Short Non-Malleable Codes from Related-Key Secure Block Ciphers

Serge Fehr[1], Pierre Karpman[2†] and Bart Mennink[1,3]

[1] CWI, Amsterdam, The Netherlands
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP[‡], LJK, 38000 Grenoble, France
[3] Digital Security Group, Radboud University, Nijmegen, The Netherlands

serge.fehr@cwi.nl, pierre.karpman@univ-grenoble-alpes.fr, b.mennink@cs.ru.nl

Abstract. A non-malleable code is an unkeyed randomized encoding scheme that offers the strong guarantee that decoding a *tampered* codeword either results in the *original* message, or in an *unrelated* message. We consider the simplest possible construction in the computational split-state model, which simply encodes a message m as $k||\mathcal{E}_k(m)$ for a uniformly random key k, where $\mathcal{E}$ is a block cipher. This construction is comparable to, but greatly simplifies over, the one of Kiayias *et al.* (ACM CCS 2016), who eschewed this simple scheme in fear of related-key attacks on $\mathcal{E}$. In this work, we prove this construction to be a strong non-malleable code as long as $\mathcal{E}$ is (i) a pseudorandom permutation under leakage and (ii) related-key secure with respect to an arbitrary but fixed key relation. Both properties are believed to hold for "good" block ciphers, such as AES-128, making this non-malleable code very efficient with short codewords of length $|m| + 2\tau$ (where τ is the security parameter, *e.g.*, 128 bits), without significant security penalty.

Keywords: Non-malleable code, split-state tampering model, related-key security, block cipher.

1 Introduction

1.1 Non-Malleable Codes

Non-malleable codes (NMCs) were introduced by Dziembowski, Pietrzak and Wichs in 2010 [DPW10]. They allow the encoding and decoding of messages in such a way that decoding a *tampered* (modified) codeword results in a message that is either the one that was originally encoded, or one uncorrelated with it. NMCs offer a very different perspective from classical error-correcting codes: when correction is not possible, decoding is *required to fail catastrophically*, outputting an unrelated message and not, *e.g.*, one that is close to the original in terms of some metric. While this is a very strong, potentially hard to fulfill condition, there is on the other hand no particular requirement that an NMC should be able to detect an error even on a single bit of a codeword; NMCs can thus in principle be designed for error patterns for which there can be no classical code, such as ones allowing to arbitrarily rewrite the entire codeword.

In a slightly more formal way, an NMC is a randomized mapping Enc from a message space to a codeword space such that (i) there is an efficient (possibly deterministic) decoding procedure Dec such that $\mathrm{Dec}(\mathrm{Enc}(m)) = m$ for every message m with probability 1, and (ii) decoding a *tampered* codeword $\hat{c} := \mathrm{T}(\mathrm{Enc}(m))$ for a function T yields either

[†] Part of this work was done when the author was at CWI.
[‡] Institute of Engineering Univ. Grenoble Alpes

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2018, No. 1, pp. 336–352
DOI:10.13154/tosc.v2018.i1.336-352

m itself or a message $\hat{m} := \mathrm{Dec}(\hat{c})$ that is uncorrelated with m. Although NMCs can be designed for very powerful error patterns, some restrictions on the set of allowed tampering functions are still necessary: indeed, it is clear that if $c := \mathrm{Enc}(m)$ is for instance tampered with the function $\mathrm{T} : x \mapsto \mathrm{Enc}(\mathrm{Dec}(x) + 1)$ with a fixed choice for the randomness of Enc, we get that $\mathrm{Dec}(\hat{c}) = m + 1$, which is strongly correlated with m. A common and well-accepted restriction for the allowed set of tampering functions is to restrict to *bipartite* functions $\mathrm{T} = (\mathrm{T_L}, \mathrm{T_R})$, which act *independently* on two distinct parts of the encoding, *i.e.*, which act as $\mathrm{T}(c) = \mathrm{T_L}(c_\mathrm{L}) || \mathrm{T_R}(c_\mathrm{R})$ given that c is of the form $c_\mathrm{L} \| c_\mathrm{R}$. When restricting the class of tampering functions to such bipartite (or, more generally, to multi-partite) functions, which act independently on different parts of the encoding, one typically speaks of *split-state* NMCs. In the extreme case of functions that must act independently on every bit of the encoding, this recovers the notion of *bitwise independent tampering* as originally introduced by Dziembowski *et al.* [DPW10].

In this work, as we explain in more detail further below, we consider a very simple split-state NMC construction that is purely based on a symmetric-key primitive (namely a block cipher). We then rigorously study its security by reducing it to different notions from the symmetric-cryptography realm.

1.2 NMCs and Tamper-Resilient Cryptography

In typical security analyses in theoretical cryptography, the algorithm under consideration is modeled as a black-box with which a potential attacker can interact only via the system's input- and output-interface. This idealistic approach leads to very nice results, but it is well understood that it does not capture reality very well. Indeed, its limitations have been impressively demonstrated by innumerable *side-channel* and *fault* attacks on real-life implementations of various cryptographic schemes.

One possible approach to address physical attacks is to actively protect the implementation of a scheme in a way that prevents the attacks or makes them too expensive to carry out, so that an algorithm hopefully does behave like a black box from the attacker's perspective. Another approach is to give up on the "black-box model" and allow the attacker some (limited) access to and/or some control over the internal state of an execution, and come up with *new schemes* that can then be proven secure in this new model. This approach is referred to as *leakage-* or *tamper-resilient cryptography*, when it aims to protect against side-channel or fault attacks respectively.

Whichever path is taken, one cannot expect the countermeasures to be universally effective in any scenario; for fault attacks, any countermeasure is designed with respect to a certain *fault model*, which specifies the type of fault (*e.g.*, transient or permanent, bit flips, random or to a constant), its granularity (*e.g.*, on a bit or on a byte), and the time and location where it occurs (*e.g.*, on an intermediate variable while performing a specific computation or on a long-term secret stored in memory). For instance, some fault attacks on the AES have been using faulty computations [BS03, PQ03], while some attacks on RSA signatures rely on faulting a secret or public parameter stored in memory [BNNT11, BM16].

One of the core tools for designing tamper-resilient cryptographic schemes are NMCs. This should not come as a surprise given their functionality: if we for instance envision the secret key of a cryptographic algorithm to be NMC-encoded, then any tampering (to which the NMC is resistant) will either not affect the decoded secret at all, in which case "black-box" arguments are enough to guarantee security; or it will make it unrelated to the original value, in which case the output of the algorithm cannot be used by an adversary to learn anything meaningful. This property makes NMCs directly applicable to protect against fault attacks on long-term parameters stored in memory, such as the ones of Brier *et al.* [BNNT11] and Bhattacharya and Mukhopadhyay [BM16]. We refer to [DPW10] for the details on how to generically use NMCs to protect against this kind of attacks that exploit a *tamper-prone memory*, assuming tamper- and leakage-proof computation.

In contrast, attacks that exploit faulty *computations*, such as the ones of Blomer and Seifert [BS03] and Piret and Quisquater [PQ03], are harder to deal with and NMCs do not offer an immediate solution; additional (or different) techniques are then necessary to achieve tamper-resilience. It is worth noticing that some proposed *ad hoc* countermeasures against faulty computations do bear some resemblance with NMCs. This is the case of *infective* countermeasures, some of which have been proposed to protect the AES [LRT12, GST12, TBM14] (although with mixed results, as these have all been attacked in various fault models [BG13, TBM14, BG16]). These countermeasures typically work by interleaving (in a secret way) the rounds of a real AES computation with ones performed on dummy states for which faults are easy to detect. If a fault is detected, a dummy ciphertext is released instead of the original one, preventing the leakage of any information about the real key.

Beyond their application to tamper-resilient cryptography, NMCs have been proven to be useful in various other contexts as well. For instance, Coretti *et al.* showed how NMCs can be used to extend single-bit public-key encryption schemes to multiple-bit ones [CMTV15]. Agrawal *et al.* achieved a similar goal in the context of non-malleable commitments, building a multi-bit commitment from a single-bit one [AGM+15], whereas Goyal *et al.* demonstrated how to use NMCs in the split-state model to directly build simple and efficient non-malleable commitments schemes [GPR16].

1.3 Related Work

Since their original formalization, many constructions of NMCs have been proposed, both in the information-theoretic (*e.g.*, [DPW10, DKO13, ADL14]) and computational (*e.g.*, [LL12, AAG+16, KLT16]) setting. A number of constructions also aim for additional properties, such as having efficient *refreshing* mechanisms that allow to update an existing codeword to a new one (decoding to the same message) without the need for decoding [FN17]. Some constructions also specifically focus on computationally restricted adversaries, either in terms of time or space complexity (*e.g.*, [FHMV17, BDSKM17]).

The construction that is the most relevant to our work is the computationally-secure split-state NMC by Kiayias *et al.* [KLT16] from CCS 2016. This NMC features the so far shortest codeword lengths, given by $|m| + 9\tau + 2\log^2(\tau)$ or $|m| + 18\tau$ (depending on how a leakage-resilient authenticated-encryption scheme is instantiated in their construction), where $|m|$ is the size of the message and τ is the security parameter. One down side of their scheme is that security is proven under the *knowledge of exponent assumption* (KEA), which is considered non-standard because it is not falsifiable [Nao03]. Their scheme encodes m into $(k, r)||(\mathcal{E}_k(m), \mathcal{H}_z(r, k))$ for random k and r, where $\mathcal{E}$ is a symmetric encryption scheme and $\mathcal{H}_z$ an instance of a "1-more extractable hash function". This can be seen as a refinement of a previous computationally-secure construction from Liu and Lysyanskaya [LL12] given by $\mathrm{Enc} : m \mapsto \mathsf{sk}||(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m), \pi)$, where the symmetric encryption replaces a public-key scheme and where the hash of the key replaces a NIZK proof π of existence of a secret key allowing to invert $\mathcal{E}_{\mathsf{pk}}(\cdot)$.

1.4 Contribution

We consider and analyze the *simplest* construction of a computationally secure NMC in the split-state model based on a block cipher $\mathcal{E} : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{M}$, denoted $\mathsf{RKNMC}[\mathcal{E}]$, where the message m is simply encoded as

$$\mathrm{Enc} : m \mapsto k||\,\mathcal{E}_k(m)\,, \tag{1}$$

for a uniformly random key $k \in \{0,1\}^\kappa$. This NMC generates short codewords of size $|m|+\kappa$ while providing $\tau = \kappa/2$ bits of security. As such, the construction we consider reduces the

codeword length down to $|m| + 2\tau$ compared to Kiayias *et al.*'s scheme. Decoding is done in the obvious way, and both encoding and decoding are depicted in Figure 1. The goal of this work is to understand the security of $\mathsf{RKNMC}[\mathcal{E}]$ in terms of basic security properties of the underlying block cipher $\mathcal{E}$.

It is rather obvious that the standard security notion of a block cipher — (strong) pseudorandom permutation security — is insufficient to imply security of this NMC: adversarial tampering may change the key k used for encryption to any other k', and in this case standard (S)PRP security does not offer any guarantee anymore. This is the reason why previous work (like [KLT16] or [LL12]) end up with more complicated schemes, where the inclusion of a hash function or NIZK proof thwarts subtle tampering of the encryption key k.

In this work, we take a different approach: instead of strengthening the construction, and this way making it more complex and expensive, we show that (1) is secure as long as the underlying block cipher satisfies certain security properties. Concretely, we give a characterization of the security of this NMC in terms of basic *sufficient* security properties of $\mathcal{E}$: we show that $\mathsf{RKNMC}[\mathcal{E}]$ is secure if

(i) $\mathcal{E}$ is *related-key secure* with respect to *any single* related-key function (Definition 4), appropriately defined to deal with output-predictable functions such as constant mappings, and

(ii) $\mathcal{E}$ has a *graceful degradation* in standard PRP security if limited information about its key is leaked (Definition 3).

Our bound is tight, as we demonstrate with a matching generic attack. It is fair to say that a violation of either of these properties would be considered a weakness for an actual block cipher (although some concrete designs do not try to achieve any sort of related-key security and thus would not fulfill (i)). In that sense, our results show that for any "good" block cipher $\mathcal{E}$, the construction in (1) is secure. For instance, it seems reasonable to conjecture that, say, AES-128 [NIS01] or SHACAL-2 [HN01] satisfy (i) and (ii). As generic attacks w.r.t. to the definitions of (i) and (ii) exist with complexity the square-root of the keyspace, we thus have obtained a simple and efficient NMC construction that offers 64 (resp. 256) bits of security unless AES-128 (resp. SHACAL-2) exhibits some unexpected behaviour.

1.5 Discussion

Our NMC significantly improves over the best previous work of Kiayias *et al.*. We reach much shorter codewords for any given security level (*e.g.*, $256 + |m|$ compared to $1250 + |m|$ for $\tau = 128$) with an efficient construction that can be readily instantiated. Our security model also compares favorably as we do not require a non-falsifiable assumption such as KEA for [KLT16], neither do we need a *common reference string* (CRS) for initialization purposes. A comparison of RKNMC with related work is provided in Table 1.

From a symmetric-cryptographic point of view, RKNMC illustrates the relevance of designing block ciphers meeting strong security requirements, such as resistance to related-key attacks. Admittedly, constructions benefiting from a related-key secure block cipher $\mathcal{E}$ are already known — one can for instance design a tweakable block cipher $\widetilde{\mathcal{E}}$ from an XOR-related-key secure block cipher $\mathcal{E}$ as $\widetilde{\mathcal{E}}(k, t, \cdot) = \mathcal{E}(k \oplus t, \cdot)$ [LRW11, BK03] — but often competitive PRP-based alternatives exist. Our non-malleable code construction can be securely instantiated with a block cipher if it is secure with respect to (i) and (ii) above; avoiding these conditions imposes the resort to more technical and more expensive schemes, such as that of Kiayias *et al.*. This could further motivate the design of related-key secure block ciphers which can attain non-trivial security with respect to (i). We believe that designs insecure with respect to (ii) should already be considered insecure in practice, but

Table 1: Comparison of computationally-secure non-malleable codes in the split-state tampering model. The data is taken from [KLT16], which also suggested competitive instantiations of previous schemes. The security parameter is denoted by τ, and the message length by $|m|$. CRS stands for common reference string model.

Reference	Codeword length	CRS	Assumption
[ADL14] + [AAG+16]	$\|m\| + \mathcal{O}(\tau^7)$	No	authenticated encryption
[LL12] + [GS08] + [NS12] + [AAG+16]	$\|m\| + \mathcal{O}(\tau^2)$	Yes	leakage-resilient PKE + robust NIZK
[KLT16]	$\|m\| + 9\tau + 2\log^2(\tau)$	Yes	1-time leakage-resilient AE + KEA
[KLT16]	$\|m\| + 18\tau$	Yes	1-time leakage-resilient AE + KEA
This work	$\|m\| + 2\tau$	No	PRP-with-leakage + fixed-related-key

our construction gives a further justification to why this should be the case. This latter definition might also be useful in different contexts, when a certain notion of resistance to *weak keys* needs to be formalized.

2 Preliminaries

2.1 Notation and Basic Definitions

For any finite non-empty set $\mathcal{M}$, $\mathrm{Perm}(\mathcal{M})$ is the set of all permutations. We denote by $x \xleftarrow{\$} \mathcal{X}$ the uniform random sampling of x from $\mathcal{X}$. In our security definitions below, an adversary $A(x)$ is an algorithm that is given some input x and, in some cases, has oracle access to a randomized oracle $\mathcal{O}(b)$, and eventually outputs a bit $\hat{b} \in \{0,1\}$. The complexity of A will be controlled by its running time t and the number q of oracle queries it makes.

We recall the notion of ε-universal hash function families [CW77].

Definition 1 (Universal Hash Function)**.** Let $\mathcal{H} : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{N}$ be a hash function family. $\mathcal{H}$ is called *ε-universal* if for any distinct $x, x' \in \mathcal{M}$ and a uniformly random key $k \xleftarrow{\$} \{0,1\}^\kappa$, the probability that $\mathcal{H}(k,x) = \mathcal{H}(k,x')$ is smaller than ε. Formally,

$$\forall x, x' \neq x \in \mathcal{M},\ \Pr\big[\mathcal{H}(k,x) = \mathcal{H}(k,x') : k \xleftarrow{\$} \{0,1\}^\kappa\big] \leq \varepsilon\,.$$

2.2 (Strong) Non-Malleable Codes

We recall here the formal definition of strong non-malleable codes for split-state tampering in the computational setting. It states that an adversary should not be able to distinguish the tampered decoding of two messages m_0 and m_1 of his choice, except in the trivial case where tampering has no effect. Indeed, in this degenerate case, it would be easy to distinguish one from the other. This is formalized through the introduction of a special "same" symbol that is returned whenever the decoded codeword is equal to the input message. This special case is consistent with the overall goal of non-malleable codes, as executing a scheme with its original secret is not per se supposed to leak any information about it.

Strong non-malleability is further parameterized by the set of tampering functions allowed to the adversary; in the present case, these are functions that independently act on the two parts of a codeword.

Definition 2 (Strong non-malleable code [DPW10] in the split-state tampering model)**.** Let $\mathsf{NMC} = (\mathrm{Enc}, \mathrm{Dec})$ be given by an encoding function

$$\mathrm{Enc} : \{0,1\}^{\kappa} \times \{0,1\}^{n} \to \{0,1\}^{\ell_L} \times \{0,1\}^{\ell_R}$$

and a corresponding decoding function

$$\mathrm{Dec} : \{0,1\}^{\ell_L} \times \{0,1\}^{\ell_R} \to \{0,1\}^{n},$$

such that $\mathrm{Dec}(\mathrm{Enc}(k,m)) = m$ for all k and m. For any m and any bipartite function

$$\mathrm{T} = \mathrm{T_L} \| \mathrm{T_R} : \{0,1\}^{\ell_L} \times \{0,1\}^{\ell_R} \to \{0,1\}^{\ell_L} \times \{0,1\}^{\ell_R}$$

which maps $(c_L, c_R) \mapsto (\mathrm{T_L}(c_L), \mathrm{T_R}(c_R))$, we consider the tampering experiment that chooses $k \xleftarrow{\$} \{0,1\}^{\kappa}$ and outputs

$$\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{NMC}}(m) := \dot{\mathrm{Dec}}^{\mathrm{Enc}_k(m)} \circ \mathrm{T} \circ \mathrm{Enc}_k(m),$$

where $\dot{\mathrm{Dec}}^{c}$ is a modified decoding algorithm s.t. $\dot{\mathrm{Dec}}^{c}(c') = \mathrm{Dec}(c')$ unless $c = c'$, in which case $\dot{\mathrm{Dec}}^{c}(c')$ outputs a special "same" symbol $\maltese$. The *SNMC advantage* of NMC is then defined as

$$\mathbf{Adv}^{\text{snmc-s}}_{\mathsf{NMC}}(t) := \max_{A,\mathrm{T}} \max_{m_0,m_1} \left| \Pr[A(\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{NMC}}(m_0)) = 1] - \Pr[A(\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{NMC}}(m_1)) = 1] \right|,$$

where the first max is over all bipartite functions $\mathrm{T} = \mathrm{T_L} \| \mathrm{T_R}$ that can be computed in time t and over all algorithms A with running time t, and the second is over all $m_0, m_1 \in \{0,1\}^{n}$.

2.3 Block Ciphers

A block cipher is a mapping $\mathcal{E} : \{0,1\}^{\kappa} \times \mathcal{M} \to \mathcal{M}$ such that for all $k \in \{0,1\}^{\kappa}$, $\mathcal{E}_k(\cdot) := \mathcal{E}(k,\cdot)$ is invertible. Its inverse is denoted $\mathcal{D}_k$. The classical security models for block ciphers are "pseudorandom permutation (PRP)" and "strong pseudorandom permutation (SPRP)" security: the former applies to settings where the adversary can only learn forward evaluations of the cipher, while in the latter case it may also learn inverse evaluations. In our work, we will use a slightly tweaked version of PRP security, see Section 2.3.1. In addition, we will require the block cipher to be related-key secure in a slightly specialized setting, see Section 2.3.2. For both notions, it is reasonable to believe that they are satisfied by a "good" block cipher.

2.3.1 PRP Security with Leakage

We consider the following variant of the standard PRP security notion where the adversary is allowed to learn some information about the key. Since we admit an arbitrary "leakage function", we have to be careful to have a definition that is still meaningful, *i.e.*, that cannot be broken *generically*. For instance, if the adversary learns the identity function applied to the key then he can trivially distinguish "real" encryptions from "ideal" encryptions.

Definition 3 (PRP-with-leakage security)**.** Let $\mathcal{E} : \{0,1\}^{\kappa} \times \mathcal{M} \to \mathcal{M}$ be a block cipher. Let $\varphi : \{0,1\}^{\kappa} \to \{0,1\}^{\kappa}$ be an arbitrary function, and let $A_{q,t}$ be an adversary with access to an oracle $\mathcal{O}_{\varphi}(b)$ and which makes at most q queries and operates in at most t time. We define the PRP-with-leakage advantage of $\mathcal{E}$ by:

$$\mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(q,t) = \max_{A_{q,t}} \max_{\varphi} \left| \Pr\left[A^{\mathcal{O}_{\varphi}(0)}_{q,t}() = 1\right] - \Pr\left[A^{\mathcal{O}_{\varphi}(1)}_{q,t}() = 1\right] \right|, \tag{2}$$

where $\mathcal{O}_{\varphi}(b)$ acts as follows:

It chooses a uniformly random key $k \xleftarrow{\$} \{0,1\}^\kappa$ and *aborts* and answers any query with $\perp$ if k is such that $\varphi(k)$ can be guessed with probability at most $p \leq 2^{-\kappa/2}$ (in other words, if $\#\{k' | \varphi(k') = \varphi(k)\} \leq 2^{\kappa/2}$). Otherwise, it announces $\varphi(k)$ to $A_{q,t}$ and answers any encryption query $m \in \mathcal{M}$ with $\mathcal{E}_k(m)$ in case $b = 0$, and with $\mathfrak{E}(m)$ in case $b = 1$, where $\mathfrak{E} \xleftarrow{\$} \mathrm{Perm}(\mathcal{M})$ is a uniformly random permutation.

Phrased slightly differently, the oracle $\mathcal{O}_\varphi(b)$ aborts if the chosen key k happens to have the property that $\Pr[\varphi(K)=\varphi(k)] \leq 2^{-\kappa/2}$ for a uniformly distributed K. As

$$2^{-\kappa} = \Pr[\,K=k\,] \geq \Pr[\,K=k \wedge \varphi(K)=k'] = \Pr[\,K=k \mid \varphi(K)=k'\,] \cdot \Pr[\,\varphi(K)=k'\,]$$

for every k and k', this ensures that if k is such that $\mathcal{O}_\varphi(b)$ does not abort then $\Pr[K = k \mid \varphi(K) = \varphi(k)] \leq 2^{-\kappa/2}$, *i.e.*, it is still hard to guess k even when given $\varphi(k)$. Thus, it is meaningful to ask for a small PRP advantage even though we have no restriction on the leakage function φ. Given that the above bound on guessing the key from the leakage can be met with a suitable choice of φ, one may expect that a good cipher achieves PRP-with-leakage advantage $\mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(q,t)$ of about $t \cdot 2^{-\kappa/2}$.

Relation With Weak-Key Classes. The security notion of PRP-with-leakage can be understood as a way of formalizing the requirement that $\mathcal{E}$ should not have (many) large "weak-key classes". Informally, one may define a weak-key class as a subset $\mathcal{K}_{weak} \subset \{0,1\}^\kappa$ of the keyspace of $\mathcal{E}$ such that the cipher can be attacked more efficiently (w.r.t. the size of $\mathcal{K}_{weak}$) knowing that $k \in \mathcal{K}_{weak}$, that is $\mathbf{Adv}^{\text{prp}}_{\mathcal{E}/\mathcal{K}_{weak}}(q,t) \gg t/\#\mathcal{K}_{weak}$ (where $\mathcal{E}/\mathcal{K}_{weak}$ denotes $\mathcal{E}$ with its keyspace restricted to $\mathcal{K}_{weak}$).

The requirement that $\mathcal{E}$ has no large weak-key classes would then correspond to the fact that for any sufficiently large subset $\mathcal{K}$ of $\{0,1\}^\kappa$, $\mathcal{E}$ with its keyspace restricted to $\mathcal{K}$ is still a "good" PRP w.r.t. the size of $\mathcal{K}$. In other words, the security of $\mathcal{E}$ degrades gracefully when one only considers subsets of its keyspace.

There is a direct relation between this requirement and $\mathcal{E}$ being secure w.r.t. PRP-with-leakage: learning the leakage $\varphi(k)$ restricts k to a subset of the keyspace of size at least $2^{\kappa/2}$, thence security for any φ follows from such a form of weak-key resistance, when "sufficiently large" is taken to be "larger than $2^{\kappa/2}$". More explicitly, given any partition $\mathcal{K}_1 \cup \mathcal{K}_2 \ldots \cup \mathcal{K}_m$ of $\{0,1\}^\kappa$, one may define an indicator function φ such that for any i, for any $k \in \mathcal{K}_i$, $\varphi(k)$ returns a fixed representative of $\mathcal{K}_i$; note that if all the $\mathcal{K}_i$'s are larger than $2^{\kappa/2}$, the game of Definition 3 never aborts when played with this function. This lets an adversary obtain "for free" the information of which subspace the key is from. Yet if $\mathcal{E}$ restricted to any of the $\mathcal{K}_i$'s is a good PRP, the adversary still cannot attack with advantage much better than $\approx t/(\min_i \#\mathcal{K}_i) \leq t/2^{\kappa/2}$, which is the best attainable security of any cipher w.r.t. Definition 3. On the other hand, if most of the restrictions of $\mathcal{E}$ to $\mathcal{K}_i$ result in a "bad" PRP, one may hope to attack with an advantage much larger than $t/2^{\kappa/2}$. As Definition 3 maximizes the adversarial advantage over the choice of φ, good security indeed requires that no such bad partition of the keyspace into *many* large weak subspaces exists.

Under this light, we believe that being secure for PRP-with-leakage is in fact a reasonably standard assumption, as it is common for practitioners to regard a block cipher as broken if it possesses even only *one* large weak-key class (see, *e.g.*, [LMR15] for an example). Smaller classes may be less of a concern as a random key is unlikely to be drawn from one, and the presence of such classes is indeed allowed in Definition 3.

2.3.2 Fixed-Related-Key Security

We define a notion of related-key security that is sufficient for our purpose of proving security of the proposed non-malleable code. In comparison to the definition suggested in,

say, [BK03], our definition is more constrained in that we allow en-/decryption queries under *one* related key only (next to the original key), but it is more liberal in that the related key can be specified by an *arbitrary* function φ, without any restriction (like being "unpredictable"), while still allowing to attain a meaningful level of security. A consequence of the latter is that we have to be careful that "real" and "ideal" encryptions are not trivially distinguishable by a clever choice of φ, *e.g.*, a constant function.

Definition 4 (F-RK security[1])**.** Let $\mathcal{E} : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{M}$ be a block cipher, let $\varphi : \{0,1\}^\kappa \to \{0,1\}^\kappa$ be an arbitrary function, and let $A_{q,t}$ be an adversary with access to an oracle $\mathcal{O}_\varphi(b)$ which makes at most q queries and operates in at most t time. We define the F-RK advantage of $\mathcal{E}$ by:

$$\mathbf{Adv}^{\mathrm{frk}}_{\mathcal{E}}(q,t) = \max_{A_{q,t}} \max_{\varphi} \left| \Pr\left[A^{\mathcal{O}_\varphi(0)}_{q,t}() = 1\right] - \Pr\left[A^{\mathcal{O}_\varphi(1)}_{q,t}() = 1\right] \right|, \tag{3}$$

where $\mathcal{O}_\varphi(b)$ acts as follows:
It chooses a uniformly random key $k \xleftarrow{\$} \{0,1\}^\kappa$ and *aborts* and answers $\perp$ if $\varphi(k)$ happens to be a value that can be guessed with probability $> 2^{-\kappa/2}$. Otherwise, it proceeds as follows. If $b = 0$ then any query of the form $(\mathtt{enc}, \mathtt{real}, m)$, $(\mathtt{enc}, \mathtt{modified}, m)$, $(\mathtt{dec}, \mathtt{real}, c)$ or $(\mathtt{dec}, \mathtt{modified}, c)$ is answered with $\mathcal{E}_k(m)$, $\mathcal{E}_{\varphi(k)}(m)$, $\mathcal{D}_k(c)$ and $\mathcal{D}_{\varphi(k)}(c)$, respectively, and if $b = 1$ then with $\mathfrak{E}_k(m)$, $\mathfrak{E}_{\varphi(k)}(m)$, $\mathfrak{D}_k(c)$ and $\mathfrak{D}_{\varphi(k)}(c)$, respectively, where $\mathfrak{E}_k$ and $\mathfrak{E}_{\varphi(k)}$ are random and independent permutations from $\mathrm{Perm}(\mathcal{M})$ if $k \neq \varphi(k)$, and random and equal if $k = \varphi(k)$, and $\mathfrak{D}_k$ and $\mathfrak{D}_{\varphi(k)}$ are the respective inverses.

We note that in comparison to Definition 3, here the oracle $\mathcal{O}_\varphi(b)$ aborts if the chosen key k happens to be so that $\Pr[\varphi(K) = \varphi(k)] > 2^{-\kappa/2}$ for a uniformly random K. This ensures that if k is so that $\mathcal{O}_\varphi(b)$ does not abort then $\varphi(k)$ is hard to guess. Hence, again, aborting as $\mathcal{O}_\varphi(b)$ does is necessary (and sufficient) for a meaningful notion of F-RK security as considered above, where the adversary can ask for en-/decryptions with respect to *any* (fixed) function φ. Similarly to Definition 3, we may expect $\mathbf{Adv}^{\mathrm{frk}}_{\mathcal{E}}(q,t) \approx t \cdot 2^{-\kappa/2}$ from a good cipher; in order to achieve a better advantage it is necessary to exploit structural properties of $\mathcal{E}$.

3 Construction

We now recall our construction and provide some intuition about its soundness, before presenting the formal security analysis in the next section.

Let $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher and $\mathcal{D}$ its inverse. We define $\mathsf{RKNMC}[\mathcal{E}]$ as the code with codewords of size $\kappa + n$ where $\mathrm{Enc} : m \mapsto k || \mathcal{E}_k(m)$ for a randomly chosen $k \xleftarrow{\$} \{0,1\}^\kappa$, and the associated decoding procedure is naturally given by $\mathrm{Dec}(c_{\mathrm{L}} || c_{\mathrm{R}}) \mapsto \mathcal{D}_{c_{\mathrm{L}}}(c_{\mathrm{R}})$, as illustrated in Figure 1.

3.1 Broken Instantiations

By definition, $\mathsf{RKNMC}[\mathcal{E}]$ is functionally a non-malleable code, but the level of security that it provides highly depends on $\mathcal{E}$. In particular, it is rather easy to see that some amount of related-key security — if maybe not sufficient — is at least necessary. Indeed, we can consider the following example: take $\mathcal{E}$ to be an Even-Mansour block cipher [EM91] defined from a public permutation $\mathcal{P}$ as $\mathcal{E}_k(m) = k \oplus \mathcal{P}(m \oplus k)$. In the single-key setting, Even-Mansour is proven to be secure up to the birthday bound in the ideal

[1] F-RK stands for *fixed* related-key and emphasizes that only en-/decryption queries under a *fixed* related key are possible. However, as should also be clear from the definition, there is no a priori restriction on the allowed related-key functions.

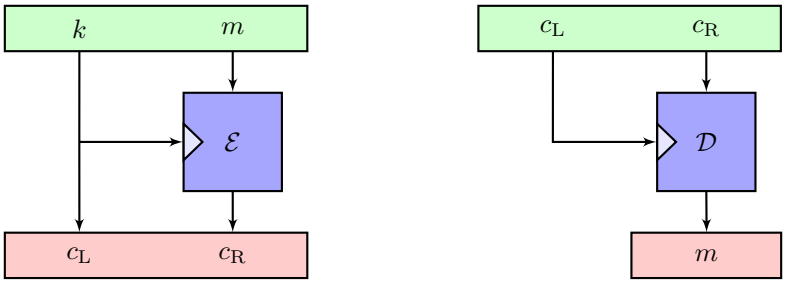

Figure 1: Non-malleable encoding (left) and decoding (right) with $\mathsf{RKNMC}[\mathcal{E}]$.

permutation model [EM91]. However it suffers from a trivial related-key distinguisher, as $\mathcal{E}_{k\oplus\Delta}(m \oplus \Delta) = \mathcal{E}_k(m) \oplus \Delta$. This directly leads to an attack on $\mathsf{RKNMC}[\mathcal{E}]$: if we let $\mathrm{T_L} = \mathrm{T_R} : x \mapsto x \oplus \Delta$ for some Δ, then

$$\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m) = \mathcal{D}_{\mathrm{T_L}(k)} \circ \mathrm{T_R} \circ \mathcal{E}_k(m) = \mathcal{D}_{k\oplus\Delta}(\mathcal{E}_k(m) \oplus \Delta) = m \oplus \Delta\,,$$

which is strongly related to m (by having a fixed difference Δ), and so the original message can be easily recovered.

3.2 Proof Idea

While the above might seem worrying, it crucially exploits related-key weaknesses of $\mathcal{E}$. We now provide some intuition why resisting related-key attacks is in fact (mostly) *sufficient* to obtain a secure non-malleable code.

First, one can notice that the decoding $\mathcal{D}_{\mathrm{T_L}(k)}(\mathrm{T_R}(\mathcal{E}_k(m)))$ of a tampered codeword can be obtained by means of a couple of related-key en-/decryption queries. It follows that distinguishing such a tampered codeword from a random uncorrelated one enables to distinguish the "real" from the "ideal" oracle in the related-key security game from Definition 4, *assuming* that the oracle there does not abort and thus provides the necessary en- and decryptions. With this observation, we already get security of $\mathsf{RKNMC}[\mathcal{E}]$ for choices of $\mathrm{T_L}$ for which $\mathrm{T_L}(k)$ is an unlikely value (except with small probability), *i.e.*, for tampering functions $\mathrm{T_L}$ that are *output-unpredictable* using the terminology of [BK03]. Unfortunately, it is crucial for the security notion in Definition 4 to make sense to have the oracle abort as soon as $\mathrm{T_L}(k)$ becomes predictable. This is because the adversary can trivially distinguish a real encryption from an ideal encryption under a predictable key $\mathrm{T_L}(k)$, for instance if $\mathrm{T_L} : k \mapsto k_0$ for a constant k_0, because then it can compute the real encryption himself.

On the other hand, for such an extreme choice of a *constant* tampering function, where the reduction to related-key security fails (because the oracle in Definition 4 aborts), we see that it actually does not harm NMC security. Indeed, in this case, $\mathcal{D}_{\mathrm{T_L}(k)}(\mathrm{T_R}(\mathcal{E}_k(m)))$ can be computed with the help of a standard encryption query only (since the decryption with the publicly-known key $\mathrm{T_L}(k) = k_0$ can be done without any oracle query), and so the security of $\mathsf{RKNMC}[\mathcal{E}]$ for such a function follows from standard PRP security.

A similar argument can be used even if $\mathrm{T_L}$ is not constant, but still has a sufficiently small image. For instance, assume that it takes possible values $\{k_0, k_1, \ldots, k_m\}$, such that every corresponding preimage $\mathcal{K}_i := {\mathrm{T_L}}^{-1}(\{k_i\})$ is still *large* (say at least $2^{\kappa/2}$), so that k remains hard to guess even when given $\mathrm{T_L}(k)$. Then, the security of $\mathsf{RKNMC}[\mathcal{E}]$ can be obtained by assuming that the PRP security of $\mathcal{E}$ degrades gracefully when its keyspace is restricted to any $\mathcal{K}_i$. In other words, security of $\mathsf{RKNMC}[\mathcal{E}]$ then reduces to PRP-with-leakage security as in Definition 3.

To prove the security of $\mathsf{RKNMC}[\mathcal{E}]$ in full generality without any restriction on $\mathrm{T_L}$, one has to combine the two above reductions. This can be done by splitting the keyspace $\{0,1\}^\kappa$ of $\mathcal{E}$ into two parts, depending on whether $\mathrm{T_L}(k)$ is an unlikely value or not, and by observing that in each case, either the oracle in the security game of Definition 3 or the oracle in the security game of Definition 4 does not abort, and so we can do the security reduction to the corresponding security property.

Overall, this strategy allows us to show that for a well-chosen $\mathcal{E}$, $\mathsf{RKNMC}[\mathcal{E}]$ has $\tau = \kappa/2$ bits of security.

4 Proof of the Construction

Recall that in the NMC security game, the decoding is done by means of a "tweaked" decoder $\dot{\mathrm{Dec}}^{(c)}$ that outputs $\vartriangle$ in case its input c' is equal to c, *i.e.*, no tampering took place.

Theorem 1. *Let $\mathcal{E} : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{M}$ be a block cipher. If the hash function family $\mathcal{H} : \mathcal{M} \times \{0,1\}^\kappa \to \mathcal{M}$ defined by $\mathcal{H}(k,x) = \mathcal{E}_x(k)$ is ε-universal, then the construction of Section 3 is a non-malleable code with:*

$$\mathbf{Adv}^{\text{snmc-s}}_{\mathsf{RKNMC}}(t) \le 2\max\left\{\mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(1, 2t+\tau_{\mathcal{D}}) + 2^{-\kappa/2}, \mathbf{Adv}^{\text{frk}}_{\mathcal{E}}(4, 2t) + \varepsilon + 2^{-n}\right\},$$

where $\tau_{\mathcal{D}}$ is the time complexity of $\mathcal{D}$.

Proof. Let $\mathrm{T} = \mathrm{T_L} \,\|\, \mathrm{T_R}$ be a bipartite tampering function that can be computed in time t, and let A be an algorithm with running time t. We need to show that the output of $A(\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m))$ is almost independent of m. Let K be uniformly distributed over $\{0,1\}^\kappa$, and let Λ be the event that

$$K \in \left\{k \;\middle|\; \Pr[\mathrm{T_L}(K) = \mathrm{T_L}(k)] \le 2^{-\kappa/2}\right\},$$

i.e., the event that "$\mathrm{T_L}(K)$ is an unlikely value". We point out that Λ is the event for which $\mathcal{O}_{\mathrm{T_L}}$ aborts in the security game of Definition 3 when queried on the function $\varphi = \mathrm{T_L}$, and it is also the event for which $\mathcal{O}_{\mathrm{T_L}}$ does *not* abort in the security game of Definition 4. We observe that

$$\Pr\left[\,\mathrm{T_L}(K) = k' \wedge \Lambda\,\right] \le 2^{-\kappa/2} \tag{4}$$

and

$$\Pr\left[\,K = k \wedge \neg\Lambda \;\middle|\; \mathrm{T_L}(K) = k'\,\right] \le 2^{-\kappa/2} \tag{5}$$

for all k and k'; the former is by definition of Λ and the latter follows from

$$\begin{aligned} 2^{-\kappa} = \Pr[\,K = k\,] &\ge \Pr[\,K = k \wedge \mathrm{T_L}(K) = k' \wedge \neg\Lambda\,] \\ &\ge \Pr[\,K = k \wedge \neg\Lambda \mid \mathrm{T_L}(K) = k'\,] \cdot \Pr[\,\mathrm{T_L}(K) = k'\,] \\ &\ge \Pr[\,K = k \wedge \neg\Lambda \mid \mathrm{T_L}(K) = k'\,] \cdot 2^{-\kappa/2}, \end{aligned}$$

where the last inequality is by observing that if k' is such that $\Pr[\,\mathrm{T_L}(K) = k'\,] \le 2^{-\kappa/2}$ then $\Pr[\,K = k \wedge \neg\Lambda \mid \mathrm{T_L}(K) = k'\,] = 0$ by the definition of Λ. Furthermore, by considering (5) and setting $k' = k$, multiplying with $\Pr[\mathrm{T_L}(K) = k]$, and summing over k, we get that

$$\Pr\left[\,\mathrm{T_L}(K) = K \wedge \neg\Lambda\,\right] \le 2^{-\kappa/2}. \tag{6}$$

We extend the notation of $\dot{\mathrm{Dec}}$ to $\mathcal{D}$ and write $\dot{\mathcal{D}}^{k,c}_{k'}(c')$ for the standard decryption but with the decrypted message replaced by $\vartriangle$ in case $k' = k$ and $c' = c$. Then, for the case

that the event Λ does not occur, we can argue as follows for any fixed choice of m:

$$\begin{aligned}
&\Pr[A(\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m)) = 1 \wedge \neg\Lambda] \\
&\quad = \Pr\Big[\, A \circ \dot{\mathrm{Dec}}^{\mathrm{Enc}_K(m)} \circ \mathrm{T} \circ \mathrm{Enc}_K(m) = 1 \wedge \neg\Lambda \,\Big] \\
&\quad = \Pr\Big[\, A \circ \dot{\mathcal{D}}^{K,\mathcal{E}_K(m)}_{\mathrm{T_L}(K)} \circ \mathrm{T_R} \circ \mathcal{E}_K(m) = 1 \wedge \neg\Lambda \,\Big] \qquad \text{(by construction)} \\
&\quad = \Pr\Big[\, A \circ \mathcal{D}_{\mathrm{T_L}(K)} \circ \mathrm{T_R} \circ \mathcal{E}_K(m) = 1 \wedge \neg\Lambda \,\Big] \pm 2^{-\kappa/2} \qquad \text{(by (6))} \\
&\quad = \Pr\Big[\, A \circ \mathcal{D}_{\mathrm{T_L}(K)} \circ \mathrm{T_R} \circ \mathfrak{E}(m) = 1 \wedge \neg\Lambda \,\Big] \pm \mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(1, 2t + \tau_{\mathcal{D}}) \pm 2^{-\kappa/2}\,,
\end{aligned}$$

where $\mathfrak{E} \xleftarrow{\$} \mathrm{Perm}(\mathcal{M})$ is a random permutation. This last approximation is by the PRP-with-leakage security by considering the leakage function $\mathrm{T_L}$ and the adversary that outputs 1 if $\mathcal{O}_{\mathrm{T_L}}$ aborts and, if it receives k' instead, queries $\mathcal{O}_{\mathrm{T_L}}$ on m to get c and then outputs $A \circ \mathcal{D}_{k'} \circ \mathrm{T_R}(c)$. Given that $\neg\Lambda$ coincides with the event that $\mathcal{O}_{\mathrm{T_L}}$ does not abort in the PRP-with-leakage security game, this shows that the two probabilities indeed differ by at most $\mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(1, 2t+\tau_{\mathcal{D}})$. The derived probability above is obviously independent of the choice of m, because $\mathfrak{E}(m)$ is uniformly random and independent of K, no matter what m is.

To argue for the case Λ, we set $K' := \mathrm{T_L}(K)$ as a short hand, we let $\widetilde{M}$ be a uniformly random message in $\mathcal{M}$ independent from m, and we write $\ddot{\mathcal{D}}^{\tilde{c},\tilde{c}',c}_{k'}(c')$ for the standard decryption but with the message replaced by $\vartriangle$ in case $\tilde{c} = \tilde{c}'$ and $c' = c$. We get that:

$$\begin{aligned}
&\Pr[A(\mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m)) = 1 \wedge \Lambda] \\
&\quad = \Pr\Big[\, A \circ \dot{\mathrm{Dec}}^{\mathrm{Enc}_K(m)} \circ \mathrm{T} \circ \mathrm{Enc}_K(m) = 1 \wedge \Lambda \,\Big] \\
&\quad = \Pr\Big[\, A \circ \dot{\mathcal{D}}^{K,\mathcal{E}_K(m)}_{K'} \circ \mathrm{T_R} \circ \mathcal{E}_K(m) = 1 \wedge \Lambda \,\Big] \qquad \text{(by construction)} \\
&\quad = \Pr\Big[\, A \circ \ddot{\mathcal{D}}^{\mathcal{E}_K(\widetilde{M}),\mathcal{E}_{K'}(\widetilde{M}),\mathcal{E}_K(m)}_{K'} \circ \mathrm{T_R} \circ \mathcal{E}_K(m) = 1 \wedge \Lambda \,\Big] \pm \varepsilon \qquad \text{(by Definition 1)} \\
&\quad = \Pr\Big[\, A \circ \ddot{\mathfrak{D}}^{\mathfrak{E}_K(\widetilde{M}),\mathfrak{E}_{K'}(\widetilde{M}),\mathfrak{E}_K(m)}_{K'} \circ \mathrm{T_R} \circ \mathfrak{E}_K(m) = 1 \wedge \Lambda \,\Big] \pm \mathbf{Adv}^{\text{frk}}_{\mathcal{E}}(4, 2t) \pm \varepsilon\,,
\end{aligned}$$

where the last approximation is by F-RK security. Concretely, it follows by considering the function $\mathrm{T_L}$ and the adversary that outputs 1 if $\mathcal{O}_{\mathrm{T_L}}$ aborts and otherwise acts as follows. First, it queries the oracle on $(\texttt{enc}, \texttt{real}, m)$ to obtain c and on $(\texttt{enc}, \texttt{real}, \tilde{m})$ and $(\texttt{enc}, \texttt{modified}, \tilde{m})$ for a random $\tilde{m}$ to obtain $\tilde{c}$ and $\tilde{c}'$, respectively. Then, if $\tilde{c} = \tilde{c}'$ and $\mathrm{T_R}(c) = c$, it runs A on input $\vartriangle$ and outputs whatever A outputs. Otherwise, it queries the oracle on $(\texttt{dec}, \texttt{modified}, \mathrm{T_R}(c))$ and runs A on the reply m' that the oracle provides, and outputs whatever A outputs. Given that Λ coincides with the event that $\mathcal{O}_{\mathrm{T_L}}$ does not abort in the F-RK security game, this shows that the two probabilities differ by at most $\mathbf{Adv}^{\text{frk}}_{\mathcal{E}}(4, 2t)$.

Here, the derived probability is *almost* independent of the choice of m. The only dependency occurs in the event that $K = K'$ and $\mathrm{T_R} \circ \mathfrak{E}_K(m) \neq \mathfrak{E}_K(m)$, in which case $\ddot{\mathfrak{D}}_{K'}$ decrypts to a random message *distinct* from m, but this is 2^{-n}-close from a random message that may include m.

Finally, given that the event Λ is independent of the choice of m, by basic properties of the statistical distance we get that the distributions of $A \circ \mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m_0)$ and $A \circ \mathsf{Tamp}^{\mathrm{T}}_{\mathsf{RKNMC}}(m_1)$ are close as claimed. □

5 A Matching Generic Attack

If $\mathcal{E}$ is an ideal cipher, it satisfies (see the remarks after Definitions 3 and 4)

$$\mathbf{Adv}^{\mathrm{frk}}_{\mathcal{E}}(q,t) \approx \mathbf{Adv}^{\text{prp-leak}}_{\mathcal{E}}(q,t) \approx t/2^{\kappa/2}\,,$$

where we equate the time complexity with the number of primitive queries, and the bound of Theorem 1 is in $O(t/2^{\kappa/2})$. We sketch a generic attack for RKNMC instantiated with a block cipher $\mathcal{E} : \{0,1\}^{\kappa} \times \{0,1\}^{n} \to \{0,1\}^{n}$. If $\kappa \leq n$, n a constant, the attack succeeds with probability $\Omega(t^2/2^{\kappa})$, closely matching the bound of the theorem (for large values of t) if $\mathcal{E}$ is frk- and prp-leak-secure.

Consider an adversary for the game of Definition 2, which is given two distinct message m_0 and m_1 and the tampering functions

$$\begin{aligned}\mathrm{T_L} : x = x_{\kappa-1}x_{\kappa-2}\dots x_0 &\mapsto 0^{\kappa/2}x_{\kappa/2-1}\dots x_0\,,\\ \mathrm{T_R} : x = x_{n-1}x_{n-2}\dots x_0 &\mapsto 0^{n/2}x_{n/2-1}\dots x_0\,,\end{aligned}$$

where 0^{α} denotes a string of α zeros. That is, both tampering functions overwrite the left half of their inputs with zeros. The adversary is then given the tampered decoding

$$c = \mathrm{Dec} \circ \mathrm{T} \circ \mathrm{Enc}_k(m_b) = \mathcal{D}_{\mathrm{T_L}(k)}(\mathrm{T_R}(\mathcal{E}_k(m_b)))\,,$$

where $k \xleftarrow{\$} \{0,1\}^{\kappa}$ and $b \xleftarrow{\$} \{0,1\}$; we assume here that $\mathrm{T_R}(\mathcal{E}_k(m_0)) \neq \mathrm{T_R}(\mathcal{E}_k(m_1))$, which is true with probability close to $2^{-n/2}$.

The idea of the attack is to use a meet-in-the-middle approach that allows to separately retrieve the two halves of the randomness. The adversary first zeroes the upper bits of k using $\mathrm{T_L}$, re-encrypts c with candidate values for its lower bits, and only keeps the ones that lead to observing a distinguishing condition on $\mathrm{T_R}(\mathcal{E}_k(m_b))$, *i.e.*, that the upper bits of the re-encryption must be zero. Remaining candidate values for k are then completed in their upper bits and used to tentatively encode both of m_0 and m_1; a match with $\mathrm{T_R}(\mathcal{E}_k(m_b))$ suggests a full candidate that may be used to answer the challenge. This procedure is detailed in Algorithm 1, with the main steps described below.

At the end of the loop on lines 3–8, the set Δ includes the "right" pair $(\mathrm{T_R}(\mathcal{E}_k(m_b)), \mathrm{T_L}(k))$, along with an expected number $w := 2^{\kappa/2}/2^{n/2}$ of false positives. The loop on lines 9–20 is executed at most $w+1$ times, and each execution takes at most $2^{\kappa/2}$ executions of the inner-loop of lines 10–19. For each pair (d,k) that is not the right one, the expected number of successful tests on lines 12 and 16 is $2^{\kappa/2}/2^{n/2}$ each. This is the same for the right pair, plus an additional successful test for the correct value of b. All in all, the attack succeeds if and only if the tampered decodings of m_0 and m_1 did not collide in the first place and if there was no false positive on line 12 or 16, that is with probability approximately

$$\left(1 - 2^{-n/2}\right) \cdot \left(\frac{1}{1 + \frac{2^{\kappa/2}}{2^{n/2}}}\right)^2 .$$

If $\kappa \leq n$, this success probability is $\Omega(1)$ (when fixing n to a constant).

The attack can also be run with fewer than $\approx 2^{\kappa/2}$ steps, decreasing its success probability accordingly. If the loop 3–8 is interrupted after at most t iterations, then the probability that the right pair was found is $t/2^{\kappa/2}$. Assuming a negligible number of false positive for simplicity (*i.e.*, assuming $\kappa \leq n$), if the loop of lines 10–19 is again interrupted after at most t iterations, the algorithm returns a correct answer with probability $t/2^{\kappa/2}$. The advantage of such a $2t$-restricted adversary is thus $\Omega(t^2/2^{\kappa})$, which is consistent with a meet-in-the-middle attack such as this one.

```
1  begin
2  |  Δ := ∅
3  |  for k_i := 0^{κ/2}||[i]_2, i < 2^{κ/2} do ▷ [x]_2 is the binary representation of x.
4  |  |  d_i := E_{k_i}(c)
5  |  |  if d_i has its n/2 leftmost bits equal to zero then
6  |  |  |  Δ := Δ ∪ (d_i, k_i)
7  |  |  end
8  |  end
9  |  forall (d, k) ∈ Δ do
10 |  |  forall k_j s.t. T_L(k_j) = k do
11 |  |  |  e_{j,0} := T_R(E_{k_j}(m_0))
12 |  |  |  if e_{j,0} = d then
13 |  |  |  |  return 0
14 |  |  |  end
15 |  |  |  e_{j,1} := T_R(E_{k_j}(m_1))
16 |  |  |  if e_{j,1} = d then
17 |  |  |  |  return 1
18 |  |  |  end
19 |  |  end
20 |  end
21 |  return 0
22 end
```

Algorithm 1: A generic attack for RKNMC.

6 Concluding Remarks

In this work, we considered the simplest construction of an NMC from a block cipher $\mathcal{E}$, and we characterized its security by means of formulating and proving sufficient security conditions on $\mathcal{E}$. Since these require some form of related-key security, it is then natural to wonder how reasonable such an assumption is and how it could be instantiated in practice. A first line of approach is to consider existing block ciphers. There is good empirical evidence that *e.g.* the NIST standard AES-128 [NIS01] and the NESSIE cipher SHACAL-2 [HN01] are both secure against related-key attacks; this yields immediate candidates for explicit instantiations targeting 64 and 256 bits of security respectively.

From a more theoretical point of view, Farshim and Procter [FP15] and, independently, Cogliati and Seurin [CS15] showed that a 3-round iterated Even-Mansour construction with identical keys is provably secure against attacks under the related-key class $\Phi^{\oplus} := \{x \mapsto x \oplus \Delta \,|\, \Delta \in \{0,1\}^{|x|}\}$. Although this result focuses on a single related-key class, it is significant insofar as it shows that a non-trivial level of related-key security can be achieved by a very simple construction. Furthermore, no attack exploiting a different related-key class is currently known for this construction.

Overall, there is good empirical and theoretical evidence that related-key security is achievable for block ciphers, and our non-malleable code could thus easily and immediately be instantiated by existing schemes.

Finally, while the construction and its analysis in Section 4 center around a simple fixed-input-length block cipher, we conclude with the observation that the approach and our result naturally extends to using variable-input-length block ciphers (VILBC). A VILBC is a family of block ciphers $\mathcal{E}^{\ell} : \{0,1\}^{\kappa} \times \{0,1\}^{\ell} \to \{0,1\}^{\ell}$ indexed by a block size parameter $\ell \in \mathcal{S}$, where $\mathcal{S}$ is a predetermined set (*e.g.*, $\mathcal{S} = [n, \infty[$). The security of the VILBC $\mathcal{E}$ is defined as the minimum security reached for any (valid) choice of the length parameter, and the security models of Section 2.3 straightforwardly generalize.

Given a secure VILBC $\mathcal{E}$, the NMC of Section 3 generalizes to messages m whose length is at most $|m| \leq \max \mathcal{S}$ as follows: injectively pad m to m^* such that $\ell = |m^*|$ is a valid size parameter for $\mathcal{E}$ (*i.e.*, $\ell \in \mathcal{S}$), and return $k || \mathcal{E}_k^\ell(m^*)$. Because $\mathcal{E}^\ell$ for fixed ℓ is a fixed-input-length block cipher, the analysis of Section 4 carries over and ensures that this is a secure encoding of m^*, and thus of m.

Many constructions of VILBC have been proposed in the literature, see, *e.g.*, [NR99, BR99]; concrete instantiations of such primitives have also been designed, *e.g.*, [HKR15, BDP+14]. Not all of these constructions natively achieve related-key security. Yet, at the cost of some overhead, related-key security can be achieved with some level of provable security, for instance by using the VILBC as a permutation within a related-key-secure Even-Mansour construction.

References

[AAG+16] Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran, *Optimal Computational Split-state Non-malleable Codes*, Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II (Eyal Kushilevitz and Tal Malkin, eds.), Lecture Notes in Computer Science, vol. 9563, Springer, 2016, pp. 393–417.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett, *Non-malleable codes from additive combinatorics*, Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014 (David B. Shmoys, ed.), ACM, 2014, pp. 774–783.

[AGM+15] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran, *Explicit Non-malleable Codes Against Bit-Wise Tampering and Permutations*, Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I (Rosario Gennaro and Matthew Robshaw, eds.), Lecture Notes in Computer Science, vol. 9215, Springer, 2015, pp. 538–557.

[BDP+14] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer, *Using* Keccak *technology for AE:* Ketje, Keyak *and more*, SHA-3 Workshop, August 2014.

[BDSKM17] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin, *Non-Malleable Codes from Average-Case Hardness: AC0, Decision Trees, and Streaming Space-Bounded Tampering*, IACR Cryptology ePrint Archive **2017** (2017), 1061.

[BG13] Alberto Battistello and Christophe Giraud, *Fault Analysis of Infective AES Computations*, 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013 (Wieland Fischer and Jörn-Marc Schmidt, eds.), IEEE Computer Society, 2013, pp. 101–107.

[BG16] Alberto Battistello and Christophe Giraud, *A Note on the Security of CHES 2014 Symmetric Infective Countermeasure*, Constructive Side-Channel Analysis and Secure Design - 7th International Workshop, COSADE 2016, Graz, Austria, April 14-15, 2016, Revised Selected Papers (François-Xavier Standaert and Elisabeth Oswald, eds.), Lecture Notes in Computer Science, vol. 9689, Springer, 2016, pp. 144–159.

[BK03] Mihir Bellare and Tadayoshi Kohno, *A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications*, Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings (Eli Biham, ed.), Lecture Notes in Computer Science, vol. 2656, Springer, 2003, pp. 491–506.

[BM16] Sarani Bhattacharya and Debdeep Mukhopadhyay, *Curious Case of Rowhammer: Flipping Secret Exponent Bits Using Timing Analysis*, Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings (Benedikt Gierlichs and Axel Y. Poschmann, eds.), Lecture Notes in Computer Science, vol. 9813, Springer, 2016, pp. 602–624.

[BNNT11] Eric Brier, David Naccache, Phong Q. Nguyen, and Mehdi Tibouchi, *Modulus Fault Attacks against RSA-CRT Signatures*, Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings (Bart Preneel and Tsuyoshi Takagi, eds.), Lecture Notes in Computer Science, vol. 6917, Springer, 2011, Full version at `https://eprint.iacr.org/2011/388`, pp. 192–206.

[BR99] Mihir Bellare and Phillip Rogaway, *On the Construction of Variable-Input-Length Ciphers*, Fast Software Encryption, 6th International Workshop, FSE'99, Rome, Italy, March 24-26, 1999, Proceedings (Lars R. Knudsen, ed.), Lecture Notes in Computer Science, vol. 1636, Springer, 1999, pp. 231–244.

[BR14] Lejla Batina and Matthew Robshaw (eds.), *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, Lecture Notes in Computer Science, vol. 8731, Springer, 2014.

[BS03] Johannes Blömer and Jean-Pierre Seifert, *Fault Based Cryptanalysis of the Advanced Encryption Standard (AES)*, Financial Cryptography, 7th International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003, Revised Papers (Rebecca N. Wright, ed.), Lecture Notes in Computer Science, vol. 2742, Springer, 2003, pp. 162–181.

[CMTV15] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi, *From Single-Bit to Multi-bit Public-Key Encryption via Non-malleable Codes*, Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I (Yevgeniy Dodis and Jesper Buus Nielsen, eds.), Lecture Notes in Computer Science, vol. 9014, Springer, 2015, pp. 532–560.

[CS15] Benoit Cogliati and Yannick Seurin, *On the Provable Security of the Iterated Even-Mansour Cipher Against Related-Key and Chosen-Key Attacks*, in Oswald and Fischlin [OF15], pp. 584–613.

[CW77] Larry Carter and Mark N. Wegman, *Universal Classes of Hash Functions (Extended Abstract)*, Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA (John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison, eds.), ACM, 1977, pp. 106–112.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski, *Non-malleable Codes from Two-Source Extractors*, Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22,

2013. Proceedings, Part II (Ran Canetti and Juan A. Garay, eds.), Lecture Notes in Computer Science, vol. 8043, Springer, 2013, pp. 239–257.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs, *Non-Malleable Codes*, Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings (Andrew Chi-Chih Yao, ed.), Tsinghua University Press, 2010, pp. 434–452.

[EM91] Shimon Even and Yishay Mansour, *A Construction of a Cipher From a Single Pseudorandom Permutation*, ASIACRYPT '91 (Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, eds.), Lecture Notes in Computer Science, vol. 739, Springer, 1991, pp. 210–224.

[FHMV17] Sebastian Faust, Kristina Hostáková, Pratyay Mukherjee, and Daniele Venturi, *Non-Malleable Codes for Space-Bounded Tampering*, Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II (Jonathan Katz and Hovav Shacham, eds.), Lecture Notes in Computer Science, vol. 10402, Springer, 2017, pp. 95–126.

[FN17] Antonio Faonio and Jesper Buus Nielsen, *Non-malleable Codes with Split-State Refresh*, Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I (Serge Fehr, ed.), Lecture Notes in Computer Science, vol. 10174, Springer, 2017, pp. 279–309.

[FP15] Pooya Farshim and Gordon Procter, *The Related-Key Security of Iterated Even-Mansour Ciphers*, Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers (Gregor Leander, ed.), Lecture Notes in Computer Science, vol. 9054, Springer, 2015, pp. 342–363.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson, *Textbook non-malleable commitments*, Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016 (Daniel Wichs and Yishay Mansour, eds.), ACM, 2016, pp. 1128–1141.

[GS08] Jens Groth and Amit Sahai, *Efficient Non-interactive Proof Systems for Bilinear Groups*, Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings (Nigel P. Smart, ed.), Lecture Notes in Computer Science, vol. 4965, Springer, 2008, pp. 415–432.

[GST12] Benedikt Gierlichs, Jörn-Marc Schmidt, and Michael Tunstall, *Infective Computation and Dummy Rounds: Fault Protection for Block Ciphers without Check-before-Output*, Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings (Alejandro Hevia and Gregory Neven, eds.), Lecture Notes in Computer Science, vol. 7533, Springer, 2012, pp. 305–321.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway, *Robust Authenticated-Encryption AEZ and the Problem That It Solves*, in Oswald and Fischlin [OF15], pp. 15–44.

[HN01] Helena Handschuh and David Naccache, *SHACAL*, NESSIE portfolio, October 2001.

[KLT16] Aggelos Kiayias, Feng-Hao Liu, and Yiannis Tselekounis, *Practical Non-Malleable Codes from ℓ-more Extractable Hash Functions*, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016 (Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, eds.), ACM, 2016, pp. 1317–1328.

[LL12] Feng-Hao Liu and Anna Lysyanskaya, *Tamper and Leakage Resilience in the Split-State Model*, Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings (Reihaneh Safavi-Naini and Ran Canetti, eds.), Lecture Notes in Computer Science, vol. 7417, Springer, 2012, pp. 517–532.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom, *A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro*, in Oswald and Fischlin [OF15], pp. 254–283.

[LRT12] Victor Lomné, Thomas Roche, and Adrian Thillard, *On the Need of Randomness in Fault Attack Countermeasures - Application to AES*, 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium, September 9, 2012 (Guido Bertoni and Benedikt Gierlichs, eds.), IEEE Computer Society, 2012, pp. 85–94.

[LRW11] Moses Liskov, Ronald L. Rivest, and David A. Wagner, *Tweakable Block Ciphers*, J. Cryptology **24** (2011), no. 3, 588–613.

[Nao03] Moni Naor, *On Cryptographic Assumptions and Challenges*, Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings (Dan Boneh, ed.), Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 96–109.

[NIS01] National Institute of Standards and Technology, *FIPS 197: Advanced Encryption Standard (AES)*, November 2001.

[NR99] Moni Naor and Omer Reingold, *On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited*, J. Cryptology **12** (1999), no. 1, 29–66.

[NS12] Moni Naor and Gil Segev, *Public-Key Cryptosystems Resilient to Key Leakage*, SIAM J. Comput. **41** (2012), no. 4, 772–814.

[OF15] Elisabeth Oswald and Marc Fischlin (eds.), *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, Lecture Notes in Computer Science, vol. 9056, Springer, 2015.

[PQ03] Gilles Piret and Jean-Jacques Quisquater, *A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD*, Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings (Colin D. Walter, Çetin Kaya Koç, and Christof Paar, eds.), Lecture Notes in Computer Science, vol. 2779, Springer, 2003, pp. 77–88.

[TBM14] Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay, *Destroying Fault Invariant with Randomization - A Countermeasure for AES Against Differential Fault Attacks*, in Batina and Robshaw [BR14], pp. 93–111.