Tamper Detection and Continuous Non-Malleable Codes

Zahra Jafargholi Daniel Wichs

Northeastern University
{zahra,wichs}@ccs.neu.edu

November 21, 2014

Abstract

We consider a public and keyless code $(\mathsf{Enc}, \mathsf{Dec})$ which is used to encode a message m and derive a codeword $c = \mathsf{Enc}(m)$. The codeword can be adversarially tampered via a function $f \in \mathcal{F}$ from some "tampering function family" $\mathcal{F}$, resulting in a tampered value $c' = f(c)$. We study the different types of security guarantees that can be achieved in this scenario for different families $\mathcal{F}$ of tampering attacks.

Firstly, we initiate the general study of *tamper-detection codes*, which must detect that tampering occurred and output $\mathsf{Dec}(c') = \perp$. We show that such codes exist for any family of functions $\mathcal{F}$ over n bit codewords, as long as $|\mathcal{F}| < 2^{2^n}$ is sufficiently smaller than the set of all possible functions, and the functions $f \in \mathcal{F}$ are further *restricted* in two ways: (1) they can only have a *few fixed points* x such that $f(x) = x$, (2) they must have *high entropy* of $f(x)$ over a random x. Such codes can also be made efficient when $|\mathcal{F}| = 2^{\mathsf{poly}(n)}$. For example, $\mathcal{F}$ can be the family of all low-degree polynomials excluding constant and identity polynomials. Such tamper-detection codes generalize the algebraic manipulation detection (AMD) codes of Cramer et al. (EUROCRYPT '08).

Next, we revisit *non-malleable codes*, which were introduced by Dziembowski, Pietrzak and Wichs (ICS '10) and require that $\mathsf{Dec}(c')$ either decodes to the original message m, or to some unrelated value (possibly $\perp$) that doesn't provide any information about m. We give a modular construction of non-malleable codes by combining tamper-detection codes and leakage-resilient codes. This gives an alternate proof of the existence of non-malleable codes with optimal rate for any family $\mathcal{F}$ of size $|\mathcal{F}| < 2^{2^n}$, as well as efficient constructions for families of size $|\mathcal{F}| = 2^{\mathsf{poly}(n)}$.

Finally, we initiate the general study of *continuous non-malleable codes*, which provide a non-malleability guarantee against an attacker that can tamper a codeword multiple times. We define several variants of the problem depending on: (I) whether tampering is *persistent* and each successive attack modifies the codeword that has been modified by previous attacks, or whether tampering is non-persistent and is always applied to the original codeword, (II) whether we can "*self-destruct*" and stop the experiment if a tampered codeword is ever detected to be invalid or whether the attacker can always tamper more. In the case of persistent tampering and self-destruct (weakest case), we get a broad existence results, essentially matching what's known for standard non-malleable codes. In the case of non-persistent tampering and no self-destruct (strongest case), we must further restrict the tampering functions to have few fixed points and high entropy. The two intermediate cases correspond to requiring only one of the above two restrictions.

These results have applications in cryptography to *related-key attack* (RKA) security and to protecting devices against tampering attacks without requiring state or randomness.

1 Introduction

Motivating Example. Consider a security-sensitive device such as a smart-card implementing a digital signature scheme. The user gives it messages as inputs and receives signatures as outputs. The computation relies on a secret signing key stored on the card. Moreover, the user's name, say "Eve", is stored on the card and the card only signs message that begin with the name "Eve". The security of signature schemes ensures that Eve cannot sign a message with any other name if she is given this card and uses it as a black-box. However, Boneh, DeMillo and Lipton [BDL01] show a surprising result: if the above is implemented using RSA signatures with Chinese remaindering, and Eve is able to simply flip a single bit of the signing key on the smart card and observe the resulting incorrectly generated signature, then she can factor the RSA modulus and completely recover the signing key. Alternatively, no matter which signature scheme is used, Eve may be able to flip a few bits of the name stored on the card (e.g., change the value from "Ev$\underline{\text{e}}$" to "Ev$\underline{\text{a}}$") without changing the signing key and then use the card to sign messages under a different name.

The above are examples of *tampering attacks*. By tampering with the internal state of a device (without necessarily knowing what it is) and then observing the outputs of the tampered device, an attacker may be able to learn additional sensitive information which would not be available otherwise. A natural approach to protecting against such attacks is to *encode* the data on the device in some way. For example, [BDL01] suggest using error-detection codes to thwart an attack that flips a small number of bits. This raises the question of what kind of codes are needed to achieve protection and what classes of tampering attacks can they protect against?

A Coding Problem. We can translate the above scenario into a coding problem. We would like to design a code $(\mathsf{Enc}, \mathsf{Dec})$ consisting of a possibly *randomized encoding* function and a *decoding* function with the correctness guarantee that $\mathsf{Dec}(\mathsf{Enc}(m)) = m$. There are no secret keys and anybody can encode and decode. We model tampering as a family of functions $\mathcal{F}$ that an attacker can apply to modify codewords.[1] We consider a "tampering experiment" with some message m and function $f \in \mathcal{F}$. The experiment begins by probabilistically encoding $c \leftarrow \mathsf{Enc}(m)$, then tampers $c' = f(c)$ and finally outputs the decoded value $m' = \mathsf{Dec}(c')$. We consider different types of security guarantees on the outcome m' of the experiment.

1.1 Tamper Detection Codes

Perhaps the simplest property that we could ask for is that tampering can always be detected with overwhelming probability, meaning that the decoded value is some special symbol $m' = \perp$ indicating an error. In other words, a tamper-detection code for a family $\mathcal{F}$ ensures that for any message m and any tampering function $f \in \mathcal{F}$ we have $\Pr[\mathsf{Dec}(f(c)) \neq \perp \ : \ c \leftarrow \mathsf{Enc}(m)]$ is negligible. We ask for which function families $\mathcal{F}$ do such codes exist.

Surprisingly, this natural problem has not been studied at this level of generality and relatively little is known beyond a few specific function families. Standard error-detection codes provide this guarantee for the family of all functions f such that the hamming distance between c and $f(c)$ is always small, but non-zero. The *algebraic manipulation detection* (AMD) codes of Cramer et al. [CDF$^+$08], consider this type of guarantee for functions f that can flip an arbitrary number of bits of c, but the error pattern is independent of c. In other words, AMD codes consider the family $\mathcal{F}_{AMD} = \{f_\Delta(c) := c \oplus \Delta \mid \Delta \neq 0^n\}$.

In this work, we show that tamper-detection codes exist for any function family $\mathcal{F}$ over n-bit codewords as long as the size of the family is bounded $|\mathcal{F}| < 2^{2^{\alpha n}}$ for some constant $\alpha < 1$ and the functions $f \in \mathcal{F}$ satisfy two additional restrictions:

- *High Entropy:* For each $f \in \mathcal{F}$, we require that $f(c)$ has sufficiently high min-entropy when $c \sim \{0,1\}^n$ is uniformly random.
- *Few Fixed Points:* For each $f \in \mathcal{F}$, we require that there aren't too many fixed points c s.t. $f(c) = c$.

Moreover, we show that such codes can achieve a rate (defined as the ratio of message size to codeword size) of $(1-\alpha)$. We also show that the restrictions on the tampering functions (high entropy, few fixed points) in the above result cannot be removed.

This existence result relies on a probabilistic method argument and, in general, such codes may not be efficient. However, when the size of the function family is $|\mathcal{F}| \leq 2^{s(n)}$ for some polynomial s, then we can use a probabilistic method argument with limited independence to get efficient codes. More precisely, this yields a family of efficient

[1] This is a departure from standard coding theory problems by focusing on the process (family of functions) $\mathcal{F}$ that modifies a codeword rather than on some notion of distance between the original and modified codeword.

codes $(\mathsf{Enc}_h, \mathsf{Dec}_h)$ indexed by some hash function h from a t-wise independent function family (for a polynomial t depending on s), such that a random member of the family is a secure tamper-detection code for $\mathcal{F}$ with overwhelming probability. We can also think of this as a construction of efficient tamper-detection codes in the *common random string* (CRS) model, where the random choice of h is specified in the CRS and known to everyone.

This result generalizes AMD codes which which provide tamper-detection for the family $\mathcal{F}_{AMD}$ of size $|\mathcal{F}_{AMD}| = 2^n - 1$ that has full entropy n, and no fixed points. For example, using this result, we can get efficient tamper-detection codes (in the CRS model) for the class $\mathcal{F}^{-}_{\mathsf{poly},d}$ of polynomials f over $\mathbb{F}_{2^n}$ of some bounded degree $d = \mathsf{poly}(n)$, as long as we exclude the identity polynomial $f(x) = x$ and the constant (degree 0) polynomials. Alternatively, we get also get such tamper-detection codes for the class $\mathcal{F}^{-}_{\mathsf{affine},r}$ of all affine functions when we interpret $x \in \{0,1\}^n$ as an (n/r)-dimensional vector over the field $\mathbb{F}_{2^r}$ with a sufficiently large r, and we exclude the identity and the constant functions.

1.2 Non-Malleable Codes

The work of Dziembowski, Pietrzak and Wichs [DPW10] introduced the notion of *non-malleable codes*, which asks for a weaker guarantee on the outcome of the tampering experiment. Instead of insisting that the decoded value after tampering is always $m' = \perp$, non-malleable security requires that either $m' = m$ is equal to the original message, or m' is a completely unrelated value (possibly $\perp$) that contains no information about the original message m. Moreover, the choice of which of these two options occurs is also unrelated to m. For example, non-malleability ensures that it should not be possible to tamper c to $c' = f(c)$ in such a way as to just flip a bit of the encoded message. Alternatively, it shouldn't be possible to tamper the codeword in such a way as to get either $m' = m$ or $m' = \perp$ depending on (say) the first bit of m.

Non-malleable codes offer strong protection against tampering. By encoding the data on an device with a non-malleable code for $\mathcal{F}$, we ensure that an attacker cannot learn anything more by tampering with the data on the device via functions $f \in \mathcal{F}$ and interacting with the tampered device, beyond what could be learned given black-box access to the device without the ability to tamper. Moreover, such codes can also protect against *continuous* tampering attacks, where the attacker repeatedly tampers with the data on the device and observes its outputs. On each invocation, the device must first decode the secret data, run the underlying functionality with the decoded value to produce some output, and then freshly *re-encode* the decoded value. This last step is important for security and ensures that each tampering attack acts on a fresh codeword. On the downside, this step usually requires fresh randomness and also requires that device is now *stateful* even if the underlying functionality is stateless.

Prior Work on Non-Malleable Codes. The work of [DPW10], gives a broad *existence* result showing that for any family $\mathcal{F}$ of tampering functions over n-bit codewords having size $|\mathcal{F}| < 2^{2^{\alpha n}}$ for $\alpha < 1$ there exists a (possibly inefficient) non-malleable code for $\mathcal{F}$. This covers various complex types of tampering attacks. The work of Cheraghchi and Guruswami [CG13a] further show that such codes can achieve a rate of $1 - \alpha$, which is optimal. These results rely on a probabilistic method argument, where the code is defined via a completely random function.

The works of [CG13a, FMVW14] also give *efficient* scaled-down versions of the above existence results for function families of singly-exponential size $|\mathcal{F}| \le 2^{s(n)}$ for some polynomial s. For example, $\mathcal{F}$ could be the class of all circuits of size at most $s(n)$. These results are derived using a probabilistic method argument with limited independence, where the code $(\mathsf{Enc}_h, \mathsf{Dec}_h)$ is parameterized by some efficient hash function h chosen from a t-wise independent family for some polynomial t depending on s. The code is secure with overwhelming probability over the choice of h, which we can think of as a common random string (CRS).

Several other works [DPW10, CKM11, LL12, DKO13, ADL13, CG13b, FMNV14, CZ14] construct explicit non-malleable codes for interesting families that are restricted through their *granularity*. In particular, these works envision that the codeword consists of several components, each of which can be tampered arbitrarily but independently of the other components. The strongest variant of this is the *split-state model*, where the codeword consists of just two components that are tampered independently of each other. Such codes were recently constructed in the information theoretic setting in the works of [DKO13, ADL13]. Another recent result [AGM$^+$14] shows how to construct non-malleable codes against functions that can permute (and perturbe) the bits of the codeword.

Our Results. We show a general way of obtaining non-malleable codes for general function families by combining tamper-detection codes for restricted function families (with few fixed points and high entropy) and a certain type of *leakage-resilient codes*, defined by Davì, Dziembowski and Venturi [DDV10]. The resulting non-malleable code construction matches that of Faust et al. [FMVW14] but with an optimized modular analysis. We show that this construction can simultaneously achieve optimal rate $(1-\alpha)$ for function families of size $|\mathcal{F}| = 2^{2^{\alpha n}}$ and also efficient encoding/decoding that scales polynomially in the security parameter and n when $|\mathcal{F}| = 2^{\mathsf{poly}(n)}$. Previously, each of these properties was known to be achievable individually but by different constrictions shown in [CG13a] and [FMVW14] respectively.

1.3 Continuous Non-Malleable Codes

As mentioned, standard non-malleable codes already provide protection against *continuous* tampering attacks on a device, if the device freshly re-encodes its state after each invocation to ensure that tampering is applied to a fresh codeword each time. However, this is undesirable since it requires that: (1) the device has access to fresh randomness on each invocation, and (2) the device is stateful and updates its state on each invocation. This is the case even if the underlying functionality that the device implements (e.g., a signature scheme) is deterministic and stateless. This brings up the natural question (posed as an open problem by [DPW10]) whether we can achieve security against continuous tampering of a single codeword *without* re-encoding. We consider four variants of such *continuous non-malleable codes* depending on:

- Whether tampering is *persistent*, meaning that each tampering attack is applied to the current version of the codeword that has been modified by previous attacks, and the original codeword is otherwise lost once tampered. Alternatively, we can consider non-persistent tampering, where tampering is always applied to the original codeword.

- Whether tampering to an invalid codeword (one that decodes to $\perp$) causes a "*self-destruct*" meaning that the experiment stops and the attacker cannot gain any additional information, or whether the attacker can always continue tampering more. We can think of this as corresponding to a physical device "self-destructing" (e.g., by erasing all internal data) if it detects that it has been tampered. This in turn corresponds to a very limited form of state, where the data on the device can only be erased but not updated otherwise.

Note that persistent tampering and self-destruct is the weakest variant, non-persistent tampering and no self destruct is the strongest variant, and the remaining two variants lie in between and are incomparable to each other. All of these variants are already stronger than standard non-malleable codes.

The notion of *continuous non-malleable codes* was first introduced in the work of Faust et al. [FMNV14], which considered the case of non-persistent tampering and self-destruct. It focused on tampering in the split-state model, where two halves of a codeword are tampered independently of each other, and showed that although such codes cannot exist information theoretically, they can be constructed under computational assumptions. The reason for focusing on non-persistent tampering is that this models attacks that have access to some "auxiliary memory" on the device beyond the n bits of "active memory" used to store the codeword. The initial tampering attack can make a copy the original codeword onto this auxiliary memory. Each subsequent attack can then tamper the original codeword from the auxiliary memory and place the tampered codeword into the active memory. In the case of persistent tampering, we implicitly assume that there is no such auxiliary memory on the device.

In this work, we give a general definition of continuous non-malleable codes that captures all of the above variants. We initiate the comprehensive study of what type of tampering attacks we can protect against under each variant in the information theoretic setting.

Our Results. We use the same template that we employed for constructing standard non-malleable codes to also construct continuous non-malleable codes. Depending on which of the four variants of continuous non-malleable codes we consider, we show that our construction achieves security for different classes of tampering attacks.

In the setting of *persistent tampering and self-destruct* (weakest), we show broad existence results which essentially match the existence results of [DPW10, CG13a] for standard non-malleable codes. In particular, we show that such codes *exist* (inefficiently) for any family $\mathcal{F}$ of tampering functions over n-bits, whose size is $|\mathcal{F}| = 2^{2^{\alpha n}}$ for $\alpha < 1$. Moreover, such codes achieve a rate of $1 - \alpha$. For example, this result shows existence of such codes (albeit inefficient) in the split-state model with rate 1/2. Furthermore, we give an *efficient* scaled-down versions of the above existence results for function families of singly-exponential size $|\mathcal{F}| = 2^{\mathsf{poly}(n)}$. Unfortunately, in this case the efficiency of the code also depends polynomially on the number of tampering attacks we want to protect against. We conjecture that this dependence can be removed, and leave this fascinating problem for future work.

In the setting of *non-persistent tampering and no self destruct* (strongest), we must place additional restrictions on the function family $\mathcal{F}$ to ensure that the functions have *high entropy* and *few fixed points*, as in tamper-detection codes. In fact, a tamper-detection code *is* continuous non-malleable in this setting since each tampering attack simply leaves the codeword invalid (decodes to $\perp$) and so the attacker does not learn anything. However, in contrast to tamper-detection codes, continuous non-malleable codes in this setting can also trivially tolerate the "identity function" $f_{\mathsf{id}}(x) = x$ and the "always constant" functions $f_c(x) = c$. Therefore, we (e.g.,) get continuous non-malleable codes in this setting for the class $\mathcal{F}_{\mathsf{poly},d}$ of *all* low-degree polynomials or the class $\mathcal{F}_{\mathsf{affine},r}$ of *all* affine functions over a sufficiently large field.

In the two intermediate settings, we only need to impose one of the two restrictions on high entropy and few fixed points. For the case of *persistent tampering and no self destruct* we only require that the functions have *few fixed points* but can have arbitrary entropy. For the case of *non-persistent tampering and self destruct* we only require that the functions have *high entropy* but can have many fixed points.

1.4 Other Related Work and RKA Security

There is a vast body of literature that considers tampering attacks using other approaches besides (non-malleable) codes. See, e.g., [BK03, GLM$^+$04, IPSW06, BC10, BCM11, FPV11, KKS11, AHI11, GOR11, Pie12, Wee12, BPT12, DFMV13, ABPP14, GIP$^+$14] etc.

One highly relevant line of work called *related key attacks* (RKA) security [BK03, BC10, BCM11, Wee12, BPT12, ABPP14] considers tampering with the secret key of a cryptographic primitive such as a pseudorandom function or a signature scheme. The definitions in those works usually require the schemes to be stateless, consider non-persistent tampering (tampering is always applied to the original key), and do not allow for self-destruct. These works focus on giving clever constructions of specific primitives that satisfy RKA security based on specific computational assumptions. We note that our results on continuous non-malleable codes in the setting of non-persistant tampering and no self-destruct provide a generic way of achieving qualitatively similar results to these works for any cryptographic scheme. In particular, by encoding the key of the scheme with such codes, we can achieve protection against the types of attacks that were considered in the RKA literature (e.g., additive tampering, low-degree polynomials) as well as many other interesting families.

Nevertheless, we note that the RKA constructions from the literature maintain some advantages, such as having uniformly random secret keys, whereas we would require the secret key to be a structured codeword. Also, the exact definitions of RKA security in the literature vary from one primitive to another and sometimes impose additional properties that our constructions would not satisfy. For example, definitions of RKA security for PRFs usually require that outputs under both the original and the tampered keys remain pseudorandom, whereas our solution would only guarantee that outputs under the original key remain pseudorandom even given outputs under a tampered key, but the latter may not be pseudorandom (they may just be $\perp$). It is not clear whether these differences are important in the envisioned applications, and therefore we view our results as providing qualitatively similar (but not equivalent) security guarantees to those studied in the context of RKA security.

2 Preliminaries

Notation. For a positive integer n, we define the set $[n] := \{1, \ldots, n\}$. Let X, Y be random variables with supports $S(X), S(Y)$, respectively. We define their *statistical distance* by $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_{s \in S(X) \cup S(Y)} |\Pr[X = s] - \Pr[Y = s]|$. We write $X \approx_\varepsilon Y$ and say that X and Y are ε-statistically close to denote that $\mathbf{SD}(X,Y) \leq \varepsilon$. We let U_n denote the uniform distribution over $\{0,1\}^n$. We use the notation $x \leftarrow X$ to denote the process of sampling a value x according to the distribution X. For a set S, we write $s \leftarrow S$ to denote the process of sampling s uniformly at random from S.

Tail Bound. We recall the following lemma from [BR94] which gives us a Chernoff-type tail bound for limited independence.

Lemma 2.1 (Lemma 2.3 of [BR94]). *Let $t \geq 4$ be an even integer. Suppose $X_1, \ldots, X_n$ are t-wise independent random variables over $\{0,1\}$. Let $X := \sum_{i=1}^n X_i$ and define $\mu := \mathbf{E}[X]$ be the expectation of the sum. Then, for any $A > 0$, $\Pr[|X - \mu| \geq A] \leq 8\left(\frac{t\mu + t^2}{A^2}\right)^{t/2}$. In particular, if $A \geq \mu$ then $\Pr[|X - \mu| \geq A] \leq 8\left(\frac{2t}{A}\right)^{t/2}$.*

It is easy to check by observing the proof of the lemma that it also holds even when $X_1, \ldots, X_n$ are not truly t-wise independent but for any $S \subseteq [n]$ of size $|S| \leq t$ they satisfy $\Pr[\bigwedge_{i \in S}\{X_i = 1\}] \leq \prod_{i \in S}\Pr[X_i = 1]$. This is because the only use of independence in the proof is to bound $\mathbb{E}[\prod_{i \in S} X_i] \leq \prod_{i \in S}\mathbb{E}[X_i]$ which holds under the above condition.

Coding Schemes. It will be useful to define the following general notion of a coding scheme.

Definition 2.2. *A* (k, n)-coding scheme *consists of two functions: a* randomized *encoding function* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$*, and a* deterministic *decoding function* $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ *such that, for each* $m \in \{0,1\}^k$, $\Pr[\mathsf{Dec}(\mathsf{Enc}(m)) = m] = 1$*. For convenience, we also define* $\mathsf{Dec}(\perp) = \perp$.

2.1 Leakage-Resilient Codes

The following definition of leakage-resilient codes is due to [DDV10]. Intuitively, it allows us to encode a message m into a codeword c in such a way that learning $f(c)$ from some class of *leakage functions* $f \in \mathcal{F}$ will not reveal anything about the message m. For convenience, we actually insist on a stronger guarantee that $f(c)$ is indistinguishable from f applied to the uniform distribution over n bit strings.

Definition 2.3. *Let* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *be a* (k, n)*-coding scheme. For a function family* $\mathcal{F}$*, we say that* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *is* $(\mathcal{F}, \varepsilon)$-leakage-resilient*, if for any* $f \in \mathcal{F}$ *and any* $m \in \{0,1\}^k$ *we have* $f(\mathtt{lrEnc}(m)) \approx_\varepsilon f(U_n)$ *where* U_n *is the uniform distribution over* n *bit strings.*

Construction. We recall the following construction from [DDV10, FMVW14]. Let $\mathcal{H}$ be a t-wise independent hash function family consisting of functions $h : \{0,1\}^v \to \{0,1\}^k$. For any $h \in \mathcal{H}$ we define the $(k, n = k+v)$-coding scheme $(\mathtt{lrEnc}_h, \mathtt{lrDec}_h)$ where: (1) $\mathtt{lrEnc}_h(m) := (r, h(r) \oplus m)$ for $r \leftarrow \{0,1\}^v$; (2) $\mathtt{lrDec}_h((r, z)) := z \oplus h(r)$.

We give an improved analysis of this construction: to handle a leakage family $\mathcal{F}$ with ℓ-bits of leakage (i.e., the output size of $f \in \mathcal{F}$ is ℓ-bits), the best prior analysis in [FMVW14] required overhead (roughly) $v = \log\log|\mathcal{F}| + \ell$, whereas our improved analysis only requires overhead (roughly) $v = \max\{\log\log|\mathcal{F}|, \ell\}$. This can yield up to a factor of 2 improvement and will be crucial in getting optimal rate for our non-malleable and continuous non-malleable codes.

Theorem 2.4. *Fix any function family* $\mathcal{F}$ *consisting of functions* $f : \{0,1\}^n \to \{0,1\}^\ell$*. With probability* $1 - \rho$ *over the choice of a random* $h \leftarrow \mathcal{H}$ *from a* t*-wise independent family* $\mathcal{H}$*, the coding scheme* $(\mathtt{lrEnc}_h, \mathtt{lrDec}_h)$ *above is* $(\mathcal{F}, \varepsilon)$*-leakage-resilient as long as* $v \geq v_{\mathsf{min}}$ *and* $t \geq t_{\mathsf{min}}$*, when* $B := \log|\mathcal{F}| + k + \log(1/\rho)$ *and*

$$\begin{aligned} &\textit{either} \quad v_{\mathsf{min}} = \log(B + 2^\ell) + \log\ell + 2\log(1/\varepsilon) + O(1) && \textit{and} \quad t_{\mathsf{min}} = O(B), \\ &\textit{or} \quad v_{\mathsf{min}} = \log(B + 2^\ell) + 2\log(1/\varepsilon) + O(1) && \textit{and} \quad t_{\mathsf{min}} = O(B + 2^\ell). \end{aligned}$$

In particular, if $\rho = \varepsilon = 2^{-\lambda}$ *for security parameter* λ *and,* $|\mathcal{F}| \geq \max\{2^k, 2^\lambda\}$*,* $\ell \leq 2^\lambda$ *we get:*

$$v_{\mathsf{min}} = \max\{\log\log|\mathcal{F}|, \ell\} + O(\lambda) \qquad \textit{and} \qquad t_{\mathsf{min}} = O(\log|\mathcal{F}|).$$

Proof. Define BAD as the event that $(\mathtt{lrEnc}_h, \mathtt{lrDec}_h)$ is not a good $(\mathcal{F}, \varepsilon)$-leakage resilient code and let $q \in [2^{\ell-1}]$ be a parameter to be set later.

$$\Pr[\mathsf{Bad}] = \Pr_h\left[\exists f \in \mathcal{F},\ m \in \{0,1\}^k\ :\ \mathbf{SD}\left(f(U_v, h(U_v) \oplus m), f(U_n)\right) \geq \varepsilon\right]$$

$$\leq \sum_{f \in \mathcal{F}} \sum_{m \in \{0,1\}^k} \Pr_h\left[\exists\ S\ \subset\ \{0,1\}^\ell,\ |S| =\ q\ :\ \Pr_{r \leftarrow U_v}[f(r, h(r) \oplus m) \in S] - \Pr[f(U_n) \in S] \geq \varepsilon/4 \cdot \tilde{p}_s\right] \tag{2.1}$$

Line 2.1 follows from the previous line and Lemma 2.5 below. Fix some $m \in \{0,1\}^k$, $f \in \mathcal{F}$ and $S \subseteq \{0,1\}^\ell$, $|S| = q$. Define $p_s := \Pr[f(U_n) \in S]$, and for each $r \in \{0,1\}^v$ let $p_{r,s} := \Pr[f(r, U_k) \in S]$. Define random variables $Y_{r,s} = 1$ if $f(r, h(r) \oplus m) \in S$ where the randomness is over the choice of h. Clearly $\sum_{r \in \{0,1\}^v} p_{r,s} = p_s$ and for each r, $\mathbf{E}[Y_{r,s}] = p_{r,s}$. Therefore $\mathbf{E}[\sum_{r \in \{0,1\}^v} Y_{r,s}] = 2^v \cdot p_s$. Note that ${Y_{r,s}}_{r \in \{0,1\}^v}$ are t-wise independent random variables. Furthermore let $\tilde{p}_s := \max\left(q \cdot 2^{-\ell}, p_s\right)$. This means that:

$$
\begin{aligned}
\Pr_h\left[\Pr_{r \leftarrow U_v}\left[f(r, h(r) \oplus m) \in S\right] - \Pr\left[f(U_n) \in S\right] \geq \varepsilon/4 \cdot \tilde{p}_s\right] &\leq \Pr_h\left[\sum_{r \in \{0,1\}^v} Y_{r,s} - 2^v \cdot p_s \geq 2^v \cdot \varepsilon/4 \cdot \tilde{p}_s\right] \\
&\leq \left(\frac{t \cdot 2^v \cdot p_s + t^2}{(\varepsilon/4 \cdot 2^v \cdot \tilde{p}_s)^2}\right)^{t/2} \leq \left(\frac{2t \cdot 2^v \cdot p_s}{(\varepsilon/4 \cdot 2^v \cdot \tilde{p}_s)^2}\right)^{t/2} \\
&\leq \left(\frac{2t}{(\varepsilon/4)^2 \cdot 2^v \cdot \tilde{p}_s}\right)^{t/2} \leq \left(\frac{2t}{\varepsilon^2 \cdot 2^{v-\ell-4} \cdot q}\right)^{t/2} \qquad (2.2) \\
\Rightarrow \Pr\left[\mathsf{Bad}\right] &\leq |\mathcal{F}| \cdot 2^k \cdot \binom{2^\ell}{q} \cdot \left(\frac{2t}{\varepsilon^2 \cdot 2^{v-\ell-4} \cdot q}\right)^{t/2} \leq \rho \qquad (2.3)
\end{aligned}
$$

The inequality in Line 2.2 is obtained by replacing $\tilde{p}_s$ with $2^{-\ell} \cdot q$. We satisfy Line 2.3 by setting the parameters appropriately, such that,

$$
\begin{aligned}
\left(\frac{2t}{\varepsilon^2 \cdot 2^{v-\ell-4} \cdot q}\right) < 1/2 \quad &\Rightarrow \quad v \geq \log t + \ell + 2\log(1/\varepsilon) - \log q + 6 \\
2^{t/2} \cdot \rho > |\mathcal{F}| \cdot 2^k \cdot \binom{2^\ell}{q} \quad &\Rightarrow \quad t/2 \geq \log|\mathcal{F}| + k + \log(1/\rho) + \log\binom{2^\ell}{q}
\end{aligned}
$$

We analyze the parameters for two values for q:

1. Set $q = 2^{\ell-1}$, we get $t_{\mathsf{min}} = O(B + 2^\ell)$ and $v_{\mathsf{min}} = \log t_{\mathsf{min}} + 2\log(1/\varepsilon) + 6$.
2. Set $q = \lceil B/\ell \rceil$ if $\lceil B/\ell \rceil \leq 2^{\ell-1}$, then we get $t_{\mathsf{min}} = O(B)$ and $v_{\mathsf{min}} = \log t_{\mathsf{min}} + \ell - \log B + \log \ell + 2\log(1/\varepsilon) + 5 = \ell + \log \ell + 2\log(1/\varepsilon) + O(1)$.

□

Lemma 2.5. *Let Y and Z be two random variables over $\{0,1\}^\ell$. If $\mathbf{SD}(Y, Z) = \varepsilon$ then*

$$\forall\ q, \quad 1 \leq q \leq 2^{\ell-1}, \quad \exists\ S \in \{0,1\}^\ell,\ |S| = q: \ \ \left|\Pr\left[Z \in S\right] - \Pr\left[Y \in S\right]\right| \geq \varepsilon/4 \cdot \max\left(\Pr\left[Z \in S\right], q \cdot 2^{-\ell}\right).$$

Proof. For all $x \in \{0,1\}^\ell$ define $z_x := \Pr[Z = x]$, $y_x := \Pr[Y = x]$, and for all $S \subset \{0,1\}^\ell$ define $p_s := \Pr[Z \in S]$. Furthermore for any subset S of size q, let $\tilde{p}_s := \max\left(q \cdot 2^{-\ell}, p_s\right)$. Assume for some q, all subsets $S \in \{0,1\}^\ell$, such that $|S| = q$, have the following property,

$$\left|\Pr\left[Z \in S\right] - \Pr\left[Y \in S\right]\right| < \varepsilon/4 \cdot \tilde{p}_s. \qquad (2.4)$$

Define set $S_z := \left\{x \in \{0,1\}^\ell\ : z_x \geq y_x\right\}$ and let $S_y := \{0,1\}^\ell \setminus S_z$. Let $S^* := S_z$ if $|S|_z \geq |S|_y$ and $S^* = S_y$ otherwise. Clearly $|S|^* \geq 2^{\ell-1}$.

$$\mathbf{SD}(Z, Y) = \varepsilon \Rightarrow \quad \left|\Pr\left[Z \in S^*\right] - \Pr\left[Y \in S^*\right]\right| = \varepsilon.$$

Order elements in S^* increasingly according to $|z_x - y_x|$. For example x_1 minimizes $|z_x - y_x|$. Let $r = |S^*| \mod q$ and $d = \lfloor |S^*|/q \rfloor$. Define M_0 to be the set of the first r values in S^*. Evidently, $\left|\Pr\left[Z \in M_0\right] - \Pr\left[Y \in M_0\right]\right| \leq \varepsilon(r/|S^*|) < \varepsilon/2$. For all $i \in \{1, 2, \ldots, d\}$ define M_i to be the first q elements in $S^* \setminus \left(\cup_{j=0}^{i-1} M_j\right)$.

$$
\begin{aligned}
\varepsilon = \left|\Pr\left[Z \in S^*\right] - \Pr\left[Y \in S^*\right]\right| &= \sum_{i=0}^{d} \left|\Pr\left[Z \in M_i\right] - \Pr\left[Y \in M_i\right]\right| \\
&\Rightarrow \sum_{i=1}^{d} \left|\Pr\left[Z \in M_i\right] - \Pr\left[Y \in M_i\right]\right| \ > \ \varepsilon/2 \qquad (2.5)
\end{aligned}
$$

By Line 2.4 we have,

$$
\begin{aligned}
\sum_{i=1}^{d} \left|\Pr\left[Z \in M_i\right] - \Pr\left[Y \in M_i\right]\right| \ &< \ \sum_{i=1}^{d} \varepsilon/4 \cdot \tilde{p}_{M_i} \\
&< \ \varepsilon/4 \cdot \sum_{i=1}^{d} (p_{M_i} + q/2^\ell) \ < \ \varepsilon/4\left((p_{S^*} - p_{m_0}) + \frac{|S^*| - r}{2^\ell}\right) \ < \ \varepsilon/2.
\end{aligned}
$$

This contradicts Line 2.5. □

3 Tamper Detection Codes

We begin by defining the notion of a *tamper-detection code*, which ensures that if a codeword is tampered via some function $f \in \mathcal{F}$ then this is detected and the modified codeword decodes to $\perp$ with overwhelming probability. We give two flavors of this definition: a default ("standard") version which guarantees security for a worst-case message m and a weak version which only guarantees security for a random message m.

Definition 3.1. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be a* (k,n)*-coding scheme and let* $\mathcal{F}$ *be a family of functions of the form* $f : \{0,1\}^n \to \{0,1\}^n \cup \perp$. *We say that* $(\mathsf{Enc}, \mathsf{Dec})$ *is a:*

- $(\mathcal{F}, \varepsilon)$*-secure tamper detection code if for any function* $f \in \mathcal{F}$ *and* any *message* $m \in \{0,1\}^k$, *we have* $\Pr[\mathsf{Dec}(f(\mathsf{Enc}(m))) \neq \perp] \leq \varepsilon$, *where the probability is over the randomness of the encoding procedure.*
- $(\mathcal{F}, \varepsilon)$*-weak tamper detection code if for any function* $f \in \mathcal{F}$ *we have* $\Pr_{m \leftarrow U_k}[\mathsf{Dec}(f(\mathsf{Enc}(m))) \neq \perp] \leq \varepsilon$, *where the probability is over a random message* m *and the randomness of the encoding procedure.*

It is easy to see that there are some small function families for which tamper detection (or even weak tamper detection) is impossible to achieve. One example is the family consisting of a single identity function $f_{\mathsf{id}}(x) = x$. Another example is the family consisting of all constant function $\mathcal{F}_{const} = \{f_c(x) = c \ : \ c \in \{0,1\}^n\}$. This family is of size only $|\mathcal{F}_{const}| = 2^n$ but no matter what code is used there is some function $f_c \in \mathcal{F}_{const}$ corresponding to a valid codeword c which breaks the tamper detection guarantee. Of course, there are other "bad" functions such as ones which are close to identity or close to some constant function. But we will show that these are the *only* bad cases. We begin by defining two restrictions on functions which ensure that they are far from the above bad cases.

Definition 3.2. *A function* $f : \{0,1\}^n \to \{0,1\}^n \cup \perp$ *is a* φ*-few fix points,* μ*-entropy function if*

- $\Pr_{x \leftarrow U_n}[f(x) = x] \leq \varphi$. *(*$\varphi$*-few fixed points)*
- $\forall y \in \{0,1\}^n, \ \Pr_{x \leftarrow U_n}[f(x) = y] \leq 2^{-\mu}$. *(*$\mu$*-entropy)*

The first property restricts the number of fixed points $x \ : f(x) = x$ to be less than $\varphi \cdot 2^n$. This ensures that the function is sufficiently far from being an identity function. The second property is equivalent to saying that the min-entropy $\mathbf{H}_\infty(f(U_n)) \geq \mu$. This ensures that the function is far from being a constant function.

3.1 Weak Tamper-Detection Codes

We begin by constructing weak tamper-detection codes. We will then show how to use weak tamper-detection codes to also construct standard tamper-detection codes.

Construction. Our construction of weak tamper detection codes will have a deterministic encoding. Let $\{0,1\}^k$ be the message space and $\{0,1\}^n$ to be the codeword space where $n = k + w$ for some $w > 0$. For a function $h : \{0,1\}^k \to \{0,1\}^w$ define $(\mathsf{Enc}_h, \mathsf{Dec}_h)$ as follow, $\mathsf{Enc}_h(m) := (m, h(m))$ and $\mathsf{Dec}_h(c)$ checks whether $c = (m, z)$ where $z = h(m)$: if so, it outputs m and otherwise it outputs $\perp$.

Theorem 3.3. *Let* $\mathcal{F}$ *be any finite family of* φ*-few fix points,* μ*-entropy functions and* $\mathcal{H} = \{h \mid h : \{0,1\}^k \to \{0,1\}^w\}$ *be a family of* $2t$*-wise independent hash functions, then*

$$\Pr_{h \leftarrow \mathcal{H}}[(\mathsf{Dec}_h, \mathsf{Enc}_h) \textit{ is a } (\mathcal{F}, \varepsilon)\textit{-weak tamper detection code}] > 1 - \rho$$

when the parameters satisfy

$$\begin{aligned} t &\geq t_{\mathsf{min}} \quad \textit{where} \quad t_{\mathsf{min}} = \log|\mathcal{F}| + k + \log(1/\rho) + 5 \\ \mu &\geq \log(t_{\mathsf{min}}) + w + \log(1/\varepsilon) + 6 \\ w &\geq \log(1/\varepsilon) + 3 \\ \varphi &\leq \varepsilon/4 \end{aligned}$$

For example, if $\rho = \varepsilon = 2^{-\lambda}$ *for security parameter* λ *and,* $|\mathcal{F}| \geq \max\{2^k, 2^\lambda\}$ *we get:*

$$t = O(\log|\mathcal{F}|) \quad , \quad \mu = \log\log|\mathcal{F}| + 2\lambda + O(1) \quad , \quad w = \lambda + O(1) \quad , \quad \varphi = 2^{-(\lambda + O(1))}$$

Proof. Define the event BAD to occur if $(\mathsf{Dec}_h, \mathsf{Enc}_h)$ is not a $(\mathcal{F}, \varepsilon)$-weak tamper detection code. Then, by the union bound:

$$\begin{aligned}\Pr_{h \leftarrow \mathcal{H}}[\mathsf{BAD}] &= \Pr_{h \leftarrow \mathcal{H}}\left[\exists f \in \mathcal{F}, \Pr_{m \leftarrow U_k}[\mathsf{Dec}_h(f(\mathsf{Enc}_h(m))) \neq \perp] > \varepsilon\right] \\ &\leq \sum_{f \in \mathcal{F}} \underbrace{\Pr_{h \leftarrow \mathcal{H}}\left[\Pr_{m \leftarrow U_k}[\mathsf{Dec}_h(f(\mathsf{Enc}_h(m))) \neq \perp] > \varepsilon\right]}_{P_f} \qquad (3.1)\end{aligned}$$

Let's fix some particular function $f \in \mathcal{F}$ and find a bound on the value of P_f. Define the indicator random variables $\{X_m\}_{m \in \{0,1\}^k}$ as $X_m = 1$ if $\mathsf{Dec}_h(f(\mathsf{Enc}_h(m))) \neq \perp$, where the randomness is only over the choice of h. Let's also define the variables $\{Y_{m,z}\}_{(m,z) \in \{0,1\}^n}$ as $Y_{m,z} = 1$ if $h(m) = z$ and $h(m') = z'$ where $(m', z') = f(m, z)$. Then $X_m = \sum_z Y_{m,z}$ and therefore:

$$P_f = \Pr\left[\sum_{m \in \{0,1\}^k} X_m > 2^k\varepsilon\right] = \Pr\left[\sum_{(m,z) \in \{0,1\}^n} Y_{m,z} \geq 2^k\varepsilon\right]. \qquad (3.2)$$

We can ignore variables $Y_{m,z}$ for values (m, z) such that $f(m, z) = \perp$ since in that case $Y_{m,z} = 0$ always. Otherwise, we have $\Pr[Y_{m,z} = 1] = \Pr[h(m) = z \wedge h(m') = z'] \leq 2^{-2w}$ if $f(m, z) = (m', z') \neq (m, z)$ is not a fixed point and $\Pr[Y_{m,z} = 1] = \Pr[h(m) = z] = 2^{-w}$ if $f(m, z) = (m, z)$ is a fixed point.

We might hope that the random variables $Y_{m,z}$ are t-wise independent but they are not. For example, say $c = (m, z)$ and $c' = (m', z')$ are two values such that $f(c) = c'$ and $f(c') = c$ then $Y_{m,z} = 1 \Leftrightarrow Y_{m',z'} = 1$. In order to analyze the dependence between these random variables we represent the tampering function f as a directed graph. We define a graph G with vertices $V = \{0,1\}^n$, representing the codewords, and edges $E = \{(c, c') \mid c' = f(c),\ c = (m, z),\ m \in \{0,1\}^k,\ z \in \{0,1\}^w\}$, representing the tampering function. Note that every vertex has at most one outgoing edge, so we can label each edge $e_{m,z}$ with the value (m, z) of its unique origin vertex. Using this representation we can associate each $Y_{m,z}$ to the unique edge $e_{m,z}$.

Now consider any subset $E' \subseteq E$ of edges such that the origin $c = (m, z)$ and destination $c' = (m', z') = f(c)$ of each edge (c, c') in E' is disjoint from the origin or destination of all other edges in E'. We call such sets E' *vertex-disjoint*. Then for any $S \subseteq E'$ of size $|S| = t$ we have:

$$\Pr\left[\bigwedge_{(m,z) \in S} Y_{m,z} = 1\right] \leq \prod_{(m,z) \in S} \Pr[Y_{m,z} = 1] \qquad (3.3)$$

This is because $Y_{m,z} = 1$ iff $h(m) = z \wedge h(m') = z'$ where $(m', z') = f(m, z)$. Let $\tilde{S} \subseteq V$ be all the vertices contained in either the origin or destination of edges in E (so $|\tilde{S}| = 2S$ since $S \subseteq E'$ is vertex-disjoint). If two vertices (m, z) and (m, z') in $\tilde{S}$ share the same m, then it cannot be the case that $h(m) = z$ and $h(m) = z'$ so $\Pr[\bigwedge_{(m,z) \in S} Y_{m,z} = 1] = 0$. Otherwise, if the values m contained in $\tilde{S}$ are all disjoint, then variables $Y_{m,z}$ are independent since h is $2t$-wise independent.

Now we will partition the set E of edges into smaller subsets each of which is vertex-disjoint. Firstly, let E_{sl} be the set of all self-loops (fixed point) edges $e_{(m,z)}$ such that $f(m, z) = (m, z)$. It is clear that E_{sl} is vertex-disjoint. Since f is a φ-few fix points, μ-entropy function, we know that $|E_{\mathsf{sl}}| \leq \varphi 2^n$. Furthermore, $\mathbf{E}[\sum_{e_{(m,z)} \in E_{\mathsf{sl}}} \sum Y_{m,z}] \leq \varphi 2^{n-w} \leq \varphi 2^k$. By applying Lemma 2.1, with some $t_{\mathsf{min}} \leq t$ we get:

$$\begin{aligned}\Pr\left[\sum_{e_{(m,z)} \in E_{\mathsf{sl}}} Y_{m,z} \geq \frac{\varepsilon}{2} 2^k\right] &\leq \Pr\left[\sum_{e_{(m,z)} \in E_{\mathsf{sl}}} Y_{m,z} \geq \varphi 2^k + (\frac{\varepsilon}{2} - \varphi) 2^k\right] \\ &\leq \Pr\left[\sum_{e_{(m,z)} \in E_{\mathsf{sl}}} Y_{m,z} \geq \varphi 2^k + (\frac{\varepsilon}{4}) 2^k\right] \qquad (3.4) \\ &\leq 8\left(\frac{2t_{\mathsf{min}}}{(\frac{\varepsilon}{4})2^k}\right)^{t_{\mathsf{min}}/2} \leq 8\left(\frac{1}{2}\right)^{t_{\mathsf{min}}} \qquad (3.5)\end{aligned}$$

where inequality 3.4 follow by requiring that $\varphi \leq \varepsilon/4$ and the right hand inequality of 3.5 follows by requiring that $(\varepsilon/4)2^k \geq 8t_{\mathsf{min}}$. This in turn follows by observing that $k \geq \mu \geq \log(t_{\mathsf{min}}) + w + \log(1/\varepsilon) + 5$.

Next we define $E_{\mathsf{nsl}} = E \setminus E_{\mathsf{sl}}$ to be the set of edges that are not self loops. Let G_{nsl} be the corresponding graph consisting of G with all self-loops removed. Since f is a φ-few fix points, μ-entropy function, we know that each vertex in graph G has in-degree of at most $2^{n-\mu}$ and out-degree 1, and therefore total degree at most $2^{n-\mu}+1$. By a theorem of Shannon [Sha49] on edge-colorings, we can color the edges of such a graph using at most $q \leq (3/2)2^{n-\mu}+1 \leq 2^{n-\mu+1}$ colors so that no two neighboring edges (ignoring direction) are colored with the same color. In other words, we can partition E_{nsl} into exactly $q := 2^{n-\mu+1}$ subsets $E_1, \ldots, E_q$ such that each E_i is vertex-disjoint (we can always add dummy colors to make this exact). Further, recall that $\mathbf{E}[Y_{m,z}] \leq 2^{-2w}$ for edges $e_{(m,z)} \in E_{\mathsf{nsl}}$ that are not self loops. This implies that:

$$\begin{aligned}
\Pr\left[\sum_{e_{m,z}\in E_{\mathsf{nsl}}} Y_{m,z} \geq \left(\frac{\varepsilon}{2}\right)2^k\right] &\leq \Pr\left[\exists i \in [q] \ : \sum_{e_{m,z}\in E_i} Y_{m,z} \geq \left(\frac{\varepsilon}{2}\right)2^k\left(\frac{1}{2}\right)\left(\frac{|E_i|}{2^n}+\frac{1}{q}\right)\right] && (3.6)\\
&\leq \sum_{i\in[q]}\Pr\left[\sum_{e_{m,z}\in E_i} Y_{m,z} \geq \left(\frac{\varepsilon}{4}\right)2^{-w}\left(|E_i|+2^{\mu-1}\right)\right] && (3.7)\\
&\leq \sum_{i\in[q]}\Pr\left[\sum_{e_{m,z}\in E_i} Y_{m,z} \geq |E_i|2^{-2w}+A\right] && (3.8)\\
&\qquad \text{where } A = \left(\frac{\varepsilon}{4}\right)2^{k-n}\left(|E_i|+2^{\mu-1}\right) - |E_i|2^{-2w}.
\end{aligned}$$

Inequality 3.6 follows by observing that $\sum_{i\in[q]}\left(\frac{|E_i|}{2^n}+\frac{1}{q}\right) \leq 2$, inequality 3.7 follows by substituting $q = 2^{n-\mu+1}$ and $w = n-k$, and inequality 3.8 follows by the union bound. We can also bound:

$$\begin{aligned}
A &= \left(\frac{\varepsilon}{4}\right)2^{-w}\left(|E_i|+2^{\mu-1}\right) - |E_i|2^{-2w}\\
&\geq |E_i|2^{-w}\left(\frac{\varepsilon}{4}-2^{-w}\right)+\frac{\varepsilon}{4}2^{-w}2^{\mu-1}\\
&\geq |E_i|2^{-2w}+\frac{\varepsilon}{4}2^{-w}2^{\mu-1} && (3.9)
\end{aligned}$$

where equation 3.9 follows by requiring that $2^{-w} \leq \frac{\varepsilon}{8} \Leftrightarrow w \geq \log(1/\varepsilon)+3$ and therefore $\left(\frac{\varepsilon}{4}-2^{-w}\right) \geq \frac{\varepsilon}{8} \geq 2^{-w}$. This shows that $A \geq |E_i|2^{-2w}$. Continuing from equation 3.8 and applying Lemma 2.1 with $t_{\mathsf{min}} \leq t$ we get:

$$\begin{aligned}
\Pr\left[\sum_{e_{m,z}\in E_{\mathsf{nsl}}} Y_{m,z} \geq \left(\frac{\varepsilon}{2}\right)2^k\right] &\leq q8\left(\frac{2t_{\mathsf{min}}}{\frac{\varepsilon}{4}2^{-w}2^{\mu-1}}\right)^{t_{\mathsf{min}}/2}\\
&\leq 2^{n-\mu+1}8\left(\frac{1}{2}\right)^{t_{\mathsf{min}}} && (3.10)
\end{aligned}$$

where 3.10 follows by requiring $\frac{\varepsilon}{4}2^{-w}2^{\mu-1} \geq 8t_{\mathsf{min}} \Leftrightarrow \mu \geq \log(t_{\mathsf{min}})+w+\log(1/\varepsilon)+6$.

Finally, combining 3.1, 3.2, 3.5 and 3.10 we have:

$$\begin{aligned}
\Pr[\mathsf{BAD}] &\leq \sum_{f\in\mathcal{F}} P_f = \sum_{f\in\mathcal{F}}\Pr\left[\sum_{(m,z)\in\{0,1\}^n} Y_{m,z} \geq 2^k\varepsilon\right]\\
&\leq \sum_{f\in\mathcal{F}}\left(\Pr\left[\sum_{e_{m,z}\in E_{\mathsf{sl}}} Y_{m,z} \geq \left(\frac{\varepsilon}{2}\right)2^k\right]+\Pr\left[\sum_{e_{m,z}\in E_{\mathsf{nsl}}} Y_{m,z} \geq \left(\frac{\varepsilon}{2}\right)2^k\right]\right)\\
&\leq |\mathcal{F}|\left(8\left(\frac{1}{2}\right)^{t_{\mathsf{min}}}+2^{n-\mu+1}8\left(\frac{1}{2}\right)^{t_{\mathsf{min}}}\right)\\
&\leq 16|\mathcal{F}|2^{n-\mu+1}\left(\frac{1}{2}\right)^{t_{\mathsf{min}}} \leq \rho
\end{aligned}$$

where the last line follows by requiring that $t_{\mathsf{min}} \geq \log|\mathcal{F}|+n-\mu+\log(1/\rho)+5$ where $n = k+w$. This proves the theorem. □

3.2 Upgrading Weak Tamper-Detection Codes

We now show how to convert weak tamper-detection codes to (standard) tamper-detection codes with security for a worst-case message. We do so via a “composed code” construction following Figure 3.1, which will be useful throughout the paper. The idea of such composed constructions, is to choose the inner code according to the weaknesses of the outer code; in other words the inner code complements the outer code to achieve a stronger notion of security which is expected from the composed code. In our case, the composed code is obtained by composing an inner “leakage-resilient (LR) code” and an outer “weak tamper-detection (WTD) code”. The weakness of the outer WTD code comes from the fact that it only guarantees security for a uniformly random message. On the other hand, the inner LR code ensures that one cannot tell between a LR encoding of some worst-case message and a uniformly random inner codeword in the context of the tampering experiment.

Let $(\mathtt{lrEnc}, \mathtt{lrDec})$ be a (k, k')-code such that $\mathtt{lrDec}$ never outputs $\perp$.
Let $(\mathtt{wtdEnc}, \mathtt{wtdDec})$ be a deterministic (k', n)-code. We define the composed (k, n)-code $(\mathsf{Enc}, \mathsf{Dec})$ via:

- $\mathsf{Enc}(m)$: Let $c_{in} \leftarrow \mathtt{lrEnc}(m)$ and output $c = \mathtt{wtdEnc}(c_{in})$.
- $\mathsf{Dec}(c)$: Let $c_{in} = \mathtt{wtdDec}(c)$. If $c_{in} = \perp$, output $\perp$ else output $m = \mathtt{lrDec}(c_{in})$.

In particular, let $(\mathtt{lrEnc}_{h_1}, \mathtt{lrDec}_{h_1})$ and $(\mathtt{wtdEnc}_{h_2}, \mathtt{wtdDec}_{h_2})$ be given by:

$$\begin{array}{ll} \mathtt{lrEnc}_{h_1}(m) = (r, h_1(r) \oplus m) : r \leftarrow U_{v_1} & \mathtt{lrDec}_{h_1}(r, x) = (h_1(r) \oplus x) \\ \mathtt{wtdEnc}_{h_2}(c_{in}) = (c_{in}, h_2(c_{in})) & \mathtt{wtdDec}_{h_2}(c_{in}, z) = c_{in} \text{ if } z = h_2(c_{in}) \text{ and } \perp \text{ otherwise.} \end{array}$$

where $h_1 : \{0,1\}^{v_1} \to \{0,1\}^k$, $h_2 : \{0,1\}^{k'} \to \{0,1\}^{v_2}$, $k' := k + v_1, n := k' + v_2$.
Then the composed code $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ is defined as:

$$\mathsf{Enc}_{h_1,h_2}(m) = \left\{ \begin{array}{c} r \leftarrow U_{v_1},\ x := m \oplus h_1(r), \\ z := h_2(r, x),\ \text{output } (r, x, z) \end{array} \right\} \quad , \quad \mathsf{Dec}_{h_1,h_2}(r, x, z) = \left\{ \begin{array}{c} \text{If } z \neq h_2(r, x),\ \text{output } \perp \\ \text{otherwise output } \ x \oplus h_1(r). \end{array} \right\}$$

Figure 3.1: Composed Code Construction

Definition 3.4. *For a function f and a coding scheme (E, D) define the binary leakage function of f on (E, D), denoted by $\mathsf{bL}_f\left[(\mathsf{E}, \mathsf{D})\right]$ as follows,*

$$\mathsf{bL}_f\left[(\mathsf{E}, \mathsf{D})\right](x) = \left\{ \begin{array}{ll} 1 & \text{if } \mathsf{D}(f(\mathsf{E}(x))) = \perp \\ 0 & \text{otherwise.} \end{array} \right.$$

For a family of functions $\mathcal{F}$, define binary leakage function family as $\mathcal{BL}_{\mathcal{F}}\left[(\mathsf{E}, \mathsf{D})\right] = \{\mathsf{bL}_f\left[(\mathsf{E}, \mathsf{D})\right] \ | \forall\ f \in \mathcal{F}\}$. When the coding scheme is implicit we omit the index (E, D) and write bL_f, $\mathcal{BL}_{\mathcal{F}}$.

Theorem 3.5. *Let $\mathcal{F}$ be a family of functions, let $(\mathtt{wtdEnc}, \mathtt{wtdDec})$ be a (k', n)-coding scheme which is $(\mathcal{F}, \varepsilon)$-weak tamper detection code and let $(\mathtt{lrEnc}, \mathtt{lrDec})$ be a (k, k')-coding scheme which is $(\mathcal{BL}_{\mathcal{F}}\left[(\mathtt{wtdEnc}, \mathtt{wtdDec})\right], \gamma)$-leakage resilient code. Then the composed code $(\mathsf{Enc}, \mathsf{Dec})$ of Figure 3.1 is a (k, n) coding scheme which is $(\mathcal{F}, \gamma + \varepsilon)$-secure tamper detection code.*

Proof. Firstly, $\forall m \in \{0,1\}^k, \forall f \in \mathcal{F}$ we have:

$$\begin{aligned} &|\Pr\left[\mathtt{wtdDec}(f(\mathtt{wtdEnc}(\mathtt{lrEnc}(m)))) \neq \perp\right] - \Pr\left[\mathtt{wtdDec}(f(\mathtt{wtdEnc}(U_{k'}))) \neq \perp\right]| \\ &= |\Pr\left[\mathsf{bL}_f(\mathtt{lrEnc}(m)) = 0\right] - \Pr[\mathsf{bL}_f(U_{k'}) = 0]| \\ &\leq \mathbf{SD}(\mathsf{bL}_f(\mathtt{lrEnc}(m)), \mathsf{bL}_f(U_{k'})) \leq \gamma \end{aligned}$$

where the second line follows from the definition of $\mathsf{bL}_f \in \mathcal{BL}_{\mathcal{F}}\left[(\mathtt{wtdEnc}, \mathtt{wtdDec})\right]$ and the last line from the fact that $(\mathtt{lrEnc}, \mathtt{lrDec})$ is a $(\mathcal{BL}_{\mathcal{F}}\left[(\mathtt{wtdEnc}, \mathtt{wtdDec})\right], \gamma)$-leakage resilient code .

Secondly, since $(\mathtt{wtdEnc}, \mathtt{wtdDec})$ is an $(\mathcal{F}, \varepsilon)$-weak tamper detection code we also have $\forall f \in \mathcal{F}$:

$$\Pr\left[\mathtt{wtdDec}(f(\mathtt{wtdEnc}(U_{k'}))) \neq \perp\right] \leq \varepsilon$$

Combining the above, we get $\forall m \in \{0,1\}^k, \forall f \in \mathcal{F}$:

$$\Pr\left[\mathtt{wtdDec}(f(\mathtt{wtdEnc}(m))) \neq \perp\right] \leq \varepsilon + \gamma$$

which proves the theorem. □

Combining the results of theorems 2.4, 3.3 and 3.5 gives us the following corollary.

Corollary 3.6. *Let* $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ *be the construction in Figure 3.1 where* h_1 *is chosen from a* t*-wise independent hash family* $\mathcal{H}_1$*, and* h_2 *is chosen from a* t*-wise independent hash family* $\mathcal{H}_2$*. Then, for any family of* φ*-few fix points,* μ*-entropy functions* $\mathcal{F}$*, the code* $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ *is an* $(\mathcal{F}, 2\varepsilon)$*-tamper detection code with probability* $1-\rho$ *over the choice of* h_1 *and* h_2*, as long as:*

$$\begin{aligned} t &\geq t_{\mathsf{min}}, & t_{\mathsf{min}} &:= O(\log|\mathcal{F}| + k + v_1 + \log(1/\rho)) \\ v_1 &\geq \log(t_{\mathsf{min}}) + 2\log(1/\varepsilon) + O(1), & v_2 &\geq \log(1/\varepsilon) + O(1). \\ \mu &\geq \log(t_{\mathsf{min}}) + v_2 + \log(1/\varepsilon) + O(1), & \varphi &\leq \varepsilon/4 \end{aligned}$$

For example, if $\rho = \varepsilon = 2^{-\lambda}$ *for security parameter* λ *and,* $|\mathcal{F}| \geq \max\{2^k, 2^\lambda\}$ *we get:*

$$t = O(\log|\mathcal{F}|)\ ,\quad \mu = \log\log|\mathcal{F}| + 2\lambda + O(1)\ ,\quad \varphi = 2^{-(\lambda+O(1))},\quad v_1 = \log\log|\mathcal{F}| + 2\lambda + O(1)\ ,\quad v_2 = \lambda + O(1).$$

Proof. By Theorem 3.5 we need the inner code $(\mathtt{lrEnc}_{h_1}, \mathtt{lrDec}_{h_1})$ to be a $(\mathcal{BL}_{\mathcal{F}}[(\mathtt{wtdEnc}_{\mathsf{h}_2}, \mathtt{wtdDec}_{\mathsf{h}_2})], \varepsilon)$-leakage resilient code, and the outer code $(\mathtt{wtdEnc}_{h_2}, \mathtt{wtdDec}_{h_2})$ to be a $(\mathcal{F}, \varepsilon)$-weak tamper detection code. We notice that the leakage-family $\mathcal{BL}_{\mathcal{F}}[(\mathtt{wtdEnc}_{\mathsf{h}_2}, \mathtt{wtdDec}_{\mathsf{h}_2})]$ is a function family of size $|\mathcal{F}|$ where each function outputs $\ell = 1$ bits of leakage. Furthermore, the definition of the leakage family is fixed independently of the choice of h_1. Therefore, we can apply Theorems 2.4, 3.3 to get the above parameters. □

When $|\mathcal{F}| \leq 2^{2^{\alpha n}}$ for some constant $\alpha < 1$, then the overhead of the code is $n - k = v_1 + v_2 = \alpha n + O(\lambda)$ and therefore the rate of the code approaches $k/n \approx (1-\alpha)$. The above codes can be made efficient when $|\mathcal{F}| \leq 2^{s(n)}$ for some polynomial s, where the efficiency of the code depends on s (and in this case, the rate approaches 1). In particular, we get an efficient family of codes indexed by hash functions h_1, h_2 such that a random member of the family is a tamper-detection code for $\mathcal{F}$ with overwhelming probability. We can also think of this as an efficient construction in the common random string (CRS) model, where h_1, h_2 are given in the CRS.

Example: Tampering via Polynomials. Let $\mathcal{F}^-_{\mathsf{poly},d}$ be the set of all polynomials $p(x)$ over the field $\mathbb{F}_{2^n}$ of degree d, excluding the identity polynomial $p(x) = x$ and the degree 0 polynomials $\{p(x) = c\ :\ c \in \mathbb{F}_{2^n}\}$. Then $\mathcal{F}^-_{\mathsf{poly},d}$ is an φ-few fix points, μ-entropy function where $\varphi = d/2^n$ and $\mu = n - \log d$. Furthermore $|\mathcal{F}^-_{\mathsf{poly},d}| = 2^{n(d+1)}$.

Using corollary 3.6, we see that there exist $(\mathcal{F}^-_{\mathsf{poly},d}, \varepsilon)$-tamper-detection codes for degrees up to $d = 2^{\alpha n}$, for constant $\alpha < \frac{1}{2}$, with security ε negligible in n. The rate of such codes approaches $(1-\alpha)$. Furthermore, when the degree $d = d(n)$ is polynomial in n, then we get an efficient construction in the CRS model with a rate that approaches 1.

Example: Tampering via Affine Functions. Let $\mathcal{F}^-_{\mathsf{affine},r}$ be the set of all affine tampering functions when we identify $\{0,1\}^n$ as the vector space $\mathbb{F}^m_{2^r}$ with $m = n/r$. We exclude the identity and constant functions. In particular, $\mathcal{F}^-_{\mathsf{affine},r}$ consists of all functions $f_{A,b}(x) = Ax + b$ where $A \in \mathbb{F}^{m\times m}_{2^r}, b \in \mathbb{F}^m_{2^r}$, and (1) A is not the all 0 matrix, (2) if A is the identity matrix then $b \neq 0$ is a non-zero vector.

In particular the family $\mathcal{F}^-_{\mathsf{affine},r}$ is a φ-few fix points, μ-entropy function where $\varphi = 2^{-r}$ and $\mu = r$. Furthermore the size of the family is $|\mathcal{F}^-_{\mathsf{affine},r}| \leq 2^{n^2+n}$. The high-entropy requirement is guaranteed by (1) and the few-fixed points requirement is guaranteed by (2) as follows. If x is a fixed point then $Ax + b = x$ means that $(A - I_n)x = b$; if A is identity then this cannot happen since $b \neq 0$ and if A is not identity then this happens with probability at most 2^{-r} over a random x.

Using corollary 3.6, we get $(\mathcal{F}^-_{\mathsf{affine},r}, \varepsilon)$-tamper-detection codes where ε is negligible in r and the rate of the code approaches 1. Furthermore, we get efficient constructions of such codes in the CRS model.

A similar result would also hold if we considered all affine functions over the vector space $\mathbb{F}^n_2$, given by $f_{A,b}(x) = Ax + b$ where $A \in \mathbb{F}^{n\times n}_2, b \in \mathbb{F}^n_2$, but in this case we would need to add the additional requirement that rank of A is at least r (to ensure high entropy), and either b is not in the column-span of $(A - I_n)$ or the rank of $(A - I_n)$ is at least r (to ensure few fixed points).

Necessity of High Entropy. We argue that the restrictions on few fixed points and high entropy cannot be removed (in general). Firstly, we argue that the restriction on high entropy is necessary. We show that for every $\mu \leq k$, there is a family $\mathcal{F}$ of μ-entropy, 0-fixed-points functions of size $|\mathcal{F}| \leq (2^n)^{2^\mu}$ such that there does not exist any tamper-detection code for $\mathcal{F}$. In other words, this shows that we need to require $\mu \geq \min\{k, \log\log|\mathcal{F}| - \log n\}$

even when $\varphi = 0$. To see this, consider the family of functions $f_{c_0,\dots,c_{2^\mu-1}}(x)$ which have 2^μ distinct values $c_i \in \{0,1\}^n$ hard-coded, and if the μ least significant bits of x represent a number j (in binary), then $f_{c_0,\dots,c_{2^\mu-1}}(x) = c_j$. Consider the family $\mathcal{F}_{\mathsf{const},\mu} = \{f_{c_0,\dots,c_{2^\mu-1}} : c_i \in \{0,1\}^n, \forall_{i\neq j} c_i \neq c_j\}$ of all such functions. Then $|\mathcal{F}_{\mathsf{const},\mu}| \leq (2^n)^{2^\mu}$ and the family has μ-entropy. We can also consider a restricted family $\mathcal{F}'_{\mathsf{const},\mu}$ where the ordering of $c_0,\dots,c_{2^\mu-1}$ is always such that the μ least significant bits of c_i do not represent i, to ensure that the functions have no fixed points. This gives us a family $\mathcal{F}'_{\mathsf{const},\mu}$ of μ-entropy, 0-fixed-points functions of size $|\mathcal{F}'_{\mathsf{const},\mu}| \leq (2^n)^{2^\mu}$, but for any code $(\mathsf{Enc},\mathsf{Dec})$ with message-size $k \geq \mu$ and codeword size n, we can find a function $f_{c_0,\dots,c_{2^\mu-1}} \in \mathcal{F}'_{\mathsf{const},\mu}$ where $c_0,\dots,c_{2^\mu-1}$ are all distinct valid codewords, and therefore tampering goes undetected with probability 1.

Necessity of Few Fixed Points. Secondly, we argue that the restriction on few fixed points is necessary. It is clear that the single identity function has all fixed-points ($\varphi = 1$) and full entropy ($\mu = n$) but cannot be detected by any code. This shows that we must require $\varphi < 1$ even when $\mu = n$ and the function family is of size 1. We can further show that for any φ, there is a small family $\mathcal{F}$ of size $|\mathcal{F}| = 2^n$ which has φ-few fixed points and fully entropy ($\mu = n$) such that no code $(\mathsf{Enc},\mathsf{Dec})$ is even an $(\mathcal{F},\varphi)$-weak tamper-detection code. At first it may seem that a single permutation with φ-few fixed points could do the trick, but perhaps the code is chosen in such a way that the valid codewords c are never fixed points $f(c) \neq c$ even though a φ-fraction of random values x are fixed points $f(x) = x$. To defeat this, we consider a family $\mathcal{F}$ of permutations with φ-few fixed points where the set of fixed points is 1-wise independent: for each $x \in \{0,1\}^n$ we have $\Pr_{f\leftarrow\mathcal{F}}[f(x) = x] = \varphi$. For example $\mathcal{F}$ could consist of functions $f_k(x)$ with a hard-coded key $k \in \{0,1\}^n$ such that, if $x \oplus k \leq \varphi \cdot 2^n$ then $f_k(x) = x$, and otherwise f_k is defined in some fixed way to ensure that it is a permutation with no other fixed points (e.g., a cycle over all other values x such that $x \oplus k > \varphi \cdot 2^n$). In that case, for any code $(\mathsf{Enc},\mathsf{Dec})$ we have

$$\Pr_{f\leftarrow\mathcal{F}, m\leftarrow\{0,1\}^k}[\mathsf{Dec}(f(\mathsf{Enc}(m))) \neq \perp] \geq \Pr_{f,m,c\leftarrow\mathsf{Enc}(m)}[f(c) = c] \geq \varphi$$

By an averaging argument, this shows that there exists some $f \in \mathcal{F}$ such that $\Pr_{m\leftarrow\{0,1\}^k}[\mathsf{Dec}(f(\mathsf{Enc}(m))) \neq \perp] \geq \varphi$, which shows that the code is not an $(\mathcal{F},\varphi)$-weak tamper-detection code.

3.3 Predictable Codes

So far, we saw that tamper-detection codes are (only) achievable for "restricted" function families with few fixed points and high entropy. However, we now observe that such tamper-detection codes for restricted families can also provide a meaningful security guarantee for function families that don't have the above restrictions. The idea is to consider how tamper-detection can "fail". Firstly, it is possible that the tampering function gets a codeword which is a fixed-point c such that $f(c) = c$. In that case, tampering does not change the codeword and therefore will not get detected. In some sense this failure is not too bad since the codeword did not change. Secondly, it is possible that $f(c) = c^*$ where c^* is some "heavy" value that has many pre-images (responsible for low-entropy of f) and is a valid codeword. Fortunately, there cannot be too many such heavy values c^*. In other words, we can essentially predict what will happen as a result of tampering: either the codeword will not change at all, or it will be tampered to an invalid value that decodes to $\perp$, or it will be tampered to one of a few "heavy" values c^*. We capture this via the following definition of a "predictable code" (a similar notion called "bounded malleable codes" was defined in [FMVW14]) which says that the outcome of tampering a codeword via some function f lies in some "small" set $\mathcal{P}(f)$ which only depends on f and not on the message that was encoded.

Definition 3.7 (Predictable Codes)**.** *Let* $(\mathsf{Enc},\mathsf{Dec})$ *be a* (k,n)*-coding scheme. For a tampering function* $f : \{0,1\}^n \to \{0,1\}^n$ *and message* $m \in \{0,1\}^k$ *consider a distribution* $\mathsf{tamper}_{f,m}$ *that chooses* $c \leftarrow \mathsf{Enc}(m)$*, sets* $c' := f(c)$*, and if* $c' = c$ *it outputs a special symbol* same*, if* $\mathsf{Dec}(c') = \perp$ *it outputs* $\perp$*, and otherwise it outputs* c'*.*

For a family of tampering functions $\mathcal{F}$ *and a predictor* $\mathcal{P} : \mathcal{F} \to powerset(\{0,1\}^n \cup \mathsf{same} \cup \perp)$ *we say that the code is* $(\mathcal{F},\mathcal{P},\varepsilon)$*-predictable if for all* $f \in \mathcal{F}, m \in \{0,1\}^k$*:* $\Pr[\mathsf{tamper}_{f,m} \notin \mathcal{P}(f)] \leq \varepsilon$*. We say that the code is* $(\mathcal{F},\ell,\varepsilon)$*-predictable if it is* $(\mathcal{F},\mathcal{P},\varepsilon)$*-predictable for some* $\mathcal{P}$ *such that for all* $f \in \mathcal{F}$*,* $|\mathcal{P}(f)| \leq 2^\ell$*.*

Let $\mathcal{F}$ be a function family consisting of functions $f : \{0,1\}^n \to \{0,1\}^n$ and $\mu \in [n], \varphi > 0$ be two parameters. We say that $c' \in \{0,1\}^n$ is μ-heavy if $\Pr[f(c) = c' : c \leftarrow \{0,1\}^n] > 1/2^\mu$. Define

$$H_f(\mu) := \{c : c \in \{0,1\}^n \text{ is } \mu\text{-heavy}\}.$$

Note that $|H_f(\mu)| \leq 2^\mu$. For any function $f \in \mathcal{F}$ define the *restricted function* f' by setting $f'(c) := f(c)$ *unless*

(I) if $f(c) \in H_f(\mu)$ then $f'(c) := \perp$.

(II) if $\Pr_{x\in\{0,1\}^n}[f(x)=x] > \varphi$ and $f(c)=c$ then we set $f'(c) := \bot$.

It is clear that f' is a φ-few fix points, μ-entropy function. Define the family $\mathcal{F}[restrict(\mu,\varphi)] = \{f' \ : \ f \in \mathcal{F}\}$.

Theorem 3.8. *For any function family $\mathcal{F}$, if $(\mathsf{Enc},\mathsf{Dec})$ is an $(\mathcal{F}[restrict(\mu,\varphi)],\varepsilon)$-TDC then it is also an $(\mathcal{F},\mathcal{P},\varepsilon)$-predictable code where $\mathcal{P}(f) = \{\bot,\mathsf{same}\}\cup H_f(\mu)$. In particular, it is $(\mathcal{F},\mu+1,\varepsilon)$-predictable.*

Furthermore, if $\mathcal{F}$ has μ-high entropy, then $\mathcal{P}(f) = \{\bot,\mathsf{same}\}$. If $\mathcal{F}$ has φ-few fixed points then $\mathcal{P}(f) = \{\bot\}\cup H_f(\mu)$. If $\mathcal{F}$ has μ-high entropy and φ-few fixed points then $\mathcal{P}(f) = \{\bot\}$.

Proof. Assume $(\mathsf{Enc},\mathsf{Dec})$ is an $(\mathcal{F}[restrict(\mu,\varphi)],\varepsilon)$-TDC. Let $f\in\mathcal{F}$ be some function and $f'\in\mathcal{F}[restrict(\mu,\varphi)]$ defined as above so that $f'(c) = f(c)$ unless conditions (I) or (II) hold in which case $f'(c)=\bot$. This in turn implies that either: (A) $\mathsf{tamper}_{f,m} = \mathsf{tamper}_{f',m}$ or (B) $\mathsf{tamper}_{f,m}\in H_f(\mu)$ or (C) $\mathsf{tamper}_{f,m} = \mathsf{same}$. Case (B) corresponds to condition (I) and case (C) corresponds to condition (II).

By the TDC security of $(\mathsf{Enc},\mathsf{Dec})$, we know:

$$\forall f'\in\mathcal{F}[restrict(\mu,\varphi)],\ \forall m\in\{0,1\}^k \ : \ \Pr\left[\mathsf{tamper}_{f',m}\neq\bot\right] = \Pr\left[\mathsf{Dec}(f'(\mathsf{Enc}(m)))\neq\bot\right]\le\varepsilon.$$

which immediately implies that $\forall f\in\mathcal{F},\ \forall m\in\{0,1\}^k$:

$$\begin{aligned}\Pr\left[\mathsf{tamper}_{f,m}\notin\mathcal{P}(f)\right] &= \Pr\left[\mathsf{tamper}_{f,m}\notin\{\bot,\mathsf{same}\}\cup H_f(\mu)\right]\\ &\le \Pr\left[\mathsf{tamper}_{f',m}\neq\bot\right]\le\varepsilon\end{aligned}$$

This proves that $(\mathsf{Enc},\mathsf{Dec})$ is $(\mathcal{F},\mathcal{P},\varepsilon)$-predictable where $\mathcal{P}(f) = \{\bot,\mathsf{same}\}\cup H_f(\mu)$.

For the second part of the theorem, we simply notice that if $\mathcal{F}$ has μ-high entropy, then $H_f(\mu)=\emptyset$ and if $\mathcal{F}$ has φ-few fixed points then we only have cases (A) and (B) but not (C), and therefore $\mathcal{P}(f)$ need not output same. □

Combining the results of Theorem 3.8 and Corollary 3.6 gives us the following corollary.

Corollary 3.9. *Let $(\mathsf{Enc}_{h_1,h_2},\mathsf{Dec}_{h_1,h_2})$ be the construction in Figure 3.1 where h_1 is chosen from a t-wise independent hash family $\mathcal{H}_1$, and h_2 is chosen from a t-wise independent hash family $\mathcal{H}_2$. For any family of functions, $\mathcal{F}$, the code $(\mathsf{Enc}_{h_1,h_2},\mathsf{Dec}_{h_1,h_2})$ is an $(\mathcal{F},\ell,\varepsilon)$-predictable code with probability $1-\rho$ over the choice of h_1 and h_2, as long as:*

$$\begin{aligned} t &> t_{\mathsf{min}} \quad \textit{where} \quad t_{\mathsf{min}} = O(\log|\mathcal{F}| + k + v_1 + \log(1/\rho))\\ v_1 &\ge \log(t_{\mathsf{min}}) + 2\log(1/\varepsilon) + O(1) \qquad , \qquad v_2\ge\log(1/\varepsilon)+O(1).\\ \ell &\ge \log(t_{\mathsf{min}}) + v_2 + \log(1/\varepsilon) + O(1)\end{aligned}$$

For example, if $\rho=\varepsilon=2^{-\lambda}$ for security parameter λ and, $|\mathcal{F}|\ge\max\left\{2^k,2^\lambda\right\}$ we get:

$$t = O(\log|\mathcal{F}|)\ ,\quad \ell = \log\log|\mathcal{F}| + 2\lambda + O(1)\ ,\quad v_1 = \log\log|\mathcal{F}| + 2\lambda + O(1),\quad v_2 = \lambda + O(1).$$

Furthermore, if $\mathcal{F}$ has μ-high entropy for some $\mu\ge\log\log|\mathcal{F}|+2\lambda+O(1)$, then the code is $(\mathcal{F},\mathcal{P},\varepsilon)$-predictable with $\mathcal{P}(f)=\{\bot,\mathsf{same}\}$ and if $\mathcal{F}$ has φ-few fixed points for some $\varphi\le 2^{-(\lambda+O(1))}$ then $\mathsf{same}\notin\mathcal{P}(f)$. If $\mathcal{F}$ has μ-high entropy and φ-few fixed points then $\mathcal{P}(f)=\{\bot\}$.

A similar result to the above corollary was also shown in [FMVW14] for "bounded malleable codes" via a direct proof that did not go through tamper-detection codes. Here, we achieve some important improvements in parameters. Most importantly, our overhead is $v_1+v_2 = \log\log|\mathcal{F}| + O(\lambda)$ whereas previously it was at least $3\log\log|\mathcal{F}|$. This will be crucial for us to argue that our constructions achieve optimal rate $(1-\alpha)$ when the function family is of size $|\mathcal{F}| = 2^{2^{\alpha n}}$.

4 Basic Non-malleable codes

We now review the definition of non-malleable codes. Several different definitions (standard, strong, super) were proposed in [DPW10, FMVW14] and here we will by default use the strongest of these which was called "super" non-malleable codes in [FMVW14]. This notion considers the tampering experiment $\mathsf{tamper}_{f,m}$ that we previously used to define predictable codes: it chooses $c\leftarrow\mathsf{Enc}(m)$, sets $c' := f(c)$, and if $c'=c$ it outputs a special symbol same, if $\mathsf{Dec}(c')=\bot$ it outputs $\bot$, and otherwise it outputs c'. Non-malleable codes ensure that the output of the experiment is independent of the message m: for any two messages m_0,m_1 the distributions tamper_{f,m_0} and tamper_{f,m_1} should be statistically close.[2]

[2] For a weaker definition in [DPW10], the experiment $\mathsf{tamper}_{f,m}$ either outputs same if $c'=f(c)$ or $\mathsf{Dec}(c')$. In contrast, in our definition it outputs c' in full when $\mathsf{Dec}(c')\neq\bot$ which provides more information and makes the definition stronger.

Definition 4.1 (Non-malleable Code)**.** *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be a* (k, n)*-coding scheme and* $\mathcal{F}$ *be a family of functions* $f : \{0,1\}^n \to \{0,1\}^n$*. We say that the scheme is* $(\mathcal{F}, \varepsilon)$*-non-malleable if for any* $m_0, m_1 \in \{0,1\}^k$ *and any* $f \in \mathcal{F}$*, we have* $\mathsf{tamper}_{f,m_0} \approx_\varepsilon \mathsf{tamper}_{f,m_1}$ *where*

$$\mathsf{tamper}_{f,m} := \left\{ \begin{array}{c} c \leftarrow \mathsf{Enc}(m),\ c' := f(c) \\ \textit{output } \mathsf{same} \textit{ if } c' = c,\ \textit{output } \perp \textit{ if } \mathsf{Dec}(c') = \perp,\ \textit{and else output } c'. \end{array} \right\}$$

We will argue that the composed code construction in Figure 3.1 already achieves non-malleability. Intuitively, we will rely on two facts: (1) we already showed that the composed code is predictable meaning that the outcome of the tampering experiment can be thought of as providing at most small amount of leakage, (2) the inner code is leakage-resilient so the small amount of leakage cannot help distinguish between two messages m and m'.

To make the above intuition formal, we need to translate a class of tampering functions into a related class of leakage functions for which we need the inner code to be leakage resilient. This translation is given in the following definition.

Definition 4.2. *Let* $(\mathtt{wtdEnc}, \mathtt{wtdDec})$ *be a deterministic coding scheme, let* $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}^n\}$ *be a family of functions, and let* $\mathcal{P}(f) : \mathcal{F} \to$ *powerset* $\{\{0,1\}^n \cup \mathsf{same} \cup \perp\}$ *be a predictor. For a tampering function* $f \in \mathcal{F}$*, define the the corresponding leakage function* $\mathsf{L}_f(x)$ *as:*

$$\mathsf{L}_f(x) := \begin{cases} \mathsf{R}_f(x) & \textit{if } \mathsf{R}_f(x) \in \mathcal{P}(f) \\ \min \mathcal{P}(f) & \textit{otherwise.} \end{cases} \qquad \textit{where} \quad \mathsf{R}_f(x) := \begin{cases} \perp & \textit{if } \mathtt{wtdDec}(f(\mathtt{wtdEnc}(x)))) = \perp \\ \mathsf{same} & \textit{else if } f(\mathtt{wtdEnc}(x))) = \mathtt{wtdEnc}(x), \\ f(\mathtt{wtdEnc}(x)) & \textit{otherwise.} \end{cases}$$

where $\min \mathcal{P}(f)$ *denotes the minimal value in* $\mathcal{P}(f)$ *according to some ordering. Define the leakage family* $\mathcal{L}_f\left[(\mathtt{wtdEnc}, \mathtt{wtdDec}), \mathcal{P}\right] := \{\mathsf{L}_f \mid \forall f \in \mathcal{F}\}$*. When the coding scheme and the predictor function are implicit we simply write* $\mathcal{L}_{\mathcal{F}}$*.*

Next we argue that the composed code construction in Figure 3.1 is non-malleable for $\mathcal{F}$ as long as the composed code is predictable for $\mathcal{F}$ and the inner code is leakage resilient for $\mathcal{L}_{\mathcal{F}}$. Intuitively, the outer predictable code ensures that tampering can only result in a few possible values, while the inner leakage-resilient code ensures that the choice of which of these few values is the actual outcome of tampering does not reveal anything about the underlying message.

Theorem 4.3. *Let* $\mathcal{F}$ *be a finite family of functions and let* $(\mathsf{Enc}, \mathsf{Dec})$ *be the composed coding scheme of Figure 3.1 constructed using an inner code* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *and an outer code* $(\mathtt{wtdEnc}, \mathtt{wtdDec})$*. If* $(\mathsf{Enc}, \mathsf{Dec})$ *is* $(\mathcal{F}, \mathcal{P}, \varepsilon)$*-predictable and* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *is a* $(\mathcal{L}_f\left[(\mathtt{wtdEnc}, \mathtt{wtdDec}), \mathcal{P}\right], \gamma)$*-leakage resilient for some predictor* $\mathcal{P}$*, then* $(\mathsf{Enc}, \mathsf{Dec})$ *is a* $(\mathcal{F}, 2(\gamma + \varepsilon))$*-non-malleable code.*

Proof. For all $m_0, m_1 \in \{0,1\}^k$ and for all $f \in \mathcal{F}$, we have:

$$\begin{aligned} \mathsf{tamper}_{m_0,f} &\equiv \mathsf{R}_f(\mathtt{lrEnc}(m_0)) \\ &\approx_\varepsilon \mathsf{L}_f(\mathtt{lrEnc}(m_0)) && (4.1) \\ &\approx_\gamma \mathsf{L}_f(U_{k'}) && (4.2) \\ &\approx_\gamma \mathsf{L}_f(\mathtt{lrEnc}(m_1)) && (4.3) \\ &\approx_\varepsilon \mathsf{R}_f(\mathtt{lrEnc}(m_1)) && (4.4) \\ &\equiv \mathsf{tamper}_{m_1,f} \end{aligned}$$

where $\equiv$ denotes distributional equivalence. Lines 4.1 and 4.4 follow from the fact that $\Pr[\mathsf{R}_f(\mathtt{lrEnc}(m)) \neq \mathsf{L}_f(\mathtt{lrEnc}(m))] \leq \Pr[\mathsf{tamper}_{f,m} \notin \mathcal{P}(f)] \leq \varepsilon$ since the combined code is $(\mathcal{F}, \mathcal{P}, \varepsilon)$-predictable. Lines 4.2 and 4.3 follow from the fact that $(\mathtt{lrEnc}, \mathtt{lrDec})$ is a $(\mathcal{L}_{\mathcal{F}}, \gamma)$-leakage resilient code. □

Corollary 4.4. *Let* $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ *be the construction in Figure 3.1 where* h_1 *is chosen from a hash family* $\mathcal{H}_1$ *which is t-wise independent, and* h_2 *is chosen from a hash family* $\mathcal{H}_2$ *which is t-wise independent. For any family of functions* $\mathcal{F}$*, the composed code* $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ *is a* $(\mathcal{F}, \varepsilon)$*-non-malleable code with probability* $1 - \rho$ *over the choice of* h_1 *and* h_2*, as long as the following holds:* $v_2 = \log(1/\varepsilon) + O(1)$*,* $v_1 \geq v_{\min}$ *and* $t \geq t_{\min}$ *for* $B := \log|\mathcal{F}| + k + v_1 + \log(1/\rho) + O(1)$ *and*

$$\begin{array}{rlll} \textit{either} & t_{\min} = O(B) & \textit{and} & v_{\min} = \log(B) + \log\log(B/\varepsilon^2) + 4\log(1/\varepsilon) + O(1) \\ \textit{or} & t_{\min} = O(B/\varepsilon^2) & \textit{and} & v_{\min} = \log(B) + 4\log(1/\varepsilon) + O(1). \end{array}$$

In particular, if $\rho = \varepsilon = 2^{-\lambda}$ and $|\mathcal{F}| \geq \{2^k, 2^\lambda\}$, $\lambda \geq \log\log\log|\mathcal{F}|$ then we get:

$$t_{\mathsf{min}} = O(\log|\mathcal{F}|) \qquad \text{and} \qquad v_{\mathsf{min}} = \log\log|\mathcal{F}| + O(\lambda)$$

Proof. By Corollary 3.9, $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ is a $(\mathcal{F}, \ell, \varepsilon/2)$-Predictable code with probability $1 - (2/3)\rho$ over the choice of h_1 and h_2 for $\ell = \log(B) + \log(1/\varepsilon) + v_2 + O(1) = \log(B) + 2\log(1/\varepsilon) + O(1) = \log(B/\varepsilon^2) + O(1)$. Moreover, we can rely on Theorem 2.4 which tells us that the inner code is $(\mathcal{L}_{\mathcal{F}}, \varepsilon)$-leakage resilient code with probability $1 - \rho/3$ over the choice of h_1 as long as $v_1 \geq v_{\mathsf{min}}$ and $t \geq t_{\mathsf{min}}$ where

$$\begin{aligned}
\text{either} \qquad t_{\mathsf{min}} = O(B) \qquad \text{and} \qquad v_{\mathsf{min}} &= \log(B + 2^\ell) + \log\ell + 2\log(1/\varepsilon) + O(1), \\
&= \log(B/\varepsilon^2) + \log\log(B/\varepsilon^2) + 2\log(1/\varepsilon) + O(1) \\
\text{or} \qquad t_{\mathsf{min}} = O(B + 2^\ell) \qquad \text{and} \qquad v_{\mathsf{min}} &= \log(B + 2^\ell) + 2\log(1/\varepsilon) + O(1). \\
&= \log(B/\varepsilon^2) + 2\log(1/\varepsilon) + O(1)
\end{aligned}$$

Both of these events occur simultaneously with probability at least $1 - \rho$ and, by Theorem 4.3, the composed code is then $(\mathcal{F}, \varepsilon)$-non-malleable code. □

The above corollary tells us that, for any tampering function family $\mathcal{F}$ of size up to $|\mathcal{F}| = 2^{2^{\alpha n}}$ for $\alpha < 1$ there exist (inefficient) non-malleable codes for $\mathcal{F}$ with additive overhead $v_1 + v_2 = \alpha n + O(\lambda)$ and therefore rate approaching $(1 - \alpha)$. This matches the positive results on the rate of non-malleable codes of [CG13a] and is known to be optimal. Furthermore, if $|\mathcal{F}| = 2^{s(n)}$ for some polynomial $s(n)$, then we can get an efficient family of codes such that a random member of the family is an $(\mathcal{F}, 2^{-\lambda})$ non-malleable code for $\mathcal{F}$ with overwhelming probability, where the efficiency of the code is $\mathsf{poly}(s(n), n, \lambda)$ and the additive overhead of the code is $v_1 + v_2 = O(\log n + \lambda)$, and therefore the rate approaches 1. For example, $\mathcal{F}$ could be the family of all circuits of size at most $s(n)$ and we can view our construction as giving an efficient non-malleable code for this family in the CRS model, where the random choice of h_1, h_2 is specified by the CRS. This matches the positive results of [FMVW14]. Therefore, we get a single construction and analysis which achieves the "best of both worlds" depending on the setting of parameters.

5 Continuous Non-malleable Codes

Intuitively, a continuous non-malleable code guarantees security against an attacker that can continuously tamper with a codeword without the codeword being refreshed. We give a unified definition of four variants of such codes depending on two binary variables: (I) a "self-destruct" flag sd which indicates whether the game stops if a codeword is ever detected to be invalid (corresponding to a device self-destructing or erasing its internals) or whether the attacker can continue tampering no matter what, (II) a "persistent" flag prs indicating whether each successive attack modifies the codeword that has been modified by previous attacks, or whether tampering can also always be applied to the original codeword (corresponding to the case where a copy of the original codeword may remain in some auxiliary memory on the device).

In more detail, the definition of continuous non-malleable codes considers a tampering experiment where a codeword $c \leftarrow \mathsf{Enc}(m)$ is chosen in the beginning. The attacker repeatedly chooses tampering functions f_i and we set the i'th tampered codeword to either be $c_i = f_i(c)$ if the "persistent" flag prs is off or $c_i = f_i \circ f_{i-1} \circ \cdots \circ f_1(c)$ if prs is on. Just like the non-continuous definition, we give the attacker the special symbol same if the codeword remains unchanged $c_i = c$, we give the attacker $\perp$ if $\mathsf{Dec}(c_i) = \perp$ and we give the attacker the entire codeword c_i otherwise. In the case where the "self-destruct" flag sd is on, the experiment immediately stops if we give the attacker $\perp$ to capture that the device self-destructs and the attacker has no more opportunity to tamper. In the case where the "persistent" flag prs is on, we also stop the experiment if the attacker ever gets the entire tampered codeword c_i (and not same or $\perp$) in some period i. This is without loss of generality since any future tampering attempts can be answered using c_i alone and hence there is no point in continuing the experiment.

Definition 5.1. *Let $(\mathsf{Enc}, \mathsf{Dec})$ be a (k, n)-coding scheme and let $\mathcal{F}$ be a family of tampering functions $f : \{0,1\}^n \to \{0,1\}^n$. We define four types of* continuous non-malleable code*s (CNMC) parameterized by the flags $\mathsf{sd} \in \{0,1\}$ (self destruct) and $\mathsf{prs} \in \{0,1\}$ (persistent tampering). We say that the scheme is a $(\mathcal{F}, T, \varepsilon)$-CNMC$[\mathsf{prs}, \mathsf{sd}]$ if for any two messages $m_0, m_1 \in \{0,1\}^k$, for any $\mathcal{F}$-legal adversary $\mathcal{A}$ we have $\mathsf{ConTamper}_{\mathcal{A},T,m_0} \approx_\varepsilon \mathsf{ConTamper}_{\mathcal{A},T,m_1}$ where the experiment $\mathsf{ConTamper}_{\mathcal{A},T,m}$ is defined in figure 5.1. For $\mathsf{prs} = 0$, an adversary $\mathcal{A}$ is $\mathcal{F}$-*legal*, if the tampering functions chosen in each round $i \in [T]$ satisfy $f_i \in \mathcal{F}$. For $\mathsf{prs} = 1$, an adversary $\mathcal{A}$ is $\mathcal{F}$-*legal *if the tampering functions f_i chosen in each round i satisfy $(f_i \circ \ldots \circ f_1) \in \mathcal{F}$.*

```
$\mathsf{ConTamper}_{\mathcal{A},T,m}[\mathsf{prs},\mathsf{sd}]$
        $c \leftarrow \mathsf{Enc}(m)$
        $f_0 := identity$
        Repeat  $i = 1, \ldots, T$
            $\mathcal{A}$ chooses a function $f'_i$
            if $\mathsf{prs} \overset{?}{=} 1 : f_i := f'_i \circ f_{i-1}$        else $f_i := f'_i$
            $c' := f_i(c)$
            if $c' \overset{?}{=} c$ :                     $\mathcal{A}$ receives same
            else if $\mathsf{Dec}(c') \overset{?}{=} \perp$ :        { $\mathcal{A}$ receives $\perp$,   if $\mathsf{sd} \overset{?}{=} 1$    experiment stops}
            else  :                            { $\mathcal{A}$ receives $c'$,   if $\mathsf{prs} \overset{?}{=} 1$    experiment stops}
            The output of the experiment is the view of $\mathcal{A}$.
```

Figure 5.1: Continuous Non-Malleability Experiment

Tamper-Resilience via Continuous Non-malleable Codes. We note that the above definition of continuous non-malleable codes directly guarantees strong protection against continuous tampering attacks on a device implementing an arbitrary cryptographic scheme. We imagine that the secret key sk of the cryptographic scheme is encoded using such a code and the codeword is stored on the device; on each invocation, the device first decodes to recover the key and then executes the original scheme. The device never needs to update the codeword. In the setting of "self-destruct" we need the device to erase the stored codeword or simply stop functioning if on any invocation it detects that the codeword decodes to $\perp$. In the setting of non-persistent tampering, we assume the tampering attacks have access to the original codeword rather than just the current value on the device (e.g., because a copy of the original codeword may remain somewhere on the device in some auxiliary memory).

We guarantee that whatever an attacker can learn by continuously tampering with the codeword stored on the device and interacting with the tampered device, could be simulated given only black-box access to the original cryptographic scheme with the original secret key sk and without the ability to tamper. The main idea is that the information that the attacker can learn by interacting with a tampered device is completely subsumed by the information provided by the experiment $\mathsf{ConTamper}_{\mathcal{A},T,sk}$ and interaction with the original untampered device: if the tampering results in $\perp$ then the attacker learns nothing from the tampered device beyond that this happened, if it results in same the attacker gets the outputs of the original device, and if it results in a new valid codeword c' then the attacker at most learns c'. On the other hand, $\mathsf{ConTamper}_{\mathcal{A},T,sk}$ does not provide any additional information about sk by the security of the continuous non-malleable codes.

A similar connection between standard non-malleable codes and tamper-resilience was formalized by [DPW10] in the case where the device updates the stored codeword after each invocation and the above claim simply says that the connection extends to the setting where the device does not update the codeword after each invocation and we use a continuous non-malleable code. The formalization of the above claim and formal proof would be essentially the same as in [DPW10].

5.1 Tool: Repeated Leakage Resilience

Towards the goal of constructing continuous non-malleable codes, we will rely on a new notion of leakage-resilient codes which we call "repeated leakage resilience".

Consider a family $\mathcal{F}$ consisting of leakage functions $f : \{0,1\}^n \to \{0,1\}^\ell$ where each function $f \in \mathcal{F}$ has an associated unique "repeat" value $\mathsf{repeat}_f \in \{0,1\}^\ell$. An attacker can specify a vector of functions $(f_1, \ldots, f_T)$ with $f_i \in \mathcal{F}$. We apply the functions to the codeword c one-by-one: if the output if $f_i(c) = \mathsf{repeat}_{f_i}$ we continue to the next function until we get the first i for which $f_i(c) \neq \mathsf{repeat}_{f_i}$ in which case we output $f_i(c)$. As in the case of standard leakage-resilient codes, we don't want the attacker to be able to distinguish between an encoding of a particular message m versus a random codeword.

Definition 5.2. *Let* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *be a* (k,n)*-coding scheme. Let* $\mathcal{F}$ *be a family of functions* $f : \{0,1\}^n \to \{0,1\}^\ell$ *where each function* $f \in \mathcal{F}$ *has an associated "repeat value"* $\mathsf{repeat}_f \in \{0,1\}^\ell$*. For an integer* $T \geq 1$*, we say that*

$(\mathtt{lrEnc}, \mathtt{lrDec})$ *is* $(\mathcal{F}, T, \varepsilon)$-repeated-leakage-resilient*, if for any* $\bar{f} = (f_1, \dots, f_T) \in \mathcal{F}^T$ *and any* $m \in \{0,1\}^k$ *we have* $\mathsf{RLeakage}(\bar{f}, \mathtt{lrEnc}(m)) \approx_\varepsilon \mathsf{RLeakage}(\bar{f}, U_n)$*, where we define* $\mathsf{RLeakage}(\bar{f}, c)$ *to output the value* $(i, f_i(c))$ *for the smallest* i *such that* $f_i(c) \neq \mathsf{repeat}_{f_i}$*, or* $\perp$ *if no such* i *exists.*

Notice that if we allow T to be as large as $T = 2^n$ then there is a simple family $\mathcal{F}$ of only 2^n leakage functions with just 1-bit output such $\mathsf{RLeakage}(\bar{f}, \mathtt{lrEnc}(m))$ completely reveals m. In particular consider the family $\mathcal{F}$ of functions $f_{c'} : \{0,1\}^n \to \{\mathsf{stop}, \mathsf{repeat}\}$ that have a hard-coded value $c' \in \{0,1\}^n$ and we define $f_{c'}(c) = \mathsf{stop}$ if $c' = c$ and $f_{c'}(c) = \mathsf{repeat}$ otherwise (1 bit output). Let $\bar{f}$ be a vector of all functions in $\mathcal{F}$. Then $\mathsf{RLeakage}(\bar{f}, c)$ completely reveals c.

In general, we can always view repeated-leakage resilience as a special case of standard leakage resilience. In particular, for any family of leakage functions $\mathcal{F}$ with associated repeat values repeat_f we can define the family $\mathcal{F}_{\mathsf{repeat},T} := \{\mathsf{RLeakage}(\bar{f}, \cdot) \ : \ \bar{f} \in \mathcal{F}^T\}$. It is clear that an $(\mathcal{F}_{\mathsf{repeat},T}, \varepsilon)$-leakage-resilient code is also $(\mathcal{F}, T, \varepsilon)$-repeated-leakage-resilient. If the function-family $\mathcal{F}$ consists of functions with ℓ-bits of leakage then the functions in $\mathcal{F}_{\mathsf{repeat},T}$ have $\leq \ell + \log T + 1$ bits of leakage (e.g., the range of $f \in \mathcal{F}_{\mathsf{repeat},T}$ is $[T] \times \{0,1\}^\ell \cup \{\perp\}$). In other words, the amount of leakage only increases by $\log T + 1$ which is not very large. Unfortunately, even though the *amount* of leakage is not much larger, the *quality* of the leakage measured as the size of the leakage family is substantially larger: $|\mathcal{F}_{\mathsf{repeat},T}| = |\mathcal{F}|^T$. This hurts our parameters and requires the efficiency of our codes to depend on and exceed T. Summarizing the above discussion with the parameters of standard leakage-resilient codes from Section 2.1: Theorems 2.4 we get the following.

Theorem 5.3. *Let* $\mathcal{F}$ *be a family of functions of the form* $f : \{0,1\}^n \to \{0,1\}^\ell$ *where each function* f *has an associated "repeat" value* $\mathsf{repeat}_f \in \{0,1\}^\ell$*. If* $(\mathtt{lrEnc}, \mathtt{lrDec})$ *is a* (k,n)*-coding scheme which is* $(\mathcal{F}_{\mathsf{repeat},T}, \varepsilon)$*-leakage-resilient then it is also* $(\mathcal{F}, T, \varepsilon)$*-repeated-leakage-resilient. In particular let* $(\mathtt{lrEnc}_h, \mathtt{lrDec}_h)$ *be the family of* $(k, n = k + v)$*-codes defined in Section 2.1. Then with probability* $1 - \rho$ *over the choice of* $h \leftarrow \mathcal{H}$ *chosen from a* t*-wise independent hash function family, the code* $(\mathtt{lrEnc}_h, \mathtt{lrDec}_h)$ *is* $(\mathcal{F}, T, \varepsilon)$*-repeated-leakage-resilient as long as:*

$$t \geq t_{\min} := O(T \cdot \log |\mathcal{F}| + k + \log(1/\rho))$$
$$v \geq \log\left(t_{\mathsf{min}} + (2^{\ell+1} \cdot T)\right) + \log(\ell + \log T + 1) + 2\log(1/\varepsilon) + O(1)$$

In the natural setting where $\varepsilon = \rho = 2^{-\lambda}$ *and* $T, \ell \leq 2^\lambda$*,* $|\mathcal{F}| \geq \max\{2^\lambda, 2^k\}$*, for security parameter* λ*, we have*

$$t = O(T \cdot \log |\mathcal{F}|) \qquad\qquad v = \max\{\ell, \log\log|\mathcal{F}|\} + O(\lambda)$$

Recall that in standard leakage-resilient codes, if we have $|\mathcal{F}| = 2^{s(n)}$ for some polynomial $s(n)$, then our construction was efficient and we required polynomial independence $t = O(s(n))$. Unfortunately, for repeated leakage resilience, the above theorem requires us to now set $t = O(Ts(n))$ depending on T. In general, we would like to have a fixed efficient construction which is secure for any polynomial number of repeated leakage queries T, rather than the reverse quantifiers where for any polynomial T we have an efficient construction which depends on T. We believe that our analysis of repeated-leakage resilience is sub-optimal and that the above goal is possible. Intuitively, we believe that the sub-optimality of the analysis comes from the fact that it completely ignores the structure of the function family $\mathcal{F}_{\mathsf{repeat},T}$ and only counts the number of functions. However, the functions $\mathsf{RLeakage}(\bar{f}, \cdot)$ in the family have significantly restricted structure and are not much more complex than the functions $f \in \mathcal{F}$. Indeed, we can think of $\mathsf{RLeakage}(\bar{f}, \cdot)$ as first computing $f_1(c), \dots, f_T(c)$ and then computing some shallow circuit on top of the T outputs. We put forward the following conjecture, and leave it as an interesting open problem to either prove or refute it.

Conjecture 5.4. *Theorem 5.3 holds in the setting* $\varepsilon = \rho = 2^{-\lambda}$ *and* $T, \ell \leq 2^\lambda$*,* $|\mathcal{F}| \geq \max\{2^\lambda, 2^k\}$*, with* $t = O(\lambda \log|\mathcal{F}|)$ *instead of* $t = O(T \log |\mathcal{F}|)$*.*

5.2 Abstract Construction of Continuous Non-Malleable Codes

We now show how to construct continuous non-malleable codes for all four cases depending on self-destruct (yes/no) and persistent tampering (yes/no). Our constructions follow the same template as the construction of standard non-malleable codes. In particular, to get continuous non-malleability for some function family $\mathcal{F}$ we will use the composed code construction from Figure 3.1 and rely on the fact that the combined code is "predictable" for $\mathcal{F}$ with some predictor $\mathcal{P}$ satisfying certain properties and that the inner code is repeated leakage resilient for a corresponding leakage family. Intuitively, we want to ensure that if the predictor $\mathcal{P}(f)$ outputs a large set of possible values that might be the outcomes of any single tampering attempt via a function f, there is at most one value $\mathsf{repeat}_f \in \mathcal{P}(f)$ which causes the continuous tamper experiment in figure 5.1 to continue and not stop. Therefore, the adversary does no learn much information during the tampering experiment.

Theorem 5.5. *Let $\mathcal{F}$ be a family of tampering functions of the form $f\ :\ \{0,1\}^n \to \{0,1\}^n$. Let $(\mathsf{Enc}, \mathsf{Dec})$ be the composed code from Figure 3.1, constructed using an inner code $(\texttt{lrEnc}, \texttt{lrDec})$ and an outer code $(\texttt{wtdEnc}, \texttt{wtdDec})$. Assume $(\mathsf{Enc}, \mathsf{Dec})$ is $(\mathcal{F}, \mathcal{P}, \varepsilon_1)$-predictable code with some predictor $\mathcal{P}$ satisfying the conditions below, and $(\texttt{lrEnc}, \texttt{lrDec})$ a $(\mathcal{L}_{\mathcal{F}}, T, \varepsilon_2)$-repeated leakage resilient code for the leakage family $\mathcal{L}_{\mathcal{F}} = \mathcal{L}_{\mathcal{F}}\left[(\texttt{wtdEnc}, \texttt{wtdDec}), \mathcal{P}\right]$ (see Definition 4.2) and where each $\mathsf{L}_f \in \mathcal{L}_{\mathcal{F}}$ has an associated repeat value repeat_f as defined below. Then the composed code $(\mathsf{Enc}, \mathsf{Dec})$ is $(\mathcal{F}, T, \varepsilon)$-CNMC$[\mathsf{sd}, \mathsf{prs}]$ with $\varepsilon = 2T \cdot \varepsilon_1 + 2\varepsilon_2$.*

- For case $\mathsf{sd} = 0, \mathsf{prs} = 0$ we require that for all $f \in \mathcal{F}$, $|\mathcal{P}(f)| = 1$. In this case, the leakage-resilience requirement is vacuous since the leakage family $\mathcal{L}_{\mathcal{F}}$ has 0-bit output.
- For case $\mathsf{sd} = 1, \mathsf{prs} = 0$ we require that for all $f \in \mathcal{F}$, either $|\mathcal{P}(f)| = 1$ or $\mathcal{P}(f) = \{\bot, \mathsf{same}\}$. For each $\mathsf{L}_f \in \mathcal{L}_{\mathcal{F}}$, we define $\mathsf{repeat}_f = \mathcal{P}(f)$ if $|\mathcal{P}(f)| = 1$ or $\mathsf{repeat}_f = \mathsf{same}$ otherwise.
- For case $\mathsf{sd} = 0, \mathsf{prs} = 1$ we require that for all $f \in \mathcal{F}$, either $|\mathcal{P}(f)| = 1$ or $\mathsf{same} \notin \mathcal{P}(f)$. For each $\mathsf{L}_f \in \mathcal{L}_{\mathcal{F}}$, we define $\mathsf{repeat}_f = \mathcal{P}(f)$ if $|\mathcal{P}(f)| = 1$ or $\mathsf{repeat}_f = \bot$ otherwise.
- For case $\mathsf{sd} = 1, \mathsf{prs} = 1$, we have no requirements on $\mathcal{P}(f)$. For each $\mathsf{L}_f \in \mathcal{L}_{\mathcal{F}}$, we define $\mathsf{repeat}_f = \mathsf{same}$.

Proof. We need to show that for any two messages m_0, m_1 and any $\mathcal{F}$-legal attacker $\mathcal{A}$, $\mathsf{ConTamper}_{\mathcal{A},T,m_0} \approx_\varepsilon \mathsf{ConTamper}_{\mathcal{A},T,m_1}$. To do so, we define a series hybrid experiments. Consider a new experiment called $\mathsf{ConTamper}^i_{\mathcal{A},T,m}$

```
$\mathsf{ConTamper}^i_{\mathcal{A},T,m}[\mathsf{prs}, \mathsf{sd}]$
        $c_{in} \leftarrow \texttt{lrEnc}(m)$
        $f_0 = \textit{identity}$
        Repeat $j = 1, \ldots, T$
            $\mathcal{A}$ chooses a function $f'_j$
            if $\mathsf{prs} \overset{?}{=} 1$ : $f_j := f'_j \circ f_{j-1}$  else $f_j := f'_j$
            if $j \leq i$ : $c' = \mathsf{L}_{f_j}(c_{in})$  else $c' = \mathsf{R}_{f_j}(c_{in})$
            $\mathcal{A}$ receives $c'$
            if $c' \overset{?}{=} \bot$ and $\mathsf{sd} \overset{?}{=} 1$,  experiment stops
            if $c' \notin \{\bot, \mathsf{same}\}$ and $\mathsf{prs} \overset{?}{=} 1$,  experiment stops
```

Once again, the output of the experiment is the view of $\mathcal{A}$. Functions R_f and L_f are the leakage functions defined in Definition 4.2. Notice that if we use R_f in every iteration then we get experiment $\mathsf{ConTamper}_{\mathcal{A},T,m}$ as defined in Figure 5.1, therefore $\mathsf{ConTamper}^0_{\mathcal{A},T,m} \equiv \mathsf{ConTamper}_{\mathcal{A},T,m}$. We call the experiment that uses L_f in every iteration, $\mathsf{ConTamper}'_{\mathcal{A},T,m} \equiv \mathsf{ConTamper}^T_{\mathcal{A},T,m}$. By triangle inequality we have

$$\mathbf{SD}(\mathsf{ConTamper}_{\mathcal{A},T,m}, \mathsf{ConTamper}'_{\mathcal{A},T,m}) \leq \sum_{i=1}^{i=T} \mathbf{SD}\left(\mathsf{ConTamper}^{i-1}_{\mathcal{A},T,m}, \mathsf{ConTamper}^{i}_{\mathcal{A},T,m}\right)$$

Next we aim to show $\mathbf{SD}\left(\mathsf{ConTamper}^{i-1}_{\mathcal{A},T,m}, \mathsf{ConTamper}^{i}_{\mathcal{A},T,m}\right) \leq \varepsilon_1$ for all $i \in [T]$. Which implies that

$$\mathbf{SD}(\mathsf{ConTamper}_{\mathcal{A},T,m}, \mathsf{ConTamper}'_{\mathcal{A},T,m}) \leq T \cdot \varepsilon_1.$$

To start, note that the two experiments $\mathsf{ConTamper}^{i-1}_{\mathcal{A},T,m}$ and $\mathsf{ConTamper}^{i}_{\mathcal{A},T,m}$ are only different in the leakage function used at step i. This function is R_{f_i} in experiment $\mathsf{ConTamper}^{i-1}_{\mathcal{A},T,m}$ and is L_{f_i} in experiment $\mathsf{ConTamper}^{i}_{\mathcal{A},T,m}$. Notice that in both experiments the tampering function f_i chosen in iteration i is chosen non-adaptively (independently of the codeword c_{in}). This is because the messages the adversary receives in the first $i-1$ attacks are completely determined by the tampering functions chosen and independent of c_{in}. First of all, L_f always outputs an element in the predictable set $\mathcal{P}(f)$. Second of all, considering the way the experiments are defined, the only value that keeps the experiments running for each tamper function f is the unique value repeat_f, defined specifically for each of the four non-malleability cases. Therefore, the ith tampering function is chosen non-adaptively. Keep in mind that the experiments must reach the ith attack, otherwise there is no difference in the two experiments.

Given that the choice of the ith function is non-adaptive and the only difference is whether L_{f_i} is used or R_{f_i}, we have

$$\mathbf{SD}\left(\mathsf{ConTamper}^{i-1}_{\mathcal{A},T,m}, \mathsf{ConTamper}^{i}_{\mathcal{A},T,m}\right) = \Pr\left[\mathsf{R}_{f_i}(c_{in}) \neq \mathsf{L}_{f_i}(c_{in})\right] = \Pr\left[\mathsf{R}_{f_i}(c_{in}) \notin \mathcal{P}(f)\right] \leq \varepsilon_1.$$

Next we prove that for any two message m_0 and m_1

$$\mathbf{SD}(\mathsf{ConTamper}'_{\mathcal{A},T,m_0}, \mathsf{ConTamper}'_{\mathcal{A},T,m_1}) \leq 2\varepsilon_2. \tag{5.1}$$

Once we have Line 5.1 we can conclude the proof using the following reasoning. For any two message m_0 and m_1 we have,

$$\begin{aligned}
&\mathbf{SD}(\mathsf{ConTamper}_{\mathcal{A},T,m_0}, \mathsf{ConTamper}_{\mathcal{A},T,m_1}) \leq \\
&\mathbf{SD}\left(\mathsf{ConTamper}_{\mathcal{A},T,m_0}, \mathsf{ConTamper}'_{\mathcal{A},T,m_0}\right) + \mathbf{SD}\left(\mathsf{ConTamper}'_{\mathcal{A},T,m_0}, \mathsf{ConTamper}'_{\mathcal{A},T,m_1}\right) \\
&\qquad + \mathbf{SD}\left(\mathsf{ConTamper}'_{\mathcal{A},T,m_1}, \mathsf{ConTamper}_{\mathcal{A},T,m_1}\right) \leq \\
&\qquad = 2T \cdot \varepsilon_1 + 2\varepsilon_2.
\end{aligned}$$

To prove Line 5.1, we use the repeated leakage resilience property of the inner coding scheme. We can think of the two experiments $\mathsf{ConTamper}'_{\mathcal{A},T,m_0}$ and $\mathsf{ConTamper}'_{\mathcal{A},T,m_1}$ as providing repeated leakage on the inner codeword c_{in} which is either an inner encoding of m_0 or an inner encoding of m_1. The leakage functions used on c_{in} for each tampering function f is L_f which comes from the leakage family $\mathcal{L}_{\mathcal{F}}$, and we associate with each such function a unique "repeat value" repeat_f. The way that the repeat value and the experiment are defined ensures that the experiment stops whenever the outcome is not repeat_f.

Now consider the sequence of tampering functions f_i that correspond to how $\mathcal{A}$ would act if he received repeat_{f_i} after each tampering step i: $\{f_1, f_2, ..., f_T\}$. Let L_{f_i} be the corresponding leakage function for f_i with the unique repeat value repeat_{f_i} as define in the theorem statement. And let $\bar{\mathsf{L}} = (\mathsf{L}_{f_1}, \mathsf{L}_{f_2}, \ldots, \mathsf{L}_{f_T})$. The view of the adversary $\mathcal{A}$ in experiment $\mathsf{ConTamper}'_{\mathcal{A},T,m}$ is completely determined by the outcome of $\mathsf{RLeakage}(\bar{\mathsf{L}}, \mathtt{lrEnc}(m))$. Therefore:

$$\begin{aligned}
&\mathbf{SD}\left(\mathsf{ConTamper}'_{\mathcal{A},T,m_0}, \mathsf{ConTamper}'_{\mathcal{A},T,m_1}\right) \\
&\leq \mathbf{SD}\left(\mathsf{RLeakage}(\bar{\mathsf{L}}, \mathtt{lrEnc}(m_0)), \mathsf{RLeakage}(\bar{\mathsf{L}}, \mathtt{lrEnc}(m_1))\right) \\
&\leq \mathbf{SD}\left(\mathsf{RLeakage}(\bar{\mathsf{L}}, \mathtt{lrEnc}(m_0)), \mathsf{RLeakage}(\bar{\mathsf{L}}, U_{k'})\right) + \mathbf{SD}\left(\mathsf{RLeakage}(\bar{\mathsf{L}}, \mathtt{lrEnc}(m_0)), \mathsf{RLeakage}(\bar{\mathsf{L}}, U_{k'})\right) \quad \leq 2\varepsilon_2
\end{aligned}$$

□

5.3 Construction Results and Parameters

We now summarize what we get when we instantiate the above construction with the code $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ from Figure 3.1. We characterize the type of function families for which we achieve the various cases of continuous non-malleable security in terms of whether the family has "few fixed points" and "high entropy".

Corollary 5.6. *Let $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ be the (k,n)-coding scheme construction in Figure 3.1 where h_1 is chosen from a t_1-wise independent hash family $\mathcal{H}_1$, and h_2 is chosen from a t_2-wise independent hash family $\mathcal{H}_2$.*

Let $\mathcal{F}$ be a family of φ-few fix points, μ-entropy functions and assume that $|\mathcal{F}| \geq \max\left\{2^k, 2^\lambda\right\}$. Then for any $T \leq 2^\lambda$ the code $(\mathsf{Enc}_{h_1,h_2}, \mathsf{Dec}_{h_1,h_2})$ is an $(\mathcal{F}, T, T \cdot 2^{-\lambda})$-CNMC[$\mathsf{sd}, \mathsf{prs}$] with probability $1 - 2^{-\lambda}$ over the choice of h_1 and h_2 with the following parameters for each of the four options, where $\mathsf{sd} = 1$ *indicates self-destruct and* $\mathsf{prs} = 1$ *indicates persistent tampering:*

[$\mathsf{sd}, \mathsf{prs}$]	$\varphi \leq$	$\mu \geq$	$t_1 =$	$t_2 =$	$v_1 + v_2 \geq$
[0 , 0]	$2^{-(\lambda+O(1))}$	$\log\log\lvert\mathcal{F}\rvert + 2\lambda + O(1)$	$O(\log\lvert F\rvert)$	$O(\log\lvert F\rvert)$	$\log\log\lvert\mathcal{F}\rvert + O(\lambda)$
[0 , 1]	$2^{-(\lambda+O(1))}$	No restriction	$O(T \cdot \log\lvert\mathcal{F}\rvert)$	$O(\log\lvert F\rvert)$	$\log\log\lvert\mathcal{F}\rvert + O(\lambda)$
[1 , 0]	No restriction	$\log\log\lvert\mathcal{F}\rvert + 2\lambda + O(1)$	$O(T \cdot \log\lvert\mathcal{F}\rvert)$	$O(\log\lvert F\rvert)$	$\log\log\lvert\mathcal{F}\rvert + O(\lambda)$
[1 , 1]	No restriction	No restriction	$O(T \cdot \log\lvert\mathcal{F}\rvert)$	$O(\log\lvert F\rvert)$	$\log\log\lvert\mathcal{F}\rvert + O(\lambda)$

All results in the table also hold if $\mathcal{F}$ includes the always identity function $f_{\mathsf{id}}(x) = x$ or the always constant functions $\{f_c(x) = c \ : \ c \in \{0,1\}^n\}$.

If Conjecture 5.4 holds then we can set $t_1 = O(\lambda \cdot \log|\mathcal{F}|)$ for the last three rows of the table.

Proof. Theorem 5.5 tells us how to build continuous non-malleable codes using predictable codes and repeated-leakage resilient codes. Corollary 3.9 gives us the parameters for predictable codes and the type of predictability they achieve. Furthermore, we can always add the identity and constant functions f to any family and maintain predictability with $|\mathcal{P}(f)| = 1$ for these functions. Theorem 5.3 gives us the parameters for repeated-leakage-resilient codes. By plugging in the parameters, the corollary follows. □

We explore the consequences of the above corollary in each of the four settings.

No Self-Destruct, Non-Persistent Tampering (Strongest). We begin with the strongest setting, where we assume no self-destruct and non-persistent tampering. In this case, an $(\mathcal{F}, \varepsilon)$-tamper-detection code is also an $(\mathcal{F}, \mathcal{P}, \varepsilon)$-predictable code with $\mathcal{P}(f) = \perp$ for each $f \in \mathcal{F}$, and hence it is also an $(\mathcal{F}, T, 2T \cdot \varepsilon)$-continuous non-malleable code in this setting. In particular, we get such codes for families of functions with high entropy and few fixed points. However, we can also add the "always identity" function $f_{\mathsf{id}}(x) = x$ or the "always constant" functions $\{f_c(x) = c \ : \ c \in \{0,1\}^n\}$ to $\mathcal{F}$ and maintain "predictability" and therefore also continuous non-malleable security in this setting.

As an example, we can achieve continuous non-malleable codes in this setting for the function families discusses in Section 3 in the context of tamper-detection. By adding in the "always identity" and the "always constant" functions to these families we get such continuous non-malleable codes for:

- The family $\mathcal{F}_{\mathsf{poly},d}$ consisting of *all* polynomials $p(x)$ of degree d over the field $\mathbb{F}_{2^n}$. For inefficient codes, we can set the degree as high as $d = 2^{\alpha n}$, for constant $\alpha < \frac{1}{2}$ and get rate $(1 - \alpha)$. For efficient codes (in the CRS model), the degree must be set to some polynomial $d = d(n)$ and the rate approaches 1.
- The family $\mathcal{F}_{\mathsf{affine},r}$ consisting of *all* affine tampering functions when we identify $\{0,1\}^n$ as the vector space $\mathbb{F}_{2^r}^m$ with $m = n/r$: i.e., all functions $f_{A,b}(x) = Ax + b$ where $A \in \mathbb{F}_{2^r}^{m \times m}, b \in \mathbb{F}_{2^r}^m$. Such codes are efficient (in the CRS model) and their security is $2^{-r+O(1)}$.

The number of tampering attacks T that such codes protect against can even be set to exponential in the security parameter $T = 2^\lambda$ without hurting efficiency.

Self Destruct, Persistent Tampering (Weakest). In the weakest setting, where we assume self-destruct and persistent tampering, we show the existence of (inefficient) continuous non-malleable codes which essentially matches the known results for standard non-malleable codes. In particular, for a function family $\mathcal{F}$ of size $|\mathcal{F}| = 2^{2^{\alpha n}}$ we can protect against even an exponential number $T = 2^\lambda$ tampering attempts with security $\varepsilon = 2^{-\lambda}$ and get a rate which approaches $(1 - \alpha)$. For example, we show the existence of such (inefficient) codes in the *split-state model* where the n-bit codeword is divided into two halves and the attacker can tamper each half of the codeword arbitrarily but independently of the other half.

Furthermore, if we take a function family of size $|\mathcal{F}| = 2^s$ for some polynomial s and want to protect against some polynomial T number of tampering attempts, then we get efficient constructions (in the CRS model) where the efficiency is $\mathsf{poly}(s, T, \lambda)$. For example, the family $\mathcal{F}$ could be all circuits of size s. The rate of such codes approaches 1. We know that the dependence between the efficiency of the code and s is necessary - for example, if the tampering functions are all circuits of size s then we know that the circuit-size of the code must exceed $O(s)$ as otherwise the tampering function could simply decode, flip the last bit, and re-encode. However, we do not know if the dependence between the efficiency of the code and T is necessary. This dependence comes from the parameters of repeated leakage-resilient codes and, if we could improve the parameters there as conjectured (see conjecture 5.4) we would get security for up to $T = 2^\lambda$ tampering attempts with a code having efficiency $\mathsf{poly}(s, \lambda)$.

Intermediate Cases. The two intermediate cases lie between the strongest and the weakest setting and correspond to either (I) having persistent tampering but no self-destruct or (II) having self-destruct but non-persistent tampering. These cases are incomparable to each other.

In case (I) we can allow function families with arbitrarily low-entropy but we need to require that the function only have few fixed points. The few-fixed points requirement is necessary in this setting as highlighted by the following attack. Consider the family $\mathcal{F}_{\mathsf{nextbit}}$ consisting of n functions $\{f_i(x) \ : \ i \in [n]\}$ where $f_i(x)$ looks at the i'th bit of x and if it a 0 then keeps x as is and otherwise it flips the $(i+1)$'st bit (if $i = n$ then the first bit) of x. We also add the identity function to the family. This family has full entropy $\mu = n$ but the functions (even the non-identity ones) have many fixed points $\varphi = 1/2$. The attacker does the following for $i = 1, \ldots, n$: he applies the function f_i and sees whether he gets back same. If so, he learns the i'th bit of the original codeword is a 0 and otherwise he learns that it was a 1. Either way the attacker applies f_i again to set the codeword back to the original. This is a legal attacker since if we compose the functions chosen by the attacker we get $f_i \circ \cdots \circ f_1 \circ f_1$ which is either just f_i or identity and therefore in the family $\mathcal{F}_{\mathsf{nextbit}}$. After n iterations the attacker completely learns the codeword and therefore also the encoded message. Notice that this attack crucially relies on the fact that there is no self-destruct, since each tampering attempt could result in $\perp$.

In case (II) we can allow function families with many fixed points but require entropy. Intuitively, the high entropy requirement is necessary in this setting to prevent the following attack. Consider the family $\mathcal{F}_{2const}$ consisting of $n2^{2n}$ functions $\{f_{i,c_0,c_1}(x) \; : \; c_0, c_1 \in \{0,1\}^n, i \in [n]\}$ which output c_0 if the ith bit of x is a 0 and c_1 if the ith bit of x is a 1. This family has $\varphi = 2^{-n+1}$ few fixed points but has low entropy $\mu = 1$. Given any code $(\mathsf{Enc}, \mathsf{Dec})$ the attacker simply chooses any two distinct valid codewords $c_0 \neq c_1$, $\mathsf{Dec}(c_0), \mathsf{Dec}(c_1) \neq \perp$. For $i = 1, \ldots, n$: he applies the tampering function f_{i,c_0,c_1} and depending on whether he gets back c_0 or c_1, he learns the i'th bit of the original codeword c. After n iterations the attacker completely learns the codeword. Notice that this attack crucially relies on the fact that tampering is non-persistent, since each tampering attempt completely overwrites the codeword but the next tampering attempt still tampers the original codeword.

The parameters of our continuous non-malleable codes in these settings matches that of the weakest case. In particular, for a function family $\mathcal{F}$ of size $|\mathcal{F}| = 2^{2^{\alpha n}}$ satisfying the appropriate entropy and fixed-point requirements, we can inefficiently protect against even an exponential number $T = 2^\lambda$ tampering attempts with security $\varepsilon = 2^{-\lambda}$ and get a rate which approaches $(1 - \alpha)$. Furthermore, if we take a function family of size $|\mathcal{F}| = 2^s$ for some polynomial s and want to protect against some polynomial T number of tampering attempts, then we get efficient constructions (in the CRS model) where the efficiency is $\mathsf{poly}(s, T, \lambda)$. Under conjecture 5.4, we would get security for up to $T = 2^\lambda$ tampering attempts with a code having efficiency $\mathsf{poly}(s, \lambda)$.

6 Conclusion

In this paper, we introduced several new notions of codes that offer protection against tampering attacks. Most importantly, we defined a general notion of tamper-detection codes and various flavors of continuous non-malleable codes and explored the question of what families of functions admit such codes. Although some of our constructions can be made efficient, all of the constructions are Monte-Carlo constructions, which can also be interpreted as constructions in the CRS model. It remains an open problem to construct explicit and efficient codes (without a CRS) for interesting families, such as low-degree polynomials or affine functions (etc.).

References

[ABPP14] Michel Abdalla, Fabrice Benhamouda, Alain Passelègue, and Kenneth G. Paterson. Related-key security for pseudorandom functions beyond the linear barrier. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 77–94. Springer, 2014.

[ADL13] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:81, 2013.

[AGM$^+$14] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations and perturbations. Cryptology ePrint Archive, Report 2014/841, 2014. `http://eprint.iacr.org/`.

[AHI11] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *ICS*, pages 45–60, 2011.

[BC10] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, pages 666–684, 2010.

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, pages 486–503, 2011.

[BDL01] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *J. Cryptology*, 14(2):101–119, 2001.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT*, pages 491–506, 2003.

[BPT12] Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In *ASIACRYPT*, pages 331–348, 2012.

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *FOCS*, pages 276–287. IEEE Computer Society, 1994.

[CDF$^+$08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 471–488. Springer, 2008.

[CG13a] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:118, 2013.

[CG13b] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. *IACR Cryptology ePrint Archive*, 2013:565, 2013.

[CKM11] Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. BiTR: Built-in tamper resilience. In *ASIACRYPT*, pages 740–758, 2011.

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:102, 2014.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2010.

[DFMV13] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT (2)*, pages 140–160, 2013.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *CRYPTO (2)*, pages 239–257, 2013.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, 2014. To appear.

[FMVW14] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *EUROCRYPT*, 2014. To appear. Available: http://eprint.iacr.org/2013/702.

[FPV11] Sebastian Faust, Krzysztof Pietrzak, and Daniele Venturi. Tamper-proof circuits: How to trade leakage for tamper-resilience. In *ICALP (1)*, pages 391–402, 2011.

[GIP$^+$14] Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 495–504. ACM, 2014.

[GLM$^+$04] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In *TCC*, pages 182–200, 2011.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In *EUROCRYPT*, pages 308–327, 2006.

[KKS11] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *CRYPTO*, pages 517–532, 2012.

[Pie12] Krzysztof Pietrzak. Subspace LWE. In *TCC*, pages 548–563, 2012.

[Sha49] Claude E. Shannon. A theorem on coloring the lines of a network. *Journal of mathematics and physics / Massachusetts Institute of Technology*, 28:148–151, 1949.

[Wee12] Hoeteck Wee. Public key encryption against related key attacks. In *Public Key Cryptography*, pages 262–279, 2012.