Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-081 December 14, 2006

Bounded CCA2-Secure Non-Malleable Encryption

Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Bounded CCA2-Secure Non-Malleable Encryption

Rafael Pass
Cornell University

abhi shelat
IBM Zurich

Vinod Vaikuntanathan
MIT CSAIL

Abstract

Under an adaptive chosen ciphertext attack (CCA2), the security of an encryption scheme must hold against adversaries that have access to a decryption oracle. We consider a weakening of CCA2 security, wherein security need only hold against adversaries making an *a-priori bounded* number of queries to the decryption oracle. Concerning this notion, which we call *bounded-CCA2* security, we show the following two results.

† Bounded-CCA2 secure *non-malleable* encryption schemes exist if and only if semantically-secure (IND-CPA-secure) encryption schemes exist. (As far as we know, bounded-CCA2 non-malleability is the strongest notion of security known to be satisfiable assuming only the existence of semantically-secure encryption schemes.)

‡ In contrast to CCA2 security, bounded-CCA2 security alone *does not* imply non-malleability. In particular, if there exists an encryption scheme that is bounded-CCA2 secure, then there exists another encryption scheme which remains bounded-CCA2 secure, but *is malleable under a simple chosen-plaintext attack*.

Keywords: Public-key Encryption, Non-Malleability, Chosen Ciphertext Security.

1 Introduction

Historically, encryption has been a mechanism helps achieve *privacy* of data. This goal of privacy is captured in the notion of semantic security [GM84] which, roughly stated, says that "whatever an adversary can learn after seeing the ciphertext, it could have learnt without seeing the ciphertext".

Non-malleability, as defined by Dolev, Dwork and Naor [DDN00], is a stronger notion of security for encryption schemes. In addition to the normal "privacy" guarantee, non-malleability guarantees that it is infeasible for an adversary to *modify* a vector of ciphertexts $\alpha_1, \ldots, \alpha_n$ into other ciphertexts of messages which are related to the decryption of $\alpha_1, \ldots, \alpha_n$. It has been widely acknowledged that this stronger notion of security is critical for many practical applications. Recently, the same authors have showed that for the weaker class of chosen-plain text attacks, any encryption scheme that is semantically secure (against chosen plain-text attacks) can be transformed into one that is non-malleable (against chosen-plaintext attacks), without relying on any additional assumptions [PSV06a].

Stronger Types of Attacks Under the traditional type of chosen-plaintext attack (CPA) on an encryption scheme, the adversary is required to act on its own without any additional help [GM84]. Naor and Yung [NY90], and Rackoff and Simon [RS93], considered the security of encryption schemes under stronger types of attacks. In the strongest of these, called *adaptive chosen cipher-text* attacks (CCA2), security is required to hold with respect to adversaries that have access to an decryption oracle. Interestingly, it has been showed that under CCA2 attack, the otherwise weaker notion of semantic security in fact implies also non-malleability [DDN00].

Nevertheless, constructions of CCA2-secure encryption schemes are rare [DDN00, CS98] and it is open whether any semantically secure encryption scheme can be transformed into one that is also CCA2 secure, without making additional complexity theoretic assumptions.

Our results In this paper we introduce a weakening of the notion of a CCA2 attack which we call bounded-CCA2 attack. In such an attack, the adversary is restricted to making an *a-priori bounded* number of queries to the decryption oracle. Thus, we may discuss the notion of an m-bounded CCA2 non-malleable encryption scheme as one that is "non-malleable" with respect to an adversary making at most m decryption queries.

With this terminology, our main result shows that bounded-CCA2 *non-malleable* encryption schemes exist if and only if CPA-secure encryption schemes exist. As far as we know, the notion of bounded-CCA2 non-malleability is the strongest notion of security for encryption schemes known to be satisfiable under only the assumption of CPA-secure encryption schemes.

Theorem 1 (Informal) *Suppose there exists a IND-CPA-secure public-key encryption scheme. Then, for any polynomial m, there exists an m-bounded CCA2 non-malleable encryption scheme.*

We mention that the encryption scheme constructed depends on the parameter m, and in fact the length of both the the public-key and the ciphertexts grows linearly with m.

Let us point out that one may also consider the notion of m-bounded IND-CCA2 semantically-secure encryption (without the extra non-malleability requirement). As mentioned above, in the case of full CCA2-attacks, semantical security has been shown to imply non-malleability. In the case of bounded-CCA2 security, however, we show that this equivalence does not hold. More dramatically, we show that bounded-CCA2 security for any fixed m does not even imply non-malleability under the simple chosen plaintext attack.

Theorem 2 (Informal) *Assume the existence of a IND-CPA-secure public-key encryption scheme. Then, for every m, there exists an encryption scheme that is m-bounded IND-CCA2-secure, but is not non-malleable (even under CPA attacks).*

This separation of notions highlights the importance of directly proving non-malleable of our scheme (which complicates the analysis). While we find the idea of setting an upper-limit on the number of decryption queries a quite reasonable relaxation of a standard CCA2 attack, we view non-malleability as a principal desiderata for the security of an encryption scheme.

Remark on Non Black-box techniques. We mention that our construction makes a non black-box use of the underlying semantically-secure encryption scheme. In particular, we use a proof that several ciphertexts are encryptions of the same message, and this may require analyzing the encryption circuit to form a theorem statement. A very recent result by Gertner, Malkin, and Myers [GMM07] shows the impossibility of black-box constructions of (fully) CCA2-secure encryption schemes from semantically secure encryption schemes, where the *decryption* algorithm of the CCA2-secure scheme does not make use of the *encryption* algorithm of the semantically secure scheme. It seems that their proof extends to rule out the same type of black-box constructions of *non-malleable* encryption schemes even under chosen plaintext attacks (and thus also under bounded-CCA2 attacks). In this sense, non-blackbox techniques may be *necessary* for our results.

On the other hand, even though our construction uses ZK proofs and thus costly general $\mathcal{NP}$ reductions, for many encryption schemes, these proofs can be substituted with much more efficient proofs (based on, say, Σ protocols) for the type of theorems we need to prove in our construction. In this sense, the non-blackbox property may not have significant overhead in practical situations.

In contrast to the above, m-bounded IND-CCA secure encryption (without the non-malleability requirement) seems possible using black-box techniques [CHK, HI]. As we show in Theorem 2, however, bounded CCA-secure non-malleable encryption is a strictly stronger requirement than bounded IND-CCA-secure encryption.

2 Definitions

Preliminaries. If A is a probabilistic polynomial time (p.p.t) algorithm that runs on input x, $A(x)$ denotes the random variable corresponding to the output of A on input x and uniformly random coins. Sometimes, we want to make the randomness used by A explicit, in which case, we let $A(x;r)$ denote the output of A on input x and random coins r. We denote computational indistinguishability [GM84] of ensembles A and B by $A \stackrel{c}{\approx} B$.

Oracles Unless otherwise noted, all of our definitions make use of the following oracle convention. In the case of a CPA attack, the oracles O_1, O_2 return the empty string on all queries. In a CCA1 attack, the oracle $O_1(\text{PK})$ returns decryptions of ciphertexts under the public key PK (which is implicit by context). Finally, in a CCA2 attack, both oracles return decryptions with the exception that $O_2(\text{PK}, y)$ returns $\perp$ when queried on a particular ciphertext y.

When we refer to a specific type of attack scenario, we will add the suffix CPA,CCA1, or CCA2 to the name of the definition, e.g., IND-CCA2. Otherwise, we omit this suffix in order to simplify the notation.

Computational Indistinguishability For the reader's convenience, we briefly summarize the notion of computational indistinguishability.

Definition 1 (Computational Indistinguishability) *Two ensembles $\{X_w\}_{w\in I}$ and $\{Y_w\}_{w\in I}$ with identical index set I are said to be* computationally indistinguishable *if for every polynomial-size circuit family $\{D_k\}_{k\in\mathbb{N}}$, every sufficiently large k, and every $w \in I \cap \{0,1\}^k$, we have that*

$$|\Pr[D_k(X_w) = 1] - \Pr[D_k(Y_w) = 1]| < \mu(k).$$

We denote such sets $\{X_w\}_{w\in I} \stackrel{c}{\approx} \{Y_w\}_{w\in I}$.

Encryption Scheme. Here we review the syntactic functionality of an encryption scheme. Note that we demand perfect correctness from an encryption scheme. This requirement can be imposed without loss of generality, since any encryption scheme can be converted into one that has perfect correctness [DNR04].

Definition 2 (Encryption Scheme) *A triple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an encryption scheme, if* Gen *and* Enc *are p.p.t. algorithms and* Dec *is a deterministic polynomial-time algorithm,*

1. Gen *on input* 1^k *produces a tuple* $(\mathrm{PK}, \mathrm{SK})$*, where* PK, SK *are the public and private keys,*
2. $\mathsf{Enc} : \mathrm{PK} \times \{0,1\}^* \to \{0,1\}^*$ *runs on input a public key* PK *and a message* $m \in \{0,1\}^*$ *and produces a ciphertext* c*,*
3. $\mathsf{Dec} : \mathrm{SK} \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ *runs on input* (SK, c) *and produces either a message* $m \in \{0,1\}^*$ *or a special symbol* $\perp$*,*

and the algorithms satisfy the perfect correctness property:

(Perfect Correctness) *There exists a polynomial* $p(k)$ *and a negligible function* $\mu(k)$ *such that for* every message m*, and* every random tape r_e*,*

$$\Pr[r_g \xleftarrow{R} \{0,1\}^{p(k)};\ (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k; r_g);\ \mathsf{Dec}_{\mathrm{SK}}(\mathsf{Enc}_{\mathrm{PK}}(m; r_e)) \neq m] \leq \mu(k).$$

2.1 Semantically-Secure (IND-CPA-Secure) Encryption

Definition 3 (IND-security) *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme and let the random variable* $\mathsf{IND}\text{-}_b(\Pi, A, k, \ell)$ *where* $b \in \{0,1\}$*,* $A = (A_1, A_2)$ *and* $k, \ell \in \mathbb{N}$ *denote the result of the following probabilistic experiment:*

$\mathsf{IND}_b(\Pi, A, R, k)$
  $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k)$
  $(m_0, m_1, z) \leftarrow A_1^{O_1}(\mathrm{PK})$
  $y \leftarrow \mathsf{Enc}(\mathrm{PK}, m_b)$
  $x \leftarrow A_2^{O_2}(y, z)$
  Output x

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is indistinguishable under a chosen-plaintext attack if* $\forall$ *p.p.t. algorithms* A *the following two ensembles are computationally indistinguishable:*

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

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

If A makes at most m queries to O_1 and O_2 together, then $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be m-**bounded** IND-CCA2-secure. Further, if $m = 0$, then the encryption scheme is said to be IND-CPA-secure.

2.2 Definition of Non-Malleable Encryption

The following definition of non-malleability was introduced in [PSV06a]. There it is shown that the definition composes, both in terms of the number of messages received by the adversary, and in terms of the number of keys under which the messages are encrypted.

Definition 4 (NME-security [BS99, PSV06a]) *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* $b \in \{0,1\}$, $A = (A_1, A_2)$ *and* $k, \ell \in \mathbb{N}$ *denote the result of the following probabilistic experiment:*

$\mathsf{NME}_b(\Pi, A, k, \ell)$:

$(\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(1^k)$

$(m_0, m_1, \text{STATE}_A) \leftarrow A_1^{O_1}(\text{PK})$ *s.t.* $|m_0| = |m_1|$

$y \leftarrow \mathsf{Enc}_{\text{PK}}(m_b)$

$(c_1, \ldots, c_\ell) \leftarrow A_2^{O_2}(y, \text{STATE}_A)$

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

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* NME*-secure if* $\forall$ *p.p.t. algorithms* $A = (A_1, A_2)$ *and for any polynomial* $p(k)$*, the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{NME}_0(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \overset{c}{\approx} \left\{\mathsf{NME}_1(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \tag{1}$$

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

If A makes at most m queries to O_1 and O_2 together, then $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be m**-bounded** NME-CCA2-secure. Further, if $m = 0$, then the encryption scheme is said to be NME-CPA-secure.

3 Strong Designated Verifier NIZK

Pass, shelat, and Vaikuntanathan [PSV06a] used designated verifier NIZK proofs to construct an NM-CPA-secure encryption scheme from an IND-CPA-secure one. We define a stronger notion of soundness for designated verifier NIZK proofs and show that the construction of [PSV06a] indeed satisfies this notion of soundness. We will later use this in the construction of our bounded-NM-CCA2-secure encryption scheme.

3.1 Defining Strong Designated Verifier NIZK Proof Systems

In the designated verifier model, a non-interactive proof system has an associated polynomial-time sampleable distribution $\mathcal{D}$ over binary strings of the form (PP, SP). During a setup phase, a trusted party samples from $\mathcal{D}$, publishes PP and privately hands the Verifier SP. The Prover and Verifier then use their respective values during the proof phase.

Definition 5 (Strong Designated Verifier NIZK Proof System) *A triple of algorithms,* $(\mathcal{D}, P, V)$*, is called a* designated verifier non-interactive zero-knowledge proof system *for an* $\mathcal{NP}$*-language* L *with witness relation* R_L*, if the algorithms* $\mathcal{D}$ *and* P *are probabilistic polynomial-time, the algorithm* V *is deterministic polynomial-time and there exists a negligible function* μ *such that the following three conditions hold:*

- COMPLETENESS: *For every* $(x, w) \in R_L$

$$\Pr\left[\ (\text{PP}, \text{SP}) \leftarrow \mathcal{D}(1^{|x|});\ \pi \leftarrow P(\text{PP}, x, w)\ :\ V(\text{PP}, \text{SP}, x, \pi) = 1\ \right] \geq 1 - \mu(|x|)$$

- m-BOUNDED STRONG SOUNDNESS: *For every oracle Turing machine B that has access to the verifier oracle $V(\text{SP}, \text{PP}, \cdot, \cdot)$, and makes at most m oracle queries to V,*

$$\Pr\left[(\text{PP}, \text{SP}) \leftarrow \mathcal{D}(1^{|x|})(', \pi') \leftarrow B^{V(\text{SP},\text{PP},\cdot,\cdot)}(\text{PP}) \ : \ \begin{array}{l} x' \notin L \text{ and} \\ V(\text{PP}, \text{SP}, x', \pi') = 1 \end{array} \right] \leq \mu(|x|)$$

- STRONG ADAPTIVE ZERO-KNOWLEDGE: *For every p.p.t. theorem chooser A, there exists a p.p.t. simulator $S = (S_1, S_2)$ such that the outputs of the following experiments are indistinguishable.*

$\text{EXPTZK}_A(k)$	$\text{EXPTZK}_A^S(k)$
$(\text{PP}, \text{SP}) \leftarrow \mathcal{D}(1^k)$	$(\text{PP}', \text{SP}', \text{STATE}) \leftarrow S_1(1^k)$
$(x, w, \text{STATE}_A) \leftarrow A(\text{PP}, \text{SP})$	$(x, w, \text{STATE}_A) \leftarrow A(\text{PP}', \text{SP}')$
$\pi \leftarrow P(\text{PP}, x, w)$	$\pi' \leftarrow S_2(\text{PP}', \text{SP}', x, \text{STATE})$
If $(x, w) \notin R_L$, *output* $\perp$	*If* $(x, w) \notin R_L$, *output* $\perp$
Else output $(\text{PP}, \text{SP}, x, \pi, \text{STATE}_A)$	*Else output* $(\text{PP}', \text{SP}', x, \pi', \text{STATE}_A)$

Some technical remarks are in order. First of all, the difference between the adaptive zero-knowledge definition here and the one in [PSV06a] is that, we give the theorem chooser SP, in addition to PP. The definition of [PSV06a] only gave PP to the theorem chooser. Despite this strengthening, we will show that the designated verifier proof system of [PSV06a] meets the stronger definition as given here. Secondly, the soundness condition is required to hold for unbounded prover algorithms B, the only restriction on B being that it can access the verifier oracle an a-priori bounded number of times. Finally, the Verifier V is a deterministic machine. This extra restriction is only used to simplify the exposition of our constructions.

3.2 The Construction

The construction is the same one presented in [PSV06a], which we briefly review for completeness. Our only complexity assumption is the existence of a semantically-secure encryption scheme. We note that Camenisch and Damgård use a similar idea in [CD00] to construct an interactive verifiable encryption scheme. The roots of this idea begin to appear much earlier in [KMO89].

Theorem 3 *Assume there exists a semantically secure encryption scheme. Then, for every polynomial $m(|x|)$, there exists a strong designated verifier NIZK proof system with $m(|x|)$-bounded soundness for any language $L \in \mathcal{NP}$.*

Proof: The NIZK protocol is in Figure 1. The completeness property follows from the completeness of the 3-round Σ protocol. The adaptive zero-knowledge property we need is stronger than the one in [PSV06a] as noted above. Nevertheless, the protocol also achieves this stronger notion of zero-knowledge. A proof is given in Appendix A.

[PSV06a] show the 0-bounded strong soundness of this protocol. More precisely,

Proposition 4 ([PSV06a]) *$(\mathcal{D}, P, V)$ is 0-bounded sound. That is, for any cheating prover B^*,*

$$\Pr\left[(\text{PP}, \text{SP}) \leftarrow \mathcal{D}(1^{|x|})(', \pi') \leftarrow B^*(\text{PP}) \ : \ \begin{array}{l} x' \notin L \text{ and} \\ V(\text{PP}, \text{SP}, x', \pi') = 1 \end{array} \right] \leq 2^{-k}$$

We will use this to show that the same protocol satisfies $m(|x|)$-bounded soundness.

Let $k \stackrel{\text{def}}{=} m(|x|) + |x|$.

Sampling Algorithm $\mathcal{D}(1^k)$**.** For $i = 1, \ldots, k$ and $b = 0, 1$, run $\mathsf{Gen}(1^k)$ $2k$ times with independent random coins, to get k key-pairs $(\text{PK}_i^b, \text{SK}_i^b)$. For $i = 1, \ldots, k$, flip coin $f_i \stackrel{R}{\leftarrow} \{0, 1\}$. Generate the receiver message σ for a two-round commitment scheme.

Let $\text{PP}_{dv} \stackrel{\text{def}}{=} [(\text{PK}_i^0, \text{PK}_i^1, \sigma)]_{i=1}^k$ and $\text{SP}_{dv} \stackrel{\text{def}}{=} [f_i, \text{SK}_i^{f_i}]_{i=1}^k$. Output $(\text{PP}_{dv}, \text{SP}_{dv})$.

Prover $P(\text{PP}_{dv}, x, w)$**.** For $i = 0, \ldots, k$, generate triples as follows:

$$
\begin{aligned}
&(a_i, s_i) \leftarrow P_1(x, w) \\
&c_{b,i} \leftarrow P_2(s, b) \text{ for both } b = 0, 1 \\
&\alpha_{b,i} \leftarrow \mathsf{Enc}_{\text{PK}_{b,i}}(c_{b,i}) \text{ for } b = 0, 1.
\end{aligned}
$$

and output $\pi \stackrel{\text{def}}{=} [(a_i, \alpha_{0,i}, \alpha_{1,i})]_{i=1}^k$.

Verifier $V(\text{PP}_{dv}, \text{SP}_{dv}, x, \pi)$**.** Parse π into k triples of the form $(a_i, \alpha_{0,i}, \alpha_{1,i})$. For $i = 1, \ldots, k$, compute $m_i \stackrel{\text{def}}{=} \mathsf{Dec}_{\text{SK}_i^{f_i}}(\alpha_{f_i,i})$ and run the verifier $V_2(a_i, f_i, m_i)$. If all k proofs are accepted, output ACCEPT, else output REJECT.

Figure 1: DESIGNATED VERIFIER NIZK PROTOCOL

Proposition 5 $(\mathcal{D}, P, V)$ *satisfies* $m(|x|)$*-bounded strong soundness.*

Proof: Suppose there is a cheating prover B that asks the verifier oracle $m(|x|)$ queries and breaks the soundness with probability $2^{-|x|}$.

We will use B to construct an algorithm B^* that breaks the normal soundness of the protocol with probability more than 2^{-k}, which is a contradiction to the [PSV06a] theorem. B^* works as follows: (1) B^* answers B's queries to the verifier oracle by flipping a random bit and returning it, and (2) When B outputs a pair (x, π) at the end, B^* outputs (x, π) too.

The probability that B^* gives the correct answers to B's queries is exactly $2^{-m(|x|)}$. In other words, with this probability, B^* simulates the verifier oracle perfectly. Thus, $\Pr[B^* \text{ succeeds}] \geq \Pr[B \text{ succeeds} \wedge B^* \text{ simulates the verifier oracle perfectly}] > 2^{-|x|}2^{-m(|x|)} = 2^{-k}$

Thus, B^* breaks the ordinary soundness of the proof system with probability 2^{-k}, which is in contradiction to the result of [PSV06a] mentioned above in Proposition 4. □ □

4 Constructing Bounded-NM-CCA2-Secure Encryption Scheme

In this section, we construct an encryption scheme that is m-bounded NM-CCA2-secure, starting from any semantically secure (IND-CPA-secure) encryption scheme. The construction is *the same* as the DDN construction [DDN00] and the construction of Pass, Shelat and Vaikuntanathan [PSV06a], except that the NIZK proof used is a designated-verifier NIZK proof with m-bounded strong soundness. By the results from the previous section (the construction of m-bounded designated verifier proof systems from semantically secure encryption schemes), our construction only relies on the assumption of the existence of a semantically secure encryption scheme.

Theorem 6 (Main Theorem) *Assume there is an encryption scheme that is* IND-CPA*-secure. Then, for every polynomial m, there exists an encryption scheme that is m-bounded* NM-CCA2*-secure.*

Our proof closely follows the proof of [PSV06a]. We highlight the crucial differences between their proof and ours in the appropriate places.

Proof:(of Theorem 6) Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be any semantically secure encryption scheme. Let $(\mathsf{Gen}_{sig}, \mathsf{Sign}, \mathsf{Ver})$ be any existentially unforgeable *strong* one-time signature scheme.[1] Without loss of generality, assume that Gen_{sig} produces verification keys of length k.[2] Define the $\mathcal{NP}$-language L as follows:

$$\big[(c_1,\ldots,c_k),(p_1,\ldots,p_k)\big] \in L \text{ if and only if}$$
$$\exists\big[m,(r_1,\ldots,r_n)\big] \text{ such that } c_i = \mathsf{Enc}_{p_i}(m;r_i) \text{ for } i=1,\ldots,n.$$

In words, the language L contains pairs consisting of a k-tuple of ciphertexts and a k-tuple of public keys such that the ciphertexts are encryptions of the *same message* m under the k public keys.

Let $(\mathcal{D}, P, V)$ be an m-bounded designated verifier NIZK proof system for L. We show that the encryption scheme $\Pi = (\mathsf{NMGen}, \mathsf{NMEnc}, \mathsf{NMDec})$ defined in Figure 4 is an m-bounded NM-CCA2-secure encryption scheme. The proof has two parts.

Just as in [DDN00] and [PSV06a], we define an encryption scheme $E' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ in which one simply encrypts a message k times with k independently chosen public keys, and we show that E' is a semantically secure encryption scheme under the assumption that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is one. This is identical to [PSV06a] and is stated in Lemma 7.

Then in Lemma 8, we show that Π is an m-bounded NM-CCA2-secure encryption scheme if E' is a semantically secure encryption scheme. The proof is concluded by noting that both m-bounded designated verifier NIZK proofs and strong one-time signatures can be constructed given any semantically secure encryption scheme (The former is true by virtue of Theorem 3. The latter follows by combining the observation that encryption implies one-way functions, Rompel's result showing that one-way functions imply universal one-way hash functions [Rom90], and the result that universal one-way hash functions imply strong one-time signature schemes [Gol04, Lam79]).□ □

The definition of the encryption scheme $E' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ below is exactly as in DDN, reproduced below for the sake of completeness.

- $\mathsf{Gen}'(1^k)$: For $i = 1,\ldots,k$, run $(\mathrm{PK}_i, \mathrm{SK}_i) \leftarrow \mathsf{Gen}(1^k)$ with independent random coins. Set $\mathrm{PK} \stackrel{\text{def}}{=} (\mathrm{PK}_1,\ldots,\mathrm{PK}_k)$ and $\mathrm{SK} \stackrel{\text{def}}{=} (\mathrm{SK}_1,\ldots,\mathrm{SK}_k)$.
- $\mathsf{Enc}'_{\mathrm{PK}}(m)$: Output $[\mathsf{Enc}_{\mathrm{PK}_1}(m;r_1),\ldots,\mathsf{Enc}_{\mathrm{PK}_k}(m;r_k)]$.
- $\mathsf{Dec}'_{\mathrm{SK}}([c_1,c_2,\ldots,c_k])$: Compute $m'_i = \mathsf{Dec}_{\mathrm{SK}_i}(c_i)$. If all the m'_i are not equal, output $\perp$, else output m'_1.

Lemma 7 *[DDN00, PSV06a] If $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is semantically secure, then $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is semantically secure.*

Lemma 8 *If $E' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is a semantically secure encryption scheme, then Π is an m-bounded* NM-CCA2*-secure encryption scheme.*

[1] A strong signature is one in which, given a signature σ of a message m, it is infeasible to produce a message m' and a valid signature σ' of m', such that $(\sigma, m) \neq (\sigma', m')$. i.e, it is infeasible also to produce a different signature for the *same message*.

[2] This is without loss of generality since we can set k to be an upperbound on the length of verification keys that Gen_{sig} produces.

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

1. For $i \in [1,k], b \in \{0,1\}$, run $\mathsf{Gen}(1^k)$ to generate key-pairs $(\mathrm{PK}_i^b, \mathrm{SK}_i^b)$.
2. Run $\mathcal{D}(1^k)$ to generate $(\mathrm{PP}, \mathrm{SP})$.

Set $\mathrm{PK} \stackrel{\text{def}}{=} \left\{(\langle \mathrm{PK}_i^0, \mathrm{PK}_i^1 \rangle)_{i=1}^k, \mathrm{PP}\right\}$ and $\mathrm{SK} \stackrel{\text{def}}{=} \left\{(\langle \mathrm{SK}_i^0, \mathrm{SK}_i^1 \rangle)_{i=1}^k, \mathrm{SP}\right\}$.

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

1. Run the signature algorithm $\mathsf{Gen}_{sig}(1^k)$ to generate $(\mathrm{SKSIG}, \mathrm{VKSIG})$. Let $(v_1, \ldots, v_k)$ be the binary representation of VKSIG.
2. Compute the ciphertexts $c_i \leftarrow \mathsf{Enc}_{\mathrm{PK}_i^{v_i}}(m)$. Let $\vec{c} \stackrel{\text{def}}{=} (c_1, c_2, \ldots, c_k)$.
3. Run the designated verifier NIZK Prover to generate a proof π that $[(c_1, \ldots, c_k), (\mathrm{PK}_1^{v_1}, \ldots, \mathrm{PK}_k^{v_k})] \in L$.
4. Compute the signature $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SKSIG}}(\langle \vec{c}, \pi \rangle)$.

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

$\mathsf{NMDec}_{\mathrm{SK}}(c)$:

1. Verify the signature with $\mathsf{Ver}_{\mathrm{VKSIG}}[\langle \vec{c}, \pi \rangle, \sigma]$; output $\perp$ upon failure.
2. Verify the proof with $V(\mathrm{PP}, \mathrm{SP}, (\vec{c}, \vec{\mathrm{PK}}), \pi)$; output $\perp$ upon failure.
3. Let $\mathrm{VKSIG} = (v_1, \ldots, v_k)$. Compute $m_1 = \mathsf{Dec}_{\mathrm{SK}_1^{v_i}}(c_1)$ and output the result.

Figure 2: THE NON-MALLEABLE ENCRYPTION SCHEME Π

Proof: To prove that Π is a non-malleable encryption scheme, we need to show that for any p.p.t. adversary A that queries the decryption oracle at most m times and for all polynomials $p(k)$,

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

We show this by a hybrid argument. The sequence of hybrid expts is the same as in [PSV06a] except that we need to handle the decryption queries of the adversary. This is done below in Step 3 in $\mathsf{NME}_b^{(1)}$ and in Step 2 in $\mathsf{NME}_b^{(2)}$. Consider the following experiments:

Experiment $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$ – Using a Simulated NIZK Proof: Proceeds exactly like NME_b except that the simulator for the designated verifier NIZK proof system is used to generate the public parameters and to compute the challenge ciphertext (as opposed to generating an honest proof by running the prover algorithm P). Let $S = (S_1, S_2)$ denote the simulator guaranteed by the adaptive zero-knowledge of $(\mathcal{D}, P, V)$. More formally, $\mathsf{NME}_b^{(1)}$ proceeds exactly like NME_b except for the following differences:

1. The encryption key $(\mathrm{PK}, \mathrm{SK})$ is generated by (1) honestly running the key-generation algorithm Gen to generate the $2k$ encryption keys $(\mathrm{PK}_i^b, \mathrm{SK}_i^b)$, *but* (2) running the simulator $S_1(1^k)$ to generate the key-pair $(\mathrm{PP}, \mathrm{SP})$ for the designated verifier NIZK (instead of running $\mathcal{D}(1^k)$ as in NMGen).

2. Generate k encryptions of m_b (just as in Steps 1 and 2 of NMEnc). *But*, instead of using the designated verifier prover, generate a "simulated proof" by running S_2. (Note that S_2 does not use the witness—namely, m_b and the randomness used for encryption—in order to generate the simulated proof).

3. **Answering the Decryption Queries:** Let the i^{th} decryption query be of the form $[\vec{c}, \pi, \text{VKSIG}, \sigma]$ (If the decryption query is not of this form, return $\perp$).

   If the signature σ is not valid under the verification key VKSIG, outputs $\perp$. Check the NIZK proof π in the ciphertext using the NIZK secret-parameter SP. If the proof is not accepting, return $\perp$. Otherwise, find a position ℓ such that $\text{VKSIG}_\ell \neq \text{VKSIG}^*_\ell$. Decrypt c_ℓ using the secret-key $\text{SK}_\ell^{\text{VKSIG}_\ell}$ and return the answer.

Experiment $\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))$ – Semantic Security of E': proceeds exactly like $\mathsf{NME}_b^{(1)}$ except for the following differences:

1. Run Gen' to get two sets of public keys $PK = \{\text{PK}_i\}_{i=1}^k$ and $PK' = \{\text{PK}'_i\}_{i=1}^k$, along with the corresponding secret-keys $SK = \{\text{SK}_i\}_{i=1}^k$ and $SK' = \{\text{SK}'_i\}_{i=1}^k$. Generate a verification key and signing key for the signature scheme $(\text{VKSIG}^*, \text{SKSIG}^*)$. Construct a public-key for Π as follows: Let v_i be the i^{th} bit of VKSIG^*. Set $\text{PK}_i^{v_i} = \text{PK}_i$, $\text{SK}_i^{v_i} = \perp$, $\text{PK}_i^{1-v_i} = \text{PK}'_i$ and $\text{SK}_i^{1-v_i} = \text{SK}'_i$. ($\mathsf{NME}_b^{(2)}$ will use the secret-keys corresponding to each PK'_i, but not PK_i, later in the experiment).

2. **Answering the Decryption Queries:** Exactly as in $\mathsf{NME}_b^{(1)}$.

3. After receiving the tuple $(\psi_1, \ldots, \psi_\ell)$ of ciphertexts from A_2, decrypt each $\psi_j = [\vec{c}_j, \pi_j, \text{VKSIG}_j, \sigma_j]$ as follows: If the signature σ_j in ψ_j does not verify, output $\perp$. If $\text{VKSIG}_j = \text{VKSIG}^*$, output $\perp$. If the NIZK proof π_j fails verification, output $\perp$. Else, decrypt one of the components of ψ_j, for which the secret-key is known (such a component is guaranteed to exist, since $\text{VKSIG}_j \neq \text{VKSIG}^*$) and output the result.

We now show that these experiments are indistinguishable. The following claim follows from the adaptive zero-knowledge property of the NIZK system. We here rely on the stronger variant of adaptive zero-knowledge (See Definition 5).

Claim 9 $\left\{\mathsf{NME}_b(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \overset{c}{\approx} \left\{\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}}$

Proof: Assume, for contradiction, that there exists a p.p.t. algorithm D which distinguishes $\mathsf{NME}_b(\Pi, A, k, p(k))$ from $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$. Then, we construct a theorem-chooser A_{zk} and a ZK distinguisher D_{zk} that violate the adaptive zero-knowledge of the proof system $(\mathcal{D}, P, V)$ for the language L. That is, D_{zk} distinguishes between the experiments $\mathsf{ZK}_{A_{\mathsf{zk}}}$ and $\mathsf{ZK}^S_{A_{\mathsf{zk}}}$, where S is the zero-knowledge simulator.

On input (PP, SP), the theorem-chooser A_{zk} works as follows:

1. Run $\mathsf{Gen}(1^k)$ $2k$ times, to generate $2k$ key-pairs $(\text{PK}_i^b, \text{SK}_i^b)_{i\in[k], b\in\{0,1\}}$. Run the adversary A_1 on input $\left[(\text{PK}_i^b)_{i\in[k],b\in\{0,1\}}, \text{PP}\right]$. A_1 returns a pair of plaintexts m_0 and m_1 and a string STATE.

2. Answer decryption queries exactly as in the experiment $\mathsf{NME}_b^{(1)}$.

3. Produce the challenge ciphertext $\vec{c}$ as follows:

   - Generate a key-pair $(\text{SKSIG}^*, \text{VKSIG}^*)$ for the signature scheme. Let $\text{VKSIG}^* = (v_1^*, v_2^*, \ldots, v_k^*)$.

- Pick a random $b \in \{0,1\}$, and for $1 \leq i \leq k$, let $c_i \leftarrow \mathsf{Enc}_{\mathrm{PK}_i^{v_i^*}}(m_b; r_i)$, where r_i is the randomness used for encryption.
  Let $\vec{c}$ denote $(c_1, c_2, \ldots, c_k)$ and $\vec{\mathrm{PK}}$ denote $(\mathrm{PK}_1^{v_1^*}, \ldots, \mathrm{PK}_k^{v_k^*})$, and $\vec{r}$ denote $(r_1, r_2, \ldots, r_k)$.

4. Let $x = (\vec{c}, \vec{\mathrm{PK}})$ and $w = (m_b, \vec{r})$. Output the theorem-witness pair (x, w). Also output the contents of the work-tape as STATE_A.

The ZK distinguisher D_{zk}, on input (PP, SP), the theorem $(\vec{c}, \vec{\mathrm{PK}})$, the proof π and the state STATE_A, does the following:

1. Run A_2 on input the ciphertext $[\vec{c}, \pi, \mathrm{VKSIG}, \mathsf{Sign}_{\mathrm{SKSIG}}(\langle \vec{c}, \pi \rangle)]$ to produce a sequence of ciphertexts $(\psi_1, \psi_2, \ldots, \psi_{p(k)})$. Run the decryption algorithm $\mathsf{Dec}_{\mathrm{SK}}(\psi_i)$ on each of these ciphertexts to get plaintexts $(\mu_1, \mu_2, \ldots, \mu_{p(k)})$.

2. Run distinguisher D on the sequence of plaintexts $(\mu_1, \mu_2, \ldots, \mu_{p(k)})$ and output whatever D outputs.

The experiment $\mathsf{ZK}_{A_{\mathsf{zk}}}$ (that is, when D_{zk} is given as input the real proof), perfectly simulates the experiment $\mathsf{NME}_b(\Pi, A, k, p(k))$, whereas the experiment $\mathsf{ZK}^S_{A_{\mathsf{zk}}}$ (that is, when D_{zk} is run with a simulated proof) perfectly simulates $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k))$. If the outputs of D in the experiments are different, then D_{zk} distinguishes between a real proof and a simulated proof, contradicting the adaptive zero-knowledge of the NIZK proof system $(\mathcal{D}, P, V)$. □ □

Next, we show that experiments $\mathsf{NME}_b^{(1)}(\cdots)$ and $\mathsf{NME}_b^{(2)}(\cdots)$ are statistically indistinguishable. To this end, we define three events, $\mathtt{badNIZK}(\mathsf{Expt})$, $\mathtt{badSig}(\mathsf{Expt})$ and $\mathtt{badKey}(\mathsf{Expt})$, corresponding to the experiment Expt. We show that the experiments $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are *identical*, under the assumption that the events $\mathtt{badNIZK}$, $\mathtt{badSig}$ and $\mathtt{badKey}$ *never* happen in these experiments. Then, we show that the bad events happen with negligible probability in both the experiments. Taken together, these two statements let us conclude that $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are statistically indistinguishable. Details follow.

The proof of the following claim is similar in structure to that in [PSV06a]. The difference is in (1) Subclaim 11, where we use the m-bounded strong soundness of the designated verifier NIZK, as opposed to ordinary soundness and, (2) Subclaim 12, where we use the strong adaptive zero-knowledge property of the designated verifier NIZK.

Claim 10 $\left\{\mathsf{NME}_0^{(1)}(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \stackrel{s}{\approx} \left\{\mathsf{NME}_0^{(2)}(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}}$

Proof: Define the event $\mathtt{badNIZK}(\mathsf{Expt})$, to capture the event that the adversary A violates the soundness of the NIZK proof system in experiment Expt (i.e, the adversary produces a false statement together with an accepting proof).

More precisely, let ψ denote a ciphertext that the adversary produces (this could either be a query to the decryption oracle or one of the ciphertexts in its output). Let $\mathtt{badNIZK}(\mathsf{Expt})$ denote the following event: In experiment Expt, there exists a ciphertext ψ that the adversary produces in which: (1) the NIZK proof in ψ_j is accepted by the verifier V, but (2) all the k ciphertexts that are part of ψ do not decrypt to the same value (in other words, ψ contains an *accepting* proof of a *false* statement).

In the subclaims below, we show that $\mathtt{badNIZK}(\mathsf{NME}_b^{(j)})$ happens only with negligible probability.

Subclaim 11 *For* $b \in \{0,1\}$, $\Pr[\mathtt{badNIZK}(\mathsf{NME}_b)] = \mathit{negl}(k)$

Proof: Suppose, for contradiction, that this is not true. That is, there is a polynomial $q(k)$ such that $\Pr[\texttt{badNIZK}(\mathsf{NME}_b] \geq \frac{1}{q(k)}$. Then, we construct a machine A_s that violates the soundness of the proof system $(\mathcal{D}, P, V)$ with probability at least $\frac{1}{p(k)q(k)}$. A_s can also access the verifier oracle at most m times.

On input a public parameter PP, A_s works as follows:

1. Simulate the experiment NME_b using PP, until A_2 outputs $p(k)$ ciphertexts. Note that A_s does not need to know the secret parameter SP to perform these steps – to answer the decryption queries, A_s simply uses the verifier oracle to check the correctness of the NIZK proof in the decryption query.

2. A_s picks at random one of the ciphertexts that the adversary produces (which includes both the adversary's queries to the decryption oracle, as well as his output ciphertexts). Say the ciphertext chosen is $[\vec{c}, \pi, \text{VKSIG}, \sigma]$. Output the pair $(\vec{c}, \pi)$.

The probability that A_s outputs a false statement and an accepting proof pair is, by our assumption, at least $\frac{1}{p(k)q(k)}$, which is a contradiction to the m-bounded strong soundness of $(\mathcal{D}, P, V)$. □

The proof of the subclaim below follows [PSV06a] exactly, except for the use of strong adaptive zero-knowledge, in the same way it was used in Claim 9.

Subclaim 12 *For* $b \in \{0,1\}$, $\Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(1)})] = \Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(2)})] = \mathit{negl}(k)$.

Proof: We start by noting that $\Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(1)})] = \Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(2)})]$. This follows because the adversary's view in experiments $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are identical until the point when the adversary A_2 outputs the ciphertexts. We proceed to show that for $b \in \{0,1\}$, $\Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(1)})]$ is negligible in k. This is shown by an argument similar to the one used in the proof of Claim 9. Assume, for contradiction, that $\Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(1)})]$ is non-negligible. Then, we construct a pair of machines $(A_{\mathsf{zk}}, D_{\mathsf{zk}})$ that violate the adaptive zero-knowledge of the proof system $(\mathcal{D}, P, V)$.

On input a public parameter PP for the NIZK proof system, A_{zk} and D_{zk} work exactly as in the proof of Claim 9, except that in Step 3, when A_2 returns a sequence of ciphertexts $(\psi_1, \ldots, \psi_{p(k)})$, D_{zk} looks for a ciphertext ψ_i such that not all the components of ψ_i decrypt to the same message, and the NIZK proof in ψ_i is accepting. If there exists such an i, then D_{zk} returns "Fail" and otherwise returns "OK".

Note that by definition, when D_{zk} receives a real proof, it outputs "Fail" with probability $\Pr[\texttt{badNIZK}(\mathsf{NME}_b)]$. On the other hand, when run on a simulated proof, it outputs "Fail" with probability $\Pr[\texttt{badNIZK}(\mathsf{NME}_b^{(1)})]$. However, in the previous subclaim, we showed that the former probability is negligible. If the latter probability is non-negligible, then D_{zk} distinguishes between a simulated proof and a real proof, contradicting the adaptive zero-knowledge property of the proof system $(\mathcal{D}, P, V)$. □

Let $\psi_i = [\vec{c}_i, \pi_i, \text{VKSIG}_i, \sigma_i]$ denote the i^{th} ciphertext returned by A_2. Define $\texttt{badSig}(\mathsf{NME}_b^{(j)})$ to be the event that, in experiment $\mathsf{NME}_b^{(j)}(\Pi, A, k, p(k))$, there exists an index i such that $\text{VKSIG}_i = \text{VKSIG}$ and $\mathsf{Ver}(\text{VKSIG}_i, \vec{c}_i, \pi_i) = \text{ACCEPT}$. Since the signature scheme is (strongly) existentially unforgeable, it follows that, for $b \in \{0,1\}$ and $j \in \{1,2\}$, $\Pr[\texttt{badSig}(\mathsf{NME}_b^{(j)})] = \mathsf{negl}(k)$.

Let $\texttt{badKey}(\mathsf{NME}_b^{(j)})$ denote the event that for one of the public keys, say $\hat{\text{PK}}$, generated in the experiment $\mathsf{NME}_b^{(j)}$, there exists a pair of messages m, m' and random coins r, r' such that $m \neq m'$ and $\mathsf{Enc}(\hat{pk}, m, r) = \mathsf{Enc}(\hat{pk}, m', r')$. Since the encryption scheme used is perfectly correct, by the union bound, we have $\Pr[\texttt{badKey}(\mathsf{NME}_b^{(j)})] = \mathsf{negl}(k)$.

Let $\texttt{fail}_b(\cdot)$ denote the event $\texttt{badNIZK}(\cdot) \vee \texttt{badSig}(\cdot) \vee \texttt{badKey}(\cdot)$. It follows, by a union bound, that $\Pr[\texttt{fail}_b(\mathsf{NME}_b^{(j)})] = \mathsf{negl}(k)$, for $j \in \{1,2\}$.

We show that conditioned on the event $\texttt{fail}_b(\mathsf{NME}_b^{(j)})$ (for $j \in \{1,2\}$) not happening, $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are identical. Note that the view of A in both the experiments is (syntactically) the same. Since $\texttt{badSig}(\mathsf{NME}_b^{(j)})$ does not happen, A uses a different verification key in all the ciphertexts ψ_i it returns. This means that $\mathsf{NME}_b^{(j)}$ can decrypt at least *one* of the components of each ψ_i, using a secret-key it knows, to get a message m_i. Since $\texttt{badNIZK}(\mathsf{NME}_b^{(j)})$ does not happen, m_i must be the message that is encrypted in all the other components of ψ_i too. Thus, ψ_i is a *valid* encryption of m_i. Also, since $\texttt{badKey}(\mathsf{NME}_b^{(j)})$ does not happen, m_i is the *unique* such message. Thus the tuple of messages returned in both $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are exactly the same, and thus the outputs of $\mathsf{NME}_b^{(1)}$ and $\mathsf{NME}_b^{(2)}$ are identical.

Combining the above with the fact that the events $\texttt{fail}_b(\cdot)$ occur with a negligible probability, we have $\mathsf{NME}_b^{(1)}(\Pi, A, k, p(k)) \stackrel{s}{\approx} \mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))$. □

The proof of the following claim is identical to that in [PSV06a].

Claim 13 *For every p.p.t. machine A, there exists a p.p.t. machine B such that for $b \in \{0,1\}$,*

$$\left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \equiv \left\{\mathsf{IND}_b(E', B, k)\right\}_{k\in\mathbb{N}}$$

Proof: The machine B is constructed as follows. B simply simulates the experiment $\mathsf{NME}_b^{(2)}$, except that instead of generating PK by itself, it uses $\text{PK} = \{\text{PK}_i\}_{i=1}^k$ received from the outside. Let (m_0, m_1) be the pair of messages the adversary A_1 returns. B then outputs (m_0, m_1) and receives a challenge ciphertext c_b from the outside. B performs the same operations as the experiment $\mathsf{NME}_b^{(2)}$ to generate the challenge ciphertext C_b for A_2. Finally, A_2 returns a sequence of ciphertexts $(\psi_1, \psi_2, \ldots, \psi_{p(k)})$. B decrypts these ciphertexts just as in $\mathsf{NME}_b^{(2)}$ and outputs the plaintexts. (Note that $\mathsf{NME}_b^{(2)}$ uses only SK' and not SK in order to decrypt the messages).

It is easy to see that B simulates the experiment $\mathsf{NME}_b^{(2)}$ perfectly using the public-keys and ciphertexts received from the outside, and thus

$$\left\{\mathsf{NME}_b^{(2)}(\Pi, A, k, p(k))\right\}_{k\in\mathbb{N}} \equiv \left\{\mathsf{IND}_b(E', B, k)\right\}_{k\in\mathbb{N}}$$

□

To conclude the proof, we combine the last three claims to conclude that for every p.p.t. adversary A, there is a p.p.t. adversary B such that $\mathsf{NME}_b(\Pi, A, k, p(k)) \stackrel{c}{\approx} \mathsf{NME}_b^{(1)}(\Pi, A, k, p(k)) \stackrel{s}{\approx} \mathsf{NME}_b^{(2)}(\Pi, A, k, p(k)) \equiv \mathsf{IND}_b(E', B, k)$. Since by the semantic security of E', $\mathsf{IND}_0(E', B, k) \stackrel{c}{\approx} \mathsf{IND}_1(E', B, k)$, it holds that $\mathsf{NME}_0(\Pi, A, k, p(k)) \stackrel{c}{\approx} \mathsf{NME}_1(\Pi, A, k, p(k))$. □

5 Separating Bounded IND-CCA2 from NM-CPA

In this section, we show that under bounded chosen cipher attacks, non-malleability of the encryption scheme is not immediately implied by indistinguishability. In particular, we show an encryption scheme that is indistinguishable-secure under a k-bounded cca attack, but *not* even non-malleable under even a chosen plaintext attack. In contrast, it has been shown that unlimited IND-CCA2 security implies (some form of) non-malleability (See [PSV06b] for a discussion).

Theorem 14 *If there exists an m-bounded* IND-CCA *secure cryptosystem Π, then there exists another m-bounded* IND-CCA *secure cryptosystem Π' that is not* NM-CPA*-secure.*

$\mathsf{Gen}'(1^k)$: Run $\mathsf{Gen}(1^k)$ and get a pair of keys (PK, SK). Suppose SK is an ℓ-bit string. Choose a random degree-m polynomial $p(x) = p_m x^m + \cdots + p_1 x + \text{SK}$ with coefficients in $GF(2^\ell)$ and whose constant term is SK. Output $\text{PK}' = \text{PK}$ and $\text{SK}' = (\text{SK}, p)$.

$\mathsf{Enc}'(\text{PK}, m)$: Get $c \leftarrow \mathsf{Enc}_{\text{PK}}(m)$ and output $(0, c)$.

$\mathsf{Dec}'(\text{SK}, c)$: Parse c as (c_1, c_2). If $c_1 = 0$, output $\mathsf{Dec}(\text{SK}, c_2)$. Else, if $c_2 > 0$, output $p(c_2)$ and otherwise return 0.

Figure 3: AN IND-m-CCA ENCRYPTION SCHEME Π' WHICH IS MALLEABLE.

Remark: Theorem 6 shows that the existence of a semantically-secure cryptosystem implies the existence of an m-bounded IND-CCA cryptosystem. Therefore, the "if" clause of the above theorem can be simplified. However, we choose to present it as is to highlight the point that bounded IND-CCA2 does not imply bounded non-malleability.

Proof: Assume that there exists an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is m-bounded IND-CCA2-secure. Then, we construct an encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ (given in Figure 5) that is also m-bounded IND-CCA2-secure, but is not NM-CPA-secure. The proof follows from the following two claims.

Claim 15 $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *is* m*-bounded* IND-CCA2*-secure.*

Proof: Suppose not. We use the adversary A that breaks the security of Π' to construct an m-bounded IND-CCA2 attack against Π. The new adversary A', on input PK, simply runs $A(\text{PK})$. When asked to decrypt a ciphertext $(0, c)$, it forwards the query to its own decryption oracle. When asked to decrypt a ciphertext of the form $(1, c_2)$, it returns either 0 if $c_2 = 0$ or a random value. Since A makes at most m queries, then A' will be able to answer all queries. The simulation is perfect because the degree-m polynomial $p(\cdot)$ is m-wise independent. This adversary A' succeeds with the same probability as A, which contradicts the assumption that Π is m-bounded secure. □

Claim 16 $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *is not* NM-CPA*-secure.*

Proof: Without loss of generality, assume that the message space of Π include the bits 0 and 1. On input a public key PK, the adversary submits as a message pair, 0 and 1.

Upon receiving a ciphertext c, the attacker first computes $\alpha = \mathsf{Enc}(\text{PK}, c)$. It then returns the vector $(\alpha, \beta_1, \ldots, \beta_{m+1})$ where $\beta_i = (1, i)$.

Notice that the output of the experiment is the vector $(c, p(1), \ldots, p(m+1))$. The distinguisher D now works as follows. It first uses $p(1), \ldots, p(m+1)$ to interpolate the secret key SK, and then runs $\mathsf{Dec}(\text{SK}, c)$ and prints the result as its output.

The distinguisher's output in the NME_0 experiment will therefore be 0 and its output in the NME_1 will be 1, which shows that Π' is not even NM-CPA secure.

As one final point, it may be that the message space of Π does not include the ciphertext — for example, the size of the ciphertext may be too big. This is easily handled. The adversary can simply encode c in a bit-by-bit fashion over many ciphertexts, and the distinguisher can simply reconstruct c to perform its test. □ □

References

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *CRYPTO*, pages 519–536, 1999. 4

[CD00] Jan Camenisch and Ivan B. Damgård. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. In *ASIACRYPT*, pages 331–345, 2000. 5

[CHK] Ronald Cramer, Dennis Hofheinz, and Eike Kiltz. A note on bounded chosen ciphertext security from black-box semantical security. manuscript, 2006. 2

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998. 1

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000. 1, 6, 7

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *EUROCRYPT*, pages 342–360, 2004. 3

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. 1, 2

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and cca-security for public-key encryption. In *TCC*, 2007. 2

[Gol04] Oded Goldreich. *Foundations of Cryptography, Volume 2*. Cambridge University Press, 2004. 7

[HI] Goichiro Hanaoka and Hideki Imai. A generic construction of chosen ciphertext secure cryptosystems without non-interactive zero-knowledge proofs. manuscript, 2006. 2

[KMO89] Joe Kilian, Silvio Micali, and Rafail Ostrovsky. Minimum resource zero-knowledge proofs. In *FOCS*, pages 474–479, 1989. 5

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, October 1979. 7

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 427–437, New York, NY, USA, 1990. ACM Press. 1

[PSV06a] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from a any semantically secure one. In *CRYPTO*, pages –, 2006. 1, 4, 5, 6, 7, 8, 10, 11, 12

[PSV06b] Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Definitions of non-malleable encryption. 2006. manuscript. 12

[Rom90] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990. 7

SIMULATOR (S_1, S_2) FOR DESIGNATED VERIFIER NIZK

$S_1(1^k)$: Follow the instructions of the sampling algorithm $\mathcal{D}(1^k)$ and output $(\text{PP}_{dv}, \text{SP}_{dv}, \epsilon)$.

$S_2(\text{PP}_{dv}, \text{SP}_{dv}, x, \text{STATE})$: Recall that SP_{dv} contains bits $f_1, \ldots, f_k$. For each $i = 1, \ldots, k$, run the Σ-protocol simulator $S_\Sigma(f_i)$ to produce transcript (a_i, f_i, c_i). Output the proof

$$\pi' \stackrel{\text{def}}{=} \left[a_i, \mathsf{Enc}_{\text{PK}_i^0}((1-f_i)\cdot c_i), \mathsf{Enc}_{\text{PK}_i^1}(f_i \cdot c_i)\right]_{i=1}^{k}$$

(One encryption is always an encryption of 0, while the other is one of c_i.)

[RS93] Charles Rackoff and Daniel R. Simon. Cryptographic defense against traffic analysis. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 672–681, New York, NY, USA, 1993. ACM Press. 1

A Proof of Adaptive Zero-knowledge of the Designated Verifier Proof System

Proposition 17 *$(\mathcal{D}, P, V)$ satisfies adaptive zero-knowledge.*

Proof: At a high level, adaptive zero-knowledge follows from the zero-knowledge of the 3-round Σ protocol and the semantic security of the encryption scheme. For any theorem-choosing algorithm A, we construct a simulator $S = (S_1, S_2)$ that works as follows.

To show that the distributions in EXPTZK_A and EXPTZK_A^S are indistinguishable, we present the following series of games. For convenience of notation, we say that the proof π consists of k triples $(a_i, \alpha_i^0, \alpha_i^1)$ where α_i^0 and α_i^1 are encryptions.

Game 0: Same as EXPTZK_A except $\mathcal{D}$ is replaced by S_1.

Game 1 **through** k**:** Same as Game 0, except that in the first i triples of the proof π, the ciphertext $\alpha_i^{1-f_i}$ is replaced by $\mathsf{Enc}_{\text{PK}_i}(0)$.

Game $k+1$ **through** $2k$**:** Same as Game k, except that the first i triples of the proof π are generated by S_2 and and the remaining $k-i$ proofs are generated by P.

Notice that EXPTZK_A is identical to Game 0 and EXPTZK_A^S is identical to Game $2k$. We establish $\text{EXPTZK}_A \stackrel{c}{\approx} \text{EXPTZK}_A^S$ through the following two claims, which contradict the assumption.

Claim 18 *Game* 1 *is indistinguishable from Game* k*.*

(Breaking the encryption.) Suppose for the sake of reaching contradiction, that there exists an algorithm D which distinguishes Game 1 from Game k with non-negligible advantage η. This implies there exists some j for which D distinguishes game Game j^* and Game j^*+1 with advantage at least η/k.

We construct an adversary B' which violates the semantic security of Enc. B' first guesses $j \in [1, k]$. It then begins to run Game j with the following modifications. Let $(a_j, c_{0,j}, c_{1,j})$ be the Σ-protocol prover messages used in the jth triple of the proof for x. B' submits the messages $(0, c_{(1-f_j),j})$ as its challenges

in the semantic security game. (Recall in an indistinguishability attack, one of these messages is randomly chosen, encrypted and returned to B'. Let us denote the returned challenge ciphertext as y.) Upon receipt of ciphertext y, B' produces the proof π exactly as described in Game j with the exception that it uses y in place of $\alpha_j^{1-f_j}$. Finally, B' feeds the resulting proof π to D and echoes D's output.

Conditioned on guessing j correctly, observe that the distribution of π is identical to that of Game j^* if y is an encryption of c_{1-f_j} and that of Game $j^* + 1$ otherwise. Thus, a probability calculation shows that B''s advantage in breaking the encryption scheme is $\frac{\eta}{k} \cdot \frac{1}{k}$, which contradicts the security of Enc.

Claim 19 *Game k is indistinguishable from Game $2k$.*

(Breaking the Σ-protocol simulator.) A hybrid argument similar to the one used in Claim 18 applies. Assume by contradiction, there exists some j^* and D which distinguishes Game j^* and Game $j^* + 1$ with advantage at least η/k

B'' receives as input a transcript (a, b, c) and must decide if the proof was simulated or not. If $V_2(a, b, c) = 0$ (i.e., the transcript is not accepting), then output 0 immediately. Otherwise, guess $j \in [1, k]$. If $f_j \neq b$, then output a random guess. Otherwise, use $(\mathrm{PK}, \mathrm{SK}, x, w)$ (which is given as non-uniform advice) to generate a proof as described in Game j. Replace the jth triple with $(a, \mathsf{Enc}_{\mathrm{PK}_j^0}((1-b) \cdot c), \mathsf{Enc}_{\mathrm{PK}_j^1}(b \cdot c))$, feed the resulting proof π to D and echo its output.

Once again, conditioned on guessing j correctly and on $f_j = b$, the distribution of π is identical to that of Game j^* if the input transcript is a real prover transcript, and identical to that of Game $j^* + 1$ if the transcript is simulated. Recall that f_j is chosen uniformly, and so $\Pr[f_j = b] = 1/2$. Thus, B'''s advantage in breaking the Σ-protocol simulator is $\frac{\eta}{2k^2}$ which is a contradiction. $\square$