# Improved, Black-Box, Non-Malleable Encryption from Semantic Security

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

March 16, 2017

**Abstract**

We give a new black-box transformation from any semantically secure encryption scheme into a non-malleable one which has a better rate than the best previous work of Coretti et al. (TCC 2016-A). We achieve a better rate by departing from the "matrix encoding" methodology used by previous constructions, and working directly with a single codeword. We also use a Shamir secret-share packing technique to improve the rate of the underlying error-correcting code.

## 1 Introduction

The basic security requirement for public key encryption (PKE) schemes, known as *semantic security* or IND-CPA (indistinguishability under chosen plaintext attack), is that an eavesdropping adversary learns nothing about the plaintext underlying a communicated ciphertext (equivalently, cannot distinguish an encryption of one plaintext from another). Often, however, this indistinguishability guarantee is not sufficient, and a PKE satisfying stronger properties is required.

**IND-CCA2 from semantic security.** A strong level of PKE security, which is sufficient for most applications, is indistinguishability under chosen-ciphertext attacks (IND-CCA2), wherein the adversary may ask adaptive queries to a decryption oracle (as long as it does not query the "challenge ciphertext" itself).

A natural question is whether one can achieve IND-CCA2 security only from IND-CPA security; however, this question still remains a big open problem. On one hand, the various known constructions of IND-CCA2 secure encryption either rely on IND-CPA security and some additional properties that computational hardness assumptions from number theory provide [CS03, Wee12, HJ12, LPJY14, KW15, LPJY15, GHKW16] or use generic assumptions seemingly stronger than IND-CPA secure encryption, e.g., non-interactive zero knowledge, identity-based encryption, or lossy trapdoor functions [DDN00, CHK04, PW08]. On the other hand, there has been only a partial black-box separation result [GMM07] between IND-CPA and IND-CCA2. Therefore, researchers have proposed and studied various intermediate notions of security.

[*] United States Naval Academy, choi@usna.edu

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

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

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

| $n$<br>scheme | $o(k)$ | $\Theta(k)$ | $\Omega(k^2)$ |
|---|---|---|---|
| [CDMW08] | $\Theta(k^3)$ | $\Theta(k^2 n)$ | $\Theta(k^2 n)$ |
| [CDTV16] | $\Theta(k^3)$ | $\Theta(k^2 n)$ | $\Theta(kn)$ |
| This work | $\Theta(k^2)$ | $\Theta(kn)$ | $\Theta(n)$ |

| $n$<br>scheme | $\Theta(k)$ | $\Theta(k^2)$ | $\Omega(k^3)$ |
|---|---|---|---|
| HE + [CDMW08] | $\Theta(k^2 n)$ | $\Theta(kn)$ | $\Theta(n)$ |
| HE + [CDTV16] | $\Theta(k^2 n)$ | $\Theta(kn)$ | $\Theta(n)$ |
| HE + this work | $\Theta(kn)$ | $\Theta(n)$ | $\Theta(n)$ |

Figure 1: The ciphertext lengths of various NM-CPA encryption schemes. The parameter $k$ is the security parameter, and $n$ is the message length. We assume the underlying IND-CPA encryption has a constant rate for messages of length $\Omega(k)$; encrypting $o(k)$-long messages with IND-CPA encryption is assumed to be $\Theta(k)$-long. HE denotes the hybrid encryption according to Herranz et al. [HHK10]. Note that hybrid encryption doesn't help reduce the ciphertext length for short messages.

**Non-malleable encryption from semantic security.** In this work, we consider the notion of non-malleability under chosen-plaintext attacks (NM-CPA), initially put forward by Dolev, Dwork and Naor [DDN00]. Roughly, non-malleability requires that it is infeasible for an adversary to modify a ciphertext into one, or many, other ciphertexts of messages related to the original plaintext. It was shown by Bellare and Sahai [BS99] and by Pass, shelat and Vaikuntanathan [PSV06] that NM-CPA is equivalent to security against adversaries with access to a non-adaptive decryption oracle, meaning that the adversary can only ask one "parallel" CCA2 decryption query. We also consider the notion of non-malleability under bounded-CCA2 attacks (NM-$q$-CCA2) [CHH$^+$07], where we allow the adversary to adaptively query the decryption oracle at most $q$ times (in contrast, NM-CCA2 allows an *unbounded* number of queries, and is equivalent to IND-CCA2 [DDN00]).

Besides being a "stepping stone" between semantically secure and CCA2 secure encryption, non-malleability (or NM-CPA security) is an important notion in its own right. As one motivating example, consider the use of PKE in auctions. Buyers place their bids for an item to a seller, encrypted under the seller's public key, and the seller sells the item to the buyer with the highest bid. We certainly want to rule out adversaries who consistently bid exactly one dollar more than the previous bidders.

**Previous work on achieving NM-CPA from IND-CPA.** Interestingly, although NM-CPA appears closer to IND-CCA2 than IND-CPA security, a sequence of results (i.e., a non-black-box construction by [PSV06] followed by a black-box construction by [CDMW08]) showed that NM-CPA schemes (and even NM-$q$-CCA2 schemes) can be constructed from any IND-CPA scheme.

In a recent work, Coretti et al. [CDTV16] revisited the work of [CDMW08], and investigated (among other results) 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, defined as $\frac{n}{c(n)}$, where $n$ is the message length and $c(n)$ is the length of the corresponding ciphertext[1].

The transformation of [CDMW08] gives an $n$-bit NM-CPA scheme such that its encryption algorithm calls the underlying $n$-bit IND-CPA scheme $\Theta(k^2)$ times, where $k$ is the security parameter.[2]

[1] We note that the result of Coretti et al. [CDTV16] showed that (a generalization) of the construction of [CDMW08] achieves not only NM-CPA security, but also a stronger notion of security—*indistinguishability under (chosen-ciphertext) self-destruct attacks (IND-SDA)*—where the adversary gets access to an adaptive decryption oracle that stops decrypting after the first invalid ciphertext is submitted.

[2] In fact, according to [CDMW08], the number $\Theta(k^2)$ of calls to IND-CPA encryption can be optimized to

For example, assuming a constant-rate IND-CPA encryption, the transformation gives a $\Theta(k)$-bit NM-CPA scheme with the ciphertext length of $\Theta(k^3)$.

Coretti et al. [CDTV16] give an improved transformation by replacing the error-correcting code used in [CDMW08] with one having a better rate, although the transformation still invokes the same number $\Theta(k^2)$ of calls to the underlying IND-CPA encryption. In particular, this allows $\Theta(k^3)$-bit ciphertexts to encrypt $\Theta(k^2)$-bit messages. See Figure 1 for more detailed comparison.

## 1.1 Our Results

In this work, we give a black-box transformation from IND-CPA encryption to NM-CPA encryption with better efficiency.

**Conceptual contribution.** Our main conceptual contribution is that *we no longer follow the framework of [DDN00] (and all subsequent constructions) of creating $k$ encryptions of the same message or codeword.*

In particular, as we elaborate on in the next section, previous constructions rely on a "matrix encoding" of the plaintext as a $k \times \ell$ matrix of elements, where each row in the matrix is an encoding of the plaintext message via an appropriate code (the message itself in [DDN00], or more sophisticated encodings in subsequent works). The rows of the matrix are indexed by a one-time signature, so we need at least $k$ (security parameter) rows. It follows that using this methodology incurs a ciphertext expansion of *at least a factor of $k$, regardless of the underlying code used.* In this sense, [CDTV16] have achieved the best possible rate within this construction framework.

We depart from this "matrix encoding" methodology and work directly with a *single* codeword. This allows us to achieve the first black-box transformation that invokes $\Theta(k)$ calls to the underlying IND-CPA encryption algorithm; previous black-box constructions need $\Theta(k^2)$ calls.

> **Main theorem (informal)** There exists a (fully) black-box construction of a non-malleable encryption scheme from any IND-CPA encryption scheme, in which the encryption algorithm calls the underlying IND-CPA encryption algorithm $\Theta(k)$ times.

We also extend the theorem to provide a black-box construction of NM-$q$-CCA2 secure encryption [CHH$^+$07] from any semantically secure encryption, calling the IND-CPA encryption algorithm $\Theta(k+q)$ times.

**NM-CPA encryption with a better rate.** Applying the aforementioned transformation, we achieve an NM-CPA encryption scheme with a better rate. For this, we use a Shamir secret-share packing technique to improve the rate of the underlying error-correcting code to encode the plaintext in the transformation. In particular, we achieve a *constant-rate NM-CPA encryption for messages of length* $\Omega(k^2)$. We compare our results with the previous work in Figure 1.

We note that one can achieve a better rate for long messages by using hybrid encryption. In particular, Herranz et al. [HHK10] showed that NM-CPA KEM plus IND-CCA2 DEM implies NM-CPA PKE. (For shorter messages, the ciphertext length is dominated by the KEM part of encrypting the $\Theta(k)$-long encapsulated key, since for the DEM part, we have a constant-rate IND-CCA2 secure

$\Theta(k \log^2 k)$; to achieve a negligible soundness error, the scheme checks $k$ random positions, but observe it's enough to check $\log^2 k$ positions since we have $1/2^{\log^2 k} = \mathsf{negl}(k)$. However, we choose to compare the results by using the non-optimized $O(k^2)$ calls, following the presentation of Coretti et al. [CDTV16].

symmetric encryption scheme [BN08].) Even considering the hybrid encryption framework, our scheme achieves better efficiency: Our scheme achieves a constant rate for messages of length $\Omega(k^2)$, rather than for messages of length $\Omega(k^3)$ in the previous schemes.

**Potential applications to other related work.** The original techniques of [CDMW08] (in particular, the properties of the matrix encoding scheme and its use for verifying consistency) have been used implicitly or explicitly in several works for different purposes. For example, there have been black-box constructions of non-malleable commitments [PW09], set intersection protocols from homomorphic encryptions [DMRY09], and a CCA2-secure encryption scheme for strings starting from one for bits [Ms09]. The works of [Wee10, LP12, KMO14, Kiy14] used these techniques in the context of black-box, round-efficient secure computation. The works of [GLOV12, GOSV14] extended the ideas to provide consistency relations beyond equality using VSS and the paradigm of MPC-in-the-head.

We hope that our improved efficiency, constant rate transformation can be used to improve efficiency in some of these or other application domains. In fact, a very recent work [BDSKM16] has already used our results to construct their non-malleble codes resilient against local tampering functions and bounded-depth circuits. Indeed, their results instantiated with the previous matrix encoding techniques would yield non-malleable codes resilient against functions with locality up to $n^c$ for some *specific* $c < 1$ (roughly $c = 1/3$). However, using our results as an ingredient, they were able to achieve resilience against locality $n^{1-\epsilon}$ for *any* constant $\epsilon < 1$ (and even $\frac{n}{\log n}$ with inefficient codes), and much better rate even in lower locality ranges.

# 2 Techniques

## 2.1 Overview of Previous Techniques

We begin with an overview of previous techniques of [DDN00, PSV06, CDMW08, CDTV16], which we will refer to below as DDN, PSV, CDMW and CDTV, respectively. We focus on the details that will be helpful towards understanding our techniques.

**Non-black-box transformations by DDN and PSV.** Let $k$ be the security parameter. The key generation algorithm generates $2k$ independent keys for the underlying semantically-secure public key encryption scheme $\textsc{pk}_i^b$ for $i = 1, \ldots, k$, and $b \in \{0, 1\}$ (and the corresponding secret keys). Encryption of message $m$ proceeds as follows:

(a) Generate a $(\textsc{vksig}, \textsc{sksig})$ pair for a one-time signature (where $|\textsc{vksig}| = k$).

(b) Generate $k$ encryptions of the message $m$. In particular, use keys $\textsc{pk}_i^{\textsc{vksig}_i}$ for $i = 1, \ldots, k$ for encryptions.

(c) Give a non-interactive zero-knowledge proof (or the relaxed "designated verifier" version) proving that all resulting ciphertexts are encryptions of the same message.

(d) Sign the entire bundle with a one-time signature.

It is in step (c) that a general $\mathsf{NP}$-reduction is used, which in return makes the construction non-black-box (and inefficient). In the proof of security, we exploit the fact that for a well-formed ciphertext, we can recover the message if we know the secret key for *any* of the $k$ encryptions.

**Black-box transformations by CDMW.** Let $k$ be the security parameter, and let $\ell = O(k)$ (or any superlogarithmic function in $k$). The key generation algorithm generates $2k\ell$ independent keys for the underlying semantically-secure public key encryption scheme $\textsc{pk}^b_{i,j}$ for $i = 1, \dots, k$, $j = 1, \dots, \ell$, and $b \in \{0, 1\}$ (and the corresponding secret keys). The encryption algorithm utilizes a Reed-Solomon error correcting code (ECC) with encoding algorithm $\mathsf{E}$ (see Appendix **??**). Now, the encryption algorithm has the following form:

(a) Generate a $(\textsc{vksig}, \textsc{sksig})$ pair for a one-time signature (where $|\textsc{vksig}| = k$).

(b) Obtain an encoding $w$ of a message $m$ by computing $w \leftarrow \mathsf{E}(m)$. Generate $k$ encryptions of the same codeword $w$, using $\ell$ public keys per each of the $k$ encryptions in a way that we explain below.

(d) Sign the entire bundle with a one-time signature.

Obviously, the scheme should provide *some mechanism for checking the consistency of $k$ encryptions*, corresponding to step (c) in DDN and PSV (i.e., the non-interactive zero-knowledge proof). That way, even if the simulated decryption in the proof of security decrypts any of the $k$ ciphertexts, the decryption should be correct with overwhelming probability. CDMW achieved this by using a codeword $w$ consisting of $\ell$ elements, and encrypting each element with a different public key, for a total of $k\ell$ encryptions. The decryption algorithm checks consistency of the $k$ encryptions of *$w$ by checking consistency of a random subset of columns* (where the randomness is determined by its secret key). Then, the decryption algorithm decrypts and error-corrects the first row, and checks that in that same subset of locations, this codeword is not corrupted. If both these column-check and codeword-check pass, output the decoded message.

We next describe the details of how the above outline is implemented, and the intuition behind its security and parameter choices. Recall that a Reed-Solomon (RS) codeword consisting of $\ell$ output symbols is simply a polynomial $p$ of degree $d = O(\ell)$ over a finite field, evaluated at $\ell$ points (say $1, \dots, \ell$). The way CDMW encode a message $m$ is via Shamir secret-sharing, which can be viewed as an instantiation of a RS code. Specifically, set $p(0) = m$, choose the values of $p(1), \dots, p(d)$ at random, and interpolate to obtain the unique degree $d$ polynomial $p$. Let the final encoding $w \leftarrow \mathsf{E}(m)$ consist of $w_1 = p(1), \dots, w_\ell = p(\ell)$. The encryptions now proceed as follows:

> Construct a $k \times \ell$ matrix $M$, where $M_{i,j} = w_j$ and $k$ is the number of bits in $\textsc{vksig} = \textsc{vksig}_1, \dots, \textsc{vksig}_k$. Each entry of $M_{i,j}$ is then encrypted under a one of two public keys $(\textsc{pk}^0_{i,j}, \textsc{pk}^1_{i,j})$, depending on whether $\textsc{vksig}_i$ is 0 or 1.

In the actual decryption algorithm, the first row of the encrypted matrix is always decrypted and decoded, whereas in the security proof, the decrypted row will be chosen based on which secret keys are available to the reduction, and it is ensured that in each submitted ciphertext there is some row for which the reduction knows all the secret keys. The key challenge is to ensure that decrypting and decoding *any one* of the $k$ rows of the encrypted matrix will yield *the same* message $\widetilde{m}$ (possibly $\perp$) as the decrypting and decoding the first row. This is where the "column check" and "codeword check" come in. In the column check, we decrypt a random subset of $t = O(\ell)$ columns, and check that all the entries in each of these columns are the same; the random subset is chosen in key generation and embedded into the private key. Intuitively, this ensures that the encoding in each row is "close" to the encoding in the first row. In the codeword check, we decrypt and decode the first row and then check the resulting codeword against the received word in the first row. Specifically, we check that $t = O(\ell)$ random positions of the first row (the same ones that were opened during the column check) agree with the corresponding $t$ positions in the decoded codeword. Intuitively, this is a type of a cut-and-choose check which ensures that the encoding in the first row is "close" to a valid codeword. If either of the checks fails, we output $\perp$. Put together, the two checks ensure that with overwhelming probability, all rows must decode to the same message (or to $\perp$), and thus provide the desired consistency.

The reason CDMW needs $\ell$ to be superlogarithmic, is that for the codeword check, we need the number of random positions $t = O(\ell)$ to satisfy $2^{-t} = \mathsf{negl}(k)$ so that a codeword that is far from valid will pass the check with negligible probability. Thus, the RS code used for each row is not constant rate.

**More efficient black-box transformation by CDTV.** The general insight of CDTV (and also the full version of CDMW [CDMW16]) is that the above construction can be generalized to work for a larger class of encoding schemes $\mathsf{E}$, beyond just Reed-Solomon codes. Specifically, Coretti et al. [CDTV16] note that using a LECSS (linear error-correcting secret sharing scheme) [CG14, CDD$^+$15] is sufficient, whereas [CDMW16] introduced a notion of reconstructable probabilistic encoding scheme (building on [DGR99]). Using these insights, the above works were able to replace the RS code described above with a constant-rate encoding scheme (for long enough messages). Specifically, each row with $\ell$ elements can in fact encode a message of length $O(\ell)$ elements, resulting in a constant rate code for each row (while still maintaining $k$ rows).

## 2.2 Our Techniques

Our encryption scheme also utilizes reconstructable probabilistic encoding (RPE) schemes. RPE schemes are, informally, error-correcting codes (ECCs) 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.

**From $k$ encryptions to a single encryption.** Our first technical contribution is identifying a property of RPE schemes and showing how it can be leveraged to eliminate the need for the "repetition" encoding in previous works. The property we use is that error-correction and decoding can be performed given *any* large enough (constant-fraction) sized subset $\lambda \cdot \ell$ of positions of the corrupted codeword (here $0 < \lambda < 1$ is a constant). We call this property the "decoding from partial views" property. Crucially, we would like this property to hold in a strong way, so that for *any* such subset, we always decode to *the same* codeword/message (possibly $\perp$), even for arbitrarily corrupt codewords, with overwhelming probability (taken over the random choice of the secret key).

We have already discussed a similar property as underlying, at least implicitly, the previous works relying on matrix encoding. However, in those works the property applied to decoding from any one of the $k$ rows (which constitute a repetition code), and was unrelated to the use of RPE for the encoding within each row. Our novel observation is that in fact a similar property can apply *directly* to RPE (with appropriate parameters). A *single* RPE codeword could then allow decoding from any partial view subset, and by correctly adapting a codeword check and another layer of (standard) encoding on the signature, we can achieve the strong version guaranteeing consistency with overwhelming probability.

Thus, encryption of a message $m$ proceeds as follows:

(a) Generate a $(\textsc{vksig}, \textsc{sksig})$ pair for a one-time signature.

(b) Let $\mathsf{E}$ be the encoding algorithm of a RPE with the output length $\ell$. Let $C$ be a linear code with relative distance $\lambda < 1$, encoding the length-$k$ $\textsc{vksig}$ to a length-$\ell$ string (note that $C$ does not have to be efficiently error correctable). Set $\vec{s} \leftarrow \mathsf{E}(m)$, where $\vec{s}$ is a vector of length $\ell$. Let $v_1 \cdots v_\ell$ be the output of $C(\textsc{vksig})$. For $j = 1, \ldots, \ell$, encrypt each entry $s_j$ under public key $\textsc{pk}_j^{v_j}$, yielding a vector of ciphertexts.

(c) Sign the entire bundle with a one-time signature.

In the actual decryption algorithm, the first $\lambda \cdot \ell$ positions of the ciphertext vector are always decrypted and decoded, whereas in the security proof, a specific subset of size $\lambda \cdot \ell$ will be chosen based on which secret keys are available to the reduction. In the proof, we use the fact that, due to unforgeability of the signature scheme, the $\textsc{vksig}$ for each submitted ciphertext must be different than the $\textsc{vksig}$ of the challenge ciphertext, and the fact that $C$ has distance $\lambda \cdot \ell$ to ensure that there is always some sufficiently large subset for which the reduction knows all the secret keys.

To ensure that decoding any of subset of size $\lambda \cdot \ell$ positions yields the same message $\widetilde{m}$ as the first subset (or both will give $\perp$), we require an analogue of the codeword check only (but no column check). As before, in the codeword check we compare the codeword obtained by decrypting and decoding the first $\lambda \cdot \ell$ positions with the received word. Specifically, we check that $t$ random positions throughout the entire received word agree with the corresponding $t$ positions in the decoded codeword.

**Constant-rate RPE.** Our second contribution is to show that the above framework is implementable with a constant rate RPE: we show that using Reed-Solomon codes with packed Shamir secret sharing yields a constant rate RPE with appropriate parameters. Compared with the Reed-Solomon based encodings used by [CDMW08] for $k$ rows, here our encoding has a single row (of length a constant times larger), and a longer message is encoded in each codeword via the packing technique [FY92]. That is, the polynomial is taken to be of a larger degree, and the message is encoded in several evaluation points of the polynomial.

## 3 Preliminaries and Definitions

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 $r$. A function $\mu$ from the natural numbers to the non-negative real numbers is *negligible* if for every positive polynomial $p(\cdot)$ there is an $N$ such that for all integers $k > N$, $\mu(k) < \frac{1}{p(k)}$. We use $\mathsf{negl}(k)$ to denote some arbitrary negligible function.

**Definition 1** (Statistical Indistinguishability)**.** *Two probability ensembles $A = \{A_k\}_{k \in \mathbb{N}}$ and $B = \{B_k\}_{k \in \mathbb{N}}$ are* statistically indistinguishable *if it holds that*

$$\sum_{\alpha} \Big| \Pr[X_k = \alpha] - \Pr[Y_k = \alpha] \Big| \leq \mathsf{negl}(k).$$

Statistical indistinguishability between two ensembles $A$ and $B$ is denoted by $A \overset{s}{\approx} B$.

**Definition 2** (Computational Indistinguishability)**.** *Two probability ensembles $A = \{A_k\}_{k\in\mathbb{N}}$ and $B = \{B_k\}_{k\in\mathbb{N}}$ are* computationally indistinguishable *if for every ppt distinguisher $D$, it holds that*

$$\left| \Pr_{a\leftarrow A_k}[D(1^k, a) = 1] - \Pr_{b\leftarrow B_k}[D(1^k, b) = 1] \right| \leq \mathsf{negl}(k).$$

Computational indistinguishability between two ensembles $A$ and $B$ is denoted by $A \overset{c}{\approx} B$.

**Distance of two strings.** Given two strings $v, w$ of length $\ell$ over an alphabet $\Sigma$, we say that $v$ and $w$ are $\delta$-far if they disagree in more than $\delta \cdot \ell$ positions, where $0 \leq \delta \leq 1$; we say that $v$ and $w$ are $\delta$-close if they agree in more than $\delta \cdot \ell$ positions.

### 3.1 Semantically Secure Encryption

In the following we define the syntax of a public key encryption scheme and the standard security notion, semantic security. We then present the two stronger security notions of non-malleability and bounded-CCA2 non-malleability.

**Definition 3** (Encryption scheme)**.** *A triple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an* encryption scheme*, if* $\mathsf{Gen}$ *and* $\mathsf{Enc}$ *are ppt algorithms and* $\mathsf{Dec}$ *is a deterministic polynomial-time algorithm which satisfies the following property:*

> Correctness. *With probability* $1-\mathsf{negl}(k)$ *over* $(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k)$*: for all* $m$, $\Pr[\mathsf{Dec}_{\text{SK}}(\mathsf{Enc}_{\text{PK}}(m)) = m] = 1$.

**Definition 4** (Semantic security)**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme and let the random variable* $\mathsf{IND}_b(\Pi, A, k)$*, where* $b \in \{0, 1\}$*,* $A = (A_1, A_2)$ *are ppt algorithms and* $k \in \mathbb{N}$*, denote the result of the following probabilistic experiment:*

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

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

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

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

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

```
$\mathsf{mIND}_b(\Pi, A, k, \ell)$ :
    For $i = 1, \ldots, \ell$: $(\textsc{pk}_i, \textsc{sk}_i) \leftarrow \mathsf{Gen}(1^k)$
    $(\langle m_0^1, \ldots, m_0^\ell \rangle, \langle m_1^1, \ldots, m_1^\ell \rangle, \textsc{state}_A) \leftarrow A_1(\langle \textsc{pk}_1, \ldots, \textsc{pk}_\ell \rangle)$
        s.t. $|m_0^1| = |m_1^1|, \ldots, |m_0^\ell| = |m_1^\ell|$
    For $i = 1, \ldots, \ell$: $y_i \leftarrow \mathsf{Enc}_{\textsc{pk}_i}(m_b^i)$
    $D \leftarrow A_2(y_1, \ldots, y_\ell, \textsc{state}_A)$
    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:*

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

## 3.2 Non-malleable Encryption

We next define the notion of non-malleability for public key encryption. Recall that, intuitively, non-malleability says that upon receiving the challenge ciphertext, which encrypts the message $m_b$, $b \in \{0, 1\}$, the *decryptions* of the set of ciphertexts produced by the adversary for $b = 0$ and $b = 1$ are computationally indistinguishable.

**Definition 5** (Non-malleable encryption [PSV06]). *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let the random variable $\mathsf{NME}_b(\Pi, A, k, \ell)$ where $b \in \{0, 1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k, \ell \in \mathbb{N}$ denote the result of the following probabilistic experiment:*

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

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

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

It was shown in [PSV06] that an encryption that is non-malleable (under Definition 5) remains non-malleable even if the adversary $A_2$ receives several encryptions under many different public keys (the formal experiment is the analogue of $\mathsf{mIND}$ for non-malleability).

## 3.3 Bounded-CCA2 Non-Malleability

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

**Definition 6** (Bounded-CCA2 non-malleable encryption [CHH+07])**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme and let the random variable* $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b(\Pi, A, k, \ell)$ *where* $b \in \{0, 1\}$*,* $A = (A_1, A_2)$ *are ppt algorithms and* $k, \ell \in \mathbb{N}$ *denote the result of the following probabilistic experiment:*

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

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

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

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

### 3.4 (Strong) One-Time Signature Schemes

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

**Definition 7** ((Digital) signature scheme)**.** *A triple* $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ *is a* (digital) signature scheme*, if* $\mathsf{GenSig}$ *and* $\mathsf{Sign}$ *are ppt algorithms and* $\mathsf{VerSig}$ *is a deterministic polynomial-time algorithm which satisfies the following property:*

> Correctness. *With probability* $1 - \mathsf{negl}(k)$ *over* $(\textsc{vksig}, \textsc{sksig}) \leftarrow \mathsf{GenSig}(1^k)$*: for all* $m$*,* $\Pr[\mathsf{VerSig}_{\textsc{vksig}}(m, \mathsf{Sign}_{\textsc{sksig}}(m)) = 1] = 1$.

**Definition 8** (Strong, existential unforgeability for a single message)**.** *Let* $\Pi = (\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ *be a signature scheme. For a pair of ppt algorithms* $A = (A_1, A_2)$ *and* $k \in \mathbb{N}$*, let the random variable* $\mathsf{Forge}(\Pi, A, k)$ *denote the result of the following probabilistic experiment:*

```
$\mathsf{Forge}(\Pi, A, k)$ :
  $(\textsc{vksig}, \textsc{sksig}) \leftarrow \mathsf{GenSig}(1^k)$
  $(m, \textsc{state}_A) \leftarrow A_1(\textsc{vksig})$
  $\sigma \leftarrow \mathsf{Sign}_{\textsc{sksig}}(m)$
  $(m', \sigma') \leftarrow A_2(\sigma, \textsc{state}_A)$
  Output 1 iff $\sigma' \neq \sigma$ and $\mathsf{VerSig}_{\textsc{vksig}}(m', \sigma') = 1$
```

$\Pi = (\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ *is* strongly existentially unforgeable for a single message *if any ppt algorithms $A = (A_1, A_2)$:*

$$\Pr[\mathsf{Forge}(\Pi, A, k) = 1] \leq \mathsf{negl}(k).$$

In this paper, signature/verification key pairs are sometimes represented as strings over a non-binary alphabet; this technique has also been used in [RS10]. This augmented version can simply be cast into the version over the binary alphabet by trivially encoding such keys into a binary string.

# 4 Reconstructable Probabilistic Encoding Scheme (RPE)

## 4.1 RPE Definition

We assume that the readers are familiar with basic notions of error correcting codes and Reed-Solomon codes (Appendix **??** gives a brief overview of the notions). We define a reconstructable probabilistic encoding (RPE) below. The secrecy property of an RPE implies that short partial codewords are not bound to any message. The reconstruction property implies that one can later bind such a short partial codeword to any target message and reconstruct the whole consistent codeword. Jumping ahead, this reconstruction procedure will be used to create two different messages sharing the same partial codeword in the reduction step of the proof.

There are several parameters in RPE schemes. A message is represented as a binary string, and the parameter $n$ specifies the length of a message (in bits). A codeword is represented as a string over alphabet $\Sigma$, and the parameter $\ell$ specifies the length of a codeword in the codeword space. The parameter $\delta$ is used to specify the relative distance between codewords. The parameter $t$ is used to specify a threshold to determine whether a partial codeword is short; every codeword of length at most $t$ is considered short.

**Definition 9** (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 $\Sigma$ is an alphabet.*

- *The* encoding algorithm $\mathsf{E}$ *is an efficient probabilistic procedure, which takes a message $m \in \{0,1\}^n$ as input and outputs a codeword $w$ over $\Sigma^\ell$. We let the code $\mathcal{W}$ be the support of* $\mathsf{E}$.
- *The* decoding algorithm $\mathsf{D}$ *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 $\mathsf{R}$ *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 $\delta$-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 $\Sigma^t$.*

*(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 $\mathsf{E}(m)|_S = (\alpha_1, \ldots, \alpha_t)$.*

We note that similar properties have been exploited already in the early work on secure multi-party computation of Ben-Or, Goldwasser, and Wigderson [BGW88], with encoding via low-degree polynomials (or Reed-Solomon codes or Shamir secret sharing with Berlekamp-Welch correction). The above notion of RPE was explicitly defined in [CDMW16], extending a definition given by Decatur, Goldreich, and Ron [DGR99], who only required error-correction and secrecy, but not reconstruction.

## 4.2 Decoding from Partial Views

The following property will be useful for our construction of non-malleable encryption. Informally, the following lemma states that for any RPE as above, given a sufficiently large "partial view" (i.e. subset of positions), decoding with error correction can be successfully performed on this partial view.

**Lemma 1.** *Let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be a* reconstructable probabilistic encoding scheme *with parameters $(n, \ell, \delta, t, \Sigma)$, and let $\lambda := (1-\delta/4)$ and $\delta' := \delta/2$. Then, there is an efficient procedure $\mathsf{D}'$ satisfying the following:*

> *For any set $S \in [\ell]$ with size $s := \lambda \cdot \ell$, for any $w \in \mathcal{W}$, and for any $w' \in \Sigma^s$ that is $(1 - \delta'/2)$-close to $w|_S$, it holds that $D'(S, w')$ outputs $w$ along with a consistent $m$.*

*Proof.* Let $S = \{i_1, \ldots, i_s\}$, where $s = \lambda \cdot \ell$. The decoding from partial views procedure $\mathsf{D}'(S, w')$ does the following. Define the string $\widetilde{w} = \widetilde{w}_1, \ldots, \widetilde{w}_\ell \in \Sigma^\ell$ in the following way: For $j \in [s], \widetilde{w}_{i_j} := w'_j$ and for $v \in [\ell] \setminus S$, $\widetilde{w}_v = \sigma$, where $\sigma$ is an arbitrary symbol in $\Sigma$. Note that if $w'$ is $(1 - \delta'/2)$-close to $w|_S$ for some $w$ in $\mathcal{W}$, then $\widetilde{w}$ is $(1 - \delta/2)$-close to $w$ for some string $w$ in $\mathcal{W}$. Therefore, running the regular decode procedure, $\mathsf{D}(\widetilde{w})$ is guaranteed to output $(w, m)$, where $w \in \mathcal{W}$ is the corrected codeword, and $m$ is the original message. □

## 4.3 RPE from Reed-Solomon Codes

In this section, we construct such a constant-rate RPE scheme with a RS (Reed-Solomon) code and packed secret-sharing [FY92]. We note this is a simple construction; similar constructions were given in different contexts.

**Construction 1. (RS-based RPE)**

For any $n, t, \gamma \in \mathbb{N}$, and for any $\delta$ with $0 < \delta < 1$, we construct an RPE scheme with parameters $(n, \ell, \delta, t, \Sigma)$, where $\ell$ is an integer such that $\ell \geq \frac{t+u-1}{1-\delta}$ with $u = \lceil n/\gamma \rceil$ and $\Sigma = \mathrm{GF}(2^\gamma)$.

We implicitly associate a string $m \in \{0,1\}^n$ with a vector $(m_1, m_2, \ldots, m_u)$ where each $m_i \in \mathrm{GF}(2^\gamma)$; an integer $i$ with $0 \leq i < 2^\gamma$ will also be implicitly encoded into a field element in $\mathrm{GF}(2^\gamma)$. We construct an RPE scheme $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ as follows:

- $\mathsf{E}(m)$: Let $d = t + u - 1$. Choose a random degree-$d$ polynomial $q$ over $\Sigma$ such that $q(\ell+1) = m_1, \ldots, q(\ell+u) = m_u$ and output $w = (q(1), q(2), \ldots, q(\ell))$.
- $\mathsf{D}(w')$: Decode $w'$ using the Berlekamp-Welch algorithm (see Appendix **??**) 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 $S = \{i_1, \ldots, i_t\}$. Compute the degree-$d$ polynomial $q$ such that $q(i_1) = \alpha_1, \ldots, q(i_t) = \alpha_t$ and $q(\ell{+}1) = m_1, \ldots, q(\ell{+}u) = m_u$ by using Lagrange interpolation polynomial (see Appendix **??**). Output $(q(1), \ldots, q(\ell))$.

The error correction property holds since we simply use the Reed-Solomon code with set of codewords $\mathcal{W}$ in encoding and decoding, where

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

Note that $\mathcal{W}$ is a code over the alphabet $\Sigma$ with minimum relative distance $\frac{\ell-d+1}{\ell} > \delta$, which means we may efficiently correct up to a $\delta/2$ fraction of errors. Secrecy and reconstruction properties hold since the codeword $(q(1), \ldots, q(\ell))$ is a $(t+u)$-out-of-$\ell$ secret-sharing of $m$ using Shamir's secret-sharing scheme, and $(\alpha_1, \ldots, \alpha_t, m_1, \ldots, m_u)$ allows the reconstruction of the (one and only) degree-$d$ polynomial.

**Decoding from partial views with better parameters.** By using the property of Reed-Solomon codes, we can obtain better parameters in terms of decoding from partial views. In particular, Lemma 2 will allow us to decode from partial views of size $s < \ell/2$, whereas Lemma 1 only allows decoding from partial views of size $s > 3\ell/4$. We illustrate concretely the improvements gained by using the specialized Lemma 2 below for Reed-Solomon codes, as opposed to using the generic Lemma 1 for any RPE, in Example instantiations 1 and 2 below.

**Lemma 2.** *Let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be an RPE scheme with parameters $(n, \ell, \delta, t, \Sigma)$ according to Construction 1; recall $\gamma = \log|\Sigma|$, $u = \lceil n/\gamma \rceil$, and $\ell \geq \frac{t+u-1}{1-\delta}$. Let $\lambda$ be an arbitrary number such that $\frac{t+u}{\ell} < \lambda \leq 1$ and $\delta' = \frac{s-(t+u-1)}{s}$. Then, there is an efficient procedure $\mathsf{D}'$ satisfying the following:*

> *For any set $V \subseteq [\ell]$ with size $s = \lambda\ell$, for any $w \in \mathcal{W}$, and for any $w' \in \Sigma^s$ that is $(1-\delta'/2)$-close to $w|_V$ where, it holds that $\mathsf{D}'(V, w')$ outputs $w$ along with a consistent $m$.*

*Proof.* Let $V = \{i_1, \ldots, i_s\}$. Then,

$$\mathcal{W}' = \{\ (q(i_1), \ldots, q(i_s)) \mid q \text{ is a degree-}d \text{ polynomial }\}$$

forms another Reed-Solomon code, where $d = t + u - 1$. Note that $\mathcal{W}'$ is a code over the alphabet $\mathrm{GF}(2^\gamma)$ with minimum relative distance $\frac{s-d+1}{s} > \delta'$, which means we can decode $w'$ using the Berlekamp-Welch algorithm, correcting up to a $\delta'/2$ fraction of errors. □

**Example instantiation 1.** By applying Construction 1 with $\delta = 0.9$ and $\gamma = n$, we obtain an RPE with parameters $(n, 10t, 0.9, t, \mathrm{GF}(2^n))$. According to Lemma 2 with $\lambda = 0.3$, the scheme can decode a partial codeword of length $3t$, correcting up to $\delta'/2 = 1/3$ fraction of errors. On the other hand, applying the generic Lemma 1, we would only be able to decode a partial codeword of length $8t$, correcting up to $\delta'/2 \approx 0.2$-fraction of errors.

**Example instantiation 2 (constant-rate RPE).** By applying Construction 1 with $\delta = 0.9$ and $\gamma = n/t$, we obtain an RPE with parameters $(n, 20t, 0.9, t, \mathrm{GF}(2^\gamma))$ with rate 0.05. According to Lemma 2 with $\lambda = 0.3$, the scheme can decode a partial codeword of length $6t$, correcting up to $\delta'/2 = 1/3$ fraction of errors. On the other hand, applying the generic Lemma 1, we would only be able to decode a partial codeword of length $16t$, correcting up to $\delta'/2 \approx 0.2$-fraction of errors.

# 5 Non-malleable Encryption from Semantic Security

## 5.1 Generic Construction Using Any RPE

Given a semantically secure encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an RPE, we construct a non-malleable encryption scheme $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$, summarized in Figure 2 and described as follows. We remark that $\mathsf{NMEnc}$ and $\mathsf{NMDec}$ has black-box access to $\mathsf{Gen}$ (as a one-way function) in order to use one-time signatures; recall that one-time signature schemes can be constructed in a black-box way from one-way functions [Rom90, Lam79].

**Key generation.** Let $k$ be the security parameter. Let $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be an RPE scheme with parameters $(n, \ell, \delta, t, \Sigma)$ and let $\lambda$ and $\delta'$ be the parameter associated with decoding the partial views. In addition, set $t = k$.

The public key contains an error correcting code $C : \Gamma^t \to \Gamma^\ell$ with distance $\lambda\ell$, where $\Gamma$ is an appropriately chosen finite alphabet in order to satisfy the distance condition. Let $g = |\Gamma|$ and we will implicitly associate $\Gamma = [g]$. According to the Gilbert-Varshamov bound [Gil52, Var57], for any $\lambda < 1 - 1/g$ and for any $R = t/\ell$ with $R < 1 - H_g(\lambda)$, there is an error correcting code with rate $R$ and relative distance $\lambda$, where $H_g(\lambda) = \lambda \log_g(g-1) - \lambda \log_g \lambda - (1-\lambda)\log_g(1-\lambda)$. We note this technique was used in [RS10]. In addition, there are $g \cdot \ell$ public keys from $\mathsf{Gen}$ indexed by a pair $(j, b) \in [\ell] \times [g]$, that is, $\{\text{PK}_j^b \mid (j,b) \in [\ell] \times [g]\}$.

The secret key contains the decryption keys $\text{SK}_j^b$ and a random subset $S$ of $[\ell]$ with size $t$ to be used in decryption for consistency checks (described below).

**Encryption.** Encryption of a message $m \in \{0,1\}^n$ proceeds as follows:

1. Generate ($\text{SKSIG}, \text{VKSIG}$) for a one-time signature where $\text{VKSIG} \in \Gamma^t$, and compute $(v_1, \ldots, v_\ell) \leftarrow C(\text{VKSIG})$.
2. Compute $(s_1, \ldots, s_\ell) \leftarrow \mathsf{E}(m)$ and compute an $\ell$-long vector $\vec{c} = (c_1, \ldots, c_\ell)$ of ciphertexts where $c_j = \mathsf{Enc}_{\text{PK}_j^{v_j}}(s_j)$:
$$\vec{c} = \left(\mathsf{Enc}_{\text{PK}_1^{v_1}}(s_1), \mathsf{Enc}_{\text{PK}_2^{v_2}}(s_2), \ldots, \mathsf{Enc}_{\text{PK}_\ell^{v_\ell}}(s_\ell)\right)$$
3. Create a signature $\sigma$ on $\vec{c}$ using $\text{SKSIG}$. The ciphertext is $[\text{VKSIG}, \vec{c}, \sigma]$.

**Decryption.** To decrypt, we verify the signature and perform consistency checks. A valid ciphertext in $\Pi$ is an encryption of a codeword in $\mathcal{W}$. We want to design consistency checks that reject ciphertexts that are "far" from being valid ciphertexts under $\Pi$. For simplicity, we will describe the consistency checks as applied to the underlying vector of plaintexts. The checks depend on a random subset $S$ of $t$ columns chosen during key generation.

`decoding-check`: Let $I = \{1, \ldots, \lambda\ell\}$. We find a codeword $w$ such that $w|_I$ is $(1 - \frac{\delta'}{2})$-close to the first $\lambda\ell$ elements of the vector $(s_1, \ldots, s_\ell)$; the check fails if no such $w$ exists. Recall that according to Lemma 1, it can correct up to a $\delta'/2$ fraction of errors in $(s_1, \ldots, s_{\lambda\ell})$.

`codeword-check`: We check that the vector $(s_1, \ldots, s_\ell)$ agrees with $w$ at the positions indexed by $S$.

Parameters: $n$ is the length of plaintexts (in bits), and $k$ is the security parameter. To achieve NM-CPA, set an RPE parameter $t = k$ and to achieve NM-$q$-CCA2, set $t = a(k + q(k))$, where $a$ is a constant such that $\left(1 - \frac{\lambda\delta'}{2}\right)^a \leq \frac{1}{2}$. Choose the other RPE parameters appropriately to ensure the existence of an RPE scheme satisfying all RPE parameters, e.g., according to Construction 1.

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme, $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$ be a strong one-time signature scheme, and $(\mathsf{E}, \mathsf{D}, \mathsf{R})$ be a reconstructable probabilistic encoding scheme with parameters $(n, \ell, \delta, t, \Sigma)$. Moreover, let $\lambda$ and $\delta'$ be the parameters associated with decoding the partial views.

In order to provide a better sense of how the actual encryption scheme would look, we will use Example instantiation 2 in Section 4.3 and describe specific RPE parameters in boxes that would be used for a NM-CPA encryption scheme with $n = \omega(k)$.

Use RPE with parameters $(n, 20k, 0.9, k, \mathrm{GF}(2^{n/k}))$; that is, $\ell = 20k$ and $\delta = 0.9$, and the message space for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathrm{GF}(2^{n/k})$. Set $\lambda = 0.3$ and $\delta' = 2/3$.

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

1. Choose an error correcting code $C : \Gamma^t \to \Gamma^\ell$ with distance $\lambda\ell$, where $\Gamma$ is an appropriately chosen finite alphabet to satisfy the distance condition. Let $g = |\Gamma|$, and we will implicitly associate $\Gamma$ with $[g]$.

   Choose an error correcting code $C : \{0,1\}^k \to \{0,1\}^{20k}$ with distance $6k$.

2. For $j \in [\ell], b \in [g]$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\mathrm{PK}_j^b, \mathrm{SK}_j^b)$.
3. Pick a random subset $S \subset [\ell]$ of size $t$.
4. Set $\mathrm{PK} = \Big(C, \{\mathrm{PK}_j^b \mid j \in [\ell], b \in [g]\}\Big)$ and $\mathrm{SK} = \Big(S, \{\mathrm{SK}_j^b \mid j \in [\lambda\ell] \cup S, b \in [g]\}\Big)$.

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

1. Run $\mathsf{GenSig}(1^k)$ to generate $(\mathrm{SKSIG}, \mathrm{VKSIG})$. Parse $\mathrm{VKSIG}$ as an element $\mathrm{VKSIG} \in \Gamma^t$. Let $C(\mathrm{VKSIG}) := (v_1, \ldots, v_\ell)$, where $v_1, \ldots, v_\ell \in \Gamma^\ell$.
2. Compute $(s_1, \ldots, s_\ell) \leftarrow \mathsf{E}(m)$, where $m \in \{0,1\}^n$. Compute the ciphertext $c_j \leftarrow \mathsf{Enc}_{\mathrm{PK}_j^{v_j}}(s_j)$, for $j \in [\ell]$.
3. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SKSIG}}(\vec{c})$ where $\vec{c} = (c_1, \ldots, c_\ell)$.
4. 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 $\vec{c} = (c_j)$ and $(v_1, \ldots, v_\ell) = C(\mathrm{VKSIG})$. For $j \in [\lambda\ell]$, compute $s_j = \mathsf{Dec}_{\mathrm{SK}_j^{v_j}}(c_j)$. Compute $((w_1, \ldots, w_\ell), m) \leftarrow \mathsf{D}'([\lambda\ell], (s_1, \ldots, s_{\lambda\ell}))$. If the decoding fails or $(w_1, \ldots, w_{\lambda\ell})$ is $\frac{\delta'}{2}$-far from $(s_1, \ldots, s_{\lambda\ell})$, then output $\perp$.

   We have $(v_1, \ldots, v_{20k}) = C(\mathrm{VKSIG})$. For $j \in [6k]$, compute $s_j = \mathsf{Dec}_{\mathrm{SK}_j^{v_j}}(c_j)$. Apply the Berlekamp-Welch algorithm to $(s_1, \ldots, s_{6k})$ to recover a degree $(2k - 1)$ polynomial $p$; if it fails, output $\perp$. Otherwise, let $(w_1, \ldots, w_{20k}) = (p(1), \ldots, p(20k))$ and $(m_1, \ldots, m_k) = (p(20k+1), \ldots, p(20k+k))$. If $(w_1, \ldots, w_{6k})$ is $\frac{1}{3}$-far from $(s_1, \ldots, s_{6k})$, then output $\perp$.

3. (`codeword-check`) Compute $s_j = \mathsf{Dec}_{sk_j^{v_j}}(c_j)$ for all $j \in S$. Check that $s_j = w_j$.
4. If all the checks accept, output the message $m$ corresponding to the codeword $w$; else, output $\perp$.

Figure 2: The Non-Malleable Encryption Scheme $\Pi$. We annotate the construction with specific parameters when the message length is $\omega(k)$ and our RS-based RPE is used.

Finally, if all the checks accept, decode the codeword $w$ and output the result; otherwise output $\perp$.

We note that we only need a partial set of the decryption keys, in particular for $I$ and $S$, in order to complete the decryption procedure.

### 5.2 Using Construction 1 for RPE

By plugging our RS-based RPE in Construction 1, using parameters $(n, 20k, 0.9, k, \mathrm{GF}(2^{n/k}))$, into the above generic NM-CPA construction, we obtain an NM-CPA encryption scheme for messages of length $\omega(k)$. In Figure 2, we briefly annotate the generic construction with the parameters for this RPE, in order to provide a better sense of how the actual NM-CPA scheme would look. The underlying IND-CPA scheme encrypts an element of $\mathrm{GF}(2^{n/k})$, and there are $20k$ of them in the overall NM-CPA ciphertext; if the underlying IND-CPA encryption is constant rate, the overall NM-CPA also achieves a constant rate. Note *we can use the binary alphabet for the error correcting code $C$, since it has a relative distance of $\lambda = 0.3$ and a rate of 0.05*; the Gilbert-Varshamov bound [Gil52, Var57] guarantees the existence of a linear code satisfying such distance and rate conditions.

When the message is of length $O(k)$, we can instantiate an NM-CPA encryption scheme by using an RPE with parameters $(n, 10k, 0.9, k, \mathrm{GF}(2^n))$. Note the overall NM-CPA ciphertext length becomes $\Theta(k^2)$. The underlying IND-CPA scheme encrypts an element of $\mathrm{GF}(2^n)$, and there are $10k$ of them in the overall NM-CPA ciphertext.

## 6 Analysis

**Theorem 1.** *If $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a semantically secure PKE, then the PKE scheme $\Pi = (\mathsf{NMGen}^{\mathsf{Gen}}, \mathsf{NMEnc}^{\mathsf{Gen},\mathsf{Enc}}, \mathsf{NMDec}^{\mathsf{Gen},\mathsf{Dec}})$ described in Figure 2 is non-malleable under a chosen plaintext attack.*

### 6.1 Proof of Main Theorem

In the hybrid argument, we consider the following variants of $\mathsf{NME}_b$ as applied to $\Pi$, where $\textsc{vksig}^* = (v_1^*, \ldots, v_\ell^*)$ denotes the verification key in the ciphertext $y = \mathsf{NMEnc}_{\mathrm{PK}}(m_b)$:

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

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

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

> $\mathsf{NMDec}^*_{\mathrm{SK}}([\vec{c}, \textsc{vksig}, \sigma])$:
>
> 1. (`sig-check`$^*$) Verify the signature with $\mathsf{VerSig}_{\textsc{vksig}}[\vec{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\textsc{vksig} = \textsc{vksig}^*$.

2. (`decoding-check`$^*$) Let $\vec{c} = (c_j)$ and $C(\textsc{vksig}) = (v_1, \dots, v_\ell)$. Let $X = (x_1, \dots, x_{\lambda\ell})$ be the smallest distinct values such that $v_{x_i} \neq v^*_{x_i}$. Note there must be these values since $C$ is an encoding with minimum distance $\lambda$. Compute $s_{x_i} = \mathsf{Dec}_{\mathrm{SK}_{x_i}^{v_{x_i}}}(c_{x_i})$, $i = 1, \dots, \lambda\ell$. Compute $w = (w_1, \dots, w_\ell) \in \mathcal{W}$ such that $(w_{x_1}, \dots, w_{x_{\lambda\ell}})$ is least $(1 - \frac{\delta'}{2})$-close to $(s_{x_1}, \dots, s_{x_{\lambda\ell}})$ by running $\mathsf{D}'(X, (s_{x_1}, \dots, s_{x_{\lambda\ell}}))$ based on the property of decoding from the partial view. If no such codeword exists, output $\perp$.
3. (`codeword-check`$^*$) For all $j \in S$, check that $\mathsf{Dec}_{\mathrm{SK}_j^{v_j}}(c_j) = w_j$.

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

**Claim 6.1.** For $b \in \{0,1\}$, $\left\{\mathsf{NME}_b(\Pi, A, k, p(k))\right\} \overset{c}{\approx} \left\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\right\}$.

*Proof.* Assume towards contradiction that there is a ppt machine $A$ that distinguishes $\mathsf{NME}_b(\Pi, A, k, p(k))$ and $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$ with non-negligible probability. We construct a ppt machine $B$ that breaks the security of the strong, one-time signature scheme, $\Pi' := (\mathsf{GenSig}, \mathsf{Sign}, \mathsf{VerSig})$.

First, note that since $A$ may not query the decryption oracle on the challenge ciphertext itself, experiments $\mathsf{NME}_b$ and $\mathsf{NME}_b^{(1)}$ can only differ (due to the difference between `sig-check` and `sig-check`$^*$) if the following event takes place:

> $A$ outputs a ciphertext $[\vec{c}, \textsc{vksig}, \sigma]$ such that the signature correctly verifies and $\textsc{vksig} = \textsc{vksig}^*$.

Thus, the above event must occur with non-negligible probability. We are now ready to describe the adversary $B$. $B$ participates externally in a $\mathsf{Forge}(\Pi', B, k)$ experiment with verification key $\textsc{vksig}^*$, while internally playing the role of challenger in the $\mathsf{NME}_b(\Pi, A, k, p(k))$ experiment. $B$ runs $(\textsc{pk}, \textsc{sk}) \leftarrow \mathsf{NMGen}(1^k)$, instantiates $A_1(\textsc{pk})$ and waits for $A_1$ to submit $(m_0, m_1, \textsc{state}_A)$. $B$ then chooses $b \leftarrow \{0,1\}$, sets the verification key to $\textsc{vksig}^*$, computes the first part of the challenge ciphertext, $\vec{c}^*$ honestly, and queries its signing oracle on $\vec{c}^*$ to obtain $\sigma^*$. $B$ then instantiates $A_2(y := [\vec{c}^*, \textsc{vksig}^*, \sigma^*], \textsc{state}_A)$ and waits for $A_2$ to output $(\psi_1, \dots, \psi_{p(k)})$. Upon receiving $(\psi_1, \dots, \psi_{p(k)})$, $B$ checks whether for any $\psi_i := [\vec{c}_i, \textsc{vksig}_i, \sigma_i], i \in [p(k)]$, it is the case that the signature correctly verifies and $\textsc{vksig}_i = \textsc{vksig}^*$. If yes, $B$ forwards $(\vec{c}_i, \sigma_i)$ to its external challenger. Recall that this event occurs with non-negligible probability, and so $B$ succeeds in forging with non-negligible probability, resulting in a contradiction to the security of $\Pi'$. □

**Claim 6.2.** For $b \in \{0,1\}$, $\left\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\right\} \overset{s}{\approx} \left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\}$.

*Proof.* We will show that both distributions are statistically close for all possible coin tosses in both experiments (specifically, those of $\mathsf{NMGen}$, $A$ and $\mathsf{NMEnc}$) except for the choice of $S$ in $\mathsf{NMGen}$. Once we fix all the coin tosses apart from the choice of $S$, the output $(\psi_1, \dots, \psi_{p(k)})$ of $A_2$ are completely determined and identical in both experiments $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$.

Recall the guarantees we would like from $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$:

- On input a ciphertext that is an encryption of a message $m$ under $\Pi$, both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ will output $m$ with probability 1.

- On input a ciphertext that is "close" to an encryption of a message $m$ under $\Pi$, both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ 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 $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ output $\perp$ with high probability.

To quantify and establish these guarantees, we consider the following promise problem $(\Pi_Y, \Pi_N)$ that again refers to the underlying vector of plaintexts. An instance is a vector of $\ell$ entries each of which lies in $\{0,1\}^n \cup \perp$.

$\Pi_Y$ (YES instances) — for some $w \in \mathcal{W}$, the instance equals $w$.

$\Pi_N$ (NO instances) — either the first $\lambda\ell$ elements of the instance is $\delta'/2$-far from the first $\lambda\ell$ elements of every codeword in $\mathcal{W}$ or the entire instance is $\frac{\lambda\delta'}{2}$-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 $\psi$ it outputs with some underlying vector $\vec{M}$ of plaintexts (which may be a YES or a NO instance or neither) and show that both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ agree on $\psi$ with high probability. To facilitate the analysis, we consider two cases:

- If $\vec{M} \in \Pi_N$, then it fails the codeword checks in both decryption algorithms with high probability, in which case both decryption algorithms output $\perp$.

  Specifically, if the first $\lambda\ell$ elements of $\vec{M}$ is $\delta'/2$-far from the first $\lambda\ell$ elements of every codeword in $\mathcal{W}$ then the decoding check in $\mathsf{NMDec}$ rejects $\vec{M}$ with probability 1. Moreover, being $\delta'/2$-far from the first $\lambda\ell$ elements for every codeword implies that $\vec{M}$ have at least $(\delta'/2) \cdot \lambda\ell$ different positions, where $c$ is some constant. Therefore, the codeword check in $\mathsf{NMDec}^*$ rejects $\vec{M}$ with probability at least $1 - \left(1 - \frac{\delta'\lambda}{2}\right)^t$, since the condition implies that $\vec{M}$ is $\frac{\delta'\lambda}{2}$-far from every codeword. From Lemma 1, both $\delta'$ and $\lambda$ are constant, and therefore with overwhelming probability in $t$, $\mathsf{NMDec}^*$ will reject $\vec{M}$ as well.

  On the other hand, if $\vec{M}$ is $\lambda\delta'/2$-far from every codeword, both codeword checks in $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ rejects $\vec{M}$ with probability $1 - (1 - \frac{\delta'\lambda}{2})^t$

  Therefore, both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ reject $\vec{M}$ with with probability at least $1 - 2 \cdot (1 - \frac{\delta'\lambda}{2})^k$, since we have $t = k$.

- 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 $S$. Note that the decoding check in both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ will be successful. This is because $\vec{M}$ is $(1 - \lambda\delta'/2)$-close to $w$, and there are at most $(\lambda\delta'/2) \cdot \ell$ erroneous positions compared with some codeword in $\mathcal{W}$. This implies that any $\lambda\ell$ elements of $\vec{M}$ has at most $\frac{(\lambda\delta'/2)\cdot\ell}{\lambda\ell} = \frac{\delta'}{2}$ fraction of errors. Moreover, the codeword check is the same in both $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$. As such, both decryption algorithms output $\perp$ with exactly the same probability, and whenever they do not output $\perp$, they output the same message $m$.

From the above analysis, the two hybrids are statistically close. □

**Claim 6.3.** Let $(n, \ell, \delta, t, \Sigma)$ be the parameters of an RPE scheme that $\Pi$ uses in Figure 2. Then, for every ppt machine $A$ and for any polynomial $p$, there exists a ppt machine $B$ such that for $b \in \{0,1\}$, it holds

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

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

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

- (pre-processing) Pick a random subset $S = \{u_1, \ldots, u_t\}$ of $[\ell]$. Choose an ECC $C$, and run $\mathsf{GenSig}(1^k)$ to generate $(\textsc{sksig}^*, \textsc{vksig}^*)$ and set $(v_1^*, \ldots, v_\ell^*) = C(\textsc{vksig}^*)$. Let $\phi : \{j \mid j \in [\ell] \setminus S\} \to [\ell - t]$ be a bijection.
- (key generation) $B$ receives $\langle \textsc{pk}_1, \ldots, \textsc{pk}_{\ell-t}\rangle$ from $\mathsf{mIND}_b$ and simulates $\mathsf{NMGen}$ as follows: for all $j \in [\ell], \beta \in [g]$,

$$(\textsc{pk}_j^\beta, \textsc{sk}_j^\beta) = \begin{cases} (\textsc{pk}_{\phi(j)}, \perp) & \text{if } \beta = v_j^* \text{ and } j \notin S \\ \mathsf{Gen}(1^k) & \text{otherwise} \end{cases}$$

- (message selection) Let $(m^0, m^1)$ be the pair of messages $A_1$ returns. $B$ chooses $(\alpha_1, \ldots, \alpha_t) \leftarrow \Sigma^t$ uniformly at random and then computes

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

  Recall that $\mathsf{R}$ is the reconstruction algorithm of the underlying RPE scheme. For $j \in S$, let $\gamma_j = w_j^0 = w_j^1$. $B$ forwards $(\langle m_1^0, \ldots, m_{\ell-t}^0\rangle, \langle m_1^1, \ldots, m_{\ell-t}^1\rangle)$ to $\mathsf{mIND}_b$, where $m_{\phi(j)}^b = w_j^b$, for $j \in [\ell] \setminus S$.
- (ciphertext generation) $B$ receives $\langle y_1, \ldots, y_{\ell-t}\rangle$ from $\mathsf{mIND}_b$ (according to the distribution $\mathsf{Enc}_{\textsc{pk}_1}(m_1^b), \ldots, \mathsf{Enc}_{\textsc{pk}_{\ell-t}}(m_{\ell-t}^b)$) and generates a ciphertext $[\vec{c}, \textsc{vksig}^*, \sigma]$ as follows:

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

  $B$ then computes the signature $\sigma \leftarrow \mathsf{Sign}_{\textsc{sksig}^*}(\vec{c})$ and forwards $[\vec{c}, \textsc{vksig}^*, \sigma]$ to $A_2$. It is straight-forward to verify that $[\vec{c}, \textsc{vksig}^*, \sigma]$ is indeed a random encryption of $m_b$ under $\Pi$.
- (decryption) Upon receiving a sequence of ciphertexts $(\psi_1, \ldots, \psi_{p(k)})$ from $A_2$, $B$ decrypts these ciphertexts using $\mathsf{NMDec}^*$ as in $\mathsf{NME}_b^{(2)}$.

Note that the view of $A$ in the original experiment is identically distributed to the one that $B$ simulates. We conclude the proof by pointing out that $B$ perfectly simulates $\mathsf{NMDec}^*$, since it has the secret keys $\{\textsc{sk}_j^\beta \mid \beta \neq v_j^* \text{ or } j \in S\}$, which $B$ generated by itself. □

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

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

We have $\mathsf{mIND}_0(\mathcal{E}, B, k, \ell - t) \overset{c}{\approx} \mathsf{mIND}_1(\mathcal{E}, B, k, \ell - t)$, due to semantic security of the underlying encryption scheme, which concludes the proof.

## 7 Achieving Bounded-CCA2 Non-Malleability

We describe how our scheme may be modified to achieve non-malleability under a bounded-CCA2 attack. Recall that, informally, an encryption scheme is non-malleable against a $q$-bounded CCA2 attack if the adversary is allowed to query $\mathsf{Dec}$ adaptively at most $q(k)$ times in the non-malleability experiment. Our modification is the straight-forward analogue of the [CHH$^+$07] modification of the [PSV06] scheme: We change the parameter $(n, \ell, \delta, t, \Sigma)$ of the underlying RPE scheme such that $t = a \cdot (k + q(k))$, where $a$ is a constant such that $\left(1 - \frac{\lambda\delta'}{2}\right)^a \leq \frac{1}{2}$. See Figure 2 for more details.

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

- Experiment $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$: This experiment proceeds exactly like $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b$, except we replace `sig-check` in $\mathsf{NMDec}$ with `sig-check`$^*$ as described in Section 6.
- Experiment $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}$: This experiment proceeds exactly like $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$ except we replace $\mathsf{NMDec}$ with $\mathsf{NMDec}^*$ as described in Section 6.

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 6.1. Moreover, it holds that $\left\{\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}(\Pi, A, k, p(k))\right\}$ and $\left\{\mathsf{mIND}_b(\mathcal{E}, B, k, \ell - t)\right\}$ are identically distributed for each $b \in \{0,1\}$, which can be shown using the reduction in the proof of Claim 6.3. (Recall that the value $p(k)$ in the various $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}$ experiments corresponds to the number of (mauled) ciphertexts that the adversary would come up with, after given the challenge ciphertext.) Therefore, we are only left to show the following claim to conclude the analysis.

**Claim 7.1.** 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 vector of $c$.

As before, we will show that both distributions are statistically close for all possible coin tosses in both experiments (specifically, those of $\mathsf{NMGen}, A$ and $\mathsf{NMEnc}$) except for the choice of $S$ in $\mathsf{NMGen}$. Fix all the coin tosses apart from the choice of $S$. Here, however, unlike the case of chosen plaintext attacks, we cannot immediately deduce that the outputs of $A_2$ in both experiments are completely determined and identical, since they depend on the adaptively chosen queries to $\mathsf{NMDec}$, 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 $\mathsf{NMDec}/\mathsf{NMDec}^*$ is at most two (since we fixed all the coin tosses except $S$). Moreover, if a query $c$ is such that $\vec{M}_c \in \Pi_N$, $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ will both give only one response of $\perp$ with overwhelming probability, according to the analysis in Claim 6.2.

This leads us to consider a binary tree of depth $q$ that corresponds informally to "unrolling" the $q$ adaptive queries that $A$ makes to $\mathsf{NMDec}/\mathsf{NMDec}^*$ in the experiments $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}/\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}$. The root node of the tree corresponds to the first query $A$ makes to $\mathsf{NMDec}/\mathsf{NMDec}^*$, and each edge from a node to its child is labeled with the answer of $\mathsf{NMDec}/\mathsf{NMDec}^*$ 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 $\mathsf{NMDec}/\mathsf{NMDec}^*$ 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 $\mathsf{NMDec}/\mathsf{NMDec}^*$ responding with a valid decryption (in which case the value returned is independent of $S$) and the other case of 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 $\mathsf{NMDec}/\mathsf{NMDec}^*$'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 $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ behave identically for queries $c$ with $\vec{M}_c \notin \Pi_N$ as shown in Claim 6.2, the computation tree is $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(1)}$ is identical to that in $\mathsf{NME}\text{-}q\text{-}\mathsf{CCA}_b^{(2)}$. Note also that $A$ makes at most $q$ adaptive queries to $\mathsf{NMDec}$, and therefore 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 $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ output $\perp$. We have

  $$\Pr_S[\mathsf{good}(S)] \geq 1 - 2 \cdot (2^{q+1}p(k)) \cdot \Big(1 - \frac{\lambda\delta'}{2}\Big)^t \geq 1 - 2 \cdot (2^{q+1}p(k)) \cdot \frac{1}{2^{k+q}} = 1 - \mathsf{negl}(k).$$

  This follows from a union bound over these ciphertexts in the tree and the analysis in Claim 6.2.
- 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 $\mathsf{NMDec}$ and $\mathsf{NMDec}^*$ always output the same answer for any $\vec{M} \notin \Pi_N$ as explained in the analysis in Claim 6.2.

□

**Acknowledgement.** We thank Marshall Ball for useful discussions. The first author was supported in part by the Office of Naval Research (ONR) awards N0001416WX01489 and N0001416WX01645, and National Science Foundation (NSF) award #1618269. The second author was supported in part by NSF CAREER award #CNS-1453045 and by a Ralph E. Powe Junior Faculty Enhancement Award. The third author 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. The fourth author 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.

## References

[BDSKM16] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. Lecture Notes in Computer Science, pages 881–908. Springer, 2016.

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

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008.

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 519–536. Springer, August 1999.

[CDD$^+$15] Ronald Cramer, Ivan Bjerre Damgård, Nico Döttling, Serge Fehr, and Gabriele Spini. Linear secret sharing schemes from error correcting codes and universal hash functions. Lecture Notes in Computer Science, pages 313–336. Springer, 2015.

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

[CDMW16] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. A black-box construction of non-malleable encryption from semantically secure encryption. 2016. Full version of [CDMW08], eprint/2016/720.

[CDTV16] Sandro Coretti, Yevgeniy Dodis, Björn Tackmann, and Daniele Venturi. Non-malleable encryption: Simpler, shorter, stronger. Lecture Notes in Computer Science, pages 306–335. Springer, 2016.

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bitwise and split-state tampering. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 440–464. Springer, February 2014.

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

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

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

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

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

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

[FY92] Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *24th Annual ACM Symposium on Theory of Computing*, pages 699–710. ACM Press, May 1992.

[GHKW16] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. Lecture Notes in Computer Science, pages 1–27. Springer, 2016.

[Gil52] E.N. Gilbert. A comparison of signalling alphabets. *Bell Systems Technical Journal*, 31:504–522, 1952.

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

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

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

[HHK10] Javier Herranz, Dennis Hofheinz, and Eike Kiltz. Some (in)sufficient conditions for secure hybrid encryption. *Inf. Comput.*, 208(11):1243–1257, 2010.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 590–607. Springer, August 2012.

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

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

[KW15] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. Lecture Notes in Computer Science, pages 101–128. Springer, 2015.

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

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

[LPJY14] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 514–532. Springer, May 2014.

[LPJY15] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. Lecture Notes in Computer Science, pages 681–707. Springer, December 2015.

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

[PSV06] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *Advances in Cryptology -*

*CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 271–289, 2006.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 187–196. ACM Press, May 2008.

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

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

[RS10] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM J. Comput.*, 39(7):3058–3088, 2010.

[Var57] R.R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akad. Nauk SSSR*, 117:739–741, 1957.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51st Annual Symposium on Foundations of Computer Science*, pages 531–540. IEEE Computer Society Press, October 2010.

[Wee12] Hoeteck Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 246–262. Springer, April 2012.