Black-Box Construction of a Non-Malleable Encryption Scheme from Any Semantically Secure One

Seung Geol Choi, Dana Dachman-Soled, Tal Malkin*, and Hoeteck Wee

Columbia University
{sgchoi,dglasner,tal,hoeteck}@cs.columbia.edu

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 encryptions; instead, we encrypt an encoding of the message with certain locally testable and self-correcting properties. We exploit the fact that low-degree polynomials are simultaneously good error-correcting codes and a secret-sharing scheme.

Key words: Public-key encryption, semantic security, non-malleability, black-box constructions.

1 Introduction

The most basic security guarantee we require of a public key encryption scheme is that of semantic security [GM84]: it is infeasible to learn anything about the plaintext from the ciphertext. In many cryptographic applications such as auctions, we would like an encryption scheme that satisfies the stronger guarantee of non-malleability [DDN00], namely that given some ciphertext c, it is also infeasible to generate ciphertexts of some message that is related to the decryption of c. Motivated by the importance of non-malleability, Pass, Shelat and Vaikuntanathan raised the following question [PSV06]:

> It is possible to *immunize* any semantically secure encryption scheme against malleability attacks?

Pass et al. gave a beautiful construction of a non-malleable encryption scheme from any semantically secure one (building on [DDN00]), thereby addressing the question in the affirmative. However, the PSV construction – as with previous constructions

* The work was partially supported by NSF grants CNS-0716245, CCF-0347839, and SBE-0245014.

achieving non-malleability from general assumptions [DDN00,S99,L06] – suffers from the curse of inefficiency arising from the use of general NP-reductions. In this work, we show that we can in fact immunize any semantically secure encryption schemes against malleability attacks without paying the price of general NP-reductions:

> **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 (from programming language lingo) that given any subroutines for computing a semantically secure encryption scheme, computes a non-malleable encryption scheme, with a multiplicative overhead in the running time that is quasi-linear in the security parameter. Before providing further details, let us first provide some background and context for our result.

1.1 Relationships amongst Cryptographic Primitives

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

Black-box usage: A construction is black-box if it refers only to the input/output behavior of the underlying primitive; we would typically also require that in the proof of security, we can use an adversary breaking the security of the construction as an oracle to break the underlying primitive. (See [RTV04] and references within for more details.). As emphasized earlier, our construction is black-box, using only oracle access to the key generation, encryption and decryption functionality of the underlying encryption scheme.

Non-black-box usage: A construction is non-black-box if it uses the code computing the functionality of the primitive. The PSV construction along with the work it builds on fall into this category: they use an NP reduction applied to the circuit computing the encryption functionality of the underlying encryption scheme in order to provide a non-interactive zero-knowledge proof of consistency.

Motivated by the fact that the vast majority of constructions in cryptography are black-box, a rich and fruitful body of work initiated in [IR89] seeks to understand the power and limitations of black-box constructions in cryptography, resulting in a fairly complete picture of the relations amongst most cryptographic primitives with respect to black-box constructions (we summarize several of the known relations pertaining to encryption in Figure 1). More recent work 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. Two notable examples are general secure multi-party computation against a dishonest majority and encryption schemes secure against adaptive chosen-ciphertext (CCA2) attacks[1] (c.f. [GMW87,DDN00]).

The general question of whether we can securely realize these tasks via black-box access to a general primitive is not merely of theoretical interest. A practical reason is related to efficiency, as non-black-box constructions tend to be less efficient due to the use of general NP reductions to order to prove statements in zero knowledge; this impacts both computational complexity as well as communication complexity (which we interpret broadly to mean message lengths for protocols and key size and ciphertext size for encryption schemes). Moreover, if resolved in the affirmative, we expect the solution to provide new insights and techniques for circumventing the use of NP reductions and zero knowledge in the known constructions. Finally, given that there has been no formal model that captures non-black-box constructions in a satisfactory manner, the pursuit of a positive result becomes all the more interesting.

Indeed, Ishai et al. [IKLP06] recently provided an affirmative answer for secure multi-party computation by exhibiting black-box constructions from some low-level primitive. Their techniques have since been used to yield secure multi-party computation via black-box access to an oblivious transfer protocol for semi-honest parties, which is complete (and thus necessary) for secure multi-party computation [H08]. This leaves the following open problem:

> Is it possible to realize CCA2-secure encryption via black-box access to a low-level primitive, e.g. enhanced trapdoor permutations or homomorphic encryption schemes?

Previous work pertaining to this problem is limited to non-black-box constructions of CCA2-secure encryption from enhanced trapdoor permutations [DDN00,S99,L06]; nothing is known assuming homomorphic encryption schemes. In work concurrent with ours, Peikert and Waters [PW07] made substantial progress towards the open problem – they constructed CCA2-secure encryption schemes via black-box access to a new primitive they introduced called lossy trapdoor functions, and in addition, gave constructions of this primitive from number-theoretic and worst-case lattice assumptions. Unfortunately, they do not provide a black-box construction of CCA2-secure encryption from enhanced trapdoor permutations.

Our work may also be viewed as a step towards closing this remaining 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

[1] These are encryption schemes that remain semantically secure even under a CCA2 attack, wherein the adversary is allowed to query the decryption oracle except on the given challenge. A CCA1 attack is one wherein the adversary is allowed to query the decryption oracle before (but not after) seeing the challenge.

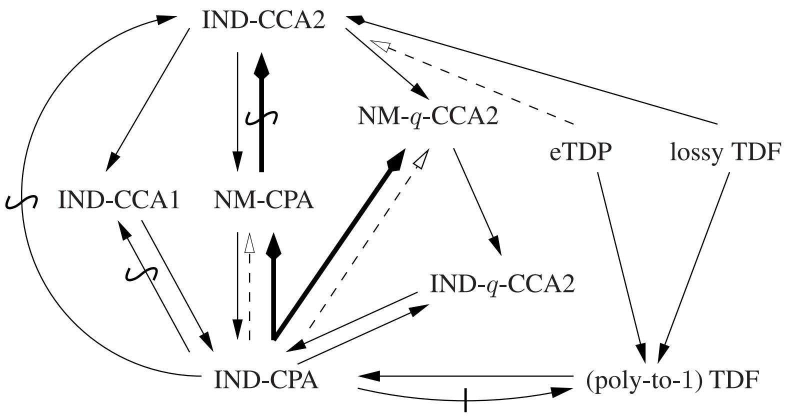

Fig. 1. Known relations among generic encryption primitives, and our results. Solid lines indicate black box constructions, and dotted lines indicate non-black-box constructions (c.f. [BHSV98,DDN00,PSV06,CHH+07,PW07]). The separations are with respect to black-box reductions, or black box shielding reductions (c.f. [GMR01,GMM07]). Our contributions are indicated with the thick arrows.

notion of non-adaptive soundness, which is sufficient for non-malleability but not for CCA2-security (c.f. [PSV06]). As a special case of our result, we obtain a black-box construction of non-malleable encryptions from any (poly-to-1) trapdoor function. Our results are incomparable with those of Peikert and Waters since we start from weaker assumptions but derive a weaker security guarantee.

Related positive results. A different line of work focuses on (very) efficient constructions of CCA2-secure encryptions under specific number-theoretic assumptions [CS98,CS04,CHK04]. Apart from those based on identity-based encryption, these constructions together with previous ones based on general assumptions can be described under the following framework (c.f. [BFM88,NY90,RS91,ES02]). Start with some cryptographic hardness assumption that allows us to build a semantically secure encryption scheme, and then prove/verify that several ciphertexts satisfy certain relations in one of two ways:

– exploiting algebraic relations from the underlying assumption to deduce additional structure in the encryption scheme (e.g. homomorphic, reusing randomness) [CS98,CS04];
– apply a general NP reduction to prove in non-interactive zero knowledge (NIZK) statements that relate to the primitive [DDN00,S99,L06].

None of the previous approaches seems to yield black-box constructions under general assumptions. Indeed, our work (also [PW07]) does not use the above framework.

1.2 Our Results

As mentioned earlier, we exhibit a black-box construction of a non-malleable encryption scheme from any semantically secure one, the main novelty being that our construction is black-box. 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 Berlekamp-Welch for decryption. The running time incurs a multiplicative overhead that is quasi-linear in the security parameter, over the running time of the underlying CPA secure scheme. Moreover, the sizes of public keys and ciphertext are independent of the computational complexity of the underlying scheme.
- *Conceptual simplicity/clarity.* Our scheme (and the analysis) is arguably much simpler than many of the previous constructions, and like [PSV06], entirely self-contained (apart from the Berlekamp-Welch algorithm). We do not need to appeal to notions of zero-knowledge, 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.

We may also derive from our construction additional positive and negative results.

Bounded CCA2 non-malleability. Cramer et al. [CHH+07] introduced the bounded CCA2 attack, a relaxation of the CCA2 attack wherein the adversary is only allowed make an a-priori bounded number of queries q to the decryption oracle, where q is fixed prior to choosing the parameters of the encryption scheme. In addition, starting from any semantically secure encryption, they obtained[2]:

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

Combining their approach for the latter construction with our main result, we obtain an encryption scheme that is non-malleable under a bounded-CCA2 attack via a black-box construction, wherein the size of the public key and ciphertext are linear in q.

[2] While semantic security and non-malleability are equivalent under a CCA2 attack [DDN00], they are not equivalent under a bounded-CCA2 attack, as shown in [CHH+07].

Separation between CCA2 security and non-malleability. Our main construction has the additional property that the decryption algorithm does not query the encryption functionality of the underlying scheme. Gertner, Malkin and Myers [GMM07] referred to such constructions as shielding and they showed that there is no shielding black-box construction of CCA1-secure encryption schemes from semantically secure encryption. Combined with the fact that any shielding construction when composed with our construction is again shielding, this immediately yields the following:

> **Corollary (informal)** There exists no shielding black-box construction of CCA1-secure encryption schemes from non-malleable encryption schemes.

Note that a CCA2-secure encryption scheme is trivially also CCA1-secure, so this also implies a separation between non-malleability and CCA2-security for shielding black-box constructions.

Our techniques. At a high level, we follow the cut-and-choose approach for consistency checks from [PSV06], wherein the randomness used for cut-and-choose is specified in the secret key. A crucial component of our construction is a message encoding scheme with certain locally testable and self-correcting properties, based on the fact that low-degree polynomials are simultaneously good error-correcting codes and a secret-sharing scheme; this has been exploited in the early work on secure multi-party computation with malicious adversaries [BGW88]. We think this technique may be useful in eliminating general NP-reductions in other constructions in cryptography (outside of public-key encryption).

Towards CCA2 Security? The main obstacle towards achieving full CCA2 security from either semantically secure encryptions or enhanced trapdoor permutations using our approach (and also the [PSV06] approach) lies in guaranteeing soundness of the consistency checks against an adversary that can adaptively determine its queries depending on the outcome of previous consistency checks. It seems conceivable that using a non-shielding construction that uses re-encryption may help overcome this obstacle.

1.3 Overview of our Construction

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

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

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

We circumvent both issues by using a more sophisticated encoding of the message m based on low-degree polynomials instead of merely making k copies of the message as in the above schemes. Specifically, we pick a random degree k polynomial p such that $p(0) = m$ and we construct a $k \times 10k$ matrix such that the i'th column of the matrix comprises entirely of the value $p(i)$. 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 issue that only a tiny number of ciphertexts are inconsistent is handled using the error-correcting properties of low-degree polynomials; specifically, each row of a valid encoding is a codeword for the Reed-Solomon code (and we output $\perp$ if it's far from any codeword).
- Low-degree polynomials are also good secret-sharing schemes, and learning a random subset of k columns in a valid encoding reveals nothing about the underlying message m. Encoding m using a secret-sharing scheme appears in the earlier work of Cramer et al. [CHH+07], but they do not consider redundancy or error-correction.

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

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

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

2 Preliminaries & Definitions

Notation. We adopt the notation used in [PSV06]. We use $[n]$ to denote $\{1, 2, \ldots, n\}$. If A is a probabilistic polynomial time (hereafter, ppt) algorithm that runs on input x, $A(x)$ denotes the random variable according to the distribution of the output of A on input x. We denote by $A(x; r)$ the output of A on input x and random coins r. Computational indistinguishability between two distributions A and B is denoted by $A \stackrel{c}{\approx} B$ and statistical indistinguishability by $A \stackrel{s}{\approx} B$.

2.1 Semantically Secure Encryption

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

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

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

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

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

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

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

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

$\mathsf{mIND}_b(\Pi, A, k, \ell)$:
For $i = 1, \ldots, \ell$: $(\text{PK}_i, \text{SK}_i) \leftarrow \mathsf{Gen}(1^k)$

$(\langle m_0^1, \ldots, m_0^\ell \rangle, \langle m_1^1, \ldots, m_1^\ell \rangle, \text{STATE}_A) \leftarrow A_1(\langle \text{PK}_1, \ldots, \text{PK}_\ell \rangle)$
s.t. $|m_0^1| = |m_1^1| = \cdots = |m_0^\ell| = |m_1^\ell|$
For $i = 1, \ldots, \ell$*:* $y_i \leftarrow \mathsf{Enc}_{\text{PK}_i}(m_b^i)$
$D \leftarrow A_2(y_1, \ldots, y_\ell, \text{STATE}_A)$
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:*

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

2.2 Non-malleable Encryption

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

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

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* non-malleable under a 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:*

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

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

2.3 (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 [R90,L79], and thus from any semantically secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ using black-box access only to Gen.

3 Construction

Given an encryption scheme $E = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, 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 Figure 2, and described as follows.

Polynomial encoding. We identify $\{0,1\}^n$ with the field $\mathrm{GF}(2^n)$. To encode a message $m \in \{0,1\}^n$, we pick a random degree k polynomial p over $\mathrm{GF}(2^n)$ such that $p(0) = m$ and construct a $k \times 10k$ matrix such that the i'th column of the matrix comprise entirely of the value $s_i = p(i)$ (where $0, 1, \ldots, 10k$ are the lexicographically first $10k+1$ elements in $\mathrm{GF}(2^n)$ according to some canonical encoding). Note that $(s_1, \ldots, s_{10k})$ is both a $(k+1)$-out-of-$10k$ secret-sharing of m using Shamir's secret-sharing scheme and a codeword of the Reed-Solomon code $\mathcal{W}$, where

$$\mathcal{W} = \{\ (p(1), \ldots, p(10k)) \mid p \text{ is a degree } k \text{ polynomial }\}.$$

Note that $\mathcal{W}$ is a code over the alphabet $\{0,1\}^n$ with minimum relative distance 0.9, which means we may efficiently correct up to 0.45 fraction errors using the Berlekamp-Welch algorithm. [**tm:** add reference]

Encryption. The public key for Π comprises $20k^2$ public keys E indexed by a triplet $(i, j, b) \in [k] \times [10k] \times \{0,1\}$; there are two keys corresponding to each entry of a $k \times 10k$ matrix. To encrypt a message m, we (a) compute $(s_1, \ldots, s_{10_k})$ as in the above-mentioned polynomial encoding, (b) generate $(\textsc{sksig}, \textsc{vksig})$ for a one-time signature, (c) compute a $k \times 10k$ matrix $\boldsymbol{c} = (c_{i,j})$ of ciphertexts where $c_{i,j} = \mathsf{Enc}_{\mathrm{PK}^{v_i}_{i,j}}(s_j)$, and (d) sign $\boldsymbol{c}$ using $\textsc{sksig}$.

$$\begin{pmatrix} \mathsf{Enc}_{\mathrm{PK}^{v_1}_{1,1}}(s_1) & \mathsf{Enc}_{\mathrm{PK}^{v_1}_{1,2}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}^{v_1}_{1,10k}}(s_{10k}) \\ \mathsf{Enc}_{\mathrm{PK}^{v_2}_{2,1}}(s_1) & \mathsf{Enc}_{\mathrm{PK}^{v_2}_{2,2}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}^{v_2}_{2,10k}}(s_{10k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{Enc}_{\mathrm{PK}^{v_k}_{k,1}}(s_1) & \mathsf{Enc}_{\mathrm{PK}^{v_k}_{k,2}}(s_2) & \cdots & \mathsf{Enc}_{\mathrm{PK}^{v_k}_{k,10k}}(s_{10k}) \end{pmatrix}$$

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 k columns chosen during key generation.

Column Check (`column-check`): We check that each of the k columns in S comprises entirely of the same value.

CODEWORD CHECK (`codeword-check`): We find a codeword w that agrees with the first row of the matrix in at least $9k$ positions; the check fails if no such w exists. Then we check that the first row of the matrix agrees with w at the k positions indexed by S.

The codeword check ensures that with high probability, the first row of the matrix agrees with w in at least $10k - o(k)$ positions. We explain its significance after describing the alternative decryption algorithm in the analysis.

Decryption. To decrypt, we (a) verify the signature and run both consistency checks, and (b) if all three checks accept, decode the codeword w and output the result, otherwise output $\perp$. Note that to decrypt we only need the $20k$ secret keys corresponding to the first row of the matrix and $2k$ secret keys corresponding to each of the k columns in S.

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

4 Analysis

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

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

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

- NMDec^* and NMDec must agree on essentially all inputs, including possibly malformed ciphertexts;
- We can implement NMDec^* without having to know the secret keys corresponding to the public keys in Γ.

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

Here is a high-level (but extremely inaccurate) description of how NMDec^* works: Γ is the set of public keys corresponding to the first row of the $k \times 10k$ matrix. To

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

1. For $i \in [k], j \in [10k], b \in \{0,1\}$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\mathrm{PK}_{i,j}^b, \mathrm{SK}_{i,j}^b)$.
2. Pick a random subset $S \subset [10k]$ of size k.

Set $\mathrm{PK} = \Big\{(\mathrm{PK}_{i,j}^0, \mathrm{PK}_{i,j}^1) \mid i \in [k], j \in [10k]\Big\}$ and $\mathrm{SK} = \Big\{S, (\mathrm{SK}_{i,j}^0, \mathrm{SK}_{i,j}^1) \mid i \in [k], j \in [10k]\Big\}$.

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

1. Pick random $\alpha_1, \ldots, \alpha_k \in \mathrm{GF}(2^n)$ and set $s_j = p(j), j \in [10k]$ where $p(x) = m_0 + \alpha_1 x + \ldots + \alpha_k x^k$.
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 $c_{i,j} \leftarrow \mathsf{Enc}_{\mathrm{PK}_{i,j}^{v_i}}(s_j)$, for $i \in [k], j \in [10k]$.
4. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SKSIG}}(\boldsymbol{c})$ where $\boldsymbol{c} = (c_{i,j})$.

Output the tuple $[\boldsymbol{c}, \mathrm{VKSIG}, \sigma]$.

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

1. (`sig-check`) Verify the signature with $\mathsf{VerSig}_{\mathrm{VKSIG}}[\boldsymbol{c}, \sigma]$.
2. Let $\boldsymbol{c} = (c_{i,j})$ and $\mathrm{VKSIG} = (v_1, \ldots, v_k)$. Compute $s_j = \mathsf{Dec}_{\mathrm{SK}_{1,j}^{v_1}}(c_{1,j})$, $j = 1, \ldots, 10k$ and the codeword $w = (w_1, \ldots, w_{10k}) \in \mathcal{W}$ that agrees with $(s_1, \ldots, s_{10k})$ in at least $9k$ positions. If no such codeword exists, output $\perp$.
3. (`column-check`) For all $j \in S$, check that $\mathsf{Dec}_{\mathrm{SK}_{1,j}^{v_1}}(c_{1,j}) = \mathsf{Dec}_{\mathrm{SK}_{2,j}^{v_2}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\mathrm{SK}_{k,j}^{v_k}}(c_{k,j})$.
4. (`codeword-check`) For all $j \in S$, check that $s_j = w_j$.

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

Fig. 2. THE NON-MALLEABLE ENCRYPTION SCHEME Π

implement NMDec^*, we will decrypt the i'th row of the matrix of ciphertexts, for some $i > 1$, which the column check (if successful) guarantees to agree with the first row in most positions; error correction takes care of the tiny fraction of disagreements.

4.1 Alternative Decryption Algorithm NMDec^*

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

COLUMN CHECK (`column-check`*): This is exactly as before, we check that the each of the k columns in S comprises entirely of the same value.

CODEWORD CHECK (`codeword-check`*): Let i be the smallest value such that $v_i \neq v_i^*$ (which exists because VKSIG $\neq$ VKSIG*). We find a codeword w that agrees with the i'th row of the matrix in at least $8k$ positions (note agreement threshold is smaller than before); the check fails if so such w exists. Then we check that the first row of the matrix agrees with w at the k positions indexed by S.

To decrypt, run the modified signature and consistency checks, and if all three checks accept, decode the codeword w and output the result, otherwise output $\perp$. To implement the modified consistency checks and decryption algorithm, we only need the $10k$ secret keys indexed by $\overline{\text{VKSIG}^*}$ for each row of the matrix, and as before, the $2k$ secret keys corresponding to each of the k columns in S.

Remark on the Codeword Check. At first, the codeword check may seem superfluous. Suppose we omit the codeword check, and as before, define w to be a codeword that agrees with the first row in $9k$ positions and with the i'th row in $8k$ positions in the respective decryption algorithms; the gap is necessary to take into account inconsistencies not detected by the column check. Now, consider a malformed ciphertext ψ for Π where in the underlying matrix of plaintexts, each row is the same corrupted codeword that agrees with a valid codeword in exactly $8.5k$ positions. Without the codeword checks, ψ will be an invalid ciphertext according to NMDec and a valid ciphertext according to NMDec^* and can be used to distinguish the intermediate hybrid distributions in the analysis; with the codeword checks, ψ is an invalid ciphertext according to both. It is also easy to construct a problematic malformed ciphertext for the case where both agreement thresholds are set to the same value (say $9k$).

4.2 A Promise Problem

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

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

To quantify and establish these guarantees, we consider the following promise problem (Π_Y, Π_N) that again refers to the underlying matrix of plaintexts. An instance is a matrix of k by $10k$ values in $\{0,1\}^n \cup \perp$.

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

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

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

- If $\boldsymbol{M} \in \Pi_N$, then it fails the column/codeword checks in both decryption algorithms with high probability, in which case both decryption algorithms output $\perp$. Specifically, if there are two rows that are 0.1-far, then column check rejects $\boldsymbol{M}$ with probability $1 - 0.9^k$. On the other hand, if the first row is 0.1-far from every codeword, then the codeword check in NMDec rejects $\boldsymbol{M}$ with probability 1 and that in NMDec^* rejects $\boldsymbol{M}$ with probability at least $1 - 0.9^k$; that is, with probability $1 - 0.9^k$, both codeword checks in NMDec and NMDec^* rejects $\boldsymbol{M}$.
- If $\boldsymbol{M} \notin \Pi_N$, then both decryption algorithms always output the same answer for all choices of the set S, provided there is no forgery. Fix $\boldsymbol{M} \notin \Pi_N$ and a set S. The first row is 0.9-close to codeword $w \in \mathcal{W}$ and we know in addition that every other row is 0.9-close to the first row and thus 0.8-close to w. Therefore, we will recover the same codeword w and message m whether we decode the first row within distance 0.1, or any other row within distance 0.2. This means that the codeword checks in both decryption algorithms compare the first row with the same codeword w. As such, both decryption algorithms output $\perp$ with exactly the same probability, and whenever they do not output $\perp$, they output the same message m.

4.3 Proof of Main Theorem

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

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

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

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

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

1. (`sig-check*`) Verify the signature with $\mathsf{VerSig}_{\text{VKSIG}}[\boldsymbol{c}, \sigma]$. Output $\perp$ if the signature fails to verify or if $\text{VKSIG} = \text{VKSIG}^*$.
2. Let $\boldsymbol{c} = (c_{i,j})$ and $\text{VKSIG} = (v_1, \ldots, v_k)$. Let i be the smallest value such that $v_i \neq v_i^*$. Compute $s_j = \mathsf{Dec}_{\text{SK}_{i,j}^{v_i}}(c_{i,j})$, $j = 1, \ldots, 10k$ and $w = (w_1, \ldots, w_{10k}) \in \mathcal{W}$ that agrees with $(s_1, \ldots, s_{10k})$ in at least $8k$ positions. If no such codeword exists, output $\perp$.
3. (`column-check*`) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}_{1,j}^{v_1}}(c_{1,j}) = \mathsf{Dec}_{\text{SK}_{2,j}^{v_2}}(c_{2,j}) = \cdots = \mathsf{Dec}_{\text{SK}_{k,j}^{v_k}}(c_{k,j})$.
4. (`codeword-check*`) For all $j \in S$, check that $\mathsf{Dec}_{\text{SK}_{1,j}^{v_1}}(c_{1,j}) = w_j$.

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

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

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

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

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

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

$$\left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\} \equiv \left\{\mathsf{mIND}_b(E, B, k, 9k^2)\right\}$$

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

- (pre-processing) Pick a random subset $S = \{u_1, \ldots, u_j\}$ of $[10k]$ and run $\mathsf{GenSig}(1^k)$ to generate $(\text{SKSIG}^*, \text{VKSIG}^*)$ and set $(v_1^*, \ldots, v_k^*) = \text{VKSIG}^*$. Let ϕ be a bijection identifying $\{(i,j) \mid i \in [k], j \in [10k] \setminus S\}$ with $[9k^2]$.
- (key generation) B receives $\langle \text{PK}_1, \ldots, \text{PK}_{9k^2} \rangle$ from the outside and simulates NMGen as follows: for all $i \in [k], j \in [10k], \beta \in \{0,1\}$,

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

– (message selection) Let (m_0, m_1) be the pair of messages A_1 returns. B then chooses k random values $(\gamma_{u_1}, \ldots, \gamma_{u_k}) \in \{0,1\}^n$ and computes two degree k polynomials p_0, p_1 where p_β interpolates the $k{+}1$ points $(0, m_\beta), (u_1, \gamma_{u_1}), \ldots, (u_k, \gamma_{u_k})$ for $\beta \in \{0,1\}$. B sets $m_\beta^{\phi(i,j)} = p_\beta(j)$, for $i \in [k], j \in [10k] \setminus S$ and forwards $(\langle m_0^1, \ldots, m_0^{9k^2} \rangle, \langle m_1^1, \ldots, m_1^{9k^2} \rangle)$ to the outside.

– (ciphertext generation) B receives $\langle y_1, \ldots, y_{9k^2} \rangle$ from the outside (according to the distribution $\mathsf{Enc}_{\mathrm{PK}_1}(m_b^1), \ldots, \mathsf{Enc}_{\mathrm{PK}_{9k^2}}(m_b^{9k^2})$) and generates a ciphertext $[\boldsymbol{c}, \mathrm{VKSIG}^*, \sigma]$ as follows:

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

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

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

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

$$\Big\{\mathsf{NME}_b(\Pi, A, k, p(k))\Big\} \stackrel{c}{\approx} \Big\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\Big\}$$
$$\stackrel{s}{\approx} \Big\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\Big\} \equiv \Big\{\mathsf{mIND}_b(E, B, k, 9k^2)\Big\}$$

By Prop 1, $\mathsf{mIND}_0(E, B, k, 9k^2) \stackrel{c}{\approx} \mathsf{mIND}_1(E, B, k, 9k^2)$, which concludes the proof of Theorem 1.

5 Achieving Bounded-CCA2 Non-Malleability

We sketch how our scheme may be modified to achieve non-malleability under a bounded-CCA2 attack. Here, we allow the adversary to query Dec at most q times in the non-malleability experiment (but it must not query Dec on y). The modification is the straight-forward analogue of the [CHH+07] modification of the [PSV06] scheme: we increase the number of columns in the matrix from $10k$ to $80(k+q)$, and the degree of the polynomial p and the size of S from k to $8(k+q)$, and propagate the changes accordingly. The analysis is basically as before, except for the following claim (where $\mathsf{NME\text{-}q\text{-}CCA}_b^{(1)}, \mathsf{NME\text{-}q\text{-}CCA}_b^{(2)}$ are the respective analogues of $\mathsf{NME}_b^{(1)}, \mathsf{NME}_b^{(1)}$):

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

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

Proof (sketch). 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. However, we cannot immediately deduce that the output of A_2 are completely determined and identical in both experiments, since they depend on the adaptively chosen queries to NMDec, and the answers depend on S. Instead, we will consider all 2^q possible computation paths of A which are determined based on the q query/answer pairs from NMDec. For each query, we consider the underlying matrix of plaintexts $\boldsymbol{M}$:

– If $\boldsymbol{M} \in \Pi_N$, then we assume NMDec returns $\perp$.
– If $\boldsymbol{M} \notin \Pi_N$, then we consider two branches depending on the two possible outcomes of the consistency checks.

We claim that with probability $1 - 2^q \cdot p(k) \cdot 0.9^{8(k+q)} > 1 - \mathrm{neg}(k)$ over the choice of S, the decryptions of $(\psi_1, \ldots, \psi_{p(k)})$ agree in both experiments in all 2^q computation paths. $\square$

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

Acknowledgments. This work was initiated while the third and fourth authors were visiting IPAM. We would like to thank Vinod Vaikuntanathan for sharing his insights on non-malleability over the last two summers.

References

[BFM88] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC*, pages 103–112, 1988.
[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC*, pages 1–10, 1988.
[BHSV98] M. Bellare, S. Halevi, A. Sahai, and S. P. Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In *CRYPTO*, pages 283–298, 1998.
[CHH+07] R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan. Bounded CCA2-secure encryption. In *ASIACRYPT*, 2007.
[CHK04] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.
[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.
[CS04] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2004.
[DDN00] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[ES02] E. Elkind and 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/.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMM07] Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *TCC*, pages 434–455, 2007.

[GMR01] Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS*, pages 126–135, 2001.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[H08] I. Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In *these proceedings*, 2008.

[IKLP06] Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank. Black-box constructions for secure computation. In *STOC*, pages 99–108, 2006.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, 1989.

[L79] L. Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.

[L06] Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. Cryptology*, 19(3):359–377, 2006.

[LP07] Y. Lindell and B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *EUROCRYPT*, pages 52–78, 2007.

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

[PSV06] R. Pass, A. Shelat, and V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In *CRYPTO*, pages 271–289, 2006.

[PW07] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. Cryptology ePrint Archive, Report 2007/279, 2007. http://eprint.iacr.org/.

[R90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[RS91] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

[RTV04] O. Reingold, L. Trevisan, and S. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, pages 1–20, 2004.

[S99] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.