J Cryptol (2018) 31:172–201
https://doi.org/10.1007/s00145-017-9254-z

Journal of
CRYPTOLOGY

A Black-Box Construction of Non-malleable Encryption from Semantically Secure Encryption*

Seung Geol Choi
US Naval Academy, Annapolis, MD, USA
choi@usna.edu

Dana Dachman-Soled
University of Maryland, College Park, MD, USA
danadach@ece.umd.edu

Tal Malkin
Columbia University, New York, NY, USA
tal@cs.columbia.edu

Hoeteck Wee
CNRS-DIENS, École Normale Supérieure, Paris, France
wee@di.ens.fr

Communicated by Eike Kiltz.

Received 6 November 2014 / Revised 30 October 2016
Online publication 13 March 2017

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.

Keywords. Public-key encryption, Non-malleable encryption, Black-box construction.

[*] An extended abstract [7] appeared in TCC 2008 under the title "Black-Box Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One"

1. Introduction

The most basic security requirement for public-key encryption (PKE) schemes, referred to as semantic security, is that an eavesdropping adversary does not learn anything about the plaintext underlying a communicated ciphertext; this security notion is proven to be equivalent to indistinguishability under a chosen-plaintext attack (IND-CPA) which requires that the adversary cannot distinguish an encryption of one plaintext from another [22,42][1]. In many applications, however, this indistinguishability guarantee is not sufficient, and a PKE satisfying the stronger notion of *non-malleability* [16] is required. Roughly, non-malleability requires that it is infeasible for an adversary to modify or *maul* 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. In order to achieve privacy of each buyer's bid against other buyers, 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. Although this auction protocol seems to be secure enough, it turns that we must also rule out an adversary who consistently bids exactly one dollar more than the previous bidders by simply mauling the ciphertexts from the 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?

We focus on this question for a passive adversary, and when we refer to "non-malleable encryption," we mean, by default, non-malleability under a 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 [46] studied this question and answered it affirmatively, providing a beautiful construction of a non-malleable encryption scheme from any semantically secure one (building on [16]). However, this PSV construction – as with previous constructions achieving non-malleability from general assumptions [16,38,53] – 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.

1.1. *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:

[1] In this work, we will use semantic security and IND-CPA security interchangeably, since the two notions are equivalent.

- *Black-box usage* A construction G is black box if it refers only to the input/output behavior of the underlying primitive f; we would typically also require the proof of security to show an efficient reduction that converts any (even inefficient) adversary A breaking the security of the construction G^f into an efficient algorithm $S^{A,f}$ breaking the underlying primitive with oracle access to the adversary A and the primitive f (this is called a fully black-box reduction—see [1,52] and references within for more details).
- *Non-black-box usage* In a non-black-box usage, a construction and/or its security proof uses the code computing the functionality of the underlying primitive.

Motivated by the fact that the majority of constructions in cryptography are black box, a rich and fruitful body of work initiated in [33] seeks to understand the power and limitations of black-box constructions in cryptography, resulting in a fairly complete picture of the relations among 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 [29,32] show a black-box construction from the minimal primitive of semi-honest oblivious transfer. Other examples include [21,28,54].

The question of whether we can securely realize a task via black-box access to a general primitive is of theoretical interest, toward 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 in 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.

1.2. *Our Contributions*

1.2.1. *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 [46], or prior to that, only using additional assumptions [16].

> **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 algorithms (e.g., Berlekamp–Welch algorithm [6]). The running time incurs a multiplicative overhead that is quasi-linear in the security parameter, over the running time of the underlying IND-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 [46], entirely self-contained (apart from some basic tools from coding theory). We do not need to appeal to notions of zero-knowledge [24,26], 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 even when instantiated with an encryption scheme with negligible decryption error. This is in contrast to the [16] and [46] constructions, which require that the underlying encryption scheme be first "immunized" against decryption errors (c.f. [19]); these constructions are otherwise susceptible to an attack described by Dwork et al. [19].

1.2.2. *Our Techniques*

At a high level, we follow the cut-and-choose approach for consistency checks from [46], 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, which we will explain later, with certain locally testable and self-correcting properties. 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 1.5).

1.2.3. *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., [16,27,45,50]). Finally, of particular relevance to us is the notion of bounded CCA2 attack, introduced by Cramer et al. [10], which is a relaxation of the CCA2 attack (and incomparable to CCA1). Here, the adversary is only allowed to 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.

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

Fig. 1. Summary of our positive results.

It is known that although indistinguishability and non-malleability are equivalent security notions under a CCA2 attack [16], non-malleability under a bounded CCA2 attack (NM-q-CCA2) is a strictly stronger security notion than indistinguishability under a bounded CCA2 attack (IND-q-CCA1); that is, every NM-q-CCA2 secure encryption is also IND-q-CCA2 secure, but the converse is not necessarily true [10].

Cramer et. al. [10] 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. Interestingly, the scheme is just the construction of [46], only except that the NIZK proof used is with stronger soundness (i.e., the soundness holds even if the adversary can query the verifier on at most q proofs and learn the validity of each proof).

Combining their approach for the latter construction with our main result (i.e., by using our NM-CPA construction but with a stronger parameter for the cut-and-choose check), 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 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 Fig. 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 et al. [23] 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:

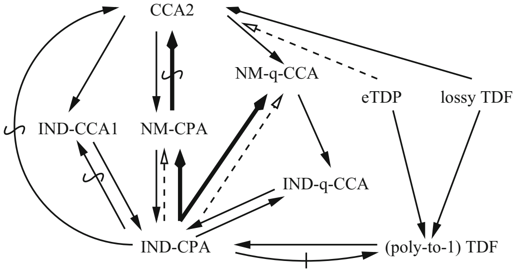

Fig. 2. Known relations among generic encryption primitives and our results. *Solid lines* indicate black-box constructions, and *dotted lines* indicate non-black-box constructions (c.f. [5,10,16,46,47]). *Arrows* with the '|' symbol (resp., the '∼' symbol) in the middle indicate the separations with respect to black-box reductions (resp., black-box shielding reductions, c.f. [23,25]). Our contributions are indicated with the *thick arrows*.

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 [23] 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 Fig. 2.

1.3. *Overview of Our Construction*

In order to prove that an encryption scheme satisfies NM-CPA, we must show that the *decryptions* of ciphertexts produced by any efficient adversary, A, upon receiving a challenge ciphertext encrypting either m_0 or m_1 are computationally indistinguishable. The high-level proof structure is to consider a sequence of *hybrid* distributions, where the first hybrid distribution corresponds to the decryptions of ciphertexts produced by the adversary, A, upon receiving an encryption of m_0, the last hybrid distribution corresponds to decryptions of ciphertexts produced by the adversary, A, upon receiving an encryption of m_1, and any two consecutive hybrid distributions are proven to be computationally indistinguishable. Thus, the difficulty in proving NM-CPA security is that in order to produce the correct distributions, one must implement a (modified) decryption oracle in each hybrid, which performs a *single* parallel decryption on all ciphertexts produced by the adversary. Recall that when proving indistinguishability of consecutive hybrids, we *reduce* a distinguisher between the hybrids to some underlying assumption and therefore that *reduction* must internally simulate the (modified) decryption oracle in each pair of consecutive hybrids. However, this produces something of a paradox, because when building NM-CPA encryption from IND-CPA encryption, it must be the case that indistinguishability of (at least) one pair of consecutive hybrids reduces to the security of the underlying IND-CPA encryption scheme. But this means that the reduction must be able to simulate the (modified) decryption oracle in this pair of consecutive hybrids, without knowing the underlying secret key. Even more puzzling, the reduction should *not know* the plaintext underlying the challenge ciphertext submitted to the adversary (so that a distinguishing adversary is useful), but should be able to decrypt *any other* ciphertext produced by the adversary, even after the adversary sees the challenge ciphertext!

The approach for solving this problem prior to our work (used by DDN [16] and later PSV [46]) was the following: To encrypt a message m under the NM-CPA encryption scheme, a set of k public keys of the underlying encryption scheme is chosen (where k is security parameter) and the same message m is encrypted k times, once under each public key in the set. The set of public keys under which to encrypt is chosen cleverly so that the following property is guaranteed: For the challenge ciphertext the reduction does not know any of the corresponding secret keys and so cannot decrypt it, while for any valid ciphertext produced by the adversary, the reduction must know at least one of the corresponding secret keys. In addition, for any ciphertext, there is a way to check whether the ciphertext is valid without decrypting and *without learning the underlying plaintext*. Given the above, the reduction implements the decryption oracle by first checking for validity and if the check passes, outputting the decryption corresponding to (one of) the secret key(s) that it knows. If the ciphertext is invalid, both the reduction and the real decryption oracle output $\perp$. On the other hand, if the ciphertext is valid (i.e., the same message m is encrypted under each of the k underlying public keys), again both the reduction and the real decryption oracle output the same message, regardless of which underlying secret key is used for decryption.

In more detail, recall the DDN [16] and PSV [46] constructions: A public key consists of k pairs of IND-CPA secure public keys $\textsc{pk} = \left\{(\textsc{pk}_i^0, \textsc{pk}_i^1) \mid i \in [k]\right\}$. To encrypt a message, one (a) generates a ($\textsc{sksig}$, $\textsc{vksig}$) pair using a one-time signature scheme, (b) generates k encryptions of the same message under independent keys, where the i-th encryption is done under public key $\textsc{pk}_i^{\textsc{vksig}_i}$, where $\textsc{vksig}_i$ denotes the i-th bit of the verification key, (c) gives a non-interactive zero-knowledge proof that all resulting ciphertexts are encryptions of the same message, and (d) signs the entire bundle with $\textsc{sksig}$. Note that due to unforgeability of the one-time signature scheme, the $\textsc{vksig}$ corresponding to a valid ciphertext produced by the adversary must be different from the $\textsc{vksig}$ used in the challenge ciphertext (otherwise, the signature on the entire bundle will fail to verify). This, in turn, means that the set of public keys corresponding to any valid ciphertext produced by the adversary differs from the set of public keys corresponding to the challenge ciphertext. This property allows us to build a reduction which does not know any of the secret keys corresponding to the challenge ciphertext, while for any valid ciphertext produced by the adversary, the reduction knows at least one of the corresponding secret keys (as described above). Moreover, the publicly verifiable signature and non-interactive zero-knowledge proof allow to check for ciphertext validity without decrypting or knowing the underlying plaintext (as described above). Note that it is in step (c) that a general NP-reduction is used, which in return makes the construction non-black box.

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 was previously used in [39] to eliminate zero-knowledge proofs in the context of secure two-party computation), namely decrypt and verify that some random, constant fraction, say $k/2$ of the ciphertexts are indeed consistent. This would mean that the reduction need only know $k/2$ of the corresponding secret keys in order to check for validity of ciphertexts and the other $k/2$ public keys can potentially be used for the reduction to IND-CPA security. Unfortunately, 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—as discussed above—it is crucial for the proof that we can enforce consistency while learning nothing about the underlying plaintext.

We circumvent both issues by using a more sophisticated encoding of the message m based on reconstructable probabilistic encoding (RPE) schemes[2] that we introduce, instead of merely making k copies of the message as in the above schemes. RPE schemes are, informally, error-correcting codes with additional secrecy and reconstruction properties. The secrecy property guarantees that the symbols at any not-too-large subset of positions in the codeword are distributed uniformly and independently of the encoded message. The reconstruction property says that furthermore, any assignment of symbols to such a subset of positions can be completed to a (correctly distributed) codeword for any given message. The parameter regime we will be interested in is the standard one, where the error correction is with respect to a constant fraction of errors, and the secrecy and reconstruction are also with respect to a (smaller) constant fraction of positions.

Specifically, let E be the encoding algorithm of the RPE scheme with output of length $\ell = O(k)$ (over the alphabet of the scheme). We first obtain an encoding w of m (i.e., $w \leftarrow \mathsf{E}(m)$) and then generate k encryptions of the same w. Thus, we construct a $k \times \ell$ matrix such that entry (i, j) holds w_j (i.e., the jth element of w). 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 first issue above—that it is difficult to detect a tiny number of inconsistent ciphertexts—is now handled using the error-correcting properties of the encoding scheme, which loosely speaking, guarantees that a small number of inconsistent ciphertexts will not affect the value of the decrypted message. 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. [10], 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:

[2] The original work [7] used an encoding scheme based on the Reed–Solomon code, and we introduce RPE as a generalization of the encoding.

- 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.

1.3.1. *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 ℓ of columns to $O(\log^2 k)$.

1.4. *Toward 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, [7], at TCC 2008.

[7] **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 [16,38,53]. A different line of work focused on (very) efficient constructions of CCA2-secure encryptions under specific number-theoretic assumptions (c.f. [11,13,14]). Apart from the construction based on identity-based encryption [11], all these constructions can be described under the following framework (c.f. [3,20,45,50]). 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) [13,14];
- apply a general NP reduction to prove in non-interactive zero knowledge (NIZK) statements that relate to the primitive [16,38,53].

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 [47] (who also do not use the above framework) made substantial progress toward 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. 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 toward 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. [46]). Indeed, the main obstacle toward achieving full CCA2 security from either semantically secure encryptions or enhanced trapdoor permutations using our approach (and also the [46] 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 [31,43]) that uses re-encryption may help overcome this obstacle.

1.4.1. *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 [51] 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 et al. [35] 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 [30] presented the first construction of CCA2-secure encryption from hardness of factoring. Wee [55] abstracted their construction and introduced a new primitive, extractable hash proofs, which is sufficient for CCA2-secure encryption. Moreover, [55] 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 [8,12,31,43] showed how to obtain multi-bit CCA2-secure encryption from single-bit CCA2-secure encryption. Another line of research (c.f. [15,41,44]) focused on black-box constructions of CCA2-secure encryption from various non-falsifiable assumptions.

1.5. *Other 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 [48], set intersection protocols from homomorphic encryptions [18], and a CCA2-secure encryption scheme for strings starting from one for bits [43]. The works of [34,36,40,54] used our encoding in the context of black-box, round-efficient secure computation. The works of [21,28] generalized our approach to proving relations beyond equality using verifiable secret sharing (VSS) and the paradigm of MPC-in-the-head. The work of [2] achieved a non-malleable code using our approach.

Coretti et al. [8] revisited the work of [7] and investigated the question of how efficient the black-box transformation can be. The measure of efficiency they consider is the rate of the resulting NM-CPA encryption scheme (i.e., $c(n)/n$, where $c(n)$ is the ciphertext length, and n is the plaintext length) and gave an improved transformation by replacing the error-correcting code (based on Reed–Solomon code) used in [7] with one having a better rate. In particular, they independently observed that the construction given in [7] can be generalized to work for more general linear error-correcting secret sharing schemes (LECSS), beyond just Reed–Solomon codes, and they were able to replace the Reed–Solomon code with an encoding scheme [9] with a better rate for long enough messages. We note that LECSS is similar to the RPE abstraction introduced in this paper. We will compare the two when we formally define RPE.

2. Preliminaries and Definitions

2.1. *Notation*

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.

2.2. *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 $(\textsc{pk}, \textsc{sk}) \leftarrow \mathsf{Gen}(1^k)$: for all m, $\Pr[\mathsf{Dec}_{\textsc{sk}}(\mathsf{Enc}_{\textsc{pk}}(m)) = m] = 1$.

We give the definition of indistinguishability under a chosen-plaintext attack (IND-CPA) for public-key encryption schemes. Roughly speaking, the definition requires that the adversary should not be able to distinguish the ciphertexts of any two messages that it chooses; to put it another way, no matter which encryption the adversary receives, its output will be indistinguishable.

Definition 2. (*IND-CPA 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:

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

(Gen, Enc, Dec) is indistinguishable under a chosen-plaintext 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 straightforward 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 $i = 1, \ldots, \ell$: $(\textsc{pk}_i, \textsc{sk}_i) \leftarrow \mathsf{Gen}(1^k)$
  $(\langle m_1^0, \ldots, m_\ell^0\rangle, \langle m_1^1, \ldots, m_\ell^1\rangle, \textsc{state}_A) \leftarrow A_1(\langle \textsc{pk}_1, \ldots, \textsc{pk}_\ell\rangle)$
    s.t. $|m_1^0| = |m_1^1| = \cdots = |m_\ell^0| = |m_\ell^1|$
  For $i = 1, \ldots, \ell$: $y_i \leftarrow \mathsf{Enc}_{\textsc{pk}_i}(m_i^b)$
  $D \leftarrow A_2(y_1, \ldots, y_\ell, \textsc{state}_A)$
  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:*

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

2.3. *Non-malleable Encryption*

We give the definition of non-malleability under a chosen-plaintext attack (NM-CPA) for public-key encryption schemes, following [46]. Roughly speaking, the definition requires that no matter which encryption the adversary receives, the decryption of the adversary's output ciphertexts should be indistinguishable. Recall that IND-CPA requires the adversary's *outputs* be indistinguishable. By requiring even the *decryption* of its output ciphertexts be indistinguishable, the definition captures the property that the adversary cannot modify the challenge ciphertext into other ciphertexts related to the original plaintext underlying the challenge ciphertext.

Definition 3. (*Non-malleable encryption* [46]) 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:

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

(Gen, Enc, Dec) is non-malleable under a chosen-plaintext 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 [46] 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).

2.4. *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* [10]) Let Π = (Gen, Enc, 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:

```
$\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b(\Pi, A, k, \ell)$ :
  $(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k)$
  $(m_0, m_1, \text{STATE}_A) \leftarrow A_1^{O_1}(\text{PK})$ s.t. $|m_0| = |m_1|$
  $y \leftarrow \mathsf{Enc}_{\text{PK}}(m_b)$
  $(\psi_1, \dots, \psi_\ell) \leftarrow A_2^{O_2}(y, \text{STATE}_A)$
  Output $(d_1, \dots, d_\ell)$ where $d_i = \begin{cases} \perp & \text{if } \psi_i = y \\ \mathsf{Dec}_{\text{SK}}(\psi_i) & \text{otherwise} \end{cases}$
```

(Gen, Enc, 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.

2.5. *(Strong) One-Time Signature Schemes*

A digital signature scheme consists of a triple of ppt algorithms (GenSig, Sign, VerSig) such that:

- GenSig takes the security parameter 1^k as input and generates a pair of keys: a public verification key VKSIG, and a secret signing key SKSIG.
- Sign takes as input a secret key SKSIG and a message m and generates a signature σ. We write this as $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}}(m)$.
- VerSig takes as input a verification key VKSIG, a message m, and a (purported) signature σ and outputs a single bit indicating acceptance or not.

For correctness, we require that for all ($\text{VKSIG}, \text{SKSIG}$) output by $\mathsf{GenSig}(1^k)$, for all messages m, and for all $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}}(m)$, we have $\mathsf{VerSig}_{\text{VKSIG}}(m, \sigma) = 1$.

2.5.1. *Strong One-Time Signature Schemes*

Informally, a strong one-time signature scheme is an existentially unforgeable digital 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 single 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.

Definition 5. (*Security of strong one-time signature schemes.*) Let $\mathcal{S} = (\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a digital signature scheme and let the random variable $\mathsf{Forge}(\mathcal{S}, A, k)$ where $A = (A_1, A_2)$ are ppt algorithms and $k \in \mathbb{N}$ denote the result of the following probabilistic experiment:

```
$\mathsf{Forge}(\mathcal{S}, A, k)$ :
  $(\text{VKSIG}, \text{SKSIG}) \leftarrow \mathsf{GenSig}(1^k)$
  $(m, \text{STATE}) \leftarrow A_1(\text{VKSIG})$
  $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}}(m)$
  $(m^*, \sigma^*) \leftarrow A_2(\sigma, \text{STATE})$
  If $\mathsf{VerSig}_{\text{VKSIG}}(m^*, \sigma^*)$ and $(m, \sigma) \neq (m^*, \sigma^*)$, output 1
  Otherwise, output 0.
```

A digital signature scheme $\mathcal{S}$ is strongly existentially unforgeable under a one-time chosen message attack if there exists a negligible function $\mu(\cdot)$ such that for all sufficiently large k, and for any ppt algorithm A, it holds

$$\Pr[\mathsf{Forge}(\mathcal{S}, A, k) = 1] \leq \mu(k).$$

Such schemes can be constructed in a black-box way from one-way functions [37,49], and thus from any semantically secure encryption scheme (Gen, Enc, Dec) using black-box access only to Gen.

2.6. *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 some 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 6. (*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 $n, \ell, t \in \mathbb{N}$, $0 < \delta < 1$, $t < \ell$, and Σ is an alphabet.

- The encoding algorithm E is an efficient probabilistic procedure, which takes a message $m \in \{0, 1\}^n$ as input and outputs a codeword w over Σ^ℓ. We let the code $\mathcal{W}$ be the support of E.
- The decoding algorithm D is an efficient procedure that takes a string $w' \in \Sigma^\ell$ as input and outputs a codeword w and a message m (or $(\perp, \perp)$ if it fails).
- The reconstruction algorithm R is an efficient procedure that takes input a set $S \subset [\ell]$ of size t, a partial codeword $(\alpha_1, \ldots, \alpha_t) \in \Sigma^t$, and a message $m \in \{0, 1\}^n$, and outputs a complete codeword $w \in \mathcal{W}$ consistent with the given partial codeword $(\alpha_1, \ldots, \alpha_t)$ and message m.

The three algorithms should satisfy the following requirements:

(1) Error correction: Any two strings in $\mathcal{W}$ are δ-far. For any string w' that is $(1-\delta/2)$-close to some codeword w in $\mathcal{W}$, it holds that $D(w')$ outputs w along with a message m consistent with w.

(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: For any set $S \subset [\ell]$ of size t, any $(\alpha_1, \ldots, \alpha_t) \in \Sigma^t$, and any $m \in \{0, 1\}^n$, it holds that $\mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), m)$ is identically distributed to $\mathsf{E}(m)$ with the constraint $\mathsf{E}(m)|_S = (\alpha_1, \ldots, \alpha_t)$.

2.6.1. *RPE vs. LECSS*

RPE schemes are related to the standard notion of linear error-correcting secrete sharing scheme (LECSS). In fact, LCESS's are just RPE schemes without property (3) above (and additionally with linearity). Concurrently with and independently from this work, Coretti et al. [8] observed that the original work of [7] can be extended to work also for LECSS's satisfying property (3).

2.6.2. *RPE Construction Based on a Reed–Solomon Code*

We can construct an RPE scheme with a Reed–Solomon code. We note the construction is implicit in [4].

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 GF(2^n); an integer i with $0 \leq i < 2^n$ will also be implicitly encoded into a field element in 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 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 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 $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, 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 Fig. 3, and described as follows.

3.1. *Encryption*

Let k be the security parameter and let $\{0, 1\}^n$ be the message space of Π. In addition, let δ be a real number with $0 < \delta < 1$, and t be an integer such that

$$t \geq \log^2 k.$$

Let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be an RPE scheme with parameters $(n, \ell, \delta, t, \Sigma)$. The public key for Π comprises $2k\ell$ public keys from Gen indexed by a triplet $(i, j, b) \in [k] \times [\ell] \times \{0, 1\}$; there are two keys corresponding to each entry of a $k \times \ell$ matrix. To encrypt a message $m \in \{0, 1\}^n$, we (a) compute $(s_1, \ldots, s_\ell) \leftarrow \mathsf{E}(m)$, (b) generate (SKSIG, VKSIG) for a one-time signature (let $(v_1, \ldots, v_k)$ be the binary representation of VKSIG), (c) compute a $k \times \ell$ 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 SKSIG. The ciphertext matrix $\vec{c}$ is shown below:

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a strong one-time signature scheme. Let k be the security parameter and let $\{0,1\}^n$ be the message space of Π. In addition, let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be an RPE scheme with parameters $(n, \ell, \delta, t, \Sigma)$.

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

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

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

1. Compute $(s_1, \ldots, s_\ell) \leftarrow \mathsf{E}(m)$, where $m \in \{0,1\}^n$.
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 $i \in [k], j \in [\ell]$.
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 $s_j = \mathsf{Dec}_{\text{SK}^{v_1}_{1,j}}(c_{1,j})$, $j = 1, \ldots, \ell$. Compute $((w_1, \ldots, w_\ell), m) \leftarrow \mathsf{D}(s_1, \ldots, s_\ell)$. If the decoding fails or $(w_1, \ldots, w_\ell)$ is $\frac{\delta}{4}$-far from $(s_1, \ldots, s_\ell)$, then 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 the checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

Fig. 3. The Non-Malleable Encryption Scheme Π.

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

3.2. *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 $(1 - \delta/4)$-close to the first row of the matrix; the check fails if no such w exists. Recall that the underlying RPE has parameters $(n, \ell, \delta, t, \Sigma)$, so it can correct up to $\frac{\delta}{2}$ 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.

3.3. *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 2ℓ secret keys corresponding to the first row of the matrix and $2t \cdot (k - 1)$ 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.*) Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA public-key encryption scheme. Then, $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$ from Fig. 3, instantiated with $t \geq \log^2 k$, is an NM-CPA public-key encryption scheme.*

We establish the theorem 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. In particular, we consider the following hybrids for $b = 0, 1$:

- **Experiment** $\mathsf{NME}_b(\Pi, A, k, p(k))$: It is the original non-malleability experiment defined in Section 2.3.
- **Experiment** $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$: This experiment proceeds exactly like $\mathsf{NME}_b(\Pi, A, k, p(k))$, except we replace `sig-check` in NMDec with an alternative `sig-check`*. In particular, let VKSIG* denote the verification key in the challenge ciphertext given to the adversary, and `sig-check`* rejects the input ciphertext to NMDec if it contains a verification key VKSIG such that VKSIG $=$ VKSIG*. It is easy to see that the unforgeability of the signature implies the indistinguishability between NME_b and $\mathsf{NME}_b^{(1)}$.

Due to this indistinguishability, the mauled ciphertexts from the adversary should have a verification key VKSIG different from VKSIG*, and therefore, in some row of each mauled ciphertext, the adversary should use *a fresh set of public keys that the challenge ciphertext does not use*.

- **Experiment** $\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))$: The experiment proceeds exactly like $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$ except we replace NMDec with an alternative decryption algorithm NMDec^*. As we will see, the algorithm NMDec^* will be able to simulate the decryption algorithm NMDec satisfying the two conflicting requirements:

  – NMDec^* works *without having to know the secret keys* corresponding to the public keys in Γ.

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

  Of course, designing NMDec^* is difficult precisely because NMDec uses the secret keys corresponding to the public keys in Γ. Intuitively, however, we can still design such algorithm NMDec^*, since the consistency check inspects *only the partial set of the columns*. Recall that there exists a row in which the adversary must use a fresh set of public keys that the challenge ciphertext does not use. This implies that the adversary should *fill up this row from scratch*, since the challenge ciphertext uses a different set of public keys for that row. Therefore, because of this row, it is infeasible for the adversary to create a mauled ciphertext that passes the consistency check such that the other rows are derived from the challenge ciphertext, even if the check inspects only a partial set of columns hidden to the adversary.
- **Experiment** $\mathsf{mIND}_b(\mathcal{E}, B, k, k(\ell - t))$: It is the semantic security experiment for multiple messages defined in Section 2.3. Since NMDec^* in $\mathsf{NME}_b^{(2)}$ never uses the public keys in Γ, one can reduce the security of $\mathsf{NME}_b^{(2)}$ to semantic security of the underlying encryption scheme.

In summary, we will show that for every ppt adversary A, there is a ppt adversary B such that for $b \in \{0, 1\}$,

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

By Proposition 1, $\left\{\mathsf{mIND}_0(E, B, k, k(\ell - t))\right\} \overset{c}{\approx} \left\{\mathsf{mIND}_1(E, B, k, k(\ell - t))\right\}$, which concludes the proof.

4.1. *Indistinguishability Between* NME_b *and* $\mathsf{NME}_b^{(1)}$

The experiment $\mathsf{NME}_b^{(1)}$ proceeds exactly like NME_b, except we replace `sig-check` in NMDec with an alternative `sig-check*` defined as follows:

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

1. (`sig-check*`)

```
    (a) $\boxed{\text{If VKSIG} = \text{VKSIG}^*\text{, then output } \perp}$.
    (b) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}(\vec{c}, \sigma)$.
2. ...
```

It is straightforward to show the two experiments are computationally indistinguishable.

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 unforgeability of the signature scheme. □

4.2. *Indistinguishability Between* $\mathsf{NME}_b^{(1)}$ *and* $\mathsf{NME}_b^{(2)}$

In this section, we show that $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are statistically indistinguishable:

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

4.2.1. *The Public Keys* Γ

Recall that Γ is the set of public keys whose secret keys are not available to NMDec^*. Let $\text{VKSIG}^* = (v_1^*, \ldots, v_k^*)$ denote the verification key in the challenge ciphertext given to the adversary. For each row i, NMDec^* is restricted to be able to decrypt *only one of the two sub-rows* according to the bit v_i^*; that is, Γ is defined as follows:

$$\Gamma = \{\text{PK}_{i,j}^{v_i^*} \mid i \in [k],\ j \in [\ell] \setminus S\}.$$

The reason that Γ does not contain columns in S is to allow NMDec^* to always perform the codeword check and the column check successfully.

4.2.2. *The Alternative Decryption Algorithm*

We describe the alternative decryption algorithm NMDec^* below, highlighting the difference from the algorithm NMDec in $\mathsf{NME}_b^{(1)}$ with boxes. Let $\text{VKSIG} = (v_1, \ldots, v_k)$ denote the verification key in the input ciphertext to NMDec^*. Instead of always choosing the first row to decrypt, NMDec^* chooses a row that it can decrypt without using the secret keys corresponding to the keys in Γ. In particular, NMDec^* chooses the xth row such that $v_x \neq v_x^*$. The existence of such row is guaranteed since $\text{VKSIG} \neq \text{VKSIG}^*$.

```
$\mathsf{NMDec}^*_{\text{SK}}([\vec{c}, \text{VKSIG}, \sigma])$:
 1. (sig-check*)
    (a) If VKSIG = VKSIG*, then output $\perp$.
    (b) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}(\vec{c}, \sigma)$.
 2. (decoding-check*)
    (a) Let $\vec{c} = (c_{i,j})$ and $\text{VKSIG} = (v_1, \ldots, v_k)$.
```

(b) $\boxed{\text{Let } x \text{ be the smallest value s.t. } v_x \neq v_x^*. \text{Compute } s_j = \mathsf{Dec}_{\mathrm{SK}^{v_x}_{x,j}}(c_{x,j}),\ j = 1, \ldots, \ell}$.

(c) Compute $((w_1, \ldots, w_\ell), m) \leftarrow \mathsf{D}(s_1, \ldots, s_\ell)$. If the decoding fails or $(w_1, \ldots, w_\ell)$ is $\boxed{\frac{\delta}{2}\text{-far}}$ from $(s_1, \ldots, s_\ell)$, 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 $\perp$.

To implement the modified decryption algorithm, we need the $\ell + t$ secret keys for each row of the matrix, that is, ℓ keys for the decryption of the entire sub-rows indexed by $\overline{\text{VKSIG}^*}$ and t keys for the decryption of the columns of S in the sub-rows indexed by VKSIG^*.

4.2.3. *Remark on the Codeword Check and the Gap of Error Fraction*

At first, the codeword check may seem superfluous, but it turns out to play a critical role in achieving indistinguishability between NMDec and NMDec^*.

To convey our point, we illustrate a problem that will arise when the codeword check is omitted. Suppose that the decryption algorithm in our scheme does not have the codeword check. Consider a ciphertext encrypting a matrix of plaintexts where the first row is $(1 - \frac{\delta}{4})$-close to a valid codeword w but the xth row is exactly the same as the first row except having exactly one more error entry and thereby not $(1 - \frac{\delta}{4})$-close to w any more. In this case, the column check will pass with non-negligible probability since the two rows have only one different entry. The problem is that *this ciphertext will pass the decoding check in* NMDec *but not in* NMDec^*, and the indistinguishability argument will break down.

To address this problem, *we first relax the allowable error fraction to $\delta/2$ in the decoding check of* NMDec^* *to embrace the above case*. Of course, this measure alone introduces a new problem. For example, consider a malformed ciphertext ψ for Π where in the underlying matrix of plaintexts, each row is the same corrupted codeword that is $\frac{\delta}{3}$-far from but $(1 - \frac{\delta}{2})$-close to a valid codeword. This time, *the ciphertext will pass the decoding check in* NMDec^* *but not in* NMDec, and the indistinguishability argument will break down again. To fix the problem, we introduce the codeword check *comparing the decrypted raw with the actual valid codeword w*. As we will see below, with the codeword check, the output of NMDec and NMDec^* will be consistent with overwhelming probability.

4.2.4. *Promise Problem*

In order to prove the claim, we would like to have the following guarantees 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 $\vec{M}$ of k by ℓ each entry of which lies in $\Sigma \cup \{\perp\}$.

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 $\frac{\delta}{4}$-far, or the first row is $\frac{\delta}{4}$-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 with respect to the underlying $k \times \ell$ matrix $\vec{M}$ of plaintexts that ψ encrypts; $\vec{M}$ may be in Π_Y or in Π_N or neither. In particular, we 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 $\frac{\delta}{4}$-far, then column check rejects $\vec{M}$ with probability $1 - (1 - \frac{\delta}{4})^t$. On the other hand, if the first row is $\frac{\delta}{4}$-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 $1 - (1 - \frac{\delta}{4})^t$; that is, with probability at least $1 - 2 \cdot (1 - \frac{\delta}{4})^t$, both consistency checks in NMDec and NMDec^* reject $\vec{M}$.
- 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 $(1 - \frac{\delta}{4})$-close to codeword $w \in \mathcal{W}$ and we know in addition that every other row is $(1 - \frac{\delta}{4})$-close to the first row and thus $(1 - \frac{\delta}{2})$-close to w. Since the underlying RPE has parameters $(n, \ell, \delta, t, \Sigma)$ and thereby corrects up to $\frac{\delta}{2}$ fraction errors, we will recover the same codeword w and message m whether we decode the first row within distance $\frac{\delta}{4}$, or any other row within distance $\frac{\delta}{2}$. 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 $\perp$ (possible from the column check or the codeword check) with exactly the same probability, and whenever they do not output $\perp$, they output the same message m.

Proof of Claim 2. 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 is completely determined and identical in both experiments. Having $t \geq \log^2 k$, we claim that with probability $1 - 2 \cdot p(k) \cdot (1 - \frac{\delta}{4})^t = 1 - \mathsf{negl}(k)$ over the choice of S, the decryptions of $(\psi_1, \ldots, \psi_{p(k)})$ agree in both experiments. This follows from the above analysis of the promise problem. □

4.3. *Reducing* $\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))$ *to Semantic Security*

In this section we show the following:

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

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

We now give the 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 $\mathsf{NME}_b^{(2)}$.

1. Recall that according to the definition of mIND_b, the adversary B_1 first receives the set of public keys. Let $\langle \text{PK}_1, \ldots, \text{PK}_{k\cdot(\ell-t)}\rangle$ be the keys B_1 received from mIND_b. Given these public keys, B simulates the key-generation procedure of $\mathsf{NME}^{(2)}$.

   - (simulating key generation of $\mathsf{NME}_b^{(2)}$) Pick a random subset S of $[\ell]$ of size t. 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 $\{(i,j) \mid i \in [k], j \in [\ell] \setminus S\}$ with $[k \cdot (\ell - t)]$. For all $i \in [k]$, $j \in [\ell]$, $\beta \in \{0,1\}$,

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

2. B_1 chooses a message pair to send to mIND_b as follows:

   - (simulating message selection of $\mathsf{NME}_b^{(2)}$) A_1 will choose a message pair and send it to B_1. Let $(\tilde{m}_0, \tilde{m}_1)$ be the pair of messages A_1 returns.

   Upon receiving the message pair, B_1 chooses $(\alpha_1, \ldots, \alpha_t)$ uniformly at random from Σ^t and then computes

$$(w_1^0, \ldots, w_\ell^0) \leftarrow \mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), \tilde{m}_0), \quad (w_1^1, \ldots, w_\ell^1) \leftarrow \mathsf{R}(S, (\alpha_1, \ldots, \alpha_t), \tilde{m}_1).$$

   Recall that R is the reconstruction algorithm of the underlying RPE scheme. Note for $j \in S$, we have $w_j^0 = w_j^1$ coming from $\{\alpha_1, \ldots, \alpha_t\}$. For $j \in S$; let $\gamma_j = w_j^0 = w_j^1$.

   For $i \in [k]$, $j \in [\ell] \setminus S$, and $\beta \in \{0,1\}$, the adversary B sets $m_{\phi(i,j)}^{\beta} = w_j^{\beta}$, and sends the following message pair to mIND_b:

$$(\langle m_1^0, \ldots, m_{k(\ell-t)}^0\rangle, \langle m_1^1, \ldots, m_{k(\ell-t)}^1\rangle)$$

3. B_2 receives challenge ciphertexts $\langle y_1, \ldots, y_{k(\ell-t)}\rangle$ from mIND, according to the distribution $\mathsf{Enc}_{\text{PK}_1}(m_1^b), \ldots, \mathsf{Enc}_{\text{PK}_{k(\ell-t)}}(m_{k(\ell-t)}^b)$. Based on these ciphertexts, B_2 creates a challenge ciphertext to send to A_2 as follows:

- (simulating ciphertext generation of $\mathsf{NME}_b^{(2)}$) B_2 first creates a $k \times \ell$ matrix of ciphertexts $(c_{i,j})$ as follows:

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

  B_2 then computes the signature $\sigma \leftarrow \mathsf{Sign}_{\text{SKSIG}^*}(\vec{c})$. Finally, B_2 sends the ciphertext $[\vec{c}, \text{VKSIG}^*, \sigma]$ to A_2.

It is straightforward to verify that $[\vec{c}, \text{VKSIG}^*, \sigma]$ is a random encryption of $\tilde{m}_b$ under Π.

4. Finally, B_2 outputs a guess using A_2's output. In particular, upon receiving a sequence of ciphertexts $(\psi_1, \ldots, \psi_{p(k)})$ from A_2, B_2 decrypts these ciphertexts using NMDec^* as in $\mathsf{NME}_b^{(2)}$ and then output the decryptions. Note that to simulate NMDec^*, it suffices for B_2 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. □

5. Achieving Bounded-CCA2 Non-malleability

Recall that 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). In our original scheme against a CPA attack, the soundness of consistency check is achieved, since the set S of checked columns are randomly chosen and hidden from the adversary. However, in a q-bounded CCA2 attack, the adversary may learn about S using q decryption queries and break the security of the scheme.

We modify our scheme to achieve non-malleability under a bounded-CCA2 attack. The modification is the straightforward analogue of the [10] modification of the [46] scheme. In other words, we increase the size of S sufficiently so that the soundness of the consistency check still holds even after q decryption queries. In particular, let η be some constant (depending on δ) such that $\left(1 - \frac{\delta}{4}\right)^{\eta} \leq \frac{1}{2}$. We change the parameter of the underlying RPE scheme in Fig. 3 such that

$$t \geq \eta \cdot (\log^2 k + q(k)).$$

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`*.
- 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^*.

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 computationally 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 $\left\{\mathsf{mIND}_b(E, B, k, k(\ell - t))\right\}$ are identically distributed for each $b \in \{0, 1\}$, which can be shown using the reduction in the proof of Claim 3. Therefore, we are only left to show the following claim to conclude the analysis.

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

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

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. Recall that the value $p(k)$ in the various NME-q-CCA experiments corresponds to the number of (mauled) ciphertexts that the adversary would come up with, after given the challenge ciphertext. 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

$$\begin{aligned}\Pr_S[\mathsf{good}(S)] &\geq 1 - 2 \cdot (2^{q+1} \cdot p(k)) \cdot \left(1 - \frac{\delta}{4}\right)^t \\ &\geq 1 - 2^{q+2} \cdot p(k) \cdot \left(\frac{1}{2}\right)^{\log^2 k + q} = 1 - \mathsf{negl}(k).\end{aligned}$$

  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.

□

5.1. *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.

Acknowledgements

This work was initiated while Tal Malkin and Hoeteck Wee were visiting IPAM. Most of 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. Seung Geol Choi was supported in part by the Office of Naval Research (ONR) awards N0001416WX01489 and N0001416WX01645, and National Science Foundation (NSF) award #1618269, and Samsung Scholarship. Dana Dachman-Soled was supported in part by NSF CAREER award #CNS-1453045 and by a Ralph E. Powe Junior Faculty Enhancement Award. Tal Malkin was supported in part by the Defense Advanced Research Project Agency (DARPA) and Army Research Office (ARO) under Contract #W911NF-15-C-0236, and NSF awards #CNS-1445424 and #CCF-1423306. Hoeteck Wee was supported in part by the Agence Nationale de la Recherche (ANR) Project EnBiD (ANR-14-CE28-0003). Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of ONR, DARPA, ARO, NSF, ANR, the U.S. Government, or the French Government. We thank Vinod Vaikuntanathan for sharing his insights on non-malleability, and Oded Goldreich for pointing out [17], and for other helpful suggestions.

Appendix: Example Instantiation of Non-Malleable Encryption Π

We instantiate a non-malleable encryption scheme Π using the RPE construction given in Lemma 1 with $\delta = 0.5$. We summarize the description below:

Let (Gen, Enc, Dec) be an encryption scheme, (GenSig, Sign, 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 $(\mathrm{PK}^b_{i,j}, \mathrm{SK}^b_{i,j})$.
2. Pick a random subset $S \subset [2t]$ of size t.
3. Set $\mathrm{PK} = \left\{(\mathrm{PK}^0_{i,j}, \mathrm{PK}^1_{i,j}) \mid i \in [k], j \in [2t]\right\}$, $\mathrm{SK} = \left\{S, (\mathrm{SK}^0_{i,j}, \mathrm{SK}^1_{i,j}) \mid i \in [k], j \in [2t]\right\}$.

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

1. Pick random $\alpha_1, \ldots, \alpha_t \in \mathrm{GF}(2^n)$ 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 $(\mathrm{SKSIG}, \mathrm{VKSIG})$. 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 $s_j = \mathsf{Dec}_{\mathrm{SK}^{v_1}_{1,j}}(c_{1,j})$ 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 $1.75t$ positions. If no such codeword exists, 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 checks accept, output the message m corresponding to the codeword w; else, output $\perp$.

References

[1] P. Baecher, C. Brzuska, M. Fischlin. Notions of black-box reductions, revisited, in *Advances in Cryptology—ASIACRYPT 2013, Part I.* Lecture Notes in Computer Science, vol. 8269. (Springer, Heidelberg, 2013), pp. 296–315

[2] M. Ball, D. Dachman-Soled, M. Kulkarni, T. Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits, in *Advances in Cryptology - EUROCRYPT 2016—35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II* (2016), pp. 881–908

[3] M. Blum, P. Feldman, S. Micali, Non-interactive zero-knowledge and its applications, in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing* (1988) pp. 103–112

[4] M. Ben-Or, S. Goldwasser, A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing* (1988), pp 1–10

[5] M. Bellare, S. Halevi, A. Sahai, S.P. Vadhan, Many-to-one trapdoor functions and their relation to public-key cryptosystems, in *Advances in Cryptology - CRYPTO '98.* Lecture Notes in Computer Science, vol. 1462 (1998), pp 283–298

[6] E.R. Berlekamp and L.R. Welch. Error correction for algebraic block codes, 1986. US Patent 4,633,470.

[7] S.G. Choi, D. Dachman-Soled, T. Malkin, H. 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*. Lecture Notes in Computer Science, vol. 4948 (2008) pp. 427–444

[8] S. Coretti, Y. Dodis, B. Tackmann, D. Venturi, Non-malleable encryption: simpler, shorter, stronger, in *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*. Lecture Notes in Computer Science, vol. 9562 (Springer, Heidelberg, 2016), pp 306–335

[9] M. Cheraghchi, V. Guruswami. Non-malleable coding against bit-wise and split-state tampering, in *TCC 2014: 11th Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 8349 (Springer, Heidelberg, 2014), pp 440–464

[10] R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, V. Vaikuntanathan. Bounded CCA2-secure encryption, in *Advances in Cryptology - ASIACRYPT 2007*. Lecture Notes in Computer Science, vol. 4833 (2007)

[11] R. Canetti, S. Halevi, J. Katz. Chosen-ciphertext security from identity-based encryption, in *Advances in Cryptology - EUROCRYPT 2004*. Lecture Notes in Computer Science, vol. 3027 (2004), pp 207–222

[12] S. Coretti, U. Maurer, B. Tackmann, D. Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes, in *TCC 2015: 12th Theory of Cryptography Conference, Part I*. Lecture Notes in Computer Science, vol. 9014 (Springer, Heidelberg, 2015), pp. 532–560.

[13] R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, in *Advances in Cryptology - CRYPTO '98*. Lecture Notes in Computer Science, vol. 1462 (1998), pp. 13–25

[14] R. Cramer, V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, in *Advances in Cryptology - EUROCRYPT 2004*. Lecture Notes in Computer Science, vol. 3027 (2004), pp. 45–64

[15] D. Dachman-Soled. A black-box construction of a CCA2 encryption scheme from a plaintext aware (sPA1) encryption scheme, in *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*. Lecture Notes in Computer Science, vol. 8383 (Springer, Heidelberg, 2014), pp. 37–55

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

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

[18] D. Dachman-Soled, T. Malkin, M. Raykova, M. Yung. Efficient robust private set intersection, in *Applied Cryptography and Network Security, 7th International Conference, ACNS 2009*. Lecture Notes in Computer Science, vol. 5536 (2009)

[19] C. Dwork, M. Naor, O. Reingold. Immunizing encryption schemes from decryption errors, in *Advances in Cryptology - EUROCRYPT 2004. Lecture Notes in Computer Science*, vol. 3027 (2004), pp 342–360.

[20] E. Elkind, A. 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/.

[21] V. Goyal, C.-K. Lee, R. Ostrovsky, I. Visconti. Constructing non-malleable commitments: A black-box approach, in *53rd Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, 2012), pp. 51–60

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

[23] Y. Gertner, T. Malkin, S. Myers. Towards a separation of semantic and CCA security for public key encryption, in *Proceedings of the 4th Theory of Cryptography Conference, TCC 2007*. Lecture Notes in Computer Science, vol. 4392 (2007), pp. 434–455

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

[25] Y. Gertner, T. Malkin, O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates, in *The 42th Annual Symposium on Foundations of Computer Science* (2001), pp. 126–135

[26] O. Goldreich, S. Micali, A. 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.

[27] O. Goldreich. *Foundations of Cryptography: Volume II, Basic Applications*. (Cambridge University Press, Cambridge, 2004)

[28] V. Goyal, R. Ostrovsky, A. Scafuro, I. Visconti. Black-box non-black-box zero knowledge, in *46th Annual ACM Symposium on Theory of Computing* (ACM Press, 2014), pp. 515–524

[29] I. Haitner. Semi-honest to malicious oblivious transfer - the black-box way, in *Proceedings of the Fifth Theory of Cryptography Conference, TCC 2008*. Lecture Notes in Computer Science, vol. 4948 (2008), pp. 412–426

[30] D. Hofheinz, E. Kiltz. Practical chosen ciphertext secure encryption from factoring, in *Advances in Cryptology – EUROCRYPT 2009*. Lecture Notes in Computer Science, vol. 5479 (Springer, Heidelberg, 2009), pp 313–332

[31] S. Hohenberger, A.B. Lewko, B. Waters. Detecting dangerous queries: a new approach for chosen ciphertext security, in *Advances in Cryptology – EUROCRYPT 2012*. Lecture Notes in Computer Science, vol. 7237 (Springer, Heidelberg, 2012), pp. 663–681

[32] Y. Ishai, E. Kushilevitz, Y. Lindell, E. Petrank. Black-box constructions for secure computation, in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing* (2006), pp. 99–108

[33] R. Impagliazzo, S. Rudich. Limits on the provable consequences of one-way permutations, in *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing* (1989), pp. 44–61

[34] S. Kiyoshima. Round-efficient black-box construction of composable multi-party computation, in *Advances in Cryptology – CRYPTO 2014, Part II*. Lecture Notes in Computer Science, vol. 8617 (Springer, Heidelberg, 2014), pp. 351–368

[35] E. Kiltz, P. Mohassel, A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security, in *Advances in Cryptology – EUROCRYPT 2010*. Lecture Notes in Computer Science, vol. 6110 (Springer, Heidelberg, 2010), pp. 673–692

[36] S. Kiyoshima, Y. Manabe, T. Okamoto. Constant-round black-box construction of composable multi-party computation protocol, in *TCC 2014: 11th Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 8349 (Springer, Heidelberg, 2014), pp. 343–367

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

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

[39] Y. Lindell, B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries, in *Advances in Cryptology - EUROCRYPT 2007*. Lecture Notes in Computer Science, vol. 4515 (2007), pp 52–78

[40] H. Lin, R. Pass. Black-box constructions of composable protocols without set-up, in *Advances in Cryptology – CRYPTO 2012*. Lecture Notes in Computer Science, vol. 7417 (Springer, Heidelberg, 2012), pp. 461–478

[41] T. Matsuda, G, Hanaoka. Chosen ciphertext security via point obfuscation, in *TCC 2014: 11th Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 8349 (Springer, Heidelberg, 2014), pp 95–120

[42] S. Micali, C. Rackoff, B. Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, 1988. Special issue on cryptography.

[43] S. Myers, A. Shelat. Bit encryption is complete, in *50th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, 2009), pp. 607–616

[44] S. Myers, M. Sergi, A. 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*. Lecture Notes in Computer Science, vol. 7485 (Springer, Heidelberg, 2012), pp. 149–165

[45] M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks, in *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing* (1990), pp. 427–437

[46] R. Pass, A. Shelat, V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one, in *Advances in Cryptology - CRYPTO 2006*. Lecture Notes in Computer Science, vol. 4117 (2006), pp. 271–289

[47] C. Peikert, B. Waters, Lossy trapdoor functions and their applications, in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (2008), pp. 187–196

[48] R. Pass, H. Wee. Black-box constructions of two-party protocols from one-way functions, in *Proceedings of the 6th Theory of Cryptography Conference, TCC 2009*. Lecture Notes in Computer Science, vol. 5444 (2009), pp. 403–418

[49] J. Rompel. One-way functions are necessary and sufficient for secure signatures, in *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing* (1990), pp. 387–394

[50] C. Rackoff, D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in *Advances in Cryptology - CRYPTO '91. Lecture Notes in Computer Science*, vol. 576 (1992) pp. 433–444

[51] A. Rosen, G. Segev. Chosen-ciphertext security via correlated products, in *TCC 2009: 6th Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 5444 (Springer, Heidelberg, 2009), pp. 419–436

[52] O. Reingold, L. Trevisan, S. Vadhan. Notions of reducibility between cryptographic primitives, in *Proceedings of the First Theory of Cryptography Conference, TCC 2004*. Lecture Notes in Computer Science, vol. 2951 (2004), pp. 1–20

[53] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security, in *The 40th Annual Symposium on Foundations of Computer Science* (1999), pp. 543–553

[54] H. Wee. Black-box, round-efficient secure computation via non-malleability amplification, in *51st Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, 2010), pp. 531–540

[55] H. Wee. Efficient chosen-ciphertext security via extractable hash proofs, in *Advances in Cryptology – CRYPTO 2010*. Lecture Notes in Computer Science, vol. 6223 (Springer, Heidelberg, 2010), pp. 314–332