Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security

Brett Hemenway[*] Benoît Libert[†] Rafail Ostrovsky[‡]

Damien Vergnaud[§]

March 22, 2012

Abstract

Lossy encryption was originally studied as a means of achieving efficient and composable oblivious transfer. Bellare, Hofheinz and Yilek showed that lossy encryption is also selective opening secure. We present new and general constructions of lossy encryption schemes and of cryptosystems secure against selective opening adversaries.

We show that *every* re-randomizable encryption scheme gives rise to efficient encryptions secure against a selective opening adversary. We show that statistically-hiding 2-round Oblivious Transfer implies Lossy Encryption and so do smooth hash proof systems. This shows that private information retrieval and homomorphic encryption *both* imply Lossy Encryption, and thus Selective Opening Secure Public Key Encryption.

Applying our constructions to well-known cryptosystems, we obtain selective opening secure commitments and encryptions from the *Decisional Diffie-Hellman*, *Decisional Composite Residuosity* and *Quadratic Residuosity* assumptions.

In an indistinguishability-based model of chosen-ciphertext selective opening security, we obtain secure schemes featuring short ciphertexts under standard number theoretic assumptions. In a simulation-based definition of chosen-ciphertext selective opening security, we also handle non-adaptive adversaries by adapting the Naor-Yung paradigm and using the perfect zero-knowledge proofs of Groth, Ostrovsky and Sahai.

Keywords: Public key encryption, commitment, selective opening security, homomorphic encryption, chosen-ciphertext security, lossy encryption

A preliminary version of this work appeared in the proceedings of ASIACRYPT 2011

[*] E-mail: bretth@math.ucla.edu. Supported in part by NSF VIGRE Fellowship and NSF grants 0716835, 0716389, 0830803 and 0916574.

[†] E-mail: benoit.libert@uclouvain.be

[‡] E-mail: rafail@cs.ucla.edu. Supported in part by IBM Faculty Award, Xerox Innovation Group Award, OKAWA Research Award, NSF grants 0430254, 0716835, 0716389, 0830803, 0916574, BSF grant, and U.C. MICRO grant.

[§] E-mail: vergnaud@di.ens.fr

Contents

1 Introduction

In Byzantine agreement, and more generally in secure multiparty computation, it is often assumed that all parties are connected to each other via private channels. In practice, these private channels are implemented using a public-key cryptosystem. An adaptive adversary in a MPC setting, however, has very different powers than an adversary in an IND-CPA or IND-CCA game. In particular, an adaptive MPC adversary may view all the encryptions sent in a given round, and then choose to corrupt a certain fraction of the players, thus revealing the decryptions of those players' messages *and the randomness used to encrypt them.* A natural question is whether the messages sent from the uncorrupted players remain secure. If the messages (and randomness) of all the players are chosen independently, then security in this setting follows immediately from the IND-CPA security of the underlying encryption. If, however, the messages are not chosen independently, the security does not immediately follow from the IND-CPA (or even IND-CCA) security of the underlying scheme. In fact, although this problem was first investigated over twenty years ago, it remains an open question whether IND-CPA (or IND-CCA) security implies this *selective opening* security.

A similar question may be asked regarded in terms of commitments as well. Suppose an adversary is allowed to see commitments to a number of related messages, the adversary may then choose a subset of the commitments for the challenger to de-commit. Does this reveal any information about the unopened commitments? This question has applications to concurrent zero-knowledge proofs.

1.1 Previous Work

There have been many attempts to design encryption protocols that can be used to implement secure multiparty computation against an adaptive adversary. The first protocols by Beaver and Haber [BH92] required interaction between the sender and receiver, required erasure and were fairly inefficient. The first non-interactive protocol was given by Canetti, Feige, Goldreich and Naor in [CFGN96]. In [CFGN96] the authors defined a new primitive called Non-Committing Encryption, and gave an example of such a scheme based on the RSA assumption. In [Bea97], Beaver extended the work of [CFGN96], and created adaptively secure key exchange under the Diffie-Hellman assumption. In subsequent work, Damgård and Nielsen improved the efficiency of the schemes of Canetti *et al.* and Beaver, they were also able to obtain Non-Committing Encryption based on one-way trapdoor functions with invertible sampling. In [CHK05], Canetti, Halevi and Katz presented a Non-Committing encryption protocols with evolving keys.

In [CDNO97], Canetti, Dwork, Naor and Ostrovsky extended the notion of Non-Committing Encryption to a new protocol which they called Deniable Encryption. In Non-Committing Encryption schemes there is a simulator, which can generate non-committing ciphertexts, and later open them to any desired message, while in Deniable Encryption, valid encryptions generated by the sender and receiver can later be opened to any desired message. The power of this primitive made it relatively difficult to realize, and Canetti *et al.* were only able to obtain modest examples of Deniable Encryption and left it as an open question whether fully deniable schemes could be created.

The notions of security against an adaptive adversary can also be applied to commitments. In fact, according to [DNRS03] the necessity of adaptively-secure commitments was realized by 1985. Despite its utility, until recently, relatively few papers directly addressed the question of commitments secure against a selective opening adversary (SOA). The work of Dwork, Naor, Reingold and Stockmeyer [DNRS03] was the first to explicitly address the problem. In [DNRS03], Dwork *et al.* showed that non-interactive SOA-secure commitments can be used to create a 3-round zero-knowledge proof systems for NP with negligible soundness error, and they gave constructions of a weak form of SOA-secure commitments, but leave open the question of whether general SOA-secure commitments exist.

The question of SOA-secure commitments was put on firm foundations by Hofheinz [Hof11b] and Bellare, Hofheinz and Yilek in [BHY09]. In [BHY09], Bellare *et al.* distinguished between simulation-based and indistinguishability-based definitions of security, and gave a number of constructions and black-box separations. In particular, Hofheinz showed that, in the simulation-based setting, non-interactive SOA-secure commitments cannot be realized in a black-box manner from standard cryptographic assumptions, but if interaction is allowed, they can be created from one-way permutations in a non-black-box manner. In the indistinguishability-based setting, they showed that any statistically-hiding scheme achieves this level of security, but that there is a black-box separation between perfectly-binding SOA-secure commitments and most standard cryptographic assumptions. Our results in the selective opening setting build on the breakthrough results of [BHY09].

The concurrent, independent work of Fehr, Hofheinz and Kiltz and Wee [FHKW10] also examines the case of CCA2 cryptosystems that are selective opening secure. In their work, they show how to adapt the universal hash proof systems of [CS02], to provide CCA2 security in the selective opening setting. Their constructions are general, and offer the first SEM-SO-CCA secure cryptosystem whose parameters are completely independent of n, the number of messages. Their work also considers selective opening security against chosen-plaintext attacks, and using techniques from Non-Committing Encryption [CFGN96] they construct SEM-SO-CPA secure systems from enhanced one-way trapdoor permutations.

The results of Bellare, Waters and Yilek [BWY11] show how to construct Identity-Based Encryption (IBE) schemes secure under selective-opening attacks based on the Decision Linear Assumption. Our work is orthogonal to theirs. Their work constructs IBE schemes secure under selective-opening attacks, while our work starts with a tag-based encryption scheme, and uses it to construct encryption schemes that are secure against a selective-opening chosen-ciphertext attack, but are not identity-based.

1.2 Our Contributions

In this paper, we primarily consider encryptions secure against a selective opening adversary. In particular, we formalize the notion of re-randomizable Public-Key Encryption and we show that re-randomizable encryption implies Lossy Encryption, as defined in [PVW08] and expanded in [BHY09]. Combining this with the recent result of Bellare, Hofheinz and Yilek [BHY09] showing that Lossy Encryption is IND-SO-ENC secure, we have an efficient construction of IND-SO-ENC secure encryption from any re-randomizable encryption (which generalizes and extends previous results). Furthermore, these constructions retain the efficiency of the underlying re-randomizable encryption protocol.

Applying our results to the Paillier cryptosystem [Pai99], we obtain an encryption scheme which attains a strong, simulation-based form of semantic security under selective openings (SEM-SO-ENC security). This is the first construction of this type from the Decisional Composite Residuosity (DCR) assumption. As far as bandwidth goes, it is also the most efficient SEM-SO-ENC secure encryption scheme to date. We note that the possible use of Paillier as a lossy encryption scheme was implicitly mentioned in [YY05]. To the best of our knowledge, its SEM-SO-ENC security was not reported earlier.

We go on to show that Lossy Encryption is also implied by (honest-receiver) statistically-hiding $\binom{2}{1}$-Oblivious Transfer and by hash proof systems [CS02]. Combining this with the results of

[PVW08], we recognize that Lossy Encryption is essentially just a different way to view the well known statistically-hiding $\binom{2}{1}$-OT primitive. Applying the reductions in [BHY09] to this result, yields constructions of SOA secure encryption from both PIR and homomorphic encryption.

These results show that the Lossy and Selective Opening Secure Encryption primitives (at least according to the latter's indistinguishability-based security definition), which have not been extensively studied until recently, are actually implied by several well-known primitives: *i.e.*, re-randomizable encryption, PIR, homomorphic encryption, hash proof systems and statistically-hiding $\binom{2}{1}$-OT. Prior to this work, the only known general[1] constructions of lossy encryption were from lossy trapdoor functions. Our results thus show that they can be obtained from many seemingly weaker primitives (see figure 1).

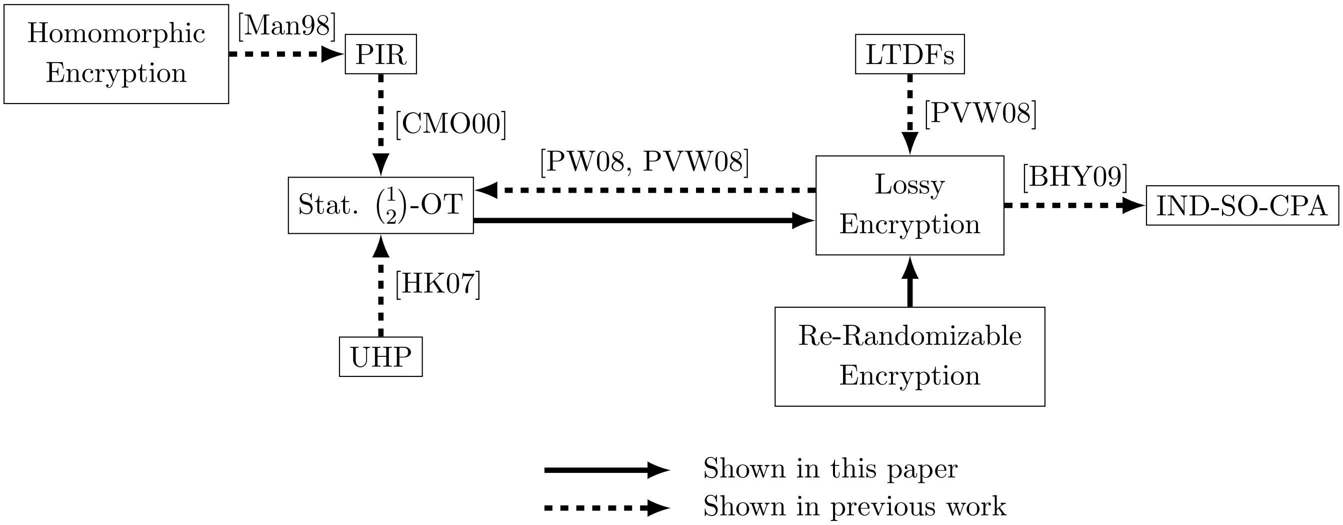

Figure 1: Constructing Lossy Encryption

Selective Opening Security Against Chosen-Ciphertext Attacks: Continuing the study of selective-opening security, we present definitions chosen-ciphertext security (CCA2) in the selective opening setting (in both the indistinguishability and simulation-based models) and describe encryption schemes that provably satisfy these enhanced forms of security. Despite recent progress, relatively few methods are known for constructing IND-CCA2 cryptosystems in the standard model. The problem is even more complex with selective openings, where some known approaches for CCA2 security do not seem to apply. We note how the Naor-Yung paradigm, even when applied with statistical zero knowledge proofs fails to prove CCA2 security in the selective opening setting. Essentially, this is because the selective opening adversary learns the randomness used in the signature scheme, which allows him to forge signatures, and thus create ciphertexts that cannot be handled by the simulated decryption oracle.

The results of Fehr, Hofheinz, Kiltz and Wee [FHKW10] show how to modify universal hash proof systems [CS02] to achieve security under selective openings.

We take a different approach and follow (a variant of) the Canetti-Halevi-Katz paradigm [CHK04]. This too encounters many obstacles in the selective opening setting. Nevertheless, under standard assumptions (such as DDH or the Composite Residuosity assumption), we construct

[1] *i.e.*, not based on specific number-theoretic assumptions

schemes featuring compact ciphertexts while resisting adaptive (*i.e.*, CCA2) chosen-ciphertext attacks according to our indistinguishability-based definition. When comparing our schemes to those of [FHKW10], we note that our public key size depends on n, the number of senders that can be possibly corrupted, while the systems of [FHKW10] are independent of n. On the other hand, to encrypt m-bit messages with security parameter λ, our ciphertexts are of length $\mathcal{O}(\lambda+m)$, while theirs are of length $\mathcal{O}(\lambda m)$. Our public-keys are longer than in [FHKW10] because our construction relies on All-But-N Lossy Trapdoor Functions (defined below), which have long description. The recent complementary work of Hofheinz [Hof11a] shows how to create All-But-Many Trapdoor Functions with short keys. Using his results in our construction eliminates the dependence of the public-key size on n. Regarding security definitions, our constructions satisfy an indistinguishability-based definition (IND-SO-CCA), whereas theirs fit a simulation-based definition (SEM-SO-CCA) which avoids the restriction on the efficient conditional re-sampleability of the message distribution.

The scheme of [FHKW10] is very different from ours and we found it interesting to investigate the extent to which well-known paradigms like [CHK04] can be applied in the present context. Moreover, by adapting the Naor-Yung paradigm [NY90], under more general assumptions, we give a CCA1 construction that also satisfies a strong simulation-based notion of adaptive selective opening security.

One advantage of our IND-SO-CCA scheme is the ability to natively encrypt multi-bit messages. It is natural to consider whether our approach applies to the scheme of Bellare, Waters and Yilek [BWY11] to achieve multi-bit IND-SO-CCA encryption. The scheme of [BWY11], like [FHKW10], encrypts multi-bit messages in a bitwise manner. Applying a Canetti-Halevi-Katz-like transformation to the construction of [BWY11] does not immediately yield IND-SO-CCA encryption schemes for multi-bit messages: the reason is that it is not clear how to prevent the adversary from reordering the bit encryptions without employing a one-time signature scheme.

2 Background

2.1 Notation

If $f : X \to Y$ is a function, for any $Z \subset X$, we let $f(Z) = \{f(x) : x \in Z\}$. If A is a PPT machine, then we use $a \overset{\$}{\leftarrow} A$ to denote running the machine A and obtaining an output, where a is distributed according to the internal randomness of A. For a PPT machine A, we use $\mathsf{coins}(A)$ to denote the distribution of the internal randomness of A. So the distributions $\{a \overset{\$}{\leftarrow} A\}$ and $\{r \overset{\$}{\leftarrow} \mathsf{coins}(A) : a = A(r)\}$ are identical. If R is a set, we use $r \overset{\$}{\leftarrow} R$ to denote sampling uniformly from R.

If X and Y are families of distributions indexed by a security parameter λ, we use $X \approx_s Y$ to mean the distributions X and Y are statistically close, *i.e.*, for all polynomials p and sufficiently large λ, we have $\sum_x |\Pr[X = x] - \Pr[Y = x]| < \frac{1}{p(\lambda)}$.

We use $X \approx_c Y$ to mean X and Y are computationally close, *i.e.*, for all PPT adversaries A, for all polynomials p, then for all sufficiently large λ, we have $|\Pr[A^X = 1] - \Pr[A^Y = 1]| < 1/p(\lambda)$.

2.2 Selective Opening Secure Encryption

We recall an indistinguishability-based definition of encryption secure against a selective opening adversary that was originally formalized in [BHY09]. We define two games, a real and an ideal game which should be indistinguishable to any efficient adversary. The key point to notice is that the adversary receives *both* the messages and the randomness for his selection. This mirrors the fact that an adaptive MPC adversary learns the entire history of corrupted players (*i.e.*, there are no secure erasures). If the adversary receives only the messages this would reduce to standard CPA security.

As in [BHY09], $\mathcal{M}$ denotes an n-message sampler outputting a n-vector $\mathbf{m} = (m_1, \ldots, m_n)$ of messages whereas $\mathcal{M}_{|I,\mathbf{m}[I]}$ denotes an algorithm that conditionally resamples another random n-vector $\mathbf{m}' = (m'_1, \ldots, m'_n)$ such that $m'_i = m_i$ for each $i \in I \subset \{1, \ldots, n\}$. If such a resampling can be done efficiently for all $I, \mathbf{m}$, then $\mathcal{M}$ is said to support efficient conditional resampling.

Definition 1. (Indistinguishability under selective openings). A public key cryptosystem (G, E, D) is indistinguishable under selective openings (IND-SO-ENC secure) if, for any message sampler $\mathcal{M}$ supporting efficient conditional resampling and any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have

$$\left|\Pr\left[\mathcal{A}^{\textsf{ind-so-real}} = 1\right] - \Pr\left[\mathcal{A}^{\textsf{ind-so-ideal}} = 1\right]\right| < \nu$$

for some negligible function ν, and where the games ind-so-real and ind-so-ideal are defined as follows.

IND-SO-ENC (Real)

```
$\mathbf{m} = (m_1, \ldots, m_n) \overset{\$}{\leftarrow} \mathcal{M}$
$r_1, \ldots, r_n \overset{\$}{\leftarrow} \mathsf{coins}(E)$
$(I, st) \overset{\$}{\leftarrow} \mathcal{A}_1\big(pk, E(m_1, r_i), \ldots, E(m_n, r_n)\big)$
$b \overset{\$}{\leftarrow} \mathcal{A}_2\big(st, (m_i, r_i)_{i \in I}, \mathbf{m}\big)$
```

IND-SO-ENC (Ideal)

```
$\mathbf{m} = (m_1, \ldots, m_n) \overset{\$}{\leftarrow} \mathcal{M}$
$r_1, \ldots, r_n \overset{\$}{\leftarrow} \mathsf{coins}(E)$
$(I, st) \overset{\$}{\leftarrow} \mathcal{A}_1\big(pk, E(m_1, r_i), \ldots, E(m_n, r_n)\big)$
$\mathbf{m}' = (m'_1, \ldots, m'_n) \overset{\$}{\leftarrow} \mathcal{M}_{|I,\mathbf{m}[I]}$
$b \overset{\$}{\leftarrow} \mathcal{A}_2\big(st, (m_i, r_i)_{i \in I}, \mathbf{m}'\big)$
```

Figure 2: IND-SO-ENC security

In the real game, the challenger samples $\mathbf{m} = (m_1, \ldots, m_n) \overset{\$}{\leftarrow} \mathcal{M}$ from the joint message distribution. Then, it generates randomness $r_1, \ldots, r_n \overset{\$}{\leftarrow} \mathsf{coins}(E)$ and sends $(E(m_1, r_1), \ldots, E(m_n, r_n)$ to $\mathcal{A}$. The adversary $\mathcal{A}$ responds with a subset $I \subset \{1, \ldots, n\}$ of size $\#I = n/2$. The challenger reveals r_i for each $i \in I$ as well as the *entire* vector $\mathbf{m} = (m_1, \ldots, m_n)$ to $\mathcal{A}$. Finally, the latter outputs a bit $b \in \{0, 1\}$.

In the ideal game, the challenger also samples $\mathbf{m} = (m_1, \ldots, m_n) \overset{\$}{\leftarrow} \mathcal{M}$ from the joint distribution. Then, it generates random coins $r_1, \ldots, r_n \overset{\$}{\leftarrow} \mathsf{coins}(E)$ and sends $(E(m_1, r_1), \ldots, E(m_n, r_n))$ to the adversary $\mathcal{A}$. The latter chooses a subset $I \subset \{1, \ldots, n\}$ with $\#I = n/2$ and the challenger reveals r_i for $i \in I$. The only difference w.r.t. the real game is that, instead of revealing $\mathbf{m}$, the challenger samples a new vector $\mathbf{m}' \overset{\$}{\leftarrow} \mathcal{M}_{|I,\mathbf{m}[I]}$ and sends $\mathbf{m}'$ to $\mathcal{A}$. Eventually, the adversary outputs a bit $b \in \{0, 1\}$.

We stress that the challenger reveals both the plaintexts m_i and the randomness r_i for indices $i \in I$. If only the messages m_i were revealed, this security would follow immediately from IND-CPA security.

2.3 Lossy Encryption

In [PVW08], Peikert, Vaikuntanathan and Waters defined Dual-Mode Encryption, a type of cryptosystem with two types public-keys, injective keys on which the cryptosystem behaves normally and "lossy" or "messy" keys on which the system loses information about the plaintext. In particular they require that the encryptions of any two plaintexts under a lossy key yield distributions that are statistically close, yet injective and lossy keys remain computationally indistinguishable.

In [BHY09] Bellare, Hofheinz and Yilek define *Lossy Encryption*, expanding on the definitions of Dual-Mode Encryption in [PVW08], and Meaningful/Meaningless Encryption in [KN08]. At a high level, a 'lossy' (or 'messy' in the terminology of [PVW08]) cryptosystem is one which has two types of public keys which specify two different modes of operation. In the normal mode, encryption is injective, while in the lossy (or 'messy') mode, the ciphertexts generated by the encryption algorithm are independent of the plaintext. We also require that no efficient adversary can distinguish normal keys from lossy keys. In [BHY09], they also require a property called *openability*, which basically allows a possibly inefficient algorithm to open a ciphertext generated under a lossy key to *any* plaintext.

Definition 2. A *lossy public-key encryption scheme* is a tuple (G, E, D) of efficient algorithms such that

- $G(1^\lambda, \mathsf{inj})$ outputs keys (pk, sk), keys generated by $G(1^\lambda, \mathsf{inj})$ are called *injective keys.*
- $G(1^\lambda, \mathsf{lossy})$ outputs keys $(pk_{\mathsf{lossy}}, sk_{\mathsf{lossy}})$, keys generated by $G(1^\lambda, \mathsf{lossy})$ are called *lossy keys.*

Additionally, the algorithms must satisfy the following properties:

1. *Correctness on injective keys.* For all plaintexts $x \in X$,
$$\Pr\left[(pk, sk) \xleftarrow{\$} G(1^\lambda, \mathsf{inj});\ r \xleftarrow{\$} \mathsf{coins}(E) : D(sk, E(pk, x, r)) = x\right] = 1.$$
2. *Indistinguishability of keys.* In lossy mode, public keys are computationally indistinguishable from those in the injective mode. Specifically, if $\mathrm{proj} : (pk, sk) \mapsto pk$ is the projection map, then
$$\{\mathrm{proj}(G(1^\lambda), \mathsf{inj})\} \approx_c \{\mathrm{proj}(G(1^\lambda, \mathsf{lossy}))\}$$
3. *Lossiness of lossy keys.* If $(pk_{\mathsf{lossy}}, sk_{\mathsf{lossy}}) \xleftarrow{\$} G(1^\lambda, \mathsf{lossy})$, then for all $x_0, x_1 \in X$, the statistical distance between the distributions $E(pk_{\mathsf{lossy}}, x_0, R)$ and $E(pk_{\mathsf{lossy}}, x_1, R)$ is negligible in λ.
4. *Openability.* If $(pk_{\mathsf{lossy}}, sk_{\mathsf{lossy}}) \xleftarrow{\$} G(1^\lambda, \mathsf{lossy})$, and $r \xleftarrow{\$} \mathsf{coins}(E)$, then for all $x_0, x_1 \in X$ with overwhelming probability, there exists $r' \in \mathsf{coins}(E)$ such that $E(pk_{\mathsf{lossy}}, x_0, r) = E(pk_{\mathsf{lossy}}, x_1, r')$. In other words, there is an (unbounded) algorithm opener that can open a lossy ciphertext to *any* arbitrary plaintext with all but negligible probability.

Although openability is implied by property (3), it is convenient to state it explicitly in terms of an algorithm. In [BHY09], it was shown that, if the algorithm opener is efficient, then the encryption scheme is actually SEM-SO-ENC secure (instead of only IND-SO-ENC).

We do not explicitly require schemes to be IND-CPA secure since semantic security follows from the indistinguishability of keys and lossiness of the lossy keys. Indeed, for any $x_0, x_1 \in X$,

$$\begin{aligned}E(\mathrm{proj}(G(1^\lambda, \mathsf{inj})), x_0, R) &\approx_c E(\mathrm{proj}(G(1^\lambda, \mathsf{lossy})), x_0, R)) \\ &\approx_s E(\mathrm{proj}(G(1^\lambda, \mathsf{lossy})), x_1, R) \approx_c E(\mathrm{proj}(G(1^\lambda, \mathsf{inj})), x_1, R).\end{aligned}$$

In [BHY09], it was shown that Lossy Encryption can notably be constructed in a straightforward manner from lossy trapdoor functions. More precisely, they observed that the IND-CPA-secure system given in [PW08] is a Lossy Encryption scheme. Next, they proved the following fact.

Theorem 1. [BHY09] Any Lossy Encryption scheme where the plaintext space admits a n-message sampler $\mathcal{M}$ that supports efficient resampling is IND-SO-ENC secure.

3 Constructing Lossy Encryption Schemes

3.1 Re-Randomizable Encryption Implies Lossy Encryption

In many cryptosystems, given a ciphertext c and a public-key, it is possible to re-randomize c to a new ciphertext c' such that c and c' encrypt the same plaintext but are statistically independent. We call a public key cryptosystem given by algorithms (G, E, D) *statistically re-randomizable*[2] if

- (G, E, D) is semantically-secure in the standard sense (IND-CPA).
- There is an efficient function ReRand such that if r' is chosen uniformly from $\mathsf{coins}(\mathsf{ReRand})$, and r_0 are chosen uniformly from $\mathsf{coins}(E)$, then the distributions

  $$\{r_0 \overset{\$}{\leftarrow} \mathsf{coins}(E) : E(pk, m, r_0)\} \approx_s \{r' \overset{\$}{\leftarrow} \mathsf{coins}(\mathsf{ReRand}) : \mathsf{ReRand}(E(pk, m, r_1), r')\}$$

  for all public keys pk and messages m, and randomness r_1.

There are many examples of re-randomizable encryption. For example, if (G, E, D) is *homomorphic* (*i.e.*, for any two pairs (m_0, r_0) and (m_1, r_1), we have $E(pk, m_0, r_0) \cdot E(pk, m_1, r_1) = E(pk, m_0 + m_1, r^*)$ for some $r^* \in \mathsf{coins}(E)$), it may be possible to take $\mathsf{ReRand}(pk, c, r') = c \cdot E(pk, 0, r')$. For all known homomorphic cryptosystems (such as Elgamal, Paillier, Damgård-Jurik, Goldwasser-Micali), we obtain statistically re-randomizable encryption with this definition of ReRand.

We note that, since re-randomization does not require any kind of group structure on the plaintext space or any method for combining ciphertexts, re-randomizable encryption appears to be a weaker primitive than homomorphic encryption. Although it is not implied by homomorphic encryption *per se*, all known homomorphic cryptosystems are re-randomizable. A more thorough discussion of the relationship between these primitives is given in Appendix B.

Our first result gives a simple and efficient method for creating lossy encryption from re-randomizable encryption. Let (G, E, D) be a statistically re-randomizable public-key cryptosystem, and we create Lossy Encryption $(\bar{G}_{\mathsf{inj}}, \bar{G}_{\mathsf{lossy}}, \bar{E}, \bar{D})$ as follows:

- **Key Generation:**
  $\bar{G}(1^\lambda, \mathsf{inj})$ generates a pair $(pk, sk) \leftarrow G(1^\lambda)$. Then $G(1^\lambda, \mathsf{inj})$ picks $r_0, r_1 \overset{\$}{\leftarrow} \mathsf{coins}(E)$, and generates $e_0 = E(pk, 0, r_0)$, $e_1 = E(pk, 1, r_1)$. $\bar{G}(1^\lambda, \mathsf{inj})$ returns $(\bar{pk}, \bar{sk}) = ((pk, e_0, e_1), sk)$.

  $\bar{G}(1^\lambda, \mathsf{lossy})$ runs $G(1^\lambda)$, generating a pair (pk, sk). Then, it picks $r_0, r_1 \overset{\$}{\leftarrow} \mathsf{coins}(E)$ and generates $e_0 = E(pk, 0, r_0)$, $e_1 = E(pk, 0, r_1)$. $\bar{G}(1^\lambda, \mathsf{lossy})$ returns $(\bar{pk}, \bar{sk}) = ((pk, e_0, e_1), sk)$.
- **Encryption:** $\bar{E}(\bar{pk}, b, r') = \mathsf{ReRand}(pk, e_b, r')$ for $b \in \{0, 1\}$.
- **Decryption** $\bar{D}(\bar{sk}, c)$, simply outputs $D(sk, c)$.

[2] We note that this definition of re-randomizable encryption requires statistical re-randomization. It is possible to define re-randomizable encryption which satisfies perfect re-randomization (stronger) or computational re-randomization (weaker). Such definitions already exist in the literature (see for example [PR07, Gro04, JJS04, CKN03]). Our constructions require statistical re-randomization, and do not go through under a computational re-randomization assumption.

We first notice that, under an injective key, the encryption mapping is clearly injective and the decryption algorithm D performs the inverse operation. In lossy mode, it will be statistically lossy by the properties of the ReRand function. The proof that this is a Lossy Encryption system is straightforward and we check the details here.

1. *Correctness on injective keys.* This follows immediately from the correctness of E.
2. *Indistinguishability of keys.* This follows immediately from the IND-CPA security of (G, E, D).
3. *Lossiness of lossy keys.* Notice that under a lossy public-key $\bar{pk}$, e_0 and e_1 are both encryptions of zero, so that $\bar{E}(\bar{pk}, b, r)$ will also be an encryption of zero for $b \in \{0, 1\}$. By the properties of ReRand, the distributions $\{\bar{E}(\bar{pk}, 0, r)\}$ and $\{\bar{E}(\bar{pk}, 1, r)\}$ will be statistically close, which is exactly what is required for a key to be "lossy".
4. *Openability.* Under a lossy public-key, we have $\bar{E}(\bar{pk}, b, r') = \mathsf{ReRand}(E(pk, 0, r_b), r')$. Since r' is chosen uniformly from $\mathsf{coins}(\mathsf{ReRand})$, the properties of ReRand guarantee that the distributions $\mathsf{ReRand}(E(pk, 0, r_b), r')$ and $\mathsf{ReRand}(E(pk, 0, r_{1-b}), r'')$ are statistically close. The existence of r'' such that $\mathsf{ReRand}(E(pk, 0, r_b), r') = \mathsf{ReRand}(E(pk, 0, r_{1-b}), r'')$ then follows from lemma 1.

Lemma 1. If R is a random variable, and $f : R \to X$, $g : R \to Y$ and

$$\sum_{z \in X \cup Y} \Pr\left[r \leftarrow R : f(r) = z\right] - \Pr\left[r \leftarrow R : g(r) = z\right] = \nu,$$

then $\Pr\left[r \leftarrow R : \forall r' \in R, f(r) \neq g(r')\right] < \nu$.

Proof. It suffices to notice that

$$\begin{aligned}\nu &= \sum_{z \in X \cup Y} \Pr\left[r \leftarrow R : f(r) = z\right] - \Pr\left[r \leftarrow R : g(r) = z\right] \\ &\geq \sum_{z \in X \setminus Y} \Pr\left[r \leftarrow R : f(r) = z\right] - \Pr\left[r \leftarrow R : g(r) = z\right] \\ &= \Pr\left[r \leftarrow R : \forall r' \in R, f(r) \neq g(r')\right].\end{aligned}$$

□

Although this scheme only allows encrypting single bits, it can be easily modified to encrypt longer messages if the underlying cryptosystem is homomorphic and if the set of encryptions of zero can be almost uniformly sampled (the details are available in Appendix B).

The above construction is easily seen to give a perfectly-binding SOA secure commitment scheme (with trusted setup). If our goal is only to construct SOA secure commitments, we do not need re-randomizable encryption, and a weaker primitive suffices. In Appendix A, we define *re-randomizable one-way functions* and show that these imply SOA secure commitments. While these constructions both require a trusted setup, in a sense, this is inevitable since it was shown in [Hof11b, BHY09] that perfectly-binding SOA secure commitments without trusted setup cannot be created in a black-box manner from any primitive with a game-based definition of security.

We also note that specific homomorphic cryptosystems such as Paillier [Pai99] or Damgård-Jurik [DJ01] provide more efficient constructions where multi-bit messages can be encrypted. In addition, as shown in Appendix C.1, the factorization of the modulus N provides a means for efficiently opening a lossy ciphertext to any plaintext. Thus this scheme is actually SEM-SO-ENC secure when instantiated with these cryptosystems. This provides the most efficient known examples of SEM-SO-ENC secure cryptosystems. See Appendix C.1 for further discussion.

3.2 Statistically-Hiding $\binom{2}{1}$-OT Implies Lossy Encryption

We briefly recall the definition of honest-receiver two-round statistically-hiding $\binom{2}{1}$-OT. Oblivious transfer is a protocol between a sender Sen and a receiver $\mathsf{Rec} = (\mathsf{Rec}_q, \mathsf{Rec}_r)$. The sender Sen has two strings s_0, s_1, and the receiver has a bit b. The receiver Rec_q generates a query q along with some state information sk and sends q to the sender. The sender evaluates $\mathsf{q}(s_0, s_1)$ and sends the result $\mathsf{rsp} = \mathsf{Sen}(\mathsf{q}, s_0, s_1)$ to the receiver Rec_r who uses sk to obtain s_b.

- **Correctness:** For all $s_0, s_1 \in \{0,1\}^k$, for all $b \in \{0,1\}$, there is a negligible function ν such that

$$\Pr[(\mathsf{q}, sk) \xleftarrow{\$} \mathsf{Rec}_q(1^\lambda, b);\ \mathsf{rsp} \xleftarrow{\$} \mathsf{Sen}(\mathsf{q}, s_0, s_1) : \mathsf{Rec}_r(sk, \mathsf{rsp}) = s_b] \geq 1 - \nu(\lambda).$$

- **Receiver Privacy:** b remains computationally hidden from Sen's view. Specifically, we must have

$$\{(\mathsf{q}, sk) \xleftarrow{\$} \mathsf{Rec}_q(1^\lambda, 0) : \mathsf{q}\} \approx_c \{(\mathsf{q}, sk) \xleftarrow{\$} \mathsf{Rec}_q(1^\lambda, 1) : \mathsf{q}\},$$

  where the distributions are taken over the internal randomness of Rec_q.

- **Sender Privacy:** for any $b \in \{0,1\}$, for any strings s_0, s_1, s_0', s_1' such that $s_b = s_b'$ and any honest receiver's query $\mathsf{q} = \mathsf{Rec}_q(1^\lambda, b)$, it must hold that

$$\{(\mathsf{q}, sk) \xleftarrow{\$} \mathsf{Rec}_q(1^\lambda, b); \mathsf{rsp} \xleftarrow{\$} \mathsf{Sen}(\mathsf{q}, s_0, s_1) : \mathsf{rsp}\} \approx_s \{(\mathsf{q}, sk) \xleftarrow{\$} \mathsf{Rec}_q(1^\lambda, b); \mathsf{rsp} \xleftarrow{\$} \mathsf{Sen}(\mathsf{q}, s_0', s_1') : \mathsf{rsp}\}$$

  where the distributions are taken over the internal randomness of Rec_q and Sen.

Let $(\mathsf{Sen}, \mathsf{Rec})$ be a two-round honest-receiver statistically-hiding $\binom{2}{1}$-OT. We construct a lossy encryption as follows:

- **Key Generation:** Define $G(1^\lambda, \mathsf{inj}) = \mathsf{Rec}_q(1^\lambda, 0)$. Set $pk = \mathsf{q}$, and $sk = sk$.

  Define $G(1^\lambda, \mathsf{lossy}) = \mathsf{Rec}_q(1^\lambda, 1)$. Set $pk = \mathsf{q}$, and $sk = \perp$.

- **Encryption:** Define $E(pk, m, (r, r^*)) = \mathsf{Sen}(\mathsf{q}, m, r; r^*)$, where r^* is the randomness used in $\mathsf{Sen}(\mathsf{q}, m, r)$ and $r \xleftarrow{\$} \{0,1\}^{|m|}$ is a random string.

- **Decryption:** to decrypt $c = \mathsf{rsp}$ in injective mode, we define $D(sk, \mathsf{rsp}) = \mathsf{Rec}_r(sk, \mathsf{rsp})$.

Lemma 2. The scheme (G, E, D) forms a lossy encryption scheme.

Proof. We need to show three things:

- **Correctness on injective keys:** This follows immediately from the correctness of OT.

- **Indistinguishability of keys:** This follows immediately from the receiver privacy of OT.

- **Lossiness of lossy keys:** This will follow from the statistical sender privacy OT. More precisely, if the cryptosystem is in lossy mode, the sender privacy of OT says that for all m_0, m_1

$$\{\mathsf{Sen}(\mathsf{q}, m_0, r)\} \approx_s \{\mathsf{Sen}(\mathsf{q}, m_1, r)\},$$

where the distribution is taken over the internal randomness of Sen. Now, if we view the randomness of Sen as an explicit input to Sen (as we do in encryption), then we have that for all m_0, m_1 and r,

$$\Delta(\mathsf{Sen}(\mathsf{q}, m_0, r; \cdot), \mathsf{Sen}(\mathsf{q}, m_1, r); \cdot) < \nu,$$

where the distributions are taken over the internal randomness of Sen. Applying lemma 3, we find

$$\Delta(\mathsf{Sen}(\mathsf{q}, m_0, \cdot; \cdot), \mathsf{Sen}(\mathsf{q}, m_1, \cdot; \cdot)) \leq \nu,$$

where the distributions range over the uniform choice of r and the internal randomness of Sen. This is exactly what is required to guarantee the lossiness of lossy keys.

□

Lemma 3. Let X, Y, Z be random variables such that $\Delta(X, Y|Z = z) < \epsilon$ for all z. Then, $\Delta(X, Y) < \epsilon$.

Proof.

$$\begin{aligned}
\Delta(X, Y) &= \sum_a |\Pr(X = a) - \Pr(Y = a)| \\
&= \sum_a \sum_z |\Pr(X = a, Z = z) - \Pr(Y = a, Z = z)| \\
&= \sum_a \sum_z |\Pr(X = a|Z = z) - \Pr(Y = a|Z = z)| \Pr(z = z) \\
&= \sum_z \Pr(Z = z) \sum_a |\Pr(X = a|Z = z) - \Pr(Y = a|Z = z)| \\
&= \sum_z \Pr(Z = z)\Delta(X, Y|Z = z) < \epsilon \sum_z \Pr(Z = z) = \epsilon.
\end{aligned}$$

□

Applying the results of [CMO00] which show that single-server Private Information Retrieval (PIR) implies statistically-hiding OT, we find the following corollary.

Corollary 1. One round (two message) Single-Server PIR implies Lossy-Encryption.

Since homomorphic encryption implies PIR [KO97, Man98, IKO05], the following result follows.

Corollary 2. Homomorphic encryption implies Lossy-Encryption.

It was shown in [Kal05, HK07] that, in the half simulation model, statistically hiding $\binom{2}{1}$-OT can be based on smooth hash proof systems that fit a slight modification of the original definition [CS02] with suitable verifiability properties. In the honest-but-curious receiver setting (which suffices here), it was already noted in [HK07][Section 1.3] that ordinary hash proof systems, as defined in [CS02], are sufficient to realize $\binom{2}{1}$-OT. In Appendix D, we describe a simplification of the construction of lossy encryption from hash proof systems and obtain the next result.

Corollary 3. Smooth projective hash functions imply Lossy Encryption.

Interestingly, the DDH-based lossy encryption scheme of [KN08, PVW08, BHY09] can be seen as a particular instance of that construction using the Projective Hashing of [CS98]. It can also be interpreted as being derived (after simplification) from the Naor-Pinkas OT protocol [NP01] via our construction.

The relationship with hash proof systems also suggests other implementations of lossy encryption based on Composite or Quadratic Residuosity (which differ from the scheme in Appendix C.1 and from Goldwasser-Micali, respectively) and the Decision Linear assumption [BBS04].

To summarize this section, by applying Theorem 1, we obtain the following theorem.

Theorem 2. Statistically-hiding 2-round honest-player $\binom{2}{1}$-OT implies IND-SO-ENC secure encryption. Moreover, single-server PIR and homomorphic encryption and smooth projective hash proof systems also imply IND-SO-ENC secure encryption.

4 Chosen-Ciphertext Security

It has long been recognized that if an adversary is given access to a decryption oracle, many cryptosystems may become insecure. The notion of chosen-ciphertext Security [NY90, RS91, DDN91] was created to address this issue, and since then there have been many schemes that achieve this level of security. The attacks of Bleichenbacher on RSA PKCS#1 [Ble98] emphasized the practical importance of security against chosen-ciphertext attacks (CCA).

The need for selective opening security was first recognized in the context of Multi-Party Computation (MPC), where an active MPC adversary can view all ciphertexts sent in a current round and then choose a subset of senders to corrupt. It is natural to imagine an adversary who, in addition to corrupting a subset of senders, can also mount a chosen-ciphertext attack against the receiver. Schemes proposed so far (based on re-randomizable encryption or described in [BHY09]) are obviously insecure in this scenario.

In this section, we extend the notion of chosen-ciphertext security to the selective opening setting. As in the standard selective-opening setting, we can define security either by indistinguishability, or by simulatability. We will give definitions of security as well as constructions for both settings.

Currently known techniques to acquire chosen-ciphertext security are delicate to use here. For instance, handling decryption queries using the Naor-Yung paradigm [NY90] and non-interactive zero-knowledge techniques [Sah99] is not straightforward as, when the adversary makes her corruption query, she should also obtain the random coins that were used to produce NIZK proofs. Hash proof systems (HPS) [CS98, CS02] seem problematic to use as well. They typically involve security reductions where simulators know the private key corresponding to the public key given to the adversary. This seems inherently at odds with the features of lossy encryption, where security relies on the property that lossy public keys (for which private keys may not exist) look like well-formed public keys. As we will see, leveraging other tools such as the Canetti-Halevi-Katz paradigm [CHK04] raises its deal of technical issues.

4.1 Chosen-Ciphertext Security: Indistinguishability

We begin with the indistinguishability-based definition (the simulation-based one is provided in Appendix E). We define two games, a real game (ind-cca2-real) and an ideal game (ind-cca2-ideal). In both games, the challenger runs the key-generation algorithm to generate a key pair $(sk, pk) \leftarrow G(1^\lambda)$ and sends pk to $\mathcal{A}$. The adversary is then allowed to adaptively make the following types of queries.

- **Challenge Query:** let $\mathcal{M}$ be a message sampler. The latter samples $\mathbf{m} = (m_1, \ldots, m_n) \overset{\$}{\leftarrow} \mathcal{M}$ and returns n "target" ciphertexts

$$\mathbf{C} = (\mathbf{C}[1], \ldots, \mathbf{C}[n]) \leftarrow (E(pk, m_1, r_1), \ldots, E(pk, m_n, r_n)).$$

- **Corrupt Query:** $\mathcal{A}$ chooses a subset $I \subset \{1, \ldots, n\}$ of cardinality $\#I = n/2$. The challenger then reveals $\{(m_i, r_i)\}_{i \in I}$ to $\mathcal{A}$.
  - In the real game, the challenger then sends $\{m_j\}_{j \notin I}$ to the adversary.
  - In the ideal game, the challenger re-samples $\mathbf{m}' = (m'_1, \ldots, m'_n) \overset{\$}{\leftarrow} \mathcal{M}_{|I, \mathbf{m}[I]}$ (*i.e.*, in such a way that $m'_j = m_j$ for each $j \in I$) and sends $\{m'_j\}_{j \notin I}$ to $\mathcal{A}$.
- **Decryption Queries:** $\mathcal{A}$ chooses a ciphertext C that has never appeared as a target ciphertext and sends C to the challenger which responds with $D(sk, C)$.

After a polynomial number of queries, exactly one of which is a challenge query and precedes the corrupt query (which is unique as well), the adversary outputs $b \in \{0, 1\}$.

Definition 3. A public key cryptosystem is IND-SO-CCA2 secure if, for any polynomial n and any n-message sampler $\mathcal{M}$ supporting efficient conditional re-sampling, any PPT adversary $\mathcal{A}$ has negligibly different outputs in the real game and in the ideal game: for some negligible function ν, we must have

$$\left| \Pr[\mathcal{A}^{\mathsf{ind\text{-}cca2\text{-}real}} = 1] - \Pr[\mathcal{A}^{\mathsf{ind\text{-}cca2\text{-}ideal}} = 1] \right| < \nu.$$

If the adversary is not allowed to make decryption queries, this reduces to IND-SO-ENC security.

Our construction of IND-SO-CCA2 secure encryption requires some basic tools outlined below.

4.2 Chameleon Hash Functions

A chameleon hash function [KR00] $\mathcal{CMH} = (\mathsf{CMKg}, \mathsf{CMhash}, \mathsf{CMswitch})$ consists of a key generation algorithm CMKg that, given a security parameter λ, outputs a pair $(hk, tk) \overset{\$}{\leftarrow} \mathcal{G}(\lambda)$. The randomized hashing algorithm outputs $y = \mathsf{CMhash}(hk, m, r)$ given the public key hk, a message m and random coins $r \in \mathcal{R}_{hash}$. On input of m, r, m' and the trapdoor key tk, the switching algorithm $r' \leftarrow \mathsf{CMswitch}(tk, m, r, m')$ outputs $r' \in \mathcal{R}_{hash}$ such that $\mathsf{CMhash}(hk, m, r) = \mathsf{CMhash}(hk, m', r')$. Collision-resistance mandates that it be infeasible to find collisions (*i.e.*, pairs $(m', r') \neq (m, r)$ such that $\mathsf{CMhash}(hk, m, r) = \mathsf{CMhash}(hk, m', r')$) without knowing tk. Finally, uniformity guarantees that the distribution of hashes is independent of the message m, in particular, for all hk, and m, m', the distributions $\{r \leftarrow \mathcal{R}_{hash} : \mathsf{CMHash}(hk, m, r)\}$ and $\{r \leftarrow \mathcal{R}_{hash} : \mathsf{CMHash}(hk, m', r)\}$ are identical. It is well-known that chameleon hashing can be based on standard number theoretic assumptions such as factoring or the discrete logarithm.

4.3 A Special Use of the Canetti-Halevi-Katz Paradigm

The Canetti-Halevi-Katz technique [CHK04] is a method to build chosen-ciphertext secure encryption schemes from weakly secure identity-based or tag-based encryption scheme. A tag-based encryption scheme (TBE) [MRY04, Kil06] is a public key cryptosystem where the encryption and decryption algorithms take an additional input, named the *tag*, which is a binary string of appropriate length with no particular structure. A TBE scheme consists of a triple

$\mathcal{TBE}$ = (TBEKg, TBEEnc, TBEDec) of efficient algorithms where, on input of a security parameter λ, TBEKg outputs a private/public key pair (pk, sk); TBEEnc is a randomized algorithm that outputs a ciphertext C on input of a public key pk, a string θ – called *tag* – and a message $m \in \mathsf{MsgSp}(\lambda)$; $\mathsf{TBEDec}(sk, \theta, C)$ is the decryption algorithm that takes as input a secret key sk, a tag θ and a ciphertext C and returns a plaintext m or $\perp$. Associated with $\mathcal{TBE}$ is a plaintext space MsgSp. Correctness requires that for all $\lambda \in \mathbb{N}$, all key pairs $(pk, sk) \leftarrow \mathsf{TBEKg}(1^\lambda)$, all tags θ and any plaintext $m \in \mathsf{MsgSp}(\lambda)$, it holds that $\mathsf{TBEDec}(sk, \theta, \mathsf{TBEEnc}(pk, \theta, M)) = m$.

Selective Opening Security for TBE Schemes. In the selective opening setting, the weak CCA2 security definition of [Kil06] can be extended as follows.

Definition 4. A TBE scheme $\mathcal{TBE}$ = (TBEKg, TBEEnc, TBEDec) is *selective-tag weakly IND-SO-CCA2 secure* (or IND-SO-stag-wCCA2 secure) if, for any polynomial n and any n-message sampler $\mathcal{M}$ supporting efficient conditional re-sampling, any PPT adversary $\mathcal{A}$ produces negligibly different outputs in the real and ideal games, which are defined as follows.

1. The adversary $\mathcal{A}$ chooses n tags $\theta_1^\star, \ldots, \theta_n^\star$ and sends them to the challenger.
2. The challenger generates a key pair $(sk, pk) \leftarrow \mathsf{TKEKg}(1^\lambda)$ and hands pk to $\mathcal{A}$. The latter then adaptively makes the following kinds of queries:
   - **Challenge Query:** let $\mathcal{M}$ be a message sampler for $\mathsf{MsgSp}(\lambda)$. The challenger samples $\mathbf{m} = (m_1, \ldots, m_n) \xleftarrow{\$} \mathcal{M}$ and returns n target ciphertexts
   $$\mathbf{C} = (\mathbf{C}[1], \ldots, \mathbf{C}[n]) \leftarrow (\mathsf{TBEEnc}(pk, \theta_1^\star, m_1, r_1), \ldots, \mathsf{TBEEnc}(pk, \theta_n^\star, m_n, r_n)).$$
   - **Corrupt Query:** $\mathcal{A}$ chooses a subset $I \subset \{1, \ldots, n\}$ of size $\#I = n/2$. The challenger then hands $\{(m_i, r_i)\}_{i \in I}$ to $\mathcal{A}$.
     - In the real game, the challenger then sends $\{m_j\}_{j \notin I}$ to the adversary.
     - In the ideal game, the challenger re-samples $(m_1', \ldots, m_n') \xleftarrow{\$} \mathcal{M}_{|I, \mathbf{m}[I]}$ and reveals $\{m_j'\}_{j \notin I}$.
   - **Decryption Queries:** $\mathcal{A}$ sends a pair (C, θ) such that $\theta \notin \{\theta_1^\star, \ldots, \theta_n^\star\}$. The challenger replies with $\mathsf{TBEDec}(sk, \theta, C) \in \mathsf{MsgSp}(\lambda) \cup \{\perp\}$.

After polynomially-many queries, one of which being a challenge query, $\mathcal{A}$ outputs a bit $b \in \{0, 1\}$. Her advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SO-stag-wCCA2}}(\lambda)$ is defined analogously to definition 3.

At first glance, one may hope to simply obtain IND-SO-CCA2 security by applying the CHK method [CHK04] to any IBE/TBE scheme satisfying some weaker level of selective opening security.

Let us assume a TBE scheme $\mathcal{TBE}$ = (TBEKg, TBEEnc, TBEDec) that is secure in the sense of definition 4 and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a strongly unforgeable one-time signature. The black-box CHK technique turns $\mathcal{TBE}$ into a public key cryptosystem $\mathcal{PKE} = (G, E, D)$ which is obtained by letting $G(1^\lambda)$ output $(sk', (\Sigma, pk'))$ where $(sk', pk') \leftarrow \mathsf{TBEKg}(1^\lambda)$. To encrypt a message m, E generates a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(1^\lambda)$, computes $C_{tbe} = \mathsf{TBEEnc}(pk, \mathsf{VK}, m)$ under the tag VK and sets the $\mathcal{PKE}$ ciphertext as $(\mathsf{VK}, C_{tbe}, \sigma)$, where $\sigma = \mathcal{S}(\mathsf{SK}, C_{tbe})$.

When we try to use this transformation in the selective opening setting, the problem is that, when the adversary makes her corruption query in the reduction, she must also obtain the random coins that were used to generate one-time signature key pairs appearing target ciphertexts. Then, she is able to re-compute the corresponding one-time private keys and make decryption queries for ciphertexts involving the same verification keys as target ciphertexts, which causes the reduction to fail. Although schemes using one-time signatures do not appear to become trivially insecure, the reduction of [CHK04, Kil06] ceases to go through and the same hurdle arises with the Boneh-Katz transformation [BK05].

It was showed in [Zha07] that chameleon hash functions [KR00] can be used to turn certain TBE schemes, termed *separable*, into full-fledged IND-CCA2 cryptosytems and supersede one-time signatures in the CHK transform. A TBE scheme is said *separable* if, on input of pk, m, θ, the encryption algorithm $\mathsf{TBEEnc}(pk, t, m)$ uses randomness $r \in \mathcal{R}_{tbe}$ and returns $C_{tbe} = (f_1(pk, m, r), f_2(pk, r), f_3(pk, \theta, r))$, where functions f_1, f_2 and f_3 are computed independently of each other and are all deterministic (and give the same outputs when queried twice on the same (m, r), r and (θ, r)).

The construction of [Zha07] uses chameleon hashing instead of one-time signatures. Key generation requires to create a TBE key pair (pk', sk') and a chameleon hashing public key hk. The private key of $\mathcal{PKE}$ is the TBE private key sk'. Encryption and decryption procedures are depicted on figure 3.

```
$E(m, pk)$
    Parse $pk$ as $(pk', hk)$
    $r_1 \leftarrow \mathcal{R}_{tbe}$; $r_2 \leftarrow \mathcal{R}_{hash}$
    $u = f_1(pk', m, r_1)$; $v = f_2(pk', r_1)$
    $\theta = \mathsf{CMhash}(hk, u||v, r_2)$
    $w = f_3(pk', \theta, r_1)$
    Return $C = (u, v, w, r_2)$

$D(sk, C)$
    Parse $C$ as $(u, v, w, r_2)$ and $sk$ as $sk'$
    $\theta = \mathsf{CMhash}(hk, u||v, r_2)$
    Return $m \leftarrow \mathsf{TBEDec}(sk', \theta, (u, v, w))$
```

Figure 3: The Separable-TBE-to-PKE transform

Unlike the fully black-box transform where tags are generated independently of the TBE ciphertext, this construction computes the ciphertext without using any other secret random coins than those of the underlying TBE ciphertext. The tag is derived from a ciphertext component u and some independent randomness r_2 that *publicly* appears in the ciphertext. For this reason, we can hope to avoid the difficulty that appears with the original CHK transform. We prove that it is indeed the case and that any separable TBE that satisfies definition 4 yields an IND-SO-CCA2 encryption scheme.

Theorem 3. If $\mathcal{TBE} = (\mathsf{TBEKg}, \mathsf{TBEEnc}, \mathsf{TBEDec})$ is a separable TBE scheme with IND-SO-stag-wCCA2 security, the transformation of figure 3 gives an IND-SO-CCA2 PKE scheme. For any IND-SO-CCA2 adversary $\mathcal{A}$, there is a TBE adversary $\mathcal{A}^{tbe}$ and a chameleon hash adversary $\mathcal{A}^{hash}$ s.t.

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SO-CCA2}}(\lambda) \leq 2 \cdot \big(\mathbf{Adv}_{\mathcal{A}^{tbe}}^{\text{IND-SO-stag-wCCA2}}(\lambda) + qn\delta + \mathbf{Adv}_{\mathcal{A}^{hash}}^{\text{CR-CMhash}}(\lambda)\big),$$

where q is the number of decryption queries and δ is the maximal probability, taken over the random choice of $r_1 \in \mathcal{R}_{tbe}$, that f_2 outputs a specific element of its range.

Proof. We first note that the definition of IND-SO-CCA2 security is equivalent to a definition where the adversary $\mathcal{A}$ is faced with a simulator and has to decide whether the latter is playing the real game, where the actual plaintexts are revealed after the corruption query, or the ideal game. The game to be played is determined by a random bit $b \in \{0, 1\}$ secretly chosen by the challenger and which $\mathcal{A}$ has to guess.

Using this definition, the proof is similar to [Zha07] and considers two kinds of adversaries.

- Type I attackers never invoke the decryption oracle on (u, v, w, r_2) for which $\mathsf{CMhash}(hk, u||v, r_2)$ collides with a tags $\theta_i^\star$ associated with target ciphertexts.
- Type II adversaries make at least one decryption query for a valid ciphertext (u, v, w, r_2) such that $\mathsf{CMhash}(hk, u||v, r_2)$ hits the tag $\theta_i^\star$ of some target ciphertext.

Type I adversaries are handled similarly to [Zha07]. We outline an adversary $\mathcal{A}^{tbe}$ against the TBE scheme using a type I IND-SO-CCA2 adversary $\mathcal{A}$. The former begins by generating a key pair $(hk, tk) \leftarrow \mathsf{CMhash}(\lambda)$ for the chameleon hash. It chooses dummy $u_i', v_i', r_{2,i}'$ in the appropriate domains and uses them to generate tags $\theta_i^\star = \mathsf{CMhash}(hk, u_i'||v_i', r_{2,i}')$ for $i = 1, \ldots, n$. These are transmitted to $\mathcal{A}^{tbe}$'s challenger $\mathcal{C}$, which replies with a TBE public key pk'. The public key $pk = (pk', hk)$ is given to $\mathcal{A}$.

Any decryption query made by $\mathcal{A}$ is forwarded to $\mathcal{A}^{tbe}$'s challenger $\mathcal{C}$ and the latter's response is relayed to $\mathcal{A}$. When $\mathcal{A}$ outputs a plaintext distribution $\mathcal{M}$, $\mathcal{A}^{tbe}$ sends $\mathcal{M}$ to her own challenger. Upon receiving the vector of target ciphertexts $\mathbf{C}_{tbe}^\star = (C_{tbe}[1]^\star, \ldots, C_{tbe}[n]^\star)$ (where $C_{tbe}[i]^\star = (u_i^\star, v_i^\star, w_i^\star)$ is associated with the tag $\theta_i^\star$), $\mathcal{A}^{tbe}$ uses the trapdoor tk to compute $r_{2,i}^\star = \mathsf{CMswitch}(tk, u_i'||v_i', r_{2,i}', u_i^\star||v_i^\star)$ (in such a way that $\theta_i^\star = \mathsf{CMhash}(hk, u_i^\star||v_i^\star, r_{2,i}^\star) = \mathsf{CMhash}(hk, u_i'||v_i', r_{2,i}')$) and sends the target vector $\mathbf{C}^\star = (\mathbf{C}[1]^\star, \ldots, \mathbf{C}[n]^\star)$, where $\mathbf{C}[i]^\star = (u_i^\star, v_i^\star, w_i^\star, r_{2,i}^\star)$ for all i, to $\mathcal{A}$.

Then, $\mathcal{A}$ makes new decryption queries, which $\mathcal{A}^{tbe}$ handles by simply transmitting them to $\mathcal{C}$ and relaying the latter's responses back to $\mathcal{A}$. When $\mathcal{A}$ decides to make her corruption query $I \subset \{1, \ldots, n\}$, $\mathcal{A}^{tbe}$ sends I to $\mathcal{C}$ that replies with plaintexts and random coins $\{(m_i^\star, r_{1,i}^\star)\}_{i \in I}$ for ciphertexts $\{\mathbf{C}_{tbe}[i]^\star\}_{i \in I}$ as well as $\{m_i\}_{i \notin I}$ for which $\mathcal{A}^{tbe}$ aims at deciding whether $m_i = m_i^\star$ for all i or $m_i \in_R \mathcal{M}$. All these elements are passed to $\mathcal{A}$ (note that $\mathcal{A}^{tbe}$ does not need to include $\{r_{2,i}^\star\}_{i \in I}$ as $\mathcal{A}$ already obtained them as part of $\mathbf{C}[i]^\star$) who makes new decryption queries.

Since $\mathcal{A}$ is assumed to be a Type I adversary, no such decryption query (u, v, w, r_2) ever results in a tag $\theta = \mathsf{CMhash}(hk, u||v, r_2)$ such that $\theta \in \{\theta_1^\star, \ldots, \theta_n^\star\}$, $\mathcal{A}^{tbe}$ can always query $\mathcal{C}$ to decrypt $((u, v, w), \theta)$ and give the answer back to $\mathcal{A}$. Eventually, $\mathcal{A}^{tbe}$ outputs the same result $b' \in \{0, 1\}$ as $\mathcal{A}$ and we easily see that, if $\mathcal{A}$ is successful, so is $\mathcal{A}^{tbe}$. Therefore, it comes that $\mathbf{Adv}^{\text{Type-I}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{IND-SO-stag-wCCA2}}(\mathcal{A}^{tbe})$.

Type II adversaries. In the expectation of a Type II adversary, we construct a collision-finder $\mathcal{A}^{hash}$ that sets up a public key (pk', hk) by obtaining the chameleon hash key hk from a challenger and generates $(sk', pk') \leftarrow \mathsf{TBEKg}(\lambda)$ on its own. It challenges the adversary $\mathcal{A}$ on the public key $pk = (pk', hk)$ and uses the private key sk' to perfectly handle all decryption queries. At the challenge step, $\mathcal{A}$ outputs a distribution $\mathcal{M}$ and obtains a vector $\mathbf{C}^\star = (\mathbf{C}[1]^\star, \ldots, \mathbf{C}[n]^\star)$ of target ciphertexts, where, for each $i \in \{1, \ldots, n\}$, $\mathbf{C}[i]^\star = (u_i^\star, v_i^\star, w_i^\star, r_{2,i}^\star)$ with $u_i^\star = f_1(pk, m_i^\star, r_{1,i}^\star)$, $v_i^\star = f_2(pk, r_{1,i}^\star)$, $\theta_i^\star = \mathsf{CMhash}(hk, u_i^\star||v_i^\star, r_{2,i}^\star)$ and $w_i^\star = f_3(pk, \theta_i^\star, r_{1,i}^\star)$ for plaintexts $m_i^\star \xleftarrow{\$} \mathcal{M}$ and random values $r_{1,i}^\star \xleftarrow{\$} \mathcal{R}_{tbe}, r_{2,i}^\star \xleftarrow{\$} \mathcal{R}_{hash}$.

In the simulation, algorithm $\mathcal{A}^{hash}$ aborts and fails in the event that, for some index $i \in \{1, \ldots, n\}$, the ciphertext $\mathbf{C}[i]^\star = (u_i^\star, v_i^\star, w_i^\star, r_{2,i}^\star)$ is such that $v_i^\star$ previously appeared in a decryption query. This only occurs with probability smaller than $qn\delta$ if δ denotes the maximal probability, taken over the random choice of $r_{1,i}^\star \xleftarrow{\$} \mathcal{R}_{tbe}$, that a specific element of the image of f_2 is reached.

If $\mathcal{A}^{hash}$ does not abort, $\mathcal{A}$ makes new decryption queries that $\mathcal{A}^{hash}$ still perfectly answers using sk'. At some point, $\mathcal{A}$ makes a corruption query I and obtains $\{(m_i^\star, r_{1,i}^\star, r_{2,i}^\star)\}_{i \in I}$. Plaintexts $\{m_i\}_{i \notin I}$ are the actual plaintexts if the challenger $\mathcal{A}^{hash}$'s random bit is $b = 0$ and random plaintexts if $b = 1$.

$\mathcal{A}$ is assumed to query at some point the decryption of some ciphertext $C = (u, v, w, r_2)$ such that $\theta = \mathsf{CMhash}(hk, u||v, r_2) = \mathsf{CMhash}(hk, u_i^\star||v_i^\star, r_{2,i}^\star) = \theta_i^\star$ for some $i \in \{1, \ldots, n\}$. If that query is made before the challenge phase, we must have $v \neq v_i^\star$ as $\mathcal{A}^{hash}$ would have aborted in the challenge phase otherwise. If the query is a post-challenge query, we also have $(u, v, r_2) \neq (u_i^\star, v_i^\star, r_{2,i}^\star)$ since, for any valid ciphertext, $(u, v) = (u_i^\star, v_i^\star)$ and $\theta = \theta_i^\star$ would imply $w = w_i^\star$ and C would be a target ciphertext. In either case, we have a collision on the chameleon hash.

The above arguments give us the upper bound $\mathbf{Adv}^{\text{Type-II}}(\mathcal{A}) \leq qn\delta + \mathbf{Adv}^{\text{CR-CMhash}}(\mathcal{A}^{hash})$.

The theorem is established by noting that $\mathcal{A}^{hash}$ can guess upfront (by flipping a coin independently of $\mathcal{A}$'s view) which kind of attack the adversary will mount and prepare the public key accordingly.

□

4.4 Lossy and All-But-n Trapdoor Functions

Lossy trapdoor functions were first defined in [PW08]. A tuple $(S_{\text{ltdf}}, F_{\text{ltdf}}, F_{\text{ltdf}}^{-1})$ of PPT algorithms is called a family of (d, k)-lossy trapdoor functions if the following properties hold:

- **Sampling injective functions:** $S_{\text{ltdf}}(1^\lambda, 1)$ outputs (s, t), where s is a function index and t its trapdoor. It is required that $F_{\text{ltdf}}(s, \cdot)$ be injective on $\{0, 1\}^d$ and $F_{\text{ltdf}}^{-1}(t, F_{\text{ltdf}}(s, x)) = x$ for all x.

- **Sampling lossy functions:** $S_{\text{ltdf}}(1^\lambda, 0)$ outputs $(s, \perp)$ where s is a function index and $F_{\text{ltdf}}(s, \cdot)$ is a function on $\{0, 1\}^d$, where the image of $F_{\text{ltdf}}(s, \cdot)$ has size at most 2^{d-k}.

- **Indistinguishability:** we have $\{(s, t) \overset{\$}{\leftarrow} S_{\text{ltdf}}(1^\lambda, 1) : s\} \approx_c \{(s, \perp) \overset{\$}{\leftarrow} S_{\text{ltdf}}(1^\lambda, 0) : s\}$.

Along with lossy trapdoor functions, Peikert and Waters [PW08] defined all-but-one (ABO) functions. Essentially, these are lossy trapdoor functions, except instead of having two branches (a lossy branch and an injective branch) they have many branches, all but one of which are injective.

The Peikert-Waters cryptosystem only requires such function families to have one lossy branch because a single challenge ciphertext must be evaluated (on a lossy branch) in the CCA2 game. Since the IND-SO-CCA security game involves $n > 1$ challenge ciphertexts, we need to generalize ABO functions into all-but-n (ABN) functions that have multiple lossy branches and where all branches except the specified ones are injective. In the case $n = 1$, ABN functions obviously boil down to ABO functions.

- **Sampling with a given lossy set:** For any n-subset $I \subset \mathcal{B}$, $S_{\text{abn}}(1^\lambda, I)$ outputs s, t where s is a function index, and t its trapdoor. We require that for any $b \in \mathcal{B} \setminus I$, $G_{\text{abo}}(s, b, \cdot)$ is an injective deterministic function on $\{0, 1\}^d$, and $G_{\text{abn}}^{-1}(t, b, G_{\text{abn}}(s, b, x)) = x$ for all x. Additionally, for each $b \in I$, the image $G_{\text{abn}}(s, b, \cdot)$ has size at most 2^{d-k}.

- **Hidden lossy sets:** For any distinct n-subsets $I_0^\star, I_1^\star \subset \mathcal{B}$, the first outputs of $S_{\text{abn}}(1^\lambda, I_0^\star)$ and $S_{\text{abn}}(1^\lambda, I_1^\star)$ are computationally indistinguishable.

Just as ABO functions can be obtained from lossy trapdoor functions [PW08], ABN functions can also be constructed generically from LTDFs.The recent results of Hofheinz [Hof11a], show how to create All-But-Many Lossy Functions, which are Lossy Trapdoor Functions with a superpolynomial number of lossy branches. The advantage of his construction is that the description of the function is independent of N. Hofheinz's All-But-Many functions can be plugged into our constructions to shrink the size of the public-key in our constructions (see [Hof11a] for details).

4.5 All-But-n Functions from Lossy Trapdoor Functions

Given a set $I \subset \mathcal{B}$, we create an unduplicatable set selector $\mathfrak{g} : \mathcal{B} \to \hat{\mathcal{B}}$. For each $\hat{b} \in \hat{\mathcal{B}}$, we will associate a lossy trapdoor function. Let $\hat{I} = \bigcup_{i \in I} \mathfrak{g}(i)$. For each $\hat{i} \in \hat{I}$, we will create a LTDF in lossy mode, and for each $\hat{b} \in \hat{\mathcal{B}} \setminus \hat{I}$, we will associate a LTDF in injective mode.

- **Sampling with a given lossy set:** Create an $(n, \lceil \log |\mathcal{B}| \rceil)$ unduplicatable set selector $\mathfrak{g}$. Suppose $\mathcal{B} \subset \{0,1\}^v$, then the construction outlined above produces $\mathfrak{g}$ which maps $\{0,1\}^v$ to subsets of $\mathbb{F}_\ell \times \mathbb{F}_\ell$, where $\ell = 2^{\lceil \log_2 2nv \rceil}$. For each element in $\mathbb{F}_\ell \times \mathbb{F}_\ell$, we will associate a lossy trapdoor function. Let $\hat{I} = \bigcup_{i \in I} \mathfrak{g}(i) \subset \mathbb{F}_\ell \times \mathbb{F}_\ell$. For each $y \in \hat{I}$ let F_y be an LTDF in lossy mode, and for each $y \in \mathbb{F}_\ell \times \mathbb{F}_\ell \setminus \hat{I}$, let F_y be an LTDF in injective mode.

  Now, define $G_{\text{abn}}(b, x) = (F_{y_1}(x), \ldots, F_{y_\ell}(x))_{y_i \in \mathfrak{g}(b)}$.

Notice that if any of the functions F_y are injective, then G_{abn} is also injective, and if the image size of F in lossy mode is 2^r, then the images size of G_{abn} on a lossy branch is $2^{r\ell}$. Finally, we notice that the lossy set is hidden by the indistinguishability of modes of the LTDF.

This construction is generic but suffers from a lack of efficiency since the description of the function and its output both have a size growing as a function of n, which is obviously not a desirable property. Luckily for specific lossy trapdoor functions, the growth of the output size can be avoided.

4.6 An IND-SO-stag-wCCA2 TBE Construction

We now give a method for constructing IND-SO-stag-wCCA2 tag-based cryptosystems from lossy trapdoor functions. Using a chameleon hash function $(\mathsf{CMKg}, \mathsf{CMhash}, \mathsf{CMswitch})$ where CMhash ranges over the set of branches $\mathcal{B}$ of the ABN family, we eventually obtain an IND-SO-CCA2 public key encryption scheme. The LTDF-based construction (and its proof) mimics the one [PW08] (in its IND-CCA1 variant).

Let $(S_{\text{ltdf}}, F_{\text{ltdf}}, F_{\text{ltdf}}^{-1})$ be a family of (d, k)-lossy-trapdoor functions, and let $(S_{\text{abn}}, G_{\text{abn}}, G_{\text{abn}}^{-1})$ be a family of (d, k') all-but-n functions with branch set $\{0,1\}^v$ where v is the length of a verification key for our one-time signature scheme. We require that $2d - k - k' \leq t - \kappa$, for $\kappa = \kappa(t) = \omega(\log t)$. Let $\mathcal{H}$ be a pairwise independent hash family from $\{0,1\}^d \to \{0,1\}^\ell$, with $0 < \ell < \kappa - 2\log(1/\nu)$, for some negligible $\nu = \nu(\lambda)$. The message space will be $\mathsf{MsgSp} = \{0,1\}^\ell$.

- $\mathsf{TBEKg}(1^\lambda)$: choose a random member $h \leftarrow \mathcal{H}$ of the pairwise independent hash family and generate
$$(s, t) \leftarrow S_{\text{ltdf}}(1^\lambda, inj), \quad (s', t') \leftarrow S_{\text{abn}}(1^\lambda, \{0, 1, \ldots, n-1\}).$$
  The public key will be $pk = (s, s', h)$ and the secret key will be $sk = (t, t')$.

- $\mathsf{TBEEnc}(m, pk, \theta)$: to encrypt $m \in \{0,1\}^\ell$ under the tag $\theta \in \mathcal{B}$, choose $x \xleftarrow{\$} \{0,1\}^d$. Compute $c_0 = h(x) \oplus m$, $c_1 = F_{\text{ltdf}}(s, x)$ and $c_2 = G_{\text{abn}}(s, \theta, x)$ and set the TBE ciphertext as
$$C = \big(c_0, c_1, c_2\big) = \big(h(x) \oplus m,\ F_{\text{ltdf}}(s, x),\ G_{\text{abn}}(s', \theta, x)\big).$$

- $\mathsf{TBEDec}(C, sk, \theta)$: given $C = \big(c_0, c_1, c_2\big)$ and $sk = t$, compute $x = F_{\text{ltdf}}^{-1}(t, c_1)$ and check whether $G_{\text{abn}}(s, \theta, x) = c_2$. If not, output $\perp$. Otherwise, output $m = c_0 \oplus h(x)$.

The scheme is easily seen to be separable since C is obtained as $c_0 = f_1(pk, m, x) = m \oplus h(x)$, $c_1 = f_2(pk, x) = F_{\text{ltdf}}(s, x)$ and $c_2 = f_3(pk, \theta, x) = G_{\text{abn}}(s', \theta, x)$.

Theorem 4. The algorithms described above form an IND-SO-stag-wCCA2 secure tag-based cryptosystem assuming the security of the lossy and all-but-n families.

Proof. The correctness of the scheme is clear, so we focus on the security. We prove security through a sequence of games which is close to the one of [PW08, Theorem 4.2].

Let Game$_0$ be the real IND-SO-stag-wCCA2 game. In this game, the adversary $\mathcal{A}$ first chooses a set of tags $\{\theta_1^\star, \ldots, \theta_n^\star\}$ under which target ciphertexts will be encrypted in the challenge phase. Recall that $\mathcal{A}$ is not allowed to query the decryption oracle w.r.t. a tag $\theta \in \{\theta_1^\star, \ldots, \theta_n^\star\}$ at any time.

Let Game$_1$ be identical to Game$_0$ except that we set the lossy branches of the all-but-n function G_{abn} to be those identified by $\{\theta_1^\star, \ldots, \theta_n^\star\}$.

Let Game$_2$ be identical to Game$_1$ except that, in the decryption algorithm, we use G_{abn}^{-1} to decrypt instead of F_{ltdf}^{-1}, *i.e.*, we set $x = G_{\text{abn}}^{-1}(t', \theta, c_2)$ instead of $x = F_{\text{ltdf}}^{-1}(t, c_1)$.

Let Game$_3$ be identical to Game$_2$ except that we replace the injective function with a lossy one, *i.e.*, during key-generation we generate $(s, \perp) \leftarrow S_{\text{ltdf}}(1^\lambda, \mathit{lossy})$, instead of $(s, t) \leftarrow S_{\text{ltdf}}(1^\lambda, \mathit{inj})$.

- Game$_1$ and Game$_0$ are indistinguishable by the indistinguishability of lossy sets in ABN functions.

- Game$_2$ does not affect $\mathcal{A}$'s view since she never makes a decryption query on a lossy-branch of G_{abn}.

- The indistinguishability of Game$_3$ and Game$_2$ follows from the indistinguishability of lossy and injective modes of lossy-trapdoor functions.

Now, if we can show that an adversary's probability of success in Game$_3$ is negligible, we will be done. To this end, we follow the proof that Lossy Encryption is selective opening secure and apply Theorem 6 in [BHY09]. The key observation is that in Game$_3$, the challenge ciphertexts are *statistically* independent of the underlying messages. We begin by showing that this is, in fact, the case.

Now, $F_{\text{ltdf}}(s, \cdot)$ and $G_{\text{abn}}(s', \theta_i^\star, \cdot)$ are lossy functions with image sizes at most 2^{d-k} and $2^{d-k'}$ respectively for each $i \in [n]$. Thus the function $x \mapsto (F_{\text{ltdf}}(s, x), G_{\text{abn}}(s', \theta_i^\star, x))$ takes on at most $2^{2d-k-k'} \leq 2^{d-\kappa}$ values. Now by Lemma 2.1 of [PW08], the average min-entropy is bounded below

$$\tilde{H}_\infty(x|c_1, c_2, s, s') \geq H_\infty(x|s, s') - (d - \kappa) = t - (d - \kappa) = \kappa.$$

Since $\ell \leq \kappa - 2\log(1/\nu)$, by Lemma 2.2 of [PW08], for each target ciphertext $C = (c_0, c_1, c_2)$, we have

$$\Delta((c_1, c_2, h, h(x)), (c_1, c_2, h, U_\ell)) \leq \nu,$$

where U_ℓ stands for the uniform distribution on $\{0, 1\}^\ell$. Now, we can incorporate the ideas of Theorem 6. Since the target ciphertexts are statistically independent of the underlying plaintexts, there is a (possibly inefficient)algorithm opener, which, given (c_0, c_1, c_2, m) outputs x such that $F_{\text{ltdf}}(s, x) = c_1$, $G_{\text{abn}}(s, \theta_i^\star, x) = c_2$, and $h(x) \oplus m = c_0$. If no such x exists, opener outputs $\perp$ (the statistical closeness guarantees that this happens with probability at most ν).

Now, let us consider a new series of games. Let Game$_{3_0}$ be identical to Game$_3$, except that target ciphertexts are opened using the output of opener instead of the actual randomness used by the challenger.

Now, for $j \in [n]$, let $\mathrm{Game}3_j$ be identical to $\mathrm{Game}3_0$ except that for $i \leq j$, the target ciphertexts are

$$(E(pk, \xi, r_1), \ldots, E(pk, \xi, r_j), E(pk, m_{j+1}, r_{j+1}), \ldots, E(pk, m_n, r_n))$$

So, the only difference between $\mathrm{Game}3_j$ and $\mathrm{Game}3_{j-1}$ lies in whether the j^{th} target ciphertext is an encryption of a dummy message ξ or m_j. Since these two distributions are *statistically* close, even an *unbounded* adversary has a negligible chance of distinguishing them. Thus by the triangle inequality, an unbounded adversary has a negligible probability of distinguishing $\mathrm{Game}3_0$ from $\mathrm{Game}3_n$.

But $\mathrm{Game}3_n$ is identical in both the real and ideal games, so an adversary has at most a negligible probability of distinguishing the two worlds. □

When the scheme is instantiated with the lossy TDF of [RS09, BFO08] and the ABN function of section 4.7, the proof of the above theorem can be adapted as follows. We simply introduce an intermediate game between Game_1 and Game_2 and consider a failure event which reveals a non-trivial factor of the modulus N if it occurs. In this game, ciphertexts are still decrypted via F_{ltdf}^{-1} and the trapdoor of the ABN function is not used. Suppose that the adversary $\mathcal{A}$ makes a decryption query involving a tag θ such that $\gcd(P(\theta), N) \neq 1$, where $P(\theta) = \prod_{i=1}^{n}(\theta - \theta_i^\star)$. Since $N > 2^\lambda$ and $\theta_i^\star \in \{0,1\}^\lambda$ for each tag $\theta_i^\star$, we cannot have $\theta = \theta_i^\star \bmod N$ for any $i \in \{1, \ldots, n\}$ since it would imply $\theta = \theta_i^\star$ (which is forbidden by the IND-stag-wCCA2 rules). Hence, the failure event would imply $p|(\theta - \theta_i^\star)$ and $q|(\theta - \theta_j^\star)$ for *distinct* $i, j \in \{1, \ldots, n\}$, which would reveal a non-trivial factor of N and *a fortiori* break the DCR assumption.

4.7 An All-but-n Function with Short Outputs

While generic, the all-but-n function of Section 4.5 has the disadvantage of long outputs, the size of which is proportional to nk. Efficient lossy and all-but-one functions can be based on the Composite Residuosity assumption [RS09, BFO08] and the Damgård-Jurik cryptosystem [DJ01]. We show that the all-but-one function of [RS09, BFO08] extends into an all-but-n function that retains short (*i.e.*, independent of n or k) outputs. Multiple lossy branches can be obtained using a technique that traces back to the work of Chatterjee and Sarkar [CS06] who used it in the context of identity-based encryption.

- **Sampling with a given lossy set:** given a security parameter $\lambda \in \mathbb{N}$ and the desired lossy set $I = \{\theta_1^\star, \ldots, \theta_n^\star\}$, where $\theta_i^\star \in \{0,1\}^\lambda$ for each $i \in \{1, \ldots, n\}$, let $\gamma \geq 4$ be a polynomial in λ.

    1. Choose random primes p, q s.t. $N = pq > 2^\lambda$.
    2. Generate a vector $\vec{U} \in (\mathbb{Z}_{N^{\gamma+1}}^*)^{n+1}$ as follows. Let $\alpha_{n-1}, \ldots, \alpha_0 \in \mathbb{Z}_{N^\gamma}$ be coefficients obtained by expanding $P[T] = (T - \theta_1^\star) \cdots (T - \theta_n^\star) = T^n + \alpha_{n-1}T^{n-1} + \cdots + \alpha_1 T + \alpha_0$ in $\mathbb{Z}_{N^\gamma}[T]$ (note that $P[T]$ is expanded in $\mathbb{Z}_{N^\gamma}$ but its roots are all in $\mathbb{Z}_N^*$). Then, for each $i \in \{0, \ldots, n\}$, set $U_i = (1+N)^{\alpha_i} a_i^{N^\gamma} \bmod N^{\gamma+1}$, where $(a_0, \ldots, a_n) \overset{\$}{\leftarrow} (\mathbb{Z}_N^*)^{n+1}$ and with $\alpha_n = 1$.
    3. Set the evaluation key as $s' = \{N, \vec{U}\}$, where $\vec{U}$ is the vector $\vec{U} = (U_0, \ldots, U_n)$, and the domain of the function as $\{0, \ldots, 2^{\gamma\lambda/2} - 1\}$. The trapdoor is defined to be $t' = \mathrm{lcm}(p-1, q-1)$.

- **Evaluation:** to evaluate $G_{\mathrm{abn}}(s', \theta, x)$, where $x \in \{0, \ldots, 2^{\gamma\lambda/2} - 1\}$ and $\theta \in \{0,1\}^\lambda$, compute $c = \big(\prod_{j=0}^{n} U_i^{(\theta^i \bmod N^\gamma)}\big)^x \bmod N^{\gamma+1}$.

- **Inversion:** for a branch θ, $c = G_{\text{abn}}(s', \theta, x)$ is a Damgård-Jurik encryption of $y = P(\theta)x \bmod N^{\gamma}$. Using the trapdoor $t' = \text{lcm}(p-1, q-1)$, the inversion procedure first applies the decryption algorithm of [DJ01] to obtain $y \in \mathbb{Z}_{N^{\gamma}}$ and returns $x = yP(\theta)^{-1} \bmod N^{\gamma}$.

As in [RS09, BFO08], $G_{\text{abn}}(s', \theta, \cdot)$ has image size smaller than N in lossy mode. Hence, the average min-entropy of x can be shown to be at least $\tilde{H}_{\infty}\big(x|(G_{\text{abn}}(s', \theta, x), N, \vec{U})\big) \geq \gamma\lambda/2 - \log(N)$ when $\theta \in I$.

We also note that the ABN function $G_{\text{abn}}(s', \theta, \cdot)$ is not strictly injective for each branch $\theta \notin I$, but only for those such that $\gcd(P(\theta), N^{\gamma}) = 1$. However, the fraction of branches $\theta \in \{0,1\}^{\lambda}$ such that $\gcd(P(\theta), N^{\gamma}) \neq 1$ is bounded by $2/\min(p,q)$, which is negligible.

Moreover, the proof of theorem 4 is not affected if the TBE scheme is instantiated with this particular ABN function and the LTDF of [RS09, BFO08]. As long as factoring is hard (which is implied by the Composite Residuosity assumption), the adversary has negligible chance of making decryption queries w.r.t. to such a problematic tag θ.

Lemma 4. The above ABN function satisfies the hidden lossy set property under the Decisional Composite Residuosity assumption.

Proof. Consider an adversary $\mathcal{A}$ that distinguishes two ABN functions with lossy sets $I_A = \{\theta^{\star}_{A,1}, \ldots, \theta^{\star}_{A,n}\}$ and $I_B = \{\theta^{\star}_{B,1}, \ldots, \theta^{\star}_{B,n}\}$ of its choice. Let $P_A[T]$ and $P_B[T]$ be the n^{th} degree polynomials having their roots in I_A and I_B, respectively. We consider a sequence of games starting with Game_A, where the adversary is given an ABN with lossy set I_A, and ending with Game_B where the ABN has lossy set I_B. Then, we consider a sequence of hybrid games where, for $j = 0, \ldots, n-1$, $\text{Game}_{H,j}$ is defined to be a game where $U_0, \ldots, U_j$ are Damgård-Jurik encryptions of the coefficients of $P_A[T]$ until degree j whereas $U_{j+1}, \ldots, U_{n-1}$ encrypt the coefficients of $P_B[T]$. Obviously, any adversary distinguishing Game_A from $\text{Game}_{H,0}$ implies a semantic security adversary against Damgård-Jurik and the same argument applies to subsequent game transitions. The result follows by noting that Game_B is identical to $\text{Game}_{H,n-1}$. □

The above ABN function yields an IND-SO-CCA2 secure encryption scheme with ciphertexts of constant (*i.e.*, independent of n) size but a public key of size $O(n)$. Encryption and decryption require $O(n)$ exponentiations as they entail an ABN evaluation. On the other hand, the private key has $O(1)$ size as well, which keeps the private storage very cheap. At the expense of sacrificing the short private key size, the decryption algorithm can be optimized by computing $x = G^{-1}_{\text{abn}}(t', \theta, c_2)$ (instead of $x = F^{-1}_{\text{ltdf}}(t, c_1)$) so as to avoid computing $G_{\text{abn}}(s', \theta, x)$ in the forward direction to check the validity of ciphertexts. In this case, the receiver has to store the coefficients $\alpha_0, \ldots, \alpha_{n-1}$ to evaluate $P(\theta)$ when inverting G_{abn}.

It is also possible to extend the DDH-based ABO function described in [PW08] into an ABN function. However, the next section describes a more efficient lossy TBE scheme based on the DDH assumption.

4.8 An IND-SO-stag-wCCA2 TBE Scheme from the DDH Assumption

The DDH problem informally consists in, given (g, g^x, g^y, g^z), to decide whether $z = xy$ or not (a rigorous definition is recalled in appendix

Rigorously,

Definition 5. The **Decisional Diffie-Hellman** (DDH) problem in a group $\mathbb{G}$, is to distinguish the distributions $D_1 = \{x, y \xleftarrow{\$} \mathbb{Z}_p : (g, g^x, g^y, g^{xy})\}$ and $D_2 = \{x, y \xleftarrow{\$} \mathbb{Z}_p; z \xleftarrow{\$} \mathbb{Z}_p \setminus \{xy\} :$

$(g, g^x, g^y, g^z)\}$. The **DDH assumption** posits that, for any PPT distinguisher $\mathcal{D}$, the following function is negligible

$$\mathbf{Adv}_{\mathbb{G},\mathcal{D}}^{\mathrm{DDH}}(\lambda) = |\Pr[\mathcal{D}(\{(g,X,Y,Z) \stackrel{\$}{\leftarrow} D_1 : g,X,Y,Z\}) = 1] - \Pr[\mathcal{D}(\{(g,X,Y,Z) \stackrel{\$}{\leftarrow} D_2 : g,X,Y,Z\}) = 1]|.$$

The system builds on the DDH-based lossy encryption scheme of [NP01, PVW08, BHY09] and could be seen as a variant of the encryption scheme described in [CKS08, Section 6.2], which is itself situated half-way between the Cramer-Shoup [CS98, CS02] and CHK methodologies [CHK04].

Again, attention must be paid to the fact that the adversary sees $n > 1$ challenge ciphertexts with different tags. To apply the technique of [CKS08] (which uses ideas that were initially proposed for identity-based encryption [BB04]) in the security proof, we need some function of the tag to cancel in the exponent for each target ciphertext. This issue can be addressed using the technique of [CS06].

$\mathsf{TBEKg}(1^\lambda)$**:** choose a group $\mathbb{G}$ of prime order $p > 2^\lambda$ with a generators $g, h \stackrel{\$}{\leftarrow} \mathbb{G}$. Pick $a_i, b_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, for $i = 0, \ldots, n$, and compute $U_i = g^{a_i}$, $V_i = h^{a_i}$, $W_i = g^{b_i}$, $Z_i = h^{b_i}$ and $Y_1 = g^y$, $Y_2 = h^y$ for a random $y \stackrel{\$}{\leftarrow} \mathbb{Z}_p$. Set the public key as $pk = \{\mathbb{G}, g, h, \vec{U}, \vec{V}, \vec{W}, \vec{Z}, X_1, X_2\}$ and define the private key to be $sk = (\vec{a}, \vec{b}, y)$, for $(n+1)$-vectors $\vec{U} = (U_0, \ldots, U_n)$, $\vec{V} = (V_0, \ldots, V_n)$, $\vec{W} = (W_0, \ldots, W_n)$, $\vec{Z} = (Z_0, \ldots, Z_n)$, $\vec{a} = (a_0, \ldots, a_n)$ and $\vec{b} = (b_0, \ldots, b_n)$.

$\mathsf{TBEEnc}(pk, \theta, m)$**:** to encrypt m under the tag $\theta \in \mathbb{Z}_p$ given pk,

1. Choose $r, s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and compute $C_0 = m \cdot Y_1^r \cdot Y_2^s$, $C_1 = g^r \cdot h^s$.
2. Set $C_2 = \left(\prod_{j=0}^n U_j^{\theta j}\right)^r \cdot \left(\prod_{j=0}^n V_j^{\theta j}\right)^s$ and $C_3 = \left(\prod_{j=0}^n W_j^{\theta j}\right)^r \cdot \left(\prod_{j=0}^n Z_j^{\theta j}\right)^s$.

Set the ciphertext as $C = \Big(C_0, C_1, C_2, C_3\Big)$.

$\mathsf{TBEDec}(sk, \theta, C)$**:** given $sk = (\vec{a}, \vec{b}, y)$, θ and $C = (C_0, C_1, C_2, C_3)$, return $\perp$ if $C_2 \neq C_1^{\sum_{j=0}^n a_j \theta^j}$ or $C_3 \neq C_1^{\sum_{j=0}^n b_j \theta^j}$. Otherwise, return $m = C_0 / C_1^y$.

This scheme is separable since functions f_1, f_2 and f_3 can be defined so that $C_0 = f_1(pk, m, (r,s))$, $C_1 = f_2(pk, (r,s))$ and $(C_2, C_3) = f_3(pk, \theta, (r,s))$. The chameleon-hash-based transformation thus applies and we only have to prove that the TBE system satisfies IND-SO-stag-wCCA2 security.

Theorem 5. For any adversary $\mathcal{A}$ making q decryption queries, we have $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SO-stag-wCCA2}}(\lambda) \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + q/2^\lambda$.

Proof. The proof consists of a sequence of games, the first one of which is the real game. In all games, we call S_i the event that the adversary $\mathcal{A}$ outputs 1 in Game$_i$.

Game$_0$: the adversary chooses n tags $\theta_1^\star, \ldots, \theta_n^\star$ and is supplied with a public key for which $\vec{U}$, $\vec{V}$, $\vec{W}$, $\vec{Z}$, Y_1, Y_2 are generated such that $Y_1 = g^y$, $Y_2 = h^y$, for some $y \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, and $U_i = g^{a_i}$, $V_i = h^{a_i}$, $W_i = g^{b_i}$ and $Z_i = h^{b_i}$ for $i \in \{0, \ldots, n\}$ where $(a_0, \ldots, a_n) \stackrel{\$}{\leftarrow} (\mathbb{Z}_p)^{n+1}$ and $(b_0, \ldots, b_n) \stackrel{\$}{\leftarrow} (\mathbb{Z}_p)^{n+1}$.

The adversary $\mathcal{A}$ makes decryption queries which the simulator $\mathcal{D}$ handles using $sk = (\vec{a}, \vec{b}, y)$, where $\vec{a} = (a_0, \ldots, a_n)$, $\vec{b} = (b_0, \ldots, b_n)$. After polynomially-many decryption queries, $\mathcal{A}$ makes a unique challenge query for a message distribution $\mathcal{M}$ of her choice. Then, $\mathcal{D}$ uniformly samples n plaintexts $(m_1^\star, \ldots, m_n^\star) \stackrel{\$}{\leftarrow} \mathcal{M}^n$ and generates a vector of ciphertexts $\mathbf{C}^\star = (\mathbf{C}[1]^\star, \ldots, \mathbf{C}[n]^\star)$.

For $i \in \{1, \ldots, n\}$, let us call $r_i^\star, s_i^\star \in \mathbb{Z}_p$ the random exponents that are used to generate $\mathbf{C}[i]^\star$ such that $\mathbf{C}[i]^\star = (C_{i,0}^\star, C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star)$ is equal to

$$\Big(m_i^\star \cdot Y_1^{r_i^\star} \cdot Y_2^{s_i^\star},\ g^{r_i^\star} \cdot h^{s_i^\star},\ (\prod_{j=0}^{n} U_j^{t^j})^{r_i^\star} \cdot (\prod_{j=0}^{n} V_j^{t^j})^{s_i^\star},\ (\prod_{j=0}^{n} W_j^{t^j})^{r_i^\star} \cdot (\prod_{j=0}^{n} Z_j^{t^j})^{s_i^\star}\Big).$$

After having obtained the vector $\mathbf{C}^\star$, $\mathcal{A}$ makes further decryption queries (C, θ) such that $\theta \notin \{\theta_1^\star, \ldots, \theta_n^\star\}$. At some point, she makes a corruption query and chooses a subset $I \subset \{1, \ldots, n\}$ such that $\#I = n/2$. At this stage, $\mathcal{D}$ returns $\{(m_i^\star, (r_i^\star, s_i^\star))\}_{i \in I}$. As for indices $i \in \{1, \ldots, n\} \setminus I$ corresponding to unopened plaintexts, $\mathcal{D}$ only returns the actual plaintexts $\{m_i^\star\}_{i \notin I}$. The adversary $\mathcal{A}$ makes further decryption queries (C, θ) subject to the rule that $\theta \notin \{\theta_1^\star, \ldots, \theta_n^\star\}$. We call S_0 the event that $\mathcal{A}$ eventually outputs 1.

Game$_1$: is the same as Game$_0$ but we modify the generation of the public key. Namely, to generate $pk = \{\mathbb{G}, g, h, f, \vec{U}, \vec{V}, \vec{W}, \vec{Z}, Y_1, Y_2\}$, the simulator $\mathcal{D}$ first computes $X_1 = g^x$ and $X_2 = h^x$, for a random $x \xleftarrow{\$} \mathbb{Z}_p$, and calculates Y_1, Y_2 and vectors $(\vec{U}, \vec{V}, \vec{W}, \vec{Z})$ in the following way. The simulator $\mathcal{D}$ uniformly picks $\alpha_n, \beta_0, \ldots, \beta_n, \gamma_0, \ldots, \gamma_n \xleftarrow{\$} \mathbb{Z}_p$. It obtains coefficients $\alpha_{n-1}, \ldots, \alpha_0$ by expanding the polynomial $P[T] = \alpha_n(T - \theta_1^\star) \ldots (T - \theta_n^\star) = \alpha_n T^n + \alpha_{n-1}T^{n-1} + \cdots + \alpha_1 T + \alpha_0$. Then, it defines $Y_1 = g^{\omega_1} X_1^{\omega_2}$ and $Y_2 = h^{\omega_1} X_2^{\omega_2}$ for randomly drawn $\omega_1, \omega_2 \xleftarrow{\$} \mathbb{Z}_p$. For each $i \in \{0, \ldots, n\}$, it sets

$$U_i = X_1^{\alpha_i} g^{\beta_i},\ V_i = X_2^{\alpha_i} h^{\beta_i},\ W_i = Y_1^{\alpha_i} g^{\gamma_i},\ Z_i = Y_2^{\alpha_i} h^{\gamma_i}.$$

This implicitly defines private keys elements $\vec{a}, \vec{b}$ and y to be $a_i = \alpha_i x + \beta_i$, $b_i = \alpha_i y + \gamma_i$, for $i \in \{0, \ldots, n\}$, and $y = \omega_1 + x\omega_2$. The distribution of pk is not modified and we have $\Pr[S_1] = \Pr[S_0]$.

Game$_2$: we modify the decryption oracle. For a decryption query (C, θ) where $C = \big(C_0, C_1, C_2, C_3\big)$ with $\theta \notin \{\theta_1^\star, \ldots, \theta_n^\star\}$, $\mathcal{D}$ evaluates the polynomials $Q_2[T] = \sum_{j=0}^{n} \beta_i T^j$ and $Q_3[T] = \sum_{j=0}^{n} \gamma_j T^j$ for $T = \theta$ and computes $A_i = (C_i / C_1^{Q_i(\theta)})^{1/P(\theta)}$ for $i \in \{2, 3\}$. The consistency of the ciphertext is verified by checking whether $C_1^{\omega_1} A_2^{\omega_2} = A_3$ and returning $\perp$ if this is not the case.

This consistency check stems from the "Twin Diffie-Hellman trapdoor test" [CKS08, Theorem 2], the idea of which is the following. If C is well-formed, for any pair (r, s) such that $C_1 = g^r h^s$, we must have $A_2 = X_1^r X_2^s$ and $A_3 = Y_1^r Y_2^s$ (so that $A_3 = C_1^{\omega_1} A_2^{\omega_2}$ and the test is successful).

Let us assume that there exists no r, s such that $C_1 = g^r h^s$, $C_2 = (g^{Q_2(\theta)} X_1^{P(\theta)})^r (h^{Q_2(\theta)} X_2^{P(\theta)})^s$ and $C_3 = (g^{Q_3(\theta)} Y_1^{P(\theta)})^r (h^{Q_3(\theta)} Y_2^{P(\theta)})^s$. The trapdoor test amounts to check whether there exists $\tau = r + \log_g(h)s$ such that $C_1 = g^\tau$, $C_2 = (g^{Q_2(\theta) + xP(\theta)})^\tau$ and $C_3 = (g^{Q_3(\theta) + yP(\theta)})^\tau$. If this is not the case, $\mathcal{D}$ obtains $A_2 = g^{x\tau_1}$ and $A_3 = g^{y\tau_2}$ such that either $\tau_1 \neq \tau$ or $\tau_2 \neq \tau$. It is easy to see that the trapdoor test cannot be satisfied if $\tau = \tau_1$ and $\tau \neq \tau_2$ and we thus assume that $\tau_1 \neq \tau$. In this case, we can write $A_2 = g^{x(\tau + \tau_1')}$, for some $\tau_1' \neq 0$, and the value $C_1^{\omega_1} A_2^{\omega_2}$ can in turn be written $g^{\tau(\omega_1 + x\omega_2)} \cdot g^{x\tau_1'\omega_2} = g^{\tau y} \cdot g^{x\tau_1'\omega_2}$, which is uniformly random from $\mathcal{A}$'s view (since the product $x\omega_2$ is perfectly hidden). Moreover, conditionally on a fixed $y = \log_g(Y_1)$, the distribution of A_3 does not depend on $x\omega_2$ since $A_3 = (C_3 / C_1^{Q_3(\theta)})^{1/P(\theta)}$ can be expressed as $A_3 = C_1^y \cdot (h^{\frac{Q_3(\theta)}{P(\theta)}} \cdot Y_2)^{s' - s}$ where (s, s') are such that $s' = s$ if $C_3 = C_1^{Q_3(\theta) + yP(\theta)}$. It comes that the condition $A_3 = C_1^{\omega_1} A_2^{\omega_2}$ cannot be satisfied with better probability than $1/q$ and C is thus rejected with probability $1 - 1/q$.

If the check succeeds, $\mathcal{D}$ returns $m = C_0 / A_3$. We have $|\Pr[S_2] - \Pr[S_1]| \leq q/p \leq q/2^\lambda$ as Game 2 and Game 1 are identical until $\mathcal{D}$ accepts a ciphertext that would have been rejected in Game 1.

Game$_3$: we modify again the generation of pk. Now, $\mathcal{D}$ computes $X_1 = g^x$ and $X_2 = h^{x'}$, where $x \xleftarrow{\$} \mathbb{Z}_p$, $x' \xleftarrow{\$} \mathbb{Z}_p \setminus \{x\}$ (instead of $X_2 = h^x$). All other calculations (including the generation of $\mathbf{C}^\star$ and the decryption oracle) remain unchanged. In particular, $\mathcal{D}$ still knows the encryption exponents $r_i^\star, s_i^\star \in \mathbb{Z}_p$ that are used to encrypt $\mathbf{C}[i]^\star$, for $i \in \{1, \ldots, n\}$, and the exponents $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$ used in the previous game.

The decryption oracle still consistently handles decryption queries as they involve tags $\theta \notin \{\theta_1^\star, \ldots, \theta_n^\star\}$. For any queried ciphertext $C = (C_0, C_1, C_2, C_3)$, given that $\log_g(X_1) \neq \log_h(X_2)$, there always exist (r, s) such that $C_1 = g^r h^s$ and $C_2 = (g^{Q_2(\theta)} X_1^{P(\theta)})^r (h^{Q_2(\theta)} X_2^{P(\theta)})^s$. For these values (r, s), the decryption oracle obtains $A_2 = X_1^r X_2^s$. Likewise, there always exists a pair of integers (r', s') satisfying $C_1 = g^{r'} h^{s'}$ and $C_3 = (g^{Q_3(\theta)} Y_1^{P(\theta)})^{r'} (h^{Q_3(\theta)} Y_2^{P(\theta)})^{s'}$ and $\mathcal{D}$ obtains $A_3 = Y_1^{r'} Y_2^{s'}$. If C is well-formed, we have $(r, s) = (r', s')$ and the oracle returns $m = C_0/A_3$ as in previous games. If $(r, s) \neq (r', s')$, A_3 can be written $A_3 = Y_1^r Y_2^{s_1}$, for some $s_1 \neq s$, so that $A_3/(C_1^{\omega_1} A_2^{\omega_2}) = Y_2^{s_1 - s} \neq 1_{\mathbb{G}}$ and the test rejects C.

Any notable difference between Game$_3$ and Game$_2$ would give a DDH-adversary. To construct a distinguisher that bridges between these games, we consider a DDH instance $(g, h, X_1 = g^x, X_2)$ and generate the public key as in Game$_1$. It comes that key generation proceeds as in Game$_2$ if $X_2 = h^x$ and mirrors Game$_3$ otherwise. Hence, $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)$.

In Game$_3$, ciphertexts $\mathbf{C}[i]^\star$ are statistically independent of plaintexts. Indeed, they are of the form

$$(C_{i,0}^\star, C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star) = \Big(m_i^\star \cdot Y_1^{r_i^\star} Y_2^{s_i^\star},\ g^{r_i^\star} h^{s_i^\star},\ (g^{r_i^\star} h^{s_i^\star})^{Q_2(t_i^\star)},\ \ (g^{r_i^\star} h^{s_i^\star})^{Q_3(t_i^\star)}\Big),$$

so that, since $\mathcal{A}$ knows $Q_2(\theta_i^\star)$ and $Q_3(\theta_i^\star)$ in the information-theoretic sense, the information revealed by $C_{i,1}^\star$, $C_{i,2}^\star$, $C_{i,3}^\star$ is redundant and leaves p equally-likely candidates for the pair $(r_i^\star, s_i^\star)$. The value $Y_1^{r_i^\star} Y_2^{s_i^\star}$ is then easily seen to statistically hide $m_i^\star$ since $\log_g(Y_1) \neq \log_h(Y_2)$. Even an all-powerful adversary would be unable to tell whether she obtains the real plaintext $m_i^\star$ or a resampled one. The proof is completed using a sequence of n hybrid games exactly as in the end of the proof of theorem 4. □

As in the Paillier-based scheme, the number n of target ciphertexts must be known at key generation since public keys have size $O(n)$. As long as n is not too large, the encryption cost remains acceptable: if n is a linear polynomial in λ for instance, the encryption algorithm has complexity $O(\lambda^4)$. Avoiding this dependency seems rather challenging (at least in the standard model) with the current state of knowledge.

On the other hand, ciphertexts consist of a constant number of group elements and decryption entails a constant number of exponentiations.

5 Conclusion

We showed that lossy encryption, which is known to provide IND-SO-CPA secure encryption schemes, is implied by the re-randomizable encryption primitive as well as by $\binom{2}{1}$-Oblivious Transfer (and thus also by PIR, homomorphic encryption and smooth hash proof systems).

Our constructions explain an existing scheme and give rise to new IND-SO-CPA secure cryptosystems based on the Decisional Composite Residuosity (DCR) and Quadratic Residuosity (QR) assumptions. These new schemes retain the efficiency of underlying protocols and immediately yield simple and efficient IND-SO-COM secure commitments. From Paillier's cryptosystem, we additionally obtained the most bandwidth-efficient SEM-SO-CPA secure encryption scheme to date and the first one based on the DCR assumption.

In the chosen-ciphertext selective opening scenario, we described new schemes fitting indistinguishability and simulation-based definitions. As for the former, we showed how to reach security in its sense using schemes with short ciphertexts. The recent results of Hofheinz [Hof11a] show how create All-But-Many Lossy Functions, which can be used to eliminate the $\mathcal{O}(n)$ complexity in terms of public key size in our constructions while retaining short ciphertexts. This significantly increases the utility of our constructions.

Acknowledgements:

We thank Yuval Ishai for suggesting a connection between Oblivious Transfer and Lossy Encryption.

References

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without Random Oracles. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, 2004.

[Bea97] Donald Beaver. Plug and play encryption. In *CRYPTO '97*, pages 75–89, London, UK, 1997. Springer-Verlag.

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In David Wagner, editor, *CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 335–359. Springer, 2008.

[BH92] Donald Beaver and Stuart Haber. Cryptographic protocols provably secure against dynamic adversaries. In *EUROCRYPT '92*, number 658 in Lecture Notes in Computer Science, pages 307–323. Springer-Verlag, 1992.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT '09*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer Berlin / Heidelberg, 2009.

[BK05] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103, 2005.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs #1. In *CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1998.

[BWY11] Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure under selective opening attack. In *TCC '11*, pages 235–252, 2011.

[BY09] Mihir Bellare and Scott Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive: Report 2009/101, 2009.

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 90–104, London, UK, 1997. Springer-Verlag.

[CFGN96] Ran Canetti, Uri Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 639–648, New York, NY, USA, 1996. ACM.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, pages 207–222. Springer, 2004.

[CHK05] Ran Canetti, Shai Halevi, and Jon Katz. Adaptively-secure, non-interactive public-key encryption. In *TCC '05*, number 3378 in Lecture Notes in Computer Science, pages 150–168. Springer-Verlag, 2005.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *STOC '98*. ACM, 1998.

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 565–582. Springer, 2003.

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. In *EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 127–145. Springer, 2008.

[CMO00] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In *EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 122–138. Springer Berlin / Heidelberg, 2000.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In - *EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64, 2002. Full version available at `http://eprint.iacr.org` Cryptology ePrint Archive, Report 2001/085.

[CS06] Sanjit Chatterjee and Palash Sarkar. Generalization of the selective-ID security model for HIBE protocols. In *9th International Conference on Theory and Practice of Public-Key Cryptography (PKC 2006)*, pages 241–256. Springer, 2006.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *STOC '91*, pages 542–552, 1991.

[DJ01] Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 119–136, London, UK, 2001. Springer-Verlag.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry Stockmeyer. Magic functions: In memoriam: Bernard m. dwork 1923–1998. *Journal of the ACM*, 50(6):852–921, 2003.

[FHKW10] Serge Fehr, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *Eurocrypt '10*, pages 381–402. Springer, 2010.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In *Proceedings of Eurocrypt 2006, volume 4004 of LNCS*, pages 339–358. Springer, 2006.

[Gro04] Jens Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *TCC '04*, volume 2951 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2004.

[HK07] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. Cryptology ePrint Archive, Report 2007/118, 2007. http://eprint.iacr.org/2007/118.

[Hof11a] Dennis Hofheinz. All-but-many lossy trapdoor functions. Cryptology ePrint Archive: Report 2011/230, 2011.

[Hof11b] Dennis Hofheinz. Possibility and impossibility results for selective decommitments. *Journal of Cryptology*, 24(3), 2011.

[IKO05] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Sufficient conditions for collision resistant hashing. In *TCC '05*, volume 3378, pages 445–456. Springer Berlin / Heidelberg, 2005.

[JJS04] Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *In Proceedings of the 2004 RSA Conference, Cryptographers track*, pages 163–178. Springer-Verlag, 2004.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In *EUROCRYPT '05*, volume 3494 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *Theory of Cryptograhy Conference 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.

[KN08] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *TCC '08 : Proceedings of the fifth annual Theory of Cryptography Conference*, pages 320–339. Springer Berlin / Heidelberg, 2008.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieva. In *FOCS '97*, pages 364–373. ACM, 1997.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *Network and Distributed System Security Symposium (NDSS 2000)*, 2000.

[Lin06] Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *Journal of Cryptology*, 19(3):359–377, 2006.

[Man98] Eran Mann. Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, 1998.

[MRY04] Philip McKenzie, Michael Reiter, and Ke Yang. Alternatives to non-malleability: Definitions, constructions, and applications. In *Theory of Cryptograhy Conference 2004*, volume 171-190. Springer, 2004.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA '01*, pages 448–457. ACM/SIAM, 2001.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC '90*, pages 427–437, 1990.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin / Heidelberg, 1999.

[PR07] Manoj Prabhakaran and Mike Rosulek. Rerandomizable RCCA encryption. In *CRYPTO '07*, volume 4622 of *Lecture Notes in Computer Science*, pages 517–534. Springer Berlin / Heidelberg, 2007.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 187–196, New York, NY, USA, 2008. ACM.

[RS91] Charles Rackoff and Daniel Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO '91*, pages 433–444, 1991.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC '09*, pages 419–436, Berlin, Heidelberg, 2009. Springer-Verlag.

[Sah99] Amit Sahai. Non-malleable non-interactive zero-knowledge, and adaptive chosen-ciphertext security. In *FOCS '99*, pages 543–553, 1999.

[SCO$^+$01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598. Springer Berlin / Heidelberg, 2001.

[YY05] Adam Young and Moti Yung. Questionable encryption and its applications. In *1st International Conference on Cryptology in Malaysia (Mycrypt'05)*, volume 3715 of *Lecture Notes in Computer Science*, pages 210–221. Springer, 2005.

[Zha07] Rui Zhang. Tweaking TBE/IBE to PKE transforms with chameleon hash functions. In *Applied Cryptography and Network Security (ACNS'07)*, pages 323–339, 2007.

Appendix

A Selective Opening Secure Commitments

A.1 Re-Randomizable One-Way Functions

A family of functions $\mathcal{F}$, indexed by a security parameter λ is called a *re-randomizable one-way function* family if the following conditions are satisfied

- **Efficiently Computable:** For all $f \in \mathcal{F}$, the function $f : M \times R \to Y$ is efficiently computable.
- **One-Way:** For all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

  $$\Pr\left[f \leftarrow \mathcal{F}; (m_0, m_1, st) \leftarrow \mathcal{A}_1(f); b \leftarrow \{0,1\}; r \leftarrow R; b' \leftarrow \mathcal{A}_2(f(m_b, r), st) : b = b'\right] < \frac{1}{2} + \nu$$

  for some negligible function ν (of λ).
- **Injective on the first input:** For all $m \neq m' \in M$, and $r, r' \in R$, $f(m,r) \neq f(m',r')$. This is equivalent to the statement $f(m,R) \cap f(m',R) = \emptyset$ for all $m \neq m' \in M$.
- **Re-randomizable:** For each f, there exists and efficient function ReRand such that, for all $m \in M$ and $r_0 \in R$, we have $\{r \leftarrow R; f(m,r)\} \approx_s \{r \leftarrow \mathsf{coins}(\mathsf{ReRand});\ \mathsf{ReRand}(f(m,r_0),r)\}$.

It is easy to see that the encryption algorithm from a re-randomizable encryption scheme is immediately a re-randomizable one-way function. We note, however, that re-randomizable one-way functions are a significantly weaker primitive since we do not require any kind of trapdoor.

A.2 Commitments from Re-Randomizable One-Way Functions

We begin by describing a construction of a simple bit commitment scheme that arises from any re-randomizable one-way function. Let $\mathcal{F}$ be a re-randomizable one-way function family. The bit commitment system is depicted on figure 4.

Parameter Generation:	**Commitment:**
$(f, \mathsf{ReRand}) \leftarrow \mathcal{F}(1^\lambda)$	$r' \leftarrow \mathsf{coins}(\mathsf{ReRand})$
$r_0, r_1 \leftarrow R$	$\mathrm{Com}(b, r') = \mathsf{ReRand}(c_b, r')$
$c_0 = f(b_0, r_0)$	**De-commitment:**
$c_1 = f(b_1, r_1)$	To de-commit, simply reveal the randomness r'.

Figure 4: Commitments from re-randomizable one-way functions

This scheme has a number of useful properties. If $b_0 = b_1$, the scheme is statistically hiding by the properties of ReRand. Alternatively, if $b_0 \neq b_1$, the scheme is perfectly binding by the injectivity of f on its first input. Now, the two modes are indistinguishable by the one-wayness of f. Combining this with the preceding observations, we also obtain that the scheme is computationally binding if $b_0 = b_1$ and computationally hiding if $b_0 \neq b_1$.

The security analysis is very straightforward but, as this will be the foundation of all our constructions, we include it hereafter.

Lemma 5. If $b_0 = b_1$, the commitment scheme of figure 4 is statistically hiding. If $b_0 \neq b_1$, then it is perfectly binding.

Proof. If $b_0 = b_1$, we have

$$\{r' \leftarrow \mathsf{coins}(\mathrm{Com}) : \mathrm{Com}(0, r')\} \approx_s \{s' \leftarrow \mathsf{coins}(\mathrm{Com}) : \mathrm{Com}(1, s')\},$$

by the definition of ReRand. On the other hand, if $b_0 \neq b_1$, $\mathrm{Com}(0, r) \in f(b_0, R)$ and $\mathrm{Com}(1, s) \in f(b_1, R)$, but by the injectivity on the first input, these sets are necessarily disjoint . □

Lemma 6. Instantiations of the scheme with $b_0 = b_1$ and $b_0 \neq b_1$ are computationally indistinguishable.

Proof. This is exactly the one-way property of f. □

Corollary 4. If $b_0 = b_1$, the scheme is computationally binding. If $b_0 \neq b_1$, it is computationally hiding.

Proof. Since the scheme is perfectly binding when $b_0 \neq b_1$, breaking the binding property amounts to a proof that $b_0 = b_1$. Since the two modes are computationally indistinguishable, no computationally bounded adversary can create such a "proof." Similarly, since the scheme is perfectly hiding when $b_0 = b_1$, breaking the hiding property amounts to showing that $b_0 \neq b_1$, since the two modes are computationally indistinguishable, no probabilistic polynomial-time adversary can break the hiding property. □

The ability to choose whether the commitment scheme will be statistically hiding or perfectly binding is a valuable property, but it is the fact that this choice can be hidden *from the committer* that makes this construction truly useful.

A.3 Definitions of Selective Opening Secure Commitments

Definition 6. (Indistinguishability of commitments under selective openings). A non-interactive commitment scheme $(\mathrm{Com}, \mathrm{Dec})$ is indistinguishable under selective openings (or IND-SO-COM secure) if, for any polynomial n, any n-message distribution $\mathcal{M}$ supporting efficient conditional resampling and any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have

$$\left|\Pr\left[\mathcal{A}^{\mathsf{ind\text{-}so\text{-}real}} = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{ind\text{-}so\text{-}ideal}} = 1\right]\right| < \nu$$

for some negligible function ν, and where the games $\mathsf{ind\text{-}so\text{-}real}$ and $\mathsf{ind\text{-}so\text{-}ideal}$ are defined as follows

```
**IND-SO-COM (Real):**
  $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$
  $r_1, \ldots, r_n \leftarrow \mathsf{coins}(\mathrm{Com})$
  $(I, st) \leftarrow \mathcal{A}_1\big(\mathsf{par}, \mathrm{Com}(m_1, r_1), \ldots, \mathrm{Com}(m_n, r_n)\big)$
  $b \leftarrow \mathcal{A}_2\big(st, \mathrm{Dec}(\mathrm{Com}(m_i, r_1))_{i \in I}, \mathbf{m}\big)$
```

```
**IND-SO-COM (Ideal):**
  $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$
  $r_1, \ldots, r_n \leftarrow \mathsf{coins}(\mathrm{Com})$
  $(I, st) \leftarrow \mathcal{A}_1\big(\mathsf{par}, \mathrm{Com}(m_1, r_1), \ldots, \mathrm{Com}(m_n, r_n)\big)$
  $\mathbf{m}' = (m'_1, \ldots, m'_n) \leftarrow \mathcal{M}_{|I, \mathbf{m}[I]}$
  $b \leftarrow \mathcal{A}_2\big(st, \mathrm{Dec}(\mathrm{Com}(m_i, r_i))_{i \in I}, \mathbf{m}'\big)$
```

Figure 5: IND-SO-COM Security

More explicitly, in the real game, the challenger samples messages $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$ from the joint message distribution and picks random coins $r_1, \ldots, r_n \leftarrow \mathsf{coins}(\mathrm{Com})$ to compute n

commitments $\mathrm{Com}(m_1, r_1), \ldots, \mathrm{Com}(m_n, r_n)$ which are sent to $\mathcal{A}$ along with a description of public parameters par. The adversary $\mathcal{A}$ responds by choosing a subset $I \subset \{1, \ldots, n\}$ of size $n/2$. Then, the challenger de-commits $\{\mathrm{Com}(m_i, r_i)\}_{i \in I}$ and hands the result $\{(m_i, r_i)\}_{i \in I}$ to $\mathcal{A}$. Finally, the challenger sends $\mathbf{m}$ to the adversary $\mathcal{A}$ who eventually outputs a bit $b \in \{0, 1\}$.

The ideal game proceeds identically to the real game until the opening query. At this stage, the challenger still de-commits $\{\mathrm{Com}(m_i, r_i)\}_{i \in I}$ by revealing $\{(m_i, r_i)\}_{i \in I}$ to $\mathcal{A}$. Instead of revealing $\mathbf{m}$ however, it samples a new vector $\mathbf{m}' \leftarrow \mathcal{M}_{|I, \mathbf{m}[I]}$ from $\mathcal{M}$ conditioned on the fact that $m_i = m'_i$ for $i \in I$ and sends it to $\mathcal{A}$ who eventually outputs a bit $b \in \{0, 1\}$.

A.4 IND-SO-COM Constructions from Re-Randomizable One-Way Functions

To construct an IND-SO-COM secure commitment scheme, it suffices to create a statistically hiding commitment scheme as was demonstrated by Bellare, Hofheinz and Yilek [BHY09].

Theorem 6. [BHY09] Statistically-hiding commitment schemes are IND-SO-COM secure.

Since the commitment scheme constructed in Appendix A.2 is statistically hiding when $b_0 = b_1$, we obtain the following corollary

Corollary 5. Re-randomizable one-way functions imply non-interactive IND-SO-COM commitments.

Since re-randomizable encryptions imply re-randomizable one-way functions, we have

Corollary 6. Re-randomizable encryption implies non-interactive IND-SO-COM secure commitments.

Perhaps more interesting is the case when $b_0 \neq b_1$. The commitment scheme constructed in Appendix A.2 is no longer perfectly hiding, so that Theorem 6 doesn't apply. In this case, we can still achieve IND-SO-COM security by using the indistinguishability of the two modes. Roughly, this follows because an IND-SO-COM adversary must have similar probabilities of success against both modes, otherwise it could be used to distinguish the modes. We then obtain the following Corollary.

Corollary 7. Re-randomizable one-way functions imply perfectly-binding IND-SO-COM commitments.

Since re-randomizable encryptions imply re-randomizable one-way functions, we have

Corollary 8. Re-randomizable encryption implies perfectly binding non-interactive IND-SO-COM secure commitments.

Proof. The proof uses an equivalent definition of IND-SO-COM security where the adversary $\mathcal{A}$ is presented with a challenger that either plays the real game or the ideal one depending on the value of a secret bit, which $\mathcal{A}$ aims to guess.

Towards a contradiction, suppose there exists an IND-SO-COM adversary $\mathcal{A}$ that succeeds against the protocol with probability $\frac{1}{2} + \epsilon$ when $b_0 = b_1$. We will use $\mathcal{A}$ to construct a distinguisher D for the one-way game against the underlying re-randomizable one-way function f. In the one-wayness game against f, the challenger samples a function f and sends it to D. D will respond by sending $\{0, 1\}$ to the one-wayness challenger and the latter samples $r \leftarrow R$ and sends $e = f(b, r)$ to D. Now, D samples $r' \leftarrow R$ and generates $e' = f(0, r')$. Then, D instantiates the commitment protocol by setting $c_0 = e, c_1 = e'$ and plays the IND-SO-COM game with the adversary $\mathcal{A}$. If $\mathcal{A}$ wins, D guesses $b = 1$ whereas, if $\mathcal{A}$ loses, D bets that $b = 0$. From Theorem 6, we know that, if $b = 0$, then $\mathcal{A}$ succeeds with advantage ν for some negligible function ν. On the other hand, by hypothesis, if $b = 1$, $\mathcal{A}$ wins the IND-SO-COM game with advantage ϵ. Now, it comes that

$$\begin{aligned}\Pr[D \text{ wins }] &= \Pr[b = 1 \cap \ \mathcal{A} \text{ wins }] + \Pr[b = 0 \cap \ \mathcal{A} \text{ loses }] \\ &= \Pr[\mathcal{A} \text{ wins}|b = 1]\Pr[b = 1] + \Pr[\mathcal{A} \text{ loses}|b = 0]\Pr[b = 0] \\ &= \frac{1}{2}\left(\frac{1}{2} + \epsilon + \frac{1}{2} - \nu\right) = \frac{1}{2} + \frac{\epsilon - \nu}{2}.\end{aligned}$$

Since ϵ is non-negligible and ν is negligible, D breaks the one-way property of f. □

We note that these constructions require trusted setup, which is necessary given the results of [BHY09], which showed a black-box separation between any primitive with a game-based definition of security and perfectly binding IND-SO-COM secure commitments without trusted setup.

B Homomorphic Encryption

A public key cryptosystem given by algorithms (G, E, D) is called *homomorphic* if

- The plaintext space forms a group X, with group operation $+$.
- The ciphertexts are members of a group Y.
- For all $x_0, x_1 \in X$, and for all $r_0, r_1 \in \mathsf{coins}(E)$, there exists an $r^* \in \mathsf{coins}(E)$ such that

$$E(pk, x_0 + x_1, r^*) = E(pk, x_0, r_0)E(pk, x_1, r_1).$$

Notice that we do not assume that the encryption is also homomorphic over the randomness, as is the case of most homomorphic encryption schemes, e.g. Elgamal, Paillier, and Goldwasser-Micali. We also do not assume that the image $E(pk, X, R)$ is the whole group Y, only that $E(pk, X, R) \subset Y$. Since the homomorphic property implies closure, we have that $E(pk, X, R)$ is a semi-group. Notice also, that while it is common to use the word "homomorphic" to describe the cryptosystem, encryption is *not* a homomorphism in the mathematical sense (although decryption is).

We now show some basic properties from all homomorphic encryption schemes. These facts are commonly used but, since our definition is weaker than the (implicit) definitions of homomorphic encryption that appear in the literature, it is important to note that they hold under this definition as well.

- $E(pk, X, R)$ is a group.
- $E(pk, 0, R)$ is a subgroup of $E(pk, X, R)$.
- For all $x \in X$, $E(pk, x, R)$ is the coset $E(pk, x, r)E(pk, 0, R)$.
- For all $x_0, x_1 \in X$, $|E(pk, x_0, R)| = |E(pk, x_1, R)|$.
- If y is chosen uniformly from $E(pk, 0, R)$, then $yE(pk, x, r)$ is uniform in $E(pk, x, R)$.

- $E(pk, X, R)$ is such that $E(pk, X, R) \simeq X \times E(pk, 0, R)$ and decryption is the homomorphism

$$E(pk, X, R) \to E(pk, X, R)/E(pk, 0, R) \simeq X.$$

We call a public key cryptosystem a *homomorphic public key encryption scheme*, if it is IND-CPA secure and homomorphic.

If we make the additional assumption that we can sample in a manner statistically close to uniform in the subgroup $E(pk, 0, R)$, then the homomorphic cryptosystem (G, E, D) will be re-randomizable.

Definition 7. A homomorphic encryption scheme is said *uniformly sampleable* if there is a PPT algorithm sample such that the output of $\mathsf{sample}(pk)$ is statistically close to uniform on the group $E(pk, 0, R)$.

We note that, for all known homomorphic cryptosystems, we may define

$$\mathsf{sample}(pk) = \{r \leftarrow \mathsf{coins}(E) : E(pk, 0, r)\}.$$

It is not hard to see that this property *does not* automatically follow from the definition of homomorphic encryption. Since all known homomorphic schemes satisfy it however, they are re-randomizable.

B.1 Efficient Re-Randomizable Encryption from Uniformly Sampleable Homomorphic Encryption

```
Parameter Generation:
  $(pk, sk) \leftarrow G(1^\lambda)$
  $r \leftarrow \mathsf{coins}(E)$
  $c = E(pk, b, r)$
  The public parameters are $(pk, c)$

Encryption:
  $r' \leftarrow \mathsf{coins}(\mathsf{sample})$
  $c' \leftarrow \mathsf{sample}(pk, r')$
  return $c^a \cdot c'$
Decryption:
  To decrypt a ciphertext $c$,
  simply return $D(c)$.
```

Figure 6: Lossy Encryption from uniformly sampleable homomorphic encryption

The scheme of section 3.1 only allows encrypting single bits. If the underlying cryptosystem (G, E, D) can encrypt more than one bit at a time, we can increase the efficiency of this system, by simply putting $c_0, c_1, \ldots, c_n$ into the public key, and an encryption of i will be $\mathsf{ReRand}(pk, c_i, r)$. In most cases, however, we can increase the size of encrypted messages without lengthening the public-key.

In particular, if $(G, E, D, \mathsf{sample})$ is a uniformly sampleable homomorphic encryption scheme and $\mathbb{Z}_N \hookrightarrow X$. Then, we can encrypt elements of $\{0, 1, \ldots, N-1\}$ instead of $\{0, 1\}$ as showed by figure 6.

If $c = E(pk, 0, r)$, the scheme is lossy since all encryptions will be uniformly distributed in the subgroup $E(pk, 0, R)$. In contrast, if $c = E(pk, 1, r)$, the scheme is injective by the correctness of the decryption algorithm. This is the natural construction when working with the Paillier or Damgård-Jurik cryptosystems. We must use caution when applying this construction to Elgamal since the inverse map $\mathbb{Z}_N \hookrightarrow X$ is not efficiently computable (it is the discrete log). In the context of commitments, it will not be a problem. On the other hand, when we want to view this as an encryption scheme for multi-bit messages, the lack of efficient inversion is an issue. Fortunately, a simple variant of Elgamal [NP01, PVW08, BHY09] is known to provide lossy encryptions from the DDH assumption. It is noteworthy that the "plain" Elgamal is itself re-randomizable although it is slightly less efficient than this modification.

C Simulation-Based Security

While we have mostly focused on an indistinguishability-based notion of security so far, Bellare *et al.* [BHY09] also formalized a simulation-based notion of security under selective openings. Their simulation-based definition of security intuitively seems stronger than the indistinguishability-based definition even though it still remains unknown whether SEM-SO-ENC implies IND-SO-ENC.

Definition 8. (Semantic Security under selective openings). A public key cryptosystem (G, E, D) is *simulatable under selective openings* (SEM-SO-ENC secure) if, for any PPT n-message sampler $\mathcal{M}$, any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and any poly-time computable relation $\mathcal{R}$, there is an efficient simulator $S = (S_1, S_2)$ s.t.

$$\left|\Pr\left[\mathcal{A}^{\mathsf{sem\text{-}so\text{-}real}} = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{sem\text{-}so\text{-}ideal}} = 1\right]\right| < \nu$$

for some negligible function ν, and where the games $\mathsf{sem\text{-}so\text{-}real}$ and $\mathsf{sem\text{-}so\text{-}ideal}$ are defined as follows

```
SEM-SO-ENC (Real):
  $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$
  $r_1, \ldots, r_n \leftarrow \mathsf{coins}(E)$
  $(I, st) \leftarrow \mathcal{A}_1\big(pk, E(m_1, r_i), \ldots, E(m_n, r_n)\big)$
  $w \leftarrow \mathcal{A}_2\big(st, (m_i, r_i)_{i \in I}\big)$
  Output $\mathcal{R}(\mathbf{m}, w)$

SEM-SO-ENC (Ideal):
  $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$
  $(I, st) \leftarrow S_1(1^\lambda)$
  $w \leftarrow S_2\big(st, \{m_i\}_{i \in I}\big)$
  Output $\mathcal{R}(\mathbf{m}, w)$
```

Figure 7: SEM-SO-ENC Security

In the real game, the challenger samples $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$ from the joint message distribution and picks random coins $r_1, \ldots, r_n \leftarrow \mathsf{coins}(E)$ to compute $E(m_1, r_1), \ldots, E(m_n, r_n)$ which are given to the adversary $\mathcal{A}$. The latter responds by choosing a $n/2$-subset $I \subset \{1, \ldots, n\}$ and gets back $\{(m_i, r_i)\}_{i \in I}$. The game ends with $\mathcal{A}$ outputting a string w and the value of the game is defined to be $\mathcal{R}(\mathbf{m}, w)$.

In the ideal game, the challenger samples messages $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$ from the joint message distribution. Without seeing any encryptions, the simulator chooses a subset I and some state information st. After having seen the messages $\{m_i\}_{i \in I}$ and the state information but without seeing any randomness, the simulator outputs a string w. The result of the game is $\mathcal{R}(\mathbf{m}, w)$.

In essence, simulation-based security demands that an efficient simulator be able to perform about as well as the adversary without having seen the challenge ciphertexts, the random coins or the public key.

In [BHY09], Bellare, Hofheinz and Yilek proved that any lossy encryption scheme endowed with an *efficient* opener procedure on lossy keys *is* SEM-SO-ENC secure.

Definition 9. A *lossy public-key encryption scheme with efficient opening* is a tuple $(G_{\mathsf{inj}}, G_{\mathsf{lossy}}, E, D)$ satisfying Definition 2, with the additional property that the algorithm opener is efficient, i.e.

- *Openability.* There is an *efficient* algorithm opener such that, if $(pk_{\mathsf{lossy}}, sk_{\mathsf{lossy}}) \leftarrow G_{\mathsf{lossy}}$, for all plaintexts $x_0, x_1 \in X$ and all $r \in \mathsf{coins}(E)$, with all but negligible probability, it holds that $E(pk_{\mathsf{lossy}}, x_0, r) = E(pk_{\mathsf{lossy}}, x_1, r')$, where $r' \leftarrow \mathsf{opener}(pk_{\mathsf{lossy}}, x_1, E(pk_{\mathsf{lossy}}, x_0, r))$.

Theorem 7. [BHY09] Lossy Encryption with efficient opening is SEM-SO-ENC secure.

Proof. This is Theorem 2 in [BHY09].
The proof is straightforward, and we only sketch it here.

We proceed in a series of games.

- Game$_0$ is the real SEM-SO-ENC experiment.
- Game$_1$ is the same as Game$_0$ but the adversary is given a lossy public key instead of a real one.
- Game$_2$ instead of giving the adversary the real randomness $\{r_i\}_{i \in I}$, the challenger uses the efficient opener procedure to generate valid randomness.
- Game$_3$ instead of giving the adversary encryptions of m_i, the adversary is given encryptions of a dummy message ξ, but the adversary is still given openings to actual messages $\{m_i\}_{i \in I}$ obtained from the opener procedure.

Now, the simulator can simulate Game$_3$ with the adversary. The simulator generates a lossy key pair, and encrypts a sequence of dummy messages and forwards the encryptions to $\mathcal{A}$. The adversary, $\mathcal{A}$, replies with a set I, which S forwards to the challenger. Then S uses the efficient opener procedure to open the selected messages for $\mathcal{A}$. At which point $\mathcal{A}$ outputs a string w, and S outputs the same string. Since the outputs of $\mathcal{A}$ in Game$_0$ and Game$_3$ are computationally close, the outputs of S, and $\mathcal{A}$ in the real and ideal experiments will also be computationally close. □

C.1 Selective Opening Security from the Composite Residuosity Assumption

Here, we discuss the application of construction of section B.1 to Paillier's cryptosystem (a review of the details of the Paillier cryptosystem can be found in Appendix F).

By defining $\mathsf{ReRand}(c, r) = c \cdot E(pk, 0, r) \mod N^2$, we easily obtain a bandwidth-efficient IND-SO-ENC secure encryption scheme via our general construction in section B.1. It was already known how to obtain IND-SO-ENC security from the DCR assumption since Rosen and Segev [RS09] and Boldyreva, Fehr and O'Neill [BFO08] showed how to build lossy-trapdoor functions using Composite Residuosity and lossy TDFs imply IND-SO secure encryption [BHY09]. By applying our construction to Paillier, we obtain a simpler and significantly more efficient construction than those following from [BFO08, RS09] under the same assumption.

While the results of [BHY09] imply that IND-SO-ENC secure encryptions follow from DCR, the question of SEM-SO-ENC secure encryptions was left open. The only previous construction of SEM-SO-ENC secure encryption was given in [BHY09] under the Quadratic Residuosity assumption (QR). From the Paillier and Damgård-Jurik cryptosystems, we readily obtain a lossy encryption scheme where the function opener is efficient. The results of [BY09, BHY09] then imply that the resulting encryption scheme achieves SEM-SO-ENC security.

To see that Paillier allows for efficient opening, recall that $E(pk, m, r) = g^m r^N \mod N^2$, where, in lossy mode, g is an N^{th} power (in which case, all ciphertexts are encryptions of 0) whereas its order is a multiple of N in injective mode. Then, any lossy ciphertext $c = E(pk, m, r)$ can be expressed as $c = r_1^N \mod N^2$ for some $r_1 \in \mathbb{Z}_N$, which the opener can compute as $r_1 = (c \bmod N)^{1/N} \mod N$ (recall that $\gcd(N, \phi(N)) = 1$) using the factorization of N and $d = N^{-1} \mod \phi(N)$. Since g is itself a N^{th} residue in $\mathbb{Z}_{N^2}$, it can compute $g_0 \in \mathbb{Z}_N$ such that $g = g_0^N \mod N^2$ in the same way. To open c to $m \in \mathbb{Z}_N$, it has to find $r' \in \mathbb{Z}_N^*$ such that $r_1^N = g_0^{mN} r'^N \mod N^2$, which is easily obtained as $r' = r_1 g_0^{-m} \mod N$.

So, the efficiency of opener reduces to the efficiency of taking N^{th} roots modulo N, which is efficiently feasible

if the factorization of N is known. Hence, we immediately obtain a simple and efficient SEM-SO-ENC secure encryption system from the DCR assumption. We note that the possible use of Paillier as a lossy encryption scheme was implicitly mentioned in [YY05] but, to the best of our knowledge, its efficient openability property was never reported so far.

Corollary 9. Under the DCR assumption, Paillier's cryptosystem is SEM-SO-ENC secure.

Since Paillier's cryptosystem (in the same way as the Damgård-Jurik extension) has smaller ciphertext expansion than the Goldwasser-Micali cryptosystem, we end up with a more efficient system than the only currently known SEM-SO-ENC secure cryptosystem.

D Lossy Encryption from Smooth Universal Hash Proof Systems

We recall the notion of a *smooth projective hash family* [CS02]. Let H be a hash family with keys in the set K, *i.e.* for each $k \in K$, $H_k : X \to \Pi$. Let $L \subset X$ and $\alpha : K \to S$. We require efficient evaluation algorithms such that, for any $x \in X$, $H_k(x)$ is efficiently computable using $k \in K$. Additionally, if $x \in L$ and a witness w for $x \in L$ is known, then $H_k(x)$ is efficiently computable given $x, w, \alpha(k)$.

Definition 10. The set $(H, K, X, L, \Pi, S, \alpha)$ is a projective hash family if, for all $k \in K$, the action of H_k on the subset L is completely determined by $\alpha(k)$.

While $\alpha(k)$ determines the output of H_k on L, we need to ensure that it does not encode "too much" information on k. This is captured by the following definition of *smooth* projective hash family.

Definition 11. Let $(H, K, X, L, \Pi, S, \alpha)$ be a projective hash family, and define two distributions Z_1, Z_2 taking values on the set $X \setminus L \times S \times \Pi$. For Z_1, we sample $k \xleftarrow{\$} K$, $x \xleftarrow{\$} X \setminus L$, and set $s = \alpha(k)$, $\pi = H_k(x)$, for Z_2 we sample $k \xleftarrow{\$} K$, $x \xleftarrow{\$} X \setminus L$, and $\pi \xleftarrow{\$} \Pi$, and set $s = \alpha(k)$. The projective hash family is called *ν-smooth* if $\Delta(Z_1, Z_2) < \nu$.

The above basically says that, given $\alpha(k)$ and $x \in X \setminus L$, $H_k(x)$ is statistically close to uniform on Π.

Let $(H, K, X, L, \Pi, S, \alpha)$ be an ν-smooth projective hash family for some negligible function ν. We show a natural construction of Lossy Encryption. While smooth hash proof systems have a natural lossiness property, the constructions of IND-CPA secure encryption from [CS02] are not lossy encryption systems. The schemes described by Cramer and Shoup have two indistinguishable types of ciphertexts: "good" ciphertexts are generated in L while "bad" ciphertexts are sampled from $X \setminus L$. By turning their construction around, we can use their ciphertexts (in the IND-CCA1 version of their schemes) as public keys and their public keys as our ciphertexts to get a construction of Lossy Encryption.

- **Injective key generation:** Sample an element $x \in L$, along with the corresponding witness w.
  Set $PK = x$, $SK = w$.
- **Lossy key generation:** Sample an $x \in X \setminus L$. Set $PK = x$, $SK = \perp$.
- **Encryption:** To encrypt a message $m \in \Pi$, pick $k \overset{\$}{\leftarrow} K$, and output $c = (\alpha(k), H_k(x) + m)$, where $H_k(x)$ is efficiently computable without the witness w because k is known.
- **Decryption:** Given a ciphertext $c = (\alpha(k), \pi)$, use the witness w and $\alpha(k)$ to compute $H_k(x)$. Output $m = \pi - H_k(x)$.

The correctness of decryption follows immediately from the definitions and the indistinguishability of modes follows immediately from the hardness of the subset decision problem $L \subset X$. It only remains to see that, in lossy mode, the ciphertext is statistically independent of the plaintext m. But this follows immediately from the ν-smoothness of the hash proof system. Thus we arrive at

Lemma 7. The scheme outlined above is a Lossy Encryption scheme.

The DDH-based lossy cryptosystem of [KN08, BY09, BHY09] is easily seen to be a particular case of this construction. Given public parameters $(g, h) \in \mathbb{G}$ for a group $\mathbb{G}$ of prime order p, we define $X = \mathbb{G}^2$ and L as the language $L = \{(Y_1, Y_2) = (g^y, h^y) : y \in \mathbb{Z}_p\}$, so that $w = y$ serves as a witness for the membership in L. We also define k to be a random pair $(r, s) \in (\mathbb{Z}_p)^2$ and $\alpha(k) = g^r \cdot h^s$ in such a way that $H_k((Y_1, Y_2)) = Y_1^r \cdot Y_2^s$ is easily computable using (r, s) and independent of $\alpha(k)$ when $(Y_1, Y_2) \notin L$.

Other known projective hash functions (e.g., [CS02]) immediately suggest new lossy encryption systems based on the Composite and Quadratic Residuosity assumptions that differ from currently known schemes. Yet another realization can be readily obtained from the Decision Linear assumption [BBS04], which is believed to be weaker than DDH.

E Chosen-Ciphertext Security: Simulatability

The simulation-based definition of [BY09, BHY09] also extends to the chosen-ciphertext scenario and involves an efficiently computable relation $\mathcal{R}$.

- **Selective opening query:** let $\mathcal{M}$ be a message distribution. The challenger samples a n-vector $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$ and generates

  $$(c_1, \ldots, c_n) = (E(pk, m_1, r_1), \ldots, E(pk, m_n, r_n)),$$

  which are sent the adversary. We call $c_1, \ldots, c_n$ the *target ciphertexts*.
- **Corruption query:** the adversary chooses a subset $I \subset [n]$ of cardinality $\#I = n/2$ and sends I to the challenger. The challenger then sends $\{(m_i, r_i)\}_{i \in I}$ to the Adversary.

  The challenger then sends $\{m_j\}_{j \notin I}$ to the adversary.

- **Decryption queries:** the adversary $\mathcal{A}$ chooses a ciphertext c that has never appeared as a target ciphertext, and sends c to the challenger. If c is a valid ciphertext (*i.e.*, $D(c) \neq \perp$) then the challenger responds with $m = D(c)$.

After adaptively making polynomially many queries, with at most one of them being a selective opening query, the adversary outputs w, and the value of the game is $\mathcal{R}(\mathbf{m}, w)$.

In the ideal game, the challenger samples $\mathbf{m} = (m_1, \ldots, m_n) \leftarrow \mathcal{M}$.

- The simulator chooses a subset, $I \leftarrow S_1$.
- The simulator views the chosen messages and outputs a w, $w \leftarrow S_2(\{m_i\}_{i \in I})$.

The value of the game is $\mathcal{R}(\mathbf{m}, w)$.

Definition 12. (SEM-SO-CCA2) A public key cryptosystem (G, E, D) is SEM-SO-CCA2 secure if, for any PPT message distribution $\mathcal{M}$, any PPT relations $\mathcal{R}$ any PPT adversary $\mathcal{A}$, there is a simulator $S = (S_1, S_2)$ s.t. the outcome of real and ideal games are identical with all but negligible probability, *i.e.*,

$$\Pr[\mathsf{sem\text{-}cca2\text{-}real} \neq \mathsf{sem\text{-}cca2\text{-}ideal}] \leq \nu.$$

For some negligible function ν.

The notion of SEM-SO-CCA1 security is defined by means of similar experiments, but no decryption query is allowed after the selective opening query in the real game.

Similarly to the indistinguishability case, we remark that, if the adversary is not allowed to make decryption queries at all, this notion reduces to SEM-SO-ENC security.

E.1 Unduplicatable Set Selection

Unduplicatable set selection was used implicitly in [NY90] and [CIO98], and formalized in [Sah99]. The description below is essentially that of [Sah99].

The goal of unduplicatable set selection is to create a mapping from $\mathfrak{g} : \{0,1\}^k \to B$ such that, for all distinct $a^1, \ldots, a^n, a^{n+1} \in \{0,1\}^k$,

$$\mathfrak{g}(a^{n+1}) \not\subset \bigcup_{i=1}^{n} \mathfrak{g}(a^i).$$

In [Sah99], Sahai gives a simple construction based on polynomials which we recall here. Let $\ell = 2^{\lceil \log_2 2nk \rceil}$, so $\ell > 2nk$, and let $Y = \mathbb{F}_\ell \times \mathbb{F}_\ell$, and $B \subset \mathcal{P}(Y)$. To each $a \in \{0,1\}^k$, we may associate a polynomial

$$f_a(x) = a_0 + a_1 x + \cdots a_{k-1} x^{k-1} \in \mathbb{F}_\ell[x].$$

Then, if we set

$$\mathfrak{g}(a) = \{(t, f_a(t)) : t \in \mathbb{F}_\ell\} \subset Y,$$

we have $|\mathfrak{g}(a)| = \ell$ and, if $a \neq a'$, it holds that $|\mathfrak{g}(a) \cap \mathfrak{g}(a')| \leq k - 1$. Thus,

$$\begin{aligned} \left| \mathfrak{g}(a^{n+1}) \setminus \bigcup_{i=1}^{n} \mathfrak{g}(a^i) \right| &= \left| \mathfrak{g}(a^{n+1}) \setminus \bigcup_{i=1}^{n} \mathfrak{g}(a^{n+1}) \cap \mathfrak{g}(a^i) \right| \\ &\geq \left| \mathfrak{g}(a^{n+1}) \right| - \sum_{i=1}^{n} \left| \mathfrak{g}(a^{n+1}) \cap \mathfrak{g}(a^i) \right| \geq \ell - n(k-1) \geq \frac{\ell}{2}. \end{aligned}$$

We call $\mathfrak{g}$ an (n,k)-unduplicatable set selector.

E.2 Non-Interactive Zero-Knowledge

One of the most successful techniques for securing cryptosystems against chosen-ciphertext attacks has been the Naor-Yung paradigm [NY90]. Roughly said, the idea is to encrypt the message twice and include a non-interactive zero-knowledge (NIZK) proof that both encryptions encrypt the same plaintext. The proof of security then uses the NIZK simulator to simulate the proof for the challenge ciphertext. This method has since been refined in [DDN91, Sah99, SCO$^+$01, Lin06] (among others).

Our construction of SEM-SO-CCA1 encryption follows the general Naor-Yung paradigm [NY90]. However, the selective opening of the encryption query poses new challenges. In particular, if we naively try to apply the Naor-Yung technique, we immediately encounter difficulties because our challenger must reveal the messages and randomness for half of the ciphertexts in the challenge. This will immediately reveal to the adversary that the proofs were simulated. It requires new ideas to overcome this difficulty.

We now give a brief definition of the properties of a non-interactive zero-knowledge proof of knowledge with honest-prover state reconstruction (originally defined and constructed in [GOS06]).

Let $\mathcal{R}$ be an efficiently computable binary relation and let $L = \{x : \exists w \text{ such that } (x, w) \in \mathcal{R}\}$. We refer to L as a language, x as a statement, and w as a witness. A non-interactive proof system for L is a triple of PPT algorithms $(\mathsf{CRSgen}, \mathsf{Prover}, \mathsf{Verifier})$ such that

- $\sigma \leftarrow \mathsf{CRSgen}(1^\lambda)$: generates a common reference string σ.
- $\pi \leftarrow \mathsf{Prover}(\sigma, x, w)$: given x and a witness w for x s.t. $\mathcal{R}(x, w) = 1$, the Prover outputs a proof π.
- $b \leftarrow \mathsf{Verifier}(\sigma, x, \pi)$: on inputs x and a purported proof π, $\mathsf{Verifier}$ outputs a bit $b \in \{0, 1\}$.

Definition 13. A triple $(\mathsf{CRSgen}, \mathsf{Prover}, \mathsf{Verifier})$ is called a non-interactive zero-knowledge (NIZK) proof of knowledge with honest-prover state reconstruction if it satisfies the following properties

- **Completeness:** For all adversaries $\mathcal{A}$, there exists a negligible function ν such that

  $$\Pr\Big[\sigma \leftarrow \mathsf{CRSgen}(1^\lambda); (x, w) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow \mathsf{Prover}(\sigma, x, w) : \mathsf{Verifier}(\sigma, x, \pi) = 1 \text{ if } (x, w) \in \mathcal{R}\Big] > 1-\nu.$$

- **Soundness:** For all adversaries $\mathcal{A}$, there is a negligible function ν such that

  $$\Pr\Big[\sigma \leftarrow \mathsf{CRSgen}(1^\lambda); (x, \pi) \leftarrow \mathcal{A}(\sigma) : \mathsf{Verifier}(\sigma, x, \pi) = 0 \text{ if } x \notin L\Big] > 1 - \nu.$$

- **Knowledge Extraction:** There is an extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$ such that, for all adversaries $\mathcal{A}$,

  $$\Big|\Pr\Big[\sigma \leftarrow \mathsf{CRSgen}(1^\lambda) : \mathcal{A}(\sigma) = 1\Big] - \Pr\Big[(\sigma, \tau) \leftarrow \mathsf{Ext}_1(1^\lambda) : \mathcal{A}(\sigma) = 1\Big]\Big| < \nu,$$

  and

  $$\Pr\Big[(\sigma, \tau) \leftarrow \mathsf{Ext}_1(1^\lambda); (x, \pi) \leftarrow \mathcal{A}(\sigma); w \leftarrow \mathsf{Ext}_2(\sigma, \tau, x, \pi) : \mathsf{Verifier}(\sigma, x, \pi) = 0 \text{ or } (x, w) \in \mathcal{R}\Big] > 1-\nu$$

  For some negligible function ν.

- **Zero-Knowledge:** There exists a simulator $S = (S_1, S_2)$, such that for all adversaries $\mathcal{A}$,

$$\left|\Pr\left[\sigma \leftarrow \mathsf{CRSgen}(1^\lambda) : \mathcal{A}^{\mathsf{Prover}(\sigma,\cdot,\cdot)}(\sigma) = 1\right] - \Pr\left[(\sigma,\tau) \leftarrow S_1(1^\lambda) : \mathcal{A}^{S'(\sigma,\tau,\cdot,\cdot)}(\sigma) = 1\right]\right| < \nu,$$

where S' is defined

$$S' = \begin{cases} S_2(\sigma,\tau,x) & \text{if } (x,w) \in \mathcal{R}, \\ \perp & \text{otherwise.} \end{cases}$$

- **Honest-Prover State Reconstruction:** There exists a simulator $\mathsf{HSR} = (\mathsf{HSR}_1, \mathsf{HSR}_2, \mathsf{HSR}_3)$ such that for all adversaries $\mathcal{A}$

$$\left|\Pr\left[\sigma \leftarrow \mathsf{CRSgen}(1^\lambda); \mathcal{A}^{\mathsf{Prover}(\sigma,\cdot,\cdot)}(\sigma) = 1\right] - \Pr\left[(\sigma,\tau) \leftarrow \mathsf{HSR}_1(1^\lambda) : \mathcal{A}^{\mathsf{HSR}(\sigma,\tau,\cdot,\cdot)}(\sigma) = 1\right]\right| < \nu,$$

where $\mathsf{Prover}(\sigma, x, w)$ samples $r \leftarrow \mathsf{coins}(\mathsf{Prover})$, sets $\pi = \mathsf{Prover}(\sigma, x, w, r)$ and returns (π, r) whereas HSR samples $r^* \leftarrow \mathsf{coins}(\mathsf{HSR}_2)$, sets $\pi' = \mathsf{HSR}_2(\sigma, \tau, x, r^*)$ and finally HSR sets $r' \leftarrow \mathsf{HSR}_3(\sigma, \tau, x, w, r^*)$ and returns (π', r'). Both oracles output $\perp$ if $(x, w) \notin \mathcal{R}$.

E.3 A SEM-SO-CCA1 Construction Based on the Naor-Yung Paradigm

Along with NIZK proofs with honest-prover state reconstruction, our construction relies on a number of common cryptographic tools. We will also require a strongly unforgeable one-time signature scheme. In the SEM-SO-CCA1 game, a single encryption query is actually n separate encryptions and we will require an unduplicatable set selector $\mathfrak{g}$ for sets of size n (see Appendix E.1 for a description of unduplicatable set selectors). Finally, we will require a lossy encryption scheme with efficient opening.

While the construction outlined below uses a one-time signature scheme (as in [DDN91]), the signature scheme can be removed and replaced by a strictly combinatorial construction as in [NY90]. We note that, although our construction is similar to the IND-CCA2 construction of [DDN91], the proof of SEM-SO-CCA1 security *does not* extend to SEM-SO-CCA2 security because the adversary learns the signing keys used for half of the ciphertexts in the challenge query, which allows her to create arbitrary signatures corresponding to those verification keys. This appears to be a significant problem when trying to adapt many of the known IND-CCA2 constructions to the IND-SO-CCA2 or SEM-SO-CCA2 settings.

Let $\Pi^{so} = (G_{so}, E, D)$ be an efficiently openable (and thus SEM-SO-ENC secure) lossy cryptosystem. Let $(\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ be a strongly unforgeable one-time signature scheme where the public key space in contained in $\{0,1\}^\lambda$. Let $\mathfrak{g}$ be an (n, λ)-unduplicatable set selector and let $\ell = |\mathfrak{g}(0^\lambda)|$ and $L = \mathfrak{g}(\{0,1\}^\lambda)$.

Let $(\mathsf{CRSgen}, \mathsf{Prover}, \mathsf{Verifier})$ be a NIZK proof of knowledge with honest-prover state reconstruction for the language given by the relation $((e_0, e_1), (m, r_0, r_1)) \in \mathcal{R}$ if $e_0 = E(m, r_0)$ and $e_1 = E(m, r_1)$.

Our SEM-SO-CCA1 scheme works as follows.

- **KeyGen:** Generate two key pairs for Π^{so} and reference strings for the NIZK proof system

$$(pk_0, sk_0) \leftarrow G_{so}(1^\lambda),\ (pk_1, sk_1) \leftarrow G_{so}(1^\lambda),\ \text{and } \sigma_i \leftarrow \mathsf{CRSgen}(1^\lambda) \text{ for } i \in L.$$

Set $pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L})$ and $sk = (sk_0, sk_1)$.

- **Encryption:** Pick random coins

  $$r^{sig} \leftarrow \mathsf{coins}(\mathsf{Sign}),\ r_0 \leftarrow \mathsf{coins}(E),\ r_1 \leftarrow \mathsf{coins}(E),\ r_i^{nizk} \leftarrow \mathsf{coins}(\mathsf{Prover}) \text{ for } i = 1, \ldots, \ell.$$

  Generate keys $(vk, sk) = \mathsf{G}(r^{sig})$ for a one-time signature using randomness r^{sig}.

  To encrypt a message m, calculate

  $$e_0 = E(pk_0, m, r_0), \quad e_1 = E(pk_1, m, r_1).$$

  Using the witness $w = (m, r_0, r_1)$, generate NIZK proofs

  $$\overline{\pi} = (\pi_1, \ldots, \pi_\ell) = (\mathsf{Prover}(\sigma_i, (e_0, e_1), w))_{i \in \mathfrak{g}(vk)}$$

  using r_i^{nizk} in the i^{th} iteration of Prover. Generate a signature $\mathsf{sig} = \mathsf{Sign}(e_0, e_1, \overline{\pi})$ and output

  $$c = (vk, e_0, e_1, \overline{\pi}, \mathsf{sig}).$$

- **Decryption:** Given a ciphertext $c = (vk, e_0, e_1, \overline{\pi}, \mathsf{sig})$, check that $\mathsf{Ver}(vk, (e_0, e_1, \overline{\pi})) = 1$, and return $\perp$ otherwise. For each $i \in \mathfrak{g}(vk)$, check that $\mathsf{Verifier}(\sigma_i, (e_0, e_1), \pi_i) = 1$ and return $\perp$ otherwise. If all checks are successful, return $m = D(sk_0, e_0)$.

Theorem 8. This scheme is SEM-SO-CCA1 secure.

Proof. We will show how to use an adversary $\mathcal{A}$ in the $\mathsf{sem\text{-}cca1\text{-}real}$ game to construct a simulator for the $\mathsf{sem\text{-}cca1\text{-}ideal}$ game. To do this, we begin by considering a series of games.

- Game$_0$: is the actual $\mathsf{sem\text{-}cca1\text{-}real}$ game.
- Game$_1$: is as Game$_0$ but the verification keys $(vk^{chal,1}, sk^{chal,1}), \ldots, (vk^{chal,n}, sk^{chal,n})$ to be used in the challenge ciphertexts are chosen during the parameter generation phase. In addition, we raise a failure event F_1, which is the occurrence of a decryption query $(vk, e_0, e_1, \overline{\pi}, \mathsf{sig})$ such that $vk = vk^{chal,j}$ for some $j \in \{1, \ldots, n\}$.
- Game$_2$: is identical to Game$_1$ but the common reference strings are now generated as

  $$\sigma_i = \begin{cases} \sigma \leftarrow \mathsf{CRSgen}(1^\lambda) & \text{if } i \in \mathfrak{g}(vk^{chal,j}) \text{ for some } j \in [n] \\ \text{the first output of } (\sigma, \tau) \leftarrow \mathsf{Ext}_1(1^\lambda) & \text{otherwise.} \end{cases}$$

  In addition, to handle decryption queries $(vk, e_0, e_1, \overline{\pi}, \mathsf{sig})$, we now use any index $i \notin \mathfrak{g}(vk) \in \{1, \ldots, \ell\}$ to recover (m, r_0, r_1) from the proof π_i using the trapdoor τ_i of the extractable reference string σ_i. Such an index $i \in \{1, \ldots, \ell\}$ must exist since $\mathfrak{g}(vk) \not\subset \bigcup_{j=1}^n \mathfrak{g}(vk^{chal,j})$.
- Game$_3$ in this game, we switch both pk_0 and pk_1 to the lossy mode and proceed as in Game$_2$.
- Game$_4$: we now use the honest-prover state reconstruction simulator $\mathsf{HSR} = (\mathsf{HSR}_1, \mathsf{HSR}_2, \mathsf{HSR}_3)$. We first bring a new change to the generation of reference strings at the beginning of the game. Namely, for each $i \in L$ such that $i \in \mathfrak{g}(v^{chal,j})$, for some $j \in [n]$, we set $(\sigma_i, \tau_i) \leftarrow \mathsf{HSR}_1(1^\lambda)$. Also, in the generation of target ciphertexts, we ignore the witnesses and simulate the "proofs"

  $$\overline{\pi} = \{\pi_i\}_{i \in \mathfrak{g}(vk^{chal,j})} = \{\mathsf{HSR}_2(\sigma_i, \tau_i, (e_0, e_1), r_i^*)\}_{i \in \mathfrak{g}(vk^{chal,j})},$$

for each $i \in \{1, \ldots, \ell\}$, $j \in \{1, \ldots, n\}$. Also, when the adversary asks for the opening of a subset of the target ciphertexts, we use the honest-prover state reconstructor to generate

$$r_i \leftarrow \mathsf{HSR}_3(\sigma_i, \tau_i, (e_0, e_1), (m, r_0, r_1, r_i^*)),$$

and return these r_i (instead of the coins r_i^* that were actually used to simulate proofs).

- Game$_5$: in this game, the challenger generates all target ciphertexts as encryptions of a dummy message ξ. In addition, the choice of $\mathbf{m} \overset{\$}{\leftarrow} \mathcal{M}$ is postponed until the moment of the opening query. When $\mathcal{A}$ asks for the opening of a subset of the target ciphertexts, we use the efficient openability of (G_{so}, E, D) to generate $\{r_i\}_{i \in I}$ that explain $\mathbf{m}[I]$. Otherwise, the simulator proceeds as in Game$_4$.

Let W_i be the distribution of the adversary's output in game i. Clearly, W_0 is almost identical to W_1 since, given that $vk^{chal,1}, \ldots, vk^{chal,n}$ are independent of the adversary's view until the challenge phase, the failure event F_1 occurs with probability smaller than $qn\delta$ if q is the number of decryption queries and δ is the maximal probability for a given verification key to be generated by G. In other words, we only need the property that vk is unpredictable and we could use a simple combinatoric argument as in [NY90]. However, a one-time signature scheme clearly has this property as well.

To show that W_1 and W_2 are only negligibly different, notice that, by the unduplicatability of $\mathfrak{g}$, there will always be at least one valid proof generated with an extractable CRS. Hence, we will always be able to answer decryption queries. It comes that any significant difference between Game$_2$ and Game$_1$ would imply the ability of the adversary to break either the soundness or the knowledge extraction property of the proof system. By virtue of the latter's security, W_2 must be negligibly close to W_1.

Since the challenger never uses the decryption keys corresponding to pk_0 and pk_1 in Game$_2$ (instead the challenger decrypts with the knowledge extractor), the distributions W_2 and W_3 must be computationally indistinguishable. Otherwise, the challenger could distinguish injective keys from lossy keys in the underlying lossy encryption scheme (G_{so}, E, D).

Now, it is easy to see that any PPT adversary that can distinguish between Game$_3$ and Game$_4$ can be used to distinguish honestly generated proofs for the real CRS of Game$_3$ and the outputs of the honest-prover reconstruction simulator $(\mathsf{HSR}_1, \mathsf{HSR}_2, \mathsf{HSR}_3)$ (really $n\ell$ such simulators) in Game$_4$. Such an adversary indeed breaks the indistinguishability of the honest-prover state reconstruction simulator, losing a factor of $n\ell$ (because we are making $n\ell$ comparisons).

Finally, we also note that, for each challenge ciphertext, HSR_2 generates proofs without using witnesses and, since pk_0 and pk_1 are both lossy keys, each challenge ciphertext is statistically independent of the plaintext. Moreover, since Π^{so} allows for efficient opening under lossy keys, the challenger can open any such ciphertext to any desired plaintext without affecting $\mathcal{A}$'s view. It comes that the statistical distance between W_5 and W_4 is negligible.

Thus, we have shown that, for any efficient adversary $\mathcal{A}$, the value of Game$_0$ will be computationally indistinguishable from the value of Game$_5$. Now, we show how to use the adversary of Game$_5$ to build a simulator for the sem-cca1-ideal game.

Specifically, the simulator runs $\mathcal{A}$ internally exactly as Game$_5$ does. In particular, it generates lossy keys pk_0, pk_1 and reference strings on its own and answers decryption queries as in Game$_2$-Game$_5$. When $\mathcal{A}$ asks for a subset I, the simulator asks for openings of the same subset I. Using $\{m_i\}_{i \in I}$, the simulator runs the efficient opening procedure of (G_{so}, E, D) to generate $\{r_i\}_{i \in I}$. As in Game$_5$, the simulator then uses the state reconstructor HSR_3 to generate randomness that look like an honest prover's random coins for the witnesses $\{(m_i, r_i)\}_{i \in I}$. Finally, when $\mathcal{A}$ outputs w, the simulator outputs the same w. Since $\mathcal{A}$'s output in Game$_5$ is indistinguishable from her output in the sem-cca1-real game, the output of the simulator will be indistinguishable from $\mathcal{A}$'s output in the sem-cca1-real game.

□

A similar argument shows that this construction will be IND-SO-CCA1 if the underlying encryption scheme is IND-SO-ENC instead of SEM-SO-ENC secure.

Notice, however, that if we consider the SEM-SO-CCA2 game, then Game$_1$ and Game$_2$ are distinguishable. This is because when an adversary gets an opening of one of the challenge ciphertexts, she also receives the secret key of the one-time signature used on that message. She can thus sign any message using that verification key. This is the primary stumbling block when trying to build SEM-SO-CCA2 (or IND-SO-CCA2) encryptions using one-time signature schemes.

F The Paillier Cryptosystem

We briefly review the Paillier cryptosystem [Pai99] that was extended by Damgård and Jurik [DJ01]. The cryptosystem works over $\mathbb{Z}^*_{N^2}$. From the Binomial Theorem, we have

$$(1+N)^a = 1 + aN \mod N^2,$$

so $(1+N)$ generates a cyclic subgroup of order N. In this group, we can compute "partial" discrete logarithms efficiently by $L(x) = \frac{x-1}{N}$, since $L((1+N)^a) = L(1+aN) = a$. Now, if g generates $\langle 1+N \rangle$ and $c = g^a \bmod N^2$, we have $a = L(c)L(g)^{-1} \bmod N$.

- **Parameter Generation:**
  - Generate primes p, q of length $\lambda/2$ and sets $N = pq$.
  - Generate $g \in \mathbb{Z}^*_{N^2}$ such that N divides the order of g.
    This condition is easy to verify if you have the factorization of N.

  The public parameters are $pk = (N, g)$. The secret key is $sk = \operatorname{lcm}(p-1, q-1)$.

- **Encryption:** to encrypt $m \in \mathbb{Z}_N$, chooose $r \overset{\$}{\leftarrow} \mathbb{Z}^*_N$ (r is actually drawn in $\mathbb{Z}_N$, but the distributions are statistically close) and compute $c = E(pk, m, r) = g^m r^N \mod N^2$.

- **Decryption:** given a ciphertext $c \in \mathbb{Z}^*_{N^2}$,

$$m = \frac{L(c^{sk} \bmod N^2)}{L(g^{sk} \bmod N^2)} \mod N.$$

This cryptosystem is IND-CPA secure under the Decisional Composite Residuosity assumption (DCR), which (informally) says the following.

Assumption 1. Decisional Composite Residuosity (DCR): If $N = pq$ is an λ-bit RSA modulus,

$$\{g \leftarrow \mathbb{Z}^*_{N^2} :\ g\} \approx_c \{g \leftarrow \mathbb{Z}^*_{N^2} :\ g^N\}.$$