sign2}(j, m)$
return $(j, \sigma_1^{(j)}, \sigma_2^{(j)})$

Figure 2: Attack game defining (k,n)-X-CMA security. Boxed parts are evaluated only in sUF game.

Game $\mathsf{G}_{\mathsf{DOTK}}^{(k,n,q)\text{-IND-tag-CCA}}(\mathcal{A})$**:**
$b \leftarrow_{\$} \{0,1\}$; $\lambda \leftarrow \mathsf{DOTK.init}()$
for $i \in [n]$**:** $(sk_i, pk_i) \leftarrow \mathsf{DOTK.gen}(\lambda)$
$\mathcal{O}_1 \leftarrow (\mathtt{enc12}, \mathtt{dec})$; $\mathcal{O}_2 \leftarrow (\mathtt{enc1}, \mathtt{enc2}, \mathtt{dec})$
$b^* \leftarrow \mathcal{A}(\lambda, \{pk_i\}_{i\in[n]};\ \mathcal{O}_k)$; **return** $(b^* = b)$
Oracle $\mathtt{enc1}(\tau)$**:**
// queried at most nq times
$j \leftarrow j+1$; $\tau^{(j)} \leftarrow \tau$; $(\varphi^{(j)}, C_1^{(j)}) \leftarrow \mathsf{DOTK.enc}_1(\lambda, \tau)$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{j\}$; **return** $(j, C_1^{(j)})$
Oracle $\mathtt{enc2}(j, i)$**:**
// $i \in [n]$; queried at most q times for each i
if $(j \notin \mathcal{Q}_1) \vee (j \in \mathcal{Q}_2)$**: return** $\perp$
$i^{(j)} \leftarrow i$; $C_2^{(j)} \leftarrow \mathsf{DOTK.enc}_2(\varphi^{(j)}, pk_{i^{(j)}})$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{j\}$; **return** $C_2^{(j)}$
Oracle $\mathtt{enc12}(\tau, i)$**:**
// $i \in [n]$; queried at most q times for each i
$(j, C_1^{(j)}) \leftarrow \mathtt{enc1}(\tau)$; $C_2^{(j)} \leftarrow \mathtt{enc2}(j, i)$
if $C_2^{(j)} = \perp$**: return** $\perp$ **else: return** $(j, C_1^{(j)}, C_2^{(j)})$
Oracle $\mathtt{dec}(i, \tau, C_1, C_2)$**:**
// $i \in [n]$; queried at most q_d times
if $\exists j.(i, \tau, C_1, C_2) = (i^{(j)}, \tau^{(j)}, C_1^{(j)}, C_2^{(j)})$**: return** $\perp$
return $K \leftarrow \mathsf{DOTK.dec}(\lambda, sk_i, \tau, C_1, C_2)$

Figure 3: Attack game defining (k,n,q)-IND-tag-CCA security.

We say that DOTK is (t, q_d, ϵ)-(k,n,q)-IND-tag-CCA-secure if $\mathrm{Adv}_{\mathrm{DOTK}}^{(k,n,q)\text{-IND-tag-CCA}}(\mathscr{A}) \leq \epsilon$ holds for any adversary $\mathscr{A}$ that runs in time t and makes at most q_d queries to the oracle dec.

Construction. Here, we give a concrete example of $(1,1)$-IND-tag-CCA-secure divisible online/offline tag-based KEM scheme with lightweight enc_2 algorithm that requires only one regular exponentiation in $\mathbb{G}_T$ of bilinear groups.

The construction of our tag-based KEM is similar to the one of partitioned KEM [14] that is based on the IBE-based KEM from Boyen et al. [17, 18]. The concrete scheme $\mathrm{DOTK}_{\mathrm{BMW}}$ is described in Figure 4, where $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$ is a bilinear group, i.e., $\mathbb{G}$, $\widehat{\mathbb{G}}$, and $\mathbb{G}_T$ are cyclic groups of prime order p, equipped with a bilinear map $e : \mathbb{G} \times \widehat{\mathbb{G}} \longrightarrow \mathbb{G}_T$ (a bilinear map $e : \mathbb{G} \times \widehat{\mathbb{G}} \longrightarrow \mathbb{G}_T$ satisfies the following properties: (1) bilinearity: $e(g^a, h^b) = e(g, h)^{ab}$ for any $g \in \mathbb{G}$, any $h \in \widehat{\mathbb{G}}$, and any $a, b \in \mathbb{Z}$; (2) efficient computability for any input pair; (3) nondegeneracy: $e(g, h) \neq 1_{\mathbb{G}_T}$ for any $g \in \mathbb{G}/1_{\mathbb{G}}$ and any $h \in \widehat{\mathbb{G}}/1_{\widehat{\mathbb{G}}}$), $H : \mathbb{G} \longrightarrow \mathbb{Z}_p$ is a target collision resistant hash function, and CH is a secure chameleon hash function. As with the partitioned KEM in [14], our scheme has no second partial ciphertext, i.e., $C_2 = \varepsilon$, which is desirable when algorithm enc_2 is executed in the bandwidth-restricted environment. We can verify that the scheme is $(1, 1)$-IND-tag-CCA-secure if the DBDH (Decisional Bilinear Diffie–Hellman) assumption on $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$ holds. The proof is similar with the original proof in [17].

3.4. Signcryptions. The two features of MDO cryptography, i.e., incremental processing and sending, can show their full advantages when applying to signcryptions, where the sender-side algorithm has multiple input arguments that can be given incrementally in most cases. Therefore, in the rest of this paper, we focus on the notion of *multidivisible online/offline signcryption (MDOSC)*.

In MDOSC schemes, a signcryption algorithm is split into three subalgorithms, each of which can be processed incrementally, and their output can be transmitted incrementally. A formal definition, security notions, and a generic construction of MDOSC are described in the following sections.

4. Multidivisible Online/Offline Signcryptions

We first introduce multidivisible online/offline signcryption (MDOSC). Then, we formally define its security models in terms of confidentiality and unforgeability.

4.1. Syntax. Definition 4. A multidivisible online/offline signcryption scheme, MDOSC, consists of the following algorithms:

$\mathrm{init}(\,) \longrightarrow \lambda$ (the public parameter generation algorithm): it outputs public parameters λ to be used by all parties

$\mathrm{gen}_{\mathrm{S}}(\lambda) \longrightarrow (\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}})$ (the sender key generation algorithm): it generates a private/public key pair $(\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}})$ for a sender

$\mathrm{gen}_{\mathrm{R}}(\lambda) \longrightarrow (\mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}})$ (the receiver key generation algorithm): it generates a private/public key pair $(\mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}})$ for a receiver

$\mathrm{sc}_1(\lambda, \mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}}) \longrightarrow (\varphi_1, \Sigma_1)$ (the first signcryption algorithm): given public parameter λ and sender's key pair $(\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}})$, it outputs state information φ_1 and partial ciphertext Σ_1

$\mathrm{sc}_2(\varphi_1, \mathrm{pk}_{\mathrm{R}}) \longrightarrow (\varphi_2, \Sigma_2)$ (the second signcryption algorithm): given state information φ_1 and recipient's public key pk_{R}, it outputs new state information φ_2 and partial ciphertext Σ_2

$\mathrm{sc}_3(\varphi_2, m) \longrightarrow \Sigma_3$ (the third signcryption algorithm): given state information φ_2 and plaintext m, it outputs partial ciphertext Σ_3

$\mathrm{unsc}(\lambda, \mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{S}}, \Sigma_1, \Sigma_2, \Sigma_3) \longrightarrow m$ (the deterministic unsigncryption algorithm): given public parameters λ, recipient's key pair $(\mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}})$, sender's public key pk_{S}, and partial ciphertexts Σ_1, Σ_2, and Σ_3, it outputs either plaintext m or error symbol $\perp$

We require the correctness for MDOSC, namely, for any $\lambda \longleftarrow \mathrm{init}(\,)$, any $(\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}}) \longleftarrow \mathrm{gen}_{\mathrm{S}}(\lambda)$, any $(\mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}}) \longleftarrow \mathrm{gen}_{\mathrm{R}}(\lambda)$, any m, any $(\varphi_1, \Sigma_1) \longleftarrow \mathrm{sc}_1(\lambda, \mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}})$, and any $(\varphi_2, \Sigma_2) \longleftarrow \mathrm{sc}_2(\varphi_1, \mathrm{pk}_{\mathrm{R}})$, we have $\mathrm{unsc}(\lambda, \mathrm{sk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{R}}, \mathrm{pk}_{\mathrm{S}}, \Sigma_1, \Sigma_2, \mathrm{sc}_3(\varphi_2, m)) = m$.

We say that a MDOSC scheme is a 2-divisible online/offline signcryption (2-DOSC, for short) scheme if Σ_2 is not null. Similarly, if Σ_1 is not null, we call it as 3-DOSC scheme.

4.2. Confidentiality. As for confidentiality, we define the indistinguishability against insider-chosen ciphertext attacks (or insider-chosen plaintext attacks) in the dynamic multiuser model, based on the same notion for standard (i.e., nondivisible) signcryption [19]. As with the indistinguishability of MDO encryption [14], we introduce the level of ciphertext divisibility, k, to express the following three distinct situations: ($k = 1$) all the ciphertext Σ_1, Σ_2, and Σ_3 are made public at the same time; ($k = 2$) Σ_1 and Σ_2 are made public at the same time, then Σ_3 is published; ($k = 3$) Σ_1, Σ_2, and Σ_3 are made public one by one. Notice that the case of $k = 1$ is covered by the standard dM-IND-iCCA notion for signcryptions [19], whereas the case of $k = 2, 3$ corresponds to the incremental sending situation of MDO cryptography. Adversaries in the latter situations have more capabilities; for example, the adversary in the case of $k = 3$ can choose the target recipient after observing Σ_1 and can choose challenge messages (m_0, m_1) after knowing Σ_2 as well as Σ_1. Our definition captures these types of adversaries using three signcryption oracles.

Definition 5. Let $k \in \{1, 2, 3\}$ and $n, q \in \mathbb{N}$. Let DOSC be a MDOSC scheme, and let $\mathscr{A}$ be an adversary. The (k, n, q)-dM-IND-iCCA-advantage of $\mathscr{A}$ against DOSC is defined as

$$\mathrm{Adv}_{\mathrm{DOSC}}^{(k,n,q)\text{-dM-IND-iCCA}}(\mathscr{A}) = \left|2\Pr\left[G_{\mathrm{DOSC}}^{(k,n,q)\text{-dM-IND-iCCA}}(\mathscr{A}) = 1\right] - 1\right|, \tag{5}$$

where attack game $G_{\mathrm{DOSC}}^{(k,n,q)\text{-dM-IND-iCCA}}$ is defined in Figure 5.

We say that DOSC is (t, q_d, ϵ)-(k, n, q)-dM-IND-iCCA-secure if $\mathrm{Adv}_{\mathrm{DOSC}}^{(k,n,q)\text{-dM-IND-iCCA}}(\mathscr{A}) \leq \epsilon$ holds for any adversary $\mathscr{A}$ that runs in time t and makes at most q_d queries to the oracle unsc.

$\mathsf{init}()$:
$g \leftarrow_\$ \mathbb{G}$; $h \leftarrow_\$ \hat{\mathbb{G}}$; $y_1, y_2 \leftarrow_\$ \mathbb{Z}_p^*$; $u_1 \leftarrow g^{y_1}$; $u_2 \leftarrow g^{y_2}$ $v_1 \leftarrow h^{y_1}$; $v_2 \leftarrow h^{y_2}$
$(tk, hk) \leftarrow \mathsf{CH.gen}()$; **return** $\lambda \leftarrow (g, h, u_1, u_2, v_1, v_2, hk)$

$\mathsf{gen}(\lambda)$:
$x \leftarrow_\$ \mathbb{Z}_p^*$; $h_0 \leftarrow h^x$; $z \leftarrow e(g, h_0)$; **return** $(sk, pk) \leftarrow (h_0, z)$

$\mathsf{enc}_1(\lambda, \tau)$:
$r \leftarrow_\$ \mathsf{CH}.\mathcal{R}$; $\tau' \leftarrow \mathsf{CH.hash}(hk, \tau, r)$
$t \leftarrow_\$ \mathbb{Z}_p^*$; $c_1 \leftarrow g^t$; $w \leftarrow H(\langle \tau' \| c_1 \rangle)$; $c_2 \leftarrow (u_1 u_2^w)^t$
$\varphi_1 \leftarrow (\lambda, t)$; $C_1 \leftarrow (c_1, c_2, r)$; **return** (φ_1, C_1)

$\mathsf{enc}_2(\varphi_1, pk)$:
$(\lambda, t) \leftarrow \varphi_1$; $z \leftarrow pk$; $K \leftarrow z^t$; $C_2 \leftarrow \varepsilon$; **return** (K, C_2)

$\mathsf{dec}(\lambda, sk, \tau, C_1, C_2)$:
if $C_2 \neq \varepsilon$: **return** $\perp$
$h_0 \leftarrow sk$; $(c_1, c_2, r) \leftarrow C_1$; $\tau' \leftarrow \mathsf{CH.hash}(hk, \tau, r)$; $w \leftarrow H(\langle \tau' \| c_1 \rangle)$
if $e(c_1, v_1 v_2^w) \neq e(c_2, h)$: **return** $\perp$ **else: return** $K \leftarrow e(c_1, h_0)$

Figure 4: Construction of $\mathrm{DOTK}_{\mathrm{BMW}}$.

Game $\mathsf{G}_{\mathsf{DOSC}}^{(k,n,q)\text{-dM-IND-iCCA}}(\mathcal{A})$:
$b \leftarrow_\$ \{0, 1\}$; $\lambda \leftarrow \mathsf{DOSC.init}()$
for $i \in [n]$: $(sk_{\mathsf{R},i}, pk_{\mathsf{R},i}) \leftarrow \mathsf{DOSC.gen_R}(\lambda)$
$\mathcal{O}_1 \leftarrow (\mathtt{sc123}, \mathtt{unsc})$
$\mathcal{O}_2 \leftarrow (\mathtt{sc12}, \mathtt{sc3}, \mathtt{unsc})$
$\mathcal{O}_3 \leftarrow (\mathtt{sc1}, \mathtt{sc2}, \mathtt{sc3}, \mathtt{unsc})$
$b^* \leftarrow \mathcal{A}(\lambda, \{pk_{\mathsf{R},i}\}_{i \in [n]}; \mathcal{O}_k)$; **return** $(b^* = b)$

Oracle $\mathtt{sc1}(sk_{\mathsf{S}}, pk_{\mathsf{S}})$:
// queried $\leq nq$ times
$j \leftarrow j + 1$; $(sk_{\mathsf{S}}^{(j)}, pk_{\mathsf{S}}^{(j)}) \leftarrow (sk_{\mathsf{S}}, pk_{\mathsf{S}})$
$(\varphi_1^{(j)}, \Sigma_1^{(j)}) \leftarrow \mathsf{DOSC.sc}_1(\lambda, sk_{\mathsf{S}}^{(j)}, pk_{\mathsf{S}}^{(j)})$; $\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{j\}$
return $(j, \Sigma_1^{(j)})$

Oracle $\mathtt{sc2}(j, i)$:
// $i \in [n]$; queried $\leq q$ times for each i
if $(j \notin \mathcal{Q}_1) \vee (j \in \mathcal{Q}_2)$: **return** $\perp$
$i^{(j)} \leftarrow i$; $(\varphi_2^{(j)}, \Sigma_2^{(j)}) \leftarrow \mathsf{DOSC.sc}_2(\varphi_1^{(j)}, pk_{\mathsf{R},i^{(j)}})$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{j\}$; **return** $\Sigma_2^{(j)}$

Oracle $\mathtt{sc3}(j, m_0, m_1)$:
// $|m_0| = |m_1|$; queried $\leq nq$ times
if $(j \notin \mathcal{Q}_2) \vee (j \in \mathcal{Q}_3)$: **return** $\perp$
$(m_0^{(j)}, m_1^{(j)}) \leftarrow (m_0, m_1)$; $\Sigma_3^{(j)} \leftarrow \mathsf{DOSC.sc}_3(\varphi_2^{(j)}, m_b^{(j)})$
$\mathcal{Q}_3 \leftarrow \mathcal{Q}_3 \cup \{j\}$; **return** $\Sigma_3^{(j)}$

Oracle $\mathtt{sc12}(sk_{\mathsf{S}}, pk_{\mathsf{S}}, i)$:
// $i \in [n]$; queried $\leq q$ times for each i
$(j, \Sigma_1^{(j)}) \leftarrow \mathtt{sc1}(sk_{\mathsf{S}}, pk_{\mathsf{S}})$; $\Sigma_2^{(j)} \leftarrow \mathtt{sc2}(j, i)$
if $\Sigma_2^{(j)} = \perp$: **return** $\perp$ **else: return** $(j, \Sigma_1^{(j)}, \Sigma_2^{(j)})$

Oracle $\mathtt{sc123}(sk_{\mathsf{S}}, pk_{\mathsf{S}}, i, m_0, m_1)$:
// $i \in [n]$; $|m_0| = |m_1|$; queried $\leq q$ times for each i
$(j, \Sigma_1^{(j)}, \Sigma_2^{(j)}) \leftarrow \mathtt{sc12}(sk_{\mathsf{S}}, pk_{\mathsf{S}}, i)$; $\Sigma_3^{(j)} \leftarrow \mathtt{sc3}(j, m_0, m_1)$
if $\Sigma_3^{(j)} = \perp$: **return** $\perp$ **else: return** $(j, \Sigma_1^{(j)}, \Sigma_2^{(j)}, \Sigma_3^{(j)})$

Oracle $\mathtt{unsc}(pk_{\mathsf{S}}, i, \Sigma_1, \Sigma_2, \Sigma_3)$:
// $i \in [n]$; queried $\leq q_d$ times
if $\exists j.(pk_{\mathsf{S}}, i, \Sigma_1, \Sigma_2, \Sigma_3) = (pk_{\mathsf{S}}^{(j)}, i^{(j)}, \Sigma_1^{(j)}, \Sigma_2^{(j)}, \Sigma_3^{(j)})$:
return $\perp$
return $m \leftarrow \mathsf{DOSC.unsc}(\lambda, sk_{\mathsf{R},i}, pk_{\mathsf{R},i}, pk_{\mathsf{S}}, \Sigma_1, \Sigma_2, \Sigma_3)$

Figure 5: Attack game defining (k, n, q)-dM-IND-iCCA security.

In addition, the (k, n, q)-dM-IND-iCPA-advantage of $\mathscr{A}$ against DOSC is defined in a similar fashion with (k, n, q)-dM-IND-iCCA-advantage, except that the adversary has no access to oracle unsc, i.e., $q_d = 0$, in game $G_{\mathrm{DOSC}}^{(k,n,q)\text{-dM-IND-iCPA}}$.

4.3. Unforgeability. As for unforgeability, we define the (strong) unforgeability against insider chosen-message attacks in the dynamic multiuser model. Similar to the confidentiality notion, we use the level k of divisibility to express the three distinct situations. Note that the case of $k = 1$ is equivalent to the standard dM-sUF-iCMA for signcryptions [19]. The adversary in the case of $k = 3$ can take more flexible strategy: it can decide the target recipient key pk_{R} after observing Σ_1 and can query the signcryption oracle with a message m after knowing Σ_2 as well as Σ_1.

Definition 6. Let $k \in \{1, 2, 3\}$ and $n, q \in \mathbb{N}$. Let $X \in \{\mathrm{UF}, \mathrm{sUF}\}$. Let DOSC be a MDOSC scheme, and let $\mathscr{A}$ be an adversary. The (k, n)-dM-X-iCMA-advantage of $\mathscr{A}$ against DOSC is defined as

$$\mathrm{Adv}_{\mathrm{DOSC}}^{(k,n)\text{-dM-X-iCMA}}(\mathscr{A}) = \Pr\left[G_{\mathrm{DOSC}}^{(k,n)\text{-dM-X-iCMA}}(\mathscr{A}) = 1\right], \tag{6}$$

where the attack game $G_{\mathrm{DOSC}}^{(k,n)\text{-dM-X-iCMA}}$ is defined in Figure 6.

We say that DOSC is (t, q_s, ϵ)-(k, n)-dM-X-iCMA-secure if $\mathrm{Adv}_{\mathrm{DOSC}}^{(k,n)\text{-dM-X-iCMA}}(\mathscr{A}) \le \epsilon$ holds for any adversary $\mathscr{A}$ that runs in time t.

5. Relations between Security Notions for MDOSCs

In this section, we clarify the relations between the security notions defined in the previous section in terms of confidentiality and unforgeability. The result is summarized in Figure 7, where (k, n, q) denotes the notion of (k, n, q)-dM-IND-iCCA-security in the left-hand side, whereas (k, n) denotes the notion of (k, n)-dM-sUF-iCMA-security in the right-hand side. Note that for any $k, k' \in \{1, 2, 3\}$, $n, n' \in \mathbb{N}$, and $q, q' \in \mathbb{N}$ such that $k \ge k'$, $n \ge n'$, and $q \ge q'$, our (k, n, q)-dM-IND-iCCA trivially implies (k', n', q')-dM-IND-iCCA and (k, n, q)-dM- IND-iCPA. Similarly, (k, n)-dM-sUF-iCMA trivially implies (k', n')-dM-sUF-iCMA and (k, n)-dM-UF-iCMA.

5.1. Confidentiality. The following two theorems say that for any $k \in \{1, 2, 3\}$, $(k, 1, 1)$-dM-IND-iCCA-security implies (k, n, q)-dM-IND-iCCA-security with reduction loss polynomial in the number n of receivers as well as the number q of challenge queries.

Theorem 1 ($(\{1, 2\}, 1, 1)$-iCCA $\longrightarrow$ $(\{1, 2\}, n, q)$-iCCA). *For $k \in \{1, 2\}$, assume a k-DOSC scheme DOSC is (t', q_d, ϵ)-$(k, 1, 1)$-dM-IND-iCCA-secure. Then, DOSC is also $(t, q_d, nq\epsilon)$-(k, n, q)-dM-IND-iCCA-secure, where $t = t' - O(nq)$.*

Using the hybrid argument in a similar fashion with [20], we can easily obtain the proof of Theorem 1.

As for the case of $k = 3$, the hybrid argument cannot be trivially applied due to the difficulty of the simulation of oracle sc_1 in $(3, n, q)$-dM-IND-iCCA, which is called nq times, using oracle sc_{1^*} in $(3, 1, 1)$-dM-IND-iCCA, which can be called only once. Responding to the adversary's query to sc_1 with $\Sigma_1^{(j)}$, the simulator for $(3, n, q)$-dM-IND-iCCA first chooses $(i^*, q^*) \in [n] \times [q]$ and uses its given single public key pk^* as a key pk_{i^*} of the user i^*. Then, the simulator uses its given oracle access to $sc1^*$ in $(3, 1, 1)$-dM-IND-iCCA *if* $\Sigma_1^{(j)}$ will be used for the q^*-th query related to the target user i^*, whereas it computes $\Sigma_1^{(j)}$ by the simulator itself otherwise. Here, the simulator has to guess whether $\Sigma_1^{(j)}$ will be used for the q^*-th challenge query to i^* before a subsequent invocation of sc_2. To overcome this issue, we use a naive guessing approach and obtain the following reduction:

Theorem 2 ($(3, 1, 1)$-iCCA $\longrightarrow$ $(3, n, q)$-iCCA). *Assume a 3-DOSC scheme DOSC is (t', q_d, ϵ)-$(3, 1, 1)$-dM-IND-iCCA-secure. Then, DOSC is also $(t, q_d, n^2q^2\epsilon)$-$(3, n, q)$-dM-IND-iCCA-secure, where $t = t' - O(nq)$.*

The proof is given in Appendix B.1.

Remark 1. We can apply the above argument to the multidivisible online/offline encryptions (MDOEs) to get a same polynomial reduction from $(3, 1, 1)$-CCA to $(3, n, q)$-CCA for MDOEs. The obtained reduction is significantly tighter than the previous one shown in [14] where only superpolynomial one was provided.

Next, we show that $(2, 1, 1)$-dM-IND-iCCA-security implies $(3, 1, 1)$-dM-IND-iCCA-security as follows:

Theorem 3 ($(2, 1, 1)$-iCCA $\Longrightarrow$ $(3, 1, 1)$-iCCA). *Assume a 3-DOSC scheme DOSC is (t, q_d, ϵ)-$(2, 1, 1)$-dM-IND-iCCA-secure. Then DOSC is also (t, q_d, ϵ)-$(3, 1, 1)$-dM-IND-iCCA-secure.*

Proof. In the single-user setting, i.e., $n = 1$, the adversary has no choice of the recipient i when it invokes oracle $\mathrm{sc}_2(j, i)$. Hence, the successive invocations of $\mathrm{sc}_1(\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}})$ and $\mathrm{sc}_2(j, i)$ in $(3, 1, 1)$-dM-IND-iCCA game is essentially equivalent to the invocation of the oracle $\mathrm{sc}_{12}(\mathrm{sk}_{\mathrm{S}}, \mathrm{pk}_{\mathrm{S}}, i)$ in $(2, 1, 1)$-dM-IND-iCCA.

Finally, we show the separation between $(1, \cdot, \cdot)$-dM-IND-iCCA and $(2, \cdot, \cdot)$-dM-IND-iCCA. Specifically, we provide a $(1, 1, 1)$-dM-IND-iCCA-secure but not $(2, 1, 1)$-dM-IND-iCCA-secure 3-DOSC scheme, $3\mathrm{DOSC}_{\mathrm{R}}$, as an evidence of the separation. Scheme $3\mathrm{DOSC}_{\mathrm{R}}$ is described in Figure 8. □

Theorem 4 ($(1, 1, 1)$-iCCA$\nrightarrow$$(2, 1, 1)$-iCPA). *Assume there exists a (divisible) online/offline tag-based KEM DOTK that is $(1, 1, 1)$-IND-tag-CCA-secure, a smooth divisible online/*

Game $\mathsf{G}^{(k,n)\text{-dM-X-iCMA}}_{\mathsf{DOSC}}(\mathcal{A})$**:**
$\lambda \leftarrow \mathsf{DOSC.init}()$
for $i \in [n]$**:** $(sk_{\mathsf{S},i}, pk_{\mathsf{S},i}) \leftarrow \mathsf{DOSC.gen_S}(\lambda)$
$(\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3) \leftarrow (\mathtt{sc123}, (\mathtt{sc12}, \mathtt{sc3}), (\mathtt{sc1}, \mathtt{sc2}, \mathtt{sc3}))$
$(i^*, sk^*_{\mathsf{R}}, pk^*_{\mathsf{R}}, \Sigma^*) \leftarrow \mathcal{A}(\lambda, \{pk_{\mathsf{S},i}\}_{i\in[n]};\ \mathcal{O}_k)$
$(\Sigma^*_1, \Sigma^*_2, \Sigma^*_3) \leftarrow \Sigma^*$
$m^* \leftarrow \mathsf{DOSC.unsc}(\lambda, sk^*_{\mathsf{R}}, pk^*_{\mathsf{R}}, pk_{\mathsf{S},i^*}, \Sigma^*_1, \Sigma^*_2, \Sigma^*_3)$
$v_1 \leftarrow (m^* \neq \perp)$
$v_2 \leftarrow \forall j.(i^*, pk^*_{\mathsf{R}}, m^* \boxed{, \Sigma^*}) \neq (i^{(j)}, pk^{(j)}_{\mathsf{R}}, m^{(j)} \boxed{, \Sigma^{(j)}})$
return $v_1 \wedge v_2$

Oracle $\mathtt{sc1}(i)$**:**
// $i \in [n]$; queried $\leq q$ times for each i
$j \leftarrow j + 1$; $i^{(j)} \leftarrow i$
$(\varphi^{(j)}_1, \Sigma^{(j)}_1) \leftarrow \mathsf{DOSC.sc}_1(\lambda, sk_{\mathsf{S},i^{(j)}}, pk_{\mathsf{S},i^{(j)}})$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{j\}$; **return** $(j, \Sigma^{(j)}_1)$

Oracle $\mathtt{sc2}(j, pk_{\mathsf{R}})$**:**
if $(j \notin \mathcal{Q}_1) \vee (j \in \mathcal{Q}_2)$**: return** $\perp$
$pk^{(j)}_{\mathsf{R}} \leftarrow pk_{\mathsf{R}}$; $(\varphi^{(j)}_2, \Sigma^{(j)}_2) \leftarrow \mathsf{DOSC.sc}_2(\varphi^{(j)}_1, pk^{(j)}_{\mathsf{R}})$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{j\}$; **return** $\Sigma^{(j)}_2$

Oracle $\mathtt{sc3}(j, m)$**:**
if $(j \notin \mathcal{Q}_2) \vee (j \in \mathcal{Q}_3)$**: return** $\perp$
$m^{(j)} \leftarrow m$; $\Sigma^{(j)}_3 \leftarrow \mathsf{DOSC.sc}_3(\varphi^{(j)}_2, m^{(j)})$
$\mathcal{Q}_3 \leftarrow \mathcal{Q}_3 \cup \{j\}$; $\Sigma^{(j)} \leftarrow (\Sigma^{(j)}_1, \Sigma^{(j)}_2, \Sigma^{(j)}_3)$; **return** $\Sigma^{(j)}_3$

Oracle $\mathtt{sc12}(i, pk_{\mathsf{R}})$**:**
// $i \in [n]$; queried $\leq q$ times for each i
$(j, \Sigma^{(j)}_1) \leftarrow \mathtt{sc1}(i)$; $\Sigma^{(j)}_2 \leftarrow \mathtt{sc2}(j, pk_{\mathsf{R}})$
if $\Sigma^{(j)}_2 = \perp$**: return** $\perp$ **else: return** $(j, \Sigma^{(j)}_1, \Sigma^{(j)}_2)$

Oracle $\mathtt{sc123}(i, pk_{\mathsf{R}}, m)$**:**
// $i \in [n]$; queried $\leq q$ times for each i
$(j, \Sigma^{(j)}_1, \Sigma^{(j)}_2) \leftarrow \mathtt{sc12}(i, pk_{\mathsf{R}})$; $\Sigma^{(j)}_3 \leftarrow \mathtt{sc3}(j, m)$
if $\Sigma^{(j)}_3 = \perp$**: return** $\perp$ **else: return** $(j, \Sigma^{(j)}_1, \Sigma^{(j)}_2, \Sigma^{(j)}_3)$

Figure 6: Attack game defining (k,n)-dM-X-iCMA security. Boxed parts are evaluated only in sUF game.

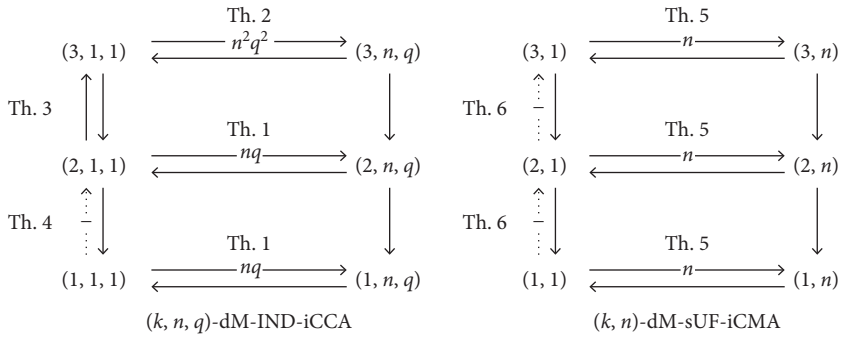

Figure 7: Implications and separations between the security notions for MDOSCs. In the left-hand side, "$(k,n,q) \longrightarrow (k', 3n', q')$" with overlapped X denotes that (k,n,q)-dM-IND-iCCA-security implies (k', n', q')-dM-IND-iCCA-security with reduction's loss X, while the dotted arrow denotes that (k,n,q)-dM-IND-iCCA-security does not necessarily imply (k', n', q')-dM-IND-iCCA-security. The ones for (k,n)-dM-sUF-iCMA are similarly represented in the right-hand side.

FIGURE 8: Constructions of 3DOSC$_R$, 3DOSC$_{S1}$, and 3DOSC$_{S2}$ used to prove Theorem 4 and Theorem 6. Boxed parts with superscript X are evaluated only in 3DOSC$_X$.

offline signature scheme DOS, and a IND-OTCCA-secure DEM DEM, where KEM.$\mathcal{K}$ = DEM.$\mathcal{K}$. Then, there exists a 3-DOSC scheme 3DOSC$_R$ which is secure in the sense of $(1,1,1)$-dM-IND-iCCA but which is not secure in the sense of $(2,1,1)$-dM-IND-iCCA.

The proof is given in Appendix B.2.

5.2. Unforgeability. First, we show the following implications with regard to the number of senders.

Theorem 5 (($\{1,2,3\},1$)-sUF $\longrightarrow$ ($\{1,2,3\},n$)-sUF). *For $k \in \{1,2,3\}$, if a k-DOSC scheme DOSC is (t', q, ϵ)-$(k,1)$-dM-sUF-iCMA-secure, then DOSC is also $(t, q, n\epsilon)$-(k,n)-dM-sUF-iCMA-secure, where $t = t' - O(nq)$.*

Proof. The adversary attacking $(k,1)$-dM-sUF-iCMA can simulate signcryption oracles for the adversary attacking (k,n)-dM-sUF-iCMA, using its given oracles (for the single target sender) as well as its generating n-1 sender key pairs. Contrary to the confidentiality notions, note that the first signcryption oracle, sc$_1$, has the sender index i as its input so that its simulation can be done without any guessing.

Next, we show separations among $(k,\cdot)$-dM-sUF-iCMA for $k \in \{1,2,3\}$. □

Theorem 6 (($1,1$)-sUF $\nrightarrow$ ($2,1$)-sUF $\longrightarrow$ ($3,1$)-sUF). *Assume there exists a $(2,1)$-sUF-CMA-secure and smooth DOS. Then, there exists 3-DOSC schemes 3DOSC$_{S1}$ and 3DOSC$_{S2}$ such that the former is secure in the sense of $(1,1)$-dM-sUF-iCMA but which is not secure in the sense of*

$(2,1)$-*dM-UF-iCMA, whereas the latter is secure in the sense of* $(2,1)$-*dM-sUF-iCMA but which is not secure in the sense of* $(3,1)$-*dM-UF-iCMA*.

Proof (sketch). $3\mathrm{DOSC}_{\mathrm{S1}}$ and $3\mathrm{DOSC}_{\mathrm{S2}}$ are described in Figure 8. In order to show that $3\mathrm{DOSC}_{\mathrm{S1}}$ (or $3\mathrm{DOSC}_{\mathrm{S2}}$) is not $(2,1)$-dM-UF-iCMA-secure (or $(3,1)$-dM-UF-iCMA-secure), we construct adversary $\mathscr{A}_1$ (or $\mathscr{A}_2$) shown in Figure 9. Note that oracle sc_3 (or sc_2) always generates $d_{\mathrm{S}}^* = \mathrm{sk}_{\mathrm{S}}$ as an answer to the query $m = \sigma_1^*$ so that $\mathscr{A}$ can output any forgery using sk_{S}.

On the other hand, we can show that $3\mathrm{DOSC}_{\mathrm{S1}}$ (or $3\mathrm{DOSC}_{\mathrm{S2}}$) is $(1,1)$-dM-sUF-iCMA-secure (or $(2,1)$-dM-sUF-iCMA-secure) in a similar fashion with the proof of Theorem 8 shown later. □

6. MDOSC Construction

In this section, we provide a generic construction of MDOSC with security proofs and compare it with previous signcryption schemes.

Using a (divisible) online/offline tag-based KEM, a divisible online/offline signature scheme, and a DEM as building blocks, we can derive 3-DOSC scheme 3DOSC. 3DOSC is described in Figure 8 without all boxed parts.

6.1. Security. The following theorems guarantee the security of our construction.

Theorem 7 (3DOSC is $(2,1,1)$-iCCA-secure). *If DOTK is* $(t', q_d', \epsilon_{DOTK})$-$(1,1,1)$-*IND-tag-CCA-secure and DEM is* $(t', q_d', \epsilon_{DEM})$-*IND-P0-C2-secure then 3DOSC is* (t, q_d, ϵ)-$(2,1,1)$-*dM-IND-iCCA-secure, where* $t = t' - O(q_d)$, $q_d = q_d'$, *and* $\epsilon = 2\epsilon_{DOTK} + \epsilon_{DEM}$.

We can prove this theorem in a similar fashion with the proof of Lemma B.2 described in Appendix B.2.

Theorem 8 (3DOSC is $(3,1)$-sUF-secure). *If DOS is* (t', q', ϵ')-$(2,1)$-*sUF-CMA-secure and DEM is one-to-one then 3DOSC is* (t, q, ϵ)-$(3,1)$-*dM-sUF- iCMA-secure, where* $t = t' - O(q)$, $q = q'$, *and* $\epsilon = \epsilon'$.

Proof. Assume we have a t-time adversary $\mathscr{A}$ that makes at most q queries to oracles sc_1, sc_2, and sc3 in order to break $(3,1)$-dM-sUF-iCMA security of 3DOSC. We will construct adversary $\mathscr{B}$ exploiting $\mathscr{A}$ as a black box to attack $(2,1)$-sUF-CMA security of DOS. For any $\mathscr{A}$, we will show the following relation holds:

$$\mathrm{Adv}_{3\mathrm{DOSC}}^{(3,1)\text{-dM-sUF-iCMA}}(\mathscr{A}) \le \mathrm{Adv}_{\mathrm{DOS}}^{(2,1)\text{-sUF-CMA}}(\mathscr{B}). \tag{7}$$

In order to prove equation (1), we use G_0 and G_1 described in Figure 10. As with the proof of Lemma B.2, we simplify the game description by omitting or modifying the statements and the variables that play no role when $n = 1$.

Game G_0 is equivalent to the original attack game played by $\mathscr{A}$ against 3DOSC. Hence, we have

$$\Pr\left[G_{3\mathrm{DOSC}}^{(3,1)\text{-dM-sUF-iCMA}}(\mathscr{A}) = 1\right] = \Pr\left[G_0(\mathscr{A}) = 1\right]. \tag{8}$$

Game G_1 is identical to G_0 except the following: computations of DOS.sign1 and DOS.sign2 are replaced by oracle invocations of sign1 and sign2, respectively; one of the winning condition ν_2 is replaced by ν_2'. Note that sign1, sign2, and ν_2' are exactly the ones in $G_{\mathrm{DOS}}^{(2,1)\text{-sUF-CMA}}$ described in Figure 2.

Oracles sign1 and sign2 here can be used to simulate the original behavior of G_0 so that $\mathscr{A}$'s views in these two games are equivalent.

In addition, we claim that $\nu_2 = 1$ implies $\nu_2' = 1$. Let us assume $\nu_2' = 0$, i.e., there exists some j^* such that $(\mathrm{pk}_{\mathrm{R}}^{(j^*)}, C_1^{(j^*)}, C_2^{(j^*)}, m^{(j^*)}, \sigma_1^{(j^*)}, \sigma_2^{(j^*)})$ equals to $(\mathrm{pk}_{\mathrm{R}}^*, C_1^*, C_2^*, m^*, \sigma_1^*, \sigma_2^*)$. Then, $K^{(j^*)} = K^*$ also holds because of the correctness property of DOTK. Furthermore, $C_3^{(j^*)} = C_3^*$ also holds because

$$\begin{aligned} C_3^{(j^*)} &= \mathrm{DEM.enc}\left(\lambda_D, K^{(j^*)}, \left(\sigma_2^{(j^*)}, m^{(j^*)}\right)\right) \\ &= \mathrm{DEM.enc}\left(\lambda_D, K^*, \left(\sigma_2^*, m^*\right)\right) \\ &= C_3^*, \end{aligned} \tag{9}$$

where the second equality follows from one-to-one property of DEM. Hence, $\nu_2 = 0$ holds since we now have j' such that $(\mathrm{pk}_{\mathrm{R}}^{(j^*)}, m^{(j^*)}, (C_1^{(j^*)}, \sigma_1^{(j^*)}), C_2^{(j^*)}, C_3^{(j^*)})$ equals to $(\mathrm{pk}_{\mathrm{R}}^*, m^*, (C_1^*, \sigma_1^*), C_2^*, C_3^*)$.

Therefore, it follows that

$$\Pr\left[G_0(\mathscr{A}) = 1\right] \le \Pr\left[G_1(\mathscr{A}) = 1\right]. \tag{10}$$

We now construct adversary $\mathscr{B}$ satisfies the following relation:

$$\Pr\left[G_1(\mathscr{A}) = 1\right] = \mathrm{Adv}_{\mathrm{DOS}}^{(2,1)\text{-sUF-CMA}}(\mathscr{B}). \tag{11}$$

The description of adversary $\mathscr{B}$ is shown in Figure 11. We can see adversary $\mathscr{B}$ makes at most q queries to sign1 as well as sign2, and its running time $t' = t + O(q)$. It is easy to verify that $\mathscr{A}$'s view in $G_{\mathrm{DOS}}^{(2,1)\text{-sUF-CMA}}(\mathscr{B})$ is equivalent to the one in G_1 so that the above relation holds.

In summary, we now have equation (1). Combining it with the assumptions that DOS is (t', q', ϵ')-$(2,1)$-sUF-CMA-secure, we have $\mathrm{Adv}_{3\mathrm{DOSC}}^{(3,1)\text{-dM-sUF-iCMA}}(\mathscr{A}) \le \epsilon'$, which implies that 3DOSC is (t, q, ϵ)-$(3,1)$-dM-sUF-iCMA-secure for $\epsilon = \epsilon'$, as desired.

From Theorems 2, 3, and 5, it follows that 3DOSC has the strongest security in the sense of both confidentiality and unforgeability: □

Corollary 1 (3DOSC is $(3,n,q)$-iCCA-secure). *Suppose DOTK is* $(t', q_d', \epsilon_{DOTK})(1,1,1)$-*IND-tag-CCA-secure and DEM is* $(t', q_d', \epsilon_{DEM})$-*IND-P0-C2-secure. Then, 3DOSC is* (t, q_d, ϵ)-$(3,n,q)$-*dM-IND-iCCA-secure, where* $t = t' - O(q_d)$, $q_d = q_d'$, *and* $\epsilon = 2n^2q^2\epsilon_{DOTK} + n^2q^2\epsilon_{DEM}$.

Adversary $\mathcal{A}_1(\lambda, pk_{\mathsf{S}}^*;\ \mathtt{sc12}, \mathtt{sc3})$**:**
$(sk_{\mathsf{R}}, pk_{\mathsf{R}}) \leftarrow \mathsf{DOTK.gen}(\lambda_{\mathsf{K}})$
$(1, \Sigma_1^{(1)}, \Sigma_2^{(1)}) \leftarrow \mathtt{sc12}(1, pk_{\mathsf{R}})$
$(C_1^{(1)}, \sigma_1^{(1)}) \leftarrow \Sigma_1^{(1)}; \quad C_2^{(1)} \leftarrow \Sigma_2^{(1)}$
$m \leftarrow \sigma_1^{(1)}; \quad \Sigma_3^{(1)} \leftarrow \mathtt{sc3}(1, m); \quad C_3^{(1)} \leftarrow \Sigma_3^{(1)}$
$K^{(1)} \leftarrow \mathsf{DOTK.dec}(\lambda_{\mathsf{K}}, sk_{\mathsf{R}}, (pk_{\mathsf{S}}^*, \sigma_1^{(1)}), C_1^{(1)}, C_2^{(1)})$
$(\sigma_2^{(1)}, m^{(1)}, d_{\mathsf{S}}^{(1)}) \leftarrow \mathsf{DEM.dec}(\lambda_{\mathsf{D}}, K^{(1)}, C_3^{(1)})$
return *any forgery generated by using* $d_{\mathsf{S}}^{(1)}$ *as a signing key*

Adversary $\mathcal{A}_2(\lambda, pk_{\mathsf{S}}^*;\ \mathtt{sc1}, \mathtt{sc2}, \mathtt{sc3})$**:**
$(sk_{\mathsf{R}}, pk_{\mathsf{R}}) \leftarrow \mathsf{DOTK.gen}(\lambda_{\mathsf{K}})$
$(1, \Sigma_1^{(1)}) \leftarrow \mathtt{sc1}(1); \quad (C_1^{(1)}, \sigma_1^{(1)}) \leftarrow \Sigma_1^{(1)}$
$pk_{\mathsf{R}} \leftarrow \sigma_1^{(1)}; \quad \Sigma_2^{(1)} \leftarrow \mathtt{sc2}(1, pk_{\mathsf{R}}); \quad (C_2^{(1)}, d_{\mathsf{S}}^{(1)}) \leftarrow \Sigma_2^{(1)}$
return *any forgery generated by using* $d_{\mathsf{S}}^{(1)}$ *as a signing key*

FIGURE 9: Description of adversary $\mathscr{A}_1$ (or $\mathscr{A}_2$) attacking $(2,1)$-dM-sUF-iCMA (or $(3,1)$-dM-sUF-iCMA) security of $3\mathrm{DOSC}_{\mathsf{S1}}$ (or $3\mathrm{DOSC}_{\mathsf{S2}}$).

Game $\mathtt{G}_i(\mathcal{A})$**:** `// for` $i \in [0,1]$
$\lambda_{\mathsf{K}} \leftarrow \mathsf{DOTK.init}(); \quad \lambda_{\mathsf{D}} \leftarrow \mathsf{DEM.init}(); \quad \lambda_{\mathsf{S}} \leftarrow \mathsf{DOS.init}()$
$\lambda \leftarrow (\lambda_{\mathsf{K}}, \lambda_{\mathsf{D}}, \lambda_{\mathsf{S}}); \quad (sk_{\mathsf{S}}^*, pk_{\mathsf{S}}^*) \leftarrow \mathsf{DOS.gen}(\lambda_{\mathsf{S}})$
$(sk_{\mathsf{R}}^*, pk_{\mathsf{R}}^*, \Sigma^*) \leftarrow \mathcal{A}(\lambda, pk_{\mathsf{S}}^*;\ \mathtt{sc1}, \mathtt{sc2}, \mathtt{sc3})$
$(\Sigma_1^*, \Sigma_2^*, \Sigma_3^*) \leftarrow \Sigma^*; \quad (C_1^*, \sigma_1^*) \leftarrow \Sigma_1^*; \quad C_2^* \leftarrow \Sigma_2^*; \quad C_3^* \leftarrow \Sigma_3^*$
$K^* \leftarrow \mathsf{DOTK.dec}(\lambda_{\mathsf{K}}, sk_{\mathsf{R}}^*, (pk_{\mathsf{S}}^*, \sigma_1^*), C_1^*, C_2^*)$
$(\sigma_2^*, m^*) \leftarrow \mathsf{DEM.dec}(\lambda_{\mathsf{D}}, K^*, C_3^*)$
$v_1 \leftarrow \mathsf{DOS.ver}(\lambda_{\mathsf{S}}, pk_{\mathsf{S}}^*, (pk_{\mathsf{R}}^*, C_1^*, C_2^*, m^*), \sigma_1^*, \sigma_2^*)$
$v_2 \leftarrow \forall j.(pk_{\mathsf{R}}^*, m^*, \Sigma^*) \neq (pk_{\mathsf{R}}^{(j)}, m^{(j)}, \Sigma^{(j)})$
$v_2' \leftarrow \forall j.((pk_{\mathsf{R}}^*, C_1^*, C_2^*, m^*), \sigma_1^*, \sigma_2^*) \neq ((pk_{\mathsf{R}}^{(j)}, C_1^{(j)}, C_2^{(j)}, m^{(j)}), \sigma_1^{(j)}, \sigma_2^{(j)})$
$\boxed{\textbf{return } v_1 \wedge v_2}^{\,0}$ $\boxed{\textbf{return } v_1 \wedge v_2'}^{\,1}$

Oracle $\mathtt{sc1}()$**:**
$j \leftarrow j + 1$
$\boxed{(\varphi_{\mathsf{S},1}^{(j)}, \sigma_1^{(j)}) \leftarrow \mathsf{DOS.sign}_1(\lambda_{\mathsf{S}}, sk_{\mathsf{S}}^*)}^{\,0}$
$\boxed{(j', \sigma_1^{(j)}) \leftarrow \mathtt{sign1}()}^{\,1}$
$(\varphi_{\mathsf{R},1}^{(j)}, C_1^{(j)}) \leftarrow \mathsf{DOTK.enc}_1(\lambda_{\mathsf{K}}, (pk_{\mathsf{S}}^*, \sigma_1^{(j)}))$
$\Sigma_1^{(j)} \leftarrow (C_1^{(j)}, \sigma_1^{(j)}); \quad \mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{j\}; \quad \textbf{return } (j, \Sigma_1^{(j)})$

Oracle $\mathtt{sc2}(j, pk_{\mathsf{R}})$**:**
if $(j \notin \mathcal{Q}_1) \vee (j \in \mathcal{Q}_2)$**: return** $\perp$
$pk_{\mathsf{R}}^{(j)} \leftarrow pk_{\mathsf{R}}; \quad (K^{(j)}, C_2^{(j)}) \leftarrow \mathsf{DOTK.enc}_2(\varphi_{\mathsf{R},1}^{(j)}, pk_{\mathsf{R}}^{(j)})$
$\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \cup \{j\}; \quad \textbf{return } \Sigma_2^{(j)} \leftarrow C_2^{(j)}$

Oracle $\mathtt{sc3}(j, m)$**:**
if $(j \notin \mathcal{Q}_2) \vee (j \in \mathcal{Q}_3)$**: return** $\perp$
$m^{(j)} \leftarrow m$
$\boxed{\sigma_2^{(j)} \leftarrow \mathsf{DOS.sign}_2(\varphi_{\mathsf{S},1}^{(j)}, (pk_{\mathsf{R}}^{(j)}, C_1^{(j)}, C_2^{(j)}, m^{(j)}))}^{\,0}$
$\boxed{\sigma_2^{(j)} \leftarrow \mathtt{sign2}(j, (pk_{\mathsf{R}}^{(j)}, C_1^{(j)}, C_2^{(j)}, m^{(j)}))}^{\,1}$
$C_3^{(j)} \leftarrow \mathsf{DEM.enc}(\lambda_{\mathsf{D}}, K^{(j)}, (\sigma_2^{(j)}, m^{(j)}))$
$\Sigma_3^{(j)} \leftarrow C_3^{(j)}; \quad \mathcal{Q}_3 \leftarrow \mathcal{Q}_3 \cup \{j\}; \quad \Sigma^{(j)} \leftarrow (\Sigma_1^{(j)}, \Sigma_2^{(j)}, \Sigma_3^{(j)})$
return $\Sigma_3^{(j)}$

FIGURE 10: Games G_0 and G_1 for proving $(3,1)$-dM-sUF-iCMA security of 3DOSC. Codes in boxes with superscript X are evaluated only in game G_X.

Adversary $\mathcal{B}(\lambda_S, pk_S^*;$ `sign1, sign2`$)$:
$\lambda_K \leftarrow \mathsf{DOTK.init}(); \quad \lambda_D \leftarrow \mathsf{DEM.init}(); \quad \lambda \leftarrow (\lambda_K, \lambda_D, \lambda_S)$
$(sk_R^*, pk_R^*, \Sigma^*) \leftarrow \mathcal{A}(\lambda, pk_S^*;$ `sc1, sc2, sc3`$)$
$(\Sigma_1^*, \Sigma_2^*, \Sigma_3^*) \leftarrow \Sigma^*; \quad (C_1^*, \sigma_1^*) \leftarrow \Sigma_1^*; \quad C_2^* \leftarrow \Sigma_2^*; \quad C_3^* \leftarrow \Sigma_3^*$
$K^* \leftarrow \mathsf{DOTK.dec}(\lambda_K, sk_R^*, (pk_S^*, \sigma_1^*), C_1^*, C_2^*)$
$(\sigma_2^*, m^*) \leftarrow \mathsf{DEM.dec}(\lambda_D, K^*, C_3^*)$
return $((pk_R^*, C_1^*, C_2^*, m^*), \sigma_1^*, \sigma_2^*)$
`// sc1, sc2, and sc3 are the same as those in` G_1 `in Fig. 10`

Figure 11: Description of adversary $\mathcal{B}$ attacking $(2,1)$-sUF-CMA security of DOS.

Table 1: Comparison regarding computational cost.

	CMSM [21]	GIESC [12]	Ours: 3DOSC
Phase 1: sc_1 *with* sk_S *and* pk_S			
Operation	—	$\mathrm{DS}_{\mathrm{BB}}$.sign $\mathrm{OTS}_{\mathrm{M}}$.gen	$\mathrm{DOS}_{\mathrm{GWXT}}.\mathrm{sign}_1$ $\mathrm{DOTK}_{\mathrm{BMW}}.\mathrm{enc}_1$
Computational cost	—	$[3, 2; 0]$	$[2, 2; 0]$
Phase 2: sc_2 *with* pk_R			
Operation	—	$\mathrm{TBKEM}_{\mathrm{tBMW1}}.\mathrm{enc}$	$\mathrm{DOTK}_{\mathrm{BMW}}.\mathrm{enc}_2$
Computational cost	—	$[3, 0; 0] + W$	$[1, 0; 0]$
Phase 3: sc_3 *with* m			
Operation	$\mathrm{TBKEM}_{\mathrm{tBMW1}}.\mathrm{enc}$ $\mathrm{DS}_{\mathrm{BB}}$.sign DEM.enc	$\mathrm{OTS}_{\mathrm{M}}$.sign DEM.enc	$\mathrm{DOS}_{\mathrm{GWXT}}.\mathrm{sign}_2$ DEM.enc
Computational cost	$[4, 0; 0] + W$	$[0, 0; 0]$	$[0, 0; 0]$

Corollary 2 (3DOSC is $(3, n)$-sUF-secure). *Suppose DOS is* $(t', q', \epsilon')(2,1)$*-sUF-CMA-secure and DEM is one-to-one. Then, 3DOSC is* $(t, q, \epsilon)(3, n)$*-dM-sUF-iCMA-secure, where* $t = t' - O(q)$*,* $q = q'$*, and* $\epsilon = n\epsilon'$*.*

6.2. Comparison. In Tables 1 and 2, we present a comparison of our 3DOSC construction with previous signcryption constructions in various viewpoints. Here, we consider the following three signcryption schemes (including ours) that satisfy the strongest notions of insider security in the multiuser setting without relying on random oracles:

CMSM: the standard (i.e., not online/offline) signcryption scheme proposed by Chiba et al. [21] that consists of TBKEM, digital signature, and DEM. Here, we instantiate it with $\mathrm{TBKEM}_{\mathrm{tBMW1}}$, TBKEM obtained from the PKE scheme by Boyen et al. [18] and Boneh-Boyen signature scheme $\mathrm{DS}_{\mathrm{BB}}$ [22].

GIESC: the incrementally executable signcryption scheme [12] that consists of TBKEM, digital signature, one-time signature, and DEM. Here, we instantiate it with $\mathrm{TBKEM}_{\mathrm{tBMW1}}$, $\mathrm{DS}_{\mathrm{BB}}$, and the specific DL-based one-time signature scheme $\mathrm{OTS}_{\mathrm{M}}$ [23].

3DOSC: our proposed MDOSC described in Figure 8 without all boxed parts, which consists of DOTK, DOS, and DEM. Here, we instantiate it with $\mathrm{DOTK}_{\mathrm{BMW}}$, a concrete $(1,1)$-IND-tag-CCA-secure DOTK described in Section 3.3, and the concrete DOS $\mathrm{DOS}_{\mathrm{GWXT}}$ proposed by Gao et al [4].

Note that we do not specify any instance of DEM since the selection of DEM does not cause the efficiency difference in the comparison.

Computational costs are measured in the same way as [21], that is, $[a, b; c]$ denotes a exponentiations, b multi-exponentiations, and c pairing computations, and W denotes computation of the Waters hash [24]. Other computational overhead caused by multiplications, hash functions, and symmetric key primitives are ignored.

In terms of storage and ciphertext size, $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the size of a bilinear group element from $\mathbb{G}$ and $\mathbb{G}_T$, respectively. Similarly, $|\mathbb{Z}_p|$ denotes the size of a group element of $\mathbb{Z}_p$, whereas $|H|$ denotes the output length of target-collision resistant hash function. The appended numerical values are instance bit-length of storage and ciphertext when we assume $|\mathbb{G}| = |\mathbb{Z}_p| = |H| = 160$ bits and $|\mathbb{G}_T| = 1024$ bits to achieve 80-bit security. We also assume $|m| = 80$ to capture typical IoT use cases where messages sent by IoT devices are significantly shorter than other cases.

Our construction exploits the multidivisible feature effectively to achieve both the shortest ciphertext size and the least computational cost in the phase 3 (i.e., online phase). With our DOSC, the expensive encryption process $\mathrm{DOTK.enc}_1$ can also be performed in the phase 1 (i.e., *before* being given the recipient's public key), whereas it can be only in the phase 2 (i.e., *after* being given the recipient's public key) with GIESC. All the advantages enable lightweight IoT devices to transmit confidential as well as authenticated messages to the recipient devices in IoT application scenarios as described in Section 1.

Table 2: Comparison regarding storage and ciphertext size.

	CMSM [21]	GIESC [12]	Ours: 3DOSC
Phase 1: sc_1 *with* sk_S *and* pk_S			
Ciphertext size $\lvert\Sigma_1\rvert$	—	—	$3\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert$; 640
Phase 2: sc_2 *with* pk_R			
Storage size $\lvert\varphi_1\rvert$	—	$4\lvert\mathbb{G}\rvert + 5\lvert\mathbb{Z}_p\rvert + 2\lvert H\rvert$; 1760	$2\lvert\mathbb{Z}_p\rvert$; 320
Ciphertext size $\lvert\Sigma_2\rvert$	—	—	—
Phase 3: sc_3 *with* m			
Storage size $\lvert\varphi_2\rvert$	—	$6\lvert\mathbb{G}\rvert + 5\lvert\mathbb{Z}_p\rvert + 2\lvert H\rvert + \lvert\mathbb{G}_T\rvert$; 3104	$\lvert\mathbb{G}_T\rvert + \lvert\mathbb{Z}_p\rvert$; 1184
Ciphertext size $\lvert\Sigma_3\rvert$	$3\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert + \lvert m\rvert$; 720	$6\lvert\mathbb{G}\rvert + 3\lvert\mathbb{Z}_p\rvert + \lvert H\rvert + \lvert m\rvert$; 1680	$2\lvert\mathbb{Z}_p\rvert + \lvert m\rvert$; 400

Game $\mathrm{G}^{X}_{\mathsf{DEM}}(\mathcal{A}, b)$:
$b \leftarrow_{\$} \{0,1\}$; $\lambda \leftarrow \mathsf{DEM.init}()$; $K^* \leftarrow_{\$} \mathsf{DEM}.\mathcal{K}$
return $b^* \leftarrow \mathcal{A}(\lambda;\ \mathtt{DEM\_enc}, \mathtt{DEM\_dec})$

Oracle $\mathtt{DEM\_enc}(m_0, m_1)$:
// $|m_0| = |m_1|$; queried only once
return $C^* \leftarrow \mathsf{DEM.enc}(\lambda, K^*, m_b)$

Oracle $\mathtt{DEM\_dec}(C)$:
// queried at most q times
if $C^* = \varepsilon$: **return** $\perp$
if $C = C^*$: **return** $\perp$ **else: return** $m \leftarrow \mathsf{DEM.dec}(\lambda, K^*, C)$

Figure 12: Attack game for defining IND-OTCCA and IND-P0-C2 security for DEM. Note that $X \in \{\text{IND-OTCCA}, \text{IND-P0-C2}\}$. Boxed parts are evaluated only in IND-OTCCA.

7. Conclusions

We presented a general concept of multidivisible online/offline cryptography (MDO cryptography), which covers the previous works including online/offline cryptographic schemes, divisible online/offline signatures, incrementally executable signcryptions, and multidivisible online/offline encryptions. We then presented the notion of multidivisible online/offline signcryptions (MDOSCs) as novel application of MDO cryptography. We defined several security notions for MDOSCs and show implications and separations between these security notions. We also presented a generic construction of MDOSC that achieves the strongest security notions with regard to confidentiality and unforgeability. MDOSC schemes allow the computationally restricted and/or bandwidth-restricted devices to transmit signed ciphertexts with low computational overhead and/or low-bandwidth network.

Appendix

A. Data Encapsulation Mechanism

A data encapsulation mechanism (DEM) consists of the following algorithms:

$\mathrm{init}(\,) \longrightarrow \lambda$ (the *public parameter generation* algorithm): it outputs public parameters λ to be used by all parties. Public parameters λ contains a description of the key space, $\mathrm{DEM}.\mathscr{K}$, which is usually identical to $\mathrm{KEM}.\mathscr{K}$. We will generally assume that all algorithms take λ as an implicit input, even if it is not explicitly stated

$\mathrm{enc}(\lambda, K, m) \longrightarrow C$ (the *encryption* algorithm): it takes as input the private key $K \in \mathrm{DEM}.\mathscr{K}$, and a message m from the associated message space, and outputs a ciphertext C

$\mathrm{dec}(\lambda, K, C) \longrightarrow m$ (the deterministic *decryption* algorithm): it takes as input the private key $K \in \mathrm{DEM}.\mathscr{K}$, and a ciphertext C, and outputs either a message m or an error symbol $\perp$

We require the correctness for DEM, namely, for any $\lambda \longleftarrow \mathrm{DEM.init}(\,)$, any $K \in \mathrm{DEM}.\mathscr{K}$, and any m, we have $\mathrm{DEM.dec}(\lambda, K, \mathrm{DEM.enc}(\lambda, K, m)) = m$.

A DEM DEM is said to be *one-to-one* if for any K, C, and C', $\mathrm{DEM.dec}(K, C) = \mathrm{DEM.dec}(K, C') \neq \perp$ implies $C = C'$. In other words, for any given K and m, there is at most one ciphertext C such that $\mathrm{DEM.dec}(K, C) = m$. As described in [25], this property is quite natural for a large number of DEMs.

We recall the security notion of indistinguishability against one-time chosen ciphertext attacks, IND-OTCCA, and its stronger variant, IND-P0-C2. The former is defined in [26], whereas the latter is defined in [27]. The decryption oracle in IND-OTCCA game can be accessed only after the adversary obtains the challenge ciphertext, whereas the one in IND-P0-C2 does not have such restriction.

Let $X \in \{\text{IND-OTCCA}, \text{IND-P0-C2}\}$. The *X-advantage of* $\mathscr{A}$ *against* DEM is defined as

$$\mathrm{Adv}^{X}_{\mathrm{DEM}}(\mathscr{A}) = \left|\Pr\left[G^{X}_{\mathrm{DEM}}(\mathscr{A},0)=1\right] - \Pr\left[G^{X}_{\mathrm{DEM}}(\mathscr{A},1)=1\right]\right|, \tag{A.1}$$

where the attack game G^{X}_{DEM} is defined in Figure 12. We say that DEM is (t,q,ϵ)-*X-secure* if $\mathrm{Adv}^{X}_{\mathrm{DEM}}(\mathscr{A}) \le \varepsilon$ holds for any adversary $\mathscr{A}$ that runs in time t and makes at most q queries to the oracle DEM_dec.

B. Proofs of Theorems

B.1. Proof of Theorem 2. Assume we have a t-time adversary $\mathscr{A}$ that takes n receivers' public keys, makes at most nq queries to sc_1, makes at most q queries per receiver to sc_2, makes at most nq queries to sc_3, and makes at most q_d queries to unsc, in order to break $(3,n,q)$-dM-IND-iCCA security of DOSC. Then, we will construct a t'-time adversary $\mathscr{B}$ that takes a single receiver's public key, makes only one query to sc_1, makes only one query to sc_2, makes only one query to sc_3, makes at most q_d queries to unsc, and exploits $\mathscr{A}$ as a black box to attack the $(3,1,1)$-dM-IND-iCCA security of DOSC. For any $\mathscr{A}$, we will show the following relation holds:

$$\mathrm{Adv}^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A}) \le n^2q^2\,\mathrm{Adv}^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B}). \tag{B.2}$$

In order to prove equation (B.2), we use the game sequence G^u_l, where $l \in [0,nq]$ and $u \in [0,1]$, shown in Figure 13. Our proof is via a hybrid argument in a similar fashion with [20]. We can see that the $\mathscr{A}$'s view in game G^0_0 is equivalent to the one in $G^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A})$ with $b=0$, while the one in game G^0_{nq} is equivalent to the one in $G^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A})$ with $b=1$. Thus, we have

$$\begin{aligned}
&\mathrm{Adv}^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A}) \\
&= \left|\Pr\left[G^0_0(\mathscr{A})=1\right] - \Pr\left[G^0_{nq}(\mathscr{A})=1\right]\right| \\
&= \left|\sum_{l\in[nq]}\left(\Pr\left[G^0_{l-1}(\mathscr{A})=1\right] - \Pr\left[G^0_l(\mathscr{A})=1\right]\right)\right|.
\end{aligned} \tag{B.3}$$

Let GD^u_l denote the event that game G^u_l does not set bad to true. Similarly, we let BD^u_l denote the event that game G^u_l sets bad to true. We can see that GD^0_l is independent of $\mathscr{A}$'s output. Hence, for any $l \in [0,nq]$, it follows that

$$\begin{aligned}
\Pr\left[G^0_l(\mathscr{A})=1\right] &= \Pr\left[G^0_l(\mathscr{A})=1 \wedge GD^0_l\right]/\Pr\left[GD^0_l\right] \\
&= nq\Pr\left[G^0_l(\mathscr{A})=1 \wedge GD^0_l\right].
\end{aligned} \tag{B.4}$$

The second equality holds because GD^0_l is equivalent to the event that $(i^{(j')}, \mathrm{ctr}^{(j')}) = (i^*, q^*)$, where the choice of $j' \longleftarrow_{\$} [nq]$ is independent of $\mathscr{A}$'s view.

Game G^1_l is the same as G^0_l except that when bad is set to true, game G^1_l outputs a random bit b^* and halts in the middle of sc_2 invocation. Game G^0_l and G^1_l are identical-until-bad so that for any $l \in [0,nq]$, we have

$$\begin{aligned}
&\Pr\left[G^0_l(\mathscr{A})=1 \wedge GD^0_l\right] \\
&= \Pr\left[G^1_l(\mathscr{A})=1 \wedge GD^1_l\right] \\
&= \Pr\left[G^1_l(\mathscr{A})=1\right] - \Pr\left[G^1_l(\mathscr{A})=1 \wedge BD^1_l\right] \\
&= \Pr\left[G^1_l(\mathscr{A})=1\right] - \frac{1}{2}\Pr\left[BD^1_l\right].
\end{aligned} \tag{B.5}$$

The last equality holds because if BD occurs then the output of game G^1_l is a random bit.

We now show how to construct adversary $\mathscr{B}$. The attack game defining $(3,1,1)$-dM-IND-iCCA of DOSC and the description of adversary $\mathscr{B}$ against it are described in Figure 14. Note that oracle sc_{1^*}, sc_{2^*}, sc_{3^*}, and unsc^* in $G^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B},b)$ are simplified but equivalent versions of sc_1, sc_2, sc_3, and *unsc*, respectively. We can see adversary $\mathscr{B}$ makes one query to sc_{1^*}, one query to sc_{2^*}, one query to sc_{3^*} and at most q_d queries to unsc^*, and its running time $t' = t + O(nq)$. In addition, for any $l \in [nq]$, we can see that the $\mathscr{A}$'s view in game $G^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B})$ with $l' = l$ as well as $b=0$ (or $b=1$) is identical to the one in game $G^1_{l-1}(\mathscr{A})$ (or $G^1_l(\mathscr{A})$). Therefore, for any $l \in [nq]$, it follows that

$$\begin{aligned}
&\left|\Pr\left[G^1_{l-1}(\mathscr{A})=1\right] - \Pr\left[G^1_l(\mathscr{A})=1\right]\right| \\
&\quad= \left|2\Pr\left[G^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B})=1 \,\middle|\, l'=l\right] - 1\right|.
\end{aligned} \tag{B.6}$$

Combining equation (B.3) with the above equation, equation (B.2) holds as follows:

$$\begin{aligned}
&\mathrm{Adv}^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A}) \\
&\le nq\left|\sum_{l\in[nq]} 2\Pr\left[G^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B})=1 \,\middle|\, l'=l\right] - 1\right| \\
&\le n^2q^2\left|2\Pr\left[G^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B})=1\right] - 1\right| \\
&\le n^2q^2\,\mathrm{Adv}^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{B}).
\end{aligned} \tag{B.7}$$

Combining equation (B.2) with the assumptions that DOSC is (t',q_d,ϵ')-$(3,1,1)$-dM-IND-iCCA-secure, we have $\mathrm{Adv}^{(3,n,q)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathscr{A}) \le n^2q^2\epsilon'$, which implies that DOSC is $(t,q_d,n^2q^2\epsilon)$-$(3,1,1)$-dM-IND-iCCA-secure, as desired. This completes the proof of Theorem 2.

B.2. Proof of Theorem 4. We first show that $3\mathrm{DOSC}_{\mathrm{R}}$ is not $(2,1,1)$-dM-IND-iCCA-secure as the following lemma:

Lemma B.1. ($3\mathrm{DOSC}_{\mathrm{R}}$ is not $(2,1,1)$-iCPA-secure). *iere exists an adversary against* $(2,1,1)$*-dM-IND-iCPA security of* $3DOSC_R$ *which has advantage one.*

Proof. We show the description of adversary $\mathscr{A}$ against it in Figure 15.

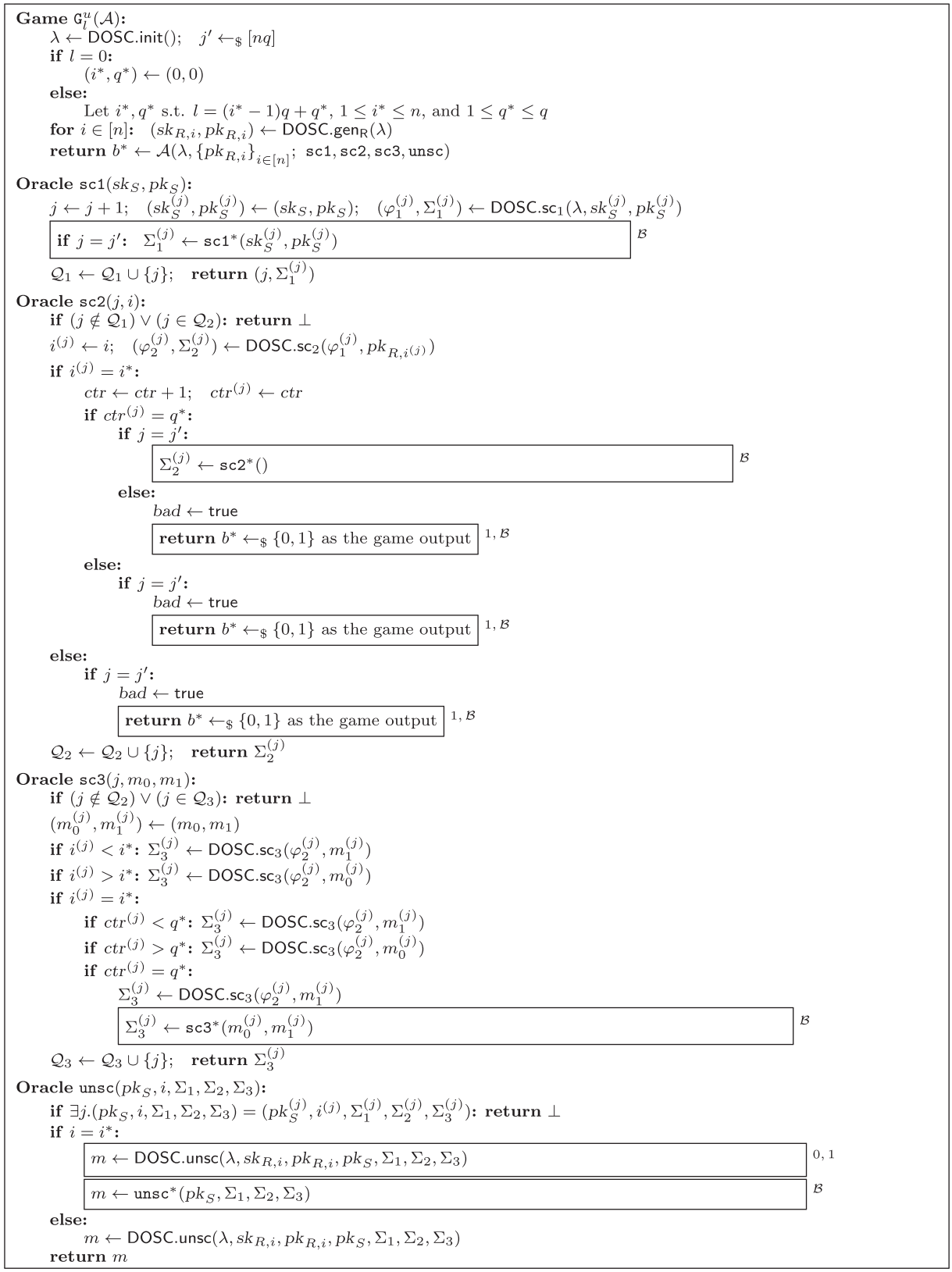

FIGURE 13: Game sequence for proving $(3, n, q)$-dM-IND-iCCA security of DOSC. Codes in boxes with superscript X are evaluated only in game G_l^X or adversary X.

Game $\mathrm{G}^{(3,1,1)\text{-dM-IND-iCCA}}_{\mathrm{DOSC}}(\mathcal{B})$**:**
$b \leftarrow_{\$} \{0,1\}$; $\lambda \leftarrow \mathsf{DOSC.init}()$; $(sk_R^*, pk_R^*) \leftarrow \mathsf{DOSC.gen_R}(\lambda)$
return $b^* \leftarrow \mathcal{B}(\lambda, pk_R^*;\ \mathtt{sc1}^*, \mathtt{sc2}^*, \mathtt{sc3}^*, \mathtt{unsc}^*)$

Oracle $\mathtt{sc1}^*(sk_S^*, pk_S^*)$**:**
// queried only once
$(\varphi_1^*, \Sigma_1^*) \leftarrow \mathsf{DOSC.sc}_1(\lambda, sk_S^*, pk_S^*)$; **return** Σ_1^*

Oracle $\mathtt{sc2}^*()$**:**
// queried only once
$(\varphi_2^*, \Sigma_2^*) \leftarrow \mathsf{DOSC.sc}_2(\varphi_1^*, pk_R^*)$; **return** Σ_2^*

Oracle $\mathtt{sc3}^*(m_0^*, m_1^*)$**:**
// $|m_0^*| = |m_1^*|$; queried only once
return $\Sigma_3^* \leftarrow \mathsf{DOSC.sc}_3(\varphi_2^*, m_b^*)$

Oracle $\mathtt{unsc}^*(pk_S, \Sigma_1, \Sigma_2, \Sigma_3)$**:**
// queried $\leq q_d$ times
if $(pk_S, \Sigma_1, \Sigma_2, \Sigma_3) = (pk_S^*, \Sigma_1^*, \Sigma_2^*, \Sigma_3^*)$**: return** $\perp$
return $m \leftarrow \mathsf{DOSC.unsc}(\lambda, sk_R^*, pk_R^*, pk_S, \Sigma_1, \Sigma_2, \Sigma_3)$

Adversary $\mathcal{B}(\lambda, pk_R^*;\ \mathtt{sc1}^*, \mathtt{sc2}^*, \mathtt{sc3}^*, \mathtt{unsc}^*)$**:**
$j' \leftarrow_{\$} [nq]$; $l' \leftarrow_{\$} [nq]$
Let i^*, q^* s.t. $l' = (i^* - 1)q + q^*$, $1 \leq i^* \leq n$, and $1 \leq q^* \leq q$
$pk_{R,i^*} \leftarrow pk_R^*$
for $i \in [n] \setminus \{i^*\}$**:** $(sk_{R,i}, pk_{R,i}) \leftarrow \mathsf{DOSC.gen_R}(\lambda)$
return $b^* \leftarrow \mathcal{A}(\lambda, \{pk_{R,i}\}_{i\in[n]};\ \mathtt{sc1}, \mathtt{sc2}, \mathtt{sc3}, \mathtt{unsc})$
// sc1, sc2, sc3, and unsc are described in Fig. 13 as $\mathcal{B}$-parts

Figure 14: Attack game for defining $(3,1,1)$-dM-IND-iCCA security of DOSC and description of adversary $\mathcal{B}$ against it.

Adversary $\mathcal{A}(\lambda, pk_{\mathsf{R}}^*;\ \mathtt{sc12}, \mathtt{sc3})$**:**
$(sk_{\mathsf{S}}^*, pk_{\mathsf{S}}^*) \leftarrow \mathsf{DOS.gen}(\lambda_{\mathsf{S}})$
$(1, \Sigma_1^*, \Sigma_2^*) \leftarrow \mathtt{sc12}(sk_{\mathsf{S}}^*, pk_{\mathsf{S}}^*, 1)$
$(C_1^*, \sigma_1^*) \leftarrow \Sigma_1^*$; $C_2^* \leftarrow \Sigma_2^*$
$m_0^* \leftarrow \sigma_1^* \oplus 1^{|\sigma_1^*|}$; $m_1^* \leftarrow \sigma_1^*$
$\Sigma_3^* \leftarrow \mathtt{sc3}(1, m_0^*, m_1^*)$; $(C_3^*, d_{\mathsf{R}}^*) \leftarrow \Sigma_3^*$
return $(d_{\mathsf{R}}^* = 1)$

Figure 15: Description of adversary $\mathcal{A}$ attacking $(2,1,1)$-dM-IND-iCPA-security of $\mathrm{3DOSC_R}$.

In game $G^{(2,1,1)\text{-dM-IND-iCPA}}_{\mathrm{3DOSC_R}}(\mathcal{A})$, when $b = 1$, oracle sc_3 always generates $d_{\mathrm{R}}^* = 1$ as an answer to the query $m_1 = \sigma_1^*$ so that $\mathcal{A}$ outputs 1 with probability one. On the other hand, when $b = 0$, oracle sc_3 always generates $d_{\mathrm{R}}^* = 1$ as an answer to the query $m_0 \neq \sigma_1^*$ so that $\mathcal{A}$ outputs 0 with probability one. Hence, advantage $\mathrm{Adv}^{(2,1,1)\text{-dM-IND-iCPA}}_{\mathrm{3DOSC_R}}(\mathcal{A})$ equals to 1. Note that this attack is CPA because adversary $\mathcal{A}$ does not need the unsigncryption oracle at all.

Next, we show that $\mathrm{3DOSC_R}$ is $(1,1,1)$-dM-IND-iCCA-secure as the following lemma: □

Lemma B.2. ($\mathrm{3DOSC_R}$ is $(1,1,1)$-iCCA-secure). *Suppose DOTK is* $(t', q_d', \epsilon_{DOTK})$*-*$(1,1,1)$*-IND-tag-CCA-secure, DOS is* ϵ_{SMTH}*-smooth, and DEM is* $(t', q_d', \epsilon_{DEM})$*-IND-OTCCA-secure. Then,* $3DOSC_R$ *is* (t, q_d, ϵ)*-*$(1,1,1)$*-dM-IND-iCCA-secure, where* $t = t' - O(q_d)$*,* $q_d = q_d'$ *and* $\epsilon = 2\epsilon_{DOTK} + 4\epsilon_{SMTH} + \epsilon_{DEM}$.

Proof. Assume we have a t-time adversary $\mathcal{A}$ that makes at most q_d queries to unsc in order to break $(1,1,1)$-dM-IND-iCCA security of $\mathrm{3DOSC_R}$. We will construct two adversaries $\mathcal{B}$ and $\mathcal{C}$ that exploit $\mathcal{A}$ as a black box, where $\mathcal{B}$ attacks the $(1,1,1)$-IND-tag-CCA security of DOTK and $\mathcal{C}$ attacks the IND-OTCCA security of DEM. For any $\mathcal{A}$, we will show the following relation holds:

$$\mathrm{Adv}^{(1,1,1)\text{-dM-IND-iCCA}}_{\mathrm{3DOSC_R}}(\mathcal{A}) \leq 2\mathrm{Adv}^{(1,1,1)\text{-IND-tag-CCA}}_{\mathrm{DOTK}}(\mathcal{B}) + 4\mathrm{Smth}_{\mathrm{DOS}} + \mathrm{Adv}^{\mathrm{IND\text{-}OTCCA}}_{\mathrm{DEM}}(\mathcal{C}). \tag{B.8}$$

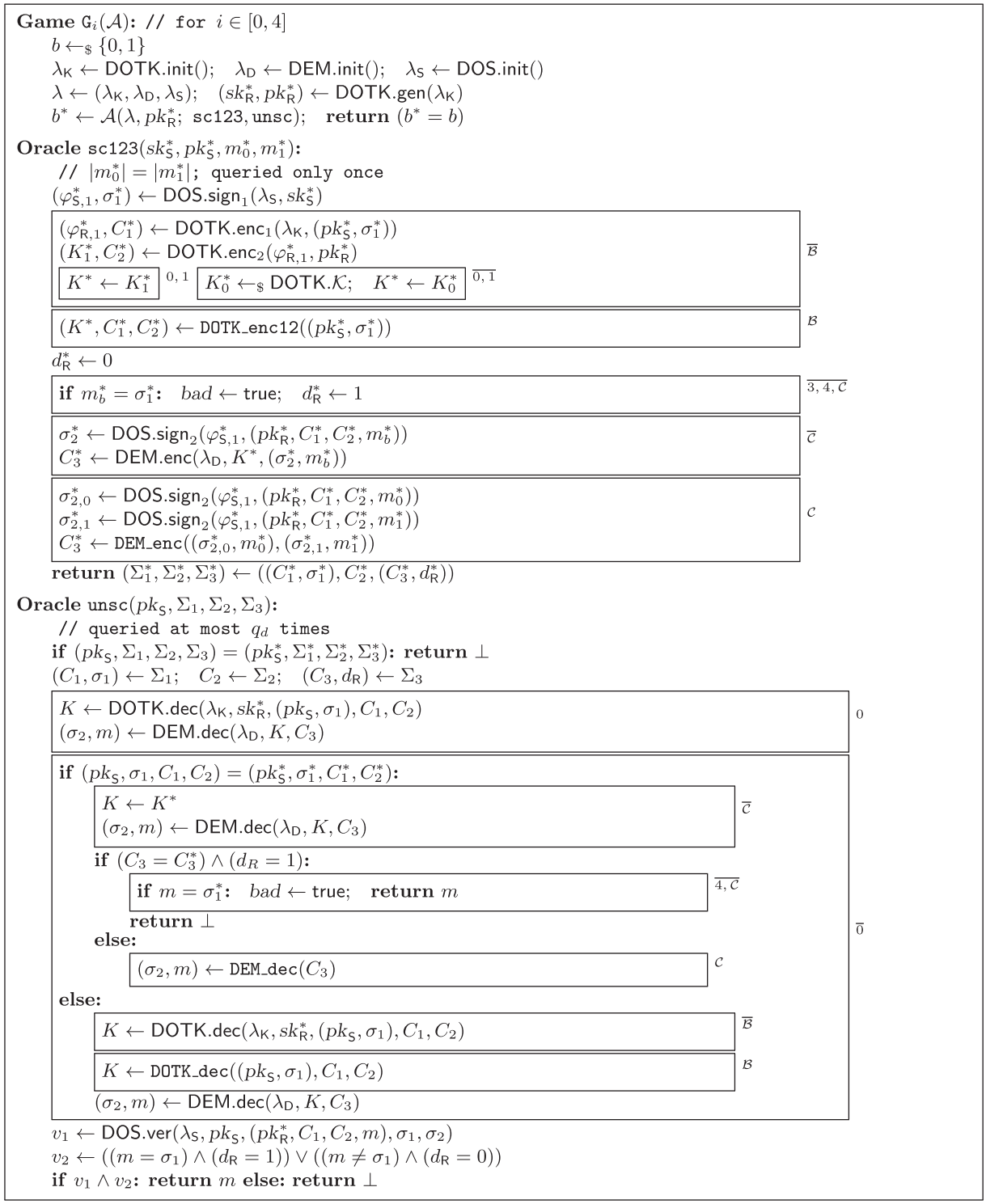

FIGURE 16: Game sequence for proving $(1,1,1)$-dM-IND-iCCA security of $3\mathrm{DOSC}_\mathrm{R}$. Codes in boxes with superscript X are evaluated only in game G_X or adversary X, whereas ones in a box with $\overline{X}$ are *not* evaluated in game G_X nor adversary X.

In order to prove equation (B.8), we describe a game sequence in Figure 16. Note that we simplify the game description by omitting or modifying the statements and the variables that play no role when $n = q = 1$. In particular, we replace the superscript $\cdot^{(j)}$ by $\cdot^*$ since j is fixed in the game when $n = q = 1$. Similarly, the public key $\mathrm{pk}_{\mathrm{R},i}$ is fixed to pk^*_R.

Game G_0 is equivalent to the original attack game played by $\mathscr{A}$ against $3\mathrm{DOSC}_\mathrm{R}$. Hence, we have

Game $\mathrm{G}_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B})$**:**
$\hat{b} \leftarrow_{\$} \{0,1\}$; $\lambda_{\mathsf{K}} \leftarrow \mathsf{DOTK.init}()$; $(sk_{\mathsf{R}}^*, pk_{\mathsf{R}}^*) \leftarrow \mathsf{DOTK.gen}(\lambda_{\mathsf{K}})$
$\hat{b^*} \leftarrow \mathcal{B}(\lambda_{\mathsf{K}}, pk_{\mathsf{R}}^*;\ \mathtt{DOTK\_enc12}, \mathtt{DOTK\_dec})$; **return** $(\hat{b^*} = \hat{b})$

Oracle $\mathtt{DOTK\_enc12}(\tau^*)$**:**
$(\varphi_{\mathsf{R},1}^*, C_1^*) \leftarrow \mathsf{DOTK.enc}_1(\lambda_{\mathsf{K}}, \tau^*)$
$(K_1^*, C_2^*) \leftarrow \mathsf{DOTK.enc}_2(\varphi_{\mathsf{R},1}^*, pk_{\mathsf{R}}^*)$
$K_0^* \leftarrow_{\$} \mathsf{DOTK}.\mathcal{K}$; $K^* \leftarrow K_{\hat{b}}^*$
return (K^*, C_1^*, C_2^*)

Oracle $\mathtt{DOTK\_dec}(\tau, C_1, C_2)$**:**
if $(\tau, C_1, C_2) = (\tau^*, C_1^*, C_2^*)$**: return** $\perp$
return $K \leftarrow \mathsf{DOTK.dec}(\lambda_{\mathsf{K}}, sk_{\mathsf{R}}^*, \tau, C_1, C_2)$

Adversary $\mathcal{B}(\lambda_{\mathsf{K}}, pk_{\mathsf{R}}^*;\ \mathtt{DOTK\_enc12}, \mathtt{DOTK\_dec})$**:**
$b \leftarrow_{\$} \{0,1\}$; $\lambda_{\mathsf{D}} \leftarrow \mathsf{DEM.init}()$; $\lambda_{\mathsf{S}} \leftarrow \mathsf{DOS.init}()$
$\lambda \leftarrow (\lambda_{\mathsf{K}}, \lambda_{\mathsf{D}}, \lambda_{\mathsf{S}})$; $b^* \leftarrow \mathcal{A}(\lambda, pk_{\mathsf{R}}^*;\ \mathtt{sc123}, \mathtt{unsc})$
return $(b^* = b)$
// sc123 and unsc are described in Fig. 16 as $\mathcal{B}$-parts

Figure 17: Attack game defining $(1,1,1)$-IND-tag-CCA security of DOTK and description of adversary $\mathcal{B}$ against it.

$$\Pr\left[G_{\mathrm{3DOSC_R}}^{(1,1,1)\text{-dM-IND-iCCA}}(\mathcal{A}) = 1\right] = \Pr\left[G_0(\mathcal{A}) = 1\right]. \quad \text{(B.9)}$$

Game G_1 behaves just like G_0, except that unsc skips the decryption process of DOTK.dec and directly decrypts (C_1, C_2) using K^* when $(\mathrm{pk_S}, \sigma_1, C_1, C_2) = (\mathrm{pk_S^*}, \sigma_1^*, C_1^*, C_2^*)$ holds. This modification does not change the view of $\mathcal{A}$ because when $(\varphi_{\mathrm{R},1}^*, C_1^*) \longleftarrow \mathrm{DOTK.enc}_1(\lambda_K, (\mathrm{pk_S^*}, \sigma_1^*))$ and $(K^*, C_2^*) \longleftarrow \mathrm{DOTK.enc}_2(\varphi_{\mathrm{R},1}^*, \mathrm{pk_R^*})$, it follows that $\mathrm{DOTK.dec}(\lambda_K, \mathrm{sk_R^*}, (\mathrm{pk_S^*}, \sigma_1^*), C_1^*, C_2^*) = K^*$ from the correctness property of DOTK. Therefore, we have

$$\Pr\left[G_0(\mathcal{A}) = 1\right] = \Pr\left[G_1(\mathcal{A}) = 1\right]. \quad \text{(B.10)}$$

Game G_2 is the same as G_1, except that K^* is replaced by K_0^*, which is uniformly chosen from $\mathrm{DOTK}.\mathcal{K} = \mathrm{DEM}.\mathcal{K}$ rather than generated by $\mathrm{DOTK.enc}_2$. We now construct adversary $\mathcal{B}$ satisfies the following relation:

$$\left|\Pr\left[G_1(\mathcal{A}) = 1\right] \text{-} \Pr\left[G_2(\mathcal{A}) = 1\right]\right| \le \mathrm{Adv}_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B}). \quad \text{(B.11)}$$

We show the attack game defining $(1,1,1)$-IND-tag-CCA of DOTK and the description of adversary $\mathcal{B}$ against it in Figure 17. We can see adversary $\mathcal{B}$ makes at most q_d queries to DOTK_dec, and its running time $t' = t + O(q_d)$. It is easy to verify that $\mathcal{A}$'s view in $G_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B})$ is equivalent to the one in G_1 (or G_2) with $\hat{b} = 1$ (or $\hat{b} = 0$). Hence, we have

$$\begin{aligned}
&\left|\Pr\left[G_1(\mathcal{A}) = 1\right] - \Pr\left[G_2(\mathcal{A}) = 1\right]\right| \\
&= \left|\Pr\left[b^* = b \mid \hat{b} = 1\right] - \left(1 - \Pr\left[b^* \neq b \mid \hat{b} = 0\right]\right)\right| \\
&= \left|2\Pr\left[(b^* = b) = \hat{b}\right] - 1\right| \\
&= \left|2\Pr\left[G_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B}) = 1\right] - 1\right| \\
&= Adv_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B}),
\end{aligned} \quad \text{(B.12)}$$

where the underlying probability spaces in the second and the third equalities are defined as $G_{DOTK}^{(1,1,1)\text{-IND-tag-CCA}}(\mathcal{B})$. Note that adversary $\mathcal{B}$ can replace all executions of DOTK_dec by the oracle invocation of DOTK_dec since $((\mathrm{pk_S^*}, \sigma_1^*), C_1^*, C_2^*)$ is never queried to DOTK_dec in game G_2.

Game G_3 is identical to G_2, except that we omit the conditional assignment of $d_{\mathrm{R}}^* \longleftarrow 1$ if $m_b^* = \sigma_1^*$. Note that adversary $\mathcal{A}$ cannot see σ_1^* before she determines (m_0^*, m_1^*) so that the probability of the event $m_b^* = \sigma_1^*$ only depends on the distribution of σ_1^*. Because of the smoothness of DOS, the following relation holds:

$$\begin{aligned}
\left|\Pr\left[G_2(\mathcal{A}) = 1\right] - \Pr\left[G_3(\mathcal{A}) = 1\right]\right| &\le \Pr\left[m_b^* = \sigma_1^*\right] \\
&\le \mathrm{Smth_{DOS}}.
\end{aligned} \quad \text{(B.13)}$$

Game G_4 is identical to G_3 except for omitting the conditional return statement in unsc. As with the above argument, we have

$$\left|\Pr\left[G_3(\mathcal{A}) = 1\right] - \Pr\left[G_4(\mathcal{A}) = 1\right]\right| \le \mathrm{Smth_{DOS}}. \quad \text{(B.14)}$$

We now construct adversary $\mathcal{C}$ satisfies the following relation:

$$\Pr\left[G_4(\mathcal{A}) = 1\right] = \Pr\left[G_{\mathrm{DEM}}^{\mathrm{IND\text{-}OTCCA}}(\mathcal{C}) = 1\right]. \quad \text{(B.15)}$$

We show the attack game defining IND-OTCCA of DEM and the description of adversary $\mathcal{C}$ against it in Figure 18. We can see adversary $\mathcal{C}$ makes at most q_d queries to DEM_dec, and its running time $t' = t + O(q_d)$. It is not hard to verify that $G_{\mathrm{DEM}}^{\mathrm{IND\text{-}OTCCA}}(\mathcal{C})$ is equivalent to $G_4(\mathcal{A})$, which implies the above equation. Note that if $(\mathrm{pk_S^*}, \sigma_1, C_1) = (\mathrm{pk_S^*}, \sigma_1^*, C_1^*)$ holds in *unsc*, then $C_2 \neq C_2^*$ always holds. Hence, C_2^* is never queried to DEM_dec in game G_3 so that adversary $\mathcal{C}$ can replace all executions of DEM.dec by the invocations of oracle DEM_dec.

In summary, equation (B.8) holds as follows:

Game $\mathsf{G}_{\mathsf{DEM}}^{\mathsf{IND\text{-}OTCCA}}(\mathcal{C})$**:**
$b \leftarrow_{\$} \{0,1\}$; $\lambda_{\mathsf{D}} \leftarrow \mathsf{DEM.init}()$; $K^* \leftarrow_{\$} \mathsf{DEM}.\mathcal{K}$ $b^* \leftarrow \mathcal{C}(\lambda_{\mathsf{D}}; \mathtt{DEM\_enc}, \mathtt{DEM\_dec})$; **return** $(b^* = b)$

Oracle $\mathtt{DEM\_enc}(m_0, m_1)$**:**
return $C_2^* \leftarrow \mathsf{DEM.enc}(\lambda_{\mathsf{D}}, K^*, m_b)$

Oracle $\mathtt{DEM\_dec}(C_2)$**:**
if $C_2^* = \varepsilon$**: return** $\perp$
if $C_2 = C_2^*$**: return** $\perp$ **else: return** $m \leftarrow \mathsf{DEM.dec}(\lambda_{\mathsf{D}}, K^*, C_2)$

Adversary $\mathcal{C}(\lambda_{\mathsf{D}}; \mathtt{DEM\_enc}, \mathtt{DEM\_dec})$**:**
$\lambda_{\mathsf{K}} \leftarrow \mathsf{DOTK.init}()$; $\lambda_{\mathsf{S}} \leftarrow \mathsf{DOS.init}()$; $\lambda \leftarrow (\lambda_{\mathsf{K}}, \lambda_{\mathsf{D}}, \lambda_{\mathsf{S}})$
$(sk_{\mathsf{R}}^*, pk_{\mathsf{R}}^*) \leftarrow \mathsf{DOTK.gen}(\lambda_{\mathsf{K}})$
return $b^* \leftarrow \mathcal{A}(\lambda, pk_{\mathsf{R}}^*; \mathtt{sc123}, \mathtt{unsc})$
`// sc123 and unsc are described in Fig. 16 as` $\mathcal{C}$`-parts`

Figure 18: Attack game defining IND – OTCCA security of DEM and description of the adversary $\mathscr{C}$ against it.

$$\begin{aligned}
&\mathrm{Adv}_{3\mathrm{DOSC_R}}^{(1,1,1)\text{-dM-IND-iCCA}}(\mathscr{A}) \\
&= 2\left|\Pr\left[G_{3\mathrm{DOSC_R}}^{(1,1,1)\text{-dM-IND-iCCA}}(\mathscr{A}) = 1\right] - 1/2\right| \\
&= 2\left|\Pr\left[G_0(\mathscr{A}) = 1\right] - 1/2\right| \\
&\le 2\sum_{0\le i\le 3}\left|\Pr\left[G_i(\mathscr{A}) = 1\right] - \Pr\left[G_{i+1}(\mathscr{A}) = 1\right]\right| + 2\left|\Pr\left[G_3(\mathscr{A}) = 1\right] - 1/2\right| \\
&\le 2\mathrm{Adv}_{\mathrm{DOTK}}^{(1,1,1)\text{-IND-tag-CCA}}(\mathscr{B}) + 4\mathrm{Smth}_{\mathrm{DOS}} + \mathrm{Adv}_{\mathrm{DEM}}^{\mathrm{IND\text{-}OTCCA}}(\mathscr{C}).
\end{aligned} \tag{B.16}$$

Combining equation (B.8) with the assumptions that DOTK is $(t', q_d', \epsilon_{\mathrm{DOTK}})$-$(1,1,1)$-IND-tag-CCA-secure, DOS is ϵ_{SMTH}-smooth, and DEM is $(t', q_d', \epsilon_{\mathrm{DEM}})$-IND-OTCCA-secure, we have $\mathrm{Adv}_{3\mathrm{DOSC_R}}^{(1,1,1)\text{-dM-IND-iCCA}}(\mathscr{A}) \le 2\epsilon_{\mathrm{DOTK}} + 4\epsilon_{\mathrm{SMTH}} + \epsilon_{\mathrm{DEM}}$, which implies that $3\mathrm{DOSC_R}$ is (t, q_d, ϵ)-$(1,1,1)$-dM-IND-iCCA-secure for $\epsilon = 2\epsilon_{\mathrm{DOTK}} + 4\epsilon_{\mathrm{SMTH}} + \epsilon_{\mathrm{DEM}}$, as desired. This completes the proof of Lemma B.2.

Combined with Lemma B.1 and Lemma B.2, the proof of Theorem 4 is completed. □

Data Availability

All the pseudocodes used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] D. Catalano, M. Di Raimondo, D. Fiore, and R. Gennaro, "Off-line/on-line signatures: theoretical aspects and experimental results," in *Public Key Cryptography—PKC 2008 (LNCS)*, R. Cramer, Ed., vol. 4939, pp. 101–120, Springer, Heidelberg, Germany, 2008.

[2] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital schemes," in *Advances in Cryptology—CRYPTO '89 Proceedings (LNCS)*, G. Brassard, Ed., vol. 435, pp. 263–275, Springer, Heidelberg, Germany, 1990.

[3] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," *Journal of Cryptology*, vol. 9, no. 1, pp. 35–67, 1996.

[4] C. Gao, B. Wei, D. Xie, and C. Tang, "Divisible on-line/off-line signatures," in *Topics in Cryptology—CT-RSA 2009 (LNCS)*, M. Fischlin, Ed., vol. 5473, pp. 148–163, Springer, Heidelberg, Germany, 2009.

[5] S. S. M. Chow, J. K. Liu, and J. Zhou, "Identity-based online/offline key encapsulation and encryption," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security—ASIACCS '11*, B. S. N. Cheung, L. C. K. Hui, R. S. Sandhu, and D. S. Wong, Eds., pp. 52–60, ACM Press, Hong Kong, China, March 2011.

[6] F. Guo, Y. Mu, and Z. Chen, "Identity-based online/offline encryption," in *Financial Cryptography and Data Security*, G. Tsudik, Ed., vol. 5143, pp. 247–261, Springer, Heidelberg, Germany, 2008.

[7] J. Lai, Y. Mu, F. Guo, and W. Susilo, "Improved identity-based online/offline encryption," in *Information Security and Privacy*, E. Foo and D. Stebila, Eds., vol. 9144, pp. 160–173, Springer, Heidelberg, Germany, 2015.

[8] J. K. Liu and J. Zhou, "An efficient identity-based online/offline encryption scheme," in *Applied Cryptography and Network Security (LNCS)*, M. Abdalla, D. Pointcheval, P. A. Fouque, and D. Vergnaud, Eds., vol. 5536, pp. 156–167, Springer,Heidelberg, Germany, 2009.

[9] J. K. Liu, J. Baek, and J. Zhou, "Online/offline identity-based signcryption revisited," in *Information Security and Cryptology—Inscrypt 2010*, pp. 36–51, Springer, Heidelberg, Germany, 2011.

[10] D. Sun, Y. Mu, and W. Susilo, "A generic construction of identity-based online/offline signcryption," in *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 707–712, IEEE, Sydney, NSW, Australia, December 2008.

[11] Z. Xu, G. Dai, and D. Yang, "An efficient online/offline signcryption scheme for MANET," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, vol. 2, pp. 171–176, IEEE, Niagara Falls, Canada, March 2007.

[12] D. Yamamoto, H. Sato, and Y. Fukuzawa, "Incrementally executable signcryptions," in *Information Security and Privacy (LNCS)*, W. Susilo and Y. Mu, Eds., vol. 8544, pp. 226–241, Springer, Heidelberg, Germany, 2014.

[13] F. Zhang, Y. Mu, and W. Susilo, "Reducing security overhead for mobile networks," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 1, pp. 398–403, IEEE, Taipei, Taiwan, March 2005.

[14] D. Yamamoto and W. Ogata, "Multi-divisible on-line/off-line encryptions," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100.A, no. 1, pp. 91–102, 2017.

[15] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281–308, Apr 1988.

[16] M. Bellare, D. Hofheinz, and E. Kiltz, "Subtleties in the definition of IND-CCA: when and how should challenge decryption be disallowed?," *Journal of Cryptology*, vol. 28, no. 1, pp. 29–48, 2015.

[17] X. Boyen, Q. Mei, and B. Waters, "Direct chosen ciphertext security from identity-based techniques," Cryptology ePrint Archive, Report 2005/288, New York, NY, USA, 2005, http://eprint.iacr.org/2005/288.

[18] X. Boyen, Q. Mei, and B. Waters, "Direct chosen ciphertext security from identity-based techniques," in *Proceedings of the 12th ACM Conference on Computer and Communications security—CCS'05*, V. Atluri, C. Meadows, and A. Juels, Eds., pp. 320–329, ACM Press, Alexandria, VA, USA, November 2005.

[19] T. Matsuda, K. Matsuura, and J. C. N. Schuldt, "Efficient constructions of signcryption schemes and signcryption composability," in *Progress in Cryptology—INDOCRYPT 2009 (LNCS)*, B. K. Roy and N. Sendrier, Eds., vol. 5922, pp. 321–342, Springer, Heidelberg, Germany, 2009.

[20] M. Bellare, A. Boldyreva, and S. Micali, "Public-key encryption in a multi-user setting: security proofs and improvements," in *Advances in Cryptology—EUROCRYPT 2000 (LNCS)*, B. Preneel, Ed., vol. 1807, pp. 259–274, Springer, Heidelberg, Germany, 2000.

[21] D. Chiba, T. Matsuda, J. C. Schuldt, and K. Matsuura, "Efficient generic constructions of signcryption with insider security in the multi-user setting," in *Proceedings of the Applied Cryptography and Network Security—ACNS 2011*, pp. 220–237, Springer, Nerja, Spain, June 2011.

[22] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," in *Advances in Cryptology—CRYPTO 2004 (LNCS)*, M. Franklin, Ed., vol. 3152, pp. 443–459, Springer, Heidelberg, Germany, 2004.

[23] P. Mohassel, "One-time signatures and chameleon hash functions," in *Selected Areas in Cryptography—SAC 2010 (LNCS)*, A. Biryukov, G. Gong, and D. R. Stinson, Eds., vol. 6544, pp. 302–319, Springer, Heidelberg, Germany, 2011.

[24] B. R. Waters, "Efficient identity-based encryption without random oracles," in *Lecture Notes in Computer Science (LNCS)*, R. Cramer, Ed., vol. 3494, pp. 114–127, Springer, Heidelberg, Germany, 2005.

[25] D. Chiba, T. Matsuda, J. C. N. Schuldt, and K. Matsuura, "Efficient generic constructions of signcryption with insider security in the multi-user setting," in *Applied Cryptography and Network Security (LNCS)*, J. Lopez and G. Tsudik, Eds., vol. 6715, pp. 220–237, Springer, Heidelberg, Germany, 2011.

[26] J. Herranz, D. Hofheinz, and E. Kiltz, "Some (in)sufficient conditions for secure hybrid encryption," *Information and Computation*, vol. 208, no. 11, pp. 1243–1257, 2010.

[27] J. Katz and M. Yung, "Characterization of security notions for probabilistic private-key encryption," *Journal of Cryptology*, vol. 19, no. 1, pp. 67–95, 2006.