Self-Destruct Non-Malleability

Sandro Coretti[1], Yevgeniy Dodis[2], Björn Tackmann[1], and Daniele Venturi[3]

[1]ETH Zürich
[2]New York University
[3]Sapienza University of Rome

October 21, 2014

Abstract

We introduce a new security notion for public-key encryption (PKE) that we dub *non-malleability under (chosen-ciphertext) self-destruct attacks* (NM-SDA), which appears to be the strongest natural PKE security notion below full-blown chosen-ciphertext (IND-CCA) security. In this notion, the adversary is allowed to ask many adaptive "parallel" decryption queries (i.e., a query consists of many ciphertexts) up to the point when the first invalid ciphertext is submitted. As such, NM-SDA security generalizes non-malleability against chosen plaintext attacks (NM-CPA, where only one parallel decryption query is allowed) and recently introduced indistinguishability against (chosen-ciphertext) self-destruct attacks (IND-SDA, where each adaptive query consists of a single ciphertext). After showing that NM-SDA is a *strict* strengthening of NM-CPA and IND-SDA and allows for more applications, we establish the following two results:

- **Domain Extension.** For any $K > 1$, there is a black-box construction of a K-bit NM-SDA PKE scheme from a single-bit NM-SDA PKE scheme. Moreover, this can be done using only $\mathcal{O}(K + \lambda)$ calls to the underlying single-bit NM-SDA scheme, where λ is the security parameter. To achieve our goal, we define and construct a novel type of continuous non-malleable code (NMC), called secret-state NMC, as we show that standard continuous NMCs are not enough for the natural "expand-then-encrypt-bit-by-bit" approach to work.
- **Black-Box Construction from IND-CPA.** Prior work showed that NM-CPA secure PKE can be constructed from any IND-CPA secure PKE in a black-box way. Here we show that the same construction actually achieves our strictly stronger notion of NM-SDA security. (This requires a non-trivial extension of the original security proof to handle multiple parallel decryption queries.) Hence, the notions of IND-CPA, NM-CPA, IND-SDA and NM-SDA security are all equivalent, lying (plausibly, strictly?) below IND-CCA security. We also show how to improve the rate of the resulting NM-SDA scheme from quadratic to linear.

1 Introduction

One of the main concerns in public-key cryptography is what security level a public-key encryption (PKE) scheme should satisfy. The most basic security notion is that of indistinguishability under chosen-plaintext attacks (IND-CPA) [20], where we demand that an adversary with no decryption capabilities be unable to distinguish between the encryption of two messages. Although extremely important and useful for a number of applications, in many cases IND-CPA security is not sufficient. For example, consider a simple setting of an electronic auction, where the auctioneer U publishes a public key pk, and invites several participants $P_1, \ldots, P_q$ to encrypt their bids b_i under pk. As was observed in the seminal paper of Dolev *et al.* [14], although IND-CPA security of encryption ensures that P_1 cannot decrypt a bid of P_2 under the ciphertext e_2, it leaves open the possibility that P_1 can construct a special ciphertext e_1 which decrypts to a *related* bid b_1 (e.g., $b_1 = b_2 + 1$). Hence, to overcome such "malleability" problems, stronger forms of security are required.

The strongest such level of PKE security is indistinguishability under chosen-ciphertext attacks (IND-CCA), where the adversary is given unrestricted, adaptive access to a decryption oracle (modulo not being able to ask on the "challenge ciphertext"). This notion is sufficient for most natural

applications of PKE, and several generic [14, 26, 28, 5, 24] and concrete [12, 13, 23, 21] constructions of IND-CCA secure encryption schemes are known by now. Unfortunately, all these constructions either rely on specific number-theoretic assumptions, or use much more advanced machinery (such as non-interactive zero-knowledge proofs or identity-based encryption) than IND-CPA secure encryption. Indeed, despite numerous efforts (e.g., a partial negative result [19]), the relationship between IND-CPA and IND-CCA security remains unresolved until now. This motivates the study of various "middle-ground" security notions between IND-CPA and IND-CCA, which are sufficient for applications, and, yet, might be constructed from simpler basic primitives (e.g., any IND-CPA encryption).

One such influential notion is non-malleability under chosen-plaintext attacks (NM-CPA), originally introduced by Dolev *et al.* [14] with the goal of precisely addressing the auction example above, by demanding that an adversary must not be able to maul ciphertexts to other ciphertexts encrypting related plaintexts. As was later shown by Bellare and Sahai [4] and Pass *et al.* [27], NM-CPA is equivalent to security against adversaries with access to a *non-adaptive* decryption oracle, meaning that the adversary can only ask one "parallel" decryption query. Although NM-CPA appears much closer to IND-CCA than IND-CPA security, a seminal result by Choi *et al.* [9] showed that one can generically build NM-CPA encryption from any IND-CPA-secure scheme. Thus, NM-CPA schemes can be potentially based on weaker assumptions than IND-CCA schemes, and yet suffice for important applications. Still, in the auction example above, the auctioneer U must change its public key pk with every new auction, if only NM-CPA secure encryption is used.

In a different vein, Coretti *et al.* [10] have recently introduced another middle-ground security notion for encryption, which we term indistinguishability under (chosen-ciphertext) self-destruct attacks (IND-SDA) in this paper.[1] Here the adversary gets access to an *adaptive* decryption oracle, which, however, stops decrypting after the first *invalid* ciphertext is submitted. Applying this notion to our auction example, it means that the auctioneer can reuse the secret key for subsequent auctions, as long as all the encrypted bids are valid. Unfortunately, if an invalid ciphertext is submitted, even the results of the *current* auction should be discarded, as IND-SDA security is not powerful enough to argue that the decryptions of the remaining ciphertexts are unrelated w.r.t. prior plaintexts.

Our Work. Motivated by the above, we introduce a new security notion that we dub *non-malleability under (chosen-ciphertext) self-destruct attacks* (NM-SDA). This notion (see Definition 3) naturally combines NM-CPA and IND-SDA, by allowing the adversary to ask many adaptive "parallel" decryption queries (i.e., a query consists of many ciphertexts) up to the point when the first invalid ciphertext is submitted. In such a case, the whole parallel decryption query containing an invalid ciphertext is still answered in full, but no future decryption queries are allowed. By being stronger (as we show below) than both NM-CPA and IND-SDA, NM-SDA security appears to be the strongest natural security notion of PKE which is still weaker (as we give evidence below) than IND-CCA. In particular, it seems to apply better to the auction example above: First, unlike with basic NM-CPA, the auctioneer can reuse the same public key pk, provided no invalid ciphertexts were submitted. Second, unlike IND-SDA, the current auction can be safely completed, even if some ciphertexts are invalid. Compared to IND-CCA, however, the auctioneer will still have to change its public key for *subsequent* auctions if some of the ciphertexts are invalid. Still, one can envision situations where parties are penalized for submitting such malformed ciphertexts, in which case NM-SDA security might be practically sufficient, potentially leading to a more efficient implementation (as compared to using full-blown IND-CCA PKE).

Having introduced and motivated NM-SDA security, we provide a comprehensive study of this notion, and its relationship to other PKE security notions. First, we observe that the prior notions of NM-CPA and IND-SDA are indeed incomparable, meaning that there are (albeit contrived) schemes that satisfy the former but not the latter notion and vice versa (cf. Theorem 20 in Appendix A). This also implies that our notion of NM-SDA security is strictly stronger than either of the two other notions. Next, we turn to study two fundamental questions related to NM-SDA. The first question is that of domain extension, i.e., how to construct a multi-bit NM-SDA PKE scheme from a single-bit NM-SDA PKE scheme (in a black-box manner). The second question is the relation between NM-SDA and IND-CPA. We detail these below.

[1] The original name used in [10] is self-destruct chosen-ciphertext attacks security.

Domain Extension. One important issue in public-key cryptography is to investigate for which security notions single-bit public-key encryption implies multi-bit public-key encryption. For IND-CPA, this question is simple [20], since the parallel repetition of a single-bit scheme (i.e., encrypting every bit of a message separately) yields an IND-CPA secure multi-bit scheme. For the other notions considered in this paper, i.e., for NM-CPA, IND-SDA, and NM-SDA, as well as for IND-CCA, the parallel repetition (even using independent public keys) is not a scheme that achieves the same security level as the underlying single-bit scheme. However, Coretti *et al.* [10] provide a more involved single-to-multi-bit transformation for IND-SDA security (based on non-malleable codes [16]; see below), and Myers and Shelat [25], as well as Hohenberger *et al.* [22], provide (much) more complicated such transformations for IND-CCA security. To complement these works, we answer the question of domain extension for NM-SDA and NM-CPA in the affirmative. In particular we show the following result:

Theorem 1 (Informal)**.** *For any $K > 1$, there is a black-box construction of a K-bit NM-SDA (resp. NM-CPA) PKE scheme from a single-bit NM-SDA (resp. NM-CPA) PKE scheme. Moreover, this can be done using only $\mathcal{O}(K + \lambda)$ calls to the underlying single-bit NM-SDA (resp. NM-CPA) scheme, where λ is the security parameter.*[2]

The proof of Theorem 1 can be found in Section 4. Our approach follows that of [10] and combines single-bit PKE with so-called *non-malleable codes (NMCs)*, introduced by Dziembowski *et al.* [16]. Intuitively, NMCs protect encoded messages against a tampering adversary, which tampers with the codeword by means of applying functions f from a particular function class $\mathcal{F}$ to it, in the sense that the decoding results in either the original message or a completely unrelated value.

Our construction, based on the one in [10], has the following simple structure (see also Figure 4): The plaintext m is first encoded using an appropriate non-malleable code into an encoding c, which is in turn encrypted bit-by-bit (under independent public keys) with the single-bit NM-SDA scheme.[3] The fact that NM-SDA security (or CCA security in general) guarantees that an attacker can either leave a ciphertext intact or replace it, which results in an unrelated message, translates to the following capability of an adversary w.r.t. decryption queries: It can either leave a particular bit of the encoding unchanged, or fix it to 0 or to 1. Therefore, the tamper class against which the non-malleable code must be resilient is the class $\mathcal{F}_{\mathsf{set}}$ of functions that tamper with each bit of an encoding individually and can either leave it unchanged or replace it by a fixed value.

The main new challenge for our construction is to deal with the *parallel* decryption queries. In particular, in order for the combined scheme to be NM-SDA secure, the NMC needs to be resilient against parallel tamper queries as well. Unfortunately, we show that no standard non-malleable code (as originally defined by Dziembowski *et al.* [16] and Faust *et al.* [17]) can achieve this notion (see Section 4.5). Fortunately, we observe that the NMC concept can be extended to allow the decoder to make use of (an initially generated) secret state, which simply becomes part of the secret key in the combined scheme. This modification of NMCs—called secret-state NMCs—allows us to achieve resilience against parallel tampering and may be of independent interest. This reduces our question to building a secret-state non-malleable code resilient against continuous parallel tampering attacks from $\mathcal{F}_{\mathsf{set}}$. We construct such a code in Section 4.3, by combining the notion of error-correcting secret sharing (see [16]) with the idea of a secret "trigger set" [9]. This construction forms one of the main technical contributions of our work.

Construction from CPA. Next, we turn to the question if NM-SDA secure PKE can be built from IND-CPA secure PKE. While this question is still wide open for building full IND-CCA security, it is resolved in the affirmative [9] for weaker NM-CPA security. As our main result in this area, we show that NM-SDA (and, hence, IND-SDA) security can be realized from IND-CPA security:

Theorem 2 (Informal)**.** *There exists a black-box construction of an NM-SDA-secure PKE scheme from an IND-CPA-secure PKE scheme.*

Hence, the notions of IND-CPA, NM-CPA, IND-SDA, and NM-SDA security are all equivalent, lying (plausibly, strictly?) below IND-CCA security. See Figure 1.

[2] Once $K = \Omega(\lambda)$, one can also use hybrid encryption; thus, in practice $\mathcal{O}(\lambda)$ calls should suffice for any K.

[3] Technically, this scheme only achieves a slight relaxation of NM-SDA security, called *replayable* NM-SDA security, but the latter can be easily transformed into the former.

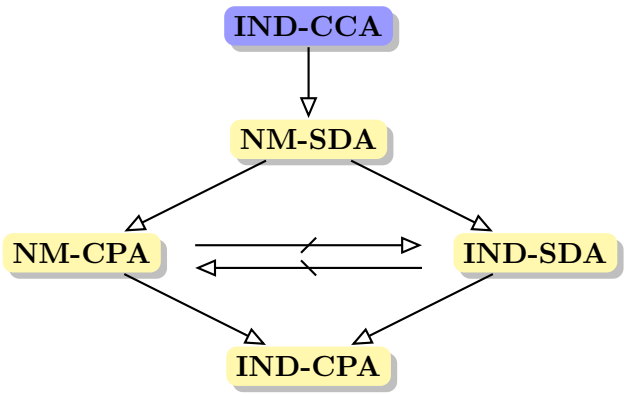

Figure 1: *Diagram of the main relationships between the security notions considered in this paper. $X \to Y$ means that X implies Y; $X \not\to Y$ indicates a separation between X and Y. Notions with the same color are equivalent under black-box transformations; notions with different colors are not known to be equivalent.*

The proof of Theorem 2 appears in Section 5. In fact, we show that the *very* same construction of Choi *et al.* already achieves NM-SDA security (rather than only NM-CPA security). Our proof much follows the pattern of the original one, except for one key step in the proof, where a brand new proof technique is required. Intuitively, we need to argue that no sensitive information about the secret "trigger set" is leaked to the adversary, unless one of the ciphertexts is invalid. This rather general technique (for analyzing security of so called "parallel stateless self-destruct games") may be interesting in its own right (e.g., it is also used in the security proof of our non-malleable code in Section 4), and is detailed in Section 6.

Along the way, we also manage to slightly abstract the transformation of [9], and to re-phrase it in terms of certain error-correcting secret-sharing schemes (ECSSs) satisfying a special property (as opposed to using Reed-Solomon codes directly as an example of such a scheme). Aside from a more modular presentation, this also allows us to instantiate the required ECSS more efficiently, and improves the rate of the transformation of [9] from quadratic to linear (while also arguing NM-SDA, instead of NM-CPA, security). Additionally, with our improved modularity the NM-SDA construction from IND-CPA security starts to look somewhat similar (although more complicated) to our domain-extension construction for NM-SDA, which also uses error-correcting secret-sharing, albeit in a somewhat more efficient way. It is an interesting open question if this "syntactic similarity" could be formalized, perhaps leading to a better, more modular understanding of how to achieve NM-SDA security from simpler components.

2 Preliminaries

In this section, we introduce notational conventions and basic concepts that we use throughout the work.

Bits and symbols. Let $\ell \in \mathbb{N}$. For any multiple $m = t\ell$ of ℓ, an m-bit string $x = (x[1], \ldots, x[m]) = (x_1, \ldots, x_t)$ can be seen as composed of its *bits* $x[j]$ or its *symbols* $x_i \in \{0,1\}^\ell$. For two m-bit strings x and y, denote by $d_{\mathsf{H}}(x, y)$ their hamming distance as the number of *symbols* in which they differ.

Oracle Algorithms. Oracle algorithms are algorithms that can make special oracle calls. An algorithm A with an oracle O is denoted by $A(O)$. Note that oracle algorithms may make calls to other oracle algorithms (e.g., $A(B(O))$).

Distinguishers and Reductions. A *distinguisher* is an (possibly randomized) oracle algorithm $D(\cdot)$ that outputs a single bit. The distinguishing advantage on two (possibly stateful) oracles S and T is defined by

$$\Delta^D(S,T) \quad := \quad |\mathsf{P}[D(S)=1] - \mathsf{P}[D(T)=1]|,$$

where the probabilities are taken over the randomness of D as well as S and T, respectively.

Reductions between distinguishing problems are modeled as oracle algorithms as well. Specifically, when reducing distinguishing two oracles U and V to distinguishing S and T, one exhibits an oracle algorithm $R(\cdot)$ such that $R(U)$ behaves as S and $R(V)$ as T; then, $\Delta^D(S,T) = \Delta^D(R(U),R(V)) = \Delta^{D(R(\cdot))}(U,V)$.

Error-correcting sharing schemes. The following notion of error-correcting sharing schemes is used in several places in this paper.

Definition 1 (Error-correcting sharing scheme)**.** *Let $n \in \mathbb{N}$ be a security parameter and $\mathbb{F}$ a field of size $L = 2^\ell$ for some $\ell \in \mathbb{N}$. A $(k, n, 2\alpha, \tau)$*-error-correcting sharing scheme (ECSS) over $\mathbb{F}$ *is a pair of algorithms (E, D), where $\mathsf{E} : \mathbb{F}^k \to \mathbb{F}^n$ is randomized and $\mathsf{D} : \mathbb{F}^n \times \mathbb{N} \to \mathbb{F}^k \cup \{\perp\}$ is deterministic, with the following properties:*

- Error correction: *It is possible to efficiently correct up to $2\alpha n$ errors, i.e., for any $x \in \mathbb{F}^k$ and any w output by $\mathsf{E}(x)$, if $d_{\mathsf{H}}(c, w) \leq t$ for some $c \in \mathbb{F}^n$ and $t \leq 2\alpha n$, then $\mathsf{D}(c, t) = x$.*[4]
- Secrecy: *The symbols of a codeword are individually uniform over $\mathbb{F}$ and and τn-wise independent (over the randomness of E).*

This paper considers two particular instantiations of ECSSs:

- *Reed-Solomon codes:* Cheraghchi and Guruswami [8] provide an ECSS based on a construction by Dziembowski *et al.* [16] and on Reed-Solomon codes with $\ell = \log n$. One can show that it achieves the following parameters (not optimized): $\alpha = 1/8$, $\tau = 1/8$ and rate $k/n \geq 1/4$.
- *Algebraic geometric codes:* Using algebraic geometric codes, Cramer *et al.* [11] provide an ECSS with $\ell = \mathcal{O}(1)$ and still constant error correction, secrecy, and rate (but with worse concrete constants than Reed-Solomon codes).

One-time signatures. A *digital signature scheme* (DSS) is a triple of algorithms $\Sigma = (KG, S, V)$, where the key-generation algorithm KG outputs a key pair $(\mathsf{sk}, \mathsf{vk})$, the (probabilistic) signing algorithm S takes a message m and a signing key sk and outputs a signature $s \leftarrow S_{\mathsf{sk}}(m)$, and the verification algorithm takes a verification key vk, a message m, and a signature s and outputs a single bit $V_{\mathsf{vk}}(m, s)$.

A *(strong) one-time signature (OTS) scheme* is a digital signature scheme that is secure as long as an adversary only observes a single signature. More precisely, OTS security is defined using the following game $G^{\Sigma,\mathsf{ots}}$ played by an adversary A: Initially, the game generates a key pair $(\mathsf{sk}, \mathsf{vk})$ and hands the verification key vk to A. Then, A can specify a single message m for which he obtains a signature $s \leftarrow S_{\mathsf{vk}}(m)$. Then, the adversary outputs a pair (m', s'). The adversary wins the game if $(m', s') \neq (m, s)$ and $V_{\mathsf{vk}}(m', s') = 1$. The *advantage* of A is the probability (over all involved randomness) that A wins the game, and is denoted by $\Gamma^A(G^{\Sigma,\mathsf{ots}})$.

Definition 2. *A DSS scheme Σ is* (t, ε)-secure *if for all adversaries A with running time at most t, $\Gamma^A(G^{\Sigma,\mathsf{ots}}) \leq \varepsilon$.*

3 Non-Malleability under Self-Destruct Attacks

A *public-key encryption (PKE) scheme* with message space $\mathcal{M} \subseteq \{0,1\}^*$ and ciphertext space $\mathcal{C}$ is defined as three algorithms $\Pi = (KG, E, D)$, where the key-generation algorithm KG outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, the (probabilistic) encryption algorithm E takes a message $m \in \mathcal{M}$ and a public key pk and outputs a ciphertext $e \leftarrow E_{\mathsf{pk}}(m)$, and the decryption algorithm takes a ciphertext $e \in \mathcal{C}$ and a secret key sk and outputs a plaintext $m \leftarrow D_{\mathsf{sk}}(e)$. The output of the decryption algorithm can be the special symbol $\perp$, indicating an invalid ciphertext. A PKE scheme is correct if $m = D_{\mathsf{sk}}(E_{\mathsf{pk}}(m))$ (with probability 1 over the randomness in the encryption algorithm) for all messages m and all key pairs $(\mathsf{pk}, \mathsf{sk})$ generated by KG.

Security notions for PKE schemes in this paper are formalized using the distinguishing game $G_b^{\Pi,q,p}$, depicted in Figure 2: The distinguisher (adversary) is initially given a public key and then specifies two

[4] The reason for requiring error correction 2α (instead of α) is purely notational convenience.

Distinguishing Game $G_b^{\Pi,q,p}$

init
- $\mathsf{ctr} \leftarrow 0$
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow KG$
- **output** pk

on $(\mathsf{chall}, m_0, m_1)$ *with* $|m_0| = |m_1|$
- $e \leftarrow E_{\mathsf{pk}}(m_b)$
- **output** e

on $(\mathsf{dec}, e^{(1)}, \ldots, e^{(p)})$
- $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
- **for** $j \leftarrow 1$ **to** p
  - $m^{(j)} \leftarrow D_{\mathsf{sk}}(e^{(j)})$
  - **if** $e^{(j)} = e$
    - $m^{(j)} \leftarrow \mathsf{test}$
- **output** $(m^{(1)}, \ldots, m^{(p)})$
- **if** $\exists j : m^{(j)} = \perp$ *or* $\mathsf{ctr} \geq q$
  - **self-destruct**

Figure 2: *Distinguishing game* $G_b^{\Pi,q,p}$*, where* $b \in \{0,1\}$*, used to define security of a PKE scheme* $\Pi = (KG, E, D)$*. The numbers* $q, p \in \mathbb{N}$ *specify the maximum number of decryption queries and their size, respectively. The command* **self-destruct** *results in all future decryption queries being answered by* $\perp$*.*

messages m_0 and m_1. One of these, namely m_b, is encrypted and the adversary is given the resulting challenge ciphertext. During the entire game, the distinguisher has access to a decryption oracle that allows him to make at most q decryption queries, each consisting of at most p ciphertexts. Once the distinguisher specifies an invalid ciphertext, the decryption oracle self-destructs, i.e., no additional decryption queries are answered.

The general case is obtained when both q and p are arbitrary (denoted by $q = p = *$), which leads to our main definition of non-malleability under (chosen-ciphertext) self-destruct attacks (NM-SDA). For readability, set $G_b^{\Pi,\mathsf{nm\text{-}sda}} := G_b^{\Pi,*,*}$ for $b \in \{0,1\}$. Formally, NM-SDA is defined as follows:

Definition 3 (Non-malleability under self-destruct attacks)**.** *A PKE scheme* Π *is* (t, q, p, ε)-NM-SDA-secure *if for all distinguishers* D *with running time at most* t *and making at most* q *decryption queries of size at most* p *each,*

$$\Delta^D(G_0^{\Pi,\mathsf{nm\text{-}sda}}, G_1^{\Pi,\mathsf{nm\text{-}sda}}) \quad \leq \quad \varepsilon.$$

All other relevant security notions in this paper can be derived as special cases of the above definition, by setting the parameters q and p to different values.

Chosen-plaintext security (IND-CPA). In this variant, the distinguisher is not given access to a decryption oracle, i.e., $q = p = 0$. For readability, set $G_b^{\Pi,\mathsf{ind\text{-}cpa}} := G_b^{\Pi,0,0}$ for $b \in \{0,1\}$ in the remainder of this paper. We say that Π is (t, ε)-IND-CPA-secure if it is, in fact, $(t, 0, 0, \varepsilon)$-NM-SDA-secure.

Non-malleability (NM-CPA). A scheme is non-malleable under chosen-plaintext attacks [27] (NM-CPA), if the adversary can make a single decryption query consisting of arbitrarily many ciphertexts, i.e., $q = 1$ and p arbitrary (denoted by $p = *$). Similarly to above, set $G_b^{\Pi,\mathsf{nm\text{-}cpa}} := G_b^{\Pi,1,*}$ for $b \in \{0,1\}$. We say that Π is (t, p, ε)-NM-CPA-secure if it is, in fact, $(t, 1, p, \varepsilon)$-NM-SDA-secure.[5]

Indistinguishability under self-destruct attacks (IND-SDA). In this variant, introduced in [10], the decryption oracle allows arbitrarily many queries, but each of them may consist of a single ciphertext only, i.e., q arbitrary (denoted by $q = *$) and $p = 1$. Once more, set $G_b^{\Pi,\mathsf{ind\text{-}sda}} := G_b^{\Pi,*,1}$. We say that Π is (t, q, ε)-IND-SDA-secure if it is, in fact, $(t, q, 1, \varepsilon)$-NM-SDA-secure.

Chosen-ciphertext security (IND-CCA). The standard notion of IND-CCA security can be obtained as a generalization of NM-SDA where $q = *$, $p = 1$, and the decryption oracle never self-destructs. We do not define this notion formally, as it is not the main focus of this paper.

Asymptotic formulation. To allow for concise statements, sometimes we prefer to use an asymptotic formulation instead of stating concrete parameters. More precisely, we will say that a PKE scheme

[5] Note that the way NM-CPA is defined here is slightly stronger than the normal notion. This is due to the adversary's ability to ask a parallel decryption query at any time—as opposed to only after receiving the challenge ciphertext in earlier definitions (cf., e.g., [27]).

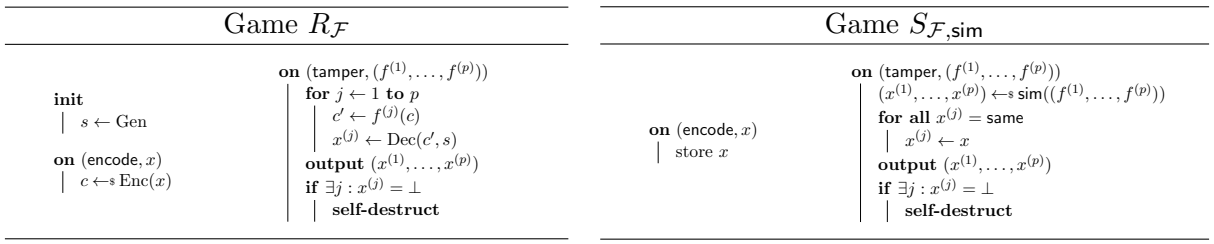

Figure 3: *Distinguishing game* $(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}})$ *used to define non-malleability of a secret-state coding scheme* $(\text{Gen}, \text{Enc}, \text{Dec})$*. The command* **self-destruct** *has the effect that all* future *queries are answered by* $\perp$*.*

Π is X-secure (for X ∈ {IND-CPA, NM-CPA, IND-SDA, NM-SDA}) if for all efficient adversaries the advantage ε in the corresponding distinguishing game is negligible in the security parameter.

Non-Malleable CPA vs. Indistinguishable SDA. We provide a separation between the notions of NM-CPA and IND-SDA security; a corresponding theorem and proof can be found in Appendix A. Given such a separation, our notion of NM-SDA security (see Definition 3) is strictly stronger than either of the two other notions.

4 Domain Extension

This section contains one of our main technical results. We show how single-bit NM-SDA PKE can be combined with so-called *secret-state non-malleable codes* resilient against *continuous parallel tampering*, which we believe is an interesting notion in its own right, to achieve multi-bit NM-SDA-secure PKE. We construct such a code and prove its security, and, additionally, we show that no code without secret state can achieve security against parallel tampering unconditionally.[6]

4.1 A New Flavor of Non-Malleable Codes

Non-malleable codes were introduced by Dziembowski *et al.* [16]. Intuitively, they protect encoded messages in such a way that any tampering with the codeword causes the decoding to either output the original message or a completely unrelated value. The original notion can be extended to include the aforementioned secret state in the decoder as follows:

Definition 4 (Code with secret state)**.** *A* (K, N)-code with secret state (CSS) *is a triple of algorithms* $(\text{Gen}, \text{Enc}, \text{Dec})$*, where the (randomized)* state-generation algorithm Gen *outputs a* secret state s *from some set* $\mathcal{S}$*, the (randomized)* encoding algorithm Enc *takes a* K*-bit plaintext* x *and outputs an* N*-bit encoding* $c \leftarrow \text{Enc}(x)$*, and the (deterministic)* decoding algorithm Dec *takes an encoding as well as some secret state* $s \in \mathcal{S}$ *and outputs a plaintext* $x \leftarrow \text{Dec}(c, s)$ *or the special symbol* $\perp$*, indicating an* invalid encoding.

Tampering attacks are captured by functions f, from a certain function class $\mathcal{F}$, that are applied to an encoding. The original definition by [16] allows an attacker to apply only a single tamper function. In order to capture continuous parallel attacks, the definition below permits the attacker to repeatedly specify parallel tamper queries, each consisting of several tamper functions. The process ends as soon as one of the tamper queries leads to an invalid codeword.

The non-malleability requirement is captured by considering a real and an ideal experiment. In both experiments, an attacker is allowed to encode a message of his choice. In the real experiment, he may tamper with an actual encoding of that message, whereas in the ideal experiment, the tamper queries are answered by a (stateful) simulator. The simulator is allowed to output the special symbol same, which the experiment replaces by the originally encoded message. In either experiment, if a component of the answer vector to a parallel tamper query is the symbol $\perp$, a self-destruct occurs, i.e., all *future* tamper queries are answered by $\perp$. The experiments are depicted in Figure 3.

[6] The question whether the notion is achievable by a computationally-secure code remains open for future work.

Definition 5 (Non-malleable code with secret state)**.** *Let $q, p \in \mathbb{N}$ and $\varepsilon > 0$. A CSS* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is* $(\mathcal{F}, q, p, \varepsilon)$-non-malleable *if the following properties are satisfied:*

- Correctness: *For each $x \in \{0,1\}^K$ and all $s \in \mathcal{S}$ output by* Gen*,* $\mathrm{Dec}(\mathrm{Enc}(x), s) = x$ *with probability* 1 *over the randomness of* Enc*.*
- Non-Malleability: *There exists a (possibly stateful) simulator* sim *such that for any distinguisher D asking at most q parallel queries, each of size at most p, $\Delta^D(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \leq \varepsilon$.*

We remark that for codes without secret-state (as the ones considered in [16]), one obtains the standard notion of non-malleability [16] by setting $q = p = 1$, and continuous non-malleability [17] by letting $p = 1$ and q arbitrary (i.e., $q = *$).

4.2 Combining Single-bit PKE and Non-Malleable Codes

Our construction of a multi-bit NM-SDA-secure PKE scheme Π' from a single-bit NM-SDA-secure scheme Π and a secret-state non-malleable (K, N)-code follows the approach of [10]: It encrypts a K-bit message m by first computing an encoding $c = (c[1], \ldots, c[N])$ of m and then encrypting each bit $c[i]$ under an independent public key of Π; it decrypts by first decrypting the individual components and then decoding the resulting codeword using the secret state of the non-malleable code (which is part of the secret key). The scheme is depicted in detail in Figure 4.

Intuitively, NM-SDA security (or CCA security in general) guarantees that an attacker can either leave a message intact or replace it by an independently created one. For our construction, which separately encrypts every bit of an encoding of the plaintext, this translates to the following capability of an adversary w.r.t. decryption queries: It can either leave a particular bit of the encoding unchanged or fix it to 0 or to 1. Therefore, the tamper class against which the non-malleable code must be resilient is the class $\mathcal{F}_{\mathsf{set}} \subseteq \{f \mid f : \{0,1\}^N \to \{0,1\}^N\}$ of functions that tamper with each bit of an encoding individually and can either leave it unchanged or replace it by a fixed value. More formally, $f \in \mathcal{F}_{\mathsf{set}}$ can be characterized by $(f[1], \ldots, f[N])$, where $f[j] : \{0,1\} \to \{0,1\}$ is the action of f on the j^{th} bit and $f[j] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}\}$ with the meaning that it either sets the j^{th} bit to 0 (zero) or to 1 (one) or leaves it unchanged (keep).

Before stating the theorem about the security of our construction Π', it needs to be pointed out that it achieves only the so-called *replayable* variant of NM-SDA security. The notion of replayable CCA (RCCA) security (in general) was introduced by Canetti *et al.* [6] to deal with the fact that for many applications (full) CCA security is unnecessarily strict. Among other things, they provide a MAC-based generic transformation of RCCA-secure schemes into CCA-secure ones, which we can also apply in our setting (as we show) to obtain a fully NM-SDA-secure scheme Π''.

Theorem 3. *Let $q, p \in \mathbb{N}$ and Π be a $(t + t_{\mathsf{1bit}}, q, p, \varepsilon_{\mathsf{1bit}})$-NM-SDA-secure 1-bit PKE scheme, (T, V) a $(t + t_{\mathsf{mac}}, 1, qp, \varepsilon_{\mathsf{mac}})$-MAC, and* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *a $(\mathcal{F}_{\mathsf{set}}, q, p, \varepsilon_{\mathsf{nmc}})$-non-malleable (K, N)-code with secret state. Then, Π'' is (t, q, p, ε)-NM-SDA-secure PKE scheme with*

$$\varepsilon \quad = \quad 2(2(N\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}) + qp \cdot 2^{-\ell} + \varepsilon_{\mathsf{mac}}) + 2(N\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}),$$

where t_{1bit} and t_{mac} are the overheads incurred by the corresponding reductions and ℓ is the length of a verification key for the MAC.

In this section, we only sketch the first part of the proof, namely that Π' is replayable NM-SDA-secure; it is based on the corresponding proof by [10] in the setting with non-parallel decryption queries. The full proof can be found in Appendix C.

We stress that an analogous statement as the one of the above theorem works for domain extension of NM-CPA, i.e., for constructing a multi-bit NM-CPA scheme out of a single-bit NM-CPA scheme. The proof is very similar to the one of Theorem 3 and therefore omitted.

Proof (sketch). The proof considers a series of N hybrid experiments. In very rough terms, the i^{th} hybrid generates the challenge ciphertext by computing an encoding $c = (c[1], \ldots, c[N])$ of the challenge plaintext and by replacing the first i bits $c[i]$ of c by random values $\tilde{c}[i]$ before encrypting

PKE Scheme $\Pi' = (KG', E', D')$

Key Generation KG'
for $i \leftarrow 1$ **to** N
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow\!\!\$\ KG$
$\mathsf{pk} \leftarrow (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$
$\mathsf{sk} \leftarrow (\mathsf{sk}_1, \ldots, \mathsf{sk}_N)$
$s \leftarrow \mathrm{Gen}$
return $(\mathsf{pk}, (\mathsf{sk}, s))$

Encryption $E'_{\mathsf{pk}}(m)$
$c = (c[1], \ldots, c[N]) \leftarrow \mathrm{Enc}(m)$
for $i \leftarrow 1$ **to** N
  $e_i \leftarrow\!\!\$\ E_{\mathsf{pk}_i}(c[i])$
return $e = (e_1, \ldots, e_N)$

Decryption $D'_{(\mathsf{sk},s)}(e)$
$(e_1, \ldots, e_N) \leftarrow e$
for $i \leftarrow 1$ **to** N
  $c[i] \leftarrow\!\!\$\ D_{\mathsf{sk}_i}(e_i)$
  if $c[i] = \perp$
    return $\perp$
$m \leftarrow \mathrm{Dec}(c[1] \cdots c[N], s)$
return m

Figure 4: *The K-bit PKE scheme $\Pi' = (KG', E', D')$ built from a 1-bit PKE scheme $\Pi = (KG, E, D)$ and a (K, N)-coding scheme with secret state* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$.

the encoding bit-wise, leading to the challenge $(e_1^*, \ldots, e_N^*)$. Moreover, when answering decryption queries $(e'_1, \ldots, e'_N)$, if $e'_j = e_j^*$ for $j \leq i$, the i^{th} hybrid sets the outcome of e'_j's decryption to be the corresponding bit $c[j]$ of the original encoding c, whereas if $e'_j \neq e_j^*$, it decrypts normally (then it decodes the resulting N-bit string normally). This follows the above intuition that a CCA-secure PKE scheme guarantees that if a decryption query is different from the challenge ciphertext, then the plaintext contained in it must have been created independently of the challenge plaintext. The indistinguishability of the hybrids follows from the security of the underlying single-bit scheme Π.

In the N^{th} hybrid, the challenge consists of N encryptions of random values. Thus, the only information about the encoding of the challenge plaintext that an attacker gets is that leaked through decryption queries. But in the N^{th} hybrid there is a 1-to-1 correspondence between decryption queries and the tamper function $f = (f[1], \ldots, f[N])$ applied to the encoding of the challenge plaintext: The case $e'_j = e_j^*$ corresponds to $f[j] = \mathsf{keep}$, and the case $e'_j \neq e_j^*$ corresponds to $f[j] = \mathsf{zero}$ or $f[j] = \mathsf{one}$, depending on whether e'_j decrypts to zero or to one. This allows a reduction to the security of the non-malleable code. □

4.3 Non-Malleable Code Construction

It remains to construct a non-malleable code (with secret state) resilient against parallel tampering. The intuition behind our construction is the following: If a code has the property (as has been the case with previous schemes secure against (non-parallel) bit-wise tampering) that changing a single bit of a valid encoding results in an invalid codeword, then the tamper function that fixes a particular bit of the encoding and leaves the remaining positions unchanged can be used to determine the value of that bit; this attack is parallelizable, and thus a code of this type cannot provide security against parallel tampering. A similar attack is also possible if the code corrects a fixed (known) number of errors. To circumvent this issue, our construction uses a—for the lack of a better word—"dynamic" error-correction bound: The secret state (which is initially chosen at random) is used to determine the positions of the encoding in which (a certain amount of) errors is tolerated.

Construction. Let $\mathbb{F}$ be a field of size $L = 2^\ell$ for some $\ell \in \mathbb{N}$, and let (E, D) be a (k, n)-ECSS (cf. Definition 1 in Section 2) with error correction 2α and secrecy τ over $\mathbb{F}$. Consider the following (K, N)-code with secret state scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ for $K = k\ell$ and $N = n\ell$:

- Gen: Choose a subset T of $[n]$ of size τn uniformly at random and output it.
- $\mathrm{Enc}(x)$ for $x \in \{0,1\}^K$: Compute $c = \mathsf{E}(x)$ and output it (as bit string).
- $\mathrm{Dec}(c, T)$ for $c \in \{0,1\}^N$: Interpret c as vector $(c_1, \ldots, c_n)$ over $\mathbb{F}$. Find a codeword $w = (w_1, \ldots, w_n)$ with $d_{\mathsf{H}}(w, c) \leq \alpha n$.[7] If no such w exists, output $\perp$. Moreover, if $w_i \neq c_i$ for some $i \in T$, output $\perp$ as well. Otherwise, decode w to $(x_1, \ldots, x_k)$ and output it (as bit string).

We prove the following theorem:

[7] Recall that the Hamming distance is over $\mathbb{F}$, i.e., it measures how many *symbols* of w and c differ.

Theorem 4. *For all $q, p \in \mathbb{N}$, coding scheme* $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is $(\mathcal{F}_{\mathsf{set}}, q, p, \varepsilon_{\mathsf{nmc}})$-non-malleable with*

$$\varepsilon_{\mathsf{nmc}} \quad = \quad p(e^{-\tau n/16+\ell s \ln 2} + e^{-\tau^2 n/4}) + pe^{-\tau^2 n},$$

where s is such that $L^s = e^{\ell s \ln 2}$ is an upper bound on the number of codewords in the range of E.

Instantiating the construction. The number $s = k + r$ of the above theorem consists of the number k of plaintext symbols and the number r of randomness symbols (both constant fractions of s). An instantiation of the above construction using algebraic geometric codes (cf. Section 2) allows to achieve constant error correction and secrecy. Moreover, the fact that the field size needed by these codes is constant, which implies that ℓ is constant, allows constant s ($s = \beta n$ for $\beta < \frac{\tau}{16\ell \ln 2}$) and thus constant rate, while $\varepsilon_{\mathsf{nmc}} = pe^{-\Omega(1)n}$. When combining the single-bit PKE scheme with our non-malleable code (cf. Theorem 3), $\varepsilon_{\mathsf{nmc}}$ becomes negligible if n is chosen, e.g., linearly in the security parameter λ of the scheme. This allows us to encrypt messages of length $K = \Omega(\lambda)$ with $n = \mathcal{O}(\lambda)$ (more generally, K-bit messages with $n = \mathcal{O}(K + \lambda)$; cf. Theorem 1).

4.4 Proof of the Non-Malleable Code Construction

For the proof of Theorem 4, fix $q, p \in \mathbb{N}$ and a distinguisher D making at most q tamper queries of size p each. Set $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$ for the rest of the proof. The goal is to show

$$\Delta^D(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \quad \leq \quad \varepsilon_{\mathsf{nmc}} \quad = \quad p(e^{-\tau n/16+\ell s \ln 2} + e^{-\tau^2 n/4}) + pe^{-\tau^2 n}$$

for a simulator sim to be determined.

On a high level, the proof proceeds as follows: First, it shows that queries that interfere with too many symbols of an encoding and at the same time do not fully fix enough symbols (called *middle* queries below) are rejected with high probability. For the remaining query types (called *low* and *high* queries), one can show that their effect on the decoding process can always be determined from the query itself and the symbols of the encoding at the positions indexed by the secret trigger set T. Since the size of T is τn, these symbols are uniformly random and independent of the encoded message, which immediately implies a simulation strategy for sim.

4.4.1 Tamper-Query Types

Recall that $f \in \mathcal{F}_{\mathsf{set}}$ can be characterized by $(f[1], \ldots, f[N])$, where $f[j] : \{0,1\} \to \{0,1\}$ is the action of f on the j^{th} bit, for $f[j] \in \{\mathsf{zero}, \mathsf{one}, \mathsf{keep}\}$, with the meaning that it either sets the j^{th} bit to 0 (zero) or to 1 (one) or leaves it unchanged (keep). Alternatively, since $N = n\ell$, f can be characterized by $(f_1, \ldots, f_n)$, where $f_i : \{0,1\}^\ell \to \{0,1\}^\ell$ is the action of f on the i^{th} symbol.[8] Abusing notation, we write $f = (f[1], \ldots, f[N]) = (f_1, \ldots, f_n)$.

Consider a tamper query $f = (f_1, \ldots, f_n) \in \mathcal{F}_{\mathsf{set}}$ and an N-bit string $c = (c_1, \ldots, c_n)$. Query f is said to *dirty* a symbol c_i of c if *at least one* of the corresponding bits of c_i is fixed to 0 or to 1; it is said to *freeze* a symbol if *all* of the corresponding bits are fixed. Note that if f freezes the i^{th} symbol, then $f_i \equiv c$ for some $c \in \{0,1\}^\ell$, in which case we set $\mathsf{val}(f_i) := c$. A tamper query f is a

- *low query* if it dirties at most τn symbols, a
- *high query* if it freezes all but τn symbols, and a
- *middle query* otherwise.

Denote by $D(f), F(f) \subseteq [n]$ the sets of indices of the symbols dirtied respectively frozen by f. Moreover, generalize the Hamming distance to compare N-bit strings and tamper queries f as follows:

$$d_{\mathsf{H}}(w, f) \quad := \quad \sum_{i \in F(f)} d_{\mathsf{H}}(w_i, \mathsf{val}(f_i)),$$

i.e., $d_{\mathsf{H}}(w, f)$ compares w to the values of the symbols fully frozen by f.[9] The distance between two codewords can be bounded in terms of their distances from a tampering query.

[8] Note that f_i still acts on each of the bits individually and may only set them to 0, to 1, or leave them unchanged.
[9] Note that $d_{\mathsf{H}}(w_i, \mathsf{val}(f_i))$ is either 0 or 1.

Proposition 5. *For any N-bit strings v, w and $f \in \mathcal{F}_{\mathsf{set}}$,*

$$d_{\mathsf{H}}(v,w) \quad \leq \quad d_{\mathsf{H}}(v,f) + d_{\mathsf{H}}(w,f) + (n - |F(f)|).$$

Proof. Let $f = (f_1, \ldots, f_n)$. Then,

$$\begin{aligned} d_{\mathsf{H}}(v,w) &= \sum_{i \in F(f)} d_{\mathsf{H}}(v_i, w_i) + \sum_{i \in [n] \setminus F(f)} d_{\mathsf{H}}(v_i, w_i) \\ &\leq \sum_{i \in F(f)} \big(d_{\mathsf{H}}(v_i, \mathsf{val}(f_i)) + d_{\mathsf{H}}(w_i, \mathsf{val}(f_i))\big) + (n - |F(f)|) \\ &= d_{\mathsf{H}}(v,f) + d_{\mathsf{H}}(w,f) + (n - |F(f)|). \end{aligned}$$

□

4.4.2 Analyzing Query Types

The following lemma states that an isolated middle query is rejected with high probability.

Lemma 6. *Let $f \in \mathcal{F}_{\mathsf{set}}$ be a middle query. Then, for any $x \in \{0,1\}^K$,*

$$\mathsf{P}[\mathrm{Dec}(f(\mathrm{Enc}(x))) \neq \perp] \quad \leq \quad e^{-\tau n/16 + \ell s \ln 2} + e^{-\tau^2 n/4}$$

where the probability is over the randomness of Enc *and the choice of the secret state T.*[10]

Proof. Fix $x \in \{0,1\}^K$ and a middle query $f = (f[1], \ldots, f[N])$. Denote by $c = (c[1], \ldots, c[N]) = (c_1, \ldots, c_n)$ and $\tilde{c} = (\tilde{c}[1], \ldots, \tilde{c}[N]) = (\tilde{c}_1, \ldots, \tilde{c}_n)$ the (random variables corresponding to the) encoding $c = \mathrm{Enc}(x)$ and the tampered encoding $\tilde{c} = f(c)$. For an arbitrary (N-bit) codeword c^*,

$$\mathbf{E}[d_{\mathsf{H}}(\tilde{c}, c^*)] \quad = \quad \sum_{i=1}^{n} \mathbf{E}[d_{\mathsf{H}}(\tilde{c}_i, c_i^*)] \quad \geq \quad \sum_{i \in T_f} \mathbf{E}[d_{\mathsf{H}}(\tilde{c}_i, c_i^*)],$$

where $T_f \subseteq [n]$ is the set containing the indices of the first τn symbols *not* completely frozen by f. Note that by the definition of middle queries, there are at least that many, i.e., $|T_f| = \tau n$.

Observe that for each $i \in T_f$, $d_{\mathsf{H}}(\tilde{c}_i, c_i^*)$ is an indicator variable with expectation $\mathbf{E}[d_{\mathsf{H}}(\tilde{c}_i, c_i^*)] \geq \frac{1}{2}$, which follows from the fact that c_i is uniformly distributed over $\{0,1\}^\ell$ and at least one bit of c_i is not frozen by f. Thus, $\mathbf{E}[d_{\mathsf{H}}(\tilde{c}, c^*)] \geq \frac{\tau n}{2}$.

Additionally, $(d_{\mathsf{H}}(\tilde{c}_i, c_i^*))_{i \in T_f}$ are independent. Therefore, using Theorem 39, for $\varepsilon > 0$

$$\mathsf{P}[d_{\mathsf{H}}(\tilde{c}, c^*) \quad < \quad (1-\varepsilon)\tau n/2] \quad \leq \quad e^{-\tau \varepsilon^2 n/4}$$

Therefore, the probability that the above does not hold for all codewords c^* is at most $e^{-\tau\varepsilon^2 n/4 + \ell s \ln 2}$ by a union bound, since there are at most $L^s = e^{\ell s \ln 2}$ codewords.

Suppose now that $d_{\mathsf{H}}(\tilde{c}, c^*) \geq (1-\varepsilon)\tau n/2$ for all codewords c^*. Then, over the choice of T,[11]

$$\mathsf{P}[\forall i \in T : d_{\mathsf{H}}(\tilde{c}_i, c_i^*) = 0] \quad \leq \quad (1 - (1-\varepsilon)\tau/2)^{\tau n} \quad \leq \quad e^{-(1-\varepsilon)\tau^2 n/2}.$$

The lemma now follows by setting $\varepsilon := \frac{1}{2}$. □

It turns out that low and high queries always result in $\perp$ or one other value.

Lemma 7. *Low queries $f \in \mathcal{F}_{\mathsf{set}}$ can result only in $\perp$ or the originally encoded message $x \in \{0,1\}^K$. High queries $f \in \mathcal{F}_{\mathsf{set}}$ can result only in $\perp$ or one other value $x'_f \in \{0,1\}^K$, which solely depends on f. Furthermore, x'_f, if existent, can be found efficiently given f.*

[10] Recall that s is such that $L^s = e^{\ell s \ln 2}$ is an upper bound on the number of codewords in the range of E.

[11] Recall that $|T| = \tau n$.

Proof. The statement for low queries is trivial, since a low query f cannot change the encoding beyond the error correction bound αn (since $\alpha \geq \tau$).

Consider now a high query f. Clearly, if w "appears" during the decoding, then $d_{\mathsf{H}}(w, f) \leq \alpha n$. Assume codewords w and w' with $d_{\mathsf{H}}(w, f) \leq \alpha n$ and $d_{\mathsf{H}}(w', f) \leq \alpha n$ exist. Then, using Proposition 5,

$$d_{\mathsf{H}}(w, w') \quad \leq \quad d_{\mathsf{H}}(w, f) + d_{\mathsf{H}}(w', f) + \tau n \quad \leq \quad \alpha n + \alpha n + \alpha n \quad \leq \quad 3\alpha n.$$

The assumed ability to correct $2\alpha n$ errors means that $3\alpha n$ is strictly less than the minimum distance of the ECSS, and therefore $w = w'$.

If $w'_f := w$ exists, it can be computed as follows from f:

1. Compute $c' \leftarrow f(0^N)$.
2. Decode c' within $2\alpha n$ (symbol) errors. If there is no codeword w'_f within $2\alpha n$, output $\perp$.
3. If $d_{\mathsf{H}}(w'_f, f) > \alpha n$, output $\perp$. Otherwise, output w'_f.

By Proposition 5, $d_{\mathsf{H}}(w'_f, f(0^N)) \leq d_{\mathsf{H}}(w'_f, f) + \underbrace{d_{\mathsf{H}}(f(0^N), f)}_{=0} + \tau n \leq 2\alpha n$, and therefore the algorithm finds w'_f. Decoding w'_f yields x'_f. □

4.4.3 Hybrids

Handling middle queries. Consider the hybrid game H_1 that behaves as $R_{\mathcal{F}}$, except that it answers all middle queries by $\perp$.

Lemma 8. $\Delta^D(R_{\mathcal{F}}, H_1) \quad \leq \quad p(e^{-\tau n/16 + \ell s \ln 2} + e^{-\tau^2 n/4})$.

The proof of Lemma 8 follows a generic paradigm, at whose core is the so-called *self-destruct lemma*, which deals with the indistinguishability of hybrids with the self-destruct property and is explained in detail in Section 6. Roughly, this lemma applies whenever the first hybrid (in this case $R_{\mathcal{F}}$) can be turned into the second one (in this case H_1) by changing ("bending") the answers to a subset (the "bending set") of the possible queries to always be $\perp$, and when additionally non-bent queries have a unique answer (cf. the statement of Lemma 19). Intuitively, the lemma states that parallelism and adaptivity do not help distinguish (much) in such cases, which allows using Lemma 6.

Proof. The lemma is proved conditioned on the message x encoded by D. To use the self-destruct lemma, note first that both $R_{\mathcal{F}}$ and H_1 answer parallel tamper queries in which each component is from the set $\mathcal{X} := \mathcal{F}$ by vectors whose components are in $\mathcal{Y} := \{0,1\}^K \cup \{\perp\}$. Moreover, both hybrids use as internal randomness a uniformly chosen element from $\mathcal{R} := \{0,1\}^\rho \times \mathcal{S}$, where ρ is an upper bound on the number of random bits used by Enc. $R_{\mathcal{F}}$ answers each component of a query $f \in \mathcal{X}$ by

$$g(f, (r, T)) \quad := \quad \mathrm{Dec}(f(\mathrm{Enc}(x; r)), T).$$

Define $\mathcal{B} \subseteq \mathcal{X}$ to be the set of all middle queries; H_1 is the $\mathcal{B}$-bending of $R_{\mathcal{F}}$ (cf. Definition 7).

Observe that queries $f \notin \mathcal{B}$ are either low or high queries. For low queries f, the unique answer is $y_f = x$, and for high queries f, $y_f = x'_f$ (cf. Lemma 7). Thus, by Lemmas 19 and 6,

$$\Delta^D(R_{\mathcal{F}}, H_1) \quad \leq \quad p \cdot \max_{f \in \mathcal{B}} \mathsf{P}[g(f, (r, T)) \neq \perp] \quad \leq \quad p(e^{-\tau n/16 + \ell s \ln 2} + e^{-\tau^2 n/4}),$$

where the probability is over the choice of (r, T). □

Handling high queries. Consider the following hybrid game H_2: It differs from H_1 in the way it decodes high queries f. Instead of applying the normal decoding algorithm to the tampered codeword $\tilde{c}$, it proceeds as follows:

1. Find w'_f (as in the proof of Lemma 7).
2. If w'_f does not exist, return $\perp$.
3. If $\tilde{c}_i = w'_{f,i}$ for all $i \in T$, return $\mathrm{Dec}(w)$. Otherwise, return $\perp$.

Lemma 9. $\Delta^D(H_1, H_2) \quad \leq \quad pe^{-\tau^2 n}$.

Proof. The lemma is proved conditioned on the message x encoded by D and the randomness r of the encoding. For the remainder of the proof, r is therefore considered fixed inside H_1 and H_2. The proof, similarly to that of Lemma 8, again uses the self-destruct lemma.

Set $\mathcal{X} := \mathcal{F}$ and $\mathcal{Y} := \{0,1\}^K \cup \{\perp\}$. However, this time, let $\mathcal{R} := \mathcal{S}$. For $f \in \mathcal{X}$ and $T \in \mathcal{R}$, define

$$g(f, T) \quad := \quad \mathrm{Dec}(\tilde{c}, T),$$

where $\tilde{c} := f(\mathrm{Enc}(x; r))$. The bending set $\mathcal{B} \subseteq \mathcal{X}$ is the set of all high queries f such that w'_f exists and $d_{\mathsf{H}}(w'_f, \tilde{c}) > \alpha n$.[12] It is readily verified that H_2 is a parallel stateless self-destruct game (cf. Definition 6) that behaves according to g, and that H_1 is its $\mathcal{B}$-bending.

Consider a query $f \notin \mathcal{B}$. If f is a low query, the unique answer is $y_f = x$; if it is a middle query, $y_f = \perp$; if it is a high query, $y_f = x'_f$ (cf. Lemma 7). Therefore,

$$\Delta^D(H_1, H_2) \quad \leq \quad \max_{f \in \mathcal{B}} \mathsf{P}[g(f, T) \neq \perp] \quad \leq \quad pe^{-\tau^2 n},$$

where the first inequality follows from the self-destruct lemma (Lemma 19) and the second one from the fact that $d_{\mathsf{H}}(x'_f, \tilde{c}) > \alpha n$ for a query in $\mathcal{B}$, and therefore the probability over the choice of T that it is accepted is at most $(1-\alpha)^{\tau n} \leq e^{-\tau^2 n}$. □

4.4.4 Simulation

By analyzing hybrid H_2, one observes that low and high queries can now be answered knowing only the query itself and the symbols of the encoding indexed by the trigger set T.

Lemma 10. *Consider the random experiment of distinguisher D interacting with H_2. There is an efficiently computable function* $\mathrm{Dec}' : \mathcal{F}_{\mathsf{set}} \times \mathcal{S} \times \{0,1\}^{\ell \tau n} \to \{0,1\}^K \cup \{\mathsf{same}, \perp\}$ *such that for any low or high query f, any fixed message x, any fixed encoding c thereof, and any output T of* Gen*,*

$$\big[\mathrm{Dec}'(f, T, (c_i)_{i \in T})\big]_{\mathsf{same}/x} \quad = \quad \mathrm{Dec}(f(c)),$$

where $[\cdot]_{\mathsf{same}/x}$ is the identity function except that same *is replaced by x and where $(c_i)_{i \in T}$ are the symbols of c specified by T.*

Proof. Consider a low query f. Due to the error correction, $\mathrm{Dec}(f(c))$ is the message originally encoded if no bit indexed by T is changed and $\perp$ otherwise. Which one is the case can clearly be efficiently computed from f, T, and $(c_i)_{i \in T}$.

For high queries f the statement follows by inspecting the definition of H_2 and Lemma 7. □

In H_2, by the τn-secrecy of the ECSS, the distribution of the symbols indexed by T is independent of the message x encoded by D. Moreover, the distribution of T is trivially independent of x. This suggests the following simulator sim: Initially, it chooses a random subset T from $\binom{[n]}{\tau n}$ and chooses τn random symbols $(c_i)_{i \in T}$. Every component f of any tamper query is handled as follows: If f is a low or a high query, the answer is $\mathrm{Dec}'(f, T, (c_i)_{i \in T})$; if f is a middle query, the answer is $\perp$. This implies:

Lemma 11. $H_2 \equiv S_{\mathcal{F}, \mathsf{sim}}$.

Proof of Theorem 4. Follows from Lemmas 8, 9, and 11 and a triangle inequality. □

[12] These are queries potentially accepted by H_2 but not by H_1.

4.5 Impossibility for Codes without State

We show that codes without secret state (as, e.g., the ones in [16, 15, 1, 18, 10, 7, 2]) cannot achieve (unconditional) non-malleability against parallel tampering. Specifically, we prove the following theorem:

Theorem 12. *Let* $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$. *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a* (K, N)*-code without secret state and noticeable rate. There exists a distinguisher* D *asking a single parallel tampering query of size* N^6 *such that, for all simulators* sim *and all* N *large enough,* $\Delta^D(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \geq 1/2$.

The above impossibility result requires that the rate of the code not be too small (in fact $N = o(2^{K/6})$ suffices, see Appendix B for the exact parameters). The distinguisher D is inefficient, so it might still be possible to construct a non-malleable code against parallel tampering with only computational security. We leave this as an interesting open question for future research.

Here, we outline an attack for the case where Dec is deterministic. A full proof and a generalization to the setting where Dec uses (independent) randomness for (each) decoding is in Appendix B.

Proof (sketch). Roughly, a possible attack works as follows: There exists an (inefficient) extraction algorithm that, by suitably tampering with an encoding in the real experiment $R_{\mathcal{F}}$, is able to recover the original plaintext with high probability. Since (modulo some technicalities) this is not possible in the ideal experiment $S_{\mathcal{F},\mathsf{sim}}$ (for any simulator sim), this constitutes a distinguishing attack.

For simplicity, suppose that the decoding algorithm Dec is deterministic. The extraction relies on the fact that for any position $i \in [N]$ with relevance in the decoding, there exist two codewords c'_i and c''_i with $\mathrm{Dec}(c'_i) \neq \mathrm{Dec}(c''_i)$ and differing in position i only. From the result of a tamper query fixing all but the i^{th} position to correspond with the bits of c'_i (or c''_i) one can therefore infer the value of the i^{th} bit of the encoding. This extraction is an independent process for every (relevant) position and thus parallelizable. In other words, a single parallel tamper query can be used to recover every relevant position of an encoding (from which the original message can be computed by filling the non-relevant positions with arbitrary values and applying the decoding algorithm). □

5 Construction from CPA Security

In this section we show that NM-SDA security can be achieved in a black-box fashion from IND-CPA security. Specifically, we prove that the scheme by Choi *et al.* [9] (dubbed the CDMW construction in the remainder of this section) is NM-SDA secure as is. The main difficulty is to extend their analysis to deal with adaptively chosen parallel decryption queries (with self-destruct).

The CDMW construction is presented in Section 5.1. We managed to slightly abstract away some features of the original transformation, using the notion of error-correcting secret-sharing codes (cf. Section 2) satisfying an additional property. We hope this might give a deeper understanding of the result of [9]. Below, we first recap the CDMW construction; the proof that it achieves NM-SDA security can be found in Section 5.2.

5.1 Construction

The CDMW construction first encodes a message using a randomized Reed-Solomon code, which is captured as a special case by the notion of error-correcting sharing scheme (ECSS) (cf. Section 2). For ease of description, we assume that the decoding algorithm returns not only the plaintext x but also the corresponding codeword w, i.e., $(x, w) \leftarrow \mathsf{D}(c, t)$, where $t \in \mathbb{N}$ specifies the number of errors to correct; moreover, the output is $(x, w) = (\perp, \perp)$ if c is not within distance t of any codeword.

The additional property the ECSS has to satisfy is roughly that given a certain number of symbols chosen uniformly at random and independently and a plaintext x, one can efficiently produce an encoding that matches the given symbols and has the same distribution as $\mathsf{E}(x)$. It is described in more detail in the proof of Lemma 18, where it is needed.[13]

[13] Of course, the Reed-Solomon-based ECSS from [9] has this property.

PKE Scheme $\Pi' = (KG', E', D')$

Key Generation KG'
- **for** $(b, i, j) \in \{0,1\} \times [\kappa] \times [n]$
  - $(\mathsf{pk}^b_{i,j}, \mathsf{sk}^b_{i,j}) \leftarrow KG$
- $\mathsf{PK} \leftarrow (\mathsf{pk}^b_{i,j})_{b,i,j}$
- $\mathsf{SK} \leftarrow (\mathsf{sk}^b_{i,j})_{b,i,j}$
- $T \leftarrow\!\!\$ \binom{[n]}{\tau n}$
- **return** $(\mathsf{PK}, (\mathsf{SK}, T))$

Encryption $E'_{\mathsf{PK}}(m)$
- $(c_1, \ldots, c_n) \leftarrow \mathsf{E}(m)$
- $(\mathsf{verk}, \mathsf{sigk}) \leftarrow KG^{\mathsf{ots}}$
- $(v[1], \ldots, v[\kappa]) \leftarrow \mathsf{verk}$
- **for** $(i, j) \in [\kappa] \times [n]$
  - $e_{i,j} \leftarrow E_{\mathsf{pk}^{v[i]}_{i,j}}(c_j)$
- $\mathbf{E} \leftarrow (e_{i,j})_{i,j}$
- $\sigma \leftarrow S_{\mathsf{sigk}}(\mathbf{E})$
- **return** $(\mathbf{E}, \mathsf{verk}, \sigma)$

Decryption $D'_{(\mathsf{SK},T)}(\mathbf{E}, \mathsf{verk}, \sigma)$
- **if** $V_{\mathsf{verk}}(\mathbf{E}, \sigma) = 0$
  - **return** $\perp$
- **for** $j \in T$
  - decrypt j^{th} column of $\mathbf{E}$
  - **if** *not all entries identical*
    - **return** $\perp$
- decrypt first row of $\mathbf{E}$ to c
- $(m, w) \leftarrow \mathsf{D}(c, \alpha n)$
- **if** $w = \perp$ *or* $\exists j \in T : c_j \neq w_j$
  - **return** $\perp$
- **return** m

Figure 5: *The CDMW PKE scheme Π' constructed from a CPA-secure scheme Π [9].*

Construction. Let $\Pi = (KG, E, D)$ be a PKE scheme with message space $\mathcal{M} = \{0,1\}^\ell$, and let $\Sigma = (KG^{\mathsf{ots}}, S, V)$ be a one-time signature scheme with verification keys of length κ. Moreover, let (E, D) be a $(k, n, 2\alpha, \tau)$-ECSS.

The CDMW construction (cf. Figure 5), to encrypt a plaintext $m \in \{0,1\}^{k\ell}$, first computes an encoding $(c_1, \ldots, c_n) \leftarrow \mathsf{E}(m)$ and then creates the $(\kappa \times n)$-matrix $\mathbf{C}$ in which this encoding is repeated in every row. For every entry $\mathbf{C}_{ij}$ of this matrix, there are two possible public keys $\mathsf{pk}^b_{i,j}$; which of them is used to encrypt the entry is determined by the i^{th} bit $v[i]$ of the verification key $\mathsf{verk} = (v[1], \ldots, v[\kappa])$ of a freshly generated key pair for Σ. In the end, the encrypted matrix $\mathbf{E}$ is signed using verk, producing a signature σ. The ciphertext is $(\mathbf{E}, \mathsf{verk}, \sigma)$.

The decryption first verifies the signature. Then, it decrypts all columns indexed by a set $T \subset [n]$, chosen as part of the secret key, and checks that each column consists of a single value only. Finally, it decrypts the first row and tries to find a codeword with relative distance at most α. If so, it checks whether the codeword matches the first row in the positions indexed by T. If all checks pass, it outputs the plaintext corresponding to the codeword; otherwise it outputs $\perp$.

We prove the following theorem in the subsequent section.

Theorem 13. *Let $t \in \mathbb{N}$ and Π be a $(t{+}t_{\mathsf{cpa}}, \varepsilon_{\mathsf{cpa}})$-IND-CPA-secure PKE scheme, (E, D) a $(k, n, 2\alpha, \tau)$-ECSS, and Σ a $(t{+}t_{\mathsf{ots}}, \varepsilon_{\mathsf{ots}})$-secure OTS scheme with verification-key length κ. Then, for any $q, p \in \mathbb{N}$, PKE scheme Π' is (t, q, p, ε)-NM-SDA-secure with*

$$\varepsilon \quad = \quad (1-\tau)\kappa n \cdot \varepsilon_{\mathsf{cpa}} + 2 \cdot \varepsilon_{\mathsf{ots}} + 4 \cdot p(1-\tau)^{\alpha n},$$

where t_{cpa} and t_{ots} represent the overhead incurred by corresponding reductions.

Instantiating the construction. When instantiated with the ECSS by [8] based on Reed-Solomon codes (cf. Section 2), the above bound becomes $\varepsilon = \frac{7}{8}\kappa n \cdot \varepsilon_{\mathsf{cpa}} + 2\varepsilon_{\mathsf{ots}} + 4pe^{-49/64n}$. This optimizes over the original CDMW construction by a factor of $\Omega(\kappa)$ (assuming constant rate k/n). The ECSS could also be instantiated with algebraic geometric codes (which also satisfy the additional property mentioned above); this would yield no significant asymptotic efficiency improvement, but would allow to potentially encrypt shorter plaintexts.

5.2 Security Proof

5.2.1 Overview

The proof follows the original one by [9]. The main change is that one needs to argue that, unless they contain invalid ciphertexts, adaptively chosen parallel queries do not allow the attacker to obtain useful information, in particular on the secret set T. This is facilitated by using the *self-destruct lemma* again (cf. Section 6). The proof proceeds in three steps using two hybrid games H_b and H'_b:

- The first hybrid H_b gets rid of signature forgeries for the verification key used to create the challenge ciphertext. The indistinguishability of the hybrid from $G_b^{\Pi', \mathsf{nm\text{-}sda}}$ follows the security of the OTS scheme and requires only minor modifications compared to the original proof.

- The second hybrid H'_b uses an alternative decryption algorithm. The indistinguishability of H'_b and H_b holds unconditionally; this step requires new techniques compared to the original proof.
- Finally, the distinguishing advantage between H'_0 and H'_1 is bounded by a reduction to the IND-CPA security of the underlying scheme Π; the reduction again resembles the one in [9].

5.2.2 Dealing with Forgeries

For $b \in \{0,1\}$, hybrid H_b behaves as $G_b^{\Pi',\mathsf{nm\text{-}sda}}$ but generates the signature key pair $(\mathsf{sigk}^*, \mathsf{verk}^*)$ used for the challenge ciphertext initially and rejects any decryption query $(\mathbf{E}', \sigma', \mathsf{verk}')$ if $\mathsf{verk}' = \mathsf{verk}^*$.

Lemma 14. *For $b \in \{0,1\}$, there exists a reduction $R'_b(\cdot)$ such that for all distinguishers D,*

$$\Delta^D(G_b^{\Pi',\mathsf{nm\text{-}sda}}, H_b) \quad \leq \quad \Gamma^{R'_b(D)}(G^{\Sigma,\mathsf{ots}}).$$

Proof. $R'_b(\cdot)$ is a standard reduction to the unforgeability of Σ. □

5.2.3 Alternative Decryption Algorithm

For $b \in \{0,1\}$, hybrid H'_b behaves as H_b but for the way it answers decryption queries $(\mathbf{E}', \sigma', \mathsf{verk}')$: As before, it first verifies the signature σ' and checks that each column of $\mathbf{E}'$ consists of encryptions of a single value. Then, it determines the first position i at which verk' and verk^* differ, i.e., where $v'[i] \neq v^*[i]$. It decrypts the i^{th} row of $\mathbf{E}$ and checks if there is a codeword w within distance $2\alpha n$.[14] If such w does not exist or else if w does not match the *first* row in a position indexed by T, the check fails. Otherwise, the plaintext corresponding to w is output.

Lemma 15. *For $b \in \{0,1\}$ and all distinguishers D,* $\Delta^D(H_b, H'_b) \leq 2 \cdot p(1-\tau)^{\alpha n}$.

The proof of Lemma 15 shows that the original and alternative decryption algorithms are indistinguishable not just for a single parallel query (as is sufficient for NM-CPA) but even against adaptively chosen parallel queries (with self-destruct). It is the main technical contribution of this section.

At the core of the proof is an analysis of how different types of encoding matrices $\mathbf{C}$ are handled inside the two decryption algorithms. To that end, one can define two games B and B' (below) that capture the behaviors of the original and the alternative decryption algorithms, respectively. The proof is completed by bounding $\Delta(B, B')$ (for all distinguishers) and showing the existence of a wrapper W_b such that $W_b(B)$ behaves as H_b and $W_b(B')$ as H'_b (also below). This proves the lemma since $\Delta^D(H_b, H'_b) = \Delta^D(W_b(B), W_b(B')) = \Delta^{D(W_b(\cdot))}(B, B')$.

The games B and B' behave as follows: Both initially choose a random size-τ subset of $[n]$. Then, they accept parallel queries with components of the type $(\mathbf{C}, i)$ for $\mathbf{C} \in \mathbb{F}^{\kappa \times n}$ and $i \in [\kappa]$. The answer to each component is computed as follows:

1. Both games check that all columns indexed by T consist of identical entries.
2. Game B tries to find a codeword w with distance less than αn from the *first* row (regardless of i), whereas B' tries to find w within $2\alpha n$ of row i. Then, if such a w is found, *both* games check that it matches the *first* row of $\mathbf{C}$ in the positions indexed by T.
3. If all checks succeed, the answer to the (component) query is w; otherwise, it is $\perp$.

Both games then output the answer vector and implement the self-destruct, i.e., if any of the answers is $\perp$, all *future* queries are answered by $\perp$.

Claim 16. *For $b \in \{0,1\}$ and all distinguishers D,* $\Delta^D(B, B') \leq 2 \cdot p(1-\tau)^{\alpha n}$.

Encoding matrices. Towards a proof of Claim 16, consider the following partition of the set of encoding matrices $\mathbf{C}$ (based on the classification in [9]):

1. There exists a codeword w within αn of the first row of $\mathbf{C}$, and all rows have distance at most αn.
2. (a) There exist two rows in $\mathbf{C}$ with distance greater than αn.

[14] Recall that the actual decryption algorithm always decrypts the first row and tries to find w within distance αn.

(b) The rest; in this case the first row differs in more than αn positions from any codeword.

Observe that queries $(\mathbf{C}, i)$ with $\mathbf{C}$ of type 1 are treated identically by both B and B': A codeword w within αn of the first row of $\mathbf{C}$ is certainly found by B; since all rows have distance at most αn, w is within $2\alpha n$ of row i and thus also found by B'. Furthermore, note that if $\mathbf{C}$ is of type 2b, it is always rejected by B (but not necessarily by B').

Consider the hybrids C and C' that behave as B and B', respectively, but always reject *all* type-2 queries. Since type-1 queries are treated identically, C and C' are indistinguishable. Moreover:

Claim 17. *For all distinguishers D,*

$$\Delta^D(B, C) \quad \leq \quad p(1-\tau)^{\alpha n} \qquad \textit{and} \qquad \Delta^D(C', B') \quad \leq \quad p(1-\tau)^{\alpha n}.$$

The proof of Claim 17 follows a generic paradigm, at whose core is the so-called *self-destruct lemma*, which deals with the indistinguishability of hybrids with the self-destruct property and is explained in detail in Section 6. Roughly, this lemma applies whenever the first hybrid (in this case B resp. B') can be turned into the second one (in this case C resp. C') by changing ("bending") the answers to a subset (the "bending set") of the possible queries to always be $\perp$, and when additionally non-bent queries have a unique answer (cf. the statement of Lemma 19). Intuitively, the lemma states that parallelism and adaptivity do not help distinguish (much) in such cases.

Proof. To use the self-destruct lemma, note that B, C, C', and B' all answer queries from $\mathcal{X} := \mathbb{F}^{\kappa \times n} \times [\kappa]$ by values from $\mathcal{Y} := \mathbb{F}^n$. Moreover, note that they use as internal randomness a uniformly chosen element T from the set $\mathcal{R} := \binom{[n]}{\tau n}$ of size-τn subsets of $[n]$.

Consider first B and C. Let $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ correspond to how B answers queries $(\mathbf{C}, i)$ (see above). Let $\mathcal{B}$ be the set $\mathcal{B}$ of all type-2a-queries. Then, C is its $\mathcal{B}$-bending (cf. Definition7). Observe that queries $x = (\mathbf{C}, i) \notin \mathcal{B}$ are either of type 1 or 2b. For the former, the unique answer y_x is the codeword w within αn of the first row of $\mathbf{C}$; for the latter, y_x is $\perp$. Therefore, using the self-destruct lemma (Lemma 19), for all distinguishers D,

$$\Delta^D(B, C) \quad \leq \quad p \cdot \max_{(\mathbf{C}, i) \in \mathcal{B}} \mathsf{P}[g((\mathbf{C}, i), T) \neq \perp],$$

where the probability is over the choice of T. Since type-2a queries have two rows with distance greater than αn, the probability over the choice of T that this remains unnoticed is at most $(1-\tau)^{\alpha n}$.

For the second part of the claim, consider B' and C'. Now, let $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ correspond to how B' answers queries $(\mathbf{C}, i)$ (see above again), and let $\mathcal{B}$ be the set $\mathcal{B}$ of all type-2-queries. Then, C' is the $\mathcal{B}$-bending of B'.

Note that all queries $x = (\mathbf{C}, i) \notin \mathcal{B}'$ are of type 1, and the unique answer y_x is the codeword w within $2\alpha n$ of row i of $\mathbf{C}$. Therefore, using Lemma 19 again, for all distinguishers D,

$$\Delta^D(B', C') \quad \leq \quad p \cdot \max_{(\mathbf{C}, i) \in \mathcal{B}'} \mathsf{P}[g'((\mathbf{C}, i), T) \neq \perp],$$

where the probability is again over the choice of T. Since type-2a queries have two rows with distance greater than αn and in type-2b queries the first row differs in more than αn positions from any codeword, the probability over the choice of T that this remains unnoticed is at most $(1-\tau)^{\alpha n}$. □

Proof (of Claim 16). The proof follows using the triangle inequality:

$$\Delta^D(B, B') \quad \leq \quad \Delta^D(B, C) + \Delta^D(C, C') + \Delta^D(C', B') \quad \leq \quad 2 \cdot p(1-\tau)^{\alpha n}.$$

□

WRAPPER. It remains to show that there exists a wrapper W_b such that $W_b(B)$ behaves as H_b and $W_b(B')$ as H'_b. The construction of W_b is straight forward: H_b and H'_b generate all keys and the challenge in the identical fashion; therefore, W_b can do it the same way. W_b answers decryption queries $(\mathbf{E}', \mathsf{verk}', \sigma')$ by first verifying the signature σ' and rejecting queries if σ' is invalid or if verk'

is identical to the verification key verk^* chosen for the challenge, decrypting the entire matrix $\mathbf{E}'$ to $\mathbf{C}'$ and submitting $(\mathbf{C}', i)$ to the oracle (either B or B'), where i is the first position at which verk' and verk^* differ, and decoding the answer w and outputting the result or simply forwarding it if it is $\perp$. Moreover, W_b implements the self-destruct. By inspection it can be seen that $W_b(B)$ implements the original decryption algorithm and $W_b(B')$ the alternative one.

5.2.4 Reduction to IND-CPA Security

Lemma 18. *There exists a reduction $R(\cdot)$ such that for all distinguishers D,*

$$\Delta^D(H'_0, H'_1) \quad = \quad (1-\tau)\kappa n \cdot \Delta^{D(R(\cdot))}(G_0^{\Pi,\mathsf{ind\text{-}cpa}}, G_1^{\Pi,\mathsf{ind\text{-}cpa}}).$$

Proof (sketch). The proof is a straight-forward generalization of the original proof by [9]; the only difference is that it needs to process multiple parallel decryption queries and implement the self-destruct feature appropriately. For ease of exposition, we describe the reduction to a many-public-key version of the CPA game for Π.[15]

Reduction $R(\cdot)$ initially chooses the secret set T and creates the challenge OTS key pair with verification key $\mathsf{verk}^* = (v^*[1], \ldots, v^*[\kappa])$ and all key pairs $(\mathsf{pk}_{i,j}^b, \mathsf{sk}_{i,j}^b)$ with $j \in T$ or $b \neq v^*[i]$. The remaining $(1-\tau)\kappa n$ key pairs are generated by the CPA game.

Recall that the ECSS is assumed to satisfy the following property: Given τn symbols $(c_i)_{i\in T}$ chosen uniformly at random and independently and any plaintext $x \in \mathbb{F}^k$, one can efficiently sample symbols $(c_i)_{i\notin T}$ such that $(c_1, \ldots, c_n)$ has the same distribution as $\mathsf{E}(x)$. Using this fact, $R(\cdot)$ creates the challenge for m_0 and m_1 as follows: It picks the random symbols $(c_i)_{i\in T}$ and completes them to two full encodings c_{m_0} and c_{m_1} with the above procedure, once using m_0 and once using m_1 as the plaintext. Let $\mathbf{C}_{m_0}$ and $\mathbf{C}_{m_1}$ be the corresponding matrices (obtained by copying the encodings κ times). Observe that the two matrices match in the columns indexed by T. These entries are encrypted by $R(\cdot)$, using the public key $\mathsf{pk}_{i,j}^b$ for entry (i,j) for which $b \neq v^*[i]$. Denote by $\mathbf{C}'_{m_0}$ and $\mathbf{C}'_{m_1}$ the matrices $\mathbf{C}_{m_0}$ and $\mathbf{C}_{m_1}$ with the columns in T removed. The reduction outputs $(\mathsf{chall}, \mathbf{C}'_{m_0}, \mathbf{C}'_{m_1})$ to its oracle and obtains the corresponding ciphertexts, which it combines appropriately with the ones it created itself to form the challenge ciphertext.

Finally, note that since the reduction knows all the secret keys $\mathsf{pk}_{i,j}^b$ with $b \neq v^*[i]$, it can implement the alternative decryption algorithm (and the self-destruct). □

5.2.5 Overall Proof

Proof (of Theorem 13). Let t_{cpa} be the overhead caused by reduction $R(\cdot)$ and t_{ots} the larger of the overheads caused by $R'_0(\cdot)$ and $R'_1(\cdot)$. Moreover, let D be a distinguisher with running time at most t. Using the triangle inequality, and Lemmas 14, 15, and 18,

$$\begin{aligned}
\Delta^D(G_0^{\Pi',\mathsf{nm\text{-}sda}}, G_1^{\Pi',\mathsf{nm\text{-}sda}}) \quad &\leq \quad \Delta^D(G_0^{\Pi',\mathsf{nm\text{-}sda}}, H_0) + \Delta^D(H_0, H'_0) \\
&\quad + \Delta^D(H'_0, H'_1) + \Delta^D(H'_1, H_1) + \Delta^D(H_1, G_1^{\Pi',\mathsf{nm\text{-}sda}}) \\
&\leq \quad \Gamma^{D(R'_0(\cdot))}(G^{\Sigma,\mathsf{ots}}) + 2 \cdot p(1-\tau)^{\alpha n} \\
&\quad + (1-\tau)\kappa n \cdot \Delta^{D(R(\cdot))}(G_0^{\Pi,\mathsf{ind\text{-}cpa}}, G_1^{\Pi,\mathsf{ind\text{-}cpa}}) \\
&\quad + 2 \cdot p(1-\tau)^{\alpha n} + \Gamma^{D(R'_1(\cdot))}(G^{\Sigma,\mathsf{ots}}) \\
&\leq \quad \varepsilon_{\mathsf{ots}} + 2 \cdot p(1-\tau)^{\alpha n} \\
&\quad + (1-\tau)\kappa n \cdot \varepsilon_{\mathsf{cpa}} + 2 \cdot p(1-\tau)^{\alpha n} + \varepsilon_{\mathsf{ots}}.
\end{aligned}$$

□

[15] In the many-public-key version of the CPA game, an attacker can play the CPA game for several independently generated public keys simultaneously; this is equivalent to the normal formulation by a standard hybrid argument [3].

6 A General Indistinguishability Paradigm

A recurring issue in this paper are proofs that certain self-destruct games answering successive parallel decryption/tampering queries are indistinguishable. We formalize such games as *parallel stateless self-destruct games.*

Definition 6. *An oracle U is a* parallel stateless self-destruct (PSSD) game *if*

- *it accepts parallel queries in which each component is from some set $\mathcal{X}$ and answers them by vectors with components from some set $\mathcal{Y}$,*
- *$\perp \in \mathcal{Y}$,*
- *there exists a function $g : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ such that every query component $x \in \mathcal{X}$ is answered by $g(x, r)$, where $r \in \mathcal{R}$ is the internal randomness of U, and*
- *the game self-destructs, i.e., after the first occurrence of $\perp$ in an answer vector all* further *outputs are $\perp$.*

A PSSD game can be transformed into a related one by "bending" the answers to some of the queries $x \in \mathcal{X}$ to the value $\perp$. This is captured by the following definition:

Definition 7. *Let U be a PSSD game that behaves according to g and let $\mathcal{B} \subseteq \mathcal{X}$. The* $\mathcal{B}$-bending of *U, denoted by U', is the PSSD game that behaves according to g', where*

$$g'(x, r) \quad = \quad \begin{cases} \perp & \text{if } x \in \mathcal{B}, \\ g(x, r) & \text{otherwise.} \end{cases}$$

The *self-destruct lemma* below states that in order to bound the distinguishing advantage between a PSSD and its bending, one merely needs to analyze a single, non-parallel query, provided that all non-bent queries x can only be answered by a unique value y_x or $\perp$.

Lemma 19. *Let U be a PSSD game and U' its $\mathcal{B}$-bending for some $\mathcal{B} \subseteq \mathcal{X}$. If for all $x \notin \mathcal{B}$ there exists $y_x \in \mathcal{Y}$ such that*

$$\{g(x, r) \mid r \in \mathcal{R}\} \quad = \quad \{y_x, \perp\},$$

then, for all distinguishers D,

$$\Delta^D(U, U') \quad \leq \quad p \cdot \max_{x \in \mathcal{B}} \mathsf{P}[g(x, R) \neq \perp],$$

where the probability is over the choice of R.

Proof. Fix a distinguisher D and denote by R and R' the random variables corresponding to the internal randomness of U and U', respectively. Call a value $x \in \mathcal{X}$ *dangerous* if $x \in \mathcal{B}$ and a query dangerous if it contains a dangerous value.

In the random experiment corresponding to the interaction between D and U, define the event E that the first dangerous query contains a dangerous value X with $g(X, R) \neq \perp$ and that the self-destruct has not been provoked yet. Similarly, define the event E' for the interaction between D and U' that the first dangerous query contains a dangerous value X' with $g(X', R') \neq \perp$ and that the self-destruct has not been provoked yet.[16]

Clearly, U and U' behave identically unless E resp. E' occur. Thus, it remains to bound $\mathsf{P}[E] = \mathsf{P}[E']$. To that end, note that adaptivity does not help in provoking E. For any distinguisher D, there exists a *non-adaptive* distinguisher $\tilde{D}$ such that whenever D provokes E, so does D'. D' proceeds as follows: First, it interacts with D only. Whenever D asks a non-dangerous query, D' answers every component $x \notin \mathcal{B}$ by y_x. As soon as D specifies a dangerous query, D' stops its interaction with D and sends all queries to U.

Fix all randomness in experiment $D'(U)$, i.e., the coins of D (inside D') and the randomness r of U. Suppose D would provoke E in the direct interaction with U. In such a case, all the answers by D' are equal to the answers by U, since, by assumption, the answers to components $x \notin \mathcal{B}$ in

[16] Note that the function g is the *same* in the definitions of either event.

non-dangerous queries are y_x or $\perp$ and the latter is excluded if E is provoked. Thus, whenever D provokes E, D' provokes it as well.

The success probability of non-adaptive distinguishers D is upper bounded by the probability over R that their first dangerous query provokes E, which is at most $p \cdot \max_{x \in \mathcal{B}} \mathsf{P}[g(x, R) \neq \perp]$. □

References

[1] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *ECCC*, 20:81, 2013. To appear in STOC 2014.

[2] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations and perturbations. *IACR Cryptology ePrint Archive*, 2014:841, 2014.

[3] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, pages 259–274, 2000.

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

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

[6] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, pages 565–582, 2003.

[7] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:102, 2014.

[8] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *TCC*, pages 440–464, 2014.

[9] 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.

[10] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. *IACR Cryptology ePrint Archive*, 2014:324, 2014.

[11] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In *ASIACRYPT*, pages 502–518, 2007.

[12] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, volume 1462 of *LNCS*, pages 13–25, 1998.

[13] 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.

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

[15] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *CRYPTO (2)*, pages 239–257, 2013.

[16] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[17] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, pages 465–488, 2014.

[18] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *EUROCRYPT*, pages 111–128, 2014.

[19] 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.

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

[21] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

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

[23] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, pages 426–442, 2004.

[24] Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. In *EUROCRYPT*, pages 241–254, 2003.

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

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

[27] 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.

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

PKE Scheme $\Pi' = (KG', E', D')$

Key Generation KG'
$(\mathsf{pk}, \mathsf{sk}) \leftarrow KG$
$\rho \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\mathsf{pk}' = \mathsf{pk}$
$\mathsf{sk}' = (\rho, \mathsf{sk})$
return $(\mathsf{pk}', \mathsf{sk}')$

Encryption $E'_{\mathsf{pk}'}(m)$
$e \leftarrow E_{\mathsf{pk}}(m)$
$e' = 0 \,\|\, e$
return e

Decryption $D'_{\mathsf{sk}'}(e')$
$e' = \beta \,\|\, e$
$m \leftarrow D_{\mathsf{sk}}(e)$
if $\beta = 1$
  return ρ
else
  if $m = \rho$
    return sk
  else
    return m

Figure 6: *PKE scheme* Π' *in the proof of Theorem 20*

A Relating IND-SDA and NM-CPA

Theorem 20. *IND-SDA and NM-CPA are incomparable notions:*

(i) Assume there exists a PKE scheme Π *that is NM-CPA secure. Then there exists a PKE scheme* Π' *that is NM-CPA secure but not IND-SDA secure.*

(ii) Assume there exists a PKE scheme Π *that is IND-SDA secure. Then there exists a PKE scheme* Π'' *that is IND-SDA secure but not NM-CPA secure.*

The proof follows the following intuition: Towards proving (i), an assumed non-malleable PKE scheme is modified as follows: An additional 0-bit is added to every ciphertext. If it is changed to 1 by an adversary, the decryption algorithm outputs a secret random value ρ (from a sufficiently large space) instead of the normal decryption, and when the decryption algorithm is fed with an encryption of ρ, it outputs the secret key. The modified scheme is not IND-SDA-secure since an adversary can obtain the secret key by making just two *adaptive* decryption queries. It is, however, still non-malleable since a non-adaptive adversary can only try to guess ρ.

For (ii), an IND-SDA-secure PKE scheme is modified as follows: Prior to encrypting a message, it is encoded using a code with the property that a sufficiently large fraction of the bits of the encoding are random. Similarly to above, ciphertexts have a format that allows an adversary, using the decryption oracle, to guess a bit of this encoding, where a wrong guess leads to a self-destruct. Since, by definition, an adversary can ask arbitrarily many (non-adaptively chosen) invalid ciphertexts in the NM-CPA game, he can recover the entire encoding of the message. At the same time, due to the randomness of the encoding, an adversary trying to guess the bits of the encoding sequentially is highly likely to produce a self-destruct in the process. Therefore, the modified scheme is not non-malleable, but it is still IND-SDA-secure.

Proof. We prove the two statements separately.

(i) Let $\Pi = (KG, E, D)$ be a PKE scheme that is NM-CPA secure, with message space $\mathcal{M} = \{0,1\}^\lambda$ for some value λ that is superpolynomial in the security parameter $\kappa \in \mathbb{N}$. Consider the PKE scheme $\Pi' = (KG', E', D')$, derived from Π, depicted in Figure 6. A secret key generated via KG' consists of a valid secret key for Π, together with a random string $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda$. A ciphertext generated via E' consists of a valid ciphertext generated via E, to which we append a single bit β that is usually set to 0 in honestly generated ciphertexts. Upon input a ciphertext with $\beta = 0$, the decryption algorithm D' behaves as D except that it returns the secret key in case the ciphertext decrypts to ρ; ciphertexts with $\beta = 1$ are always decrypted to ρ.

Claim 21. *There exists a distinguisher* D *such that* $\Delta^D(G_0^{\Pi',2,1}, G_1^{\Pi',2,1}) = 1$.

Proof. Consider the following distinguisher D that breaks IND-SDA security of Π' with probability 1, using two adaptive decryption queries, and without ever provoking a self-destruct. Given the challenge ciphertext e, the distinguisher creates $e_1 = 1 \,\|\, \bar{e}$ where $\bar{e} \leftarrow E_{\mathsf{pk}}(\bar{m})$ for a fixed message $\bar{m} \in \mathcal{M}$, and queries (dec, e_1); note that the answer to such query will be the value ρ (no matter what the value of

PKE Scheme $\Pi'' = (KG'', E'', D'')$

Key Generation KG''
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow KG$
- $\bar{m} \leftarrow\!\!\$\ \{0,1\}^k$
- $\mathsf{pk}'' = (\mathsf{pk}, \bar{m})$
- $\mathsf{sk}'' = \mathsf{sk}$
- **return** $(\mathsf{pk}'', \mathsf{sk}'')$

Encryption $E''_{\mathsf{pk}''}(m)$
- $c \leftarrow \mathrm{Enc}(m)$
- $e \leftarrow E_{\mathsf{pk}}(c)$
- $e'' = 0 \,\|\, 0 \,\|\, 0 \,\|\, e$
- **return** e''

Decryption $D''_{\mathsf{sk}''}(e'')$
- $e'' = \beta_1 \,\|\, \beta_2 \,\|\, \alpha \,\|\, e$
- $c \leftarrow D_{\mathsf{sk}}(e)$
- $m \leftarrow \mathrm{Dec}(c)$
- **if** $\beta_1 = 0$
  - **if** $(\beta_2 = 0) \wedge (\mathsf{ctr} = 0)$
    - **return** m
  - **else**
    - **return** $\perp$
- **else**
  - **if** $(m'[\alpha] = \beta_2)$
    - **return** $\bar{m}$
  - **else**
    - **return** $\perp$

Figure 7: *PKE scheme Π'' in the proof of Theorem 20*

$\bar{m}$ is). Given ρ, the distinguisher encrypts $e_2 \leftarrow\!\!\$\ 0 \,\|\, e_\rho$ where $e_\rho \leftarrow E_{\mathsf{pk}}(\rho)$ and queries (dec, e_2). Note that, by definition of Π', the answer to such query will be the secret key sk that can be used to decrypt the challenge ciphertext e with probability 1. □

Claim 22. *For all $p \in \mathbb{N}$, and for all distinguishers D', there exists a distinguisher D such that*

$$\Delta^D(G_0^{\Pi,1,p}, G_1^{\Pi,1,p}) \quad \geq \quad \Delta^{D'}(G_0^{\Pi',1,p}, G_1^{\Pi',1,p}) - p \cdot 2^{-\lambda}.$$

Proof. We build distinguisher D (based on D') as follows:

1. At the beginning D samples a random $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda$ (but not sk), and forwards the public key pk it receives from game $G_b^{\Pi,\mathsf{nm\text{-}cpa}}$ to D'.
2. Upon input $(\mathsf{chall}, m_0, m_1)$ from D', distinguisher D calls $(\mathsf{chall}, m_0, m_1)$ to its own challenge oracle obtaining a ciphertext e, and returns $e' := 0 \,\|\, e$ to D'.
3. Upon input the decryption query $(\mathsf{dec}, e'_1, \ldots, e'_p)$ from D', distinguisher D parses $e'_i = \beta_i \,\|\, e_i$ for all $i \in [p]$ and queries $(\mathsf{dec}, e_1, \ldots, e_p)$ to its own decryption oracle, obtaining $(m_1, \ldots, m_p)$. Hence, D proceeds as follows for each $i \in [p]$:
   - In case $\beta_i = 1$, replace m_i with $m_i := \rho$.
   - In case $m_i = \rho$, abort.

   Return $(m_1, \ldots, m_p)$.
4. Output whatever D' outputs.

Note that D perfectly simulates the key generation of Π' (this is because the value ρ is chosen uniformly and independently by D). The same holds for the simulation of the challenge ciphertext. Furthermore, the simulation of the parallel decryption query is perfect unless D aborts; thus, it suffices to bound the probability that D aborts. By the union bound, this probability is at most $p \cdot 2^{-\lambda}$. The claim follows. □

The two claims above conclude the proof of statement (i), by observing that λ and p are, respectively, super-polynomial and polynomial in κ.

(ii) Let $\Pi = (KG, E, D)$ be a PKE scheme, with message space $\mathcal{M} = \{0,1\}^n$, that is IND-SDA secure. Moreover let $(\mathrm{Enc}, \mathrm{Dec})$ be a (k, n)-encoding scheme with the property that the bits of a codeword are individually uniform and τn-wise independent (over the randomness of the encoding). Consider the PKE scheme $\Pi'' = (KG'', E'', D'')$, with message space $\mathcal{M}'' = \{0,1\}^k$, obtained from Π as depicted in Figure 7. The key generation is identical in the two schemes, except that KG'' includes

a message $\bar{m} \leftarrow\!\!\$\ \{0,1\}^k$ in the public key. The encryption algorithm first encodes the plaintext m obtaining a codeword c, encrypts c via pk, and appends to the ciphertext two bits β_1, $\beta_2 \in \{0,1\}$ and a special value $\alpha \in [n]$ (normally set to 0). The decryption algorithm simply decrypts the ciphertext and runs the decoding algorithm, in case $\beta_1 = 0$. Otherwise it checks that the bit in position α of c equals β_2; if this is the case it returns the message $\bar{m}$, else it returns $\perp$.

Claim 23. *There exists a distinguisher D'' such that $\Delta^{D''}(G_0^{\Pi'',1,n}, G_1^{\Pi'',1,n}) = 1$.*

Proof. Consider the following distinguisher D'' playing the NM-CPA game. At the beginning D'' receives the public key $(\mathsf{pk}, \bar{m})$, and issues $(\mathsf{chall}, m_0, m_1)$, for some m_0, m_1, obtaining a challenge $e'' = (0 \,\|\, 0 \,\|\, 0 \,\|\, e)$.

Next, D'' defines a sequence of n ciphertexts $e_1, \ldots, e_n$ and issues $(\mathsf{dec}, e_1, \ldots, e_n)$. For all $i \in [n]$, the i-th ciphertext e_i has a type $(1 \,\|\, \beta_{2,i} \,\|\, i \,\|\, e)$ where $\beta_{2,i}$ is a guess for $c[i]$ (i.e., the i-th bit of the codeword $c \leftarrow D_{\mathsf{sk}}(e)$).

Let $(m_1, \ldots, m_n)$ be the answer from the decryption oracle. Distinguisher D'' defines a codeword $c = (c[1], \ldots, c[n])$ where $c[i] = \beta_{2,i}$ if $m = \bar{m}$ (and $c[i] = 1 - \beta_{2,i}$ otherwise).[17] By inspection, this is exactly the codeword c encoding the challenge, and thus D'' wins the game with probability 1.[18] □

Claim 24. *For all $q \in \mathbb{N}$, and for all distinguishers D'', there exists a distinguisher D such that*

$$\Delta^{D}(G_0^{\Pi,q,1}, G_1^{\Pi,q,1}) \quad \geq \quad \Delta^{D''}(G_0^{\Pi'',q,1}, G_1^{\Pi'',q,1}) - 2^{-\tau n}.$$

Proof. We build distinguisher D (based on D'') as follows:

1. At the beginning D samples $\bar{m} \leftarrow\!\!\$\ \{0,1\}^k$, and forwards $\mathsf{pk}'' = (\mathsf{pk}, \bar{m})$ to D'', where pk is the public key it receives from its own challenger. Furthermore D samples n uniform and independent bits $b = (b_1, \ldots, b_n)$.

2. Upon input $(\mathsf{chall}, m_0, m_1)$ at the inside, D defines $c_0 \leftarrow \mathrm{Enc}(m_0)$ and $c_1 \leftarrow \mathrm{Enc}(m_1)$ and issues $(\mathsf{chall}, c_0, c_1)$ to its own challenger. Given the challenge ciphertext e, distinguisher D returns $e'' := (0 \,\|\, 0 \,\|\, 0 \,\|\, e)$ to D''.

3. For all $i \in [q]$, upon input the i-th decryption query (dec, e_i'') from D'', distinguisher D parses $e_i'' = (\beta_{1,i} \,\|\, \beta_{2,i} \,\|\, \alpha_i \,\|\, e_i)$ and proceeds as follows:

   (a) If $\beta_{1,i} = \beta_{2,i} = 0$ and $\alpha_i = 0$, issue (dec, e_i) to the decryption oracle receiving a value $c_i \in \{0,1\}^n$ and compute $m_i \leftarrow \mathrm{Dec}(c_i)$; return m_i to D''.

   (b) If $\beta_{1,i} = 0$ and $\alpha_i \neq 0$ or $\beta_{2,i} \neq 0$, output $\perp$ and self-destruct.

   (c) If $\beta_{1,i} = 1$, check whether $b[\alpha_i] = \beta_{2,i}$; if that is the case return $\bar{m}$, and otherwise return $\perp$ and self-destruct.

4. D outputs whatever D'' does.

Clearly D perfectly simulates the public key and the challenge ciphertext. As for decryption queries, the simulation for queries of type 3(a) and 3(b) is also perfect. Define the event E that D'' issues more than τn decryption queries of type 3(c), such that the guess $\beta_{2,i}$ for $b[\alpha_i]$ is correct. It easy to see that in case E does not happen, the simulation of decryption queries with type 3(c) is correct, because any subset of dimension up to τn of the codeword c corresponding to the challenge ciphertext is uniformly distributed. Thus, it suffices to bound the probability of the event E which is at most $2^{-\tau n}$. The claim follows. □

The two claims above imply statement (ii), by setting τ to be super-logarithmic in the security parameter κ.[19] □

[17] Recall that, by definition of Π'', the values $\bar{m}$ and $\perp$ are the only possible outcomes for each of the m_i.

[18] The latter holds true in case the encryption scheme and the encoding scheme have perfect correctness. However, it is straightforward to generalize the proof to the case of a small correctness error.

[19] Suitable encoding schemes $(\mathrm{Enc}, \mathrm{Dec})$ with such τ exist unconditionally [16].

B Necessity of Codes with Secret State

In this section we prove Theorem 12, stating that (unconditional) non-malleability under parallel tampering is impossible—already for the case $q = 1$—in case the coding scheme has no secret state (even if decoding is randomized). Before proving the theorem, let us give a precise definition of coding schemes with no secret state and randomized decoding.

Definition 8 (Code without secret state)**.** *A (K, N)-code is a pair of algorithms* $(\mathrm{Enc}, \mathrm{Dec})$*, where the (randomized)* encoding algorithm Enc *takes a K-bit plaintext x and outputs an N-bit encoding $c \leftarrow \mathrm{Enc}(x)$, and the (randomized)* decoding algorithm Dec *takes an N-bit encoding c and outputs a K-bit plaintext $x \leftarrow \mathrm{Dec}(c)$ or the special symbol $\perp$, indicating an* invalid encoding*.*

We say that $(\mathrm{Enc}, \mathrm{Dec})$ has correctness error ν if for all $x \in \{0,1\}^K$, it holds that $\mathrm{Dec}(\mathrm{Enc}(x)) = x$ with probability at least $1 - \nu$ over the randomness of Enc and Dec.

Let us restate Theorem 12 for the reader's convenience.

Theorem 12. *Let $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$. Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a (K, N)-code without secret state and noticeable rate. There exists a distinguisher D asking a single parallel tampering query of size N^6 such that, for all simulators* sim *and all N large enough,* $\Delta^D(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \geq 1/2$.

The proof of the above theorem follows directly by the following lemma:

Lemma 25. *Let $\mathcal{F} := \mathcal{F}_{\mathsf{set}}$. Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a (K, N)-code without secret state, and with correctness error $\nu < 1/2 - 1/N$. There exists a distinguisher D such that, for all simulators* sim*,*

$$\Delta^D(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \geq 1 - (\nu + 2/N + 2Ne^{-1/2N} + (2N^6 + 1) \cdot 2^{-K}).$$

The distinguisher D asks a single tampering query (i.e., $q = 1$) with $p = N^6$.

It remains to prove Lemma 25. The proof is non constructive, meaning that the distinguisher D depends on some "auxiliary information" that is fixed once the code is given, but might be hard to compute. However, as we show below, such auxiliary information always exists (for any code) provided that the correctness error of the code is not too large (but a good code would typically have a small correctness error).

We discuss some intuition for the proof of Lemma 25. The main idea is to define an extraction algorithm that (almost) always succeeds to extract the encoded message when it interacts with $R_{\mathcal{F}}$, but only does so with a small probability when interacting with $S_{\mathcal{F},\mathsf{sim}}$ (for any sim). For simplicity, let us first assume that decoding is deterministic and that the code has perfect correctness (i.e, $\nu = 0$). Define a position $i \in [N]$ to be *relevant* if there exists a pair of codewords (c'_i, c''_i), differing only at position i, for which decoding c'_i and c''_i leads to different values. One can show that the set of all relevant position is not empty (by correctness of the code), and is always fixed once the coding scheme is given (this is because the code has no secret state). Additionally, in order to decode any codeword $c \in \{0,1\}^N$, one needs to know only the values $c[i]$ for the relevant positions; all other values can be set to 0 and play no role in decoding a codeword.

Consider now the following distinguisher D that is given a set of pairs (c'_i, c''_i), one for each *relevant* position $i \in [N]$. At the beginning D encodes a value x, which defines a target encoding c. Next, D attempts to extract the i-th *relevant* bit of c via a tampering query $f_i \in \mathcal{F}$ that keeps the bit in position i and replaces all other values with the bits of c'_i or c''_i (recall that the two codewords only differ at position i). Since c'_i and c''_i decode to different values, D can determine with a single tampering query (of size at most N) all relevant values $c[i]$ with certainty; the non-relevant bits can be set to 0, as they play no role in decoding c. Distinguisher D outputs 1 if and only if the above extraction procedure leads to the chosen value x. Clearly, D always outputs 1 when interacting with $R_{\mathcal{F}}$. On the other hand, one can show that D almost never outputs 1 when interacting with $S_{\mathcal{F},\mathsf{sim}}$, which concludes the proof.

The impossibility proof extends to the case where the decoding is probabilistic, with correctness error $\nu > 0$. That means, in particular, that for a fixed codeword c, the value $\mathrm{Dec}(c)$ is a random variable. For each position i and a parameter $\mu \in [0,1]$, either of the following cases applies: (i) There

exists a pair of codewords (c'_i, c''_i), differing only at position i, such that for the statistical distance it holds that $\Delta(\mathrm{Dec}(c'_i), \mathrm{Dec}(c''_i) \geq \mu$; (ii) For all codewords c'_i and c''_i again differing only at position i, $\Delta(\mathrm{Dec}(c'_i), \mathrm{Dec}(c''_i) < \mu$. In case (i), position i is called relevant; in case (ii) it is called non-relevant. This distinction can then be exploited as in the deterministic case, but with two adaptations:

1. The extraction has to be repeated roughly $\Theta(\mu^{-3})$ times in order to get a good estimation of the i-th relevant bit. Since the individual decoding attempts use independent randomness, by the Hoeffding bound we should get a good estimation of that bit after a polynomial number of queries.

2. As before, the decoding is at the end computed by filling the non-relevant positions with sufficiently many 0s. Since the statistical distance for *all* strings with each of the bits flipped in the non-relevant positions varies by at most μ, the triangle inequality allows to (lower) bound the probability of actually computing the correct value.

In order to formalize the above intuition, we state a general lemma that will be useful in the sequel. Let $(\mathrm{Enc}, \mathrm{Dec})$ be a (K, N)-coding scheme without secret state. Fix a parameter $\mu \in [0, 1]$. Define a position $i \in [N]$ to be *μ-relevant* for the encoding scheme if there exists two codewords $c'_i, c''_i \in \{0,1\}^N$ that differ *only* in position i, for which $\Delta(\mathrm{Dec}(c'_i), \mathrm{Dec}(c''_i)) \geq \mu$. Let $\mathcal{R} = \mathcal{R}(\mu) \subseteq [N]$ be the set of all relevant positions.

Lemma 26. *Let $\mu \in [0, 1]$ and consider a (K, N)-coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ without secret state, and with correctness error $\nu \in [0, 1]$. The following holds:*

(i) The set $\mathcal{R}(\mu)$ is not empty, whenever $0 \leq \mu < \frac{1-2\nu}{2N}$.

(ii) Let $c' \leftarrow \mathrm{Enc}(x')$ for some $x' \in \{0,1\}^K$. For all c'' such that $c''[i] = c'[i]$ for all $i \in \mathcal{R}(\mu)$ we have that

$$\mathsf{P}[\mathrm{Dec}(c'') = x'] \geq 1 - \nu - 2N \cdot \mu.$$

Proof. We start by proving statement (i). For the sake of contradiction, assume that $\mathcal{R}(\mu)$ is empty. This means that for all $i \in [N]$, and for all possible pairs of codewords c'_i, c''_i (differing only at coordinate i) we have that $\Delta(\mathrm{Dec}(c'_i), \mathrm{Dec}(c''_i)) < \mu$. By the triangle inequality, this implies that any two codewords c', c'' satisfy $\Delta(\mathrm{Dec}(c'), \mathrm{Dec}(c'')) \leq N \cdot \mu$. Fix now two values $x', x'' \in \{0,1\}^K$, such that $x' \neq x''$. Let $c' \leftarrow \mathrm{Enc}(x')$ and $c'' \leftarrow \mathrm{Enc}(x'')$ be the corresponding encodings. We have:

$$\begin{aligned} 2N \cdot \mu &\geq \sum_{x \in \{0,1\}^K} \left| \mathsf{P}[\mathrm{Dec}(c'') = x] - \mathsf{P}[\mathrm{Dec}(c') = x] \right| \\ &\geq \mathsf{P}[\mathrm{Dec}(c'') = x''] - \mathsf{P}[\mathrm{Dec}(c') \neq x'], \end{aligned}$$

where the first inequality follows by definition of statistical distance, and the second inequality follows by the fact that $\mathsf{P}[\mathrm{Dec}(c') \neq x'] \geq \mathsf{P}[\mathrm{Dec}(c') = x'']$. Using the correctness property of the code, and our choice of $\mu < \frac{1-2\nu}{2N}$, we have obtained

$$\mathsf{P}[\mathrm{Dec}(c') \neq x'] \quad \geq \quad 1 - \nu - 2N \cdot \mu > \nu,$$

a contradiction.

We now show statement (ii). Let c', c'' be as in the statement of the lemma. First observe that, since the code has no secret state, the event that c'' could decode to something different than x' depends only on the randomness of the decoding process. Without loss of generality, assume that $c'' \neq c'$ (otherwise there is nothing to prove). By definition of $\mathcal{R}(\mu)$ and by using the triangle inequality, we get that $\Delta(\mathrm{Dec}(c'), \mathrm{Dec}(c'')) \leq |\mathcal{R}(\mu)| \cdot \mu \leq N \cdot \mu$. Hence,

$$\begin{aligned} 2N \cdot \mu &\geq \sum_{x \in \{0,1\}^K} \left| \mathsf{P}[\mathrm{Dec}(c') = x] - \mathsf{P}[\mathrm{Dec}(c'') = x] \right| \\ &\geq \mathsf{P}[\mathrm{Dec}(c') = x'] - \mathsf{P}[\mathrm{Dec}(c'') = x'] \\ &\geq 1 - \nu - \mathsf{P}[\mathrm{Dec}(c'') = x'], \end{aligned}$$

which concludes the proof. □

We now turn to the proof of Lemma 25.

Proof of Lemma 25. Fix parameters $\mu \in [0,1]$ and $\rho \in \mathbb{N}$ to be determined later. Let $\mathcal{R}(\mu) \subseteq [n]$, with $|\mathcal{R}(\mu)| := r \leq N$, the set of all μ-relevant positions, and denote with $(c'_i, c''_i)_{i \in \mathcal{R}(\mu)}$ the corresponding pairs of codewords that are ensured to exist by definition. Additionally, let $\mathcal{X}_i := \{x \in \{0,1\}^K : \mathsf{P}[\mathrm{Dec}(c'_i) = x] > \mathsf{P}[\mathrm{Dec}(c''_i) = x]\}$ be the set of all values x for which the probability that decoding c'_i leads to x is larger than than the probability that decoding c''_i leads to x; also let $p'_i := \mathsf{P}[\mathrm{Dec}(c'_i) \in \mathcal{X}_i]$ and $p''_i := \mathsf{P}[\mathrm{Dec}(c''_i) \in \mathcal{X}_i]$. We define the "auxiliary information" of the code to be:

$$\mathrm{aux} \quad := \quad \left\{\mathcal{R}(\mu), (c'_i, c''_i, \mathcal{X}_i, p'_i, p''_i)_{i \in \mathcal{R}(\mu)}\right\}.$$

Note that information aux is fixed once a particular encoding scheme $(\mathrm{Enc}, \mathrm{Dec})$ is given. This is because the code has no secret state.

Consider the following extraction algorithm $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ (with the above auxiliary information hard-coded), issuing a single parallel tampering query and outputting a value $\bar{x} \in \{0,1\}^K \cup \{\perp\}$.

1. For all $i \in \mathcal{R}(\mu)$, consider a function $f_i \in \mathcal{F}$ being specified via $(f_i[1], \ldots, f_i[N])$ where $f_i[i] = \mathsf{keep}$ and $f_i[j]$ is set to $c'_i[j]$ or $c''_i[j]$ for all other positions $j \in [N] \setminus \{i\}$ (recall that c'_i and c''_i differ only at position $i \in \mathcal{R}(\mu)$).

2. $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ defines the following parallel tampering query $(\mathsf{tamper}, (f_i, \ldots, f_i)_{i \in \mathcal{R}(\mu)})$, where we take ρ copies of each function f_i. Let $(x_{1,1}, \ldots, x_{1,\rho}, \ldots, x_{r,1}, \ldots, x_{r,\rho})$ be the answers corresponding to $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$'s query.

3. For each $i \in \mathcal{R}(\mu)$, $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ checks whether

$$\#\{x_{i,j} \in \mathcal{X}_i : \ j \in [\rho]\} \quad \geq \quad \left(\frac{p'_i}{2} + \frac{p''_i}{2}\right)\rho;$$

   if that is the case it sets $\bar{c}[i] := c'_i[i]$, and otherwise $\bar{c}[i] := 1 - c'_i[i] = c''_i[i]$. All other values $\bar{c}[i]$, with $i \notin \mathcal{R}(\mu)$, are set to 0.

4. Finally, $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ outputs $\bar{x} = \mathrm{Dec}(\bar{c})$.

Claim 27. *Let $0 < \mu < \frac{1-2\nu}{2N}$, and $\rho \in \mathbb{N}$. For all $x \in \{0,1\}^K$, let $\bar{x}$ be the value returned by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ after tampering with $c \leftarrow \mathrm{Enc}(x)$. Then, $\mathsf{P}[\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ outputs $\bar{x} = x] \geq 1 - (\nu + 2N/\mu + 2Ne^{-1/2\rho\mu^2})$.*

Proof. Note that by Lemma 26, the range of values allowed for the parameter μ ensures that the set $\mathcal{R}(\mu)$ is not empty. We need to show that $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ extracts the bits $c[i]$ of the target encoding of x in all μ-relevant positions, with high enough probability (the other bits are set to 0 by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ and play almost no role in decoding $\bar{c}$).

Let $\tilde{c}_i = f_i(c)$ be the i-th tampered codeword as defined by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ in step 2. For all $j \in [\rho]$, and for all $i \in \mathcal{R}(\mu)$, let $X'_{i,j}$ (resp. $X''_{i,j}$) be a binary random variable which equals 1 if and only if $x_{i,j}$ happens to be in the set $\mathcal{X}_i$, where $x_{i,j}$ is the value obtained by decoding $\tilde{c}_i = c'_i$ (resp. $\tilde{c}_i = c''_i$). Note that, for each $i \in \mathcal{R}(\mu)$, the random variables $X'_{i,j}$ (resp. $X''_{i,j}$) are independent and follow the Bernoulli distribution with parameter p'_i (resp. p''_i). Let $X'_i = \sum_{j=1}^{\rho} X'_{i,j}$ and similarly $X''_i = \sum_{j=1}^{\rho} X''_{i,j}$. We bound the probability that $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ extracts the i-th relevant bit of the target encoding incorrectly as follows:

$$\begin{aligned}
\mathsf{P}[\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho} \text{ sets } \bar{c}[i] \text{ incorrectly}] \quad &\leq \quad \mathsf{P}[\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho} \text{ sets } \bar{c}[i] \text{ incorrectly} | \tilde{c}_i = c'_i] \\
&\qquad + \mathsf{P}[\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho} \text{ sets } \bar{c}[i] \text{ incorrectly} | \tilde{c}_i = c''_i] \\
&= \quad \mathsf{P}[X'_i \leq (p'_i + p''_i)\rho/2] + \mathsf{P}[X''_i \geq (p'_i + p''_i)\rho/2] \\
&\leq \quad \mathsf{P}[X'_i \leq (p'_i - \mu/2)\rho] + \mathsf{P}[X''_i \geq (p''_i + \mu/2)\rho] \\
&\leq \quad 2e^{-1/2\rho\mu^2},
\end{aligned}$$

where the first inequality comes from the fact that $\mu \leq p'_i - p''_i$, and the second inequality follows from the Hoeffding inequality.[20] By the union bound, we get that the probability of extracting *at least one position incorrectly* is bounded by $2Ne^{-1/2\rho\mu^2}$, so the probability that $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ extracts all μ-relevant positions correctly is at least $1 - 2Ne^{-1/2\rho\mu^2}$.

Finally, we observe that the probability that decoding $\bar{c}$ as defined by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ in step 3 leads to the correct value x is at least $1 - \nu - 2N/\mu$ (see Lemma 26). The claim follows. □

Define now the following distinguisher D (depending on $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$). The distinguisher chooses $x \in \{0,1\}^K$ and issues (encode, x) to the oracle it has access to. Then it fixes $\mu = N^{-2}$ and $\rho = N^5$, and it lets $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ interact with that oracle. In case at least one of the values $\{x_{i,j}\}_{i \in \mathcal{R}(\mu), j \in [\rho]}$ seen by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ in step 2 happens to be equal to x, D outputs 0. Otherwise, whenever $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ returns a value $\bar{x} \in \{0,1\}^K \cup \{\perp\}$, D outputs 1 if and only if $\bar{x} = x$.

Claim 28. $\mathsf{P}[D(R_{\mathcal{F}}) = 1] \geq 1 - (\nu + 2/N + 2Ne^{-1/2N} + N^6 \cdot 2^{-K+1})$.

Proof. The statement follows by definition of D. Note that D outputs 1 provided that $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ returns the value x chosen uniformly at random by D, except in case it happens that one of the answers $x_{i,j}$ corresponding to $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$'s tampering query happens to be equal to x. Denote by E the latter event. Since we already have a lower bound on the probability of success of $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ (see Claim 27), it suffices to bound the probability of the event E.

Note that for each tampering function $f_i \in \mathcal{F}$ specified by $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$, we have that the tampered codeword $\tilde{c}_i = f_i(c)$ overwrites all the bits with known values but the bit in position i, which is copied from the original codeword c. It follows that the output of such tampering query can decrease the entropy of x by at most one, and hence the probability that each $x_{i,j}$ happens to be equal to x is at most 2^{-K+1}. By a union bound, we get that $\mathsf{P}[E] \leq r\rho \cdot 2^{-K+1} \leq N\rho \cdot 2^{-K+1}$.

The claim now follows by the choice of $\mu = N^{-2}$ and $\rho = N^5$, which in particular ensures that $\mathcal{R}(\mu)$ is not empty (see Lemma 26) by our assumption that $\nu < 1/2 - 1/N$. □

Claim 29. *For all simulators* sim*, we have that* $\mathsf{P}[D(S_{\mathcal{F},\mathsf{sim}}) = 1] \leq 2^{-K}$.

Proof. We first observe that we can assume, without loss of generality, that sim never outputs same. In fact, if sim happens to return same for any of the queries f_i defined by D (via $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$), then the oracle $S_{\mathcal{F},\mathsf{sim}}$ would replace the output of the simulator with x. By definition in this case D outputs 1 with probability 0.

On the other hand, assume that sim never outputs same. In such a case the view of $\mathrm{Ext}^{\mathrm{aux}}_{\mu,\rho}$ is completely independent of x, and so the extractor will output $\bar{x} = x$ with probability at most 2^{-K}. The claim follows. □

Lemma 25 follows by combining Claim 28 and Claim 29. □

C Composing Non-Malleable Codes, PKE, and MACs

This section contains the proof of Theorem 3, which is split into two parts. First, we prove that the PKE scheme Π' resulting from combining a single-bit PKE Π and a non-malleable code with secret state $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ as shown in Section 4.2 (cf. Figure 4) is replayable NM-SDA secure (NM-RSDA); the proof is based on the corresponding one in [10] for IND-SDA security. Then, we show that a MAC-based transformation suggested by [6] to obtain IND-CCA security from IND-RCCA security also works in our setting, i.e., the transformation applied to Π' yields a fully NM-SDA secure PKE scheme Π''. All results in this section also apply to NM-CPA security. Overall, we prove:

[20] The general form is: Let $X_1, \ldots, X_n$ be i.i.d. with $X_i \sim \mathrm{Be}(p)$. Then, for $X := \sum_i X_i$, and for any $\varepsilon \in (0,1]$,

$$\mathsf{P}[X \leq (p-\varepsilon)n] \leq e^{-2\varepsilon^2 n} \quad \text{and} \quad \mathsf{P}[X \geq (p+\varepsilon)n] \leq e^{-2\varepsilon^2 n}.$$

Theorem 3. *Let $q, p \in \mathbb{N}$ and Π be a $(t + t_{\mathsf{1bit}}, q, p, \varepsilon_{\mathsf{1bit}})$-NM-SDA-secure 1-bit PKE scheme, (T, V) a $(t + t_{\mathsf{mac}}, 1, qp, \varepsilon_{\mathsf{mac}})$-MAC, and $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ a $(\mathcal{F}_{\mathsf{set}}, q, p, \varepsilon_{\mathsf{nmc}})$-non-malleable (K, N)-code with secret state. Then, Π'' is (t, q, p, ε)-NM-SDA-secure PKE scheme with*

$$\varepsilon \quad = \quad 2(2(N\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}) + qp \cdot 2^{-\ell} + \varepsilon_{\mathsf{mac}}) + 2(N\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}),$$

where t_{1bit} and t_{mac} are the overheads incurred by the corresponding reductions and ℓ is the length of a verification key for the MAC.

C.1 Replayable NM-SDA Security

The notion of *replayable CCA* security was introduced by Canetti *et al.* [6] to deal with the artificial strictness of CCA security. Intuitively, it potentially allows an attacker to maul a target ciphertext into one that decrypts to the *same* message.[21] This idea carries over seamlessly to the definition of NM-SDA security; the corresponding distinguishing game $G_b^{\Pi,\mathsf{nm\text{-}rsda}}$ is obtained by changing $G_b^{\Pi,\mathsf{nm\text{-}sda}}$ (cf. Figure 2) to answer test whenever a ciphertext $e^{(j)}$ decrypts to m_0 or m_1 (instead of only when $e^{(j)}$ equals the challenge ciphertext).

Definition 9. *A PKE scheme Π is* replayable (t, q, p, ε)-NM-SDA-secure (NM-RSDA) *if for all distinguishers D with running time at most t and making at most q decryption queries of size at most p each,*

$$\Delta^D(G_0^{\Pi,\mathsf{nm\text{-}rsda}}, G_1^{\Pi,\mathsf{nm\text{-}rsda}}) \quad \leq \quad \varepsilon.$$

C.2 Non-Malleable Codes and PKE

In this section we show that the PKE scheme Π' is NM-RSDA if the underlying single-bit scheme Π is NM-SDA secure. Concretely, we prove:

Theorem 30. *Let $q, p \in \mathbb{N}$ and Π be a $(t_{\mathsf{rsda}} + t_{\mathsf{1bit}}, q, p, \varepsilon_{\mathsf{1bit}})$-NM-SDA-secure 1-bit PKE scheme and let $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ be $(\mathcal{F}_{\mathsf{set}}, q, p, \varepsilon_{\mathsf{nmc}})$-non-malleable. Then, Π' is $(t_{\mathsf{rsda}}, q, p, \varepsilon_{\mathsf{rsda}})$-NM-RSDA-secure PKE scheme with*

$$\varepsilon_{\mathsf{rsda}} \quad = \quad 2(N\varepsilon_{\mathsf{1bit}} + \varepsilon_{\mathsf{nmc}}),$$

where t_{1bit} represents the overhead incurred by the reductions.

The proof follows directly from the following lemma:

Lemma 31. *For $b \in \{0, 1\}$ and $i \in [N]$, there exist reductions $R_{b,i}(\cdot)$ and $W_b(\cdot)$ such that for all distinguishers D,*

$$\Delta^D(G_0^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}) \quad \leq \quad \sum_{b,i} \Delta^{D(R_{b,i}(\cdot))}(G_0^{\Pi,\mathsf{nm\text{-}sda}}, G_1^{\Pi,\mathsf{nm\text{-}sda}}) + \sum_b \Delta^{D(W_b(\cdot))}(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}),$$

where sim is the simulator for the non-malleable code. Moreover, all reductions preserve the number q and the size p of the queries.

Proof. Let t_{1bit} be the maximal occurring overhead caused by the reductions $R_{b,i}(\cdot)$. Fix a distinguisher D having running time t_{rsda} and making at most q decryption queries of size at most p. Due to the preservation property of the reductions, $\Delta^{D(R_{b,i}(\cdot))}(G_0^{\Pi,\mathsf{nm\text{-}sda}}, G_1^{\Pi,\mathsf{nm\text{-}sda}}) \leq \varepsilon_{\mathsf{1bit}}$ and $\Delta^{D(W_b(\cdot))}(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}) \leq \varepsilon_{\mathsf{nmc}}$, which completes the proof. □

Towards a proof of Lemma 31, consider the following hybrids for $b \in \{0, 1\}$ and $i \in [N]$: $H_{b,i}$ proceeds as $G_b^{\Pi',\mathsf{nm\text{-}rsda}}$ except that the challenge query $(\mathsf{chall}, m_0, m_1)$ and decryption queries $(\mathsf{dec}, e^{(1)}, \ldots, e^{(p)})$ are handled differently:

[21] In contrast, full CCA security requires that any ciphertext created by the attacker (other than the target ciphertext) decrypt to an independent message.

- **Challenge query:** The first i bits of the encoding $c = (c[1], \ldots, c[N])$ of m_b are replaced by uniformly random and independent bits. The resulting N-bit string is then encrypted bit-wise (as done by E'). This results in the challenge ciphertext $e^* = (e_1^*, \ldots, e_N^*)$.
- **Decryption query:** Every component $e^{(l)} = (e'_1, \ldots, e'_N)$ is answered as follows: Hybrid $H_{b,i}$ computes $c' = (c'[1], \ldots, c'[N])$, where

$$c'[i] \quad = \quad \begin{cases} c[j] & \text{if } e'_j = e_j^*, \text{ and} \\ D_{\mathsf{sk}_j}(e'_j) & \text{otherwise.} \end{cases}$$

  Then, $H_{b,i}$ outputs $\mathrm{Dec}(c', s)$ as the answer to the component of the decryption query.[22]

Let $H_{b,0} := G_b^{\Pi',\mathsf{nm\text{-}rsda}}$.

Lemma 32. *For all $b \in \{0,1\}$ and $i \in [N]$, there exist a reduction $R_{b,i}(\cdot)$ such that for all D*

$$\Delta^D(H_{b,i-1}, H_{b,i}) \quad = \quad \Delta^{D(R_{b,i}(\cdot))}(G_0^{\Pi,\mathsf{nm\text{-}sda}}, G_1^{\Pi,\mathsf{nm\text{-}sda}}).$$

Proof. Fix b and i. Hybrid $R_{b,i}(\cdot)$ works as follows: Initially, it generates the secret state $s \leftarrow \mathrm{Gen}$ and $N-1$ key pairs $(\mathsf{pk}_j, \mathsf{sk}_j)$ for $j \in [N] \setminus \{i\}$, obtains pk_i (but not sk_i) from the oracle (from $G_0^{\Pi,\mathsf{nm\text{-}sda}}$ or $G_0^{\Pi,\mathsf{nm\text{-}sda}}$), and outputs $\mathsf{pk} := (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$. When it receives $(\mathsf{chall}, m_0, m_1)$, it computes an encoding $c = (c[1], \ldots, c[N]) \leftarrow \mathrm{Enc}(m_b)$. Then, it chooses i random bits $\tilde{c}[1], \ldots, \tilde{c}[i]$ and computes

$$e_j^* \quad = \quad \begin{cases} E_{\mathsf{pk}_j}(\tilde{c}[j]) & \text{for } j < i, \text{ and} \\ E_{\mathsf{pk}_j}(c[j]) & \text{for } j > i. \end{cases}$$

Moreover, it outputs $(\mathsf{chall}, c[i], \tilde{c}[i])$ to its oracle and obtains a ciphertext e_i^*. It finally returns $e^* = (e_1^*, \ldots, e_N^*)$.

When $R_{b,i}(\cdot)$ receives a (parallel) decryption query, for each component $e' = (e'_1, \ldots, e'_N)$ it proceeds as follows: For $j \neq i$, it computes $c'[j]$ as $H_{b,i}$ does. Moreover, if $e'_i = e_i^*$, it sets $c'[i] \leftarrow c[i]$. Otherwise, it outputs (dec, e'_i) to its oracle and obtains the answer $c'[i]$.[23] Then, it computes $m' \leftarrow \mathrm{Dec}(c')$. The answer to the component of the decryption query is m', unless $m' \in \{m_0, m_1\}$, in which case the it is test. If one of the component answers is $\perp$, $R_{b,i}(\cdot)$ implements the self-destruct mode, i.e., answers all future queries by $\perp$.

Consider $R_{b,i}(G_0^{\Pi,\mathsf{nm\text{-}sda}})$ and $H_{b,i-1}$. Both generate the public key in the same fashion. As to the challenge ciphertext, the first $i-1$ ciphertext components e_j generated by $R_{b,i}(G_0^{\Pi,\mathsf{nm\text{-}sda}})$ are encryptions of random bits $\tilde{c}[j]$, whereas the i^{th} and the remaining components are encryptions of the corresponding bits of an encoding of m_b (generated by $G_0^{\Pi,\mathsf{nm\text{-}sda}}$ and $R_{b,i}(\cdot)$, respectively). The same is true for $H_{b,i-1}$. The answer to a decryption query component sent to $R_{b,i}(G_0^{\Pi,\mathsf{nm\text{-}sda}})$ is $\mathrm{Dec}(c')$ for $c' = (c'[1], \ldots, c'[N])$, where $c'[j] = D_{\mathsf{sk}_j}(e'_j)$ unless $j < i$ and $e'_j = e_j$, in which case $c'[j] = \tilde{c}[j]$. Again, the same holds for $H_{b,i-1}$. Moreover, both $R_{b,i}(G_0^{\Pi,\mathsf{nm\text{-}sda}})$ and $H_{b,i-1}$ answer test if $\mathrm{Dec}(c') \in \{m_0, m_1\}$. Thus, they behave identically.

$R_{b,i}(G_1^{\Pi,\mathsf{nm\text{-}sda}})$ and $H_{b,i}$ are compared similarly. This concludes the proof. □

Lemma 33. *For $b \in \{0,1\}$, there exists a wrapper $W_b(\cdot)$ such that*

1. *$W_b(R_{\mathcal{F}})$ behaves as $H_{b,N}$, and*
2. *$W_0(S_{\mathcal{F},\mathsf{sim}})$ and $W_1(S_{\mathcal{F},\mathsf{sim}})$ behave identically.*

Proof. Wrapper $W_b(\cdot)$ works as follows: Initially, it generates N key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for $i \in [N]$ and outputs $\mathsf{pk} := (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$. When it receives $(\mathsf{chall}, m_0, m_1)$, it picks N random values $\tilde{c}[1], \ldots, \tilde{c}[N]$, computes $e_i^* \leftarrow\!\!\$\, E_{\mathsf{pk}}(\tilde{c}[i])$ for $i = 1, \ldots, N$, and returns $e = (e_1, \ldots, e_N)$. Additionally, it outputs (encode, m_b) to its oracle.

[22] Assume here and below that $\mathrm{Dec}(c') = \perp$ if any of the bits $c'[j]$ equal $\perp$.

[23] In fact, it is important that $R_{b,i}(\cdot)$ output a single parallel decryption query containing all e'_i for the individual components; but it is less cumbersome to describe how individual components are handled.

When it gets a (parallel) decryption query, for every component $e' = (e'_1, \ldots, e'_N)$, it proceeds as follows: First, it creates a tamper query $f = (f[1], \ldots, f[N])$ where

$$f[i] \quad = \quad \begin{cases} \mathsf{zero} & \text{if } e'_i \neq e^*_i \text{ and } D_{\mathsf{sk}_i}(e'_i) = 0, \\ \mathsf{one} & \text{if } e'_i \neq e^*_i \text{ and } D_{\mathsf{sk}_i}(e'_i) = 1, \text{ and} \\ \mathsf{keep} & \text{if } e'_i = e^*_i. \end{cases}$$

Then, it outputs (tamper, f) to its oracle and obtains an answer x'. If $x' \in \{m_0, m_1\}$, the answer to the component query test.[24] Otherwise, it is x'. If one of the component answers is $\perp$, $W_b(\cdot)$ implements the self-destruct mode, i.e., answers all future queries by $\perp$.

Consider $W_b(R_{\mathcal{F}})$ and $H_{b,N}$. Both generate the public key in the same fashion. Furthermore, in either case, the challenge ciphertext consists of N encryptions of random bits. Finally, both answer a decryption query by applying the same tamper function to an encoding of m_b before decoding it. When the decoding of the tampered codeword results in m_0 or m_1, both answer test. Therefore, they behave identically.

Due to the fact that test is output when a decryption query results in m_0 or m_1, the observable behavior is the same in $W_0(S_{\mathcal{F},\mathsf{sim}})$ and $W_1(S_{\mathcal{F},\mathsf{sim}})$.[25]

□

Proof (of Lemma 31). Lemma 31 follows using a triangle inequality. Specifically, for any distinguisher D,

$$\begin{aligned} \Delta^D(G_0^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}) \quad &\leq \quad \sum_i \Delta^D(H_{0,i-1}, H_{0,i}) + \Delta^D(W_0(R_{\mathcal{F}}), W_0(S_{\mathcal{F},\mathsf{sim}})) \\ &\quad + \Delta^D(W_1(S_{\mathcal{F},\mathsf{sim}}), W_1(R_{\mathcal{F}})) + \sum_i \Delta^D(H_{1,i-1}, H_{1,i}) \\ &\leq \quad \sum_{b,i} \Delta^D(R_{b,i}(G_0^{\Pi,\mathsf{nm\text{-}sda}}), R_{b,i}(G_1^{\Pi,\mathsf{nm\text{-}sda}})) \\ &\quad + \sum_b \Delta^{D(W_b(\cdot))}(R_{\mathcal{F}}, S_{\mathcal{F},\mathsf{sim}}). \end{aligned}$$

□

C.3 From Replayable to Full NM-SDA Security

Message authentication codes. A *message authentication code (MAC)* is a pair of algorithms (T, V), where the tagging algorithm T takes as input a message m and a key K and outputs a tag $t \leftarrow T_K(m)$ and where the verification algorithm V takes a key K, a message m, and a tag t and outputs a bit $V_K(m, t)$.

MAC security is defined using the following game G^{mac} played by an adversary A: Initially, the game chooses a random key K. Then, A gets access to a tagging oracle, which returns a tag $t \leftarrow T_K(m)$ when given a message m, and to a verification oracle, which outputs $V_K(m, t)$ when given a message m and a tag K. The adversary wins the game if he submits to the verification oracle a pair (m, t) that is not a query-answer pair for the tagging oracle and for which $V_K(m, t) = 1$.

Definition 10. *A MAC Σ is* (t, u, v, ε)-secure *if for all adversaries A with running time at most t, making at most u tag queries and at most v verification queries, $\Gamma^A(G^{\mathsf{mac}}) \leq \varepsilon$.*

An NM-RSDA-secure PKE scheme $\Pi' = (KG', E', D')$ and a message-authentication code (MAC) (V, T) can be combined as follows to obtain a fully NM-SDA-secure scheme Π'' [6]: The key generation remains unchanged. To encrypt a message m, the new encryption algorithm first chooses a key K for the MAC and computes an encryption $e_1 \leftarrow E'_{\mathsf{pk}}(m \,\|\, K)$ and $e_2 \leftarrow T_K(e_1)$; the ciphertext is (e_1, e_2).

[24] Again, $W_b(\cdot)$ needs to output a single parallel tamper query containing the tamper functions f for the individual components.

[25] This is where the proof reflects that Π' is only NM-RSDA secure.

The new decryption algorithm decrypts e_1 to (m, K) and verifies the tag e_2. If the tag is valid, the decryption algorithm outputs m; otherwise, it outputs $\perp$.

Theorem 34. *Let Π' be a $(t + t_{\mathsf{rsda}}, q, p, \varepsilon_{\mathsf{rsda}})$-NM-RSDA secure PKE scheme and (V, T) a $(t + t_{\mathsf{mac}}, \varepsilon_{\mathsf{mac}})$-secure MAC. Then, Π'' is a (t, q, p, ε)-NM-SDA-secure PKE scheme for*

$$\varepsilon \quad \leq \quad 2(\varepsilon_{\mathsf{rsda}} + qp \cdot 2^{-\ell} + \varepsilon_{\mathsf{mac}}) + \varepsilon_{\mathsf{rsda}},$$

where ℓ is the length of the MAC key.

The theorem follows from the following lemma:

Lemma 35. *For $b \in \{0, 1\}$ there exist reductions $R_b(\cdot)$, $R'(\cdot)$, and $R''_b(\cdot)$, such that for all distinguishers D,*

$$\begin{aligned}\Delta^D(G_0^{\Pi'',\mathsf{nm\text{-}sda}}, G_1^{\Pi'',\mathsf{nm\text{-}sda}}) \quad \leq \quad & \sum_b \left(\Delta^{D(R_b(\cdot))}(G_b^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}) + qp \cdot 2^{-\ell} + \Gamma^{D(R''_b(\cdot))}(G^{\mathsf{mac}})\right) \\ & + \Delta^{D(R'(\cdot))}(G_0^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}).\end{aligned}$$

where ℓ is the length of the MAC key. Moreover, reductions $R_b(\cdot)$ and $R'(\cdot)$ preserve the number q and the size p of the queries, and reduction $R''_b(\cdot)$ asks a single tag query and qp verification queries.

Proof. Let t_{rsda} be the maximal occurring overhead caused by the reductions $R_b(\cdot)$, $R'(\cdot)$ and t_{mac} that by the reductions $R''_b(\cdot)$. Fix a distinguisher D having running time t_{rsda} and making at most q decryption queries of size at most p. Due to the preservation properties of the above reductions, the distinguishing advantages on $G_b^{\Pi',\mathsf{nm\text{-}rsda}}$ are at most $\varepsilon_{\mathsf{rsda}}$ and $\Gamma^{D(R''_b(\cdot))}(G^{\mathsf{mac}})$ is at most $\varepsilon_{\mathsf{mac}}$. □

Hybrid 1. The first hybrid H_b captures the fact that the MAC key in the challenge ciphertext is computationally hidden; it differs from $G_b^{\Pi'',\mathsf{nm\text{-}sda}}$ as follows:

- It generates the challenge ciphertext using two independent MAC keys K^* and K, i.e., $(e_1^*, e_2^*) \leftarrow (E'_{\mathsf{pk}}(m_b \,\|\, K^*), T_K(e_1^*))$.
- When answering (components of parallel) decryption queries $(e'_1, e'_2) \leftarrow (E'_{\mathsf{pk}}(m_b \,\|\, K'), e'_2)$, if $K' = K^*$, the tag is verified using K instead of K^*.

Lemma 36. *There exists a reduction $R_b(\cdot)$ such that for all distinguishers D asking at most q parallel queries of size at most p each,*

$$\Delta^D(G_b^{\Pi'',\mathsf{nm\text{-}sda}}, H_b) \quad \leq \quad \Delta^{D(R_b(\cdot))}(G_0^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}) + qp \cdot 2^{-\ell},$$

where ℓ is the length of the MAC key.

Proof (sketch). Initially, reduction $R_b(\cdot)$ outputs (to D) the public key obtained from its oracle. When it gets $(\mathsf{chall}, m_0, m_1)$, it outputs $((\mathsf{chall}, m_b \,\|\, K, m_b \,\|\, K^*))$ to its oracle and gets a response e_1^*. Then, it computes $e_2^* \leftarrow T_K(e_1^*)$ and outputs (e_1^*, e_2^*). As long as no self-destruct has occurred, $R_b(\cdot)$ answers (components of parallel) decryption queries (e'_1, e'_2) (different from the challenge ciphertext) as follows: It outputs (dec, e'_1) to its oracle. If the answer is test, H_b verifies the tag e'_2 with K and returns m_b to D if it is valid. If the answer is $m' \,\|\, K'$, H_b verifies the tag with K' and returns m' if it is valid.

By inspection one observes that $R_b(G_0^{\Pi',\mathsf{nm\text{-}rsda}})$ behaves as $G_b^{\Pi'',\mathsf{nm\text{-}sda}}$ unless D asks a query (e'_1, e'_2) where e'_1 is an encryption of a message concatenated with K^*; however, since the view of D when interacting with $R_b(G_0^{\Pi',\mathsf{nm\text{-}rsda}})$ is independent of K^*, the probability of this event is bounded by $2^{-\ell}$.

On the other hand, observe that $R_b(G_1^{\Pi',\mathsf{nm\text{-}rsda}})$ behaves exactly as hybrid H_b. □

Hybrid 2. The second hybrid H'_b behaves as H_b except that queries (e'_1, e'_2) where e'_1 contains K^* are always rejected.

Lemma 37. *There exists a reduction $R''_b(\cdot)$ such that for all distinguishers D,*

$$\Delta^D(H_b, H'_b) \quad \leq \quad \Gamma^{D(R''_b(\cdot))}(G^{\mathsf{mac}}).$$

Proof. $R''_b(\cdot)$ is a standard reduction to the strong unforgeability of the MAC. □

Reduction to NM-RSDA. Distinguishing $G_0^{\Pi',\mathsf{nm\text{-}rsda}}$ and $G_1^{\Pi',\mathsf{nm\text{-}rsda}}$ can now be reduced to distinguishing H'_0 and H'_1.

Lemma 38. *There exists a reduction $R'(\cdot)$ such that for all distinguishers D,*

$$\Delta^D(H'_0, H'_1) \quad = \quad \Delta^{DR'(\cdot)}(G_0^{\Pi',\mathsf{nm\text{-}rsda}}, G_1^{\Pi',\mathsf{nm\text{-}rsda}}).$$

Proof (sketch). The reduction translates between the NM-SDA game for Π'' and the NM-RSDA game for Π', using the fact that decryption queries for which the first component contains K^* can be rejected. In particular, when the NM-RSDA game outputs test, a ciphertext can be rejected. □

D Miscellaneous

D.1 Chernoff Bound

We make use of the following Chernoff bound.

Theorem 39. *Let $X_1, \ldots, X_n$ be i.i.d. with $X_i \sim \mathrm{Be}(p_i)$. Then, for $X := \sum_i X_i$ and $\mu := \sum_i p_i$,*

$$\mathsf{P}[X \leq (1-\varepsilon)\mu] \quad \leq \quad e^{-\mu\varepsilon^2/2}$$

for any $\varepsilon \in (0, 1]$.