This is the full version of the paper with same title that appeared in *IEEE Transactions on Information Theory*, Volume 53, Number 11, 2007. It extends the previously published versions [Ku, BBS].

Multi-Recipient Encryption Schemes: Efficient Constructions and their Security

M. BELLARE* A. BOLDYREVA† K. KUROSAWA‡ J. 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 re-use. In order to analyze security of our constructions we introduce a new notion of multi-recipient 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 re-using MRES is secure. The results and applications cover both asymmetric and symmetric encryption.

Keywords: Encryption, randomness, provable security, broadcast encryption.

1 Introduction

1.1 Multi-Recipient 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

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. mihir@cs.ucsd.edu. http://www-cse.ucsd.edu/users/mihir.

†College of Computing, Georgia Institute of Technology, 801 Atlantic dr., Atlanta, Georgia 30332, USA. sasha@gatech.edu. www.cc.gatech.edu/~aboldyre.

‡Dept. of Computer and Information Sciences, Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan. kurosawa@cis.ibaraki.ac.jp.

§CSL, Palo Alto Research Center, 3333 Coyote Hill Rd. Palo Alto, CA 94304, USA. staddon@parc.com.

(because encryption for each receiver is done independently), we first define a new setting of *multi-recipient encryption* as follows[1]. 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 *multi-recipient encryption algorithm* $\overline{\mathcal{E}}$ 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 *multi-recipient 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 multi-recipient encryption algorithm. Key generation and decryption are just like in a standard scheme. We will also consider a scenario when a 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 re-use*, accordingly we start with the description of this idea and the corresponding subclass of MRESs that exploit randomness re-use.

1.2 Randomness Re-using MRESs

We propose to consider MRESs constructed from the standard encryption schemes by applying what we call randomness re-use. Namely, we suggest, that re-using random coins when computing ciphertexts for different receivers may often provide computational and bandwidth savings. Consider a multi-recipient 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 ($1 \le i \le n$). The corresponding MRES is called the *randomness re-using* MRES (RR-MRES) associated to the underlying standard encryption scheme.

1.3 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 $1 \le i \le n$. 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 $1 \le i \le n$. 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 multi-cast by the sender, since in that case the transmission would be $\mathbf{C} = (g^r, g^{x_1 r} \cdot M_1, \ldots, g^{x_n r} \cdot M_n)$, 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 [CrSh], permits similar computational savings.

[1] Let us restrict our attention for the moment to asymmetric-key setting. We turn to symmetric-key setting later.

DHIES. This is a Diffie-Hellman-based asymmetric encryption scheme proposed in [ABR] and adopted by draft standards ANSI X9.63EC and IEEE P1363a. It has ElGamal-like cost in public-key operations. Unlike ElGamal and Cramer-Shoup it does not assume the plaintext is a group element, but handles arbitrary plaintext strings via a hybrid construction involving a symmetric encryption scheme. Randomness re-use for this scheme is attractive since it results in bandwidth and computational savings in various applications just as for the ElGamal scheme.

CBC. We also consider the symmetric setting. We consider popular CBC encryption with random IV, based on a given block cipher. The IV is the randomness underlying the encryption. Randomness re-use 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 re-used 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.

1.4 Security Notions for MRESs

The above examples shows that there are MRESs that are more efficient than the naive one. But are they secure? The first step towards 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 multi-user setting of [BBM, BPS], encryptions to other recipients use independent coins, but ciphertexts created by a multi-recipient 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 4 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 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.

1.5 The Reproducibility Theorem for Randomness Re-using MRESs

Many RR-MRESs offer performance benefits, but not all are secure. (We illustrate the latter in Section 5 by showing how Håstad's attacks [Hå] can be exploited to break RR-MRESs based on RSA-OAEP [BR].) We are interested in determining which RR-MRESs are secure MRESs. Direct case by case analyses 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 re-use (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 sk_2 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.

1.6 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 DDH assumption ElGamal is IND-CPA secure Cramer-Shoup is IND-CCA secure [CrSh], imply that under the same assumption the ElGamal RR-MRES is IND-CPA secure and the Cramer-Shoup based one is IND-CCA secure.

We then extend these results by providing reductions of improved concrete security. These improvements do not use the reproducibility theorem, instead directly exploiting the reproducibility property of the base schemes and, as in [BBM], using self-reducibility properties of the DDH problem [St, NR, Sh].

DHIES. Our analysis exploits both asymmetric and symmetric reproducibility. We show that if the underlying symmetric scheme is reproducible then so is the resulting (asymmetric) DHIES scheme. In particular, if the symmetric encryption scheme is CBC (the most popular choice in practice) then DHIES is reproducible. DHIES achieves Cramer-Shoup-like security (IND-CCA), although the proof [ABR] relies on significantly stronger assumptions than the DDH assumption used in [CrSh]. As usual, our reproducibility theorem then implies that the corresponding randomness re-using multi-recipient scheme is IND-CCA under the assumptions used to establish that DHIES is IND-CCA.

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 [BDJR], the reproducibility theorem implies that the randomness re-using 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 [BBM] 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 multi-user 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 [BBM] do not imply that the hybrid SM-MRES we proposed is secure. Our results (see Section 10) 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) security and the SM-MRES is IND-CPA (resp. IND-CCA) secure, then the corresponding hybrid SM-MRES is IND-CPA (resp. IND-CCA) secure [2]. 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 re-used) 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 re-used in the encryption algorithm of the hybrid SM-MRES.

1.7 Minimal assumptions for secure randomness re-use

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 8, are obtained by transforming a given standard encryption scheme into another standard encryption scheme that permits secure randomness re-use. 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.

1.8 Discussion and related work

ON RE-USING RANDOMNESS. At first glance, re-using 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 a 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 very recent paper [BKS] shows how to utilize re-using randomness to achieve even better efficiency for some schemes. They consider stateful encryption that generalizes MRES, and show that batching can

[2] 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 10

also be exploited when multiple messages are sent to receivers (multiple or single.)

USING PRGS. A natural question is, instead of re-using randomness, why not use pseudorandom bit generators? Indeed, randomness generation costs for encryption can be reduced by picking a single, short random seed s and applying a pseudorandom bit generator 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 [BM, Y], 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 re-use permits applications that usage of pseudorandomness does not permit. A case in point is the efficiency improvements discussed above. Furthermore, randomness re-use 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) [FN] 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 $\mathbf{C}$ 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 $(\mathbf{C}[1], \ldots, \mathbf{C}[n])$ 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 multi-recipient 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.

2 Preliminaries

2.1 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 $X \stackrel{\$}{\leftarrow} S$ denotes that X is selected uniformly at random from S. For convinience for any $k \in \mathbb{N}$ we will often write $X_1, X_2, \ldots, X_k \stackrel{\$}{\leftarrow} S$ as a shorhand for $X_1 \stackrel{\$}{\leftarrow} S; X_2 \stackrel{\$}{\leftarrow} S; \ldots; X_n \stackrel{\$}{\leftarrow} S$. 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 $X \stackrel{\$}{\leftarrow} A(X_1, X_2, \ldots, X_k)$ 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.

2.2 Definitions

A function $f\colon \mathbb{N} \to [0,1]$ 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 [BBM] 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:

$$I \xleftarrow{\$} \mathcal{G}(1^k)\ ;\ (pk, sk) \xleftarrow{\$} \mathcal{K}(I)\ ;\ M \xleftarrow{\$} \mathrm{MsgSp}(I)$$
$$\text{If } \mathcal{D}_{I,sk}(\mathcal{E}_{I,pk}(M)) = M \text{ then return 1 else return 0}$$

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 [GoMi, MRS, RS].

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

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

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

Above st denotes the state information the adversary wants to preserve. It is mandated that $|M_0| = |M_1|, M_0, M_1 \in \mathrm{MsgSp}(I)$ and 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 as follows:

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

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 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.

3 Multi-Recipient Asymmetric Encryption Schemes

3.1 Syntax

An asymmetric *multi-recipient encryption scheme (MRES)* $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ 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 *multi-encryption* algorithm $\overline{\mathcal{E}}$ takes input a common key I, a *public-key vector* $\mathbf{pk} = (\mathbf{pk}[1], \ldots, \mathbf{pk}[n])$ and a *plaintext vector* $\mathbf{M} = (\mathbf{M}[1], \ldots, \mathbf{M}[n])$ and returns a *ciphertext vector* $\mathbf{C} = (\mathbf{C}[1], \ldots, \mathbf{C}[n])$. Associated to each common key I is a *message space* $\mathrm{MsgSp}(I)$ from which the components of $\mathbf{M}$ are allowed to be drawn. We require that the following experiment returns 1 with probability 1:

$$\begin{array}{l} I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)\,;\ \text{For } i = 1, \ldots, n \text{ do } (\mathbf{pk}[i], \mathbf{sk}[i]) \stackrel{\$}{\leftarrow} \mathcal{K}(I) \text{ EndFor;} \\ M \stackrel{\$}{\leftarrow} \mathrm{MsgSp}(I)\,;\ \mathbf{C} \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{I,\mathbf{pk}}(\mathbf{M}) \\ j \stackrel{\$}{\leftarrow} \{1, \ldots, n\}\,;\ \text{If } \mathcal{D}_{I,\mathbf{sk}[j]}(\mathbf{C}[j]) = \mathbf{M}[i] \text{ then return 1 else return 0} \end{array}$$

We do not specify how $\mathbf{C}[i]$ is communicated to user i. It could be that the whole ciphertext vector $\mathbf{C}$ is sent via a broadcast or multi-cast channel and, if all $\mathbf{C}[i]$ have a common part due to a randomness re-use, this part can be sent only once. It could also be that $\mathbf{C}[i]$ 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 Multi-Recipient Encryption Schemes (*SM-MRESs*), also called a broadcast encryption scheme, 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 a single-message multi-recipient encryption scheme (SM-MRES) if there exists a multi-encryption algorithm $\overline{\mathcal{E}}$ such that $(\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ is a MRES as defined above and $\mathcal{E}^1$ is defined by

$$\begin{array}{l} \mathcal{E}^1_{I,\mathbf{pk}}(M) \\ \quad \text{Let } n \text{ be the number of components of } \mathbf{pk} \\ \quad \text{For } i = 1, \ldots n \text{ do } \mathbf{M}[i] \leftarrow M \text{ EndFor} \\ \quad \mathbf{C}[i] \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M}, r) \\ \quad \text{Return } \mathbf{C} \end{array}$$

3.2 Randomness Re-using MRESs

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

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

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

For examples of RR-MRESs see Section 7.

4 Security of Asymmetric Multi-Recipient 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 $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ 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 $1 \leq l \leq n(k)$, 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; a $(n(k) - l)$-vector of random coins which are is 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 $\mathbf{M}$. The latter is encrypted via the multi-encryption algorithm to yield a challenge ciphertext $\mathbf{C}$ 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 $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ be a multi-receiver asymmetric encryption scheme. For $\text{atk} \in \{\text{cpa}, \text{cca}\}$ and $b \in \{0, 1\}$ consider the experiments:

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

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

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

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

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

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

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

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

(8) $d \xleftarrow{\$} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\text{guess}, \mathbf{C}, st)$

(9) Return d

Above, the oracles for $1 \leq i \leq l$ are defined as follows: If $\text{atk} = \text{cpa}$ then $\mathcal{O}_i(\cdot) = \varepsilon$ and if $\text{atk} = \text{cca}$ then $\mathcal{O}_i(\cdot) = \mathcal{D}_{I,\mathbf{sk}[i]}(\cdot)$. It is mandated that for all $1 \leq i \leq l$ we have $|M_0[i]| = |M_1[i]|$ and all message vector componenets are in the scheme's message space, and also that if $\text{atk} = \text{cca}$ then the adversary B does not query $\mathcal{O}_i(\cdot)$ on $\mathbf{C}[i]$. 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 ind-atk *advantage* of an adversary B is

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

We will say that MRES $\overline{\mathcal{AE}}$ is IND-CPA (resp. IND-CCA) secure if the function $\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$ (respectively $\mathbf{Adv}^{\text{mr-cca}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$) 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 $\text{atk} = \{\text{cpa}, \text{cca}\}$ 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 Section 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) $I \xleftarrow{\$} \mathcal{G}(1^k)$; $(1^l, st) \xleftarrow{\$} B(\text{select}, n(k), I)$ $\ \ \mathbf{[}\, l = n(k) \mathbf{]}$

...

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

...

Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ be a single-message multi-recipient encryption scheme. The ind-atk *advantage* of an adversary B is

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

We say that SM-MRES $\overline{\mathcal{AE}}$ is IND-CPA (resp. IND-CCA) secure if the function $\mathbf{Adv}^{\text{smmr-cpa}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$ (respectively $\mathbf{Adv}^{\text{-smmr-cca}}_{\overline{\mathcal{AE}},B,n(\cdot)}(\cdot)$) 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 multi-user setting [BBM, BPS] only considered outsider attacks, meaning the adversary was not one of the receivers. However, in the multi-recipient 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, rf. [Ku]). 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 multi-recipient setting with randomness re-use the adversary can register this public key and later after receiving a ciphertext can obtain the random string 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 multi-recipient scheme $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}', \overline{\mathcal{E}}, \mathcal{D})$, 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 $\overline{\mathcal{E}}_{I,\mathbf{pk}}(\mathbf{M})$
  $r \stackrel{\$}{\leftarrow} \mathbb{Z}_q$
  For $i = 1, \ldots, n$ do
    $\mathbf{C}'[i] \stackrel{\$}{\leftarrow} \mathcal{E}'_{pk'_i}(r)$; $Y_i \leftarrow g^r$; $W_i \leftarrow (g^{x_i})^r M[i]$; $\mathbf{C}[i] \leftarrow (Y_i, W_i, \mathbf{C}'[i])$
  EndFor
  Return $\mathbf{C}$

We omit the description of $\mathcal{D}$. We claim that $\overline{\mathcal{AE}}$ is wIND-CPA secure while it is 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 $\mathbf{C}$ it decrypts $\mathbf{C}'[1]$ using sk'_1 and obtains r. Now A can compute $\mathbf{M}[i]$ 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 $\mathbf{C}_D$, picks a random r' and computes a challenge $\mathbf{C}_B$ for B such that $\mathbf{C}_B[i] = (\mathbf{C}_D[i], \mathcal{E}'_{I,pk'_i}(r'))$. 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 [Ku] that the latter scheme is wIND-CPA, so this would imply that $\overline{\mathcal{AE}}$ is also wIND-CPA.

Moreover, for analyses of multi-recepient 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 re-use. 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 [BG], 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.

5 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 [BR] that has been proven to be IND-CCA secure (in the random oracle model) [FOPS] and hence is also IND-CCA secure in a multi-user setting [BBM, BPS]. 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 $\mathbf{M} = (M, M, M)$. 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 $\mathbf{C}[i] = x^3 \bmod N_i$, and send $\mathbf{C}[i]$ to i. An adversary given $\mathbf{C}$ can use Håstad's attack [Hå] (based on the fact that the modulii 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 re-use 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.

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

$I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)\ ;\ (pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}(I)\ ;\ M \stackrel{\$}{\leftarrow} \mathrm{MsgSp}(I)\ ;\ r \stackrel{\$}{\leftarrow} \mathrm{Coins}_{\mathcal{E}}(I)$

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

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 there exists a RPTA R called the reproduction algorithm such that $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{AE},R}(k)$ outputs 1 with the 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 $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ 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

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

The proof is Appendix A.

7 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.

The security of the schemes we consider here is based on the hardness of the Decisional Diffie-Hellman (DDH) problem for appropriate prime-order-group generators. Accordingly we begin with definitions for the latter.

A *prime-order-group generator* is a RPTA that 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 $\mathbb{G}$. There can be numerous such prime-order-group generators. We will not specify a particular one but will use it as a parameter to the computational problems we consider. The description of a group should specify the algorithms for group operations (multiplication and inverse), the algorithm for testing group membership, and also the random group element sampling algorithm. All of these algorithms are assumed to be PTAs. Here and further in the paper we assume that the group elements are uniquely encoded as strings. We let $\hat{1}$ denote the identity element of $\mathbb{G}$. Let T^{exp}_q denote the worst time needed to perform an exponentiation operation with respect to a base element in $\mathbb{G}$ and an exponent in $\mathbb{Z}_q$, for any $\tilde{\mathbb{G}}, q, g$ output of $\mathcal{G}(1^k)$. This operation is assumed to be polynomial in k.

Definition 7.1 [DDH] Let $\mathcal{G}$ be a prime-order-group generator. Let D be an adversary that on input $\tilde{\mathbb{G}}, q, g$ and three elements $X, Y, T \in \mathbb{G}$ returns a bit. We consider the following experiments

Experiment $\mathbf{Exp}^{\mathrm{ddh\text{-}real}}_{\mathcal{G},D}(k)$	Experiment $\mathbf{Exp}^{\mathrm{ddh\text{-}rand}}_{\mathcal{G},D}(k)$
$(1^k, \tilde{\mathbb{G}}, q, g) \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$	$(1^k, \tilde{\mathbb{G}}, q, g) \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$
$x \stackrel{\$}{\leftarrow} \mathbb{Z}_q\ ;\ X \leftarrow g^x\ ;\ y \stackrel{\$}{\leftarrow} \mathbb{Z}_q\ ;\ Y \leftarrow g^y$	$x \stackrel{\$}{\leftarrow} \mathbb{Z}_q\ ;\ X \leftarrow g^x\ ;\ y \stackrel{\$}{\leftarrow} \mathbb{Z}_q\ ;\ Y \leftarrow g^y$
$T \leftarrow g^{xy}$	$T \stackrel{\$}{\leftarrow} \mathbb{G}$
$d \stackrel{\$}{\leftarrow} D(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$	$d \stackrel{\$}{\leftarrow} D(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$
Return d	Return d

The advantage of D in solving the Decisional Diffie-Hellman (DDH) problem for $\mathcal{G}$ is the function of the security parameter defined by

$$\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) \;=\; \Pr\Big[\,\mathbf{Exp}^{\mathrm{ddh\text{-}real}}_{\mathcal{G},D}(k)=1\,\Big] - \Pr\Big[\,\mathbf{Exp}^{\mathrm{ddh\text{-}rand}}_{\mathcal{G},D}(k)=1\,\Big]\;.$$

We say that the DDH problem is hard for $\mathcal{G}$ if the function $\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(\cdot)$ is negligible for every RPTA D. ▮

We will refer to (g, X, Y, T) as to a *valid* Diffie-Hellman tuple if $\log_g X = \log_Y T$, and as to a *random* Diffie-Hellman tuple otherwise.

A common case is that $\mathbb{G}$ is a subgroup of order q of $\mathbb{Z}_p^*$ where p is a prime such that q divides $p-1$. Another example is when $\mathbb{G}$ is an appropriate elliptic curve group. Our setting is general enough to encompass both these cases.

7.1 ElGamal

The ElGamal scheme in a group of prime order is known to be IND-CPA under the assumption that the decision Diffie-Hellman (DDH) problem is hard. (This is noted in [C, NR, CrSh, TY].) 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:

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

The message space associated to a common key (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 7.2 The ElGamal encryption scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible.

Proof: On input $(I, pk, X, (g^r, g^{rx} \cdot M), M', pk', sk')$, where $I = (1^k, \tilde{\mathbb{G}}, q, g)$, $pk = g^x$, $pk' = g^{x'}$, $sk' = x'$, a PTA R returns $(g^r, (g^r)^{x'} \cdot M')$. 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 7.2 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. The following theorem shows that it is possible to obtain a tighter relation than the one implied by Theorem 6.2. It means that re-using randomness while encrypting messgages for different receivers almost does not comprise security and, as we discussed in the introduction, reduces bandwidth and computational costs by about 50%.

Theorem 7.3 Let $\mathcal{G}$ be a prime-order-group generator, $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ the associated ElGamal encryption scheme, and $\overline{\mathcal{EG}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ the associated RR-MRES as per Construction 3.1. Then for any adversary B there exists a adversary D such that for any k

$$\mathbf{Adv}^{\mathrm{mr\text{-}cpa}}_{\overline{\mathcal{EG}},B,n(\cdot)}(k) \;\le\; 2 \cdot \mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + \frac{1}{2^{k-2}},$$

where the running time of D is one of B plus $O(n(k) \cdot k^3)$. ▮

The proof of the above theorem is in Appendix B.

$\underline{\mathcal{G}(1^k)}$:

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

$\underline{\mathcal{E}_{I,pk}(M)}$:
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)

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

$\underline{\mathcal{D}_{I,sk}(u_1, u_2, e, v)}$:
Parse I as $(1^k, \tilde{\mathbb{G}}, g_1, g_2, K)$
Parse sk as $(x_1, x_2, y_1, y_2, z_1, z_2)$
$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
(D1) If $v \neq {u_1}^{x_1 + y_1\alpha} {u_2}^{x_2 + y_2\alpha}$
Then return $\perp$ kwfontEndIf
(D2) $f \leftarrow u_1^{z_1} u_2^{z_2}$
(D3) $M \leftarrow e/f$
Return M

Figure 1: Cramer-Shoup scheme

7.2 Cramer-Shoup

We now consider an RR-MRES based on the Cramer-Shoup scheme [CrSh, CrSh2] 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 $\overline{\mathcal{G}}$ 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 depicted in Figure 1 [CrSh2, Sec.6.1]. The message space associated to a common key $(1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K)$ is $\mathbb{G}$.

Lemma 7.4 The Cramer-Shoup encryption scheme $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible.

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 $e' = u_1^{z_1'} u_2^{z_2'} M' = g_1^{rz_1'} g_2^{rz_2'} M' = (h')^r M'$. 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 $\Pr\left[\, \mathbf{Exp}^{\mathrm{repr}}_{\mathcal{CS},R}(1^k) = 1 \,\right] = 1$. ▮

Before we analyze the scheme let us recall the definition of collision resistance of hash function families, since it will be used in our analysis.

COLLISION-RESISTANT HASH FUNCTIONS. A family of hash functions $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ is defined by a probabilistic generator algorithm $\mathcal{GH}$, which takes as input 1^k, where $k \in \mathbb{N}$ is the 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 $\mathcal{EH}_K(X) \in \{0,1\}^{k-1}$.

Definition 7.5 Let $\mathcal{H} = (\mathcal{GH}, \mathcal{EH})$ be a family of hash functions and let C be an adversary that on input a key K and a string X_0, returns a string X_1. Now, we consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{cr}}_{\mathcal{H},C}(k)$
$K \stackrel{\$}{\leftarrow} \mathcal{GH}(1^k), X_0 \stackrel{\$}{\leftarrow} \mathbb{G}^3$; $X_1 \leftarrow C(K, X_0)$
If $(X_0 \neq X_1)$ and $\mathcal{EH}_K(X_0) = \mathcal{EH}_K(X_1)$ then return 1 else return 0

We define the *advantage* of adversary C via

$$\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k) = \Pr\left[\, \mathbf{Exp}^{\mathrm{cr}}_{\mathcal{H},C}(k) = 1 \,\right] .$$

We say that the family of hash functions $\mathcal{H}$ is *target-collision-resistant* if $\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k)$ is negligible for every RPTA C. ▮

The notion of target-collision-resistant family of hash functions was shown by Cramer and Shoup [CrSh2]. It is a special case of universal one-way hash function UOWH family introduced by Naor and Yung [NY], where a UOWH family can be built from arbitrary one-way functions [NY, Rom].

If the DDH problem is hard for $\mathcal{G}$ and if $\mathcal{H}$ is target-collision-resistant then $\mathcal{CS}$ is IND-CCA secure [CrSh, CrSh2]. This fact, Theorem 6.2 and Lemma 7.4 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 linarly with the number of users. We get a better security result than the one implied by Theorem 6.2, and the following theorem states our improvement.

Theorem 7.6 Let $\mathcal{G}$ be a prime-order-group generator, $\mathcal{CS} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ the associated Cramer-Shoup encryption scheme and $\overline{\mathcal{CS}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ the associated RR-MRES as per Construction 3.1. Let $n(\cdot)$ be a polynomial. Then for any adversary B, which makes $Q(\cdot)$ decryption oracle queries in total, there exists probabilistic algorithms D and C such that

$$\mathbf{Adv}^{\mathrm{mr\text{-}cca}}_{\overline{\mathcal{CS}},B,n(\cdot)}(k) \;\leq\; 2\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + 2n(k) \cdot \mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k) + \frac{4(Q(k)+n(k)+3)}{2^k},$$

where the running times of D and C are essentially the same as that of B.

Note that the security of $\overline{\mathcal{CS}}$ is tightly related to the security of DDH. The proof of the above theorem is in Appendix C.

7.3 DHIES

We consider the other DDH-based encryption scheme DHIES [ABR] which is in several draft standards. It combines asymmetric and symmetric key encryption methods, a message authentication code and a hash function and provides security against chosen-ciphertext attacks. Let $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ be a symmetric

$\underline{\mathcal{E}_{I,pk}(M)}$:
Parse pk as (q, g, X)
$r \stackrel{\$}{\leftarrow} \mathbb{Z}_q$; $Y \leftarrow g^r$; $W \leftarrow H(X^r)$
Let sk_m be the first ml bits of W
Let sk_e be the last kl bits of W
$C \stackrel{\$}{\leftarrow} \mathcal{SE}_{sk_e}(M)$; $T \leftarrow \mathcal{T}_{sk_m}(C)$
Return (Y, C, T)

$\underline{\mathcal{D}_{I,sk}((Y, C, T))}$:
Parse sk as (q, g, x)
$W \leftarrow H(Y^x)$
Let sk_m be the first ml bits of W
Let sk_e be the last kl bits of W
$M \leftarrow \mathcal{SD}_{sk_e}(C)$
If $\mathcal{V}_{sk_m}(M, T) = 1$ then Return M
else Return $\perp$ EndIf

Figure 2: DHIES

encryption scheme[3] with key length kl. Let $\mathsf{MAC} = (\mathcal{T}, \mathcal{V})$ be a message authentication code with key length ml, where $\mathcal{T}$ is a tagging algorithm taking input the random key and a message and returning the tag, and $\mathcal{V}$ is a verification algorithm taking input the random key, a message and a tag and returning the 1 (if the tag is valid) and 0, otherwise. Let $H\colon \{0,1\}^{gl} \to \{0,1\}^{ml+kl}$ be a function. We assume MAC is deterministic. The common key and key generation algorithms of $\mathcal{DHIES}[\mathcal{SE}, H, \mathsf{MAC}] = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ are the same as the ones of ElGamal encryption scheme. The rest of the algorithms are presented in Figure 2.

Below we use the notion of reproducibility for symmetric encryption and the corresponding reproducibility theorem; please refer to Section 9 where we properly describe how the notions and results of this paper related to asymmetric multi-recipient schemes can be naturally extended for a case of symmetric encryption schemes.

Lemma 7.7 $\mathcal{DHIES}[\mathcal{SE}, H, \mathsf{MAC}] = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible if $\mathcal{SE}$ is reproducible.

Proof: Since $\mathcal{SE}$ is reproducible then there exists an RPTA reproduction algorithm R' for $\mathcal{SE}$ which takes a ciphertext and a random message and a secret key and outputs a ciphertext of this message under this secret key such that it is created using the same random coins as the given ciphertext. We present an RPT reproduction algorithm R for $\mathcal{DHIES}$ which uses R'.

Algorithm $R(I, pk, (g^r, C, T), M', X', x')$
Parse I as $(1^k, \tilde{\mathbb{G}}, q, g)$
$K \leftarrow H((g^r)^{x'})$
Let sk_m be the first ml bits of K ; let sk_e be the last kl bits of K
$C' \stackrel{\$}{\leftarrow} R'(C, M', sk_e)$; $T' \stackrel{\$}{\leftarrow} \mathcal{T}_{sk_m}(C')$
Return (g^r, C', T')

Note that R first computes symmetric keys for $\mathcal{SE}$ and MAC using given g^r and then uses R' to output a valid symmetric ciphertext which is created using the same random coins as the given ciphertext C and therefore the whole output (g^r, C', T') is always a valid ciphertext computed using the same coins as the original ciphertext (g^r, C, T). ▮

$\mathcal{DHIES}[\mathcal{SE}, H, \mathsf{MAC}]$ is proven to be IND-CCA secure if $\mathcal{SE}$ is IND-CPA, MAC is strongly universally-unforgeable under chosen-message attack and the Oracle Diffie-Hellman assumption[4] is hard for $\mathcal{G}$ and H. This fact, Theorem 6.2 and Lemma 7.7 imply that it is also RR-IND-CCA or, equivalently, the corresponding MRES is IND-CCA secure, under the same assumptions.

[3]We recall the syntax of symmetric encryption schemes in Section 9.1.
[4]See [ABR] for details.

8 From IND-CPA (IND-CCA) to RR-IND-CPA (RR-IND-CCA)

As Section 5 and Section 7 show, 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 pseudo-randomness.

PSEUDORANDOM FUNCTION FAMILIES. Let $kl\colon \mathbb{N} \to \mathbb{N}, il\colon \mathbb{N} \to \mathbb{N}, ol\colon \mathbb{N} \to \mathbb{N}$ 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 $g \stackrel{\$}{\leftarrow} F$ is a shorthand for $K \stackrel{\$}{\leftarrow} \{0,1\}^{kl}$; $g \leftarrow F(K, \cdot)$. 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 $g \stackrel{\$}{\leftarrow} R$ 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, whre $k \in \mathbb{N}$ 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. Define the *advantage of* D as

$$\mathbf{Adv}^{\mathrm{prf}}_{F,D}(k) \;=\; \Pr\left[\, D^{g(\cdot)}(1^k) = 1 \;:\; g \stackrel{\$}{\leftarrow} F \,\right] - \Pr\left[\, D^{g(\cdot)}(1^k) = 1 \;:\; g \stackrel{\$}{\leftarrow} R \,\right] .$$

The function family F is said to be *pseudorandom* if $\mathbf{Adv}^{\mathrm{prf}}_{F,D}(\cdot)$ 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 $il\colon \mathbb{N} \to \mathbb{N}, ol\colon \mathbb{N} \to \mathbb{N}$ such that for all k $|(I, pk)| = il$ and $\mathrm{Coins}(I) = \{0,1\}^{ol}$ 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 $kl\colon \mathbb{N} \to \mathbb{N}$ and fix a function family $F\colon \{0,1\}^{kl} \times \{0,1\}^{il} \to \{0,1\}^{ol}$. 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 $\mathcal{E}'_{I,pk}(M, r')$
  $r \leftarrow F(r', (I, pk))$; $C \stackrel{\$}{\leftarrow} \mathcal{E}_{I,pk}(M, r)$
  Return C ▮

In practice a block cipher such as AES can be often used in place F (if its 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 $il\colon \mathbb{N} \to \mathbb{N}, ol\colon \mathbb{N} \to \mathbb{N}$ 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 (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 above theorem states the asymptotic security result. In Appendix D we prove the concrete security result and the statement of the theorem follows.

The above 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 (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 [IL] and the existence of one-way functions implies the existence of pseudorandom generators [HILL] which in turn implies the existence of PRFs [GGM]. 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. ▌

9 Multi-Recipient 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.

9.1 Symmetric Encryption Schemes

SYNTAX. Following [BDJR], a symmetric encryption scheme $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ 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 [BDJR] we recall the security of a symmetric-key encryption scheme under chosen-plaintext and chosen-ciphertext attacks. An adversary attacking the encryption scheme is given an encryption oracle $\mathcal{SE}_K(\cdot)$ which returns an encryption of an input plaintext. An adversary wins if it can find two equal-length messages and is given a challenge ciphertext that is an encryption of one of the messages. The adversary wins if it correctly guesses which plaintext goes with the challenge ciphertext. In the case of chosen-ciphertext attacks the adversary is also given a decryption oracle, which decrypts input ciphertexts except the challenge ciphertext.

Definition 9.1 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 oraces. For $b = 0, 1$ define the experiments

Experiment $\mathbf{Exp}^{\mathrm{cpa}-b}_{\mathcal{SE},A_{\mathrm{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\mathrm{cca}-b}_{\mathcal{SE},A_{\mathrm{cca}}}(k)$
$sk \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$	$sk \stackrel{\$}{\leftarrow} \mathcal{K}(1^k)$
$(M_0, M_1, st) \stackrel{\$}{\leftarrow} A_{\mathrm{cpa}}{}^{\mathcal{SE}_{sk}(\cdot)}(\mathrm{find}, k)$	$(M_0, M_1, st) \stackrel{\$}{\leftarrow} A_{\mathrm{cca}}{}^{\mathcal{SE}_{sk}(\cdot),\mathcal{SD}(\cdot)}(\mathrm{find}, k)$
$C \stackrel{\$}{\leftarrow} \mathcal{SE}_{sk}(M_b)$	$C \stackrel{\$}{\leftarrow} \mathcal{SE}_{sk}(M_b)$
$d \stackrel{\$}{\leftarrow} A_{\mathrm{cpa}}{}^{\mathcal{SE}_{sk}(\cdot)}(\mathrm{guess}, C, st)$	$d \stackrel{\$}{\leftarrow} A_{\mathrm{cca}}{}^{\mathcal{SE}_{sk}(\cdot),\mathcal{SD}(\cdot)}(\mathrm{guess}, C, st)$
Return d	Return d

Above st denotes the state information the adversary wants to preserve. 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 as follows:

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

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

We will also use weaker definitions of security for symmetric encryption schemes, wIND-CPA and wIND-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 weakly IND-CPA secure (resp. weakly IND-CCA secure). We remark that the latter weak definition of security is called Find-Guess (FG) security definition in [FO].

9.2 Symmetric-Key MRESs

We now consider MRESs in the symmetric-key setting. Syntax for such schemes $\overline{\mathsf{SE}} = (\mathcal{SK}, \overline{\mathcal{SE}}, \mathcal{SD})$ can be defined similarly to syntax of asymmetric MRESs defined in Section 2.2. 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 for a symmetric-key RR-MRES 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 given an encryption oracle which takes as input a message vector and outputs a ciphertext vector.

The adversary runs in two stages. In both stages it is given an encryption oracle which takes as input $n(k)$ messages and outputs a ciphertext vector. At the end of the find stage the adversary outputs two vectors of n messages. In the guess stage the adversary gets as input a challenge ciphertext vector which is a ciphertext vector corresponding to a random choice of two vectors, and outputs its guess. We now provide a formal definition.

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 $b \in \{0, 1\}$ and a polynomial n define the experiments:

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

$(1^l, \mathbf{sk}', st) \xleftarrow{\$} B(\text{select}, k, n(\cdot))\;;\quad [\,1 \le l \le n(k)\;;\; |\mathbf{sk}'| = n(k) - l\,]$

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

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

$(\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}, st) \xleftarrow{\$} B^{\overline{\mathcal{SE}}_{\mathbf{sk}}(\cdot)}(\text{find}, st)$

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

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

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

$d \xleftarrow{\$} B^{\overline{\mathcal{SE}}_{\mathbf{sk}}(\cdot)}(\text{guess}, \mathbf{C}, st)$

Return d

It is required that $|M_0[i]| = |M_1[i]|$, and are in $\text{MsgSp}(k)$ for all $1 \le i \le n(k)$. We define the *advantage* $\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathsf{SE}},B}()$ of the adversary, IND-CPA security of the symmetric MRES analogously to the definitions for the asymmetric case described in Section 4. ▮

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

$sk \xleftarrow{\$} \mathcal{SK}(1^k)\;;\; M \xleftarrow{\$} \text{MsgSp}(k)\;;\; r \xleftarrow{\$} \text{Coins}_{\mathcal{SE}}(k)\;;\; C \xleftarrow{\$} \mathcal{SE}_{sk}(M, r)$

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

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. If $\mathcal{SE}$ is reproducible then for any RPTA B, there exists an RPTA A, such that

$$\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathsf{SE}},B,n(\cdot)}(k) \quad \le \quad n(k)\mathbf{Adv}^{\text{cpa}}_{\mathcal{SE},A}(k) \quad ▮$$

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 it's 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-BASES MRES. We recall CBC encryption scheme. The message space is a set of all strings whose length is multiple of s bits. The scheme uses a family of permutations $F : \{0,1\}^s \times \{0,1\}^k \to \{0,1\}^s$. F^{-1}

denotes the inverse permutation. A key-generation algorithm of $\mathcal{CBC}[F] = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ simply outputs a random k-bit string sk, which specifies a function $F(sk, \cdot)$ with a domain and range $\{0,1\}^s$. Usually F is a block cipher such as AES and $k = 128$. The encryption and decryption algorithms are defined as follows:

$\underline{\mathcal{SE}_{sk}(M)}$	$\underline{\mathcal{SD}_{sk}(C)}$
Parse M as $M_1, \ldots, M_p$,	Parse C as $C_0, \ldots, C_p$,
[s.t. $\lvert M_i \rvert = s$ for $1 \le i \le p$]	[s.t. $\lvert M_i \rvert = s$ for $0 \le i \le p$]
$C_0 \xleftarrow{\$} \{0,1\}^s$	For $i = 1, \ldots, p$ do
For $i = 1, \ldots, p$ do	$M_i \leftarrow F^{-1}(sk, C_i) \oplus C_{i-1}$
$C_i \leftarrow F(sk, C_{i-1} \oplus M_i)$	EndFor
EndFor	$M \leftarrow M_1 \Vert \ldots \Vert M_p$
Return $C_0 \Vert C_1 \Vert \ldots \Vert C_p$	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 $R(C_0 \Vert C_1 \Vert \ldots \Vert C_p, M', sk')$ 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 $\mathbf{Exp}^{\mathrm{repr}}_{\mathcal{CBC}[F],R}(k)$ will always output 1. ▌

The result of [BDJR] 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.

10 Secure Hybrid SM-MRES

Construction 10.1 Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ be an asymmetric MRES and let $\mathcal{SE} = (\mathcal{SK}, \mathcal{SE}, \mathcal{SD})$ be a symmetric encryption scheme. The single-message multi-recipient 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.

$\underline{\mathcal{HE}_{I,pk}(\mathbf{M})}$	$\underline{\mathcal{HD}_{I,sk}(C)}$
$K \xleftarrow{\$} \mathcal{SK}(1^k)$	Parse C as $C' \Vert C''$
For $i = 1, \ldots, n(k)$ do $\mathbf{K}[i] \leftarrow K$ EndFor	$K \leftarrow \mathcal{D}_{I,sk}(C')$
$\mathbf{C}' \xleftarrow{\$} \overline{\mathcal{E}}_{\mathbf{pk}}(I, \mathbf{K})$; $C'' \xleftarrow{\$} \mathcal{SE}_{\mathbf{K}[1]}(\mathbf{M}[1])$	$M \leftarrow \mathcal{SD}_{\mathbf{K}[1]}(C'')$
For $i = 1, \ldots, n(k)$ do $\mathbf{C}[i] \leftarrow \mathbf{C}'[i] \Vert C''$ EndFor	Return M
Return $\mathbf{C}$	

Note that the second part of $\mathbf{C}[i]$ 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 $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ 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 a SM-MRES constructed as per Construction 10.1. Then for any RPTA A there exist RPTAs B, C such that for $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$

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

The proof is in Appendix E.

References

[BPS] O. BAUDRON, D. POINTCHEVAL AND J. STERN, "Extended notions of security for multicast public key cryptosystems," *ICALP 2000.*

[ABR] M. ABDALLA, M. BELLARE AND P. ROGAWAY, "The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES," *CT-RSA 01, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed, Springer-Verlag, 2001.*

[BBM] M. BELLARE, A. BOLDYREVA AND S. MICALI, "Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements," *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.

[BBKS] M. BELLARE, A. BOLDYREVA, K. KUROSAWA AND J. STADDON "Multi-recipient encryption schemes: efficient construction and their security", Extended version of this paper. Available at `www.cc.gatech.edu/~aboldyre/publications.html`

[BBS] M. BELLARE, A. BOLDYREVA AND J. STADDON "Multi-recipient encryption schemes: efficient construction and their security", *Proceedings of the International workshop on practice and theory in Public Key Cryptography (PKC'03)*, 2003.

[BDJR] M. BELLARE, A. DESAI, E. JOKIPII AND P. ROGAWAY, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[BDPR] M. BELLARE, A. DESAI, D. POINTCHEVAL AND P. ROGAWAY, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[BG] M. BELLARE AND O. GOLDREICH, "On defining proofs of knowledge," *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.

[BKS] M. BELLARE, T. KOHNO AND V. SHOUP, "Stateful Public-Key Cryptosystems: How to Encrypt with One 160-bit Exponentiation," Proceedings of the ACM Conference on Computer and Communications Security (CCS), ACM, 2006.

[BR] M. BELLARE AND P. ROGAWAY, "Optimal asymmetric encryption – How to encrypt with RSA," *Advances in Cryptology – EUROCRYPT '94*, Lecture Notes in Computer Science Vol. 950, A. De Santis ed., Springer-Verlag, 1994.

[Ber] S. BERKOVITS, "How to Broadcast a Secret", *Advances in Cryptology – EUROCRYPT '91*, Lecture Notes in Computer Science Vol. 547, D. Davies ed., Springer-Verlag, 1991.

[BM] M. BLUM AND S. MICALI, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM J. on Computing* Vol. 13, No. 4, November 1984.

[B] D. BONEH. "Simplified OAEP for the RSA and Rabin Functions," *Advances in Cryptology – CRYPTO '01*, Lecture Notes in Computer Science Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.

[BF] D. BONEH AND M. FRANKLIN. "Identity-based encryption from the Weil Pairing," *Advances in Cryptology – CRYPTO '01*, Lecture Notes in Computer Science Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.

[CM] J. CAMENISCH AND M. MICHELS, "Confirmer signature schemes secure against adaptive adversaries," *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.

[C] R. CANETTI,, "Towards Realizing Random Oracles: Hash Functions that Hide All Partial Information,", *Advances in Cryptology – CRYPTO '97*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[CrSh] R. CRAMER AND V. SHOUP, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[CrSh2] R. Cramer and V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, SIAM Journal on Computing, Volume 33, Number 1, pp. 167-226 (2003)

[ElG] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol 31, 1985.

[FN] A. Fiat and M. Naor, "Broadcast Encryption", *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science Vol. 773, D. Stinson ed., Springer-Verlag, 1993.

[FO] E. Fujisaki, T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[FOPS] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern, "RSA-OAEP is Secure under the RSA Assumption," *Advances in Cryptology – CRYPTO '01*, Lecture Notes in Computer Science Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.

[GoMi] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, pp. 270–299, 1984.

[GGM] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions,"*Journal of the ACM*, Vol. 33, No. 4, 210–217, 1986.

[Hå] J. Håstad, "Solving simultaneous modular equations of low degree," *SIAM J. on Computing* Vol. 17, No. 2, April 1988.

[HILL] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, "A pseudorandom generation from any one-way function ," *SIAM Journal on Computing*, Vol. 28, No. 4, 1999.

[IL] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography," *Proceedings of the 30th Symposium on Foundations of Computer Science*, IEEE, 1989

[Ku] K. Kurosawa, "Multi-Recipient Public-Key Encryption with Shortened Ciphertext," *Proceedings of the International workshop on practice and theory in Public Key Cryptography (PKC'02)*, 2002.

[MRS] S. Micali, C. Rackoff and R. H. Sloan, "The notion of security for probabilistic cryptosystems," *Advances in Cryptology – CRYPTO '86*, Lecture Notes in Computer Science Vol. 263, A. Odlyzko ed., Springer-Verlag, 1986.

[NR] M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.

[NY] M.Naor and M.Yung, Universal One-Way Hash Functions and their Cryptographic Applications, STOC 1989, pp.33-43 (1989)

[Rom] J.Rompel: One-Way Functions are Necessary and Sufficient for Secure Signatures, STOC 1990, pp.387-394 (1990)

[RS] C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attack," *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.

[Sh] V. Shoup, "On formal models for secure key exchange, " Theory of Cryptography Library Record 99-12, `http://philby.ucsd.edu/cryptolib/`.

[St] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology – EUROCRYPT '96*, Lecture Notes in Computer Science Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[TY] Y. Tsiounis and M. Yung, "On the security of ElGamal based encryption," *Proceedings of the First International workshop on practice and theory in Public Key Cryptography (PKC'98)*, Lecture Notes in Computer Science Vol. 1431, H. Imai and Y. Zheng eds., Springer-Verlag, 1998.

[Y] A. C. Yao. "Theory and application of trapdoor functions," *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982.

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

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

This implies the statement of the Theorem 6.2. 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 $\mathbf{ExpH}_j(k)$ **[** $0 \leq j \leq n(k)$**]**

$I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$; $(1^l, st) \stackrel{\$}{\leftarrow} B(\text{select}, n(k), I)$
For $i = 1, \ldots, l$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \stackrel{\$}{\leftarrow} \mathcal{K}(I)$ EndFor
$(\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}, \mathbf{coins}, st) \stackrel{\$}{\leftarrow} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\text{find}, \mathbf{pk}, st)$
For $i = l+1, \ldots, n(k)$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \stackrel{\$}{\leftarrow} \mathcal{K}(I, \mathbf{coins}_{\mathcal{K}}(I)[i])$ EndFor
$\mathbf{pk} \leftarrow (\mathbf{pk}[1], \ldots, \mathbf{pk}[n(k)])$
If $j \leq l$
  then $\mathbf{M} \leftarrow (\mathbf{M}_0[1], \ldots, \mathbf{M}_0[j], \mathbf{M}_1[j+1], \ldots, \mathbf{M}_1[l], \mathbf{M}[l+1], \ldots, \mathbf{M}[n(k)])$
  else $\mathbf{M} \leftarrow (\mathbf{M}_1[0], \ldots, \mathbf{M}_0[l], \mathbf{M}[l+1], \ldots, \mathbf{M}[n(k)])$
EndIf
$\mathbf{C} \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{I,\mathbf{pk}}(\mathbf{M})$
$d \stackrel{\$}{\leftarrow} B(\text{guess}, \mathbf{C}, st)$
Return d

Let $P_j \stackrel{\text{def}}{=} \Pr\left[\, \mathbf{ExpH}_j(k) = 0 \,\right]$ 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

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

and after subtraction Equation (1) follows. We now justify the two equations above. In experiment $\mathbf{ExpH}_{n(k)}(k)$ we have $j = n(k)$ and a challenge ciphertext C is computed by encrypting the "left" vector of messages $\mathbf{M}_0$ 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 $\mathbf{ExpH}_0(k)$ 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 $A(\text{find}, I, pk)$

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

$st \leftarrow (j, l, \mathbf{pk}, \mathbf{sk}, \mathbf{M}_0, \mathbf{M}_1, \mathbf{M}, st)$
Return $(\mathbf{M}_0[j], \mathbf{M}_1[j], st)$
Adversary $A(\text{guess}, C, st)$
For $i \in \{1, \ldots, j-1, j+1, \ldots, n(k)\}$ do
If $i \leq j$ then $M' \leftarrow \mathbf{M}_0[i]$ else $M' \leftarrow \mathbf{M}_1[i]$ EndIf
$\mathbf{C}[i] \leftarrow R(I, pk, C, M', \mathbf{pk}[i], \mathbf{sk}[i])$
EndFor
$\mathbf{C}' \leftarrow (\mathbf{C}[1], \ldots, \mathbf{C}[j-1], C, \mathbf{C}[j+1], \ldots, \mathbf{C}[n(k)])$
$d \xleftarrow{\$} B(\text{guess}, \mathbf{C}', st')$
Return d

We claim that

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

Subtracting and exploiting the collapse of the sums we get

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

The statement of the theorem follows, so it remains to justify Equations (2). 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 a A's challenge ciphertext. And reproductivity 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 $\mathbf{ExpH}_j(k)$. Similarly, the experiment $\mathbf{Exp}^{\text{cpa}-1}_{\mathcal{AE},A}(k)$ is the same as $\mathbf{ExpH}_{j-1}(k)$.

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. ▌

B Proof of Theorem 7.3

Let A be an adversary attacking $\overline{\mathcal{EG}}$ scheme. We will design an adversary D for the DDH problem which we recalled in Definition 7.1 so that

$$\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) \;\geq\; \frac{1}{2} \cdot \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{EG}},B,n(\cdot)}(k) - \frac{1}{2^{k-1}}\;. \quad (3)$$

This implies the statement of Theorem 7.3. So it remains to specify D. We present the code for D in Figure 3.

We now proceed to analyze D. First consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above satisfy $T = g^{xy}$ where $X = g^x$ and $Y = g^y$ for some x, y in $\mathbb{Z}_q$. Using DDH random self-reducibility and its analysis done in [St, NR, Sh, BBM] we claim that for all $i \in 2, \ldots l$ the triples (X_i, Y, T_i) computed by D are also valid Diffie-Hellman triples and X_i, T_i are all uniformly and independently distributed over G_q. Thus $X_1, \ldots, X_l$ have the proper distribution of public keys. Since the second triple elements are equal all ciphertexts are computed using the same random string. Thus, the challenge vector of l ciphertexts together

Adversary $D(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$

$X_1 \leftarrow X$; $T_1 \leftarrow T$; $I \leftarrow (\tilde{\mathbb{G}}, 1^k, q, g)$; $pk_1 \leftarrow X_1$

$(l, st) \xleftarrow{\$} B(\text{select}, n(\cdot), I)$

For $i = 2, \dots l$ do

$v_i \xleftarrow{\$} \mathbb{Z}_q$; $w_i \xleftarrow{\$} \mathbb{Z}_q$; $X_i \leftarrow (X_1)^{w_i} \cdot g^{v_i}$; $T_i \leftarrow T_1^{w_i} \cdot Y^{v_i}$

$pk_i \leftarrow X_i$

EndFor

$(M_{1,0}, M_{2,0}, \dots, M_{l,0}, M_{1,1}, M_{2,1}, \dots, M_{l,1}, M_{l+1}, \dots, M_{n(k)},$

$pk_{l+1}, sk_{l+1}, \dots, pk_{n(k)}, sk_{n(k)}, st) \xleftarrow{\$} B(\text{find}, pk_1, \dots, pk_l, st)$

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

For $i = 1, \dots l$ do

$\mathbf{C}[i] \leftarrow (Y, T_i \cdot M_{i,b})$

EndFor

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

$\mathbf{C}[i] \leftarrow (Y, Y^{sk_i} \cdot M_i)$

EndFor

$\mathbf{C} \leftarrow \mathbf{C}[1], \dots, \mathbf{C}[n]$

Run $A(\text{guess}, \mathbf{C}, st)$

Eventually A halts

If it outputs a bit d and $b = d$ then return 1 else return 0

Figure 3: Adversary D for the proof of Theorem 7.3

---

with the $n - l$ ciphertexts are distributed exactly like a ciphertext in RR-MRES ElGamal scheme under public keys $pk_1, \dots pk_n$. We use it to see that for any k

$$\begin{aligned} \Pr\left[\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k) = 1\right] &= \frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\text{mr-cpa-0}}_{\mathcal{EG},B,n(\cdot)}(1^k) = 0\right] \\ &+ \frac{1}{2} \cdot \left(1 - \Pr\left[\mathbf{Exp}^{\text{mr-cpa-0}}_{\mathcal{EG},B,n(\cdot)}(1^k) = 0\right]\right) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\text{mr-cpa}}_{\mathcal{EG},B,n(\cdot)}(k) \ . \end{aligned} \tag{4}$$

Now consider $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$. In this case, the inputs X, Y, T to D above are all uniformly distributed over G_q. Clearly, for $1 \le i \le l$ X_i, T_i are all uniformly and independently distributed over G_q. Again, we have a proper distribution public keys for the ElGamal cryptosystem. But now $T_1, \dots, T_l$ are random elements in G_q and are independent of anything else. The rest $n - l$ ciphertexts cannot give any additional information to the adversary since A could compute them itself using Y and $x_{l+1}, \dots, x_n$. This means that the challenge ciphertext gives B no information about b, in an information-theoretic sense. We have

$$\Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k) = 1\right] \le \frac{1}{2} + \frac{1}{2^{k-1}} \ . \tag{5}$$

The last term accounts for the maximum probability that random inputs to D happen to have the distribution of a valid Diffie-Hellman triple. For any q this probability is less then $\frac{1}{2^{k-1}}$ since $2^{k-1} < q < 2^k$. Subtracting Equations 4 and 5 we get

$$\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(k) = \Pr\left[\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k) = 1\right]$$

$$\geq \quad \frac{1}{2} \cdot \mathbf{Adv}^{\text{mr-cpa}}_{\mathcal{EG},B,n(\cdot)}(k) - \frac{1}{2^{k-1}} \ ,$$

which is Equation (3).

It remains to specify D's running time. The overhead for D is that of performing at most $2n$ exponentiation operations with respect to a base element in G_q and an exponent in $\mathbb{Z}_q$ and $2n$ multiplication operations of the elements in G_q, which we can bound by $O(n(k)k^3)$, and that's the added cost in time of D.

C Proof of Theorem 7.6

The proof is very similar to that of the Cramer-Shoup scheme [CrSh2, Theorem 1]. We first change (6) and (9) of $\mathbf{Exp}^{\text{mr-cca-b}}_{\mathcal{AE},B,n(\cdot)}(k)$ as follows (see Def. 4.1).

(6') $b \stackrel{\$}{\leftarrow} \{0,1\}, \mathbf{M} \leftarrow (\mathbf{M}_b[1], \ldots, \mathbf{M}_b[l], \mathbf{M}[1], \ldots, \mathbf{M}[n(k)-l])$

(9') Return $d' = d \oplus b$.

This experiment is denoted by $\mathbf{Exp2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k)$. The advantage of an adversary B is defined as

$$\mathbf{Adv2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) = \Pr[\mathbf{Exp2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) = 0] - 1/2.$$

It is easy to see that

$$\mathbf{Adv}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) = 2\mathbf{Adv2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k).$$

More concretely, $\mathbf{Exp2}^{mr-cca}_{\mathcal{CS},B,n(\cdot)}(k)$ is described as follows.

$\mathbf{Exp2}^{mr-cca}_{\mathcal{CS},B,n(\cdot)}(k)$

(1) $I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$, where $I = (1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K)$;
  $(1^l, st) \stackrel{\$}{\leftarrow} B(\text{select}, n(k), I)$ **[** $1 \leq l \leq n(k)$**]**

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

(3) $(\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}, \mathbf{coins}, st) \stackrel{\$}{\leftarrow} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\text{find}, \mathbf{pk}, st)$
  [$|\mathbf{M}_0| = |\mathbf{M}_1| = l$; $|\mathbf{M}| = n(k) - l$**]**
  [$|\mathbf{pk}| = l$; $|\mathbf{coins}| = n(k) - l$**]**

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

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

(6) $b \stackrel{\$}{\leftarrow} \{0,1\}, \mathbf{M} \leftarrow (\mathbf{M}_b[1], \ldots, \mathbf{M}_b[l], \mathbf{M}[1], \ldots, \mathbf{M}[n(k)-l])$

(7) $\mathbf{C} \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{I,\mathbf{pk}}(\mathbf{M})$

(8) $d \stackrel{\$}{\leftarrow} B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\text{guess}, \mathbf{C}, st)$

(9) Return $d' = d \oplus b$,

where $\overline{\mathcal{E}}_{I,\mathbf{pk}}(\mathbf{M})$ can be written as

Parse I as $(1^k, \tilde{\mathbb{G}}, g_1, g_2, K)$
(E1) $r \stackrel{\$}{\leftarrow} \mathbb{Z}_q$
(E2) $u_1^* \leftarrow g_1^r$; $u_2^* \leftarrow g_2^r$
Parse $\mathbf{M}$ as $(M_1, \cdots, M_n)$.
For $i = 1, \cdots, n(k)$, do:
Parse $\mathbf{pk}[i]$ as (c_i, d_i, h_i)
(E3) $f_i^* \leftarrow h_i^r$
(E4) $e_i^* \leftarrow f_i^* M_i$
(E5) $\alpha_i^* \leftarrow \mathcal{EH}_K(u_1^*, u_2^*, e_i^*)$
(E6) $v_i^* \leftarrow (c_i^*)^r (d_i^*)^{r\alpha_i^*}$
Return $(u_1^*, u_2^*, e_1^*, v_1^*, \cdots, e_n^*, v_n^*)$

In the above, we let $\mathbf{pk}[i] = (c_i, d_i, h_i)$ for $1 \leq i \leq n(k)$. Similarly, let $\mathbf{sk}[i] = (x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}, z_{1,i}, z_{2,i})$ for $1 \leq i \leq n(k)$. Let w be such that $g_2 = g_1^w$.

Let $\mathbf{G}_0$ be $\mathbf{Exp2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k)$, and let T_0 be the event that $\mathbf{G}_0 = 0$, so that

$$\mathbf{Adv2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) = \Pr[T_0] - 1/2.$$

We shall define a sequence $\mathbf{G}_0, \mathbf{G}_1, \cdots, \mathbf{G}_5$ of modefied experiments, and let T_i be the event that $\mathbf{G}_i = 0$ for $1 \leq i \leq 5$.

In $\mathbf{G}_1$, we modify the encryption algorithm slightly in such a way that steps E3 and E6 are replaced by the corresponding part of the decryption algorithm:

(E3') $f_i^* \leftarrow (u_1^*)^{z_{1,i}} (u_2^*)^{z_{2,i}}$.

(E6') $v_i^* \leftarrow (u_1^*)^{x_{1,i}+y_{1,i}w} (u_2^*)^{x_{2,i}+y_{2,i}w}$.

This change is purely conceptual. It is clear that

$$\Pr[T_1] = \Pr[T_0]. \tag{6}$$

In $\mathbf{G}_2$, we further modify steps E1 and E2:

(E1') $r_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q, r_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_q \setminus \{r_1\}$.

(E2') $u_1^* \leftarrow g_1^{r_1}$; $u_2^* \leftarrow g_2^{r_2}$.

Under the DDH assumption, B cannot distinguish $\mathbf{G}_2$ from $\mathbf{G}_1$. More precisely, we have

Lemma C.1 There exists a probabilistic algorithm D, whose running time is essentially the same as that of the adversary B, such that

$$|\Pr[T_2] - \Pr[T_1]| \leq \mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + 3/q. \tag{7}$$

Proof: The proof is the same as that of [CrSh2, Lemma 5]. We show a distinguisher D for the DDH problem. D takes as an input $(1^k, \tilde{\mathbb{G}}, q, g, X, Y, T)$, where $X = g^w, Y = g^{r_1}, T = g^{wr_2}$ for some w, r_1, r_2. Our D executes the experiment $\mathbf{Exp2}^{mr-cca}_{\mathcal{CS},B,n(\cdot)}(k)$ with the following modification of step (1) and step (E2') using (g, X, Y, T).

(1') Let $I = (1^k, \tilde{\mathbb{G}}, q, g_1, g_2, K)$ be such that $g_1 = g, g_2 = X$.

(E2") $u_1^* \leftarrow Y$; $u_2^* \leftarrow T$.

Finally, D outputs d' of step (9) of the experiment.

It is easy to see that the above experiment is the same as $\mathbf{G}_1$ if (g, X, Y, T) is a DDH-tuple because $r_2 = r_1$ in this case. Hence we have

$$\Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-real}}(k) = 0\,\right] = \Pr[T_1].$$

On the other hand, the above experiment is the same as $\mathbf{G}_2$ if (g, X, Y, T) is a random tuple such that $r_2 \neq r_1$. Hence we have

$$\Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-rand}}(k) = 0 \mid r_2 \neq r_1\,\right] = \Pr[T_2].$$

Therefore, we have

$$\begin{aligned} |\Pr[T_2] - \Pr[T_1]| &= \left|\Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-rand}}(k) = 0 \mid r_2 \neq r_1\,\right] - \Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-real}}(k) = 0\,\right]\right| \\ &\leq \left|\Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-rand}}(k) = 0\,\right] - \Pr\left[\,\mathbf{Exp}_{\mathcal{G},D}^{\text{ddh-real}}(k) = 0\,\right]\right| + 3/q \\ &= \mathbf{Adv}_{\mathcal{G},D}^{\text{ddh}}(k) + 3/q, \end{aligned}$$

where the last inequality comes from [CrSh2, Lemma 1]. ▮

In $\mathbf{G}_3$, we modify the decryption oracles in $\mathbf{G}_2$. We replace steps D1 and D2 with:

(D1') Test if $u_2 = u_1^w$ and $v = {u_1}^{x+y\alpha}$; return $\perp$ and halt if this is not the case.

(D2') $f \leftarrow u_1^z$,

where $x = x_1 + x_2 w, y = y_1 + y_2 w, z = z_1 + z_2 w$.

Lemma C.2 Suppose that the adversary B makes $Q(k)$ decryption oracle queries in total. Then there exists a RPTA C, whose running time is essentially the same as that of B, such that

$$|\Pr[T_3] - \Pr[T_2]| \leq n(k)(\mathbf{Adv}_{\mathcal{H},\mathcal{C}}^{\text{tcr}}(k) + 1/q) + Q(k)/q. \tag{8}$$

The proof is almost the same as that of [CrSh2], and it is given in the next subsection.

In $\mathbf{G}_4$, we modify the encryption oracle in $\mathbf{G}_3$ slightly. We replace step E4 with:

(E4') $s_i \stackrel{\$}{\leftarrow} \mathbb{Z}_q$, $e_i^* \stackrel{\$}{\leftarrow} g_1^{s_i}$.

It is clear that

$$\Pr[T_4] = 1/2. \tag{9}$$

Lemma C.3

$$\Pr[T_4] = \Pr[T_3]. \tag{10}$$

Proof: In $\mathbf{G}_3$, before the challenge ciphertext $\mathbf{C}$ is given, all the information that B knows on $(z_{1,i}, z_{2,i})$ is $h_i = g_1^{z_{1,i}} g_2^{z_{2,i}}$ for each i. At step (E3'), f_i^* is computed as $f_i^* = (u_1^*)^{z_{1,i}}(u_2^*)^{z_{2,i}}$. ¿From these equations, we have

$$\begin{aligned} \log_{g_1} h_i &= z_{1,i} + w z_{2,i} \\ \log_{g_1} f_i^* &= r_1 z_{1,i} + r_2 w z_{2,i} \end{aligned}$$

The above two equations are linearly independent because $r_1 \neq r_2$. Hence there exists a bijection between f_i^* and a solution $(z_{1,i}, z_{2,i})$. This implies that f_i^* is random because $(z_{1,i}, z_{2,i})$ is randomly chosen. Therefore, $e_i^* = f_i^* M_i$ is also random. This means that $\Pr[T_4] = \Pr[T_3]$. ∎

¿From Equations (6) - (10), we obtain that

$$\mathbf{Adv2}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) \leq \mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + n(k)\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k) + (Q(k) + n(k) + 3)/q.$$

Therefore,

$$\begin{aligned} \mathbf{Adv}^{mr-cca}_{\mathcal{AE},B,n(\cdot)}(k) &\leq 2\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + 2n(k)\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k) + 2(Q(k) + n(k) + 3)/q \\ &\leq 2\mathbf{Adv}^{\mathrm{ddh}}_{\mathcal{G},D}(k) + 2n(k)\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{H},\mathcal{C}}(k) + 4(Q(k) + n(k) + 3)/2^k. \end{aligned}$$

C.1 Proof of Lemma C.2

Proposition C.4 [CrSh2, Lemma 4.] Let U_1, U_2 and F be events defined on some probability space. Suppose that the event $U_1 \wedge \neg F$ occurs if and only if $U_2 \wedge \neg F$ occurs. Then

$$|\Pr[U_1] - \Pr[U_2]| \leq \Pr[F].$$

Let R_3 be the event that in $\mathbf{G}_3$, the adversary B queries a ciphertext $C_i = (u_1, u_2, e, v)$ to some decryption oracle $\mathcal{O}_i$ such that C_i is rejected at step D1' but that would pass the test in step D1 of the decryption algorithm. It happens if and only if $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$ with $r_1 \neq r_2$ and

$$v = u_1^{x_{1,i} + y_{1,i}\alpha} u_2^{x_{2,i} + y_{2,i}\alpha},$$

where $\alpha = \mathcal{EH}_K(u_1, u_2, e)$. It is clear that $\mathbf{G}_2$ and $\mathbf{G}_3$ proceed identically until event R_3 occurs. In particular, the events $T_2 \wedge \neg R_3$ and $T_3 \wedge \neg R_3$ are identical. So by Proposition C.4, we have

$$|\Pr[T_3] - \Pr[T_2]| \leq \Pr[R_3]. \tag{11}$$

So it suffices to bound $\Pr[R_3]$ in order to prove Lemma C.2.

In $\mathbf{G}_4$, define the event R_4 in the same way as the event R_3 in $\mathbf{G}_3$. Then it is easy to see that

$$\Pr[R_4] = \Pr[R_3] \tag{12}$$

because $\mathbf{G}_3$ and $\mathbf{G}_4$ are identical as shown in the proof of Lemma C.3.

We next introduce $\mathbf{G}_5$, where $\mathbf{G}_5$ is the same as $\mathbf{G}_4$ except for the following special rejection rule. After step (7), if B queries a ciphertext $C_i = (u_1, u_2, e, v)$ to some decryption oracle $\mathcal{O}_i$ such that $(u_1, u_2, e) \neq (u_1^*, u_2^*, e_i^*)$ but $v = v_i^*$, then the decryption oracle immediately outputs **reject** and halts (before executing step D1').

In $\mathbf{G}_5$, we define an event R_5 in the same way as R_3 in $\mathbf{G}_3$. We also define C_5 as an event such that some decryption oracle rejects a ciphertext using the above special rejection rule.

It is clear that $\mathbf{G}_4$ and $\mathbf{G}_5$ proceed identically until event C_5 occurs. In particular, the events $R_4 \wedge \neg C_5$ and $R_5 \wedge \neg C_5$ are identical. So by Proposition C.4, we have

$$|\Pr[R_5] - \Pr[R_4]| \leq \Pr[C_5]. \tag{13}$$

Lemma C.5 There exists a probabilistic algorithm D, whose running time is essentially the same as that of the adversary B, such that

$$\Pr[C_5]/n(k) \leq \mathbf{Adv}^{\rm tcr}_{\mathcal{H},\mathcal{D}}(k) + 1/q. \tag{14}$$

Proof: The proof is almost the same as that of [CrSh2, Lemma 7]. We show an algorithm D which breaks the target collision resistance of $\mathcal{H}$. D takes as input a key K of $\mathcal{H}$ and a string $X_0 = (a^*, b^*, e^*) \in \mathbb{G}^3$. Then our D executes $\mathbf{Exp2}^{mr-cca}_{\mathcal{CS},B,n(\cdot)}(k)$ with the following modification of step (E2') and (E4') using $X_0 = (a^*, b^*, e^*)$.

E2''' Let $u_1^* \leftarrow a^*$ and $u_2^* \leftarrow b^*$.

E4'' This step is the same as (E4') except for the following modification. Choose $1 \leq i \leq n(k)$ randomly, and let $e_i^* \leftarrow e^*$ for this i.

If the decryption oracle $\mathcal{O}_i$ (with the above index i) invokes the special rejection rule for a ciphertext $C_i = (u_1, u_2, e, v)$, then D immediately outputs $X_1 = (u_1, u_2, e)$ and halts. Otherwise, D the aborts.

It is easy to see that if (a^*, b^*, e^*) is sampled randomly in such a way that $\log_{g_1} a^* \neq \log_{g_1} b^*$, then D succeeds in finding a collision with probability $\Pr[C_5]/n(k)$. On the other hand, in the definition of $\mathbf{Adv}^{\rm tcr}_{\mathcal{H},\mathcal{D}}(k)$, the input is sampled from the uniform distribution over $\mathbb{G}^3$. Eq.(14) follows from this statistical difference. ▌

Lemma C.6 Suppose that the adversary makes $Q(k)$ decryption queries in total. Then we have

$$\Pr[R_5] \leq Q(k)/q. \tag{15}$$

Proof: The proof is obtained in the say way as that of [CrSh2, Lemma 8]. This is because the above probability is essentially taken over $\mathbf{sk}[i] = (x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i})$ for all i, and $\mathbf{sk}[i]$ is independently chosen for each i. Hence we can apply the proof technique of [CrSh2, Lemma 8] to each $\mathbf{sk}[i] = (x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i})$ independently. ▌

Inequality (8) now follows immediately from Equations (11) - (15).

D Proof of Theorem 8.3

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

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

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 $\{0,1\}^{il} \rightarrow \{0,1\}^{ol}$. Let A be an RPTA adversary attacking the security of the multi-recipient scheme $\overline{\mathcal{AE}'[F]}$. We will construct a RPT adversary D which attacks F as a pseudorandom function family and an adversary B which attacks the security of $\mathcal{AE}$ such that

$$\mathbf{Adv}^{\mathrm{prf}}_{F,D}(k) = \frac{1}{2} \cdot (\mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[F]},A,n(\cdot)}(k) - \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k)) \tag{16}$$

$$\mathbf{Adv}^{\mathrm{cpa}}_{\mathcal{AE},B,n(\cdot)}(k) \geq \frac{1}{n(k)} \cdot \mathbf{Adv}^{\text{mr-cpa}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k) \tag{17}$$

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 $g\colon \{0,1\}^{il} \rightarrow \{0,1\}^{ol}$. Here is the algorithm for D.

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

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

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

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

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

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

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

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

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

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

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

$d \xleftarrow{\$} A(\text{guess}, \mathbf{C}, st)$

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

Above $\overline{\mathcal{E}'}^{g(\cdot)}_{\mathbf{pk}}$ denotes the procedure which substitutes all applications of $F(r', \cdot)$ in $\overline{\mathcal{E}'}_{\mathbf{pk}}(\cdot)$with an application of $g(\cdot)$.

We now analyze the adversary. We claim that

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

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

The above 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 it's view in $\mathbf{Exp}^{\text{mr-cpa-b}}_{\overline{\mathcal{AE}'[R]},A,n(\cdot)}(k)$. After subtraction we get Equation (16).

We now prove Equation (17). Let A be an adversary which attacks the security of $\overline{\mathcal{AE}'[R]}$. We will use the hybrid experiments $\mathbf{ExpH}_j(k)$ for $0 \leq j \leq n(k)$ we defined in the proof of Theorem 6.2, which are associated to A and the encryption scheme $\overline{\mathcal{AE}'[R]}$. Let $P_j \stackrel{\text{def}}{=} \Pr\left[\, \mathbf{ExpH}_j(k) = 0 \,\right]$ for $j = 0, 1, \ldots, n(k)$.

Similarly to the proof of Theorem 6.2 we claim that

$$\mathbf{Adv}^{\text{mr-cpa}}_{\mathcal{AE}'[R],A,n(\cdot)}(k) \;=\; P_{n(k)} - P_0 \;. \tag{18}$$

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

Adversary $B(\text{find}, I, pk)$
$(l, st') \xleftarrow{\$} A(\text{select}, n(k), I)$; $j \xleftarrow{\$} \{1, \ldots, n(k)\}$
If $j \le l$ then For $i \in \{1, \ldots, j-1, j+1, \ldots, l\}$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$; $\mathbf{pk}[j] \leftarrow pk$ EndFor
else For $i = 1, \ldots l$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathcal{K}(I)$ EndFor EndIf
$(\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}, \mathbf{coins}, st) \xleftarrow{\$} A(\text{find}, \mathbf{pk}, st)$
For $i = l+1, \ldots, n(k)$ do $\mathbf{M}_0[i] \leftarrow \mathbf{M}[i]$; $\mathbf{M}_1[i] \leftarrow \mathbf{M}[i]$ EndFor
$st \leftarrow (I, j, l$; $\mathbf{pk}, \mathbf{sk}, \mathbf{M}_0, \mathbf{M}_1$; $st')$
Return $(\mathbf{M}_0[j], \mathbf{M}_1[j], st)$
Adversary $B(\text{guess}, C, st)$
For $i \in \{1, \ldots, j-1, j+1, \ldots, n(k)\}$ do
If $\mathbf{pk}[i] = pk$ then $M \leftarrow \mathcal{D}_{I,sk_i}(C)$; If $M = \mathbf{M}_0[j]$ then Return 0 else Return 1
Else
If $\exists p\colon 1 \le p < i, \mathbf{pk}[p] = \mathbf{pk}[i]$ 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 $\mathbf{C}[i] \leftarrow \mathcal{E}_{I,\mathbf{pk}[i]}(\mathbf{M}_0[i], r_i)$
For $i = j+1, \ldots, n$ do $\mathbf{C}[i] \leftarrow \mathcal{E}_{\mathbf{pk}[i]}(\mathbf{M}_{1[i]}, r_i)$
$C_j \leftarrow C$; $d \xleftarrow{\$} A(\text{guess}, \mathbf{C}, st')$
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 $\mathbf{ExpH}_j(k)$. This is true since the only potential difference among these experiments from A's view is how the values r_i used as coin tosses for $\mathcal{E}_{I,pk_i}$ are computed. In the experiment $\mathbf{ExpH}_j(k)$ 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 $\mathbf{ExpH}_j(k)$, since for them to be the same B should compute the component of $\mathbf{C}$ 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

$$\Pr\left[\,\mathbf{Exp}^{\text{cpa}-0}_{\mathcal{AE},B}(1^k) = 0\,\right] \;\ge\; \frac{1}{n(k)} \cdot \sum_{j=1}^{n(k)} P_j \quad \text{and} \quad \Pr\left[\,\mathbf{Exp}^{\text{cpa}-1}_{\mathcal{AE},B}(1^k) = 0\,\right] \;\le\; \frac{1}{n(k)} \cdot \sum_{j=1}^{n(k)} P_{j-1} \;. \tag{19}$$

Subtracting and exploiting the collapse of the sums we get

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

The above implies Equation (17).

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 $1 \leq i \leq l$. 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 one of A plus the time to pick a number $j \leq n(k)$ at random. The running time of D is one of A. ▮

E Proof of Theorem 10.2

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

Experiment $\mathbf{ExpH}^{m\text{-atk}}_{\mathcal{HS},A_{\text{atk}}}(k)$ **[** $m \in \{1,2,3,4\}$; $\text{atk} \in \{\text{cpa},\text{cca}\}$**]**
$I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$; For $i = 1, \ldots, n(k)$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \stackrel{\$}{\leftarrow} \mathcal{K}(I)$ EndFor
$(1^l, st) \stackrel{\$}{\leftarrow} A_{\text{atk}}(\text{select}, n(k), I)$; If $l \neq n(k)$ then abort EndIf
$(\mathbf{M}_0, \mathbf{M}_1, st) \stackrel{\$}{\leftarrow} A_{\text{atk}}(\text{find}, \mathbf{pk}, st)$
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
$K \stackrel{\$}{\leftarrow} \mathcal{SK}(k)$; $K' \stackrel{\$}{\leftarrow} \mathcal{SK}(k)$
If $l = 1$ or $l = 2$ then $C_1 \stackrel{\$}{\leftarrow} \mathcal{SE}_K(\mathbf{M}_0[1])$ EndIf
If $l = 3$ or $l = 4$ then $C_1 \stackrel{\$}{\leftarrow} \mathcal{SE}_K(\mathbf{M}_1[1])$ EndIf
For $i = 1, \ldots, n(k)$ do
If $m = 1$ or $m = 4$ then $\mathbf{C}_0 \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{I,pk}(K)$ EndIf
If $m = 2$ or $m = 3$ then $\mathbf{C}_0 \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{I,pk}(K')$ EndIf
$\mathbf{C} \leftarrow \mathbf{C}_0[i] \| C_1$
EndFor
If $\text{atk} = \text{cca}$ and A_{cca} during find stage makes a decryption oracle query $\mathbf{C}'$ to oracle $\mathcal{HD}_{I,sk_i}(\cdot)$
$M \leftarrow \mathcal{HD}_{I,sk}(\mathbf{C}'[1])$ EndIf
return M to A_{cca}
EndIf
$d \stackrel{\$}{\leftarrow} A_{\text{atk}}(\text{guess}, \mathbf{C}, st)$
If $\text{atk} = \text{cca}$ and A_{cca} during guess stage makes a decryption oracle query $\mathbf{C}'$ to oracle $\mathcal{HD}_{I,sk_i}(\cdot)$:
If $m = 1$ or $m = 4$ then $M \leftarrow \mathcal{HD}_{I,sk}(\mathbf{C}'[1])$ EndIf
If $m = 2$ or $m = 3$ then parse $\mathbf{C}'[1]$ as $C'_0 \| C'_1$
If $C'_0 = \mathbf{C}_0[1]$ then $M \leftarrow \mathcal{SD}_K(C'_1)$ else $M \leftarrow \mathcal{HD}_{I,sk}(\mathbf{C}'[1])$ EndIf
EndIf
return M to A_{cca}
EndIf
Return d

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

$$\mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{HS},A,n(\cdot)}(k) \;=\; P_4^{\text{atk}} - P_1^{\text{atk}} = (P_4^{\text{atk}} - P_3^{\text{atk}}) + (P_3^{\text{atk}} - P_2^{\text{atk}}) + (P_2^{\text{atk}} - P_1^{\text{atk}})\;. \qquad (20)$$

Adversary $B_1(\text{select}, n(\cdot), I)$
$(1^l, st') \leftarrow A_{\text{cca}}(\text{select}, n(\cdot), I)$; If $l \neq n(k)$ then abort EndIf
Return $(n(k), st')$
Adversary $B_1^{\mathcal{D}_{sk_1}(\cdot),\ldots,\mathcal{D}_{sk_{n(k)}}(\cdot)}(\text{find}, \mathbf{pk}, st)$
$K \xleftarrow{\$} \mathcal{SK}(k)$; $K' \xleftarrow{\$} \mathcal{SK}(k)$
Run $A_{\text{cca}}^{\mathcal{HD}_{sk_1}(\cdot)\ldots\mathcal{HD}_{sk_{n(k)}}(\cdot)}(\text{find}, \mathbf{pk}, st')$
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 $C'_0 \| C'_1$; $K'' \leftarrow \mathcal{D}_{I,sk_i}(C'_0)$; $M \leftarrow \mathcal{SD}_{K''}(C'_1)$
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 $B_1^{\mathcal{D}_{sk_1}(\cdot),\ldots,\mathcal{D}_{sk_{n(k)}}(\cdot)}(\text{guess}, \mathbf{C}_0, st)$
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 $A_{\text{cca}}^{\mathcal{HD}_{sk_1}(\cdot),\ldots,\mathcal{HD}_{sk_{n(k)}}(\cdot)}(\text{find}, \mathbf{C}, st')$ as follows
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

Figure 4: The adversary for the proof of Claim E.1

We now claim that

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

$$P_4^{\text{atk}} - P_3^{\text{atk}} \;\leq\; \mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{AE},B_1,n(\cdot)}(k)\ .$$

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

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

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

$$P_2^{\text{atk}} - P_1^{\text{atk}} \;\leq\; \mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{AE},B_2,n(\cdot)}(k)\ .$$

For a fixed $k \in \mathbb{N}$, if $\mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{AE},B_1,n(\cdot)}(k) \geq \mathbf{Adv}^{\text{smmr-atk}}_{\mathcal{AE},B_2,n(\cdot)}(k)$ then define an adversary $B = B_1$ and $B = B_2$ otherwise. Then the statement of the theorem follows from Equation (20) and Claim E.1, Claim E.2, Claim E.3. It remains to prove the latter claims. ∎

Proof of Claim E.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 an adversary B_1 in Figure E.

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)

Adversary $C^{\mathcal{SD}_K(\cdot)}(\text{find}, k)$

$I \stackrel{\$}{\leftarrow} \mathcal{G}(1^k)$; For $i = 1 \ldots n(k)$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \stackrel{\$}{\leftarrow} \mathcal{K}(I)$ EndFor

$K' \stackrel{\$}{\leftarrow} \mathcal{SK}(k)$; $(1^l, st') \stackrel{\$}{\leftarrow} A_{\text{cca}}(\text{select}, I)$

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

$(\mathbf{M}_0, \mathbf{M}_1, st') \stackrel{\$}{\leftarrow} {A_{\text{cca}}}^{\mathcal{HD}_{sk_1}(\cdot) \ldots \mathcal{HD}_{sk_{n(k)}}(\cdot)}(\text{find}, \mathbf{pk}, st')$ **[** B_1 answers A_{cca}'s decryption queries using $sk_1, \ldots, sk_{n(k)}$**]**

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

$\mathbf{C}_0 \stackrel{\$}{\leftarrow} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$; $C_1 \stackrel{\$}{\leftarrow} \mathcal{SE}_K(\mathbf{M}_0[1])$; $st \leftarrow (\mathbf{pk}, \mathbf{C}_0, K', st')$

Return $(\mathbf{M}_0[1], \mathbf{M}_1[1], st)$

Adversary $C^{\mathcal{D}_{sk}(\cdot)}(\text{guess}, C_1, st)$

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 $\mathcal{HD}_{I,sk_i}(\cdot)$ **[** $1 \leq i \leq n(k)$ **]**

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

Figure 5: The adversary for the proof of Claim E.2

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

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

and that B_1 runs in polynomial time. ▌

Proof of Claim E.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 Figure E.

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

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

and that C runs in polynomial time. ▮

Proof of Claim E.3: The proof is similar to the proof of Claim E.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 E.1 outputs (K', K). ▮