Non-malleable Codes from Two-Source Extractors*

Stefan Dziembowski[1], Tomasz Kazana[2], and Maciej Obremski[2]

[1] University of Warsaw and Sapienza University of Rome
[2] University of Warsaw

Abstract. We construct an efficient information-theoretically non-malleable code in the split-state model for one-bit messages. Non-malleable codes were introduced recently by Dziembowski, Pietrzak and Wichs (ICS 2010), as a general tool for storing messages securely on hardware that can be subject to tampering attacks. Informally, a code $(\mathsf{Enc} : \mathcal{M} \to \mathcal{L} \times \mathcal{R}, \mathsf{Dec} : \mathcal{L} \times \mathcal{R} \to \mathcal{M})$ is *non-malleable in the split-state model* if any adversary, by manipulating *independently* L and R (where (L, R) is an encoding of some message M), cannot obtain an encoding of a message M' that is not equal to M but is "related" M in some way. Until now it was unknown how to construct an information-theoretically secure code with such a property, even for $\mathcal{M} = \{0, 1\}$. Our construction solves this problem. Additionally, it is leakage-resilient, and the amount of leakage that we can tolerate can be an arbitrary fraction $\xi < 1/4$ of the length of the codeword. Our code is based on the inner-product two-source extractor, but in general it can be instantiated by any two-source extractor that has large output and has the property of being *flexible*, which is a new notion that we define.

We also show that the non-malleable codes for one-bit messages have an equivalent, perhaps simpler characterization, namely such codes can be defined as follows: if M is chosen uniformly from $\{0, 1\}$ then the probability (in the experiment described above) that the output message M' is not equal to M can be at most $1/2 + \epsilon$.

1 Introduction

Real-life attacks on cryptographic devices often do not break their mathematical foundations, but exploit vulnerabilities in their implementations. Such "physical attacks" are usually based on passive measurements such as running-time, electromagnetic radiation, power consumption (see e.g. [24]), or active tampering where the adversary maliciously modifies some part of the device (see e.g. [3]) in order to force it to reveal information about its secrets. A recent trend in theoretical cryptography, initiated by [34,31,30] is to design cryptographic schemes

* This work was partly supported by the WELCOME/2010-4/2 grant founded within the framework of the EU Innovative Economy (National Cohesion Strategy) Operational Programme. The European Research Council has provided financial support for this work under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no CNTM-207908.

R. Canetti and J.A. Garay (Eds.): CRYPTO 2013, Part II, LNCS 8043, pp. 239–257, 2013.

that already on the abstract level guarantee that they are secure even if implemented on devices that may be subject to such physical attacks. Contrary to the approach taken by the practitioners, security of these constructions is always analyzed formally in a well-defined mathematical model, and hence covers a broad class of attacks, including those that are not yet known, but may potentially be invented in the future. Over the last few years several models for passive and active physical attacks have been proposed and schemes secure in these models have been constructed (see e.g. [31,30,22,2,35,7,15,25]). In the passive case the proposed models seem to be very broad and correspond to large classes of real-life attacks. Moreover, several constructions secure in these models are known (including even general compliers [27] for any cryptographic functionality). The situation in the case of active attacks is much less satisfactory, usually because the proposed models include an assumption that some part of the device is tamper-proof (e.g. [26]) or because the tampering attacks that they consider are very limited (e.g. [30] or [13] consider only probing attacks, and in [37] the tampering functions are assumed to be as linear). Hence, providing realistic models for tampering attacks, and constructing schemes secure in these models is an interesting research direction.

In a recent paper [23] the authors consider a very basic question of storing messages securely on devices that may be subject to tampering. To this end they introduce a new primitive that they call the *non-malleable codes*. The motivating scenario for this concept is as follows. Imagine we have a secret message $m \in \mathcal{M}$ and we want to store it securely on some hardware $\mathcal{D}$ that may be subject to the tampering attacks. In order to increase the security, we will encode the message m by some (randomized) function Enc and store the codeword $x := \mathsf{Enc}(m)$ on $\mathcal{D}$. Since we later want to recover m from $\mathcal{D}$ we obviously also need a decoding function $\mathsf{Dec} : \mathcal{X} \to \mathcal{M} \cup \{\perp\}$ such that for every $m \in \mathcal{M}$ we have $\mathsf{Dec}(\mathsf{Enc}(m)) = m$. Now, suppose the adversary can tamper with the device in some way, which we model by allowing him to choose a function $F : \mathcal{X} \to \mathcal{X}$, from some fixed set $\mathcal{F}$ of *tampering functions* and substitute the contents of $\mathcal{D}$ by $F(x)$. Let $m' := \mathsf{Dec}(F(\mathsf{Enc}(m)))$ be the result of decoding such modified codeword.

Let us now think what kind of security properties one could expect from such an encoding scheme. Optimistically, e.g., one could hope to achieve tamper-detection by which we would mean that $m' = \perp$ if $F(x) \neq x$. Unfortunately this is usually unachievable, as, e.g., if the adversary chooses F to be a constant function equal to $\mathsf{Enc}(\tilde{m})$ then $m' = \tilde{m}$. Hence, even for very restricted classes $\mathcal{F}$ (containing only the constant functions), the adversary can force m' to be equal to some message of his choice. Therefore, if one hopes to get any meaningful security notion, one should weaken the tamper-detection requirement.

In [23] the authors propose such a weakening based on the concept of *non-malleability* introduced in the seminal paper of Dolev et al. [19]. Informally, we say that a code $(\mathsf{Enc}, \mathsf{Dec})$ is *non-malleable* if either (1) the decoded message m' is equal to m, or (2) the decoded message m' is "independent" from m. The formal definition appears in Section 3, and for an informal discussion of this concept the reader may consult [23]. As argued in [23] the non-malleable codes

can have vast applications to tamper-resistant cryptography. We will not discuss them in detail here, but let us mention just one example, that looks particularly appealing to us. A common practical way of breaking cryptosystems is based on the so-called related-key attacks (see, e.g. [5,4]), where the adversary that attacks some device $\mathcal{D}(K)$ (where K is the secret key) can get access to an identical device containing a *related* key $K' = F(K)$ (by for example tampering with K). Non-malleable codes provide an attractive solution to this problem. If $(\mathsf{Enc}, \mathsf{Dec})$ is a non-malleable code secure with respect to same family $\mathcal{F}$, then we can store the key K on $\mathcal{D}$ in an encoded form, and prevent the related key attacks as long as the "relation F" is in $\mathcal{F}$. This is because, the only thing that the adversary can achieve by applying F to $\mathsf{Enc}(K)$ is to produce encoding of either a completely unrelated key K', or to keep $K' = K$. It is clear that both cases do not help him in attacking $\mathcal{D}(K)$.

It is relatively easy to see that if the family $\mathcal{F}$ of tampering functions is equal to the entire space of functions from $\mathcal{X}$ to $\mathcal{X}$ then it is impossible to construct such a non-malleable code secure against $\mathcal{F}$. This is because in this case the adversary can always choose $F(x) = \mathsf{Enc}(H(\mathsf{Dec}(x)))$ for any function $H : \mathcal{M} \to \mathcal{M}$, which yields $m' = \mathsf{Dec}(x) = \mathsf{Dec}(\mathsf{Enc}(H(\mathsf{Dec}(\mathsf{Enc}(m))))) = H(m)$, and therefore he can relate m' to m in an arbitrary way. Therefore non-malleable codes can exist only with respect to restricted classes $\mathcal{F}$ of functions. The authors of [23] propose some classes like this and provide constructions of non-malleable codes secure with respect to them. One example is the class of bit-wise tampering functions, which tamper with every bit of x "independently", more precisely: the ith bit x'_i of x' is a function of x_i, and does not depend on any x_j for $j \neq i$. This is a very strong assumption and it would be desirable to weaken it. One natural idea for such weakening would be to allow x'_i to depend on the bits of x from positions on some larger subset $\mathcal{I}_i \subsetneq \{1, \ldots, |x|\}$. Observe that $\mathcal{I}$ always needs to be a proper subset of $\{1, \ldots, |x|\}$, as, for the reasons described above, allowing x_i to depend on entire x would render impossible any secure construction. It is of course not clear what would be the right "natural" subsets $\mathcal{S}_i$ that one could use here. The authors of [23] solve this problem in the following simple way. They assume that the codeword consists of two parts (usually of equal size), i.e.: $x = (L, R) \in \mathcal{L} \times \mathcal{R}$, and the adversary can tamper in an arbitrary way with both parts, i.e., $\mathcal{F}$ consists of *all* functions $\mathsf{Mall}^{f,g}$ that can be defined as $\mathsf{Mall}^{f,g}(L, R) = (f(L), g(R))$ (for some $f : \mathcal{L} \to \mathcal{L}$ and $g : \mathcal{R} \to \mathcal{R}$). In practical applications this corresponds to a scenario in which L and R are stored on two separate memory parts that can be tampered independently. A similar model has been used before in the context of leakages and is called a *split-state model* [22,14,28,16]. The authors of [23] show existence of non-malleable codes secure in this model in a non-constructive way (via the probabilistic argument). They also provide a construction of such codes in a random oracle model, and leave constructing explicit information-theoretically secure codes as an open problem. A very interesting partial solution to this problem came recently from Liu and Lysyanskaya [33] who constructed such codes with computationally-security, assuming a common reference string. Their construction comes with an additional feature of being leakage-resilient, i.e.

they allow the adversary to obtain some partial information about the codeword via memory leakage (the amount of leakage that they can tolerate is a $\frac{1}{2} - o(1)$ fraction of the length of the codeword). However, constructing the information-theoretically secure nonmalleable codes in this model remained an open problem, even if messages are of length 1 only (i.e. $\mathcal{M} = \{0, 1\}$).

Our Contribution. We show a construction of efficient information-theoretically secure non-malleable codes in the split-state model for $\mathcal{M} = \{0, 1\}$. Additionally to being non-malleable, our code is also leakage-resilient and the amount of leakage that we can tolerate is an arbitrary constant $\xi < \frac{1}{4}$ of the length of the codeword (cf. Thm. 2). Our construction is fairly simple. The codeword is divided into two parts, L and R, which are vectors from a linear space $\mathbb{F}^n$, where $\mathbb{F}$ is a field of exponential size (and hence $\log |\mathbb{F}|$ is linear). Essentially, to encode a bit $B = 0$ one chooses at a random pair $(L, R) \in \mathbb{F}^n \times \mathbb{F}^n$ of orthogonal vectors (i.e. such that $\langle L, R \rangle = 0$), and to encode $B = 1$ one chooses a random pair of non-orthogonal vectors (clearly both encoding and decoding can be done very efficiently in such a code). Perhaps surprisingly, the assumption that $\mathbb{F}$ is large is important, as our construction is *not* secure for small $\mathbb{F}$'s. An interesting consequence is that our code is "non-balanced", in the sense that a random element of the codeword space with an overwhelming probability encodes 1. We actually use this property in the proof.

Our proof also very strongly relies on the fact that the inner product over finite field is a two-source extractor (cf. Sect. 2). We actually show that in general a split-state non-malleable code for one-bit messages can be constructed from any two source-extractor with sufficiently strong parameters (we call such extractors *flexible*, cf. Sect. 2).

We also provide a simple argument that shows that our scheme is secure against affine mauling functions (that look at the entire codeword, hence *not* in the split-state model).

Typically in information-theoretic cryptography solving a certain task for one-bit messages automatically gives a solution for multi-bit messages. Unfortunately, it is not the case for the non-malleable codes. Consider for example a naive idea of encoding n bits "in parallel" using the one bit encoding function Enc, i.e. letting $\mathsf{Enc}'(m_1, \ldots, m_n) := ((L_1, \ldots, L_n), (R_1, \ldots, R_n))$, where each $(L_i, R_i) = \mathsf{Enc}(m_i)$. This encoding is obviously malleable, as the adversary can, e.g., permute the bits of m by permuting (in the same way) the blocks $L_1, \ldots, L_n$ and $R_1, \ldots, R_n$. Nevertheless we believe that our solution is an important step forward, as it may be useful as a building blocks for other, more advanced constructions, like, e.g., tamper-resilient generic compilers (in the spirit of [31,30,13,20,27]). This research direction looks especially promising since many of the leakage-resilient compliers (e.g. [20,27]) are based on the same inner-product extractor.

We also show that for one-bit messages non-malleable codes can be defined in an alternative, and perhaps simpler way. Namely we show (cf. Lemma 2) that any code $(\mathsf{Enc}, \mathsf{Dec})$ (not necessarily defined in the split-state model) in non-malleable with respect to some family $\mathcal{F}$ of functions if and only if "it is hard to

negate the encoded bit B with functions from $\mathcal{F}$", by which we mean that for a bit B chosen *uniformly* from $\{0,1\}$ any $F \in \mathcal{F}$ we have that

$$P\left[\mathsf{Dec}(F(\mathsf{Enc}(B))) \neq B\right] \leq \frac{1}{2}. \tag{1}$$

(the actual lemma that we prove involves also some small error parameter ϵ both in the non-malleability definition and in (1), but for the purpose of this informal discussion let us omit them). Therefore, the problem of constructing non-malleable bit encoding in the split state model can be translated to a much simpler and perhaps more natural question: can one encode a random bit B as (L, R) in such a way that independent manipulation of L and R produces an encoding (L', R') of $\overline{B}$ with probability at most $1/2$? Observe that, of course, it is easy to negate a random bit with probability exactly $1/2$, by deterministically setting (L', R') to be an encoding of a fixed bit, 0, say. Informally speaking, $(\mathsf{Enc}, \mathsf{Dec})$ is non-malleable if this is the best that the adversary can achieve.

In the full version of this paper [21] we analyze the general relationship between the two-source extractors and the non-malleable codes in the split state model pointing out some important differences. We also compare the notion of the non-malleable codes with the *leakage-resilient storage* [14] also showing that they are fundamentally different.

Related and Subsequent Work. Some of the related work was already described in the introduction. There is no space here to mention all papers that propose theoretical countermeasures against tampering. This research was initiated by Ishai et al. [30,26]. Security against both tampering and leakage attacks were also recently considered in [32]. Unlike us, they construct concrete cryptosystems (not encoding schemes) secure against such attacks. Another difference is that their schemes are computationally secure, while in this work we are interested in the information-theoretical security.

The notion of non-malleability (introduced in [19]) is used in cryptography in several contexts. In recent years it was also analyzed in the context of randomness extractors, starting from the work of Dodis and Wichs [18] on non-malleable extractors (see also [17,12]). Informally speaking an extractor ext is non-malleable if its output $\mathsf{ext}(S, X)$ is (almost) uniform even if one knows the value $\mathsf{ext}(F(S), X)$ for some "related" seed $F(S)$ (such that $F(S) \neq S$). Unfortunately, it does not look like this primitive can be used to construct the non-malleable codes in the split-state model, as this definition does not capture the situation when X is also modified.

Constructions of non-malleable codes secure in different (not split-state) models were recently proposed in [8,9,10].

Recently, Aggarwal, Dodis and Lovett [1] solved the main open problem left in this paper, by showing a non-malleable code that works for messages of arbitrary length. This exciting result is achieved by combining the inner-product based encoding with sophisticated methods from the additive combinatorics.

Acknowledgments. We are very grateful to Divesh Aggarwal and to the anonymous CRYPTO reviewer for pointing out errors in the proof of Lemma 3 in the

previous versions of this paper. We also thank Yevgeniy Dodis, Konrad Durnoga and Karol Cwalina for helpful discussions.

2 Preliminaries

If $\mathcal{Z}$ is a set then $Z \leftarrow \mathcal{Z}$ will denote a random variable sampled uniformly from $\mathcal{Z}$. We start with some standard definitions and lemmas about the statistical distance. Recall that if A and B are random variables over the same set $\mathcal{A}$ then the *statistical distance between A and B* is denoted as $\Delta(A;B)$, and defined as $\Delta(A;B) = \frac{1}{2}\sum_{a\in\mathcal{A}} |P[A=a] - P[B=a]|$. If the variables A and B are such that $\Delta(A,B) \leq \epsilon$ then we say that A is ϵ-close to B, and write $A \approx_\epsilon B$. If $\mathcal{X}, \mathcal{Y}$ are some events then by $\Delta(A|\mathcal{X}\ ;\ B|\mathcal{Y})$ we will mean the distance between variables A' and B', distributed according to the conditional distributions $P_{A|\mathcal{X}}$ and $P_{B|\mathcal{Y}}$.

If B is a uniform distribution over $\mathcal{A}$ then $d(A|\mathcal{X}) := \Delta(A|\mathcal{X};B)$ is called *statistical distance of A from uniform given the event $\mathcal{X}$*. If moreover C is independent from B then $d(A|C) := \Delta((A,C);(B,C))$ is called *statistical distance of A from uniform given the variable C*. More generally, if $\mathcal{X}$ is an event then $d(A|C,\mathcal{X}) := \Delta((A,C)|\mathcal{X};(B,C)|\mathcal{X})$. It is easy to see that $d(A|C)$ is equal to $\sum_c P[C=c]\cdot d(A|C=c)$.

Extractors. As described in the introduction, the main building block of our construction is a two-source randomness extractor based on the inner product over finite fields. The two source extractors were introduced (implicitly) by Chor and Goldreich [11], who also showed that the inner product over Z_2 is a two-source extractor. The generalization to any field is shown in [36].

Our main theorem (Thm. 1) does not use any special properties of the inner product (like, e.g., the linearity), besides of the fact that it extracts randomness, and hence it will be stated in a general form, without assuming that the underlying extractor is necessarily an inner product. The properties that we need from our two-source extractor are slightly non-standard. Recall that a typical way to define a strong two-source extractor[1] (cf. e.g. [36]) is to require that $d(\mathsf{ext}(L,R)|L)$ and $d(\mathsf{ext}(L,R)|R)$ are close to uniform, provided that L and R have min-entropy at least m (for some parameter m). For the reasons that we explain below, we need a slightly stronger notion, that we call *flexible* extractors. Essentially, instead of requiring that $\mathbf{H}_\infty(L) \geq m$ and $\mathbf{H}_\infty(R) \geq m$ we will require only that $\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) \geq k$ (for some k). Note that if $k = 2m$ then this requirement is obviously weaker than the standard once, and hence the flexibility strengthens the standard definition.

Formally, let $\mathcal{L}, \mathcal{R}$ and $\mathcal{C}$ be some finite sets. A function $\mathsf{ext} : \mathcal{L}\times\mathcal{R} \to \mathcal{C}$ is a *strong flexible (k,ϵ)-two source extractor* if for every $L \in \mathcal{L}$ and $R \in \mathcal{R}$ such that $\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) \geq k$ we have that $d(\mathsf{ext}(L,R)|L) \leq \epsilon$ and $d(\mathsf{ext}(L,R)|R) \leq \epsilon$. Since we are not going to use any weaker version of this notion we will often

[1] Recall also that a random variable A has *min-entropy k*, denoted $\mathbf{H}_\infty(A) = k$ if $k = \min_a(-\log P[A=a])$.

simply call such extractors "flexible" without explicitly stating that they are strong. As it turns out the inner product over finite fields is such an extractor.

Lemma 1. *For every finite field* $\mathbb{F}$ *and any* n *we have that* $\mathsf{ext} : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ *defined as* $\mathsf{ext}^n_{\mathbb{F}}(L, R) = \langle L, R \rangle$ *is a strong flexible* (k, ϵ)*-extractor for any* k *and* ϵ *such that*

$$\log(1/\epsilon) = \frac{k - (n+4)\log|\mathbb{F}|}{3} - 1. \tag{2}$$

Although this lemma appears to be folklore, at least in case of the "weak" flexible extractors (i.e. when we require only that $d(\mathsf{ext}(L, R)) \leq \epsilon$), we were not able to find it in the literature for the *strong* flexible extractors. Therefore for completeness in the full version of this paper [21] we provide a proof of it (which is straightforward adaptation of the proof of Theorem 3.1 in [36]).

Note that since ϵ can be at most 1, hence (2) makes sense only if $k \geq 6 + 4\,|\mathbb{F}| + n \log|\mathbb{F}|$. It is easy to see that it cannot be improved significantly, as in any flexible (k, ϵ)-extractor $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ we need to have $k > \max(\log|\mathcal{L}|, \log|\mathcal{R}|)$. To see why it is the case, suppose we have such a flexible (k, ϵ)-extractor ext for $k = \log|\mathcal{L}|$ (the case $k = \log|\mathcal{R}|$ is obviously symmetric). Now let L' be a random variable uniformly distributed over $\mathcal{L}$ and let $R' \in \mathcal{R}$ be constant. Then obviously $\mathbf{H}_\infty(L') + \mathbf{H}_\infty(R') = \log|\mathcal{L}| + 0 = k$, but $\mathsf{ext}(L', R')$ is a deterministic function of L', and hence $d(\mathsf{ext}(L', R')|L')$ is large. Therefore, in terms of the entropy threshold k, the inner product is optimal in the class of flexible extractors (up to a small additive constant). Note that this is in contrast with the situation with the "standard" two-source extractors where a better extractor is known [6].

The reason why we need the "flexibility" property is as follows. In the proof of Lemma 3 we will actually use in two different ways the fact that ext is an extractor. In one case (in the proof of Claim 2 within the proof of Lemma 3) we will use it in the "standard" way, i.e. we will apply it to two independent random variables with high min-entropy. In the other case (proof of Claim 1) we will use the fact that $d(\mathsf{ext}(L, R)|R) \leq \epsilon$ even if L has relatively low min-entropy ($\mathbf{H}_\infty(L) = k - |R|$) while R is completely uniform (and hence $\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) = k$).[2] Hence we will treat ext as standard seeded extractor. It should not be surprising that we can use the inner product in this way, as it is easy to see that the inner product is a universal hash function, and hence the fact that it is a seeded strong extractor follows from the leftover hash lemma [29]. Hence Lemma 1 in some sense "packs" these two properties of the inner product into one simple statement.

The observation that the inner product extractor is flexible allows us as also to talk about the sum of leakages in Section 5, instead of considering bounded leakage from L and R separately (as it is done, e.g., in [14]). We would like to stress that this is actually not the main reason for introducing the "flexibility" property, as it would be needed even if one does not incorporate leakages into the model.

[2] We will also use a symmetric fact for $d(\mathsf{ext}(L, R)|L)$.

3 Non-malleable Codes and the Hardness of Negation

In this section we review the definition of the non-malleable codes from [23], which has already been discussed informally in the introduction. Formally, let $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X}, \mathsf{Dec} : \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ be an encoding scheme. For $F : \mathcal{X} \to \mathcal{X}$ and for any $m \in \mathcal{M}$ define the experiment Tamper_m^F as:

$$\mathsf{Tamper}_m^F = \left\{ \begin{array}{c} X \leftarrow \mathsf{Enc}(m), \\ X' := F(X), \\ m' := \mathsf{Dec}(X') \\ \text{output: } m' \end{array} \right\}$$

Let $\mathcal{F}$ be a family of functions from $\mathcal{X}$ to $\mathcal{X}$. We say that an encoding scheme $(\mathsf{Enc}, \mathsf{Dec})$ is *ϵ-non-malleable with respect to $\mathcal{F}$* if for every function $F \in \mathcal{F}$ there exists distribution D^F on $\mathcal{M} \cup \{\mathsf{same}^*, \perp\}$ such that for every $m \in \mathcal{M}$ we have

$$\mathsf{Tamper}_m^F \approx_\epsilon \left\{ \begin{array}{c} d \leftarrow D^F \\ \text{if } d = \mathsf{same}^* \text{ then output } m \\ \text{otherwise output } d. \end{array} \right\} \tag{3}$$

The idea behind the "$\perp$" symbol is that it should correspond to the situation when the decoding function detects tampering and outputs an error message. Since the codes that we construct in this paper do not need this feature, we will usually drop this symbol and have $\mathsf{Dec} : \mathcal{X} \to \mathcal{M}$. The "$\perp$" symbol is actually more useful for the *strong* non-malleable codes (another notion defined in [23]) where it is required that *any* tampering with X should be either "detected" or should produce encoding of an unrelated message. Our codes do not have this property. This is because, for example, permuting the elements of the vectors L and R in the same manner *does* change these vectors, but *does not* change their inner product. Fortunately, for all applications that we are aware of this stronger notion is not needed. The following lemma, already informally discussed in Sect. 1, states that for one-bit messages non-malleability is equivalent to the hardness of negating a random encoded bit. It turns out that such a characterization of the non-malleable codes is much simpler to deal with. We also believe that it may be of independent interest.

Lemma 2. *Suppose $\mathcal{M} = \{0,1\}$. Let $\mathcal{F}$ be any family of functions from $\mathcal{X}$ to $\mathcal{X}$. An encoding scheme $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X}, \mathsf{Dec} : \mathcal{X} \to \mathcal{M})$ is ϵ-non-malleable with respect to $\mathcal{F}$ if and only if for any $F \in \mathcal{F}$ and $B \leftarrow \{0,1\}$ we have*

$$P\left[\mathsf{Dec}\left(F(\mathsf{Enc}(B))\right) \neq B\right] \leq \frac{1}{2} + \epsilon. \tag{4}$$

The proof of this lemma appears in the full version of this paper [21]. In this paper we are interested in the split-state codes. A *split-state code* is a pair $(\mathsf{Enc} : \mathcal{M} \to \mathcal{L} \times \mathcal{R}, \mathsf{Dec} : \mathcal{L} \times \mathcal{R} \to \mathcal{M})$. We say that it is *$\epsilon$-non-malleable* if it is ϵ-non-malleable with respect to a family of *all* functions $\mathsf{Mall}^{f,g}$ defined as $\mathsf{Mall}^{f,g}(L,R) = (f(L), g(R))$.

4 The Construction

In this section we present a construction of a non-malleable code in the split-state model, together with a security proof. Before going to the technical details, let us start with some intuitions. First, it is easy to see that any such code $(\mathsf{Enc}, \mathsf{Dec})$ needs to be a 2-out-of-2 secret sharing scheme, where Enc is the sharing function, Dec is the reconstruction function, and $(L, R) = \mathsf{Enc}(M)$ are shares of a secret M. Informally speaking, this is because if one of the "shares", L, say, reveals some non-trivial information about M then by modifying L we can "negate" stored secret M with probability significantly higher than $1/2$. More precisely, suppose that $\mathcal{M} = \{0, 1\}$ and that we know that there exist some values $\ell_0, \ell_1 \in \mathcal{L}$ such that for $b = 0, 1$ if $L = \ell_b$ then M is significantly more likely to be equal to b. Then (f, g) where g is an identity and f is such that $f(\ell_0) = \ell_1$ and $f(\ell_1) = \ell_0$ would lead to $M' = \mathsf{Dec}(f(L), g(R)) = 1 - M$ with probability significantly higher than $1/2$ (this argument is obviously informal, but it can be formalized).

It is also easy to see that not every secret sharing scheme is a non-malleable code in the split-state model. As an example consider $\mathsf{Enc} : Z_a \to Z_a \times Z_a$ (for some $a \geq 2$) defined as $\mathsf{Enc}(M) := (L, L + M \pmod a)$, where $L \leftarrow Z_a$, and $\mathsf{Dec}(L, R) := L + R \bmod m$. Obviously it is a good 2-out-of-2 secret sharing scheme. However, unsurprisingly, it is malleable, as an adversary can, e.g., easily add any constant $w \in Z_a$ to a encoded message, by choosing an identity function as f, and letting g be such that that $g(R) = R + w \bmod a$. Obviously in this case for every L and R that encode some M we have $\mathsf{Dec}(f(L), g(R)) = M + w \bmod a$.

We therefore need to use a secret sharing scheme with some extra security properties. A natural idea is to look at the two-source randomness extractors, as they may be viewed exactly as "2-out-of-2 secret sharing schemes with enhanced security", and since they have already been used in the past in the context of the leakage-resilient cryptography. The first, natural idea, is to take the inner product extractor $\mathsf{ext} : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ and use it as a code as follows: to encode a message $M \in \mathbb{F}$ take a random pair $(L, R) \in \mathbb{F}^n \times \mathbb{F}^n$ such that $\langle L, R \rangle = M$ (to decode (L, R) simply compute $\langle L, R \rangle$). This way of encoding messages is a standard method to provide leakage-resilience in the split-state model (cf. e.g. [14]). Unfortunately, it is easy to see that this scheme can easily be broken by exploiting the linearity attacks of the inner product. More precisely, if the adversary chooses $f(L) := a \cdot L$ and $g(R) := R$ (for any $a \in \mathbb{F}$) then the encoded secret gets multiplied by a. Obviously, this attack does not work for $\mathbb{F} = Z_2$, as in this case the only choices are $a = 0$ (which means that the secret is deterministically transformed to 0) and $a = 1$ (which leaves the secret unchanged). Sadly, it turns out that for $\mathbb{F} = Z_2$ another attack is possible. Consider f and g that leave their input vectors unchanged except of setting the first coordinate of the vector to 1, i.e.: $f(L_1, \ldots, L_n) := (1, L_2, \ldots, L_n)$ and $g(R_1, \ldots, R_n) := (1, R_2, \ldots, R_n)$. Then it is easy to see that $\langle f(L), g(R) \rangle \neq \langle L, R \rangle$ if and only if $L_1 \cdot R_1 = 0$, which happens with probability $3/4$ both for $M = 0$ and for $M = 1$.

Note that the last attack is specific for small $\mathbb{F}$'s, as over larger fields the probability that $L_1 \cdot R_1 = 0$ is negligible. At the first glance, this fact should not bring any hope for a solution, since, as described above, for larger fields another

attack exists. Our key observation is that for one-bit messages it is possible to combine the benefits of the "large field" solution with those of the "small field" solution in such a way that the resulting scheme is secure, and in particular both attacks are impossible! Our solution works as follows. The codewords are pairs of vectors from $\mathbb{F}^n$ for a large $\mathbb{F}$. The encoding of 0 remains as before – i.e. we encode it as a pair (L, R) of orthogonal vectors. To encode 1 we choose a random pair (L, R) of non-orthogonal vectors, i.e. such that $\langle L, R\rangle$ is a random non-zero element of $\mathbb{F}$. Before going to the technical details let us first "test" this construction against the attacks described above. First, observe that multiplying L (or R) by some constant $a \neq 0$ never changes the encoded bit as $\langle a \cdot L, R\rangle = a \langle L, R\rangle$ which is equal to 0 if and only if $\langle L, R\rangle = 0$. On the other hand if $a = 0$ then $\langle a \cdot L, R\rangle = 0$, and hence the secret gets deterministically transformed to 0, which is also ok. It is also easy to see that the second attack (setting the first coordinates of both the vectors to 1) results in $\langle f(L), g(R)\rangle$ close to uniform (no matter what was the value of $\langle L, R\rangle$), and hence $\mathsf{Dec}(f(L), g(R)) = 1$ with an overwhelming probability.

Let us now define our encoding scheme formally. As already mentioned in Sect. 2 our construction uses a strong flexible two-source extractor $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ in a black-box way (later we show how to instantiate it with an inner product extractor, cf. Thm. 2). This in particular means that we do not use any special properties of the inner product, like the linearity. Also, since $\mathcal{C}$ does not need to be a field, hence obviously the choice to encode 0 is by a pair of vectors such that $\langle L, R\rangle = 0$ (in the informal discussion above) was arbitrary, and one can encode 0 as any pair (L, R) such that $\langle L, R\rangle = c$, for some fixed $c \in \mathbb{F}$. Let $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ be a strong flexible (k, ϵ)-extractor, for some parameters k and ϵ, and let $c \in \mathcal{C}$ be arbitrary. We first define the decoding function. Let $\mathsf{D}^c_{\mathsf{ext}} : \mathcal{L} \times \mathcal{R} \to \{0, 1\}$ be defined as:

$$\mathsf{D}^c_{\mathsf{ext}}(L, R) = \begin{cases} 0 \text{ if } \mathsf{ext}(X) = c \\ 1 \text{ otherwise.} \end{cases}$$

Now, let $\mathsf{E}^c_{\mathsf{ext}} : \{0, 1\} \to \mathcal{L} \times \mathcal{R}$ be an encoding function defined as $\mathsf{E}^c_{\mathsf{ext}}(b) := (L, R)$, where (L, R) is a pair chosen uniformly at random from the set $\{(L, R) : \mathsf{D}^c_{\mathsf{ext}}(L, R) = b\}$. We also make a small additional assumption about ext. Namely, we require that $\tilde{L}$ and $\tilde{R}$ are completely uniform over $\mathcal{L}$ and $\mathcal{R}$ (resp.) then $\mathsf{ext}(\tilde{L}, \tilde{R})$ is completely uniform. More formally

$$\text{for } \tilde{L} \leftarrow \mathcal{L} \text{ and } \tilde{R} \leftarrow \mathcal{R} \text{ we have } d(\mathsf{ext}(\tilde{L}, \tilde{R})) = 0. \tag{5}$$

The reason why we impose this assumption is that it significantly simplifies the proof, thanks to the following fact. It is easy to see that if ext satisfies (5), then for every $x \in \mathcal{C}$ the cardinality of each set $\{(\ell, r) : \mathsf{ext}(\ell, r) = x\}$ is exactly $1/|\mathbb{F}|$ fraction of the cardinality of $\mathcal{L} \times \mathcal{R}$. Hence, if $B \leftarrow \{0, 1\}$ and $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(B)$, then in the distribution of (L, R) every (ℓ, r) such that $\mathsf{ext}(\ell, r) = c$ is exactly $(|\mathcal{C}| - 1)$ more likely than any (ℓ', r') such that $\mathsf{ext}(\ell', r') \neq c$. Formally:

$$P\left[(L, R) = (\ell, r)\right] = (|\mathcal{C}| - 1) \cdot P\left[(L, R) = (\ell', r')\right]. \tag{6}$$

It is also straightforward to see that every extractor can be easily converted to an extractor that satisfies (5)[3]. Lemma 3 below is the main technical lemma of this paper. It states that $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ is non-malleable, for an appropriate choice of ext. Since later (in Sect. 5) we will re-use this lemma in the context of non-malleability with leakages, we prove it in a slightly more general form. Namely, (cf. (8)) we show that it is hard to negate an encoded bit even if one knows that the codeword (L, R) happens to be an element of some set $\mathcal{L}' \times \mathcal{R}' \subseteq \mathcal{L} \times \mathcal{R}$. Note that we do not explicitly assume any lower bound on the cardinality of $\mathcal{L}' \times \mathcal{R}'$. This is not needed, since this cardinality is bounded implicitly in (7) by the fact that in any flexible extractor the parameter k needs to be larger than $\max(\log|\mathcal{L}|, \log|\mathcal{R}|)$ (cf. Sect. 2). If one is not interested in leakages then one can read Lemma 3 and its proof assuming that $\mathcal{L}' \times \mathcal{R}' = \mathcal{L} \times \mathcal{R}$. Lemma 3 is stated abstractly, but one can, of course, obtain a concrete non-malleable code, by using as ext the two-source extractor $\mathsf{ext}^n_{\mathbb{F}}$. We postpone presenting the choice of concrete parameters $\mathbb{F}$ and n until Section 5, where it is done in a general way, also taking into account leakages.

Lemma 3. *Let $\mathcal{L}'$ and $\mathcal{R}'$ be some subsets of $\mathcal{L}$ and $\mathcal{R}$ respectively. Suppose $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ is a strong flexible (k, ϵ)-extractor that satisfies (5), where, for some parameter δ we have:*

$$k = \frac{2}{3} \cdot (\log|\mathcal{L}'| + \log|\mathcal{R}'|) - \frac{2}{3} \cdot \log(1/\delta). \tag{7}$$

Take arbitrary functions $f : \mathcal{L} \to \mathcal{L}$ and $g : \mathcal{R} \to \mathcal{R}$, let B be chosen uniformly at random from $\{0,1\}$ and let $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(B)$. Then

$$P\left[\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) \neq B \mid (L, R) \in (\mathcal{L}', \mathcal{R}')\right] \leq \frac{1}{2} + \frac{3}{2}|\mathcal{C}|^{-1} + 6|\mathcal{C}|^2 \epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon), \tag{8}$$

and, in particular $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ is $\left(\frac{3}{2}|\mathcal{C}|^{-1} + 6|\mathcal{C}|^2 \epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon)\right)$-non-malleable.

Proof. Before presenting the main proof idea let us start with some simple observations. First, clearly it is enough to show (8), as then the fact that $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ is $\left(|\mathcal{C}|^{-1} + 2|\mathcal{C}|^2 \epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon)\right)$-non-malleable can be obtained easily by assuming that $\mathcal{L}' \times \mathcal{R}' = \mathcal{L} \times \mathcal{R}$ and applying Lemma 2. Observe also that (8) implies that $\log|\mathcal{L}'| + \log|\mathcal{R}'| \geq k$, and hence, from the fact that ext is a (k, ϵ)-two source extractor we obtain that if $\tilde{L} \leftarrow \mathcal{L}'$ and $\tilde{R} \leftarrow \tilde{R}'$ then

$$d(\mathsf{ext}(\tilde{L}, \tilde{R})) \leq \epsilon. \tag{9}$$

We will use this fact later. The basic idea behind the proof is a as follows. Denote $B' := \mathsf{Mall}^{f,g}(\mathsf{Enc}(B))$. Recall that our code is "non-balanced" in the sense that

[3] The inner-product extractor satisfies (5) if we assume, e.g., that the fist coordinate of $\mathcal{L}$ and the last coordinate of $\mathcal{R}$ are non-zero. In general, if $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ is any extractor, then $\mathsf{ext}' : (\mathcal{L} \times \mathcal{C}) \times \mathcal{R} \to \mathcal{C}$ defined as $\mathsf{ext}((C, L), R) = ext(L, R) + C$ (assuming that $(\mathcal{C}, +)$ is a group) satisfies (5).

a random codeword $(L, R) \in \mathcal{L}' \times \mathcal{R}'$ with only negligible probability encodes 0. We will exploit this fact. Very informally speaking, we would like to prove that if $B = 1$ then the adversary cannot force B' to be equal to 0, as any independent modifications of L and R that encode 1 are unlikely to produce an encoding of 0. In other words, we would hope to show that $P[B' = 0 | B = 1]$ is small. Note that if we managed to show it, then we would obviously get that $P[B' \neq B]$ cannot be much larger than $1/2$ (recall that B is uniform), and then the proof would be finished. Unfortunately, this is too good to be true, as the adversary can choose f and g to be constant such that always $\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 0$, which would result in $B' = 0$ for any value of B. Intuitively, what we will actually manage to prove is that the only way to obtain $B' = 0$ if $B = 1$ is to apply such a "constant function attack". Below we show how to make this argument formal.

Let us first observe that any attack where f and g are constant will never work against any encoding scheme, as in this case $(f(L), g(R))$ carries no information about the initial value of B. Our first key observation is that for our scheme, thanks to the fact that it is based on extractor, this last statement holds even if any of f and g is only "sufficiently close to constant". Formalizing this property is a little bit tricky, as, of course, the adversary can apply "mixed" strategies, e.g., setting f to be constant on some subset of $\mathcal{L}'$ and to be injective (and hence "very far from constant") on the rest of $\mathcal{L}'$. In order to deal with such cases we will define subsets $\mathcal{L}_{\mathsf{FFC}} \subseteq \mathcal{L}'$ and $\mathcal{R}_{\mathsf{FFC}} \subseteq \mathcal{R}'$ on which f and g (resp.) are "very far from constant". Formally, for $\tilde{L} \leftarrow \mathcal{L}'$ and $\tilde{R} \leftarrow \mathcal{R}'$ let

$$\mathcal{L}_{\mathsf{FFC}} := \left\{ \ell \in \mathcal{L}' : \mathbf{H}_\infty(\tilde{L} \mid f(\tilde{L}) = f(\ell)) < k + 1 - \log |\mathcal{R}'| \right\},$$

and

$$\mathcal{R}_{\mathsf{FFC}} := \left\{ r \in \mathcal{R}' : \mathbf{H}_\infty(\tilde{R} \mid g(\tilde{R}) = g(r)) < k + 1 - \log |\mathcal{L}'| \right\},$$

where FFC stands for "far from constant". Hence, in some sense, we define a function to be "very far from constant on some argument x" if there are only a few other arguments of this function that collide with x. We now state the following claim (whose proof appears in the full version of this paper [21]) that essentially formalizes the intuition outlined above, by showing that if either $L \notin \mathcal{L}_{\mathsf{FFC}}$ or $R \notin \mathcal{R}_{\mathsf{FFC}}$ then (f, g) cannot succeed in negating B.

Claim 1. *Let $B \leftarrow \{0,1\}$ and $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(B)$. Then:*

$$P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R)) \neq B \mid L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}} \right] \leq \frac{1}{2} + \frac{3}{4} \cdot |\mathcal{C}|^{-1} + 6\,|\mathcal{C}|^2\,\epsilon. \tag{10}$$

Hence, what remains is to analyze the case when $(L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$. We will do it only for the case $B = 1$, and when $\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$ is relatively large, more precisely we will assume that

$$|\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}| \geq \delta \cdot |\mathcal{L}' \times \mathcal{R}'|. \tag{11}$$

This will suffice since later we will show (cf. (23)) that the probability that $\mathsf{Enc}(B) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$ is small for small δ's (note that this is not completely trivial as (L, R) does not have a uniform distribution over $\mathcal{L}' \times \mathcal{R}'$). We now have the following claim whose proof appears in the full version of this paper [21].

Claim 2. *Let* $(L^1, R^1) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(1)$ *and suppose* $\mathcal{L}_{\mathsf{FFC}}$ *and* $\mathcal{R}_{\mathsf{FFC}}$ *are such that (11) holds. Then*

$$P\left[\mathsf{D}^c_{\mathsf{ext}}\left(\mathsf{Dec}(f(L^1), g(R^1))\right) = 0 \mid (L^1, R^1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right] \leq 2\left|\mathcal{C}\right|^{-1} + 2\epsilon. \quad (12)$$

To finish the proof we need to combine the two above claims. A small technical difficulty, that we need still to deal with, comes from the fact that Claim 2 was proven only under the assumption (11). Let us first expand the left-hand-side of (8). We have

$$P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R) \neq B | (L, R) \in \mathcal{L}' \times \mathcal{R}'\right] \quad (13)$$

$$= \overbrace{P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R) \neq B \mid L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}}\right]}^{(*)} \cdot P\left[L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}}\right] \quad (14)$$

$$+ \overbrace{P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R) \neq B \mid (L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}^{(**)} \cdot P\left[(L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right] \quad (15)$$

From Claim 1 we get that $(*)$ is at most $\frac{1}{2} + \frac{1}{2} \cdot \left|\mathcal{C}\right|^{-1} + 2\left|\mathcal{C}\right|^2 \epsilon$. Now consider two cases.

Case 1 First, suppose that (11) holds (i.e. $|\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}| \geq \delta \cdot |\mathcal{L} \times \mathcal{R}|$). In this case we get that $(**)$ is a equal to

$$\overbrace{P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R) \neq B | B = 0 \wedge (L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}^{\leq 2|\mathcal{C}|^{-1} + 2\epsilon \text{ by Claim 2}} \cdot \overbrace{P\left[B = 0 | (L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}^{\geq \frac{1}{2} - |\mathcal{C}|\epsilon} + \quad (16)$$

$$\underbrace{P\left[\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L, R) \neq B | B = 1 \wedge (L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}_{\leq 1} \cdot \underbrace{P\left[B = 1 | (L, R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}_{\leq \frac{1}{2} + |\mathcal{C}|\epsilon} \quad (17)$$

$$\leq \frac{1}{2} + \left|\mathcal{C}\right|^{-1} - \epsilon + \left|\mathcal{C}\right|(\epsilon - \epsilon^2) \leq \frac{1}{2} + \left|\mathcal{C}\right|^{-1} + \left|\mathcal{C}\right|\epsilon. \quad (18)$$

The inequalities in (16) and (18) follow from the fact that $\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$ is a large set and the fact that B depends on $\mathsf{ext}(L, R)$, where ext is a randomness

extractor. The detailed proof of these inequalities appears in the full version of this paper [21]. Now, since (13) is a weighted average of $(*)$ and $(**)$, hence obviously

$$(13) \tag{19}$$

$$\leq \max\left(\frac{1}{2} + \frac{3}{2}\cdot|\mathcal{C}|^{-1} + 6\,|\mathcal{C}|^{2}\,\epsilon, \frac{1}{2} + |\mathcal{C}|^{-1} + |\mathcal{C}|\,\epsilon\right) \tag{20}$$

$$\leq \frac{1}{2} + \frac{3}{2}\,|\mathcal{C}|^{-1} + 6\,|\mathcal{C}|^{2}\,\epsilon. \tag{21}$$

Case 2 Now consider the case when (11) does not hold, i.e.:

$$|\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}| < \delta \cdot |\mathcal{L} \times \mathcal{R}| \tag{22}$$

We now give a bound on the probability that (L, R) is a member of $\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$.

$$\begin{aligned}
&P\left[(L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right] \\
&= \frac{1}{2}\cdot P\left[\mathsf{E}^{c}_{\mathsf{ext}}(0) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right] + \frac{1}{2}\cdot P\left[\mathsf{E}^{c}_{\mathsf{ext}}(1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right] \\
&= \frac{1}{2}\cdot P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} \mid \mathsf{ext}(\tilde{L},\tilde{R}) = c\right] + \\
&\quad \frac{1}{2}\cdot P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} \mid \mathsf{ext}(\tilde{L},\tilde{R}) \neq c\right] \\
&\leq \frac{1}{2}\cdot \frac{P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}{P\left[\mathsf{ext}(\tilde{L},\tilde{R}) = c\right]} + \frac{1}{2}\cdot \frac{P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}{P\left[\mathsf{ext}(\tilde{L},\tilde{R}) \neq c\right]} \\
&\leq \frac{1}{2}\cdot \frac{P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}{|\mathcal{C}|^{-1} - \epsilon} + \frac{1}{2}\cdot \frac{P\left[(\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right]}{(|\mathcal{C}|-1)\cdot|\mathcal{C}|^{-1} - \epsilon} \qquad (23) \\
&\leq \delta/(|\mathcal{C}|^{-1} - \epsilon),
\end{aligned}$$

where in (23) we used (9). Hence, in this case, (15) is at most equal to $\delta/(|\mathcal{C}|^{-1} - \epsilon)$, and therefore, altogether, we can bound (13) by

$$(13) \leq (*) + \delta/(|\mathcal{C}|^{-1} - \epsilon) \tag{24}$$

$$= \frac{1}{2} + \frac{3}{2}\,|\mathcal{C}|^{-1} + 6\,|\mathcal{C}|^{2}\,\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon) \tag{25}$$

Since analyzing both cases gave us bounds (21) and (25), hence all in all we can bound (13) by their maximum, which is at most

$$\frac{1}{2} + \frac{3}{2}\,|\mathcal{C}|^{-1} + 6\,|\mathcal{C}|^{2}\,\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon).$$

Hence (8) is proven.

5 Adding Leakages

In this section we show how to incorporate leakages into our result. First, we need to extend the non-malleability definition. We do it in the following, straightforward way. Observe that we can restrict ourselves to the situation when the leakages happen *before* the mauling process (as it is of no help to the adversary to leak from $(f(L), g(R)$ if he can leak already from (L, R)). For any split-state encoding scheme $(\mathsf{E}^c_{\mathsf{ext}} : \mathcal{M} \to \mathcal{L} \times \mathcal{R}, \mathsf{D}^c_{\mathsf{ext}} : \mathcal{L} \times \mathcal{R} \to \mathcal{M})$, a family of functions $\mathcal{F}$, any $m \in \mathcal{M}$ and any adversary $\mathcal{A}$ define a game $\mathsf{Tamper}^{\mathcal{A}}_m$ (where λ is some parameter) as follows. First, let $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(m)$. Then the adversary $\mathcal{A}$ chooses a sequence of functions $(v^1, w^1, \ldots, v^t, w^t)$, where each v^i has a type $v^i : \mathcal{L} \to \{0,1\}^{\lambda_i}$ and each w^i has a type $w^i : \mathcal{R} \to \{0,1\}^{\rho_i}$ where the λ's and ρ's are some parameters such that

$$\lambda_1 + \cdots + \lambda_t + \rho_1 + \cdots \rho_t \leq \lambda. \tag{26}$$

He learns $\mathsf{Leak}(L, R) = \big(v^1(L), w^1(R), \ldots, v^t(L), w^t(R)\big)$. Moreover this process is *adaptive*, i.e. the choice of an ith function in the sequence (26) can depend on the $i-1$ first values in the sequence $\mathsf{Leak}(L, R)$. Finally the adversary chooses functions $f : \mathcal{L} \to \mathcal{L}$ and $g : \mathcal{R} \to \mathcal{R}$. Now define the output of the game as: $\mathsf{Tamper}^{\mathcal{A}}_m := (f(L), g(R))$. We say that the encoding scheme $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ is ϵ-*non-malleable with leakage* λ if for every adversary $\mathcal{A}$ there exists distribution $D^{\mathcal{A}}$ on $\mathcal{M} \cup \{\mathsf{same}^*\}$ such that for every $m \in \mathcal{M}$ we have

$$\mathsf{Tamper}^{\mathcal{A}}_m \approx_\epsilon \left\{ \begin{array}{c} d \leftarrow D^{\mathcal{A}} \\ \text{if } d = \mathsf{same}^* \text{ then output } m, \\ \text{otherwise output } d. \end{array} \right\}$$

Theorem 1. *Suppose* $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ *is a flexible* (k, ϵ)*-extractor that satisfies (5), where, for some parameters* δ *and* λ *we have*

$$k = \frac{2}{3} \cdot (\log |\mathcal{L}| + \log |\mathcal{R}| - \lambda) - \frac{4}{3} \cdot \log(1/\delta). \tag{27}$$

Then the encoding scheme is $\big(\frac{3}{2} |\mathcal{C}|^{-1} + 6 |\mathcal{C}|^2 \epsilon + 2\delta/(|\mathcal{C}|^{-1} - \epsilon)\big)$*-non-malleable with leakage* λ*.*

The proof of this theorem appears in the full version of this paper [21]. We now show how to instantiate Theorem 1 with the inner-product extractor from Sect. 2.

Theorem 2. *Take any* $\xi \in [0, 1/4)$ *and* $\gamma > 0$ *then there exist an explicit split-state code* $(\mathsf{Enc} : \{0,1\} \to \{0,1\}^{N/2} \times \{0,1\}^{N/2}, \mathsf{Dec} : \{0,1\}^{N/2} \times \{0,1\}^{N/2} \to \{0,1\})$ *that is* γ*-non-malleable with leakage* $\lambda := \xi N$ *such that* $N = \mathcal{O}(\log(1/\gamma) \cdot (1/4 - \xi)^{-1})$*. The encoding and decoding functions are computable in* $\mathcal{O}(N \cdot \log^2(\log(1/\gamma)))$ *and the constant hidden under the* $\mathcal{O}$*-notation in the formula for* N *is around* 100*.*

The proof of this theorem appears in the full version of this paper [21]. We would like to remark that it does not look like we could prove, with our current proof techniques, a better relative leakage bound than $\xi < \frac{1}{4}$. Very roughly speaking it is because we used the fact that the inner product is an extractor twice in the proof. On the other hand we do not know any attack on our scheme for relative leakage $\xi \in \left(\frac{1}{4}, \frac{1}{2}\right)$ (recall that for $\xi = \frac{1}{2}$ obviously any scheme is broken). Hence, it is quite possible, that with a different proof strategy (perhaps relying on some special features of the inner product function) one could show a higher leakage tolerance of our scheme.

6 Security against Affine Mauling

Interestingly, we can also show that our encoding scheme $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$, instantiated with the inner product extractor, is secure in the model where $(L, R) \in \mathbb{F}^n \times \mathbb{F}^n$ can be mauled simultaneously (i.e. we do not use the split-model assumption), but the class of the mauling functions is restricted to the affine functions over $\mathbb{F}$, i.e. each mauling function h is of a form

$$h((L_1, \ldots, L_n), (R_1, \ldots, R_n)) = M \cdot (L_1, \ldots, L_n, R_1, \ldots, R_n)^T + V^T, \qquad (28)$$

where M is an $(2n \times 2n)$-matrix over $\mathbb{F}$ and $V \in \mathbb{F}^{2n}$. We now argue informally why it is the case, by showing that every h that breaks the non-malleability of this scheme can be transformed into a pair of functions (f, g) that breaks the non malleability of the scheme

$$\left(\mathsf{E}^c_{\mathsf{ext}} : \mathcal{F}^{n+2} \times \mathcal{F}^{n+2} \to \{0,1\}, \mathsf{D}^c_{\mathsf{ext}} : \{0,1\} \to \mathcal{F}^{n+2} \times \mathcal{F}^{n+2}\right)$$

in the split-state model. Let $(L, R) \in \mathbb{F}^{n+2} \times \mathbb{F}^{n+2}$ denote the codeword in this scheme. Our attack works only under the assumption that it happened that $(L, R) \in \mathcal{L}' \times \mathcal{R}'$, where $\mathcal{L}' \times \mathcal{R}' := (\mathbb{F}^n \times \{0\} \times \{0\}) \times (\mathbb{F}^n \times \{0\} \times \{0\})$ (in other words: the two last coordinates of both L and R are zero). Since $\mathcal{L}' \times \mathcal{R}'$ is large, therefore this clearly suffices to obtain the contradiction with the fact that our scheme is secure even if (L, R) happen to belong to some large subdomain of the set of all codewords (cf. Lemma 3). Clearly, to finish the argument it is enough to construct the functions f and g such that

$$\langle f(L), g(R) \rangle = \left\langle (L'_1, \ldots, L'_{n+2}), (R'_1, \ldots, R'_{n+2}) \right\rangle,$$

where $(L'_1, \ldots, L'_{n+2}, R'_1, \ldots, R'_{n+2}) = h(L_1, \ldots, L_n, R_1, \ldots, R_n)$. It is easy to see that, since h is affine, hence the value of $\left\langle (L'_1, \ldots, L'_{n+2}), (R'_1, \ldots, R'_{n+2}) \right\rangle$ can be represented as a sum of monomials over variables L_i and R_j where each variable appears in power at most 1. Hence it can be rewritten as the following sum:

$$\sum_{i=1}^{n} \left(L_i \cdot \sum_{j \in J_i} R_j \right) + \sum_{j \in J_{n+1}} L_j + \sum_{i,j \in K_{n+1}} L_i L_j + y + \sum_{j \in J_{n+2}} R_j + \sum_{i,j \in K_{n+2}} R_i R_j,$$

where each J_i is a subset of the indices $\{1, \ldots, n\}$ and $y \in \mathbb{F}$ is a constant. It is also easy to see that the above sum is equal to the inner product of vectors V and W defined as:

$$V := \Big(L_1, \ldots, L_n, \sum_{j \in J_{n+1}} L_j + \sum_{i,j \in K_{n+1}} L_i L_j, 1\Big)$$

$$W := \Big(\sum_{j \in J_1} R_j, \ldots, \sum_{j \in J_n} R_j, 1, y + \sum_{j \in J_{n+2}} R_j + \sum_{i,j \in K_{n+2}} R_i R_j\Big).$$

Now observe that V depends only on the vector L, and similarly, W depends only on R. We can therefore set $f(L) := V$ and $g(R) := W$. This finishes the argument.

References

1. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. Cryptology ePrint Archive, Report 2013/201 (2013), http://eprint.iacr.org/
2. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
3. Anderson, R., Kuhn, M.: Tamper resistance - a cautionary note. In: The Second USENIX Workshop on Electronic Commerce Proceedings (November 1996)
4. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 647–647. Springer, Heidelberg (2003)
5. Biham, E.: New types of cryptanalytic attacks using related keys. Journal of Cryptology 7(4), 229–246 (1994)
6. Bourgain, J.: More on the sum-product phenomenon in prime fields and its applications. International Journal of Number Theory 1(1), 1–32 (2005)
7. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 501–510. IEEE (2010)
8. Chabanne, H., Cohen, G., Flori, J., Patey, A.: Non-malleable codes from the wiretap channel. In: 2011 IEEE Information Theory Workshop (ITW), pp. 55–59. IEEE (2011)
9. Chabanne, H., Cohen, G., Patey, A.: Secure network coding and non-malleable codes: Protection against linear tampering. In: 2012 IEEE International Symposium on Information Theory Proceedings (ISIT), pp. 2546–2550 (2012)
10. Choi, S.G., Kiayias, A., Malkin, T.: BiTR: Built-in tamper resilience. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 740–758. Springer, Heidelberg (2011)
11. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM Journal on Computing 17(2), 230–261 (1988)
12. Cohen, G., Raz, R., Segev, G.: Non-malleable extractors with short seeds and applications to privacy amplification. In: Computational Complexity (CCC), pp. 298–308 (2012)

13. Dachman-Soled, D., Kalai, Y.T.: Securing circuits against constant-rate tampering. In: Safavi-Naini, R. (ed.) CRYPTO 2012. LNCS, vol. 7417, pp. 533–551. Springer, Heidelberg (2012)
14. Davì, F., Dziembowski, S., Venturi, D.: Leakage-resilient storage. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 121–137. Springer, Heidelberg (2010)
15. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 511–520. IEEE Computer Society (2010)
16. Dodis, Y., Lewko, A., Waters, B., Wichs, D.: Storing secrets on continually leaky devices. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 688–697. IEEE (2011)
17. Dodis, Y., Li, X., Wooley, T., Zuckerman, D.: Privacy amplification and non-malleable extractors via character sums. In: FOCS 2011, pp. 668–677 (2011)
18. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: STOC, pp. 601–610 (2009)
19. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Review 45(4), 727–784 (2003)
20. Dziembowski, S., Faust, S.: Leakage-resilient circuits without computational assumptions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 230–247. Springer, Heidelberg (2012)
21. Dziembowski, S., Kazana, T., Obremski, M.: Non-malleable codes from two-source extractors. Cryptology ePrint Archive (2013), Full version of this paper, `http://eprint.iacr.org/`
22. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008, pp. 293–302. IEEE (2008)
23. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: ICS, pp. 434–452 (2010)
24. ECRYPT. European Network of Excellence. Side Channel Cryptanalysis Lounge, `http://www.emsec.rub.de/research/projects/sclounge`
25. Faust, S., Pietrzak, K., Venturi, D.: Tamper-proof circuits: How to trade leakage for tamper-resilience. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 391–402. Springer, Heidelberg (2011)
26. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (2004)
27. Goldwasser, S., Rothblum, G.: How to compute in the presence of leakage. In: FOCS 2012, pp. 31–40 (2012)
28. Halevi, S., Lin, H.: After-the-fact leakage in public-key encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 107–124. Springer, Heidelberg (2011)
29. Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)
30. Ishai, Y., Prabhakaran, M., Sahai, A., Wagner, D.: Private circuits II: Keeping secrets in tamperable circuits. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 308–327. Springer, Heidelberg (2006)
31. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
32. Kalai, Y.T., Kanukurthi, B., Sahai, A.: Cryptography with tamperable and leaky memory. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 373–390. Springer, Heidelberg (2011)

33. Liu, F.-H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini, R. (ed.) CRYPTO 2012. LNCS, vol. 7417, pp. 517–532. Springer, Heidelberg (2012)
34. Micali, S., Reyzin, L.: Physically observable cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
35. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
36. Rao, A.: An exposition of bourgain 2-source extractor. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 14, p. 034 (2007)
37. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 262–279. Springer, Heidelberg (2012)