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

Gianluca Brian[1], Antonio Faonio[2], João Ribeiro[3] and Daniele Venturi[4]

[1] Sapienza University of Rome, Rome, Italy
brian@di.uniroma1.it
[2] EURECOM, Sophia-Antipolis, Biot, France
antonio.faonio@eurecom.fr
[3] Carnegie Mellon University, Pittsburgh, PA, USA
jlourenc@cs.cmu.edu
[4] Sapienza University of Rome, Rome, Italy
venturi@di.uniroma1.it

Abstract. We construct non-malleable codes in the split-state model with codeword length $m+3\lambda$ or $m+5\lambda$, where m is the message size and λ is the security parameter, depending on how conservative one is. Our scheme is very simple and involves a single call to a block cipher meeting a new security notion which we dub *entropic* fixed-related-key security, which essentially means that the block cipher behaves like a pseudorandom permutation when queried upon inputs sampled from a distribution with sufficient min-entropy, even under related-key attacks with respect to an arbitrary but fixed key relation. Importantly, indistinguishability only holds with respect to the original secret key (and not with respect to the tampered secret key).
In a previous work, Fehr, Karpman, and Mennink (ToSC 2018) used a related assumption (where the block cipher inputs can be chosen by the adversary, and where indistinguishability holds even with respect to the tampered key) to construct a non-malleable code in the split-state model with codeword length $m + 2\lambda$. Unfortunately, no block cipher (even an ideal one) satisfies their assumption when the tampering function is allowed to be cipher-dependent. In contrast, we are able to show that entropic fixed-related-key security holds in the ideal cipher model with respect to a large class of cipher-dependent tampering attacks (including those which break the assumption of Fehr, Karpman, and Mennink).

Keywords: non-malleability, split-state model, block ciphers, related-key security.

1 Introduction

Consider the classical setting in which a message μ is encoded via an algorithm Encode, yielding a codeword σ. The decoding algorithm Decode allows one to recover μ from σ, efficiently. The goal of such an encoding procedure is to prevent modifications to the codeword, either benign (e.g., because of errors introduced by the communication medium) or adversarial (e.g., because of malicious tampering attacks). In more detail, let $\tilde{\sigma} = f(\sigma)$ be a modified codeword for some tampering function f over the codeword space. The property of *error correction* guarantees that $\mathsf{Decode}(\tilde{\sigma})$ still results in the original message μ. The property of *error detection*, instead, guarantees that $\mathsf{Decode}(\tilde{\sigma})$ either results in the original message μ or in an error symbol $\perp$ (denoting that tampering occurred but the original message cannot be recovered). While the goals of error correction/detection are very well understood, it is well known that these guarantees are simply impossible for certain classes of tampering functions, particularly so for those classes that model *adversarial* tampering (e.g., the family of constant functions).

Motivated by this shortcoming, Dziembowski, Pietrzak, and Wichs introduced the beautiful notion of *non-malleable* codes [DPW10, DPW18], which guarantees that $\mathsf{Decode}(\tilde{\sigma})$ either results in the original message or in a completely *unrelated* value. While being weaker than error correction and error detection, non-malleability can be achieved for much larger classes of tampering functions. Moreover, a non-malleable code can be used to protect arbitrary cryptographic primitives against tampering attacks targeting the memory (a.k.a. related-key attacks). The latter is achieved by simply storing the secret key in encoded form, and by decoding it prior to invoking the underlying cryptographic algorithms. Intuitively, this ensures that memory tampering either results in the same key (and thus has no effect) or to a completely unrelated key (which does not harm[1] security).

In this work, we focus on the so-called *split-state* model in which a codeword consists of two parts (σ_0, σ_1) that can be tampered arbitrarily yet independently; namely, a tampering function f has a type $f = (f_0, f_1)$ and the mauled codeword is of the form $\tilde{\sigma} = (\tilde{\sigma}_0, \tilde{\sigma}_1) = (f_0(\sigma_0), f_1(\sigma_1))$. While non-malleable codes in the split-state model exist unconditionally, Cheraghchi and Guruswami [CG16] established that the best achievable rate for such codes in the information-theoretic setting is $1/2$, where the rate refers to the (asymptotic) ratio between the length of the message and the length of the codeword when the message length goes to infinity. This lower bound motivated cryptographers to build more efficient codes under (as weak as possible) computational assumptions. We refer the reader to Section 1.3 for a summary of known results.

1.1 The Work of Fehr, Karpman, and Mennink

The starting point of our work is a paper by Fehr, Karpman, and Mennink [FKM18] (improving previous works by Aggarwal, Agrawal, Gupta, Maji, Pandey, and Prabhakaran [AAG+16] and Kiayias, Liu, and Tselekounis [KLT16]), where the authors show how to construct non-malleable codes in the split-state model assuming sufficiently strong block ciphers. Their construction is the simplest possible cipher-based split-state non-malleable code: The left part of the codeword is the key κ for a block cipher, and the right part of the codeword is the ciphertext γ encrypting the message. Namely,

$$\mathsf{Encode}(\mu) \to (\kappa, \mathsf{Encrypt}(\kappa, \mu)) \,.$$

The length of the codeword in their candidate construction is $m + 2\lambda$, where m is the size of the message and λ is a security[2] parameter, which is the shortest known today.

One of the explicit goals of Fehr *et al.* was to understand the assumptions needed from the block cipher in order to prove non-malleability of the above simple construction without relying on trusted setup or other (non-falsifiable) assumptions. Their main technical result is that the latter is indeed possible assuming the underlying block cipher is: (i) a pseudorandom permutation (PRP) under leakage (a.k.a. PRP-with-leakage security), and (ii) related-key secure with respect to an arbitrary but fixed key relation (a.k.a. FRK security). Property (i) means that the block cipher behaves as a PRP even given arbitrary leakage on the secret key, so long as the latter is still unpredictable given the leakage. Property (ii) means that the block cipher behaves as a PRP even if the adversary is allowed to ask en-/decryption queries under a *single* related key (next to the original key), so long as the related key is hard to guess.

As observed by an anonymous ToSC 2021 reviewer, the notion of FRK security as defined in [FKM18] is impossible to achieve whenever the tampering function f is allowed to

[1] Of course, for this to work we must assume that the computations carried over by the underlying cryptographic primitive are tamper proof.

[2] One should think of 2λ as the length of the block cipher key for security level λ; the reason for the factor 2 is related to the security assumptions made on the block cipher.

depend on the block cipher.[3] There is a simple attack (originally due to Bernstein [Ber10]) against FRK security. Before describing the attack, we give more details on the definition of FRK security from [FKM18]. As mentioned, the attacker is allowed to specify a tampering function f that is applied to the secret key. The security experiment samples a uniformly random key κ and computes $\tilde{\kappa} = f(\kappa)$. At this point, depending on the challenge bit b, the adversary gets oracle access either to the oracles $\mathsf{Encrypt}(\kappa, \cdot), \mathsf{Decrypt}(\kappa, \cdot), \mathsf{Encrypt}(\tilde{\kappa}, \cdot)$ and $\mathsf{Decrypt}(\tilde{\kappa}, \cdot)$ or to two uniformly random permutations π and π' and their respective inverses. The security guarantee states that, for any tampering function f, the adversary cannot tell the difference between these two scenarios except with negligible advantage. It is easy to see that, without further restrictions, this notion is not achievable. Indeed, an adversary can fix $\tilde{\kappa}$ to be the 0^k (where k is the key length) string and trivially distinguish the two scenarios. Thus, the FRK security experiment additionally first checks that the pre-image of $\tilde{\kappa}$ under the function f is a set with at most $2^{k/2}$ elements, and, if not, the experiment aborts and the adversary is not allowed to query the oracle. Thanks to this restriction the related key $\tilde{\kappa}$ is hard to guess if κ is chosen uniformly at random.

Now we are ready to describe the attack. Fix a block cipher $\Pi = (\mathsf{Encrypt}, \mathsf{Decrypt})$ and consider the function f that outputs $\lceil \mathsf{Encrypt}(\kappa, 0^m) \rceil_k$, where k is the key size, m is the input size, and $\lceil w \rceil_k$ denotes the first k bits of a string w. Since $f(\kappa)$ is a truncated evaluation of the block cipher, which behaves like a PRP, the tampering function f satisfies the property that $f(\kappa)$ is hard to predict for a random κ, and so the check on the preimage size described in the previous paragraph does not cause the experiment to abort except with extremely small probability. In fact, if there were many keys that map 0^m to the same value then we would have a distinguisher against the (standard) PRP security of Π. The attacker against FRK security using this tampering function f behaves as follows:

1. **Extract Tampered Key.** Obtain an encryption of 0^m under κ from the oracle, call it x.

2. **Test Oracle.** Obtain an encryption of 0^m under the tampered key $f(\kappa)$ from the oracle, call it y; compute offline the encryption $z = \mathsf{Encrypt}(\lceil x \rceil_k, 0^m)$ and check if $z = y$.

In the real world, we have $z = y$ with probability one, while if $\mathsf{Encrypt}$ is replaced by an independent and truly random permutation in the oracle, then $z = y$ only with very small probability. This contradicts the FRK security assumption from [FKM18], as it ranges over *all* (hard to guess) functions, even ones that can depend on the block cipher. Note that this attack applies even in the ideal cipher model (i.e., assuming that the block cipher behaves like a truly random permutation for every choice of the key).

An analogous argument shows that the notion of PRP-with-leakage security as defined in [FKM18] is impossible to achieve whenever the leakage function g is allowed to depend on the block cipher. The latter can be seen by considering the leakage function g that returns the first bit of $\mathsf{Encrypt}(\kappa, 0^m)$ and later obtains an encryption of 0^m from the oracle. Clearly, the secret key is still unpredictable given the leakage; yet, the attacker can distinguish the block cipher from a truly random permutation by comparing the output of the leakage function with the first bit of the output obtained from the oracle.

The above attacks fit into a class of cipher-dependent attacks studied by Albrecht, Farshim, Paterson, and Watson [AFPW11] in the context of modelling related-key attacks in the ideal cipher model. Their class includes the attack of Bernstein [Ber10], as well as another attack by Harris [Har09]. This discussion showcases the subtle challenges imposed by cipher-dependent attacks, and we find it interesting to study how to handle such attacks with as little impact as possible on the performance of the candidate schemes.

[3] An earlier version of this paper which made use of the faulty assumptions from [FKM18] was submitted to ToSC 2021.

1.2 Our Contributions

In this paper, we put forward a meaningful weakening of FRK security which intrinsically rules out the above cipher-dependent attacks, while still being sufficient to formally prove security of a slight tweak of the original construction by Fehr, Karpman, and Mennink. The codeword size in our construction can be as small as $m + 3\lambda$. Furthermore, we provide evidence of the robustness of our new assumption by proving that it holds unconditionally in the ideal cipher model with respect to a broad class of tampering functions covering, in particular, Bernstein's attack. We elaborate on these contributions below.

1.2.1 Entropic Fixed-Related-Key Security

In Section 3, we consider a different form of FRK security in which the attacker has a limited access to the encryption and decryption oracle under the original key κ. In particular, our notion of security relaxes the definition of [FKM18], which we discussed in Section 1.1, in two ways:

1. The attacker has arbitrary oracle access to the encryption and decryption oracle under the tampered key, but it is allowed to observe the output of the block cipher under the original key κ only for random inputs sampled from a distribution with sufficiently high min-entropy.

2. We do not require indistinguishability from a random permutation for the block cipher instantiated with the tampered key.

We refer to our notion as *entropic* FRK security. Briefly, the rationale for the first relaxation is that such a limited access to the encryption oracle under the original key would rule out the "Extract Tampered Key" part of the aforementioned classes of cipher-dependent attacks from Section 1.1; the rationale for the second relaxation is that having access to the *real-world* encryption and decryption oracles under the tampered key, independently of the challenge bit, would rule out the "Test Oracle" part of the aforementioned classes of cipher-dependent attacks. To understand why this is indeed the case, consider the following scenario: The oracle samples n messages $\mu_1, \ldots, \mu_n$ independently from a distribution $\mathcal{D}$ with s bits of min-entropy, i.e.,

$$\forall \mu \in \{0,1\}^m : \Pr_{D \sim \mathcal{D}}[D = \mu] \leq 2^{-s}.$$

Then, the adversary learns the message/ciphertext pairs $(\mu_i, \gamma_i = \mathsf{Encrypt}(\kappa, \mu_i))_{i \in [n]}$; the attacker has no further oracle access to $\mathsf{Encrypt}(\kappa, \cdot)$. In order to carry out the cipher-dependent related-key attack described in Section 1.2, the adversary musts learn the encryption under κ of a message that was also queried by the tampering function f on input κ. Let τ_{tamp} denote the running time of f (so that $f(\kappa)$ can compute encryptions and decryptions of at most τ_{tamp} inputs). By a union bound, the probability that f queries the cipher on one of the messages $\mu_1, \ldots, \mu_n$ or ciphertexts $\gamma_1, \ldots, \gamma_n$ is at most

$$2n \cdot \tau_{\mathsf{tamp}} \cdot 2^{-s}. \tag{1}$$

Therefore, if the min-entropy parameter s satisfies $s \gg \log \tau_{\mathsf{tamp}} + \log n$, it follows that the probability that the cipher-dependent attack above succeeds is extremely small.

Moreover, we notice that the second relaxation means that we do not require any privacy guarantee from the block cipher instantiated with the tampered key, besides that oracle access to $\mathsf{Encrypt}(f(\kappa), \cdot)$ and $\mathsf{Decrypt}(f(\kappa), \cdot)$ for a tampering function f cannot help in breaking the privacy of the ciphertexts computed under the original key. In contrast, Fehr, Karpman and Mennink require indistinguishability of the block cipher from a random permutation to hold even with respect to the tampered key. For this reason, their definition requires that the tampered key cannot be easily guessed by the adversary, as otherwise

there would be a trivial distinguisher. By giving up on the indistinguishability of the block cipher under the tampered key, we additionally gain that we do not need anymore any restriction on the unpredictability of the tampered key. Informally speaking, the attacker could decide to tamper the original key and set it to an "easy to guess" tampered key $\tilde{\kappa}$, thus receiving oracle access to $\mathsf{Encrypt}(\tilde{\kappa}, \cdot)$ and $\mathsf{Decrypt}(\tilde{\kappa}, \cdot)$ (independently of the challenge bit). If such a tampered key is easy to guess, however, the same oracle access could have been simulated by the adversary *in its head*. Thus, predictable tampered keys cannot harm the security definition and can be allowed.

1.2.2 Our Construction

The non-malleable code construction we consider is a slight variation of the original construction by [FKM18]. Namely, in Section 4, we consider the non-malleable code in the split-state model that encodes a message μ as described below:

$$\mathsf{Encode}(\mu) \to (\kappa, \mathsf{Encrypt}(\kappa, \mu \| \rho)) \,,$$

where κ is a uniformly random secret key and ρ is a uniformly random λ-bit string. Notice that the only difference between our construction and the construction of Fehr, Karpman, and Mennink is that we additionally sample a random string ρ and encrypt the concatenation of $\mu \| \rho$.

As a bonus, we substantially simplify the security analysis. Indeed, the original security proof involves a case analysis according to the unpredictability of the tampered codeword: if the tampered key $\tilde{\kappa}$ is unpredictable, then the security of the non-malleable code reduces to the FRK security of the block cipher. Otherwise, it reduces to the PRP-with-leakage security of the block cipher. In the latter case, the reduction leaks the full tampered key and uses this leakage to simulate oracle access to $\mathsf{Decrypt}(\tilde{\kappa}, \cdot)$.

In our case, entropic FRK security directly provides oracle access to $\mathsf{Decrypt}(\tilde{\kappa}, \cdot)$, and it considers such an oracle as the only *leakage* an adversary can get from a tampering attack. This allows us to bypass the case analysis and reduce directly to the entropic FRK security of the block cipher (independently of the unpredictability of the tampered key), without explicitly assuming any form of leakage resilience from the block cipher.

1.2.3 Security in the Ideal Cipher Model

Recall that Bernstein's cipher-dependent attack on the FRK security notion from [FKM18] described in Section 1.1 applies even in the ideal cipher model. To further validate our approach, in Section 5, we prove that, unlike the notion of FRK security, entropic FRK security does hold (unconditionally) in the ideal cipher model, albeit with respect to a restricted (but still broad) family of tampering functions which includes cipher-dependent attacks such as Bernstein's.

The main ideas behind our analysis follow what we already described in Section 1.2.1. The intuition is that the tampering function, even with access to the original key κ, cannot query the ideal cipher on the challenge messages, because those messages are sampled from a distribution with high min-entropy. Thus, the tampered key is independent of the challenge ciphertexts, and so are the queries to the tampered encryption and decryption oracle. It is easy to see that, in the ideal cipher model, the only way for the adversary to distinguish the ideal from the real experiment is to query the ideal cipher on the original key. Thus, we give a bound on the unpredictability of the original key when the adversary has oracle access to the tampered encryption and decryption oracle. However, we need to make a simplifying assumption on the structure of the tampering functions. The problem is that the tampered key could be chosen as a function of the outputs of the ideal cipher queried on the tampered key itself. In principle, this allows for rejection-sampling adversarial strategies that can leak partial information about the original key. For example, consider

the tampering function that sets the tampered key to an arbitrary string $\tilde{\kappa}$ such that the first bit of an encryption of the all-zero string under $\tilde{\kappa}$ matches the first bit of the original key. The adversary can leak the first bit of κ through its oracle access. While these kind of tampering functions do not seem to help breaking entropic FRK security, nevertheless they make the analysis more complicated as they can bias in unexpected ways the distribution of the original key given oracle access to the ideal cipher and the tampered encryption and decryption oracle. Specifically, we cannot anymore easily argue that the output of the ideal cipher queried on the tampered key is independent of the tampered key.

Our solution to avoid these contrived tampering attacks is to additionally assume that the tampering functions do not query the ideal cipher on the tampered key. We notice that this additional assumption holds true for the tampering functions of Bernstein's [Ber10] and Harris' attacks [Har09] as discussed in [AFPW11]. Moreover, we point out that such a restriction was already considered by Albrecht, Farshim, Paterson and Watson [AFPW11] under the more generic notion of oracle-independence. We conjecture that full entropic FRK security holds in the ideal cipher model, and leave a formal proof of this fact as an interesting open problem.

Remark 1. It is natural to wonder whether one can prove the security of our proposed construction directly in the ideal cipher model. While we believe that this would indeed be possible, we do not pursue this direction. Our main goal is to base security on a falsifiable assumption which is plausibly satisfied by real-world block ciphers and is easier to evaluate in practice. Moreover, we believe that the notion of Entropic FRK security might have other applications (for example, hybrid encryption and tamper-resilient secret-key encryption).

1.2.4 Parameter Instantiations

In Section 6, we give two possible parameter instantiations for our construction. The first instantiation simply assumes that practical block ciphers (such as AES-128 and SHACAL-2) directly have good entropic FRK security; this yields codewords of size close to $m + 3\lambda$ at about λ bits of security.

Alternatively, we can be more conservative and consider the advantage upper bound on entropic FRK security we establish in the ideal cipher model; this yields slightly longer codeword size close to $m + 5\lambda$ at about λ bits of security.

1.3 Related Work

A long line of research explores constructions of non-malleable codes in the split-state model, both with information-theoretic [DPW10, DKO13, ADL14, CG16, CG17, ADKO15, CGL16, Li17, Li19, AO20, AKO+22] and computational [LL12, AAG+16, KLT16] security. Currently, the best explicit non-malleable code in the information-theoretic setting achieves rate 1/3 [AKO+22] (versus 1/2, which is the best possible rate in the information-theoretic setting [CG16]). More precisely, this means that if m denotes the message length and $n = n(m)$ denotes the codeword length corresponding to m-bit messages, then

$$\frac{m}{n} \to \frac{1}{3}$$

when $m \to \infty$. Aggarwal *et al.* [AAG+16] show how to compile an information-theoretic non-malleable code in the split-state model with rate bounded away from 1 into a non-malleable code with much lower redundancy in the computational setting. Their construction encodes the secret key κ of a secret key encryption scheme under the poor-rate non-malleable code, obtaining a codeword (σ_0, σ_1), and then encrypts the message μ under the key κ obtaining a ciphertext γ. The final encoding is $\sigma'_0 = (\sigma_0, \gamma)$ and $\sigma'_1 = \sigma_1$. Therefore, encoding m-bit messages using a k-bit key under this construction leads to

codewords of length

$$m + n(k), \tag{2}$$

where $n(k)$ is the codeword length of the underlying information-theoretic non-malleable code on k-bit messages. The security proof requires the encryption scheme to be non-malleable, i.e., a so-called authenticated encryption scheme, and the underlying non-malleable code to satisfy a slightly stronger non-malleability flavor known as *augmented* non-malleability, which is satisfied by the construction in [AKO+22].

Given the above, it is natural to compare our construction with the one obtained by combining the compiler of Aggarwal *et al.* and the best known information-theoretic non-malleable code. In general, known constructions of information-theoretic non-malleable codes rely heavily on tools from pseudorandomness, such as randomness extractors, which suffer from large hidden constants in their various parameters and from an impractical running time (although asymptotically polynomial), such as the GUV seeded extractor [GUV09]. With this in mind:

1. The codeword length obtained by combining [AAG+16] and [AKO+22] according to Eq. (2) would be
$$m + 3k + o(k), \tag{3}$$
where the $o(k)$ term satisfies $\frac{o(k)}{k} \to 0$ when $k \to \infty$ but hides a large constant.

2. As mentioned above, the running time of our encoding/decoding algorithms essentially only involve evaluating the block cipher, while the encoding/decoding of [AKO+22] involves objects from pseudorandomness whose running time is impractical.

3. The resulting security error would be at least the statistical security error $\epsilon(k)$ of the underlying information-theoretic non-malleable code on k-bit messages. The rate-1/3 code from [AKO+22] achieves statistical error $2^{-\Omega(k/\log^3 k)}$, with $\Omega(\cdot)$ hiding a big constant. Hence, even ignoring hidden constants, one needs to take the key length k to be $k > \lambda \cdot \log^3(\lambda)$ in order to get overall (computational) security error $2^{-\lambda}$. For usual values of the security parameter (say, $\lambda = 256$), this implies an extra multiplicative factor of at least $8^3 = 512$.

Combining Eq. (3) with the last item, we conclude that, even ignoring the large hidden constants and impractical encoding/decoding procedures, achieving security error comparable to $2^{-\lambda}$ would require codewords of length larger than $m + 200\lambda$ for currently reasonable values of λ. In contrast, our construction only requires codewords of length close to $m + 3\lambda$ or $m + 5\lambda$, depending on how conservative one is, to achieve the same security level, and is easy to implement in practice since it only involves encoding/decoding via a single call to a block cipher. Without considering [FKM18], the best previous construction [KLT16] with concrete security required codewords of length $m + 18\lambda$ (or $m + 9\lambda + 2\log^2 \lambda$) to obtain security $2^{-\lambda}$, and relied on non-falsifiable assumptions.

2 Preliminaries

2.1 Notation

We denote by $[n]$ the set $\{1, \ldots, n\}$; for any $a \leq b$, we let $[a, b] := \{a, \ldots, b\}$. For a string $x \in \{0,1\}^*$, we denote its length by $|x|$. We denote sets by calligraphic letters such as $\mathcal{X}$. The size of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. When x is chosen randomly from $\mathcal{X}$, we write $x \leftarrow\!\!\$\, \mathcal{X}$. We denote the family of permutations over a set $\mathcal{X}$ by $\mathcal{P}(\mathcal{X})$. Sometimes we will denote the family $\mathcal{P}(\{0,1\}^m)$, for a natural number m, with $\mathcal{P}(m)$. Similarly, we denote the family of keyed-permutations with keys ranging over $\{0,1\}^k$ and permutation set $\{0,1\}^m$ by $\mathcal{P}(k, m)$. If $\mathcal{I} \subseteq [n]$ is a set and $x \in \mathcal{S}^n$ is a string, we define the projection

$x_{\mathcal{I}} = (x_i)_{i\in\mathcal{I}}$. When A is a randomized algorithm, we write $y \leftarrow\!\!\$\ \mathsf{A}(x)$ to denote a run of A on input x (and implicit random coins ρ) and output y; the value y is a random variable and $\mathsf{A}(x;\rho)$ denotes a run of A on input x and randomness ρ.

2.2 Non-Malleable Codes in the Split-State Model

We start by giving the definition of coding schemes in the split-state model.

Definition 1. A split-state coding scheme Σ with *message space* $\mathcal{M}$ and *codeword space* $\mathcal{S} = \mathcal{S}_0 \times \mathcal{S}_1$ is a pair of algorithms $(\mathsf{Encode}, \mathsf{Decode})$ such that (i) Encode is a *randomized* encoding function $\mathsf{Encode} : \mathcal{M} \to \mathcal{S}$, (ii) Decode is a *deterministic* decoding function $\mathsf{Decode} : \mathcal{S} \to \mathcal{M} \cup \{\perp\}$ and (iii) for all $\mu \in \mathcal{M}$, $\Pr\left[\mathsf{Decode}(\mathsf{Encode}(\mu)) = \mu\right] = 1$ (over the randomness of the encoding algorithm). In the above, $\perp$ is a special symbol stating that the input codeword is invalid.

Let $\Sigma = (\mathsf{Encode}, \mathsf{Decode})$ be a 2-split state coding scheme. Fix any messages $\mu_0, \mu_1 \in \mathcal{M}$ and arbitrary *tampering functions* $f_0 : \mathcal{S}_0 \to \mathcal{S}_0$ and $f_1 : \mathcal{S}_1 \to \mathcal{S}_1$ with running time τ_0 and τ_1 respectively. For $b \in \{0,1\}$, consider the experiment:

$$\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0, \mu_1, b) := \widetilde{\mathsf{Decode}}(\tilde{\sigma}_0, \tilde{\sigma}_1),$$

where $(\sigma_0, \sigma_1) \leftarrow\!\!\$\ \mathsf{Encode}(\mu_b)$ and $\tilde{\sigma}_j = f_j(\sigma_j)$ for all $j \in \{0,1\}$, and where $\widetilde{\mathsf{Decode}}$ is the algorithm that outputs $\diamond$ if and only if $\tilde{\mu} := \mathsf{Decode}(\tilde{\sigma}_0, \tilde{\sigma}_1) \in \{\mu_0, \mu_1\}$ and returns $\tilde{\mu}$ otherwise.

Definition 2 (Non-Malleability)**.** Let $\Sigma = (\mathsf{Encode}, \mathsf{Decode})$ be a split-state coding scheme. The *non-malleability advantage* of Σ is

$$\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau) := \max_{\substack{\mu_0,\mu_1 \\ \mathsf{A}_\tau, f_0, f_1}} \left| \Pr\left[\mathsf{A}_\tau(\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0,\mu_1,0)) = 1\right] - \Pr\left[\mathsf{A}_\tau(\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0,\mu_1,1)) = 1\right] \right|,$$

where the maximum is over all $\mu_0, \mu_1 \in \mathcal{M}$, all algorithms A_τ running in time at most τ, and all (f_0, f_1) with running time at most τ.

Informally, the goal of an adversary A_τ is to learn whether the encoded message is μ_0 or μ_1, and the only information A_τ gets to see is the result on the reconstructed message corresponding to the tampering functions (f_0, f_1) of his choice. Intuitively, by requiring for $\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau)$ to be small, we are saying that no adversary (running in time at most τ) is able to tell the difference between μ_0 and μ_1 with meaningful advantage.

3 Tamper-Resilient Block Ciphers

A block cipher $\Pi = (\mathsf{Encrypt}, \mathsf{Decrypt})$ is a pair of polynomial-time algorithms specified as follows:

- The deterministic encryption algorithm takes as input a key $\kappa \in \{0,1\}^k$ and a message $\mu \in \{0,1\}^m$, and outputs a ciphertext $\gamma \in \{0,1\}^m$.
- The deterministic decryption algorithm takes as input a key $\kappa \in \{0,1\}^k$ and a ciphertext $\gamma \in \{0,1\}^m$, and outputs a value in $\{0,1\}^m$.

We require the block cipher satisfies perfect correctness, namely for all $\kappa \in \{0,1\}^k$ and $\mu \in \{0,1\}^m$ it holds that $\mathsf{Decrypt}(\kappa, \mathsf{Encrypt}(\kappa, \mu)) = \mu$.

We proceed to discuss the notion of tamper resilience we require for the block ciphers we use. Before presenting the main definition of this section, we introduce the notion of *(samplable) s-entropic distributions.*

Definition 3 (Entropic distribution)**.** We say that a family of distributions $\mathcal{D}(1^\lambda, \mathsf{aux})$, parameterized by a security parameter λ and an auxiliary input aux, is *s-entropic* for a function $s(\cdot)$ if for any λ, aux, and any possible output x we have

$$\Pr[\mathcal{D}(1^\lambda, \mathsf{aux}) = x] \leq 2^{-s(\lambda)}.$$

We say that $\mathcal{D}$ is *τ_{samp}-samplable* if there is a randomized algorithm running in time τ_{samp} which generates a sample of $\mathcal{D}(1^\lambda, \mathsf{aux})$ given $(1^\lambda, \mathsf{aux})$ as input.

Let $\mathcal{O}_{\Pi,f,\mathcal{D},n}(b)$ be an oracle depending on block cipher Π, an arbitrary tampering function $f : \{0,1\}^k \to \{0,1\}^k$, an arbitrary s-entropic distribution $\mathcal{D}$ supported on the message space $\{0,1\}^m$, and a natural number n. Whenever it is clear from context, we omit the parameters Π and n from the definition of the oracle. The oracle acts as follows when interacting with some adversary A who is allowed to make multiple queries:

- It chooses a uniformly random key $\kappa \leftarrow\!\!\$\ \{0,1\}^k$.
- It samples n messages $\mu_1^*, \ldots, \mu_n^*$ independently according to the distributions $\mathcal{D}(1^\lambda, j)$, for $j = 1, \ldots, n$. If $b = 0$, it computes $\gamma_j^* = \mathsf{Encrypt}(\kappa, \mu_j^*)$, and otherwise computes $\gamma_j^* \leftarrow \pi(\mu_j^*)$ where $\pi \leftarrow\!\!\$\ \mathcal{P}(\{0,1\}^m)$.
- The oracle reveals the message-ciphertext pairs $(\mu_j^*, \gamma_j^*)_{j\in[n]}$ to the adversary A and answers its queries as follows:
  - Upon input $(\mathtt{enc}, \mu)$, if $\mu = \mu_j^*$ for some j and $f(\kappa) = \kappa$ then return the special symbol $\diamond$. Else it returns $\mathsf{Encrypt}(f(\kappa), \mu)$.
  - Upon input $(\mathtt{dec}, \gamma)$, if $\gamma = \gamma_j^*$ for some j and $f(\kappa) = \kappa$ then return the special symbol $\diamond$. Else it returns $\mathsf{Decrypt}(f(\kappa), \gamma)$.

Note that this oracle differs from the one used in the FRK security definition from [FKM18] (discussed in Section 1.1) because it does not allow unrestricted query access to the functions $\mathsf{Encrypt}(\kappa, \cdot)$ and $\mathsf{Decrypt}(\kappa, \cdot)$. Namely, the adversary only observes outputs of these functions on messages sampled independently from some distribution with enough min-entropy.

Definition 4 (Entropic FRK security)**.** For any function $f : \{0,1\}^k \to \{0,1\}^k$ and a distribution $\mathcal{D}$ consider the oracle $\mathcal{O}_{\mathtt{frk}} := \mathcal{O}_{f,\mathcal{D}}$. For a block cipher Π, we define the FRK advantage of Π as:

$$\mathbf{Adv}_{\Pi}^{\mathtt{frk}}(q, \tau, k, s, n) := \max_{\mathsf{A}_{q,\tau},\mathcal{D},f} \left| \Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(0)}() = 1\right] - \Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(1)}() = 1\right] \right|,$$

where the probability above is taken over $\mu_j \leftarrow\!\!\$\ \mathcal{D}(1^\lambda, j)$ independently for all $j \in [n]$ and $\kappa \leftarrow\!\!\$\ \{0,1\}^k$, and where the first maximum is over all algorithms $\mathsf{A}_{q,\tau}$ running in time at most τ and asking q oracle queries and over all s-entropic distributions running in time at most τ, and all tampering functions $f : \{0,1\}^k \to \{0,1\}^k$ running in time at most τ.

We note that in this work we only require Entropic FRK security with respect to $q = 1$ oracle queries and $n = 1$ message-ciphertext pairs. Nevertheless, we define this notion with respect to general q and n as we believe it may find applications elsewhere at this level of generality.

4 Our Construction

Our construction of non-malleable codes in the split-state model is based only on block ciphers and is depicted in Fig. 1. The encoding of a message μ is (κ, γ), where κ is a

Let $\Pi = (\mathsf{Encrypt}, \mathsf{Decrypt})$ be a block cipher and consider the split-state coding scheme $\Sigma = (\mathsf{Encode}, \mathsf{Decode})$ defined below.

$\underline{\mathsf{Encode}(\mu)}$: The encoding algorithm proceeds as follows:

1. Sample $\kappa \leftarrow_{\$} \{0,1\}^k$, $\rho \leftarrow \{0,1\}^s$ and compute $\gamma := \mathsf{Encrypt}(\kappa, \mu \| \rho)$;
2. Return $\sigma := (\kappa, \gamma)$.

$\underline{\mathsf{Decode}(\sigma)}$: The decoding algorithm proceeds as follows:

1. Parse σ as κ, γ;
2. Return μ where $\mu \| \rho \leftarrow \mathsf{Decrypt}(\kappa, \gamma)$.

Figure 1: Description of our non-malleable code.

random block cipher key and γ is an encryption of the string $\mu||\rho$ for a random λ-bit string ρ. The decoding algorithm simply decrypts the ciphertext and discards the last λ bits of the resulting plaintext.

The theorem below characterizes the non-malleability of our code in terms of the security of the underlying block cipher.

Theorem 1. *Let Π be a block cipher and let $\tau' = 2\tau + O(1)$. Then, the split-state coding scheme Σ described in Fig. 1 satisfies*

$$\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau) \leq 2\mathbf{Adv}^{\mathtt{frk}}_{\Pi}(1, \tau', k, s, 1)\,.$$

We give the main ideas behind the proof of the theorem. Given a tampering function (f_0, f_1) against the non-malleable code, our reduction uses f_0 to define the related key $\tilde{\kappa}$ in the Entropic FRK security experiment, and uses f_1 to compute a ciphertext $\tilde{\gamma}$ that can be decrypted using its oracle access to the tampered decryption oracle, thus providing a decoding of the codeword. We can then replace the real codeword from $(\kappa, \mathsf{Encrypt}(\kappa, \mu\|\rho))$ with a fake codeword (κ, ω), where ω is an uniformly random string. Clearly, such a fake codeword is independent of the encoded message. We now present the formal proof.

Proof. Consider an adversary A running in time τ and tampering function (f_0, f_1) with running time $\tau_{\mathsf{tamp}} \leq \tau$ which maximizes the advantage, i.e., such that

$$\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau) = \Big|\Pr\left[\mathsf{A}(\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0, \mu_1, 0)) = 1\right] - \Pr\left[\mathsf{A}(\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0, \mu_1, 1)) = 1\right]\Big|\,.$$

Let the hybrid experiment $\mathbf{Exp}_1(\mu_0, \mu_1, b)$, for $b \in \{0,1\}$, be the same as $\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0, \mu_1, b)$ except that it computes $\gamma = \pi(\mu_b \| \rho)$, where π is a uniformly random permutation. Let $\mathcal{D}(1^\lambda, 1)$ be the distribution which outputs a string $\mu||\rho$ such that $\mu = \mu_b$ and ρ is uniformly distributed over $\{0,1\}^s$. Let B be the following adversary against Entropic FRK security of Π with related-key function f_0.

Adversary $\mathsf{B}()$ with oracle access to $\mathcal{O}_{\mathtt{frk}}(b) = \mathcal{O}_{\Pi, f_0, \mathcal{D}, n}(b)$:

1. Receive from the oracle the pair $(\mu_b||\rho, \gamma)$, where $\mu_b||\rho \leftarrow_{\$} \mathcal{D}(1^\lambda, 1)$ and γ is either the output of $\mathsf{Encrypt}(\kappa, \mu_b||\rho)$ or the output of $\pi(\mu_b||\rho)$ depending on the experiment.
2. Compute $\tilde{\gamma} = f_1(\gamma)$.
3. Query the oracle $\mathcal{O}_{\mathtt{frk}}$ with $(\mathtt{dec}, \tilde{\gamma})$, thus receiving the message $\tilde{\mu}\|\tilde{\rho}$. Note that f_0 is used implicitly in this step, since the oracle $\mathcal{O}_{\mathtt{frk}}$ attempts to decrypt $\tilde{\gamma}$ using the related key $f_0(\kappa)$.

4. If $\tilde{\mu} \in \{\mu_0, \mu_1\}$, set $\tilde{\mu} = \diamond$.
5. Run $b' \leftarrow \mathsf{A}(\tilde{\mu})$.
6. Return 0 if $b' = b$ and return 1 otherwise.

Observe that B computes f_1, runs A and performs some constant-time operation. Therefore, its running time is bounded by $\tau + \tau_{\mathsf{tamp}} + O(1) \leq \tau'$. Also notice that, by design of the FRK experiment, when both $\tilde{\kappa} = \kappa$ and $\tilde{\gamma} = \gamma$ the adversary B sets the message $\tilde{\mu}$ to be the output of the decryption oracle that is equal to $\diamond$. Finally, we observe that the simulation performed by B is perfect. We can verify this by inspection. Indeed, when B is interacting with $\mathcal{O}_{\mathtt{frk}}(0)$, by definition of the security game for FRK we have that $\gamma = \mathsf{Encrypt}(\kappa, \mu \| \rho)$ as in experiment $\mathbf{Exp}^{\mathtt{nm}}$. On the other hand, when B is interacting with $\mathcal{O}_{\mathtt{frk}}(1)$, we have that $\gamma = \pi(\mu \| \rho)$ as in experiment $\mathbf{Exp}_1$. Therefore,

$$\begin{aligned}&\left|\Pr\left[\mathsf{A}(\mathbf{Exp}^{\mathtt{nm}}_{\Sigma}(\mu_0, \mu_1, b)) = 1\right] - \Pr\left[\mathsf{A}(\mathbf{Exp}_1(\mu_0, \mu_1, b)) = 1\right]\right| \\ &\quad = \left|\Pr\left[\mathsf{B}^{\mathcal{O}_{\mathtt{frk}}(0)}() = 1\right] - \Pr\left[\mathsf{B}^{\mathcal{O}_{\mathtt{frk}}(1)}() = 1\right]\right| \leq \mathbf{Adv}^{\mathtt{frk}}_{\Pi}(1, \tau', k, s, 1).\end{aligned}$$

Finally, we show that, for all $b \in \{0, 1\}$,

$$\Pr\left[\mathsf{A}(\mathbf{Exp}_1(\mu_0, \mu_1, 0)) = 1\right] = \Pr\left[\mathsf{A}(\mathbf{Exp}_1(\mu_0, \mu_1, 1)) = 1\right]$$

In fact, the only value that might depend on the bit b in the experiment $\mathbf{Exp}_1(\mu_0, \mu_1, 0)$ is the random variable $\gamma = \pi(\mu_b \| \rho)$. However, such random variable is uniformly distributed over $\{0, 1\}^m$. □

5 Entropic FRK Security in the Ideal Cipher Model

In this section, we show that entropic FRK security holds in the ideal cipher model for a natural family of cipher-dependent tampering functions (generalizing Bernstein's attack, as described in Section 1). Since Bernstein's attack applies equally well in the ideal cipher model, this result stands in sharp contrast with the fact that the notion of FRK security considered in [FKM18] is unachievable even in the ideal cipher model.

Before we proceed to state and prove the result, we formally define what is meant by entropic FRK security in the ideal cipher model w.r.t. a tampering family. The definition is almost identical to the one in Section 3, except that both the adversary and the tampering function are given oracle access to a random keyed permutation $\Pi \leftarrow\!\!\$\ \mathcal{P}(k, m)$. The latter essentially means that a random permutation is chosen for every possible key, and the attacker as well as the tampering function, can query such permutations in the forward direction (via queries $(\mathtt{enc\text{-}ideal}, \kappa', \mu)$) and in the backward direction (via queries $(\mathtt{dec\text{-}ideal}, \kappa', \gamma)$). An identical formalization was used in [AFPW11].

Definition 5 (Entropic FRK security in the ideal cipher model)**.** Consider the same oracle $\mathcal{O}_{\mathtt{frk}} := \mathcal{O}_{\Pi, f, \mathcal{D}, n}$ specified in Section 3, and let $\mathcal{F}$ be a family of cipher-dependent tampering functions $f^{\Pi} : \{0, 1\}^k \to \{0, 1\}^k$ which can make queries to the encryption and decryption algorithms of Π. We define the *entropic-FRK advantage in the ideal cipher model w.r.t.* $\mathcal{F}$ as:

$$\mathbf{Adv}^{\mathtt{frk\text{-}ideal}}_{\mathcal{F}}(q_{\mathcal{O}}, q_{\Pi}, k, m, s, n) := \max_{\mathsf{A}, \mathcal{D}, f \in \mathcal{F}} \left|\Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(0), \Pi}() = 1\right] - \Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(1), \Pi}() = 1\right]\right|,$$

where the probabilities above are taken over $\mu_j \leftarrow\!\!\$\ \mathcal{D}(1^{\lambda}, j)$ for all j, $\kappa \leftarrow\!\!\$\ \{0, 1\}^k$ and $\Pi \leftarrow\!\!\$\ \mathcal{P}(k, m)$, and where the maximum is over all algorithms A asking $q_{\mathcal{O}}$ oracle queries to $\mathcal{O}_{\mathtt{frk}}$, all s-entropic distributions $\mathcal{D}$, and over all tampering functions $f^{\Pi} \in \mathcal{F}$, where A and f ask cumulatively q_{Π} oracle queries to Π.

As discussed earlier, our security analysis in the ideal cipher model only holds w.r.t. a large class of cipher-dependent tampering functions which we name *oracle-independent*. The latter roughly means that $f^{\Pi}(\kappa)$ does not output a key κ' which was used as part of an $(\texttt{enc-ideal}, \kappa', \cdot)$ or a $(\texttt{dec-ideal}, \kappa', \cdot)$ query. A similar notion (which was probabilistic in nature and considered multiple tampering functions) was considered in [AFPW11].

Definition 6 (Oracle independence)**.** For any f, κ and Π, let $\mathcal{Q}_{f,\kappa,\Pi}$ be the set of keys κ' such that $f^{\Pi}(\kappa)$ queries $(\texttt{enc-ideal}, \kappa', \mu)$ for some μ or $(\texttt{dec-ideal}, \kappa', \gamma)$ for some γ. We say that f is *oracle-independent* if for any κ and Π we have $f^{\Pi}(\kappa) \notin \mathcal{Q}_{f,\kappa,\Pi}$. Moreover, we call $\mathcal{F}^*$ the set of all oracle-independent tampering functions.

Theorem 2. *For any parameters $q_{\mathcal{O}}, q_{\Pi}, k, m, s, n$ such that $q_{\Pi} \leq 2^{k/4}$ and $q_{\mathcal{O}} \leq 2^{m-1}$ we have*

$$\mathbf{Adv}^{\texttt{frk-ideal}}_{\mathcal{F}^*}(q_{\mathcal{O}}, q_{\Pi}, k, m, s, n) \leq 4q_{\Pi} \cdot (n \cdot 2^{-s} + 2^{-k/4} + 2^{-k/2}) + 6q_{\mathcal{O}} \cdot n \cdot 2^{-m}.$$

Proof. At a high level, we begin by introducing a hybrid experiment $\mathcal{H}(b)$ parameterized by a challenge bit b. Then, we show that the advantage of any given adversary A in distinguishing between $\mathcal{H}(0)$ and $\mathcal{H}(1)$ is appropriately close to the advantage of the same adversary in distinguishing between the original Entropic FRK security experiment described in Definition 5 with challenge bit $b = 0$ and $b = 1$, respectively. Finally, we argue that the advantage in distinguishing between the experiments $\mathcal{H}(0)$ and $\mathcal{H}(1)$ is small, which concludes the proof.

Let A be an adversary, f be a tampering function, and $\mathcal{D}$ be an s-entropic distribution such that A asks at most $q_{\mathcal{O}}$ oracle queries to $\mathcal{O}_{\texttt{frk}}$, f is oracle-independent, A and f ask cumulatively at most q_{Π} oracle queries to Π, and A, f, and $\mathcal{D}$ maximize the advantage in the FRK security experiment, i.e.,

$$\mathbf{Adv}^{\texttt{frk-ideal}}_{\mathcal{F}^*}(q_{\mathcal{O}}, q_{\Pi}, k, m, s, n) = \left| \Pr\left[\mathsf{A}^{\mathcal{O}_{\texttt{frk}}(0),\Pi}() = 1\right] - \Pr\left[\mathsf{A}^{\mathcal{O}_{\texttt{frk}}(1),\Pi}() = 1\right] \right|. \quad (4)$$

We now define the hybrid experiment $\mathcal{H}(b)$ where we run A, f, and $\mathcal{D}$ in an experiment that is similar to the original FRK experiment with challenge bit b but where the hybrid experiment aborts and outputs $\perp$ if certain events hold. We describe the hybrid experiment in the following and in pseudo-code in Fig. 2. Specifically, the oracles $\mathcal{O}_{\texttt{frk}}$ and Π in the hybrid $\mathcal{H}(b)$ are modified to raise flags flg_1, flg_2, and flg_3. The hybrid experiment returns $\perp$ if at least one of the flags is set to 1 during the experiment, otherwise it returns the response bit output by A. Notice that in the pseudo-code description we assume that all the variables are shared between the hybrid experiment and the two oracles. Thus, for example, the flags are initially set to 0 by the hybrid experiment and might be updated (and set to 1) at each invocation of the oracles by the adversary. For $i = 1, 2, 3$ we define the event E_i to be the event that the flag flg_i is set to 1. The events are as follows:

- **Event** E_1. The function $f^{\Pi}(\kappa)$ sends a query to Π of the form $(\texttt{enc-ideal}, \kappa', \mu^*_j)$ or $(\texttt{dec-ideal}, \kappa', \gamma^*_j)$ for some $j \in [n]$ and some κ'. In this case, we says that $f^{\Pi}(\kappa)$ *queries on* μ^*_j. (Recall that $\mu^*_1, \ldots, \mu^*_n$ are the messages sampled by the sampler $\mathcal{D}$, and $\gamma^*_1, \ldots, \gamma^*_n$ are the corresponding ciphertexts.)
- **Event** E_2. The adversary A queries the secret key κ to Π.
- **Event** E_3. The adversary A finds a collision. Namely, it holds that $f^{\Pi}(\kappa) = \kappa$ and either the adversary sends a query of the form $(\texttt{enc}, x)$ with $x \notin \{\mu^*_1, \ldots, \mu^*_n\}$ and receives $\mathsf{Encrypt}(\kappa, x) \in \{\gamma^*_1, \ldots, \gamma^*_n\}$, or sends a query of the form $(\texttt{dec}, y)$ with $y \notin \{\gamma^*_1, \ldots, \gamma^*_n\}$ and receives $\mathsf{Decrypt}(\kappa, y) \in \{\mu^*_1, \ldots, \mu^*_n\}$.

Note that the events E_i may at first sight have different probabilities in $\mathcal{H}(0)$ and $\mathcal{H}(1)$. In fact, as we shall argue, E_1 and E_2 have the same probability of happening in $\mathcal{H}(0)$ and

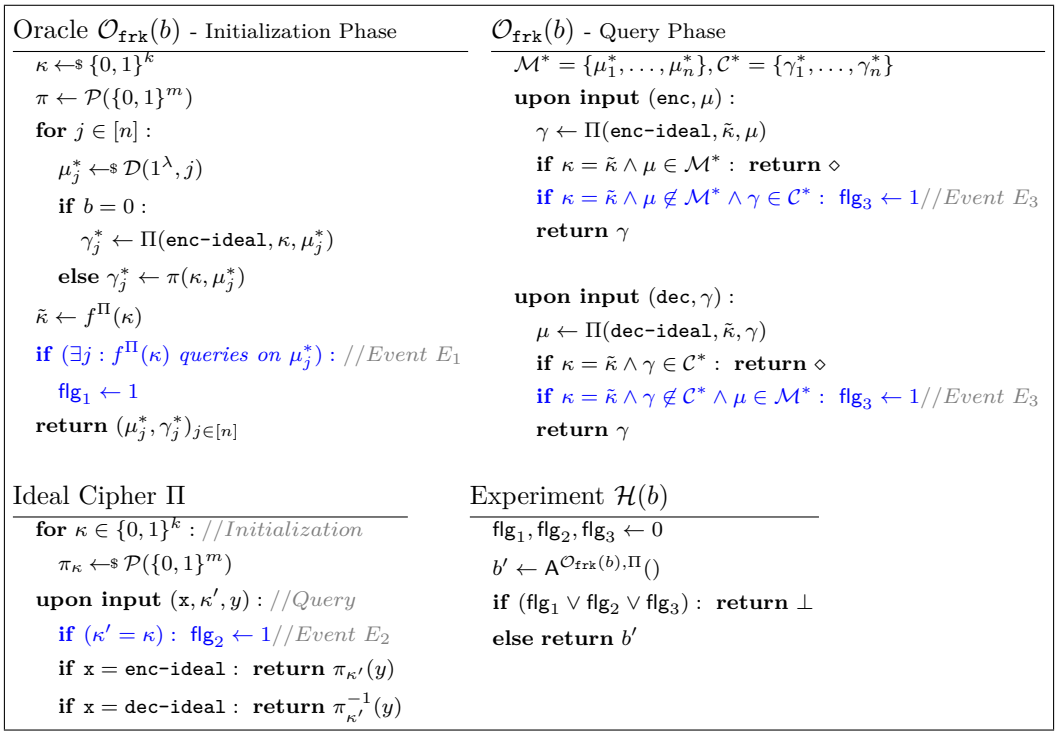

Figure 2: The hybrid experiment $\mathcal{H}(b)$ and the corresponding modified oracles. The differences in the oracles between the original security experiment in Definition 5 and the hybrid experiment are highlighted in blue.

$\mathcal{H}(1)$, while E_3 does not. To avoid overloading the notation, we avoid explicitly writing down whether we are referring to event E_i in $\mathcal{H}(0)$ or $\mathcal{H}(1)$ as this will always be clear from context.

By inspection of the hybrid experiment we can notice that if the flags are not raised, namely if $(\neg E_1 \wedge \neg E_2 \wedge \neg E_3)$, then the hybrid experiment and the FRK experiment are exactly the same. In fact, the changes introduced in the hybrid do not influence the outputs of the oracles. Thus it follows that for $b \in \{0,1\}$ we have

$$\left|\Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(b),\Pi}() = 1\right] - \Pr\left[\mathcal{H}(b) = 1\right]\right| \leq \Pr\left[E_1\right] + \Pr\left[E_2|\neg E_1\right] + \Pr\left[E_3|\neg E_2 \wedge \neg E_1\right]. \tag{5}$$

We proceed to bound the three rightmost terms appropriately. First, notice that for any choice of Π, index j, and key κ', the probability that the i-th query of f^Π is of the form $(\mathtt{enc\text{-}ideal}, \kappa', \mu_j^*)$ or $(\mathtt{dec\text{-}ideal}, \kappa', \gamma_j^*)$ is at most $2^{-s(\lambda)}$ since $\mathcal{D}$ is s-entropic and the message samples from $\mathcal{D}$ are i.i.d. for different j. Taking a union bound over all $j \in [n]$ and the q_Π queries made by f^Π, it follows that

$$\Pr\left[E_1\right] \leq q_\Pi \cdot n \cdot 2^{-s(\lambda)}. \tag{6}$$

Notice that the event E_1 is independent of the challenge bit b, i.e., $\Pr\left[E_1^0\right] = \Pr\left[E_1^1\right]$. In fact the event E_1 depends only on $f^\Pi(\kappa)$ and it is independent of the query made by A to the oracle $\mathcal{O}_{\mathtt{frk}}(b)$.

We now move to bound the probability of the event $(E_2|\neg E_1)$. To bound this event we make use of our assumption that the tampering function is oracle-independent (recall Definition 6). Since Π is an ideal cipher, the tuples (μ_j^*, γ_j^*) for any j are independent from κ. Moreover, conditioned on $\neg E_1$ such tuples are independent of the tampered key $\tilde{\kappa}$,

because $\tilde{\kappa} = f^{\Pi}(\kappa)$ and $f^{\Pi}(\kappa)$ has not queried $\tilde{\kappa}$ on μ_j^* for any j. Additionally, since f is oracle-independent and Π is an ideal cipher, the queries of A to $\mathcal{O}_{\mathtt{frk}}$ and the oracle answers are independent of $\tilde{\kappa}$ and κ. This means that the key κ is uniformly distributed over a subset given the values $(\mu_j^*, \gamma_j^*)_{j\in[n]}$. Also, note that if the i-th query of A to Π features a key κ' different from κ and a message/ciphertext of its choice, then, because Π is an ideal cipher, it only learns whether $\kappa' \in \{\kappa, \tilde{\kappa}\}$ and whether $\kappa' \in (f^{\Pi})^{-1}(\tilde{\kappa})$. To see the latter, namely that the adversary can learn whether κ' is in the pre-image of the tampered key or not, notice that the tampering function could set $\tilde{\kappa}$ according to an arbitrary predicate[4] that depends on the queries it does to the ideal cipher on the original key κ. Thus, by querying on κ', the adversary could verify if the predicate is satisfied or not by κ'. For any key κ' we set $\Lambda(\kappa')$ to be the event that is true if and only if

$$|(f^{\Pi})^{-1}(f^{\Pi}(\kappa'))| < 2^{k/2}.$$

Note that

$$\Pr\left[E_2|\neg E_1\right] \leq \Pr\left[\Lambda(\kappa) \wedge E_2|\neg E_1\right] + \Pr\left[E_2|\neg E_1, \neg\Lambda(\kappa)\right]. \tag{7}$$

We begin by bounding the leftmost term in the right hand side of Eq. (7). More precisely, we show that

$$\Pr\left[\Lambda(\kappa) \wedge E_2|\neg E_1\right] \leq q_{\Pi} \cdot 2^{-k/2}. \tag{8}$$

We can assume that $\Pr\left[\Lambda(\kappa)|\neg E_1\right] > q_{\Pi}\cdot 2^{-k/2}$, as otherwise Eq. (8) holds trivially. For each κ and Π such that $\Lambda(\kappa)$ holds there are at most $2^{k/2}$ values κ' such that $f^{\Pi}(\kappa) = f^{\Pi}(\kappa')$, and furthermore $\Lambda(\kappa') = 1$ for all such κ'. This comes readily by the definition of the event $\Lambda(\kappa)$. We are interested in bounding the probability that A queries Π on κ for the first time in the i-th query conditioned on $\Lambda(\kappa)$ holding. Note that there are exactly $2^k \cdot \Pr\left[\Lambda(\kappa)|\neg E_1\right]$ possible values for the key κ conditioned on $\Lambda(\kappa)$ holding. Furthermore, each previous query to $\kappa' \neq \kappa$ such that $\Lambda(\kappa')$ holds rules out at most $2^{k/2}$ key values by definition of $\Lambda(\kappa')$. More precisely, it rules out κ' along with all values in the preimage $(f^{\Pi})^{-1}(\kappa')$, which are fewer than $2^{k/2}$. Therefore, the probability that A queries Π on κ for the first time in the i-th query conditioned on $\Lambda(\kappa)$ holding is at most

$$\frac{1}{2^k \Pr\left[\Lambda(\kappa)|\neg E_1\right] - (i-1)2^{k/2}} \leq \frac{1}{q_{\Pi}\cdot 2^{k/2} - (i-1)2^{k/2}} \leq 2^{-k/2},$$

where the leftmost inequality uses our assumption that $\Pr\left[\Lambda(\kappa)|\neg E_1\right] \geq q_{\Pi}\cdot 2^{-k/2}$. Taking a union bound over all q_{Π} queries yields Eq. (8).

We now bound the rightmost term in the right hand side of Eq. (7). Conditioned on the event $\neg\Lambda(\kappa)$, we consider the worst-case scenario where the adversary A knows the value $\tilde{\kappa}$ and that all its queries are in $(f^{\Pi})^{-1}(\tilde{\kappa})$. Since $\neg\Lambda(\kappa)$ holds, we know that there are at least $2^{k/2}$ keys κ' such that $f^{\Pi}(\kappa') = f^{\Pi}(\kappa)$. Moreover, each query to such a key $\kappa' \neq \kappa$ only rules out κ' itself, and κ is still uniformly distributed over the remaining set of keys. Therefore, the probability that A queries Π on κ for the first time in the i-th query is at most

$$\frac{1}{2^{k/2} - (i-1)} \leq 2^{-k/4},$$

where the last inequality uses our hypothesis that $q_{\Pi} \leq 2^{k/4}$. Taking a union bound over all q_{Π} queries shows that

$$\Pr\left[E_2|\neg E_1, \neg\Lambda(\kappa)\right] \leq q_{\Pi}\cdot 2^{-k/4}.$$

[4] For example, the tampering function could set the tampered key to 0^k if the first bit of $\pi_{\kappa}(0)$ is equal to 0, or to 1^k otherwise. By querying $\pi_{\kappa'}(0)$, the adversary can assert whether κ' is in the pre-image of $\tilde{\kappa}$.

Combining this inequality with Eqs. (7) and (8) yields

$$\Pr[E_2|\neg E_1] \leq q_\Pi \cdot (2^{-k/4} + 2^{-k/2}), \tag{9}$$

as desired. Notice that the event E_2 is independent of the challenge bit b, namely the probability of E_2 is the same in the distributions defined by $\mathcal{H}(0)$ and $\mathcal{H}(1)$. In fact, the tuples $(\mu_j^*, \gamma_j^*)_{j\in[n]}$ and the key κ are identically distributed in $\mathcal{H}(0)$ and $\mathcal{H}(1)$ given the full views before the first query of A that triggers the event E_2.

Finally, we bound the probability of the event $(E_3|\neg E_2)$. The only way to find collisions conditioned on $\neg E_2$ is for the adversary to query $\mathcal{O}_{\mathtt{frk}}$. Notice that if $b = 0$ then the probability of E_3 is 0 because Π is a keyed permutation and we must have $f^\Pi(\kappa) = \kappa$ for E_3 to hold. We now focus on the case $b = 1$. In this case, for any choice of $(\mu_i^*, \gamma_i^*)_{i\in[n]}$ the probability that the i-th *distinct* query of the form $(\mathtt{enc}, \mu)$ with $\mu \notin \{\mu_1^*, \ldots, \mu_n^*\}$ yields $\pi_\kappa(\mu) \in \{\gamma_1^*, \ldots, \gamma_n^*\}$ is at most

$$\frac{n}{2^m - i} \leq n \cdot 2^{-m+1},$$

because $\pi_\kappa(x)$ is uniformly random and distinct from the answers to all the previous encryption queries. The last inequality uses our hypothesis that $q_\mathcal{O} \leq 2^{m-1}$. An analogous argument shows that the probability that the i-th *distinct* query of the form $(\mathtt{dec}, y)$ with $\gamma \notin \{\gamma_1^*, \ldots, \gamma_n^*\}$ yields $\pi_\kappa^{-1}(\gamma) \in \{\mu_1^*, \ldots, \mu_n^*\}$ is also at most $\frac{n}{2^m - i} \leq n \cdot 2^{-m+1}$. Combining these bounds with a union bound over all $q_\mathcal{O}$ queries to $\mathcal{O}_{\mathtt{frk}}$ yields

$$\Pr\left[E_3|\neg E_2 \wedge \neg E_1\right] \leq q_\mathcal{O} \cdot n \cdot 2^{-m+1}. \tag{10}$$

From Eqs. (6), (9) and (10) combined with Eq. (5) it follows that

$$\left|\Pr\left[\mathsf{A}^{\mathcal{O}_{\mathtt{frk}}(b),\Pi}() = 1\right] - \Pr\left[\mathcal{H}(b) = 1\right]\right| \leq q_\Pi \cdot (2n \cdot 2^{-s} + 2^{-k/4} + 2^{-k/2}) + 2q_\mathcal{O} \cdot n \cdot 2^{-m} \tag{11}$$

for $b \in \{0, 1\}$.

It remains to upper bound the advantage in distinguishing between $\mathcal{H}(0)$ and $\mathcal{H}(1)$. We claim that

$$\left|\Pr\left[\mathcal{H}(0) = 1\right] - \Pr\left[\mathcal{H}(1) = 1\right]\right| \leq q_\mathcal{O} \cdot n \cdot 2^{-m+1}. \tag{12}$$

First, notice that for $b \in \{0, 1\}$ we have $\Pr\left[\mathcal{H}(b) = 1|E_1 \vee E_2 \vee E_3\right] = 0$ since, under this conditioning, the hybrid always outputs $\perp$. Thus, letting $\bar{E} := (\neg E_1 \wedge \neg E_2 \wedge \neg E_3)$, we are left to show that

$$\Pr\left[\mathcal{H}(0) = 1 \wedge \bar{E}\right] - \Pr\left[\mathcal{H}(1) = 1 \wedge \bar{E}\right] \leq q_\mathcal{O} \cdot n \cdot 2^{-m+1}.$$

We first argue that $\Pr\left[\mathcal{H}(0) = 1|\bar{E}\right] = \Pr\left[\mathcal{H}(1) = 1|\bar{E}\right]$. Conditioned on $\neg E_1$ to hold, we have that $\tilde{\kappa}$ when $\tilde{\kappa} \neq \kappa$ is independent of $(\mu_i^*, \gamma_i^*)_{i\in[n]}$ even fixing Π, and thus the queries to $\mathcal{O}_{\mathtt{frk}}$ made by A are independent of $(\mu_i^*, \gamma_i^*)_{i\in[n]}$ (and thus of b). Conditioned on $\neg E_2$ to hold, the joint distribution of $(\mu_i^*, \gamma_i^*)_{i\in[n]}$ and the outputs of the queries to Π is independent of b, because Π is an ideal cipher and none of the queries by the adversary A to Π intersect with the queries made by the oracle $\mathcal{O}_{\mathtt{frk}}$ to Π. Conditioned on $\neg E_3$ to hold, when $f^\Pi(\kappa) = \kappa$ the joint distribution of $(\mu_i^*, \gamma_i^*)_{i\in[n]}$ and the outputs of the queries to $\mathcal{O}_{\mathtt{frk}}$ is independent of the challenge bit b, because both when $b = 0$ and $b = 1$ these values are uniformly random from the set $\{0, 1\}^m$ and distinct.

Therefore, Eq. (12) follows if we show that

$$\left|\Pr_{\mathcal{H}(0)}\left[E_1 \vee E_2 \vee E_3\right] - \Pr_{\mathcal{H}(1)}\left[E_1 \vee E_2 \vee E_3\right]\right| \leq q_\mathcal{O} \cdot n \cdot 2^{-m+1},$$

where we stress that the first probability in the left hand side is over the probability space induced by $\mathcal{H}(0)$, while the second probability is over the probability space induced by $\mathcal{H}(1)$. Notice that, as we have already argued, the events E_1 and E_2 are independent of the challenge bit b (and so they have the same probabilities in both probability spaces), while E_3 is not. As a result, we have

$$\begin{aligned}
&\left|\Pr_{\mathcal{H}(0)}\left[\neg E_1 \wedge \neg E_2 \wedge \neg E_3\right] - \Pr_{\mathcal{H}(1)}\left[\neg E_1 \wedge \neg E_2 \wedge \neg E_3\right]\right| \\
&= \left|\Pr_{\mathcal{H}(0)}\left[E_3 | \neg E_1 \wedge \neg E_2\right] - \Pr_{\mathcal{H}(1)}\left[E_3 | \neg E_1 \wedge \neg E_2\right]\right| \\
&= \Pr_{\mathcal{H}(1)}\left[E_3 | \neg E_1 \wedge \neg E_2\right] \\
&\leq q_{\mathcal{O}} \cdot n \cdot 2^{-m+1},
\end{aligned}$$

as desired. The second equality holds because $\Pr_{\mathcal{H}(0)}\left[E_3 | \neg E_1 \wedge \neg E_2\right] = 0$ as argued above, and the last inequality follows from Eq. (10). Combining Eqs. (4), (11) and (12) with the triangle inequality concludes the proof. □

6 Setting Parameters

We provide two possible parameter instantiations for our coding scheme. The first instantiation, based on arguments from [FKM18], leads to codewords of length close to $m + 3\lambda$ and non-malleability advantage comparable to $\tau_{\mathsf{tamp}} \cdot 2^{-\lambda}$. A more conservative instantiation, based on Theorem 2, leads to codewords of length close to $m + 5\lambda$ for the same non-malleability advantage. In both cases, we achieve codewords significantly shorter than the state of the art.

Fehr, Karpman, and Mennink [FKM18, Remarks after Definitions 3 and 4, and Section 6] argue that a good cipher, such as AES-128 and SHACAL-2, with keylength k should have advantage close to

$$\tau_{\mathsf{tamp}} \cdot 2^{-k/2}$$

against fixed related-key attacks, with τ_{tamp} denoting the runtime of the tampering function. Although, as we have shown, this argument breaks down with respect to their fixed related-key security assumption, we believe that the cipher-dependent attacks which break their assumption are necessarily contrived. Therefore, since our weaker entropic fixed related-key security assumption precludes the relevant attacks, we find it reasonable to assume that good ciphers Π with keylength k satisfy

$$\mathbf{Adv}^{\mathtt{frk}}_{\Pi}(1, \tau_{\mathsf{tamp}}, k, s, 1) \approx \tau_{\mathsf{tamp}} \cdot (2^{-k/2} + 2^{-s}). \tag{13}$$

The additional rightmost term $\tau_{\mathsf{tamp}} \cdot 2^{-s}$ stems from the discussion surrounding Eq. (1) – the probability that the adversary learns encryptions (under the true key κ) of messages which were also queried by the tampering function is at most $\tau_{\mathsf{tamp}} \cdot 2^{-s}$ via a union bound and the fact that padding the original message with an s-bit random string yields an s-entropic distribution. Therefore, taking into account Theorem 1 and Eq. (13), in order to obtain non-malleability advantage

$$\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau_{\mathsf{tamp}}) \approx \tau_{\mathsf{tamp}} \cdot 2^{-\lambda}$$

it is enough to set $k \approx 2\lambda$ and $s \approx \lambda$ in our coding scheme Σ described in Fig. 1. This leads to codewords of overall length $m + k + s \approx m + 3\lambda$.

As an alternative, more conservative instatiation method, we may instead consider the advantage upper bound provided by Theorem 2 in the ideal cipher model. This result states

that an ideal cipher with keylength k has entropic fixed related-key security advantage roughly

$$\tau_{\mathsf{tamp}} \cdot (2^{-k/4} + 2^{-s}) \tag{14}$$

when the message is padded with an s-bit random string. We extrapolate that a good cipher should satisfy this property in practice, and remark that it is our belief that this bound is loose and the true advantage in the ideal cipher model should be $\tau_{\mathsf{tamp}} \cdot (2^{-k/2} + 2^{-s})$. We leave it as an interesting open problem to prove this conjecture. Similarly to the previous paragraph, taking into account Theorem 1 and Eq. (14), in order to obtain non-malleability advantage

$$\mathbf{Adv}^{\mathtt{nm}}_{\Sigma}(\tau_{\mathsf{tamp}}) \approx \tau_{\mathsf{tamp}} \cdot 2^{-\lambda}$$

it is enough to set $k \approx 4\lambda$ and $s \approx \lambda$ in our coding scheme Σ described in Fig. 1. This more conservative instantiation leads to codewords of overall length $m + k + s \approx m + 5\lambda$.

7 Conclusions and Future Directions

We have given a construction of non-malleable codes in the split-state model with codeword length $m+3\lambda$, where m is the message size and λ is the security parameter. Our construction involves a single call to a block cipher, and can be proven secure under a form of related-key security which we named entropic FRK security. Previous work either achieved rather worse codeword length under non-falsifiable assumptions [KLT16], or a similar (in fact, slightly better) codeword length under a much stronger form of related-key security [FKM18] that unfortunately does not hold in the presence of cipher-dependent tampering attacks. In contrast, entropic FRK security holds unconditionally in the ideal cipher model w.r.t. a large class of *oracle-independent* tampering functions (which includes cipher-dependent tampering attacks which break the assumption from [FKM18]).

Natural directions for future work include reducing the codeword length even further, for example through a better analysis of the entropic FRK assumption in the ideal cipher model or through a different set of assumptions (indeed, while the assumptions of [FKM18] do not hold against cipher-dependent tampering functions, the adaptation of Bernstein's attack [Ber10] to the context of non-malleable codes does not seem to trivially violate the non-malleability of Fehr, Karpman, and Mennink's construction) and establishing that entropic FRK security holds in the ideal cipher model even for tampering functions that are not oracle-independent. It would also be interesting to extend our techniques to obtain practical non-malleable secret sharing [GK18] with very short shares based on related-key secure block ciphers.

References

[AAG+16] Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Optimal computational split-state non-malleable codes. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 393–417. Springer, Heidelberg, January 2016.

[ADKO15] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 459–468. ACM Press, June 2015.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In David B. Shmoys, editor, *46th ACM STOC*, pages 774–783. ACM Press, May / June 2014.

[AFPW11] Martin R. Albrecht, Pooya Farshim, Kenneth G. Paterson, and Gaven J. Watson. On cipher-dependent related-key attacks in the ideal-cipher model. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 128–145. Springer, Heidelberg, February 2011.

[AKO+22] Divesh Aggarwal, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, Maciej Obremski, and Sruthi Sekar. Rate one-third non-malleable codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1364–1377, New York, NY, USA, 2022. Association for Computing Machinery.

[AO20] Divesh Aggarwal and Maciej Obremski. A constant rate non-malleable code in the split-state model. In *61st FOCS*, pages 1285–1294. IEEE Computer Society Press, November 2020.

[Ber10] Daniel J. Bernstein. E-mail discussion among the participants of the Early Symmetric Crypto Seminar 2010, 2010.

[CG16] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. *IEEE Transactions on Information Theory*, 62(3):1097–1118, 2016.

[CG17] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. *J. Cryptol.*, 30(1):191–241, 2017.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 285–298. ACM Press, June 2016.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 239–257. Springer, Heidelberg, August 2013.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 434–452. Tsinghua University Press, January 2010.

[DPW18] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. *J. ACM*, 65(4), apr 2018.

[FKM18] Serge Fehr, Pierre Karpman, and Bart Mennink. Short non-malleable codes from related-key secure block ciphers. *IACR Trans. Symm. Cryptol.*, 2018(1):336–352, 2018.

[GK18] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 685–698. ACM Press, June 2018.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *J. ACM*, 56(4), jul 2009.

[Har09] David G. Harris. Generic ciphers are more vulnerable to related-key attacks than previously thought. In *International Workshop on Coding and Cryptography*, 2009.

[KLT16] Aggelos Kiayias, Feng-Hao Liu, and Yiannis Tselekounis. Practical non-malleable codes from l-more extractable hash functions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1317–1328. ACM Press, October 2016.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 1144–1156. ACM Press, June 2017.

[Li19] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:49, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 517–532. Springer, Heidelberg, August 2012.