Multirecipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security

Mihir Bellare, Alexandra Boldyreva, *Member, IEEE*, Kaoru Kurosawa, *Member, IEEE*, and Jessica Staddon

***Abstract*—This paper proposes several new schemes which allow a sender to send encrypted messages to multiple recipients more efficiently (in terms of bandwidth and computation) than by using a standard encryption scheme. Most of the proposed schemes explore a new natural technique called randomness reuse. In order to analyze security of our constructions, we introduce a new notion of multirecipient encryption schemes (MRESs) and provide definitions of security for them. We finally show a way to avoid *ad hoc* analyses by providing a general test that can be applied to a standard encryption scheme to determine whether the associated randomness reusing MRES is secure. The results and applications cover both asymmetric and symmetric encryption.**

***Index Terms*—Cryptography, encryption, provable security, randomness.**

I. Introduction

THIS paper is an extension of [5], [27]. Some extra results are presented in [4].

A. Multirecipient Encryption Schemes (MRESs)

Consider a common scenario when a sender needs to encrypt messages for several recipients. A traditional approach for this task is for a sender to encrypt messages independently using an encryption algorithm of some standard encryption scheme. Depending on the application, the ciphertexts can be sent to the receivers together via broadcast or separately, possibly over some period of time.

In this paper, we propose and analyze the ways to achieve computational and bandwidth savings possible in this scenario due to batching. Since the setting of standard encryption does not allow to exploit batching (because encryption for each receiver is done independently), we first define a new setting of *multirecipient encryption* as follows. (Let us restrict our attention for the moment to asymmetric-key setting. We turn to symmetric-key setting later.) There are n receivers, numbered $1, \ldots, n$. Each receiver i has generated for itself a secret decryption key sk_i and corresponding public encryption key pk_i. The sender now applies a *multirecipient encryption algorithm* $\bar{\mathcal{E}}$ to $pk_1, \ldots, pk_n$ and messages $M_1, \ldots, M_n$ to obtain ciphertexts $C_1, \ldots, C_n$. Each receiver i can apply to sk_i and C_i a decryption algorithm that recovers M_i. We refer to the primitive enabling this type of encryption as a *multirecipient encryption scheme* (MRES). We note that its syntax differs from that of a standard encryption scheme only in that the encryption algorithm of the latter is replaced by a multirecipient encryption algorithm. Key generation and decryption are just like in a standard scheme. We will also consider a scenario when an MRES is used to encrypt a *single* message for all receivers. It can often arise in broadcast applications. We call this subclass of MRESs *single-message MRESs* or SM-MRESs.

A common use of a standard encryption we mentioned above can be described by a naive MRES as follows. For each i, let C_i be the result of applying the encryption algorithm $\mathcal{E}$ of a standard scheme to pk_i, M_i. However, it is possible to exploit batching and construct more efficient MRESs. To exemplify this, we sketch the constructions of several MRESs we propose and discuss the efficiency savings they permit. Further, we discuss the security of proposed schemes. Since most of the schemes we present explore an interesting and natural technique, which we call *randomness reuse*; accordingly, we start with the description of this idea and the corresponding subclass of MRESs that exploit randomness reuse.

B. Randomness Reusing MRESs

We propose to consider MRESs constructed from the standard encryption schemes by applying what we call randomness reuse. Namely, we suggest, that reusing random coins when computing ciphertexts for different receivers may often provide computational and bandwidth savings. Consider a multirecipient encryption algorithm that works as follows: given messages $M_1, \ldots, M_n$ and keys $pk_1, \ldots, pk_n$, it picks at random coins r for a single application of the encryption algorithm $\mathcal{E}$ of an underlying standard encryption scheme, and then outputs $(C_1, \ldots, C_n)$, where $C_i = \mathcal{E}_{pk_i}(M_i, r)$ is the encryption of message M_i under key pk_i and coins r $(1 \leq i \leq n)$. The corresponding MRES is called the *randomness reusing* MRES

Manuscript received September 22, 2006; revised April 10, 2007. The work of M. Bellare was supported by the National Science Foundation under Grants CNS-0524765, CNS-0627779, and a gift from Intel Corporation. The work of A. Boldyreva was supported in part by the National Science Foundation under CAREER Award 0545659.

M. Bellare is with the Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: mihir@cse.ucsd.edu).

A. Boldyreva is with the College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332 USA (e-mail: aboldyre@cc.gatech.edu).

K. Kurosawa is with the Department of Computer and Information Sciences, Ibaraki University, Ibaraki, 316-8511, Japan (e-mail: kurosawa@mx.ibaraki.ac.jp).

J. Staddon is with the Computing Science Laboratory, Palo Alto Research Center, Palo Alto, CA 94304USA (e-mail: staddon@parc.com.)

Communicated by E. Okamoto, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2007.907471

(RR-MRES) associated to the underlying standard encryption scheme.

C. Efficient MRESs

ELGAMAL AND CRAMER–SHOUP. Suppose receiver i has secret key $x_i \in \mathbb{Z}_q$ and public key g^{x_i}, operations being in some global, fixed group of order q. The naive ElGamal-based MRES is the following: Pick $r_1, \ldots, r_n$ independently at random from $\mathbb{Z}_q$ and let $C_i = (g^{r_i}, g^{x_i r_i} \cdot M_i)$ for $1 \leq i \leq n$. Instead, we suggest that one pick just one r at random from $\mathbb{Z}_q$ and set $C_i = (g^r, g^{x_i r} \cdot M_i)$ for $1 \leq i \leq n$. In other words we propose the ElGamal-based RR-MRES.

The associated RR-MRES is of interest because compared to the naive one it permits reductions in both computation and broadcast ciphertext size. First, it results in bandwidth reduction in the case that the ciphertexts are being broadcast or multicast by the sender, since in that case the transmission would be $\boldsymbol{C} = (g^r, g^{x_1 r} \cdot M_1, \ldots, g^{x_n r} \cdot M_n)$, which is about half as many bits as required to transmit the ciphertexts computed by the naive method. Second, the suggested scheme (approximately) halves the computational cost (number of exponentiations) for encryption as compared to the naive method. We also suggest that the RR-MRES derived in a similar way from the Cramer–Shoup encryption scheme [16] permits similar computational savings.

CBC. We also consider the symmetric setting. We consider popular CBC encryption with random initial vector (IV), based on a given block cipher. The IV is the randomness underlying the encryption. Randomness reuse is interesting in this context because it means that CBC encrypted ciphertexts to different receivers can use the same IV, thereby yielding savings in bandwidth for broadcast. If the message is one block long then the CBC-based RR-MRES allows to reduce the length of the broadcast ciphertext by 50%.

HYBRID ENCRYPTION. In practice, asymmetric and symmetric encryption schemes are usually used together in the following "hybrid" manner. A sender uses an asymmetric encryption scheme to encrypt a random "session" symmetric key under the receiver's public key and then uses a symmetric encryption scheme to encrypt a message under the symmetric key.

Now consider a scenario when a sender uses a hybrid encryption scheme to encrypt a *single* message under public keys of several recipients, and send, possibly via broadcast, the resulting ciphertexts to the receivers. A naive SM-MRES would ask a sender to use fresh random coins each time it encrypts a message. This includes picking a new symmetric key for each recipient. However, we propose a sender to use the same session symmetric key for all receivers. This is attractive since when a single symmetric key is used the symmetric ciphertext is the same for all receivers and can be sent (broadcast) only once, thus providing bandwidth savings. Moreover, the random coins can possibly be reused when encrypting the symmetric key thereby providing additional savings.

We note that security results for the above schemes do not follow from any previously known results. We need to specifically address security of the above schemes and MRESs in general.

D. Security Notions for MRESs

The preceding examples show that there are MRESs that are more efficient than the naive one. But are they secure? The first step toward answering this important question is to ask what "secure" means in this context. That is, we need appropriate models and definitions of security, in particular extensions of standard definitions such as IND-CPA and IND-CCA to the MRES context.

We envision a very powerful adversary. As usually, we consider the standard chosen-plaintext (resp., chosen-ciphertext) attacks. In addition, we take into account a scenario where the adversary is one of the recipients, enabling it to mount what we call *insider attacks*. As a legitimate recipient it could decrypt a received ciphertext, and might then obtain the coins underlying that ciphertext. This is not a concern if, as in the multiuser setting of [3], [1], encryptions to other recipients use independent coins, but ciphertexts created by a multirecipient encryption algorithm might be based on related coins. So in the latter case, possession of the coins underlying a ciphertext sent to one recipient might enable the adversary to compromise the security of ciphertexts sent to other, legitimate recipients. Our model takes this into account by allowing the adversary to corrupt some fraction of the users and thereby come into possession of their decryption keys.

A stronger form of insider attack that one could consider is to allow the adversary to specify the (public) encryption keys of the corrupted recipients. (In such a *rogue-key* attack, it would register public keys created as a function of public keys of other, legitimate users or would register "invalid" public keys that cannot normally be output by the key-generation algorithm.) Such attacks can be extremely damaging, as we illustrate in Section IV with a rogue-key attack that breaks the above-mentioned ElGamal-based MRES. It is important to be aware of such attacks, but it is for such reasons that certification authorities require (or should require in certain scenarios) that a user registering a public encryption key prove knowledge of the corresponding secret decryption key and "validity" of the public key. This can be done by the user proving knowledge of the random coins used in the key-generation algorithm. (In that case, our attack fails.) Accordingly, our model does allow rogue-key attacks, but does not give the adversary complete freedom in specifying encryption keys of corrupted recipients. Rather, we require that it may do so only if it also provides coins that are used by the key-generation algorithm to output a pair of a public and secret keys.

SECURITY OF SINGLE-MESSAGE MRESS. We also consider a definition of security for SM-MRESs, which is a special case of a more general security definition for MRESs. The difference is that in the case of SM-MRESs, insider attacks are not a threat since all users receive a single message. Accordingly, the adversary is not allowed to corrupt recipients.

E. Reproducibility Theorem for Randomness Reusing MRESs

Many RR-MRESs offer performance benefits, but not all are secure. (We illustrate the latter in Section V by showing how Håstad's attacks [24] can be exploited to break RR-MRESs based on RSA-OAEP [10].) We are interested in determining which RR-MRESs are secure MRESs. Direct case-by-case

analysis of different schemes is possible but would be prohibitive. Instead, we introduce a paradigm based on which one can determine whether a standard encryption scheme permits secure randomness reuse (meaning the associated RR-MRES is a secure MRES) based on existing security results about the underlying (base) standard encryption scheme. It takes two parts: definition of a property of encryption schemes called *reproducibility*, and a theorem, called the *reproducibility theorem*. The latter says that if a standard encryption scheme is reproducible and is IND-CPA (resp., IND-CCA) in the standard, single-receiver setting, then the corresponding RR-MRES is also IND-CPA (resp., IND-CCA) with respect to our notions of security for such schemes. It is usually easy to check whether a given encryption scheme is reproducible, so numerous applications follow. The approach and result hold for both asymmetric and symmetric encryption.

Reproducibility itself is quite simply explained. Considering first the case where the standard encryption scheme is asymmetric, let pk_1, pk_2 be public encryption keys, and let $C_1 = \mathcal{E}_{pk_1}(M_1, r)$ be a ciphertext of a message M_1 created under key pk_1 based on random string r. We say that the encryption scheme is *reproducible* if, given pk_1, pk_2, C_1, any message M_2, and the secret decryption key $sk2$ corresponding to pk_2, there is a polynomial time *reproduction algorithm* that returns the ciphertext $C_2 = \mathcal{E}_{pk_2}(M_2, r)$. The symmetric case is analogous except that the reproduction algorithm is denied the first encryption key because this is also the decryption key.

F. Security of the Proposed MRESs

We now discuss security of the MRESs we discussed before. ELGAMAL AND CRAMER–SHOUP. We show that the base ElGamal and Cramer–Shoup schemes are both reproducible. Our reproducibility theorem together with known results stating that under the decision Diffie–Hellman (DDH) assumption ElGamal is IND-CPA secure Cramer–Shoup is IND-CCA secure [16], imply that under the same assumption the ElGamal RR-MRES is IND-CPA secure and the Cramer–Shoup based one is IND-CCA secure.

In [4], we extend these results by providing reductions with improved concrete security. These improvements do not use the reproducibility theorem, instead directly exploiting the reproducibility property of the base schemes and, as in [3], using self-reducibility properties of the DDH problem [34], [29], [33]. There we also show that the RR-MRES based on DHIES scheme proposed in [2] and adopted by draft standards ANSI X9.63EC and IEEE P1363a is IND-CCA under the assumptions used to establish that DHIES is IND-CCA, and permits badwidth and computational savings similarly to ElGamal and Cramer–Shoup.

CBC ENCRYPTION. We show that the base CBC encryption scheme is reproducible. Since it is known to be IND-CPA assuming the block cipher is a pseudorandom permutation [6], the reproducibility theorem implies that the randomness reusing CBC MRES is IND-CPA under the same assumption.

HYBRID ENCRYPTION. It is well known that if the asymmetric and symmetric schemes are both IND-CPA (resp., IND-CCA) secure, then the standard hybrid scheme is also IND-CPA (resp., IND-CCA) secure. The results of [3] imply that if the hybrid scheme is IND-CPA (resp., IND-CCA) secure, then it is also IND-CPA (resp., IND-CCA) secure in the multiuser setting. But this assumes that a sender uses fresh random coins each time it encrypts a message including picking a new symmetric key for each recipient. Thus, the results of [3] do not imply that the hybrid SM-MRES we proposed is secure. Our results (see Section X) imply that if the asymmetric and symmetric schemes are both IND-CPA (resp., IND-CCA) secure, then the corresponding hybrid SM-MRES is also IND-CPA (resp., IND-CCA) secure. More precisely, we construct a hybrid SM-MRES using any symmetric encryption scheme and an asymmetric SM-MRES. We show that if the symmetric scheme is IND-CPA (resp., IND-CCA) secure and the SM-MRES is IND-CPA (resp., IND-CCA) secure, then the corresponding hybrid SM-MRES is IND-CPA (resp., IND-CCA) secure.[1] We note that since not all hybrid SM-MRESs fall into a subclass of RR-MRESs (savings can be achieved even when the coins used to encrypt the symmetric keys are not reused) we do not apply the reducibility theorem in our analysis. However, our results imply that if the underlying SM-MRES is a secure RR-MRES, all random coins can be reused in the encryption algorithm of the hybrid SM-MRES.

G. Minimal Assumptions for Secure Randomness Reuse

A basic theoretical question is: under what assumptions can one prove the existence of a standard encryption scheme whose associated RR-MRES is a secure MRES? We determine minimal assumptions. We show that there exists a standard encryption scheme whose associated RR-MRES is IND-CPA (resp., IND-CCA) secure if and only if there exists a standard IND-CPA (resp., IND-CCA) secure encryption scheme. These results, detailed in Section VIII, are obtained by transforming a given standard encryption scheme into another standard encryption scheme that permits secure randomness reuse. The transformation uses a pseudorandom function and is simple and efficient. However, one should note that the resulting RR-MRES does not yield savings in bandwidth for broadcast encryption.

H. Discussion and Related Work

ON REUSING RANDOMNESS. At first glance, reusing coins for different encryptions sounds quite dangerous. This is because of the well-known fact that privacy in the sense of IND-CPA is not met if two messages are encrypted using the same coins under the same key. (An attacker can tell whether or not the messages are the same by seeing whether or not the ciphertexts are the same.) However, in an RR-MRES, the different encryptions, although using the same coins, are under *different* keys. Our results indicate that in this case, security is possible. We consider this an interesting facet of the role of randomness in encryption.

A recent paper [9] shows how to utilize reusing randomness to achieve even better efficiency for some schemes. They consider stateful encryption that generalizes MRES, and show that batching can also be exploited when multiple messages are sent to receivers (multiple or single.)

[1] In fact, similarly to the case of regular hybrid encryption schemes, the symmetric scheme can satisfy a weaker security definition. We provide the details in Section X.

USING PRGS. A natural question is, instead of reusing randomness, why not use pseudorandom bit generators (PRGs)? Indeed, randomness generation costs for encryption can be reduced by picking a single, short random seed s and applying a PSG G to obtain a sequence $r_1, r_2, \ldots$ of strings to play the role of coins for successive encryptions. If G is cryptographically secure in the sense of [12], [36], then it is easy to see that the resulting encryption preserves semantic security, not only for encryption to different receivers, but even for multiple encryptions to a single receiver.

However, randomness reuse permits applications that usage of pseudorandomness does not permit. A case in point is the efficiency improvements discussed above. Furthermore, randomness reuse is attractive even in the absence of such applications because it is simple and efficient. A hardware implementation, for example, would benefit from not having to spend real estate on implementation of a pseudorandom bit generator.

RELATION TO BROADCAST ENCRYPTION. MRESs and broadcast encryption schemes (BESs) [19] differ as follows.

- In a BES, the key generation process may be executed by the sender and yields a sequence of possibly related encryption keys, one per recipient, while in a MRES, key generation is like that of a standard scheme, meaning each recipient produces (and registers) its own encryption keys for its own use.
- In a BES, the encryption process takes as input a sequence of encryption keys and a *single* message and produces a *single* ciphertext $\boldsymbol{C}$ called a broadcast ciphertext, while in a general MRES, the encryption process takes as input a sequence of encryption keys and a *sequence* of messages, and produces a corresponding *sequence* of ciphertexts $(\boldsymbol{C}[1], \ldots, \boldsymbol{C}[n])$ one for each recipient.

Perhaps more succinctly, an MRES is simply a way to mimic, or duplicate, the functionality of a standard encryption scheme while attempting to use batching to obtain some cost benefits, while broadcast encryption has a different goal. However, any MRES can be transformed into a natural associated BES as follows. Recipients are given independently generated keys, and message M is encrypted by running the multirecipient encryption algorithm with all messages set to M to yield a vector which plays the role of the broadcast ciphertext and is sent to all recipients. Each recipient extracts the component of the vector pertinent to it and decrypts this to obtain the broadcast message.

II. PRELIMINARIES

A. Notation

Let $\mathbb{N} = \{1, 2, 3, \ldots\}$. For $k \in \mathbb{N}$ let $\mathbb{Z}_k$ denote the ring of integers modulo k. We denote by $\{0,1\}^*$ the set of all binary strings of finite length. If X is string then $|X|$ denotes its length in bits and if X, Y are strings then $X \| Y$ denotes the concatenation of X and Y. If S is a set then $X \xleftarrow{\$} S$ denotes that X is selected uniformly at random from S. For convinience, for any $k \in \mathbb{N}$ we will often write $X_1, X_2, \ldots, X_k \xleftarrow{\$} S$ as a shorthand for $X_1 \xleftarrow{\$} S; X_2 \xleftarrow{\$} S; \ldots; X_n \xleftarrow{\$} S$. If $k \in \mathbb{N}$ then 1^k denotes the string consisting of k consecutive "1" bits. If A is a randomized algorithm and $k \in \mathbb{N}$, then the notation $X \xleftarrow{\$} A(X_1, X_2, \ldots, X_k)$ denotes that X is assigned the outcome of the experiment of running A on inputs $X_1, X_2, \ldots, X_k$. If A is deterministic, we might drop the dollar sign above the arrow. When describing algorithms, $X \leftarrow Y$ denotes that X is assigned the value Y. "RPT" (resp., "PT") stands for "randomized, polynomial-time," (resp., "polynomial-time") and "RPTA" (resp., "PTA") for "RPT algorithm" (resp., "PT algorithm").

Everywhere in text $k \in \mathbb{N}$ is the security parameter and $n(\cdot)$ is a polynomial that denotes the number of recipients of encrypted messages.

B. Definitions

A function $f : \mathbb{N} \to [0,1]$ is called *negligible* if it approaches zero faster than the reciprocal of any polynomial, i.e., for any polynomial p, there exists $n_p \in \mathbb{N}$ such that for all $n \geq n_p$, $f(n) \leq 1/p(n)$.

ASYMMETRIC ENCRYPTION SCHEMES. We recall the standard definitions, following [3] in extending the usual syntax to allow a "common key generation" algorithm. Thus, an *asymmetric (public-key) encryption scheme* $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms.

- The RPT *common-key generation* algorithm $\mathcal{G}$ takes as input 1^k, where $k \in \mathbb{N}$ is a security parameter and returns a *common key* I.
- The RPT *key generation* algorithm $\mathcal{K}$ takes as input a common key I and returns a pair (pk, sk) consisting of a public key and a corresponding secret key.
- The RPT *encryption* algorithm $\mathcal{E}$ takes input a common key I, a public key pk, and a plaintext M and returns a ciphertext.
- The PT *decryption* algorithm $\mathcal{D}$ takes a common key I, a secret key sk, and a plaintext M and returns the corresponding plaintext or a special symbol $\perp$ indicating that the ciphertext was invalid.

Associated to each common key I is a *message space* $\mathrm{MsgSp}(I)$ from which M is allowed to be drawn. We require that the following experiment returns 1 with probability 1.

$$I \xleftarrow{\$} \mathcal{G}(1^k); (pk, sk) \xleftarrow{\$} \mathcal{K}(I); M \xleftarrow{\$} \mathrm{MsgSp}(I).$$

If $\mathcal{D}_{I,sk}(\mathcal{E}_{I,pk}(M)) = M$ then return 1 else return 0.

We will use the terms "plaintext" and "message" interchangeably.

In our context, it is important to make explicit the random choices underlying the randomized key-generation and encryption algorithms $\mathcal{K}, \mathcal{E}$. The notation $(pk, sk) \xleftarrow{\$} \mathcal{K}(I)$ is a shorthand for $r \xleftarrow{\$} \mathrm{Coins}_{\mathcal{K}}(I)$; $(pk, sk) \xleftarrow{\$} \mathcal{K}(I, r)$ and the notation $C \xleftarrow{\$} \mathcal{E}_{I,pk}(M)$ is thus shorthand for

$$r \xleftarrow{\$} \mathrm{Coins}_{\mathcal{E}}(I); C \leftarrow \mathcal{E}_{I,pk}(M, r)$$

where $\mathrm{Coins}_{\mathcal{K}}(I), \mathrm{Coins}_{\mathcal{E}}(I)$ are set from which $\mathcal{K}, \mathcal{E}$, respectively, draw their coins. As the notation indicates, these sets can depend on I.

As an example to illustrate the addition of a common-key generation algorithm to the usual syntax, consider a Diffie–Hellman

based scheme. Here the common key I could include a description of a group and a generator for this group. Different parties may have different keys, but the algorithms are all in the same group.

SECURITY OF ASYMMETRIC ENCRYPTION. We recall the standard notion of security of asymmetric encryption schemes in the sense of indistinguishability. We consider both chosen-plaintext and chosen-ciphertext attacks. The ideas are from [22], [28], [32].

Definition 2.1: Let $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let $A_{\rm cpa}, A_{\rm cca}$ be adversaries which run in two stages and in both stages the latter has access to an oracle. For $b = 0, 1$ and $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ define the experiments

$\mathbf{Experiment}\ \mathbf{Exp}^{\mathrm{atk-b}}_{\mathcal{AE}, A_{\mathrm{atk}}}(k)$
$I \xleftarrow{\$} \mathcal{G}(1^k); (pk, sk) \xleftarrow{\$} \mathcal{K}(I)$
$(M_0, M_1, st) \xleftarrow{\$} A^{\mathcal{O}}_{\mathrm{atk}}(\mathrm{find}, I, pk)$
$C \xleftarrow{\$} \mathcal{E}_{I,pk}(M_b)$
$d \xleftarrow{\$} A^{\mathcal{O}}_{\mathrm{atk}}(\mathrm{guess}, C, st)$
Return d

In the preceding st denotes the state information the adversary wants to preserve. If $\mathrm{atk} = \mathrm{cpa}$ then $\mathcal{O} = \varepsilon$ and if $\mathrm{atk} = \mathrm{cca}$ then $\mathcal{O} = \mathcal{D}_{I,sk}(\cdot)$. It is mandated that $|M_0| = |M_1|, M_0, M_1 \in \mathrm{MsgSp}(I)$ and $A_{\rm cca}$ does not make oracle query C in the guess stage. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ we define the *advantages* of the adversaries $\mathbf{Adv}^{\mathrm{atk}}_{\mathcal{AE}, A_{\mathrm{atk}}}(k)$ as

$$\Pr\left[\mathbf{Exp}^{\mathrm{atk-0}}_{\mathcal{AE}, A_{\mathrm{atk}}}(k) = 0\right] - \Pr\left[\mathbf{Exp}^{\mathrm{atk-1}}_{\mathcal{AE}, A_{\mathrm{atk}}}(k) = 0\right].$$

The scheme $\mathcal{AE}$ is said to be *IND-CPA secure* (resp., *IND-CCA secure)* if the function $\mathbf{Adv}^{\mathrm{cpa}}_{\mathcal{AE}, A_{\mathrm{cpa}}}(\cdot)$ (resp., $\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{AE}, A_{\mathrm{cca}}}(\cdot)$) is negligible for any RPT adversary. □

The concrete-security considerations we will enter at some points in this paper are facilitated by adopting some conventions. Namely, the "time complexity" of the adversary above is the worst case execution time of the associated experiment plus the size of the code of the adversary, in some fixed random-access memory (RAM) model of computation. (Note that the execution time refers to the entire experiment, not just the adversary. In particular, it includes the time for key generation, challenge generation, and computation of responses to oracle queries, if any.) The same convention is used for all other definitions in this paper.

III. MULTIRECIPIENT ASYMMETRIC ENCRYPTION SCHEMES

A. Syntax

An asymmetric *multirecipient encryption scheme (MRES)* $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ consists of four algorithms. The common-key generation algorithm $\mathcal{G}$, key generation algorithm $\mathcal{K}$, and decryption algorithm $\mathcal{D}$ are just like those of an ordinary asymmetric encryption scheme. The RPT *multiencryption* algorithm $\bar{\mathcal{E}}$ takes input a common key I, a *public-key vector* $\boldsymbol{pk} = (\boldsymbol{pk}[1], \ldots, \boldsymbol{pk}[n])$ and a *plaintext vector* $\boldsymbol{M} = (\boldsymbol{M}[1], \ldots, \boldsymbol{M}[n])$ and returns a *ciphertext vector* $\boldsymbol{C} = (\boldsymbol{C}[1], \ldots, \boldsymbol{C}[n])$. Associated to each common key I is a *message space* $\mathrm{MsgSp}(I)$ from which the components of $\boldsymbol{M}$ are allowed to be drawn. We require that the following experiment returns 1 with probability 1.

$I \xleftarrow{\$} \mathcal{G}(1^k)$
For $i = 1, \ldots, n$ do
$(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I); \boldsymbol{M}[i] \xleftarrow{\$} \mathrm{MsgSp}(I)$ EndFor
$\boldsymbol{C} \xleftarrow{\$} \bar{\mathcal{E}}_{I,\boldsymbol{pk}}(\boldsymbol{M})$
$j \xleftarrow{\$} \{1, \ldots, n\}$
If $\mathcal{D}_{I,\boldsymbol{sk}[j]}(\boldsymbol{C}[j]) = \boldsymbol{M}[j]$ then return 1 else return 0

We do not specify how $\boldsymbol{C}[i]$ is communicated to user i. It could be that the whole ciphertext vector $\boldsymbol{C}$ is sent via a broadcast or multicast channel and, if all $\boldsymbol{C}[i]$ have a common part due to a randomness reuse, this part can be sent only once. It could also be that $\boldsymbol{C}[i]$ is sent to party i directly. This issue depends on the specific application and is not relevant for security of the scheme.

SENDING A SINGLE MESSAGE USING MRESS. In a single-message multirecipient encryption schemes (*SM-MRESs*), the encryption algorithm takes input a single message M (rather than a vector of messages) and returns a vector of ciphertexts. Formally, we say that $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \mathcal{E}^1, \mathcal{D})$ is an SM-MRES if there exists a multiencryption algorithm $\bar{\mathcal{E}}$ such that $(\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ is an MRES as defined above and $\mathcal{E}^1$ is defined by

$\mathcal{E}^1_{I,\boldsymbol{pk}}(M)$:
Let n be the number of components of $\boldsymbol{pk}$
For $i = 1, \ldots n$ do $\boldsymbol{M}[i] \leftarrow M$ EndFor
$\boldsymbol{C}[i] \xleftarrow{\$} \bar{\mathcal{E}}_{\boldsymbol{pk}}(\boldsymbol{M}, r)$
Return $\boldsymbol{C}$

B. Randomness Reusing MRESs

Construction 3.1: The *randomness-reusing MRES (RR-MRES)* associated to a given asymmetric encryption scheme $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is the multirecipient encryption scheme $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ in which the common key generation, key generation algorithms, and decryption algorithms are that of $\mathcal{AE}$ and the multirecipient encryption algorithm is defined as follows:

$\bar{\mathcal{E}}_{I,\boldsymbol{pk}}(\boldsymbol{M})$
Let n be the number of components of $\boldsymbol{M}$ and $\boldsymbol{pk}$
$r \xleftarrow{\$} \mathrm{Coins}_{\mathcal{E}}(I)$
For $i = 1, \ldots n$ do $\boldsymbol{C}[i] \leftarrow \mathcal{E}_{pk[i]}(\boldsymbol{M}[i], r)$ EndFor
Return $\boldsymbol{C}$

We refer to $\mathcal{AE}$ as the *base scheme* of $\overline{\mathcal{AE}}$. □

For examples of RR-MRESs see Section VII.

IV. Security of Asymmetric Multirecipient Schemes

We provide the definition and follow it with a discussion illustrating how it takes into account the various security issues mentioned in the Introduction.

Model and Definition. Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be an asymmetric MRES. (We are particularly interested in the case where this is an RR-MRES scheme, but the definition is not restricted to this case.) Let B be an adversary attacking $\overline{\mathcal{AE}}$. B runs in three stages. In the select stage, the adversary is given the number of users and an initial information string and outputs a state information st and an integer l such that $1 \le l \le n(k)$, which indicates that it wants to corrupt $n(k) - l$ users, assumed without loss of generality to be users $l+1, \ldots, n(k)$. In the find stage, the adversary is given the common key I, st, and the public keys of the honest users $1, \ldots, l$. It outputs *two* l-vectors of messages corresponding to choices for the honest users; *one* $(n(k) - l)$-vector of messages corresponding to choices for the corrupted users; an $(n(k) - l)$-vector of random coins which are later used in the key-generation algorithm to create keys for the corrupted users (see the discussion below). Based on a challenge bit b, one of the two l-vectors is selected, and the components of the $(n(k) - l)$-vector of messages are appended to yield a challenge n-vector of messages $\boldsymbol{M}$. The latter is encrypted via the multiencryption algorithm to yield a challenge ciphertext $\boldsymbol{C}$ that is returned to the adversary, now in its guess stage. Finally, B returns a bit d as its guess of the challenge bit b. In each stage, the adversary will output state information that is returned to it in the next stage. In case of chosen-ciphertext attacks in the find and guess stages B is given l decryption oracles corresponding to the secret keys of the honest users. We now provide a formal definition.

Definition 4.1: Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be a multireceiver asymmetric encryption scheme. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ and $b \in \{0,1\}$ consider the experiments:

Experiment $\mathbf{Exp}^{\text{mr-atk-b}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$

(1) $I \xleftarrow{\$} \mathcal{G}(1^k); (1^l, \mathrm{st}) \xleftarrow{\$} B\,(\mathrm{select}, n(k), I)$
$[1 \le l \le n(k)]$

(2) For $i = 1, \ldots, l$ do $(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor

(3) $(\boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathbf{coins}, s) \xleftarrow{\$} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\mathrm{find}, \boldsymbol{pk}, s)$
$[|\boldsymbol{M}_0| = |\boldsymbol{M}_1| = l; |\boldsymbol{M}| = n(k) - l]$
$[|\boldsymbol{pk}| = l; |\mathbf{coins}| = n(k) - l]$

(4) For $i = l+1, \ldots, n(k)$ do
$(\boldsymbol{pk}'[i], \boldsymbol{sk}'[i]) \xleftarrow{\$} \mathcal{K}(I, \mathbf{coins}[i])$ EndFor

(5) $\boldsymbol{pk} \leftarrow (\boldsymbol{pk}[1], \ldots, \boldsymbol{pk}[l], \boldsymbol{pk}'[l+1], \ldots, \boldsymbol{pk}'[n(k)])$

(6) $\boldsymbol{M} \leftarrow (\boldsymbol{M}_b[1], \ldots, \boldsymbol{M}_b[l], \boldsymbol{M}[1], \ldots, \boldsymbol{M}[n(k) - l])$

(7) $\boldsymbol{C} \xleftarrow{\$} \bar{\mathcal{E}}_{I,\boldsymbol{pk}}(\boldsymbol{M})$

(8) $d \xleftarrow{\$} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\mathrm{guess}, \boldsymbol{C}, s)$

(9) Return d

Above, the oracles for $1 \le i \le l$ are defined as follows: If $\mathrm{atk} = \mathrm{cpa}$ then $\mathcal{O}_i(\cdot) = \varepsilon$ and if $\mathrm{atk} = \mathrm{cca}$ then $\mathcal{O}_i(\cdot) = \mathcal{D}_{I,\boldsymbol{sk}[i]}(\cdot)$. It is mandated that for all $1 \le i \le l$ we have $|M_0[i]| = |M_1[i]|$ and all message vector components are in the scheme's message space, and also that if $\mathrm{atk} = \mathrm{cca}$ then the adversary B does not query $\mathcal{O}_i(\cdot)$ on $\boldsymbol{C}[i]$. The restriction on decryption oracle queries is necessary since otherwise the adversary can decrypt the corresponding part of the challenge ciphertext vector and therefore distinguish which plaintext vector was encrypted.

The adversary's ind-atk advantage $\mathbf{Adv}^{\text{mr-atk}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$ is defined as

$$\Pr\left[\mathbf{Exp}^{\text{mr-atk-0}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) = 0\right] - \Pr\left[\mathbf{Exp}^{\text{mr-atk-1}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) = 0\right].$$

We say that MRES $\overline{\mathcal{AE}}$ is IND-CPA (resp., IND-CCA) secure if the function $\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$ (resp., $\mathbf{Adv}^{\text{mr-cca}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$) is negligible for any RPTA B and any polynomial n. □

Security of SM-MRESs. In order to define security for a SM-MRES $\overline{\mathcal{AE}}$ for $\mathrm{atk} = \{\mathrm{cpa}, \mathrm{cca}\}$ we define $\mathbf{Exp}^{\text{smmr-atk-b}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$ similarly to $\mathbf{Exp}^{\text{mr-atk-b}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$ defined in Definition 4.1, except now the adversary is not allowed to corrupt users. Below, we specify the lines of the experiment description that are different from those of $\mathbf{Exp}^{\text{mr-atk-b}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$, the rest of the description is identical.

Experiment $\mathbf{Exp}^{\text{smmr-atk-b}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$

(1) $I \xleftarrow{\$} \mathcal{G}(1^k); (1^l, \mathrm{st}) \xleftarrow{\$} B\,(\mathrm{select}, n(k), I)$
$[l = n(k)]$

$\ldots$

(3) $(\boldsymbol{M}_0, \boldsymbol{M}_1) \xleftarrow{\$} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\mathrm{find}, \boldsymbol{pk})$
$[|\boldsymbol{M}_0| = |\boldsymbol{M}_1| = n(k); \boldsymbol{M}_0[i] = \boldsymbol{M}_0[j];$
$\boldsymbol{M}_1[i] = \boldsymbol{M}_1[j] \forall 1 \le i, j \le n(k)]$

$\ldots$

Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be a single-message multirecipient encryption scheme. The adversary's ind-atk advantage $\mathbf{Adv}^{\text{smmr-atk}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k)$ is defined as

$$\Pr\left[\mathbf{Exp}^{\text{smmr-atk-0}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) = 0\right] - \Pr\left[\mathbf{Exp}^{\text{smmr-atk-1}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) = 0\right].$$

We say that SM-MRES $\overline{\mathcal{AE}}$ is IND-CPA (resp., IND-CCA) secure if the function $\mathbf{Adv}^{\text{smmr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$ (resp., $\mathbf{Adv}^{\text{smmr-cca}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$) is negligible for any RPTA B and any polynomial n. □

Asymmetric Schemes Yielding Secure RR-MRESs. It is convenient to introduce a notion of security for base encryption schemes based on the security of the corresponding RR-MRES. We stress that the following is a notion of security for (standard) asymmetric encryption schemes, not for MRESs.

Definition 4.2: Let $\mathcal{AE}$ be an asymmetric encryption scheme. We say that it is RR-IND-CPA (resp., RR-IND-CCA, RR) secure if the RR-MRES $\overline{\mathcal{AE}}$ associated to $\mathcal{AE}$ is IND-CPA (resp., IND-CCA, IND-CPA, or IND-CCA) secure. □

Discussion. The previous works on the multiuser setting [3], [1] only considered outsider attacks, meaning the adversary was not one of the receivers. However, in the multirecipient setting it is necessary to consider insider attacks. The adversary should be allowed to corrupt some fraction of the users and choose secret and public keys for them.

To justify this claim consider the RR-MRES associated to the ElGamal scheme. It can be shown to be wIND-CPA (a notion similar to our IND-CPA, but that does not take into account insider attacks, [27]). Now consider a modified encryption scheme which differs from ElGamal in that its encryption algorithm when invoked on one particular public key (e.g., g^3) in addition to the ciphertext returns the randomness used to compute it. When this scheme used in a multirecipient setting with randomness reuse, the adversary can register this public key and later, after receiving a ciphertext, can obtain the random coins used to compute the ciphertexts of other users and thus break the scheme. Under our model, the advantage of such adversary in breaking this scheme will be 1. Even though the modified scheme is contrived, this simple example shows an example of insider attacks.

Consider another example which shows the importance of the stronger model. Let $\mathcal{AE}' = (\mathcal{G}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$ be some IND-CPA secure encryption scheme. Consider a multirecipient scheme $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}', \bar{\mathcal{E}}, \mathcal{D})$, where $\mathcal{G}$ runs $\mathcal{G}'$ to get I' and outputs I' and also a description of a group of prime order q and a generator g, $\mathcal{K}$ runs $\mathcal{K}'$ to get (pk', sk'), picks a random element of $\mathbb{Z}_q$, and outputs $((g^x, pk')(x, sk')$. Let us also assume that the message space of $\mathcal{AE}'$ includes $\mathbb{Z}_q$. Let the encryption algorithm of $\mathcal{AE}'$ be as follows.

Algorithm $\bar{\mathcal{E}}_{I,\boldsymbol{pk}}(\boldsymbol{M})$

$r \xleftarrow{\$} \mathbb{Z}_q$

For $i = 1, \ldots, n$ do

$\boldsymbol{C}'[i] \xleftarrow{\$} \mathcal{E}'_{pk'_i}(r); Y_i \leftarrow g^r; W_i \leftarrow (g^{x_i})^r M[i]$

$\boldsymbol{C}[i] \leftarrow (Y_i, W_i, \boldsymbol{C}'[i])$ EndFor

Return $\boldsymbol{C}$

We omit the description of $\mathcal{D}$. We claim that $\overline{\mathcal{AE}}$ is wIND-CPA secure while it is insecure in our model. We first prove the latter claim by presenting a practical attack. An adversary A "corrupts" the first user and chooses $pk_1 = (g^{x_1}, pk'_1)$ in normal way so that it knows x_1, sk'_1. When A receives a ciphertext vector $\boldsymbol{C}$ it decrypts $\boldsymbol{C}'[1]$ using sk'_1 and obtains r. Now A can compute $\boldsymbol{M}[i]$ as $W_i(g^{x_i})^{-r}$. Under our model of security A would have advantage 1. We now show that $\overline{\mathcal{AE}}$ is secure under the weaker notion (wIND-CPA). Let B be an adversary attacking wIND-CPA security of $\overline{\mathcal{AE}}$. Then it is possible to construct an adversary D which attacks ElGamal RR-MRES. D simply provides the common key and all the public keys it is given to B and outputs message vectors that B outputs. D then receives a challenge ciphertext vector $\boldsymbol{C}_D$, picks a random r', and computes a challenge $\boldsymbol{C}_B$ for B such that $\boldsymbol{C}_B[i] = (\boldsymbol{C}_D[i], \mathcal{E}'_{I,pk'_i}(r'))$. Since $\mathcal{AE}'$ is IND-CPA, then the view of B in the simulated experiment is indistinguishable from the real experiment. Therefore, the advantage of B is at most the advantage of D. But it is proven in [27] that the latter scheme is wIND-CPA, so this would imply that $\overline{\mathcal{AE}}$ is also wIND-CPA.

Moreover, for analyses of multirecipient schemes it is important to take into account the possibility of rogue-key attack. This can be particularly damaging in the context of random-string reuse. For example, suppose the adversary registers public keys $(g^x)^2 = g^{2x}$ and $(g^x)^3 = g^{3x}$ where g^x is the key of a legitimate user. Suppose that messages M_1, M, M are ElGamal encrypted with the same randomness r under public keys g^x, g^{2x}, g^{3x} and broadcast to the users. Thus, the adversary sees the three corresponding ciphertexts $(g^r, g^{rx} \cdot M_1), (g^r, g^{2rx} \cdot M), (g^r, g^{3rx} \cdot M)$. From them it can compute $M_1 = [g^{rx} \cdot M_1] \cdot [g^{2rx} \cdot M] \cdot [g^{3rx} \cdot M]^{-1}$ and obtain the message addressed the legitimate user.

As we mentioned in the Introduction, to prevent attacks of this type we put some limitation on the adversary in this regard, in particular to disallow it from creating public keys whose corresponding secret keys it does not know. The model incorporates this by requiring the adversary to supply a list of random coins that are later used in the key-registration algorithm to create the public and secret keys for the corrupted users. This models the effect of appropriate proofs of knowledge of the random coins used in the key-generation algorithm that are assumed to be done as part of the key certification process. The alternative is to explicitly consider the certification process in the model, and then, in proofs of security, use the extractors, guaranteed by the proof of knowledge property [8], to extract the secret keys from the adversary. This being quite a complication of the model, we have chosen to build in the intended effects of the proofs of knowledge.

V. Not Every RR-MRES Scheme Is Secure

We consider general embedding schemes which first apply a randomized invertible transform to a message and then apply a trapdoor permutation to the result. An example of such a scheme is RSA-OAEP [10] that has been proven to be IND-CCA secure (in the random oracle model) [21] and hence is also IND-CCA secure in a multiuser setting [3], [1]. Nonetheless, the associated RR-MRES scheme is insecure. The attack is as follows. Assume all users use public moduli of equal length and have encryption exponent 3. Let N_i be the public modulus of user i. Suppose the sender wants to send a single message M to three receivers, namely, $\boldsymbol{M} = (M, M, M)$. Under the RR-MRES scheme, it will pick a random string r, using M and a random r will compute a transform x, which with high probability will be in $\mathbb{Z}^*_{\mathbb{N}}$ for all i, set $\boldsymbol{C}[i] = x^3 \bmod N_i$, and send $\boldsymbol{C}[i]$ to i. An adversary given $\boldsymbol{C}$ can use Håstad's attack [24] (based on the fact that the moduli are relatively prime) to recover x, and them recover M by inverting the transform. The same attack applies regardless of embedding method, since the latter must be an invertible transform.

This indicates that secure randomness reuse is not possible for *all* base encryption schemes: there exist base encryption schemes that are secure, yet the associated RR-MRES is not secure. In fact, no encryption scheme where the random string used by the encryption algorithm can be obtained by the legitimate receiver who performs the decryption, can be a base of a secure RR-MRES. However, there are large classes of base encryption schemes for which the associated RR-MRES scheme are secure.

VI. Reproducibility Test and Theorem

We provide a condition under which a given encryption scheme can be a base of a secure RR-MRES. Informally speaking, the condition is satisfied for those encryption schemes

for which it is possible, using a public key and a ciphertext of a random message, to create ciphertexts for arbitrary messages under arbitrary keys, such that all ciphertexts employ the same random string as that of the given ciphertext.

Definition 6.1: Fix a public-key encryption scheme $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. Let R be an algorithm that takes as input a common and a public keys and ciphertext of a random message, another random message together with a public-secret key pair, and returns a ciphertext. Consider the following experiment.

Experiment $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{AE},R}(k)$

$I \xleftarrow{\$} \mathcal{G}(1^k); (pk, sk) \xleftarrow{\$} \mathcal{K}(I)$

$M \xleftarrow{\$} \mathrm{MsgSp}(I); r \xleftarrow{\$} \mathrm{Coins}_{\mathcal{E}}(I)$

$C \xleftarrow{\$} \mathcal{E}_{I,pk}(M, r); (pk', sk') \xleftarrow{\$} \mathcal{K}(I); M' \xleftarrow{\$} \mathrm{MsgSp}(I)$

If $\mathcal{E}_{pk'}(M', r) = R(I, pk, C, M', pk', sk')$

Then return 1 else return 0 EndIf

We say that $\mathcal{AE}$ is *reproducible* if for any $k \in \mathbb{N}$ there exists an RPTA R called the reproduction algorithm such that $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{AE},R}(k)$ outputs 1 with probability 1. □

Later we will show that many popular discrete-log-based encryption schemes are reproducible. It is an open question whether there exist reproducible asymmetric encryption schemes of other types.

We now state the main reproducibility theorem. It implies that if an encryption scheme is reproducible and is IND-CPA (resp., IND-CCA) secure, then it is also RR-IND-CPA (resp., RR-IND-CCA) secure.

Theorem 6.2: Fix a public-key encryption scheme $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ and a polynomial n. Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be the associated RR-MRES. If $\mathcal{AE}$ is reproducible then for any RPTA B_{atk}, there exists an RPTA A_{atk}, where $\mathrm{atk} = \{\mathrm{cpa}, \mathrm{cca}\}$, such that for any k

$$\mathbf{Adv}^{\text{mr-atk}}_{\overline{\mathcal{AE}},B_{\mathrm{atk}},n(\cdot)}(k) \le n(k) \cdot \mathbf{Adv}^{\mathrm{atk}}_{\mathcal{AE},A_{\mathrm{atk}}}(k). \qquad \square$$

The proof is given in Appendix A.

VII. Analysis of Specific Schemes

In this section, we show that many popular encryption schemes are reproducible. Using the known results about security of these schemes and the result of Theorem 6.2 this would imply that these schemes are also RR secure.

A. ElGamal

The ElGamal scheme in a group of prime order is known to be IND-CPA under the assumption that the DDH problem is hard. (This is noted in [15], [29], [16], [35].) We will look at the IND-CPA security of the corresponding RR-MRES constructed as per Construction 3.1. We recall the ElGamal scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. The common-key generation algorithm $\mathcal{G}$ on input 1^k, where $k \in \mathbb{N}$ is the security parameter, returns a tuple $(1^k, \tilde{\mathbb{G}}, q, g)$, where q is a prime with $2^{k-1} < q < 2^k$, $\tilde{\mathbb{G}}$ is a description of a group $\mathbb{G}$ of order q, and g is a generator of $\mathbb{G}$. The rest of the algorithms are as follows:

$\underline{\mathcal{K}((1^k, \tilde{\mathbb{G}}, q, g))}$:	$\underline{\mathcal{E}_{(1^k,\tilde{\mathbb{G}},q,g),X}(M)}$:
$x \xleftarrow{\$} \mathbb{Z}_q; X \leftarrow g^x$	$r \xleftarrow{\$} \mathbb{Z}_q; Y \leftarrow g^r$
$pk \leftarrow X; sk \leftarrow x$	$T \leftarrow X^r; W \leftarrow TM$
Return (pk, sk)	Return (Y, W)

$\underline{\mathcal{D}_{(1^k,\tilde{\mathbb{G}},q,g),x}((Y, W))}$:

$T \leftarrow Y^x$

$M \leftarrow WT^{-1}$

Return M

The message space associated to the common key $(1^k, \tilde{\mathbb{G}}, q, g)$ is the group $\mathbb{G}$ itself. Note that a generator g is the output of the common key generation algorithm, which means we fix g for all keys.

Lemma 7.1: The ElGamal encryption scheme

$$\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$$

is reproducible.

Proof: On input $(I, pk, X, (g^r, g^{rx}M), M', pk', sk')$, where $I = (1^k, \tilde{\mathbb{G}}, q, g), pk = g^x, pk' = g^{x'}, sk' = x'$, a PTA R returns $(g^r, (g^r)^{x'} M')$. It is easy to see that R always outputs a valid ciphertext which is created using the same random string as the given ciphertext and therefore the experiment $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{EG},R}(1^k)$ always outputs 1. □

The fact that the ElGamal scheme in a group of prime order is known to be IND-CPA under the assumption that the DDH problem is hard, Theorem 6.2 and Lemma 7.1 imply that the ElGamal scheme is also RR-IND-CPA or, equivalently, $\overline{\mathcal{EG}}$ is IND-CPA secure. However, according to Theorem 6.2 the security degrades linearly as the number of users $n(k)$ increases. In [4], we prove that it is possible to obtain a tighter relation than the one implied by Theorem 6.2 that implies that security of ElGamal RR-MRES almost does not degrade as we add more users.

B. Cramer–Shoup

We now consider an RR-MRES based on the Cramer–Shoup scheme [16], [17] in order to get cost and bandwidth efficiency and IND-CCA security properties. We first recall the Cramer–Shoup scheme. The scheme uses a family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ defined by a probabilistic generator algorithm $\mathcal{GH}$—which takes as input 1^k, where $k \in \mathbb{N}$ is a security parameter and returns a key K, and a deterministic evaluation algorithm $\mathcal{EH}$ which takes as input the key K and a string $X \in \mathbb{G}^3$ and returns a string $\mathcal{EH}_K(X) \in \{0,1\}^{k-1}$. Without loss of generality, we assume that $K \in \{0,1\}^k$. Let $\bar{\mathcal{G}}$ be a prime-order-group generator. The algorithms of the

associated Cramer–Shoup scheme $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ are as follows:

$\underline{\mathcal{G}(1^k)}$:
$(1^k, \tilde{\mathbb{G}}, q, g_1) \xleftarrow{\$} \bar{\mathcal{G}}$
$g_2 \xleftarrow{\$} \mathbb{G}/\{\hat{1}\}$
$K \xleftarrow{\$} \mathcal{GH}(1^k)$
$I \leftarrow (1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K)$
Return I

$\underline{\mathcal{K}((1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K))}$:
$x_1, x_2, y_1, y_2, z_1, z_2 \xleftarrow{\$} \mathbb{Z}_q$
$c \leftarrow g_1^{x_1} g_2^{x_2}$
$d \leftarrow g_1^{y_1} g_2^{y_2}$
$h \leftarrow g_1^{z_1} g_2^{z_2}$
$pk \leftarrow (c, d, h)$
$sk \leftarrow (x_1, x_2, y_1, y_2, z_1, z_2)$
Return (pk, sk)

$\underline{\mathcal{E}_{I,pk}(M)}$:
Parse I as $(1^k, \tilde{\mathbb{G}}, g_1, g_2, K)$
Parse pk as (c, d, h)
$r \xleftarrow{\$} \mathbb{Z}_q$
$u_1 \leftarrow g_1^r; u_2 \leftarrow g_2^r$
$e \leftarrow h^r M$
$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
$v \leftarrow c^r d^{r\alpha}$
Return (u_1, u_2, e, v)

$\underline{\mathcal{D}_{I,sk}((u_1, u_2, e, v))}$:
Parse I as $(1^k, \tilde{\mathbb{G}}, g_1, g_2, K)$
Parse sk as $(x_1, x_2, y_1, y_2, z_1, z_2)$
$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
If $v \neq {u_1}^{x_1 + y_1 \alpha} {u_2}^{x_2 + y_2 \alpha} = v$
Then return $\perp$ EndIf
$f \leftarrow u_1^{z_1} u_2^{z_2}$
$M \leftarrow e/f$
Return M

The message space associated to the common key $(1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K)$ is $\mathbb{G}$.

Lemma 7.2: The Cramer–Shoup encryption scheme

$$\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$$

is reproducible.

The proof of the above lemma is simple and is similar to the proof of Lemma 7.1.

Proof: We present a PTA R which takes as input a common and a public key and a ciphertext of a random message under this key, another random message and a public-secret key pair and returns a ciphertext.

Algorithm $R(I, pk, C, M', pk', sk')$
Parse I as $(1^k, \tilde{\mathbb{G}}, g_1, g_2, K)$
Parse pk as (c, d, h); Parse C as (u_1, u_2, e, v)
Parse pk' as (c', d', h')
Parse sk' as $(x_1', x_2', y_1', y_2', z_1', z_2')$
$e' \leftarrow u_1^{z_1'} u_2^{z_2'} M'; \alpha' \leftarrow \mathcal{EH}_K(u_1, u_2, e')$
$v' \leftarrow u_1^{x_1' + y_1' \alpha'} u_2^{x_2' + y_2' \alpha'}$
Return (u_1, u_2, e', v')

Let us denote the random string used in a challenge ciphertext C as r. First, we note that first two elements $u_1 = g_1^r, u_2 = g_2^r$ of the output ciphertext are equal to the first two elements of a challenge ciphertext C as they should due to a fact that r is fixed. Next we note that $e' = u_1^{z_1'} u_2^{z_2'} M' = g_1^{r z_1'} g_2^{r z_2'} M' = (h')^r M'$. This means that e' and, thus, α' are of the right form. Similarly

$$v' = u_1^{x_1' + y_1' \alpha'} u_2^{x_2' + y_2' \alpha'} = g_1^{r(x_1' + y_1' \alpha')} g_2^{r(x_2' + y_2' \alpha')} = (c')^r (d')^{r\alpha}$$

which is valid computation. Therefore, R always outputs a valid ciphertext which is created using the same random string as a given ciphertext and therefore $\Pr[\mathbf{Exp}^{\text{repr}}_{\mathcal{CS},R}(1^k) = 1] = 1$. □

$\mathcal{CS}$ is proven to be IND-CCA secure assuming that the DDH is hard and $\mathcal{H}$ is target-collision resistant [16], [17]. This being a fact, Theorem 6.2 and Lemma 7.2 imply that it is also RR-IND-CCA or, equivalently, $\overline{\mathcal{CS}}$ is IND-CCA secure. As for the ElGamal scheme, the security of the associated RR-MRES degrades linearly with the number of users. In [4], we provide a better security result than the one implied by Theorem 6.2.

VIII. From IND-CPA (IND-CCA) to RR-IND-CPA (RR-IND-CCA)

As shown in Sections V and VII, some practical encryption schemes such as ElGamal and Cramer–Shoup are RR secure, while some, e.g., RSA-OAEP, are not. We now provide a simple method for an efficient transformation of any encryption scheme which meets the standard notion of security into RR secure one. The construction will use a pseudorandom function family; accordingly, we first recall the notion of pseudorandomness.

Pseudorandom Function Families. Let $kl : \mathbb{N} \to \mathbb{N}$, $il : \mathbb{N} \to \mathbb{N}, ol : \mathbb{N} \to \mathbb{N}$ be polynomially bounded, polynomial-time computable functions, and let $k \in \mathbb{N}$ be a security parameter. A family of functions F is a map $\{0,1\}^{kl} \times \{0,1\}^{il} \to \{0,1\}^{ol}$ which takes a key $K \in \{0,1\}^{kl}$ and an input $x \in \{0,1\}^{il}$ and returns a string $y = F(K, M)$ where $y \in \{0,1\}^{ol}$. The notation $g \xleftarrow{\$} F$ is a shorthand for $K \xleftarrow{\$} \{0,1\}^{kl}; g \leftarrow F(K, \cdot)$. We call g a random instance of F. Let R denote the family of all functions of $\{0,1\}^{il}$ to $\{0,1\}^{ol}$ so that $g \xleftarrow{\$} R$ denotes the operation of selecting at random a function of $\{0,1\}^{il}$ to $\{0,1\}^{ol}$. We call g a random function. An adversary D takes as input 1^k, where $k \in \mathbb{N}$ is the security parameter, and has access to an oracle for a function $g : \{0,1\}^{il} \to \{0,1\}^{ol}$ and outputs a bit.

Definition 8.1: Let F, R be as above, let D be a adversary. The adversary's advantage $\mathbf{Adv}^{\text{prf}}_{F,D}(k)$ is defined as

$$\Pr\left[D^g(1^k) = 1 : g \xleftarrow{\$} F\right] - \Pr\left[D^g(1^k) = 1 : g \xleftarrow{\$} R\right].$$

The function family F is said to be pseudorandom if $\mathbf{Adv}^{\text{prf}}_{F,D}(\cdot)$ is negligible for any RPT adversary. □

We now describe the transformation.

Construction 8.2: Fix an asymmetric encryption scheme $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ and let k be a security parameter. Let (I, pk) denote a string containing I and pk. We assume that there exist polynomially bounded, polynomial-time computable functions $il : \mathbb{N} \to \mathbb{N}, ol : \mathbb{N} \to \mathbb{N}$ such that for all k $|(I, pk)| = il$ and $\text{Coins}(I) = \{0,1\}^{ol}$ for all I generated by $\mathcal{G}(1^k)$ and all pk generated by $\mathcal{K}(1^k)$. Fix a polynomially bounded, polynomial-time computable function $kl : \mathbb{N} \to \mathbb{N}$ and fix a function family $F : \{0,1\}^{kl} \times \{0,1\}^{il} \to \{0,1\}^{ol}$. Then a transformed asymmetric encryption scheme $\mathcal{AE}'[F] = (\mathcal{G}, \mathcal{K}, \mathcal{E}', \mathcal{D})$ has the same common-key-generation, key-generation and decryption algorithms as $\mathcal{AE}$ and the encryption algorithm is defined as follows:

Algorithm $\mathcal{E}'_{I,pk}(M, r')$

$r \leftarrow F(r', (I, pk)); C \xleftarrow{\$} \mathcal{E}_{I,pk}(M, r)$

Return C □

In practice, a block cipher such as AES can be often used in place F (if it is fixed key, input and output lengths satisfy the assumptions described above). Hence, the cost of the transform is negligible.

Theorem 8.3: Fix an asymmetric encryption scheme $\mathcal{AE}$. Assume that there exist functions $il : \mathbb{N} \to \mathbb{N}, ol : \mathbb{N} \to \mathbb{N}$ satisfying the conditions defined above. Let $\mathcal{AE}'[F]$ be a transformed encryption scheme as per Construction 8.2. Let it be a base scheme for the RR-MRES $\overline{\mathcal{AE}'[F]}$ which is defined as per Construction 3.1. Then if $\mathcal{AE}$ is IND-CPA (resp., IND-CCA) secure and F is a pseudorandom function family then $\mathcal{AE}'[F]$ is RR-IND-CPA (resp., RR-IND-CCA) secure, or, equivalently, $\overline{\mathcal{AE}'[F]}$ is IND-CPA (resp., IND-CCA) secure. □

The preceding theorem states the asymptotic security result. In Appendix B, we prove the concrete security result and the statement of the theorem follows.

The preceding results show that one can efficiently modify any RSA embedding encryption scheme, e.g., RSA-OAEP, which is IND-CCA secure (in the random oracle model), by adding one application of a block cipher such that the resulting scheme becomes RR-IND-CCA.

Corollary 8.4: The existence of IND-CPA (resp., IND-CCA) secure asymmetric encryption scheme is a necessary and sufficient condition for the existence of RR-IND-CPA (resp., RR-IND-CCA) encryption scheme.

Proof: It follows from Construction 8.2 and Theorem 8.3 that the existence of IND-CPA schemes and the existence of PRF function families imply the existence of RR-IND-CPA schemes. It is known that the existence of IND-CPA schemes implies the existence of one-way functions [26] and the existence of one-way functions implies the existence of pseudorandom generators [25] which in turn implies the existence of PRFs [23]. Therefore, the existence of IND-CPA schemes implies the existence of RR-IND-CPA schemes. Similarly, for the case of IND-CCA schemes. Another direction of the corollary is trivial. □

IX. Multirecipient Symmetric Encryption Schemes

The results of this paper for the asymmetric-key setting can be easily adjusted to the symmetric-key setting. We first recall syntax for symmetric encryption schemes and the corresponding notion of security under a chosen-plaintext attack.

A. Symmetric Encryption Schemes

Syntax. Following [6], a symmetric encryption scheme $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ associated with a message space $\mathrm{MsgSp}(k)$ consists of three algorithms.

- An RPT key generation algorithm $\mathcal{SK}$ takes a security parameter k and returns a key sk.
- An RPT encryption algorithm $\mathcal{SE}$ takes sk and a message $M \in \mathrm{MsgSp}(k)$ to return a ciphertext C.
- A PT decryption algorithm $\mathcal{D}$ takes sk and a ciphertext C and returns a message M.

We require that for all $k \in \mathbb{N}$, $\mathcal{SD}_{sk}(\mathcal{SE}_{sk}(M)) = M$ for all $M \in \mathrm{MsgSp}(k)$.

Security. Following [6], we recall the security definition of a symmetric-key encryption scheme under chosen-plaintext and chosen-ciphertext attacks.

Definition 9.2: Let $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ be a symmetric-key encryption scheme. Let $A_{\mathrm{cpa}}, A_{\mathrm{cca}}$ be adversaries which run in two stages and in both stages the former has access to an oracle and the latter has access to two oracles. For $b \in \{0, 1\}$ and $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ define the following experiment.

Experiment $\mathbf{Exp}^{\mathrm{atk}-b}_{\mathcal{SE},A_{\mathrm{atk}}}(k)$

$sk \xleftarrow{\$} \mathcal{K}(1^k)$

$(M_0, M_1, \mathrm{st}) \xleftarrow{\$} A_{\mathrm{atk}}{}^{\mathcal{SE}_{sk}(\cdot),\mathcal{O}(\cdot)}(\mathrm{find}, k)$

$C \xleftarrow{\$} \mathcal{SE}_{sk}(M_b)$

$d \xleftarrow{\$} A_{\mathrm{atk}}{}^{\mathcal{SE}_{sk}(\cdot),\mathcal{O}(\cdot)}(\mathrm{guess}, C, \mathrm{st})$

Return d

Above, st denotes the state information the adversary wants to preserve. If $\mathrm{atk} = \mathrm{cpa}$ then $\mathcal{O} = \varepsilon$ and if $\mathrm{atk} = \mathrm{cca}$ then $\mathcal{O} = \mathcal{D}_{sk}(\cdot)$. It is mandated that $|M_0| = |M_1|$ and $M_0, M_1 \in \mathrm{MsgSp}(k)$ above. We require that A_{cca} does not make oracle query C in the guess stage. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ we define the advantages of the adversaries $\mathbf{Adv}^{\mathrm{atk}}_{\mathcal{SE},A_{\mathrm{atk}}}(k)$ as follows:

$$\Pr\left[\mathbf{Exp}^{\mathrm{atk}-0}_{\mathcal{SE},A_{\mathrm{atk}}}(k) = 0\right] - \Pr\left[\mathbf{Exp}^{\mathrm{atk}-1}_{\mathcal{SE},A_{\mathrm{atk}}}(k) = 0\right].$$

The scheme $\mathcal{SE}$ is said to be IND-CPA secure (resp., IND-CCA secure) if the function $\mathbf{Adv}^{\mathrm{cpa}}_{\mathcal{SE},A_{\mathrm{cpa}}}(\cdot)$ (resp., $\mathbf{Adv}^{\mathrm{cca}}_{\mathcal{SE},A_{\mathrm{cca}}}(\cdot)$) is negligible for any RPT adversary. □

We will also use weaker definitions of security for symmetric encryption schemes, 1-IND-CPA and 1-IND-CCA. The only difference with the above standard definitions is that an adversary is not given the encryption oracles.

Obviously, any symmetric encryption scheme that is IND-CPA secure (resp., IND-CCA secure) is also 1-IND-CPA secure (resp., 1-IND-CCA secure). We remark that the latter weak definition of security is called Find-Guess (FG) security definition in [25].

B. Symmetric-Key MRESs

We now consider MRESs in the symmetric-key setting. The syntax for such schemes $\overline{\mathsf{SE}} = (\mathcal{SK}, \overline{\mathcal{SE}}, \mathcal{SD})$ can be defined similarly to the syntax of asymmetric MRESs defined in Section II-B. The only difference is that in the symmetric-key case we do not consider a common-key generation algorithm and instead of a public/secret key pairs there are symmetric keys.

Again, we are interested in RR-MRESs. We can define them in a symmetric-key setting similarly to Definition 3.1 for a public-key setting. The only changes are as mentioned above.

Security. Unlike the public-key environment, in the symmetric-key setting the possibility of a common randomness being learned by a receiver after performing decryption is not a threat since it cannot help a user to get any information about non-legitimate messages. Moreover, for many symmetric encryption schemes the random string used in an encryption

algorithm is often public and a part of a ciphertext. Nevertheless, we still allow the model to consider insider attacks. The reason is that it is reasonable to assume that secret keys could be chosen by users and are not always random and independent. The definition is analogous to the one for asymmetric setting, but now the adversary is not asked to output random coins for key generation and is given an encryption oracle which takes as input a message vector and outputs a ciphertext vector.

Definition 9.2: Let $\overline{\mathsf{SE}} = (\mathcal{SK}, \overline{\mathcal{SE}}, \mathcal{SD})$ be a symmetric-key MRES. Let B be an adversary. B has access to an oracle which takes a vector. For $b \in \{0,1\}, \mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ and a polynomial n define the experiments:

Experiment $\mathbf{Exp}^{\text{mr-atk-}b}_{\overline{\mathsf{SE}},B,n(\cdot)}(k)$

$(1^l, \boldsymbol{sk'}, \mathrm{st}) \xleftarrow{\$} B(\mathrm{select}, k, n(\cdot))$

$[1 \le l \le n(k); |\boldsymbol{sk'}| = n(k) - l]$

For $i = 1, \ldots, l$ do $\boldsymbol{sk}[i] \xleftarrow{\$} \mathcal{K}(1^k)$ EndFor

$\boldsymbol{sk} \leftarrow \boldsymbol{sk} || \boldsymbol{sk'}$

$(\boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathrm{st}) \xleftarrow{\$} B^{\overline{\mathcal{SE}}_{\boldsymbol{sk}}(\cdot), \mathcal{O}_1(\cdot), \ldots, \mathcal{O}_l(\cdot)}(\mathrm{find}, \mathrm{st})$

$[|\boldsymbol{M}_0| = |\boldsymbol{M}_1| = l; |\boldsymbol{M}| = n(k) - l]$

$\boldsymbol{M} \leftarrow (\boldsymbol{M}_b[1], \ldots, \boldsymbol{M}_b[l], \boldsymbol{M}[1], \ldots, \boldsymbol{M}[n(k) - l])$

$\boldsymbol{C} \xleftarrow{\$} \overline{\mathcal{SE}}_{\boldsymbol{sk}}(\boldsymbol{M})$

$d \xleftarrow{\$} B^{\overline{\mathcal{SE}}_{\boldsymbol{sk}}(\cdot), \mathcal{O}_1(\cdot), \ldots, \mathcal{O}_l(\cdot)}(\mathrm{guess}, \boldsymbol{C}, \mathrm{st})$

Return d

Above, the oracles for $1 \le i \le l$ are defined as follows: If $\mathrm{atk} = \mathrm{cpa}$ then $\mathcal{O}_i(\cdot) = \varepsilon$ and if $\mathrm{atk} = \mathrm{cca}$ then $\mathcal{O}_i(\cdot) = \mathcal{D}_{\boldsymbol{sk}[i]}(\cdot)$. It is required that $|M_0[i]| = |M_1[i]|$, and are in $\mathrm{MsgSp}(k)$ for all $1 \le i \le n(k)$. If $\mathrm{atk} = \mathrm{cca}$, then the adversary B does not query $\mathcal{O}_i(\cdot)$ on $\boldsymbol{C}[i]$. We define the advantage $\mathbf{Adv}^{\text{mr-atk}}_{\overline{\mathsf{SE}},B}(\ )$ of the adversary, IND-CPA and IND-CCA security of the symmetric MRES analogously to the definitions for the asymmetric case described in Section IV. □

REPRODUCIBILITY OF SYMMETRIC-KEY ENCRYPTION SCHEMES. The definition of reproducible schemes defined in Definition 6.1 can be naturally lifted for the symmetric-key setting.

Definition 9.3: Fix a symmetric-key encryption scheme $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$. Let R be an algorithm that takes as input a ciphertext of a random message, another random message, and a secret key, and returns a ciphertext. Consider the following experiment.

Experiment $\mathbf{Exp}^{\text{repr}}_{\mathcal{SE},R}(k)$

$sk \xleftarrow{\$} \mathcal{SK}(1^k); M \xleftarrow{\$} \mathrm{MsgSp}(k)$

$r \xleftarrow{\$} \mathrm{Coins}_{\mathcal{SE}}(k); C \xleftarrow{\$} \mathcal{SE}_{sk}(M, r)$

$sk' \xleftarrow{\$} \mathcal{SK}(1^k); M' \xleftarrow{\$} \mathrm{MsgSp}(k)$

If $\mathcal{SE}_{sk'}(M', r) = R(C, M', sk')$

Then return 1 else return 0 EndIf

We say that $\mathcal{SE}$ is reproducible if for any k there exists an RPTA R such that $\mathbf{Exp}^{\text{repr}}_{\mathcal{SE},R}(k)$ outputs 1 with probability 1.□

The analog of Theorem 6.2 also holds for a symmetric-key setting. It implies that if $\mathcal{SE}$ is reproducible and IND-CPA then it is also RR-IND-CPA.

Theorem 9.4: Fix a symmetric-key encryption scheme $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$. Let $\overline{\mathsf{SE}} = (\mathcal{SK}, \overline{\mathcal{SE}}, \mathcal{SD})$ be the corresponding RR-MRES. For $\mathrm{atk} = \{\mathrm{cpa}, \mathrm{cca}\}$, if $\mathcal{SE}$ is reproducible then for any RPTA B, there exists an RPTA A, such that

$$\mathbf{Adv}^{\text{mr-atk}}_{\overline{\mathsf{SE}},B,n(\cdot)}(k) \le n(k)\mathbf{Adv}^{\text{atk}}_{\mathcal{SE},A}(k). \qquad \square$$

The proof follows the proof of Theorem 6.2, presenting the adversary A which tries to break a symmetric encryption scheme and uses the adversary B which attacks the associated symmetric key RR-MRES. The main difference is that in this case A has to answer B's encryption oracle queries. The problem is that A does not know one secret key corresponding to its own challenge. But A has access to an encryption oracle corresponding to this key. So it can query this oracle and then use the reproduction algorithm to get the rest of the ciphertexts to form a ciphertext vector as an answer to B's query. The rest of the proof in analogous.

CBC-BASED MRES. We recall the CBC encryption scheme. The message space is a set of all strings whose length is a multiple of s bits. The scheme uses a family of permutations $F : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$. F^{-1} denotes the inverse permutation. The key-generation algorithm of $\mathcal{CBC}[F] = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ simply outputs a random k-bit string sk, which specifies the permutation $F(sk, \cdot)$ with a domain and range $\{0,1\}^s$. Usuall,y F is a block cipher such as AES. The encryption and decryption algorithms are defined as follows:

$\underline{\mathcal{SE}_{sk}(M)}$

Parse M as $M_1, \ldots, M_p$,

[s.t. $|M_i| = s$ for $1 \le i \le p$]

$C_0 \xleftarrow{\$} \{0,1\}^s$

For $i = 1, \ldots, p$ do

$C_i \leftarrow F(sk, C_{i-1} \oplus M_i)$

EndFor

Return $C_0 || C_1 || \ldots || C_p$

$\underline{\mathcal{SD}_{sk}(C)}$

Parse C as $C_0, \ldots, C_p$,

[s.t. $|C_i| = s$ for $0 \le i \le p$]

For $i = 1, \ldots, p$ do

$M_i \leftarrow F^{-1}(sk, C_i) \oplus C_{i-1}$

EndFor

$M \leftarrow M_1 || \ldots || M_p$

Return M

C_0 is often called the initial vector (IV).

Lemma 9.5: CBC encryption scheme $\mathcal{CBC}[F] = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ is reproducible for any F.

Proof: An RPTA R takes as input $R(C_0\|C_1\|\ldots\|C_p, M', sk')$ and returns $C' = \mathcal{SE}_{sk'}(M', C_0)$. It is easy to see that R always outputs a valid ciphertext which is created using the same random string C_0 as a given ciphertext and therefore $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{CBC}[F],R}(k)$ always output 1. □

The result of [6] states that if F is a pseudorandom function family then $\mathcal{CBC}[F]$ is IND-CPA. It follows from this result and form the reproduction theorem and Lemma 9.5 that $\mathcal{CBC}[F]$ is RR-IND-CPA.

X. Secure Hybrid SM-MRES

Construction 10.1: Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be an asymmetric MRES and let $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ be a symmetric encryption scheme. The single-message multirecipient hybrid encryption scheme $\mathcal{HS} = (\mathcal{G}, \mathcal{K}, \mathcal{HE}, \mathcal{HD})$ is an asymmetric SM-MRES encryption scheme and its common key generation and key generation algorithms are the same as those of $\overline{\mathcal{AE}}$. The rest of algorithms are as follows.

$\underline{\mathcal{HE}_{I,pk}(\boldsymbol{M})}$
$K \xleftarrow{\$} \mathcal{SK}(1^k)$
For $i = 1, \ldots, n(k)$ do
$\boldsymbol{K}[i] \leftarrow K$ EndFor
$\boldsymbol{C}' \xleftarrow{\$} \bar{\mathcal{E}}_{\boldsymbol{pk}}(I, \boldsymbol{K})$
$C'' \xleftarrow{\$} \mathcal{SE}_{\boldsymbol{K}[1]}(\boldsymbol{M}[1])$
For $i = 1, \ldots, n(k)$ do
$\boldsymbol{C}[i] \leftarrow \boldsymbol{C}'[i]\|C''$ EndFor
Return $\boldsymbol{C}$

$\underline{\mathcal{HD}_{I,sk}(C)}$
Parse C as $C'\|C''$
$K \leftarrow \mathcal{D}_{I,sk}(C')$
$M \leftarrow \mathcal{SD}_K(C'')$
Return M

Note that the second part of $\boldsymbol{C}[i]$ for all $1 \le i \le n(k)$ is the same and can be sent only once thus permitting bandwidth savings. The following theorem states that the above SM-MRES is secure given that $\overline{\mathcal{AE}}$ and $\mathcal{SE}$ meet the corresponding notions of security.

Theorem 10.2: Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \bar{\mathcal{E}}, \mathcal{D})$ be an asymmetric MRES and let $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ be a symmetric encryption scheme. Let $\mathcal{HS} = (\mathcal{G}, \mathcal{K}, \mathcal{HE}, \mathcal{HD})$ be an SM-MRES constructed as per Construction 10.1. Then for any RPTA A there exist RPTAs B, C such that for $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$

$$\mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{HS},A,n(\cdot)}(k) \le 2\mathbf{Adv}^{\text{smmr-atk}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) + \mathbf{Adv}^{\text{1-atk}}_{\mathcal{SE},C}(k). \quad \square$$

The proof is in Appendix C.

Appendix A
Proof of Theorem 6.2

We first consider the case of chosen-plaintext attacks only and then indicate how to extend the argument to the case of chosen-ciphertext attacks. Let B be an adversary attacking the RR-MRES $\overline{\mathcal{AE}}$. We will design an adversary A attacking the scheme $\mathcal{AE}$ so that

$$\mathbf{Adv}^{\mathrm{cpa}}_{\mathcal{AE},A}(k) \ge \frac{1}{n(k)}\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k).$$

This implies the statement of the Theorem 6.2 for $\mathrm{atk} = \mathrm{cpa}$. We begin by describing some hybrid experiments associated to B and $\overline{\mathcal{AE}}$. It is convenient to parameterize the hybrids via an integer j, where j is ranging from 0 to $n(k)$.

Experiment $\mathbf{Exp}\,\boldsymbol{H}_j(k)$ $[0 \le j \le n(k)]$
$I \xleftarrow{\$} \mathcal{G}(1^k); (1^l, \mathrm{st}) \xleftarrow{\$} B(\mathrm{select}, n(k), I)$
For $i = 1, \ldots, l$ do $(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor
$(\boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathbf{coins}, \mathrm{st}) \xleftarrow{\$} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\mathrm{find}, \boldsymbol{pk}, \mathrm{st})$
For $i = l+1, \ldots, n(k)$ do
$(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I, \mathbf{coins}_{\mathcal{K}}(I)[i])$ EndFor
$\boldsymbol{pk} \leftarrow (\boldsymbol{pk}[1], \ldots, \boldsymbol{pk}[n(k)])$
If $j \le l$ then $\boldsymbol{M} \leftarrow (\boldsymbol{M}_0[1], \ldots, \boldsymbol{M}_0[j], \boldsymbol{M}_1[j+1], \ldots, \boldsymbol{M}_1[l], \boldsymbol{M}[1], \ldots, \boldsymbol{M}[n(k)-l])$
else $\boldsymbol{M} \leftarrow (\boldsymbol{M}_1[0], \ldots, \boldsymbol{M}_0[l], \boldsymbol{M}[1], \ldots, \boldsymbol{M}[n(k)-l])$
EndIf
$\boldsymbol{C} \xleftarrow{\$} \bar{\mathcal{E}}_{I,\boldsymbol{pk}}(\boldsymbol{M})$
$d \xleftarrow{\$} B(\mathrm{guess}, \boldsymbol{C}, \mathrm{st})$
Return d

Let $P_j \stackrel{\mathrm{def}}{=} \Pr[\mathbf{ExpH}_j(k) = 0]$ for $j = 0, 1, \ldots, n(k)$. Now we claim that

$$\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k) = P_{n(k)} - P_0. \tag{1}$$

This is justified as follows. We claim that

$$\Pr\left[\mathbf{Exp}^{\text{mr-cpa-0}}_{\overline{\mathcal{AE}},B,n(\cdot)}(1^k) = 0\right] = P_{n(k)}$$

and

$$\Pr\left[\mathbf{Exp}^{\text{mr-cpa-1}}_{\overline{\mathcal{AE}},B,n(\cdot)}(1^k) = 0\right] = P_0$$

and after subtraction (1) follows. We now justify the two equations above. In experiment $\mathbf{Exp}\,\boldsymbol{H}_{n(k)}(k)$ we have $j = n(k)$ and a challenge ciphertext C is computed by encrypting the "left" vector of messages $\boldsymbol{M}_0$ under l different public keys plus the encryptions of the rest $n(k) - l$ messages, so that the B's "view" is the same as in experiment $\mathbf{Exp}^{\text{mr-cpa-0}}_{\overline{\mathcal{AE}},B,n(\cdot)}(1^k)$. On the other hand, in experiment $\mathbf{Exp}\,\boldsymbol{H}_0(k)$ we have $j = 0$, and a challenge ciphertext C consists of l encryptions of messages from a "right" vector of messages under l different public keys, plus the encryptions of the rest $n(k) - l$ messages, so that B's "view" is the same as in experiment $\mathbf{Exp}^{\text{mr-cpa-1}}_{\overline{\mathcal{AE}},B,n(\cdot)}(1^k)$.

Now we turn to the description of A.

Adversary $A(\mathrm{find}, I, pk)$
$(1^l, \mathrm{st}') \xleftarrow{\$} B(\mathrm{select}, n(k), I); j \xleftarrow{\$} \{1, \ldots, n(k)\}$
If $j \le l$ then For $i \in \{1, \ldots, j-1, j+1, \ldots, l\}$ do
$(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I); \boldsymbol{pk}[j] \leftarrow pk$ EndFor
else For $i = 1, \ldots l$ do $(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor
EndIf
$(\boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathbf{coins}, \mathrm{st}') \xleftarrow{\$} B(\mathrm{find}, I, \boldsymbol{pk}, \mathrm{st}')$

For $i = l+1, \ldots, n(k)$ do

$(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I, \mathbf{coins}[i])$ EndFor

If $j > l$ then $\boldsymbol{M}_0[j] \leftarrow \boldsymbol{M}[j]; \boldsymbol{M}_1[j] \leftarrow \boldsymbol{M}[j]$ EndIf

$\mathrm{st} \leftarrow (j, l, \boldsymbol{pk}, \boldsymbol{sk}, \boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathrm{st})$

Return $(\boldsymbol{M}_0[j], \boldsymbol{M}_1[j], \mathrm{st})$

Adversary $A(\mathrm{guess}, C, \mathrm{st})$

For $i \in \{1, \ldots, j-1, j+1, \ldots, n(k)\}$ do

If $i \geq l+1$ then $M' \leftarrow \boldsymbol{M}[i]$ EndIf
If $i \leq j$ then $M' \leftarrow \boldsymbol{M}_0[i]$ else $M' \leftarrow \boldsymbol{M}_1[i]$ EndIf

$\boldsymbol{C}[i] \leftarrow R(I, pk, C, M', \boldsymbol{pk}[i], \boldsymbol{sk}[i])$

EndFor

$\boldsymbol{C}' \leftarrow (\boldsymbol{C}[1], \ldots, \boldsymbol{C}[j-1], C, \boldsymbol{C}[j+1], \ldots, \boldsymbol{C}[n(k)])$

$d \xleftarrow{\$} B(\mathrm{guess}, \boldsymbol{C}', \mathrm{st}')$

Return d

We claim that

$$\Pr\left[\mathbf{Exp}^{\text{cpa-0}}_{\mathcal{AE},A}(k) = 0\right] = \frac{1}{n(k)} \cdot \sum_{j=1}^{n(k)} P_j \tag{2}$$

and

$$\Pr\left[\mathbf{Exp}^{\text{cpa-1}}_{\mathcal{AE},A}(k) = 0\right] = \frac{1}{n(k)} \cdot \sum_{j=1}^{n(k)} P_{j-1}. \tag{3}$$

Subtracting and exploiting the collapse of the sums we get

$$\begin{aligned}\mathbf{Adv}^{\text{cpa}}_{\mathcal{AE},A}(k) &= \frac{1}{n(k)} \sum_{j=1}^{n(k)} P_j - P_{j-1} \\ &= \frac{1}{n(k)}[P_{n(k)} - P_0] \\ &= \frac{1}{n(k)} \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(k).\end{aligned}$$

The statement of the theorem follows, so it remains to justify (2), (3). Each value of j in $\{1, \ldots, n(k)\}$ is equally likely for A. The j's ciphertext in B's challenge ciphertext vector is an A's challenge ciphertext. Reproducibility of $\mathcal{AE}$ guarantees that all $n(k)$ ciphertexts in a challenge ciphertext are computed using the same random string. It is easy to see that the experiment $\mathbf{Exp}^{\text{cpa-0}}_{\mathcal{AE},A}(k)$ is the same as $\mathbf{Exp}\,\boldsymbol{H}_j(k)$. Similarly, the experiment $\mathbf{Exp}^{\text{cpa-1}}_{\mathcal{AE},A}(k)$ is the same as $\mathbf{Exp}\,\boldsymbol{H}_{j-1}(k)$.

The running time of A is one of B plus one of R plus the time to pick a number $j \leq n(k)$ at random.

We provide a sketch of how to extend the proof to the case of chosen-ciphertext attacks. The definition of the hybrid experiments is the same with regard to how the inputs to B are computed. Decryption queries are however answered truthfully, using the correct secret key. The adversary A is given also the decryption oracle $\mathcal{D}_{I,sk}(\cdot)$ where sk is the secret key corresponding to its input public key pk. It proceeds as before. The novel elements is to provide answers to decryption oracle queries. When the query is to $\mathcal{D}_{I,sk_i}(\cdot)$ for $1 \leq i \leq l, i \neq j$, algorithm A can easily provide the answer since it is in possession of sk_i. When $i = j$, it provides the answer by invoking its own given decryption oracle. The analysis proceeds as before. □

Appendix B
Proof of Theorem 8.3

We prove that for any RPTA A_{atk}, there exist an RPTA adversary B_{atk}, where $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$ and an RPT adversary D, such that for any $k \in \mathbb{N}$

$$\begin{aligned}&\mathbf{Adv}^{\text{mr-atk}}_{\overline{\mathcal{AE}'[F]},A_{\mathrm{atk}},n(\cdot)}(k) \\ &\qquad \leq n(k)\mathbf{Adv}^{\mathrm{atk}}_{\mathcal{AE},B_{\mathrm{atk}}}(k) + 2 \cdot \mathbf{Adv}^{\mathrm{prf}}_{F,D}(k).\end{aligned}$$

The statement of Theorem 8.3 is implied by this result. We first prove it for the case of chosen-plaintext attacks and then show how the proof can be extended for the case of chosen-ciphertext attacks. Let R be a family of all functions of $\{0,1\}^{il} \to \{0,1\}^{ol}$. Let A be an RPTA adversary attacking the security of the multirecipient scheme $\overline{\mathcal{AE}'[F]}$. We will construct an RPT adversary D which attacks F as a pseudorandom function family and an adversary B which attacks the security of $\mathcal{AE}$ such that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{F,D}(k) = \frac{1}{2} \cdot \Big(&\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[F]},A,n(\cdot)}(k) \\ &- \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k)\Big)\end{aligned} \tag{4}$$

$$\mathbf{Adv}^{\text{cpa}}_{\mathcal{AE},B,n(\cdot)}(k) \geq \frac{1}{n(k)} \cdot \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k) \tag{5}$$

where $\overline{\mathcal{AE}'[R]}$ denotes the encryption scheme which uses a random function in place of the random instance of the pseudorandom function family. This implies the statement of the theorem. It remains to specify the strategies of D and B. The adversary D takes k and has access to an oracle $g : \{0,1\}^{il} \to \{0,1\}^{ol}$. Here is the algorithm for D.

Adversary $D^{g(\cdot)}(1^k)$

$b \xleftarrow{\$} \{0,1\}$

$I \xleftarrow{\$} \mathcal{G}(1^k); (1^l, \mathrm{st}) \xleftarrow{\$} A(\mathrm{select}, n(k), I)$
$[1 \leq l \leq n(k)]$

For $i = 1, \ldots, l$ do $(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor

$(\boldsymbol{M}_0, \boldsymbol{M}_1, \boldsymbol{M}, \mathbf{coins}, \mathrm{st}) \xleftarrow{\$} A(\mathrm{find}, \boldsymbol{pk}, \mathrm{st})$

$[|\boldsymbol{M}_0| = |\boldsymbol{M}_1| = l; |\boldsymbol{M}| = n(k) - l]$

$[|\boldsymbol{pk}| = l; |\mathbf{coins}| = n(k) - l]$

For $i = l+1, \ldots, n(k)$ do

$(\boldsymbol{pk}'[i], \boldsymbol{sk}'[i]) \xleftarrow{\$} \mathcal{K}(I, \mathbf{coins}[i])$ EndFor

$\boldsymbol{pk} \leftarrow (\boldsymbol{pk}[1], \ldots, \boldsymbol{pk}[l], \boldsymbol{pk}'[l+1], \ldots, \boldsymbol{pk}'[n(k)])$

$\boldsymbol{M} \leftarrow (\boldsymbol{M}_b[1], \ldots, \boldsymbol{M}_b[l], \boldsymbol{M}[1], \ldots, \boldsymbol{M}[n(k) - l])$

$\boldsymbol{C} \xleftarrow{\$} \bar{\mathcal{E}}'^{g(\cdot)}_{\boldsymbol{pk}}(\boldsymbol{M})$

$d \xleftarrow{\$} A(\mathrm{guess}, \boldsymbol{C}, \mathrm{st})$

If $b = d$ then return 1 else return 0

In the preceding algorithm, $\bar{\mathcal{E}}'^{g(\cdot)}_{\boldsymbol{pk}}$ denotes the procedure which substitutes all applications of $F(r',\cdot)$ in $\bar{\mathcal{E}}'_{\boldsymbol{pk}}(\cdot)$with an application of $g(\cdot)$.

We now analyze the adversary. We claim that

$$\Pr\left[D^g(k)=1 : g \xleftarrow{\$} F\right] = \frac{1}{2}+\frac{1}{2}\cdot \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[F]},A,n(\cdot)}(k)$$
$$\Pr\left[D^g(k)=1 : g \xleftarrow{\$} R\right] = \frac{1}{2}+\frac{1}{2}\cdot \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k).$$

The preceding equations are justified as follows. If g is an instance of F then A's view in the simulated experiment is indistinguishable from its view in $\mathbf{Exp}^{\text{mr-cpa-}b}_{\overline{\mathcal{AE}'[F]},A,n(\cdot)}(k)$. This is true since in the real experiment the challenge ciphertext vector for A's guess stage is computed using an instance of the function family F specified by the key, which is the random string used by the encryption algorithm. In the simulated experiment, D uses its oracle which is also a random instance of the function family F. Similarly, if g is an instance of R then A's view in the simulated experiment is indistinguishable from its view in $\mathbf{Exp}^{\text{mr-cpa-}b}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k)$. After subtraction we get (4).

We now prove (5). Let A be an adversary which attacks the security of $\overline{\mathcal{AE}'[R]}$. We will use the hybrid experiments $\mathbf{Exp}\,\boldsymbol{H}_j(k)$ for $0 \le j \le n(k)$ we defined in the proof of Theorem 6.2, which are associated to A and the encryption scheme $\overline{\mathcal{AE}'[R]}$. Let $P_j \stackrel{\text{def}}{=} \Pr[\mathbf{Exp}\,\boldsymbol{H}_j(k)=0]$ for $j=0,1,\ldots,n(k)$. Similarly to the proof of Theorem 6.2 we claim that

$$\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k) = P_{n(k)} - P_0. \tag{6}$$

We now present the adversary B which attacks the security of $\mathcal{AE}$. It will use A. Here is the code for B:

Adversary $B(\text{find}, I, pk)$

$(l,\text{st}') \xleftarrow{\$} A(\text{select}, n(k), I)$; $j \xleftarrow{\$} \{1,\ldots,n(k)\}$

If $j \le l$ then For $i \in \{1,\ldots,j-1,j+1,\ldots,l\}$ do

$(\boldsymbol{pk}[i],\boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$; $\boldsymbol{pk}[j] \leftarrow pk$ EndFor

else For $i=1,\ldots l$ do

$(\boldsymbol{pk}[i],\boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor

EndIf

$(\boldsymbol{M}_0,\boldsymbol{M}_1,\boldsymbol{M},\mathbf{coins},\text{st}) \xleftarrow{\$} A(\text{find},\boldsymbol{pk},\text{st})$

For $i=l+1,\ldots,n(k)$ do

$(\boldsymbol{pk}[i],\boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I,\mathbf{coins}[i])$

$\boldsymbol{pk} \leftarrow (\boldsymbol{pk}[1],\ldots,\boldsymbol{pk}[l],\boldsymbol{pk}[l+1],\ldots,\boldsymbol{pk}[n(k)])$

$\boldsymbol{M}_0[i] \leftarrow \boldsymbol{M}[i]$; $\boldsymbol{M}_1[i] \leftarrow \boldsymbol{M}[i]$ EndFor

$\text{st} \leftarrow (I,j,l;\boldsymbol{pk},\boldsymbol{sk},\boldsymbol{M}_0,\boldsymbol{M}_1;\text{st}')$

Return $(\boldsymbol{M}_0[j],\boldsymbol{M}_1[j],\text{st})$

Adversary $B(\text{guess}, C, \text{st})$

For $i \in \{1,\ldots,j-1,j+1,\ldots,n(k)\}$ do

If $\boldsymbol{pk}[i]=pk$ then $M \leftarrow \mathcal{D}_{I,sk_i}(C)$

If $M=\boldsymbol{M}_0[j]$ then Return 0 else Return 1

Else

If $\exists p: 1 \le p < i, \boldsymbol{pk}[p]=\boldsymbol{pk}[i]$ then $r_i \leftarrow r_p$

Else $r_i \xleftarrow{\$} \text{Coins}_{\mathcal{E}}(I)$ EndIf

EndIf

EndFor

For $i=1,\ldots,j-1$ do $\boldsymbol{C}[i] \leftarrow \mathcal{E}_{I,\boldsymbol{pk}[i]}(\boldsymbol{M}_0[i],r_i)$

For $i=j+1,\ldots,n(k)$ do $\boldsymbol{C}[i] \leftarrow \mathcal{E}_{\boldsymbol{pk}[i]}(\boldsymbol{M}_{1[i]},r_i)$

$C_j \leftarrow C$; $d \xleftarrow{\$} A(\text{guess},\boldsymbol{C},\text{st}')$

Return d

We now analyze the adversary B. All values of j in $\{1,\ldots n(k)\}$ are equally likely for B, so we focus on one particular value of j. If all the public keys created by B and those which are output by A are different from B's "challenge" public key pk, then we claim that the view of A in the experiment simulated by B is indistinguishable from A's view in the experiment $\mathbf{Exp}\,\boldsymbol{H}_j(k)$. This is true since the only potential difference among these experiments from A's point of view is how the values r_i used as coin tosses for $\mathcal{E}_{I,pk_i}$ are computed. In the experiment $\mathbf{Exp}\,\boldsymbol{H}_j(k)$, the values r_i are computed as the output of a random function and B computes r_i by dynamically simulating a random function.

If at least one of the public keys created by B or one of those which are output by A happens to be the same as B's "challenge" public key pk, then A's view in the simulated experiment is different from its view in the experiment $\mathbf{Exp}\,\boldsymbol{H}_j(k)$, since for them to be the same B should compute the component of $\boldsymbol{C}$ corresponding to this public key using the same randomness as was used to compute its own challenge ciphertext C (since this randomness is the output of the random function invoked on the same inputs), but B has no way of learning this randomness. However, in this case B learns the challenge secret key and can always win its game by decrypting the challenge ciphertext. Thus, we claim that

$$\Pr\left[\mathbf{Exp}^{\text{cpa-0}}_{\mathcal{AE},B}(1^k)=0\right] \ge \frac{1}{n(k)}\sum_{j=1}^{n(k)} P_j$$

and

$$\Pr\left[\mathbf{Exp}^{\text{cpa-1}}_{\mathcal{AE},B}(1^k)=0\right] \le \frac{1}{n(k)}\sum_{j=1}^{n(k)} P_{j-1}.$$

Subtracting and exploiting the collapse of the sums we get

$$\begin{aligned}\mathbf{Adv}^{\text{cpa}}_{\mathcal{AE},A}(k) &\ge \frac{1}{n}\sum_{j=1}^{n(k)}[P_j - P_{j-1}]\\ &= \frac{1}{n(k)}[P_{n(k)}-P_0]\\ &= \frac{1}{n(k)}\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k).\end{aligned}$$

The above implies (5).

We now sketch out how to extend the proof to the case of chosen-ciphertext attacks. Both D and B now have to answer A's decryption oracle queries, which can be made to $\mathcal{D}_{sk_i}$ for $1 \le i \le l$. D can easily do so since it possesses all the secret keys $sk_1, \ldots, sk_l$. B knows all but one secret key, it does not know sk_j but it has access to a decryption oracle which corresponds to this key. When A makes a query to $\mathcal{D}_{sk_j}$, B provides an answer by invoking its own decryption oracle. The definition of hybrid experiments remains the same, except that A can ask decryption oracle queries, which are answered truthfully, using the correct secret key. The rest of the analysis is as before.

It remains to specify running times of D and B. The running time of B is that of A plus the time to pick a number $j \le n(k)$ at random. The running time of D is one of A. □

Appendix C
Proof of Theorem 10.2

Let A_{atk} be an adversary attacking SM-MRES $\mathcal{HS}$. We first define the following experiment:

Experiment $\mathbf{Exp}\,\boldsymbol{H}^{m\text{-atk}}_{\mathcal{HS},A_{\mathrm{atk}}}(k)$

$[m \in \{1,2,3,4\}; \mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}]$

$I \xleftarrow{\$} \mathcal{G}(1^k)$

For $i = 1, \ldots, n(k)$ do $(\boldsymbol{pk}[i], \boldsymbol{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor

$(1^l, \mathrm{st}) \xleftarrow{\$} A_{\mathrm{atk}}(\mathrm{select}, n(k), I)$;

If $l \ne n(k)$ then abort EndIf

$(\boldsymbol{M}_0, \boldsymbol{M}_1, \mathrm{st}) \xleftarrow{\$} A_{\mathrm{atk}}(\mathrm{find}, \boldsymbol{pk}, \mathrm{st})$

If $\exists 1 \le i, j \le n(\cdot)$ such that

$\boldsymbol{M}_0[i] \ne \boldsymbol{M}_0[j]$ or $\boldsymbol{M}_1[i] \ne \boldsymbol{M}_1[j]$ then abort EndIf

$K \xleftarrow{\$} \mathcal{SK}(k)$; $K' \xleftarrow{\$} \mathcal{SK}(k)$

If $l = 1$ or $l = 2$ then $C_1 \xleftarrow{\$} \mathcal{SE}_K(\boldsymbol{M}_0[1])$ EndIf

If $l = 3$ or $l = 4$ then $C_1 \xleftarrow{\$} \mathcal{SE}_K(\boldsymbol{M}_1[1])$ EndIf

For $i = 1, \ldots, n(k)$ do

If $m = 1$ or $m = 4$ then $\boldsymbol{C}_0 \xleftarrow{\$} \bar{\mathcal{E}}_{I,\boldsymbol{pk}}(K)$ EndIf

If $m = 2$ or $m = 3$ then $\boldsymbol{C}_0 \xleftarrow{\$} \bar{\mathcal{E}}_{I,\boldsymbol{pk}}(K')$ EndIf

$\boldsymbol{C} \leftarrow \boldsymbol{C}_0[i] || C_1$

EndFor

If $\mathrm{atk} = \mathrm{cca}$ and A_{cca} during find stage makes

a decryption oracle query $\boldsymbol{C}'$ to oracle $\mathcal{HD}_{I,sk_i}(\cdot)$

$M \leftarrow \mathcal{HD}_{I,sk}(\boldsymbol{C}'[1])$ EndIf

return M to A_{cca}

EndIf

$d \xleftarrow{\$} A_{\mathrm{atk}}(\mathrm{guess}, \boldsymbol{C}, \mathrm{st})$

If $\mathrm{atk} = \mathrm{cca}$ and A_{cca} during guess stage makes

a decryption oracle query $\boldsymbol{C}'$ to oracle $\mathcal{HD}_{I,sk_i}(\cdot)$:

If $m = 1$ or $m = 4$ then $M \leftarrow \mathcal{HD}_{I,sk}(\boldsymbol{C}'[1])$ EndIf

If $m = 2$ or $m = 3$ then parse $\boldsymbol{C}'[1]$ as $C'_0 || C'_1$

If $C'_0 = \boldsymbol{C}_0[1]$ then $M \leftarrow \mathcal{SD}_K(C'_1)$

else $M \leftarrow \mathcal{HD}_{I,sk}(\boldsymbol{C}'[1])$ EndIf

EndIf

return M to A_{cca}

EndIf

Return d

Let $P^{\mathrm{atk}}_m \stackrel{\mathrm{def}}{=} \Pr[\mathbf{Exp}\,\boldsymbol{H}^{m\text{-atk}}_{\mathcal{HS},A_{\mathrm{atk}}}(k) = 0]$ for $m \in \{1,2,3,4\}$. It is not difficult to see that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{smmr\text{-}atk}}_{\overline{\mathcal{HS}},A,n(\cdot)}(k) &= P^{\mathrm{atk}}_4 - P^{\mathrm{atk}}_1 \\ &= \left(P^{\mathrm{atk}}_4 - P^{\mathrm{atk}}_3\right) \\ &\quad + \left(P^{\mathrm{atk}}_3 - P^{\mathrm{atk}}_2\right) + \left(P^{\mathrm{atk}}_2 - P^{\mathrm{atk}}_1\right). \quad (7)\end{aligned}$$

We now claim the following.

Claim C.1: For any $k \in \mathbb{N}$ there exists an RPTA B_1 such that

$$P^{\mathrm{atk}}_4 - P^{\mathrm{atk}}_3 \le \mathbf{Adv}^{\mathrm{smmr\text{-}atk}}_{\overline{\mathcal{AE}},B_1,n(\cdot)}(k).$$

Claim C.2: For any $k \in \mathbb{N}$ there exists an RPTA C such that

$$P^{\mathrm{atk}}_3 - P^{\mathrm{atk}}_2 \le \mathbf{Adv}^{\mathrm{1\text{-}atk}}_{\mathcal{SE},C}(k).$$

Claim C.3: For any $k \in \mathbb{N}$ there exists an RPTA B_2 such that

$$P^{\mathrm{atk}}_2 - P^{\mathrm{atk}}_1 \le \mathbf{Adv}^{\mathrm{smmr\text{-}atk}}_{\overline{\mathcal{AE}},B_2,n(\cdot)}(k).$$

For a fixed $k \in \mathbb{N}$, if

$$\mathbf{Adv}^{\mathrm{smmr\text{-}atk}}_{\overline{\mathcal{AE}},B_1,n(\cdot)}(k) \ge \mathbf{Adv}^{\mathrm{smmr\text{-}atk}}_{\overline{\mathcal{AE}},B_2,n(\cdot)}(k)$$

then define adversary $B = B_1$ and $B = B_2$ otherwise. Then the statement of the theorem follows from (7) and Claims C.1, C.2, and C.3. It remains to prove the latter claims. □

Claim C.1: We consider a more general case of chosen-ciphertext attacks and then specify the changes pertaining to the case of chosen-plaintext attacks. We present a pseudocode for adversary B_1 in Fig. 1.

Adversary $B_1(\text{select}, n(\cdot), I)$
$(1^l, st') \xleftarrow{\$} A_{\text{cca}}(\text{select}, n(\cdot), I)$
If $l \neq n(k)$ then abort EndIf
Return $(n(k), st')$
Adversary $B_1^{\mathcal{D}_{I,\mathbf{sk}[1]}(\cdot),\ldots,\mathcal{D}_{I,\mathbf{sk}[n(k)]}(\cdot)}(\text{find}, \mathbf{pk}, st)$
$K \xleftarrow{\$} \mathcal{SK}(k)$; $K' \xleftarrow{\$} \mathcal{SK}(k)$
Run $A_{\text{cca}}^{\mathcal{HD}_{I,\mathbf{sk}[1]}(\cdot)\ldots\mathcal{HD}_{I,\mathbf{sk}[n(k)]}(\cdot)}(\text{find}, \mathbf{pk}, st')$
When A_{cca} makes a query C'
to its decryption oracle $\mathcal{HD}_{I,\mathbf{sk}[i]}(\cdot)$ [$1 \leq i \leq n(k)$]
parse C' as $C'_0 \| C'_1$
$K'' \leftarrow \mathcal{D}_{I,\mathbf{sk}[i]}(C'_0)$; $M \leftarrow \mathcal{SD}_{K''}(C'_1)$
Return M to A_{cca}
Until A_{cca} outputs $(\mathbf{M}_0, \mathbf{M}_1, st')$
If $\exists\ 1 \leq i, j \leq n(\cdot)$ such that $\mathbf{M}_0[i] \neq \mathbf{M}_0[j]$ or
$\mathbf{M}_1[i] \neq \mathbf{M}_1[j]$ then abort EndIf
$C_1 \xleftarrow{\$} \mathcal{SE}_K(\mathbf{M}_0[1])$; $st \leftarrow (\mathbf{pk}, K, K', C_1, st')$
Return (K, K', st)
Adversary $B_1^{\mathcal{D}_{I,\mathbf{sk}[1]}(\cdot),\ldots,\mathcal{D}_{I,sk[n(k)]}(\cdot)}(\text{guess}, \mathbf{C}_0, st)$
Parse st as $(\mathbf{pk}, K, K', C_1, st')$
For $i = 1 \ldots n(k)$ do $\mathbf{C}[i] \leftarrow \mathbf{C}_0[i] \| C_1$ EndFor
Run $A_{\text{cca}}^{\mathcal{HD}_{I,\mathbf{sk}[1]}(\cdot),\ldots,\mathcal{HD}_{I,\mathbf{sk}[n(k)]}(\cdot)}(\text{find}, \mathbf{C}, st')$:
When A_{cca} makes a query C' to its decryption
oracle $\mathcal{HD}_{I,sk[i]}(\cdot)$ [$1 \leq i \leq n(k)$]
parse C' as $C'_0 \| C'_1$
If $C'_0 \neq \mathbf{C}_0[1]$ then $K'' \leftarrow \mathcal{D}_{I,sk[i]}(C'_0)$
$M \leftarrow \mathcal{SD}_{K''}(C'_1)$ else $M \leftarrow \mathcal{SD}_K(C'_1)$ EndIf
Return M to A_{cca}
When A_{cca} outputs d, return d

Fig. 1. The adversary for the proof of Claim C.1.

Adversary $C^{\mathcal{SD}_K(\cdot)}(\text{find}, k)$
$I \xleftarrow{\$} \mathcal{G}(1^k)$; For $i = 1 \ldots n(k)$ do
$(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor
$K' \xleftarrow{\$} \mathcal{SK}(k)$; $(1^l, st') \xleftarrow{\$} A_{\text{cca}}(\text{select}, I)$
If $l \neq n(k)$ then abort EndIf
$(\mathbf{M}_0, \mathbf{M}_1, st') \xleftarrow{\$} A_{\text{cca}}^{\mathcal{HD}_{I,\mathbf{sk}[1]}(\cdot)\ldots\mathcal{HD}_{I,\mathbf{sk}[n(k)]}(\cdot)}(\text{find},$
$\mathbf{pk}, st')$
[C answers A_{cca}'s decryption queries
using $\mathbf{sk}[1], \ldots, \mathbf{sk}[n(k)]$]
If $\exists\ 1 \leq i, j \leq n(\cdot)$ such that $\mathbf{M}_0[i] \neq \mathbf{M}_0[j]$ or
$\mathbf{M}_1[i] \neq \mathbf{M}_1[j]$ then abort EndIf
$\mathbf{C}_0 \xleftarrow{\$} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$; $C_1 \xleftarrow{\$} \mathcal{SE}_K(\mathbf{M}_0[1])$
$st \leftarrow (\mathbf{pk}, \mathbf{C}_0, K', st')$
Return $(\mathbf{M}_0[1], \mathbf{M}_1[1], st)$
Adversary $C^{\mathcal{D}_K(\cdot)}(\text{guess}, C_1, st)$
Parse st as $(\mathbf{pk}, \mathbf{C}_0, K', st')$
For $i = 1 \ldots n(k)$ do $\mathbf{C}[i] \leftarrow \mathbf{C}_0[i] \| C_1$ EndFor
Run $A_{\text{cca}}(\text{find}, \mathbf{C}, st')$ as follows
When A_{cca} makes a query C' to its decryption oracle
$\mathcal{HD}_{I,\mathbf{sk}[i]}(\cdot)$ [$1 \leq i \leq n(k)$]
parse C' as $C'_0 \| C'_1$
If $C'_0 \neq \mathbf{C}_0[1]$ then $K'' \leftarrow \mathcal{D}_{I,sk[i]}(C'_0)$
$M \leftarrow \mathcal{SD}_{K''}(C'_1)$ else $M \leftarrow \mathcal{SD}_K(C'_1)$ EndIf
Return M to A_{cca}
When A_{cca} outputs d, return d

Fig. 2. The adversary for the proof of Claim C.2.

We comment on how B_1 answers A_{cca}'s decryption oracle queries. If the first (asymmetric) part of the ciphertext queried by A_{cca} is different from the elements of B_1's challenge ciphertext (which are all equal) or if the challenge ciphertext is not yet known to B_1, then B_1 can answer A_{cca}'s decryption query by using the corresponding decryption oracle on the asymmetric part of the ciphertext to compute the symmetric key and then use the latter to decrypt the symmetric part of the ciphertext. If the asymmetric part of the ciphertext queried by A_{cca} is the same as the elements of B_1's challenge ciphertext, then B_1 cannot use its decryption oracles, but in this case, B_1 knows the symmetric key K and can just decrypt the symmetric part of the queried ciphertext.

For the case of chosen-plaintext attacks, B_1 and A_{cpa} are not given the decryption oracles, hence, B_1 would not need to answer A_{cpa}'s decryption queries.

Analyzing the adversary we claim that

$$\begin{aligned}\mathbf{Adv}^{\text{smmr-atk}}_{\overline{\mathcal{AE}},B_1,n(\cdot)}(k) &= \Pr\left[\mathbf{Exp}^{\text{smmr-atk-0}}_{\overline{\mathcal{AE}},B_1,n(\cdot)}(k) = 0\right] \\ &\quad - \Pr\left[\mathbf{Exp}^{\text{smmr-atk-1}}_{\overline{\mathcal{AE}},B_1,n(\cdot)}(k) = 0\right] \\ &\leq \Pr\left[\mathbf{Exp}\,\boldsymbol{H}^{\text{4-atk}}_{\mathcal{HS},A_{\text{atk}}}(k)\right] \\ &\quad - \Pr\left[\mathbf{Exp}\,\boldsymbol{H}^{\text{3-atk}}_{\mathcal{HS},A_{\text{atk}}}(k)\right] \\ &= P_4^{\text{atk}} - P_3^{\text{atk}}\end{aligned}$$

and that B_1 runs in polynomial time. □

Claim C.2: Again we consider a more general case of chosen-ciphertext attacks and then specify the changes pertaining to the case of chosen-plaintext attacks. We present a pseudocode for an adversary C in Fig. 2.

We comment on how C answers A_{cca}'s decryption oracle queries. If the first (asymmetric) part of the ciphertext queried by A_{cca} is different from C's challenge ciphertext (which are all equal) or when the challenge ciphertext is not known to C yet, then C can answer A_{cca}'s decryption query by using the asymmetric secret keys. If the asymmetric part of the ciphertext queried by A_{cca} is the same as C's challenge ciphertext, then C can just decrypt the symmetric part of the queried ciphertext by querying it to its own decryption oracle.

For the case of chosen-plaintext attacks, C and A_{cpa} are not given the decryption oracles, hence, C would not need to answer A_{cpa}'s decryption queries. Thus, we have

$$\mathbf{Adv}^{\text{1-atk}}_{\mathcal{SE},C}(k) \leq P_3^{\text{atk}} - P_2^{\text{atk}}$$

and that C runs in polynomial time. □

Claim C.3: The proof is similar to the proof of Claim C.1. The main difference is that B_2 will output (K', K) at the end of its find stage, when B_2 the proof of Claim C.1 outputs (K', K). □

REFERENCES

[1] O. Baudron, D. Pointcheval, and J. Stern, "Extended notions of security for multicast public key cryptosystems," in *International Colloquium on Automata, Languages and Programming (ICALP'00) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 2000.

[2] M. Abdalla, M. Bellare, and P. Rogaway, "The oracle diffie-hellman assumptions and an analysis of DHIES," in *CT-RSA 01 (Lecture Notes in Computer Science)*, D. Naccache, Ed. Berlin, Germany: Springer-Verlag, 2001, vol. 2020.

[3] M. Bellare, A. Boldyreva, and S. Micali, "Public-key encryption in a multi-user setting: Security proofs and improvements," in *Advances in Cryptology-EUROCRYPT'00 (Lecture Notes in Computer Science)*, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, 2000, vol. 1807.

[4] M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon, "Multi-Recipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security." Extended version of this paper. [Online]. Available: www.cc.gatech.edu/aboldyre/publications.html

[5] M. Bellare, A. Boldyreva, and J. Staddon, "Multi-recipient encryption schemes: Efficient constructions and their security," in *Proc. International Workshop on Practice and Theory in Public Key Cryptography (PKC'03) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 2003.

[6] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," in *Proc. 38th IEEE Symp. Foundations of Computer Science*, Miami Beach, FL, Oct. 1997, pp. 394–403.

[7] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Advances in Cryptology—CRYPTO'98 (Lecture Notes in Computer Science)*, H. Krawczyk, Ed. Berlin, Germany: Springer-Verlag, 1998, vol. 1462.

[8] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Advances in Cryptology—CRYPTO'92 (Lecture Notes in Computer Science)*, E. Brickell, Ed. Berlin, Germany: Springer-Verlag, 1992, vol. 740.

[9] M. Bellare, T. Kohno, and V. Shoup, "Stateful public-key cryptosystems: How to encrypt with one 160-bit exponentiation," in *Proc. of the ACM Conference on Computer and Communications Security (CCS), ACM*, 2006.

[10] M. Bellare and P. Rogaway, "Optimal asymmetric encryption—How to encrypt with RSA," in *Advances in Cryptology—EUROCRYPT'94 (Lecture Notes in Computer Science)*, D. Santis, Ed., 1994, vol. 950, Springer-Verlag.

[11] S. Berkovits, "How to broadcast a secret," in *Advances in Cryptology—EUROCRYPT'91 (Lecture Notes in Computer Science)*, D. Davies, Ed. Berlin, Germany: Springer-Verlag, 1991, vol. 547.

[12] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM J. Comput.*, vol. 13, no. 4, Nov. 1984.

[13] D. Boneh, "Simplified OAEP for the RSA and Rabin functions," in *Advances in Cryptology—CRYPTO'01 (Lecture Notes in Computer Science)*, J. Kilian, Ed. Berlin, Germany: Springer-Verlag, 2001, vol. 2139.

[14] J. Camenisch and M. Michels, "Confirmer signature schemes secure against adaptive adversaries," in *Advances in Cryptology—EUROCRYPT'00 (Lecture Notes in Computer Science)*, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, 2000, vol. 1807.

[15] R. Canetti, "Toward realizing random oracles: Hash functions that hide all partial information," in *Advances in Cryptology—CRYPTO'97 (Lecture Notes in Computer Science)*, B. Kalisk_i, Ed. Berlin, Germany: Springer-Verlag, 1997, vol. 1294.

[16] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Advances in Cryptology—CRYPTO'98 (Lecture Notes in Computer Science)*, H. Krawczyk, Ed. Berlin, Germany: Springer-Verlag, 1998, vol. 1462.

[17] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, 2003.

[18] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.

[19] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO'93 (Lecture Notes in Computer Science)*, D. Stinson, Ed. Berlin, Germany: Springer-Verlag, 1993, vol. 773.

[20] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Advances in Cryptology—CRYPTO'99 (Lecture Notes in Computer Science)*, M. Wiener, Ed. Berlin, Germany: Springer-Verlag, 1999, vol. 1666.

[21] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, "RSA-OAEP is secure under the RSA assumption," in *Advances in Cryptology—CRYPTO'01 (Lecture Notes in Computer Science)*, J. Kilian, Ed. Berlin, Germany: Springer-Verlag, 2001, vol. 2139.

[22] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comp. Syst. Sci,*, vol. 28, pp. 270–299, 1984.

[23] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. Assoc. Comput. Mach.*, vol. 33, no. 4, pp. 210–217, 1986.

[24] J. Håstad, "Solving simultaneous modular equations of low degree," *SIAM J. Comput.*, vol. 17, no. 2, pp. 336–341, Apr. 1988.

[25] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, "A pseudorandom generation from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[26] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography," in *Proc. 30th IEEE Symp. Foundations of Computer Science*, Research Triangle Park, NC, Oct./Nov. 1989, pp. 230–235.

[27] K. Kurosawa, "Multi-recipient public-key encryption with shortened ciphertext," in *Proc. Int. Workshop on Practice and Theory in Public Key Cryptography (PKC'02) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 2002.

[28] S. Micali, C. Rackoff, and R. H. Sloan, "The notion of security for probabilistic cryptosystems," in *Advances in Cryptology—CRYPTO'86 (Lecture Notes in Computer Science)*, A. Odlyzko, Ed. Berlin, Germany: Springer-Verlag, , 1986, vol. 263.

[29] M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," in *Proc 38th IEEE Symp. Foundations of Computer Science*, Miami Beach, FL, Oct. 1997, pp. 458–467.

[30] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proc. ACM Symp. Theory of Computing (STOC'89)*, Seattle, WA, May 1989, pp. 33–43.

[31] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," in *Proc. ACM Symp. Theory of Computing (STOC'90)*, Baltimore, MD, May 1990, pp. 387–394.

[32] C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attack," in *Advances in Cryptology—CRYPTO'91 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1991.

[33] V. Shoup, "On formal models for secure key exchange," Univ. Calif. San Diego, La Jolla, CA, 1999, IBM Res. Rep. RZ 3120 [Online]. Available: http:philby.ucsd.edu/cryptolib

[34] M. Stadler, "Publicly verifiable secret sharing," in *Advances in Cryptology—EUROCRYPT'96 (Lecture Notes in Computer Science)*, U. Maurer, Ed. Berlin, Germany: Springer-Verlag, 1996, vol. 1070.

[35] Y. Tsiounis and M. Yung, "On the security of ElGamal based encryption," in *Proc. Int. Workshop on Practice and Theory in Public Key Cryptography (PKC'98) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1998.

[36] A. C. Yao, "Theory and application of trapdoor functions," in *Proc. 23rd IEEE Symp. Foundations of Computer Science*, Chicago, IL, Nov. 1982, pp. 80–91.