Bit encryption is complete

Steven Myers
School of Informatics
Indiana University
samyers@indiana.edu

abhi shelat*
Computer Science Dept.
University of Virginia
abhi@virginia.edu

Abstract— Under CPA and CCA1 attacks, a secure bit encryption scheme can be applied bit-by-bit to construct a secure many-bit encryption scheme. The same construction fails, however, under a CCA2 attack. In fact, since the notion of CCA2 security was introduced by Rackoff and Simon [21], it has been an open question to determine whether single bit CCA2 secure encryption implies the existence of many-bit CCA2 security. We positively resolve this long-standing question and establish that bit encryption is complete for CPA, CCA1, and CCA2 notions.

Our construction is black-box, and thus requires novel techniques to avoid known impossibility results concerning trapdoor predicates [10]. To the best of our knowledge, our work is also the first example of a non-shielding reduction (introduced in [9]) in the standard (i.e., not random-oracle) model.

Keywords-Chosen-ciphertext secure public key encryption

1. INTRODUCTION

A fundamental research program in cryptography is to classify the minimal assumptions that are sufficient to build secure primitives. We investigate this question for the case of encryption schemes secure against a full chosen-ciphertext (CCA2) attack. Prior results—some quite recent—show that CCA2-secure encryption can be constructed from enhanced trapdoor permutations [7], [24], lossy trapdoor functions [20], non-correlateable functions [23], and of course specific number theoretic assumptions such as decisional Diffie-Hellman [5]. Despite this recent progress, a seemingly straightforward question has remained open: given a CCA2-secure encryption primitive that encrypts only one bit messages, is is possible to construct a CCA2-secure scheme that encrypts longer messages? We show that one-bit primitives are in fact necessary and sufficient, i.e., that one-bit encryption is complete for CCA2 encryption.

Theorem 1. *{CPA,CCA}-secure encryption schemes exist iff 1-bit {CPA,CCA}-secure encryption exists.*

One direction of this theorem follows by inspection; the non-trivial direction ultimately follows from a new line of reasoning in Theorem 4. For the weaker cases of CPA and CCA1-secure schemes, the completeness of one-bit encryption for CPA was established almost immediately in the seminal encryption paper by Goldwasser and Micali [13], and for CCA1 by Naor and Yung [17]. The solution is to encrypt the long message bit-by-bit. A simple hybrid argument proves the security of such constructions. In both cases, it is essential that the adversary does not have access to a decryption oracle after receiving the challenge ciphertext. Removing this constraint as per the case in CCA2 security totally breaks the bit-by-bit scheme. Recall that in a CCA2 attack, the adversary's decryption oracle responds to all queries except the challenge ciphertext. Therefore, an adversary can easily reorder the bit-by-bit encryption to form a new encryption, submit this to its decryption oracle, and use the oracle's response to recover the challenge plaintext.

One may imagine several ad-hoc mechanisms to prevent this type of attack. However, the adversary has another strategy that is more cumbersome to defeat: the adversary can "*quote*" a single-bit ciphertext from the long challenge ciphertext in a long ciphertext of its own making. By using the decryption oracle's output on this new cut-and-pasted ciphertext, the adversary can eventually learn the plaintext bit hidden by the single-bit ciphertext. By repeatedly applying this technique, the adversary can then decode all of the single-bit ciphertexts in the long challenge and eventually retrieve the underlying many-bit plaintext. We call such attacks *quoting attacks*. It is not immediately obvious how to prevent quoting attacks when using an encryption scheme that only handles one-bit messages. The crux of our technical contribution is to prevent this type of attack.

We note that in the Random Oracle Model it is known that single-bit encryption implies multi-bit encryption (e.g., [8]), but the foundational deficiencies of the Random Oracle model are well known [2], [12].

1.1. Background and Prior Work

There are three common definitions for security of an encryption scheme: i) Semantic or CPA Security introduced by Goldwasser and Micali [13], ii) CCA1 Security introduced by Naor and Yung [17], and iii) CCA2 security introduced by Rackoff and Simon [21]. Rackoff and Simon did not present a traditional encryption scheme that satisfied their notion, but subsequently, Dolev, Dwork, and Naor [7] present one that relies on the existence of trapdoor permutations. Dolev et al. also introduce the notion of non-malleability of encryption schemes and show that CCA2-security and non-malleable CCA2-security can coincide. Cramer and Shoup [5] present

*Research supported by NSF award CNS-0854811.

a practical CCA2-encryption scheme based on the specific Decisional Diffie-Hellman assumption. They later generalize their theory and show that smooth projective hash functions suffice [6]. Sahai [24] shows that ideas presented by Naor and Yung [17] can be applied to build CCA2 schemes, but this construction uses simulation sound NIZK proofs which—as far as we know—also require trapdoor permutations. Lindell [16] simplifies the construction of Sahai, but still requires trapdoor permutations. Recently, Peikert and Waters [20] show that lossy trapdoor functions suffice to construct CCA2 encryption and can be instantiated under lattice-based assumptions. Rosen and Segev [23] extend the idea to show how to use trapdoor functions secure under correlated products. Contemporaneously, Wee [26] shows that a CCA2 encryption scheme for $\omega(\log k)$ bits can be used to construct a many-bit CCA2 scheme. To the best of our knowledge, our result establishes the first *necessary and sufficient* condition for the existence of many-bit CCA2 encryption.

1.2. Overview of our techniques

As mentioned above, one key challenge we face with using a one-bit scheme to construct a many bit CCA2 scheme is the prevention of quoting attacks; i.e., when the adversary submits decryption queries that contain part of the challenge ciphertext.

To be sure, suppose that the adversary A never *quotes* the challenge ciphertext. In this case, the security of the many bit scheme can be directly reduced to the security of the one-bit CCA2 scheme: simulate the one-bit security experiment for adversary A and use the one-bit decryption oracle directly to answer all of A's decryption queries. The oracle will answer all of A's queries because by assumption, all of them will differ from the one-bit challenge ciphertext. In fact, as a first step in §3, we formalize this notion of "unquoted CCA2" security as UCCA security and show that one-bit CCA2 security implies many-bit UCCA security.

What this suggests is that any successful attack to a proposed many-bit construction must involve a quoted query. Thus, our goal is to construct a scheme which prevents quoting. One simple approach is to make the ciphertext "self-authenticating" by encoding the random coins used to produce the ciphertext as part of the ciphertext (without, of course, breaking its security). The decryption algorithm can then decrypt the many-bit ciphertext and re-encrypt the appropriate single-bits with the encoded random bits to verify the ciphertext's correctness. This approach would seem to prevent single ciphertexts from being quoted from the challenge ciphertext because the adversary cannot determine the appropriate random bits to encode.

One problem with this simple approach arises from a result by Gertner, Malkin, and Reingold [10] who show that a black-box construction of a poly-to-one trapdoor function from a CCA2 secure one-bit encryption primitive is impossible.[1] As a result, this simple approach cannot hope to recover all of the random bits used to form the ciphertext, since doing so would seem to create a poly-to-one trapdoor function for many natural constructions.

These observations lead us to our first candidate construction which is based on nested encryption, i.e. it is an encryption of an encryption. The inner-layer, denoted $\alpha = \mathsf{enc}(\kappa)$, consists of an encryption of a pseudo-random function key κ. The bits of the ciphertext α are then encrypted in a bit-by-bit fashion using the 1-bit CCA2-secure scheme. The random tape used in this bit-by-bit process is generated by using a pseudo-random function (PRFG) keyed on κ to produce a pseudo-random string of appropriate length (i.e., $F_\kappa(1)||F_\kappa(2)||\ldots$).[2] These outer ciphertexts are denoted as $\beta = \beta_1, \beta_2, \ldots, \beta_{|\alpha|}$. (Notice that the random coins used to encrypt the inner-layer are not recovered so as to avoid the noted impossibility result.) To encrypt a message, we treat $F_\kappa(0)$ as a random key, and use a result of Shoup [25] which shows that a CCA2-secure *key encapsulation mechanism* (KEM) suffices to construct CCA2-secure encryption scheme. To decrypt a ciphertext C, first decrypt β and then α to recover κ. Then verify that $\beta = \mathsf{enc}(\alpha; F_\kappa(1)||\ldots)$. Upon verification, use the KEM mechanism to recover the message.

Unfortunately, even this scheme may still not prevent quoting. It might be possible to reorder the outer-layer β ciphertexts in a way that reorders the α ciphertext. Furthermore, this reordering might change the PRFG key κ into a related key κ' such that the pseudo-random bits constructed by $F_{\kappa'}$ are consistent with those of F_κ. If possible, this "mauling strategy" would pass the validity check on the outer-layer ciphertexts and allow the adversary to learn the original message. To prevent this, we instead use a non-malleable inner-layer encryption system. More specifically, we modify a construction of Choi et al. [3] to construct a non-malleable UCCA scheme from a 1-bit CCA2 scheme. One key factor that we discuss below is that we only need the inner scheme to satisfy a weak form of non-malleability. We take advantage of the black-box construction of Choi et al. to ensure that our construction is also entirely black-box.

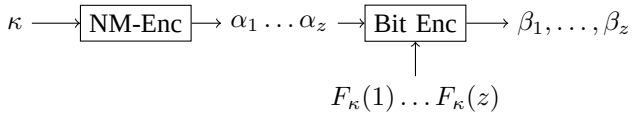

Figure 1. Cartoon of our scheme CCA2 KEM scheme $\mathbb{P}$

Intuitively, these measures make a quoting attack difficult because the adversary needs to find the pseudo-randomness

[1] Technically, they prove their result from trapdoor predicates, but it is easy to see that their oracle trapdoor predicate represents a CCA2 secure one-bit encryption scheme.

[2] A pseudo-random generator would suffice, but would introduce cumbersome notation and awkward parsing.

used to generate the outer layer encryptions before it can perform such an attack. Formally proving this, however, is not so straightforward. In the inner-ciphertext α, information about the plaintext κ is used to generate the random tapes $F_\kappa(i)$ used to form β, and therefore standard encryption security for β does not immediately apply. Moreover, information about the key to F_κ is present in the ciphertext α, and so standard security of the PRFG also does not immediately apply. In other words, since we cannot guarantee outright security of the pseudo-random key, it may be possible for the adversary to use the decryption oracle to advantage to learn about the pseudo-random key and then unwind the construction's supposed security from there.

Proof techniques: To overcome these issues, our security proof carefully analyzes the type of decryption queries that an adversary must make to break our system. Informally, there are two types of quoting queries with respect to the challenge ciphertext C^*: (1) an α-quoted query is one that quotes from the α component of C^*, and (2) a β-quoted query is one that quotes from the β-component of C^*.

Our first observation in Lemma 1 is that an adversary that makes an α-quoted query must have (except with low probability) already (or concurrently) submitted a β-quoted query. This argument is based on a series of hybrid experiments and relies on the UCCA security of both the inner and outer schemes.

Next, in Lemma 2, we consider an adversary A that breaks our scheme. If adversary A never asks a β-quoted query, then by Lemma 1, it also (almost) never asks an α-quoted query. Since such an adversary never asks a quoted query, it would therefore violate the UCCA security of the inner or outer schemes. Thus, in Lemma 2, we conclude that any adversary that breaks our scheme must ask a successful (i.e., one that does not decrypt to $\perp$) β-quoted query with noticeable probability.

Finally, in Theorem 4, we complete the security argument. We show that any adversary that makes a successful β-quoted query can be used to violate the non-malleability of the inner encryption scheme. In particular, such an adversary, when given $\alpha = \mathsf{enc}(\kappa)$ can construct a related ciphertext $\alpha' = \mathsf{enc}(\kappa')$ for which $\mathsf{enc}(\alpha_s;\ F_\kappa(s)) = \mathsf{enc}(\alpha'_t;\ F_{\kappa'}(t))$, for some s and t.

Roadmap: In §3.1, we introduce a new notion of unquoted security—denoted UCCA-security—and show that although the bit-by-bit scheme is not CCA2 secure, it does satisfy UCCA-security. Notice that non-malleability is not immediately implied by the unquotable CCA2 notion, as the adversary can still modify the challenge ciphertext, even if it cannot directly decrypt it. Thus, our next step in §3.2 is to transform a UCCA-secure scheme into a 1-wise non-malleable UCCA-secure scheme using ideas from Choi et al [3]. In this limited form of non-malleability, the adversary can only submit one decryption query. Finally, in §4 we combine the pieces and construct the fully CCA2 secure many-bit encryption scheme.

2. Basic Definitions & Notation

We assume familiarity with the standard notions and notation for negligible functions, indisitinguishibility, and encryption schemes. When A is a randomized algorithm, we use the notation $A(x;\ r)$ to indicate that algorithm A is executed on input x using random tape r. When r is not specified, it is assumed that A is executed on x using a uniformly chosen random tape. We use $||$ to denote concatenation, and $[n] = \{1, .., n\}$.

We will use $\Pi = (\mathsf{g}, \mathsf{e}, \mathsf{d})$ to denote a CCA2-secure one-bit encryption scheme and $\Pi_\mathsf{v} = (\mathsf{gv}, \mathsf{ev}, \mathsf{dv})$ to denote the system that results from concatenating encryptions from Π together to form a multi-bit unquoted CCA2 secure encryption system. We will use $\Pi_{\mathsf{NM}} = (\mathsf{nmg}, \mathsf{nme}, \mathsf{nmd})$ to denote our intermediate 1-wise NM UCCA secure scheme, and finally, $\mathbb{P} = (\mathbb{G}, \mathbb{E}, \mathbb{D})$ to denote a CCA2-secure key encapsulation mechanism. One of our constructions makes use of a strong one-time signature scheme, essentially a digital signature scheme that can be used to sign only one message, but is strongly existentially unforgeable so that an adversary cannot even create an alternate signature to the signed message. In the proceedings version we refer the reader to [15], [22] for details.

Definition 1 (1-bit CCA2 security)**.** *Let $\Pi = (\mathsf{g}, \mathsf{e}, \mathsf{d})$ be a one-bit encryption scheme and let the random variable $\mathsf{CCA2}_b(\Pi, A, k)$, where $b \in \{0, 1\}$, A is a p.p.t. algorithm and $k \in \mathbb{N}$, denote the result of the following probabilistic experiment:*

$\mathsf{CCA2}_b(\Pi, A, k)$
(1) $(pk, sk) \leftarrow \mathsf{g}(1^k)$
(2) $y \leftarrow \mathsf{e}_{pk}(b)$
(3) $b' \leftarrow A^{\mathsf{d}^*(sk,\cdot)}(y)$ $(b' \in \{0, 1\})$
(4) *Output* b'

(The decryption oracle d^* returns $\perp$ when queried on the ciphertext y, but otherwise decrypts the query using sk.)
Π is CCA2 secure if for all p.p.t. algorithms A the following two ensembles are computationally indistinguishable:

$$\left\{\mathsf{CCA2}_0(\Pi, A, k)\right\}_{k\in\mathbb{N}} \approx_s \left\{\mathsf{CCA2}_1(\Pi, A, k)\right\}_{k\in\mathbb{N}}$$

We give the definition of a CCA2 Key Encapsulation Mechanism below.

Definition 2. *A key-encapsulating mechanism $\mathbb{P} = (\mathbb{G}, \mathbb{E}, \mathbb{D})$ is CCA2 secure if, for all probabilistic polynomial time (resp. poly size circuit families) adversaries $A = (A_1, A_2)$ and for all sufficiently large k: $\Pr[\mathsf{CCA2KEM}(\mathbb{P}, A, k) = 1] \leq 1/2 + \mu(k)$, for some negligible function μ. We define the experiment* CCA2KEM *as follows:*

$\mathsf{CCA2KEM}(\mathbb{P}, A, k)$
(1) $(PK, SK) \leftarrow \mathbb{G}(1^k)$
(2) $S \leftarrow A_1^{\mathbb{D}_{SK}}(PK)$
(3) $b \leftarrow \{0,1\}$
(4) $(k_b, C^*) \leftarrow \mathbb{E}_{PK}(1^k)$
(5) $k_{1-b} \in_{\mathcal{R}} \{0,1\}^k$
(6) $b' \leftarrow A_2^{\mathbb{D}^*_{SK}}(C^*, k_0, k_1, S)$
(7) *Output* 1 *if* $b = b'$ *else Output* 0.

The oracle $\mathbb{D}^*_{SK}$ is the standard decryption oracle that answers all queries except the challenge C^*.

3. Unquoted CCA2 Encryption and q-wise Non-Malleability

As mentioned earlier, our many-bit CCA2 secure primitive will be constructed from two other schemes. For one of these, we need a many-bit encryption scheme that has some minimal non-malleability properties as well as unquoted CCA2 security properties. In this section we define these properties, and describe the construction. Let us first give an informal summary of these two notions.

Unquoted CCA2 Security: We say that a scheme $(\mathsf{gen}, \mathsf{enc}, \mathsf{dec})$ is unquoted CCA2 secure if it conforms to a weakening of the traditional CCA2 definition: the decryption oracle that the adversary queries after being issued the challenge ciphertext does not respond to queries that have been "quoted" from the challenge ciphertext. In particular, let C^* be the challenge ciphertext in the CCA2 definition, the limited decryption oracle will respond with $\perp$ to any query C' in which $\mathsf{dec}_{sk}(C')$ and $\mathsf{dec}_{sk}(C^*)$ make an identical query to d (Since the construction of dec is black-box, we can discuss it making queries to a d oracle). We note that it is exactly such queries an adversary cannot simulate in many proofs of security. Concatenated bit-by-bit encryption achieves this limited security definition.

q-wise Non-Malleability: The natural weaker notion of non-malleability that we introduce is denoted q-NM-CCA1 security. Recall that the original definition of non-malleable encryption systems from [7] was shown by Bellare and Sahai [1] and Pass et al. [19] to be roughly equivalent to a notion of indistinguishability of the plaintexts that result from a parallel query to a decryption oracle in two related experiments. More specifically, the adversary is asked to produce two plaintexts, m_0 and m_1, and based on the experiment $b \in \{0,1\}$, the adversary receives a challenge encryption of m_b (note the adversary does not know b). In the traditional NM-definition, an adversary outputs a tuple of ciphertexts whose *decryptions* should be computationally independent in the two experiments. The adversary can choose the size of this tuple, and the size can be polynomially-related to the security parameter. In a q-NM-S definition, the size of this output tuple of ciphertexts is upper-bounded by q. That is, a scheme is q-NM-S secure if the adversary cannot find a tuple of q encryptions in each experiment whose plaintexts can be efficiently distinguished. For our construction of a many-bit CCA2 primitive, we only need a construction to be 1-NM-UCCA secure scheme, but in principal our construction works for any constant q. This weaker notion of q-wise non-malleability may be of independent interest. Notice that 1-NM-UCCA security implies UCCA security. Notice also that q-wise non-malleability is stronger than the q-bounded CCA2 security notion put forth in Cramer et al. [4] because the q-wise non-malleable adversary can make arbitrarily many queries *before* receiving the challenge ciphertext.

3.1. Unquoted CCA2 Security

We begin by giving a formal definition of the CCA2 Security property.

Definition 3 (Unquoted CCA2 Security)**.** *Let $\Pi' = (\mathsf{gen}^\Pi, \mathsf{enc}^\Pi, \mathsf{dec}^\Pi)$ be an encryption scheme that makes black-box access to a one-bit encryption scheme $\Pi = (\mathsf{g}, \mathsf{e}, \mathsf{d})$. Π' is UCCA secure with respect to Π if for all p.p.t. adversaries $B = (B_1, B_2)$:*

$$\{\mathsf{UCCA}_0(\Pi, B, k)\} \approx_s \{\mathsf{UCCA}_1(\Pi, B, k)\}$$

Here UCCA_b is defined as the following experiment:

$\mathsf{UCCA}_b(\Pi', B, k)$
$(pk, sk) \leftarrow \mathsf{gen}(1^k)$
$(m_0, m_1, S_1) \leftarrow B_1^{\mathsf{dec}_{sk}}(\text{NPK})$ *s.t.* $|m_0| = |m_1|$
$y \leftarrow \mathsf{enc}_{pk}(m_b)$
$b' \leftarrow B_2^{\lfloor \mathsf{dec}_{sk} \rfloor}(y, S_1)$
Output b'

Let $Q_{\mathsf{dec}_{sk}(c)}$ be the set of queries to the oracle d made during computation $\mathsf{dec}_{sk}(c)$. We define $\lfloor \mathsf{dec}_{sk} \rfloor$ as decryption oracle that responds with $\perp$ to the challenge query y and to queries $c \neq y$ if $Q_{\mathsf{dec}_{sk}(c)} \cap Q_{\mathsf{dec}_{sk}(y)} \neq \emptyset$. Further, we say such a query c contains a **quoted query***, or in cases where the context is clear we may say query c is quoted.*

Observe that testing whether a query is quoted may not be efficiently computable by the adversary; one might worry that the use of the oracle could leak information about whether a query is quoted, and therefore leak information about the secret key. For the constructions that we propose, however, testing for a quoted query is easy.

Theorem 2. *Let $\Pi = (\mathsf{g}, \mathsf{e}, \mathsf{d})$ be a 1-bit CCA2 secure encryption scheme. For a polynomial p define the many-bit concatenating construction $\Pi_\mathsf{v} = (\mathsf{gv}, \mathsf{ev}, \mathsf{dv})$ where $\mathsf{gv} = \mathsf{g}$; for a security parameter k and message $m = (m_1, \ldots, m_{p(k)})$ of length $p(k)$ we define $\mathsf{ev}_{pk}(m) = \mathsf{e}_{pk}(m_1)||\ldots||\mathsf{e}_{pk}(m_{p(k)})$, and dv is defined analogously. Then Π_v is UCCA-secure.*

Proof: Omitted. See final version ■

3.2. q-NM Security

Definition 4 (q-wise NM Security)**.** *Let $\Pi_{\mathsf{NM}} = (\mathsf{nmg}, \mathsf{nme}, \mathsf{nmd})$ be an encryption scheme. For $S \in \{$CPA,CCA1,CCA2,UCCA2$\}$, let $\mathcal{O}_{S,1}$ and $\mathcal{O}_{S,2}$ be the pair*

of decryption oracles made available to the adversary in the S security experiment before ($\mathcal{O}_{S,1}$) and after ($\mathcal{O}_{S,1}$) the adversary is given the challenge ciphertext. We say that Π_{NM} is q-wise NM secure with respect to an S-adversary if for all p.p.t. S adversaries A and distinguishers D respectively:

$$\{\mathsf{q\text{-}NM\text{-}S}_0(\Pi_{\mathsf{NM}}, A, D, k)\} \approx_s \{\mathsf{q\text{-}NM\text{-}S}_1(\Pi_{\mathsf{NM}}, A, D, k)\}$$

q-NM-S *is defined as follows:*

$\mathsf{q\text{-}NM\text{-}S}_b(\Pi, A, D, k)$

1) $(\text{NPK}, \text{NSK}) \leftarrow \mathsf{nmg}(1^k)$
2) $(m_0, m_1, \sigma_1) \leftarrow A_1^{\mathcal{O}_{S,1}}(\text{NPK})$ such that $|m_0| = |m_1|$
3) $y \leftarrow \mathsf{nme}_{\text{NPK}}(m_b)$
4) $(c_1, \ldots, c_q, \sigma_2) \leftarrow A_2^{\mathcal{O}_{S,2}}(y, \sigma_1)$
5) Output $D(d_1, ..., d_q, \sigma_2)$ where
   $d_i \leftarrow \mathsf{nmd}(\text{NSK}, c_i)$ if $c_i \neq y$

Note that in contrast to the definition of non-malleability in the work of Pass et al. [18] and Choi et al. [3], we permit the adversary to pass state information to the distinguisher. Such information is usually redundant since the adversary can encrypt this state information and include it in its output vector; however, since we only allow q output ciphertexts, this extra information may be needed. Such state information was present in the definitions of Bellare and Sahai [1].

3.2.1. 1-NM-UCCA *Secure construction:* We give our construction in Fig. 2. The scheme is a modification of that of Choi et al. [3]. They produce a NM-CPA secure encryption scheme from a CPA secure primitive. In contrast, we start and end with stronger primitives. We construct from a many-bit UCCA2 secure primitive a many-bit 1-NM-UCCA2 secure primitive. It may be possible to use the notion of designated verifier NIZK employed by Pass et al. [18] to complete this step; however, doing so would be more complicated and would make the construction non-blackbox.

Theorem 3. *If Π_{v} is a multi-bit* UCCA *secure scheme, then the encryption scheme Π_{NM} in Fig. 2 is a 1-NM-UCCA2 secure scheme.*

Proof: The proof follows ideas from [3] with modifications. See full version. ■

4. Many Bit CCA2 Secure Encryption Scheme

We now construct a CCA2 multi-bit encryption scheme using our 1-NM-UCCA2 secure scheme Π_{NM} and our many-bit UCCA2 encryption scheme Π_{v}. Both Π_{NM} and Π_{v} are constructed in a black-box manner from our 1-bit CCA2 encryption scheme Π. Let F be a PRFG (equivalently, a PRG can be used, but the PRFG simplifies the notation). We construct a many-bit scheme that effectively encrypts a pseudo-random seed of appropriate length. We can then use this in a hybrid encryption scheme, as described by Cramer and Shoup [5], [25], to construct a many-bit arbitrary message encryption scheme using a PRFG and MAC. Note that the existence of a PRFG and a MAC are implied by the existence of a 1-bit CCA2 encryption scheme [14], [11]. We give the definition of a CCA2 secure public key encapsulation mechanism (PKEM) in Def. 2.

We construct a many-bit CCA2 secure PKEM system as described in Fig. 3 (pg. 6). The construction consists of two layers. In the first inner layer, a random key k for a PRFG F is encrypted using a 1-NM UCCA2 scheme. The ciphertext output of the inner-layer, denoted α, is then encrypted using an outer-layer encryption scheme which is UCCA2 secure, but in which the random-bits used to encrypt are given by $F_k(i)$. Call this output β. The pseudo-random bits used to construct the ciphertext β are checked by the decryption algorithm to ensure β's validity. It is this property that makes

- Assume that the input message length equals the security parameter k. Let $\Pi_{\mathsf{v}} = (\mathsf{gv}, \mathsf{ev}, \mathsf{dv})$ be a many-bit UCCA secure encryption scheme. Let $\Sigma = (\mathsf{GenSignKey}, \mathsf{Sign}, \mathsf{Verify})$ be a strong one-time signature scheme in which verification keys are of length k. Let $z(k) = |\mathsf{nme}_{\text{NPK}}(m)|$ for $(pk, sk) \leftarrow \mathsf{nmg}(1^k)$ and $m \in \{0,1\}^k$. Let F be a PRFG s.t. for $K \in_{\mathcal{R}} \{0,1\}^k$: $F_K : \{0,1\}^{z(k)} \rightarrow \mathcal{S}$ where $\mathcal{S} = \{S \mid S \subset [10k] \wedge |S| = k\}$.
- $\mathsf{nmg}(1^k)$
  1) $(pk^b_{i,j}, sk^b_{i,j}) \leftarrow \mathsf{gv}(1^k)$ for all $i \in [k], j \in [10k], b \in \{0,1\}$
  2) Pick $K \in_{\mathcal{R}} \{0,1\}^k$
  3) *Output* $\text{NPK} = \{(pk^0_{i,j}, pk^1_{i,j} | i \in [k], j \in [10k]\}$ and $\text{NSK} = \{K, (sk^0_{i,j}.sk^1_{i,j}) | i \in [k], j \in [10k]\}$
- $\mathsf{nme}(\text{NPK}, m)$
  1) Pick a deg-k poly $p(x) = m_0 + \alpha_1 x + \cdots + \alpha_k x^k$, $\alpha_1, \ldots, \alpha_k \in_{\mathcal{R}} GF(2^k),\ \alpha_k \neq 0$
  2) $s_j \leftarrow p(j),\ \forall j \in [10k]$
  3) $(\mathsf{SigSK}, \mathsf{SigVK}) \leftarrow \mathsf{GenSignKey}(1^k)$
  4) Let $(v_1, ..., v_k)$ be the bits of SigVK
  5) $c_{i,j} \leftarrow \mathsf{ev}_{pk^{v_i}_{i,j}}(s_j),\ \forall i \in [k], j \in [10k]$
  6) $\sigma \leftarrow \mathsf{Sign}_{\mathsf{SigSK}}([c_{i,j}]_{\{i \in [k], j \in [10k]\}})$.
  7) *Output* $(c, \mathsf{SigVK}, \sigma)$
- $\mathsf{nmd}(\text{NSK}, C = (c, \mathsf{SigVK}, \sigma)))$
  1) If $\mathsf{Verify}_{\mathsf{SigVK}}(\sigma, c) = \perp$ Then *Output* $\perp$
  2) $s_j \leftarrow \mathsf{dv}_{sk^{v_1}_{1,j}}(c_{1,j})\ \forall j \in [10k]$
  3) Let $w = (w_1, ..., w_{10k})$ be a Reed-Solomon codeword that agrees with $(s_1, \ldots, s_{10k})$ in at least $9k$ positions.
  4) If no such codeword *Output* $\perp$
  5) $S \leftarrow F_K(C)$ (Choose Columns)
  6) $\forall j \in S$ if
     $$\left(\left(w_j = \mathsf{dv}_{sk^{v_1}_{1,j}}(c_{1,j}) = \cdots = \mathsf{dv}_{sk^{v_k}_{k,j}}(c_{k,j})\right) \neq \top\right)$$
     Output $\perp$ Else *Output* m

Figure 2. The Non-malleable Encryption Scheme Π_{NM}

our construction a non-shielding construction, as defined by Gertner et al. [9].

We will refer to the inner encryption and its scheme as the α layer and α encryption system, respectively. Similarly, we refer to the outer encryption and its scheme as the β layer and the β encryption scheme.

Let $\Pi_{\mathsf{v}} = (\mathsf{gv}, \mathsf{ev}, \mathsf{dv})$ be a many-bit UCCA scheme
Let $\Pi_{\mathsf{NM}} = (\mathsf{nmg}, \mathsf{nme}, \mathsf{nmd})$ be a many-bit 1-NM-UCCA scheme (as per §3.2).

- $\mathbb{G}(1^k)$
  1) $(pk, sk) \leftarrow \mathsf{gv}(1^k)$.
  2) $(\text{NPK}, \text{NSK}) \leftarrow \mathsf{nmg}(1^k)$.
  3) Output $PK = (pk, \text{NPK})$ and $SK = (sk, \text{NSK})$.
- $\mathbb{E}(PK, R)$
  1) Choose $\kappa, i, r \in_{\mathcal{R}} \{0,1\}^k$
  2) $\alpha \leftarrow \mathsf{nme}_{\text{NPK}}(\kappa;\ r)$
  3) $Z \leftarrow F_\kappa(1)||F_\kappa(2)||...||F_\kappa(|\alpha|)$
  4) $\beta = \mathsf{ev}_{pk}(\alpha;\ Z)$
  5) Output $(F_\kappa(0), \beta)$
- $\mathbb{D}(SK, C')$
  1) Set $\alpha' \leftarrow \mathsf{dv}_{sk}(C')$. If $\perp \in \alpha'$ then output $\perp$.
  2) Set $\kappa \leftarrow \mathsf{nmd}_{\text{NSK}}(\alpha')$. If $\kappa = \perp$ then output $\perp$.
  3) If $\mathsf{ev}_{pk}(\alpha';\ (F_\kappa(1)||F_\kappa(2)||...||F_\kappa(|\alpha'|))) \neq C'$ then output $\perp$.
  4) Output $F_\kappa(0)$.

Figure 3. CCA2 KEM ENCRYPTION SCHEME $\mathbb{P}$

Note that scheme $\mathbb{P}$ is a black-box construction that can be viewed as $(\mathbb{G}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}, \mathbb{E}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}, \mathbb{D}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}})$, where $\Pi_{\mathsf{NM}} = (\mathsf{nmg}^{\Pi}, \mathsf{nme}^{\Pi}, \mathsf{nmd}^{\Pi})$ and $\Pi_{\mathsf{v}} = (\mathsf{gv}^{\Pi}, \mathsf{ev}^{\Pi}, \mathsf{dv}^{\Pi})$.

In order to prove that $\mathbb{P}$ is secure, we need to formalize the notion of *quoted* queries made by the adversary. Specifically, we are interested in queries to the decryption oracle $\mathbb{D}_{SK}$ that result in queries to the constituent decryption algorithms $\mathsf{nmd}_{\text{NSK}}$ or dv_{sk} that would not be permitted by the decryption oracles in the unquotable CCA2 (UCCA2) security definitions of the constituent schemes Π_{v} or Π_{NM}. To this end we define *α- and β-quoted queries* with respect to our construction $\mathbb{P} = (\mathbb{G}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}, \mathbb{E}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}, \mathbb{D}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}})$.

Definition 5 (α-quoted, β-quoted, and successful queries)**.** *Let $\beta_{SK,C} = \{(sk_i, c_i)\}$ be the set of queries made to the one-bit encryption scheme $\mathsf{d}(\cdot)$ by $\mathsf{dv}(\cdot)$ in line (1) when invoked by decryption algorithm $\mathbb{D}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}(SK, C)$.*

Similarly, let $\alpha_{SK,C} = \{(sk_i, c_i)\}$ be the set of queries made to the one-bit decryption algorithm d that $\mathsf{nmd}^{\Pi}(\text{NSK}, c)$ makes in line (2) when invoked by $\mathbb{D}^{\Pi_{\mathsf{NM}},\Pi_{\mathsf{v}}}(SK, C)$.

— *A decryption query C is a* β-quoted query with respect to challenge C^* *if* $\beta_{SK,C} \cap \beta_{SK,C^*} \neq \emptyset$.
— *A decryption query C is* an α-quoted query with respect to challenge ciphertext C^* *if* $\alpha_{SK,C} \cap \alpha_{SK,C^*} \neq \emptyset$.
— *Finally, a query C is* successful *if* $D(SK, C) \neq \perp$.

We note that our experiments require an efficient algorithm to determine whether a query C is a β-quoted query with respect to challenge C^*. For our proposed schemes $\mathbb{P}$ and Π_{v}, this task can be done by parsing and string comparisons. Similarly, in certain experiments the adversary will have access to the α ciphertext that is encrypted in the challenge ciphertext C. In these cases, it is also possible to check whether a query C is an α-quoted query with respect to C^*. Finally, there are certain situations in which C^* represents an encryption of either an α_0 or α_1 encryption. In these situations, we refer to an α_0-quoted query, or an α_1-quoted query.

Our goal is to show that any purported adversary of $\mathbb{P}$ that breaks its CCA2 security must ask a successful β-quoted query to the decryption oracle. In order to do this, we first show that any α-quoted query must be preceded by a β-quoted query.[3] Then we argue that if there are no quoted queries, then the UCCA-security suffices. Thus, we conclude that the adversary must make either a β-quoted query or an α-quoted query. Altogether, these two observations imply that any adversary that successfully breaks the CCA2KEM security of $\mathbb{P}$ must make an interesting β-quoted query with non-negligible probability.

Intuitively, a β-quoted query must be made, because everything contained in the challenge ciphertext is encrypted bit-by-bit with a CCA2 secure encryption system, so each of those bits is individually secure, and no information is leaked about them. Further, none of the *individual* bit encryptions can be mauled to try and learn information, so the adversary must maul the ciphertext by performing a β-quoting. The reason why an α-quoted query must be preceded by a β-quoted query is because the α-quoted queries contained in the challenge ciphertext are encrypted by the outer-layer of β-quoted queries. Therefore, for the adversary to embed an α-quoted query, it must first try to decrypt β-quoted queries to learn how to make an α-quoting. Alternately, it may simply attempt to cut-and-paste an α-query in to a ciphertext it creates, but in those cases the β-quoted encryptions on the outer layer of the α-query will be decrypted prior to the α-quoted query being decrypted.

For the following lemma, one needs to define a natural ordering on α- and β-quoted queries given a series of decryption oracle queries made by the adversary. Intuitively, the decryption algorithm $\mathbb{D}$ must process the β-quoted ciphertexts before the α-quoted queries as it encounters them first.

Definition 6. *Suppose an adversary A makes a series of decryption oracle queries $C_1, C_2, ...$, and suppose that two queries C_i and C_j contain α- or β-quoted queries with respect to a challenge ciphertext C^* given to the adversary. Then if $i < j$, we say that the quoted query in C_i preceded*

[3]The notion of "preceded" is formalized in Def. 6

the quoted query in C_j. If $i = j$, then we say that the β-quoted query in C_i precedes the α-quoted query in C_j.

To simplify, we often assume that the adversary is given auxiliary information containing an index denoting the first query of a certain form to a decryption oracle. In all cases, the value of these indices can be guessed by the adversary for a polynomial factor reduction in the adversary's effectiveness in the security reductions.

Lemma 1. *Let $A = (A_1, A_2)$ be a* CCA2KEM *adversary for the many-bit construction $\mathbb{P} = (\mathbb{G}, \mathbb{E}, \mathbb{D})$ described in Fig. 3. The probability that A makes an α-quoted query that is not preceded by a β-quoted query is negligible.*

Proof: (Sketch) Suppose that for infinitely many k, adversary A_2, on input a challenge ciphertext C^*, asks an α-quoted query $C' \neq C^*$ that has not been preceded by a β-quoted query with non-negligible probability $1/p(k)$, for some polynomial p. We argue that A_2 can be used to break the UCCA2 security of the Π_{v} encryption scheme. First, we assume that A_2 is in a normalized form so that it always makes a fixed number of queries. Let i_k be the smallest index for which there is a non-negligible probability that the $i_k{}^{\text{th}}$ oracle query by A_2 in $\mathsf{CCA2KEM}(\mathbb{P}, A, k)$ contains an α-quoted query that has not been preceded by a β-quoted query. We consider the following experiment $Expr_1$ simulating $\mathsf{CCA2KEM}(\mathbb{P}, A, k)$ until the i_kth query.

$Expr_1(\mathbb{P}, A, k, i_k)$
1) $(PK, SK) \leftarrow \mathbb{G}(1^k)$
2) $S \leftarrow A_1^{\mathbb{D}_{SK}}(PK)$
3) $b \in_{\mathcal{R}} \{0,1\}$
4) $(K_b, C^*) \leftarrow \mathbb{E}_{PK}(1^k)$
5) $K_{1-b} \in \{0,1\}^k$
6) Simulate $A_2^{\mathbb{D}_{SK}}(C^*, K_0, K_1, S)$ for i_k queries
7) *Output* 1 if the i_kth query contains an α-quoted query that has not been preceded by an α- or β-quoted query.
8) *Output* 0

By inspection, it holds that for infinitely many k, $\Pr[Expr_1(\mathbb{P}, A, k, i_k) = 1] \geq 1/p(k)$. We define a new experiment $Expr_2(\mathbb{P}, A, k, i_k)$ that is identical to $Expr_1(\mathbb{P}, A, k, i_k)$, except that we replace the algorithm $\mathbb{E}$ that is called in line 4 of $Expr_1$, with a modification of it named $\mathbb{E}^2$ described below. It modifies $\mathbb{E}$ so that the PRFG key used to generate the KEM key and produce randomness for the outer-layer of encryptions is independent of the key encrypted in the inner layer. Changes are in lines 1,2 and 4.

$\mathbb{E}^2(PK, R)$
1) Choose $\kappa, \kappa' \in_{\mathcal{R}} \{0,1\}^k$
2) $\alpha \leftarrow \mathsf{nme}_{\text{NPK}}(\kappa')$
3) $\beta = \mathsf{ev}_{pk}(\alpha; F_\kappa(1)||F_\kappa(2)||...||F_\kappa(|\alpha|))$
4) $Output\{F_\kappa(0), \beta\}$

Claim 1. $\{Expr_1(\mathbb{P}, A, k, i_k)\} \approx_s \{Expr_2(\mathbb{P}, A, k, i_k)\}$

The proof is by contradiction. If not, then we show how A can be used to break the UCCA security of Π_{NM}. We construct an adversary $B = (B_1, B_2)$ that simulates one of the experiments for $A = (A_1, A_2)$.

$B_1^{\mathsf{nmd}_{\text{NSK}}}(\text{NPK}, 1^k)$
1) $(pk, sk) \leftarrow \mathsf{gv}(1^k)$
2) Let $PK = (\text{NPK}, pk)$ and $SK = (\text{NSK}, sk)$.
3) Run $(S_A) \leftarrow A^{\mathbb{D}_{SK}}(PK)$. Answer queries to $\mathbb{D}_{SK}(C)$ using oracle $\mathsf{nmd}_{\text{NSK}}$ and key sk.
4) Output $(\kappa_0, \kappa_1, S_B = (PK, sk, \kappa_0, \kappa_1, S_A))$.

$B_2^{\mathsf{nmd}^*{}_{\lfloor \text{NSK} \rfloor}}(pk, S_B, c^*)$
1) $b \in_{\mathcal{R}} \{0,1\}$
2) Let $C^* \leftarrow \mathsf{ev}_{pk}(c^*;\ (F_{\kappa_0}(1)||F_{\kappa_0}(2)||...||F_{\kappa_0}(|c^*|)))$
3) Let $K_b \leftarrow F_{\kappa_0}(0)$ and let $K_{1-b} \in_{\mathcal{R}} \{0,1\}^k$.
4) Simulate $A_2^{\mathbb{D}_{SK}}(C^*, K_0, K_1, S_A)$ for i_kth queries. Answer queries to $\mathbb{D}_{SK}(C)$ by using the decryption oracle $\lfloor \mathsf{nmd}_{\text{NSK}} \rfloor$ and key sk to run $\mathbb{D}$. If an α- or β- quoted query is encountered prior to the i_kth decryption query, then quit the simulation and output 0.
5) If query i_k is an α-quoted query and not a β-quoted query Then output 1. Else output 0.

Notice that both $Expr_1$ and $Expr_2$ output 0 if there are early α- or β-quoted queries. This allows adversary B to mimic this behavior because B_2 can detect when A submits an early α- or β-quoted query and then output 0.

Observe that $\Pr[\mathsf{UCCA}_0(\Pi_{\mathsf{NM}}, B, k) = 1] = \Pr[Expr_1(\mathbb{P}, A, k, i_k) = 1]$. On the other hand, $\Pr[\mathsf{UCCA}_1(\Pi_{\mathsf{NM}}, B, k) = 1] = \Pr[Expr_2(\mathbb{P}, A, k, i_k) = 1]$. Therefore

$$\begin{aligned}&\Pr[\mathsf{UCCA}_0(\Pi_{\mathsf{NM}}, B, k) = 1] - \Pr[\mathsf{UCCA}_1(\Pi_{\mathsf{NM}}, B, k) = 1] \\ &= \Pr[Expr_1(\mathbb{P}, A, k, i_k) = 1] - \Pr[Expr_2(\mathbb{P}, A, k, i_k) = 1] \\ &\geq 1/q(k)\end{aligned}$$

This difference is noticeable and therefore contradicts the UCCA security of Π_{NM}.

We define a new experiment $Expr_3(\mathbb{P}, A, k, i_k)$ that is identical to $Expr_2(\mathbb{P}, A, k, i_k)$, except that we modify $\mathbb{E}^2$ by replacing the output of the PRFG F_κ with that of a random function. Specifically, we replace line 2 of $\mathbb{E}^2$ with $\beta \leftarrow \mathsf{ev}_{pk}(\alpha;\ R)$ for a randomly chosen R. Similarly, we replace line 5 of $\mathbb{E}^2$ with $Output\{R', \beta\}$ for randomly chosen $R' \in_{\mathcal{R}} \{0,1\}^k$.

Claim 2. $\{Expr_2(\mathbb{P}, A, k, i_k)\} \approx_s \{Expr_3(\mathbb{P}, A, k, i_k)\}$

This is a standard argument that follows from the security of a PRFG. (Proof omitted for space.)

We now introduce a new adversary B, that by virtue of $Expr_3(\mathbb{P}, A, k, i_k)$, can be used to break the UCCA2

security of $\Pi_{\mathsf{v}k} = (\mathsf{gv}, \mathsf{ev}, \mathsf{dv})$. The adversary B simulates A's execution in $Expr_3$.

— $B_1^{\mathsf{dv}_{sk}}(pk, 1^k)$

1) $(\text{NPK}, \text{NSK}) \leftarrow \mathsf{nmg}(1^k)$
2) Let $PK = (\text{NPK}, pk)$ and $SK = (\text{NSK}, sk)$.
3) Simulate $S_A \leftarrow A_1^{\mathbb{D}_{SK}}(PK)$
   For queries $\mathbb{D}_{SK}(C)$ simulate the decryption algorithm using the decryption oracle dv_{sk} and NSK.
4) $\kappa_0, \kappa_1 \in_{\mathcal{R}} \{0,1\}^k$
5) $\alpha_a \leftarrow \mathsf{nme}_{\text{NPK}}(\kappa_a),\ a \in \{0,1\}$
6) Output $(\alpha_0, \alpha_1, S_B = (S_A, \text{NSK}, PK))$

— $B_2^{\lfloor \mathsf{dv}_{sk} \rfloor}(pk, S_B, C^*)$

1) $K_0, K_1 \in_{\mathcal{R}} \{0,1\}^k$.
2) $g \in_{\mathcal{R}} \{0,1\}$.
3) Simulate $A_2^{\mathbb{D}_{SK}}(PK, C^*, K_0, K_1)$ until the i_kth query. Answer queries to $\mathbb{D}_{SK}(C)$ by using the decryption oracle $\lfloor \mathsf{dv}_{sk} \rfloor$ and NSK. If an α_0-,α_1- or β-quoted query is encountered prior to the i_kth decryption query then *Output* g. Otherwise let C_{i_k} be the i_kth query.
4) If C_{i_k} has a β-quoted query, output g.
5) Let $\alpha' \leftarrow \mathsf{dv}_{sk}(C_i)$.
6) If α' contains quoted queries with respect to α_a then output a.
7) Output g.

Notice that B runs a perfect simulation of $Expr_3$ for A until such time as there is a quoted query. When this event happens, note that B_2 executes and halts on line 6. Denote by QUOTE the event that B_2 executes line 6 and let $p'(k)$ denote its probability. Let us now analyze $\mathsf{UCCA}_b(\Pi_{\mathsf{v}}, B, k, \ell)$.

$$
\begin{aligned}
&\Pr[\mathsf{UCCA}_b(\Pi_{\mathsf{v}}, B, k, \ell) = b] \\
&\quad = \Pr[\mathsf{UCCA}_b(\Pi_{\mathsf{v}}, B, k, \ell) = b | \text{QUOTE}] \cdot \Pr[\text{QUOTE}] \\
&\qquad + \Pr[\mathsf{UCCA}_b(\Pi_{\mathsf{v}}, B, k, \ell) = b | \overline{\text{QUOTE}}] \Pr[\overline{\text{QUOTE}}] \\
&\quad = \Pr[\mathsf{UCCA}_b(\Pi_{\mathsf{v}}, B, k, \ell) = b | \text{QUOTE}] \cdot p'(k) \\
&\qquad + 1/2 \cdot (1 - p'(k)) \\
&\quad = (1 - \mu(k)) \cdot p'(k) + 1/2 \cdot (1 - p'(k)) \qquad (5) \\
&\quad \geq 1/2 + p'(k)/4 \qquad (7) \\
&\quad \geq 1/2 + 1/4 \cdot (1/p(k) - \eta(k) - \mu(k)) \qquad (8)
\end{aligned}
$$

Line 5 follows for a negligible function μ by the CPA security of Π. In particular, B_2 and thus the simulation of A_2 is statistically independent of α_{1-b}. Therefore, the only method for A_2 to generate a decryption query that has a α_{1-b} quoted query occurs with negligible probability. A formal argument appears in the full version.

Lines 7 follows for sufficiently large k. For line 8 we know that A in experiment $Expr_3$ will, with probability at least $1/p(k) - \eta(k)$, for negligible function η, execute without making any α_b or β-quoted queries before making an α_b-quoted query in its i_kth decryption oracle query. (Recall that $\{Expr_3(\mathbb{P}, A, k, i_k)\} \approx_s Expr_2(\mathbb{P}, A, k, i_k)\} \approx_s \{Expr_1(\mathbb{P}, A, k, i_k)\}$ and $\Pr[Expr_1(\mathbb{P}, A, k, i_k) = 1] \geq 1/p(k)$.) By B's perfect simulation of $Expr_3$ until a quoted query and the definition of $p'(k)$ we see that $p'(k) \geq 1/p(k) - \eta(k) - \mu(k)$. But this breaks the UCCA security of the scheme which proves the lemma.

■

Lemma 2. *Let A be a* CCA2KEM *adversary for the scheme $\mathbb{P} = (\mathbb{G}, \mathbb{E}, \mathbb{D})$ described. If A breaks $\mathbb{P}$'s security, that is for infinitely many k:* $\Pr[\mathsf{CCA2KEM}(\mathbb{P}, A, k) = 1] \geq 1/2 + 1/\mathsf{poly}(k)$*, then for infinitely many k the adversary A must make a successful $\beta-$quoted query in* $\mathsf{CCA2KEM}(\mathbb{P}, A, k)$ *with probability at least* $1/\mathsf{poly}'(k)$.

We prove the contrapositive. Suppose that for all sufficiently large k the probability that A will ask a successful β-quoted query in $\mathsf{CCA2KEM}(\mathbb{P}, A, k)$ is less than $\mu_1(k)$ for some negligible function μ_1. Let $\Pr[\mathsf{CCA2KEM}(\mathbb{P}, A, k) = 1] = v(k)$. We will show that $v(k) \leq 1/2 + \mu_2(k)$ for some negligible function μ_2, thus proving the security of $\mathbb{P}$.

We modify A so that it never asks a successful β-quoted query; in the event it were to attempt to ask a β-quoted query, it immediately simulates getting the response $\perp$. Call the modified version $\hat{A}$. It is easy to see that $\{\mathsf{CCA2KEM}(\mathbb{P}, A, k)\} \approx_s \{\mathsf{CCA2KEM}(\mathbb{P}, \hat{A}, k)\}$. We can now use the previous lemma (1) to conclude that since A makes no β-quoted queries, it makes no α-queries except with negligible probability. This allows us to use UCCA2 decryption oracles to simulate the decryption oracle for A. Therefore, we can now continue our proof in a manner similar to that of Lemma 1. We consider a series of hybrid experiments. The first simulates the CCA2KEM experiment. Then we consider an experiment in which the encryption of the challenge ciphertext uses a different key for the PRFG providing random bits to the outer β-layer encryption and for the generation of the KEMDEM key, then that which is encrypted in the inner α-layer of the challenge cipher (This is very similar the proof of Claim 1 in the proof of Lemma 1). Next, we consider another modification of the experiment where the PRFG is replaced with a truly random function, as we did in Claim 2 in the proof of Lemma 1). We then observe that since the two keys given to the adversary in the modified CCA2KEM experiment are random, the adversary cannot distinguish them. The proof is given in the full version.

Theorem 4. *If Π is a 1-bit CCA-secure scheme then construction $\mathbb{P}$ is* CCA2KEM*-secure.*

Proof: If Π is 1-bit secure, then by Thm. 2 and Thm. 3, schemes Π_{v} and Π_{NM} are respectively UCCA- and 1-NM-UCCA-secure.

Suppose that there exists an adversary A that can break the CCA2KEM security of $\mathbb{P}$. By Lemma 2, for infinitely many $k \in \mathbb{N}$, adversary A asks a successful β-quoted query with non-negligible probability $1/p(k)$. Therefore, there must be some i_kth query, which is the first query position for which A has a non-negligible chance of asking a successful β-quoted query. Modify A so that any query

before i_k which is a β-quoted query is answered with $\perp$; this has no computationally observable effect on A's behavior, as any such query's response should either have been $\perp$, or was an alternate reply but the latter can occur with only negligible probability. We now claim that A can be used to effectively break the 1-wise non-malleability of $\Pi_{\mathsf{NM}} = (\mathsf{nmg}, \mathsf{nme}, \mathsf{nmd})$. First, we note that the modified A makes no β-quoted queries before the i_kth query, and so by Lemma 1, except with negligible probability, it makes no α-quoted queries before the ith query.

We now construct an adversary B for the 1-NM-UCCA experiment (abbreviated as 1-NM for convenience) that runs experiment CCA2KEM for A. The description of B is given below. The adversary works on the premise that if A makes a successful β-query in its i_kth query, then the creation of the ciphertext query violates the non-malleability property of the inner encryption system Π_{NM}. This is because this adversary B, when given an encryption $\alpha = \mathsf{nme}(\kappa)$ simulates $\mathbb{P}$ and A until the i_kth query and finds a related encryption $\alpha' = \mathsf{nme}(\kappa')$ and indices s, t such that $\mathsf{e}_{pk}(\alpha_s; F_\kappa(s)) = \mathsf{e}_{pk}(\alpha'_t;\ F_{\kappa'}(t))$, where α_j denotes the jth bit of α.

- Adversary (B, D) for the 1-NM experiment
- $B_1^{\mathsf{nmd}_{\mathrm{NSK}}}(\mathrm{NPK}, 1^k)$
  - (1) $(pk, sk) \leftarrow g(1^k)$
  - (2) Set $PK \leftarrow (\mathrm{NPK}, pk)$. Define $SK = (\mathrm{NSK}, sk)$
  - (3) Execute $(S_A) \leftarrow A_1^{\mathbb{D}_{SK}}(PK, 1^k)$. When A_1 queries its oracle $\mathbb{D}_{SK}$, use the decryption oracle $\mathsf{nmd}_{\mathrm{NSK}}$ along with the key sk to answer.
  - (4) $\kappa_0, \kappa_1 \in_{\mathcal{R}} \{0,1\}^k$
  - (5) Output $M_0 = \kappa_0$, $M_1 = \kappa_1$ and state information $S_B = (S_A, sk, PK, M_0, M_1)$.
- $B_2^{\lfloor \mathsf{nmd}_{\mathrm{NSK}} \rfloor}(\mathrm{NPK}, c^*, S_B)$
  - (1) $g, x \in_{\mathcal{R}} \{0,1\}$
  - (2) $K_{1-x} \in_{\mathcal{R}} \{0,1\}^k$ and $K_x \leftarrow F_{\kappa_g}(0)$
  - (3) $Z \leftarrow F_{\kappa_g}(1)||F_{\kappa_g}(2)||\cdots||F_{\kappa_g}(|c^*|)$
  - (4) Compute $C^* \leftarrow \mathsf{ev}_{pk}(c^*;\ Z)$.
  - (5) Simulate $A_2^{D_{SK}}(S_A, (C^*, K_0, K_1)$ until the i_kth decryption query. Denote this query as C'. Since there are no α- or β-quoted queries (except with negligible probability), answer queries to oracle D_{SK} by using the UCCA2 oracle $\lfloor \mathsf{nmd}_{\mathrm{NSK}} \rfloor$ and sk.
  - (6) If C' is not a β-quoted query output $\perp$.
  - (7) Compute $\alpha' \leftarrow \mathsf{dv}_{sk}(C')$.
  - (8) Output ciphertext α' and state information $S = (\alpha', pk, C', g, \kappa_1)$
- The distinguisher $D(x' = \kappa', S = (\alpha', pk, C', g, \kappa_1))$
  - (1) Let $Z' \leftarrow F_{\kappa'}(1)||F_{\kappa'}(2)||\cdots||F_{\kappa'}(|\alpha'|)$
  - (2) (Test for successfulness of β-query) If $\mathsf{ev}_{pk}(\alpha'; Z') = C'$, output g.
  - (3) Otherwise output 0.

Here we briefly sketch why B breaks the assumed 1-NM-security of Π_{NM} and therefore completes the proof of our main theorem.

To simplify our analysis, we condition the following analysis on the event that α-quoted queries do not happen; this event occurs with $1 - \mu(k)$ probability, for a negligible function μ, and so affects our analysis by at most a negligible amount. Thus, we now assume they do not happen. Let i_k be (non-uniform) advice specifying the index of the first query for which there is a noticeable chance that A makes a successful β-quoted query. Let b be the bit specified by the $\mathsf{1\text{-}NM}_b$ experiment. With probability $1/2$, adversary B selects $g = b$ (line 1), and so the ciphertext C^* computed in line 4 of B_2 is a valid ciphertext for the $\mathbb{P}$ scheme. In this case, B executes a perfect simulation of the CCA2KEM experiment for adversary A, and by lemma 2, A makes a successful β-quoted query in the experiment with probability $1/p(k)$. Thus, with probability $1/p(k)$, the query C' is a successful β-quoted query.

Notice that in this case, the distinguisher D outputs the bit b chosen in the $\mathsf{1\text{-}NM}_b$ experiment with noticeable advantage. This follows because when C' is a successful β-quoted query, as tested in line 6 of B_2 and line 2 of the distinguisher D, then D outputs $g = b$. Also, when A does not ask a successful β query, then D outputs 0. For convenience, we use $\mathsf{1\text{-}NM}_b$ to denote the random value $\mathsf{1\text{-}NM}_b(\Pi_{\mathsf{NM}}, B, D, k)$. Let event Q denote a successful β-quoted query. All together, we have

$$\Pr[\mathsf{1\text{-}NM}_1 = 1 \mid g = 1] = \Pr[Q] \cdot 1 + \Pr[\overline{Q}] \cdot 0 = 1/p(k)$$
$$\Pr[\mathsf{1\text{-}NM}_0 = 1 \mid g = 0] \leq \Pr[Q] \cdot 0 + \Pr[\overline{Q}] \cdot 0 = 0$$

The concern is that when $b \neq g$, the advantage just described that B has in the 1-NM experiment is lost. This however, is not possible. When $b \neq g$, then the ciphertext C^* is an invalid ciphertext for the $\mathbb{P}$ scheme. In particular, the ciphertext C^* is comprised of the inner encryption $\alpha = \mathsf{nme}(\kappa_b)$ that has been encrypted to the β ciphertexts using the randomness $F_{\kappa_g}(1)||\ldots$; i.e. $C' = \mathsf{ev}_{\mathsf{pk}_o}(\alpha; F_{\kappa_g}(1)||\ldots)$. In this case, query C' would be a successful beta query with respect to C^* only if a portion of ciphertext C' is shared with the challenge C^*, and C' decrypts correctly. We argue that on input $C^* = \mathsf{ev}_{\mathsf{pk}_o}(\alpha = \mathsf{nme}(\kappa_b); F_{\kappa_{1-b}}(1)||\ldots)$, the query C' produced by A is a successful β query with negligible probability. As a result, the distinguisher always outputs 0 in these cases and therefore we have the following claim whose proof we defer.

Claim 3. $\Pr[\mathit{1\text{-}NM}_0 = 1 \mid g = 1] = \eta(k)$, *for some negligible function* η.

Given this claim, let us compute the final probabilities. We have that

$$\begin{aligned}
&\Pr[\mathsf{1\text{-}NM}_1 = 1] - \Pr[\mathsf{1\text{-}NM}_0 = 1] \\
&\quad = \Pr[g = 1] \cdot \Pr[\mathsf{1\text{-}NM}_1 = 1 | g = 1] \\
&\quad - \Pr[g = 1] \cdot \Pr[\mathsf{1\text{-}NM}_0 = 1 | g = 1] \\
&\quad + \Pr[g = 0] \cdot \Pr[\mathsf{1\text{-}NM}_1 = 1 | g = 0] \\
&\quad - \Pr[g = 0] \cdot \Pr[\mathsf{1\text{-}NM}_0 = 1 | g = 0] \\
&\geq 1/2p(k) - \eta(k)/2
\end{aligned}$$

which violates the 1-wise NM security of the scheme Π_{NM}. ■

ACKNOWLEDGEMENTS

We thank our anonymous referees for helpful comments.

REFERENCES

[1] M. Bellare and A. Sahai, "Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization." in *CRYPTO'99*, 1999.

[2] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," in *Proceedings of the 30th Annual Symposium on Theory Of Computing (STOC)*. ACM Press, May 1998, pp. 209–218.

[3] S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee, "Black-box construction of a non-malleable encryption scheme from any semantically secure one," in *TCC 2008*, 2008, pp. 427–444.

[4] R. Cramer, G. Hanaoka, D. Hofheinz, H. Imai, E. Kiltz, R. Pass, A. Shelat, and V. Vaikuntanathan, "Bounded cca2-secure encryption," in *ASIACRYPT*, 2007, pp. 502–518.

[5] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack." in *CRYPTO*, 1998, pp. 13–25.

[6] ——, "Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption," in *EUROCRYPT'02*, 2002.

[7] D. Dolev, C. Dwork, and M. Naor, "Nonmalleable cryptography." *SIAM J. Comput.*, vol. 30, no. 2, pp. 391–437, 2000.

[8] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *CRYPTO*, 1999, pp. 537–554.

[9] Y. Gertner, T. Malkin, and S. Myers, "Towards a separation of semantic and cca security for public key encryption," in *TCC'07*, 2007, pp. 434–455.

[10] Y. Gertner, T. Malkin, and O. Reingold, "On the impossibility of basing trapdoor functions on trapdoor predicates," in *FOCS'01*, 2001, pp. 126–135.

[11] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[12] Goldwasser and Taumann-Kilai, "On the (in)security of the fiat-shamir paradigm," in *FOCS'03*, 2003, pp. 102–115.

[13] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. of Comp. and Sys. Sci*, vol. 28, no. 2, pp. 270–299, 1984.

[14] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby, "Construction of pseudorandom generator from any one-way function," *SIAM Journal of Computing*, vol. 28, no. 4, pp. 1364–1396, 1998.

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

[16] Y. Lindell, "A simpler construction of cca2-secure public-key encryption under general assumptions," *J. Cryptol.*, vol. 19, no. 3, pp. 359–377, 2006.

[17] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen cypher-text attack," in *STOC'90*, 1990, pp. 427–437.

[18] R. Pass, A. Shelat, and V. Vaikuntanathan, "Construction of a non-malleable encryption scheme from a any semantically secure one," in *CRYPTO*, 2006, pp. –.

[19] ——, "Relations among notions of non-malleability for encryption," in *ASIACRYPT*, 2007, pp. –.

[20] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *STOC '08*. New York, NY, USA: ACM, 2008, pp. 187–196.

[21] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *CRYPTO'91*, 1992, pp. 433–444.

[22] Rompel, "One-way functions are necessary and sufficient for secure signatures," in *STOC'90*, 1990.

[23] A. Rosen and G. Segev, "Chosen ciphertext security via correlated products," in *TCC'09*, 2009.

[24] A. Sahai, "Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security," in *FOCS'99*, 1999, pp. 543–553.

[25] V. Shoup, "A proposal for an iso standard for public key encryption," Dec 20 2001, ver 2.1.

[26] H. Wee, "Chosen-ciphertext security: the short, the long and the streaming," July 2009, personal communication.