University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Maciej Obremski

Flexible Two-Source Extractors and their Applications

PhD dissertation

Supervisor

dr hab. Stefan Dziembowski

Institute of Informatics
University of Warsaw

November 2012

Author's declaration:
aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

November, 2012 .....

date *Maciej Obremski*

Supervisor's declaration:
the dissertation is ready to be reviewed

November, 2012 .....

date *dr hab. Stefan Dziembowski*

Abstract

We introduce a new notion *flexible* extractor. It is a generalization of the standard concept of a two-source-extractor which require each of a sources to have some entropy, *flexible* extractor requires the sum of sources entropy to exceed fixed value. We distinguish between a strong and a weak *flexible* extractors and (similarly to two-source-extractors case) prove that every weak *flexible* extractor is also a strong extractor just with a slightly worse parameters. Moreover we prove that two common two-source extractors are in fact *flexible* which can be viewed as a generalization of the Leftover Hash Lemma for those extractors. We use that notion in joint work with Stefan Dziembowski and Tomasz Kazana "Non-Malleable Codes from Two-Source Extractors" currently under submission. In that work we use the flexible extractors to 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 the property of being *flexible*. 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 $1/2 + \epsilon$.

Key words: non-malleable codes, two-source extractors, flexible two-source extractors

AMS Classification: 68P20, 68P25, 68P30, 94A60.

Streszczenie

Prezentujemy nowe pojęcie *elastycznego* extraktora dwuźródłowego. W przeciwieństwie do standardowych dwuźródłowych ekstraktorów, które wymagają by każde ze źródeł osobno miało pewną entropię, *elastyczny* ekstraktor wymaga by sumaryczna entropia źródeł przekraczała daną wartość. Wyróżniamy słabe i silne *elastyczne* ekstraktory i podobnie jak w przypadku słabych i silnych ekstraktorów dwuźródłowych dowodzimy, że każdy słaby *ekstraktor* jest też silny kosztem nieznacznego pogorszenia jego parametrów. Ponadto dowodzimy, że dwa z powszechnie znanych i używanych ekstraktorów są *elastyczne* co znacząco wzmacnia tezę Leftover Hash Lemma dla tych ekstraktorów. Pojęcia *elastycznych* ekstraktorów używamy we wspólnej pracy ze Stefanem Dziembowskim i Tomaszem Kazaną "Non-Malleable Codes from Two-Source Extractors", praca ta została wysłana na międzynarodową konferencję. Konstruujemy w niej wydajny, teorio-informacyjnie bezpieczny kod niekowalny w modelu z przepołowioną pamiecią dla wiadomości jednobitowych. Pojęcie kodów niekowalnych zostało wprowadzone przez S.Dziembowskiego, K.Pietrzaka i D.Wichsa (ICS 2010), jako narzędzie do składowania danych na urządzeniu, które może być poddane działaniu przeciwnika modyfikującego dane. Nieformalnie ujmując, schemat $(\mathsf{Enc} : \mathcal{M} \to \mathcal{L} \times \mathcal{R}, \mathsf{Dec} : \mathcal{L} \times \mathcal{R} \to \mathcal{M})$ jest *kodem niekowalnym w modelu z przepołowioną pamięcią* jeśli wspomniany przeciwnik manipulujący *niezależnie* L i R (gdzie (L, R) koduje pewną wiadomość m) nie może otrzymac, kodu wiadomości m', która byłaby różna od m ale z nią "skorelowana"(np. $m' = m + 1$). Do teraz efektywna konstrukcja informacyjnie bezpiecznego kodu o takiej wlasności pozostawala nieznana nawet dla wiadomości ze zbioru $\{0, 1\}$. Nasza konstrukcja rozwiązuje ten problem. Ponadto dowodzimy jej odporności na wycieki, w następującym sensie: przeciwnik zanim wybierze dwie funkcje manipulujące (jedną na L, drugą na R) może poznać dowolną, ustaloną wcześniej funkcję wycieku z (L, R). Formalnie, dla każdego $\xi < 1/4$ potrafimy podać efektywną konstrukcje kodu niekowalnego taką, że przeciwnik przed wyborem funkcji manipulacji pozna wartości wybranych przez siebie adaptywnie funkcji $F_1(L), F_2(R), F_3(L), F_4(R)...$ byle tylko sumaryczna dlugość wyjścia tych funkcji nie przekraczała $\xi \cdot (|L| + |R|)$. Konstrukcja naszego kodu jest oparta na iloczynie skalarnym nad ciałami skończonymi, ale pokazujemy jak zbudowac kod z dowolnego innego dwuźródłowego ekstraktora, który jest *elastyczny (flexible)*. Poza tym pokazujemy, ze definicja kodów niekowalnych w przypadku wiadomości jednobitowych ma równoważną, prostszą charakteryzację mianowicie: jeśli wybierzemy wiadomość m jednostajnie z $\{0, 1\}$ wtedy prawdopodobieństwo, że przeciwnik bedzie w stanie uzyskac (w sposób opisany powyżej) wiadomość przeciwną do m jest niewiększe niż $1/2 + \epsilon$.

Acknowledgments

I would like to thank my advisor and co-author dr hab. Stefan Dziembowski for his input in this dissertation and my research work. I would also like to thank professor dr hab. Stanisław Kwapień for his support and discussions during my research work. I'm very grateful to professors Yevgeniy Dodis and Krzysztof Pietrzak for stimulating discussions regarding non-malleable codes in different models. I am also grateful to dr hab. Wojciech Niemiro for his guidance. Also I want to thank my college and co-author of all research papers Tomasz Kazana. Last but not least i want to express my deepest gratitude to my muse S.G.

Contents

1 Introduction 7
1.1 Our contribution 11
1.2 Related work 12
2 Preliminaries 13
2.1 Entropy 15
3 Extractors 17
3.1 Inner product as flexible extractor 19
3.2 Other flexible extractor example 21
4 Inner product, leakage and Leftover Hash Lemma 25
4.1 Leftover Hash Lemma and non-adaptive leakage 25
4.2 Adaptive leakage 27
4.3 Leakage-resilient storage in the split-state model 28
5 Definition of the non-malleable codes and equivalence to the hardness of negation 29
6 The construction 32
7 Adding Leakages 45
8 Non-malleable codes vs. extractors 49
9 Non-malleable codes vs. leakage-resilent storage 49
10 Security against affine malling 50

1 Introduction

The notion of randomness plays a central role in computer science. For example, several algorithmic tasks are much easier to solve if one allows the algorithm to have access to a string X of random bits. Also in cryptography it is known that some primitives, such as, e.g., the public-key encryption are impossible to construct in a deterministic (i.e. not randomized) way. What is usually assumed in these constructions is that the randomness that they use is uniform. Since uniform randomess rarely appears in practice, a natural approach is to weaken the uniformity requirement and assume that X is far from uniform (for example X could be a result of a measurement of some physical process). To reason in a modular way, the simplest method is to take a standard randomized algorithm A and to run it on some randomness $\mathsf{ext}(X)$ *extracted* from X. For this reason *extractors* were introduced as functions that take weak random sources and output uniformly distributed bits.

Typically an input to the extractor is a random variable X that is not necessarily uniform but it has some randomness. This property is usually formalized by requiring that X has a large *min-entropy*, a notion which is a variant of Shannon's entropy (for a formal definition of min-entropy see Section 2.1). The output of an extractor should be close to uniform distribution (we define this notion formally in Section 2).

Unfortunately in general it is not possible to extract randomness from one variable, i.e., there does not exist a deterministic function ext that on every high-min-entropy variable X outputs $\mathsf{ext}(X)$ close to uniform (cf. [40], Remark 1). This problem is solved by introducing an additional assumption that ext takes an extra input Y, which is a different random variable that is independent from X. There are two basic classes of extractors. The first one is called the *seeded extractors*. In this case Y is guaranteed to be uniform. In order to exclude a trivial solution ($\mathsf{ext}_Y(X) = Y$) the output of $\mathsf{ext}_Y(X)$ must be larger than the length of variable Y. An important subclass of the *seeded extractors* is a *strong seeded extractors* class, where an output of $\mathsf{ext}_Y(X)$ is close to uniform even if the value of variable Y is revealed, more precisely the conditional random variable $\mathsf{ext}_Y(X)|Y$ is close to uniform. The second basic class of extractors are the *2-source extractors*. In this case we do not require independent variable Y to be uniform. We only assume that the two independent random variables X and Y both have high min-entropy. For a formal definition of extractors and further discussion see Section 3.

Extractors are also a very useful tool to thwart leakage and tampering attacks. The main contribution of this work is a new notion of a *flexible extractor* which is generalization of a *2-source extractors* notion. In order to

return uniformly distributed bits a flexible extractor does not require both sources to have min-entropy exceeding some fixed value, instead it only requires that the sum of sources min-entropies exceeds fixed value. This relaxation allows us to use such extractors in a construction of schemes resilient against tampering and leakage, which are explained below.

Leakage and tampering attacks. Real-life attacks on cryptographic devices often do not break their mathematical foundations, but exploit vulnerabilities in their implementations. For example: the PCs can be infected with viruses. In case of a dedicated cryptographic devices such "physical attacks" are usually based on passive measurements such as running-time, electromagnetic radiation, power consumption (see e.g. [36]), or active tampering where the adversary maliciously modifies some part of the device (see e.g. [2]) 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 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. e.g. [31, 30, 22, 1, 35]).

One of the models simulating a computer infected by passive virus (i.e. a virus that can only steal the data not tamper with it) is the Bounded Retrieval Model (see e.g. [16]). In this model the adversary can use a virus to leak the data from the computer. We assume that the secret key inside that computer is so large that it is not efficient for the adversary to download the whole data. To simulate that restriction we let the adversary to choose any function $f : \{0,1\}^n \to \{0,1\}^l$ where l is arbitrary and much smaller then n and then the adversary learns $f(sk)$, where sk is a secret key. On the other hand we expect that the protocols in the BRM have „locality" property. Informally speaking „locality" means that protocol should not require processing whole secret key sk instead it should only need to access part of it (possibly randomly chosen part).

In case of passive measurements on a dedicated cryptographic devices the most common model is very similar to BRM. We simulate passive measurements by any function $f : \{0,1\}^n \to \{0,1\}^l$ chosen by the adversary. The adversary learns $f(m)$ where m is whole memory of the device, as before l

is smaller then m (if $l = cm$ for $c \in (0,1)$ we call it a linear leakage). If the memory of the device is split in to two (or more) parts L, R, we model passive measurement as choice of two functions $f, g : \{0,1\}^{n/2} \to \{0,1\}^{l/2}$ and revile to adversary $f(L)$ and $g(R)$. Idea is to prove that a given scheme is secure even if the adversary leaks that additional information. For more detail and other models see [15, 24, 1].

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 [26] 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. [25]) or because the tampering attacks that they consider are very limited (e.g. [30] or [14] consider only probing attacks, and in [41] the tampering functions is 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 5, 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 on 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: each 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 practi-

cal 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, 15, 27]. 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. $\mathcal{M} = \{0, 1\}$).

1.1 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. 18). 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. 3). 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. 3).

We also provide a simple argument that shows that our scheme is secure against against affine malling functions (that look at the entire codeword,

hence *not* in the split-state model).

Typically in theoretical 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, 14, 20, 26]). This research direction looks especially promising since many of the leakage-resilient compliers (e.g. [20, 26]) 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 15) 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

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

(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.

1.2 Related 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]. Security against both tampering and leakage attacks were also recently con-

sidered 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-theoreticl security.

It is also worth to compare our result with "Algorithmic Tamper-Proof Security" from Gennaro et al. [25]. The idea of that work is to use public-key signature scheme to prevent stored message from being tampered with. Formaly we store secret message s along with signature $\sigma = Sign_{sk}(s)$, where sk is private key. In sense of encoding scheme we can think about it as functions $\mathsf{Enc}_{sk}(s) = (s, Sign_{sk}(s))$ and Dec_{pk} which only verifies the signature using public key pk and in case of invalid signature function Dec_{pk} outputs $\perp$. However this solution does not fulfill security definition of non-malleable codes. In [23] authors show very easy attack on that scheme using independent bit tampering. Simply set first bit of message to 1. If decoding function returns $\perp$ that means first bit was 0, in case decoding function returns valid message we can assume (with high probability) that first bit of message was 1 since it is unlikely that there was same signature for 2 messages that are different only on first coordinate.

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, 13]). 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.

Finally, let us mention that constructions non-malleable codes secure in different (not split-state) models were recently proposed in [8, 9, 10].

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 $\Delta(A, B) \leqslant \epsilon$ 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 $\sum_c P(C=c)\cdot d(A|C=c)$. We now have the following standard lemmas whose proofs can be found e.g. in [21].

Lemma 1. *If A and B are random variables over $\{0,1\}$ then for any $b \in \{0,1\}$ we have $\Delta(A;B) = |P(A=b) - P(B=b)|$.*

Lemma 2. *For any random variables A and B and any function φ we have that $|\Delta(\varphi(A);\varphi(B))| \leqslant \Delta(A;B)$. and in particular $d(A) \leqslant d(A|B)$.*

Lemma 3. *For every random variable A and events $\mathcal{X}$ and $\mathcal{Y}$ we have*

$$\Delta(A|\mathcal{Y}\ ;\ A|\mathcal{X}\wedge\mathcal{Y}) \leqslant 1 - P(\mathcal{X}|\mathcal{Y}). \tag{2}$$

Proof. First observe that for every a we have that

$$\begin{aligned} & P(A=a \mid \mathcal{Y}) - P(A=a \mid \mathcal{X}\wedge\mathcal{Y}) && (3)\\ = \ & P(A=a \mid \mathcal{Y}) - \frac{P(A=a\wedge\mathcal{X} \mid \mathcal{Y})}{P(\mathcal{X} \mid \mathcal{Y})} && (4)\\ \leqslant \ & P(A=a \mid \mathcal{Y}) - P(A=a\wedge\mathcal{X} \mid \mathcal{Y}) && (5)\end{aligned}$$

Hence, the left hand side of (2) is at most equal to

$$\begin{aligned} & \sum_{a:P(A=a \mid \mathcal{Y})>P(A=a \mid \mathcal{X}\wedge\mathcal{Y})} P(A=a \mid \mathcal{Y}) - P(A=a\wedge\mathcal{X} \mid \mathcal{Y}) \leqslant \\ \leqslant \ & \sum_{a:P(A=a \mid \mathcal{Y})>P(A=a \mid \mathcal{X}\wedge\mathcal{Y})} P(A=a \mid \mathcal{Y}) - && (6)\\ & - \sum_{a:P(A=a)>P(A=a \mid \mathcal{X}\wedge\mathcal{Y})} P(A=a\wedge\mathcal{X} \mid \mathcal{Y}) = \\ = \ & P(\mathcal{A}|\mathcal{Y}) - \underbrace{P(\mathcal{A}\wedge\mathcal{X} \mid \mathcal{Y})}_{\geqslant P(\mathcal{A}|\mathcal{Y})-(1-P(\mathcal{X}|\mathcal{Y}))} \leqslant && (7)\\ \leqslant \ & 1 - P(\mathcal{X}|\mathcal{Y}) \end{aligned}$$

(where $\mathcal{A}$ in (7) denotes the event that $A \in \{a : P(A=a \mid \mathcal{Y}) > P(A=a \mid \mathcal{X}\wedge\mathcal{Y})\}$). This finishes the proof. □

Lemma 4. *Let $(A,B) \in \mathcal{A}\times\mathcal{B}$ be a random variable such that $d(A|B) \leqslant \epsilon$. Then for every $a \in \mathcal{A}$ we have*

$$\Delta(B|A=a\ ;\ B) \leqslant 2|\mathcal{A}|\epsilon. \tag{8}$$

Proof. Let U be uniform over $\mathcal{A}$ and independent from B. We have

$$\begin{aligned}
\epsilon &\geqslant \Delta((U,B);(A,B)) \\
&= \frac{1}{2}\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} |P\left((U,B)=(a,b)\right) - P\left((A,B)=(a,b)\right)| \\
&= \frac{1}{2}\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} |P\left(U=a\right)\cdot P\left((U,B)=(a,b)|U=a\right) - \\
&\qquad\qquad P\left(A=a\right)\cdot P\left((A,B)=(a,b)|A=a\right)| \\
&\geqslant \frac{1}{2}\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} |P\left(U=a\right)\cdot P\left((U,B)=(a,b)|U=a\right) - \qquad (9) \\
&\qquad\qquad P\left(U=a\right)\cdot P\left((A,B)=(a,b)|A=a\right)| - \\
&\qquad\qquad |P\left(U=a\right)\cdot P\left((A,B)=(a,b)|A=a\right) - \\
&\qquad\qquad P\left(A=a\right)\cdot P\left((A,B)=(a,b)|A=a\right)| \\
&\geqslant \frac{1}{2}\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} P\left(U=a\right)\cdot |P\left((U,B)=(a,b)|A=a\right) - P\left((A,B)=(a,b)|U=a\right)| - \\
&\qquad \overbrace{\frac{1}{2}\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} P\left((A,B)=(a,b)|A=a\right)\cdot |P\left(U=a\right) - P\left(A=a\right)|}^{\leqslant d(A)\ \leqslant d(A|B)\ \leqslant \epsilon} \\
&= \frac{1}{2}\cdot\frac{1}{|\mathcal{A}|}\cdot\sum_{(a,b)\in\mathcal{A}\times\mathcal{B}} |P\left((A,B)=(a,b)|A=a\right) - P\left((U,B)=(a,b)|U=a\right)| - \epsilon \qquad (10) \\
&= \frac{1}{|\mathcal{A}|}\cdot\sum_{a\in\mathcal{A}}\frac{1}{2}\cdot\sum_{b\in\mathcal{B}} |P\left((A,B)=(a,b)|A=a\right) - P\left((U,B)=(a,b)|U=a\right)| - \epsilon \\
&= \frac{1}{|\mathcal{A}|}\cdot\sum_{a\in\mathcal{A}} \Delta(B|A=a\ ;\ B) - \epsilon
\end{aligned}$$

where (9) follows from the triangle inequality, and (10) comes from the fact that U is uniform on $\mathcal{A}$ and from the fact that $d(A) \leqslant d(A|B) \leqslant \epsilon$ (cf. Lemma 2). Therefore we obtain (8). This finishes the proof. □

2.1 Entropy

Entropy theory was introduced to measure a 'randomness' or 'chaos' of given object. Entropy is used in various fields of the modern science from the codes

theory(see. [12, 42]) to the medical screening tests(see. [39]). Let us begin with the definition from Shannon(see. [7]), let X be a random variable on $\mathcal{X}$, for $x \in \mathcal{X}$ define $X(x) = P(X = x)$. Shannon defined entropy (denoted $\mathbf{H}(\cdot)$) as

$$\mathbf{H}(X) = \sum_{x \in \mathcal{X}} X(x) \log \frac{1}{X(x)}$$

Let us remind a few basic properties of Shannon entropy, for C a constant variable and U uniformly distributed over $\mathcal{X}$

$$\mathbf{H}(C) = 0 \qquad\qquad \mathbf{H}(U) = \log |\mathcal{X}|$$

where $|\mathcal{X}|$ size(number of elements) of the set $\mathcal{X}$. For an arbitrary chosen, independent X, Y on $\mathcal{X}$ and a random variable Z_p defined as follows: with probability p choose element according to distribution X and with probability $1 - p$ choose element according to distribution Y such inequality holds:

$$\mathbf{H}(Z_p) \geqslant p\mathbf{H}(X) + (1 - p)\mathbf{H}(Y)$$

Last inequality shows why Shannon entropy is not good measure of randomness for some cryptographic purposes. Take $X \equiv 0$ and Y distributed uniformly over $\mathcal{X}$ then

$$\mathbf{H}(Z_p) \geqslant (1 - p) \log |\mathcal{X}|$$

take p equal 0.9 and $\mathcal{X}$ very large set, then unfortunately $\mathbf{H}(Z)$ is large but if it is used as cryptographic key we can easly break scheme simply assuming $Z = 0$ and we will be right with probability over 0.9. It is quite clear that we require other measure of randomness. Let X be random variable on $\mathcal{X}$, as earlier $X(x) = P(X = x)$, we define min-entropy(denoted as $\mathbf{H}_\infty(\cdot)$) as

$$\mathbf{H}_\infty(X) = \log \left(\frac{1}{\max_{x \in \mathcal{X}} X(x)} \right) = - \log \left(\max_{x \in \mathcal{X}} X(x) \right)$$

This measures how easly can we guess random variable. Notice that it has similar properties as Shanon entropy, for C constant variable and U uniformly distributed over $\mathcal{X}$

$$\mathbf{H}_\infty(C) = 0 \qquad \mathbf{H}_\infty(U) = \log |\mathcal{X}| \qquad \mathbf{H}_\infty(X) \leqslant \mathbf{H}(X).$$

It is easy to see that for $Z_{0.9}$ defined earlier $\mathbf{H}_\infty(Z_{0.9}) \leqslant -\log(0.9)$ which is close to 0.15 is small indicating we should not consider this random variable as 'truly random' for cryptographic purposes.

3 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 [38].

The idea behind a randomness extractors is as follows. Suppose we got a random variable X with some entropy but which is not uniform. For many cryptographic applications we require an uniformly random variables for example to use them as a secret key. Since achieving a perfectly uniform variable is not necessary (and hard) we will be satisfied with a variable that is indistinguishable from a uniform distribution, formally speaking we need a variable for which statistical distance from a uniform distribution is negligible. Therefore we need to somehow extract the randomness from X to get a shorter (a variable with smaller support) but uniformly distributed output. However it is easy to see that there does not exist a deterministic function f_k which for every random variable X with $\mathbf{H}_\infty(X) \geqslant k$ would achieve $f(X)$ being close to uniform even is output of f is very short compare to the length of X.

Lemma 5. *Let* $\log|\mathcal{X}| - 1 \geqslant k \geqslant 0$*, and* $\frac{1}{2} > \epsilon > 0$ *There does not exist deterministic function* $f : \mathcal{X} \to Z_2$ *such that for every random variable* X*, with* $\mathbf{H}_\infty(X) \geqslant k$ *we get* $\Delta((f(X)); U_{\mathcal{Z}}) \leqslant \epsilon$

Proof. Let $f^{-1}(0)$ denote all $x \in \mathcal{X}$ such that $f(x) = 0$. Let $U_{f^{-1}(0)}$ be random variable uniformly distributed on set $f^{-1}(0)$, analogously for $U_{f^{-1}(1)}$. Since $f^{-1}(0) \cup f^{-1}(1) = \mathcal{X}$ and $f^{-1}(0) \cap f^{-1}(1) = \emptyset$ hence that, for some $i \in \{0, 1\}$ we get $|f^{-1}(i)| \geqslant \frac{|\mathcal{X}|}{2}$ therefore $\mathbf{H}_\infty(U_{f^{-1}(i)}) \geqslant \log|\mathcal{X}| - 1$ while $f(U_{f^{-1}(i)}) = i$ therefore $d(f(U_{f^{-1}(i)})) = 1/2$. □

We can fix this problem in two ways, first by introducing *seeded extractors*, second by *two-source extractors. Seeded extractor* ext takes long random vector X, $\mathbf{H}_\infty(X) \geqslant k$ and short uniform seed S which is independent of X and as a result $\Delta(\mathsf{ext}(X, S); U) \leqslant \epsilon$. This is quite trivially achieved simply by choosing $\mathsf{ext}(X, S) = S$ so we also require output of ext to be much longer then length of S.

Let us focus on second solution that is 2-source extractors. We say that $\mathsf{ext} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is (k, ϵ)-2-source extractor if for every two independent random variables X on $\mathcal{X}$ and Y on $\mathcal{Y}$, such that $\mathbf{H}_\infty(X) \geqslant k$ and $\mathbf{H}_\infty(Y) \geqslant k$, result of $\mathsf{ext}(X, Y)$ is random variable Z which is not further then ϵ from

uniform distribution over $\mathcal{Z}$ formally $\Delta(Z, U_{\mathcal{Z}}) \leqslant \epsilon$, where $U_{\mathcal{Z}}$ is uniformly distributed over $\mathcal{Z}$.

There are some variations of 2-source extractors. We say that $\mathsf{ext} : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is strong (k, ϵ)-2-source extractor if for every two independent random variables X on $\mathcal{X}$ and Y on $\mathcal{Y}$, such that $\mathbf{H}_\infty(X) \geqslant k$ and $\mathbf{H}_\infty(Y) \geqslant k$, we get $\Delta\left((\mathsf{ext}(X, Y), X), (U_{\mathcal{Z}}, X)\right) \leqslant \epsilon$. Which basically means that even if we show adversary one of inputs (X or Y) he still can not distinguish between result of extractor and uniform distribution with probability significantly greater then $\frac{1}{2}$. In [38] (Theorem 5.1) we can find proof attributed to Boaz Barak that every 2-source extractor with sufficiently small error is also strong 2-source extractor with slightly worse parameters.

Below we introduce two new notions from our paper namely weak flexible 2-source extractor and strong flexible 2-source extractor. Similary to 2-source and strong 2-source extractors we prove that every weak flexible 2-source extractor is also strong flexible 2-source extractor or as we will call it later simply flexible extractor with worse parameters then original weak flexible extractor.

Our main theorem (Thm. 17) 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. [38]) 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 $\mathbf{H}_\infty(L) \geqslant m$ and $\mathbf{H}_\infty(R) \geqslant m$ we will require only that $\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) \geqslant k$ (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* (or: *flexible* (k, ϵ)*-extractor* for short) if for every $L \in \mathcal{L}$ and $R \in \mathcal{R}$ such that $\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) \geqslant k$ we have that $d(\mathsf{ext}(L, R)|L) \leqslant \epsilon$ and $d(\mathsf{ext}(L, R)|R) \leqslant \epsilon$. As it turns out an the inner product over finite fields is such an extractor.

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

3.1 Inner product as flexible extractor

Lemma 6. *For every finite fields* $\mathbb{F}$ *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 flexible* (k, ϵ)*-extractor for any* k *and* ϵ *such that*

$$\log(1/\epsilon) = \frac{k - (n+4)\log|\mathbb{F}|}{3} - 1. \tag{11}$$

Proof. First, it is easy to see, by inspection of the proof of Lemma 3.1 in [38] (cf. the line before Remark 3.2), that $\mathsf{ext}^n_{\mathbb{F}}$ is a *weak* flexible $\left(k, 2^{(n\log|\mathbb{F}| - k + \log|\mathbb{F}|)/2}\right)$-extractor, for any k. This obviously does not finish the proof, since we need our result to holds for the *strong* flexible extractors. Fortunately [38] provides also an argument, attributed there to Boaz Barak, that every weak extractor is also a strong (for slightly weaker parameters). Since in [38] this argument is stated for the classical definition of strong extractors, we need to check if it also holds for the flexible ones. Fortunately it turn out to be true, as shown below (what follows is copied almost verbatim from [38]).

Claim 1. *Let* $\mathsf{ext} : (\{0,1\}^N)^2 \to \{0,1\}^M$ *be a* weak *flexible* (K, ϵ)*-extractor, for* $K \geqslant N$*. Then for any* $K' \geqslant K$ *we have that* ext *is a* strong *flexible* (K', ϵ')*-extractor where* $\epsilon' = 2^M(\epsilon + 2^{K-K'})$.

Proof. Let X and Y be random variables such that $\mathbf{H}_\infty(X) + \mathbf{H}_\infty(Y) \geqslant K'$. Without loss of generality, assume that X and Y have flat distribution. Clearly, it suffices to show that

$$\sum_{y \in \mathsf{supp}(Y)} 2^{-\mathbf{H}_\infty(Y)} \Delta(\mathsf{ext}(X, y); U_M) \leqslant 2^M \left(2^{K-K'} + \epsilon\right), \tag{12}$$

where U_M is a uniform distribution over $\{0,1\}^M$. For any $z \in \{0,1\}^M$, define the set B_z of *bad y's for z* as follows:

$$B_z := \left\{ y : |P\left(\mathsf{ext}(X, y) = z\right) - 2^{-M}\right)| \geqslant \epsilon \right\}.$$

Now, we claim that for every z it holds that

$$|B_z| < 2^{\mathbf{H}_\infty(Y) - K' + K}. \tag{13}$$

(Observe that the exponent in (13) is non-negative, since $(\mathbf{H}_\infty(Y) - K') + K \geqslant -\mathbf{H}_\infty(X) + K \geqslant -N + K > 0$). To show (13) suppose it does not hold. Then the flat distribution on B_z and the variable X are two independent sources for which the extractor ext fails, because $\mathbf{H}_\infty(B_z) + \mathbf{H}_\infty(X) \geqslant \mathbf{H}_\infty(Y) - K' +$

$K + \mathbf{H}_\infty(X) \geqslant K$. This contradiction proves (13). Now let $B = \cup_z B_z$. We see that $|B| < 2^{\mathbf{H}_\infty(Y)-K'+K}2^M$. Therefore,

$$\begin{aligned}
& \sum_{y\in\mathsf{supp}(Y)} 2^{-\mathbf{H}_\infty(Y)}\Delta(\mathsf{ext}(X,y);U_M) \\
= & \sum_{y\in\mathsf{supp}(Y)\cap B} 2^{-\mathbf{H}_\infty(Y)}\Delta(\mathsf{ext}(X,y);U_M) + \sum_{y\in\mathsf{supp}(Y)\setminus B} 2^{-\mathbf{H}_\infty(Y)}\Delta(\mathsf{ext}(X,y);U_M) \\
\leqslant & \; 2^{-\mathbf{H}_\infty(Y)}2^{\mathbf{H}_\infty(Y)-K'+K+M} + \epsilon 2^M \\
= & \; 2^M(2^{K-K'}+\epsilon),
\end{aligned}$$

which, obviously, implies (12). □

Now take any k and set $M := \log|\mathbb{F}|$ and $N := n\log|\mathbb{F}|$ and $K' = k$ and $K := \frac{1}{3}(n+1)\log|\mathbb{F}| + \frac{2}{3}\cdot K'$ and $\epsilon := 2^{(n\log|\mathbb{F}|-K+\log|\mathbb{F}|)/2}$. From the remarks at the beginning of the proof we get that $\mathsf{ext}^n_{\mathbb{F}}$ is a weak flexible (K,ϵ)-extractor. Then, applying Claim 1 we get that it is also a strong flexible (K',ϵ')-extractor for

$$\begin{aligned}
& \epsilon' \\
= & \; 2^M\cdot\left(\epsilon + 2^{K-K'}\right) \\
= & \; |\mathbb{F}|\left(2^{\left(n\log|\mathbb{F}|-\frac{1}{3}\cdot(n+1)\log|\mathbb{F}|-\frac{2}{3}\cdot k+\log|\mathbb{F}|\right)/2} + 2^{\frac{1}{3}\cdot(n+1)\log|\mathbb{F}|-\frac{1}{3}\cdot k}\right) \\
= & \; 2^{\left(\frac{1}{3}\cdot n+\frac{4}{3}\cdot\log|\mathbb{F}|\right)-\frac{1}{3}\cdot k+1},
\end{aligned}$$

and hence

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

Thus the lemma 6 is proven. □

Note that since ϵ can be at most 1, hence (11) makes sense only if $k \geqslant 6 + 4|\mathbb{F}| + n\log|\mathbb{F}|$. 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 16 we will actually use in two different ways the fact that ext is an extractor. In one case (in the proof of Claim 3 within the proof of Lemma 16) 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 2) we will use the fact that $d(\mathsf{ext}(L,R)|R) \leqslant \epsilon$ 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 [28]. Hence Lemma 6 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 7, instead of considering bounded leakage from L and R separately (as it is done, e.g., in [15]). 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.

3.2 Other flexible extractor example

Another extractor that could be used to build a non-malleable code was introduced in [29] and is defined as follows. Take two independent random variables X, Y defined on $GF(2^n)$. Where $GF(2^n)$ denotes Galois field of polynomials of order $n-1$ over Z_2. Briefly speaking it is the field of a binary strings of length n where the addition is defined as a simple coordinate-wise addition over Z_2 while only for purpose of the multiplication binary strings are interpreted as a polynomials of order $n-1$ and the multiplication is simply a multiplication of a polynomials modulo some irreducible polynomial of order n. Holenstein in [29] proves that $\mathsf{ext}(X,Y) := (XY)_\lambda$ is a strong seeded extractor, where $(\cdot)_\lambda$ means trimming given sequence to the λ most significant bits. We will prove something stronger:

Lemma 7. $\mathsf{ext} : GF(2^n) \times GF(2^n) \to GF(2^\lambda)$, *defined as* $\mathsf{ext}(X,Y) = (X \cdot Y)_\lambda$ *is* $(k, 2^{\frac{n-k+2\lambda-2}{2}})$ *weak flexible 2-source extractor.*

Therefore by Claim 1 we will obtain that it is also strong flexible 2-source extractor. To begin the proof we need to introduce few definitions regarding

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

abstract harmonic analysis. All definitions and facts below are classical results proved in 1950-60. The proofs can be found in [37].

Let C^1 be a multiplicative group of $\{z \in C \mid |z| = 1\}$ where C denotes complex numbers and the multiplication is standard multiplication over complex numbers.
Let G be a abelian, locally compact group. For the purpose of this section we will assume that G is a finite discrete group. Reason for that is because we do not need a general approach, however this theorem is much more general and all facts and definitions below can be rewritten for any arbitrary abelian, locally compact group.
We will call $\psi : G \to C^1$ the *character* of group G if and only if ψ is homomorphism from G to C^1.
All characters of group G form Pontryagin dual group $\widehat{G}$ with group operation defined as follows let ψ, ϕ be characters of G then $(\psi \cdot \phi) : G \to C^1$, and $(\psi \cdot \phi)(g) = \psi(g)\phi(g)$. Neutral element of that group is *trivial* character (character constant, equal to 1). Characters have a following important geometric property that holds for any character ψ:

$$\sum_{g \in G} \psi(g) = \begin{cases} |G|, \text{ if } \psi \equiv 1 \\ 0, \text{ in other case} \end{cases} \tag{14}$$

The first case is obvious, let us focus on the second one. If ψ is non trivial then there exists $a \in G$ such that $\psi(a) \neq 1$. Using property of homomorphism we obtain:

$$\sum_{g \in G} \psi(g) = \sum_{g \in G} \psi(g + a) = \sum_{g \in G} (\psi(g)\psi(a)) = \psi(a) \sum_{g \in G} \psi(g)$$

Since $\psi(a) \neq 1$ sum above must be 0. We require a few more definitions regarding the norm metric. Let $f, h : G \to C$ be arbitrary functions, we define *norm metrics* and *inner product* as

$$||f||_1 = \sum_{g \in G} |f(g)|, \qquad ||f||_2 = \left(\sum_{g \in G} |f(g)|^2 \right)^{1/2},$$
$$||f||_\infty = max_{g \in G} |f(g)|, \qquad \langle f, h \rangle = \sum_{g \in G} f(g)\bar{h}(g).$$

Let $f : G \to C$ be an arbitrary function, we define $\widehat{f} : \widehat{G} \to C$ as follows

$$\widehat{f}(\psi) = \frac{1}{|G|} \sum_{g \in G} f(g)\psi(g)$$

We call function $\widehat{f}$ a Fourier-Stjelties transform of a function f. Let X be random variable on a group G, we will identify that random variable with its distribution $X : G \to [0,1]$, where $X(g) = P(X = g)$. Let us show few facts about distribution function.

$$||X(\cdot)||_\infty = 2^{-\mathbf{H}_\infty(X)} \qquad ||X(\cdot)||_2 \leqslant \sqrt{||X(\cdot)||_\infty}$$

The first equality follows from definition of min-entropy, second one is simply statement that average of real numbers is less or equal the largest number. Now for $f, h : G \to C$

$$\langle f, g\rangle \leqslant ||f||_2||g||_2 \qquad ||\widehat{f}||_2 = \frac{1}{|G|}||f||_2$$

The first one is the Cauchy-Schwartz inequality, the second is Parsaval identity stating that up to constant factor Fourier-Stjelties transform is isometric. Key idea behind using harmonic analysis in extractors theory is XOR-lemma.

Theorem 8. (Generalized Vazirani's XOR - Lemma, see also [38]) *Let G,H be finite abelian groups. Let X be a distribution on G with $|\mathbf{E}(\psi(X))| \leqslant \epsilon$ for every non-trivial character ψ of G and let U be the uniform distribution on G. Let $\sigma : G \to H$ be a function such that for every character ϕ of H, we have that*

$$||\widehat{\phi \circ \sigma}||_1 = \sum_{\psi \in \widehat{G}} |\frac{1}{|G|} \sum_{g \in G} \phi(\sigma(g))\psi(g)| \leqslant \tau$$

then

$$\Delta((\sigma(X)), (\sigma(U))) \leqslant \frac{1}{2}\epsilon\tau\sqrt{|H|}$$

Proof of this theorem can be found in [38]. The XOR-Lemma is crucial for the proof that $\sigma(X \cdot Y)$ is a flexible 2-source extractors.

Proof. (of Lemma 7) Let $X, Y \in GF(2^n)$ be random variables such that $\mathbf{H}_\infty(X) = k_x$ and $\mathbf{H}_\infty(Y) = k_y$ and $\sigma : GF(2^n) \to GF(2^\lambda)$ defined as a function that trims given sequence to λ most significant bits. At first we will

prove that $|\mathbf{E}(\psi(X \cdot Y))| \leqslant 2^{\frac{n-k_x-k_y}{2}}$ for every non-trivial character ψ of G.

$$|\mathbf{E}(\psi(X \cdot Y))| = \left| \sum_{i,j \in GF(2^n)} Y(j)X(i)\psi(j \cdot i) \right| =$$

$$\left| 2^n \sum_{j \in GF(2^n)} Y(j) \frac{1}{2^n} \sum_{i \in GF(2^n)} X(i)\psi(j \cdot i) \right| =$$

$$\left| 2^n \sum_{j \in GF(2^n)} Y(j)\widehat{X}(\gamma_j) \right| = \left| 2^n \langle \widehat{X}(\gamma_\cdot), Y(\cdot) \rangle \right| \leqslant \tag{15}$$

$$2^n ||\widehat{X}||_2 ||Y||_2 \leqslant 2^n \left(\frac{1}{\sqrt{2^n}} ||X||_2 ||Y||_2 \right) \leqslant \tag{16}$$

$$2^{\frac{n-k_x-k_y}{2}} \tag{17}$$

Where, (15) follows from fact that $\gamma_j(x) := \psi(j \cdot x)$ is character of $GF(2^n)$, (16) follows from the Cauchy-Schwartz inequality and Parseval identity, and (17) we obtain from $||X||_2 \leqslant \sqrt{||X||_\infty} = 2^{-\mathbf{H}_\infty(X)/2}$. Now to use XOR-Lemma for every ϕ character of $GF(2^\lambda)$ we need to bound

$$||\widehat{\phi \circ \sigma}||_1 = \sum_{\psi \in \widehat{GF(2^n)}} \left| \frac{1}{2^n} \sum_{g \in GF(2^n)} \phi(\sigma(g))\psi(g) \right|$$

At first let us notice that σ defined earlier is homomorphism from $GF(2^n)$ to $GF(2^\lambda)$. Thus $\phi \circ \sigma$ is a homomorphism from $GF(2^n)$ to C^1 therefore it is a character of $GF(2^n)$. Moreover $\phi(\sigma(g))\psi(g)$ is also a character and since characters form group for every ϕ there exist only one character ψ for which $\phi(\sigma(g))\psi(g) \equiv 1$. By (14) we get $\sum_{g \in GF(2^n)} \phi(\sigma(g))\psi(g) = 0$ for all characters but one, for that last character this sum is equal $|G|$ which ends the proof that

$$||\widehat{\phi \circ \sigma}||_1 = \sum_{\psi \in \widehat{GF(2^n)}} \left| \frac{1}{2^n} \sum_{g \in GF(2^n)} \phi(\sigma(g))\psi(g) \right| = 1.$$

Therefore by applying the XOR-Lemma we obtain that for random variables $X, Y \in GF(2^n)$ such that $\mathbf{H}_\infty(X) + \mathbf{H}_\infty(Y) = k$

$$\Delta((XY)_\lambda, U) \leqslant \frac{1}{2}\, 2^{\frac{n-k}{2}} |GF(2^\lambda)| = 2^{\frac{n-k+2\lambda-2}{2}}.$$

This ends the proof that $\mathsf{ext}(X, Y) = (X \cdot Y)_\lambda$ is flexible $(k, 2^{\frac{n-k+2\lambda-2}{2}})$ 2-source extractor. □

4 Inner product, leakage and Leftover Hash Lemma

Let us begin with the introduction to the area of leakage resilient cryptography. In the classical cryptography one usually assumes that the device is black-boxed, which means that that any adversary attempting to break it can only use its input-output interface without access to its internal data (such as, e.g., the secret key). It turns out that this model does not correspond well to the real life attacks as in practice the adversary can get such information, via number of so called *side channels* based on passive measurements such as running-time, electromagnetic radiation, power consumption (see e.g. [36]). A recent trend in cryptography is to extend the black-box attacks models by assuming that the adversary can get some partial information about device's internal data. As, obviously, giving to the adversary the complete secret state of the device ruins any security, every such model needs to somehow limit the amount of the information that the adversary learns about the secret state. A typical approach is to assume that the adversary can learn any function $f : \{0,1\}^* \to \{0,1\}^*$ of the state, subject to the restriction that the output of f is much shorter then the length of the state (and hence inevitably f „forgets" some information about its input). In this dissertation we are particularly interested in variant of this model (called the split-state model) where it is assumed that the secret state of the device is split into two parts L and R and the adversary can learn some bounded information independently from L and R. In the most basic form this means that the adversary can chose two functions f and g such that $f(L) \ll |L|$ and $g(R) \ll |R|$ and learn $f(L)$ and $g(R)$. In a more general, *adaptive* case the adversary can actually chose a sequence of functions f_i and g_i in an *adaptive* way (i.e. his choice of each f_i and g_i can depend on what he learned before). For the formal definitions of these notions see Sections 4.2 and Section 4.3. As highlighted in the introduction the very important method for studying leakage resilience are the randomness extractors. In the next two sections we discuss the technical tools that are useful in this context.

4.1 Leftover Hash Lemma and non-adaptive leakage

Let us begin with definition from [3]. A family $\mathcal{H}$ of (deterministic) functions $h : \mathcal{X} \to \{0,1\}^v$ is a called *p-universal hash family* (on space $\mathcal{X}$), if for any $x_1 \neq x_2 \in \mathcal{X}$ we have $P_{h \leftarrow \mathcal{H}}[h(x_1) = h(x_2)] \leqslant p$. When $p = \frac{1}{2^v}$, we say that $\mathcal{H}$ is *universal*. We can finally state the *Leftover Hash Lemma (LHL).*

Lemma 9. *(Leftover-Hash Lemma) Assume that the family $\mathcal{H}$ of functions $h : \mathcal{X} \to \{0,1\}^v$ is $\frac{1+\gamma}{2^v}$-*universal hash family*. Then the extractor $\mathsf{ext}(x;h) =$*

$h(x)$, where h is uniform over $\mathcal{H}$, is an (m, ϵ)-extractor, where $\epsilon = \frac{1}{2}\sqrt{\gamma + \frac{1}{2^{m-v}}}$.

Proof of this lemma can be found in [3]. Let $x, y \in \mathbb{F}^n$, now define function $h_y(x) = \langle x, y\rangle$. It is not hard to see that family $\mathcal{H} = \{h_y | y \in \mathbb{F}^n\}$ is a *universal hash family*. Therefore we obtain that for X, Y independent such that $\mathbf{H}_\infty(X) \geqslant m$ and Y is uniform:

$$\Delta[(\langle X, Y\rangle, Y); (U_{\mathbb{F}}, Y)] \leqslant \frac{1}{2}\sqrt{\frac{1}{2^{m-\log|\mathbb{F}|}}}$$

This result can be translated to leakage, simply by choosing $f(X) \equiv 0$ and $g(Y) = Y$. Of course if X a priori has high min-entropy then we can choose f such that $P_{x\leftarrow X}(\mathbf{H}_\infty(X|f(X) = x) \geqslant m) \geqslant 1-\epsilon$, then LHL result translates to

$$P\left(\Delta[(\langle X, Y\rangle, f(X), Y); (U_{\mathbb{F}}, f(X), Y)] \leqslant \frac{1}{2}\sqrt{\frac{1}{2^{m-\log|\mathbb{F}|}}}\right) \geqslant 1 - \epsilon$$

which satisfies us when ϵ is negligible factor. We will show how to relax assumption that Y has to be uniform and to do that we will use *flexibility* notion. In Theorem 6 we showed that for $X, Y \in \mathbb{F}^n$ we get that $\mathsf{ext}(X, Y) = \langle X, Y\rangle$ is $(k, 2^{-\left[\frac{k-(n+4)\log|\mathbb{F}|}{3}-1\right]})$ -strong flexible extractor. Therefore we obtain

Theorem 10. *For any X and Y independent random variables on $\mathbb{F}^n$ such that $\mathbf{H}_\infty(Y) \geqslant k-m$ and for any $f : \mathbb{F}^n \to \mathcal{G}$ such that $P_{x\leftarrow f(U_{\mathbb{F}^n})}(\mathbf{H}_\infty(X|f(X) = x) \geqslant m) \geqslant 1 - \epsilon$. We get:*

$$P\left(\Delta[(\langle X, Y\rangle, f(X), Y); (U_{\mathbb{F}}, f(X), Y)] \leqslant 2^{-\left[\frac{k-(n+4)\log|\mathbb{F}|}{3}-1\right]}\right) \geqslant 1 - \epsilon$$

therefore:

$$\Delta[(\langle X, Y\rangle, f(X), Y); (U_{\mathbb{F}}, f(X), Y)] \leqslant 2^{-\left[\frac{k-(n+4)\log|\mathbb{F}|}{3}-1\right]} + \epsilon$$

Natural question rises how to choose G such that $P_{x\leftarrow X}(\mathbf{H}_\infty(X|f(X) = x) \geqslant m) \geqslant 1 - \epsilon$, to answer that question we present this lemma which is the generalization of Lemma 5 from [15].

Lemma 11. *For every X be random variable on $\mathcal{X}$, such that $\mathbf{H}_\infty(X) = k$, and for $f : \mathcal{X} \to \{0,1\}^\lambda$ then*

$$P_{y\leftarrow f(U_{\mathcal{X}})}\left(\mathbf{H}_\infty(X|f(X) = y) \leqslant m\right) \leqslant 2^{-k+\lambda+m}.$$

Proof. Let us define set $\mathcal{A}_y \subset \mathcal{X}$ as follows:

$$\mathcal{A}_y = \{x \in \mathcal{X} | f(x) = y\}$$

We will say that set $\mathcal{A}_y$ is *small* if $P(X \in \mathcal{A}_y) \leqslant 2^{m-k}$. First observe that $\mathbf{H}_\infty(X|X \notin \text{ small set}) \geqslant m$ which follows straightforward from conditional probability. Notice that there is at most 2^λ sets that are small. Therefore $P(X \in \text{ small set}) \leqslant 2^{-k+m+\lambda}$ which ends the proof. □

4.2 Adaptive leakage

Theorem 12. *Let* $\mathsf{ext} : \mathcal{X} \times \mathcal{X} \to \mathcal{G}$ *be flexible* $(k, \epsilon)-$*extractor. Let* X *and* Y *be independent random variables such that* $\mathbf{H}_\infty(X) = k_x$ *and* $\mathbf{H}_\infty(Y) = k_y$. *For any adaptive sequence of functions* $f_i : \mathcal{X} \to \{0,1\}^{a_i}$ *and* $g_i : \mathcal{X} \to \{0,1\}^{b_i}$ *(where i'th function can depend on results of* $i-1$ *previous functions* f_i *and* g_i*), such that* $\lambda_x + \lambda_y \leqslant \lambda$ *where* λ *is parameter and* $\sum_i a_i = \lambda_x$ *and* $\sum_i b_i = \lambda_y$ *we get:*

$$d(\mathsf{ext}(X,Y)|\mathsf{view}_{f,g}) \leqslant \epsilon + 2^{-k_x-k_y+k+\lambda}$$

where $\mathsf{view}_{f,g} = (f_1(X), g_1(Y), f_2(X), g_2(Y)...)$.

To prove that theorem we require lemma from [15] (Lemma 4.).

Lemma 13. *Let* X *and* Y *be independent random variables then*

$$I(X, Y|\mathsf{view}_{f,g}) = 0$$

where I *denotes Shannon's information, and* $\mathsf{view}_{f,g}$ *is defined same way as above.*

Now we can prove Theorem 12.

Proof. Let us notice that in order to prove Th. 12 by Lemma 13 its sufficient to estimate probability

$$P_c\left(\mathbf{H}_\infty(X|\mathsf{view}_{f,g} = c) + \mathbf{H}_\infty(Y|\mathsf{view}_{f,g} = c) \geqslant k\right)$$

By Lemma 11 we know that

$$P_c\left(\mathbf{H}_\infty(X|\mathsf{view}_{f,g} = c) \leqslant m_x\right) \leqslant 2^{-k_x+m_x+\lambda_x}$$

same for

$$P_c\left(\mathbf{H}_\infty(Y|\mathsf{view}_{f,g} = c) \leqslant m_y\right) \leqslant 2^{-k_y+m_y+\lambda_y}.$$

Therefore

$$P_c\left(\mathbf{H}_\infty(X|\mathsf{view}_{f,g}=c) \leqslant k_x-\lambda_x-\alpha_x\right) \leqslant 2^{-\alpha_x}$$

and

$$P_c\left(\mathbf{H}_\infty(Y|\mathsf{view}_{f,g}=c) \leqslant k_y-\lambda_y-\alpha_y\right) \leqslant 2^{-\alpha_y}.$$

Thus that by $\lambda_x+\lambda_y \leqslant \lambda$ and by choosing $\alpha=\alpha_x+\alpha_y$ we get

$$P_c\left(\mathbf{H}_\infty(X|\mathsf{view}_{f,g}=c)+\mathbf{H}_\infty(Y|\mathsf{view}_{f,g}=c) \geqslant k_x+k_y-\lambda-\alpha\right) \geqslant 1-2^{-\alpha}$$

now take $\alpha=-k+k_x+k_y-\lambda$ we get

$$P_c\left(\mathbf{H}_\infty(X|\mathsf{view}_{f,g}=c)+\mathbf{H}_\infty(Y|\mathsf{view}_{f,g}=c) \geqslant k\right) \geqslant 1-2^{-k_x-k_y+k+\lambda}$$

Therefore since ext is flexible $(k,\epsilon)-$extractor

$$P_c\left(d(\mathsf{ext}(X,Y)|\mathsf{view}_{f,g}=c) \leqslant \epsilon\right) \geqslant 1-2^{-k_x-k_y+k+\lambda}$$

from that we get Theorem 12. □

4.3 Leakage-resilient storage in the split-state model

In [15] notion of the *leakage-resilient storage* was introduced, for purpose of this dissertation we will generalize the definitions from [15]. For the reasons explained in the introduction to this section we are interested in creating storage schemes that are resilient to leakage of the information. Let $\mathsf{ext}: \mathcal{X}\times\mathcal{X}\to\mathbb{F}$ be a (k,ϵ)- flexible extractor. Let L and R be a independent random variables such that $\mathbf{H}_\infty(L)+\mathbf{H}_\infty(R)>k$. Now let us define a $((k,\epsilon)-\mathsf{ext},L,R)$-scheme as a pair of functions $\mathsf{Enc}:\mathbb{F}\to\mathcal{X}\times\mathcal{X}$ and $\mathsf{Dec}:\mathcal{X}\times\mathcal{X}\to\mathbb{F}$ defined as follows:

$$\mathsf{Dec}(l,r)=\mathsf{ext}(l,r)$$
$$\mathsf{Enc}(m)=(l,r) \text{ such that } l\leftarrow L, r\leftarrow R \text{ and } \mathsf{ext}(l,r)=m.$$

Now let us define the (λ,t)-*split-state model adversary* (or in short the $(\lambda,t)-$SSM adversary) similarly as in [15]. Assume memory of the device is split in two parts L,R, we execute t-times following procedure: for $i=1,2,...,t$ the adversary chooses $f_i:\mathcal{X}\to\{0,1\}^{a_i}$ and $g_i:\mathcal{X}\to\{0,1\}^{b_i}$ and then learns $f_i(L)$ and $g_i(R)$. The adversary can choose any a_i and b_i such that $\sum_i a_i+b_i<\lambda$. We will denote the vector $(f_1(L),g_1(R),f_2(L),...,g_t(R))$ of the outputs reviled to the adversary $\mathcal{A}$ as $\mathsf{view}_{\mathcal{A}}(L,R)$.

We will say that the $((k,\epsilon)-\mathsf{ext},L,R)$-scheme is (λ,t,δ)-weak secure in the split-state model if for every (λ,t)-SSM adversary $\mathcal{A}$ following condition is fulfilled

$$d(\mathsf{ext}(L,R)|\mathsf{view}_{\mathcal{A}}(L,R)) \leqslant \delta.$$

Now let us recall a definition from [15] and rewrite it for the split-state model. We say that the scheme $(\mathsf{Enc}, \mathsf{Dec})$ is $(\lambda, t, \delta)-$secure if for any two messages m_0 and m_1 and for every (λ, t)-SSM adversary following condition is fulfilled:

$$\Delta\left(\mathsf{view}_{\mathcal{A}}(\mathsf{Enc}(m_0)); \mathsf{view}_{\mathcal{A}}(\mathsf{Enc}(m_1))\right) \leqslant \delta$$

Informally speaking that means that for any two messages m_0 and m_1 we can not distinguish their encodings with a probability greater then $1/2 + \delta/2$ even if additional information leaked.

Theorem 14. *If the $((k, \epsilon) - \mathsf{ext}, L, R)$-scheme is (λ, t, δ)-weak secure then it is $(\lambda, t, 4|\mathbb{F}| \cdot \delta)-$secure.*

Proof. Let the $((k, \epsilon) - \mathsf{ext}, L, R)$-scheme be (λ, t, δ)-weak secure. Thus for every (λ, t)-SSM adversary $\mathcal{A}$ we get

$$d(\mathsf{ext}(L, R)|\mathsf{view}_{\mathcal{A}}(L, R)) \leqslant \delta.$$

By Lemma 4 we get that for every $m \in \mathbb{F}$

$$\Delta\left((\mathsf{view}_{\mathcal{A}}(L, R)|\mathsf{ext}(L, R) = m); \mathsf{view}_{\mathcal{A}}(L, R)\right) \leqslant 2|\mathbb{F}| \cdot \delta.$$

Therefore by triangle inequality, for every $m_0, m_1 \in \mathbb{F}$ we get

$$\Delta\left((\mathsf{view}_{\mathcal{A}}(L, R)|\mathsf{ext}(L, R) = m_0); (\mathsf{view}_{\mathcal{A}}(L, R)|\mathsf{ext}(L, R) = m_1)\right) \leqslant 4|\mathbb{F}| \cdot \delta$$

thus

$$\Delta\left(\mathsf{view}_{\mathcal{A}}(\mathsf{Enc}(m_0)); \mathsf{view}_{\mathcal{A}}(\mathsf{Enc}(m_1))\right) \leqslant 4|\mathbb{F}| \cdot \delta.$$

□

In Theorem 12 we showed that the $((k, \epsilon) - \mathsf{ext}, L, R)$- scheme is $(\lambda, \infty, \epsilon + 2^{-\mathbf{H}_\infty(L) - \mathbf{H}_\infty(R) + k + \lambda})$-weak secure thus Theorem 14 we get that schemes based on the flexible extractors are $(\lambda, \infty, 4|\mathbb{F}|(\epsilon + 2^{-\mathbf{H}_\infty(L) - \mathbf{H}_\infty(R) + k + \lambda}))$- secure against leakage.

5 Definition of the non-malleable codes and equivalence to 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

$$\mathsf{Tamper}_m^F \approx_\epsilon \left\{ \begin{array}{c} d \leftarrow D^F \\ \text{if } d = \mathsf{same}^* \text{ then output } m \\ \text{otherwise output } d. \end{array} \right\} \tag{18}$$

The idea behind the "$\perp$" symbols is that is 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. 1.1, 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 15. *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*

$$P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(B))\right) \neq B\right) \leqslant \frac{1}{2} + \epsilon. \tag{19}$$

Proof. First assume that $(\mathsf{Enc}, \mathsf{Dec})$ is ϵ-non-malleable and show that (19) holds. Fix any $F : \mathcal{X} \to \mathcal{X}$. Since $(\mathsf{Enc}, \mathsf{Dec})$ is ϵ-non-malleable, hence there

exists a distribution D^F such that (18) holds. Therefore (cf. Lemma 1) we have

$$\epsilon \geqslant |P\left(\mathsf{Tamper}_0^F = 1\right) - P\left(D^F = 1\right)| \tag{20}$$

and

$$\epsilon \geqslant |P\left(\mathsf{Tamper}_1^F = 0\right) - P\left(D^F = 0\right)|. \tag{21}$$

Adding sidewise (20) and (21) we obtain

$$\begin{aligned} 2\epsilon &\geqslant |P\left(\mathsf{Tamper}_0^F = 1\right) - P\left(D^F = 1\right)| + && (22)\\ &\quad |P\left(\mathsf{Tamper}_1^F = 0\right) - P\left(D^F = 0\right)| \\ &\geqslant \Big|P\left(\mathsf{Tamper}_0^F = 1\right) + P\left(\mathsf{Tamper}_1^F = 0\right) - && (23)\\ &\quad \left(P\left(D^F = 1\right) + P\left(D^F = 0\right)\right)\Big|, && (24) \end{aligned}$$

where (24) comes from the triangle inequality. Since obviously $P\left(D^F = 1\right) + P\left(D^F = 0\right) \leqslant 1$, hence (24) implies that

$$1 + 2\epsilon \geqslant P\left(\mathsf{Tamper}_0^F = 1\right) + P\left(\mathsf{Tamper}_1^F = 0\right). \tag{25}$$

On the other hand it is easy to see that

$$\begin{aligned} & P\left(\mathsf{Tamper}_0^F = 1\right) + P\left(\mathsf{Tamper}_1^F = 0\right) \\ = \;& P\left(F(\mathsf{Enc}(B)) \neq B | B = 0\right) + P\left(F(\mathsf{Enc}(B)) \neq B | B = 1\right) \\ = \;& \frac{1}{2} \cdot P\left(F(\mathsf{Enc}(B)) \neq B\right), && (26) \end{aligned}$$

where (26) comes from the fact that B has uniform distribution over $\{0, 1\}$. Obviously (25) and (26) imply (19). Hence this part of the lemma is proven. To show the opposite direction of the lemma assume now that (19) holds. We will show that $(\mathsf{Enc}, \mathsf{Dec})$ is ϵ-non-malleable. Again, fix any $F : \mathcal{X} \to \mathcal{X}$. Denote

$$\begin{aligned} \epsilon' := \frac{1}{2} \cdot \max(0, & P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(1))\right) = 0\right) + && (27)\\ & P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(0))\right) = 1\right) - 1). && (28) \end{aligned}$$

Clearly from (19) we get that $\epsilon' \leqslant \epsilon$. Now, define D^F as follows

$$D^F := \begin{cases} 0 & \text{with prob. } P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(1))\right) = 0\right) - \epsilon' \\ 1 & \text{with prob. } P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(0))\right) = 1\right) - \epsilon' \\ \mathsf{same}^* & \text{otherwise.} \end{cases}$$

It is easy to verify that the probabilities above are non-negative, and, from the definition of ϵ' they sum up to 1. Hence the distribution D^F is defined correctly. Now look at the experiment (18). It is obvious that for $b = 1$ we have

$$P\left(\mathsf{Tamper}_1^F = 0\right) = P\left(\mathsf{Dec}\left(F(\mathsf{Enc}(1))\right) = 0\right) - \epsilon'.$$

Hence in this case $\mathsf{Tamper}_1^F \approx_{\epsilon'} \mathsf{Dec}\left(F(\mathsf{Enc}(1))\right)$. By a symmetric argument we also get $\mathsf{Tamper}_0^F \approx_{\epsilon'} \mathsf{Dec}\left(F(\mathsf{Enc}(0))\right)$. Since $\epsilon' \leqslant \epsilon$ this implies that $(\mathsf{Enc}, \mathsf{Dec})$ is ϵ-non-malleable. □ □

In this paper we are interested in the split-state codes. A *split-state code* is an 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 ϵ-*non-malleable* if it is ϵ-non-malleable with respect to a family of *all* functions $\mathsf{Mall}^{f,g}$ defined as $\mathsf{Mall}^{f,g}(L, R) = (f(L), g(R))$.

6 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 let us begin with 2-out-of-2 secret sharing definition, let S be secret we want to share between 2 parties in such way that neither of those can reconstruct secret (or something related to it) without knowledge of second share. Formaly we say that scheme is 2-out-of-2 secret sharing if distribution of $(S|\text{left share}(S) = L)$ is identical as distribution of S (same for right share R). Analogously scheme is $\epsilon-$2-out-of-2 secret sharing if $\Delta\left((S|\text{left share}(S) = L); (S)\right) \leqslant \epsilon$ and $\Delta\left((S|\text{right share}(S) = R); (S)\right) \leqslant \epsilon$.

It is easy to see that any such code $(\mathsf{Enc}, \mathsf{Dec})$ needs to be a ϵ-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 $M \in \{0, 1\}$. 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 $1/2$. Precisely speaking assume that $\mathcal{A} : \mathcal{L} \to Z_2$ is program that gets vector $L \in \mathcal{L}$ and makes a guess what secret is coded using L(symmetric argument works for R). Now assume our scheme is not $(2\delta)-$secret-sharing scheme that means $\Delta\left((S|\text{left share}(S) = L); (S)\right) \geqslant 2\delta$ (without loss of generality let us assume scheme is not secret sharing with respect to left share). Then there exist such program $\mathcal{A}$ that $P\left(B \leftarrow \{0, 1\}; L, R \leftarrow \mathsf{Enc}(B); \mathcal{A}(L) = B\right) \geqslant \frac{1}{2} + \delta$. Adversary has access to such program $\mathcal{A}$ and can chose any three vectors $l_0, l_1 \in \mathcal{L}, r \in \mathcal{R}$ such that $\mathsf{Dec}(l_0, r) = 0$ and $\mathsf{Dec}(l_1, r) = 1$ (he can do that

because scheme itself is not secret). Then he chose function g to be constant $g \equiv r$, and defines f with help of program $\mathcal{A}$, precisely $f(L) = l_{\mathcal{A}(L)+1}$. Let us calculate $P\left(\mathsf{Dec}\left(\mathsf{Mall}^{f,g}(\mathsf{Enc}(B))\right) \neq B\right)$.

$$P\left(\mathsf{Dec}\left(\mathsf{Mall}^{f,g}(\mathsf{Enc}(B))\right) \neq B\right) = P\left(L, R \leftarrow \mathsf{Enc}(B), \mathcal{A}(L) = B\right) \geqslant \frac{1}{2} + \delta.$$

Therefor by lemma 15 we get that such scheme can not be $\epsilon-$non-malleable if $\delta > \epsilon$.

In general case of any arbitrary message space $\mathcal{M}$ it is also true that scheme needs to be $2\epsilon-$2-out-of-2 secret sharing in order to be $\epsilon-$non-malleable. The idea is very similar to idea for just $\{0,1\}$ messages. Assume scheme is not secret sharing, then there exists two messages $m_0, m_1 \in \mathcal{M}$ for which scheme is not secret sharing (now we assume scheme is only sharing m_0 or m_1) and then by earlier reasoning we get that coding scheme can not be non-malleable.

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 $a \geqslant 2$) 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 $\mathsf{ext} : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$ 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. [15]). 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. 3 our construction uses a flexible two-source extractor $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ in a black-box way (later we show how it instantiate it with an inner product extractor, cf. Thm. 18). 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 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 is $\tilde{L}$ 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

$$\text{for } \tilde{L} \leftarrow \mathcal{L} \text{ and } \tilde{R} \leftarrow \mathcal{R} \text{ we have } d(\mathsf{ext}(\tilde{L},\tilde{R})) = 0. \tag{29}$$

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 (29), 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:

$$P\left((L,R) = (\ell,r)\right) = (|\mathcal{C}|-1) \cdot P\left((L,R) = (\ell',r')\right). \tag{30}$$

It is also straightforward to see that every extractor can be easily converted to an extractor that satisfies (29)[3]. Lemma 16 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. 7) 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. (32)) 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 (31) 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. 3). If one is not interested in leakages then one can read Lemma 16 and its proof assuming that $\mathcal{L}' \times \mathcal{R}' = \mathcal{L} \times \mathcal{R}$. Lemma 16 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 7, where it is done in a general way, also taking into account leakages.

Lemma 16. *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 flexible (k,ϵ)-extractor that satisfies (29), where, for some parameter δ we have:*

$$k = \frac{2}{3} \cdot (\log|\mathcal{L}'| + \log|\mathcal{R}'|) - \frac{2}{3} \cdot \log(1/\delta). \tag{31}$$

[3] The inner-product extractor satisfies (29) 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 (29).

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*

$$P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R)) \neq B \mid (L,R) \in (\mathcal{L}',\mathcal{R}')\right) \leqslant \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon), \tag{32}$$

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

Proof. Before presenting the main proof idea let us start with some simple observations. First, clearly it is enough to show (32), as then the fact that $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ is $\left(\frac{1}{2}\cdot|\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon)\right)$-non-malleable can be obtained easily by assuming that $\mathcal{L}' \times \mathcal{R}' = \mathcal{L} \times \mathcal{R}$ and applying Lemma 15. Observe also that (32) implies that $\log|\mathcal{L}'| + \log|\mathcal{R}'| \geqslant k$, 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

$$d(\mathsf{ext}(\tilde{L},\tilde{R})) \leqslant \epsilon. \tag{33}$$

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 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 an 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 of $\mathcal{L}_{\mathsf{FFC}} \subseteq \mathcal{L}'$ 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

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

and

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

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 that essentially formalizes the intuition outlined above, by showing that if either $L \in \mathcal{L}_{\mathsf{FFC}}$ or $R \in \mathcal{R}_{\mathsf{FFC}}$ then (f, g) cannot succeed in negating B.

Claim 2. *Let $B \leftarrow \{0,1\}$ and $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(B)$. Then:*

$$P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R)) \neq B \mid L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}}\right) \leqslant \frac{1}{2} + \frac{1}{2} \cdot |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon. \tag{34}$$

Proof. We will actually prove only that

$$P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right) \leqslant \frac{1}{2} + \frac{1}{2} \cdot |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon. \tag{35}$$

This will suffice, as, obviously, because of the symmetry of L and R, the following inequality can be proven analogously:

$$P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \mid R \notin \mathcal{R}_{\mathsf{FFC}}\right) \leqslant \frac{1}{2} + \frac{1}{2} \cdot |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon, \tag{36}$$

and (35) and (36) together imply (34). Let $(\tilde{L}, \tilde{R})$ be chosen uniformly at random from $\mathcal{L}' \times \mathcal{R}'$. From the definition of $\mathcal{L}_{\mathsf{FFC}}$ for every $y \notin \overrightarrow{f}(\mathcal{L}_{\mathsf{FFC}})$ we have that

$$\mathbf{H}_\infty(\tilde{L} \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}} \wedge f(\tilde{L}) = y) \geqslant k - \log|\mathcal{R}'|. \tag{37}$$

Since $\tilde{R}$ is uniform and independent from $\tilde{L}$ hence we also have that

$$\mathbf{H}_\infty(\tilde{R} \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}} \wedge f(\tilde{L}) = y) = \mathbf{H}_\infty(\tilde{R}) = \log|\mathcal{R}'|, \tag{38}$$

and, moreover, clearly $\tilde{L}$ and $\tilde{R}$ are independent conditioned on the event $(\tilde{L} \notin \mathcal{L}_{\mathsf{FFC}} \wedge f(\tilde{L}) = y)$. Since ext is a flexible (k, ϵ)-extractor, hence we get:

$$d\left(\mathsf{ext}(\tilde{L}, \tilde{R}) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}} \wedge f(\tilde{L}) = y,\ \tilde{R}\right) \leqslant \epsilon, \tag{39}$$

which, since we quantified over all y's such that $y \notin \overrightarrow{f}(\mathcal{L}_{\mathsf{FFC}})$, clearly implies that

$$d\left(\mathsf{ext}(\tilde{L},\tilde{R}) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}},\ f(\tilde{L}),\ \tilde{R}\right) \leqslant \epsilon. \tag{40}$$

Basically, what it means is: once it happened that $\tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}$, then $\mathsf{ext}(\tilde{L},\tilde{R})$ is close to uniform even if we give to the adversary $f(\tilde{L})$ and the *entire* $\tilde{R}$. Note that in this argument we implicitly used ext as a strong seeded extractor, which we are allowed to do because of its flexibility (cf. Sect. 3). Since $\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right)$ is clearly a function of $(f(\tilde{L},\tilde{R}))$ hence, by Lemma 2, Eq. (40) implies that

$$d\left(\mathsf{ext}(\tilde{L},\tilde{R}) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}},\ \mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right)\right) \leqslant \epsilon. \tag{41}$$

This is, of course, still very far from what we need, for several reasons, one of them being that we want to reason about the distance of $\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right)$ from uniformity, and in (41) this term appears on the right-hand-side of the condition symbol "|". Fortunately, we can apply now Lemma 4 to "invert" (41) obtaining that

$$\begin{aligned} 2|\mathcal{C}|\epsilon \geqslant \Delta\Big(& \underbrace{\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}} \wedge \mathsf{ext}(\tilde{L},\tilde{R}) = c\right)}_{(*)}; \\ & \underbrace{\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)}_{(**)}\Big) \end{aligned} \tag{42}$$

From the construction of $\mathsf{D}^c_{\mathsf{ext}}$ it is easy to see that the conditional distribution $(*)$ is equal to the distribution of $\mathsf{D}^c_{\mathsf{ext}}\left(f(L),g(R)\right)$ conditioned on the event that $B = 0$ and $L \notin \mathcal{L}_{\mathsf{FFC}}$. We now show that $(**)$ is close the the distribution of $\mathsf{D}^c_{\mathsf{ext}}\left(f(L),g(R)\right)$ conditioned on the event that $B = 1$ and $L \notin \mathcal{L}_{\mathsf{FFC}}$.

$$\begin{aligned} & \Delta\Big(\overbrace{\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)}^{(**)}\Big); \\ & \left(\mathsf{D}^c_{\mathsf{ext}}\left(f(L),g(R)\right) | B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right)\Big) \\ = \ & \Delta\Big(\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right) \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)\Big); \\ & \left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\tilde{L}),g(\tilde{R})\right) | \mathsf{ext}(\tilde{L},\tilde{R}) \neq c \wedge \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)\Big) \\ \leqslant \ & \left(1 - P\left(\mathsf{ext}(\tilde{L},\tilde{R}) \neq c\right)\right) \qquad (43) \\ = \ & P\left(\mathsf{ext}(\tilde{L},\tilde{R}) = c\right) \\ \leqslant \ & |\mathcal{C}|^{-1} + \epsilon, \qquad (44) \end{aligned}$$

where in (43) we used Lemma 3, and in (44) we used (33). Hence, applying the triangle inequality to (42) and (44) we obtain

$$\begin{aligned} & 2|\mathcal{C}|\epsilon + |\mathcal{C}|^{-1} + \epsilon \\ \geqslant\ & \Delta((\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) \mid B = 0 \wedge L \notin \mathcal{L}_{\mathsf{FFC}})\ ; \\ & (\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) \mid B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}})). \end{aligned} \tag{45}$$

Now observe that $\mathsf{D}^c_{\mathsf{ext}}$ takes values in a binary set, and hence, by Lemma 1 the right-hand-side of (45) is equal to

$$\begin{aligned} & |P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 0 \mid B = 0 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right) - \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 0 \mid B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right)| \\ = \ & |\left(1 - P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 1 \mid B = 0 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right)\right) - \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 0 \mid B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right)|, \end{aligned}$$

and therefore

$$\begin{aligned} & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 1 \mid B = 0 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right) + & (46) \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) = 0 \mid B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right) \\ & \leqslant \frac{1}{2} + |\mathcal{C}|\epsilon + \frac{1}{2} \cdot |\mathcal{C}|^{-1} + \frac{1}{2} \cdot \epsilon & (47) \end{aligned}$$

What (47) essentially states is that
$P\left(B = 1 | B = 0 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right) + P\left(B = 0 | B = 1 \wedge L \notin \mathcal{L}_{\mathsf{FFC}}\right)$ is at most (approximately) 1/2. Unfortunately this is still not what we need, as it could be the case, e.g., that the first summand is equal 1, the second is equal to 0 (and hence (47) holds), but $P\left(B = 0 | L \notin \mathcal{L}_{\mathsf{FFC}}\right)$ is overwhelming, and hence the total probability of negating B is much higher than 1/2. Intuitively, this should not happen, as one can expect the distribution of B conditioned on $L \notin \mathcal{L}_{\mathsf{FFC}}$ to be close to uniform. We confirm this intuition below. First, from (40) we get that

$$|P\left(\mathsf{ext}(\tilde{L}, \tilde{R}) = c \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right) - \frac{1}{|\mathcal{C}|}| \leqslant \epsilon \tag{48}$$

and

$$|P\left(\mathsf{ext}(\tilde{L}, \tilde{R}) \neq c \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right) - \frac{|\mathcal{C}| - 1}{|\mathcal{C}|}| \leqslant \epsilon. \tag{49}$$

Therefore

$$\frac{1 - |\mathcal{C}|\epsilon}{|\mathcal{C}| - 1 + |\mathcal{C}|\epsilon} \leqslant \frac{P\left(\mathsf{D}^c_{\mathsf{ext}}(\tilde{L}, \tilde{R}) = 0 \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(\mathsf{D}^c_{\mathsf{ext}}(\tilde{L}, \tilde{R}) = 1 \mid \tilde{L} \notin \mathcal{L}_{\mathsf{FFC}}\right)} \leqslant \frac{1 + |\mathcal{C}|\epsilon}{|\mathcal{C}| - 1 - |\mathcal{C}|\epsilon}$$

Now, to get from the uniform $(\tilde{L}, \tilde{R})$ to (L, R) (which does not have a uniform distribution, as it comes from $(L, R) \leftarrow \mathsf{E}^c_{\mathsf{ext}}(B)$) we use the observation (30) and obtain

$$\frac{(|\mathcal{C}|-1)\,(1-|\mathcal{C}|\epsilon)}{|\mathcal{C}|-1+|\mathcal{C}|\epsilon} \leqslant \frac{P\left(\mathsf{D}^c_{\mathsf{ext}}(L,R)=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(\mathsf{D}^c_{\mathsf{ext}}(L,R)=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)} \leqslant \frac{(|\mathcal{C}|-1)\,(1+|\mathcal{C}|\epsilon)}{|\mathcal{C}|-1-|\mathcal{C}|\epsilon},$$

which implies that

$$1-|\mathcal{C}|^2\epsilon \leqslant \frac{P\left(B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)} \leqslant 1+|\mathcal{C}|^2\epsilon \tag{50}$$

Now, from (47) we get

$$\begin{aligned} & \frac{1}{2}+|\mathcal{C}|\epsilon+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+\frac{1}{2}\cdot\epsilon \\ \geqslant\ & \frac{P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(B=0|L \notin \mathcal{L}_{\mathsf{FFC}}\right)}+ \\ & \frac{P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(B=1|L \notin \mathcal{L}_{\mathsf{FFC}}\right)}, \end{aligned}$$

and therefore

$$\begin{aligned} & \left(\frac{1}{2}+|\mathcal{C}|\epsilon+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+\frac{1}{2}\cdot\epsilon\right)\cdot P\left(B=0|L \notin \mathcal{L}_{\mathsf{FFC}}\right) \\ \geqslant\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)+ \\ & \frac{P\left(B=0|L \notin \mathcal{L}_{\mathsf{FFC}}\right)}{P\left(B=1|L \notin \mathcal{L}_{\mathsf{FFC}}\right)}\cdot\ P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right) \\ \geqslant\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)+ \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)\cdot\left(1-|\mathcal{C}|^2\epsilon\right) \\ \geqslant\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)+ \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)-|\mathcal{C}|^2\epsilon, \end{aligned} \tag{51}$$

where (51) comes from (50). Thus we get that

$$\begin{aligned} & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right)+ \\ & P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1 \mid L \notin \mathcal{L}_{\mathsf{FFC}}\right) \end{aligned} \tag{52}$$

is at most

$$\frac{1}{2}+|\mathcal{C}|\epsilon+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+\frac{1}{2}\cdot\epsilon\cdot P\left(B=0|L \notin \mathcal{L}_{\mathsf{FFC}}\right)+|\mathcal{C}|^2\epsilon$$

By a similar argument we can obtain another bound on (52), name that can show that (52) is also at most is also at most

$$\left(\frac{1}{2}+|\mathcal{C}|\epsilon+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+\frac{1}{2}\cdot\epsilon\right)\cdot P\left(B=1|L\notin\mathcal{L}_{\mathsf{FFC}}\right)+|\mathcal{C}|^2\epsilon.$$

We therefore get

$$\begin{aligned}
& P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=1\wedge B=0\mid L\notin\mathcal{L}_{\mathsf{FFC}}\right)+\\
& P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L),g(R))=0\wedge B=1\mid L\notin\mathcal{L}_{\mathsf{FFC}}\right)\\
\leqslant\ & \frac{1}{2}+|\mathcal{C}|\epsilon+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+\\
& \frac{1}{2}\cdot\epsilon\cdot\underbrace{\left(P\left(B=1|L\notin\mathcal{L}_{\mathsf{FFC}}\right)+P\left(B=0|L\notin\mathcal{L}_{\mathsf{FFC}}\right)\right)}_{=1}+|\mathcal{C}|^2\epsilon\\
\leqslant\ & \frac{1}{2}+\frac{1}{2}\cdot|\mathcal{C}|^{-1}+2|\mathcal{C}|^2\epsilon
\end{aligned}$$

which clearly implies (35). □

□

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

$$|\mathcal{L}_{\mathsf{FFC}}\times\mathcal{R}_{\mathsf{FFC}}|\geqslant\delta\cdot|\mathcal{L}'\times\mathcal{R}'|. \tag{53}$$

This will suffice since later we will show (cf. (69)) 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 $\mathcal{L}'\times\mathcal{R}'$).

Claim 3. *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 (53) holds. Then*

$$P\left(\mathsf{D}^c_{\mathsf{ext}}\left(\mathsf{Dec}(f(L^1),g(R^1))\right)=0\mid(L^1,R^1)\in\mathcal{L}_{\mathsf{FFC}}\times\mathcal{R}_{\mathsf{FFC}}\right)\leqslant 2|\mathcal{C}|^{-1}+2\epsilon. \tag{54}$$

Proof. Let $(\hat{L},\hat{R})$ be distributed uniformly over $\mathcal{L}_{\mathsf{FFC}}\times\mathcal{R}_{\mathsf{FFC}}$. Recall that $\mathcal{L}_{\mathsf{FFC}}$ and $\mathcal{R}_{\mathsf{FFC}}$ consist of those elements of $\mathcal{L}$ and $\mathcal{R}$ (resp.) that do not collide with too many other elements under the functions f and g (resp.). To explain the basic proof idea first let us go to the extreme and assume that f and g are injective on $\mathcal{L}_{\mathsf{FFC}}$ and $\mathcal{R}_{\mathsf{FFC}}$. This implies that the min-entropies of $f(\hat{L})$ and $g(\hat{R})$ are equal to the min-entropies of $\tilde{L}$ and $\tilde{R}$ (resp.), and hence, by the assumption (53) their sum is at least $\log|\mathcal{L}'|+\log|\mathcal{R}'|-\log(1/\delta)$.

Since normally this would be a large value, we could use the fact that ext is an extractor and obtain that $d(\mathsf{ext}(\hat{L},\hat{R}))$ is close to uniform, which would clearly imply that the probability that $\mathsf{ext}(\hat{L},\hat{R}) = c$ is close to $|C|^{-1}$, and hence, in turn, that the probability that $\mathsf{D}^c_{\mathsf{ext}}(\hat{L},\hat{R}) = 0$ is negligible.

There are two problems with the above argument. Firstly, the distribution of $(\hat{L},\hat{R})$ is not equal to the distribution of (L^1,R^1) conditioned on the event that $(L^1,R^1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$. Secondly, f and g are only "close to injective", and the proof needs to take it into account. Below we show how to deal with both problems. We start with showing that the distribution of $(\hat{L},\hat{R})$ is close to the distribution of (L^1,R^1) (conditioned on $(L^1,R^1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}$). This is actually not surprising, as a random vector $(\tilde{L},\tilde{R})$ with an overwhelming probability encodes 1. Formally, this can be shown using the following transformations.

$$
\begin{aligned}
& \Delta\left((\hat{L},\hat{R})\ ;\ \left((L^1,R^1)|(L^1,R^1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)\right) \\
= \ & \Delta\left((\hat{L},\hat{R})\ ;\ (\hat{L},\hat{R}) \mid \mathsf{D}^c_{\mathsf{ext}}(\hat{L},\hat{R}) = 1\right) && (55)\\
= \ & \Delta\left((\hat{L},\hat{R})\ ;\ (\hat{L},\hat{R}) \mid \mathsf{ext}(\hat{L},\hat{R}) \neq c\right) && (56)\\
\leqslant \ & 1 - P\left(\mathsf{ext}(\hat{L},\hat{R}) \neq c\right) && (57)\\
= \ & P\left(\mathsf{ext}(\hat{L},\hat{R}) = c\right), && (58)
\end{aligned}
$$

where (55) comes from the assumption that $(L^1,R^1) \leftarrow \mathsf{Enc}(1)$, Eq. (56) comes from the construction of $\mathsf{E}^c_{\mathsf{ext}}$ and Eq. (57) follows from Lemma 3. Now, from the assumption (53) we get that $\mathbf{H}_\infty(\hat{L}) + \mathbf{H}_\infty(\hat{R}) = \log|\mathcal{L}_{\mathsf{FFC}}| + \log|\mathcal{R}_{\mathsf{FFC}}| - \log(1/\delta)$, which from (31) is at least $3k/2 \geqslant k$. Hence, we can use the fact that ext is an (k,ϵ)-two source extractor, and obtain that (58) is at most $|\mathcal{C}|^{-1} + \epsilon$. Hence Eq. (58) implies that

$$
\begin{aligned}
P\left(\mathsf{D}^c_{\mathsf{ext}}\left((f(L^1),g(R^1))\right) = 0 \mid (L^1,R^1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right) \leqslant \\
P\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(\hat{L}),g(\hat{R})\right) = 0\right) + |\mathcal{C}|^{-1} + \epsilon. \qquad (59)
\end{aligned}
$$

Now let us deal with the second problem. Observe that

$$
\begin{aligned}
\mathbf{H}_\infty(f(\hat{L})) &= -\log\left(\max_y P\left(f(\hat{L}) = y\right)\right) \\
&\geqslant -\log\frac{2^{k-\log|\mathcal{R}'|}}{|\mathcal{L}_{\mathsf{FFC}}|} \\
&= \log|\mathcal{R}'| - k + \log|\mathcal{L}_{\mathsf{FFC}}|
\end{aligned}
$$

and, by the symmetry of $\hat{L}$ and $\hat{R}$ also

$$\mathbf{H}_\infty(f(\hat{R})|\hat{R} \in \mathcal{R}_{\mathsf{FFC}}) \geqslant |\mathcal{L}'| - k + \log|\mathcal{R}_{\mathsf{FFC}}|.$$

Therefore:

$$\begin{aligned}
& \mathbf{H}_\infty(f(\hat{L})) + \mathbf{H}_\infty(f(\hat{R})) \\
\geqslant\ & \log|\mathcal{L}'| + \log|\mathcal{R}'| - 2k + \log|\mathcal{L}_{\mathsf{FFC}}| + \log|\mathcal{R}_{\mathsf{FFC}}| \\
\geqslant\ & \log|\mathcal{L}'| + \log|\mathcal{R}'| - 2k + \log|\mathcal{L}'| + \log|\mathcal{R}'| - \log(1/\delta) \qquad (60) \\
\geqslant\ & k, \qquad (61)
\end{aligned}$$

where (60) comes from (53), and (61) from (31). Thus, from the assumption that ext is a (k,ϵ)-two source extractor, and from the fact that $\hat{L}$ and $\hat{R}$ are independent and uniform, we get that

$$d(\mathsf{ext}(f(\hat{L}), g(\hat{R}))) \leqslant \epsilon,$$

and thus

$$P(\underbrace{\mathsf{ext}(f(\hat{L}), g(\hat{R})) = c}_{\mathsf{D}^c_{\mathsf{ext}}\left(f(\hat{L}),g(\hat{R})\right)=0}) \leqslant |\mathcal{C}|^{-1} + \epsilon.$$

Combining it with (59) we obtain

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

Hence the claim is proven. □ □

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 3 was proven only under the assumption (53). Let us first expand the left-hand-side of (32). We have

$$\begin{aligned}
& P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B | (L,R) \in \mathcal{L}' \times \mathcal{R}'\right) \qquad (62) \\
= \quad & \overbrace{P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \mid L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}}\right)}^{(*)} \cdot \\
& \cdot P\left(L \notin \mathcal{L}_{\mathsf{FFC}} \vee R \notin \mathcal{R}_{\mathsf{FFC}}\right) \\
& + \overbrace{P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \mid (L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)}^{(**)} \\
& \cdot P\left((L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right) \qquad (63)
\end{aligned}$$

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

Case 1 First, suppose that (53) holds (i.e. $|\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}| \geqslant \delta \cdot |\mathcal{L} \times \mathcal{R}|$). In this case we get that $(**)$ is a equal to

$$\overbrace{P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \wedge (L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} | B=0\right)}^{\leqslant 2|\mathcal{C}|^{-1}+2\epsilon \text{ by Claim 3}} \cdot \overbrace{P\left(B=0\right)}^{=\frac{1}{2}} + \underbrace{P\left(\mathsf{D}^c_{\mathsf{ext}}(\mathsf{Mall}^{f,g}(L,R) \neq B \wedge (L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} | B=1\right)}_{\leqslant 1} \cdot \underbrace{P\left(B=1\right)}_{=\frac{1}{2}}$$
$$\leqslant \frac{1}{2} + |\mathcal{C}|^{-1} + \epsilon.$$

Now, since (62) is a weighted average of and $(*)$ and $(**)$, hence obviously

$$\begin{aligned} & (62) && (64)\\ \leqslant\ & \max\left(\frac{1}{2} + \frac{1}{2} \cdot |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon, \frac{1}{2} + |\mathcal{C}|^{-1} + \epsilon\right) && (65)\\ \leqslant\ & \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon. && (66) \end{aligned}$$

Case 2 Now consider the case when (53) does not hold, i.e.:

$$|\mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}| < \delta \cdot |\mathcal{L} \times \mathcal{R}| \tag{67}$$

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

$$\begin{aligned} & P\left((L,R) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right) && (68)\\ =\ & \frac{1}{2} \cdot P\left(\mathsf{E}^c_{\mathsf{ext}}(0) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right) + \frac{1}{2} \cdot P\left(\mathsf{E}^c_{\mathsf{ext}}(1) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right) && (68)\\ =\ & \frac{1}{2} \cdot P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} \mid \mathsf{ext}(\tilde{L},\tilde{R}) = c\right) + && (68)\\ & \frac{1}{2} \cdot P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}} \mid \mathsf{ext}(\tilde{L},\tilde{R}) \neq c\right) && (68)\\ \leqslant\ & \frac{1}{2} \cdot \frac{P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)}{P\left(\mathsf{ext}(\tilde{L},\tilde{R}) = c\right)} + \frac{1}{2} \cdot \frac{P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)}{P\left(\mathsf{ext}(\tilde{L},\tilde{R}) \neq c\right)} && (68)\\ \leqslant\ & \frac{1}{2} \cdot \frac{P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)}{|\mathcal{C}|^{-1} - \epsilon} + \frac{1}{2} \cdot \frac{P\left((\tilde{L},\tilde{R}) \in \mathcal{L}_{\mathsf{FFC}} \times \mathcal{R}_{\mathsf{FFC}}\right)}{(|\mathcal{C}|-1) \cdot |\mathcal{C}|^{-1} - \epsilon} && (69)\\ \leqslant\ & \delta/(|\mathcal{C}|^{-1} - \epsilon), && (68) \end{aligned}$$

where in (69) we used (33). Hence, in this case, (63) is at most equal to $\delta/(|\mathcal{C}|^{-1} - \epsilon)$, and therefore, altogether, we can bound (62) by

$$\begin{aligned} (62) &\leqslant (*) + \delta/(|\mathcal{C}|^{-1} - \epsilon) && (70) \\ &= \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon) && (71) \end{aligned}$$

Since analyzing both cases gave us bounds (66) and (71), hence all in all we can bound (62) by their maximum, which is at most

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

Hence (32) is proven. □

7 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 malling 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

$$\lambda_1 + \cdots + \lambda_t + \rho_1 + \cdots \rho_t \leqslant \lambda. \tag{72}$$

He learns $\mathsf{Leak}(L, R) = (v^1(L), w^1(R), \ldots, v^t(L), w^t(R))$. Moreover this process is *adaptive*, i.e. the choice of an ith function in the sequence (72) can depend on the $i-1$ first values in the sequence $\mathsf{Leak}(L, R)$. Finally the adversary chooses a 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 17. *Suppose* $\mathsf{ext} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ *is a flexible* (k, ϵ)*-extractor that satisfies (29), where, for some parameters* δ *and* λ *we have*

$$k = \frac{2}{3} \cdot (\log|\mathcal{L}| + \log|\mathcal{R}| - \lambda) - \frac{4}{3} \cdot \log(1/\delta). \tag{73}$$

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

Proof. Fix some adversary $\mathcal{A}$. Let $B \leftarrow \{0,1\}$ and consider the game $\mathsf{Tamper}_B^{\mathcal{A}}$. Let $\ell = \mathsf{Leak}(L, R)$ and let (f, g) be functions chosen by $\mathcal{A}$. By Lemma 15 we need to show that for we have

$$P\left(\mathsf{D}^c_{\mathsf{ext}}(f(L), g(R)) \neq B\right) \leqslant \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + 2\delta/(|\mathcal{C}|^{-1} - \epsilon). \tag{74}$$

It is a standard argument (cf. e.g. [15]) that the set $\{(L, R) \in \mathcal{L} \times \mathcal{R} : \mathsf{Leak}(L, R) = \ell\}$ can be presented as a product $\mathcal{L}^\ell \times \mathcal{R}^\ell$ for some $\mathcal{L}^\ell \subseteq \mathcal{L}$ and $\mathcal{R}^\ell \subseteq \mathcal{R}$. By a counting argument for uniform $\tilde{L} \times \tilde{R} \leftarrow \mathcal{L} \times \mathcal{R}'$ have

$$P\left(|\mathcal{L}^\ell \times \mathcal{R}^\ell| < |\mathcal{L} \times \mathcal{R}| \cdot 2^{-\lambda} \cdot \delta\right) \leqslant \delta,$$

where the probability is taken over $\ell \leftarrow \mathsf{Leak}(\tilde{L}, \tilde{R})$. Therefore (cf. (30)) if $\ell \leftarrow \mathsf{Leak}(L, R)$ then

$$P\left(|\mathcal{L}^\ell \times \mathcal{R}^\ell| < |\mathcal{L} \times \mathcal{R}| \cdot 2^{-\lambda} \cdot \delta\right) \leqslant \delta \cdot |\mathcal{C}|. \tag{75}$$

Thus, assume that

$$|\mathcal{L}^\ell \times \mathcal{R}^\ell| \geqslant |\mathcal{L} \times \mathcal{R}| \cdot 2^{-\lambda} \cdot \delta,$$

which is the same as

$$\log|\mathcal{L}^\ell| + \log|\mathcal{R}^\ell| + \lambda + \log(1/\delta) \geqslant \log|\mathcal{L}| + \log|\mathcal{R}|.$$

Therefore from (73) we get

$$\begin{aligned} k &= \frac{2}{3} \cdot (\log|\mathcal{L}| + \log|\mathcal{R}| - \lambda) - \frac{4}{3} \cdot \log(1/\delta) \\ &\leqslant \frac{2}{3} \cdot \left(\log|\mathcal{L}^\ell| + \log|\mathcal{R}^\ell| + \lambda + \log(1/\delta) - \lambda\right) - \frac{4}{3} \cdot \log(1/\delta) \\ &= \frac{2}{3} \cdot \left(\log|\mathcal{L}^\ell| + \log|\mathcal{R}^\ell|\right) - \frac{2}{3} \cdot \log(1/\delta). \end{aligned}$$

We can therefore use Lemma 16 with $\mathcal{L}' \times \mathcal{R}' = \mathcal{L}^\ell \times \mathcal{R}^\ell$ and obtain that

$$\begin{aligned} & P\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(L), g(R)\right) \neq B \mid (L, R) \in (\mathcal{L}^\ell, \mathcal{R}^\ell) \wedge |\mathcal{L}^\ell \times \mathcal{R}^\ell| \geqslant |\mathcal{L} \times \mathcal{R}| \cdot 2^{-\lambda} \cdot \delta\right) \\ \leqslant\ & \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon). \end{aligned}$$

Therefore we get:

$$\begin{aligned} & P\left(\mathsf{D}^c_{\mathsf{ext}}\left(f(L), g(R)\right) \neq B \mid (L,R) \in (\mathcal{L}^\ell, \mathcal{R}^\ell)\right) \geqslant |\mathcal{L}\times\mathcal{R}| \cdot 2^{-\lambda} \cdot \delta \\ \leqslant\ & \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + \delta/(|\mathcal{C}|^{-1} - \epsilon) + \delta \cdot |\mathcal{C}| \qquad (76) \\ \leqslant\ & \frac{1}{2} + |\mathcal{C}|^{-1} + 2|\mathcal{C}|^2\epsilon + 2\delta/(|\mathcal{C}|^{-1} - \epsilon), \end{aligned}$$

(the "$+\delta \cdot |\mathcal{C}|$" term in (76) accounts for the probability that $|\mathcal{L}^\ell \times \mathcal{R}^\ell| < |\mathcal{L}\times\mathcal{R}| \cdot 2^{-\lambda} \cdot \delta$ — cf. (75)). Hence (74) is proven. □ □

We now show for to instantiate Theorem 17 be the inner-product extractor from Sect. 3.

Theorem 18. *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.

Proof. Set

$$N := 2 \cdot \left\lceil \frac{56}{1 - 4\xi} \right\rceil \cdot (3 + \log(1/\gamma)).$$

Clearly such $N = \mathcal{O}\left(\log(1/\gamma) \cdot (1/4 - \xi)^{-1}\right)$. We will "plug-in" the inner-product extractor into Theorem 17. To this end take $\mathbb{F} := \mathrm{GF}\left(2^{3-\lceil \log(\gamma) \rceil}\right)$ and $n := \mathit{suf}_{\frac{56}{1-4\xi}}$ and $\delta := |\mathbb{F}|^{-2}$. Set

$$k := \frac{2}{3} \cdot (2n \log|\mathbb{F}| - \lambda) - \frac{4}{3}\log(1/\delta).$$

From Lemma 6 we get that $\mathsf{ext} : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ defined as $\mathsf{ext}(L,R) = \langle L, R\rangle$ is a flexible (k,ϵ)-extractor for ϵ such that

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

Hence, by Thm. 17, the encoding scheme $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ (constructed in Sect. 3) is $\left(2|\mathbb{F}|^2\epsilon + 2\delta/(|\mathbb{F}|^{-1} - \epsilon) + |\mathbb{F}|^{-1}\right)$-non-malleable with leakage λ. We now

have

$$\begin{aligned} & 2|\mathbb{F}|^2\epsilon + 2\delta/(|\mathbb{F}|^{-1} - \epsilon) + |\mathbb{F}|^{-1} \\ \leqslant\ & 2|\mathbb{F}|^2|\mathbb{F}|^{-3} + \frac{2|\mathbb{F}|^{-2}}{|\mathbb{F}|^{-1} - |\mathbb{F}|^{-3}} + |\mathbb{F}|^{-1} \qquad (77) \\ \leqslant\ & 6|\mathbb{F}|^{-1} \\ \leqslant\ & 6 \cdot 2^{\lceil \log(\gamma) \rceil - 3} \\ \leqslant\ & \gamma, \end{aligned}$$

where in (77) we use the fact $\epsilon < |\mathbb{F}|^{-3}$ that comes from:

$$\begin{aligned} \log(1/\epsilon) &= \frac{(\frac{1}{3}n - 4)\log|\mathbb{F}| - \frac{2}{3}\lambda - \frac{4}{3}\log(1/\delta)}{3} - 1 \\ &= \frac{(\frac{1}{3}n - 4)\log|\mathbb{F}| - \frac{2}{3}2\xi n\log|\mathbb{F}| - \frac{4}{3}\log(1/\delta)}{3} - 1 \\ &= \frac{\frac{1}{3}n(1-4\xi)\log|\mathbb{F}| - 4\log|\mathbb{F}| - \frac{4}{3}\log(1/\delta)}{3} - 1 \\ &= \frac{\frac{1}{3}n(1-4\xi)\log|\mathbb{F}| - 4\log|\mathbb{F}| - \frac{8}{3}\log|\mathbb{F}|}{3} - 1 \\ &\geqslant \frac{12}{3}\log|\mathbb{F}| - 1 \\ &\geqslant 3\log|\mathbb{F}| \end{aligned}$$

Clearly the dominating cost in computing both $\mathsf{E}^c_{\mathsf{ext}}$ and $\mathsf{D}^c_{\mathsf{ext}}$ is the time need for n multiplications in $\mathbb{F}$. Using a standard FFT algorithm each multiplication can be done in time $\mathcal{O}(\log|\mathbb{F}| \cdot \log^2\log|\mathbb{F}|)$, and hence the total cost of encoding and decoding is $\mathcal{O}(n \cdot \log|\mathbb{F}| \cdot \log^2\log|\mathbb{F}|) = \mathcal{O}(N \cdot \log^2(\log(1/\gamma)))$ □

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 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.

8 Non-malleable codes vs. extractors

In this section we discuss the relationship between the non-malleable and the two-soruce randomness-extractors. Consider, for example, what happens to our encoding scheme $(\mathsf{E}^c_{\mathsf{ext}}, \mathsf{D}^c_{\mathsf{ext}})$ if, instead of basing it on an extractor, we base it on an additive secret sharing scheme over Z_m. More precisely: let $\mathcal{M} = Z_2$, let $\mathsf{Enc}(B)$ be a random pair (L, R) such that $L + R = 0$ if and only if $B = 0$ (obviously, the decoding function just computes $L + R$ and checks it $L + R = 0$). We now show an attack on this scheme. Assume that m is even (a similar attack exists also for odd m) and let $f(L) = (L + 1) \bmod 2$ and $g(R) = R \bmod 2$. It is easy to verify that if the encoded bit B was equal to 0 then the decoded bit B' will always be equal to 1. On the other hand if B was equal to 1 then the decoded bit will be equal to 0 with probability (around) 1/4. Hence the probability that for a random B we get $B' \neq B$ significantly larger than 1/2 and therefore the code is malleable.

This brings a natural question if we could show some relationship between the extractors and the non-malleable codes in the split-state model. Unfortunately, there is no obvious way of formalizing the conjecture that the non-malleable codes need to be based on extractors, since both of these objects are known to exist unconditionally, and therefore implications of a type "the existence of the non-malleable codes implies the existence of extractors" are trivially true.

Observe also that obviously not every decoding function needs to be a two-source extractor, as, e.g., our function $\mathsf{D}^c_{\mathsf{ext}} : \mathcal{L} \times \mathcal{R} \to \mathcal{C}$ is not an extractor, because even for uniformly random $L \leftarrow \mathcal{L}$ and $R \leftarrow \mathcal{R}$ its output $\mathsf{D}^c_{\mathsf{ext}}(L, R)$ is almost certainly 1 and hence it is very far from a uniform distribution on $\mathcal{C}$. The same is true in the other direction, argued already in Sect. 6 there exist examples (namely: the inner product over a small field) when ext is a good extractor, but it cannot be used directly as a decoding function in a non-malleable code.

9 Non-malleable codes vs. leakage-resilent storage

If one looks again at the example from Sect. 8 then, intuitively, the attack presented there is based on the fact that the additive secret sharing is not leakage-resilient, by which we mean that the adversary can obtain significant knowledge about the encoded secret by retrieving only one bit of information from L and R independently. More precisely, suppose that he learns $\lambda(L) = L \bmod 2$ and $\rho(R) = R \bmod 2$. Then by checking if $\lambda(L) = \lambda(R)$ is gets non-trivial information about B (as $\lambda(L) = \lambda(R)$ holds always in $B = 0$ and

holds with probability around $1/2$ if $B = 1$). Note that the functions λ and ρ look very similar to the functions f and g that we constructed to show the malleability of this encoding.

Hence one could conjecture that every split-state non-malleable code needs to be leakage resilient (the opposite is obviously not true as, e.g, the encoding based on the inner product over Z_2 s leakage resilient, cf. e.g. [15], but, as shown in Sect. 6 is malleable). The following is example show that this conjecture is false. More precisely, there exists encoding scheme in the split-state model that is non-malleable but is not resilient to leakage of an arbitrary small fraction α of information from both L and R. To construct this example take any non-malleable code $(\mathsf{Enc} : \mathcal{M} \to \mathcal{L} \times \mathcal{R}, \mathsf{Dec} : \mathcal{L} \times \mathcal{R} \to \mathcal{M})$ and construct a new code $(\mathsf{Enc}' : \mathcal{M} \to \mathcal{L}' \times \mathcal{R}', \mathsf{Dec} : \mathcal{L}' \times \mathcal{R}' \to \mathcal{M})$ as follows. Set $\mathcal{L}' = \mathcal{L}^t$ and $\mathcal{R}' = \mathcal{R}^t$ for $t := suf\alpha^{-1}$. Now to compute $\mathsf{Enc}'(M)$ for any $M \in \mathcal{M}$ first calculate $(L, R) = \mathsf{Enc}(M)$ and then let

$$\mathsf{Enc}'(M) = (\overbrace{(L, \ldots, L)}^{t \text{ times}}, \overbrace{(R, \ldots, R)}^{t \text{ times}}).$$

The decoding function is defined as: $\mathsf{Dec}'((L_1, \ldots, L_t), (R_1, \ldots, R_t) = \mathsf{Dec}(L_1, R_1)$, in other words, it just applies Dec to the first blocks of the inputs and ignores the rest. It is easy to show that $(\mathsf{Enc}', \mathsf{Dec}')$ is non-malleable (as any functions that breaks it can be easily transformed into a function that breaks $(\mathsf{Enc}, \mathsf{Dec})$). On the other hand leaking just L from $(L, \ldots, L)$ and R from $(R, \ldots, R)$ suffices to recover $M = \mathsf{Dec}(L, R)$ completely. This finishes the argument as obviously $|\mathcal{L}|/|\mathcal{L}'| = |\mathcal{R}|/|\mathcal{R}'| = 1/t \leqslant \alpha$. Hence we conclude that leakage-resilience and non-malleability are two orthogonal properties of an encoding scheme.

10 Security against affine malling

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 malled simultaneously (i.e. we do not use the split-model assumption), but the class of the malling functions is restricted to the affine functions over $\mathbb{F}$, i.e. each malling function h is of a form

$$h((L_1, \ldots, L_n), (R_1, \ldots, R_n)) = M \cdot (L_1, \ldots, L_n, R_1, \ldots, R_n)^T + V^T, \quad (78)$$

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 16). 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:

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

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:

$$\begin{aligned} V &:= \left(L_1, \ldots, L_n, \sum_{j \in J_{n+1}} L_j + \sum_{i,j \in K_{n+1}} L_i L_j, 1 \right) \\ W &:= \left(\sum_{j \in J_1} R_j, \ldots, \sum_{j \in J_n} R_j, 1, y + \sum_{j \in J_{n+2}} R_j + \sum_{i,j \in K_{n+2}} R_i R_j \right). \end{aligned}$$

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] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. *TCC*, pages 474–495, 2009.

[2] R. Anderson and M. Kuhn. Tamper resistance - a cautionary note. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, November 1996.

[3] B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F.-X. Standaert, and Y. Yu. Leftover hash lemma, revisited. 2011. `http://eprint.iacr.org/`.

[4] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. *EUROCRYPT 2003*, pages 647–647, 2003.

[5] E. Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.

[6] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(01):1–32, 2005.

[7] C.E.Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, 1948.

[8] H. Chabanne, G. Cohen, J. Flori, and A. Patey. Non-malleable codes from the wire-tap channel. In *Information Theory Workshop (ITW), 2011 IEEE*, pages 55–59. IEEE, 2011.

[9] H. Chabanne, G. Cohen, and A. Patey. Secure network coding and non-malleable codes: Protection against linear tampering. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2546–2550, 2012.

[10] S. Choi, A. Kiayias, and T. Malkin. Bitr: built-in tamper resilience. *ASIACRYPT 2011*, pages 740–758, 2011.

[11] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[12] S.-Y. Chung, G. D. F. Jr., T. J. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications Letters*, 5.

[13] G. Cohen, R. Raz, and G. Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *Computational Complexity (CCC)*, pages 298–308, 2012.

[14] D. Dachman-Soled and Y. Kalai. Securing circuits against constant-rate tampering. *CRYPTO 2012*, pages 533–551, 2012.

[15] F. Davì, S. Dziembowski, and D. Venturi. Leakage-resilient storage. *Security and Cryptography for Networks*, pages 121–137, 2010.

[16] Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. pages 621–630, 2009.

[17] Y. Dodis, X. Li, T. Wooley, and D. Zuckerman. Privacy amplification and non-malleable extractors via character sums. In *FOCS 2011*, pages 668–677, 2011.

[18] Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, pages 601–610, 2009.

[19] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003.

[20] S. Dziembowski and S. Faust. Leakage-resilient circuits without computational assumptions. *TCC*, pages 230–247, 2012.

[21] S. Dziembowski and K. Pietrzak. Intrusion-resilient secret sharing. In *FOCS'07*, pages 227–237.

[22] S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *FOCS'08*, pages 293–302. IEEE.

[23] S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. *ICS*, pages 434–452, 2010.

[24] S. Faust, T. Rabin, L. Reyzin, E. Tromer, and V. Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. pages 135–156, 2010.

[25] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. *TCC*, pages 258–277, 2004.

[26] S. Goldwasser and G. Rothblum. How to compute in the presence of leakage, 2012. accepted to FOCS 2012.

[27] S. Halevi and H. Lin. After-the-fact leakage in public-key encryption. *TCC*, pages 107–124, 2011.

[28] J. HÅstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[29] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[30] Y. Ishai, M. Prabhakaran, A. Sahai, and D. Wagner. Private circuits ii: Keeping secrets in tamperable circuits. *EUROCRYPT*, pages 308–327, 2006.

[31] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. *CRYPTO*, pages 463–481, 2003.

[32] Y. Kalai, B. Kanukurthi, and A. Sahai. Cryptography with tamperable and leaky memory. *CRYPTO 2011*, pages 373–390, 2011.

[33] F. Liu and A. Lysyanskaya. Tamper and leakage resilience in the split-state model. *CRYPTO 2012*, pages 517–532, 2012.

[34] S. Micali and L. Reyzin. Physically observable cryptography. *TCC*, pages 278–296, 2004.

[35] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. *CRYPTO 2009*, pages 18–35, 2009.

[36] E. N. of Excellence (ECRYPT). Side channel cryptanalysis lounge. `http://www.emsec.rub.de/research/projects/sclounge`.

[37] L. Pontryagin and R. Gamkrelidze. *Topological Groups*. Gordon and Breach Science Publishers.

[38] A. Rao. An exposition of bourgain 2-source extractor. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, page 034, 2007.

[39] R.Dorfman. The detectation of defective members of large population. *Ann.Math.Statist.*, 1943.

[40] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[41] H. Wee. Public key encryption against related key attacks. *PKC 2012*, pages 262–279, 2012.

[42] H. Yamamoto. Rate-distortion theory for the shannon cipher system. *IEEE Transactions on Information Theory*, 43.