Multi-recipient Encryption, Revisited*

Alexandre Pinto
Royal Holloway,
University of London
and
Instituto Universitário da Maia

Bertram Poettering
Royal Holloway,
University of London

Jacob C.N. Schuldt
Royal Holloway,
University of London

ABSTRACT

A variant of public key encryption that promises efficiency gains due to batch processing is multi-recipient public key encryption (MR-PKE). Precisely, in MR-PKE, a dedicated encryption routine takes a vector of messages and a vector of public keys and outputs a vector of ciphertexts, where the latter can be decrypted individually, as in regular PKE.

In this paper we revisit the established security notions of MR-PKE and the related primitive MR-KEM. We identify a subtle flaw in a security model by Bellare, Boldyreva, and Staddon, that also appears in later publications by different authors. We further observe that these security models rely on the knowledge-of-secret-key (KOSK) assumption—a requirement that is rarely met in practice. We resolve this situation by proposing strengthened security notions for MR-PKE and MR-KEMs, together with correspondingly secure yet highly efficient schemes. Importantly, our models abstain from restricting the set of considered adversaries in the way prior models did, and in particular do not require the KOSK setting. We prove our constructions secure assuming hardness of the static Diffie-Hellman problem, in the random oracle model.

1. INTRODUCTION

Standard public key encryption is concerned with two communicating parties, a sender and a receiver, and enables the sender to encrypt a confidential message for the receiver using the latter's public key. Should a sender want to encrypt the same or different messages for two or more receivers, the use of a standard public key encryption scheme would require the sender to encrypt the messages under the intended receivers' public keys independently of each other.

In contrast to this, a *multi-recipient multi-message public key encryption* scheme (MR-MM-PKE) enables a sender to simultaneously encrypt many messages for many receivers in a single operation. More specifically, an MR-MM-PKE scheme provides an encryption algorithm which, given a vector $(pk_1, \ldots, pk_n)$ consisting of the public keys of n receivers and a corresponding vector $(m_1, \ldots, m_n)$ of n messages, returns a vector $(c_1, \ldots, c_n)$ of n ciphertexts. It is required that these ciphertexts can be decrypted individually, i.e., each receiver i (holding the public/private key pair (pk_i, sk_i)) be able to decrypt c_i without knowledge of the public keys of the other recipients or the remaining ciphertexts c_j, $j \neq i$. Hence, syntactically, decryption is identical to that of an ordinary public key encryption scheme.

By simultaneously processing messages for multiple receivers, MR-MM-PKE schemes allow computational and, in some settings, bandwidth savings compared to the parallel use of standard public key encryption. This property is highly attractive in applications where batch processing of encryptions naturally occurs. For example, consider an online banking system which, following a regular schedule, sends encrypted bank account statements to all customers who have provided the bank with a corresponding public key. By employing an MR-MM-PKE scheme as opposed to an ordinary public key encryption scheme, the bank could considerably reduce the computational workload for providing this service.

1.1 The evolution of MR-MM-PKE

A promising approach towards the construction of efficient MR-MM-PKE schemes is to use an ordinary encryption scheme to encrypt the messages for their respective receivers, but to use the same randomness for all encryptions [13, 5, 4]. Depending on the structure of the encryption scheme, this can potentially provide a significant reduction in the computational requirements for the encryption process, but at the same time requires that the used encryption scheme remain secure under randomness re-use.

In [13], Kurosawa shows that both ElGamal encryption [11] and (a version of) Cramer-Shoup encryption [8] are secure in this setting. Furthermore, it is highlighted that the MR-MM-PKE schemes obtained by employing randomness reuse in combination with these encryption schemes provide a reduction of almost 50% in terms of computational cost, compared to individually encrypting the messages. However, the MR-MM-PKE security model used in [13] does not take into account malicious receivers: all public keys used in the (multi-recipient) encryption producing the challenge ciphertext need to be honestly generated, and the corresponding private keys need to remain unknown to the adversary. Hence, the security guarantees provided *in practice* are rather weak.

*The research of the second and third authors was supported by an EPSRC Leadership Fellowship, EP/H005455/1.

ASIA CCS'14, June 4–6, 2014, Kyoto, Japan.

http://dx.doi.org/10.1145/2590296.2590329.

This issue was identified and addressed by Bellare, Boldyreva, and Staddon [5] who introduced a strengthened variant of Kurosawa's security definition. The updated model explicitly considers insiders, i.e., malicious receivers, by allowing the adversary to provide some of the public keys used in the encryption generating the challenge ciphertext. However, for these keys the adversary also has to reveal the corresponding secret keys—a requirement generally known as the *knowledge-of-secret-key (KOSK)* assumption.

Commonly seen justifications for this quite strong assumption argue that in all practical systems, in order to be considered valid, public keys are registered with a Certificate Authority (CA), and that this CA will require zero-knowledge proofs of knowledge of the corresponding private keys. This, however, is rarely done in practice. In fact, we are not aware of any CA on the Internet which requires a user to provide a full proof of knowledge of her private key. It is important to note here that KOSK-based security arguments generally do not translate to more realistic settings. Indeed, as we discuss in Section 3.2, multi-recipient ElGamal encryption in the plain (i.e., non-KOSK) setting is trivially breakable, in strong contrast with the (KOSK-based) results from [5].

Besides the strengthened security model, Bellare *et al.* also introduce the notion of *reproducibility* for an encryption scheme and obtain the powerful result that all schemes fulfilling this additional requirement are amenable to a generic conversion to an MR-MM-PKE via randomness re-use. In particular, it is shown that ElGamal and Cramer-Shoup encryption are reproducible, and hence give rise to secure MR-MM-PKE in a generic way. Note that this re-establishes the results by Kurosawa, but in a stronger security model.

1.2 Our Contributions

We first revisit the (already strengthened) security notion for MR-MM-PKE as defined by Bellare *et al.* [5] and observe that it contains a subtle technical flaw. Specifically, we note that when the challenge ciphertext is produced, the used public keys are always prefixed by honest ones. This feature makes it possible to define an (admittedly artificial) MR-MM-PKE scheme which can be shown secure in the model of [5], but which is obviously insecure in any practical setting where the adversary can influence the order in which the encryption algorithm receives its public keys. Furthermore, we note that the security model of [5] does not allow repetition of (honest) public keys. Hence, security is not guaranteed should the sender encrypt more than one message for a given receiver. See Section 3.2 for a more detailed discussion of these issues.

We fix these issues by further strengthening the model of [5]. In the same step we also drop the KOSK assumption, i.e., the adversary is from now on allowed to introduce maliciously generated public keys for which he does not know the decryption keys. We argue that, to ensure security in practice, MR-MM-PKE schemes should achieve the level of security captured by this new model.

As discussed above, the schemes proposed in [5] might become insecure in the new setting, and some definitively do. Unfortunately, this seems to generally hold for all schemes obtained via the reproducibility-based transformation, as the latter inherently requires the KOSK assumption. Hence, in order to find stronger instantiations of MR-MM-PKE, we need to follow a different approach.

As a first step towards an MR-MM-PKE scheme that is secure in our new security model, we introduce the notion of *multi-recipient multi-key key encapsulation mechanism* (MR-MK-KEM). Following the well-known hybrid approach, by combining this KEM with an appropriate *data encapsulation mechanism* (DEM) we obtain an MR-MM-PKE scheme.

We then show that the 'hashed ElGamal' technique from [1] combined with randomness re-use gives rise to a secure MR-MK-KEM and, by consequence, to a secure MR-MM-PKE scheme. The latter is efficient and provides compact ciphertexts: when compared to the parallel use of the original hashed ElGamal scheme, the reduction in the ciphertext size and in the computational cost associated with encryption is approximately 50%. In addition, although providing much stronger security guarantees and not limiting the sender to encrypt the same message for all receivers, our scheme is as efficient as the best known MR-PKE scheme for *single* messages (see related work).

1.3 Related Work

Research on multi-recipient encryption was initiated by Kurosawa [13] who defined the first security model for MR-MM-PKE schemes and proposed randomness re-using constructions based on ElGamal and Cramer-Shoup encryption. The primary goal of [13] was to shorten ciphertexts, and computational advantages of the new primitive were considered only of subordinate importance. As pointed out above, Bellare *et al.* [5] improved upon this work by introducing a stronger security model and describing a general transformation from reproducible encryption schemes to MR-MM-PKE schemes.

The concept of multi-recipient key encapsulation (MR-KEM) was introduced by Smart in [15]. Intuitively, this primitive is closely related to MR-PKE due to the hybrid approach for obtaining public key encryption by generically combining key encapsulation (KEM) with data encapsulation (DEM). It is important to note, however, that Smart only considers single-key MR-KEMs, i.e., MR-KEMs that establish for all recipients the same session key. In combination with a DEM, such KEMs are helpful to construct *multi-recipient single-message public key encryption (MR-SM-PKE)* schemes which are restricted to applications where the same message is encrypted for all receivers. This is an often considered subclass of the more general MR-MM-PKE primitive. Note that the work of Smart does not consider insider attacks; indeed, it seems that in the single-key setting such a notion is inapplicable.

The work of Barbosa and Farshim [3] lifts the MR-KEM results by Smart to the identity-based setting, by correspondingly adapting the definitions from [15] and carefully taking into account the arising subtleties. By consequence, the paper restricts attention to the single-key setting and does not consider insider attacks.

In apparently independent work, Baek, Safavi-Naini, and Susilo [2] construct a single-key MR-KEM in the identity-based setting that computes the ciphertexts for an arbitrary number of recipients using a single pairing computation in total. While this is highly attractive from an efficiency-centric point of view, unfortunately the scheme is proven secure only in a 'selective-id' model.

Chatterjee and Sarkar [7] complement the results from [3, 2] by constructing two multi-recipient single-key identity-

based KEMs that do not require the random oracle heuristic. Their constructions require a hierarchical IBE scheme (HIBE) with constant-size ciphertexts as a building block and are hence interesting more from a theoretical perspective.

Broadcast encryption [12, 14, 6] is closely related to MR-SM-PKE. More specifically, a broadcast encryption scheme allows a sender to encrypt a message for any subset of recipients belonging to a pre-defined universe of users. The latter is typically fixed at the time the scheme is initialized, but specific constructions, denoted *dynamic* broadcast encryption [10], allow users to be added as prospective recipients *after* scheme initialization. In contrast, MR-SM-PKE allows addressing any set of users on an ad-hoc basis, regardless of when their public/private keys were generated. Independently of that, broadcast encryption requires a central authority to generate a public key for the universe of users as well as a private decryption key for each individual. These private keys would then have to be delivered to the users through a confidential and authenticated channel. In contrast, MR-SM-PKE allows users to independently generate their own public/private keys and can hence be used in settings where a trusted third party is not available. Lastly, note that the more general MR-MM-PKE is not comparable to broadcast encryption as it supports encrypting different messages to the recipients.

2. PRELIMINARIES

2.1 Notation

We write $[n]$ for $\{1, \ldots, n\}$ and use arrow notation for vectors (e.g., $\vec{a} = (a_1, \ldots, a_n)$ if $n = |\vec{a}|$). If A is a finite set, we write $a \leftarrow_R A$ for sampling a uniformly from A. If A is an algorithm, we write $a \leftarrow_R A^{\mathcal{O}}(x)$ for assigning to a the outcome of an execution of A on input x with uniformly picked random coins and oracle access to $\mathcal{O}$. In experiments and algorithms we write "`Require` X" as a shortcut for "`Return` $\perp$ `unless` X".

2.2 Static Diffie-Hellman Assumption

We recall the cyclic group setting and the static Diffie-Hellman assumption.

Definition 1 (Group generator). *A group generator $\mathcal{G}$ is an efficient algorithm which, on input security parameter 1^λ, returns the description of an efficient prime-order group $\mathbb{G}$. We assume that the group order p and a distinguished generating element $g \in \mathbb{G}$ are part of this description. We will use the notation $(\mathbb{G}, p, g) \leftarrow_R \mathcal{G}(1^\lambda)$ accordingly.*

Definition 2 (SDH assumption). *Let $\mathcal{G}$ be a group generator. The advantage of an algorithm $\mathcal{A}$ in solving the* static Diffie-Hellman problem *with respect to $\mathcal{G}$ is defined as*

$$\mathrm{Adv}^{\mathsf{SDH}}_{\mathcal{G},\mathcal{A}}(\lambda) = \Pr\Big[(\mathbb{G}, p, g) \leftarrow_R \mathcal{G}(1^\lambda);\ u \leftarrow_R \mathbb{Z}_p;\ v \leftarrow_R \mathbb{Z}_p;\ Z \leftarrow_R \mathcal{A}^{\mathcal{O}_u, \mathcal{O}_v}(\mathbb{G}, p, g, g^u, g^v) :\ Z = g^{uv}\Big],$$

where $\mathcal{O}_u(\cdot,\cdot)$ and $\mathcal{O}_v(\cdot,\cdot)$ are decisional oracles which on input $X, Y \in \mathbb{G}$ return 1 if and only if $X^u = Y$, and if and only if $X^v = Y$, respectively. The probability is taken over the random coins used to sample u and v and those consumed by $\mathcal{G}$ and $\mathcal{A}$. The static Diffie-Hellman assumption *holds with respect to $\mathcal{G}$ if for all efficient algorithms $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{SDH}}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible in λ.*

2.3 Data Encapsulation Mechanisms

The concept of hybrid encryption and the related notion of data encapsulation was first introduced in [9]. We recall the definition and one of the possible security definitions.

Definition 3 (Data encapsulation mechanism). *A* data encapsulation mechanism (DEM) *$D = (\mathcal{K}, \mathsf{DEM}, \mathsf{DEM}^{-1})$ consists of a keyspace $\mathcal{K} = \{0,1\}^{l(\lambda)}$, for a polynomial l in the security parameter, and two efficient algorithms as follows:*

- $\mathsf{DEM}(K, m)$. *On input a key $K \in \mathcal{K}$ and a message $m \in \{0,1\}^*$, this algorithm returns a ciphertext c.*
- $\mathsf{DEM}^{-1}(K, c)$. *On input a key $K \in \mathcal{K}$ and a ciphertext c, this algorithm returns either a message m or the error symbol $\perp$.*

A DEM is correct *if for all $K \in \mathcal{K}$ and all $m \in \{0,1\}^*$ we have $\mathsf{DEM}^{-1}(K, \mathsf{DEM}(K, m)) = m$.*

Definition 4 (One-time CCA security). *A DEM $D = (\mathcal{K}, \mathsf{DEM}, \mathsf{DEM}^{-1})$ is said to be* indistinguishable against a one-time chosen ciphertext attack (IND-OT-CCA) *if for all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ interacting in the experiments $\mathrm{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA},b}_{D,\mathcal{A}}$ from Figure 1 the following advantage function is negligible in λ,*

$$\mathrm{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{D,\mathcal{A}}(\lambda) = \Big|\Pr\Big[\mathrm{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA},1}_{D,\mathcal{A}}(1^\lambda) = 1\Big] - \Pr\Big[\mathrm{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA},0}_{D,\mathcal{A}}(1^\lambda) = 1\Big]\Big|\ ,$$

where the probabilities are taken over the random coins used in the experiment (including those consumed by $\mathcal{A}$).

$\underline{\mathrm{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA},b}_{D,\mathcal{A}}(1^\lambda)}$:

(a) $K \leftarrow_R \mathcal{K}$
(b) $(m^0, m^1, st) \leftarrow \mathcal{A}_1(1^\lambda)$
(c) `Require` $|m^0| = |m^1|$
(d) $c^* \leftarrow \mathsf{DEM}(K, m^b)$
(e) $d \leftarrow \mathcal{A}_2^{\mathcal{D}}(c^*, st)$
(f) `Return` d

Oracle $\mathcal{D}(c)$:

(a) `Require` $c \neq c^*$
(b) $m \leftarrow \mathsf{DEM}^{-1}(K, c)$
(c) `Return` m

Figure 1: IND-OT-CCA experiment for DEMs

3. MULTI-RECIPIENT PKE

In this section we expose the most important concepts of multi-recipient encryption. After motivating its purpose and specifying the syntax, we critically review the security models that appear so far in corresponding publications. Indeed, concerning the latter we identify a couple of shortcomings and show how to resolve them. We defer the construction of a multi-recipient encryption scheme that is secure in the new model to Sections 4.2 and 5.

3.1 Syntax of MR-MM-PKE

We recall the notion of multi-recipient encryption from [5]. In contrast to plain public key encryption this primitive allows batch processing, i.e., the execution of many encryption operations in one shot. The work of [5] restricts attention to the multi-message setting where for each specified recipient key an individual message is encrypted. An important property of multi-recipient encryption is that the decryption operation is oblivious of the other keys used for creating a given ciphertext. The main advantage of this primitive is the potential efficiency gain due to cost amortization in the encryption process where some computations might be used for the creation of several ciphertexts.

DEFINITION 5 (MR-MM-PKE). *A* multi-recipient multi-message public key encryption *scheme (MR-MM-PKE)* $E = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *consists of four efficient algorithms as follows:*

- $\mathsf{PGen}(1^\lambda)$. *On input security parameter* 1^λ*, this algorithm outputs public parameters* pp.

  We will assume implicitly that the following algorithms are defined in respect to a single distinguished instance of pp.

- $\mathsf{KGen}(\,)$. *This probabilistic algorithm outputs a key pair* (sk, pk).

- $\mathsf{Enc}(\vec{pk}, \vec{m})$. *On input vectors* $\vec{pk} = (pk_1, \ldots, pk_n)$ *of public keys and* $\vec{m} = (m_1, \ldots, m_n)$ *of messages, this probabilistic algorithm outputs a vector* $\vec{c} = (c_1, \ldots, c_n)$ *of ciphertexts.*

- $\mathsf{Dec}(sk, c)$. *On input a secret key* sk *and a ciphertext* c*, this algorithm outputs either a message or the error symbol* $\perp$.

For fixed parameters pp *and any* $n \in \mathbb{N}$ *let* $(sk_j, pk_j) \leftarrow_R \mathsf{KGen}(\,)$ *and* $m_j \in \{0,1\}^*$ *for all* $j \in [n]$. *The MR-MM-PKE is* correct *if for all encryptions*

$$(c_1, \ldots, c_n) \leftarrow_R \mathsf{Enc}\big((pk_1, \ldots, pk_n), (m_1, \ldots, m_n)\big)$$

we have $\mathsf{Dec}\big(sk_j, c_j\big) = m_j$ *for all* $j \in [n]$.

Note that we obtain regular public key encryption as a special case of MR-MM-PKE by restricting the Enc algorithm to accept only single-element vectors $\vec{pk}$ and $\vec{m}$. Moreover, a canonic (but rather uninteresting) way to construct MR-MM-PKE from regular public key encryption is to create all ciphertexts independently of each other by invoking $c_j \leftarrow_R \mathsf{Enc}\big(pk_j, m_j\big)$ once for each $j \in [n]$.

3.2 Security of MR-MM-PKE

To model the security of MR-MM-PKE schemes, [5] proposes a generalization of the standard indistinguishability notion for public key encryption (in fact, the authors propose two notions: one corresponding to CPA security, the other to CCA security; here we focus on the CCA variant only). Briefly, in their game-based definition, the adversary is required to tell apart an encryption of one set of messages from an encryption of another set. However, after closely studying the formalizations from [5], we came to the conclusion that a couple of technical artifacts severely weaken the security guarantees provided by the model in practice. Before we elaborate on our findings and fix the model accordingly, let us first recall the definitions from [5] in more detail.

3.2.1 The security model by Bellare, Boldyreva, and Staddon [5]

In Figure 2 we reproduce details of the CCA security experiment from [5] (slightly adapting the notation towards our needs). The experiment simulates to the adversary an environment with k honest users by providing her, in line (c), with the corresponding public keys $\vec{pk} = (pk_1, \ldots, pk_k)$ and decryption oracles. Among others, the adversary outputs two vectors of messages, $\vec{m}^0$ and $\vec{m}^1$, where either the one or the other shall be encrypted for the keys in $\vec{pk}$ in challenge ciphertext $\vec{c}$. Also encrypted, but for the adversarially-generated keys in $\vec{pk}^*$, shall be the messages in vector $\vec{m}^*$. Finally, in $\vec{sk}^*$ the adversary is required to reveal the decryption keys corresponding to $\vec{pk}^*$ (see discussion below). The consistency of adversary's output is checked in lines (d)–(f). Challenge ciphertext $\vec{c}$ is created in line (g); observe here that in the recipient list the honest public keys come first, followed by the corrupt ones. The remaining part of the experiment is as expected, with the natural restrictions on the second-phase decryption oracle. According to [5], an MR-MM-PKE scheme E is secure if the return value d of the experiment is computationally independent of parameter b, for all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

3.2.2 Shortcomings in the model from [5]

We next discuss a couple of properties of the experiment from Figure 2 that indicate that schemes proved secure in respect to the model might, in practice, not offer the intuitively expected level of security.

The first problem stems from the fact that in line (g) the vector of encryption keys is always prefixed by honestly chosen ones. To see that this is problematic, fix an arbitrary MR-MM-PKE scheme E that is secure in the model from [5] and, without loss of generality, assume that each ciphertext of E starts with prefix `"0"`. Execute the KGen algorithm of E to create a distinguished key pair $(sk^\times, pk^\times)$. Consider now the scheme E' obtained from E by modifying the encryption algorithm such that, on input $\vec{pk} = (pk_1, \ldots, pk_n)$ and $\vec{m} = (m_1, \ldots, m_n)$, ciphertext vector $\vec{c}$ is computed as usual if $pk_1 \neq pk^\times$, but is set to $\big(\texttt{"1"} \,\|\, m_1, \ldots, \texttt{"1"} \,\|\, m_n\big)$ otherwise. Assume in addition that the decryption routine is adapted such that ciphertexts of the latter form are 'decrypted' correctly. It is not difficult to see that scheme E' is secure in the sense of [5]; indeed, as case $pk_1 = pk^\times$ occurs only with negligible probability in experiment $\mathrm{Expt}^{\mathsf{MR\text{-}PKE\text{-}IND\text{-}BBS}}$, security of E implies security of E'. However, intuitively, scheme E'

$\underline{\text{Expt}^{\text{MR-PKE-IND-BBS},b}_{E,\mathcal{A},k,n}(1^\lambda)}$:

(a) $pp \leftarrow_R \mathsf{PGen}(1^\lambda)$

(b) $(\vec{sk}, \vec{pk}) \xleftarrow{k}_R \mathsf{KGen}()$

(c) $(\vec{m}^0, \vec{m}^1, \vec{m}^*, \vec{pk}^*, \vec{sk}^*, st) \leftarrow_R \mathcal{A}_1^{\mathcal{D}_1}(pp, \vec{pk})$

(d) `Require` $|\vec{m}^0| = |\vec{m}^1| = k \wedge$
$\forall i \in [k] : |m_i^0| = |m_i^1|$

(e) `Require` $|\vec{m}^*| = |\vec{pk}^*| = |\vec{sk}^*| = n - k$

(f) `Require` $\forall j \in [n-k] : pk_j^*$ matches sk_j^*

(g) $\vec{c} \leftarrow_R \mathsf{Enc}(\vec{pk} \,\|\, \vec{pk}^*, \vec{m}^b \,\|\, \vec{m}^*)$

(h) $d \leftarrow_R \mathcal{A}_2^{\mathcal{D}_2}(\vec{c}, st)$

(i) `Return` d

$\underline{\text{Oracle } \mathcal{D}_1(i, c)}$:

(a) `Require` $i \in [k]$

(b) $m \leftarrow \mathsf{Dec}(sk_i, c)$

(c) `Return` m

$\underline{\text{Oracle } \mathcal{D}_2(i, c)}$:

(a) `Require` $i \in [k]$

(b) `Require` $c \neq c_i$

(c) $m \leftarrow \mathsf{Dec}(sk_i, c)$

(d) `Return` m

Figure 2: Security experiment for MR-MM-PKE from [5]

is weak: by claiming $(sk^\times, pk^\times)$ as her own key pair and by arranging $pk^\times$ to appear first in the input to an encryption operation, the confidentiality of all other ciphertext components is trivially broken.

As an independent (and possibly less far-fetched) problem we identify the fact that experiment $\text{Expt}^{\text{MR-PKE-IND-BBS},b}_{E,\mathcal{A},k,n}$ does not allow the adversary to make the same honest public key appear multiple times on the input to the encryption routine. For instance, intuitively, an MR-MM-PKE scheme where the encryption routine consistently transforms inputs of the form $\vec{pk} = (pk, pk)$ and $\vec{m} = (m, m)$ into ciphertext pairs of the form $\vec{c} = (c, c)$ is weak: if messages m, m_0, m_1 are chosen such that $m = m_0 \neq m_1$, encryptions of (m, m_0) and (m, m_1) for (pk, pk) can trivially be distinguished from each other, since in exactly one of the cases the ciphertext vector has the form (c, c). However, in principle such a scheme can be secure in respect to the experiment from Figure 2, simply because the logic of the game will effectively prevent the special $\vec{pk} = (pk, pk)$ situation to occur (observe that if one of the copies of pk is part of $\vec{pk}^*$, then $\mathcal{A}$ will not be able to reveal the corresponding decryption key sk^*).

3.2.3 A criticism of the KOSK assumption

Recall that in line (c) of $\text{Expt}^{\text{MR-PKE-IND-BBS},b}_{E,\mathcal{A},k,n}$ the adversary is expected to reveal the decryption keys $\vec{sk}^*$ corresponding to the potentially maliciously-chosen encryption keys $\vec{pk}^*$. This represents what is commonly known as the *knowledge-of-secret-key* (KOSK) assumption and shall provide some indication of well-formedness of the keys in $\vec{pk}^*$. We observe, however, that it is generally unclear how to verify that the adversary's outputs $\vec{pk}^*$ and $\vec{sk}^*$ are indeed consistent, i.e., how to accurately implement the corresponding test in line (f). Indeed, in a follow-up work by the authors of [5], this situation is formally clarified by restricting the adversary even further than it is in Figure 2: in [4], the adversary does not have to output $\vec{pk}^*$ and $\vec{sk}^*$ any more, but instead the random coins used to create them with KGen.

Generally speaking, security models assuming KOSK are obviously so strong that one might question their practical relevance. Authors typically argue in support of the KOSK assumption by claiming that in any practical setting public keys are certified by trusted authorities (CAs) anyway, and in the certification process the latter could require zero-knowledge proofs of knowledge of secret keys (or random coins). However, we are not aware of any CA on the Internet or elsewhere that would require such a proof (or would at least have corresponding infrastructure available).

The following simple example illustrates that the approach to obtain MR-MM-PKE from ElGamal encryption [11] by re-using the ephemeral randomness does not lead to a secure scheme. We stress that this holds even though the very same scheme was proved secure in [5] (under the KOSK assumption). In detail, in the DL setting, the scheme encrypts messages $m_1, m_2 \in \mathbb{G}$ for public keys $X_1, X_2 \in \mathbb{G}$ by picking a random exponent $r \in \mathbb{Z}_p$ and computing ciphertexts $c_1 = (g^r, X_1^r \cdot m_1)$ and $c_2 = (g^r, X_2^r \cdot m_2)$. Now, if X_1 is an honestly generated key and the adversary claims $X_2 = X_1^t$ as hers, for arbitrary $t \in \mathbb{Z}_p$, then from any (multi-)encryption

$$(c_1, c_2) = \Big((g^r, X_1^r \cdot m_1), (g^r, X_1^{tr} \cdot m_2)\Big)$$

of secret message m_1 and known message m_2 under keys (X_1, X_2), message m_1 can readily be recovered via $m_1 = (X_1^r \cdot m_1)/((X_1^{tr} \cdot m_2)/m_2)^{t^{-1}}$.

This clearly illustrates the danger of adopting MR-MM-PKE schemes shown secure under the KOSK assumption in practical systems where proofs of knowledge of private keys are not required. However, as we show in Sections 4 and 5, simultaneously efficient and secure MR-MM-PKE schemes can be achieved without requiring the KOSK assumption or any third-party certification of public keys.

3.2.4 Our strengthened security model for MR-MM-PKE

We proceed with the exposition of our new security model for multi-recipient encryption that tackles the issues discussed above. We particularly highlight that in the new model the adversary is allowed to specify arbitrary vectors of encryption keys to be challenged on (i.e., it may arrange the public keys in any order, and also repetitions are allowed); additionally, the KOSK assumption is not required any more. It is easy to see that our model is (strictly) stronger than the one from [5] as our experiment encompasses the one from Figure 2 as a special case.

Definition 6. (Indistinguishability of MR-MM-PKE)

An MR-MM-PKE scheme $E = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* indistinguishable (MR-PKE-IND-secure) *if for all* k *and* n *polynomially dependent on the security parameter and all efficient adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *interacting in experiments*

$\text{Expt}^{\text{MR-PKE-IND}}_{E,\mathcal{A},k,n}$ *from Figure 3 the following advantage function is negligible in* λ*, where the probabilities are taken over the random coins of the experiment (including over* $\mathcal{A}$*'s randomness):*

$$\text{Adv}^{\text{MR-PKE-IND}}_{E,\mathcal{A},k,n}(\lambda) = \Big|\Pr\left[\text{Expt}^{\text{MR-PKE-IND},1}_{E,\mathcal{A},k,n}(1^\lambda) = 1\right] - \Pr\left[\text{Expt}^{\text{MR-PKE-IND},0}_{E,\mathcal{A},k,n}(1^\lambda) = 1\right]\Big| \ .$$

$\underline{\text{Expt}^{\text{MR-PKE-IND}}_{E,\mathcal{A},k,n}(1^\lambda)}$:

(a) $pp \leftarrow_R \mathsf{PGen}(1^\lambda)$
(b) $(\vec{sk}, \vec{pk}) \xleftarrow{k}_R \mathsf{KGen}(\,)$
(c) $(\vec{m}^0, \vec{m}^1, \vec{pk}^*, st) \leftarrow_R \mathcal{A}_1^{\mathcal{D}_1}(pp, \vec{pk})$
(d) `Require` $|\vec{m}^0| = |\vec{m}^1| = |\vec{pk}^*| = n$
(e) `Require` $\forall j \in [n] : pk_j^* \in \vec{pk} \Rightarrow |m_j^0| = |m_j^1|$
(f) `Require` $\forall j \in [n] : pk_j^* \notin \vec{pk} \Rightarrow m_j^0 = m_j^1$
(g) $\vec{c} \leftarrow_R \mathsf{Enc}(\vec{pk}^*, \vec{m}^b)$
(h) $d \leftarrow_R \mathcal{A}_2^{\mathcal{D}_2}(\vec{c}, st)$
(i) `Return` d

$\underline{\text{Oracle } \mathcal{D}_1(i, c)}$:

(a) `Require` $i \in [k]$
(b) $m \leftarrow \mathsf{Dec}(sk_i, c)$
(c) `Return` m

$\underline{\text{Oracle } \mathcal{D}_2(i, c)}$:

(a) `Require` $i \in [k]$
(b) `Require` $\nexists j \in [n] : pk_j^* = pk_i \wedge c_j = c$
(c) $m \leftarrow \mathsf{Dec}(sk_i, c)$
(d) `Return` m

Figure 3: Our strengthened security experiment for MR-MM-PKE

In experiment $\text{Expt}^{\text{MR-PKE-IND}}_{E,\mathcal{A},k,n}$ the adversary first receives the public keys of k honest users and has access to corresponding decryption oracles. She then outputs two vectors of messages, $\vec{m}^0$ and $\vec{m}^1$, on which she wants to be challenged. She also outputs a vector $\vec{pk}^*$ of public keys which may contain honest and malicious keys in any configuration. The only (and natural) condition on $\vec{m}^0, \vec{m}^1, \vec{pk}^*$ is that the messages in $\vec{m}^0$ and $\vec{m}^1$ have the same length when targeting honest public keys, and that the messages are equal when targeting malicious keys (see lines (e) and (f), respectively). Note that the rules for the second-phase decryption oracle are also as liberal as they can possibly be.

4. MULTI-RECIPIENT MULTI-KEY KEY ENCAPSULATION

A natural building block for the construction of a multi-recipient encryption scheme seems to be a multi-recipient multi-key key encapsulation mechanism (MR-MK-KEM). In this section we formalize this primitive and specify its security properties. In addition, by showing that MR-MK-KEMs can be combined with appropriate DEMs to obtain secure MR-MM-PKE we provide evidence that our formalizations are indeed helpful and accurate.

4.1 Syntax and security of MR-MK-KEM

In an MR-MK-KEM, the encapsulation algorithm takes a number of public keys and creates vectors of ciphertexts and corresponding (symmetric) keys. Using the decapsulation algorithm, each such key can individually be recovered from the corresponding ciphertext component. Similarly to the MR-MM-PKE case, the functionality of multi-recipient KEMs is readily emulated by running appropriately-many instantiations of a regular KEM in parallel. Correspondingly, regular KEMs are obtained from MR-MK-KEMs by restricting the input of the encapsulation algorithm to a single element. We formalize the primitive as follows:

Definition 7 (MR-MK-KEM). *A* multi-recipient multi-key key encapsulation mechanism *(MR-MK-KEM)* $M = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *consists of four algorithms as follows:*

- $\mathsf{PGen}(1^\lambda)$. *On input security parameter* 1^λ*, this algorithm outputs public parameters* pp *and the description of a keyspace* $\mathcal{K}$*.*

  We will assume implicitly that the following algorithms are defined in respect to a single distinguished copy of $(pp, \mathcal{K})$*.*

- $\mathsf{KGen}(\,)$. *This probabilistic algorithm outputs a key pair* (sk, pk)*.*

- $\mathsf{Encap}(\vec{pk})$. *On input a vector* $\vec{pk} = (pk_1, \ldots, pk_n)$ *of public keys, this probabilistic algorithm outputs vectors* $\vec{c} = (c_1, \ldots, c_n)$ *of ciphertexts and* $\vec{K} = (K_1, \ldots, K_n)$ *of keys from keyspace* $\mathcal{K}$*.*

- $\mathsf{Decap}(sk, c)$. *On input a secret key* sk *and a ciphertext* c*, this algorithm outputs either a key in* $\mathcal{K}$ *or* $\perp$*.*

For fixed parameters pp *and any* $n \in \mathbb{N}$ *let* $(sk_j, pk_j) \leftarrow_R \mathsf{KGen}(\,)$ *for all* $j \in [n]$*. The MR-MK-KEM is* correct *if for all encapsulations*

$$\left((c_1, \ldots, c_n), (K_1, \ldots, K_n)\right) \leftarrow_R \mathsf{Encap}(pk_1, \ldots, pk_n)$$

we have $\mathsf{Decap}(sk_j, c_j) = K_j$ *for all* $j \in [n]$*.*

We proceed with our security definition for MR-MK-KEMs. Similarly to Section 3 we assume particularly strong adversaries: the availability of decapsulation oracles ensures CCA security, arbitrary configurations are allowed for the challenge public key vector, and, importantly, we do not require the premises of the KOSK assumption.

Definition 8. (Indistinguishability of MR-MK-KEM) *An MR-MK-KEM scheme* $M = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *is* indistinguishable (MR-KEM-IND-secure) *if for all* k *and* n *polynomially dependent on the security parameter and all efficient adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *interacting in experiments* $\text{Expt}^{\text{MR-KEM-IND},b}_{M,\mathcal{A},k,n}$ *from Figure 4 the following advantage function is negligible in* λ*, where the probabilities are taken over the random coins of the experiment (including over* $\mathcal{A}$*'s randomness):*

$$\mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{M,\mathcal{A},k,n}(\lambda) = \left| \Pr\left[\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},1}_{M,\mathcal{A},k,n}(1^\lambda) = 1\right] - \Pr\left[\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},0}_{M,\mathcal{A},k,n}(1^\lambda) = 1\right] \right| .$$

$\underline{\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}_{M,\mathcal{A},k,n}(1^\lambda)}$:

(a) $(pp, \mathcal{K}) \leftarrow_R \mathsf{PGen}(1^\lambda)$
(b) $(\vec{sk}, \vec{pk}) \overset{k}{\leftarrow}_R \mathsf{KGen}(\,)$
(c) $(\vec{pk}^*, st) \leftarrow_R \mathcal{A}_1^{\mathcal{D}_1}(pp, \mathcal{K}, \vec{pk})$
(d) `Require` $|\vec{pk}^*| = n$
(e) $(\vec{c}, \vec{K}^1) \leftarrow_R \mathsf{Encap}(\vec{pk}^*)$
(f) $\forall j \in [n]$:
  - `if` $pk_j^* \in \vec{pk}$ `then` $K_j^0 \leftarrow_R \mathcal{K}$
  - `if` $pk_j^* \notin \vec{pk}$ `then` $K_j^0 \leftarrow K_j^1$
(g) $d \leftarrow_R \mathcal{A}_2^{\mathcal{D}_2}(\vec{c}, \vec{K}^b, st)$
(h) `Return` d

$\underline{\text{Oracle } \mathcal{D}_1\,(i, c)}$:

(a) `Require` $i \in [k]$
(b) $K \leftarrow \mathsf{Decap}\,(sk_i, c)$
(c) `Return` K

$\underline{\text{Oracle } \mathcal{D}_2\,(i, c)}$:

(a) `Require` $i \in [k]$
(b) `Require` $\nexists j \in [n] : pk_j^* = pk_i \wedge c_j = c$
(c) $K \leftarrow \mathsf{Decap}\,(sk_i, c)$
(d) `Return` K

Figure 4: Our security experiment for MR-MK-KEMs

4.2 Constructing MR-MM-PKE from MR-MK-KEMs

We formally verify that the composition of an MR-MK-KEM and a DEM yields a secure MR-MM-PKE scheme.

CONSTRUCTION 1. (MR-MM-PKE FROM MR-MK-KEM) *Let* $M = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *be an MR-MK-KEM and* $D = (\mathcal{K}, \mathsf{DEM}, \mathsf{DEM}^{-1})$ *be a DEM such that the KEM's and DEM's keyspaces coincide. We build an MR-MM-PKE* $E = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *by implementing its algorithms in the following way:*

- $E.\mathsf{PGen}(1^\lambda) = M.\mathsf{PGen}(1^\lambda)$
- $E.\mathsf{KGen}(\,) = M.\mathsf{KGen}(\,)$
- $E.\mathsf{Enc}(\vec{pk}, \vec{m})$. *Run* $(\vec{c^1}, \vec{K}) \leftarrow_R M.\mathsf{Encap}(\vec{pk})$ *and* $\forall j \in [n] : c_j^2 \leftarrow_R D.\mathsf{DEM}\left(K_j, m_j\right)$. *Return* $((c_j^1, c_j^2))_{j\in[n]}$.
- $E.\mathsf{Dec}\,(sk, c)$. *Parse* $(c^1, c^2) \leftarrow c$ *and compute* $K \leftarrow_R M.\mathsf{Decap}(sk, c^1)$. *Return* $\perp$ *if* $K = \perp$*; otherwise, return* $D.\mathsf{DEM}^{-1}(K, c^2)$.

THEOREM 1. *If* M *is an* MR-KEM-IND*-secure MR-MK-KEM and* D *is an* IND-OT-CCA*-secure DEM, then* E *defined according to Construction 1 is an* MR-PKE-IND*-secure MR-MM-PKE. Specifically, given* $k, n \in \mathbb{N}$ *and any efficient adversary* $\mathcal{A}$ *against* E*, we can build efficient adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *against* M *and* D*, respectively, such that*

$$\mathrm{Adv}^{\mathsf{MR\text{-}PKE\text{-}IND}}_{E,\mathcal{A},k,n}(\lambda) \leq 2 \cdot \mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{M,\mathcal{B},k,n}(\lambda) + n \cdot \mathrm{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{D,\mathcal{C}}(\lambda) .$$

We leave the proof for the appendix.

REMARK 1. *While Theorem 1 establishes that the security notions from Definitions 4 and 8 are sufficiently strong to obtain* MR-PKE-IND*-secure MR-MM-PKE schemes, conceivably also other combinations of KEM/DEM security notions will imply a secure hybrid. Indeed, if the requirements in Definition 8 are relaxed such that (some of) the keys* K *established for different occurrences of the same public key in an* Encap *invocation may coincide, security of the hybrid scheme is still provided if the requirements on the DEM are simultaneously strengthened from* IND-OT-CCA *to* IND-CCA *[9] (observe how, in this setting, the stronger DEM thwarts the second attack described in Section 3.2.2).*

5. CONSTRUCTING MULTI-RECIPIENT MULTI-KEY KEY ENCAPSULATION

Taking into account the results from Section 4.2, the missing building block on our way towards MR-MM-PKE is an MR-MK-KEM. We propose a construction that is indistinguishable in the sense of Definition 8 and in particular does not rely on the KOSK assumption.

CONSTRUCTION 2 (HASHED ELGAMAL KEM). *Let* $\mathcal{G}$ *be a group generator as in Definition 1 and let* l *be a polynomial. The algorithms of our MR-MK-KEM* HEK *are specified as follows:*

- $\mathsf{PGen}(1^\lambda)$. *Let* $(\mathbb{G}, p, g) \leftarrow_R \mathcal{G}(1^\lambda)$. *Fix keyspace* $\mathcal{K} = \{0,1\}^{l(\lambda)}$ *and choose a hash function* $H : \mathbb{G} \times \mathbb{G} \times \mathbb{N} \to \mathcal{K}$. *Return public parameters* $pp = (\mathbb{G}, p, g, H)$.
- $\mathsf{KGen}(\,)$. *Sample* $x \leftarrow_R \mathbb{Z}_p$ *and return* $(sk, pk) = (x, g^x)$.
- $\mathsf{Encap}(\vec{pk})$. *Let* $\vec{pk} = (pk_1, \ldots, pk_n)$. *Sample* $r \leftarrow_R \mathbb{Z}_p$ *and compute* $\hat{c} \leftarrow g^r$. *For all* $j \in [n]$ *let* $c_j \leftarrow (\hat{c}, j)$ *and* $K_j \leftarrow H\left((pk_j)^r, pk_j, j\right)$. *Return* $(\vec{c}, \vec{K})$.
- $\mathsf{Decap}\left(sk, (\hat{c}, j)\right)$. *Return* $K = H(\hat{c}^{sk}, pk, j)$, *where* $pk = g^{sk}$.

THEOREM 2. *Our MR-MK-KEM* HEK *from Construction 2 is* MR-KEM-IND*-secure under the static Diffie-Hellman assumption, in the random oracle model. Specifically, given* $k, n \in \mathbb{N}$ *and any efficient adversary* $\mathcal{A}$ *against* HEK*, we can build an efficient adversary* $\mathcal{B}$ *solving the SDH problem in* $\mathbb{G}$ *such that*

$$\mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{\mathsf{HEK},\mathcal{A},k,n}(\lambda) \leq \mathrm{Adv}^{\mathsf{SDH}}_{\mathcal{G},\mathcal{B}}(\lambda) + \frac{q_{d1}}{p} + \frac{q_{h1}}{p} ,$$

where q_{d1} *and* q_{h1} *represent the number of queries* $\mathcal{A}_1$ *issues to the decapsulation and the hash oracles, respectively.*

$\mathcal{B}^{\mathcal{O}_u,\mathcal{O}_v}(\mathbb{G},p,g,U,V)$:

(a) $HL \leftarrow [\,]$, $DL \leftarrow [\,]$
(b) $\mathsf{Bad}_1 \leftarrow 0$, $\mathsf{Bad}_2 \leftarrow 0$
(c) Fix keyspace $\mathcal{K} = \{0,1\}^{l(\lambda)}$
(d) $pp \leftarrow (\mathbb{G},p,g)$
(e) $\forall i \in [k]$: $w_i \leftarrow_R \mathbb{Z}_p$
(f) $\forall i \in [k]$: $pk_i \leftarrow V^{w_i}$
(g) $(\vec{pk}^*, st) \leftarrow_R \mathcal{A}_1^{\mathcal{D}_1,\mathcal{H}}(pp,\mathcal{K},\vec{pk})$
(h) `Require` $|\vec{pk}^*| = n$
(i) $\forall j \in [n]$
- $c_j \leftarrow (U,j)$
- $K_j^* \leftarrow_R \mathcal{K}$
- `if` $\exists i \in [k] : pk_j^* = pk_i$: $DL[i,U,j] \leftarrow K_j^*$

(j) $d \leftarrow_R \mathcal{A}_2^{\mathcal{D}_2,\mathcal{H}}(\vec{c},\vec{K}^*,st)$
(k) `Halt with output` $\perp$

Simulation of $\mathcal{D}_{phase}(i,\hat{c},j)$ oracle:

(a) `Require` $i \in [k]$
(b) `if` $phase = 1 \wedge \hat{c} = U$: $\mathsf{Bad}_1 \leftarrow 1$
(c) `if` $phase = 2$: `Require` $pk_j^* \neq pk_i \vee \hat{c} \neq U$
(d) `if` $\exists Z \in \mathbb{G}, t \in \mathcal{K}$: $HL[Z,pk_i,j] = t \wedge \mathcal{O}_v(\hat{c}^{w_i},Z) = 1$: `Return` t
(e) `if` $DL[i,\hat{c},j] \neq \varepsilon$: `Return` $DL[i,\hat{c},j]$
(f) $t \leftarrow_R \mathcal{K}$
(g) $DL[i,\hat{c},j] \leftarrow t$
(h) `Return` t

Simulation of $\mathcal{H}(Z,W,j)$ oracle:

(a) `if` $\exists i \in [k] : W = pk_i \wedge \mathcal{O}_v(U^{w_i},Z) = 1$: `Halt with output` Z^{1/w_i}
(b) `if` $phase = 1 \wedge \mathcal{O}_u(W,Z) = 1$ $\mathsf{Bad}_2 \leftarrow 1$
(c) `if` $phase = 2 \wedge j \in [n] \wedge W = pk_j^* \wedge \mathcal{O}_u(W,Z) = 1$: `Return` K_j^*
(d) `if` $\exists i \in [k], \hat{c} \in \mathbb{G}, t \in \mathcal{K}$: $W = pk_i \wedge DL[i,\hat{c},j] = t \wedge \mathcal{O}_v(\hat{c}^{w_i},Z) = 1$: `Return` t
(e) `if` $HL[Z,W,j] \neq \varepsilon$: `Return` $HL[Z,W,j]$
(f) $t \leftarrow_R \mathcal{K}$
(g) $HL[Z,W,j] \leftarrow t$
(h) `Return` t

Figure 5: The code for adversary $\mathcal{B}$, trying to break the SDH assumption

Proof. Throughout the proof we will denote by DH the function $\mathbb{G} \times \mathbb{G} \to \mathbb{G}$ that maps pairs (g^x, g^y) to g^{xy} for all $x, y \in \mathbb{Z}_p$. The proof is in the random oracle model, i.e., hash function H is implemented as a random function.

Given fixed k, n and an MR-KEM-IND adversary $\mathcal{A}$ for these parameters, we build an SDH adversary $\mathcal{B}$ that receives a challenge $(\mathbb{G}, p, g, U = g^u, V = g^v)$ and has access to oracles $\mathcal{O}_u(\cdot,\cdot)$ and $\mathcal{O}_v(\cdot,\cdot)$, as defined in Definition 2, and aims at computing $\mathsf{DH}(U,V)$, by the help of $\mathcal{A}$. The description of $\mathcal{B}$ is given in Figure 5.

We argue that the environment that adversary $\mathcal{B}$ provides to $\mathcal{A}$ is (almost) indistinguishable from experiments $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}$, $b \in \{0,1\}$. Indeed, it is readily verified that the vector $\vec{pk}$ of public keys that $\mathcal{A}_1$ receives follows the right (uniform) distribution (cf. lines (e) and (f) of $\mathcal{B}$). In addition, the vector $\vec{K}^*$ of keys given in line (j) to $\mathcal{A}_2$ as a result of the challenge encapsulation is completely random, which is, due to the random oracle model, in agreement with both experiments $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}$, assuming specific hash queries are not submitted.

To ensure consistency, decapsulation and hash queries are taken care of by accurately designed oracles: precisely, line (c) of the $\mathcal{H}$ oracle ensures consistency between the hash function and the components of $\vec{K}^*$; observe that a corresponding condition in the $\mathcal{D}$ oracle is not required as challenge ciphertexts may not be queried for decapsulation. Lines (d) of $\mathcal{D}$ and (d) of $\mathcal{H}$ ensure consistency between the oracles for queries not related to the challenge. Line (b) in the $\mathcal{D}$ oracle and line (b) in the $\mathcal{H}$ oracle mark $\mathcal{A}_1$-queries related to the (yet unknown) challenge ciphertext as bad; we show that this happens only with negligible probability. Lines (e)–(h) of $\mathcal{D}$ and (e)–(h) of $\mathcal{H}$ are standard for implementing consistent random oracles. Note that line (c) of $\mathcal{D}$ is just a rewriting of condition (b) in the $\mathcal{D}_2$ oracle of Figure 4.

Hash function queries related to challenge encapsulations for honest keys are handled in line (a) of $\mathcal{H}$. These queries are of the form $(\mathsf{DH}(U,pk_i),pk_i,j)$. As mentioned above, if no such hash query is submitted by $\mathcal{A}$, its views in experiments $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},0}$ and $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},1}$ are identical. On the other hand, if $\mathcal{A}$ does make such a query, an event we denote E, this allows the extraction of the SDH solution g^{uv} by $\mathcal{B}$, as implemented in line (a).

We continue with the analysis of $\mathcal{B}$'s advantage, starting with bounding the probabilities that flags Bad_1 and Bad_2 are set, and that event E occurs. Observe that the two flags can be set only in the first phase of $\mathcal{B}$'s simulation, and that the corresponding conditions depend on $\mathcal{A}_1$ guessing the right value of U *before* obtaining any information about it. Since U is uniformly distributed, the probability that in any query $\mathcal{A}_1$ finds the right value is $1/|\mathbb{G}|$. That is, the probabilities of these flags being set are bounded by

$$\Pr[\mathsf{Bad}_1] \leq \frac{q_{d1}}{p} \quad \text{and} \quad \Pr[\mathsf{Bad}_2] \leq \frac{q_{h1}}{p} \ .$$

As highlighted above, if the event E occurs, $\mathcal{B}$ solves the SDH problem, i.e., $\Pr[E] = \mathrm{Adv}^{\mathsf{SDH}}_{\mathcal{G},\mathcal{B}}$.

Observe that $\mathcal{B}$ provides a perfect simulation of the experiments $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}_{\mathsf{HEK},\mathcal{A},k,n}$, $b \in \{0,1\}$, if the events Bad_1, Bad_2, and E do not occur. Denote by S_b the event that $\mathcal{A}_2$ outputs 1 at the end of $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}_{\mathsf{HEK},\mathcal{A},k,n}$. The above allows us to conclude

$$\Pr[S_0 \mid \neg E, \neg\mathsf{Bad}_1, \neg\mathsf{Bad}_2] = \Pr[S_1 \mid \neg E, \neg\mathsf{Bad}_1, \neg\mathsf{Bad}_2] \ ,$$

which implies

$$\begin{aligned}\mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{\mathsf{HEK},\mathcal{A},k,n}(\lambda) &= |\Pr[S_0] - \Pr[S_1]| \\ &\leq \Pr[E] + \Pr[\mathsf{Bad}_1] + \Pr[\mathsf{Bad}_2] \ .\end{aligned}$$

Combining this with the above established bounds on $\Pr[E]$, $\Pr[\mathsf{Bad}_1]$, and $\Pr[\mathsf{Bad}_2]$, we have

$$\mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{\mathsf{HEK},\mathcal{A},k,n}(\lambda) \leq \mathrm{Adv}^{\mathsf{SDH}}_{\mathcal{G},\mathcal{B}}(\lambda) + \frac{q_{d1}}{p} + \frac{q_{h1}}{p}.$$

□

Remark 2. *It is instructive to trace how the various components of Construction 2 contribute to its security: Firstly,*

we exploit the power of the static Diffie-Hellman assumption, as opposed to standard CDH, for getting rid of the KOSK restriction required in [5]; more precisely, in the proof of Theorem 2 it is exactly the oracle from Definition 2 that allows a consistent simulation of the challenge encryption and the random oracle—even in the presence of maliciously chosen public keys. Secondly, by incorporating the counter i into the input of hash function H in the Encap *routine we prevent the second attack described in Section 3.2.2 from working: intuitively, if the same public key appears more than once in $\vec{pk}$, the different values of i will ensure that all occurrences are treated independently of each other.*

Conclusion

We revisit the security notion for multi-recipient encryption introduced in [5] and point out a number of subtle yet serious technical flaws. We propose a strengthened security model that fixes the identified issues and also avoids the unrealistic knowledge-of-secret-key assumption. We proceed by lifting the hybrid KEM/DEM construction methodology for achieving public key encryption to the multi-recipient setting and formally prove the soundness of this approach. Finally, we propose a new multi-recipient KEM (and, hence, a multi-recipient PKE scheme) that we prove secure under the static Diffie-Hellman assumption, in the random oracle model. We leave the construction of a standard model multi-recipient PKE scheme that achieves the level of security implied by our definitions for future work.

6. REFERENCES

[1] M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In D. Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 143–158, San Francisco, CA, USA, Apr. 8–12, 2001. Springer, Berlin, Germany.

[2] J. Baek, R. Safavi-Naini, and W. Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In S. Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 380–397, Les Diablerets, Switzerland, Jan. 23–26, 2005. Springer, Berlin, Germany.

[3] M. Barbosa and P. Farshim. Efficient identity-based key encapsulation to multiple parties. In N. P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 428–441, Cirencester, UK, Dec. 19–21, 2005. Springer, Berlin, Germany.

[4] M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon. Multirecipient encryption schemes: How to save on bandwidth and computation without sacrificing security. *IEEE Transactions on Information Theory*, 53(11):3927–3943, 2007.

[5] M. Bellare, A. Boldyreva, and J. Staddon. Randomness re-use in multi-recipient encryption schemes. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 85–99, Miami, USA, Jan. 6–8, 2003. Springer, Berlin, Germany.

[6] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275, Santa Barbara, CA, USA, Aug. 14–18, 2005. Springer, Berlin, Germany.

[7] S. Chatterjee and P. Sarkar. Multi-receiver identity-based key encapsulation with shortened ciphertext. In R. Barua and T. Lange, editors, *INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 394–408, Kolkata, India, Dec. 11–13, 2006. Springer, Berlin, Germany.

[8] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25, Santa Barbara, CA, USA, Aug. 23–27, 1998. Springer, Berlin, Germany.

[9] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Computing*, 33(1):167–226, 2003.

[10] C. Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 200–215, Kuching, Malaysia, Dec. 2–6, 2007. Springer, Berlin, Germany.

[11] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18, Santa Barbara, CA, USA, Aug. 19–23, 1984. Springer, Berlin, Germany.

[12] A. Fiat and M. Naor. Broadcast encryption. In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491, Santa Barbara, CA, USA, Aug. 22–26, 1993. Springer, Berlin, Germany.

[13] K. Kurosawa. Multi-recipient public-key encryption with shortened ciphertext. In D. Naccache and P. Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 48–63, Paris, France, Feb. 12–14, 2002. Springer, Berlin, Germany.

[14] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62, Santa Barbara, CA, USA, Aug. 19–23, 2001. Springer, Berlin, Germany.

[15] N. P. Smart. Efficient key encapsulation to multiple parties. In C. Blundo and S. Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 208–219, Amalfi, Italy, Sept. 8–10, 2004. Springer, Berlin, Germany.

APPENDIX

Proof of Theorem 1

PROOF. Fix $k, n \in \mathbb{N}$ and an efficient adversary $\mathcal{A}$ against E. The proof proceeds by a series of game hops. Let

$$G^0 = \mathrm{Expt}^{\mathsf{MR\text{-}PKE\text{-}IND},0}_{E,\mathcal{A},k,n}(1^\lambda) \quad G^1 = \mathrm{Expt}^{\mathsf{MR\text{-}PKE\text{-}IND},1}_{E,\mathcal{A},k,n}(1^\lambda)$$

(see Figure 3).

Next, define two other games, H^0 and H^1. For $b \in \{0,1\}$, game H^b is equal to game G^b with the following exceptions:

- in games G^b, the challenge ciphertext is created by

$$(\vec{c^1}, \vec{K}^1) \leftarrow_R M.\mathsf{Encap}\left(\vec{pk}^*\right);$$
$$\vec{c^2} \leftarrow_R D.\mathsf{DEM}(\vec{K}^1, \vec{m}^b);$$
$$\vec{c} \leftarrow ((c_j^1, c_j^2))_{j\in[n]}.$$

  while in games H^b the challenge ciphertext is created by

$$(\vec{c^1}, \vec{K}^1) \leftarrow_R M.\mathsf{Encap}(\vec{pk}^*);$$
$$\vec{c^2} \leftarrow_R D.\mathsf{DEM}(\vec{K}^0, \vec{m}^b);$$
$$\vec{c} \leftarrow ((c_j^1, c_j^2))_{j\in[n]},$$

  where the components of $\vec{K}^0$ are defined like this

$$\begin{cases} K_j^0 \leftarrow_R \mathcal{K}, & pk_j^* \in \vec{pk} \\ K_j^0 \leftarrow K_j^1, & pk_j^* \notin \vec{pk} \end{cases}$$

  for $j \in [n]$.
- when handling decryption queries in games G^b, a DEM key is computed from the first component of the submitted ciphertext, and then used to decrypt the second component according to the scheme's specifications; in games H^b, the key vector $\vec{K}^0$ will be used in decryption queries involving the challenge encapsulations $\vec{c^1}$. That is, when the decryption oracle receives a query $(i, (\hat{c}^1, \hat{c}^2))$ for which $\exists j : pk_j^* = pk_i \wedge \hat{c}^1 = c_j^1$, decryption of $\hat{c}^2$ will be done using K_j^0. In all the other cases, it proceeds normally, first decapsulating $\hat{c}^1$ to obtain a DEM key, and then decrypting $\hat{c}^2$ with it.

We can write

$$\begin{aligned} \mathrm{Adv}^{\mathsf{MR\text{-}PKE\text{-}IND}}_{E,\mathcal{A},k,n}(\lambda) = \Big|\Pr[G^1 = 1] - \Pr[G^0 = 1]\Big| \leq \\ \Big|\Pr[G^1 = 1] - \Pr[H^1 = 1]\Big| + \\ \Big|\Pr[H^1 = 1] - \Pr[H^0 = 1]\Big| + \\ \Big|\Pr[H^0 = 1] - \Pr[G^0 = 1]\Big| \quad . \end{aligned}$$

Note that the difference between games G^0 and H^0, and the difference between games G^1 and H^1, correspond to the difference between the KEM experiments $\mathrm{Expt}^{\mathsf{MR\text{-}KEM\text{-}IND},b}_{E,\mathcal{A},k,n}(1^\lambda)$, $b \in \{0,1\}$. Hence, there exist KEM adversary $\mathcal{B}^0$ and $\mathcal{B}^1$ such that

$$\Big|\Pr[G^l = 1] - \Pr[H^l = 1]\Big| \leq \mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{M,\mathcal{B}^l,k,n}$$

for $l \in \{0,1\}$.

Furthermore, the only difference in games H^0 and H^1 is in the challenge encryption: H^0 always encrypts $\vec{m}^0$ and H^1 always encrypts $\vec{m}^1$. To analyze the distance between these games, we proceed by a hybrid argument. Define a game E^j for $0 \leq j \leq n$ such that $E^0 = H^1$ and $E^n = H^0$: if $\vec{m}$ is the vector that is encrypted, then $m_k = m_k^0$ for all $1 \leq k \leq j$ and $m_k = m_k^1$ for all $j < k \leq n$. That is, in two consecutive games, E^{j-1} and E^j, the only difference is in the component c_j^2 of the challenge ciphertext: in E^{j-1} it is the encryption of an element in $\vec{m}^1$ whereas in E^j it is the encryption of an element in $\vec{m}^0$ of the same length.

It easily follows that there exist adversaries $\mathcal{C}^j$ for $1 \leq j \leq n$, such that $\mathcal{C}^j$ interpolates between the games E^{j-1} and E^j, and

$$\mathrm{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{D,\mathcal{C}^j}(\lambda) = \Big|\Pr[E^{j-1} = 1] - \Pr[E^j = 1]\Big|.$$

Thus,

$$\begin{aligned} &\Big|\Pr\Big[H^1 = 1\Big] - \Pr\Big[H^0 = 1\Big]\Big| \\ &= \sum_{j=1}^{n} \Big|\Pr[E^{j-1} = 1] - \Pr[E^j = 1]\Big| \\ &= \sum_{j=1}^{n} \mathrm{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{D,\mathcal{C}^j}(\lambda) \end{aligned}$$

Hence, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that[1]

$$\begin{aligned} \mathrm{Adv}^{\mathsf{MR\text{-}PKE\text{-}IND}}_{E,\mathcal{A},k,n}(\lambda) \leq\ & 2 \cdot \mathrm{Adv}^{\mathsf{MR\text{-}KEM\text{-}IND}}_{M,\mathcal{B},k,n}(\lambda) + \\ & n \cdot \mathrm{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{D,\mathcal{C}}(\lambda) \quad . \end{aligned}$$

□

[1] $\mathcal{B}$ and $\mathcal{C}$ can be constructed by randomly picking and running an adversary from $\{\mathcal{B}^b\}_{b\in\{0,1\}}$ and $\{\mathcal{C}^j\}_{j\in[n]}$, respectively, which will yield an advantage for $\mathcal{B}$ and $\mathcal{C}$ corresponding to the average advantage of their underlying adversaries.