Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 23 (2020)

Non-Malleability against Polynomial Tampering

Marshall Ball
Columbia University
marshall@cs.columbia.edu

Eshan Chattopadhyay
Cornell University
eshanc@cornell.edu

Jyun-Jie Liao
Cornell University
jl3825@cornell.edu

Tal Malkin
Columbia University
tal@cs.columbia.edu

Li-Yang Tan
Stanford University
liyang@cs.stanford.edu

June 25, 2020

Abstract

We present the first explicit construction of a *non-malleable code* that can handle tampering functions that are *bounded-degree polynomials*. Prior to our work, this was only known for degree-1 polynomials (affine tampering functions), due to Chattopadhyay and Li (STOC 2017). As a direct corollary, we obtain an explicit non-malleable code that is secure against tampering by bounded-size arithmetic circuits.

We show applications of our non-malleable code in constructing *non-malleable secret sharing schemes* that are robust against bounded-degree polynomial tampering. In fact our result is stronger: we can handle adversaries that can adaptively choose the polynomial tampering function based on initial leakage of a bounded number of shares.

Our results are derived from explicit constructions of *seedless non-malleable extractors* that can handle bounded-degree polynomial tampering functions. Prior to our work, no such result was known even for degree-2 (quadratic) polynomials.

1 Introduction

1.1 Non-malleable codes

Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs [DPW18] as a natural and useful modification of error correcting codes, which can handle stronger forms of adversarial tampering attacks (including ones that can change all symbols of the codeword),

ISSN 1433-8092

while still providing meaningful guarantees. Informally, a non-malleable code is a pair of algorithms $(\mathrm{Enc}, \mathrm{Dec})$, and it is secure against a tampering function family $\mathcal{F}$ if for every tampering function $f \in \mathcal{F}$, the decoding of a tampered codeword, namely $\mathrm{Dec}(f(\mathrm{Enc}(s)))$ for an arbitrary message s, will either be the original message s, or a value completely unrelated to s. (See Section 3.3 for a formal definition.)

As an example of an application of non-malleable codes, one can consider s as being the signing key of a digital signature scheme, and is stored as $\mathrm{Enc}(s)$ in memory. The non-malleability guarantee ensures that for any tampering attack which turns $\mathrm{Enc}(s)$ into $f(\mathrm{Enc}(s))$, the tampered signature is signed under either s or a completely unrelated key. In both cases the tampered signature does not help the adversary learn how to forge a valid signatures on its own.

Non-malleable codes have also found other useful applications in cryptography, such as in constructing non-malleable commitments [GPR16], public-key encryption systems [CMTV15], and, as we discuss in Section 1.2, non-malleable secret sharing [GK18a, GK18b, BS18, ADN+18].

Dziembowski at al. [DPW18] observed that some restrictions on the tampering function family is necessary. Indeed, it is impossible to achieve non-malleability if the adversary is able to decode the codeword, tamper the message, and then re-encode the tampered message. In the last 10 years, non-malleable codes have been shown to exist for numerous rich tampering function families and in various settings. In this work we focus on *explicit, information-theoretic* constructions.

A successful line of work focused on *split-state* tampering functions, where the codeword is broken into several disjoint parts and the adversary can tamper each part arbitrarily but independently [DKO13, CG14, CZ14, ADKO15, CGL16, Li17, KOS17, GMW18, KOS18, ADL18, Li19, AO19]. This line of work has culminated in the construction of near-optimal codes in this setting.

Recently there has been significant interest and progress on constructing non-malleable codes in a more general setting, where the tampering functions are not restricted to fixed partitions, and can act *globally* on the codeword. Global tampering classes that have been studied include permutations and bit flipping [AGM+15], local functions [BDKM16], affine functions over $\mathbb{F}_2$ [CL17], small-depth circuits [CL17, BDSG+18], and small-depth decision trees [BGW19]. Our work fits into this line of research.

Our Results. We consider the tampering class of *bounded-degree polynomials*. This is a natural class of tampering functions, and significantly generalizes the class of affine tampering functions (i.e. degree-1 polynomials) studied in [CL17]. We define the setting more precisely as follows. Let q be a prime, and $\mathrm{Poly}_{n,q,d}$ denote the family of n-variate polynomials over $\mathbb{F}_q$ of degree at most d. We are interested in the following family of tampering functions:

$$\mathcal{F}_{n,q,d} := \{(p_1, \ldots, p_n) : \forall i \in [n], p_i \in \mathrm{Poly}_{n,q,d}\}.$$

For $P = (p_1, \ldots, p_n) \in \mathcal{F}_{n,q,d}$, and $x \in \mathbb{F}_q$, define $P(x) := (p_1(x), \ldots, p_n(x))$.

The following is our main result.

Theorem 1.1 (NMCs for bounded-degree polynomials)**.** *There exists a constant* $C > 0$ *such that for all integers* n, d, m*, any* $\varepsilon > 0$ *and any prime* $q > (Cn^2 d^4 m 2^{2m}/\varepsilon^2) \cdot \log(nd/\varepsilon)$*, there exists a non-malleable code on alphabet* $[q]$*, with block length* n*, message length* m*, relative rate* $\Omega(m/n \log q)$ *and error* ε*, with respect to the family* $\mathcal{F}_{n,q,d}$*.*

Prior to our work, no explicit construction of a non-malleable code was known even for quadratic polynomials ($d = 2$).

To prove Theorem 1.1, we construct new explicit seedless non-malleable extractors that can handle the tampering class $\mathcal{F}_{n,q,d}$. A similar strategy was adopted in [CL17], where they constructed seedless non-malleable extractors against affine tampering functions (i.e, $\mathcal{F}_{n,q,1}$). However, their construction of such extractors heavily exploit the linearity of the tampering functions and explicit constructions of extractors that are linear, and their techniques seem to break down even against quadratic tampering functions. We introduce a completely different approach to construct seedless non-malleable extractors against higher degree polynomial tampering. We discuss this in detail in Section 1.3.

We use Theorem 1.1 to derive a non-malleable code that is secure against tampering by arithmetic circuits. Consider the following family of tampering functions:

$$\mathcal{E}_{n,q,s} := \{(e_1, \ldots, e_n) : \ e_i \text{ is an } n\text{-variate size-}s \text{ arithmetic circuit over } \mathbb{F}_q\}.$$

For $E = (e_1, \ldots, e_n) \in \mathcal{E}_{n,q,s}$ and $x \in \mathbb{F}_q$, we define $E(x) := (e_1(x), \ldots, e_n(x))$.

Corollary 1.2 (NMCs for arithmetic circuits)**.** *There exists a constant* $C > 0$ *such that for all integers* n, s, m*, any* $\varepsilon > 0$ *and any prime* $q > (Cn^2 s m 2^{4s+2m}/\varepsilon^2) \cdot \log(n/\varepsilon)$*, there exists a non-malleable code on alphabet* $[q]$*, with block length* n*, message length* m*, relative rate* $\Omega(m/n \log q)$ *and error* ε*, with respect to the family* $\mathcal{E}_{n,q,s}$*.*

To our knowledge, this is the first explicit construction of a non-malleable code that can handle tampering by arithmetic circuits.

Corollary 1.2 follows as a straightfoward consequence of Theorem 1.1, using the fact that a size-s arithmetic circuit computes a polynomial of degree at most 2^s.

1.2 Non-malleable secret sharing

A t-out-of-n secret sharing scheme [Sha79, Bla79] allows a dealer to share a secret $s \in \{0,1\}^m$ among n parties such that any t parties can collectively recover the secret, and yet any colluding $(t-1)$ parties learn nothing about the secret. Recently, Goyal and Kumar [GK18a] initiated the study of the more robust notion of *non-malleable secret sharing*. A non-malleable secret sharing scheme further requires the shares to be non-malleable against a family of tampering functions $\mathcal{F}$. That is, when the shares are tampered by any function $f \in \mathcal{F}$, for any t parties the reconstructed secret should be either s or a value completely unrelated to m.

Similar to non-malleable codes, non-malleable secret sharing schemes aim to provide protection against tampering attacks, and there are strong connections between non-malleable secret sharing schemes and non-malleable codes. In fact, it can be shown that non-malleable codes in the 2-split-state model are 2-out-of-2 secret sharing schemes. In [GK18a], the authors constructed t-out-of-n non-malleable secret sharing schemes in different tampering models. A detailed comparison of these models and references to other related work can be found in [ADN+18]. These models have in common that the tampering functions are "compartmentalized," applying the function independently to different disjoint parts.

A natural direction of investigation is to construct non-malleable secret sharing against tampering functions that are *not* compartmentalized. Recently, Lin et al. [LCG+19] construct a t-out-of-n secret sharing against affine tampering for every t and large enough n, and Chattopadhyay and Li [CL19] construct a non-malleable ramp secret sharing against affine tampering composed with joint tampering.

Our Results. We construct a non-malleable secret sharing scheme that is secure against the class of polynomial tampering functions. Prior to our work, no such explicit construction was known even against the tampering class of quadratic polynomials. The following is an informal version of our result:

Theorem 1.3 (NM secret sharing for polynomial tampering)**.** *For all integers n, d, r, any prime $q > \mathrm{poly}(2^m, n, d)$ and $1 \leq r \leq n$, there exists an r-out-of-n non-malleable secret sharing scheme with respect to polynomial tampering $\mathcal{F}_{n,q,d}$ for m-bit secrets.*

In fact our construction is stronger and can handle an adaptive tampering adversary who chooses the polynomial tampering function $f \in \mathcal{F}_{n,q,d}$ depending on any $r-1$ of the shares.

As in the case of non-malleable codes, the above theorem directly yields explicit non-malleable secret sharing schemes that are secure against the tampering class of bounded-size arithmetic circuits.

1.3 Seedless non-malleable extractors

Informally, a *randomness extractor* is a deterministic algorithm that produces nearly uniform bits of randomness from defective sources of randomness. The study of randomness extractors is motivated by the fact that many applications in computer science require high-quality random bits, whereas most naturally occurring sources of randomness are of much lower quality. Before defining a randomness extractor formally, we first define the notion of min-entropy that is typically used as a measure of the quality of a source:

Definition 1.4 (Min-entropy and (n, k)-sources)**.** *Let X be a distribution on $\{0,1\}^n$. The min-entropy of X, denoted by $H_\infty(X)$, is defined as $\min_x(\log(1/\Pr[X = x]))$.*
An (n, k)-source is a distribution on $\{0,1\}^n$ with min-entropy at least k.

For two distributions D_1 and D_2 on the same universe Ω, we use $|D_1 - D_2|$ to denote the statistical distance between them. We are now ready to define a randomness extractor for a class of sources.

Definition 1.5 (Extractor)**.** *Let $\mathcal{X}$ be a family of sources on $\{0,1\}^n$. A function* Ext : $\{0,1\}^n \to \{0,1\}^m$ *is called an extractor for the family $\mathcal{X}$ with error ε if for any $X \in \mathcal{X}$,*

$$|\mathrm{Ext}(X) - \mathrm{U}_m| \leq \varepsilon.$$

It turns out that there cannot exist an extractor that works for the family of distributions on $\{0,1\}^n$ with min-entropy at least $n-1$. To circumvent this difficulty, a long line of work has focused on extracting from a weak source X assuming access to a short independent seed Y. Such extractors are called *seeded extractors* [NZ96] and we now have almost optimal constructions of such extractors [GUV09, DKSS09]. Another successful line of research focused on extracting random bits assuming more structure on the source X. Such extractors are called as *seedless extractors*. Examples include assuming that the weak source consists of multiple independent sources [CG88, Bou05, BIW06, CZ19], assuming that the source is supported on an affine subspace [Bou07, GR08] or an algebraic variety [Dvi12], or even simply assuming that there are some unknown coordinates of the source that are uniform and independent [CGH$^+$85]. Explicit constructions of seeded and seedless extractors have found numerous applications in complexity theory [Zuc06], coding theory [TSZ04] and cryptography [BBR88, Lu02].

Recently, several works studied a a more robust notion of a randomness extractor called *non-malleable extractor*. The main motivations for studying this stronger variant is from applications in cryptography. Surprisingly, explicit constructions of non-malleable extractors have led to improved constructions of standard extractors. As in the case of standard extractors, there are *seeded non-malleable extractors* and *seedless non-malleable extractors*. The seeded variant was introduced by Dodis and Wichs [DW09] with applications to the problem of privacy application [BBR88]. The seedless variant of non-malleable extractors was introduced by Cheraghchi and Guruswami [CG14] with applications to constructions of non-malleable codes.

We focus on the seedless variant of non-malleable extractors. For the sake of simplicity, we define seedless non-malleable extractors in slightly less generality and refer the reader to Section 3.3 for the more general definition.

Definition 1.6 (Seedless non-malleable extractor)**.** *Let $\mathcal{X}$ be a family of sources on $\{0,1\}^n$ and $\mathcal{F}$ be a class of tampering functions acting on $\{0,1\}^n$. Further assume that all $f \in \mathcal{F}$ does not have any fixed points. A function* nmExt : $\{0,1\}^n \to \{0,1\}^m$ *is defined to be a non-malleable extractor with respect to $\mathcal{X}$ and $\mathcal{F}$ with error ε if the following hold: for any $X \in \mathcal{X}$ and $f \in \mathcal{F}$, we have*

$$|(\mathrm{nmExt}(X), \mathrm{nmExt}(f(X))) - (\mathrm{U}_m, \mathrm{nmExt}(f(X)))| \leq \varepsilon.$$

An informal way of interpreting the above definition is as follows. Let X be a source from the family $\mathcal{X}$. The distribution $X' = f(X)$ represents the tampered distribution, where $f \in \mathcal{F}$ (note that $X' \neq X$). The task of the non-malleable extractor nmExt is to remove the correlation between the random variables X and X' (which are clearly dependent).

Chattopadhyay and Zuckerman [CZ14] gave explicit constructions of seedless non-malleable extractors assuming X consists of 10 independent sources, and each source is arbitrarily tampered. This was improved by Chattopadhyay, Goyal and Li [CGL16] to construct seedless non-malleable extractors for 2 independent sources. Chattopadhyay and Li [CL17] constructed a seedless non-malleable extractor against the class of affine functions. In another work, Chattopadhyay and Li [CL19] constructed seedless non-malleable extractors when the source X consists of 2 independent sources that are interleaved in an unknown way. They also consider some generalizations such as composition of linear tampering and partitioned tampering.

Our results. We give a seedless non-malleable extractor that can handle polynomial tampering. Prior to our work, Chattopadhyay and Li [CL17] handled the special case of affine tampering. Their construction heavily relied on linearity of the tampering functions and linearity properties of extractors, and their techniques do not seem to extend even to the case tampering functions that are quadratic polynomials. While a seedless non-malleable extractor for uniform source is sufficient for the reduction in [CG14], we show that our non-malleable extractor in fact works for *skew affine source* defined below. This generality is useful in our construction of non-malleable secret sharing schemes that are robust to polynomial tampering.

Definition 1.7. *Let $\mathbb{F}_q$ be a finite field, and let $X = (X_1, \ldots, X_n)$ be a distribution on $\mathbb{F}_q^n$. We say X is an* affine source *if X is uniform over an affine subspace $W \subseteq \mathbb{F}_q^n$. We define the dimension of X to be the dimension of W. We say X is a* skew affine source *if X is an affine source and for every $i \in [n]$, X_i has support size greater than 1.*

We are now ready to state our result on explicit non-malleable extractors against polynomial tampering.

Theorem 1.8. *There exists a constant $C > 0$ such that for all integers n, d, m, any prime q and any $\varepsilon > 0$ such that $q > (Cn^2 d^4 m 2^{2m}/\varepsilon^2) \cdot \log(nd/\varepsilon)$, there exists an explicit function $\mathrm{nmExt} : \mathbb{F}_q^n \to \{0,1\}^m$, that is a seedless non-malleable extractor with respect to the family of sources*

$$\mathcal{X} = \{X : X \textit{ is a skew affine source on } \mathbb{F}_q^n \textit{ of dimension} \geq 1\}$$

and the tampering family $\mathcal{F}_{n,q,d}$.

Prior to our work, no explicit construction of a seedless non-malleable extractor was known against even quadratic polynomials ($d = 2$).

We use the above theorem to derive a non-malleable extractor against arithmetic circuits.

Corollary 1.9. *There exists a constant $C > 0$ such that for all integers n, s, m, any prime q and any $\varepsilon > 0$ such that $q > (Cn^2 s m 2^{4s+2m}/\varepsilon^2) \cdot \log(n/\varepsilon)$, there exists an explicit function $\mathrm{nmExt} : \mathbb{F}_q^n \to \{0,1\}^m$, that is a seedless non-malleable extractor with respect to the*

$$\mathcal{X} = \{X : X \textit{ is a skew affine source on } \mathbb{F}_q^n \textit{ of dimension} \geq 1\}$$

and the tampering family $\mathcal{E}_{n,q,s}$.

To the best of our knowledge, this is the first explicit construction of a non-malleable extractor that can handle tampering by arithmetic circuits.

We in fact show that the non-malleable extractors constructed are efficiently invertible, i.e, given any output z, there exists an efficient sampling algorithm that produces a sample from a distribution that is close to uniform on the set $\mathrm{nmExt}^{-1}(z)$. We discuss the sampling algorithm in Section 5. We then use the connection established in [CG14] (see Section 3.4) to derive the explicit non-malleable codes with respect to polynomials (Theorem 1.1) and arithmetic circuits (Corollary 1.9).

Organization. We give an overview of our techniques in Section 2. We discuss some preliminaries in Section 3. In Section 4, we explicitly construct a non-malleable extractor against polynomial tampering functions. In Section 5, we present efficient sampling algorithms necessary to construct efficient non-malleable codes. We use Section 6 to construct a non-malleable secret sharing scheme that can handle polynomial tampering.

2 Overview of techniques

In this section we discuss the main ideas that are used in our explicit constructions of non-malleable codes, non-malleable extractors, and non-malleable secret sharing schemes. We start by discussing the explicit non-malleable extractor against polynomial tampering (Theorem 1.8). We then discuss ideas that go into using this construction to construct efficient non-malleable codes and non-malleable secret sharing schemes that are robust to polynomial tampering.

Seedless non-malleable extractors against polynomials. We discuss the main ideas that goes into the construction of the non-malleable extractor from Theorem 1.8. We consider the simpler setting and assume the the source is uniform (instead of being a skew affine source as in Theorem 1.8). This setting cleanly captures our main ideas. The setup is as follows:

Let n, d be arbitrary integers, and fix any $\varepsilon > 0$. Let $q = \mathrm{poly}(n, d, 1/\varepsilon)$ be a large enough prime (for exact details, see the statement of Theorem 1.8). Let X be the uniform distribution on $\mathbb{F}_q^n$. Our goal is to construct a polynomial time function $\mathrm{nmExt} : \mathbb{F}_q^n \to \{0,1\}^m$ such that for any tampering function $P = (p_1, \ldots, p_n)$ from the class $\mathcal{F}_{n,q,d}$, such that there exits $i \in [n]$ for which $p_i(x) \neq x_i$, we have

$$|\mathrm{nmExt}(X), \mathrm{nmExt}(P(X)) - \mathrm{U}_m, \mathrm{nmExt}(P(X))| \leq \varepsilon.$$

The high level idea of our construction is to observe that we can express X as a convex combination of distributions that are flat on lines in $\mathbb{F}_q^n$, and then design a non-malleable extractor for such line sources. We note that Gabizon and Raz [GR08] used such an approach for constructing affine extractors on large fields.

We now describe our approach more precisely. Our plan is to construct a low-degree multivariate polynomial $h : \mathbb{F}_q^n \to \mathbb{F}_q$ such that the following hold: for all $\beta \in \mathbb{F}_q$, the polynomial

$$g_\beta = h(x) + \beta h(P(x))$$

is non-constant. (We stress that the choice of h cannot depend on P.) Now, for a suitable choice of m (we pick $m = \nu \log q$ for some small enough ν), we claim that for such an h, defining

$$\mathrm{nmExt}(x) = h(x) \pmod{2^m}$$

would satisfy the conclusion of Theorem 1.8.

Before constructing such an h, we first discuss why this is indeed enough. For any $a \in \mathbb{F}_q^n$, $b \in \mathbb{F}_q^n \setminus \{0^n\}$, define the line $L_{a,b} = \{(a_1 + tb_1, \ldots, a_n + tb_n) : t \in \mathbb{F}_q\}$. We abuse notation, also use $L_{a,b}$ to denote the flat distribution on $L_{a,b}$. Then clearly, X can be sampled by first uniformly sampling a, b (from their respective domains), and then sampling from $L_{a,b}$.

The first observation is the following: let $D = deg(g_\beta)$, and let $g_{\beta,a,b}(t)$ be the univariate restriction of g to the line $L_{a,b}$. We note that the coefficient of t^D is $g(b)$. Appealing to the fact that a low degree polynomial has few roots (Lemma 3.5), it follows that with high probability (over sampling a, b), the univariate polynomial $g_{\beta,a,b}(t)$ is a non-constant polynomial of degree D. Fix such vectors a, b so that $g_{\beta,a,b}$ is a non-constant polynomial. We now use a deep result from algebraic geometry known as the Weil bound (see Theorem 3.1) to conclude that for any non-trivial character[1] χ of $\mathbb{F}_q$, we have

$$|\mathbb{E}_{t \sim \mathbb{F}_q}[\chi(g_{\beta,a,b}(t))]| \leq D/\sqrt{q}.$$

Roughly, this asserts the fact that the non-trivial Fourier coefficients of the distribution $g_{\beta,a,b}(U_{\mathbb{F}_q})$ are bounded, where $U_{\mathbb{F}_q}$ denotes the uniform distribution on $\mathbb{F}_q$. Such a bound can be now be translated into statistical closeness of the distribution $\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(P(L_{a,b}))$ to $\mathrm{U}_m, \mathrm{nmExt}(P(L_{a,b}))$ using known XOR lemmas (see Lemma 3.2, Lemma 3.3). To conclude that $\mathrm{nmExt}(X), \mathrm{nmExt}(P(X))$ is close to $\mathrm{U}_m, \mathrm{nmExt}(P(X))$, we combine the fact that X is a convex combination of the flat sources $L_{a,b}$, and that for most a, b, we have $\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(P(L_{a,b}))$ is close to $\mathrm{U}_m, \mathrm{nmExt}(P(L_{a,b}))$.

Given the above discussion, all that remains to construct the required non-malleable extractor is to find such an h. We recall the guarantee we need from h for convenience of the reader:

- for all $\beta \in \mathbb{F}_q$ and $P = (p_1, \ldots, p_n) \in \mathcal{F}_{n,q,d}$ satisfying that for some $i \in [n]$ $p_i(x) \neq x_i$, the polynomial $g(x) = h(x) + \beta h(P(x))$ is a non-constant polynomial.
- h must a low degree polynomial. In particular, we require $deg(h) \ll q^{1/2}$.

An initial attempt to construct such an h could be to use a polynomial similar to the one used by Gabizon and Raz [GR08] in their affine extractor construction and define

$$h(x_1, x_2, \ldots, x_n) = x_1^{c_1} + x_2^{c_2} + \ldots + x_n^{c_n},$$

[1] See Section 3 for a quick recap of characters of finite fields.

where $c_1, c_2, \ldots, c_n$ are arbitrary distinct positive integer. It is not hard to see that this does not work as follows. It is always possible to find $\beta, \gamma_1, \gamma_2, \ldots, \gamma_n \in \mathbb{F}_q^*$ such that $\gamma_i^{c_i} = -\beta^{-1}$, such that at least one $\gamma_i \neq 1$. Now defining $P = (\gamma_1 x_1, \ldots, \gamma_n x_n)$ gives the desired counterexample since for this choice of β and P, $h(x) + \beta h(P(x)$ is identically the zero polynomial.

We avoid the above counterexample as follows: Pick $c_1, c_2, \ldots, c_{2n}$ from an arithmetic progression such that the common difference is co-prime with $q-1$, and define

$$h(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i^{c_{2i-1}} + x_i^{c_{2i}}.$$

For this choice of h, it is not hard to prove that if each $p_i(x) = \gamma_i x_i$ (for some $\gamma_i \in \mathbb{F}_q$), and $g(x)$ is a constant polynomial, it must be that each γ_i is 1, and $\beta = -1$. However this contradicts our assumption on P that for some i, $p_i(x) \neq x_i$. Thus we avoid the counterexample discussed above.

We in fact that this choice of h works for all $P \in \mathcal{F}_{n,q,d} \setminus \{(x_1, \ldots, x_n)\}$. To prove this, we rely on a result (Lemma 3.4) which shows that for such a choice of c_i's, for any distinct $i_1, i_2 \in [n]$, $deg(p_{i_1}^{c_{i_1}})$ is well separated from $deg(p_{i_2}^{c_{i_2}})$. With a careful case analysis, we use this to show that some monomial (of degree at least 1) in $g(x)$ survives. We provide the details in Section 4.

Non-malleable extractors for skew affine sources against polynomial tampering. In the previous paragraph we sketched how to construct a non-malleable extractor against polynomial tampering assuming access to a uniform source on $\mathbb{F}_q^n$. In Section 4, we actually show that the non-malleable extractor works for any affine source which is non-constant on every coordinate. We call such source a *skew affine source*. In other words, our non-malleable extractor is resilient to affine leakage which does not reveal any single coordinate in the source. We will see the application of this property in non-malleable secret sharing.

To prove this stronger property of the non-malleable extractor, recall that in previous section we defined a polynomial $g(x) = h(x) + \beta h(P(x))$, and its restriction to the line $L_{a,b}$, denoted by $g_{\beta,a,b}(t)$. We then sketched a proof that $g_{\beta,a,b}$ is non-constant if $g(b) \neq 0$, which happens with high probability over b. In Section 4, we actually show the following stronger result: $\forall i, b_i \neq 0$ is a sufficient condition for $g_{\beta,a,b}$ to be non-constant. In fact, it is also a necessary condition. If there exists i such that $b_i = 0$, the adversary can set $p_j(x) = x_j$ for every $j \neq i$ and $p_i(x) = c$ for a constant $c \neq a_i$. One can verify that $g_{-1,a,b}$ is a constant in this case.

The proof idea is that a similar case analysis as sketched in the previous section also works for $g_{\beta,a,b}$ if $b_i \neq 0$ for every i. We then show that every skew affine source is a convex combination of line source $L_{a,b}$ where $b_i \neq 0$ for every i (Lemma 4.1) to finish the proof.

Non-malleable codes against polynomial tampering. We now turn to cryptographic applications of our non-malleable extractors. To build a non-malleable code against polynomial tampering, we use the connection between non-malleable code and non-malleable extractor established in [CG14]. To apply the reduction in [CG14], we need an efficient algorithm which samples almost uniformly from a pre-image of our non-malleable extractor on any output.

Recall that our non-malleable extractor is of the form $\mathrm{nmExt}(x) = \sigma(h(x))$, where σ is modulo 2^m and h is a bounded-degree polynomial. Inverting σ is easy, and there exists an algorithm by Cheraghchi and Shokrollahi [CS09] which almost-uniformly samples a pre-image of bounded-degree polynomial (over any large enough prime field). An initial attempt to sample from $\mathrm{nmExt}^{-1}(z)$ would be first sample $y \in \sigma^{-1}(z)$ and then sample from $h^{-1}(y)$. However this does not work since $h^{-1}(y)$ might have different size for different $y \in \mathbb{F}_q$. So we need to sample $y \in \sigma^{-1}(z)$ with probability proportional to $|h^{-1}(y)|$. A possible way to perform such weighted sampling from $\sigma^{-1}(z)$ is to do a rejection sampling which samples $y \in \sigma^{-1}(z)$ uniformly in each round and accept with probability proportional to $|h^{-1}(y)|$. However, we need to (approximately) count $|h^{-1}(y)|$ in this approach, which is difficult in general.

Chattopadhyay and Zuckerman [CZ14] handled a similar sampling task while constructing efficient non-malleable codes in the split-state model, with the crucial difference being that they were dealing with polynomials on a constant number of variables. In [CZ14], they adopted a similar sampling strategy as the one sketched above, and they count $|h^{-1}(y)|$ with an algorithm from [HW98], which has running time doubly exponential in the number of variables (which, in their case, still takes constant time).

To get around this difficulty, we observe that the algorithm in [CS09] is actually a rejection sampling which has accepting probability proportional to $|h^{-1}(y)|$ in each round. Therefore, we can embed an uniform sampling of y in each round of [CS09] and bypass the computation of $|h^{-1}(y)|$. We provide the details of our sampling algorithm in Section 5.

Non-malleable secret sharing against polynomial tampering. As another application of our non-malleable extractor, we build a non-malleable secret sharing that can handle polynomial tampeering. We obtain this by plugging in our extractor into a scheme by Lin, Cheraghchi, Guruswami, Safavi-Naini and Wang [LCG+19]. In this scheme, they take an efficiently invertible non-malleable extractor nmExt and a linear erasure code $(\mathrm{Enc}, \mathrm{Dec})$, then define the sharing function to be $\mathrm{Enc} \circ \mathrm{nmExt}^{-1}$ and the reconstruction function to be $\mathrm{nmExt} \circ \mathrm{Dec}$. If in the erasure code $(\mathrm{Enc}, \mathrm{Dec})$, Dec only needs r symbols in the codeword to reconstruct the original message, then so does $\mathrm{nmExt} \circ \mathrm{Dec}$ in the secret sharing scheme. Therefore the correctness holds as long as there is an efficient inverter for nmExt which succeeds with high probability.

To prove privacy and non-malleability we need the following guarantee on nmExt. To guarantee non-malleability, for every tampering function f, nmExt should be non-malleable against the composed tampering function $\mathrm{Dec} \circ f \circ \mathrm{Enc}$. For polynomial tampering, taking the erasure code to be a linear code over $\mathbb{F}_q$ naturally satisfies this requirement. To guarantee privacy, given a uniform source X, $\mathrm{nmExt}(X)$ should be uniform conditioned on that some symbols of $\mathrm{Enc}(X)$ is leaked to the adversary. When $(\mathrm{Enc}, \mathrm{Dec})$ is a linear code, this means nmExt should be an affine extractor. This is also true for our extractor (see Appendix A).

We in fact achieve a stronger result and construct a non-malleable secret sharing scheme where the adversary can choose the polynomial tampering function based on some of the shares. If given a secret the adversary can learn a symbol of $\mathrm{nmExt}^{-1}(s)$ from their shares, the secret sharing scheme sketched above will become malleable. We show that we can avoid this problem by taking Enc to be a "truncated systematic MDS code". That is, we take a MDS code for which the encoding is in the form $f(x) = (x, f'(x))$, then we discard x and only keep $f'(x)$. For $x \in \mathbb{F}_q^r$, we can prove that given any $r-1$ symbols in $f'(x)$, it is not possible to recover any symbol in x. Roughly speaking, if given $r-1$ symbols in $f'(x)$ it is possible to recover x_i, then these symbols together with x_i contain "redundant information", which violates the MDS property. This is in fact very similar to Shamir's secret sharing scheme, and the only difference is we want to hide every single symbol in the message while Shamir's secret sharing is only hiding the first symbol because the others are random. Because our extractor is non-malleable given any other form of affine leakage (using the fact that our non-malleable extractor works for any skew affine source of dimension at least 1), we can conclude that the corresponding r-out-of-n secret sharing is non-malleable even if the adversary choose their tampering function based on $r-1$ shares. We provide more details of our non-malleable secret sharing scheme in Section 6.

3 Preliminaries

Define $e(x) = e^{2\pi i x}$, where $i = \sqrt{-1}$.
For any distribution D, let $D(x)$ denote $\Pr[D = x]$, and let $\mathrm{Supp}(D)$ denote the support of D.
Let U_m denote the uniform distribution over m bits. Let U_Σ denote the uniform distribution over the finite set Σ.
For two distributions D_1 and D_2 on the same universe, we use $|D_1 - D_2|$ to denote the statistical distance. We use $D_1 \approx_\varepsilon D_2$ to denote the fact that D_1 and D_2 are ε-close in statistical distance.
For non-negative integers $\lambda_1, \ldots, \lambda_n$ that sum to 1, and arbitrary distributions $D_1, \ldots, D_n$, we use $\sum_i \lambda_i D_i$ to denote the distribution that places weight $\sum_i \lambda_i D_i(x)$ at the point x.
For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For non-negative integer k, we use $\binom{[n]}{k}$ denote the set of all subsets of $[n]$ of size k. Let Σ be a set of symbol. For sequence $X = (x_1, \ldots, x_n) \in \Sigma^k$ and $S = \{i_1, \ldots, i_k\} \subseteq [n]$ such that $i_1 < i_2 < \ldots < i_k$, we use X_S to denote the sequence $(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$.

3.1 Characters sums over finite fields

Let q be a prime. The additive characters of $\mathbb{F}_q$ are of the form $\chi_j(x) = e(xj/q)$, for $j = 0, 1, \ldots, q-1$. χ_0 is called the trivial character, and the others are called as non-trivial characters of $\mathbb{F}_q$. We now recall a deep result from algebraic geometry that has found various applications in pseudorandomness.

Theorem 3.1 (Weil bound [Wei48])**.** *Let p be a non-constant univariate polynomial of degree $d < q$ over $\mathbb{F}_q$. For any non-trivial additive character χ of $\mathbb{F}_q$, we have*

$$\left| \sum_{y \in \mathbb{F}_q} \chi(p(y)) \right| \leq d\sqrt{q}.$$

We record a couple of XOR lemmas that lets us translate bounds on expectations of characters under a distribution D, to the closeness of D in statistical distance to the uniform distribution.

Lemma 3.2 ([Rao07])**.** *For every prime q, there exists an efficiently computable map $\sigma : \mathbb{F}_q \to \{0,1\}^m$ such that if Y is a distribution on $\mathbb{F}_q$ such that for every non-trivial additive character χ of $\mathbb{F}_q$,*

$$\mathbb{E}[\chi(Y)] \leq \delta,$$

then it is the case that

$$|\sigma(Y) - \mathrm{U}_m| \leq \varepsilon,$$

where $\varepsilon = \delta 2^{m/2} + O(2^m/q)$.

Lemma 3.3 ([Rao07, DLWZ14])**.** *For every prime q, there exists an efficiently computable map $\sigma : \mathbb{F}_q \to \{0,1\}^m$ such that if (Y, Y') is a distribution on $\mathbb{F}_q \times \mathbb{F}_q$ where for all additive characters χ, ϕ of $\mathbb{F}_q$, where χ is non-trivial,*

$$\mathbb{E}[\chi(Y)\phi(Y')] \leq \delta,$$

then it is the case that

$$|\sigma(Y), \sigma(Y') - \mathrm{U}_m, \sigma(Y')| \leq \varepsilon,$$

where $\varepsilon = \delta 2^m + O(2^m/q)$.

3.2 Useful lemmas about polynomials

We recall a useful result from [DGW09] (Lemma 4.2).

Lemma 3.4. *Let n, r, d, λ be arbitrary positive integers, and q be a prime. Let $p_1(x), \ldots, p_r(x) \in \mathrm{Poly}_{n,d,q}$ be non-constant polynomials. Suppose that $d_i = deg(p_i)$. Define $c_i = \lambda(2dr + 1) + \lambda i$. Then, for all $1 \leq i < j \leq r$, we have*

$$|deg(p_i^{c_i}) - deg(p_j^{c_j})| = |c_i \cdot d_i - c_j \cdot d_j| \geq \lambda.$$

We also record the Schwartz-Zippel Lemma.

Lemma 3.5 ([Zip79, Sch79])**.** *Let $p(x) \in \mathrm{Poly}_{n,d,q}$ be a non-zero polynomial. Then,*

$$\Pr_{x \in \mathbb{F}_q^n}[p(x) = 0] \leq d/q.$$

3.3 Non-malleable codes and seedless non-malleable extractors

Definition 3.6 (Coding schemes)**.** *Let Σ be a finite alphabet set. A pair of functions* $(\mathrm{Enc}, \mathrm{Dec})$*, where* $\mathrm{Enc} : \{0,1\}^k \to \Sigma^n$ *is a randomized function and* $\mathrm{Dec} : \Sigma \to \{0,1\}^k \cup \{\perp\}$ *is a deterministic function, is defined to be a coding scheme with block length n and message length k if for all $z \in \{0,1\}^k$,* $\Pr[\mathrm{Dec}(Enc(s)) = s] = 1$.

Definition 3.7 (Tampering functions)**.** *Let Σ be a finite alphabet set. For any $n > 0$, let $\mathcal{H}_{\Sigma,n}$ denote the set of all functions $h : \Sigma^n \to \Sigma^n$. Any subset $\mathcal{G} \subseteq \mathcal{H}_{\Sigma,n}$ is a family of tampering functions.*

For simplicity, we sometimes do not specify the domain of tampering functions when it is clear from the context. We define a function that will be useful in defining non-malleable codes:

$$\mathrm{copy}(x, y) = \begin{cases} x & \text{if } x \neq \text{same} \\ y & \text{if } x = \text{same}. \end{cases}$$

Definition 3.8 (Non-malleable codes)**.** *Let Σ be a finite alphabet set. A coding scheme* $(\mathrm{Enc}, \mathrm{Dec})$ *on alphabet Σ with block length n and message length k is a non-malleable code with respect to a tampering family $\mathcal{G} \subset \mathcal{H}_{\Sigma,n}$ and error ε if for every $g \in \mathcal{H}_{\Sigma,n}$ there is a random variable D_g supported on* $\{0,1\}^k \cup \{\text{same}\}$ *that is independent of the randomness in* Enc*, and any message $z \in \{0,1\}^k$, we have*

$$|\mathrm{Dec}(f(\mathrm{Enc}(z))) - \mathrm{copy}(D_g, z)| \leq \varepsilon$$

We define the rate of a non-malleable code $\mathcal{C}$ to be the quantity $\frac{k}{n \log(|\Sigma|)}$.

Definition 3.9 (Seedless non-malleable extractors)**.** *Let Σ be a finite alphabet set, $\mathcal{G}$ be a class of tampering functions $\Sigma^n \to \Sigma^n$ and $\mathcal{X}$ be a class of distribution over Σ^n. A function* $\mathrm{nmExt} : \Sigma^n \to \{0,1\}^m$ *is called a seedless non-malleable extractor that works for $\mathcal{X}$ with respect to $\mathcal{G}$ with error ε if for every distribution $X \in \mathcal{X}$ and every tampering function $g \in \mathcal{G}$, there exists a random variable D_g on* $\{0,1\}^m \cup \{\text{same}\}$ *that is independent of X, such that*

$$|(\mathrm{nmExt}(X), \mathrm{nmExt}(g(X))) - (\mathrm{U}_m, \mathrm{copy}(D_g, \mathrm{U}_m))| \leq \varepsilon.$$

3.4 Non-malleable codes via seedless non-malleable extractors

Cheraghchi and Guruswami [CG14] established the following connection between non-malleable codes and seedless non-malleable extractors.

Theorem 3.10. *Let Σ be some finite alphabet set. Let* $\mathrm{nmExt} : \Sigma^n \to \{0,1\}^m$ *be a polynomial time computable seedless non-malleable extractor that works for uniform distribution with respect to a class of tampering functions $\mathcal{G}$ acting on Σ^n. Suppose there is a sampling algorithm* Samp *that on any input $z \in \{0,1\}^m$ runs in time* $\mathrm{poly}(n, \log|\Sigma|)$ *and samples from a distribution that is δ-close to uniform on the pre-image set* $\mathrm{nmExt}^{-1}(s)$.

Then there exists an efficient construction of a non-malleable code on alphabet Σ with block length n, relative rate $\frac{m}{n}$, error $2^m\varepsilon + \delta$ with respect to the tampering family $\mathcal{G}$.

Given such an invertible non-malleable extractor, the non-malleable code for $\mathcal{G}$ is defined as follows: Any message $v \in \{0,1\}^m$ is encoded as $\mathrm{Samp}(v)$. The decoding of a codeword $c \in \Sigma^n$ is $\mathrm{nmExt}(c) \in \{0,1\}^m$.

3.5 MDS code

Definition 3.11. *Let $C \subseteq \mathbb{F}_q^n$ be a linear subspace of dimension k where $\mathbb{F}_q$ is the finite field with q elements. We say C is a $[n,k,d]_q$ code if every two distinct codewords $c_1, c_2 \in C$ coincide in at most $n-d$ coordinates. We say C is a $[n,k]_q$* MDS (maximum distance separable) code *if C is a $[n,k,n-k+1]$ code, i.e. C matches Singleton bound [Sin64].*

Definition 3.12. *Let C be a $[n,k,d]_q$ code and* Enc *be a bijective linear mapping from $\mathbb{F}_q^k$ to C. We say* Enc *is* systematic encoding *of C if there exists a function $\mathrm{Enc}' : \mathbb{F}_q^k \to \mathbb{F}_q^{n-k}$ such that for every $x \in \mathbb{F}_q^k$, $\mathrm{Enc}(x) = (x, \mathrm{Enc}'(x))$.*

The distance property of a $[n,k]_q$ MDS code guarantees that the codewords remain distinct even when restricted to only k out of n symbols. Moreover, it is well-known that Reed-Solomon code [RS60] is a MDS code, and every linear code has a systematic encoding. (For example, see [LF04] for a systematic encoding of Reed-Solomon code.) Therefore we have the following lemma.

Lemma 3.13. *For every finite field $\mathbb{F}_q$ of q element, and every integer k, n such that $k \le n \le q$, there exists a $[n,k]_q$ MDS code $C \subseteq \mathbb{F}_q^n$ and an efficient systematic encoding $\mathrm{Enc} : \mathbb{F}_q^k \to C$. Moreover, for every $R \subseteq [n]$ of size $|R| = k$, there exists an efficient decoding algorithm $\mathrm{Dec}_R : \mathbb{F}_q^k \to \mathbb{F}_q^k$ such that for every $x \in \mathbb{F}_q^k$, $\mathrm{Dec}_R(\mathrm{Enc}(x)_R) = x$, where $\mathrm{Enc}(x)_R$ denote the restriction of $\mathrm{Enc}(x)$ on the coordinates specified by R.*

3.6 Other useful lemmas

We will also use the following lemma for statistical distance in [LCG+19] (Lemma 13).

Lemma 3.14. *Let $\mathcal{V}, \mathcal{W}$ be finite sets, and let (V,W), (V',W') be joint distribution on $\mathcal{V} \times \mathcal{W}$. Let $\varepsilon > 0$ be real number such that*

$$(V,W) \approx_\varepsilon (V',W').$$

Then for every event $\mathcal{E} \subseteq \mathrm{Supp}(W) \cap \mathrm{Supp}(W')$,

$$|(V \mid W \in \mathcal{E}) - (V' \mid W' \in \mathcal{E})| \le \frac{\varepsilon}{\Pr[W \in \mathcal{E}]}.$$

4 Non-malleable extractors against polynomials

We present the proof of Theorem 1.8 in this section. On a high level, our idea is to express X as a convex combination of sources on lines in $\mathbb{F}_q^n$, and design a non-malleable extractor for such line sources. We note that Gabizon and Raz [GR08] adopted such an approach for constructing affine extractors over large fields. First we show that a skew affine source is a convex combination of skew line source.

Lemma 4.1. *Let q be a prime, $n < q$ be a integer and $X \in \mathbb{F}_q^n$ be a skew affine source of dimension k. Then there exists a distribution $A \in \mathbb{F}_q^n$ and a vector $b \in (\mathbb{F}_q \backslash \{0\})^n$ such that $X \equiv A + Tb$, where T is uniform over $\mathbb{F}_q$.*

Proof. Suppose X is uniform over the affine subspace $W + z$ where W is a linear subspace of $\mathbb{F}_q^n$ and $z \in \mathbb{F}_q^n$ is a fixed vector. Our goal is to find a vector $b \in W$ s.t. $b_i \neq 0$ for every $i \in [n]$. Given such b we can set $A \equiv X$, and the lemma holds because $X + w \equiv X$ for every $w \in W$ and $tw \in W$ for every $w \in W$ and $t \in \mathbb{F}_q$.

Fix a basis $\{w_1, \dots, w_k\}$ of W. For every $i \in [k]$, define $S_i = \{j \in [n] : w_j \neq 0\}$ and $\overline{S}_i = \bigcup_{j=1}^{i} S_j$. Note that $\overline{S}_k = [n]$ because $W + z$ does not have any constant coordinate. we will prove by induction that for every $i \in [k]$ there exists $v_i \in \text{span}(w_1, \dots, w_i)$ s.t. $(v_i)_j \neq 0$ for every $j \in \overline{S}_i$. Assume that there exists v_{i-1} which satisfies the induction hypothesis. (Note that $v_0 = 0$.) Consider the set of q distinct vectors $L_i = \{v_{i-1} + tw_i : t \in \mathbb{F}_q\} \subseteq \text{span}(w_1, \dots, w_i)$. Observe that for every $j \in S_i$, there exists at most one vector $u_j \in L_i$ satisfying that $(u_j)_j = 0$. Since $n < q$, there must exist $u^* \in L_i$ s.t. $(u^*)_j \neq 0$ for every $j \in S_i$. Moreover, for every $j \in \overline{S_i} \backslash S_i$, $(u^*)_j = (v_{i-1})_j \neq 0$. Therefore $(u^*)_j \neq 0$ for every $j \in \overline{S}_i$. Finally observe that v_k is a valid choice of b because $\overline{S}_k = [n]$ and $\text{span}(w_1, \dots, w_k) = W$. □

Next we present the extractor construction and prove correctness. Let B be the smallest integer greater than 3 such that $\gcd(B, q-1) = 1$. Note that B must be a prime. We can deduce an upper bound on B as follows. Define the primorial function $\nu(\ell)$ as the product of the first ℓ primes. It is known that $\nu(\ell) = e^{(1+o(1))\ell \log(\ell)}$ [Dus10]. Further, it is known that the ℓ'th smallest prime number is at most $O(\ell \log(\ell))$ [Ros39, Rob88]. Hence, it must be that $B \leq \mu \log q$, for some large enough constant μ. We can thus find such a B efficiently.

For $i \in [2n]$, define $c_i = B(4dn + 1) + Bi$. Define the function $h : \mathbb{F}_q^n \to \mathbb{F}_q$ as

$$h(x_1, \dots, x_n) = \sum_{i=1}^{n} (x_i^{c_{2i-1}} + x_i^{c_{2i}}).$$

Let $\sigma : \mathbb{F}_q \to \{0,1\}^m$ be the mapping from Lemma 3.3. We now define the non-malleable extractor:

$$\text{nmExt}(x) = \sigma(h(x)).$$

For any $a \in \mathbb{F}_q^n$ and $b \in \mathbb{F}_q^n \setminus \{0^n\}$, define the line $L_{a,b} = \{a + tb : t \in \mathbb{F}_q\}$. We overload notation, and also use $L_{a,b}$ to denote the flat source on this line. We will show that nmExt is a non-malleable extractor against $\text{Poly}_{n,d,q}$ for every skew line source. Theorem 1.8 then follows using Lemma 4.1.

Lemma 4.2. *Let $a \in \mathbb{F}_q^n, b \in (\mathbb{F}_q \backslash \{0\})^n$. For every tampering function $P \in \mathrm{Poly}_{n,d,q}$ which is not identity on $L_{a,b}$,*

$$\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(P(L_{a,b})) \approx_\varepsilon \mathrm{U}_m, \mathrm{nmExt}(P(L_{a,b})),$$

where $\varepsilon = O\left(\frac{2^m d^2 n \log q}{\sqrt{q}}\right)$

The following bound is the key ingredient. Indeed, Lemma 4.2 then follows using Lemma 3.3.

Lemma 4.3. *Let χ, ϕ be additive characters of $\mathbb{F}_q$ such that χ is non-trivial. Then,*

$$|\mathbb{E}[\chi(h(L_{a,b}))\phi(h(P(L_{a,b})))]| \leq O((d^2 n \log q)/\sqrt{q}).$$

Let $\chi(y) = e^{2\pi\alpha y/q}$ and $\phi(y) = e^{2\pi\alpha' y/q}$. Since χ is non-trivial, we know that $\alpha \neq 0$. Let $\beta = \alpha'/\alpha$. Define the polynomial

$$g(x) = h(x) + \beta h(P(x)).$$

We note that

$$|\mathbb{E}[\chi(h(X))\phi(h(P(X)))]| \leq \left|\mathbb{E}\left[e\left(\frac{\alpha g(X)}{q}\right)\right]\right|.$$

Let $g_{a,b}(t)$ be the univariate polynomial obtained by restricting $g(x)$ to the line $L_{a,b}$. The following two claims directly yields Lemma 4.3.

Lemma 4.4. *Suppose for some $a, b \in \mathbb{F}_q^n$, $g_{a,b}$ is a non-constant polynomial. Then,*

$$\left|\mathbb{E}_{t \sim \mathbb{F}_q}\left[e\left(\frac{\alpha \cdot g_{a,b}(t)}{q}\right)\right]\right| \leq O((d^2 n \log q)/\sqrt{q}).$$

Lemma 4.5. *For every $a \in \mathbb{F}_q^n$, $b \in (\mathbb{F}_q \backslash \{0\})^n$, $g_{a,b}$ is a constant polynomial only if P is identity on $L_{a,b}$.*

Lemma 4.4 is indeed simple to obtain using the Weil bound.

Proof of Lemma 4.4. Follows directly from Theorem 3.1 using the fact that $deg(g_{a,b}(t)) \leq O(d^2 n \log q)$. □

Now we prove Lemma 4.5.

Proof of Lemma 4.5. For every $i \in [n]$, define the polynomial $q_i(t) = p_i(a + tb)$. Since $a + tb$ is an affine function, $deg(q_i) \leq deg(p_i) \leq d$. Let $d_i = deg(q_i)$. For every $i \in [n]$, define

$$w_i(t) = (a_i + tb_i)^{c_{2i-1}} + (a_i + tb_i)^{c_{2i}} + \beta q_i(t)^{c_{2i-1}} + \beta q_i(t)^{c_{2i}}.$$

Recall that

$$g_{a,b}(t) = \sum_i w_i(t).$$

First we prove that $deg(w_i) \in \{0, c_{2i}d_i, c_{2i}, c_{2i-1}, c_{2i}-1\}$. Moreover, $deg(w_i) = 0$ if and only if $\beta = -1$ and $q_i(t) = a_i + tb_i$. To prove this statement, first we consider the case $deg(q_i) \geq 2$. Suppose that the leading coefficient in q_i is $s_i \neq 0$. If $\beta \neq 0$, the coefficient of $t^{c_{2i}d_i}$ in w_i is $\beta s_i^{c_{2i}} \neq 0$. Therefore $deg(w_i) = c_{2i}d_i$. If $\beta = 0$, the coefficient of $t^{c_{2i}}$ in w_i is $\beta b_i^{c_{2i}} \neq 0$. Therefore $deg(w_i) = c_{2i}$. Next consider the case $deg(q_i) = 0$. With an argument similar to the case $\beta = 0$, we also have $deg(w_i) = c_{2i}$. Finally consider the case $deg(q_i) = 1$. Suppose $q_i(t) = r_i + ts_i$. Observe that the coefficient of $t^{c_{2i}}$ in w_i is $b_i^{c_{2i}} + \beta s_i^{c_{2i}}$ and the coefficient of $t^{c_{2i}-1}$ in w_i is $c_{2i}(a_i b_i^{c_{2i}-1} + \beta r_i s_i^{c_{2i}-1})$. In this case either $deg(w_i) \in \{c_{2i}, c_{2i}-1\}$ or

$$b_i^{c_{2i}} = -\beta s_i^{c_{2i}} \text{ and } a_i b_i^{c_{2i}-1} = -\beta r_i s_i^{c_{2i}-1}.$$

The equations hold only when there exists $k \in \mathbb{F}_q$ s.t.

$$r_i = ka_i, s_i = kb_i \text{ and } k^{c_{2i}} = -\beta^{-1}.$$

If such k exists, we can write $w_i(t) = (1 - k^{-B}(a_i + tb_i)^{c_{2i-1}}$. If $\beta = -1$, we have $k = 1$, $w_i(t) = 0$ and $q_i(t) = a_i + tb_i$. If $\beta \neq -1$, then $k \neq 1$, which implies $(1 - k^{-B}) \neq 0$ because $(B, q-1) = 1$. Therefore w_i contains a monomial of degree c_{2i-1} with coefficient $(1 - k^{-B})b_i^{c_{2i-1}} \neq 0$, and hence $deg(w_i) = c_{2i-1}$.

Now we show that $g_{a,b}(t)$ is a constant polynomial only if $\beta = -1$ and $q_i(t) = a_i + tb_i$ for every $i \in [n]$. Consider the set of index $I = \{i \in [n] : deg(w_i) > 0\}$. Then for every $i \in I$, $deg(w_i) \in \{d_i c_{2i}, c_{2i}, c_{2i-1}, c_{2i}-1\}$ if $d_i > 0$, or $deg(w_i) \in \{c_{2i}, c_{2i-1}, c_{2i}-1\}$ if $d_i = 0$. By Lemma 3.4, for every pair $i, j \in I$, $i \in j$, $deg(w_i) \neq deg(w_j)$. Therefore $deg(g_{a,b}) > 0$ if I is non-empty. If $g_{a,b}$ is a constant polynomial, it must be the case that $deg(w_i) = 0$ for every $i \in [n]$. This only happens when $\beta = -1$ and $q_i(t) = a_i + tb_i$ for every $i \in [n]$.

□

Finally we prove Theorem 1.8 formally.

Theorem 4.6 (Theorem 1.8, restated)**.** *There exists a constant $C > 0$ such that for every integers n, m, d, any $\varepsilon > 0$, any prime q such that $q > Cn^2 d^4 m 2^{2m} \cdot \log(nd/\varepsilon)$, any skew affine source $X \in \mathbb{F}_q^n$ of dimension ≥ 1 and any tampering function $f \in \mathrm{Poly}_{n,d,q}$, there exists a distribution D_f on $\{0,1\}^m \cup \{\mathrm{same}\}$ that is independent of X, such that*

$$|\mathrm{nmExt}(X), \mathrm{nmExt}(f(X)) - \mathrm{U}_m, \mathrm{copy}(D_f, \mathrm{U}_m)| \leq \varepsilon.$$

Proof. By Lemma 4.1, there exists a distribution A on $\mathbb{F}_q^n$ and vector b such that $X = \sum_a \Pr[A = a] \cdot L_{a,b}$. Define $I = \{a \in \mathbb{F}_q^n : f \text{ is identity on } L_{a,b}\}$. For every $a \in I$, define $(D_f)_a = \mathrm{same}$. For every $a \notin I$ define $(D_f)_a = \mathrm{nmExt}(f(L_{a,b}))$. Then we claim that

$D_f = \sum_a \Pr[A=a]\cdot(D_f)_a$ satisfies the requirement:

$$\begin{aligned}
&|\mathrm{nmExt}(X), \mathrm{nmExt}(f(X)) - \mathrm{U}_m, \mathrm{copy}(D_f, \mathrm{U}_m)| \\
&\le \sum_a \Pr[A=a]\cdot|\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(f(L_{a,b})) - \mathrm{U}_m, \mathrm{copy}((D_f)_a, \mathrm{U}_m)| \\
&= \sum_{a\in I} \Pr[A=a]\cdot|\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(L_{a,b}) - \mathrm{U}_m, \mathrm{U}_m| \\
&+ \sum_{a\notin I} \Pr[A=a]\cdot|\mathrm{nmExt}(L_{a,b}), \mathrm{nmExt}(f(L_{a,b})) - \mathrm{U}_m, \mathrm{nmExt}(f(L_{a,b}))| \\
&\le \sum_{a\in I} \Pr[A=a]\cdot\varepsilon + \sum_{a\in I} \Pr[A=a]\cdot\varepsilon \\
&= \varepsilon
\end{aligned}$$

The first inequality is by the convexity of statistical distance, and the second inequality is by Lemma 4.2. □

5 Efficient sampling

Recall that to construct efficient non-malleable codes using the connection established in [CG14], we need to efficiently sample from the pre-image of any given output of the non-malleable extractor constructed in the previous section. (We discuss this connection in Section 3.4.) In this section we show how to construct such a sampler for the non-malleable extractor constructed in Theorem 1.8. Note that Corollary 1.9 use the same non-malleable extractors.

Theorem 5.1. *Let* $\mathrm{nmExt} : \mathbb{F}_q^n \to \{0,1\}^m$ *be the non-malleable extractor against* $\mathcal{F}_{n,q,d}$ *tampering in Theorem 1.8. Then there exists a randomized algorithm* $\overline{\mathrm{nmExt}^{-1}}$ *such that for every* $z \in \{0,1\}^m$ *the distribution of* $\overline{\mathrm{nmExt}^{-1}}(z)$ *is* ε*-close to uniform distribution on* $\mathrm{nmExt}^{-1}(z)$. *The running time of* $\overline{\mathrm{nmExt}^{-1}}$ *is bounded by* $\mathrm{poly}(n, d, \log q, \log(1/\varepsilon))$.

Our starting point to prove Theorem 5.1 is a sampling algorithm from [CZ14], which has running time $O(d^{n^{O(n)}}(\log q)^{O(1)})$ and error $O(d^{O(n^n)}/\sqrt{q})$. We will show how to modify this algorithm and get an improved running time of $\mathrm{poly}(n, d, \log q, \log(1/\varepsilon))$ for arbitrarily small error ε.

Let nmExt be the non-malleable extractor from Theorem 1.8. Recall that $\mathrm{nmExt} = \sigma \circ h$ where $\sigma : \mathbb{F}_q \to \{0,1\}^m$ is defined as $\sigma(x) = x \pmod{2^m}$ and $h : \mathbb{F}_q^n \to \mathbb{F}_q$ is a multivariate polynomial of degree d over $\mathbb{F}_q$. Given $z \in \{0,1\}^m$, the pre-image of z under nmExt is

$$\mathrm{nmExt}^{-1}(z) = \bigcup_{y\in\sigma^{-1}(z)} h^{-1}(y),$$

and our goal is to sample from $\mathrm{nmExt}^{-1}(z)$ almost uniformly. The sampling algorithm in [CZ14] is based on the following rejection sampling strategy.
Let $M \ge \max_y |h^{-1}(y)|$.

1. Sample $y \in \sigma^{-1}(z)$ uniformly at random.
2. Compute $|h^{-1}(y)|$ (approximately), and accept y with probability $|h^{-1}(y)|/M$. If y is rejected, go back to step 1.
3. Output an (almost) uniform sample from $h^{-1}(y)$.

In [CZ14], the second step is achieved by an algorithm from [HW98] that has running time $O(d^{n^{O(n)}}(k \log q)^{O(1)})$.

The third step is based on the following algorithm in [CS09].

Lemma 5.2 ([CS09]). *Let q be a sufficiently large prime, $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be polynomials of total degree bounded by d, and each polynomial has at most ℓ monomials. Let $S \subseteq \mathbb{F}_q^n$ be the set of common zeroes of f. There exists a randomized algorithm which takes f as input (as a list of monomials) and outputs a random value $X \in \mathbb{F}_q^n$ such that the distribution of X is $O(d^{O(1)}/q)$-close to uniform distribution on S. The worst-case running time of this algorithm is* $\mathrm{poly}(\log q, d, n, \ell)$.

Thus the bottleneck in achieving a polynomial time sampling algorithm is Step (2) which takes time that is doubly exponential in n. We get around this difficulty as follows: first note that the rejection sampling in Step (2) is to ensure that the subset $h^{-1}(y)$ is selected with probability proportional to $|h^{-1}(y)|$. Our crucial observation is that the algorithm in Lemma 5.2 is actually a rejection sampling which accepts an output with probability proportional to $|h^{-1}(y)|$ in each round. Therefore we can actually combine the rejection sampling in step 2 and 3, and bypass the computation of $|h^{-1}(y)|$.

First we explain the relation between the algorithm in Lemma 5.2 and rejection sampling. A naive way to sample from the variety $h^{-1}(y)$ is to repeatedly sample a point $x \in \mathbb{F}_q^n$ and verify if $h(x) = y$. However, the success probability of the naive rejection sampling is only $|h^{-1}(y)|/q^n$, which is too small. The idea in [CS09] is that the space $\mathbb{F}_q^n$ can be split into lines, and the variety S is split into many "slices" by these lines. The naive rejection sampling is equivalent to first sampling a line and then sampling a point from this line. Since each line has q points, the probability of a certain point in the variety being chosen is still $1/q^{n-1} \cdot 1/q$. However, if we choose a *good direction* to split the space, each slice of the variety only has at most d points where $d \ll q$, and these points can be enumerated efficiently. Therefore instead of sampling every point in this subspace with equal probability we can sample only from the *slice of variety* instead. This allows us to increase the accepting probability in each round to $|h^{-1}(y)|/dq^{n-1}$, which is high enough and still proportional to $|h^{-1}(y)|$. With the ideas above we get the following theorem.

Lemma 5.3. *Let $h : \mathbb{F}_q^n \to \mathbb{F}_q$ be a n-variate polynomial of degree $d < q/2$ with ℓ monomials, and $\sigma : \mathbb{F}_q \to \{0,1\}^m$ be any function. Suppose we have access to an oracle* Samp_σ *which takes input z and outputs a sample from $\sigma^{-1}(z)$ uniformly at random. Then for every $\varepsilon > 0$, there exists a randomized algorithm A such that for every $z \in \{0,1\}^m$, the algorithm either outputs a uniformly random sample from $(\sigma \circ h)^{-1}(z)$ or output $\perp$. The probability that the algorithm outputs $\perp$ is at most ε.*

Moreover, the expected running time of A on z is $T \cdot \mathrm{poly}(\log q, n, d, \ell)$ plus T oracle calls to Samp_σ, where

$$T = O\left(\frac{q^{n-1} \cdot d \cdot |\sigma^{-1}(z)|}{|(\sigma \circ h)^{-1}(z)|} \log(1/\varepsilon)\right).$$

Before we formally prove Lemma 5.3, first we show how to prove Theorem 5.1 based on Lemma 5.3. The following corollary shows that the algorithm in Lemma 5.3 is efficient when $\sigma \circ h$ is an "extractor for uniform distribution" and σ does not concentrate on certain output.

Corollary 5.4. *Suppose that $\sigma(h(\mathrm{U}_{\mathbb{F}_q^n})) \approx_{1/2^{m+1}} \mathrm{U}_m$, and $|\sigma^{-1}(z)| \leq Cq/2^m$ for every z. Then the running time of the algorithm in Lemma 5.3 is $C \log(1/\varepsilon)\mathrm{poly}(n, \ell, \log q, d)$.*

Proof. The number of rounds of rejection sampling in the algorithm from Lemma 5.3 is $T = O\left(\frac{q^{n-1} \cdot d \cdot |\sigma^{-1}(z)|}{|(\sigma \circ h)^{-1}(z)|} \log(1/\varepsilon)\right)$.

Observe that

$$\left|(\sigma \circ h)^{-1}(z)\right| = q^n \cdot \Pr[\sigma(h(\mathrm{U}_{\mathbb{F}_q^n})) = z] \geq q^n \cdot (1/2^m - 1/2^{m+1}) = q^n/2^{m+1}.$$

Plugging this in, and the upper on $\sigma^{-1}(z)$, we have $T = O(d \log(1/\varepsilon))$. The corollary now follows directly from Lemma 5.3. □

Proof of Theorem 5.1. To prove Theorem 5.1 we only need to show that our non-malleable extractor satisfies the condition in Corollary 5.4. The fact that $\sigma(h(\mathrm{U}_{\mathbb{F}_q^n}))$ is close to U_m follows Theorem 1.8, and the second condition is also true because $\sigma(x) = x \mod 2^m$, which satisfies $|\sigma^{-1}(z)| \leq \lceil q/2^m \rceil$ for every $z \in \{0,1\}^m$. □

We now prove Lemma 5.3. First we need the following lemma which is analogous to Proposition 4.3 in [CS09]. Note that we slightly tweak the lemma to make the sampling algorithm able to handle arbitrarily small error. The lemma says a random direction is a good direction to split the space with high probability.

Lemma 5.5. *Let $h : \mathbb{F}_q^n \to \mathbb{F}_q$ be a n-variate polynomial of degree at most d, and let $b = (b_1, \ldots, b_n)$ be uniformly random samples from $\mathbb{F}_q$. Then with probability at least $1 - d/q$, $h_{a,b}(t) = h(a_1 + b_1 t, \ldots, a_n + b_n t)$ is a non-constant polynomial of t for every $a = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$.*

Proof. Let g be the highest-degree homogeneous part of h. Then observe that $h_{a,b}(t)$ has degree at most d, and its coefficient of t^d equals to $g(b_1, \ldots, b_n)$. By Lemma 3.5, the probability that $g(b_1, \ldots, b_n)$ is non-zero is at least $1 - d/q$. Therefore with probability $1 - d/q$ over b, $h_{a,b}(t)$ has degree exactly D for every $a \in \mathbb{F}_q^n$. □

Proof of Lemma 5.3. In algorithm A, first we repeatedly sample $b \in \mathbb{F}_q^n$ uniformly at random until we find b which satisfies the condition in Lemma 5.5. If we fail to find such b in $\log(1/\varepsilon) + 1$ rounds, abort and output $\perp$. Then repeat the following steps for at most T rounds:

Sample $y \in \sigma^{-1}(z)$ with oracle Samp_σ, and sample $a = (a_1, \ldots, a_n)$ uniformly at random. Compute the restriction of $h(x) = y$ on the line $L_{a,b} = \{(a_1 + b_1 t, \ldots, a_n + b_n t) : b \in \mathbb{F}_q\}$, i.e. $h_{a,b}(t) = y$ where $h_{a,b}(t) = h(a_1 + b_1 t, \ldots, a_n + b_n t)$. Note that $h_{a,b}$ is a non-constant polynomial of degree at most d. Then we run Berlekamp-Rabin algorithm [Rab80] to enumerate all the roots of $h_{a,b}$ in $\mathbb{F}_q$, denoted by $t_1, \ldots, t_k$ where $k \leq d$. Now pick a number $i \in [d]$ uniformly at random. If $i \leq k$, the algorithm succeeds, and we will return $(a_1 + b_1 t_i, \ldots, a_n + b_n t_i)$. Otherwise sample y and a again and repeat. If no value is returned after all T rounds, return $\perp$.

To prove the correctness of A, first we compute the distribution $A(z)$ conditioned on that the algorithm succeeds. Observe that $A(z)$ never returns an element which is not in $(\sigma \circ h)^{-1}(z)$. Moreover, for every $v \in (\sigma \circ h)^{-1}(z)$, in each round the probability that $A(z)$ outputs v is

$$\frac{1}{|\sigma^{-1}(z)|} \cdot \frac{1}{q^{n-1}} \cdot \frac{1}{d}.$$

The first factor is the probability that $y = h(v)$, the second factor is the probability that $L_{a,b} \ni v$, and the third factor is the probability that v is chosen from the list of roots of $h_{a,b}$. Since this formula does not depend on v, we can conclude that $A(z)$ is a uniform distribution on $(\sigma \circ h)^{-1}(z)$, conditioned on $A(z) \neq \perp$.

Now we compute the probability that A fails. Assuming $q \geq 2d$, the probability that we fail to find a b satisfying the condition in Lemma 5.5 in $log(1/\varepsilon) + 1$ rounds is at most $(d/q)^{\log(1/\varepsilon)+1} \leq \varepsilon/2$. If we find such b successfully, observe that A successfully returns a sample with probability

$$p = \frac{|(\sigma \circ h)^{-1}(z)|}{|\sigma^{-1}(z)| \cdot q^{n-1} \cdot d}$$

in one round. Now define

$$T = \frac{C \log(1/\varepsilon)}{p},$$

for a large enough constant C. Then the probability that A does not output any element after T rounds is at most $(1-p)^T < \varepsilon/2$. Therefore $\Pr_A[A(z) = \perp] \leq \varepsilon$.

Finally we analyze the running time of A. Finding a vector b which satisfies Lemma 5.5 (or abort and output $\perp$) takes at most $\log(1/\varepsilon)\mathrm{poly}(n, \ell, \log q, d)$ steps. After finding b, we run at most T rounds of rejection sampling, where in each round we first make an oracle call to Samp_σ, sample a and compute the polynomial $h_{a,b}$ which takes $\mathrm{poly}(n, \ell, \log q, d)$ steps, and run Berlekamp-Rabin which takes expected $\mathrm{poly}(n, \ell, \log q, d)$ steps. Therefore the total expected running time is as claimed. □

Remark 5.6. *While we only show the expected running time in Lemma 5.3, it is possible to bound the worst-case running time by introducing a small error to the output distribution as follows. In each of the T rounds, we are running Berlekamp-Rabin algorithm to factorize a polynomial of degree at most d. Recall that in Berlekamp-Rabin algorithm, we are repeatedly trying to factorize a polynomial into two non-trivial factors. Moreover, each attempt of factoring succeeds with probability at least $1/2$. To factorize T polynomials of degree d,* *we need at most Td successful attempts. Note that the probability that there are less than Td success in the first $7Td$ attempts are at most $(1/2)^T < \varepsilon$. Therefore, we can force the algorithm to terminate and output $\perp$ after $6Td$ unsuccessful attempts of Berlekamp-Rabin. This ensures that the worst-case running time is still $T \cdot \mathrm{poly}(n, \ell, \log q, d)$. Besides, since the* time-out *event happens with probability at most ε, the output distribution is still ε-close to uniform distribution on $(\sigma \circ h)^{-1}(z)$.*

6 Non-malleable secret sharing

In this section we construct a non-malleable secret sharing scheme that is non-malleable against polynomial tampering. This extends a recent work of Lin et al. [LCG+19] where they could handle affine tampering functions. We use the framework that was introduced in [LCG+19] to derive our secret sharing scheme. In short, the framework in [LCG+19] takes a linear erasure code $(\mathrm{Enc}, \mathrm{Dec})$ and an invertible affine extractor Ext, and define the share function to be $\mathrm{Enc}(\mathrm{Ext}^{-1})$. If Ext is non-malleable against a class of tampering function $\mathcal{F}$ which is closed under composition with linear function, the non-malleability will be inherited by the secret sharing scheme. We show that the non-malleable extractor in Theorem 1.8 is also an extractor for arbitrary affine source (see Appendix A). Thus the framework in [LCG+19] directly gives a non-malleable secret sharing against polynomial tampering.

Besides the direct application, we further show how to construct a r-out-of-n secret sharing which is non-malleable against adversaries who can (adaptively) corrupt $(r-1)$ shares and choose the polynomial tampering functions based on the corrupted shares. To handle such adaptive adversary, we cannot directly plug our extractor into the framework in [LCG+19] because our extractor is non-malleable only for skew affine source. Nevertheless, we will show that non-malleablility for skew affine source is sufficient if we choose a proper erasure code in the [LCG+19] scheme.

First we formally define the non-malleable secret sharing.

Definition 6.1 (Adaptive adversary)**.** *Let Σ denote a set of symbols. We say $\mathcal{A} : \Sigma^n \to \Sigma^k$ is a (n,k)-adaptive adversary if $\mathcal{A}(x_1, \ldots, x_n) = (x_{s_1}, \ldots, x_{s_k})$ for indices $s_1, \ldots, s_k$ defined as follows.*

- *s_1 is fixed.*
- *For every i, there exists a function $f_i : \Sigma^i \to [n]$ such that $s_{i+1} = f_i(x_{s_1}, \ldots, x_{s_i})$.*

Definition 6.2 (Non-malleable secret sharing)**.** *Let Σ be a finite alphabet set. Let* Share : $\{0,1\}^m \to \Sigma^n$ *be a randomized algorithm mapping m bits to into n shares, each being an alphabet from Σ. Let $\mathcal{F} : \Sigma^n \to \Sigma^n$ be a family of tampering function. We say* Share *is a r-out-of-n ε-non-malleable secret sharing with respect to $\mathcal{F}$ if the following properties hold.*

- ***Correctness.*** *For every authorized set $R \subseteq [n]$ of size $|R| = r$, there exists a deterministic algorithm* $\mathrm{Rec}_R : \Sigma^r \to \{0,1\}^m$ *such that for every secret $s \in \{0,1\}^m$,*

$$\Pr[\mathrm{Rec}_R(\mathrm{Share}(s)_R) = s] \geq 1 - \varepsilon,$$

where $\mathrm{Share}(s)_R$ *denotes the* r *shares in* $\mathrm{Share}(s)$ *identified by the set* R*.*

- ***Privacy.*** *For every* $(n, r-1)$*-adaptive adversary* $\mathcal{A}$ *and every pair of secret* $a, b \in \{0,1\}^m$*,*
$$\mathcal{A}\left(\mathrm{Share}(a)\right) \approx_\varepsilon \mathcal{A}\left(\mathrm{Share}(b)\right).$$

- ***Non-malleability.*** *For every* $(n, r-1)$*-adaptive adversary* $\mathcal{A}$*, every reconstruction strategy* $\mathcal{R} : \Sigma^{r-1} \to \binom{[n]}{r}$*, every secret* $s \in \{0,1\}^m$ *and every tampering strategy* $\mu : \Sigma^{r-1} \to \mathcal{F}$*, define the tampering experiment*
$$\tilde{S} = \left\{ \begin{array}{c} \mathit{share} \leftarrow \mathrm{Share}(s) \\ v \leftarrow \mathcal{A}(\mathit{share}) \\ f \leftarrow \mu(v) \\ R \leftarrow \mathcal{R}(v) \\ \widetilde{\mathit{share}} \leftarrow f(\mathit{share}) \\ \mathit{Output} : \mathrm{Rec}_R(\widetilde{\mathit{share}_R}) \end{array} \right\}$$
which is a random variable over the randomness of Share*. Then there exists a distribution* $D_{\mathcal{A},\mathcal{R},\mu}$ *on* $\{0,1\}^m \cup \{\mathrm{same}\}$ *which does not depend on* s *such that*
$$\tilde{S} \approx_\varepsilon \mathrm{copy}(D_{\mathcal{A},\mathcal{R},\mu}, s).$$

As observed in [LCG+19], since the tampering function f can be based on the view of adversary, the adversary can jointly tamper $(r-1)$ adaptively chosen shares arbitrarily. The tampering on shares which the adversary cannot see depends on how strong $\mathcal{F}$ is. In our construction $\mathcal{F}$ would be bounded-degree polynomials. With the non-malleable extractor in Theorem 1.8, we show the following.

Theorem 6.3. *There exists a constant* $C > 0$ *such that for all integers* n, d, r*, any prime* q *and any* $\varepsilon > 0$ *such that* $q > (C2^m n^2 d^4/\varepsilon^2) \cdot \log(nd/\varepsilon)$ *and* $1 \le r \le n$*, there exists a* r*-out-of-*n ε*-non-malleable secret sharing scheme with respect to polynomial tampering* $\mathcal{F}_{n,q,d}$ *for* m*-bit secret.*

Proof. First we specify the construction. Let $\mathrm{nmExt} : \mathbb{F}_q^r \to \{0,1\}^m$ be the non-malleable extractor with respect to $\mathcal{F}_{r,q,d}$ with error $\varepsilon/2^{m+2}$ in *Theorem* 1.8. Let $\mathrm{Enc}(x) = (x, \mathrm{Enc}'(x))$ be the systematic encoding of a $[n+r, r]_q$ MDS code in Lemma 3.13. Let $\overline{\mathrm{nmExt}^{-1}}$ be the sampling algorithm in Theorem 5.1 with error $\varepsilon/2^{m+2}$. Then we define
$$\mathrm{Share}(s) = \mathrm{Enc}'(\overline{\mathrm{nmExt}^{-1}}(m)),$$
where $\overline{\mathrm{nmExt}^{-1}}$ is the almost-uniform inverter of nmExt in Section 4. Next we prove the three properties in Definition 6.2. The proof basically follows [LCG+19], but additionally we need to show that the decoded shares is a skew affine source conditioned on adversary view.

- **Correctness.** For every authorized set $R \subseteq [n]$ of size $|R| = r$, let Dec_R denote the decoding function of Enc' specified by R in Lemma 3.13. Then we define

  $$\mathrm{Rec}_R(v) = \mathrm{nmExt}(\mathrm{Dec}_R(v)).$$

  Rec is a correct reconstruction because for every secret s,

  $$\Pr[\mathrm{Rec}_R(\mathrm{Share}(s)_R) = s] = \Pr[\mathrm{nmExt}\left(\mathrm{Dec}_R\left(\mathrm{Enc}(\overline{\mathrm{nmExt}^{-1}}(s))_R\right)\right) = s] \geq 1 - \varepsilon.$$

  Note that the correctness is not perfect because $\overline{\mathrm{nmExt}^{-1}}(x)$ does not always output a pre-image of x.

- **Privacy.** Let $S = \mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r})$, and define $X = \overline{\mathrm{nmExt}^{-1}}(S)$. Fix any $(n, r-1)$-adaptive adversary $\mathcal{A} : \mathbb{F}_q^n \to \mathbb{F}_q^{r-1}$. Since $\overline{\mathrm{nmExt}^{-1}}$ is an inverter of nmExt with error $\varepsilon/2^{m+2}$, we have $(X, S) \approx_{\varepsilon/2^{m+2}} (\mathrm{U}_{\mathbb{F}_q^n}, S)$, which implies

  $$\left(\mathcal{A}\left(\mathrm{Enc}'(X)\right), S\right) \approx_{\varepsilon/2^{m+2}} \left(\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right), \mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r})\right).$$

  Define $V = \mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right)$. We claim that for every $v \in \mathbb{F}_q^{r-1}$, $Y_v = (\mathrm{U}_{\mathbb{F}_q^r} \mid V = v)$ is a skew affine source with positive min-entropy. Observe that there exists a set $T_v \in \binom{[n]}{r-1}$ uniquely determined by v such that $\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right) = \mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})_{T_v}$. Since Enc' is a linear mapping, $V = v$ corresponds to $r-1$ linear constraints for Y_v. Therefore Y_v is an affine source with positive min-entropy. Now assume for contradiction that Y_v is not skew. Then there exists $i \in [r]$ such that $(Y_v)_i$ is a constant. Since Y_v is not a constant, there exist two distinct value $y_1, y_2 \in \mathrm{Supp}(Y_v)$. Observe that $\mathrm{Enc}'(y_1)_{T_v} = v = \mathrm{Enc}'(y_2)_{T_v}$ and $(y_1)_i = (y_2)_i$. Then $\mathrm{Enc}(y_1) := (y_1, \mathrm{Enc}'(y_1))$ and $\mathrm{Enc}(y_2) := (y_2, \mathrm{Enc}'(y_2))$ coincide on $(r-1)+1$ coordinates, which contradicts to the fact that Enc is a MDS code. Therefore Y_v is skew. By Theorem 1.8,

  $$\left(\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right), \mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r})\right) \approx_{\varepsilon/2^{m+2}} \left(\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right), \mathrm{U}_m\right).$$

  By triangle inequality we have

  $$\left(\mathcal{A}\left(\mathrm{Enc}'(X)\right), S\right) \approx_{\varepsilon/2^{m+1}} \left(\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right), \mathrm{U}_m\right),$$

  which by Lemma 3.14 implies

  $$\left(\mathcal{A}\left(\mathrm{Enc}'(X)\right) \mid S = a\right) \approx_{\varepsilon/2} \left(\mathcal{A}\left(\mathrm{Enc}'(\mathrm{U}_{\mathbb{F}_q^r})\right)\right) \approx_{\varepsilon/2} \left(\mathcal{A}\left(\mathrm{Enc}'(X)\right) \mid S = b\right)$$

  for every $a, b \in \mathrm{Supp}(S)$. Finally, observe that $\mathrm{Supp}(S) = \{0,1\}^m$ because S is $\varepsilon/2^{m+2} < 1/2^m$ close to uniform. Therefore for every $a, b \in \{0,1\}^m$,

  $$\mathcal{A}\left(\mathrm{Enc}'\left(\overline{\mathrm{nmExt}^{-1}}(a)\right)\right) \approx_{\varepsilon} \mathcal{A}\left(\mathrm{Enc}'\left(\overline{\mathrm{nmExt}^{-1}}(b)\right)\right).$$

- **Non-malleability.** Let $S = \mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r})$, and define $X = \overline{\mathrm{nmExt}^{-1}}(S)$. Fix any $(n, r-1)$-adaptive adversary $\mathcal{A} : \mathbb{F}_q^n \to \mathbb{F}_q^{r-1}$, any reconstruction strategy $\mathcal{R} : \mathbb{F}_q^{r-1} \to \binom{[n]}{r}$ and any tampering strategy $\mu : \mathbb{F}_q^{r-1} \to \mathcal{F}_{n,q,d}$. Recall the tampering experiment

$$\widetilde{S} = \left\{ \begin{array}{c} share \leftarrow \mathrm{Enc}'(X) \\ V \leftarrow \mathcal{A}(share) \\ f \leftarrow \mu(V) \\ R \leftarrow \mathcal{R}(V) \\ \widetilde{share} \leftarrow f(share) \\ Output : \mathrm{Rec}_R(\widetilde{share}_R) \end{array} \right\}$$

Note that this tampering experiment is equivalent to applying the tampering experiment in Definition 6.2 on S. Now define

$$\widetilde{S}' = \left\{ \begin{array}{c} share' \leftarrow \mathrm{Enc}(\mathrm{U}_{\mathbb{F}_q^n}) \\ V' \leftarrow \mathcal{A}(share') \\ f \leftarrow \mu(V') \\ R \leftarrow \mathcal{R}(V') \\ \widetilde{share}' \leftarrow f(share') \\ Output : \mathrm{Rec}_R(\widetilde{share}_R) \end{array} \right\}$$

Since $\overline{\mathrm{nmExt}^{-1}}$ is an inverter of nmExt with error $\varepsilon/2^{m+2}$, we have $(S, X) \approx_{\varepsilon/2^{m+2}} (S, \mathrm{U}_{\mathbb{F}_q^n})$ which implies

$$(S, \widetilde{S}) \approx_{\varepsilon/2^{m+2}} (S, \widetilde{S}').$$

For every $v \in \mathbb{F}_q^{r-1}$, define $Y_v = (\mathrm{U}_{\mathbb{F}_q^r} \mid V' = v)$. With the same proof in the privacy part, we can show that Y_v is a skew affine source with positive min-entropy. Now define $f_v = \mu(v)$, $R_v = \mathcal{R}(v)$ and $g_v : \mathbb{F}_q^r \to \mathbb{F}_q^r$ to be $g_v(x) := \mathrm{Dec}_{R_v}(f_v(\mathrm{Enc}'(x))_{R_v})$. Since both Enc' and Dec_{R_v} are linear and $f_v \in \mathcal{F}_{n,q,d}$, we have $g_v \in \mathcal{F}_{r,q,d}$. By Theorem 1.8, there exists a distribution D_{g_v} on $\{0,1\}^m \cup \{\text{same}\}$ such that

$$(\mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r}), \mathrm{nmExt}(g_v(\mathrm{U}_{\mathbb{F}_q^r})) \mid V' = v) \approx_{\varepsilon/2^{m+2}} (\mathrm{U}_m, \mathrm{copy}(D_{g_v}, \mathrm{U}_m)).$$

Define $D_{\mathcal{A},\mathcal{R},\mu} = \sum_v \Pr[V' = v] \cdot D_{g_v}$. By convexity of statistical distance,

$$(S, \widetilde{S}') = (\mathrm{nmExt}(\mathrm{U}_{\mathbb{F}_q^r}), \widetilde{S}') \approx_{\varepsilon/2^{m+2}} (\mathrm{U}_m, \mathrm{copy}(D_{\mathcal{A},\mathcal{R},\mu}, \mathrm{U}_m)),$$

which by triangle inequality implies

$$(S, \tilde{S}) \approx_{\varepsilon/2^{m+1}} (\mathrm{U}_m, \mathrm{copy}(D_{\mathcal{A},\mathcal{R},\mu}, \mathrm{U}_m))).$$

Finally by Lemma 3.14 and the fact that $\mathrm{Supp}(S) = \{0,1\}^m$ we can conclude that for every $s \in \{0,1\}^m$,

$$(\tilde{S} \mid S = s) \approx_\varepsilon \mathrm{copy}(D_{\mathcal{A},\mathcal{R},\mu}, s).$$

□

7 Open Questions

Obvious questions that arise from our work include improving the parameters (such as rate and error) of our non-malleable code against polynomials, and similarly obtaining seedless non-malleable extractors against polynomials with smaller error.

Another interesting direction is to construct such non-malleable codes and extractors against polynomials over smaller fields. In particular, over $\mathbb{F}_2$ would be the most interesting. We expect this to require significantly different ideas from our construction: we crucially rely on exponential sum estimates for our non-malleable extractor construction, and such estimates are not available over smaller fields.

More broadly, we believe it to be a very interesting question to construct non-malleable codes against other natural complexity classes (e.g., small-width branching programs, AC^0 with PARITY gates, etc.).

Acknowledgments

Marshall Ball is supported by an IBM Research PhD Fellowship. Tal Malkin and Marshall Ball: This work is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via Contract No. 2019-1902070006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Eshan Chattopadhyay and Jyun-Jie Liao are supported by NSF grant CCF-1849899. Li-Yang Tan is supported by NSF grant CCF-1921795.

References

[ADKO15] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 459–468. ACM, 2015.

[ADL18] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *SIAM Journal on Computing*, 47(2):524–546, 2018.

[ADN+18] Divesh Aggarwal, Ivan Damgard, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, Joao Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret-sharing schemes for general access structures. *IACR Cryptology ePrint Archive*, 2018:1147, 2018.

[AGM+15] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 375–397, 2015.

[AO19] Divesh Aggarwal and Maciej Obremski. A constant-rate non-malleable code in the split-state model. *IACR Cryptology ePrint Archive*, 2019:1299, 2019.

[BBR88] C.H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17:210–229, 1988.

[BDKM16] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In *TCC*, 2016.

[BDSG+18] Marshall Ball, Dana Dachman-Soled, Siyao Guo, Tal Malkin, and Li-Yang Tan. Non-malleable codes for small-depth circuits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 826–837. IEEE, 2018.

[BGW19] Marshall Ball, Siyao Guo, and Daniel Wichs. Non-malleable codes for decision trees. *IACR Cryptology ePrint Archive*, 2019:379, 2019.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM J. Comput.*, 36(4):1095–1118, December 2006.

[Bla79] George R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, 1979.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 01(01):1–32, 2005.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[BS18] Saikrishna Badrinarayanan and Akshayaram Srinivasan. Revisiting non-malleable secret sharing. *IACR Cryptology ePrint Archive*, 2018:1144, 2018.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *Theory of Cryptography Conference*, pages 440–464. Springer, 2014.

[CGH+85] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *STOC*, 2016.

[CL17] Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1171–1184. ACM, 2017.

[CL19] Eshan Chattopadhyay and Xin Li. Non-malleable codes, extractors and secret sharing for interleaved tampering and composition of tampering. Technical report, Cryptology ePrint Archive, Report 2018/1069, 2018., 2019.

[CMTV15] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. In *Theory of Cryptography Conference*, pages 532–560. Springer, 2015.

[CS09] Mahdi Cheraghchi and Amin Shokrollahi. Almost-uniform sampling of points on high-dimensional algebraic varieties. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 277–288, 2009.

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 306–315, 2014.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019.

[DGW09] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

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

[DKSS09] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 181–190, 2009.

[DLWZ14] Yevgeniy Dodis, Xin Li, Trevor D Wooley, and David Zuckerman. Privacy amplification and nonmalleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[DPW18] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, 2018.

[Dus10] Pierre Dusart. Estimates of some functions over primes without RH. *arXiv preprint arXiv:1002.0442*, 2010.

[Dvi12] Zeev Dvir. Extractors for varieties. *Computational complexity*, 21(4):515–572, 2012.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, pages 601–610, 2009.

[GK18a] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 685–698. ACM, 2018.

[GK18b] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, pages 501–530, 2018.

[GMW18] Divya Gupta, Hemanta K Maji, and Mingyuan Wang. Constant-rate non-malleable codes in the split-state model. Technical report, Technical Report Report 2017/1048, Cryptology ePrint Archive, 2018.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1128–1141. ACM, 2016.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

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

[HW98] Ming-Deh A. Huang and Yiu-Chung Wong. An algorithm for approximate counting of points on algebraic sets over finite fields. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 514–527. Springer, 1998.

[KOS17] Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Four-state non-malleable codes with explicit constant rate. In *Theory of Cryptography Conference*, pages 344–375. Springer, 2017.

[KOS18] Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Non-malleable randomness encoders and their applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 589–617. Springer, 2018.

[LCG$^+$19] Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Non-malleable secret sharing against affine tampering. *arXiv preprint arXiv:1902.06195*, 2019.

[LF04] Jérôme Lacan and Jérôme Fimes. Systematic MDS erasure codes based on vandermonde matrices. *IEEE Communications Letters*, 8(9):570–572, 2004.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1144–1156, 2017.

[Li19] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, pages 28:1–28:49, 2019.

[Lu02] Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In *Annual International Cryptology Conference*, pages 257–271. Springer, 2002.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

[Rab80] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM J. Comput.*, 9(2):273–280, 1980.

[Rao07] Anup Rao. An exposition of Bourgain's 2-source extractor. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

[Rob88] Guy Robin. Permanence de relations de récurrence dans certains développements asymptotiques. *Pub. Inst. Math. Beograd*, 43(57):17–25, 1988.

[Ros39] Barkley Rosser. The n-th prime is greater than nlogn. *Proceedings of the London Mathematical Society*, 2(1):21–44, 1939.

[RS60] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.

[Sch79] Jacob T Schwartz. Probabilistic algorithms for verification of polynomial identities. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 200–215. Springer, 1979.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sin64] Richard C. Singleton. Maximum distance q -nary codes. *IEEE Trans. Information Theory*, 10(2):116–118, 1964.

[TSZ04] Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004.

[Wei48] André Weil. On some exponential sums. *Proceedings of the National Academy of Sciences of the United States of America*, 34(5):204, 1948.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 216–226. Springer, 1979.

[Zuc06] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 681–690, 2006.

A Extraction from arbitrary affine source

To get a non-malleable secret sharing scheme against polynomial tampering in the non-adaptive setting (i.e., the choice of the tampering function does not depend on the shares), the original scheme in [LCG+19] (where they use an arbitrary erasure code instead of the truncated MDS code that we use in Section 6) suffices. This follows directly from [LCG+19] relying on the additional fact that the non-malleable extractor constructed in Theorem 1.8 (see Section 4) is also an extractor for affine sources. In this section we include a proof that our non-malleable extractor is indeed a (standard) extractor for affine sources. This is stated below as Lemma A.1.

For convenience of the reader, we first recall the setup and construction of the non-malleable extractor from Section 4:

For any integers n, d and any $\varepsilon > 0$, let $q > Cn^2d^4\varepsilon^2 \log(nd/\varepsilon)$ be a prime, for some large enough constant C. Now define the function $h : \mathbb{F}_q^n \to \mathbb{F}_q$ as

$$h(x_1, \ldots, x_n) = \sum_{i=1}^{n} (x_i^{c_{2i-1}} + x_i^{c_{2i}}),$$

where the $c_i = B(dn + 1) + Bi$, $B \leq O(\log q)$. Let $\sigma : \mathbb{F}_q \to \{0, 1\}^m$ be the mapping from Lemma 3.3. We define the non-malleable extractor: $\mathrm{nmExt}(x) = \sigma(h(x))$.

Lemma A.1. *For any affine source Y on $\mathbb{F}_q^n$ of dimension ≥ 1, we have*

$$|\mathrm{nmExt}(Y) - \mathrm{U}_m| \leq \varepsilon.$$

Proof. Since any affine source can be written as a convex combination of sources that are flat on lines (i.e, affine source of dimension 1), without loss of generality suppose the source Y is a flat source of dimension 1. Thus, assume Y is flat on some line $L_{a,b} = \{a + tb : t \in \mathbb{F}_q\}$, where $a, b \in \mathbb{F}_q^n$ and b is not the all 0's vector.

We first claim that the polynomial the univariate polynomial $h_{a,b}(t) = h(a+tb)$ is a non-constant polynomial (over $\mathbb{F}_q$) of degree at most $O(n \log q)$. The upper bound on the degree is direct from the fact that $c_{2n} \leq O(n \log q)$. Next note that

$$h_{a,b}(t) = \sum_{j \in [n]} \left((a_j + tb_j)^{c_{2j-1}} + (a_j + tb_j)^{c_{2j}}\right).$$

since $b \neq 0^n$, the set $S = \{j \in [n] : b_j \neq 0\}$ is non-empty and let i be the largest integer in S. It is now easy to see that $deg(h_{a,b}) = c_{2i}$. Indeed the coefficient of $t^{c_{2i}}$ in the polynomial $h_{a,b}$ is $b_i^{c_{2i}}$ which by assumption is non-zero. This completes the proof that $h_{a,b}$ is non-constant polynomial of degree at most $O(n \log q)$.

It now follows by Theorem 3.1, that for any non-trivial additve character χ of $\mathbb{F}_q$, we have

$$\left|\mathbb{E}_{t \in F_q}[\chi(h_{a,b}(t))]\right| \leq O(n \log q)/\sqrt{q}.$$

Thus, by Lemma 3.2, it follows that

$$|\sigma(h(Y)) - \mathrm{U}_m| \leq 2^{m/2} \cdot \frac{O(n \log q)}{\sqrt{q}} + 2^m \cdot \frac{1}{q}.$$

The bound now follows using the fact that $q > Cn^2 d^4 \varepsilon^2 \log(nd/\varepsilon)$ and $m = \nu \cdot \log q$ for some small enough constant ν. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il