A Black-Box Construction of Non-Malleable Encryption from Semantically Secure Encryption[*]

Seung Geol Choi[†] Dana Dachman-Soled[‡] Tal Malkin[§] Hoeteck Wee[¶]

Abstract

We show how to transform any semantically secure encryption scheme into a non-malleable one, with a black-box construction that achieves a quasi-linear blow-up in the size of the ciphertext. This improves upon the previous non-black-box construction of Pass, Shelat and Vaikuntanathan (Crypto '06). Our construction also extends readily to guarantee non-malleability under a bounded-CCA2 attack, thereby simultaneously improving on both results in the work of Cramer et al. (Asiacrypt '07).

Our construction departs from the oft-used paradigm of re-encrypting the same message with different keys and then proving consistency of encryption. Instead, we encrypt an encoding of the message; the encoding is based on an error-correcting code with certain properties of reconstruction and secrecy from partial views, satisfied, e.g., by a Reed-Solomon code.

1 Introduction

The most basic security requirement for public key encryption (PKE) schemes is that an eavesdropping adversary does not learn anything about the plaintext underlying a communicated ciphertext (equivalently, cannot distinguish an encryption of one plaintext from another). This notion is denoted IND-CPA (for indistinguishability under chosen plaintext attack), and is often referred to as "semantic security" [GM84]. In many applications, however, this indistinguishability guarantee is not sufficient, and a PKE satisfying the stronger notion of *non-malleability* [DDN00] is required. Roughly, non-malleability requires that it is infeasible for an adversary to modify a ciphertext into one, or many, other ciphertexts of messages related to the original plaintext. As one example for the importance of non-malleability, consider the use of PKE in auctions. Buyers place their bids for an item to a seller, encrypted under the seller's public key, and the seller sells the item to the buyer with the highest bid. We certainly want to rule out adversaries who consistently bid exactly one dollar more than the previous bidders. This motivates the following question.

> Is it possible to *immunize* any semantically secure encryption scheme, transforming it into a scheme that is non-malleable?

[*] An extended abstract [CDMW08] appeared in TCC 2008 under the title "Black-Box Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One". The work was done while all the authors were at Columbia University, supported in part by NSF Grants CNS-0716245, CCF-0347839, and SBE-0245014, and by Samsung Scholarship.

[†] US Naval Academy, choi@usna.edu

[‡] University of Maryland, danadach@ece.umd.edu

[§] Columbia University, tal@cs.columbia.edu

[¶] CNRS-DIENS, École Normale Supérieure, wee@di.ens.fr

We focus on this question for a passive adversary, and when we refer to "non-malleable encryption" we mean, by default, non-malleability under chosen plaintext attack (NM-CPA). Later, we will also discuss the implications of our results to the active case, where the adversary can mount a limited chosen ciphertext attack.

Prior to our work, Pass, Shelat, and Vaikuntanathan [PSV06] studied this question and answered it affirmatively, providing a beautiful construction of a non-malleable encryption scheme from any semantically secure one (building on [DDN00]). However, this PSV construction – as with previous constructions achieving non-malleability from general assumptions [DDN00, Sah99, Lin06] – suffers from the curse of inefficiency arising from the use of general NP-reductions. In this paper we overcome this problem and answer the above question affirmatively using a *black-box* reduction. Before explaining our results, we provide some background and motivation.

Black-Box Complexity of Cryptographic Primitives. Much of the modern work in foundations of cryptography rests on general cryptographic assumptions like the existence of one-way functions and trapdoor permutations. General assumptions provide an abstraction of the functionalities and hardness we exploit in specific assumptions such as hardness of factoring and discrete log without referring to any specific underlying algebraic structure. Constructions based on general assumptions may use the primitive guaranteed by the assumption in one of two ways:

- *Black-box usage.* A construction is black-box if it refers only to the input/output behavior of the underlying primitive; we would typically also require that in the proof of security, we can use an adversary breaking the security of the construction as an oracle to break the underlying primitive (this is called a fully black-box reduction – see [RTV04] and references within for more details).

- *Non-black-box usage.* A construction is non-black-box if it uses the code computing the functionality of the primitive.

Motivated by the fact that the majority of constructions in cryptography are black-box, a rich and fruitful body of work initiated in [IR89] seeks to understand the power and limitations of black-box constructions in cryptography, resulting in a fairly complete picture of the relations amongst many cryptographic primitives with respect to black-box constructions. Recent work (including this paper), has turned to tasks for which the only constructions we have are non-black-box, yet the existence of a black-box construction is not ruled out. A notable example is general secure multi-party computation against a dishonest majority, for which the recent works of [IKLP06, Hai08] show a black-box construction from the minimal primitive of semi-honest oblivious transfer. Other examples include [GLOV12, GOSV14, Wee10a].

The question of whether we can securely realize a task via black-box access to a general primitive is of theoretical interest, towards a better understanding of the complexity and minimal assumptions necessary, as well as of practical significance, since black-box (thus, modular) constructions are typically simpler and more efficient. Indeed, non-black-box constructions tend to be less efficient due to the typical use of general NP reductions to order to prove statements in zero knowledge; this impacts both computational complexity as well as communication complexity (which we interpret broadly to mean message lengths for protocols and key size and ciphertext size for encryption schemes). Moreover, if resolved in the affirmative, the solution can provide new insights and techniques for circumventing the use of NP reductions and zero knowledge in the known constructions.

1.1 Our Contributions

Non-Malleability against Chosen Plaintext Attacks. As mentioned above, in this paper we provide a black-box construction of non-malleable encryption from semantically-secure encryption, where previous work achieved it only through a non-black-box construction [PSV06], or prior to that, only using additional assumptions [DDN00].

> **Main theorem (informal)** There exists a (fully) black-box construction of a non-malleable encryption scheme from any semantically secure one.

That is, we provide a "wrapper program" that given any subroutines for computing a semantically secure encryption scheme, computes a non-malleable encryption scheme. While this is interesting in and of itself, our construction also compares favorably with previous work in several regards:

- *Improved parameters.* We improve on the computational complexity of previous constructions based on general assumptions. In particular, we do not have to do an NP-reduction in either encryption or decryption, although we do have to pay the price of the running time of error correcting code alogrithms (e.g., Berlekamp-Welch algorithm [BW86]). The running time incurs a multiplicative overhead that is quasi-linear in the security parameter, over the running time of the underlying CPA secure scheme. Moreover, the sizes of public keys and ciphertext are independent of the computational complexity of the underlying scheme.

- *Conceptual simplicity/clarity.* Our scheme (and the analysis) is arguably much simpler than many of the previous constructions, and unlike [PSV06], entirely self-contained (apart from some basic tools from coding theory). We do not need to appeal to notions of zero-knowledge [GMR89, GMW91], nor do we touch upon subtle technicalities like adaptive vs non-adaptive NIZK. Our construction may be covered in an introductory graduate course on cryptography without requiring zero knowledge as a pre-requisite.

- *Ease of implementation.* Our scheme is easy to describe and can be easily implemented in a modular fashion.

- *Robustness.* Our construction achieves non-malleability as is even when instantiated with an encryption scheme with negligible decryption error. This is in contrast to the [DDN00] and [PSV06] constructions, which require that the underlying encryption scheme be first "immunized" against decryption errors (c.f. [DNR04]); these constructions are otherwise susceptible to an attack described by Dwork, Naor and Reingold [DNR04].

Our Techniques. At a high level, we follow the cut-and-choose approach for consistency checks from [PSV06], wherein the randomness used for cut-and-choose is specified in the secret key. A crucial component of our construction is a message encoding scheme based on a "reconstructable probabilistic encoding scheme" (an extension of probabilistic encoding schemes introduced by [DGR99]). Reconstructable probabilistic encoding schemes can correct a constant fraction of errors, and they have a secrecy property which allows a large number of positions in the outputted codeword to be revealed, without leaking any information about the encoded message; in addition, given a message and a partial codeword for it, the schemes allow the reconstruction of the whole codeword consistent with them. Similar properties of encoding via low-degree polynomials have been exploited in the early work on secure multi-party computation with malicious adversaries

$$\begin{tikzcd}
\text{IND-CPA} \arrow[r] \arrow[d] \arrow[dr] & \text{NM-CPA} \\
\text{IND-}q\text{-CCA} & \text{NM-}q\text{-CCA} \\
\text{IND-CCA2} \arrow[r, phantom, "="] & \text{NM-CCA2}
\end{tikzcd}$$

Figure 1: Summary of our positive results.

[BGW88]. We think this technique may be useful in eliminating general NP-reductions in other constructions in cryptography (outside of public-key encryption). Indeed, this has already proven true in several subsequent works (see Section 1.4).

Implications for Chosen Ciphertext Attacks. While the notion of (passive) non-malleability is important and interesting in its own right, it is also interesting as an intermediate notion between semantic security and fully active chosen ciphertext attacks, where the adversary is allowed to query the decryption oracle as well. Recall that in CCA1 attacks, the adversary may access the decryption oracle only before seeing the challenge, while in the stronger CCA2, adaptive decryption queries (after seeing the challenge) are also allowed, except for the challenge itself (cf., [Gol04, NY90, RS92, DDN00]).[1] Finally, of particular relevance to us is the notion of bounded CCA2 attack, introduced by Cramer et al. [CHH$^+$07], which is a relaxation of the CCA2 attack (and incomparable to CCA1). Here, the adversary is only allowed make an a priori bounded number of queries q to the decryption oracle, where q is fixed prior to choosing the parameters of the encryption scheme. We will consider both the indistinguishability and non-malleability security guarantees; while these are equivalent under a CCA2 attack (namely every IND-CCA2 secure scheme is also NM-CCA2 secure)[DDN00], the latter is a strictly stronger security notion than the former under a bounded CCA2 attack, i.e., every NM-q-CCA2 secure encryption is also IND-q-CCA2 secure, but the converse is not necessarily true [CHH$^+$07].

Cramer et. al. [CHH$^+$07] obtained two constructions, starting from any semantically secure (IND-CPA) encryption:

- an encryption scheme that achieves indistinguishability under a bounded-CCA2 attack via a black-box construction, wherein the size of the public key and ciphertext are quadratic in q; and

- an encryption scheme that is non-malleable under a bounded-CCA2 attack via a non-black-box construction, wherein the size of the public key and ciphertext are linear in q.

Combining their approach for the latter construction with our main result, we obtain a result that simultaneously improves over both the above.

> **Corollary (informal)** There exists a (fully) black-box construction of an encryption scheme that is non-malleable under a bounded CCA2 attack (NM-q-CCA2) from any

[1] We note that non-malleability against a passive CPA attack is equivalent to indistinguishability against a CCA2 attack that allows only a *single* adaptive stage (with possibly many parallel queries) after seeing the challenge.

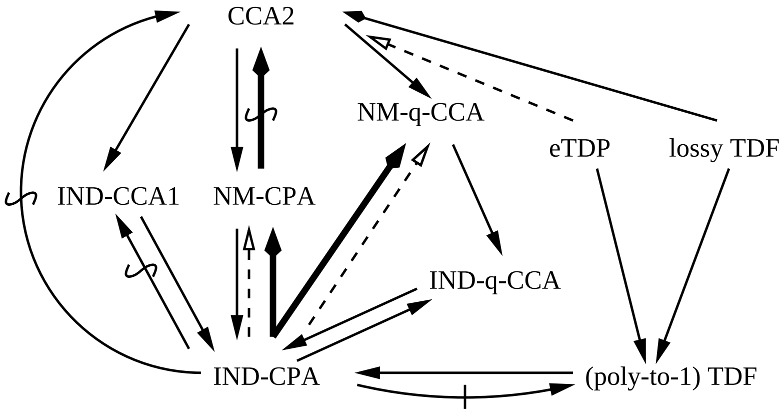

Figure 2: Known relations among generic encryption primitives, and our results. Solid lines indicate black box constructions, and dotted lines indicate non-black-box constructions (c.f. [BHSV98, DDN00, PSV06, CHH$^+$07, PW08]). Arrows with the '|' symbol (resp., the '$\sim$' symbol) in the middle indicate the separations with respect to black-box reductions (resp., black box shielding reductions, c.f. [GMR01, GMM07]). Our contributions are indicated with the thick arrows.

> semantically secure (IND-CPA) encryption scheme. Moreover, for this construction, the size of the public key and ciphertext are linear in the number of queries q.

Our positive results are summarized in Figure 1.

We also use our construction to obtain a negative (separation) result between non-malleability and CCA security. Our main construction has the additional property that the decryption algorithm does not query the encryption functionality of the underlying scheme. Gertner, Malkin and Myers [GMM07] referred to such constructions as "shielding" and they showed that there is no shielding black-box construction of IND-CCA1 secure encryption schemes from semantically secure ones. Combined with the fact that any shielding construction when composed with our construction is again shielding, this yields the following:

> **Corollary (informal)** There exists no shielding black-box construction of a IND-CCA1, NM-CCA1, or CCA2 encryption scheme from non-malleable (NM-CPA) encryption.

This corollary for IND-CCA1 follows from combining [GMM07] with our result, and immediately implies the same separation for NM-CCA1 and CCA2, as both these notions trivially imply IND-CCA1 security. Our results, as well as other known relationships between relevant primitives, are summarized in Figure 2.

We believe that our work and techniques provide useful insight towards constructions of CCA2-secure schemes. First, our work can be viewed as guiding potential approaches towards answering the open problem of whether CCA2-secure schemes can be constructed from CPA ones (directing the efforts towards constructions of CCA2 schemes from non-malleable schemes, either in a non-black-box, or in a black-box non-shielding way). Second, our techniques may be useful towards a black-box construction of CCA2-secure encryptions from enhanced trapdoor permutations [Gol04] (rather than the non-black-box transformation of [DDN00] via NIZK). More generally, our work indicates that the barrier towards CCA2-security may not lie in black-box vs non-black-box use of the underlying primitive (specifically, previous constructions are all non-black-box and do not shed any light on this issue). We discuss this in more detail (as well as discussing some related work) next.

1.2 Overview of Our Construction

Recall the DDN [DDN00] and PSV [PSV06] constructions: to encrypt a message, one (a) generates k encryptions of the same message under independent keys, (b) gives a non-interactive zero-knowledge proof that all resulting ciphertexts are encryptions of the same message, and (c) signs the entire bundle with a one-time signature. It is in step (b) that a general NP-reduction is used, which in return makes the construction non-black-box. In the proof of security, we exploit that fact that for a well-formed ciphertext, we can recover the message if we know the secret key for any of the k encryptions.

How do we guarantee that a tuple of k ciphertexts are encryptions of the same plaintext without using a zero-knowledge proof and without revealing any information about the underlying plaintext? Naively, one would like to use a cut-and-choose approach (as has been previously used in [LP07] to eliminate zero-knowledge proofs in the context of secure two-party computation), namely decrypt and verify that some constant fraction, say $k/2$ of the ciphertexts are indeed consistent. There are two issues with this approach:

- First, if only a constant number of ciphertexts are inconsistent, then we are unlikely to detect the inconsistency. To circumvent this problem, we could decrypt by outputting the majority of the remaining $k/2$ ciphertexts.
- The second issue is more fundamental: decrypting any of the ciphertexts will immediately reveal the underlying message, whereas it is crucial for the proof that we can enforce consistency while learning nothing about the underlying message.

We circumvent both issues by using a more sophisticated encoding of the message m based on reconstructable probabilistic encoding schemes, instead of merely making k copies of the message as in the above schemes. Specifically, let E be the encoding algorithm of the reconstructable probabilistic encoding scheme with output of length $2k$ (over the alphabet of the scheme). We first obtain an encoding w of m (i.e., $w \leftarrow \mathsf{E}(m)$) and then generates k encryptions of the same w. Thus, we construct a $k \times 2k$ matrix such that entry (i, j) holds w_j. To verify consistency, we will decrypt a random subset of k columns, and check that all the entries in each of these columns are the same; the random subset will be chosen in key generation and embedded into the private key. The issue that only a tiny number of ciphertexts are inconsistent is now handled using the error-correcting properties of the encoding scheme. The second issue is addressed since, due to the secrecy properties of the encoding scheme, learning a random subset of k columns in a valid encoding reveals nothing about the underlying message m. We note that encoding m using a secret-sharing scheme appears in the earlier work of Cramer et al. [CHH$^+$07], but they do not consider redundancy or error-correction.

As before, we encrypt all the entries of the matrix using independent keys and then sign the entire bundle with a one-time signature. It is important that the encoding also provides a robustness guarantee similar to that of repeating the message k times: we are able to recover the message for a valid encryption if we can decrypt *any* row in the matrix. Indeed, this is essentially our entire scheme with two technical caveats:

- As with previous schemes, we will associate one pair of public/secret key pairs with each entry of the matrix, and we will select the public key for encryption based on the verification key of the one-time signature scheme.
- To enforce consistency, we will need a codeword check (checking if the first row has only a small number of errors) in addition to the column check outlined above. The reason for this is fairly subtle and we will highlight the issue in the formal exposition of our construction.

Decreasing ciphertext size. To encrypt an n-bit message with security parameter k, our construction yields $O(k^2)$ encryptions of n-bit messages in the underlying scheme. It is easy to see that this may be reduced to $O(k \log^2 k)$ encryptions while maintaining security against ppt adversaries, by reducing the number of columns to $O(\log^2 k)$.

1.3 Towards Full CCA2 Security?

One of the biggest open problems remaining in the area is the construction of CCA2-secure encryption via black-box access to a low-level general primitive (e.g., enhanced trapdoor permutations), or the construction (whether black-box or not) of CCA2-secure encryption from semantically secure encryption. Below we describe the perspective on achieving full CCA2 security, both pre and post publication of our original work, [CDMW08], at TCC 2008.

[CDMW08] and prior works. Early works pertaining to this open problem were limited to non-black-box constructions of CCA2-secure encryption from enhanced trapdoor permutations [DDN00, Sah99, Lin06]. A different line of work focused on (very) efficient constructions of CCA2-secure encryptions under specific number-theoretic assumptions (c.f. [CS98, CS04, CHK04]). Apart from the construction based on identity-based encryption [CHK04], all these constructions can be described under the following framework (c.f. [BFM88, NY90, RS92, ES02]). Start with some cryptographic hardness assumption that allows us to build a semantically secure encryption scheme, and then prove/verify that several ciphertexts satisfy certain relations in one of two ways:

- exploiting algebraic relations from the underlying assumption to deduce additional structure in the encryption scheme (e.g. homomorphic, reusing randomness) [CS98, CS04];
- apply a general NP reduction to prove in non-interactive zero knowledge (NIZK) statements that relate to the primitive [DDN00, Sah99, Lin06].

These previous approaches do not yield black-box constructions under general assumptions and, indeed, our work does not use the above framework.

Peikert and Waters [PW08] (who also do not use the above framework), made substantial progress towards the open problem. They constructed CCA2-secure encryption schemes via black-box access to a new primitive they introduced called lossy trapdoor functions, and in addition, gave constructions of this primitive from number-theoretic and worst-case lattice assumptions.[2]

[2] Note that our results are incomparable, since we start from weaker assumptions (any semantically secure encryption, or any poly-to-1 trapdoor function, rather than lossy trapdoor functions), but derive a weaker security guarantee (non-malleability, rather than CCA2 security).

Unfortunately, their work does not provide a black-box construction of CCA2-secure encryption from enhanced trapdoor permutations.

Our work may be viewed as a step towards solving this gap (and a small step in the more general research agenda of understanding the power of black-box constructions). Specifically, the security guarantee provided by non-malleability lies between semantic security and CCA2 security, and we show how to derive non-malleability in a black-box manner from the minimal assumption possible, i.e., semantic security. In the process, we show how to enforce consistency of ciphertexts in a black-box manner. This issue arises in black-box constructions of both CCA2-secure and non-malleable encryptions. However, our consistency checks only satisfy a weaker notion of non-adaptive soundness, which is sufficient for non-malleability but not for CCA2-security (c.f. [PSV06]). Indeed, the main obstacle towards achieving full CCA2 security from either semantically secure encryptions or enhanced trapdoor permutations using our approach (and also the [PSV06] approach) lies in guaranteeing soundness of the consistency checks against an adversary that can adaptively determine its queries depending on the outcome of previous consistency checks. It seems conceivable that using a non-shielding construction (as in [Ms09, HLW12]) that uses re-encryption may help overcome this obstacle.

Subsequent works. Recently there has been significant, renewed effort on constructing CCA2-secure encryption from new assumptions. Notably, all of these subsequent works deviate from the classic encrypt-and-prove paradigm discussed above. We next discuss several of these recent works. Rosen and Segev [RS09] introduced a new assumption of trapdoor functions secure under correlated products, showed that this assumption is weaker than the assumption of lossy trapdoor functions, and presented a simple, black-box construction of CCA2-secure encryption under this assumption. Kiltz, Mohassel and O'Neill [KMO10] formalized an even weaker assumption called adaptive trapdoor functions, and showed that it is sufficient for black-box constructions of CCA2-secure encryption. Hofheinz and Kiltz [HK09] presented the first construction of CCA2-secure encryption from hardness of factoring. Wee [Wee10b] abstracted their construction and introduced a new primitive, extractable hash proofs, which is sufficient for CCA2-secure encryption. Moreover, [Wee10b] showed a construction of extractable hash proofs from the CDH assumption, which yields the first construction of CCA2-secure encryption from CDH. Other works such as [Ms09, HLW12] showed how to obtain multi-bit CCA2-secure encryption from single-bit CCA2-secure encryption. Another line of research (c.f. [MH14, MSs12, Dac14]) focused on black-box constructions of CCA2-secure encryption from various non-falsifiable assumptions.

1.4 Applications of Our Techniques in Subsequent Works

Since the publication of this work at TCC 2008, the encoding scheme introduced here has been used in a number of follow-up works. There have been black-box constructions of non-malleable commitments [PW09], set intersection protocols from homomorphic encryptions [DMRY09], and a CCA2-secure encryption scheme for strings starting from one for bits [Ms09]. The works of [Wee10a, LP12, KMO14, Kiy14] used our encoding in the context of black-box, round-efficient secure computation. The works of [GLOV12, GOSV14] generalized our approach to proving relations beyond equality using VSS and the paradigm of MPC-in-the-head.

2 Preliminaries and Definitions

Notation. We adopt the notation used in [PSV06]. We use $[n]$ to denote $\{1, 2, \ldots, n\}$. If A is a probabilistic polynomial time (hereafter, ppt) algorithm that runs on input x, $A(x)$ denotes the random variable according to the distribution of the output of A on input x. We denote by $A(x; r)$ the output of A on input x and random coins r. Computational indistinguishability between two ensembles A and B is denoted by $A \overset{c}{\approx} B$, and statistical indistinguishability between two distributions A and B is denoted by $A \overset{s}{\approx} B$. Given two strings v, w of length ℓ over an alphabet Σ, we say that v and w are δ-far if they disagree in greater than $\delta \cdot \ell$ positions, where $0 \leq \delta \leq 1$; we say that v and w are δ-close if they agree in greater than $\delta \cdot \ell$ positions.

2.1 Semantically Secure Encryption

Definition 1 (Encryption scheme)**.** *A triple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an* encryption scheme*, if* Gen *and* Enc *are ppt algorithms and* Dec *is a deterministic polynomial-time algorithm which satisfies the following property:*

> Correctness. *There exists a negligible function $\mu(\cdot)$ such that for all sufficiently large k, we have that with probability $1 - \mu(k)$ over $(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k)$: for all m, $\Pr[\mathsf{Dec}_{\text{SK}}(\mathsf{Enc}_{\text{PK}}(m)) = m] = 1$.*

Definition 2 (Semantic security)**.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let the random variable $\mathsf{IND}_b(\Pi, A, k)$, where $b \in \{0, 1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k \in \mathbb{N}$, denote the result of the following probabilistic experiment:*

```
$\mathsf{IND}_b(\Pi, A, k)$ :
  $(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k)$
  $(m_0, m_1, \text{STATE}_A) \leftarrow A_1(\text{PK})$ s.t. $|m_0| = |m_1|$
  $y \leftarrow \mathsf{Enc}_{\text{PK}}(m_b)$
  $D \leftarrow A_2(y, \text{STATE}_A)$
  Output $D$
```

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* indistinguishable under a chosen-plaintext (CPA) attack*, or* semantically secure*, if for any ppt algorithms $A = (A_1, A_2)$ the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{IND}_0(\Pi, A, k)\right\}_{k \in \mathbb{N}} \overset{c}{\approx} \left\{\mathsf{IND}_1(\Pi, A, k)\right\}_{k \in \mathbb{N}}$$

It follows from a straight-forward hybrid argument that semantic security implies indistinguishability of multiple encryptions under independently chosen keys:

Proposition 1. *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a semantically secure encryption scheme and let the random variable $\mathsf{mIND}_b(\Pi, A, k, \ell)$, where $b \in \{0, 1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k \in \mathbb{N}$, denote the result of the following probabilistic experiment:*

```
$\mathsf{mIND}_b(\Pi, A, k, \ell)$ :
  For $i = 1, \ldots, \ell$: $(\text{PK}_i, \text{SK}_i) \leftarrow \mathsf{Gen}(1^k)$
  $(\langle m_0^1, \ldots, m_0^\ell \rangle, \langle m_1^1, \ldots, m_1^\ell \rangle, \text{STATE}_A) \leftarrow A_1(\langle \text{PK}_1, \ldots, \text{PK}_\ell \rangle)$
    s.t. $|m_0^1| = |m_1^1| = \cdots = |m_0^\ell| = |m_1^\ell|$
    For $i = 1, \ldots, \ell$: $y_i \leftarrow \mathsf{Enc}_{\mathrm{PK}_i}(m_b^i)$
    $D \leftarrow A_2(y_1, \ldots, y_\ell, \mathrm{STATE}_A)$
    Output $D$
```

then for any ppt algorithms $A = (A_1, A_2)$ and for any polynomial $p(k)$ the following two ensembles are computationally indistinguishable:

$$\left\{\mathsf{mIND}_0(\Pi, A, k, p(k))\right\}_{k \in N} \overset{c}{\approx} \left\{\mathsf{mIND}_1(\Pi, A, k, p(k))\right\}_{k \in N}$$

2.2 Non-malleable Encryption

Definition 3 (Non-malleable encryption [PSV06])**.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let the random variable $\mathsf{NME}_b(\Pi, A, k, \ell)$ where $b \in \{0,1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k, \ell \in \mathbb{N}$ denote the result of the following probabilistic experiment:*

```
$\mathsf{NME}_b(\Pi, A, k, \ell)$ :
    $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k)$
    $(m_0, m_1, \mathrm{STATE}_A) \leftarrow A_1(\mathrm{PK})$ s.t. $|m_0| = |m_1|$
    $y \leftarrow \mathsf{Enc}_{\mathrm{PK}}(m_b)$
    $(\psi_1, \ldots, \psi_\ell) \leftarrow A_2(y, \mathrm{STATE}_A)$
    Output $(d_1, \ldots, d_\ell)$ where $d_i = \begin{cases} \perp & \text{if } \psi_i = y \\ \mathsf{Dec}_{\mathrm{SK}}(\psi_i) & \text{otherwise} \end{cases}$
```

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* non-malleable under a chosen plaintext (CPA) attack *if for any ppt algorithms $A = (A_1, A_2)$ and for any polynomial $p(k)$, the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{NME}_0(\Pi, A, k, p(k))\right\}_{k \in \mathbb{N}} \overset{c}{\approx} \left\{\mathsf{NME}_1(\Pi, A, k, p(k))\right\}_{k \in \mathbb{N}}$$

It was shown in [PSV06] that an encryption that is non-malleable (under Definition 3) remains non-malleable even if the adversary A_2 receives several encryptions under many different public keys (the formal experiment is the analogue of mIND for non-malleability).

2.3 Bounded-CCA2 Non-Malleability

The definition of Bounded-CCA2 Non-Malleability is almost identical to the definition of Non-Malleability except here, we allow the adversary to query Dec at most q times in the non-malleability experiment (but it must not query Dec on the challenge ciphertext).

Definition 4 (Bounded-CCA2 non-malleable encryption [CHH$^+$07])**.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let the random variable $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b(\Pi, A, k, \ell)$ where $b \in \{0,1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k, \ell \in \mathbb{N}$ denote the result of the following probabilistic experiment:*

```
$\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b(\Pi, A, k, \ell)$ :
    $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k)$
    $(m_0, m_1, \mathrm{STATE}_A) \leftarrow A_1^{O_1}(\mathrm{PK})$ s.t. $|m_0| = |m_1|$
    $y \leftarrow \mathsf{Enc}_{\mathrm{PK}}(m_b)$
```

$(\psi_1, \ldots, \psi_\ell) \leftarrow A_2^{O_2}(y, \text{STATE}_A)$

Output $(d_1, \ldots, d_\ell)$ *where* $d_i = \begin{cases} \perp & \textit{if } \psi_i = y \\ \mathsf{Dec}_{\text{SK}}(\psi_i) & \textit{otherwise} \end{cases}$

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* non-malleable under a bounded-CCA2 attack *for a function* $q(k) : \mathbb{N} \to \mathbb{N}$ *if* $\forall$ *ppt algorithms* $A = (A_1, A_2)$ *which make* $q(k)$ *total queries to the oracles and for any polynomial* $p(k)$*, the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_0(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \overset{c}{\approx} \left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_1(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}}$$

The oracle $O_1 = \mathsf{Dec}_{\text{SK}}(\cdot)$ *is the decryption oracle.* $O_2 = \mathsf{Dec}^y_{\text{SK}}(\cdot)$ *is the decryption oracle except that* O_2 *returns* $\perp$ *when queried on* y*.*

2.4 (Strong) One-Time Signature Schemes

Informally, a (strong) one-time signature scheme $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ is an existentially unforgeable signature scheme, with the restriction that the signer signs at most one message with any key. This means that an efficient adversary, upon seeing a signature on a message m of his choice, cannot generate a valid signature on a different message, or a different valid signature on the same message m. Such schemes can be constructed in a black-box way from one-way functions [Rom90, Lam79], and thus from any semantically secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ using black-box access only to Gen.

2.5 Reconstructable Probabilistic Encoding Scheme

Informally, reconstructable probabilistic encoding (RPE) schemes can correct a constant fraction of errors, and they have a secrecy property which allows a large number of positions in the output codeword to be revealed, without leaking any information about the encoded message. In addition, given a message and a partial codeword for it, the schemes allow the reconstruction of the whole codeword consistent with them.

Definition 5 (Reconstructable probabilistic encoding)**.** *We say a triple* $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ *is a* reconstructable probabilistic encoding scheme *with parameters* $(n, \ell, \delta, t, \Sigma)$*, where* $n, \ell, t \in \mathbb{N}$*,* $0 < \delta < 1$*, and* Σ *is an alphabet, if it satisfies the following properties:*

1. ***Error correction.*** $\mathsf{E} : \{0,1\}^n \to \Sigma^\ell$ *is an efficient probabilistic procedure, which maps a message* $m \in \{0,1\}^n$ *to a distribution over* Σ^ℓ*. If we let* $\mathcal{W}$ *denote the support of* E*, any two strings in* $\mathcal{W}$ *are* δ*-far. Moreover,* D *is an efficient procedure that given any* $w' \in \Sigma^\ell$ *that is* $(1 - \delta/2)$*-close to some string* w *in* $\mathcal{W}$*, outputs* w *along with a consistent* m*.*

2. ***Secrecy of partial views.*** *For all* $m \in \{0,1\}^n$ *and all sets* $S \subset [\ell]$ *of size* t*, the projection of* $\mathsf{E}(m)$ *onto the coordinates in* S*, as denoted by* $\mathsf{E}(m)|_S$*, is identically distributed to the uniform distribution over* Σ^t*.*

3. ***Reconstruction from partial views.*** R *is an efficient procedure that given any set* $S \subset [\ell]$ *of size* t*, any* $(\alpha_1, \ldots, \alpha_t) \in \Sigma^t$*, and any* $m \in \{0,1\}^n$*, samples from the distribution* $\mathsf{E}(m)$ *with the constraint* $\mathsf{E}(m)|_S = (\alpha_1, \ldots, \alpha_t)$*.*

We can construct an RPE scheme with a Reed-Solomon code. We note the construction is implicit in [BGW88].

Lemma 1. *For any $n, t \in \mathbb{N}$ and any constant δ such that $0 < \delta < 1$, there is an RPE scheme with parameters $(n, \lceil \frac{t}{1-\delta} \rceil, \delta, t, \mathrm{GF}(2^n))$.*

Proof. We will implicitly identify $\{0,1\}^n$ with the field $\mathrm{GF}(2^n)$; an integer i with $0 \leq i < 2^n$ will also be implicitly encoded into a field element in $\mathrm{GF}(2^n)$. Set $\ell = \lceil \frac{t}{1-\delta} \rceil$ and $\Sigma = \mathrm{GF}(2^n)$. We construct an RPE scheme $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ as follows:

- $\mathsf{E}(m)$: Choose a random degree-t polynomial q over $\mathrm{GF}(2^n)$ such that $q(0) = m$ and output $w = (q(1), q(2), \ldots, q(\ell))$.
- $\mathsf{D}(w')$: Decode w' using the Berlekamp-Welch algorithm and output (w, m), where w is the corrected codeword, and m is the original message.
- $\mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), m)$: Let $S = \{i_1, \ldots, i_t\}$. Determine the degree-t polynomial q such that $q(0) = m$, $q(i_1) = \alpha_1$, $q(i_2) = \alpha_2$, $\ldots$, $q(i_t) = \alpha_t$. Output $(q(1), \ldots, q(\ell))$.

Property (1) holds since we simply use the Reed-Solomon code $\mathcal{W}$ in encoding and decoding, where

$$\mathcal{W} = \{\ (q(1), \ldots, q(\ell)) \mid q \text{ is a degree } t \text{ polynomial } \}.$$

Note that $\mathcal{W}$ is a code over the alphabet $\mathrm{GF}(2^n)$ with minimum relative distance δ, which means we may efficiently correct up to $\delta/2$ fraction errors. Properties (2) and (3) hold since the codeword $(q(1), \ldots, q(\ell))$ is a $(t+1)$-out-of-ℓ secret-sharing of m using Shamir's secret-sharing scheme, and $(m, \alpha_1, \ldots, \alpha_t)$ allows the reconstruction of the (one and only) degree-t polynomial. □

3 Construction

Given an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we construct a new encryption scheme $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$, summarized in Figure 3, and described as follows.

Encryption. For simplicity we assume the security parameter k is a power of two. Let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be an RPE scheme with parameters $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$ according to Lemma 1, where $t = \log^2 k$. The public key for Π comprises $4kt$ public keys from Gen indexed by a triplet $(i, j, b) \in [k] \times [2t] \times \{0,1\}$; there are two keys corresponding to each entry of a $k \times 2t$ matrix. To encrypt a message $m \in \{0,1\}^n$, we (a) compute $(s_1, \ldots, s_{2t}) \leftarrow \mathsf{E}(m)$, (b) generate $(\textsc{sksig}, \textsc{vksig})$ for a one-time signature, (c) compute a $k \times 2t$ matrix $\vec{c} = (c_{i,j})$ of ciphertexts where $c_{i,j} = \mathsf{Enc}_{\mathrm{PK}_{i,j}^{v_i}}(s_j)$, and (d) sign $\vec{c}$ using $\textsc{sksig}$.

$$\begin{pmatrix} \mathsf{Enc}_{\mathrm{PK}_{1,1}^{v_1}}(s_1) & \mathsf{Enc}_{\mathrm{PK}_{1,2}^{v_1}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}_{1,2t}^{v_1}}(s_{2t}) \\ \mathsf{Enc}_{\mathrm{PK}_{2,1}^{v_2}}(s_1) & \mathsf{Enc}_{\mathrm{PK}_{2,2}^{v_2}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}_{2,2t}^{v_2}}(s_{2t}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{Enc}_{\mathrm{PK}_{k,1}^{v_k}}(s_1) & \mathsf{Enc}_{\mathrm{PK}_{k,2}^{v_k}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}_{k,2t}^{v_k}}(s_{2t}) \end{pmatrix}$$

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme, $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a strong one-time signature scheme, and $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be a reconstructable probabilistic encoding scheme with parameters $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$ where $t = \log^2 k$.

$\mathsf{NMGen}(1^k)$:

1. For $i \in [k], j \in [2t], b \in \{0,1\}$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\mathrm{PK}^b_{i,j}, \mathrm{SK}^b_{i,j})$.
2. Pick a random subset $S \subset [2t]$ of size t.
3. Set $\mathrm{PK} = \Big\{(\mathrm{PK}^0_{i,j}, \mathrm{PK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$ and $\mathrm{SK} = \Big\{S, (\mathrm{SK}^0_{i,j}, \mathrm{SK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$.

$\mathsf{NMEnc}_{\mathrm{PK}}(m)$:

1. Compute $(s_1, \ldots, s_{2t}) \leftarrow \mathsf{E}(m)$, where $m \in \{0,1\}^n$.
2. Run $\mathsf{GenSig}(1^k)$ to generate $(\mathrm{SKSIG}, \mathrm{VKSIG})$. Let $(v_1, \ldots, v_k)$ be the binary representation of VKSIG.
3. Compute the ciphertext $c_{i,j} \leftarrow \mathsf{Enc}_{\mathrm{PK}^{v_i}_{i,j}}(s_j)$, for $i \in [k], j \in [2t]$.
4. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SKSIG}}(\vec{c})$ where $\vec{c} = (c_{i,j})$.
5. Output the tuple $[\vec{c}, \mathrm{VKSIG}, \sigma]$.

$\mathsf{NMDec}_{\mathrm{SK}}([\vec{c}, \mathrm{VKSIG}, \sigma])$:

1. (`sig-check`) Verify the signature with $\mathsf{VerSig}_{\mathrm{VKSIG}}[\vec{c}, \sigma]$.
2. (`decoding-check`) Let $\vec{c} = (c_{i,j})$ and $\mathrm{VKSIG} = (v_1, \ldots, v_k)$. Compute $s_j = \mathsf{Dec}_{\mathrm{SK}^{v_1}_{1,j}}(c_{1,j})$, $j = 1, \ldots, 2t$. Compute $((w_1, \ldots, w_{2t}), m) \leftarrow \mathsf{D}(s_1, \ldots, s_{2t})$. If the decoding fails or $(w_1, \ldots, w_{2t})$ is not 0.9-close to $(s_1, \ldots, s_{2t})$, then output $\perp$.
3. (`column-check`) For all $j \in S$, check that $\mathsf{Dec}_{\mathrm{SK}^{v_1}_{1,j}}(c_{1,j}) = \mathsf{Dec}_{\mathrm{SK}^{v_2}_{2,j}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\mathrm{SK}^{v_k}_{k,j}}(c_{k,j})$.
4. (`codeword-check`) For all $j \in S$, check that $s_j = w_j$.
5. If all the checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

Figure 3: The Non-Malleable Encryption Scheme Π

Consistency Checks. A valid ciphertext in Π satisfies two properties: (1) the first row is an encryption of a codeword in $\mathcal{W}$ and (2) every column comprises k encryptions of the same plaintext. We want to design consistency checks that reject ciphertexts that are "far" from being valid ciphertexts under Π. For simplicity, we will describe the consistency checks as applied to the underlying matrix of plaintexts. The checks depend on a random subset S of t columns chosen during key generation.

> Decoding Check (`decoding-check`): We find a codeword w that is 0.9-close to the first row of the matrix; the check fails if no such w exists. Recall that the underlying RPE has parameters $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$, so it can correct up to 0.25 fraction errors.
>
> Column Check (`column-check`): We check that each of the columns in S comprises entirely of the same value.
>
> Codeword Check (`codeword-check`): We check that *the first row of the matrix agrees with w at the positions indexed by S.*

The codeword check reassures that with high probability, the first row of the matrix is $(1 - o(1))$-close to w. We explain its significance after describing the alternative decryption algorithm in the analysis.

Decryption. To decrypt, we (a) verify the signature and run both consistency checks, and (b) if all the checks accept, decode the codeword w and output the result, otherwise output $\perp$. Note that to decrypt we only need the $4t$ secret keys corresponding to the first row of the matrix and $2t(k-1)$ additional secret keys corresponding to columns in S.

Note that the decryption algorithm may be stream-lined, for instance, by running the codeword check only if the column check succeeds. We choose to present the algorithm as is in order to keep the analysis simple; in particular, we will run both consistency checks independent of the outcome of the other.

4 Analysis

Having presented our construction, we now formally state and prove our main result:

Theorem 1. (Main Theorem, restated). *Assume there exists an encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is semantically secure under a CPA attack. Then there exists an encryption scheme $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$ that is non-malleable under a CPA attack.*

We establish the theorem (as in [DDN00, PSV06], etc) via a series of hybrid arguments and deduce indistinguishability of the intermediate hybrid experiments from the semantic security of the underlying encryption scheme under some set of public keys Γ. To do so, we will need to implement an alternative decryption algorithm NMDec^* that is used in the intermediate experiments to simulate the actual decryption algorithm NMDec in the non-malleability experiment. We need NMDec^* to achieve two conflicting requirements:

- NMDec^* and NMDec must agree on essentially all inputs, including possibly malformed ciphertexts;

- We can implement NMDec^* without having to know the secret keys corresponding to the public keys in Γ.

Of course, designing NMDec^* is difficult precisely because NMDec uses the secret keys corresponding to the public keys in Γ.

Here is a high-level (but extremely inaccurate) description of how NMDec^* works: Γ is the set of public keys corresponding to the first row of the $k \times 2t$ matrix. To implement NMDec^*, we will decrypt the i'th row of the matrix of ciphertexts, for some $i > 1$, which the column check (if successful) guarantees to agree with the first row in most positions; error correction takes care of the tiny fraction of disagreements.

4.1 Alternative Decryption Algorithm NMDec^*

Let $\textsc{vksig}^* = (v_1^*, \ldots, v_k^*)$ denote the verification key in the challenge ciphertext given to the adversary in the non-malleability experiment, and let $\textsc{vksig} = (v_1, \ldots, v_k)$ denote the verification key in (one of) the ciphertext(s) generated by the adversary. First, we modify the signature check to also output $\perp$ if there is a forgery, namely $\textsc{vksig} = \textsc{vksig}^*$. Next, we modify the consistency checks (again, as applied to the underlying matrix of plaintexts) as follows:

> Decoding Check (`decoding-check`*): Let i be the smallest value such that $v_i \neq v_i^*$ (which exists because $\textsc{vksig} \neq \textsc{vksig}^*$). We find a codeword w that is 0.8-close to the i'th row of the matrix (note agreement threshold is smaller than before); the check fails if so such w exists.
>
> Column Check (`column-check`*): This is exactly as before; we check that the each of the columns in S comprises entirely of the same value.
>
> Codeword Check (`codeword-check`*): This is exactly before; we check that the first row of the matrix agrees with w at the positions indexed by S.

To decrypt, run the modified signature and consistency checks, and if all the checks accept, decode the codeword w and output the result, otherwise output $\perp$. To implement the modified consistency checks and decryption algorithm, we need the $3t$ secret keys for each row of the matrix, that is, $2t$ keys for the decryption of the entire row indexed by $\overline{\textsc{vksig}^*}$ and t keys for the decryption of the columns of S in the row indexed by $\textsc{vksig}^*$.

Remark on the Codeword Check. At first, the codeword check may seem superfluous. Suppose we omit the codeword check, and as before, define w to be a codeword that is 0.9-close to the first row and 0.8-close to the i'th row in the respective decryption algorithms; the gap is necessary to take into account inconsistencies not detected by the column check. Now, consider a malformed ciphertext ψ for Π where in the underlying matrix of plaintexts, each row is the same corrupted codeword that is 0.15-far from a valid codeword. Without the codeword checks, ψ will be an invalid ciphertext according to NMDec and a valid ciphertext according to NMDec^* and can be used to distinguish the intermediate hybrid distributions in the analysis; with the codeword checks, ψ is an invalid ciphertext according to both. It is also easy to construct a problematic malformed ciphertext for the case where both agreement thresholds are set to the same value (say 0.9-close).

4.2 A Promise Problem

Recall the guarantees we would like from NMDec and NMDec^*:

- On input a ciphertext that is an encryption of a message m under Π, both NMDec and NMDec^* will output m with probability 1.
- On input a ciphertext that is "close" to an encryption of a message m under Π, both NMDec and NMDec^* will output m with the same probability (the exact probability is immaterial) and $\perp$ otherwise.
- On input a ciphertext that is "far" from any encryption, then both NMDec and NMDec^* output $\perp$ with high probability.

To quantify and establish these guarantees, we consider the following promise problem (Π_Y, Π_N) that again refers to the underlying matrix of plaintexts. An instance is a matrix of k by $2t$ each entry of which lies in $\{0,1\}^n \cup \perp$.

Π_Y (YES instances) — for some $w \in \mathcal{W}$, every row equals w.

Π_N (NO instances) — either there exist two rows that are 0.1-far, or the first row is 0.1-far from every codeword in $\mathcal{W}$.

Valid encryptions correspond to the YES instances, while NO instances will correspond to "far" ciphertexts. To analyze the success probability of an adversary, we examine each ciphertext ψ it outputs with some underlying matrix $\vec{M}$ of plaintexts (which may be a YES or a NO instance or neither) and show that both NMDec and NMDec^* agree on ψ with high probability. To facilitate the analysis, we consider two cases:

- If $\vec{M} \in \Pi_N$, then it fails the column/codeword checks in both decryption algorithms with high probability, in which case both decryption algorithms output $\perp$. Specifically, if there are two rows that are 0.1-far, then column check rejects $\vec{M}$ with probability $1 - 0.9^t$. On the other hand, if the first row is 0.1-far from every codeword, then the decoding check in NMDec rejects $\vec{M}$ with probability 1 and the codeword check in NMDec^* rejects $\vec{M}$ with probability at least $1 - 0.9^t$; that is, with probability at least $1 - 2 \cdot 0.9^t$, both codeword checks in NMDec and NMDec^* rejects $\vec{M}$.

- If $\vec{M} \notin \Pi_N$, then both decryption algorithms always output the same answer for all choices of the set S, provided there is no forgery. Fix $\vec{M} \notin \Pi_N$ and a set S. The first row is 0.9-close to codeword $w \in \mathcal{W}$ and we know in addition that every other row is 0.9-close to the first row and thus 0.8-close to w. Since the underlying RPE has parameters $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$ and thereby corrects up to 0.25 fraction errors, we will recover the same codeword w and message m whether we decode the first row within distance 0.1, or any other row within distance 0.2. This means that the codeword checks in both decryption algorithms compare the first row with the same codeword w. As such, both decryption algorithms output $\perp$ with exactly the same probability, and whenever they do not output $\perp$, they output the same message m.

4.3 Proof of Main Theorem

In the hybrid argument, we consider the following variants of NME_b as applied to Π, where $\textsc{vksig}^*$ denotes the verification key in the ciphertext $y = \mathsf{NMEnc}_{\text{PK}}(m_b)$:

Experiment $\mathsf{NME}_b^{(1)}$: $\mathsf{NME}_b^{(1)}$ proceeds exactly like NME_b, except we replace `sig-check` in NMDec with `sig-check`*:

> (`sig-check`*) Verify the signature with $\mathsf{VerSig}_{\textsc{vksig}}[\vec{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\textsc{vksig} = \textsc{vksig}^*$.

Experiment $\mathsf{NME}_b^{(2)}$: $\mathsf{NME}_b^{(2)}$ proceeds exactly like $\mathsf{NME}_b^{(1)}$ except we replace NMDec with NMDec^*:

> $\mathsf{NMDec}^*_{\text{SK}}([\vec{c}, \textsc{vksig}, \sigma])$:
>
> 1. (`sig-check`*) Verify the signature with $\mathsf{VerSig}_{\textsc{vksig}}[\vec{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\textsc{vksig} = \textsc{vksig}^*$.
> 2. (`decoding-check`*) Let $\vec{c} = (c_{i,j})$ and $\textsc{vksig} = (v_1, \ldots, v_k)$. Let i be the smallest value such that $v_i \neq v_i^*$. Compute $s_j = \mathsf{Dec}_{\text{SK}_{i,j}^{v_i}}(c_{i,j})$, $j = 1, \ldots, 2t$. Compute $w = (w_1, \ldots, w_{2t}) \in \mathcal{W}$ that is at least 0.8-close to $(s_1, \ldots, s_{2t})$. If no such codeword exists, output $\perp$.
> 3. (`column-check`*) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}_{1,j}^{v_1}}(c_{1,j}) = \mathsf{Dec}_{\text{SK}_{2,j}^{v_2}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\text{SK}_{k,j}^{v_k}}(c_{k,j})$.
> 4. (`codeword-check`*) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}_{1,j}^{v_1}}(c_{1,j}) = w_j$.
>
> If all the checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

Claim 1. *For $b \in \{0,1\}$, we have* $\left\{\mathsf{NME}_b(\Pi, A, k, p(k))\right\} \overset{c}{\approx} \left\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\right\}$

Proof. This follows readily from the security of the signature scheme. □

Claim 2. *For $b \in \{0,1\}$, we have* $\left\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\right\} \overset{s}{\approx} \left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\}$

Proof. We will show that both distributions are statistically close for all possible coin tosses in both experiments (specifically, those of NMGen, A and NMEnc) except for the choice of S in NMGen. Once we fix all the coin tosses apart from the choice of S, the output $(\psi_1, \ldots, \psi_{p(k)})$ of A_2 are completely determined and identical in both experiments. We claim that with probability $1 - 2 \cdot p(k) \cdot 0.9^t = 1 - 2 \cdot p(k) \cdot 0.9^{\log^2 k} = 1 - \mathsf{negl}(k)$ over the choice of S, the decryptions of $(\psi_1, \ldots, \psi_{p(k)})$ agree in both experiments. This follows from the analysis of the promise problem in Section 4.2. □

Claim 3. *For every ppt machine A, there exists a ppt machine B such that for $b \in \{0,1\}$,*

$$\left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\} \equiv \left\{\mathsf{mIND}_b(E, B, k, k \cdot t)\right\}$$

Proof. The machine B is constructed as follows: B participates in the experiment mIND_b (the "outside") while internally simulating $A = (A_1, A_2)$ in the experiment $\mathsf{NME}_b^{(2)}$.

- (pre-processing) Pick a random subset $S = \{u_1, \ldots, u_t\}$ of $[2t]$ and run $\mathsf{GenSig}(1^k)$ to generate $(\text{SKSIG}^*, \text{VKSIG}^*)$ and set $(v_1^*, \ldots, v_k^*) = \text{VKSIG}^*$. Let ϕ be a bijection identifying $\{(i,j) \mid i \in [k], j \in [2t] \setminus S\}$ with $[k \cdot t]$.

- (key generation) B receives $\langle \text{PK}_1, \ldots, \text{PK}_{k \cdot 2t} \rangle$ from the outside and simulates NMGen as follows: for all $i \in [k], j \in [2t], \beta \in \{0,1\}$,

$$(\text{PK}_{i,j}^{\beta}, \text{SK}_{i,j}^{\beta}) = \begin{cases} (\text{PK}_{\phi(i,j)}, \perp) & \text{if } \beta = v_i^* \text{ and } j \notin S \\ \mathsf{Gen}(1^k) & \text{otherwise} \end{cases}$$

- (message selection) Let (m_0, m_1) be the pair of messages A_1 returns. B chooses $(\alpha_1, \ldots, \alpha_t) \leftarrow \mathrm{GF}(2^n)^t$ uniformly at random. and then computes $(w_1^0, \ldots, w_{2t}^0) \leftarrow \mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), m_0)$, $(w_1^1, \ldots, w_{2t}^1) \leftarrow \mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), m_1)$. Recall that R is the reconstruction algorithm of the underlying RPE scheme, and for $j \in S$, we have $w_j^0 = w_j^1$ whose value is a member of $\{\alpha_1, \ldots, \alpha_t\}$. For $j \in S$, let $\gamma_j = w_j^0 = w_j^1$.

  B sets $m_{\phi(i,j)}^b = w_j^b$, for $i \in [k], j \in [2t] \setminus S$ and forwards $(\langle m_1^0, \ldots, m_{k \cdot t}^0 \rangle, \langle m_1^1, \ldots, m_{k \cdot t}^1 \rangle)$ to the outside.

- (ciphertext generation) B receives $\langle y_1, \ldots, y_{k \cdot t} \rangle$ from the outside (according to the distribution $\mathsf{Enc}_{\text{PK}_1}(m_1^b), \ldots, \mathsf{Enc}_{\text{PK}_{k \cdot t}}(m_{k \cdot t}^b))$ and generates a ciphertext $[\vec{c}, \text{VKSIG}^*, \sigma]$ as follows:

$$c_{i,j} = \begin{cases} y_{\phi(i,j)} & \text{if } j \notin S \\ \mathsf{Enc}_{\text{PK}_{i,j}^{v_i^*}}(\gamma_j) & \text{otherwise} \end{cases}$$

  B then computes the signature $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}^*}(\vec{c})$ and forwards $[\vec{c}, \text{VKSIG}^*, \sigma]$ to A_2. It is straight-forward to verify that $[\vec{c}, \text{VKSIG}^*, \sigma]$ is indeed a random encryption of m_b under Π.

- (decryption) Upon receiving a sequence of ciphertexts $(\psi_1, \ldots, \psi_{p(k)})$ from A_2, B decrypts these ciphertexts using NMDec^* as in $\mathsf{NME}_b^{(2)}$. Note that to simulate NMDec^*, it suffices for B to possess the secret keys $\{\text{SK}_{i,j}^{\beta} \mid \beta = 1 - v_i^* \text{ or } j \in S\}$, which B generated by itself.

□

Combining the three claims, we conclude that for every ppt adversary A, there is a ppt adversary B such that for $b \in \{0,1\}$,

$$\begin{aligned} \Big\{\mathsf{NME}_b(\Pi, A, k, p(k))\Big\} &\overset{c}{\approx} \Big\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\Big\} \\ &\overset{s}{\approx} \Big\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\Big\} \equiv \Big\{\mathsf{mIND}_b(E, B, k, k \cdot t)\Big\} \end{aligned}$$

By Proposition 1, $\mathsf{mIND}_0(E, B, k, k \cdot t) \overset{c}{\approx} \mathsf{mIND}_1(E, B, k, k \cdot t)$, which concludes the proof of Theorem 1.

5 Achieving Bounded-CCA2 Non-Malleability

We describe how our scheme may be modified to achieve non-malleability under a bounded-CCA2 attack. Informally, an encryption scheme is non-malleable against a q-bounded CCA2 attack if the adversary is allowed to query Dec at most $q(k)$ times in the non-malleable experiment (but it must not query Dec on the challenge ciphertext). The modification is the straight-forward analogue of the [CHH$^+$07] modification of the [PSV06] scheme: We change the parameter $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$ of the underlying RPE scheme such that $t = 8(\log^2 k + q(k))$. See Figure 4 for more details.

We analyze security of the encryption scheme using the hybrid argument. We define the following hybrid experiments as before.

- Experiment $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$: $\mathsf{NME}_b^{(1)}$ proceeds exactly like $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b$, except we replace `sig-check` in NMDec with `sig-check`*:

  (`sig-check`*) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}[\vec{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\text{VKSIG} = \text{VKSIG}^*$.

- Experiment $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}$: $\mathsf{NME}_b^{(2)}$ proceeds exactly like $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$ except we replace NMDec with NMDec^*:

  $\mathsf{NMDec}^*_{\text{SK}}([\vec{c}, \text{VKSIG}, \sigma])$:

  1. (`sig-check`*) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}[\vec{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\text{VKSIG} = \text{VKSIG}^*$.
  2. (`decoding check`*) Let $\vec{c} = (c_{i,j})$ and $\text{VKSIG} = (v_1, \ldots, v_k)$. Let i be the smallest value such that $v_i \neq v_i^*$. Compute $s_j = \mathsf{Dec}_{\text{SK}^{v_i}_{i,j}}(c_{i,j})$, $j = 1, \ldots, 2t$ and $w = (w_1, \ldots, w_{2t}) \in \mathcal{W}$ that is at least 0.8-close to $(s_1, \ldots, s_{2t})$. If no such codeword exists, output $\perp$.
  3. (`column-check`*) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}^{v_1}_{1,j}}(c_{1,j}) = \mathsf{Dec}_{\text{SK}^{v_2}_{2,j}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\text{SK}^{v_k}_{k,j}}(c_{k,j})$.
  4. (`codeword-check`*) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}^{v_1}_{1,j}}(c_{1,j}) = w_j$.

  If all the checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

We note that $\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b(\Pi, A, k, p(k))\right\}$ and $\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}(\Pi, A, k, p(k))\right\}$ are computaionally indistinguishable for each $b \in \{0, 1\}$, which can be argued based on security of the signature scheme as in Claim 1. Moreover, $\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}(\Pi, A, k, p(k))\right\}$ and $\left\{\mathsf{mIND}_b(E, B, k, k \cdot t)\right\}$ are identically distributed for each $b \in \{0, 1\}$, which can be shown using the reduction in the proof of Claim 3. (Recall that the value $p(k)$ in the various $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}$ experiments corresponds to the number of (mauled) ciphertexts that the adversary would come up with, after given the challenge ciphertext.) Therefore, we are only left to show the following claim to conclude the analysis.

Claim 4. *For $b \in \{0, 1\}$, we have*

$$\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}(\Pi, A, k, p(k))\right\} \overset{s}{\approx} \left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}(\Pi, A, k, p(k))\right\}$$

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme, $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a strong one-time signature scheme, and $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be a reconstructable probabilistic encoding scheme with parameters $(n, 2t, 0.5, t, \mathrm{GF}(2^n))$ where $t = 8(\log^2 k + q(k))$.

$\mathsf{NMGen}(1^k)$:

1. For $i \in [k], j \in [2t], b \in \{0, 1\}$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\mathrm{PK}^b_{i,j}, \mathrm{SK}^b_{i,j})$.
2. Pick a random subset $S \subset [2t]$ of size t.
3. Set $\mathrm{PK} = \Big\{(\mathrm{PK}^0_{i,j}, \mathrm{PK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$ and $\mathrm{SK} = \Big\{S, (\mathrm{SK}^0_{i,j}, \mathrm{SK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$.

$\mathsf{NMEnc}_{\mathrm{PK}}(m)$:

1. Compute $(s_1, \ldots, s_{2t}) \leftarrow \mathsf{E}(m)$, where $m \in \{0, 1\}^n$.
2. Run $\mathsf{GenSig}(1^k)$ to generate $(\mathrm{SKSIG}, \mathrm{VKSIG})$. Let $(v_1, \ldots, v_k)$ be the binary representation of VKSIG.
3. Compute the ciphertext $c_{i,j} \leftarrow \mathsf{Enc}_{\mathrm{PK}^{v_i}_{i,j}}(s_j)$, for $i \in [k], j \in [2t]$.
4. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SKSIG}}(\vec{c})$ where $\vec{c} = (c_{i,j})$.
5. Output the tuple $[\vec{c}, \mathrm{VKSIG}, \sigma]$.

$\mathsf{NMDec}_{\mathrm{SK}}([\vec{c}, \mathrm{VKSIG}, \sigma])$:

1. (`sig-check`) Verify the signature with $\mathsf{VerSig}_{\mathrm{VKSIG}}[\vec{c}, \sigma]$.
2. (`decoding-check`) Let $\vec{c} = (c_{i,j})$ and $\mathrm{VKSIG} = (v_1, \ldots, v_k)$. Compute $s_j = \mathsf{Dec}_{\mathrm{SK}^{v_1}_{1,j}}(c_{1,j})$, $j = 1, \ldots, 2t$. Compute $((w_1, \ldots, w_{2t}), m) \leftarrow \mathsf{D}(s_1, \ldots, s_{2t})$. If the decoding fails or $(w_1, \ldots, w_{2t})$ is not 0.9-close to $(s_1, \ldots, s_{2t})$, then output $\perp$.
3. (`column-check`) For all $j \in S$, check that $\mathsf{Dec}_{\mathrm{SK}^{v_1}_{1,j}}(c_{1,j}) = \mathsf{Dec}_{\mathrm{SK}^{v_2}_{2,j}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\mathrm{SK}^{v_k}_{k,j}}(c_{k,j})$.
4. (`codeword-check`) For all $j \in S$, check that $s_j = w_j$.
5. If all the checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

Figure 4: The Bounded-CCA2 Non-Malleable Encryption Scheme Π

Proof. Let $q = q(k)$ and for a ciphertext c, let $\vec{M}_c$ denote the underlying plaintext matrix of c.

As before, we will show that both distributions are statistically close for all possible coin tosses in both experiments (specifically, those of NMGen, A and NMEnc) except for the choice of S in NMGen. Fix all the coin tosses apart from the choice of S. Here, however, unlike the case of chosen plaintext attacks, we cannot immediately deduce that the outputs of A_2 in both experiments are completely determined and identical, since they depend on the adaptively chosen queries to NMDec, and the answers depend on S. Still, the choice of S only affects whether the consistency checks accept or not; therefore, for each query, the number of possible responses of NMDec is at most two (since we fixed all the coin tosses except S). Moreover, if a query c is such that $\vec{M}_c \in \Pi_N$, NMDec will give only one response of $\perp$ with overwhelming probability, according to the analysis in Section 4.2.

This leads us to consider a binary tree of depth q that corresponds informally to "unrolling" the q adaptive queries that A makes to NMDec in the experiment $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$. The root node of the tree corresponds to the first query A makes to NMDec, and each edge from a node to its child is labeled with the answer of NMDec to the node's query. In particular, the tree is inductively built as follows:

- When A makes a query c with $\vec{M}_c \in \Pi_N$, we only consider the computation path corresponding to NMDec responding with $\perp$.
- When A makes a query c with $\vec{M}_c \notin \Pi_N$, we consider two computation paths, that is, one case of NMDec responding with a valid decryption (in which case the value returned is independent of S) and the other case of NMDec responding with $\perp$.
- The query at an internal node (except the root) corresponds to the query that A makes when following the computation path from the root to the node while NMDec's answers correspond to the labels of the edges in the path. Each leaf node contains $p(k)$ ciphertexts output by A at the end of the experiment.

Observe that the construction of the computation tree is completely deterministic and independent of the choice of S. Moreover, since A makes at most q adaptive queries to NMDec, the total number of ciphertexts in the tree is at most $2^{q+1}p(k)$. The claim follows from combining the following two observations:

- Let $\mathsf{good}(S)$ be an event in which given the choice S, for every ciphertext c in the tree such that $\vec{M}_c \in \Pi_N$, both NMDec and NMDec^* output $\perp$. We have

  $$\Pr_S[\mathsf{good}(S)] \geq 1 - 2 \cdot (2^{q+1} \cdot p(k)) \cdot 0.9^t = 1 - 2^{q+2} \cdot p(k) \cdot 0.9^{8(\log^2 k + q)} = 1 - \mathsf{negl}(k).$$

  This follows from a union bound over these ciphertexts in the tree and the analysis in Section 4.2.
- For every S such that $\mathsf{good}(S)$ is true, the outputs in both experiments are the same. This follows readily by induction on the queries made by A, and using the fact both NMDec and NMDec^* always output the same answer for any $\vec{M} \notin \Pi_N$ as explained in Section 4.2.

□

Remark on achieving (full) CCA2 security. It should be clear from the preceding analysis that the barrier to obtaining full CCA2 security lies in handling queries outside Π_N. Specifically, with even just a (full) CCA1 attack, an adversary could query NMDec on a series of adaptively chosen ciphertexts corresponding to matrices outside Π_N to learn the set S upon which it could readily break the security of our construction.

Acknowledgments. This work was initiated while the third and fourth authors were visiting IPAM. We thank Vinod Vaikuntanathan for sharing his insights on non-malleability, and Oded Goldreich for pointing out [DGR99], and for other helpful suggestions.

References

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 103–112, 1988.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988.

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil P. Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 283–298, 1998.

[BW86] Elwyn R. Berlekamp and Lloyd R. Welch. Error correction for algebraic block codes, 1986. US Patent 4,633,470.

[CDMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *Proceedings of the 5th Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 427–444, 2008.

[CHH$^+$07] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, 2007.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, 2004.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, 1998.

[CS04] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 45–64, 2004.

[Dac14] Dana Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme. In *PKC 2014: 17th International Workshop on Theory and Practice in Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 37–55. Springer, 2014.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DGR99] Scott E. Decatur, Oded Goldreich, and Dana Ron. Computational sample complexity. *SIAM Journal on Computing*, 29(3):854–879, 1999.

[DMRY09] Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. Efficient robust private set intersection. In *Applied Cryptography and Network Security, 7th International Conference, ACNS 2009*, volume 5536 of *Lecture Notes in Computer Science*, 2009.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360, 2004.

[ES02] Edith Elkind and Amit Sahai. A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/024, 2002. http://eprint.iacr.org/.

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE Computer Society Press, 2012.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In *Proceedings of the 4th Theory of Cryptography Conference, TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455, 2007.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary Version in 17th STOC, 1985.

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *The 42th Annual Symposium on Foundations of Computer Science*, pages 126–135, 2001.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *Journal of ACM*, 38(3):691–729, 1991. Preliminary Version in 27th FOCS, 1986.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Volume II, Basic Applications.* Cambridge University Press, 2004.

[GOSV14] Vipul Goyal, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Black-box non-black-box zero knowledge. In *46th Annual ACM Symposium on Theory of Computing*, pages 515–524. ACM Press, 2014.

[Hai08] Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In *Proceedings of the Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 412–426, 2008.

[HK09] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 313–332. Springer, 2009.

[HLW12] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 663–681. Springer, 2012.

[IKLP06] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 99–108, 2006.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 44–61, 1989.

[Kiy14] Susumu Kiyoshima. Round-efficient black-box construction of composable multi-party computation. In *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 351–368. Springer, 2014.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 673–692. Springer, 2010.

[KMO14] Susumu Kiyoshima, Yoshifumi Manabe, and Tatsuaki Okamoto. Constant-round black-box construction of composable multi-party computation protocol. In *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 343–367. Springer, 2014.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.

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

[LP07] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 52–78, 2007.

[LP12] Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 461–478. Springer, 2012.

[MH14] Takahiro Matsuda and Goichiro Hanaoka. Chosen ciphertext security via point obfuscation. In *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 95–120. Springer, 2014.

[Ms09] Steven Myers and abhi shelat. Bit encryption is complete. In *50th Annual Symposium on Foundations of Computer Science*, pages 607–616. IEEE Computer Society Press, 2009.

[MSs12] Steven Myers, Mona Sergi, and abhi shelat. Blackbox construction of a more than non-malleable CCA1 encryption scheme from plaintext awareness. In *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 2012.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 427–437, 1990.

[PSV06] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 271–289, 2006.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 187–196, 2008.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *Proceedings of the 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 403–418, 2009.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 387–394, 1990.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, 1992.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2009.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *Proceedings of the First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, 2004.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *The 40th Annual Symposium on Foundations of Computer Science*, page 543, 1999.

[Wee10a] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51st Annual Symposium on Foundations of Computer Science*, pages 531–540. IEEE Computer Society Press, 2010.

[Wee10b] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 314–332. Springer, 2010.

A Non-Malleable Encryption with Reed-Solomon Code

We describe a non-malleable encryption scheme $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$, where $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a semantically secure encryption scheme by directly using a Reed-Solomon code instead of using an RPE scheme. We summarize the description in Figure 5.

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme, $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a strong one-time signature scheme. Let k be the security parameter and $t = \log^2 k$.

$\mathsf{NMGen}(1^k)$:

1. For $i \in [k], j \in [2t], b \in \{0, 1\}$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\text{PK}^b_{i,j}, \text{SK}^b_{i,j})$.
2. Pick a random subset $S \subset [2t]$ of size t.
3. Set $\text{PK} = \Big\{(\text{PK}^0_{i,j}, \text{PK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$, $\text{SK} = \Big\{S, (\text{SK}^0_{i,j}, \text{SK}^1_{i,j}) \mid i \in [k], j \in [2t]\Big\}$.

$\mathsf{NMEnc}_{\text{PK}}(m)$:

1. Pick random $\alpha_1, \ldots, \alpha_t \in \text{GF}(2^n)$ and set $p(x) = m_0 + \alpha_1 x + \ldots + \alpha_t x^t$. Set $s_j = p(j)$ for $j \in [2t]$.
2. Run $\mathsf{GenSig}(1^k)$ to generate $(\text{SKSIG}, \text{VKSIG})$. Let $(v_1, \ldots, v_k)$ be the binary representation of VKSIG.
3. Compute the ciphertext $c_{i,j} \leftarrow \mathsf{Enc}_{\text{PK}^{v_i}_{i,j}}(s_j)$, for $i \in [k], j \in [2t]$.
4. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}}(\vec{c})$ where $\vec{c} = (c_{i,j})$.
5. Output the tuple $[\vec{c}, \text{VKSIG}, \sigma]$.

$\mathsf{NMDec}_{\text{SK}}([\vec{c}, \text{VKSIG}, \sigma])$:

1. (`sig-check`) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}[\vec{c}, \sigma]$.
2. (`decoding-check`) Let $\vec{c} = (c_{i,j})$ and $\text{VKSIG} = (v_1, \ldots, v_k)$. Compute $s_j = \mathsf{Dec}_{\text{SK}^{v_1}_{1,j}}(c_{1,j})$ for $j = 1, \ldots, 2t$. Compute the codeword $w = (w_1, \ldots, w_{2t}) \in \mathcal{W}$ that agrees with $(s_1, \ldots, s_{2t})$ in at least $1.8t$ positions. If no such codeword exists, output $\perp$.
3. (`column-check`) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}^{v_1}_{1,j}}(c_{1,j}) = \mathsf{Dec}_{\text{SK}^{v_2}_{2,j}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\text{SK}^{v_k}_{k,j}}(c_{k,j})$.
4. (`codeword-check`) For all $j \in S$, check that $s_j = w_j$.
5. If all checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

Figure 5: The Non-Malleable Encryption Scheme with a Reed-Solomon Code