Randomness Reuse: Extensions and Improvements

M. Barbosa[1] and P. Farshim[2]

[1] Departamento de Informática, Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal.
mbb@di.uminho.pt

[2] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, United Kingdom.
farshim@cs.bris.ac.uk

Abstract. We extend the generic framework of reproducibility for reuse of randomness in multi-recipient encryption schemes as proposed by Bellare et al. (PKC 2003). A new notion of *weak reproducibility* captures not only encryption schemes which are (fully) reproducible under the criteria given in the previous work, but also a class of efficient schemes which can only be used in the single message setting. In particular, we are able to capture the single message schemes suggested by Kurosawa (PKC 2002), which are more efficient than the direct adaptation of the multiple message schemes studied by Bellare et al. Our study of randomness reuse in key encapsulation mechanisms provides an additional argument for the relevance of these results: by taking advantage of our weak reproducibility notion, we are able to generalise and improve multi-recipient KEM constructions found in literature. We also propose an efficient multi-recipient KEM provably secure in the standard model and conclude the paper by proposing a notion of *direct reproducibility* which enables tighter security reductions.

Keywords. Randomness Reuse. Multi-Recipient. Hybrid Encryption.

1 Introduction

Generating randomness for cryptographic applications is a costly and security-critical operation. It is often assumed in security analysis that parameters are sampled from a perfect uniform distribution and are handled securely. Moreover, various operations performed by a cryptographic algorithm depend on the random coins used within the algorithm. These operations, such as a group exponentiation, can be quite costly and prevent the use of the scheme in constrained devices. Therefore, minimising the amount of fresh randomness required in cryptographic algorithms is important for their overall efficiency and security.

One approach to minimise this problem is to reuse randomness across multiple instantiations of cryptographic algorithms, namely in the context of batch operations where (possibly different) messages are encrypted to multiple recipients. This avenue must be pursued with caution, since randomness reuse may hinder the security of cryptographic schemes. However, when possible, this technique allows for significant savings in processing load and bandwidth, since partial results (and even ciphertext elements) can be shared between multiple instances of a cryptographic algorithm.

Examples of this method are the multi-recipient encryption schemes proposed by Kurosawa [11], the mKEM scheme by Smart in [12], the certificateless encryption scheme in [3] where randomness is shared between the identity-based and public-key components, and the cryptographic workflow scheme in [2].

Bellare et al. [4], building on the work of Kurosawa [11], systematically study the problem of reusing randomness. The authors examine multi-recipient encryption, and consider the particular case of constructing such schemes by running multiple instances of a public-key encryption (PKE) scheme, whilst sharing randomness across them. An interesting result in this work is a general method for identifying PKE schemes that are secure when used in this scenario. Schemes which satisfy the so-called *reproducibility test* are guaranteed to permit a hybrid argument proof strategy which is generally captured in a *reproducibility theorem*. Bellare et al. later leveraged on these results to propose a stateful encryption framework [5] which enables more efficient encryption operations.

In this paper we extend the above theoretical framework supporting the reuse of randomness to construct multi-recipient encryption schemes. The main contribution of this paper is a more permissive test that permits constructing a wider class of efficient *single message* multi-recipient schemes. Of particular interest are the optimised modified versions of the ElGamal and Cramer-Shoup multi-recipient encryption schemes briefly mentioned by Kurosawa in the final section of [11]. We show that these schemes do not fit the randomness reuse framework originally proposed by Bellare et al. and propose extensions to the original definitions which capture these as well as other similar schemes. The technique that we employ to prove the main Theorem (Theorem 1) deviates from that of [4] and may be of independent interest in other contexts.

We then turn our attention to the KEM/DEM paradigm and focus on key encapsulation mechanisms [7]. Adaptation of the results in [4] is straightforward if one focuses on multi-recipient KEMs generating independent keys for each recipient. The interesting case arises when one considers single key multi-recipient KEMs. To construct these schemes efficiently by reusing randomness, we define the notion of *public key independent* KEM[3]. However, we find that if such a KEM satisfies an appropriate modification of the reproducibility test of Bellare et al. it cannot be secure. To compensate for this negative result, we propose an alternative generic construction of efficient single key multi-recipient KEMs based on weakly secure and weakly reproducible PKEs. We also present a concrete efficient construction, which is secure in the standard model.

The paper is structured as follows. In Section 2 we define what we mean by secure multi-recipient PKEs and full reproducibility, and go on to define weak reproducibly. Concrete schemes are analysed in Section 3. In Section 4 we examine extensions of the previous results to KEMs. Finally, in Section 5

[3] This closely related to the notion of *partitioned* identity-based KEMs independently proposed by Abe et al. in [1].

we propose a new approach to the reproducibility generalisation that captures tighter security reductions, discuss the results we obtained using this strategy, and conclude with some associated open problems.

2 A New Notion of Reproducibility

2.1 Multi-Recipient PKEs

An n-Multi-Recipient PKE (n-MR-PKE) [4] is defined similarly to a standard PKE scheme, with two exceptions: (1) The key generation algorithm is parameterised with a domain parameter I which ensures compatibility between users' key pairs and various spaces[4]. We denote the randomness, message and ciphertext spaces by the letters $\mathcal{R}$, $\mathcal{M}$ and $\mathcal{C}$ respectively; and (2) The encryption algorithm takes a list of n message/public key tuples and outputs a list of n ciphertexts.

Given a PKE scheme we can build the *associated* n-MR-PKE as follows. The key generation and decryption algorithms are identical to the underlying PKE (which we call the *base* PKE). Encryption is defined naturally by running multiple parallel instances of the base PKE encryption algorithm. If the randomness tapes in all instantiations are constant, the resulting n-MR-PKE scheme is called *randomness reusing.* In case there are common parameters that may be shared by all public keys to improve overall efficiency[5], these are included in the domain parameter I. The formal security model for an n-MR-PKE, as defined in [4], considers the possibility of insider attacks by allowing the adversary to corrupt some of the users by maliciously choosing their public keys. This ensures that security is still in place between the legitimate recipients, or in other words, that there is no "cross-talk" between the ciphertexts intended for different recipients.

In this work we are interested in a special case of n-MR-PKEs where the same message is sent to all recipients. We refer to this special case as *single message* (n-SM-PKE for simplicity), and note that such a scheme can also take advantage of randomness reuse. This specific case is a recurring use-case of n-MR-PKEs in practice, and one could ask if the single message restriction makes it any easier to construct n-MR-PKE schemes. More precisely, is there a wider range of schemes that can be used to construct efficient n-SM-PKEs through randomness reuse?

Below is the simplified security model for n-SM-PKEs. There is an important difference to the n-MR-PKE model: the adversary is no longer able to corrupt users. The reason for this is that, since all recipients will be getting the same message, there is no need to enforce security across the individual ciphertexts. We will also see in Section 4 that this weaker model is particularly relevant in the hybrid encryption scenario. Throughout the game, the adversary also has access to $\mathcal{O}_1$ and $\mathcal{O}_2$, which denote a set of oracles, as follows:

[4] This parameter is generated once for all users and henceforth, unless specifically stated otherwise, we leave its generation as implicit to simplify notation.

[5] For example in a Diffie–Hellman based scheme, this might include a domain modulus and generator which all parties use to create key pairs.

– If $\mathtt{atk} = \mathtt{CPA}$ then $\mathcal{O}_1 = \mathcal{O}_2 = \text{NULL}$;
– If $\mathtt{atk} = \mathtt{CCA}$[6] then $\mathcal{O}_1$ is a set of decryption oracles one for each $\mathtt{PK}_i$ and $\mathcal{O}_2$ is same as $\mathcal{O}_1$ except that no component C_i^* of the challenge ciphertext C^* can be submitted to an oracle corresponding to $\mathtt{PK}_i$.

```
IND-atk
 1. For $i = 1, \ldots, n$
      $(\mathtt{SK}_i, \mathtt{PK}_i) \leftarrow \mathbb{G}_{n-\mathtt{SM}-\mathtt{PKE}}(I)$
 2. $(M_0, M_1, s) \leftarrow A_1^{\mathcal{O}_1}(\mathtt{PK}_1, \ldots, \mathtt{PK}_n)$
 3. $b \leftarrow \{0, 1\}$
 4. $C^* \leftarrow \mathbb{E}_{n-\mathtt{SM}-\mathtt{PKE}}(M_b, (\mathtt{PK}_i)_{i=1}^n)$
 5. $b' \leftarrow A_2^{\mathcal{O}_2}(C^*, s)$
```

$$\mathrm{Adv}_{n-\mathtt{SM}-\mathtt{PKE}}^{\mathtt{IND}-\mathtt{atk}}(A) := |2\Pr[b' = b] - 1|.$$

2.2 Weak Reproducibility for PKEs

Definition 1. *A PKE scheme is* fully reproducible *if there exists a probabilistic polynomial time (PPT) algorithm R such that the following experiment returns* 1 *with probability* 1*.*

1. $(\mathtt{PK}, \mathtt{SK}), (\mathtt{PK}', \mathtt{SK}') \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$
2. $r \leftarrow \mathcal{R}_{\mathtt{PKE}}(I)$*;* $M, M' \leftarrow \mathcal{M}_{\mathtt{PKE}}(I)$
3. $C \leftarrow \mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}; r)$
4. If $R(\mathtt{PK}, C, M', \mathtt{PK}', \mathtt{SK}') = \mathbb{E}_{\mathtt{PKE}}(M', \mathtt{PK}'; r)$ *return* 1*, else return* 0

It is shown in [4] that an IND-atk secure PKE scheme satisfying the above definition can be used to construct an efficient IND-atk secure n-MR-PKE, by reusing randomness across n PKE instances. This result is interesting in itself, as it constitutes a generalisation of a proof strategy which can be repeated, almost without change, for all schemes satisfying the reproducibility test. This is a hybrid argument where an n-MR-PKE attacker is used to construct an attacker against the base scheme. The reproducibility algorithm generalises the functionality required to extend a challenge in the single-user PKE security game, to construct a complete challenge for the n-MR-PKE security game.

The SM security model proposed in the previous section is somewhat simpler than the original model in [4], so it is conceivable that a wider range of PKE schemes can be used to construct secure n-SM-PKEs, namely efficient randomness reusing ones. Hence, we are interested in defining a less restrictive version of reproducibility that permits determining whether a PKE scheme can be safely used in the single message scenario, even if it does not satisfy the full reproducibility test above. The following definition achieves this.

Definition 2. *A PKE scheme is* weakly reproducible *(wREP) if there exists a PPT algorithm R such that the following experiment returns* 1 *with probability* 1*.*

[6] In this paper we use IND-CCA to denote a fully adaptive chosen ciphertext attack sometimes denoted by IND-CCA2.

```
1. $(\mathtt{SK}_1, \mathtt{PK}_1), (\mathtt{SK}_2, \mathtt{PK}_2), (\mathtt{SK}_3, \mathtt{PK}_3) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$
2. $r \leftarrow \mathcal{R}_{\mathtt{PKE}}(I)$; $M, M' \leftarrow \mathcal{M}_{\mathtt{PKE}}(I)$
3. $C_1 \leftarrow \mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}_1; r)$; $C_2 \leftarrow \mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}_2; r)$
4. If $R(\mathtt{PK}_1, C_1, M, \mathtt{PK}_2, \mathtt{SK}_2) \neq C_2$ return $0$
5. If $R(\mathtt{PK}_1, C_1, M', \mathtt{PK}_3, \mathtt{SK}_3) \neq R(\mathtt{PK}_2, C_2, M', \mathtt{PK}_3, \mathtt{SK}_3)$ return $0$, else return $1$
```

Similarly to the original REP definition, the wREP definition follows from the generalisation of the hybrid argument which allows reducing the security of a randomness reusing n-SM-PKE to that of its base scheme. The intuition behind the definition is as follows. We are dealing which single message schemes. Therefore we only require correct reproduction when the two messages are the same. When the messages are different, we relax the definition and require only that R is source-$\mathtt{PK}$ independent (condition 5). This property is easy to check.

To see why more schemes might satisfy this definition, note that R is not even required to produce a valid ciphertext when the messages are different. In Section 3 we analyse specific PKE schemes and give a formal separation argument which establishes that the wREP definition is meaningful: there are schemes which satisfy this definition and which are not fully reproducible. Conversely, it is easy to check that the following Lemma holds, and that wREP fits in the original reproducibility generalisation.

Lemma 1. *Any scheme which is fully reproducible is also weakly reproducible.*

The following theorem shows that the wREP definition is sufficient to guarantee n-SM-PKE security. The proof uses techniques which are somewhat different from that in [4] and may be of independent interest in other contexts.

Theorem 1. *The associated randomness reusing n-SM-PKE scheme of an IND-atk public-key encryption scheme is IND-atk secure if the base PKE is weakly reproducible. More precisely, any PPT attacker A with non negligible advantage against the randomness reusing n-SM-PKE scheme can be used to construct attackers B and D against the base PKE, such that:*

$$\mathrm{Adv}^{\mathtt{IND-atk}}_{n-\mathtt{SM-PKE}}(A) \leq n \cdot \mathrm{Adv}^{\mathtt{IND-atk}}_{\mathtt{PKE}}(B) + (n-1) \cdot \mathrm{Adv}^{\mathtt{IND-atk}}_{\mathtt{PKE}}(D).$$

Proof. We present the argument for the IND-CPA case, since the IND-CCA version is a straightforward extension where simulators use their knowledge of secret keys and external oracles to answer decryption queries. We begin by defining the following experiment, parameterised with an IND-atk attacker A against the randomness reusing n-SM-PKE scheme, and indexed by a coin b and an integer l such that $0 \leq l \leq n$.

```
$\mathtt{Exp}_{l,b}(A)$
1. $(\hat{\mathtt{PK}}, \hat{\mathtt{SK}}) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$
2. $(\mathtt{PK}_i, \mathtt{SK}_i) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$, for $1 \leq i \leq n$
3. $(M_0, M_1, s) \leftarrow A_1(\mathtt{PK}_1, \ldots, \mathtt{PK}_n)$
4. $\hat{C} \leftarrow \mathbb{E}_{\mathtt{PKE}}(M_b, \hat{\mathtt{PK}})$
5. $C_i \leftarrow R(\hat{\mathtt{PK}}, \hat{C}, M_1, \mathtt{PK}_i, \mathtt{SK}_i)$, for $1 \leq i \leq l$
6. $C_i \leftarrow R(\hat{\mathtt{PK}}, \hat{C}, M_0, \mathtt{PK}_i, \mathtt{SK}_i)$, for $l+1 \leq i \leq n$
7. $c \leftarrow A_2(C_1, \ldots, C_n, s)$
8. Return $c$
```

Looking at this experiment, and recalling from the wREP definition that R performs perfect reproduction when the input message is the same as that inside the input ciphertext, we can write the following equation:

$$\mathrm{Adv}_{n-\mathtt{SM}-\mathtt{PKE}}^{\mathtt{IND}-\mathtt{atk}}(A) = |\Pr[\mathtt{Exp}_{n,1}(A) = 1] - \Pr[\mathtt{Exp}_{0,0}(A) = 1]|.$$

This follows from the advantage definition, and fact that when $(l, b) = (n, 1)$, then $\hat{C}$ will encapsulate M_1, and all challenge ciphertexts are reproduced with M_1, which gives rise to a valid n-IND-atk ciphertext encapsulating M_1. The same happens for M_0, when $(l, b) = (0, 0)$.

We now define a probabilistic algorithm B which tries to break the base PKE scheme using A.

```
$B_1(\bar{\mathtt{PK}})$

1. Select $l$ at random such that $1 \le l \le n$
2. $(\mathtt{PK}_l, \mathtt{SK}_l) \leftarrow (\bar{\mathtt{PK}}, \perp)$
3. $(\mathtt{PK}_i, \mathtt{SK}_i) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$, for $1 \le i \le n$ and $i \ne l$
4. $(M_0, M_1, s) \leftarrow A_1(\mathtt{PK}_1, \dots, \mathtt{PK}_n)$
5. Return $(M_0, M_1, (M_0, M_1, l, \bar{\mathtt{PK}}, (\mathtt{PK}_1, \mathtt{SK}_1), \dots, (\mathtt{PK}_n, \mathtt{SK}_n), s))$

$B_2(\bar{C}, (M_0, M_1, l, \bar{\mathtt{PK}}, (\mathtt{PK}_1, \mathtt{SK}_1), \dots, (\mathtt{PK}_n, \mathtt{SK}_n), s))$

1. $C_l \leftarrow \bar{C}$
2. $C_i \leftarrow R(\bar{\mathtt{PK}}, \bar{C}, M_1, \mathtt{PK}_i, \mathtt{SK}_i)$, for $1 \le i \le l - 1$
3. $C_i \leftarrow R(\bar{\mathtt{PK}}, \bar{C}, M_0, \mathtt{PK}_i, \mathtt{SK}_i)$, for $l + 1 \le i \le n$
4. $\hat{b} \leftarrow A_2(C_1, \dots, C_n, s)$
5. Return $\hat{b}$
```

To continue the proof, we will require the following two Lemmas, which we shall prove shortly.

Lemma 2. *For $1 \le l \le n - 1$, and for any PPT adversary A, there is an adversary D such that*

$$\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND}-\mathtt{atk}}(D) = |\Pr[\mathtt{Exp}_{l,1}(A) = 1] - \Pr[\mathtt{Exp}_{l,0}(A) = 1]|.$$

Lemma 3. *For $1 \le i \le n$, the output of algorithm B and that of $\mathtt{Exp}_{l,b}(A)$ are related as follows:*

$$\Pr[\hat{b} = 1 | l = i \wedge \bar{b} = 1] = \Pr[\mathtt{Exp}_{i,1}(A) = 1]$$
$$\Pr[\hat{b} = 1 | l = i \wedge \bar{b} = 0] = \Pr[\mathtt{Exp}_{i-1,0}(A) = 1].$$

Here $\bar{b}$ is the hidden bit in $\bar{C}$.

Let us now analyse overall the probability that B returns 1, conditional on the value of the hidden challenge bit $\bar{b}$. Since B choses l uniformly at random, we may write:

$$\Pr[\hat{b}=1|\bar{b}=1] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\hat{b}=1|l=i \wedge \bar{b}=1]$$
$$\Pr[\hat{b}=1|\bar{b}=0] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\hat{b}=1|l=i \wedge \bar{b}=0].$$

Taking advantage of Lemma 3, we can rewrite these as:

$$\Pr[\hat{b}=1|\bar{b}=1] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\mathtt{Exp}_{i,1}(A)=1]$$
$$\Pr[\hat{b}=1|\bar{b}=0] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\mathtt{Exp}_{i-1,0}(A)=1].$$

Subtracting the previous equations and rearranging the terms, we get

$$\begin{aligned} & n(\Pr[\hat{b}=1|\bar{b}=1] - \Pr[\hat{b}=1|\bar{b}=0]) - \\ & \qquad (\sum_{i=1}^{n-1} \Pr[\mathtt{Exp}_{i,1}(A)=1] - \sum_{i=1}^{n-1} \Pr[\mathtt{Exp}_{i,0}(A)=1]) \\ & \qquad\qquad = \Pr[\mathtt{Exp}_{n,1}(A)=1] - \Pr[\mathtt{Exp}_{0,0}(A)=1]. \end{aligned}$$

Considering the absolute values of both sides and using Lemma 2, we can write

$$n\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-atk}}(B) + (n-1)\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-atk}}(D) \geq \mathrm{Adv}_{n-\mathtt{SM-PKE}}^{\mathtt{IND-atk}}(A).$$

In other words

$$\mathrm{Adv}_{n-\mathtt{SM-PKE}}^{\mathtt{IND-atk}}(A) \leq (2n-1)\epsilon,$$

where ϵ is negligible and the theorem follows. □

We now prove the required lemmas.

Proof. (Lemma 2) We build an algorithm $D_l = (D_{1,l}, D_{2,l})$ which runs A in exactly the same conditions as it is run in $\mathtt{Exp}_{l,b}$, and which can be used to win the IND-atk game against the base PKE with an advantage which is the same as A's capability of distinguishing between $\mathtt{Exp}_{l,0}$ and $\mathtt{Exp}_{l,1}$.

```
$D_{1,l}(\bar{\mathtt{PK}})$

  1. $(\mathtt{PK}_i, \mathtt{SK}_i) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$, for $1 \leq i \leq n$
  2. $(M_0, M_1, s) \leftarrow A_1(\mathtt{PK}_1, \ldots, \mathtt{PK}_n)$
  3. Return $(M_0, M_1, (M_0, M_1, \bar{\mathtt{PK}}, (\mathtt{PK}_1, \mathtt{SK}_1), \ldots, (\mathtt{PK}_n, \mathtt{SK}_n), s))$
```

```
$D_{2,l}(\bar{C}, (M_0, M_1, \bar{\mathtt{PK}}, (\mathtt{PK}_1, \mathtt{SK}_1), \ldots, (\mathtt{PK}_n, \mathtt{SK}_n), s))$

  1. $C_i \leftarrow R(\bar{\mathtt{PK}}, \bar{C}, M_1, \mathtt{PK}_i, \mathtt{SK}_i)$, for $1 \leq i \leq l$
  2. $C_i \leftarrow R(\bar{\mathtt{PK}}, \bar{C}, M_0, \mathtt{PK}_i, \mathtt{SK}_i)$, for $l+1 \leq i \leq n$
  3. $\hat{b} \leftarrow A_2(C_1, \ldots, C_n, s)$
  4. Return $\hat{b}$
```

D simply uses the challenge public key $\bar{\mathtt{PK}}$ in place of $\hat{\mathtt{PK}}$ in the experiment, and uses the PKE challenge $\bar{C}$ in place of $\hat{C}$. Note that the only visible difference to the definition of $\mathtt{Exp}$ is that D does not know $\bar{\mathtt{SK}}$, which it does not need, and that the IND-atk hidden bit $\bar{b}$ is used in place of b. We can therefore write, for a given value of l:

$$\Pr[\hat{b} = 1 | \bar{b} = 1] = \Pr[\mathtt{Exp}_{l,1}(A) = 1]|$$
$$\Pr[\hat{b} = 1 | \bar{b} = 0] = \Pr[\mathtt{Exp}_{l,0}(A) = 1]|,$$

and consequently

$$\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-atk}}(D) = |\Pr[\mathtt{Exp}_{l,1}(A) = 1] - \Pr[\mathtt{Exp}_{l,0}(A) = 1]|.$$

□

Proof. (Lemma 3) We present here the proof for the first case in the Lemma, and leave the second case, which is proved using a similar argument, for Appendix A. The first case of the Lemma states that

$$\Pr[\hat{b} = 1 | l = i \wedge \bar{b} = 1] = \Pr[\mathtt{Exp}_{i,1}(A) = 1].$$

We must show that the probability distribution of the inputs presented to A is exactly the same in the scenarios corresponding to both sides of the equation above. This is trivially true for the public keys that A_1 receives, since all of them are independently generated using the correct algorithm. Regarding the challenge ciphertext that A_2 gets, we start by expanding the values of $(C_1, \ldots, C_n)$.

In $\mathtt{Exp}_{i,1}(A)$, we have $\hat{C} = \mathbb{E}_{\mathtt{PKE}}(M_1, \hat{\mathtt{PK}}; r)$ and:

$$C_j = R(\hat{\mathtt{PK}}, \hat{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } 1 \leq j \leq i$$
$$C_j = R(\hat{\mathtt{PK}}, \hat{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } i+1 \leq j \leq n.$$

On the other hand, in $B_2(\bar{C}, \bar{s})$, given that $l = i$ and $\bar{b} = 1$ we have $\bar{C} = \mathbb{E}_{\mathtt{PKE}}(M_1, \bar{\mathtt{PK}}; r)$ and:

$$C_i = \bar{C}$$
$$C_j = R(\bar{\mathtt{PK}}, \bar{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } 1 \leq j \leq i-1$$
$$C_j = R(\bar{\mathtt{PK}}, \bar{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } i+1 \leq j \leq n.$$

To show that the distributions are identical, we split the argument in three parts and fix the values of all random variables, considering the case where the public keys provided to A in both cases are the same, and that the implicit randomness in both $\hat{C}$ and $\bar{C}$ is the same r. We show that the resulting challenge ciphertexts in both cases are exactly the same:

- $j = i$: Note that in the second scenario we have $C_i = \bar{C}$, while in the first scenario we have $C_i = R(\hat{\mathtt{PK}}, \hat{C}, M_1, \mathtt{PK}_i, \mathtt{SK}_i)$. Since $\hat{C}$ encrypts M_1, the result of R is perfect and equal to $\mathbb{E}_{\mathtt{PKE}}(M_1, \mathtt{PK}_i; r) = \bar{C}$.

- $j < i$: In this range, challenge components are identical in both scenarios: they are perfect reproductions $\mathbb{E}_{\mathtt{PKE}}(M_1, \mathtt{PK}_j; r)$, since M_1 is passed to R both in encrypted and plaintext form.
- $j > i$: In this range, challenge components are outputs of R, but in this case we cannot claim that they are identical without resorting to the properties of the wREP algorithm. For different message reproduction, condition 5 of Definition 2 ensures that

$$R(\hat{\mathtt{PK}}, \hat{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j) = R(\bar{\mathtt{PK}}, \bar{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j)$$

as required.

This means that the first case of the Lemma follows. □

3 Kurosawa's Efficient Schemes

In this section we analyse modified versions of ElGamal and Cramer-Shoup encryption schemes briefly mentioned by Kurosawa [11] as a way to build efficient single-message multiple-recipient public key encryption schemes. These schemes permit establishing a separation between the original reproducibility notion proposed by Bellare et al. and the one we introduced in the previous section.

3.1 Modified ElGamal

The *modified ElGamal* encryption scheme is similar to the ElGamal encryption scheme and operates as follows. The key generation algorithm $\mathbb{G}_{\mathtt{PKE}}(I)$ on input $I := (p, g)$ returns the key pair $(\mathtt{SK}, \mathtt{PK}) = (1/x, g^x)$ for $x \leftarrow \mathbb{Z}_p^*$. The encryption algorithm $\mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}; r)$ returns the ciphertext $(u, v) := ((g^x)^r, m \cdot g^r)$ for $r \leftarrow \mathbb{Z}_p^*$. The decryption algorithm $\mathbb{D}_{\mathtt{PKE}}(u, v, 1/x)$ returns the message $m := v/(u^{1/x})$.

Theorem 2 establishes the security of the modified ElGamal scheme as well as its weak reproducibility property. Theorem 3 shows that modified ElGamal establishes a separation between the notions of full and weak reproducibility.

Theorem 2. *Modified ElGamal is (1) IND-CPA secure under the decisional Diffie–Hellman assumption, and (2) weakly reproducible.*

Proof. (1) The proof is similar to that for the ElGamal encryption scheme.

(2) The weak reproducibility algorithm R on input $(g^x, u, v, m', g^{x'}, 1/x')$ returns $((v/m')^{x'}, v)$. We now check that R satisfies the two properties required by the wREP definition. If $m' = m$, then $v/m' = (m \cdot g^r)/m = g^r$ and the output is a valid encryption of m' under $g^{x'}$ using random coins r. Note also that R's output does not dependent on the public key g^x and hence the second property is also satisfied. □

Theorem 3. *The modified ElGamal encryption is not fully reproducible under the CDH assumption.*

Proof. Let $(g, g^a, g^b) \in G^3$ denote the CDH problem instance. Our goal is to compute g^{ab}. The reproduction algorithm on input $(p, g, g^x, g^{rx}, m \cdot g^r, m', g^y, 1/y)$ outputs $(g^{ry}, m' \cdot g^r)$. We pass to R the input $(p, g^a, g, g^b, 1, 1, g^a, 1)$ which could be written as $(p, h, h^{1/a}, h^{b \cdot 1/a}, 1, 1, h, 1)$ where $h = g^a$. Note that since R succeeds with probability 1, it will run correctly on the above input instance even though its distribution is far away from those that R takes. Here implicitly we have $x = 1/a$, from $rx = b/a$ we get $r = b$, and $m = h^{-b}$. Hence the first component of the output of R will be $(h^1)^b = g^{ab}$. □

3.2 Modified Cramer-Shoup

Another construction of an efficient n-SM-PKE hinted at by Kurosawa in [11] is based on the CS1a encryption scheme of Cramer and Shoup [7], modified in an analogous manner to the ElGamal encryption scheme as presented in the previous section. In this case, the construction is secure against adaptive chosen ciphertext attacks in the standard model. Modified versions of the other schemes presented in [7] also pass the weak reproducibility test without being fully reproducible. The following scheme, however, is the most efficient as it shares $\hat{g}$ as a domain parameter.

The scheme is defined as follows. The domain parameter is $I := (p, g, \hat{g}, H)$, where g and $\hat{g}$ are generators of a group G of prime order p and H denotes a cryptographic hash function. The key generation algorithm $\mathbb{G}_{\texttt{PKE}}(I)$ outputs (x_1, x_2, y_1, y_2, z), a random element of $(\mathbb{Z}_p)^5$, as the secret key and the public key is set to be $(g^{x_1}\hat{g}^{x_2}, g^{y_1}\hat{g}^{y_2}, g^z)$. Encryption and decryption algorithms are:

```
$\mathbb{E}_{\texttt{PKE}}(m, \texttt{PK})$
  – $(e, f, h) \leftarrow \texttt{PK}$
  – $u \leftarrow \mathbb{Z}_p$
  – $\hat{a} \leftarrow \hat{g}^u$
  – $b \leftarrow h^u$
  – $c \leftarrow m \cdot g^u$
  – $v \leftarrow \mathrm{H}(\hat{a}, b, c)$
  – $d \leftarrow e^u f^{uv}$
  – Return $(\hat{a}, b, c, d)$
```

```
$\mathbb{D}_{\texttt{PKE}}((\hat{a}, b, c, d), \texttt{SK})$
  – $(x_1, x_2, y_1, y_2, z) \leftarrow \texttt{SK}$
  – $v \leftarrow \mathrm{H}(\hat{a}, b, c)$
  – $a \leftarrow b^{1/z}$
  – If $a^{x_1+y_1v}\hat{a}^{x_2+y_2v} \neq d$ return $\perp$
  – $m \leftarrow c/a$
  – Return $m$
```

Theorem 4. *The modified Cramer-Shoup scheme is (1) IND-CCA under the DDH assumption and (2) weakly reproducible.*

The proof of the first part of theorem is essentially that of the standard Cramer-Shoup scheme in [7]. We omit the proof details due to space limitations. Regarding the second part of Theorem 4, the weak reproduction algorithm is a natural extension of the one presented for modified ElGamal, returning

$$(\hat{a}, (c/m')^{z'}, c, (c/m')^{u(x'_1+v'y'_1)}\hat{a}^{u(x'_2+v'y'_2)}),$$

where $v' := H(\hat{a}, (c/m')^{z'}, c)$. A very important distinction in this case, however, is that the reproduction algorithm produces an output which may not be a valid ciphertext. In fact, for different message reproduction, the encryption algorithm would never be able to produce something like the resulting ciphertext. The returned output is, however, indistinguishable from a valid ciphertext under the decisional Diffie–Hellman assumption. The fact that the outputs of R may not be identically distributed to the outputs of the encryption algorithm, but merely indistinguishable, implies that the proof strategy presented in [4] does *not* apply for this scheme. On the other hand, note that the technique presented in the proof of Theorem 1 covers this and other similar schemes.

4 Hybrid Encryption

Practical applications of public key encryption are based on the hybrid paradigm, where public key techniques are used to encapsulate symmetric encryption keys. Formally, this is captured by the KEM/DEM framework [7]. Sharing randomness across multiple instances of a KEM may be justified, as before, as a means to achieve computational savings when performing batch operations. In this section we study randomness reuse for KEMs, a problem which has not been formally addressed by previous work.

The KEM primitive takes the recipient's public key as the single parameter to the encapsulation algorithm. In particular, unlike what happens in PKEs, one does not control the value of the encapsulated key: this is internally generated inside the KEM primitive, and its value depends only on the recipient's public key and on the randomness tape of the encapsulation algorithm. Since in this work we are interested on the role of randomness inside cryptographic algorithms, this leads us to the following categorisation of KEMs.

Definition 3. *A KEM scheme is* public key independent *if the following experiment returns* 1 *with probability* 1*.*

1. $(\mathtt{SK},\mathtt{PK}),(\mathtt{SK}',\mathtt{PK}') \leftarrow \mathbb{G}_{\mathtt{KEM}}(I)$
2. $r \leftarrow \mathcal{R}_{\mathtt{KEM}}(I)$
3. $(K,C) \leftarrow \mathbb{E}_{\mathtt{KEM}}(\mathtt{PK};r)$*;* $(K',C') \leftarrow \mathbb{E}_{\mathtt{KEM}}(\mathtt{PK}';r)$
4. *If* $K = K'$ *return* 1*, else return* 0

Considering what happens when one shares randomness across several instances of an encapsulation algorithm immediately suggests two independent adaptations of KEMs to the multi-recipient setting. The first, which we generically call multi-recipient KEMs (n-MR-KEMs), are functionally equivalent to the independent execution of n KEM instances, thereby associating an independent encapsulated secret key to each recipient. The second, which we will call single-key multi-recipient KEMs (n-SK-KEMs), given that the same secret key is encapsulated to all recipients, is akin to the mKEM notion introduced in [12].

Adaptation of the results in [4] to n-MR-KEMs is straightforward. The same is not true, however, for n-SK-KEMs. To justify why this is the case, we present a reproducibility test for KEMs in Definition 4. It is a direct adaptation of the reproducibility test for PKEs, considering that there is no message input to the encapsulation algorithm and that this returns also the encapsulated secret key. It can be easily shown that any KEM satisfying this test can be used to construct an efficient n-MR-KEM with randomness reuse.

Definition 4. *A KEM is called* reproducible *if there exists a PPT algorithm R such that the following experiment returns* 1 *with probability* 1.

1. $(\mathtt{SK}, \mathtt{PK}), (\mathtt{SK}', \mathtt{PK}') \leftarrow \mathbb{G}_{\mathtt{KEM}}(I)$
2. $r \leftarrow \mathcal{R}_{\mathtt{KEM}}(I)$
3. $(K, C) \leftarrow \mathbb{E}_{\mathtt{KEM}}(\mathtt{PK}; r)$*;* $(K', C') \leftarrow \mathbb{E}_{\mathtt{KEM}}(\mathtt{PK}'; r)$
4. *If* $R(\mathtt{PK}, C, \mathtt{PK}', \mathtt{SK}') = (K', C')$ *return* 1*, else return* 0

An n-SK-KEM, referred to as an mKEM in [12], is a key encapsulation mechanism which translates the hybrid encryption paradigm to the multi-cast setting: it permits encapsulating the same secret key to several different receivers. The point is that encrypting a single message to all these recipients can then be done using a single DEM instantiation based on that unique session key, rather than n different ones. This provides, not only computational savings, but also bandwidth savings, and captures a common use of hybrid encryption in practice. The natural security model for n-SK-KEMs is shown below.

```
IND-atk
 1. For $i = 1, \ldots, n$
      $(\mathtt{SK}_i, \mathtt{PK}_i) \leftarrow \mathbb{G}_{n-\mathtt{SK-KEM}}(I)$
 2. $s \leftarrow A_1^{\mathcal{O}_1}(\mathtt{PK}_1, \ldots, \mathtt{PK}_n)$
 3. $b \leftarrow \{0, 1\}$
 4. $(K_0, C^*) \leftarrow \mathbb{E}_{n-\mathtt{SK-KEM}}((\mathtt{PK}_i)_{i=1}^n)$
 5. $K_1 \leftarrow \{0, 1\}^\kappa$
 6. $b' \leftarrow A_2^{\mathcal{O}_2}(C^*, K_b, s)$
```

$$\mathrm{Adv}_{n-\mathtt{SK-KEM}}^{\mathtt{IND-atk}}(A) := |2\Pr[b' = b] - 1|.$$

As usual, the adversary also has access to $\mathcal{O}_1$ and $\mathcal{O}_2$, which denote a set of oracles, as follows:

- If $\mathtt{atk} = \mathtt{CPA}$ then $\mathcal{O}_1 = \mathcal{O}_2 =$ NULL;
- If $\mathtt{atk} = \mathtt{CCA}$ then $\mathcal{O}_1$ is a set of decapsulation oracles, one for each $\mathtt{PK}_i$, and $\mathcal{O}_2$ is same as $\mathcal{O}_1$ except that no component C_i^* of the challenge ciphertext C^* can be submitted to an oracle corresponding to $\mathtt{PK}_i$.

Unlike n-MR-KEMs there does not seem to be a natural way of constructing n-SK-KEMs from single-recipient KEMs. The fact that the same key should be encapsulated for all recipients makes public key independent KEMs the only possible candidates to be used as base KEMs. However, any public key independent scheme which satisfies a reproducibility test such as that in Definition 4 must be insecure, as anyone would be able to use the reproducibility algorithm to obtain the secret key in an arbitrary ciphertext. In the following we show how the weak reproducibility notion for PKEs we obtained in Theorem 1 actually fills this apparent theoretical gap, as it permits capturing the efficient n-SK-KEMs constructions we have found in literature. We conclude this section proposing a concrete construction of an efficient n-SK-KEM secure in the standard model.

4.1 Generic Construction of n-SK-KEMs

One trivial way to build secure randomness reusing n-SK-KEMs is to use a secure weakly reproducible encryption scheme, and to set a random message to be the ephemeral key. However, the underlying encryption scheme must have the same security guarantees as those required for the KEM. A more practical way to build a fully secure n-SK-KEM is to use a weaker PKE through the following generic construction, which generalises the mKEM scheme proposed in [12] and extends a construction by Dent in [8]. The domain parameters and key generation algorithm are the same as those of the underlying PKE. Encapsulation and decapsulation algorithms are:

```
$\mathbb{E}_{n-\texttt{SK}-\texttt{KEM}}(\texttt{PK}_1, \ldots, \texttt{PK}_n)$
  – $M \leftarrow \mathcal{M}_{\texttt{PKE}}(I)$
  – $r \leftarrow H(M)$
  – For $i = 1 \ldots, n$
    $C_i \leftarrow \mathbb{E}_{\texttt{PKE}}(M, \texttt{PK}_i; r)$
  – $C \leftarrow (C_1, \ldots, C_n)$
  – $K \leftarrow \mathrm{KDF}(M)$
  – Return $(K, C)$
```

```
$\mathbb{D}_{n-\texttt{SK}-\texttt{KEM}}(C, \texttt{SK})$
  – $M \leftarrow \mathbb{D}_{\texttt{PKE}}(M, \texttt{SK})$
  – If $M =\perp$ return $\perp$
  – $r \leftarrow H(M)$
  – If $C \neq \mathbb{E}_{\texttt{PKE}}(M, \texttt{PK}; r)$ return $\perp$
  – $K \leftarrow \mathrm{KDF}(M)$
  – Return $K$
```

Here H and KDF are cryptographic hash functions. The security of this scheme is captured via the following theorem, proved in Appendix B.

Theorem 5. *The above construction is an IND-CCA secure n-SK-KEM, if the underlying PKE is IND-CPA and weakly reproducible, and if we model H and* KDF *as random oracles. More precisely, any PPT attacker A with non negligible advantage against the generic n-SK-KEM can be used to construct an attacker B against the base PKE, such that:*

$$\mathrm{Adv}^{\texttt{IND-CCA}}_{n-\texttt{SK}-\texttt{KEM}}(A) \leq 2n(q_H + q_K + q_D)\mathrm{Adv}^{\texttt{IND-CPA}}_{\texttt{PKE}}(B) + \epsilon,$$

where q_H, q_K and q_D are the number of queries the adversary makes to H, KDF *and decapsulation oracles and ϵ denotes a negligible quantity.*

The security argument for this construction has two parts. The first part establishes the one-way security of the n-SK-PKE scheme associated with the base PKE. This follows directly from the weak reproducibility theorem in Section 3.2 and the fact that one-wayness is implied by indistinguishability[7]. The second part builds on the previous result to achieve IND-CCA security in the n-SK-KEM setting, using a general construction laid out by Dent in [8]. In this construction one models the hash function H and KDF as random oracles and shows that the queries placed by any adversary with non-negligible advantage in breaking the n-SK-KEM scheme can be used to invert the one-wayness of the underlying n-SM-PKE scheme.

[7] We assume that various message spaces have exponential size in the security parameter.

The mKEM in [12] fits the general framework we introduced in this paper by instantiating the above construction with the ElGamal encryption scheme. The results in this work permit introducing two interesting enhancements over the mKEM in [12] if the above construction is instantiated with the modified ElGamal scheme:

- Stronger security guarantees disallowing benign malleability. The security model in [12] disallows decapsulation queries on any ciphertext which decapsulates to the same key as that implicit in the challenge.
- More efficient encryption algorithm, saving $n-1$ group operations.

4.2 An Efficient n-SK-KEM Secure in the Standard Model

In this section we propose an efficient n-SK-KEM scheme which is IND-CCA secure in the standard model. To the best of our knowledge, it is the first such construction to achieve this level of security and efficiency. The scheme is an adaptation of a KEM proposed by Cramer and Shoup in [7], which is public key dependent and therefore cannot be used as a black-box to construct an n-SK-KEM. The adapted scheme is defined as follows.

The domain parameter is $I := (p, g, \hat{g}, H, \mathrm{KDF})$, where g and $\hat{g}$ are generators of a group G of prime order p, H is a cryptographic hash function and KDF is a key derivation function. The key generation algorithm $\mathbb{G}_{n-\mathtt{SK}-\mathtt{KEM}}(I)$ outputs $\mathtt{SK} = (x_1, x_2, y_1, y_2, z)$, a random element of $\mathbb{Z}_p^4 \times \mathbb{Z}_p^*$, as the secret key and $\mathtt{PK} = (e, f, h) = (g^{x_1}\hat{g}^{x_2}, g^{y_1}\hat{g}^{y_2}, g^z)$ as the public key. Encapsulation and decapsulation algorithms are:

```
$\mathbb{E}_{n-\mathtt{SK}-\mathtt{KEM}}(\mathtt{PK}_1, \ldots, \mathtt{PK}_n)$
  – $u \leftarrow \mathbb{Z}_p$
  – $\hat{a} \leftarrow \hat{g}^u$
  – $b \leftarrow g^u$
  – $K \leftarrow \mathrm{KDF}(\hat{a}, b)$
  – For $1 \leq i \leq n$
      $(e_i, f_i, h_i) \leftarrow \mathtt{PK}_i$
      $a_i \leftarrow h_i^u$
      $v_i \leftarrow H(\hat{a}, a_i)$
      $d_i \leftarrow e_i^u f_i^{uv_i}$
  – Return $(K, \hat{a}, a_1, \ldots, a_n, d_1, \ldots, d_n)$
```

```
$\mathbb{D}_{n-\mathtt{SK}-\mathtt{KEM}}((\hat{a}, a, d), \mathtt{SK})$
  – $(x_1, x_2, y_1, y_2, z) \leftarrow \mathtt{SK}$
  – $v \leftarrow H(\hat{a}, a)$
  – $b \leftarrow a^{1/z}$
  – If $b^{x_1+vy_1}\hat{a}^{x_2+vy_2} \neq d$
      return $\perp$
  – $K \leftarrow \mathrm{KDF}(\hat{a}, b)$
  – Return $K$
```

A proof that the n-SK-KEM scheme proposed above is IND-CCA secure under the decisional Diffie–Hellman assumption, provided that the hash function is target collision resistant, and that the KDF function is entropy smoothing will appear in the full version of this paper.

5 Tighter Reductions

In [4] the authors present tighter security reductions for the multi-recipient randomness reusing schemes associated with the ElGamal and Cramer-Shoup encryption schemes. These reductions rely on the random self-reducibility property of the DDH problem. The tighter reductions are achieved by using this property to unfold a single DDH problem instance, so that it can be embedded in the multiple challenge ciphertext components required in the multiple user setting. In these proofs, the extra public keys and challenge ciphertexts required in the reduction are chosen in a controlled manner. For instance, one public key might have a known discrete logarithm with respect to another. The following notion of reproducibility could be viewed as a generalisation of this type of proof strategy for tight reductions.

Definition 5. *A PKE scheme is called* directly reproducible *(dREP) if there exists a set of PPT algorithms $R = (R_1, R_2, R_3)$ such that the following experiment returns* 1 *with probability* 1*.*

1. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}_{\mathtt{PKE}}(I)$
2. $(\mathtt{PK}', s) \leftarrow R_1(\mathtt{PK})$
3. $M \leftarrow \mathcal{M}_{\mathtt{PKE}}(I)$*;* $r \leftarrow \mathcal{R}_{\mathtt{PKE}}(I)$
4. $C \leftarrow \mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}; r)$*;* $C' \leftarrow \mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}'; r)$
5. *If* $C' \neq R_2(C, s)$ *return* 0
6. *If* $C \neq R_3(C', s)$ *return* 0*, else return* 1

We require the distributions of $\mathtt{PK}$ *and* $\mathtt{PK}'$ *to be identical.*

Note that R_1 controls the generation of the public keys and the main reproduction algorithm (R_2) may take advantage of the state information produced by the first algorithm. The existence of the third algorithm is required for the simulation of decryption oracles for CCA secure schemes. It is easy to verify that

Theorem 6. *The associated randomness reusing n-SM-PKE scheme of a directly reproducible and IND-atk secure encryption scheme is also secure in the IND-atk sense. More precisely, any PPT attacker A against the randomness reusing n-SM-PKE scheme can be used to build an attacker B against the base scheme, such that:*

$$\mathrm{Adv}^{\mathtt{IND-atk}}_{n-\mathtt{SM-PKE}}(A) \leq \mathrm{Adv}^{\mathtt{IND-atk}}_{\mathtt{PKE}}(B).$$

The above notion of reproducibility, not only permits deriving tighter security reductions, but also gives rise to a new test for detecting additional schemes which allow randomness reuse. In fact, it can be shown that a modified version of the escrow ElGamal encryption scheme is directly but not weakly reproducible (see Appendix C).

Furthermore, unlike weak and full reproducibility, this new notion respects the Fujisaki-Okamoto transformation [9] for building IND-CCA secure schemes, as it does not explicitly handle the encrypted message. It therefore establishes a new set of chosen-ciphertext secure single message multi-recipient schemes with tight security reductions in the random oracle model.

Direct reproducibility also poses an interesting problem, which concerns public key encryption schemes with chosen ciphertext security in the standard model. In particular, the case of the Cramer-Shoup encryption scheme remains open, as we were unable to construct the required reproduction algorithms. We leave it as an open problem to find such an algorithm, or to design an analogous reproducibility test which admits encryption schemes which are IND-CCA secure in the standard model.

References

1. M. Abe, Y. Cui, H. Imai and E. Kiltz. Efficient Hybrid Encryption from ID-Based Encryption. *Cryptology ePrint Archive*, Report 2007/023, 2007.
2. S.S. Al-Riyami, J. Malone-Lee and N.P. Smart. Escrow-Free Encryption Supporting Cryptographic Workflow. *International Journal of Information Security*, Vol 5:217–230, 2006.
3. S.S. Al-Riyami and K.G. Paterson. Certificateless Public-Key Cryptography. *Advances in Cryptology – ASIACRYPT 2003*, LNCS 2894:452–473. Springer-Verlag, 2003.
4. M. Bellare, A. Boldyreva and J. Staddon. Randomness Re-Use in Multi-recipient Encryption Schemes. *Public Key Cryptography – PKC 2003*, LNCS 2567:85–99. Springer-Verlag, 2003.
5. M. Bellare, T. Kohno and V. Shoup. Stateful Public-Key Cryptosystems: How to Encrypt with One 160-bit Exponentiation. *3th ACM Conference on Computer and Communications Security – CCS*, ACM, 2006.
6. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32:586–615. 2003.
7. R. Cramer and V. Shoup. A Practical Public-Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. *Advances in Cryptology – CRYPTO '98*, LNCS 1462:13–25. Springer-Verlag, 1998.
8. A.W. Dent. A Designer's Guide to KEMs. *Cryptography and Coding*, LNCS 2898:133–151. Springer-Verlag, 2003.
9. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *Advances in Cryptology – CRYPTO '99*, LNCS 1666:537–554. Springer-Verlag, 1999.
10. J. Hastad. Solving Simultaneous Modular Equations of Low Degree. *SIMA Journal on Computing*, Vol. 17, No. 2, April 1988.
11. K. Kurosawa. Multi-Recipient Public-Key Encryption with Shortened Ciphertext. *5th International Workshop on Practice and Theory in Public Key Cryptosystems – PKC 2002*, LNCS 2274:48–63, Springer-Verlag, 2002.
12. N.P. Smart. Efficient Key Encapsulation to Multiple Parties. *Security in Communication Networks*, LNCS 3352:208–219. Springer-Verlag, 2005.

A – Proof of the Second Case of Lemma 3

Proof. We now prove the Lemma for the case

$$\Pr[\hat{b} = 1 | l = i \wedge \bar{b} = 0] = \Pr[\texttt{Exp}_{i-1,0}(A) = 1].$$

The argument is similar to the previous case. We must show that the probability distribution of the inputs presented to A is exactly the same in the scenarios corresponding to both sides of the equation above. This is trivially true for the public keys that A_1 receives, since all of them are independently generated using the correct algorithm. Regarding the challenge ciphertext that A_2 gets, we start by expanding the values of $(C_1, \ldots, C_n)$.

In $\mathtt{Exp}_{i-1,0}(A)$, we have $\hat{C} = \mathbb{E}_{\mathtt{PKE}}(M_0, \hat{\mathtt{PK}}; r)$ and

$$\begin{aligned} C_j &= R(\hat{\mathtt{PK}}, \hat{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } 1 \leq j \leq i-1 \\ C_j &= R(\hat{\mathtt{PK}}, \hat{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } i \leq j \leq n. \end{aligned}$$

On the other hand, in $B_2(\bar{C}, \hat{s})$, given that $l = i$ and $\bar{b} = 0$ we have $\bar{C} = \mathbb{E}_{\mathtt{PKE}}(M_0, \mathtt{PK}; r)$ and

$$\begin{aligned} C_i &= \bar{C} \\ C_j &= R(\mathtt{PK}, \bar{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } 1 \leq j \leq i-1 \\ C_j &= R(\mathtt{PK}, \bar{C}, M_0, \mathtt{PK}_j, \mathtt{SK}_j) \text{ for } i+1 \leq j \leq n. \end{aligned}$$

To show that the distributions are identical, we split the argument in three parts and fix the values of all random variables, considering the case where the public keys provided to A in both cases are the same, and that the implicit randomness in both $\hat{C}$ and $\bar{C}$ is the same r. We show that the resulting challenge ciphertexts in both cases are exactly the same:

- $j = i$: Note that in the second scenario we have $C_i = \bar{C}$, while in the first scenario we have $C_i = R(\hat{\mathtt{PK}}, \hat{C}, M_0, \mathtt{PK}_i, \mathtt{SK}_i)$. Since $\hat{C}$ encrypts M_0, the result of R is perfect and equal to $\mathbb{E}_{\mathtt{PKE}}(M_0, \mathtt{PK}_i; r) = \bar{C}$.
- $j < i$: In this range, challenge components are outputs of R, but in this case we cannot claim that they are identical without resorting to the properties of R described in Definition 2 for different message reproduction, which ensure that
$$R(\hat{\mathtt{PK}}, \hat{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j) = R(\bar{\mathtt{PK}}, \bar{C}, M_1, \mathtt{PK}_j, \mathtt{SK}_j)$$
as required.
- $j > i$: In this range, challenge components are identical in both scenarios: they are perfect reproductions $\mathbb{E}_{\mathtt{PKE}}(M_0, \mathtt{PK}_j; r)$, since M_0 is passed to R both in encrypted and plaintext form.

This means that the second case of the Lemma follows. □

B – Proof of Theorem 5

Proof. Let A denote an IND-CCA adversary against the generic construction with non-negligible advantage. Modelling hash functions as random oracles, we construct an algorithm B with non-negligible advantage in the OW-CPA game for the n-SM-PKE. One-way security notion can be easily adapted to multi-recipient schemes. Note that one-wayness of an n-SM-PKE is not necessarily implied by the one-wayness of its base PKE [10]. However, since indistinguishability implies one-wayness and indistinguishability property is inherited from the base scheme due to the wREP property, we do have that the n-SM-PKE is OW-CPA. The concrete reduction is:

$$\mathrm{Adv}^{\mathtt{OW-CPA}}_{n-\mathtt{SM-PKE}}(A) \leq \mathrm{Adv}^{\mathtt{IND-CPA}}_{n-\mathtt{SM-PKE}}(B) + \epsilon_1 \leq n\mathrm{Adv}^{\mathtt{IND-CPA}}_{\mathtt{PKE}}(C) + \epsilon_2.$$

Here ϵ_1 and ϵ_2 are negligible quantities, assuming that the message space has a size super-polynomial in the security parameter. We omit the straightforward details of the proof.

On receiving the n public keys for the OW-CPA game, B passes these values on to algorithm A. During A's first stage, algorithm B replies to A's oracle queries as follows:

- H queries: B maintains a list $L \subseteq \mathcal{M}_{n-\mathtt{SM-PKE}}(I) \times \mathcal{R}_{n-\mathtt{SM-PKE}}(I)$ which contains at most q_H pairs (M, r). On input of M, if $(M, r) \in L$ then B returns r, otherwise it selects r at random from the appropriate randomness space, appends (M, r) to the list and returns r.
- KDF queries: B maintains a list $L_K \subseteq \mathcal{M}_{n-\mathtt{SM-PKE}}(I) \times \mathcal{K}_{n-\mathtt{SK-KEM}}(I)$ which contains at most $q_K + q_D$ pairs (M, k). On input of M, if $(M, k) \in L_K$ then B returns k, otherwise it selects k at random from the appropriate key space, appends (M, k) to the list and returns k.
- Decapsulation queries: on input $(C, \mathtt{PK})$, B checks for each $(M, r) \in L$ if $\mathbb{E}_{\mathtt{PKE}}(M, \mathtt{PK}; r) = C$; if such a pair exists, B calls the KDF simulation procedure on value M and returns the result to A. Otherwise B returns $\perp$.

At some point A will complete its first stage and return some state information. At this point, B calls the outside challenge oracle, and obtains a challenge ciphertext $(C_1, \ldots, C_n)$ on some unknown M^*. Algorithm B now checks if A has queried for decapsulation on a tuple $(C_\ell, \mathtt{PK}_\ell)$ during its first stage. If this is the case, algorithm B terminates. Otherwise it generates a random K^* and provides this to A along with the challenge ciphertext.

In the second stage, B answers A's oracle queries as in stage one. When A terminates, B randomly returns a message from L or L_K.

Now we analyse the probability that this answer is correct.

B's execution has no chance of success (event S_B) if it terminates at the end of A's first stage (event T). Therefore:

$$\Pr[S_B] = \Pr[S_B \wedge \neg T] = \Pr[S_B | \neg T] \Pr[\neg T]$$

Note that the challenge encapsulation is independent of A's view in the first stage, so that A could only have queried decapsulation for one of the challenge encapsulations by pure chance. However, the size of the valid encapsulation space for each public key is the same as the message space. This means that the probability that B continues to execute is

$$\Pr[\neg T] = 1 - \frac{q_D}{M}$$

where $M = |\mathcal{M}_{n-\mathtt{SM-PKE}}(I)|$.

Given that termination does not take place, B's simulation could be imperfect if one of the following events occur:

– Event E_1: The adversary places a decapsulation query for a valid ciphertext, and B returns $\perp$.
– Event E_2: The adversary queries H or KDF for the unknown M^* value.

Event E_1 occurs if A finds a valid ciphertext without querying H to obtain the randomness required to properly construct it. The probability of this is

$$\Pr[E_1] \leq \frac{q_D \gamma_n}{R},$$

where $R = |\mathcal{R}_{n-\texttt{SM}-\texttt{PKE}}(I)|$ and $\gamma_n = \gamma_n(I)$ is the least upper bound such that for every n-tuple $(\texttt{PK}_i)_{i=1}^n$, every $M \in \mathcal{M}_{n-\texttt{SM}-\texttt{PKE}}(I)$, every $j \in \{1, \ldots, n\}$ and every $C \in \mathcal{C}_{n-\texttt{SM}-\texttt{PKE}}(I)$ we have

$$|\{r \in \mathcal{R}_{n-\texttt{SM}-\texttt{PKE}}(I) : [\mathbb{E}_{n-\texttt{SM}-\texttt{PKE}}(M, (\texttt{PK}_i)_{i=1}^n; r)]_j = C\}| \leq \gamma_n(I).$$

This follows from the fact that, since H is modelled as a random oracle, A can only achieve this by guessing the randomness value. Moreover, the probability that a given randomness generates a valid ciphertext is at most γ_n/R and there are at most q_D such queries.

Note that we can write

$$\Pr[E_1 \vee E_2] \leq \Pr[E_1] + \Pr[E_2] \leq \frac{q_D \gamma_n}{R} + \Pr[E_2].$$

On the other hand, since A operates in the random oracle it can have no advantage if event $E_1 \vee E_2$ does not occur. Hence we can write

$$\begin{aligned}(1/2)\mathrm{Adv}_{n-\texttt{SK}-\texttt{KEM}}^{\texttt{IND}-\texttt{CCA}}(A) &= \Pr[S_A] - 1/2 \\ &= \Pr[S_A \wedge (E_1 \vee E_2)] + \Pr[S_A \wedge \neg(E_1 \vee E_2)] - 1/2 \\ &\leq \Pr[E_1 \vee E_2] + 1/2 - 1/2 \\ &\leq \Pr[E_2] + \frac{q_D \gamma_n}{R}.\end{aligned}$$

Now:

$$\begin{aligned}\mathrm{Adv}_{n-\texttt{SM}-\texttt{PKE}}^{\texttt{OW}-\texttt{CPA}}(B) = \Pr[S_B] &= \Pr[S_B|\neg T](1 - \frac{q_D}{M}) \\ &= \frac{1}{|L| + |L_K|} \Pr[E_2](1 - \frac{q_D}{M}) \\ &\geq \frac{1}{q_H + q_K + q_D}(\Pr[E_2] - \frac{q_D}{M}).\end{aligned}$$

and rearranging the terms

$$\Pr[E_2] \leq (q_H + q_K + q_D)\mathrm{Adv}_{n-\texttt{SM}-\texttt{PKE}}^{\texttt{OW}-\texttt{CPA}}(B) + \frac{q_D}{M}$$

Putting the above two results together we get:

$$\mathrm{Adv}_{n-\texttt{SK}-\texttt{KEM}}^{\texttt{IND}-\texttt{CCA}}(A) \leq 2(q_H + q_K + q_D)\mathrm{Adv}_{n-\texttt{SM}-\texttt{PKE}}^{\texttt{OW}-\texttt{CPA}}(B) + 2q_D(\frac{1}{M} + \frac{\gamma_n}{R}).$$

□

C – Direct and Weak Reproducibility Separation

Let us consider a modified version of a scheme proposed by Boneh and Franklin [6] known as escrow ElGamal. In this scheme the domain parameter is $I := (p, g, h)$ where $h = g^t$ for $t \leftarrow \mathbb{Z}_p^*$. The key generation algorithm outputs $(1/x, g^x)$ as the secret-public key pair for $x \leftarrow \mathbb{Z}_p^*$. The encryption algorithm on input a message m and a public key g^x returns $(u, v) := ((g^x)^r, m \cdot e(g, h)^r)$ where r is random in $\mathbb{Z}_p^*$. One is able to decrypt this ciphertext using the secret key $1/x$ by computing $m := v/e(u, h^{1/x})$. Here $e : G \times G \to G_T$ is a non-degenerate efficiently computable bilinear map [6].

The randomness reuse properties of this scheme are as follows.

Theorem 7. *The modified escrow ElGamal encryption scheme given above is (1) IND-CPA under the decisional bilinear Diffie–Hellman assumption; (2) directly reproducible; and (3) not weakly reproducible if the computational Diffie–Hellman assumption holds in G.*

Proof. (1) The security proof is analogous to that of escrow ElGamal.

(2) The direct reproducibility algorithm $R = (R_1, R_2, R_3)$ operates as follows. Algorithm R_1 on input a public key g^x returns $((g^x)^s, s)$ where s is a random element in $\mathbb{Z}_p^*$. The algorithm R_2 on input a ciphertext $(u, v) = (g^{xr}, m \cdot e(g, h)^r)$ and state information s returns (u^s, v). It is easily seen that R produces a valid encryption of m under $(g^x)^s$. Algorithm R_3 returns $(u^{1/s}, v)$. Note that the public key $(g^x)^s$ is identically distributed to public keys returned by the key generation algorithm.

(3) Let $(g, g^a, g^b) \in G^3$ denote the CDH problem instance. Our goal is to compute g^{ab}. The reproduction algorithm on input

$$(p, g, h, g^x, g^{rx}, m \cdot e(g, h)^r, m, g^y, 1/y)$$

outputs $(g^{ry}, m \cdot e(g, h)^r)$. To compute g^{ab} we pass to R the input

$$(p, g^a, g, g, g^b, e(g^b, g^a), 1, g^a, 1).$$

This could be written as:

$$(p, g', g'^{1/a}, g'^{1/a}, g'^{b \cdot 1/a}, e(g', g'^{1/a})^b, 1, g', 1),$$

where $g' = g^a$. Note again that since R succeeds with probability 1, it will run correctly on the above input instance. Here implicitly we have $x = 1/a$, and from $rx = b/a$ we have $r = b$, and $m = 1$. Therefore the first component of the output will be $(g'^1)^b = g^{ab}$. □