A Black-Box Construction of a CCA2 Encryption Scheme from a Plaintext Aware Encryption Scheme

Dana Dachman-Soled
University of Maryland
danadach@ece.umd.edu

October 14, 2013

Abstract

We present a construction of a CCA2-secure encryption scheme from a plaintext aware, weakly simulatable public key encryption scheme. The notion of plaintext aware, weakly simulatable public key encryption has been considered previously by Myers, Sergi and shelat (SCN, 2012) and natural encryption schemes such as the Damgård Elgamal Scheme (Damgård, Crypto, 1991) and the Cramer-Shoup Lite Scheme (Cramer and Shoup, SIAM J. Comput., 2003) were shown to satisfy these properties.

Recently, Myers, Sergi and shelat (SCN, 2012) defined an extension of non-malleable CCA1 security, called cNM-CCA1, and showed how to construct a cNM-CCA1-secure encryption scheme from a plaintext aware and weakly simulatable public key encryption scheme. Our work extends and improves on this result by showing that a full CCA2-secure encryption scheme can be constructed from the same assumptions.

Key words: CCA2-secure encryption, plaintext aware encryption, weakly simulatable public key encryption, black-box

1 Introduction

The basic security requirement for public key encryption schemes is Chosen Plaintext Attack (CPA) security [GM84] (also known as semantic security), which ensures security against a *passive*, eavesdropping adversary. A stronger security requirement for public key encryption schemes, which ensures that they remain secure even in the face of an *active* adversary, is known as Adaptive Chosen Ciphertext Attack (CCA2) security. More specifically, a CCA2-secure encryption scheme is guaranteed to be secure even against an adversary who has access to a decryption oracle and may use it to decrypt any ciphertext of its choice except for the challenge ciphertext itself. This captures real-life scenarios where the adversary has control over network traffic which allows the adversary, in effect, to decrypt all ciphertexts of its choice.

There is a significant body of work on constructing CCA2-secure encryption schemes from specific computational hardness assumptions (c.f. [CS02, HK08, CKS09, HJKS10]), as well as from various lower level primitives (c.f. [DDN00, CHK04, Kil06, PW11, RS10, KMO10, Wee10]). Nevertheless, the central question in this area remains open: To determine the relationship between CCA2 and CPA-secure encryption—whether a CCA2-secure encryption scheme can be constructed assuming only the existence of a CPA-secure encryption scheme, or whether CCA2-security requires stronger assumptions. Although a partial answer was given in [GMM07], the larger question remains open for both black-box and non-black-box constructions. Moreover, several important variants of the question such as whether a CCA2 secure encryption scheme can be constructed from a CCA1-secure encryption scheme[1] remain open.

In this paper, we consider a strong type of CPA-secure public key encryption scheme which is also *plaintext aware*, *weakly simulatable*, and enjoys *perfect correctness*[2] and show how to construct full CCA2-secure public key encryption schemes from such a CPA-secure encryption scheme. Moreover, the CCA2 construction presented is black-box in the underlying CPA-secure scheme.

Although the required assumptions are strong—we discuss and provide more details on the assumptions of plaintext awareness and weak simulatability below—we view our new construction of CCA2 encryption from plaintext aware, weakly simulatable PKE as meaningful progress since our underlying assumption is an assumption which was not previously known to imply CCA2 security. Moreover, to the best of our knowledge, this is the first construction of a CCA2 scheme from encryption schemes with seemingly weaker or incomparable security to CCA2 and requiring no additional assumptions. Finally, we present new proof techniques for proving CCA2 security, which may be useful for constructing CCA2 secure encryption from other lower-level primitives.

1.1 Our Assumptions

Our work relies on a strong assumption on the underlying CPA-secure encryption scheme called *plaintext awareness*. The notion of a plaintext aware encryption scheme was first introduced in the seminal paper of Bellare and Rogaway [BR94] and the notion was further studied by Bellare et al. [BDPR98]. Both of these works dealt with the notion of plaintext awareness in the Random Oracle model. Subsequently, Bellare and Palacio [BP04b] considered extending the notion of plaintext awareness to the plain model[3]. In this work, we are also interested in the notion of plaintext awareness in the plain model without random oracles. Informally, an encryption scheme is plaintext aware (called sPA1 in [BP04b]) if for every efficient ciphertext creator, C, there exists an efficient plaintext extractor, C^*, that outputs the same value as the decryption algorithm on ciphertexts outputted by C. This type of assumption is known as

[1] A CCA1-encryption scheme is one where the adversary has oracle access to the decryption oracle up to the point that it receives the challenge ciphertext.

[2] We can remove the requirement of perfect correctness by using the transformation of [DNR04] to transform a public key encryption scheme with decryption error to a public key encryption scheme with perfect correctness. Note that each transformation in the sequence of transformations given in the proof of Theorem 3 of [DNR04] preserves both simulatability and plaintext awareness of the underlying encryption scheme.

[3] We note that prior to the work of [BP04b], Herzog et al. [HLM03] considered a notion of plaintext awareness in the key registration model.

a knowledge assumption (other examples of knowledge assumptions include the knowledge of exponent assumption [HT98, BP04a] and extractable collision resistant hash functions [BCCT12]) and is thus a non-falsifiable assumption. Despite the strength of the assumption, the notion of plaintext awareness is meant to capture an intuitive property of certain encryption schemes that an efficient adversary cannot create a valid ciphertext without "knowing" the corresponding plaintext.

It is not hard to see that any plaintext aware encryption scheme is itself also CCA1-secure, since the plaintext extractor can be used to simulate the decryption oracle in the CCA1 experiment. However, plaintext aware encryption *does not* directly imply CCA2-secure encryption since the plaintext extractor is not guaranteed to work correctly when the ciphertext creator receives a valid encryption as input. Thus, when the adversary queries the CCA2 decryption oracle after receiving the challenge ciphertext CT^* in the CCA2 experiment, the extractor may not be able to simulate the decryption oracle. In fact, since we are given no guarantees on the output of the plaintext extractor when the ciphertext creator receives CT^* as input, it would seem that constructing CCA2-secure encryption from plaintext aware encryption is just as hard as constructing CCA2-secure encryption from CCA1-secure encryption; we have the extra guarantee of a plaintext extractor, but the extractor seems useless for queries made after the challenge ciphertext is received.

Recently, a fascinating result by Myers, Sergi and shelat [MSS12], showed that by adding an additional assumption that the plaintext aware public key encryption scheme is also weakly simulatable, the above problem can be partially overcome. Essentially, they present a new construction and show that the plaintext extractor can still be useful for simulating the decryption oracle for a *constant* number of parallel queries made after the adversary receives the challenge ciphertext when the underlying plaintext aware public key encryption scheme is also *weakly simulatable*.

The notion of *simulatable* public key encryption was first introduced by Damgård and Nielsen [DN00] in the context of non-committing encryption. Loosely speaking, [DN00] define a simulatable public key encryption scheme to be an encryption scheme with special algorithms for obliviously sampling public keys and random ciphertexts without learning the corresponding secret keys and plaintexts; in addition, both of these oblivious sampling algorithms should be eficiently invertible. An incomparable notion of *simulatable* public key encryption was introduced by [Den06] and was shown to imply CCA2-secure encryption. Here, the public key encryption scheme has an invertible algorithm f for obliviously sampling random ciphertexts (but not public keys) and in addition, $f(r)$, where r is a random string r and C, where C is an honestly generated ciphertext are indistinguishable, *even when given access to a decryption oracle*. The weakly simulatable encryption schemes used in this work are strictly weaker than both of the above notions. They are weaker than the [DN00] notion since only the *ciphertext* and not the public key has an invertible oblivious sampling algorithm and they are weaker than the [Den06] notion since the attacker is not given access to the decryption oracle.

In their work, [MSS12] defined an extension of non-malleable CCA1 security, called cNM-CCA1, where an adversary can make c adaptive parallel decryption queries after seeing the challenge ciphertext. Then, [MSS12] showed how to construct cNM-CCA1 encryption from plaintext aware and weakly simulatable public key encryption for any constant c. Similar assumptions of plaintext aware and weakly simulatable public key encryption were previously made by [Den06]. Moreover, as shown by Myers, Sergi and shelat [MSS12] natural encryption schemes such as the Dåmgard Elgamal encryption scheme (DEG) and the lite version of Cramer-Shoup encryption scheme (CS-lite) satisfy both of these properties under the DDH assumption and a suitable extension of the Diffie-Hellman Knowledge (DHK) assumption (see [BP04b] for discussion of the DHK assumption).

Following the work of [MSS12], it is interesting to explore how far we can take the assumption of the existence of a plaintext aware and weakly simulatable public key encryption scheme and what the power of this assumption is relative to the assumption of the existence of a CCA2-secure encryption scheme.

1.2 Our Results

Informally, we show the following:

Theorem 1.1 (Informal) *There is a black-box construction of* CCA2*-secure encryption from plaintext aware and weakly simulatable public key encryption with perfect correctness.*

Our result extends the work of [MSS12] by showing that plaintext aware and weakly simulatable public key encryption can, in fact, be used to achieve full CCA2 security.

Finally, the assumption of a plaintext aware encryption scheme can be viewed as an assumption that allows us to use strong non-black-box techniques on the *adversary* in the security reduction. More specifically, we leverage the code of the adversary by using it to extract crucial information that the adversary must "know." This raises the intriguing question of whether we can present a construction of CCA2 from CPA where the security proof uses non-black-box access to the adversary. Such reductions are known to be more powerful than black-box reductions in the setting of multiparty computation as first shown in the seminal work of Barak [Bar01]. But it has not been clear how to leverage these techniques in the non-interactive setting of public key encryption.

1.3 Technical Overview

We adapt and combine many of the techniques of [HLW12], [MSS12] and, in addition, we introduce new techniques as discussed in detail below.

The construction. On security parameter k, the scheme will consist of a one-time signature as well as both inner and outer ciphertexts, with corresponding public keys. More specifically, two inner ciphertexts will be encrypted under public keys pk_{in_0}, pk_{in_1}, and k outer ciphertexts will be encrypted using k public keys chosen out of k pairs of public keys $(pk_1^0, pk_1^1), \ldots, (pk_k^0, pk_k^1)$. The selection of the k public keys $pk_1^{b_1}, \ldots, pk_k^{b_k}$ will depend on bits of the verification key, vksig, chosen for the one-time signature (as in [DDN00, MSS12]).

In particular, a ciphertext will consist of the following:

Verification key: A verification key, vksig, for the one-time signature scheme, generated by GenSig.

Inner ciphertexts: Two ciphertexts $CT_{in_0} = \mathsf{Enc}(pk_{in_0}, \tilde{s}_0)$, $CT_{in_1} = \mathsf{Enc}(pk_{in_1}, \tilde{s}_1)$ where $\tilde{s}_0, \tilde{s}_1$ are additive secret shares of $m||r$, m is the message to be encrypted, r is the randomness used to encrypt the outer ciphertexts (as described below), and $||$ denotes concatenation.

Outer ciphertexts: k ciphertexts $CT_1, \ldots, CT_k$ computed the following way: $r_1||\cdots||r_k \leftarrow \mathsf{prg}(r)$, where prg is a pseudorandom generator. Each $CT_i = \mathsf{Enc}(pk_i^{\mathsf{vksig}_i}, CT_{in_0}||CT_{in_1}; r_i)$.

Signature: A signature $\sigma = \mathsf{Sign}(\mathsf{sksig}, CT_1||\cdots||CT_k)$.

The security reduction. We consider a modified CCA2 experiment where the decryption oracle is replaced with the plaintext extractor guaranteed by the plaintext awareness property of the underlying encryption scheme. Note that once the adversary receives the challenge ciphertext in the CCA2 experiment, we have no guarantees on whether the plaintext extractor returns messages that are consistent with the answers of the decryption oracle. Therefore, we define a *bad extraction event* as the event that the plaintext extractor and decryption oracle disagree on a query submitted by the adversary A to the decryption oracle. We consider a sequence of hybrids and show that (1) In the first hybrid, the probability of bad extraction event ocurring is negligible (due to the security guarantees of the plaintext aware, weakly simulatable encryption scheme) and (2) In consecutive hybrids the probability of bad extraction event occurring differs by a negligible amount (since the occurrence of a bad extraction event can be detected in each hybrid). Put together, these imply that the decryption oracle and plaintext extractor agree (even for queries after the challenge ciphertext is received) in the original experiment with all but negligible probability. Furthermore, this implies that the CCA2 experiment can be simulated without knowing the secret key of the inner encryption scheme (by using the plaintext extractor to decrypt oracle

queries), which immediately implies the CCA2 security of the scheme. To show (1), we use techniques similar to those of [MSS12]. To show (2), we build upon the sequence of hybrids used by [HLW12].

The main new technical challenge in this work is showing that property (2) holds for each pair of consecutive hybrids. More specifically, in the final two hybrids, which we denote here by $\tilde{H}_0, \tilde{H}_1$, we run the CCA2 experiment with the CCA2 adversary, but use the plaintext extractor to decrypt the inner ciphertexts CT_{in_0}, CT_{in_1}. Additionally, in $\tilde{H}_0$, the value $\tilde{s}_0 \oplus \tilde{s}_1$ is set to a random string, while in $\tilde{H}_1$, value $\tilde{s}_0 \oplus \tilde{s}_1$ is set honestly to $(r||m_\beta)$. Note that if a bad extraction event does not occur, then the view of the adversary in $\tilde{H}_1$ is identical to its view in the original CCA2 experiment. By previous arguments, we have that the probability of a bad extraction event is negligible in $\tilde{H}_0$. To argue that the probability of bad extraction event occurring differs by a negligible amount in these final two hybrids, we must reduce to the semantic security of the inner encryption scheme. However, a bad extraction event—in which the plaintext extractor disagrees with the decryption oracle—cannot be detected unless the adversary has the secret keys corresponding to the inner encryptions and if this is the case, it seems that we cannot hope to reduce to semantic security.

Thus, we consider a modified experiment where at the beginning of the experiment we fix a bit $b \leftarrow\!\!\$ \{0,1\}$ and a *modified bad extraction event* defined as the event that the plaintext extractor and decryption oracle disagree specifically on the decryption of CT_{in_b} for a query submitted by the adversary A. Since $b \leftarrow\!\!\$ \{0,1\}$ is chosen uniformly at random, independent of all other variables, we show that the probability that the *first* bad extraction event occurs on CT_{in_b} is exactly half the probability that the first bad extraction event occurs on either CT_{in_0} or CT_{in_1}. Now, if the first bad extraction event occurs on CT_{in_b}, then the semantic security adversary successfully detects the first bad extraction event. In the following, we will give additional details on the semantic security adversary.

An adversary attempting to break the semantic security of the inner encryption simulates the modified experiment in the final two hybrids by fixing $b \leftarrow\!\!\$ \{0,1\}$, embedding its challenge public key in place of $pk_{in_{1-b}}$ and embedding a public key with a known secret key in place of pk_{in_b}. Now, due to the additive secret sharing of the inner plaintexts, we have that in both hybrids $\tilde{H}_0$ and $\tilde{H}_1$, each of the inner plaintexts $\tilde{s}_0, \tilde{s}_1$ are *individually* uniformly distributed. Leveraging this property, the semantic security adversary chooses a random plaintext $\tilde{s}_b$ and computes an honest encryption $CT^*_{in_b}$ of $\tilde{s}_b$. To embed the remaining inner ciphertext, the adversary chooses $\beta \in \{0,1\}$ and computes: $\tilde{s}^0_{1-b} \leftarrow\!\!\$ \{0,1\}^\ell, \tilde{s}^1_{1-b} = (r||m_\beta) \oplus \tilde{s}_b$ where m_0, m_1 are the plaintexts submitted by the internal CCA2 adversary, and submits $\tilde{s}^0_{1-b}, \tilde{s}^1_{1-b}$ to the external semantic security experiment. The semantic security adversary receives in return a ciphertext $CT_{\mathsf{pa\text{-}cpa}}$, which is an encryption of either $\tilde{s}^0_{1-b}$ or $\tilde{s}^1_{1-b}$ and sets $CT^*_{in_{1-b}} = CT_{\mathsf{pa\text{-}cpa}}$. Note that when the external experiment returns an encryption of $\tilde{s}^0_{1-b}$, the generated ciphertext is identically distributed to the challenge ciphertext in $\tilde{H}_0$ and when the external experiment returns an encryption of $\tilde{s}^1_{1-b}$, the generated ciphertext is identically distributed to the challenge ciphertext in $\tilde{H}_1$. Finally, since the adversary has the secret key corresponding to pk_{in_b}, it will successfully detect the case that a bad extraction event occurs for the first time on $CT^*_{in_b}$. Thus, we have that the occurrence of modified bad extraction event with non-negligible probability in $\tilde{H}_1$, implies an attack on the CPA security of the underlying scheme.

1.4 Related Work

In their seminal work, Dolev et al. [DDN00] presented the first construction of CCA2-encryption from the lower-level primitive of enhanced trapdoor permutations. However, the [DDN00] construction is not black-box and requires the use of generic non-interactive zero knowledge proofs. Subsequently, Pass et al. [PSV06] presented a new definition of non-malleability and presented a construction from CPA to non-malleable CPA requiring non-black box use of the underlying encryption scheme. Choi et al. [CDSMW08] gave a black-box version of this result thereafter. Myers and shelat [MS09] showed how to construct many-bit CCA2-encryption from single-bit cca2-encryption and Hohenberg et al. [HLW12] extended their result and showed how to build CCA2-encryption from any detectable chosen ciphertext (DCCA) secure encryption scheme. As discussed previously, [MSS12] show how to construct a cNM-CCA1-

secure encryption scheme from a plaintext aware, weakly simulatable public key encryption scheme.

A different line of work introduced new low-level primitives and showed how to construct CCA2 encryption from these low-level primtives. Examples are constructions of CCA2-secure encryption from the primitives of identity-based encryption [CHK04], tag-based encryption [Kil06], lossy trapdoor functions [PW11], correlated products [RS10], adaptive trapdoor functions [KMO10], and extractable hash proofs [Wee10].

Finally, several works [CS02, HK08, CKS09, HJKS10] construct CCA2-encryption directly from various number-theoretic assumptions.

2 Preliminaries

2.1 CCA2 Security

Definition 2.1 (CCA2 Security) *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let the random variable $\mathsf{CCA2\text{-}Expr}_\beta(\mathcal{E}, A, k)$ where $\beta \in \{0,1\}$, $A = (A_1, A_2)$ are ppt algorithms and $k \in \mathsf{N}$, denote the result of the following probabilistic experiment:*

$\mathsf{CCA2\text{-}Expr}_\beta(\mathcal{E}, A, k)$

- $(pk, sk) \leftarrow\!\!\$\ \mathsf{Gen}(1^k)$
- $(m_0, m_1, state_A) \leftarrow\!\!\$\ A_1^{\mathsf{Dec}(sk,\cdot)}(pk)$
- $y \leftarrow\!\!\$\ \mathsf{Enc}(pk, m_\beta)$
- $D \leftarrow\!\!\$\ A_2^{\mathsf{Dec}(sk,\cdot)}(y, state_A)$

We require that the output of A_1 satisfies $|m_0| = |m_1|$ and that A_2 does not query y to its oracle.

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA2-secure if for any ppt algorithms $A = (A_1, A_2)$ the following two ensembles are computationally indistinguishable:

$$\{\mathsf{CCA2\text{-}Expr}_0(\mathcal{E}, A, k)\}_{k\in\mathsf{N}} \stackrel{c}{\approx} \{\mathsf{CCA2\text{-}Expr}_1(\mathcal{E}, A, k)\}_{k\in\mathsf{N}}.$$

2.2 Plaintext Awareness for Multiple key Setup

We follow [MSS12] for the following definition.

$\mathsf{sPA1}_\ell(E, C, C^*, k)$:

- Let $R[C]$, $R[C^*]$ be randomly chosen bit strings for C and C^*.
- $((pk_i, sk_i))_{i\in[\ell(k)]} \leftarrow\!\!\$\ \mathsf{Gen}(1^k)$
- $st \leftarrow \big((pk_i)_{i\in[\ell(k)]}, R[C]\big)$
- $C^{C^*(st,\cdot)}\big((pk_i)_{i\in[\ell(k)]}\big)$
- Let $Q = \{(q_i = (pk_{j_i}, c_i), m_i)\}$ be the set of queries C made to C^* until it halted and C^*'s responses to them. Return $\wedge_{i=1}^{|Q|}(m_i = \mathsf{Dec}_{sk_{j_i}}(c_i))$.

In the above experiment, C is a ciphertext creator, and C^* is a stateful ppt algorithm called the *extractor* that takes as input the state information st and a ciphertext given by the ciphertext creator C, and will return the decryption of that ciphertext and the updated state st. The state information is initially set to the public key pk and the adversary C's random coins. It gets updated by C^* as C^* answers each query that the adversary C submits. The above experiment returns 1 if all the extractor's answers to queries are the true decryption of those queries under sk. Otherwise, the experiment returns 0.

Definition 2.2 ($\mathsf{sPA1}_\ell$) *Let ℓ be a polynomial. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an asymmetric encryption scheme. Let the ciphertext-creator adversary C and the extractor C^* be ppt algorithms. For $k \in \mathsf{N}$, the* sPA1*-advantage of C relative to C^* is defined as:*

$$Adv^{\mathsf{sPA1}_\ell}(\mathcal{E}, C, C^*) = \Pr[\mathsf{sPA1}(\mathcal{E}, C, C^*, k) = 0]$$

The extractor C^* is a successful $\mathsf{sPA1}_\ell$-extractor for the ciphertext-creator adversary C if for all $k \in \mathsf{N}$, the function $Adv^{\mathsf{sPA1}_\ell}(\mathcal{E}, C, C^*)$ is negligible. The encryption scheme $\mathcal{E}$ is called $\mathsf{sPA1}_\ell$ multi-key secure if for any ppt ciphertext creator there exists a successful $\mathsf{sPA1}_\ell$-extractor.

As shown by [MSS12], both the Damgard Elgamal encryption scheme (DEG) and the lite version of Cramer-Shoup encryption scheme (CS-lite) are $\mathsf{sPA1}_\ell$ secure under a suitable generalization of the DHK1 assumption.

2.3 Weakly Simulatable Encryption Scheme

As in [MSS12], we consider a notion of simulatability similar to the one of Dent [Den06], but where the attacker is not given access to the decryption oracle. If an encryption scheme satisfies this weaker notion of simulatability, we say it is weakly simulatable.

Definition 2.3 (Weakly Simulatable Encryption Scheme) *An asymmetric encryption scheme* (Gen, Enc, Dec) *is weakly simulatable if there exist two poly-time algorithms (f, f^{-1}) where f is deterministic and f^{-1} is probabilistic, such that for all $k \in \mathsf{N}$ there exists the polynomial function $p(\cdot)$ where $l = p(k)$, we have the following correctness properties:*

- *f on inputs of public key pk (in the range of Gen) and a random string $r \in \{0,1\}^l$, returns elements in $\mathcal{C}$, where $\mathcal{C}$ is the set of all possible "ciphertext"-strings that can be submitted to the decryption oracle (notice that $C \in \mathcal{C}$ might not be a valid ciphertext).*
- *f^{-1} on input of a public key pk (in the range of Gen) and an element $C \in \mathcal{C}$ outputs elements of $\{0,1\}^l$.*
- *$f(pk, f^{-1}(pk, C)) = C$ for all $C \in \mathcal{C}$.*

And the following security properties. No polynomial time attacker A has probability better than $1/2 + \mu(k)$ of winning the following experiment, where μ is some negligible function.

- *The challenger generates a random key pair $(pk, sk) \leftarrow_{\$} \mathsf{Gen}(1^k)$, and chooses randomly $b \in \{0,1\}$.*
- *The attacker A executes on the input 1^k and the public key pk outputs $m \in \mathcal{M}$. The challenger sends A the pair $(f^{-1}(pk, c = \mathsf{Enc}(pk, m)), c)$ if $b = 0$, or $(r, f(pk, r))$ for some randomly generated element $r \in \{0,1\}^l$ if $b = 1$. The attacker A terminates by outputting a guess b' for b. A wins if $b = b'$ and its advantage is defined in the usual way.*

Lemma 2.4 *If $\mathcal{E}$ is a weakly simulatable encryption scheme, then $\mathcal{E}$ is CPA-secure.*

[MSS12] show that DEG and CS-lite schemes can both be weakly simulatable when instantiated in proper groups.

2.4 $\mathsf{PA1}^+$–An Extension of Plaintext Awareness

[MSS12] additionally consider an augmented notion of plaintext awareness in which the ciphertext creator has access to an oracle that produces random bits, $\mathsf{PA1}^+$. The extractor receives the answers to any queries generated by the creator, but only at the time these queries are issued. This oracle is meant is to model the fact that the plaintext extractor might not receive all of the random coins used by the ciphertext creator *at the beginning* of the experiment. By introducing this oracle, we require the extractor to work even when it receives the random coins at the same time as the ciphertext creator. This modification has implications when the notion of plaintext awareness is computational. However, in our case, as in [MSS12], we require statistical plaintext awareness, and as argued by [MSS12], allowing access to such an oracle does not affect the $\mathsf{sPA1}_\ell$ security.

Any encryption scheme that is $\mathsf{sPA1}_\ell$ secure is also $\mathsf{sPA1}^+_\ell$ secure.

Definition 2.5 *Define the* $\mathsf{sPA1}^+_\ell$ *experiment in a similar way to the* $\mathsf{sPA1}_\ell$ *experiment. The only difference between the two is that during the* $\mathsf{sPA1}^+_\ell$ *experiment, the ciphertext creator has access to a random oracle* $\mathcal{O}$ *that takes no input, but returns independent uniform random strings upon each access. Any time the creator accesses the oracle, the oracle's response is forwarded to both the creator and the extractor.*

If an encryption scheme would be deemed $\mathsf{sPA1}_\ell$ *secure, when we replace the* $\mathsf{sPA1}_\ell$ *experiment in the definition with the modified* $\mathsf{sPA1}^+_\ell$ *experiment, then the encryption scheme is said to be* $\mathsf{sPA1}^+_\ell$ *secure.*

Lemma 2.6 (Appeared in [MSS12].) *If an encryption scheme* $\mathcal{E}$ *is* $\mathsf{sPA1}_\ell$ *secure, then it is* $\mathsf{sPA1}^+_\ell$ *secure.*

2.5 Strong One-Time Signature Scheme

We follow here the definition of [CDSMW08]. Informally, a strong one-time signature scheme $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{Ver})$ 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 [Lam79, Rom90], and thus from any semantically-secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

3 The Scheme

We present a $\mathsf{CCA2}$-secure encryption scheme $\mathcal{E}_{\mathsf{cca}} = (\mathsf{Gen}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$ from any scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}} = (\mathsf{Gen}_{\mathsf{pa\text{-}cpa}}, \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}, \mathsf{Dec}_{\mathsf{pa\text{-}cpa}})$ which is a plaintext aware, weakly simulatable public key encryption scheme with perfect correctness and any scheme $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{Ver})$, which is a strong one-time signature scheme and any pseudorandom generator prg. See Figure 1.

Theorem 3.1 *Encryption scheme* $\mathcal{E}_{\mathsf{cca}}$*, presented in Figure 1, is* $\mathsf{CCA2}$*-secure under the assumptions that* $\mathcal{E}_{\mathsf{pa\text{-}cpa}} = (\mathsf{Gen}_{\mathsf{pa\text{-}cpa}}, \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}, \mathsf{Dec}_{\mathsf{pa\text{-}cpa}})$ *is a plaintext aware, weakly simulatable public key encryption scheme with perfect correctness, the scheme* $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{Ver})$ *is a strong one-time signature scheme and* prg *is a pseudorandom generator.*

Note that the Damgard Elgamal encryption scheme (DEG) and the lite version of Cramer-Shoup encryption scheme (CS-lite) are plaintext aware, weakly simulatable and have perfect correctness.

Since strong one-time signature schemes and pseudorandom generators can be constructed in a black-box manner from CPA-secure public key encryption (and hence plaintext aware, weakly simulatable public key encryption) we have the following corollary:

Corollary 3.2 *There is a black-box construction of a* $\mathsf{CCA2}$*-secure public key encryption scheme from any plaintext aware, weakly simulatable public key encryption scheme with perfect correctness.*

Encryption Scheme $\mathcal{E}_{\mathsf{cca}} = (\mathsf{Gen}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$

Key Generation $\mathsf{Gen}_{\mathsf{cca}}(1^k)$:

- $[pk_{in_b}, sk_{in_b}]_{b\in\{0,1\}} \leftarrow\!\!\$\ \mathsf{Gen}_{\mathsf{pa\text{-}cpa}}(1^k)$
- $[pk_i^b, sk_i^b]_{b\in\{0,1\},i\in[k]} \leftarrow\!\!\$\ \mathsf{Gen}_{\mathsf{pa\text{-}cpa}}(1^k)$
- $pk \leftarrow ([pk_{in_b}]_{b\in\{0,1\}}, [pk_i^b]_{b\in\{0,1\},i\in[k]})$; $sk \leftarrow ([sk_{in_b}]_{b\in\{0,1\}}, [sk_i^b]_{b\in\{0,1\},i\in[k]})$
- Return (pk, sk)

Encryption $\mathsf{Enc}_{\mathsf{cca}}(pk, m)$**:**

- $(\mathsf{vksig}, \mathsf{sksig}) \leftarrow\!\!\$\ \mathsf{GenSig}(1^k)$
- $r \leftarrow\!\!\$\ \{0,1\}^k$
- $\tilde{s}_0 \leftarrow\!\!\$\ \{0,1\}^\ell$, where $\ell = k + |m|$; $\tilde{s}_1 \leftarrow (r||m) \oplus \tilde{s}_0$
- $CT_{in_0} \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_0)$; $CT_{in_1} \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_1}, \tilde{s}_1)$
- $r_1||\cdots||r_k = \mathsf{prg}(r)$
- For $1 \le i \le k$, $CT_i \leftarrow \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}_i}, CT_{in_0}||CT_{in_1}; r_i)$
- Return $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma = \mathsf{Sign}(\mathsf{sksig}, CT_1||\cdots||CT_k))$

Decryption $\mathsf{Dec}_{\mathsf{cca}}(sk, (CT = CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma))$

- If $\mathsf{Ver}(\mathsf{vksig}, CT, \sigma) = \perp$, output $\perp$.
- Otherwise, $CT_{in_0}||CT_{in_1} \leftarrow \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_1^{\mathsf{vksig}_1}, CT_1)$
- $\tilde{s}_0 \leftarrow \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_0}, CT_{in_0})$
- $\tilde{s}_1 \leftarrow \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_1}, CT_{in_1})$
- $(r||m) \leftarrow \tilde{s}_0 \oplus \tilde{s}_1$
- $(r_1||\cdots||r_k) \leftarrow \mathsf{prg}(r)$
- If for all i, $CT_i = \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}_i}, CT_{in_0}||CT_{in_1}; r_i)$ return m
- Else return $\perp$.

Figure 1: The CCA2-Secure Encryption Scheme $\mathcal{E}_{\mathsf{cca}}$

4 Security Analysis

We begin by defining an experiment which is different than the regular CCA2 experiment, but will be useful in our analsysis of $\mathcal{E}_{\mathsf{cca}}$:

Nested Indistinguishability Experiment for scheme $\mathcal{E}_{\mathsf{cca}}$:

We define the expriment $\mathsf{Nested\text{-}Expr}(\beta, z)$ for $\beta, z \in \{0,1\}$.

For every adversary $A = (A_1, A_2)$ participating in a CCA2 experiment, we consider a corresponding ciphertext creator C_A (described below) and ciphertext extractor C^* (as guaranteed by the security of the encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$), interacting with an oracle $\mathcal{O}$ (described below). Let the random variable $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$, where $\beta, z \in \{0,1\}$ and $k \in \mathsf{N}$, denote the result of the following probabilistic experiment:

$\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$**:**

- C_A receives public keys $[pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[n]}$ from the $\mathsf{sPA1}^+_{2k+2}$ experiment

- C_A chooses $(\mathsf{sksig}^*, \mathsf{vksig}^*) \leftarrow\!\!\$\ \mathsf{GenSig}(1^k; r_{\mathsf{sksig}})$, where r_{sksig} consists of the first k bits of C_A's random tape.
- C_A sets $pk = [pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[n]}$.
- C_A chooses a random tape for A and begins an emulation of A_1 on input pk.
- Whenever C_A receives query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ from A, C_A checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, C_A returns $\perp$. If so, C_A submits CT_i, where i is the first index s.t. $\mathsf{vksig}_i^* \neq \mathsf{vksig}$, to the extractor to obtain $(CT_{in_0}||CT_{in_1})$. If there is no such index, C_A returns $\perp$ and halts. Otherwise, C_A submits CT_{in_0} and CT_{in_1} to the extractor to obtain $\tilde{s}_0, \tilde{s}_1$. C_A computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, C_A returns $\perp$. If so, C_A returns m. Eventually A_1 returns (m_0, m_1, st) and halts. C_A outputs (m_0, m_1).
- C_A queries its oracle $\mathcal{O}$ and $\mathcal{O}$ returns $r_1, \ldots, r_k$ where $(r_1, CT_1^*) = f^{-1}(pk_1^{\mathsf{vksig}_1^*}, CT_1^*), \ldots, (r_k, CT_k^*) = f^{-1}(pk_1^{\mathsf{vksig}_k^*}, CT_k^*)$ and where $CT_1^*, \ldots, CT_k^*$ are computed in the following way:
  1. $r \leftarrow\!\!\$\ \{0,1\}^k, r_1, \ldots, r_n \leftarrow \mathsf{prg}(r)$.
  2. $(\mathsf{sksig}, \mathsf{vksig}) \leftarrow\!\!\$\ \mathsf{GenSig}(1^k)$
  3. $\tilde{s}_0 \leftarrow\!\!\$\ \{0,1\}^\ell$
  4. If $z = 0$ then $\tilde{s}_1 \leftarrow\!\!\$\ \{0,1\}^\ell$.
  5. Else if $z = 1$ then $\tilde{s}_1 \leftarrow (r||m_\beta) \oplus \tilde{s}_0$.
  6. $CT_{in_0}^* \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_0)$; $CT_{in_1}^* \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_1}, \tilde{s}_1)$
  7. For $1 \leq i \leq k$, $CT_i^* \leftarrow \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}_i^*}, CT_{in_0}^*||CT_{in_1}^*; r_i)$
- C_A computes $CT_i^* = f(x_i)$ for each i and the signature σ^*. C_A returns $CT^* = (CT_1^*||\cdots||CT_k^*, \mathsf{vksig}^*, \sigma^*)$ to A
- Whenever C_A receives query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ from A, C_A checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, C_A returns $\perp$. If so, C_A submits CT_i, where i is the first index s.t. $\mathsf{vksig}_i \neq \mathsf{vksig}_i^*$, to the extractor to obtain $(CT_{in_0}||CT_{in_1})$. If there is no such index, C_A returns $\perp$ and halts. Otherwise, C_A submits CT_{in_0} and CT_{in_1} to the extractor to obtain $\tilde{s}_0, \tilde{s}_1$. C_A computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, C_A returns $\perp$. If so, C_A returns m. Eventually A_2 outputs D and halts.

We require that the output of A_1 satisfies $|m_0| = |m_1|$ and that A_2 does not query CT^* to its oracle.

Definition 4.1 (Nested Indistinguishability) *We say that* $\mathcal{E}_{\mathsf{cca}} = (\mathsf{Gen}_{\mathsf{cca}}, \mathsf{Enc}_{\mathsf{cca}}, \mathsf{Dec}_{\mathsf{cca}})$ *is* nested-indistinguishable *if for any ppt algorithms* $A = (A_1, A_2)$ *and for* $\beta \in \{0,1\}$ *the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{Nested\text{-}Expr}_{\beta,0}(\mathcal{E}_{\mathsf{cca}}, A, k)\right\}_{k\in\mathsf{N}} \overset{c}{\approx} \left\{\mathsf{Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)\right\}_{k\in\mathsf{N}}.$$

Consider the following event:

Definition 4.2 (The Bad Extraction Event) *We say that a* bad extraction event *has occurred during an execution of the nested indistinguishability experiment if at some point* A *submits a decryption query* $CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma)$ *such that one of the following occurs:*

- $C^*(st, CT_i) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_i^{\mathsf{vksig}_i}, CT_i)$ *where* i *is the first index such that* $\mathsf{vksig}_i^* \neq \mathsf{vksig}_i$.
- $C^*(st, CT_{in_0}) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_0}, CT_{in_0})$

- $C^*(st, CT_{in_1}) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_1}, CT_{in_1})$

Definition 4.3 (The Forging Signature Event) *We say that a* forging signature event *has occurred during an execution of the nested indistinguishability experiment if at some point A submits a decryption query $(CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma))$ such that $\mathsf{vksig} = \mathsf{vksig}^*$ and $\mathsf{Ver}(\mathsf{vksig}, CT, \sigma) = 1$.*

Our main theorem, Theorem 3.1, is immediately implied by the following two lemmas:

Lemma 4.4 *Assume that the scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}} = (\mathsf{Gen}_{\mathsf{pa\text{-}cpa}}, \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}, \mathsf{Dec}_{\mathsf{pa\text{-}cpa}})$ is a plaintext aware, weakly simulatable public key encryption scheme with perfect correctness. Then encryption scheme $\mathcal{E}_{\mathsf{cca}}$ is nested-indistinguishable.*

Lemma 4.5 *Assume that the scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}} = (\mathsf{Gen}_{\mathsf{pa\text{-}cpa}}, \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}, \mathsf{Dec}_{\mathsf{pa\text{-}cpa}})$ is a plaintext aware, weakly simulatable public key encryption scheme with perfect correctness, the scheme $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{Ver})$ is a strong one-time signature scheme and prg is a pseudorandom generator. Then for $\beta \in \{0,1\}$ and for every ppt adversary A:*

$$\{\mathsf{Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)\}_{k\in\mathsf{N}} \overset{s}{\approx} \{\mathsf{CCA2\text{-}Expr}_{\beta}(\mathcal{E}_{\mathsf{cca}}, A, k)\}_{k\in\mathsf{N}}$$

Lemma 4.4 follows by a straightforward reduction to semantic security of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$. Lemma 4.5 follows in a straightforward manner from the fact that Bad Extraction Event and Forging Signature Event occur with at most negligible probability when $z = 1$ along with the perfect correctness of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$.

Thus, in what follows, we focus our attention on proving that Bad Extraction Event and Forging Signature Event occur with at most negligible probability when $z = 1$. To show this we proceed in the following way:

- In Section 4.1 we prove that Bad Extraction Event occurs with negligible probability in the Nested Indistinguishability Experiment when $z = 0$.
- In Section 4.2 we use the fact that Bad Extraction Event occurs with negligible probability in the Nested Indistinguishability Experiment when $z = 0$ to prove that Bad Extraction Event also occurs with negligible probability in the Nested Indistinguishability Experiment when $z = 1$.
- In Section 4.3 we use the fact that Bad Extraction event occurs with negligible probability in the Nested Indistinguishability Experiment when $z = 1$ to prove that Forging Signature Event occurs with negligible probability in the Nested Indistinguishability Experiment when $z = 1$.

4.1 Bad Extraction Event when $z = 0$

In this section we prove the following lemma:

Lemma 4.6 *Bad Extraction Event occurs with negligible probability when $z = 0$.*

We proceed by considering a sequence of hybrids:

Hybrid H_0: Proceeds exactly as the nested indistinguishability game for the case where $z = 0$.

Hybrid H_1: Proceeds exactly like H_0 except that fresh randomness r_i is used to encrypt each $CT_i^* = \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}_i}, CT_{in_0}^*||CT_{in_1}^*; r_i)$, instead of the output of the prg.

Claim 4.7 The probability of a Bad Extraction Event in H_1 and H_0 differs by a negligible amount.

This follows in a straightforward manner from the security of the prg.

Hybrid H_2: Proceeds exactly like H_1 except the oracle $\mathcal{O}$ returns uniformly random $r_1, \ldots, r_k$.

Claim 4.8 The probability of Bad Extraction Event in H_2 is negligible.

The claim follows due to the fact that the view of C_A in the nested indistinguishability experiment in Hybrid H_2 is identical to the view of C_A in the $\mathsf{sPA1}^{+}_{2k+2}$ experiment (since in H_2 the oracle $\mathcal{O}$ simply returns uniformly random coins $r_1, \ldots, r_k$, as does the oracle in the $\mathsf{sPA1}^{*}_{2k+2}$ experiment). Thus, by the $\mathsf{sPA1}^{+}_{2k+2}$-secuirty of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$, C^* is guaranteed to return the same value as $\mathsf{Dec}_{\mathsf{pa\text{-}cpa}}$ on all ciphertexts submitted by C_A with all but negligible probability.

Claim 4.9 The probability of a Bad Extraction Event in H_1 and H_2 differs by a negligible amount.

Proof: Assume towards contradiction that there exists a ppt adversary A such that a Bad Extraction Event in H_1 and H_2 differs by a non-negligible amount $p = p(k)$ when interacting with A, C_A, C^*. We present a ppt adversary B breaking the weak simulatability of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$.

B participates in an external experiment where B plays the security game of the weakly simulatable encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$ while internally interacting with the adversary A and the corresponding ciphertext creator C_A and extractor C^* in the following way:

- B receives $\hat{pk}_1, \ldots, \hat{pk}_k$ from the external simulatability security experiment.
- B chooses a random tape r_{C_A} for the ciphertext creator C_A.
- B computes $(\mathsf{sksig}^*, \mathsf{vksig}^*) \leftarrow\!\!\$\ \mathsf{GenSig}(1^k; r_{\mathsf{sksig}})$, where r_{sksig} consists of the first k bits of r_{C_A}.
- B generates public key, secret key pairs $[pk_{in_b}, sk_{in_b}]_{b\in\{0,1\}}, \{pk_i^{1-\mathsf{vksig}_i^*}, sk_i^{1-\mathsf{vksig}_i^*}\}_{i\in[k]}$ and for $i \in [k]$ sets $pk_i^{\mathsf{vksig}_i^*} = \hat{pk}_i$.
- B instantiates C_A with random tape r_{C_A} on input $[pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[k]}$.
- Eventually C_A outputs (m_0, m_1). At this point, B plays the part of the oracle $\mathcal{O}$ and does the following:
  1. Choose $\tilde{s}_0, \tilde{s}_1 \leftarrow\!\!\$\ \{0,1\}^\ell$
  2. Compute $CT^*_{in_0} \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_0)$; $CT^*_{in_1} \leftarrow\!\!\$\ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_1}, \tilde{s}_1)$
  3. Submit $CT^*_{in_0}||CT^*_{in_1}$ to its external challenger.
  4. Receives $(r_1, CT^*_1), \ldots, (r_k, CT^*_k)$ from its external challenger, where for each i, $r_i = f^{-1}(pk_i, c = \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i, CT^*_{in_0}||CT^*_{in_1}))$ if $b = 0$ or $(r_i, f(pk_i, r_i))$ for randomly generated r_i if $b = 1$.

  B forwards $r_1, \ldots, r_k$ to C_A on behalf of oracle $\mathcal{O}$ and continues the emulation of C_A.
- If at any point during the emulation, Bad Extraction Event occurs (which B can check by decrypting using $[sk_{in_b}]_{b\in\{0,1\}}, \{sk_i^{1-\mathsf{vksig}_i^*}\}_{i\in[k]}$), B aborts and outputs 1.
- Otherwise, B outputs 0.

Note that for $\beta \in \{0,1\}$, B perfectly simulates C_A's view in Hybrid H_1 when $b = 0$ by returning $r_1, \ldots, r_k$ where for each i, $r_i = f^{-1}(pk_i, c = \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i, CT^*_{in_0}||CT^*_{in_1}))$ and perfectly simulates C_A's view in Hybrid H_2 by returning $r_1, \ldots, r_k$ where for each i, r_i is chosen uniformly at random. Thus, B outputs 1 in the case that $b = 0$ in the external experiment with probability p_1 and B outputs 1 in the case that $b = 1$ in the external experiment with probability p_2 where $p_1 - p_2 > p$. Since by hypothesis, p is non-negligible, we have that B breaks the security of the weakly simulatable encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$. ∎

Lemma 4.6 follows immediately from Claims 4.7, 4.8 and 4.9.

4.2 Bad Extraction Event when $z = 1$

In this section we prove the following lemma:

Lemma 4.10 *Bad Extraction Event occurs with negligible probability when $z = 1$.*

To aid in our analysis, we define a second experiment "Modified Nested Indistinguishability" and a second Bad Extraction Event, "Modified Bad Extraction Event". The Modified Nested Indistinguishability experiment is *identical* to the Nested Indistinguishability experiment except that an additional random variable $b \leftarrow\$ \{0,1\}$ is chosen at the very beginning of the experiment The Modified Bad Extraction Event will then depend on the value of b chosen during the experiment. Details follow.

For every adversary $A = (A_1, A_2)$ participating in a CCA2 experiment, we consider a corresponding ciphertext creator C_A (described below) and ciphertext extractor C^* (as guaranteed by the security of the encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$), interacting with the oracle $\mathcal{O}$ (described below). Let the random variable $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$, where $\beta, z \in \{0,1\}$ and $k \in \mathsf{N}$, denote the result of the following probabilistic experiment:

$\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$**:**

- $b \leftarrow\$ \{0,1\}$
- C_A receives public keys $[pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[n]}$ from the $\mathsf{sPA1}^+_{2k+2}$ experiment
- C_A chooses $(\mathsf{sksig}^*, \mathsf{vksig}^*) \leftarrow\$ \mathsf{GenSig}(1^k; r_{\mathsf{sksig}})$, where r_{sksig} are the first k bits of C_A's random tape.
- C_A sets $pk = [pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[n]}$.
- C_A chooses a random tape for A and begins an emulation of A_1 on the input of pk.
- Whenever C_A receives query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ from A, C_A checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, C_A returns $\perp$. If so, C_A submits CT_i, where i is the first index s.t. $\mathsf{vksig}_i^* \neq \mathsf{vksig}$, to the extractor to obtain $(CT_{in_0}||CT_{in_1})$. If there is no such index, C_A returns $\perp$ and halts. Otherwise, C_A submits CT_{in_0} and CT_{in_1} to the extractor to obtain $\tilde{s}_0, \tilde{s}_1$. C_A computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, C_A returns $\perp$. If so, C_A returns m. Eventually A_1 returns (m_0, m_1, st) and halts. C_A outputs (m_0, m_1).
- C_A queries its oracle $\mathcal{O}$ and $\mathcal{O}$ returns $r_1, \ldots, r_k$ where $(r_1, CT_1^*) = f^{-1}(pk_1^{\mathsf{vksig}_1^*}, CT_1^*), \ldots, (r_k, CT_k^*) = f^{-1}(pk_1^{\mathsf{vksig}_k^*}, CT_k^*)$ and where $CT_1^*, \ldots, CT_k^*$ are computed in the following way:
  1. $r \leftarrow\$ \{0,1\}^k, r_1, \ldots, r_n \leftarrow \mathsf{prg}(r)$.
  2. $(\mathsf{sksig}, \mathsf{vksig}) \leftarrow\$ \mathsf{GenSig}(1^k)$
  3. $\tilde{s}_0 \leftarrow\$ \{0,1\}^\ell$
  4. If $z = 0$ then $\tilde{s}_1 \leftarrow\$ \{0,1\}^\ell$.
  5. Else if $z = 1$ then $\tilde{s}_1 \leftarrow (r||m_\beta) \oplus \tilde{s}_0$.
  6. $CT_{in_0}^* \leftarrow\$ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_0)$; $CT_{in_1}^* \leftarrow\$ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_1}, \tilde{s}_1)$
  7. For $1 \le i \le k$, $CT_i^* \leftarrow \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}}, CT_{in_0}^*||CT_{in_1}^*; r_i)$
- C_A computes $CT_i^* = f(x_i)$ for each i and the signature σ^*. C_A returns $CT^* = (CT_1^*||\cdots||CT_k^*, \mathsf{vksig}^*, \sigma^*)$ to A
- Whenever C_A receives query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ from A, C_A checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, C_A returns $\perp$. If so, C_A submits CT_i, where i is the first index s.t. $\mathsf{vksig}_i \neq \mathsf{vksig}_i^*$ to the extractor to obtain $(CT_{in_0}||CT_{in_1})$. If there is no such index, C_A returns $\perp$ and halts. Otherwise, C_A submits CT_{in_0} and CT_{in_1} to the extractor to obtain $\tilde{s}_0, \tilde{s}_1$. C_A computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, C_A returns $\perp$. If so, C_A returns m. Eventually A_2 outputs D and halts.

Definition 4.11 (The Modified Bad Extraction Event) *We say that a* modified bad extraction event *has occurred during an execution of the nested indistinguishability experiment if at some point A submits a decryption query $CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma)$ such that one of the following occurs:*

- $C^*(st, CT_i) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_i^{\mathsf{vksig}_i}, CT_i)$ *where i is the first index such that* $\mathsf{vksig}_i^* \neq \mathsf{vksig}_i$.

- $C^*(st, CT_{in_b}) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_b}, CT_{in_b})$

Claim 4.12 For every ppt adversary $A = (A_1, A_2)$ and for $\beta \in \{0, 1\}$, Modified Bad Extraction Event occurs in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ with negligible probability when $z = 0$.

This follows immediately from the fact that for every ppt adversary $A = (A_1, A_2)$, Bad Extraction event occurs in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ with negligible probability when $z = 0$.

Claim 4.13 If for some ppt adversary $A = (A_1, A_2)$ we have that Bad Extraction Event occurs with probability p_1 in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ when $z = 1$ then Modified Bad Extraction Event occurs with probability at least $p_1/2$ in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ when $z = 1$.

Proof: Let A be a ppt adversary such that Bad Extraction Event occurs with probability p_1 in the experiment $\mathsf{Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$. Let event E be the event that for some query, $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$, one of the following occurs:

$$C^*(st, CT_i) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_i^{\mathsf{vksig}_i}, CT_i) \tag{1}$$

where i is the first index such that $\mathsf{vksig}_i^* \neq \mathsf{vksig}_i$.

OR

$$C^*(st, CT_{in_0}) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_0}, CT_{in_0}) \tag{2}$$

OR

$$C^*(st, CT_{in_1}) \neq \mathsf{Dec}_{\mathsf{pa\text{-}cpa}}(sk_{in_1}, CT_{in_1}) \tag{3}$$

and this is the *first such query* made by A during the experiment. Note that the probability that event E occurs in $\mathsf{Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ and the probability that E occurs in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ is p_1.

We consider an experiment, $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$, identical to the Modified Nested Indistinguishability experiment except the value of b is chosen "on the fly" at the first point when event E occurs. It is straightforward to see that the probability of event E in $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ is also p_1 (the same as the probability of E in the experiment $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$).

Now, if event E was triggered by a query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ in $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ such that (1) occurs, then modified bad extraction event also occurs. Alternatively, if event E was triggered by a query $CT = (CT_1||\cdots||CT_k, \mathsf{vksig}, \sigma)$ in $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ such that (2) or (3) occurs, then modified bad extraction event will occur with probability exactly $1/2$. Thus, modified bad extraction event occurs in $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ with probability at least $p_1/2$. Since the view of C_A is identical in $\mathsf{Modified\text{-}Nested\text{-}Expr}'_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ and in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ we must also have that modified bad extraction event occurs in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ with probability at least $p_1/2$. ∎

Claim 4.14 The probability of a Modified Bad Extraction Event when $z = 0$ and $z = 1$ differs by a negligible amount.

Proof: Assume towards contradiction that there is a ppt adversary A such that the probability of a Modified Bad Extraction Event in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{beta,0}(\mathcal{E}_{\mathsf{cca}}, A, k)$ is $p_0 = p_0(k)$, the probability of a Modified Bad Extraction Event in $\mathsf{Modified\text{-}Nested\text{-}Expr}_{beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ is $p_1 = p_1(k)$ and $p(k) = p_1(k) - p_0(k)$ is non-negligible. We present a ppt adversary B that uses A to break the semantic security of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$.

B participates in an external semantic security experiment for encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$ while internally emulating a run of $\mathsf{Modified\text{-}Nested\text{-}Expr}$ with C_A, A and playing the part of the oracle $\mathcal{O}$. More specifically, B receives a public key $pk_{\mathsf{pa\text{-}cpa}}$ from the semantic security experiment for the encryption scheme $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$ and does the following:

- B chooses $b \leftarrow_{\$} \{0,1\}$.
- B sets $pk_{in_{1-b}} = pk_{\mathsf{pa\text{-}cpa}}$
- B chooses $(pk_{in_b}, sk_{in_b}) \leftarrow_{\$} \mathsf{Gen}_{\mathsf{pa\text{-}cpa}}(1^k)$ and $[pk_i^b, sk_i^b]_{b\in\{0,1\},i\in[k]} \leftarrow_{\$} \mathsf{Gen}_{\mathsf{pa\text{-}cpa}}(1^k)$.
- B chooses a random tape r_{C_A} for C_A and begins an emulation of C_A with input $([pk_{in_b}]_{b\in\{0,1\}}, [pk_i^b]_{b\in\{0,1\},i\in[k]})$.
- At some point C_A outputs m_0, m_1. At this point, B, playing the part of the oracle $\mathcal{O}$, returns $r_1, \ldots, r_k$ where $(r_1, CT_1^*) = f^{-1}(pk_1^{\mathsf{vksig}_1^*}, CT_1^*), \ldots, (r_k, CT_k^*) = f^{-1}(pk_1^{\mathsf{vksig}_k^*}, CT_k^*)$ and $CT_1^*, \ldots, CT_k^*$ are computed in the following way:
  - $r \leftarrow_{\$} \{0,1\}^k, r_1, \ldots, r_n \leftarrow \mathsf{prg}(r)$.
  - $(\mathsf{sksig}, \mathsf{vksig}) \leftarrow_{\$} \mathsf{GenSig}(1^k)$
  - $\tilde{s}_b \leftarrow_{\$} \{0,1\}^\ell$
  - Choose $\tilde{s}_{1-b}^0 \leftarrow_{\$} \{0,1\}^\ell$. and set $\tilde{s}_{1-b}^1 \leftarrow (r||m_\beta) \oplus \tilde{s}_0$.
  - B returns $M_0 = \tilde{s}_{1-b}^0, M_1 = \tilde{s}_{1-b}^1$ to its external challenger and receives ciphertext $CT_{\mathsf{pa\text{-}cpa}}$ in return.
  - B sets $CT_{in_b}^* \leftarrow_{\$} \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_b)$ and sets $CT_{in_{1-b}}^* = CT_{\mathsf{pa\text{-}cpa}}$.
  - For $1 \le i \le k$, $CT_i^* \leftarrow \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}}, CT_{in_0}^* || CT_{in_1}^*; r_i)$
- B continues the emulation of C_A, A.
- If the event Modified Bad Extraction Event occurs, B aborts and outputs 1.
- Otherwise, B outputs 0.

Note that for $\beta \in \{0,1\}$, B perfectly simulates C_A's view in the experiment $\mathsf{Modified\text{-}Nested\text{-}Expr}(\beta, 0)$. Thus, B outputs 1 in the case that the external challenger returned an encryption of M_1 with probability p_1 and B outputs 1 in the case that the external challenger returned an encryption of M_0 with probability p_2 where $p_1 - p_2 > p$. Since by hypothesis, p is non-negligible, we have that B breaks the semantic security of $\mathcal{E}_{\mathsf{pa\text{-}cpa}}$. ∎

Together, Claims 4.12, 4.13 and 4.14 immediately imply Lemma 4.10.

4.3 Forging Signature Event when $z = 1$

In this section, we prove the following lemma:

Lemma 4.15 *Forging Signature Event occurs with negligible probability when $z = 1$.*

Proof: Assume towards contradiction that there exists a ppt adversary A such that Forging Signature Event occurs in $\mathsf{Nested\text{-}Expr}$ with non-negligible probability, $p = p(k)$. We use A to construct a ppt adversary B breaking the security of the signature scheme. B does the following:

- B receives vksig^* from its external challenger.
- B generates public, secret keys $[pk_{in_b}, sk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b, sk_i^b\}_{b\in\{0,1\},i\in[n]}$ from the $\mathsf{sPA1}^+_{2k+2}$ experiment and sets $pk = [pk_{in_b}]_{b\in\{0,1\}}, \{pk_i^b\}_{b\in\{0,1\},i\in[n]}$
- B chooses random coins for A and begins an emulation of A_1 on input pk.
- Whenever B receives query $CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma)$ from A_1, B checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, B returns $\perp$. If so, B decrypts CT_i, where i is the first index s.t. $\mathsf{vksig}_i \neq \mathsf{vksig}_i^*$ using $sk_i^{\mathsf{vksig}_i}$ to obtain $(CT_{in_0}||CT_{in_1})$. B decrypts CT_{in_0} and CT_{in_1} using sk_{in_0}, sk_{in_1} to obtain $\tilde{s}_0, \tilde{s}_1$. B computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, B returns $\perp$. If so, B returns m. Eventually A_1 returns (m_0, m_1, st) and halts.
- B does the following:
  1. $r \leftarrow\$ \{0,1\}^k, r_1, \ldots, r_n \leftarrow \mathsf{prg}(r)$.
  2. $(\mathsf{sksig}, \mathsf{vksig}) \leftarrow\$ \mathsf{GenSig}(1^k)$
  3. $\tilde{s}_0 \leftarrow\$ \{0,1\}^\ell$, $\tilde{s}_1 \leftarrow (r||m_\beta) \oplus \tilde{s}_0$.
  4. $CT^*_{in_0} \leftarrow\$ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_0}, \tilde{s}_0)$; $CT^*_{in_1} \leftarrow\$ \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_{in_1}, \tilde{s}_1)$
  5. For $1 \leq i \leq k$, $CT^*_i \leftarrow \mathsf{Enc}_{\mathsf{pa\text{-}cpa}}(pk_i^{\mathsf{vksig}}, CT^*_{in_0}||CT^*_{in_1}; r_i)$
  6. B requests a signature for message $CT^*_1||\cdots||CT^*_k$ and receives $(CT^*_1||\cdots||CT^*_k, \mathsf{vksig}^*, \sigma^*)$ from its oracle.
  7. B returns $CT^* = (CT^*_1||\cdots||CT^*_k, \mathsf{vksig}^*, \sigma^*)$ to A
- Whenever B receives query $CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma)$ from A, B checks $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_k, \sigma) = 1$. If not, B returns $\perp$. If so, B decrypts CT_i, where i is the first index such that $\mathsf{vksig}_i \neq \mathsf{vksig}_i^*$ using $sk_i^{\mathsf{vksig}_i}$ to obtain $(CT_{in_0}||CT_{in_1})$. B decrypts CT_{in_0} and CT_{in_1} using sk_{in_0}, sk_{in_1} to obtain $\tilde{s}_0, \tilde{s}_1$. B computes $r||m = \tilde{s}_0 \oplus \tilde{s}_1$ and checks that $CT_1, \ldots, CT_n$ were computed correctly. If not, B returns $\perp$. If so, B returns m. Eventually A_2 outputs D and halts.
- If at any point during the execution, A submits a query $CT = (CT_1||\cdots||CT_n, \mathsf{vksig}, \sigma)$ such that $\mathsf{vksig} = \mathsf{vksig}^*$ and $\mathsf{Ver}(\mathsf{vksig}, CT_1||\cdots||CT_n, \sigma) = 1$, (i.e. Forging Signature Event occurs), then B aborts the experiment and submits σ to its challenge oracle.

We claim that the probability that Forging Signature Event occurs in B's simulation differs by at most a negligible amount from the probability that Forging Signature Event occurs in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$. Thus, if Forging Signature Event occurs with non-negligible probability $p = p(k)$ in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ then Forging Signature Event occurs with non-negligible probability $p/2$ during B's emulation and contradicts the security of the signature scheme $(\mathsf{GenSig}, \mathsf{Sign}, \mathsf{Ver})$.

To see that the probabilities differ by a negligible amount, note that the response returned to A in B's emulation is identical to the response returned to A in $\mathsf{Nested\text{-}Expr}_{\beta,1}(\mathcal{E}_{\mathsf{cca}}, A, k)$ unless Bad Extraction Event occurs. Since we have by Lemma 4.10 that Bad Extraction Event occurs with at most negligible probability in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$ when $z = 1$, this immediately implies that the probability that Forging Signature Event occurs in B's simulation differs by at most a negligible amount from the probability that Forging Signature Event occurs in $\mathsf{Nested\text{-}Expr}_{\beta,z}(\mathcal{E}_{\mathsf{cca}}, A, k)$. ∎

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *ITCS*, pages 326–349, 2012.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.

[BP04a] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *CRYPTO*, pages 273–289, 2004.

[BP04b] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In *ASIACRYPT*, pages 48–62, 2004.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111, 1994.

[CDSMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In *TCC*, pages 427–444, 2008.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[CKS09] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. *J. Cryptology*, 22(4):470–504, 2009.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

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

[Den06] Alexander W. Dent. The cramer-shoup encryption scheme is plaintext aware in the standard model. In *EUROCRYPT*, pages 289–307, 2006.

[DN00] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO*, pages 432–450, 2000.

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

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

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and cca security for public key encryption. In *TCC*, pages 434–455, 2007.

[HJKS10] Kristiyan Haralambiev, Tibor Jager, Eike Kiltz, and Victor Shoup. Simple and efficient public-key encryption from computational diffie-hellman in the standard model. In *Public Key Cryptography*, pages 1–18, 2010.

[HK08] Goichiro Hanaoka and Kaoru Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational diffie-hellman assumption. In *ASIACRYPT*, pages 308–325, 2008.

[HLM03] Jonathan Herzog, Moses Liskov, and Silvio Micali. Plaintext awareness via key registration. In *CRYPTO*, pages 548–564, 2003.

[HLW12] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting dangerous queries: A new approach for chosen ciphertext security. In *EUROCRYPT*, pages 663–681, 2012.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *CRYPTO*, pages 408–423, 1998.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *EUROCRYPT*, pages 673–692, 2010.

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

[MS09] Steven Myers and Abhi Shelat. Bit encryption is complete. In *FOCS*, pages 607–616, 2009.

[MSS12] Steven Myers, Mona Sergi, and Abhi Shelat. Blackbox construction of a more than non-malleable cca1 encryption scheme from plaintext awareness. In *SCN*, pages 149–165, 2012.

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

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM J. Comput.*, 40(6):1803–1844, 2011.

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

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

[Wee10] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *CRYPTO*, pages 314–332, 2010.