A preliminary version of this paper, entitled "Randomness re-use in multi-recipient encryption schemes," appeared in *Public-Key Cryptography '03*, Lecture Notes in Computer Science Vol. 2567 , Y. Desmdedt ed., Springer-Verlag, 2003. This is the full version.

Multi-Recipient Encryption Schemes: Security Notions and Randomness Re-Use

Mihir Bellare* Alexandra Boldyreva† Jessica Staddon‡

March 18, 2003

Abstract

This paper begins by refining Kurosawa's [Ku] definitions of security for multi-recipient encryption schemes (MRESs). It then considers a subclass of MRESs, that are formed by transforming standard encryption schemes via a natural technique called randomness re-use, and that offer important performance benefits. The main result is a way to avoid ad-hoc analyses of such schemes: we provide a general test that can be applied to a standard encryption scheme to determine whether the associated randomness re-using MRES is secure. This is applied to identify numerous specific secure and efficient randomness re-using MRESs. The results and applications cover both asymmetric and symmetric encryption.

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

*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. Supported in part by NSF Grant CCR-0098123 and NSF Grant ANR-0129617.

†Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. aboldyre@cs.ucsd.edu. http://www-cse.ucsd.edu/users/aboldyre. Supported in part by grants of the other authors and by a SDSC Graduate Student Diversity Fellowship.

‡CSL, Palo Alto Research Center, 3333 Coyote Hill Rd. Palo Alto, CA 94304, USA. staddon@parc.com. Partially supported by DARPA grant N66001-00-1-8921.

Contents

1 Introduction

We begin by introducing the notion of multi-recipient encryption schemes and recalling a motivating example. We then proceed to discuss our contributions.

1.1 Multi-recipient encryption schemes

The setting of standard encryption is the following. A receiver has generated for itself a secret decryption key sk and corresponding public encryption key pk.[1] The sender applies an encryption algorithm $\mathcal{E}$ to pk and a message M to obtain a ciphertext C. The receiver can apply to sk and C a decryption algorithm that recovers M.

The setting of multi-recipient encryption is the following. 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). It is worth noting 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.

There is of course a naive, or obvious way to build a MRES: 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, viewing the task of producing multiple ciphertexts as being done by a single process allows one to explore reductions in cost that might arise from batching. To exemplify this let us consider an example due to Kurosawa [Ku].

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 El Gamal based MRES is the following: Pick $r_1, \ldots, r_n$ independently at random from $\mathbb{Z}_q$ and let $C_i = (g^{r_i}, g^{x_i r_i} \cdot M_i)$ for $1 \leq i \leq n$. Instead, Kurosawa [Ku] suggests that one pick just one r at random from $\mathbb{Z}_q$ and set $C_i = (g^r, g^{x_i r} \cdot M_i)$ for $1 \leq i \leq n$.

Kurosawa points out that his MRES brings two benefits compared to the naive one. 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, his suggested scheme (approximately) halves the computational cost (number of exponentiations) for encryption as compared to the naive method. Kurosawa also notes a MRES derived in a similar way from the Cramer-Shoup encryption scheme [CrSh].

1.2 Security notions for MRESs

The above example 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.

First steps towards this end were taken by Kurosawa, who proposed definitions that are simple adaptations of the definitions of privacy for *standard* encryption schemes in the multi-user setting as given in [BPS, BBM]. The first contribution of our work is to point to weaknesses in Kurosawa's

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

model and provide new definitions of security. Full definitions are in Section 4. Here we expand briefly on some of the underlying issues.

Kurosawa's model assumes the adversary is an outsider. But the adversary might be 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 setting of [BPS, BBM], 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. To illustrate the power of insider attacks, we present in Section 4 a variant of Kurosawa's scheme that is provably secure in his model but insecure in our model.

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.) Such attacks can be extremely damaging, as we illustrate in Section 4 with a rogue-key attack that breaks Kurosawa's above-mentioned El Gamal based MRES. It is important to be aware of such attacks, but it is for such reasons that certification authorities require (or should require) that a user registering a public encryption key prove knowledge of the corresponding secret decryption key. (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 a valid corresponding decryption key.

1.3 Randomness re-using MRESs and the reproducibility theorem

Having defined security for MRESs, we want to assess the security of Kurosawa's schemes and also to find new, secure MRESs that provide cost reductions compared to the naive ones. Rather than proceed in an ad hoc manner, we provide general paradigms towards these ends. We introduce a subclass of MRESs that include all Kurosawa's MRESs and provide a simple way to test whether or not a MRES from this subclass is secure. Let us first expand on these items and then turn to applications.

Randomness re-using MRESs. 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 $(1 \leq i \leq n)$. The corresponding MRES is called the *randomness re-using* MRES (RR-MRES) associated to the underlying standard encryption scheme.

Note that Kurosawa's El Gamal based MRES is the RR-MRES associated to the El Gamal scheme, and his Cramer-Shoup based MRES is the RR-MRES associated to the Cramer-Shoup encryption scheme. Later we will see other examples that offer cost benefits. RR-MRESs are the subclass of MRESs on which we focus.

The reproducibility test and theorem. 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 embedding schemes such as PKCS#1.) 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 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.

1.4 Secure and efficient MRESs

We now use the above to identify various secure MRESs that offer some cost benefits. These applications cover asymmetric, symmetric and hybrid schemes.

El Gamal and Cramer-Shoup. The associated RR-MRESs are those of Kurosawa [Ku], of interest because, as noted previously, they permit reductions in both computation and broadcast ciphertext size. Kurosawa proved these MRESs secure under the DDH (Decisional Diffie-Hellman) assumption but, as noted above, his target notion of security is weak. Thus one needs to ask whether the schemes remain secure under our stronger notion of security.

We show that the base El Gamal and Cramer-Shoup schemes are both reproducible. Our reproducibility theorem then implies that indeed, Kurosawa's MRESs remain secure with respect to our more stringent security notions.

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, Ku], using self-reducibility properties of the DDH problem [St, NR, Sh].

CBC encryption. A novel element of our work compared to [Ku, BBM, BPS] is consideration of the symmetric setting. We show that reproducibility and the corresponding theorem apply in this setting too.

We consider 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 re-using randomness allows to reduce the length of the broadcast ciphertext by 50%. With regard to security, 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.

DHIES. This is a Diffie-Hellman based asymmetric encryption scheme adopted by draft standards ANSI X9.63EC and IEEE P1363a. It has El Gamal-like cost in public-key operations while

achieving Cramer-Shoup-like security (IND-CCA), although the proof [ABR] relies on significantly stronger assumptions than the DDH assumption used in [CrSh]. Unlike El Gamal 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 El Gamal scheme, so it is important to assess security.

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. 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.

Pairings-based escrow El Gamal. Boneh and Franklin [BF] introduced an El Gamal like scheme with global escrow capabilities, based on the Weil pairing. We show that this scheme is reproducible. Our reproducibility theorem coupled with the result of [BF] then implies that the corresponding randomness re-using multi-recipient scheme is IND-CPA in the random oracle model under the Bilinear Diffie-Hellman assumption. Our reproducibility algorithm exploits properties of the Weil pairing. Again, as for El Gamal scheme, re-using randomness permits computational and bandwidth savings.

1.5 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.

1.6 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.

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 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 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 randomized *common-key generation* algorithm $\mathcal{G}$ takes as input a security parameter $k \in \mathsf{N}$ and, in $\mathrm{poly}(k)$ time, returns a *common key* I; we write $I \xleftarrow{R} \mathcal{G}(k)$.
- The randomized *key generation* algorithm $\mathcal{K}$ takes as input the common key I and, in $\mathrm{poly}(k)$-time, returns a pair (pk, sk) consisting of a public key and a corresponding secret key; we write $(pk, sk) \xleftarrow{R} \mathcal{K}(I)$.
- The randomized *encryption* algorithm $\mathcal{E}$ takes input a public key pk and a plaintext M and, in $\mathrm{poly}(k)$-time, returns a ciphertext; we write $C \xleftarrow{R} \mathcal{E}_{pk}(M)$.
- The deterministic, *decryption* algorithm $\mathcal{D}$ takes the secret key sk and a ciphertext C to return in $\mathrm{poly}(k)$-time the corresponding plaintext or a special symbol $\perp$ indicating that the ciphertext was invalid; we write $x \leftarrow \mathcal{D}_{sk}(C)$.

Associated to each common key I is a *message space* $\mathrm{MsgSp}(I)$ from which M is allowed to be drawn. We require that $\mathcal{D}_{sk}(\mathcal{E}_{pk}(M)) = M$ for all $M \in \mathrm{MsgSp}(I)$. We will use the terms "plaintext" and "message" interchangeably.

In our context it is important to make explicit the random choices underlying the randomized encryption algorithm $\mathcal{E}$. The notation $C \xleftarrow{R} \mathcal{E}_{pk}(M)$ is thus shorthand for $r \xleftarrow{R} \mathrm{Coins}_{\mathcal{E}}(I, pk)$; $C \leftarrow \mathcal{E}_{pk}(M; r)$, where $\mathrm{Coins}_{\mathcal{E}}(I, pk)$ is a set from which $\mathcal{E}$ draws its coins.

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.

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 $A_{\text{cpa}}, A_{\text{cca}}$ be adversaries such that the latter has access to an oracle, $\mathcal{D}_{sk}(\cdot)$. Let I be some initial information string. For $b = 0, 1$ define the experiments

Experiment $\mathbf{Exp}^{\text{cpa}-b}_{\mathcal{AE},A_{\text{cpa}}}(k)$	**Experiment** $\mathbf{Exp}^{cca-b}_{\mathcal{AE},A_{\text{cca}}}(k)$
$I \xleftarrow{R} \mathcal{G}(k)$; $(pk, sk) \xleftarrow{R} \mathcal{K}(I)$	$I \xleftarrow{R} \mathcal{G}(k)$; $(pk, sk) \xleftarrow{R} \mathcal{K}(I)$
$(m_0, m_1, st) \leftarrow A_{\text{cpa}}(\text{find}, I, pk)$	$(m_0, m_1, st) \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk}(\cdot)}(\text{find}, I, pk)$
$C \xleftarrow{R} \mathcal{E}_{pk}(m_b)$	$C \xleftarrow{R} \mathcal{E}^r_{pk}(m_b)$
$d \leftarrow A_{\text{cpa}}(\text{guess}, C, st)$	$d \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk}(\cdot)}(\text{guess}, C, st)$
Return d	Return d

It is mandated that $|m_0| = |m_1|$ above. We require that A_{cca} does not make oracle query C in the guess stage. For $\text{atk} \in \{\text{cpa}, \text{cca}\}$ we define the *advantages* of the adversaries via as follows:

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

The scheme $\mathcal{AE}$ is said to be *polynomially-secure against chosen-plaintext attack or IND-CPA secure (resp. chosen-ciphertext attack or IND-CCA secure)* if the function $\mathbf{Adv}^{\text{cpa}}_{\mathcal{AE},A_{\text{cpa}}}(\cdot)$ (resp. $\mathbf{Adv}^{\text{cca}}_{\mathcal{AE},A_{\text{cca}}}(\cdot)$) is negligible for any adversary of $\text{poly}(k)$ time-complexity. ▌

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

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 randomized *multi-encryption* algorithm $\overline{\mathcal{E}}$ takes input 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, in $\text{poly}(k)$-time, returns a *ciphertext vector* $\mathbf{C} = (\mathbf{C}[1], \ldots, \mathbf{C}[n])$; we write $\mathbf{C} \xleftarrow{R} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$. Associated to each common key I is a *message space* $\text{MsgSp}(I)$ from which the components of $\mathbf{M}$ are allowed to be drawn. We require that for all $\mathbf{M}$ with components in the message space, the following experiment returns 1 with probability 1:

For $i = 1, \ldots, n$ do $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{R} \mathcal{K}(k)$ EndFor; $\mathbf{C} \xleftarrow{R} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$
$i \xleftarrow{R} \{1, \ldots, n\}$
If $\mathcal{D}_{\mathbf{sk}[i]}(\mathbf{C}[i]) = \mathbf{M}[i]$ then return 1 else return 0

The notation $\mathbf{C} \xleftarrow{R} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$ is shorthand for $r \xleftarrow{R} \mathrm{Coins}_{\overline{\mathcal{E}}}(I, \mathbf{pk})$; $\mathbf{C} \leftarrow \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M}; r)$, where $\mathrm{Coins}_{\overline{\mathcal{E}}}(I, \mathbf{pk})$ is a set from which $\mathcal{E}$ draws its coins.

We are interested in a specific class of MRESs, those obtained from a given asymmetric encryption scheme by using the same coins to encrypt the different messages in the message vector.

Construction 3.1 The *randomness-reusing MRES (RR-MRES)* associated to a given asymmetric encryption scheme $\mathcal{AE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is the 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:

$\overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$
Let n be the number of components of $\mathbf{M}$ [and also of $\mathbf{pk}$]
$r \xleftarrow{R} \mathrm{Coins}_{\mathcal{E}}(I, \mathbf{pk})$
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 $\overline{\mathcal{AE}}$. ▌

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. 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 n be a polynomial. Let B be an adversary attacking $\overline{\mathcal{AE}}$. B runs in three stages. In the select stage the adversary is given an initial information string and outputs l such that $1 \leq l \leq n$, which indicates that it wants to corrupt $n - l$ users, assumed without loss of generality to be users $l + 1, \ldots, n$. In the findstage the adversary is given I 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* $(n-l)$-vector of messages corresponding to choices for the corrupted users; a $(n-l)$-vector of public keys for the corrupted users; and a $(n - l)$-vector of corresponding secret keys (see the discussion below.) Based on a challenge bit b, one of the two l-vectors is selected, and the components of the $(n - l)$-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 guessstage. 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, let n be a polynomial. For atk $\in \{\text{cpa}, \text{cca}\}$ and $b \in \{0,1\}$ consider the experiment:

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

$I \xleftarrow{R} \mathcal{G}(k)$; $l \leftarrow B(\text{select}, I)$ $\quad[1 \le l \le n(k)]$

For $i = 1, \ldots, l$ do $(pk_i, sk_i) \xleftarrow{R} \mathcal{K}(I)$ EndFor

$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\ ;\ m_{1,1}, m_{2,1}, \ldots, m_{l,1}\ ;\ m_{l+1}, \ldots, m_{n(k)}\ ;\ pk_{l+1}, sk_{l+1}, \ldots, pk_{n(k)}, sk_{n(k)}\ ;\ st)$
$\quad \leftarrow B^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_l(\cdot)}(\text{find}, I, pk_1, \ldots, pk_l)$

$\mathbf{pk} \leftarrow (pk_1, \ldots, pk_{n(k)})$

$\mathbf{M} \leftarrow (m_{1,b}, \ldots, m_{l,b}, m_{l+1}, \ldots, m_{n(k)})$

$\mathbf{C} \xleftarrow{R} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$

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

Return d

Above, the oracles are defined as follows for $1 \le i \le l$: If atk = cpa then $\mathcal{O}_i(\cdot) = \varepsilon$ and if atk = cca then $\mathcal{O}_i(\cdot) = \mathcal{D}_{sk_i}(\cdot)$. It is mandated that for all $1 \le i \le l$ we have $|m_{i,0}| = |m_{i,1}|$ and also that if atk = 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

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

Let $\overline{\mathcal{AE}} = (\mathcal{G}, \mathcal{K}, \overline{\mathcal{E}}, \mathcal{D})$ be a multi-recipient encryption scheme. We say that it is IND-CPA (resp. IND-CCA) secure if the function $\mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathcal{AE}},B}(\cdot)$ (respectively $\mathbf{Adv}^{n\text{-mr-cca}}_{\overline{\mathcal{AE}},B}(\cdot)$) is negligible for any poly(k)-time adversary B and any polynomial n. ▌

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) secure (or, briefly RRS) if the RR-MRES $\overline{\mathcal{AE}}$ associated to $\mathcal{AE}$ is IND-CPA (resp. IND-CCA) secure. ▌

Discussion and comparison with the model of security of [Ku]. Previous works [BPS, BBM, Ku] only considered outsider attacks, meaning the adversary was not one of the receivers. A novel element of our model relative to [BPS, BBM, Ku] is the consideration of insider attacks. The adversary is allowed to corrupt some fraction of the users and choose secret and public keys for them.

We argue that it is necessary for a model of security of multi-recipient schemes to take into account insider attacks. The model of [Ku] does not address this problem and we show that there exist MRESs which can be proven secure using the model of [Ku] but are obviously insecure and can easily be shown insecure using our model of security.

It is proved in [Ku] that El Gamal scheme permits secure randomness re-use in the multi-recipient setting. Now consider a modified encryption scheme which differs from El Gamal in that its encryption algorithm when invoked on one particular public key (e.g. g^3) in addition to a

ciphertext returns randomness used to compute it. Assume this fact is known to the adversary. When this scheme used in a multi-recipient setting with randomness re-use the adversary can certify 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. But in the model of [Ku] all the public keys assumed to be random, and the scheme can be proven secure.

Consider another example which exploits a different weakness of the model of [Ku]. Let $\mathcal{AE}' = (\mathcal{G}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$ be some IND-CPA secure encryption scheme. Consider a multi-recipient scheme $\overline{\mathcal{AE}}$ with user i's public key $pk_i = (g^{x_i}, pk'_i)$, where g^{x_i} is a public key for El Gamal encryption and pk'_i is a public key of $\mathcal{AE}'$. Let the encryption algorithm of $\mathcal{AE}'$ be as follows.

Algorithm $\overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$
$r \stackrel{R}{\leftarrow} Z_q$; For $i = 1, \ldots, n$ do $\mathbf{C}'[i] \leftarrow \mathcal{E}'_{pk'_i}(r)$
$Y_i \leftarrow g^r$; $W_i \leftarrow (g^{x_i})^r M[i]$
$\mathbf{C}[i] \leftarrow (Y_i, W_i, \mathbf{C}'[i])$ EndFor
Return $\mathbf{C}$

We claim that there exists an attack on $\overline{\mathcal{AE}}$ but the scheme can be proven secure under the model of [Ku]. We first show that $\overline{\mathcal{AE}}$ is insecure in practice by presenting an 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 model of [Ku]. Let B be an adversary attacking $\overline{\mathcal{AE}}$ under the model of [Ku]. Then it is possible to construct an adversary D which attacks El Gamal RR-MRES. But [Ku] proves the latter scheme is secure, so this would imply that $\overline{\mathcal{AE}}$ is secure. D simply provides 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 $\mathbf{C}_B[i] = (\mathbf{C}_D[i], \mathcal{E}'_{pk'_i}(r'))$. 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 is negligible.

Moreover, the model of [Ku], as well as of [BBM, BPS] do not 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 symmetric session keys K_1, K, K are El Gamal 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 $(g^r, g^{rx} \cdot K_1), (g^r, g^{2rx} \cdot K), (g^r, g^{3rx} \cdot K)$. From them it can compute $K_1 = [g^{rx} \cdot K_1] \cdot [g^{2rx} \cdot K] \cdot [g^{3rx} \cdot K]^{-1}$ and obtain the session key of the legitimate user. As a consequence, the adversary will be able to decrypt the secret information encrypted under this session key addressed to 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, along with public keys for the corrupted users, corresponding secret keys. This models the effect of appropriate proofs of knowledge of the secret key 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. The example of such schemes is RSA-PKCS#1 [PKCS] 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 [BPS, BBM]. Nonetheless, the associated RR-MRES scheme is insecure. The attack is as follows. Let N_i be the public modulus of user i and assume all users have encryption exponent 3. 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, 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 (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 in encryption algorithm is a by-product of 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 is secure.

6 Reproducibility Test and Theorem

We provide a condition under which a given encryption scheme can be a base of the 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 n be polynomial in k, and let R be an algorithm that takes as input a public key 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 $\mathbf{Exp}^{repr}_{\mathcal{AE},R}(k)$
$I \xleftarrow{R} \mathcal{G}(k)$; $(pk, sk) \xleftarrow{R} \mathcal{K}(I)$; $M \xleftarrow{R} \mathrm{MsgSp}(I)$; $r \xleftarrow{R} \mathrm{Coins}_{\mathcal{E}}(I, pk)$
$C \leftarrow \mathcal{E}_{pk}(M, r)$; $(pk', sk') \xleftarrow{R} \mathcal{K}(I)$; $M' \xleftarrow{R} \mathrm{MsgSp}(I)$
If $\mathcal{E}_{pk'}(M', r) = R(pk, C, M', pk', sk')$ then Return 1 else Return 0 EndIf

We say that $\mathcal{AE}$ is *reproducible* if for any k there exists a probabilistic, poly(k)-time algorithm R called the reproduction algorithm such that $\mathbf{Exp}^{repr}_{\mathcal{AE},R}(k)$ 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 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 $n(\cdot)$. 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 poly-time

adversary B_{atk}, there exists apoly(k)-time adversary A_{atk}, where atk = {cpa, cca}, such that for any k

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

Proof: We first consider the case of chosen-plaintext attack only and then briefly 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

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

The statement of the Theorem 6.2 follows, so it remains to design A. 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 $\mathbf{ExpH}_j$ $\;[\,0 \le j \le n(k)]$
$I \xleftarrow{R} \mathcal{G}(k)$; $l \leftarrow B(\text{select}, I)$
For $i = 1,\ldots,l$ do $(pk_i, sk_i) \xleftarrow{R} \mathcal{K}(I)$ EndFor
$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\,;\, m_{1,1}, m_{2,1}, \ldots, m_{l,1}\,;\, m_{l+1}, \ldots, m_n; pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n, st) \leftarrow$
  $B(\text{find}, I, pk_1, \ldots, pk_l)$
$\mathbf{pk} \leftarrow (pk_1, \ldots, pk_n)$
If $j \le l$
  then $\mathbf{M} \leftarrow (m_{1,0}, \ldots, m_{j,0}, m_{j+1,1}, \ldots, m_{l,1}, m_{l+1}, \ldots, m_n)$
  else $\mathbf{M} \leftarrow (m_{1,0}, \ldots, m_{l,0}, m_{l+1}, \ldots, m_n)$
EndIf
$\mathbf{C} \xleftarrow{R} \overline{\mathcal{E}}_{\mathbf{pk}}(\mathbf{M})$; $d \leftarrow B(\text{guess}, \mathbf{C}, st)$; Return d

Let $P_j \stackrel{\text{def}}{=} \Pr\left[\,\mathbf{ExpH}_j = 0\,\right]$ denote the probability that experiment $\mathbf{ExpH}_j$ returns 0, for $j = 0, 1, \ldots, n$. Now we claim that

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

This is justified as follows. We claim that

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

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

Now we turn to the description of A.

Adversary $A(find, I, pk)$
$l \leftarrow B(\text{select}, I)$; $j \xleftarrow{R} \{1, \ldots, n\}$
If $j \le l$ then For $i \in \{1, \ldots, j-1, j+1, \ldots, l\}$ do $(pk_i, sk_i) \xleftarrow{R} \mathcal{K}(I)$; $pk_j \leftarrow pk$ EndFor
else For $i = 1, \ldots l$ do $(pk_i, sk_i) \xleftarrow{R} \mathcal{K}(I)$ EndFor

EndIf
$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\ ;\ m_{1,1}, m_{2,1}, \ldots, m_{l,1}\ ;\ m_{l+1}, \ldots, m_n\ ;\ pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n, st')$
$\leftarrow B(\text{find}, I, pk_1, \ldots, pk_l)$
If $j > l$ then $m_{j,0} \leftarrow m_j$; $m_{j,1} \leftarrow m_j$ EndIf
$st \leftarrow (j, l\ ;\ pk_1, sk_1, \ldots, pk_l, sk_l\ ;\ m_{1,0}, m_{2,0}, \ldots, m_{l,0}\ ;\ m_{1,1}, m_{2,1}, \ldots, m_{l,1}\ ;\ m_{l+1}, \ldots, m_n$
$pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n, st')$
Return $(m_{j,0}, m_{j,1}, st)$

Adversary $A(guess, C, st)$
For $i \in \{1, \ldots, j-1, j+1, \ldots, n\}$ do
If $i \leq j$ Then $M' \leftarrow m_{i,0}$ Else $M' \leftarrow m_{i,1}$ EndIf
$C_i \leftarrow R(pk, C, M', pk_i, sk_i)$
EndFor
$\mathbf{C}' \leftarrow (C_1, \ldots, C_{j-1}, C, C_{j+1}, \ldots, C_n)$; $d \leftarrow B(\text{guess}, \mathbf{C}', st')$; Return d

We claim that

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

Subtracting and exploiting the collapse of the sums we get

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

The statement of the theorem follows, so it remains to justify Equations (2). Each value of j in $\{1, \ldots, n\}$ 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 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 $\mathbf{ExpH}_j$. Similarly, the experiment $\mathbf{Exp}^{\text{cpa}-1}_{\mathcal{AE},A}(k)$ is the same as $\mathbf{ExpH}_{j-1}$.

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 brief 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 $\mathcal{D}_{sk}(\cdot)$ 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 $\mathcal{D}_{sk_i}(\cdot)$ 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. ∎

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 RRS.

We first consider three DDH-based schemes which work over a group of prime order. A *prime-order-group generator* is a probabilistic algorithm that on input a security parameter k returns a pair (q, g) satisfying the following conditions: q is a prime with $2^{k-1} < q < 2^k$; $2q + 1$ is a prime; and g is a generator of G_q.

7.1 El Gamal

Let $\mathcal{G}$ be a prime-order-group generator. This is the common key generation algorithm of the El Gamal scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, the rest of the algorithms are as follows:

$\underline{\mathcal{K}(q,g)}$:	$\underline{\mathcal{E}_{pk}(M)}$:	$\underline{\mathcal{D}_{sk}(Y,W)}$:
$x \xleftarrow{R} Z_q$; $X \leftarrow g^x$	Parse pk as (q,g,X)	Parse sk as (q,g,x)
$pk \leftarrow (q,g,X)$; $sk \leftarrow (q,g,x)$	$r \xleftarrow{R} Z_q$; $Y \leftarrow g^r$	$T \leftarrow Y^x$
Return (pk, sk)	$T \leftarrow X^r$; $W \leftarrow TM$	$M \leftarrow WT^{-1}$
	Return (Y, W)	Return M

The message space associated to a common key (q, g) is the group G_q itself. Note that a generator g is the output of the common key generation algorithm, which means we fix g for all keys.

Lemma 7.1 The El Gamal encryption scheme $\mathcal{EG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible.

Proof: On input $(pk, (g^r, g^{rx} \cdot M), M', pk', sk')$, where $pk = (q, g, g^x), pk' = (q, g, g^{x'}), sk' = (q, g, x')$, a poly$(k)$-time reproduction algorithm 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 $\mathbf{Exp}^{repr}_{\mathcal{EG},R}(k)$ always outputs 1. ▌

The El Gamal 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]). Accordingly we define the DDH problem.

Definition 7.2 [DDH] Let $\mathcal{G}$ be a prime-order-group generator. Let D be an adversary that on input q, g and three elements $X, Y, T \in G_q$ returns a bit. We consider the following experiments

Experiment $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D}(k)$	Experiment $\mathbf{Exp}^{\text{ddh-rand}}_{\mathcal{G},D}(k)$
$(q,g) \xleftarrow{R} \mathcal{G}(k)$	$(q,g) \xleftarrow{R} \mathcal{G}(k)$
$x \xleftarrow{R} Z_q$; $X \leftarrow g^x$; $y \xleftarrow{R} Z_q$; $Y \leftarrow g^y$	$x \xleftarrow{R} Z_q$; $X \leftarrow g^x$; $y \xleftarrow{R} Z_q$; $Y \leftarrow g^y$
$T \leftarrow g^{xy}$; $d \leftarrow D(q,g,X,Y,T)$; Return d	$T \xleftarrow{R} G_q$ $d \leftarrow D(q,g,X,Y,T)$; 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

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

We say that the DDH problem is hard for $\mathcal{G}$ if the function $\mathbf{Adv}^{\text{ddh}}_{\mathcal{G},D}(\cdot)$ is negligible for every poly(k)-time algorithm D. ▌

Theorem 6.2 and Lemma 7.1 imply that it is also RR-IND-CPA or, equivalently, $\overline{\mathcal{EG}}$ is IND-CPA secure and the security degrades linearly as the number of users n increases. The following theorem shows that it is possible to obtain a tighter relation than the one implied by Theorem 6.2.

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

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

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 A. [Ku] proves a similar result but for a weaker notion of security of multi-recipient schemes.

$\underline{\mathcal{G}(k)}$:	$\underline{\mathcal{K}(q, g_1, g_2, K)}$:	$\underline{\mathcal{E}_{pk}(M)}$:	$\underline{\mathcal{D}_{sk}(u_1, u_2, e, v)}$:
$(q, g_1) \xleftarrow{R} \overline{\mathcal{G}}$	$x_1, x_2, y_1, y_2, z \xleftarrow{R} Z_q$	Parse pk as (g_1, g_2, c, d, h, K)	Parse sk as (x_1, x_2, y_1, y_2, z)
$g_2 \xleftarrow{R} G_q$	$c \leftarrow g_1^{x_1} g_2^{x_2}$; $d \leftarrow g_1^{y_1} g_2^{y_2}$	$r \xleftarrow{R} Z_q$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$
$K \xleftarrow{R} \mathcal{GH}(k)$	$h \leftarrow g_1^z$	$u_1 \leftarrow g_1^r$; $u_2 \leftarrow g_2^r$	If ${u_1}^{x_1+y_1\alpha} {u_2}^{x_2+y_2\alpha} = v$
Return (q, g_1, g_2, K)	$pk \leftarrow (g_1, g_2, c, d, h, K)$	$e \leftarrow h^r M$	then $M \leftarrow e/{u_1}^z$
	$sk \leftarrow (x_1, x_2, y_1, y_2, z)$	$\alpha \leftarrow \mathcal{EH}_K(u_1, u_2, e)$	else $M \leftarrow \perp$
		$v \leftarrow c^r d^{r\alpha}$	EndIf
	Return (pk, sk)	Return (u_1, u_2, e, v)	Return M

Figure 1: Cramer-Shoup scheme

7.2 Cramer-Shoup

We now consider an RR-MRES based on the Cramer-Shoup scheme [CrSh] in order to get IND-CCA security properties. We first recall the Cramer-Shoup scheme. 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.

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

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

Proof: We present a polynomial time algorithm R which takes as input a random 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 $R(pk, C, M', pk', sk')$
Parse pk as (g_1, g_2, c, d, h, K); Parse C as (u_1, u_2, e, v)
Parse pk' as $(g_1, g_2, c', d', h', K)$; Parse sk' as $(x'_1, x'_2, y'_1, y'_2, z')$
$e' \leftarrow u_1^{z'} M'$; $\alpha' \leftarrow \mathcal{EH}_K(u_1, u_2, e')$; $v' \leftarrow u_1^{x'_1+y'_1\alpha'} u_2^{x'_2+y'_2\alpha'}$
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 $e' = u_1^{z'} M' = g_1^{rz'} M' = (h')^r M'$. 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 $\Pr\left[\mathbf{Exp}^{repr}_{\mathcal{CS},R}(k) = 1\right] = 1$. ∎

Let $\mathbf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k)$ denote the advantage of an adversary C breaking collision-resistance of $\mathcal{H}$ (Section B.1 recalls the formal definition of collision resistance). If the DDH problem is hard for $\mathcal{G}$ and if $\mathcal{H}$ is collision-resistant then $\mathcal{CS}$ is IND-CCA secure [CrSh]. Theorem 6.2 and Lemma 7.4 imply that it is also RR-IND-CCA or, equivalently, $\overline{\mathcal{CS}}$ is IND-CCA secure. We match the result of [Ku] in getting a better security result than the one implied by Theorem 6.2 but we do it for a stronger notion of security of multi-recipient schemes. The following theorem states our improvement.

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

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

Figure 2: DHIES

Theorem 7.5 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 be a polynomial. Then for any adversary B, which makes q_{d} decryption oracle queries, there exists an adversary A, a distinguisher D and an adversary C such that for any k

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

and the running time of D and C is that of B plus $O(n(k) \cdot k^3)$. ∎

Note that the security of $\overline{\mathcal{CS}}$ is tightly related to the security of DDH and does not depend on the number of the users in the system. The proof of the above theorem is in Appendix B.2.

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 $\mathsf{SE} = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ be a symmetric encryption scheme with key length kl and let $\mathsf{MAC} = (\mathcal{T}, \mathcal{V})$ be a message authentication code with key length ml, tagging algorithm $\mathcal{T}$ and verification algorithm $\mathcal{V}$. Let $H\colon \{0,1\}^{gl} \rightarrow \{0,1\}^{ml+kl}$ be a function. We assume MAC is deterministic. The common key and key generation algorithms of $\mathcal{DHIES}[\mathsf{SE}, H, \mathsf{MAC}] = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ are the same as the ones of El Gamal 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 7.6 $\mathcal{DHIES}[\mathsf{SE}, H, \mathsf{MAC}] = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible if SE is reproducible.

Proof: Since SE is reproducible then there exists a poly(k)-time reproduction algorithm R' for 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 a poly(k)-time reproduction algorithm R for $\mathcal{DHIES}$ which uses R'.

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

Note that R first computes symmetric keys for 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). ▮

7.4 Escrow El Gamal

Boneh and Franklin [BF] suggested the El Gamal encryption scheme with global escrow capabilities. The $\mathcal{EEG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ scheme uses Weil pairing and is defined as follows. The algorithm $\mathcal{G}$ on input a security parameter k chooses a k-bit prime p such that $p \equiv 2 \bmod 3$ and $p \equiv 6q - 1$ for some prime $q \geq 3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over F_p. Then it chooses a random $P \in E/\mathsf{F}_p$ of order q, computes $Q = sP$ for a random $s \in \mathbb{Z}_q^*$ and chooses a hash function $H\colon \mathsf{F}_{p^2} \to \{0,1\}^m$. The message space is $\{0,1\}^m$. The escrow key is s. $\mathcal{G}$ outputs (p, m, P, Q, H). The rest of the algorithms are as follows:

$\underline{\mathcal{K}(p,m,P,Q,H)}$:	$\underline{\mathcal{E}_{pk}(M)}$:	$\underline{\mathcal{D}_{sk}(U,V)}$:
$x \stackrel{R}{\leftarrow} Z_q^*$; $X \leftarrow xP$	Parse pk as (p, P, Q, X)	Parse sk as (p, P, Q, x)
$pk \leftarrow (p, P, Q, X)t$	$r \stackrel{R}{\leftarrow} Z_q^*$	$M \leftarrow V \oplus H(\hat{e}(U, xQ))$
$sk \leftarrow (p, P, Q, x)$	$g \leftarrow \hat{e}(X, Q)$	Return M
Return (pk, sk)	Return $(rP, M \oplus H(g^r))$	

We do not define the decryption using the escrow key since it is not relevant for our goal.

Lemma 7.7 The escrow El Gamal encryption scheme $\mathcal{EEG} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is reproducible.

Proof: The reproduction algorithm R takes $(pk, (rP, M \oplus H((\hat{e}(X,Q))^r), M', pk', sk')$ where $pk = (p,\ P, Q, X)$, $pk' = (p, P, Q, X')$, $sk' = (p, P, Q, x')$ and returns $(rP,\ M' \oplus H(\hat{e}(rP, x'Q)))$. Since $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$ one can check that R always outputs a valid ciphertext which is created using the same random string as the given ciphertext and therefore $\mathcal{EEG}$ is reproducible. ▮

A standard argument shows that $\mathcal{EEG}$ is IND-CPA secure in the random oracle model assuming Bilinear Diffie-Hellman assumption (see [BF] for proper definitions). The results of Theorem 6.2 and Lemma 7.7 can be extended for the random oracle model and they would imply that $\mathcal{EEG}$ is also RR-IND-CPA or, equivalently, the corresponding multi-recipient scheme $\overline{\mathcal{EEG}}$ is IND-CPA secure, both in the random oracle model.

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 El Gamal and Cramer-Shoup are RRS, while some, e.g. RSA-PKCS#1 are not. We now provide a simple method for an efficient transformation of any encryption scheme which meets the standard notion of security into RRS one. The construction will use a pseudorandom function family; we recall the notion of pseudorandomness [GGM] and define the advantage $\mathbf{Adv}^{\mathrm{prf}}_{F,D}(k)$ of the distinguisher D breaking pseudorandomness of the function family F in Appendix C.1.

Construction 8.1 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, poly(k)-time computable functions $il\colon \mathsf{N} \to \mathsf{N}, ol\colon \mathsf{N} \to \mathsf{N}$ such that for all

k $|(I, pk)| = il(k)$ and $\text{Coins}(I, pk) = \{0,1\}^{ol(k)}$ for all I generated by $\mathcal{G}(k)$ and all pk generated by $\mathcal{K}(k)$. Fix a polynomially bounded, poly-time computable function $kl\colon \mathsf{N} \to \mathsf{N}$ and fix a function family $F\colon \{0,1\}^{kl(k)} \times \{0,1\}^{il(k)} \to \{0,1\}^{ol(k)}$. 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 $\mathcal{E}'_{pk}(M, r')$
$r \leftarrow F(r', (I, pk))$; $C \leftarrow \mathcal{E}_{pk}(M, r)$
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 8.2 Fix an asymmetric encryption scheme $\mathcal{AE}$. Assume that there exist functions $il\colon \mathsf{N} \to \mathsf{N}, ol\colon \mathsf{N} \to \mathsf{N}$ satisfying the conditions defined above. Let $\mathcal{AE}'[F]$ be a transformed encryption scheme as per Construction 8.1. 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 C 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-PKCS#1, which is IND-CCA secure, by adding one application of a block cipher such that the resulting scheme becomes RR-IND-CCA.

Corollary 8.3 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 8.1 and Theorem 8.2 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.

SYNTAX. Following [BDJR], a symmetric encryption scheme $\mathsf{SE} = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ consists of three algorithms.

- A randomized poly(k)-time key generation algorithm K takes a security parameter k and returns a key sk; we write $sk \xleftarrow{R} \mathrm{K}(k)$.

- A randomized poly(k)-time encryption algorithm E takes the key sk and a message $M \in \mathrm{MsgSp}(k)$ to return a ciphertext C; we write $C \xleftarrow{R} \mathrm{E}_{sk}(M)$ or $r \xleftarrow{R} \mathrm{Coins}_{\mathrm{E}}(k)$; $C \leftarrow \mathrm{E}_{sk}(M, r)$.
- A deterministic decryption algorithm $\mathcal{D}$ takes sk and a ciphertext C and returns a message M; we write $M \leftarrow \mathrm{D}_{sk}(C)$.

We require that $\mathrm{D}_{sk}(\mathrm{E}_{sk}(M)) = M$ for all $M \in \mathrm{MsgSp}(k)$.

NOTION OF SECURITY FOR SYMMETRIC-KEY ENCRYPTION. Following [BDJR] we recall the security of a symmetric-key encryption scheme under chosen-plaintext attacks. An adversary wins if it can find two equal-length messages whose ciphertexts it can later distinguish. An adversary attacking the encryption scheme is given an encryption oracle $\mathrm{E}_K(\cdot)$ which returns an encryption of an input plaintext.

Definition 9.1 [Indistinguishability of symmetric-key ciphertexts] Let $\mathsf{SE} = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ be a symmetric encryption scheme. For an adversary A, $k \in N$ and $b \in \{0, 1\}$ define the experiments

Experiment $\mathbf{Exp}^{\mathrm{cpa}-b}_{\mathsf{SE},A}$

$sk \xleftarrow{R} \mathcal{K}(k)$; $(m_0, m_1, st) \leftarrow A^{\mathrm{E}_{sk}(\cdot)}(\mathrm{find})$

$C \xleftarrow{R} \mathrm{E}_{sk}(m_b)$

$d \leftarrow A^{\mathrm{E}_{sk}(\cdot)}(\mathrm{guess}, C, st)$

Return d

It is mandated that $|m_0| = |m_1|$ above. Now define the *advantage* of A as follows:

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

The scheme SE is said to be *polynomially-secure against chosen-plaintext attacks* if the function $\mathbf{Adv}^{\mathrm{cpa}}_{\mathsf{SE},A}(\cdot)$ is negligible for any poly(k)-time adversary. ▌

SYMMETRIC-KEY MRESS. We now consider MRESs in the symmetric-key setting. Syntax for such schemes $\overline{\mathsf{SE}} = (\mathrm{K}, \overline{\mathrm{E}}, \mathrm{D})$ can be defined similarly to syntax of asymmetric MRESs defined in Section 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 random string being a by-product of a decryption algorithm 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 $\overline{\mathsf{SE}} = (\mathrm{K}, \overline{\mathrm{E}}, \mathrm{D})$ be a symmetric-key MRES. Let n be polynomial in k and let B be an adversary. B has access to an oracle which takes a vector. For $b \in \{0,1\}$ define the experiments:

Experiment $\mathbf{Exp}^{n\text{-mr-cpa-b}}_{\overline{\mathsf{SE}},B}(k)$
$(l, sk_{l+1}, \ldots, sk_n) \leftarrow B(\text{select})$
For $i = 1, \ldots, l$ do $sk_i \xleftarrow{R} \mathcal{K}(k)$ EndFor
$\mathbf{sk} \leftarrow (sk_1, \ldots, sk_n)$
$(m_{1,0}, \ldots, m_{l,0}\,;\, m_{1,1}, \ldots, m_{l,1}\,;\, m_{l+1}, \ldots, m_n) \leftarrow B^{\overline{\mathrm{E}}_{\mathbf{sk}}(\cdot)}(\text{find})$
$\mathbf{M} \leftarrow (m_{1,b}, \ldots, m_{l,b}, m_{l+1}, \ldots, m_n, st)$
$\mathbf{C} \xleftarrow{R} \overline{\mathrm{E}}_{\mathbf{sk}}(\mathbf{M})$
$d \leftarrow B^{\overline{\mathrm{E}}_{\mathbf{sk}}(\cdot)}(\text{guess}, C, st)$
Return d

It is required that $|m_{i,0}| = |m_{i,1}|$ for all $1 \le i \le n(k)$. We define the *advantage* $\mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathsf{SE}},B}(k)$ of the adversary, IND-CPA security of the symmetric MRES analogously to the definitions for a public-key case of 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 $\mathsf{SE} = (\mathrm{K}, \mathrm{E}, \mathrm{D})$. 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 $\mathbf{Exp}^{repr}_{\mathsf{SE},R}(k)$
$sk \xleftarrow{R} \mathrm{K}(k)\,;\, M \xleftarrow{R} \mathrm{MsgSp}(k)\,;\, r \xleftarrow{R} \mathrm{Coins}_{\mathrm{E}}(k)\,;\, C \leftarrow \mathrm{E}_{sk}(M, r)$
$sk' \xleftarrow{R} \mathrm{K}(k)\,;\, M' \xleftarrow{R} \mathrm{MsgSp}(k)\,;\, C' \leftarrow R(C, M', sk')$
If $(\mathrm{E}_{sk'}(M', r) = C')$ then Return 1 else Return 0 EndIf

We say that SE is *reproducible* if for any k there exists a probabilistic, poly(k)-time algorithm R such that $\mathbf{Exp}^{repr}_{\mathsf{SE},R}(k)$ outputs 1 with probability 1. ▌

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

Theorem 9.4 Fix a symmetric-key encryption scheme $\mathsf{SE} = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ and a polynomial n. Let $\overline{\mathsf{SE}} = (\mathrm{K}, \overline{\mathrm{E}}, \mathrm{D})$ be the corresponding RR-MRES. If SE is reproducible then for any polynomial-time adversary B, there exists a polynomial-time adversary A, such that

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

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. 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 function family F with input and output length s and a key length k. A key-generation algorithm of $\mathcal{CBC}[F] = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ simply outputs a random k-bit key sk, which specify a function f 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:

Algorithm $\mathrm{E}_{sk}(M)$	Algorithm $\mathrm{D}_{sk}(c_0, c_1, \ldots, c_p)$
Parse M as $M_1, \ldots, M_p$	For $i = 1, \ldots, p$ do $M_i \leftarrow f^{-1}(c_i) \oplus c_{i-1}$
$c_0 \stackrel{R}{\leftarrow} \{0,1\}^s$	$M \leftarrow M_1 \| \ldots \| M_p$
For $i = 1, \ldots, p$ do $c_i \leftarrow f(c_{i-1} \oplus M_i)$	Return M
Return $(c_0, c_1, \ldots, c_p)$	

c_0 is often called an initial vector or IV.

Lemma 9.5 CBC encryption scheme $\mathcal{CBC}[F] = (\mathrm{K}, \mathrm{E}, \mathrm{D})$ is reproducible for any F.

Proof: A polynomial time reproduction algorithm R takes as input $R((c_0, c_1, \ldots, c_p), M', sk')$ and returns $C' = \mathrm{E}_{sk'}(M', c_0)$. 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 $\mathbf{Exp}^{repr}_{\mathcal{CBC}[F],R}(k)$ 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 Acknowledgements

We thank Diana Smetters for useful discussions. Part of this research has been done when Alexandra Boldyreva was at PARC.

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.

[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.

[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.

[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.

[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, 1364–1396, 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 Fifth 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.

[PKCS] "PKCS-1," RSA Labs, `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/`.

[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/`.

Adversary $D(q, g, X, Y, T)$
$X_1 \leftarrow X$; $T_1 \leftarrow T$; $pk_1 \leftarrow (q, g, X_1)$
For $i = 2, \ldots l$ do
$v_i \stackrel{R}{\leftarrow} Z_q$; $w_i \stackrel{R}{\leftarrow} Z_q$; $X_i \leftarrow (X_1)^{w_i} \cdot g^{v_i}$; $T_i \leftarrow T_1^{w_i} \cdot Y^{v_i}$
$pk_i \leftarrow (q, g, X_i)$
EndFor
$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\ ;\ m_{1,1}, m_{2,1}, \ldots, m_{l,1}\ ;\ m_{l+1}, \ldots, m_n\ ;\ pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n, st)$
$\leftarrow B(\text{find}, q, g, pk_1, \ldots, pk_l)$
$b \stackrel{R}{\leftarrow} \{0, 1\}$
For $i = 1, \ldots l$ do
$\mathbf{C}[i] \leftarrow (Y, T_i \cdot m_{i,b})$
EndFor
For $i = l + 1, \ldots n$ do
Parse pk_i as q, g, g^{x_i}
Parse sk_i as q, g, x_i
$\mathbf{C}[i] \leftarrow (Y, Y^{x_i} \cdot m_i)$
EndFor
$\mathbf{C} \leftarrow \mathbf{C}[1], \ldots, \mathbf{C}[n]$
$d \leftarrow A(\text{guess}, \mathbf{C}, st)$
If $b = d$ then return 1 else return 0

Figure 3: Adversary D for the proof of Theorem 7.3

[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 El Gamal 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 7.3

The proof is similar to the corresponding proof of [Ku]. We still provide the details since we use the different notion of security of multi-recipient schemes. Let A be an adversary attacking $\overline{\mathcal{EG}}$ scheme. We will design a distinguisher D for the DDH problem which we recalled in Definition 7.2 so that

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

The statement of Theorem 7.3 follows. 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 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 with the $n-l$ ciphertexts are distributed exactly like a ciphertext in RR-MRES El Gamal scheme under public keys $pk_1, \ldots pk_n$. We use it to see that for any k

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

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 El Gamal cryptosystem. But now $T_1, \ldots, T_l$ 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 $x_{l+1}, \ldots, x_n$. This means that the challenge ciphertext gives B no information about b, in an information-theoretic sense. We have

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

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

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

which is Equation (3).

It remains to show that D runs in time polynomial in k. 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 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.

B Definition and Proof for Section 7.2

B.1 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 the security parameter k and returns a key K— and a deterministic evaluation algorithm $\mathcal{EH}$ —which takes as input the key K and a string $M \in \{0,1\}^*$ and returns a string $\mathcal{EH}_K(M) \in \{0,1\}^{k-1}$.

Definition B.1 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 returns two strings. Now, we consider the following experiment:

Experiment $\mathbf{Exp}^{\text{cr}}_{\mathcal{H},C}(k)$
$K \xleftarrow{R} \mathcal{GH}(k)\,;\ (x_0, x_1) \leftarrow C(K)$
If $(x_0 \ne x_1)$ and $\mathcal{EH}_K(x_0) = \mathcal{EH}_K(x_1)$ then return 1 else return 0

Adversary $D(q, g, X, Y, T)$

$K \xleftarrow{R} \mathcal{GH}(k)$; $l \leftarrow B(\text{select}, q, g)$; $g_1 \leftarrow g$; $g_2 \leftarrow X$; $u_1 \leftarrow Y$; $u_2 \leftarrow T$

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

$x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}, z_{1,i}, z_{2,i} \xleftarrow{R} Z_q$; $c_i \leftarrow g_1^{x_{1,i}} g_2^{x_{2,i}}$; $d_i \leftarrow g_1^{y_{1,i}} g_2^{y_{2,i}}$; $h_i \leftarrow g_1^{z_{1,i}} g_2^{z_{2,i}}$

$pk_i \leftarrow (g_1, g_2, c_i, d_i, h_i)$

EndFor

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

$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\ ;\ m_{1,1}, m_{2,1}, \ldots, m_{l,1}\ ;\ m_{l+1}, \ldots, m_n\ ;\ pk_{l+1}, sk_{l+1} \ldots, pk_n, sk_n, st)$

$\leftarrow B(\text{find}, q, g, pk_1, \ldots, pk_l)$

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

$e_i \leftarrow u_1^{z_{1,i}} u_2^{z_{2,i}} m_{i,b}$; $\alpha_i \leftarrow \mathcal{EH}_K(u_1, u_2, e_i)$; $v_i \leftarrow u_1^{x_{1,i}+y_{1,i}\alpha_i} u_2^{x_{2,i}+y_{2,i}\alpha_i}$; $C_i \leftarrow (u_1, u_2, e_i, v_i)$

EndFor

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

Parse sk_i as $(x_{1,i}, x_{2,i}, y_{1,i}, y_{2,i}, z_{1,i}, z_{2,i})$

$e_i \leftarrow u_1^z M_i$; $\alpha_i \leftarrow \mathcal{EH}_K(u_1, u_2, e_i)$; $v_i \leftarrow u_1^{x_{1,i}+y_{1,i}\alpha_i} u_2^{x_{2,i}+y_{2,i}\alpha_i}$; $C_i \leftarrow (u_1, u_2, e_i, v_i)$

EndFor

$\mathbf{C} \leftarrow (C_1, \ldots, C_n)$

$d \leftarrow B(\text{guess}, \mathbf{C}, st)$

reply to B's decryption queries at any stage as follows:

$B \overset{\mathcal{D}_{sk_i}}{\rightarrow} \bar{C}$ [This denotes that B makes a query $\bar{C}$ to $\mathcal{D}_{sk_i}$]

parse $\bar{C}$ as $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$; $\bar{\alpha} \leftarrow \mathcal{EH}_K(\bar{u}_1, \bar{u}_2, \bar{e})$

If $\bar{u}_1^{x_{1,i}+y_{1,i}\bar{\alpha}} \bar{u}_2^{x_{2,i}+y_{2,i}\bar{\alpha}} = \bar{v}$ then $m \leftarrow \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}}$ else $m \leftarrow \perp$ EndIf

B gets m

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

Figure 4: Adversary D for the proof of Theorem 7.5

We define the *advantage* of adversary C via

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

We say that the family of hash functions $\mathcal{H}$ is *collision-resistant* if $\mathbf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k)$ is negligible for every poly(k)-time algorithm C. ▌

B.2 Proof of Theorem 7.5

The proof is similar to the corresponding proof of [Ku] which in turn uses the techniques of [CrSh, BBM], but the novel element is that we use the notion of security where the adversary is allowed to corrupt hones users. We present a distinguisher D in Figure 4. To analyze D we first consider $\mathbf{Exp}^{\text{ddh-real}}_{\mathcal{G},D_A}(k)$. We show that the view of the adversary B under D's simulation is exactly as in the actual experiment. Without loss of generality we assume that B is deterministic, otherwise the following arguments can be made for each choice of the random coins of the adversary. Thus we assume that B outputs a fixed number l in the end of its select stage. We show that l public keys and a challenge ciphertext vector given to B have the correct distribution and that decryption oracle queries are answered as in a real experiment.

The input to D has the form $q, g, g^{r_1}, g^{r_2}, g^{r_1 r_2}$. We can read this also as q, g_1, g_2, u_1, u_2, where

$u_2 = g_1^{r_2}$ and $u_2 = g_2^{r_2}$. Obviously, for all $1 \le i \le l$ c_i, d_i have the right distribution of public keys since they are computed exactly like in the actual experiment. In the proof in [CrSh] the distinguisher has to compute only one public key and it is shown there that h has the right distribution even though it is computed differently from an actual experiment.. We compute all h_i similarly and the same argument can be applied to show that they have the right distribution. Therefore, l public keys computed by D have the right distribution.

Now we show that the challenge ciphertext vector has the right distribution. Clearly, u_1, u_2 are of the right form. We note that $e_1, \alpha_1, v_1, \ldots, e_l, \alpha_l, v_l$ are all computed using fixed u_1, u_2 and use the claims of [CrSh] to see that besides that they have the right distribution. It is also easy to see that the rest ciphertexts $C_{l+1}, \ldots C_n$ are computed exactly like in an actual experiment. Thus, the challenge ciphertext vector $(u_1, u_2, (e_1, v_1), \ldots, (e_n, v_n))$ has the right distribution.

Finally we show that the decryption oracle queries $(\bar{u}_1, \bar{u}_2, \bar{e}, \bar{v})$ are answered correctly. This is because the condition of a valid ciphertext is computed as in the actual experiment and the plaintext is computed as $M = \bar{e}/\bar{u}_1^{z_{1,i}} \bar{u}_2^{z_{2,i}} = \bar{e}/h_i^{r_2}$ for $1 \le i \le l$ if the query was made to $\mathcal{D}_{sk_i}$, which is as in the actual decryption algorithm. Thus for any k and polynomial n we have

$$\Pr\left[\, \mathbf{Exp}^{\text{ddh-real}}_{\overline{\mathcal{G}},D}(k) = 1 \,\right] \;=\; \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{n\text{-mr-cca}}_{\overline{\mathcal{CS}},B}(k)\ . \tag{6}$$

Similarly to the corresponding proofs of [CrSh, BBM, Ku] one can show that if D's input is a random tuple and if H is a collision-resistant function family, then B's view in the simulated experiment is independent from it's challenge bit. More precisely, there exists a polynomial time adversary C such that for every k

$$\Pr\left[\, \mathbf{Exp}^{\text{ddh-rand}}_{\overline{\mathcal{G}},D}(k) = 1 \,\right] \;\le\; \frac{1}{2} + \mathbf{Adv}^{cr}_{\mathcal{H},\mathcal{C}}(k) + \frac{q_d(k) + 2}{2^{k-2}} \tag{7}$$

where q_d is the number of decryption oracle queries made by B. We omit the details.

The statement of Theorem 7.5 follows Equation (6) and Equation (7). It remains to show that D runs in time polynomial in k. To show that D runs in time polynomial in k we note that the overhead for D is that of performing at most $O(n)$ exponentiation operations with respect to a base element in G_q and an exponent in Z_q and $O(n)$ 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 Definition and Proof for Section 8

C.1 Pseudorandom function families

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

Definition C.1 Let F, R be as above, let D be a distinguisher. Define the *advantage of D* as

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

The function family F is said to be *pseudorandom* if $\mathbf{Adv}^{\mathrm{prf}}_{F,D}(\cdot)$ is negligible for any adversary whose running time is polynomial in k. ▌

C.2 Proof of Theorem 8.2

We prove that for any poly-time adversary A_{atk}, there exist a poly-time adversary B_{atk}, where $\mathrm{atk} = \{\mathrm{cpa}, \mathrm{cca}\}$ and a distinguisher D, such that for any k

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

The statement of Theorem 8.2 is implied by the 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 $\{0,1\}^{il(k)} \to \{0,1\}^{ol(k)}$. Let A be a poly-time adversary attacking the security of the multi-recipient scheme $\overline{\mathcal{AE}'[F]}$. We will construct a distinguisher D which attacks F as a pseudorandom function family and an adversary B which attacks the security of $\mathcal{AE}$ such that their running time is polynomial and

$$\mathbf{Adv}^{\mathrm{prf}}_{F,D}(k) \;=\; \frac{1}{2} \cdot (\mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathcal{AE}'[F]},A}(k) - \mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathcal{AE}'[R]},A}(k)) \tag{8}$$

$$\mathbf{Adv}^{\mathrm{cpa}}_{\mathcal{AE},B}(k) \;\geq\; \frac{1}{n(k)} \cdot \mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathcal{AE}'[R]},A}(k) \tag{9}$$

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. The statement of the theorem follows. It remains to specify the strategies of D and B. The distinguisher D takes k and has access to an oracle $g\colon \{0,1\}^{il(k)} \to \{0,1\}^{ol(k)}$. D will run A as a subroutine. Here is the algorithm for D.

Distinguisher $D^{g(\cdot)}(k)$

$I \leftarrow \mathcal{G}(k)$; $l \leftarrow A(\mathrm{select}, I)$; For $i = 1 \ldots, l$ do $(pk_i, sk_i) \xleftarrow{R} \mathcal{K}(I)$ EndFor

$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\,;\; m_{1,1}, m_{2,1}, \ldots, m_{l,1}\,;\; m_{l+1}, \ldots, m_n\,;\; pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n, st)$
$\quad \leftarrow A(\mathrm{find}, I, pk_1, \ldots, pk_l)$

$b \xleftarrow{R} \{0,1\}$; $\mathbf{M} \leftarrow (m_{1,b}, \ldots, m_{l,b}, m_{l+1}, \ldots m_n)$; $\mathbf{pk} \leftarrow (pk_1, \ldots, pk_n)$; $\mathbf{C} \leftarrow \overline{\mathcal{E}'}^{g(\cdot)}_{\mathbf{pk}}(\mathbf{M})$

$d \leftarrow A(\mathrm{guess}, \mathbf{C}, st)$

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 distinguisher. We claim that

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

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

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 $\mathbf{Exp}^{n\text{-mr-cpa-b}}_{\overline{\mathcal{AE}'[F]},A}(k)$. 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 $\mathbf{Exp}^{n\text{-mr-cpa-b}}_{\overline{\mathcal{AE}'[R]},A}(k)$. After subtraction we get Equation (8).

We now prove Equation (9). Let A be an adversary which attacks the security of $\overline{\mathcal{AE}'[R]}$. We will use the hybrid experiments $\mathbf{ExpH}_j$ for $0 \le j \le n(k)$ we defined in the proof of Theorem 6.2, which are associated to A and the encryption scheme $\overline{\mathcal{AE}'[R]}$. Let $P_j \stackrel{\text{def}}{=} \Pr\left[\, \mathbf{ExpH}_j = 0 \,\right]$ denote the probability that experiment $\mathbf{ExpH}_j$ returns 0, for $j = 0, 1, \ldots, n$. Similarly to the proof of Theorem 6.2 we claim that

$$\mathbf{Adv}^{n\text{-mr-cpa}}_{\overline{\mathcal{AE}'[R]},A}(k) \;=\; P_n - P_0 \;. \tag{10}$$

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

Adversary $B(\text{find}, I, pk)$
$l \leftarrow A(\text{select}, I)$; $j \stackrel{R}{\leftarrow} \{1, \ldots, n\}$
If $j \le l$ then For $i \in \{1, \ldots, j-1, j+1, \ldots, l\}$ do $(pk_i, sk_i) \stackrel{R}{\leftarrow} \mathcal{K}(I)$; $pk_j \leftarrow pk$ EndFor
else For $i = 1, \ldots l$ do $(pk_i, sk_i) \stackrel{R}{\leftarrow} \mathcal{K}(I)$ EndFor EndIf
$(m_{1,0}, m_{2,0}, \ldots, m_{l,0}\,;\, m_{1,1}, m_{2,1}, \ldots, m_{l,1}\,;\, m_{l+1}, \ldots, m_n\,;\, pk_{l+1}, sk_{l+1}, \ldots, pk_n, sk_n\,;\, st')$
  $\leftarrow A(\text{find}, I, pk_1, \ldots, pk_l)$
For $i = l+1, \ldots, n$ do $m_{i,0} \leftarrow m_i$; $m_{i,1} \leftarrow m_i$ EndFor
$st \leftarrow (I, j, l\,;\, pk_1, sk_1, \ldots, pk_n, sk_n\,;\, m_{1,0}, m_{2,0}, \ldots, m_{j,0}, m_{j,1}, \ldots, m_{n,1}\,;\, st')$
Return $(m_{j,0}, m_{j,1}, st)$

Adversary $B(\text{guess}, C, st)$
For $i \in \{1, \ldots, j-1, j+1, \ldots, n\}$ do
  If $pk_i = pk$ then $m \leftarrow \mathcal{D}_{sk_i}(C)$; If $m = m_{j,0}$ then Return 0 else Return 1
  Else
    If $\exists k\colon 1 \le k < i, pk_k = pk_i$ then $r_i \leftarrow r_k$; Else $\mathbf{pk} = (pk_1, \ldots, pk_n)$; $r_i \stackrel{R}{\leftarrow} \text{Coins}(I, \mathbf{pk})$ EndIf
  EndIf
EndFor
For $i = 1, \ldots, j-1$ do $C_i \leftarrow \mathcal{E}_{pk_i}(m_{i,0}, r_i)$
For $i = j+1, \ldots, n$ do $C_i \leftarrow \mathcal{E}_{pk_i}(m_{i,1}, r_i)$
$C_j \leftarrow C$; $\mathbf{C} \leftarrow (C_1, \ldots, C_n)$; $d \leftarrow A(\text{guess}, \mathbf{C}, st')$
Return d

We now analyze the adversary B. All values of j in $\{1, \ldots n\}$ 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 $\mathbf{ExpH}_j$. 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 $\mathcal{E}_{pk_i}$ are computed. In the experiment $\mathbf{ExpH}_j$ 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 $\mathbf{ExpH}_j$, 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 a challenge secret key and can always win its game by decrypting the challenge ciphertext. Thus we claim that

$$\Pr\left[\, \mathbf{Exp}^{\text{cpa}-0}_{\mathcal{AE},B}(k) = 0 \,\right] \;\ge\; \frac{1}{n} \cdot \sum_{j=1}^{n} P_j \quad \text{and} \quad \Pr\left[\, \mathbf{Exp}^{\text{cpa}-1}_{\mathcal{AE},B}(k) = 0 \,\right] \;\le\; \frac{1}{n} \cdot \sum_{j=1}^{n} P_{j-1} \;. \tag{11}$$

Subtracting and exploiting the collapse of the sums we get

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

Equation (9) follows.

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. ▌