On the Local Leakage Resilience of Linear Secret Sharing Schemes*

Fabrice Benhamouda† Akshay Degwekar‡ Yuval Ishai§ Tal Rabin¶

June 3, 2019

Abstract

We consider the following basic question: to what extent are standard secret sharing schemes and protocols for secure multiparty computation that build on them resilient to leakage? We focus on a simple *local leakage* model, where the adversary can apply an arbitrary function of a bounded output length to the secret state of each party, but cannot otherwise learn joint information about the states.

We show that additive secret sharing schemes and high-threshold instances of Shamir's secret sharing scheme are secure under local leakage attacks when the underlying field is of a large prime order and the number of parties is sufficiently large. This should be contrasted with the fact that any linear secret sharing scheme over a small characteristic field is clearly insecure under local leakage attacks, regardless of the number of parties. Our results are obtained via tools from Fourier analysis and additive combinatorics.

We present two types of applications of the above results and techniques. As a positive application, we show that the "GMW protocol" for honest-but-curious parties, when implemented using shared products of random field elements (so-called "Beaver Triples"), is resilient in the local leakage model for sufficiently many parties and over certain fields. This holds even when the adversary has *full access* to a constant fraction of the views. As a negative application, we rule out multiparty variants of the share conversion scheme used in the 2-party homomorphic secret sharing scheme of Boyle et al. (Crypto 2016).

*A preliminary version of this paper appeared in [BDIR18].

†IBM Research. Email: fabrice.benhamouda@normalesup.org. Research supported by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236.

‡MIT. Email: akshayd@mit.edu. Part of the work was done when the author was a summer intern at IBM Research. Research supported in part by NSF Grants CNS-1413920 and CNS-1350619, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

§Technion. Email: yuvali@cs.technion.ac.il. Research supported in part by ERC grant 742754, ISF grant 1709/14, and NSF-BSF grant 2015782, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India.

¶Algorand Foundation. Email: tal@algorand.foundation. Research done while at IBM Research and supported by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236.

Contents

1 Introduction

The recent attacks of Meltdown and Spectre [KGG+18, LSG+18] have brought back to the forefront the question of side-channel leakage and its effects. Starting with the early works of Kocher et al. [Koc96, KJJ99], side-channel attacks have demonstrated vulnerabilities in cryptographic primitives. Moreover, there are often inherent tradeoffs between efficiency and leakage resilience, where optimizations increase the susceptibility to side-channel attacks.

A large body of work on the theory of *leakage-resilient cryptography* (cf. [BBR88, BBCM95, Riv97, DSS01, CDH+00, MR04, DP08, AGV09]) studies the possibility of constructing cryptographic schemes that remain secure in the presence of partial leakage of the internal state. One prominent direction of investigation has been designing leakage-resilient cryptographic protocols for general computations [ISW03, FRR+10, DF12, Rot12, GR15, GIM+16].

The starting point for most of these works is the observation that some standard cryptographic schemes are vulnerable to very simple types of leakage. Moreover, analyzing the leakage resilience of others seems difficult. This motivates the design of new cryptographic schemes that deliver strong provable leakage resilience guarantees.

In this work, we forgo designing *special-purpose* leakage-resilient schemes and focus on studying the properties of existing common designs. We want to understand:

To what extent are standard *cryptographic schemes leakage resilient?*

We restrict our attention to *linear* secret sharing schemes and secure multiparty computation (MPC) protocols that build on them. In particular, we would like to understand the leakage resilience properties of the most commonly used secret sharing schemes, like additive secret sharing and Shamir's scheme, as well as simple MPC protocols that rely on them.

Analyzing existing schemes has a big advantage, as it can potentially allow us to enjoy their design benefits while at the same time enjoying a strong leakage-resilience guarantee. Indeed, classical secret sharing schemes and MPC protocols have useful properties which the specially designed leakage-resilient schemes are not known to achieve. For instance, linear secret sharing schemes can be manipulated via additive (and sometimes multiplicative) homomorphism, and standard MPC protocols can offer resilience to faults and to a large number of *fully corrupted* servers. Finally, classical schemes are typically more efficient than special-purpose leakage-resilient schemes.

Local Leakage. We study leakage resilience under a simple and natural model of *local leakage* attacks. To motivate the model, consider servers sharing some secret data and possibly performing some computation on their shares. The local leakage model has the following three properties: (1) The attacker can leak information about each server's state *locally*, independently of the other servers' states; this is justified by physical separation. (2) Only a *few bits of information* can be leaked about the internal state of each server; this is justified by the limited precision of measurements of physical quantities such as time or power. (3) The leakage is *adversarial*, in the sense that the adversary can decide what function of the secret state to leak. This is due to the fact that the adversary may have permission to legally execute programs on the server or have other forms of influence that can somewhat control the environment.

The local leakage model we consider is closely related to other models that were considered in the literature under the names "only computation leaks" (OCL) [MR04, BDL14, GR15, DLZ15], "intrusion resilience" [DP07], or "bounded communication leakage" [GIM+16]. These alternative models are typically more general in that they allow the leakage to be *adaptive*, or computable by an interactive protocol, whereas the leakage model we consider is non-adaptive.

Despite its apparent simplicity, our local leakage model can be quite powerful and enable very damaging attacks. In particular, in any linear secret sharing scheme over a field $\mathbb{F}_{2^k}$ of characteristic 2, an adversary can learn a bit of the secret by leaking just one bit from each share. Surprisingly, in the case of Shamir's scheme, full recovery of a multi-bit secret is possible, in some settings, by leaking only one bit from each share [GW17]. Some of the most efficient implementations of MPC protocols (such as the ones in [DPSZ12, KOS16, AFL+16]) are based on secret sharing schemes over $\mathbb{F}_{2^k}$ and are thus susceptible to such an attack.

As mentioned earlier, most prior works on leakage-resilient cryptography (see Section 1.2 below) design *special-purpose* leakage-resilient schemes. These works have left open the question of analyzing (variants of) standard schemes and protocols. Such an analysis is motivated by the hope to obtain better efficiency and additional security features.

1.1 Our Results

We obtain three kinds of results. First, we analyze the local leakage resilience of linear secret sharing schemes. Then, we apply these results to prove the leakage resilience of some natural MPC protocols. Finally, we present a somewhat unexpected application of these techniques to rule out the existence of certain *local share conversion* schemes. Our results are based on Fourier analysis techniques developed in the context of additive combinatorics. See Section 1.2 for details. We now give a more detailed overview of these results.

Leakage resilience of linear secret sharing schemes. In a linear secret sharing scheme over a finite field $\mathbb{F}$, the secret is an element $s \in \mathbb{F}$ and the share obtained by each party consists of one or more linear combinations of s and some random field elements. Two commonly used linear secret sharing schemes are the *additive* scheme, where the shares are random field elements that add up to the secret, and *Shamir's scheme*, where the shares are evaluations of a random degree-bounded polynomial whose free coefficient is equal to the secret.

We consider a scenario where n parties hold a linear secret sharing of either s_0 or s_1 specified by the adversary $\mathcal{A}$. (Due to linearity, we can assume without loss of generality that $s_0 = 0$ and $s_1 = 1$.) The adversary can also specify arbitrary leakage functions that output from each party's share m bits of leakage. The adversary's goal is to determine if the secret shared is s_0 or s_1. In this setting, we prove the following theorems.

Theorem 1.1 (Informally, Additive Secret Sharing)**.** *Additive secret sharing scheme over $\mathbb{F}_p$ is local leakage resilient even when up to $\log_2(p) - 1$ bits (namely, all but one bit) are leaked from every share. Concretely, the adversary's distinguishing advantage, in distinguishing between any two secrets, is at most $p \cdot 2^{-\Omega(n/p^2)}$ where n is the number of parties. In particular, when p is fixed and n tends to infinity, the advantage is $2^{-\Omega(n)}$.*

For a more precise statement see Corollaries 4.8, 4.10, and 4.11. There are many other parameter settings possible, for example if $p > n$, then additive secret sharing is leakage resilient when $(\log p)/4$ bits are leaked from each share. The adversary's advantage degrades as $2^{-\Omega(\sqrt{n})}$. In contrast to the theorem above, if the additive secret sharing were over $\mathbb{F}_{2^k}$, the adversary could distinguish between the two secrets by just leaking the least significant bit of each share and adding those up to reveal the least significant bit of the secret.

We show the following result for Shamir's secret sharing.

Theorem 1.2 (Informally, Shamir's Secret Sharing). *Let $p > n$ be a prime, where n is the number of parties. Then, (n, t)-Shamir's secret sharing*[1] *over $\mathbb{F}_p$ is local leakage resilient for the following parameters:*

1. *$t = \alpha n$ for some constant $\alpha < 1$ when a constant number of bits are leaked from each share. The adversary's advantage degrades as $2^{-\Omega(n)}$. When 1 bit is leaked, $\alpha = 0.85$ suffices.*
2. *$t = n - n^{1/4}$ when a quarter of the bits ($(\log p)/4$ of $\log p$) are leaked from every share, where $n < p \leq 2n$. The adversary's advantage degrades as $2^{-\Omega(\sqrt{n})}$.*

For a more precise statement see Corollaries 4.9, 4.12, and 4.13.

Shamir's secret sharing is typically used with threshold $t = n/2$ or $t = n/3$, in which case the above result is not applicable. While we cannot prove local leakage resilience, we do not know of attacks in this parameter regime. We conjecture the following:

Conjecture 1.3 (Shamir's Secret Sharing). *For large enough n, $(n, t = \alpha n)$-Shamir's secret sharing is 1-bit local leakage resilient for any constant $\alpha > 0$.*

Observe that proving the conjecture for a specific constant α immediately implies the conjecture for any constant $\alpha' > \alpha$. This follows from the fact that $(n, \alpha n)$-Shamir's shares can be locally converted to random $(n, \alpha' n)$-Shamir's shares for $\alpha' > \alpha$.[2]

Application to leakage-resilient MPC. We use the leakage resilience of linear secret sharing schemes to show that the *honest-but-curious* variant of the GMW [GMW87] protocol with a "Beaver Triples" setup [Bea91] (that we call GMW with shared product preprocessing) is local leakage resilient.

For the MPC setting, we modify the leakage model as follows to allow for a stronger adversary. The adversary $\mathcal{A}$ is allowed to corrupt a fraction of the parties, see their shares and views of the entire protocol execution. In addition, $\mathcal{A}$ specifies local leakage functions for the non-corrupted parties and receives the corresponding leakage on their individual views.

The honest-but-curious GMW protocol with shared product preprocessing works as follows. The parties wish to evaluate an arithmetic circuit C on an input x. The parties receive random shares of the input x under a linear secret sharing scheme and random shares of Beaver triples under the same scheme.[3] The protocol proceeds gate by gate where the parties maintain a secret sharing of the value at each gate. For input, addition and inverse (-1) gates, parties locally manipulate their existing shares to generate the shares for these gates. For multiplication gates, where we multiply z_1 and z_2 to get z, the parties first construct $z_1 - a$ and $z_2 - b$ by broadcasting the differences of the shares of the inputs and of the shares of a and b from a fresh Beaver triple (a, b, ab). Then the parties can locally construct a secret sharing of $z = z_1 \cdot z_2$ by using the following relation:

$$z = (z_1 - a)(z_2 - b) + a(z_2 - b) + b(z_1 - a) + ab\ .$$

[1] In the whole paper, a (n, t)-Shamir's secret sharing scheme or Shamir's secret sharing scheme with (reconstruction) threshold t uses polynomials of degree $t - 1$, so that the secret cannot be recovered from a collusion of less t parties. The secret can be recovered from the shares of t parties.

[2] This can be done by locally adding shares of an arbitrary $(n, \alpha' n)$-Shamir's sharing of 0 to the given $(n, \alpha n)$-Shamir's shares for $\alpha' > \alpha$.

[3] A Beaver triple consists of (a, b, ab) where a, b are randomly chosen field elements.

We show that when the underlying secret sharing scheme is local leakage resilient, this protocol can also tolerate local leakage. We can prove leakage resilience in a simulation-based definition. See Section 5 for details. Informally, when the additive secret sharing scheme is used, we show the following.

Theorem 1.4 (Informally, Leakage Resilience of GMW)**.** *For any prime p, the GMW protocol with shared product preprocessing and additive secret sharing over $\mathbb{F}_p$ is local leakage resilient. The adversary can corrupt $n/2$ parties, learn their entire state and, then locally leak a constant number of bits each from all the uncorrupted parties. The adversary's distinguishing advantage for this attack is $2^{-\Omega(n)}$.*

On the impossibility of local share conversion. In the problem of local share conversion [CDI05, BIKO12], n parties hold a share of a secret s under a secret sharing scheme $\mathcal{L}$. Their goal is to *locally*, without interaction, convert their shares to shares of a related secret s' under a different secret sharing scheme $\mathcal{L}'$ such that (s, s') satisfy a pre-specified relation R. We assume R is not trivial in the sense that it is not permissible to map shares of every secret s to shares of a fixed constant. Local share conversion has been used to design protocols for Private Information Retrieval [BIKO12]. More recently, different kinds of local share conversion were used to construct Homomorphic Secret Sharing (HSS) schemes [BGI16, DHRW16, FGJI17]. Using techniques similar to the ones for leakage resilience, we rule out certain nontrivial instances of local share conversion. We first state our results and then discuss their relevance to constructions of HSS schemes.

Theorem 1.5 (Informally, Impossibility of Local Share Conversion)**.** *Three-party additive secret sharing over $\mathbb{F}_p$, for any prime $p > 2$, cannot be converted to additive secret sharing over $\mathbb{F}_2$, with constant success probability ($> 5/6$), for any non-trivial relation R on the secrets.*

The proof of this result uses a Fourier analysis technique similar to the analysis of the Blum-Luby-Rubinfeld linearity test [BLR93]. We also show a similar impossibility result for Shamir's secret sharing. This result relies crucially on a technique by Green and Tao [GT10]. We elaborate more in Section 2. See Theorems 6.5 and 6.6 for the precise general statements.

Relevance to HSS Schemes. At the heart of the DDH-based 2-party HSS scheme of Boyle et al. [BGI16] and its Paillier-based variant of Fazio et al. [FGJI17] is an efficient local share conversion algorithm of the following special form. The two parties hold shares g^x and g^y respectively of $b \in \{0, 1\}$, such that $g^b = g^x \cdot g^y$. The conversion algorithm enables them to *locally* compute additive shares of the bit b over the *integers* $\mathbb{Z}$, with small (inverse polynomial) failure probability. Note that this implies similar conversion to additive sharing over $\mathbb{F}_2$. One approach to constructing 3-party HSS schemes would be to generalize this local share conversion scheme to 3 parties, i.e., servers holding random g^x, g^y and g^z respectively, such that $g^b = g^x \cdot g^y \cdot g^z$, can locally convert these shares to additive shares of the bit b over integers. We rule out this approach by showing that even when given the exponents x, y and z in the clear (i.e. $x + y + z = b$ over $\mathbb{F}_p$), locally computing additive shares of b over $\mathbb{F}_2$ (or the integers) with small failure probability is impossible. A similar share conversion from (noisy) additive sharing over $\mathbb{F}_p$ to additive sharing over $\mathbb{F}_2$ was used by Dodis et al. [DHRW16] (and recently by Boyle et al. [BKS19]) to obtain an LWE-based construction of 2-party HSS and spooky encryption. However, in this case there is an alternative route of reducing the multiparty case to the 2-party case. Our negative result only rules out a *direct* generalization of the 2-party solution to the multi-party case.

1.2 Related Work

Our work was inspired by the surprising result of Guruswami and Wootters [GW17] mentioned above. This work turned attention to the fact that some natural linear secret sharing schemes miserably fail to offer local leakage resilience over fields of characteristic 2, in that leaking only one bit from each share is sufficient to fully recover a multi-bit secret.

The traditional "leakage" model considered in multiparty cryptography allows the adversary to fully corrupt up to t parties and learn their entire secret state. This t-bounded leakage model motivated secret sharing schemes designed to protect *information* [Sha79, Bla79] and secure multiparty computation (MPC) protocols designed to protect *computation* [Yao86, GMW87, BGW88, CCD88]. The same leakage model was also considered at the hardware level, where parties are replaced by atomic gates [ISW03]. The t-bounded leakage considered in all these works is quite different from the local leakage model we consider: we allow *partial* leakage from *every* secret state, whereas the t-bounded model allows *full* leakage from up to t secret states. While resilience to t-bounded leakage was shown to imply resilience to certain kinds of "noisy leakage" [FRR+10, DDF14] or "low-complexity leakage" [BIVW16], it clearly does not imply local leakage resilience in general. Indeed, additive secret sharing over $\mathbb{F}_{2^k}$ is highly secure in the t-bounded model and yet is totally insecure in the local leakage model.

The literature on leakage-resilient cryptography is extensive, thus we discuss a few of the most relevant works. Secret-sharing schemes that offer local leakage resilience were first constructed by Dziembowski and Pietrzak [DP07]. Their scheme involved an interactive reconstruction procedure, which was needed for allowing the reconstruction to access only small part of the shares. Simpler constructions (without the latter efficiency feature) were proposed by Davì et al. [DDV10]. In particular, they presented a simple two-party scheme based on any two-source extractor, such as the inner-product extractor. For stronger or more general constructions of leakage-resilient secret-sharing schemes, see the recent works of Goyal and Kumar [GK18], Srinivasan and Vasudevan [SV18], and Kumar et al. [KMS18] and references therein. All the above works design specialized (and non-linear) secret-sharing schemes that have strong leakage resilience properties. In contrast, we are interested in exploring the leakage resilience of standard (linear) schemes.

Subsequent to our work, Nielsen and Simkin [NS19] studied the question of leakage resilience of Shamir's secret sharing, and more generally information theoretic secret sharing schemes. They show that, in the local leakage model, when the total number of bits leaked exceeds total entropy of all the shares jointly, the secret is revealed. In our results, the total entropy of the shares is significantly higher than the total bits leaked. Closing this gap and showing either better leakage resilience or better attacks remains an open question.

We turn to survey some relevant works on leakage-resilient MPC. Boyle et al. [BGK11] consider the problem of leakage-resilient coin-tossing and reduce it to a certain kind of leakage-resilient verifiable secret sharing. Here too, a new construction of (nonlinear) secret sharing is developed in order to achieve these results.

Goldwasser and Rothblum [GR15] give a general transformation that takes any algorithm and creates a related algorithm that computes the same function and can tolerate leakage. This approach can be viewed as a special-purpose MPC protocol for a constant number of parties that offers local leakage resilience (and beyond) [BDL14]. However, this construction is quite involved and offers poor concrete leakage resilience and efficiency overhead.

Most relevant to our MPC-related results is the recent work of Goyal et al. [GIM+16] on leakage-resilient secure two-party computation (see also [GIW17]). This work analyzes the resilience of a GMW-style protocol under a similar (in fact, more general) type of leakage to the local leakage model we consider. One key difference is that the protocol from [GIM+16] modifies the underlying circuit (incurring a considerable overhead) whereas we apply the GMW protocol to the original circuit. Also, our approach applies to a large number of parties of which a large fraction can be entirely corrupted, whereas the construction in [GIM+16] is restricted to the two-party setting.

Our results use techniques developed in the context of additive combinatorics. See Tao and Vu [TV06] for an exposition on Fourier analysis methods used in additive combinatorics. The works most relevant to ours are works by Green and Tao [GT10] and follow-ups by Gowers and Wolf [GW10, GW11a, GW11b]. The relation of these works and their techniques to ours is discussed in Section 2.4.

1.3 Publication Note

An abridged version of this paper appeared in the proceedings of Crypto 2018 [BDIR18]. The current version provides better security bounds using new ideas to bound the sum of Fourier coefficients appearing in the analysis. In particular, while the local leakage resilience of Shamir's secret sharing was only proved for threshold $t = n - O(\log n)$ in [BDIR18], here we show local leakage resilience for threshold $t = \Theta(n)$ (for constant-size leakage, when the number of parties n goes to infinity). In addition, the current version includes all the security proofs, improves several notations, fixes some minor mistakes, and discusses additional related works.

2 Overview of the Techniques

2.1 Leakage Resilience of Secret Sharing Schemes

Very simple local leakage attacks exist for linear secret sharing schemes over small characteristic fields. These attacks stem from the existence of small additive subgroups in these fields. This gives rise to the hope that linear schemes over fields of prime order, that lack such subgroups, are leakage resilient. We start by considering the simpler case of additive secret sharing.

Additive secret sharing. We define $\mathsf{AddSh}(s)$ to be a function that outputs random shares $s^{(1)}, \ldots, s^{(n)}$ such that $\sum s^{(i)} = s$.

Let $\boldsymbol{\tau} = \tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)}$ be some leakage functions. We want to show that for all secrets $s_0, s_1 \in \mathbb{F}$, the leakage distributions are statistically close. That is,

$$\left\{\boldsymbol{\tau}(\mathbf{s}) : \mathbf{s} \leftarrow \mathsf{AddSh}(s_0)\right\} \approx \left\{\boldsymbol{\tau}(\mathbf{s}) : \mathbf{s} \leftarrow \mathsf{AddSh}(s_1)\right\} \ ,$$

where $\boldsymbol{\tau}(\mathbf{s}) = \tau^{(1)}(s^{(1)}), \ldots, \tau^{(n)}(s^{(n)})$ is the total leakage the adversary sees on the shares $\mathbf{s} = s^{(1)}, s^{(2)}, \ldots, s^{(n)}$.

We know that there is a local leakage attack on $\mathbb{F}_{2^k}$: simply leak the least significant bit (lsb) from all the parties and add the outputs to reconstruct the lsb of the secret. What enables the attack on $\mathbb{F}_{2^k}$ while $\mathbb{F}_p$ is unaffected?

To understand this difference, it is instructive to start with an example. Let us consider additive secret sharing over $\mathbb{F}_{2^k}$ for 3 parties. We know that,

$$\mathsf{lsb}(s) = \mathsf{lsb}(s^{(1)}) + \mathsf{lsb}(s^{(2)}) + \mathsf{lsb}(s^{(3)}) \ .$$

This attack works because $\mathbb{F}_{2^k}$ has many subgroups that are closed under addition. Let $A_0 = \mathsf{lsb}^{-1}(0)$ and $A_1 = \mathsf{lsb}^{-1}(1)$. The set A_0 is an additive subgroup of $\mathbb{F}_{2^k}$ and A_1 is a coset of A_0. Furthermore, the lsb function is a homomorphism from $\mathbb{F}_{2^k}$ to the quotient group[4] $\mathbb{F}_{2^k}/A_0$. The lsb leakage tells us which coset each share $s^{(j)}$ is in. Then by adding these leakages, we can infer whether $s \in A_0$ or $s \in A_1$ (i.e., to which coset it belongs).

Let us consider the analogous situation over $\mathbb{F}_p$ for a prime p. The group $\mathbb{F}_p$ does not have any subgroups. In fact, it has an opposite kind of expansion property: that adding *any* two sets results in a larger set.

Theorem 2.1 (Cauchy-Davenport Inequality)**.** *Let $A, B \subseteq \mathbb{F}_p$. Let $A + B = \{a + b : a \in A \text{ and } b \in B\}$. Then,*

$$|A + B| \geq \min(p, |A| + |B| - 1) \ .$$

So, if we secret shared a random secret over $\mathbb{F}_p$ and got back leakage output indicating that $s^{(1)} \in B_1$, $s^{(2)} \in B_2$, and $s^{(3)} \in B_3$, we can infer that $s \in B_1 + B_2 + B_3$. But because of this expansion property, the set $B_1 + B_2 + B_3$ is a lot larger than the sets B_i's individually. This is in contrast to the $\mathbb{F}_{2^k}$ case where e.g. $A_0 + A_1$ was the same size as A_0.

This gives an idea of why the lsb attack does not work. Some information is lost because of expansion. This is not sufficient for us though. What we need to show is stronger. We want to show that even given the leakage, the secret is *almost completely hidden*. This is a more "distributional" statement.

We model it as follows: Let us say that we have n parties where party j holds the share $s^{(j)}$. The adversary $\mathcal{A}$ has specified leakage functions $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$ and received back the leakage $\boldsymbol{\ell} = \ell_1, \ell_2, \ldots, \ell_n$ where $\ell_j = \tau^{(j)}(s^{(j)})$: the leakage on the j-th share. We want to show that even conditioned on this leakage, the probability that the secret was s_0 vs s_1 is close to a half. That is, we want to show the following:

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}(s_0)}[\boldsymbol{\tau}(\mathbf{s}) = \boldsymbol{\ell}] \approx \Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}(s_1)}[\boldsymbol{\tau}(\mathbf{s}) = \boldsymbol{\ell}] \ . \tag{1}$$

Below, we will sketch an argument showing that leaking from the additive shares of 0 is statistically close to leaking from a vector of uniformly random elements: if U is the uniform distribution over $\mathbb{F}_p^n$,

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}(0)}[\boldsymbol{\tau}(\mathbf{s}) = \boldsymbol{\ell}] \approx \Pr_{\mathbf{u} \leftarrow U}[\boldsymbol{\tau}(\mathbf{u}) = \boldsymbol{\ell}] \ . \tag{2}$$

This argument is not specific to 0 and shows that additive secret sharing is local leakage resilient. More precisely, from Eq. (2), Eq. (1) follows by a simple hybrid argument as shares of any other secret s are simply shares of 0 with the secret s added to the first party's share. That is, let $\mathbf{e}_1 = (1, 0, 0, \ldots, 0)$,

$$\{\mathbf{s} + s \cdot \mathbf{e}_1 : \mathbf{s} \leftarrow \mathsf{AddSh}(0)\} \equiv \{\mathbf{y} : \mathbf{y} \leftarrow \mathsf{AddSh}(s)\} \ .$$

We want to understand the probability of getting a particular value of leakage under both the uniform distribution and the additive shares of 0. To understand this probability better, let us consider the following operator:

$$\Lambda(f_1, f_2, \ldots, f_n) = \mathop{\mathbb{E}}_{\mathbf{s} \leftarrow \mathsf{AddSh}(0)}\left[f_1(s^{(1)}) \cdot f_2(s^{(2)}) \cdots f_n(s^{(n)})\right] \ .$$

[4] To recall, in the quotient group $\mathbb{F}_{2^k}/A_0$, the elements are the cosets A_0, A_1. The sum of two cosets is the coset formed by the sum of elements of the first coset with elements of the second coset. Concretely, we have $A_0 + A_0 = A_0$, $A_0 + A_1 = A_1$, and $A_1 + A_1 = A_0$.

By picking the functions f_j's appropriately, we can model the probability of getting a particular value of leakage under the secret sharing. Define $1_{\ell_j} : \mathbb{F}_p \to \{0, 1\}$ as follows: $1_{\ell_j}(s) = 1$ if the output of the leakage function $\tau^{(j)}$ on input s is ℓ_j, i.e., $\tau^{(j)}(s) = \ell_j$ and, 0 otherwise. Notice that we can write the probability of leakage output being $\boldsymbol{\ell}$ in terms of the operator Λ as follows,

$$\Pr_{\mathbf{s}\leftarrow \mathsf{AddSh}(0)}[\boldsymbol{\tau}(\mathbf{s}) = \boldsymbol{\ell}] = \Lambda(1_{\ell_1}, 1_{\ell_2}, \ldots, 1_{\ell_n})\,.$$

The probability of the leakage being $\boldsymbol{\ell}$ on the uniform distribution is simply a product of the expectations:

$$\Pr_{\mathbf{u}\leftarrow U}[\boldsymbol{\tau}(\mathbf{u}) = \boldsymbol{\ell}] = \mathop{\mathbb{E}}_{\mathbf{u}\leftarrow U}[\mathbf{1}_{\boldsymbol{\ell}}(\mathbf{u})] = \mathop{\mathbb{E}}_{\mathbf{u}\leftarrow U}\left[1_{\ell_1}(u^{(1)}) \cdot 1_{\ell_2}(u^{(2)}) \cdots 1_{\ell_n}(u^{(n)})\right]$$

where $1_{\boldsymbol{\ell}}(\mathbf{u}) = 1_{\ell_1}(u^{(1)}) \cdot 1_{\ell_2}(u^{(2)}) \cdots 1_{\ell_n}(u^{(n)})$. So, we want to show:

$$\Lambda(1_{\ell_1}, 1_{\ell_2}, \ldots, 1_{\ell_n}) = \mathop{\mathbb{E}}_{\mathbf{u}\leftarrow U}[\mathbf{1}_{\boldsymbol{\ell}}(\mathbf{u})] + \varepsilon\,.$$

The tool we use to bound the difference $|\Lambda(\mathbf{1}_{\boldsymbol{\ell}}) - \mathbb{E}_{\mathbf{u}\leftarrow U}[\mathbf{1}_{\boldsymbol{\ell}}(\mathbf{u})]|$ is Fourier analysis. At the heart of this is the Poisson summation formula for the Λ operator: the Fourier spectrum of Λ takes a form highly similar to the definition of Λ as follows. For Λ defined over a linear code C:

$$\Lambda(f_1, f_2, \ldots, f_n) = \mathop{\mathbb{E}}_{\mathbf{s}\leftarrow C}\left[f_1(s^{(1)}) \cdot f_2(s^{(2)}) \cdots f_n(s^{(n)})\right]\,,$$

Λ can be equivalently represented on the dual code $C^\perp$ (see Lemma 4.16) as,

$$= \sum_{\vec{\alpha}\in C^\perp} \widehat{f_1}(\alpha_1) \cdot \widehat{f_2}(\alpha_2) \cdots \widehat{f_n}(\alpha_n)\,,$$

with the 'Fourier coefficients' $\widehat{f}(\alpha) = \mathbb{E}_{x\leftarrow\mathbb{F}_p}\left[f(x)\cdot\omega^{\alpha x}\right]$ where $\omega = \exp(2\pi i/p)$ is a root of unity. Observe that as $\widehat{1_\ell}(0) = \mathbb{E}_x[1_\ell(x)]$. So, $\mathbb{E}_{\mathbf{u}\leftarrow U}[\mathbf{1}_{\boldsymbol{\ell}}(\mathbf{u})] = \widehat{1_{\ell_j}}(0)\cdot\widehat{1_{\ell_j}}(0)\cdots\widehat{1_{\ell_n}}(0)$ is the term corresponding to the all-zeros codeword in the dual code. Hence, the error term we have to bound is the following:

$$\Lambda(\mathbf{1}_{\boldsymbol{\ell}}) - \mathop{\mathbb{E}}_{\mathbf{u}\leftarrow U}[\mathbf{1}_{\boldsymbol{\ell}}(\mathbf{u})] = \sum_{\vec{\alpha}\in C^\perp\setminus\{\vec{0}\}} \widehat{1_{\ell_1}}(\alpha_1)\cdot\widehat{1_{\ell_2}}(\alpha_2)\cdots\widehat{1_{\ell_n}}(\alpha_n)\,. \tag{3}$$

Note that, at this point, it is interesting to observe how the presence of subgroups (over $\mathbb{F}_{2^k}$) and the lack thereof (over $\mathbb{F}_p$) manifests itself. Over $\mathbb{F}_{2^k}$ because of the non-trivial subgroups, these non-zero Fourier coefficients can be large and hence the error term is not small. On the other hand, over $\mathbb{F}_p$, we can show that each non-zero Fourier coefficient is *strictly smaller* than the zero-th coefficient and noticeably so. This lets us bound the error term. First we elaborate on the large Fourier coefficient over $\mathbb{F}_{2^k}$ and we some intuition for bounds on $\mathbb{F}_p$.

Large coefficients over $\mathbb{F}_{2^k}$. Each Fourier basis function over $\mathbb{F}_{2^k}$ is indexed by a vector $\vec{\alpha} \in \{0,1\}^k$ and the Fourier coefficient for $\vec{\alpha}$ is given by $\widehat{f}(\vec{\alpha}) = \mathbb{E}_{\vec{x}\leftarrow\mathbb{F}_{2^k}}\left[f(\vec{x})(-1)^{\langle\vec{\alpha},\vec{x}\rangle}\right]$.[5] Over $\mathbb{F}_{2^k}$, non-zero Fourier coefficients can be as large as the zero-th coefficient, which is always the largest for binary valued functions.

[5] We abuse notation and sometimes consider elements of $\mathbb{F}_{2^k}$ as vectors in $\mathbb{F}_2^k$.

To use the running example, in the case of the lsb function, let $\tau^{(j)} = \mathsf{lsb}$ and consider the $1_{\mathsf{lsb}=1}$ to be the function which returns 1 if the lsb is 1 and 0 otherwise. So, $1_{\mathsf{lsb}=1}$ is 1 on the set A_1 and 0 on A_0. The non-zero Fourier coefficient indexed by $\vec{e}_k = (0, 0, \dots 0, 1) \in \{0,1\}^k$ is as large as the zero-th Fourier coefficient since: $\widehat{1}_{\mathsf{lsb}=1}(\vec{0}) = \mathbb{E}_{\vec{x}}\left[1_{\mathsf{lsb}=1}(\vec{x})\right] = 0.5$ as half of the inputs satisfy $\mathsf{lsb} = 1$, and also, $\widehat{1}_{\mathsf{lsb}=1}(\vec{e}_k) = \mathbb{E}_{\vec{x}}\left[1_{\mathsf{lsb}=1}(\vec{x}) \cdot (-1)^{x_k}\right] = \mathbb{E}_{\vec{x}}\left[1_{\mathsf{lsb}=1}(\vec{x}) \cdot (-1)\right] = -0.5$ because when $1_{\mathsf{lsb}=1}(x) = 1$, then $x_k = 1$ and $1_{\mathsf{lsb}=1}(\vec{x}) \cdot (-1)^{x_k} = -1$. So, these two Fourier coefficients are equally large in magnitude. Hence the error term can be quite large.

Bounding Fourier Coefficients on $\mathbb{F}_p$. Back to the prime order setting (i.e., the setting on which we focus), we want to bound $\widehat{1_{\ell_j}}(\alpha)$ for non-zero $\alpha \in \mathbb{F}_p$. For now, let us consider a single leakage function $\tau : \mathbb{F}_p \to \{0,1\}^m$. Observe that τ partitions $\mathbb{F}_p$ in to 2^m sets $A_1, A_2, \dots, A_{2^m}$ where each $A_\ell = 1_\ell^{-1}(1) = \left\{x \in \mathbb{F}_p : \tau(x) = \ell\right\}$. For simplicity, assume that each set A_i is approximately of size $p/2^m$ (actually, this is the hardest case). We want to understand,

$$\widehat{1_\ell}(\alpha) = \mathop{\mathbb{E}}_{y \leftarrow \mathbb{F}_p}\left[1_\ell(y) \cdot \omega^{\alpha y}\right] = \frac{\sum_{a \in A_\ell} \omega^{\alpha a}}{p} .$$

$\widehat{1_\ell}(0) = \sum_{a \in A_\ell} \omega^{0 \cdot a} = |A_\ell|/p > |\widehat{1_\ell}(\alpha)|$ for all $\alpha \neq 0$. Sums of the form $\sum_{b \in B} \omega^b$ are maximized when the set B is an interval (see Lemma 3.11 and Fig. 1a). Leveraging this, we can show that there is a constant $c_m < 1$ such that,

$$\max_{\alpha \neq 0} |\widehat{1_\ell}(\alpha)| \leq c_m \cdot |A_\ell|/p .$$

As written, this equation is only true for sets of size $p/2^m$ but arguments based on convexity allow us to plug this back into Eq. (3) and show that,

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U)) \leq \frac{1}{2} \cdot \left|C^\perp\right| \cdot c_m^t ,$$

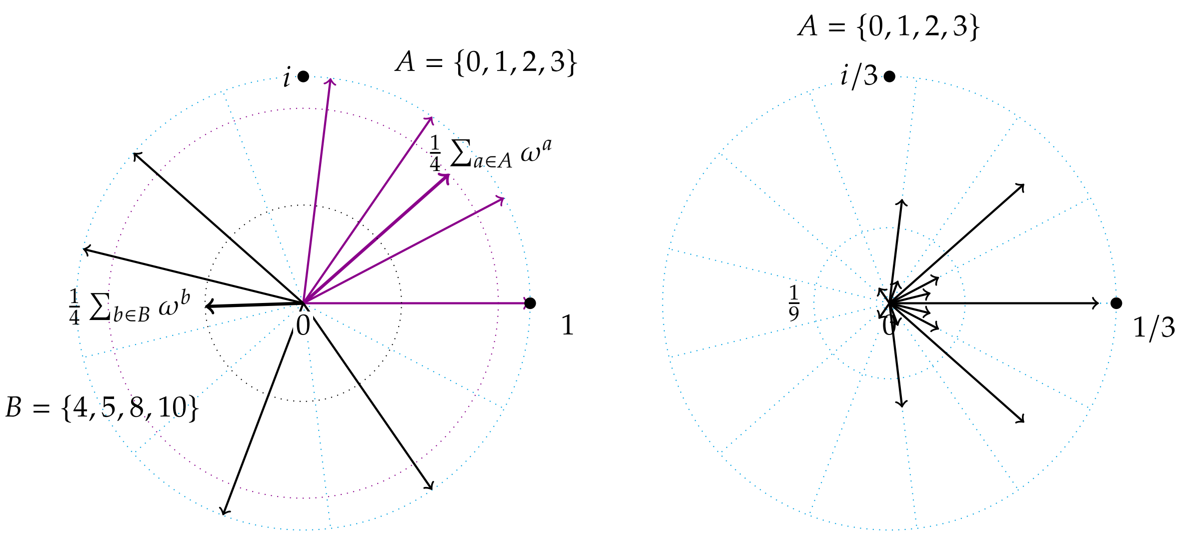
where SD denotes the statistical distance between the two distributions, $\boldsymbol{\tau}(C) = \left\{\boldsymbol{\tau}(\mathbf{s}) : \mathbf{s} \leftarrow C\right\}$, $\boldsymbol{\tau}(U) = \left\{\boldsymbol{\tau}(\mathbf{s}) : \mathbf{s} \leftarrow U\right\}$ (with U being the uniform distribution over $\mathbb{F}_p^n$), and t is the minimum distance of the dual code $C^\perp$. Formally, the theorem is stated in Theorem 4.5. The factor $|C^\perp|$ comes from summing over all dual codewords after using the triangle inequality.

When applied to the code $C = \mathsf{AddSh}(0)$, we have $|C^\perp| = p$ and $t = n$, and this implies that additive secret sharing is leakage resilient, proving Theorem 1.1. We can also apply the result to Reed Solomon Codes, the codes underlying (n, t)-Shamir's secret sharing. In this case, $|C^\perp| = p^{n-t+1}$ and hence this proof works only when $n - t = O(n/\log p)$ because we need $c_m^t \ll p^{n-t}$. Furthermore, this bound has a peculiar character that it becomes worse as the prime used increases.[6] This is unnatural.

Till now, we have utilized the fact that the largest non-zero Fourier coefficient is bounded away from the zero-th Fourier coefficient. To improve our bound, we next utilize another fact about Fourier coefficients: most non-zero Fourier coefficients are a lot smaller than the largest one. For an illustration of this fact, see Fig. 1b. In particular, Parseval's identity (Theorem 3.9(a)) implies that for any set A,

$$\|\widehat{1_A}\|_2^2 = \sum_{\alpha \in \mathbb{F}_p} |\widehat{1_A}(\alpha)|^2 = \mathop{\mathbb{E}}_{y \leftarrow \mathbb{F}_p}\left[1_A(y)^2\right] = \frac{|A|}{p}$$

[6] While the constant c_m has a some dependence on p, it decreases as p increases, it is dwarfed by the p^{n-t} term.



(a) Fourier Sums are maximized for intervals. (The scaling by 4 of the sums is for convenience.)

(b) Fourier Coefficients for $A = \{0,1,2,3\}$ over $\mathbb{F}_{13}$.

Figure 1: Illustrations of Fourier Sums and Coefficients

Hence, an "average" non-zero Fourier coefficient is of size approximately $\sqrt{|A|}/p$, a size lot smaller than $c_m|A|/p$, the maximum possible. We want to leverage this fact. And the way to do so is Cauchy-Schwarz inequality. We describe the idea in the case of additive secret sharing. In the general case, the manipulations are more involved. In the case of additive secret sharing, the dual code $C^{\perp} = \boldsymbol{\alpha} : \alpha \in \mathbb{F}_p$. Roughly speaking, we can bound the sum from Eq. (3) as,

$$\sum_{\alpha\in\mathbb{F}_p\setminus\{0\}} \left|\widehat{1_{\ell_1}}(\alpha)\cdot\widehat{1_{\ell_2}}(\alpha)\cdots\widehat{1_{\ell_n}}(\alpha)\right| \le \|\widehat{1_{\ell_1}}\|_2\cdot\|\widehat{1_{\ell_2}}\|_2\cdot\max_{\alpha\neq 0}|\widehat{1_{\ell_3}}(\alpha)|\cdots\max_{\alpha\neq 0}|\widehat{1_{\ell_n}}(\alpha)| \ .$$

This allows us to derive a sharper bound on the error, showing that for additive secret sharing $\mathrm{SD}(\boldsymbol{\tau}(C),\boldsymbol{\tau}(U)) \le \frac{1}{2}\cdot 2^m\cdot c_m^{t-2}$. And for general MDS codes, we can show a similar result that: For an $[n, t-1, n-t+2]$ code C (i.e., C is a linear subspace of $\mathbb{F}_p^n$ of dimension $t-1$ and such that the Hamming weight of any non-zero vector of C is at least $n-t+2$),

$$\mathrm{SD}(\boldsymbol{\tau}(C),\boldsymbol{\tau}(U)) \lesssim \frac{1}{2}\cdot 2^{5m(n-t)}\cdot c_m^t \ .$$

This bound has two desirable properties: first of all, it does not become worse as the prime increases, and secondly, it allows us to show that Shamir's secret sharing is leakage resilient when $t = cn$ for some constant c. For more precise statements and parameters see Sections 4.2.1 and 4.2.3.

2.2 Application to Leakage Resilience of MPC protocols

Given the leakage resilience of additive secret sharing over $\mathbb{F}_p$, we can show that the following *honest-but-curious* variant of the GMW protocol [GMW87] (GMW with shared product preprocessing) using Beaver Triples [Bea91] is leakage resilient. The protocol is described in Fig. 2. Recall that in our leakage model, the adversary $\mathcal{A}$ is allowed to corrupt a fraction of the parties, see their views of the entire protocol execution and then specify leakage functions $\tau^{(j)}$ for the non-corrupted parties and receive this leakage on their individual views.

GMW Protocol with Shared Product Preprocessing

Setup: Given an arithmetic circuit C over field $\mathbb{F}$ computing f. C has gates from the basis $\mathbb{B} = \{+, \times, -1\}$ where the -1 gate negates the input. We also have input gates that read a field element from the input.

Input Encoding: On input $\vec{x}$, randomly secret share $\vec{x}$ using additive secret sharing, i.e., $\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(n)} \leftarrow \mathsf{AddSh}(\vec{x})$. Party j gets $\vec{x}^{(j)}$.

Randomness: Let $G_\times$ be the set of multiplication gates in C. For each multiplication gate g in $G_\times$, generate a Beaver triple: $\mathbf{a}_g \leftarrow \mathsf{AddSh}(a_g)$, $\mathbf{b}_g \leftarrow \mathsf{AddSh}(b_g)$ and $(\mathbf{ab})_g \leftarrow \mathsf{AddSh}(a_g \cdot b_g)$ for $a_g, b_g \leftarrow \mathbb{F}$.

Protocol Π: Party j receives an input $\vec{x}^{(j)}$ and randomness $(a_g^{(j)}, b_g^{(j)}, (ab)_g^{(j)})_{g \in G_\times}$. The parties traverse the gates in the circuit C in a predetermined order where every gate is traversed only after its input gates. Let $\mathbf{z}_g$ denote the secret sharing of the value z_g at gate g. For each gate, the parties do the following:

1. If gate g is not a multiplication gate, the parties *locally* generate:

$$\mathbf{z}_g = \begin{cases} \mathbf{x}_i & \text{if } g \text{ is an input gate reading } x_i \\ -\mathbf{z}_{g_1} & \text{if } g \text{ is a } -1 \text{ gate with input } g_1 \\ \mathbf{z}_{g_1} + \mathbf{z}_{g_2} & \text{if } g \text{ is a } + \text{ gate with inputs } g_1 \text{ and } g_2 \end{cases}$$

2. If g is a multiplication gate, with inputs g_1 and g_2, then the parties do the following:
   (a) Locally compute $\mathbf{a}'_g = \mathbf{z}_{g_1} - \mathbf{a}_g$ and $\mathbf{b}'_g = \mathbf{z}_{g_2} - \mathbf{b}_g$ and broadcast these values.
   (b) Receive the corresponding values from other parties.
   (c) Locally compute $z_{g_1} - a_g$ and $z_{g_2} - b_g$ by adding all the values received.
   (d) Locally compute $\mathbf{z}_g = (z_{g_1} - a_g)(z_{g_2} - b_g) \cdot \mathbf{1} + (z_{g_1} - a_g) \cdot \mathbf{b}_g + \mathbf{a}_g \cdot (z_{g_2} - b_g) + (\mathbf{ab})_g$ where $\mathbf{1}$ a fixed secret sharing of the value 1.

Figure 2: GMW Protocol with Shared Product Preprocessing

We consider two settings, the first being with *private outputs* where the adversary does not see the output of the non-corrupted parties and the second with *public outputs* where the parties broadcast their output shares at the end to reconstruct the final output and the adversary sees them.

In both models, we show that the adversary's view (i.e., the views of the corrupted parties and the leakage on all the uncorrupted parties' views) can be simulated by a simulator which gets nothing (in the private-outputs setting) or gets all the shares of the output (in the public-outputs setting).

To prove the result, we need two ingredients: (a) the leakage resilience of additive secret sharing over $\mathbb{F}_p$ and, (b) a lemma formalizing the following intuition: *In the GMW protocol, each party learns a share of a secret sharing of the value at each gate in the circuit and nothing more.* The first ingredient we have shown above, and we now describe the second. In Lemmas 5.8 and 5.9, we formally state and prove this intuition in both the private-outputs and public-outputs setting and here we provide an informal statement.

Lemma 2.2 (Informal)**.** *On an input $\vec{x}$, let z_g denote the value at multiplication gate $g \in G_\times$. The joint view of any subset Θ of the parties, $\mathsf{view}^{(\Theta)}$, can be simulated given their shares of the inputs and of the values at each multiplication gate:*

$$\mathsf{view}^{(\Theta)}(x) \equiv \mathsf{Sim}(\vec{x}^{(\Theta)}, (z_g^{(\Theta)})_{g \in G_\times}) \ .$$

Given the lemma, proving local leakage resilience in the private-outputs setting is a hybrid argument. Because of the lemma, the adversary can leak from party j a function of $\vec{x}^{(j)}$ and $(z_g^{(j)})_{g \in G_\times}$. The simulator $\mathsf{LeakSim}$, not knowing the input $\vec{x}$, picks random values $\vec{x}', (z_g{}')_{g \in G_\times}$ instead, secret shares them and then leaks from these values according to the leakage functions $\tau^{(j)}$ specified by $\mathcal{A}$.

Then we show that these two distributions are close to each other. If the local leakage can distinguish between the two distributions, then we can use them to construct leakage functions that violate the local leakage resilience of a single instance of the underlying secret sharing scheme. Because of the homomorphic properties of the secret sharing schemes, this transformation is lossless and does not degrade with circuit size as a hybrid argument would.

The proof in the public-outputs setting has a subtlety that the adversary sees not only the local leakage from the uncorrupted parties, but also their final outputs. In this case, we first observe that the final output is a fixed linear function of the circuit values z_g of the multiplication gates and of the input values x_i. Using this observation, the simulator picks the shares of the multiplication gates conditioned on the output values seen. And we can show a similar reduction to the local leakage resilience of the underlying secret sharing scheme. This proves Theorem 1.4.

2.3 On Local Share Conversion

In this section, we sketch the techniques used to show Theorem 1.5: that three-party additive secret sharing over $\mathbb{F}_p$, for any prime $p > 2$, cannot be converted to additive secret sharing over $\mathbb{F}_2$, even with a small error, for any non-trivial relation R on the secrets.

Our results on impossibility of local share conversion are derived by viewing the output of the share conversion schemes as leakage on the original shares, where the adversary instead of being able to do arbitrary computation, can only add the leakage outputs over $\mathbb{F}_2$.

Impossibility of Share Conversion of Additive Secret Sharing from $\mathbb{F}_p$ to $\mathbb{F}_2$. We start with the impossibility of local share conversion of additive secret sharings from $\mathbb{F}_p$ to $\mathbb{F}_2$ for any non-trivial relation R on the secrets.[7] The analysis is inspired by Fourier analysis reinterpretations of linearity testing [BLR93] and group homomorphism testing [BCLR08].

Assume that $g_1, g_2, g_3 : \mathbb{F}_p \to \mathbb{F}_2$ form a 3-party local share conversion scheme for additive secret sharing for some relation R where shares of 0 in $\mathbb{F}_p$ have to be mapped to shares of 0 in $\mathbb{F}_2$ and shares of 1 in $\mathbb{F}_p$ have to be mapped to shares of 1 in $\mathbb{F}_2$ (with high probability, say 99%).[8] That is, if $x_1 + x_2 + x_3 = b$, then $g_1(x_1) + g_2(x_2) + g_3(x_3) = b$ for $b \in \{0, 1\}$. It is convenient for us to define the real-valued analogues $G_i(x) = (-1)^{g_i(x)}$. At the heart of this proof is the following operator:

$$\Lambda(G_1, G_2, G_3) = \mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathsf{AddSh}(0)} [G_1(x_1) \cdot G_2(x_2) \cdot G_3(x_3)] \ .$$

[7] A relation is trivial if no matter what secret is shared, a constant output by the conversion scheme would satisfy correctness. Or put another way, in a non-trivial relation R, there exist s_0 and s_1 such that s_0 has to be mapped to 0 and s_1 has to be mapped to 1 by the relation R.

[8] We consider more general case in Section 6 which also tolerates a higher error probability of 1/6.

The first observation is that if shares of 0 over $\mathbb{F}_p$ are mapped to shares of 0 over $\mathbb{F}_2$ with high probability (say 99%), then the value of this operator is quite high as,

$$\Lambda(G_1, G_2, G_3) = 1 - 2 \cdot \Pr_{\mathbf{x} \leftarrow \mathsf{AddSh}(0)} \left[g_1(x_1) + g_2(x_2) + g_3(x_3) \neq 0 \right] \geq 0.98 \ .$$

The crux of the argument is an 'inverse theorem' style lemma (Lemma 6.9) which characterizes functions G_1's that result in a large value for Λ. Lemma 6.9 shows that if $\Lambda(G_1, G_2, G_3)$ is high, then each of the functions G_1, G_2 and G_3 are 'almost' constant functions, i.e., for most x's, $G_i(x)$ is the same fixed value. Given this lemma, the impossibility result follows. Because the functions G_i's (and hence g_i's) are almost always constant, even given secret shares of 1 as input, they would still output shares of 0 as output.

To complete the proof, we need to argue that G_1 is an almost constant function. This proof has two parts: the first part which is generic to any field $\mathbb{F}$ is to show that if Λ is large, then G_1 has a large Fourier coefficient. In the second part, we show that if G_1 has a large Fourier coefficient, then G_1 is an almost constant function. This part is specific to $\mathbb{F}_p$.

To show the first part, we rewrite $\Lambda(G_1, G_2, G_3)$ over the Fourier basis (using Lemma 4.16) to get

$$\Lambda(G_1, G_2, G_3) = \sum_{a \in \mathbb{F}_p} \widehat{G}_1(a) \cdot \widehat{G}_2(a) \cdot \widehat{G}_3(a)$$

this follows from Lemma 4.16 as the dual code of additive shares of 0 is the code generated by the all-ones vector. We can now use Cauchy-Schwarz inequality with the fact that $\sum_a |\widehat{G}_i(a)|^2 = 1$ to get that,

$$\leq \left| \widehat{G}_1 \right|_\infty \cdot (\sum_a |\widehat{G}_2(a)|^2) \cdot (\sum_a |\widehat{G}_3(a)|^2) \leq \left| \widehat{G}_1 \right|_\infty \ .$$

This implies that $|\widehat{G}_1|_\infty$ is large. Now we show the second part, which is specific to $\mathbb{F}_p$. We need to show that G_1 is almost constant function. We want to show that if some Fourier coefficient of G_1 is large (larger than $\frac{2}{3}$), then it has to be the zero-th coefficient. The zero-th coefficient measures the bias of G_1: if the coefficient is small, then G_1 is close to balanced, and if this coefficient is large, then G_1 is an almost constant function. Although proving this for all primes is somewhat tedious (see Lemma 6.7), the intuition is easy to grasp. Let $p = 3$ and $\omega = \exp(2\pi i/3)$ be a root of unity. A non-zero Fourier coefficient of G_1 takes the following form: $\widehat{G}_1(a) = \mathbb{E}_{x \in \mathbb{F}_3}[G_1(x) \cdot \omega^{ax}]$ for $a \neq 0$. Because G_1 takes values in $\{-1, 1\}$ and ω^{ax} takes all values $\left\{1, \omega, \omega^2\right\}$, these two functions cannot be too correlated. And hence the Fourier coefficient cannot be too large: $|\widehat{G}_1(a)| \leq 2/3$. This completes the proof.

The Impossibility of Share Conversion from Shamir's Secret Sharing from $\mathbb{F}_p$ to Additive Sharing on $\mathbb{F}_2$. We now briefly discuss the techniques used to prove the result on local conversion of (n, t)-Shamir's secret sharing over $\mathbb{F}_p$, for $(n + 3)/2 \leq t \leq n$. Again consider a relation R where Shamir's shares of 0 over $\mathbb{F}_p$ have to be mapped to additive shares of 0 over $\mathbb{F}_2$ and Shamir's shares of 1 have to be mapped to additive shares of 1 over $\mathbb{F}_2$. Let $g_1, g_2, \ldots, g_n$ be the local share conversion functions used. We want to follow a similar strategy: first show that the corresponding function $G_i = (-1)^{g_i}$ has a large Fourier coefficient. Then, similar to the additive secret sharing proof, show that if G_i has a large Fourier coefficient, then G_i is 'almost constant' and hence derive a contradiction.

In the first part, we want to use the fact that Shamir's shares of 0 over $\mathbb{F}_p$ are converted to additive shares of 0 over $\mathbb{F}_2$ to infer that G_1 (say) has a large Fourier coefficient. This is proved in Lemma 6.10. The proof is a specialized case of the work of Green and Tao [GT10]. In the proof, the value of an appropriately defined operator Λ:

$$\Lambda(G_1, G_2, \ldots, G_n) = \mathop{\mathbb{E}}_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(s)} [G_1(s_1) \cdot G_2(s_2) \cdots G_n(s_n)] \ ,$$

(where $\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(s)$ is a random (n, t)-Shamir's secret sharing of s) is bound by the "Gowers' Uniformity Norm" (the U^2 norm) of the function G_1. Then using a connection between the U^2 norm and Fourier bias, we can derive that G_1 has a large Fourier coefficient. For details see Section 6.

2.4 Additive Combinatorics Context

We provide some context for these techniques. Such Λ style operators have been studied quite a bit in number theory. They can be used to represent many fascinating questions about the distribution of prime numbers. To give some examples, *What is the density of three-term arithmetic progressions in primes?* is a question about the operator $\Lambda = \mathbb{E}_{x,d}[1_P(x) \cdot 1_P(x+d) \cdot 1_P(x+2d)]$ where 1_P is 1 if x is a prime and 0 otherwise. Also, the *twin primes conjecture* can be framed in terms of the operator $\Lambda = \mathbb{E}_x[1_P(x) \cdot 1_P(x+2)]$. Green and Tao [GT10] and subsequent works by Wolf and Gowers [GW10, GW11a, GW11b] tried to understand the following question: let $L_1, L_2, \ldots, L_m$ be linear equations from $\mathbb{F}^n$ to $\mathbb{F}$. Can we bound the following expectation:

$$\Lambda(f_1, f_2, \ldots, f_m) = \mathop{\mathbb{E}}_{\vec{x} \leftarrow \mathbb{F}^n} \left[f_1(L_1(\vec{x})) \cdot f_2(L_2(\vec{x})) \cdots f_m(L_m(\vec{x})) \right] \ ?$$

This is a very general question. And roughly speaking, they give the following answer. These works define two measures of complexity (termed as Cauchy-Schwarz Complexity and True Complexity respectively) and show that if a system of linear equations has complexity k, then,[9]

$$\Lambda(f_1, f_2, \ldots, f_m) < C \cdot \min_i \left\| f_i \right\|_{U^k} \ ,$$

where $\left\| f_i \right\|_{U^k}$ is the k-th order Gowers' Uniformity Norm [Gow01]. This method of bounding Λ by the Gowers' norm has been very influential in number theory. This method is what we use to prove the results on Shamir's secret sharing. We first bound an appropriately defined operator Λ by the Gowers' U^2 norm and then exploit a connection between the U^2 and Fourier analysis. Such a technique does not suffice to give desired results in the case of leakage resilience of $(n, t = \alpha n)$-Shamir's secret sharing for two reasons (for some constant $\alpha > 0$). The first reason is that the constant C derived from this method is often extremely large and has an exponential dependence on the number of equations m. Also the second reason is that in our setting, the functions f_i's are chosen by the adversary. So, showing that $\left\| f_i \right\|_{U^k}$ is small is either very challenging or just not true for some adversarially chosen functions f_i's. On the other hand, we do not know how to translate this into an local leakage attack on Shamir's secret sharing either and hence a strong win-win result eludes us.

[9] Both complexity measures do not assign complexity to all possible linear forms. To give an example, the linear form $(L_1(x) = x, L_2(x) = x + 2)$, which corresponds to the twin primes conjecture, is not assigned a complexity value and the twin primes conjecture is still open.

3 Preliminaries

We denote by $\mathbb{C}$ the field of complex numbers, by SD the statistical distance (or total variation distance), and by $\equiv$ the equality of distributions. For a vector space $\mathbb{F}_p^n$, we define $U = U_n$ to be the uniform distribution over $\mathbb{F}_p^n$. For any finite set S, $x \leftarrow S$ denotes the fact of sampling an independent element x uniformly from S. For any positive integer n, the set $[n]$ is the integer interval $\{1, \dots, n\}$. As we are using extensively codes, we use the conventions of coding theory: vectors are always row vectors.

3.1 Linear Codes

Secret sharing schemes are closely related to linear codes, that we define next.

Definition 3.1 (Linear Code). *A subset $C \subseteq \mathbb{F}^n$ is an $[n, k, d]$* linear code *over field $\mathbb{F}$ if C is a subspace of $\mathbb{F}^n$ of dimension k such that: for all $\vec{x} \in C \setminus \{\vec{0}\}$,* $\mathsf{HammingDistance}(\vec{x}) \geq d$ *(i.e., the minimum Hamming distance between two elements of the code is at least d). A code is called* Maximum Distance Separable (MDS) *if $n - k + 1 = d$. The* dual code *of the code C is defined as $C^\perp = \left\{\vec{y} \in \mathbb{F}^n \ : \ \forall \vec{x} \in C, \left\langle \vec{x}, \vec{y} \right\rangle = 0\right\}$. A* generator matrix *for an $[n, k, d]$ linear code is a matrix $G \in \mathbb{F}_p^{k \times n}$ such that its rows form a basis C, or in other words: $C = \left\{\vec{y} \in \mathbb{F}_p^n \ : \ \exists \vec{x} \in \mathbb{F}_p^k, \ \vec{y} = \vec{x} \cdot G\right\}$. A* parity check matrix *H of C is a generator matrix of the dual code $C^\perp$.*

Proposition 3.2. *The dual code $C^\perp$ of an $[n, k, d]$ MDS code C is itself an MDS code with parameters $[n, n - k, k + 1]$.*

Example 3.3 (Generalized Reed Solomon Code). An $[n, k, n - k + 1]$ generalized Reed Solomon code over $\mathbb{F}$ such that $|\mathbb{F}| > n$ interprets a message $\vec{m} \in \mathbb{F}^k$ as $p(x) = m_1 + m_2 x + \cdots + m_k x^{k-1}$ and encodes it as $(u_1 p(\alpha_1), u_2 p(\alpha_2), \dots, u_n p(\alpha_n))$ where $A = \{\alpha_1, \alpha_2 \dots, \alpha_n\} \subseteq \mathbb{F}$ is a fixed set of n distinct evaluation points and $u_1, \dots, u_n \in \mathbb{F}_p$ are non-zero coefficients. Generalized Reed Solomon codes are MDS.

Moreover, the dual code of such a code C is itself a $[n, n - k, k + 1]$ generalized Reed Solomon code C' over $\mathbb{F}$ with the same evaluation points and the coefficients $v_i = u_i^{-1} \cdot \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1}$ for $i \in [n - k]$. Indeed given messages $p(x) = m_1 + m_2 x + \cdots + m_k x^{k-1}$ and $q(x) = m'_1 + m'_2 x + \cdots + m'_{n-k} x^{n-k-1}$, the inner product of the corresponding codewords for C and C' is:

$$\sum_{i=1}^{n} u_i v_i p(\alpha_1) q(\alpha_1) = \sum_{i=1}^{n} \frac{1}{\prod_{j \neq i} (\alpha_i - \alpha_j)} p(\alpha_1) q(\alpha_1) \ ,$$

which is the Lagrange interpolation of the coefficient x^{n-1} of $p(x)q(x)$, namely 0. This proves that C' is the dual code of C.

3.2 Linear Secret Sharing Schemes

We recall the definition of (threshold) secret sharing schemes.

Definition 3.4 (Secret Sharing Scheme). *An (n, t)-*secret sharing scheme *over field $\mathbb{F}$ is defined by a pair* $(\mathsf{Share}, \mathsf{Rec})$ *where* Share *is a randomized mapping of an input $s \in \mathbb{F}$ to shares for each party $\mathbf{s} = (s^{(1)}, s^{(2)}, \dots, s^{(n)})$ and the reconstruction algorithm* Rec *is a function mapping a set $A \subseteq [n]$ and the corresponding shares $\mathbf{s}^{(A)} = \left(s^{(j)}\right)_{j \in A}$ to a secret $s \in \mathbb{F}$, such that the following properties hold:*

1. Reconstruction. *$\mathsf{Rec}(A, \mathbf{s}^{(A)})$ outputs the secret s for all sets $A \subseteq [n]$ where $|A| \geq t$.*
2. Security. *For any set A such that $|A| < t$, the joint distribution of shares received by the subset of parties A, $\mathbf{s}^{(A)} = \left(s^{(j)}\right)_{j \in A}$ where $\mathbf{s} \leftarrow \mathsf{Share}(s)$, is independent of the secret s.*

We extend secret sharing schemes to handle vectors of secrets naturally as follows. If $(\mathsf{Share}, \mathsf{Rec})$ is a secret sharing scheme and if $\vec{s} \in \mathbb{F}^k$ is a vector of k secrets, we define:

$$\vec{\mathbf{s}} = (\vec{s}^{(1)}, \dots, \vec{s}^{(n)}) \leftarrow \mathsf{Share}(\vec{s}) \qquad \text{where } \forall i \in [k],\ (s_i^{(1)}, \dots, s_i^{(n)}) \leftarrow \mathsf{Share}(s_i)$$
$$\vec{s} = (s_1, \dots, s_k) = \mathsf{Rec}(A, \vec{\mathbf{s}}^{(A)}) \qquad \text{where } \forall i \in [k],\ s_i = \mathsf{Rec}(A, \mathbf{s}_i^{(A)})$$

where $\vec{s}^{(j)} = (s_1^{(j)}, \dots, s_n^{(j)})$ and $\mathbf{s}_i^{(A)} = \left(s_i^{(j)}\right)_{j \in A}$.

An important particular case of secret sharing scheme are linear secret sharing schemes. Actually all the schemes we consider in this paper are linear.

Definition 3.5. *An (n, t)-secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ over a finite field $\mathbb{F}$ is* linear *if*

1. *the codomain of Share is the vector space $(\mathbb{F}^\ell)^n$, for some positive integer ℓ (i.e., each share is a vector of ℓ field elements),*
2. *for any $s \in \mathbb{F}$, $\mathsf{Share}(s)$ is uniformly distributed over an affine subspace of $(\mathbb{F}^\ell)^n$,*
3. *for any $\lambda_0, \lambda_1, s_0, s_1 \in \mathbb{F}$:*

$$\left\{ \lambda_0 \mathbf{s}_0 + \lambda_1 \mathbf{s}_1 \ : \ \begin{matrix} \mathbf{s}_0 \leftarrow \mathsf{Share}(s_0) \\ \mathbf{s}_1 \leftarrow \mathsf{Share}(s_1) \end{matrix} \right\} \equiv \mathsf{Share}(\lambda_0 s_0 + \lambda_1 s_1) \ .$$

Let us now recall the two classical linear secret sharing schemes we are using.

Example 3.6 (Additive Secret Sharing $(\mathsf{AddSh}_n, \mathsf{AddRec}_n)$). The additive secret sharing scheme $(\mathsf{AddSh}_n, \mathsf{AddRec}_n)$ for n parties over a field $\mathbb{F}$ is a linear (n, n)-secret sharing scheme defined as follows. Shares $\mathsf{AddSh}_n(s) = \mathbf{s}$ of a secret $s \in \mathbb{F}$ are generated as follows: $(s^{(1)}, \dots, s^{(n-1)}) \leftarrow \mathbb{F}^{n-1}$, and $s^{(n)} = s - (s^{(1)} + \dots + s^{(n-1)})$. The reconstruction of s from $\mathbf{s}$ is done as follows: $\mathsf{AddRec}_n(\mathbf{s}) = s^{(1)} + \dots + s^{(n)}$.

Example 3.7 (Shamir's Secret Sharing $(\mathsf{ShaSh}_{n,t}, \mathsf{ShaRec}_{n,t})$). The Shamir's secret sharing scheme $(\mathsf{ShaSh}_{n,t}, \mathsf{ShaRec}_{n,t})$ of n parties and threshold t over a field $\mathbb{F}$ (with $|\mathbb{F}| > n$) is a linear (n, t)-secret sharing scheme defined as follows. Let $\alpha_1, \dots, \alpha_n \in \mathbb{F} \setminus \{0\}$ be n distinct arbitrary non-zero field elements. Shares $\mathsf{ShaSh}_{n,t}(s) = \mathbf{s}$ of a secret $s \in \mathbb{F}$ are generated as follows: generate a uniformly random polynomial P of degree at most $t - 1$ over $\mathbb{F}$ with constant coefficient s (i.e., $P(0) = s$), the share $s^{(j)}$ is $s^{(j)} = P(\alpha_j)$. Given shares $s^{(A)}$ with $A \subseteq [n]$ and $|A| \geq t$, the reconstruction works as follows: it computes the Lagrange coefficients $\lambda_j = \prod_{i \in A \setminus \{j\}} (\alpha_i / (\alpha_i - \alpha_j))$ and output $\mathsf{ShaRec}_{n,t}(A, s^{(A)}) = \sum_{j \in A} \lambda_j s^{(j)} \in \mathbb{F}$.

3.3 Fourier Analysis

In this section, we present the notion of Fourier coefficients of a function and some of its properties. Most of the calculations needed about Fourier coefficients are deferred to the corresponding sections for the ease of readability. For an excellent survey on how Fourier Analytic methods are used in Additive Combinatorics, see [Gre07].

Let $\mathbb{G}$ be any finite Abelian group. A character is a homomorphism $\chi : \mathbb{G} \to \mathbb{C}$ from the group $\mathbb{G}$ to $\mathbb{C}$, i.e., $\chi(a+b) = \chi(a) \cdot \chi(b)$ for all $a, b \in \mathbb{G}$. For any finite Abelian group $\mathbb{G}$, the set of characters

$\widehat{\mathbb{G}}$ is a group (under the operation point-wise product) isomorphic to $\mathbb{G}$. The reader should note that while we define Fourier coefficients in generality, we would be primarily use Fourier analysis on the groups $\mathbb{F}_p$ for some prime p.

Definition 3.8 (Fourier Coefficients)**.** *For functions $f : \mathbb{G} \to \mathbb{C}$, the* Fourier basis *is composed of the group $\widehat{\mathbb{G}}$ of characters $\chi : \mathbb{G} \to \mathbb{C}$. We define the* Fourier coefficient *$\widehat{f}(\chi)$ corresponding to a character χ as*

$$\widehat{f}(\chi) = \mathop{\mathbb{E}}_{x \leftarrow \mathbb{G}} \big[f(x) \cdot \chi(x) \big] \in \mathbb{C} \ .$$

As we will use Fourier analysis on the additive group $\mathbb{F}_p$, we describe the Fourier characters over $\mathbb{F}_p$. Let $\omega = \exp(2\pi i/p)$ be a primitive p-th root of unity. Then, the characters for $\mathbb{F}_p$ are given by $\chi_\alpha(x) = \omega^{\alpha \cdot x}$ where $\alpha \in \mathbb{F}_p$. We sometimes abuse notation and write $\widehat{f}(\alpha)$ instead of $\hat{f}(\chi_\alpha)$.

We follow the "standard" notation in additive combinatorics. In this notation, when working on the group $\mathbb{G}$, the *Haar measure* is used which assigns the weight $|\mathbb{G}|^{-1}$ to every $x \in \mathbb{G}$ and when working on $\widehat{\mathbb{G}}$, the *counting measure* is used which assigns the weight 1 to every $\alpha \in \widehat{\mathbb{G}}$. Using these measures generally eliminates the need for normalization. So, when we talk about norms, these will always be taken with respect to the underlying measure. That is,

$$\|f\|_1 = \mathop{\mathbb{E}}_{x} \big[|f(x)| \big] \quad \text{whereas} \quad \|\widehat{f}\|_2 = \Big(\sum_{\alpha} |\widehat{f}(\alpha)|^2 \Big)^{1/2} \ .$$

We note that the Fourier Transform has the following properties. These follow easily from the orthogonality relation on the characters: $\sum_{x \in \mathbb{F}_p} \omega^{a \cdot x}$ is p when $a = 0$ and 0 otherwise.

Theorem 3.9. *Let $f, g : \mathbb{G} \to \mathbb{C}$ be two functions. Let $\widehat{\mathbb{G}}$ denote the group of characters of $\mathbb{G}$. The following hold:*

(a) (Parseval's identity) *We have,*

$$\mathop{\mathbb{E}}_{x \leftarrow \mathbb{G}} \Big[f(x) \cdot \overline{g(x)} \Big] = \sum_{\chi \in \widehat{\mathbb{G}}} \widehat{f}(\chi) \cdot \overline{\widehat{g}(\chi)} \ .$$

In particular, $\|f\|_2 = \|\widehat{f}\|_2$ where $\|f\|_2^2 = \mathbb{E}_{x \leftarrow \mathbb{G}} \Big[\big|f(x)\big|^2 \Big]$ and $\|\widehat{f}\|_2^2 = \sum_{\chi \in \widehat{\mathbb{G}}} \Big| \widehat{f}(\chi) \Big|^2$.

(b) (Fourier Inversion Formula) *For any $x \in \mathbb{G}$, $f(x) = \sum_{\chi \in \widehat{\mathbb{G}}} \widehat{f}(\chi) \cdot \overline{\chi(x)}$.*

Finally, we introduce the notion of bias. A function is biased if it is highly correlated with some Fourier character.

Definition 3.10 (Bias)**.** *For a function $f : \mathbb{G} \to \mathbb{C}$, the* bias *of f is defined as,*

$$\text{bias}(f) = \|\widehat{f}\,\|_\infty = \max_{\chi \in \widehat{\mathbb{G}}} |\widehat{f}(\chi)| \ .$$

We need a calculation on certain sums of roots of unity. Let A be a subset of $\mathbb{Z}_k$. And let $\gamma = e^{i \cdot 2\pi/k}$. We want to bound sums of the form $\boldsymbol{\gamma}^A = \sum_{x \in A} \gamma^x$. We state and prove the Lemma below. We will use the lemma to show that non-trivial Fourier coefficients of certain functions have to be smaller than the trivial one.

Lemma 3.11. *Let k be a positive integer. Let $\zeta_k : [0,k] \to \mathbb{R}_{\geq 0}$ be defined as $\zeta_k(x) = \frac{\sin(x\pi/k)}{\sin(\pi/k)}$ with $\zeta_k(0) = 0$. Let $A \subseteq \mathbb{Z}_k$ of size t. Let $A^\star = \{0, 1, \ldots, t-1\}$. Then*

$$|\boldsymbol{\gamma}^A| \leq |\boldsymbol{\gamma}^{A^\star}| = \frac{\sin(\pi t/k)}{\sin(\pi/k)} = \zeta_k(t) \ .$$

We will show that the sum is maximized when A is an interval. The proof of the claim is an extremal argument. If an element does not lie in the direction of the sum, we can remove it and add something in the direction to increase the norm.

Proof. First, the fact that $\boldsymbol{\gamma}^{A^\star} = \zeta_k(t)$ is derived using a basic trigonometry calculation:

$$\left|\boldsymbol{\gamma}^{A^\star}\right| = \left|\sum_{i=0}^{t-1} \gamma^i\right| = \frac{\left|\gamma^t - 1\right|}{\left|\gamma - 1\right|} = \frac{2\sin(\pi t/k)}{2\sin(\pi/k)} \ ,$$

where the last equality follows from the fact that the angle between γ^t and -1 is $(\pi - 2t\pi/k)$ and hence, $\left|\gamma^t - 1\right| = 2\cos((\pi - 2t\pi/k)/2) = 2\sin(\pi t/k)$.

Let us now show that the sum is indeed maximum when $A = A^\star$. An interval $[a, b)$ over $\mathbb{Z}_p$ consists of the elements $\left\{a \bmod q, a+1 \bmod q, \ldots, b-1 \bmod q\right\}$. Note that the intervals $[a, b)$ and $[b, a)$ are distinct. Observe that for any two intervals A, B of the same size, $\boldsymbol{\gamma}^A = \gamma^k \cdot \boldsymbol{\gamma}^B$ for some $k \in \mathbb{Z}_p$, hence $\boldsymbol{\gamma}^A$ and $\boldsymbol{\gamma}^B$ have the same magnitude.

Let $A \subseteq \mathbb{Z}_p$ of size t, such that $\boldsymbol{\gamma}^A$ is maximum. We want to prove that $\boldsymbol{\gamma}^A = \boldsymbol{\gamma}^{A^\star}$. The cases when $t = 0$ or p are vacuously true. If A is an interval, i.e., a set of the form above, we are done. Else, we want to show that there exists an interval A' of same size, such that $|\boldsymbol{\gamma}^A| \leq |\boldsymbol{\gamma}^{A'}|$.

Let $\xi = \boldsymbol{\gamma}^A = \sum_{a \in A} \omega^a$. We have $|\xi| \geq \left|\boldsymbol{\gamma}^{A^\star}\right| > 0$. We consider the interval $A' = [a', a'+t)$ consisting of all the roots of unity most 'aligned' with ξ. That is, a' is chosen as:

$$a' \in \operatorname*{argmax}_{a' \in \mathbb{Z}_p, A' = [a', a'+t)} \boldsymbol{\gamma}^{A'} \circ \xi \ ,$$

where $\circ$ is the complex dot product.[10] Equivalently, the interval $A' = [a, a+t)$ is the interval of size t such that, for all $a \in A'$ and $b \in \{0, 1 \ldots, k-1\} \setminus A'$, $\gamma^a \circ \xi \geq \gamma^b \circ \xi$.

Let us now show that $|\boldsymbol{\gamma}^A| \leq |\boldsymbol{\gamma}^{A'}|$. For that, let $B \subseteq \mathbb{Z}_k$ be a set of size t such that $|\xi| = |\boldsymbol{\gamma}^A| = |\boldsymbol{\gamma}^B|$ and the size of the intersection of A' and B is maximum. Let us prove that $B = A'$, which will conclude the proof.

Pick $a \in A' \setminus B$ and $b \in B \setminus A'$. Consider the set $B' = (B \setminus \{b\}) \cup \{a\}$. We remark that the intersection of A' and B' is larger than the intersection of A' and B. Let us now prove that $|\boldsymbol{\gamma}^{B'}| \geq |\boldsymbol{\gamma}^B|$, which is a contradiction (B was not the set of size t with the largest sum and the largest interesection with A'). Observe that $\boldsymbol{\gamma}^{B'} = \xi - \gamma^b + \gamma^a$. And as $\xi \circ \gamma^a \geq \xi \circ \gamma^b$, $\xi \circ (\gamma^a - \gamma^b) \geq 0$. Hence, $\cos\theta \geq 0$ where θ is the angle between ξ and $(\gamma^a - \gamma^b)$. This implies that $\theta \in [-\pi/2, \pi/2]$ and hence $|\xi - \gamma^b + \gamma^a| = |\xi + (\gamma^a - \gamma^b)| \geq |\xi|$.

And the result follows. □

[10] $z_1 \circ z_2 = x_1 x_2 + y_1 y_2$ where $z_b = x_b + i \cdot y_b$ is the dot product of z_1 and z_2. Equivalently, $z_1 \circ z_2 = |z_1||z_2|\cos\theta$ where θ is the angle between z_1 and z_2.

4 On Leakage Resilience of Secret Sharing Schemes

4.1 Definitions and Basic Properties

We consider a model of leakage where the adversary can first choose a subset of $\Theta \subseteq [n]$ parties and get their full shares and then leak m bits each from all the shares of all the (other) parties. Formally, what is learned by the adversary on a sharing $\mathbf{s}$ is the following:

$$\mathsf{Leak}_{\Theta,\boldsymbol{\tau}} = (\mathbf{s}^{(\Theta)},\ (\tau^{(i)}(\mathbf{s}^{(\Theta)}, s^{(i)}))_{i\in[n]})\ , \tag{4}$$

where $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ is a family of n leakage functions that output m bits and $\mathbf{s}^{(\Theta)} = \left(s^{(j)}\right)_{j\in\Theta}$ are the complete shares of the parties corrupted. The adversary can choose the functions $\boldsymbol{\tau}$ arbitrarily.

Definition 4.1 (Local Leakage Resilient)**.** *Let Θ be a subset of $[n]$. A secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *is said to be* (Θ, m, ε)-local leakage resilient *(or* (Θ, m, ε)-LL resilient *for short) if for every leakage function family $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ where $\tau^{(j)}$ has an m-bit output, and for every pair of secrets s_0, s_1,*

$$\mathrm{SD}\Big(\big\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\mathbf{s})\ :\ \mathbf{s} \leftarrow \mathsf{Share}(s_0)\big\}, \big\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\mathbf{s})\ :\ \mathbf{s} \leftarrow \mathsf{Share}(s_1)\big\}\Big) \leq \varepsilon\ .$$

A secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ *is said to be (θ, m, ε)-LL resilient if it is (Θ, m, ε)-LL resilient for any subset $\Theta \subseteq [n]$ of size at most θ.*

Remark 4.2. We remark that we can consider an equivalent definition where for each distribution $\mathcal{D}$ of leakage function family $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$:

$$\mathrm{SD}\left(\left\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\mathbf{s})\ :\ \begin{matrix}\mathbf{s} \leftarrow \mathsf{Share}(s_0)\\ \boldsymbol{\tau} \leftarrow \mathcal{D}\end{matrix}\right\}, \left\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\mathbf{s})\ :\ \begin{matrix}\mathbf{s} \leftarrow \mathsf{Share}(s_1)\\ \boldsymbol{\tau} \leftarrow \mathcal{D}\end{matrix}\right\}\right) \leq \varepsilon\ .$$

Observe that a (n, t)-secret sharing scheme is $(t, 0, 0)$-Local Leakage resilient: that is, complete access to the shares of t parties and no information about the others.

Note that in the leakage model, the adversary is not allowed to adaptively choose the leakage functions. As discussed in the introduction, this is a very meaningful and well-motivated leakage model. Next, we demonstrate some attacks in this model. In particular, we formalize the observation that linear secret sharing schemes over small characteristic fields are not local leakage resilient.

Example 4.3 (Attack on Schemes Over Small Characteristic Fields). Over fields of small characteristic like $\mathbb{F}_{2^k}$ that have many additive subgroups, secret sharing schemes with linear reconstruction are not local leakage resilient even for 1-bit leakage. We give some examples of such attacks. They are not hard to generalize. Let $s \in \mathbb{F}_{2^k}$ be the secret that is shared among n-parties as shares $(s^{(1)}, s^{(2)}, \ldots, s^{(n)})$. Consider the following attacks:

- *Additive Secret Sharing.* The adversary can locally leak the least significant bit of each share $s^{(j)}$. Adding them up, the adversary can reconstruct the least significant bit of s.
- *Shamir's Secret Sharing.* For a similar attack, observe that $s = \lambda_1 s^{(1)} + \lambda_2 s^{(2)} + \cdots + \lambda_n s^{(n)}$ where λ_j's are *fixed* Lagrange coefficients. So to attack the scheme, the adversary locally multiplies the share $s^{(j)}$ with λ_j and leaks the least significant bit. This again reveals the least significant bit of s. The recent work of Guruswami and Wootters [GW17] shows how such leakage can be used to even *completely reconstruct* s, in some settings.

Example 4.4 (Attack on Few Parties). If the number of parties n is a constant, then the additive secret sharing over $\mathbb{F}_p$ is not LL-resilient. The adversary can distinguish between secrets $< p/2$ and $> p/2$ by local leakage. The adversary locally leaks $\tau^{(j)}(s^{(j)}) = 1$ if the share $s^{(j)} < p/(2n)$ (seeing the share as integer in $\{0, \dots, p-1\}$). If all the leakages output 1, the adversary can conclude that the secret $s = s^{(1)} + \cdots + s^{(n)} < p/2$. On the other hand, if the secret is larger than $p/2$, then all the leakage outputs will never be 1 simultaneously. In the $< p/2$ case, the probability of all the secrets being $< p/2n$ is about $(1/2n)^n$, a constant. Similar attacks can also be performed on Shamir's secret sharing. We stress that this is not the most effective attack, but it is an attack nonetheless. This attack is similar to the one in [KP10, Footnote 8].

4.2 Leakage Resilience of Additive and Shamir's Secret Sharing Schemes

We are now in a position to state the main technical result of this section. That, no family of local leakage functions can distinguish between shares picked from a 'good' linear code and uniformly random shares. We then apply these results to get local leakage resilience for additive and Shamir's secret sharing schemes.

4.2.1 Main Technical Theorem: Leakage Resilience of Linear Codes

We describe two versions of our bounds: they differ in their dependence on the underlying prime p. The first bound has tighter constants but a worse dependence on the prime while the latter bound, the bound on the distinguishing advantage does not degrade with increasing primes.

Theorem 4.5. *Let $C \subseteq \mathbb{F}_p^n$ be any $[n, t-1, n-t+2]$ linear code. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} < 1$ (when $2^m < p$). Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \tfrac{1}{2} \cdot p^{n-t+1} \cdot c_m^t \ ,$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$ and:

$$\boldsymbol{\tau}(C) = \left\{ \left(\tau^{(i)}(x_i)\right)_{i \in [n]} \ : \ \vec{x} \leftarrow C \right\} \quad \textit{and} \quad \boldsymbol{\tau}(U_n) = \left\{ \left(\tau^{(i)}(x_i)\right)_{i \in [n]} \ : \ \vec{x} \leftarrow U_n \right\} \ .$$

The bound above is not tight. In particular, the p^{n-t+1} factor leads to an unnatural situation where our bounds become worse as the prime increases. To give some intuition about what parameters it can support, if a bit is leaked from each share, i.e., $m = 1$, then c_m is a constant and hence the statistical distance is bound as $e^{(n-t+1)\cdot \log p} \cdot c_m^t$ then we can set $n - t \approx O(n/\log p)$ and the distance is negligible. But we cannot set $n - t = \Omega(n)$.

Next, we describe stronger bounds removing this dependence in p: the p^{n-t+1} dependence is replaced by a $(2^{O(m)})^{(n-t)}$ style term. This gives the "natural interpretation" in that our bounds get stronger as the prime p increases, since the c_m term decreases as p increases. The key idea behind this proof is using Cauchy-Schwarz inequality to reduce the number of terms we need to bound. The constant here is not very optimized, but it suffices.

Theorem 4.6. *Let $C \subseteq \mathbb{F}_p^n$ be any $[n, t-1, n-t+2]$ linear code. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. Let $c'_m = \frac{2^m \sin(\pi/2^m + \pi/2^{4m})}{p \sin(\pi/p)}$. Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \frac{1}{2} \cdot 2^{(5m+1)\cdot(n-t)+m} \cdot (c'_m)^{2t-n-2} \ ,$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$.

In the case of additive secret sharing, we can improve the constants more, and the proof serves as an instructive warmup for the proof of Theorem 4.6. We state the bound below.

Theorem 4.7. *[Additive Secret Sharing] Let $C \subseteq \mathbb{F}_p^n$ be the code generated by* $\mathsf{AddSh}(0)$. *Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} < 1$ (when $2^m < p$). Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \tfrac{1}{2} \cdot 2^m \cdot c_m^{n-2}$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$.

We remark that a slightly weaker version of Theorem 4.7 above can be obtained by invoking Theorem 4.6 on the $[n, n-1, 2]$ code C generated by $\mathsf{AddSh}(0)$ ($t = n$). More precisely we would get:

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq 2^{m-1} \cdot (c'_m)^{n-2} \ ,$$

which is almost the same bound except c_m is replaced by the slightly larger constant c'_m.

4.2.2 Local Leakage Resilience of Additive and Shamir's Secret Sharing Schemes

Additive Secret Sharing. We observe that Theorems 4.5 to 4.7 yield the following two corollaries for additive secret sharing and Shamir's secret sharing. We first prove the corollaries assuming Theorems 4.5 to 4.7 and then prove these theorems next. We describe example parameter settings in Section 4.2.3.

Let c_m and c'_m be defined as follows: For $2^m < p$,

$$c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} \quad \text{and} \quad c'_m = \frac{2^m \sin(\pi/2^m + \pi/2^{4m})}{p \sin(\pi/p)} \ .$$

Corollary 4.8 (Leakage Resilience of Additive Secret Sharing)**.** *The additive secret sharing* AddSh_n *for n parties is (θ, m, ε)-LL resilient where:*

$$\varepsilon = p \cdot c_m^{n-\theta} \quad \text{or} \quad \varepsilon = 2^m \cdot c_m^{n-\theta-2} \ .$$

Proof. This corollary follows from Theorems 4.5 and 4.7 the following claim after remarking that, when θ parties reveal their share, an additive secret sharing with n parties becomes a random additive secret sharing with $n - \theta$ parties.

Claim 4.8.1. *Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of m-bit output leakage functions. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} < 1$ (when $2^m < p$). Then for all secrets $s_0, s_1 \in \mathbb{F}_p$,*

$$\mathrm{SD}(\boldsymbol{\tau}(\mathsf{AddSh}_n(s_0)),\ \boldsymbol{\tau}(\mathsf{AddSh}_n(s_1))) \leq p \cdot c_m^n \ ,$$

Proof. The proof is a simple hybrid argument. Let C be the support of $\mathsf{AddSh}(0)$. Note that C is an $[n, n-1, 2]$ linear code and $\mathsf{AddSh}(0)$ is uniformly distributed on C. Also note that the distribution $\mathsf{AddSh}(s)$ is a coset of $\mathsf{AddSh}(0)$, i.e., $\mathsf{AddSh}(s)$ can be obtained by first sampling $\mathbf{x} \leftarrow \mathsf{AddSh}(0)$ and then adding a fixed vector $s \cdot \mathbf{e} = (s, 0, 0, \ldots, 0)$ to $\mathbf{x}$. So, for any secret s,

$$\begin{aligned} \mathrm{SD}(\boldsymbol{\tau}(\mathsf{AddSh}(s)),\ \boldsymbol{\tau}(U_n)) &= \mathrm{SD}(\boldsymbol{\tau}(\mathsf{AddSh}(0) + s\mathbf{e}),\ \boldsymbol{\tau}(U_n)) \\ &= \mathrm{SD}(\boldsymbol{\tau}'(\mathsf{AddSh}(0)),\ \boldsymbol{\tau}'(U_n - s\mathbf{e})) \end{aligned}$$

where $\tau'^{(1)}(x) = \tau^{(1)}(x+s)$ and $\tau'^{(j)} = \tau^{(j)}$ for $j > 1$.

$$= \mathrm{SD}(\boldsymbol{\tau}'(\mathsf{AddSh}(0)),\ \boldsymbol{\tau}'(U_n))\ .$$

Using triangle inequality, we can complete the proof:

$$\begin{aligned}
&\mathrm{SD}(\boldsymbol{\tau}(\mathsf{AddSh}(s_0)),\ \boldsymbol{\tau}(\mathsf{AddSh}(s_1))) \\
&\leq \mathrm{SD}(\boldsymbol{\tau}(\mathsf{AddSh}(s_0)),\ \boldsymbol{\tau}(U_n))\ +\ \mathrm{SD}(\boldsymbol{\tau}(U_n),\ \boldsymbol{\tau}(\mathsf{AddSh}(s_1))) \\
&\leq p \cdot c_m^n \quad \text{or} \quad 2^m \cdot c_m^{n-2}\ ,
\end{aligned}$$

where the last line follows from either Theorem 4.5 or Theorem 4.7. □

This concludes the proof of Corollary 4.8. □

Shamir's Secret Sharing. Next we argue the corresponding statement for Shamir's secret sharing.

Corollary 4.9 (Leakage Resilience of Shamir's secret sharing)**.** *The (n,t)-Shamir's secret sharing scheme $\mathsf{ShaSh}_{n,t}$ is (θ, m, ε)-LL resilient where:*

$$\varepsilon = p^{n-t+1} \cdot c_m^{t-\theta} \quad \textit{or} \quad \varepsilon = 2^{(5m+1)(n-t)+m} \cdot (c'_m)^{2t-n-\theta-2}\ .$$

Proof. This corollary follows from the following lemma after remarking that, when θ parties reveal their share, a Shamir's secret sharing $\mathsf{ShaSh}_{n,t}(s)$ on the remaining $n-\theta$ parties is an still an MDS code, up to an additive shift. We prove this claim first and then finish the proof.

Claim 4.9.1. *For Shamir's secret sharing scheme $\mathsf{ShaSh}_{n,t}$ of n parties and threshold t over a field $\mathbb{F}$ (with $|\mathbb{F}| > n$), let $\Theta \subseteq [n]$ be a set of $\theta < t$ parties. Consider the following experiment where for a given secret s, n shares $\mathbf{s} = \mathsf{ShaSh}_{n,t}(s)$ are generated and the shares for parties in Θ leaked. Let the leaked values be $\mathbf{x}^{(\Theta)}$. Let $\mathsf{ShaSh}_{n,t}(s)|_{\mathbf{s}^{(\Theta)}=\mathbf{x}^{(\Theta)}}$ be the distribution on shares conditioned on the revealed values $\mathbf{s}^{(\Theta)}$ being $\mathbf{x}^{(\Theta)}$. Then, there exists an $[n-\theta, t-1-\theta, n-t+2]$ MDS code $C \subseteq \mathbb{F}^{n-\theta}$ and a shift vector $\mathbf{b} \in \mathbb{F}^n$ such that,*

$$\mathsf{ShaSh}_{n,t}(s)|_{\mathbf{s}^{(\Theta)}=\mathbf{x}^{(\Theta)}} \equiv \left\{(\mathbf{y}^{(\overline{\Theta})}|\mathbf{0}^{(\Theta)}) + \mathbf{b}\ :\ \mathbf{y} \leftarrow C\right\}\ ,$$

where $(\mathbf{y}^{(\overline{\Theta})}|\mathbf{0}^{(\Theta)})$ denotes a vector where the positions in Θ are 0 while the positions in $\overline{\Theta}$ are filled by $\mathbf{y}$.

Given the claim, the proof follows. The adversary sees $\mathbf{s}^{(\Theta)}$ for the θ parties corrupted. Then, the adversary specifies leakage functions $\boldsymbol{\tau}^{(\overline{\Theta})} = (\tau^{(i)})_{i\in\overline{\Theta}}$ be any family of m-bit output leakage functions. We bound $\mathrm{SD}\left(\boldsymbol{\tau}^{\overline{\Theta}}(\mathsf{ShaSh}_{n,t}(s)^{(\overline{\Theta})}),\ \boldsymbol{\tau}(U_n)\right)$ and use the triangle inequality to complete the proof and lose a factor of 2 as above. Observe that,

$$\begin{aligned}
\mathrm{SD}\left(\boldsymbol{\tau}^{(\overline{\Theta})}(\mathsf{ShaSh}_{n,t}(s)^{(\overline{\Theta})}),\ \boldsymbol{\tau}(U_n)\right) &= \mathrm{SD}\left(\boldsymbol{\tau}^{(\overline{\Theta})}(C + \mathbf{b}^{(\overline{\Theta})}),\ \boldsymbol{\tau}(U_n)\right) \\
&\leq \frac{1}{2} \cdot p^{n-t+1} \cdot c_m^{t-\theta} \quad \text{or} \quad \frac{1}{2} \cdot 2^{(5m+1)(n-t)+m} \cdot (c'_m)^{2t-n-\theta-2}\ ,
\end{aligned}$$

where the equality follows from Claim 4.9.1 and the inequality follows from Theorems 4.5 and 4.6. Next, we prove the claim to finish the proof.

Proof of Claim 4.9.1. The proof follows from considering alternate ways of sampling the conditional distribution.

$$\mathsf{ShaSh}_{n,t}(s)|_{\mathbf{s}^{(\Theta)}=\mathbf{x}^{(\Theta)}} \equiv \mathsf{ShaSh}_{n,t}(0)|_{\mathbf{s}^{(\Theta)}=\mathbf{x}^{(\Theta)}-s\mathbf{1}^{(\Theta)}} + s\mathbf{1} \ ,$$

where $\mathbf{1} \in \mathbb{F}^n$ is the vector $\mathbf{1} = (1, \ldots, 1)$. This follows because $\mathsf{ShaSh}_{n,t}(s) \equiv \mathsf{ShaSh}_{n,t}(0) + s\mathbf{1}$. For the next transformation, pick a polynomial p of degree at most $\theta - 1 < t$ such that p when evaluated at points in Θ evaluates to $\mathbf{x}^{(\Theta)}$. Let $\mathbf{p}$ denote the evaluation of p at the evaluation points of Shamir's secret sharing. Then,

$$\equiv \mathsf{ShaSh}_{n,t}(0)|_{\mathbf{s}^{(\Theta)}=\mathbf{0}} + \mathbf{p} + (s - p(0)) \cdot \mathbf{1} \ .$$

The last equivalence follows from observing that p is a polynomial of degree $\leq t-1$ and a Shamir's sharing is obtained from a random degree-$(t-1)$ polynomial. Hence an element of the right-hand-side distribution is a Shamir's sharing of s. Note that $\mathsf{ShaSh}_{n,t}(0)|_{\mathbf{s}^{(\Theta)}=\mathbf{0}}$ has a clean characterization as follows: sample a random polynomial q of degree $t - \theta - 1$ such that $q(0) = 0$, consider the augmented polynomial $q'(x) = q \cdot \prod_{i \in \Theta}(x - \alpha_i)$ of degree $t-1$ where $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is the set of evaluation points for Shamir's secret sharing scheme. Finally the codeword is the evaluations of q' on A. This characterization allows us to see that $\mathsf{ShaSh}_{n,t}(0)|_{\mathbf{s}^{(\Theta)}=\mathbf{0}}$ is an $[n, t-\theta-1, n-t+2]$ code.

Now we are done. Consider C to be the restriction of $\mathsf{ShaSh}_{n,t}(0)|_{\mathbf{s}^{(\Theta)}=\mathbf{0}}$ to $\overline{\Theta}$ and the shift $\mathbf{b} = \mathbf{p} + (s - p(0)) \cdot \mathbf{1}$. Code C is an $[n - \theta, t - \theta - 1, n - t + 2]$ code because all the points excluded are 0 and hence do not affect the distance. This completes the argument. □

□

4.2.3 Example Parameter Settings

Let us now simplify the bounds for some specific parameter settings. All the statements in this section assumes that the parameters $n, p, m, \theta, \varepsilon$ are functions of some implicit parameter $\lambda \in \mathbb{N}$.

Additive Secret Sharing. The following corollary shows that for additive secret sharings, if a constant number of bits is leaked per share ($m = O(1)$), as long as the prime order p is larger than 2^m (i.e., not all the bits are leaked), if $n - \theta$ goes to infinity, the adversary advantage goes to 0 exponentially fast in $n - \theta$.

Corollary 4.10 (Additive Secret Sharing with Constant-Size Leakage)**.** *If $m = O(1)$, $p > 2^m$, the additive secret sharing* AddSh_n *for n parties is (θ, m, ε)-LL resilient where $\varepsilon = 2^{-\Omega(n-\theta)}$.*

Proof. Since $c_m < 1$ as soon as $p > 2^m$, there exists a constant $c > 0$ such that $c_m \leq 2^{-c}$ for all the values of the implicit parameter $\lambda \in \mathbb{N}$. Corollary 4.8 implies that the additive secret sharing AddSh_n is (θ, m, ε)-LL resilient when:

$$\varepsilon = 2^m \cdot c_m^{n-\theta-2} \leq 2^{m-c(n-\theta-2)} = 2^{-\Omega(n-\theta)} \ .$$

□

The following corollary shows in particular that for additive secret sharings, if all-but-one bit is leaked per share ($m = \lfloor \log_2 p - 1 \rfloor$), and $n - \theta = \Omega(p^2 \log p)$, the scheme is $\varepsilon = 1/3$-LL resilient.

Corollary 4.11 (Additive Secret Sharing with All-but-One Bit of Leakage)**.** *Let $\eta > 0$ be a constant. If p goes to infinity, $\theta < n-2$, and $m = \lfloor \log_2 p - \eta \rfloor$, then the additive secret sharing AddSh_n for n parties is (θ, m, ε)-LL resilient where $\varepsilon = 2^{m-\Omega(n-\theta)/p^2}$.*

The corollary is actually stronger than the informal statement above, as it holds even if "less than a bit" is not leaked (more precisely if the remaining min-entropy of each share is η).

Proof. We have:

$$\begin{aligned}
c_m &= \log \frac{2^m/\pi \cdot \sin(\pi/2^m)}{p/\pi \cdot \sin(\pi/p)} \\
&= \frac{\frac{2^m}{\pi} \cdot \left(\frac{\pi}{2^m} - \frac{\pi^3}{6 \cdot 2^{3m}} + O\left(\frac{1}{2^{5m}}\right)\right)}{\frac{p}{\pi} \cdot \left(\frac{\pi}{p} - \frac{\pi^3}{6 \cdot p^3} + O\left(\frac{1}{p^5}\right)\right)} \\
&= \left(1 - \frac{\pi^2}{6 \cdot 2^{2m}} + O\left(\frac{1}{2^{4m}}\right)\right) \cdot \left(1 + \frac{\pi^2}{6 \cdot p^2} + O\left(\frac{1}{p^4}\right)\right) \\
&\leq 1 - \frac{\pi^2}{6p^2} \cdot (2^{2\eta} - 1) + O\left(\frac{1}{p^4}\right) \ ,
\end{aligned}$$

where the inequality comes from the fact that $2^m \leq p/2^\eta$. We denote by c the constant $c = \pi^2(2^{2\eta} - 1)/(6 \log 2) > 0$, where log corresponds to the natural logarithm (while $\log_2$ corresponds to the logarithm in base 2). From the inequality $\log(1+x) \leq x$ for $x > -1$, we have:

$$\log_2 c_m \leq -\frac{c}{p^2} + O\left(\frac{1}{p^4}\right) \ . \tag{5}$$

Finally, Corollary 4.8 implies that the additive secret sharing AddSh_n is (θ, m, ε)-LL resilient when:

$$\varepsilon = 2^m \cdot c_m^{n-\theta-2} \leq 2^{m-(n-\theta-2)\cdot c/p^2 + O((n-\theta)/p^4)} \ ,$$

where the inequality comes Eq. (5). When the implicit parameter λ is large enough, p is small enough and the term $O((n-\theta)/p^4)$ in the inequality above is less than $(n-\theta-2) \cdot c/(2p^2)$. Thus, for large enough implicit parameter:

$$\varepsilon \leq 2^{m-(n-\theta-2)\cdot c/(2p^2)} \ .$$

This concludes the proof. □

Shamir's Secret Sharing. The following corollary shows that for Shamir's secret sharings, if a constant number of bits is leaked per share ($m = O(1)$) and a constant number of shares are completely leaked ($\theta = O(1)$), there exists $\alpha < 1$, such that if $t \geq \alpha n$ and if n goes to infinity, the adversary advantage goes to 0 exponentially fast in n.

Corollary 4.12 (Shamir's Secret Sharing with Constant-Size Leakage)**.** *If $m = O(1)$, $\theta = O(1)$, and n goes to infinity, there exists $\alpha < 1$, such that the Shamir's secret sharing scheme $\mathsf{ShaSh}_{n,t}$ for n parties and threshold $t \geq \alpha n$ is (θ, m, ε)-LL resilient where $\varepsilon = 2^{-\Omega(n)}$.*

Proof. When the implicit parameter $\lambda \in \mathbb{N}$ is large enough, $p > n$ is large, and $c'_m \leq 2^{-c}$ for some constant $c > 0$. Corollary 4.9 implies that the additive secret sharing AddSh_n is (θ, m, ε)-LL resilient where:

$$\begin{aligned} \varepsilon &= 2^{(5m+1)(n-t)+m} \cdot (c'_m)^{2t-n-\theta-2} && (6) \\ &\leq 2^{(5m+1)(n-t)+m-c(2t-n-\theta-2)} && (7) \\ &\leq 2^{(5m+1+c)n-(5m+1+2c)t+m+c\theta+2c} \quad . && (8) \end{aligned}$$

Hence choosing $\alpha > (5m+1+c)/(5m+1+2c)$ but still $\alpha < 1$, if $t \geq \alpha n$, we have:

$$\varepsilon \leq 2^{((5m+1+c)-\alpha(5m+1+2c))n+m+c\theta+2c} \leq 2^{-\Omega(n)} \quad ,$$

because $((5m+1+c) - \alpha(5m+1+2c))$ is a negative constant and $m + c\theta = O(1)$. □

Corollary 4.13 (Shamir's Secret Sharing with Constant-Fraction Leakage)**.** *Let $\theta = O(1)$. For sufficiently large n, for $n < p \leq 2n$ and $m = \lfloor(\log p)/4\rfloor$, the Shamir's secret sharing scheme $\mathsf{ShaSh}_{n,t}$ for n parties and threshold $t > n - n^{1/4}$ is (θ, m, ε)-LL resilient where $\varepsilon = 2^{-\Omega(\sqrt{n})}$.*

The proof relies on the following bound for c_m proved in Appendix A:

Proposition 4.14. *Let $m \geq 1$ and $p \geq 2$ be two integers. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)}$. We have:*

$$\log c_m \leq -\frac{1}{2^{2m+1}} + \frac{4}{p^2} \quad .$$

Corollary 4.13. Corollary 4.9 implies that Shamir's secret sharing is leakage resilient where $\varepsilon = p^{n-t+1} \cdot c_m^t$. Hence by Proposition 4.14, for $m = \lfloor(\log p)/4\rfloor$, we get that $\log c_m \leq -\frac{1}{2^{(\log p)/2+1}} - \frac{4}{p^2} < -\frac{1}{3\sqrt{p}}$ for large enough $n < p$. We have:

$$\varepsilon = p^{n-t+1} \cdot c_m^{t-\theta} \leq e^{\log p \cdot (n-t+1) - (t-\theta)\frac{1}{3\sqrt{p}}} \leq e^{\frac{\log p}{n^{1/4}} - \frac{t-\theta}{3\sqrt{p}}} \quad .$$

To complete the proof, observe that, $\frac{\log p}{n^{1/4}} - \frac{t-\theta}{3\sqrt{p}} < \frac{\log p}{n^{1/4}} - \frac{3n/4}{3\sqrt{p}} < -\sqrt{n}/16$ for large enough n as $\sqrt{p} < 2\sqrt{n}$, $\frac{\log p}{n^{1/4}} < \frac{\sqrt{n}}{16}$ and $\theta = O(1)$. Hence, $\varepsilon \leq 2^{-\sqrt{n}/16}$ as desired. □

4.3 Proofs of Theorems 4.5, 4.6, and 4.7

The proofs of all three statements follow a very similar outline. We describe the common parts of the proof and then specialize the proofs as required. For a linear code C and leakage functions $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ our overarching goal is to bound the statistical distance $\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n))$. The first part of the proof common to all the theorems is to write this statistical distance in a Fourier representation. The second part, which is specialized, uses different methods to bound this statistical distance.

Lemma 4.15. *Let $C \subseteq \mathbb{F}_p^n$ be any $[n, t-1, n-t+2]$ linear code. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. We abuse notation and define $1_{\ell_j}(x) = 1$ if $\tau^{(j)}(x) = \ell_j$ and 0 otherwise. We then have*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) = \frac{1}{2} \sum_{\vec{\ell} \in \{0,1\}^{m \times n}} \left| \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} \prod_j \widehat{1_{\ell_j}}(\alpha_j) \right| \quad .$$

Before proving the lemma, we want to briefly describe how the proofs of Theorems 4.5 to 4.7 follow. The three theorems apply different bounds for the Fourier expression above. Theorem 4.5 has the simplest proof, which consists of bounding each of the terms $\widehat{1_{\ell_j}}(\alpha_j)$ and then invoking convexity. But the proof yields a dependence on the number of terms (p^{n-t}) which is undesirable. This can be improved by using Cauchy-Schwarz inequality. This is done in Theorem 4.7 and Theorem 4.6. The case of additive secret sharing (Theorem 4.7) serves as a warm-up for the more intricate proof of Theorem 4.6. We start by proving Lemma 4.15.

Recall that $\omega = \exp(2\pi i/p)$ is a primitive p-th root of unity.

Proof of Lemma 4.15. We start by proving the Poisson Summation Formula for linear codes C. It shows that the expectation of product of functions over a code can be represented as a sum of products over the dual code. Then we show how this can prove the lemma.

Lemma 4.16 (Poisson Summation Formula). *Let $p > 2$ be a prime. Let $C \subseteq \mathbb{F}_p^n$ be a linear code with dual code is $C^\perp$. Let $f_1, f_2, \ldots, f_n : \mathbb{F}_p \to \mathbb{C}$ be functions. Let Λ be defined as follows:*

$$\Lambda(f_1, f_2, \ldots, f_n) = \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[f_1(x_1) \cdot f_2(x_2) \cdots f_n(x_n)\right] ,$$

where $\vec{x} = (x_1, x_2, \ldots, x_n)$. Then, the following holds:

$$\Lambda(f_1, f_2, \ldots, f_n) = \sum_{\vec{\alpha} \in C^\perp} \widehat{f_1}(\alpha_1) \cdot \widehat{f_2}(\alpha_2) \cdots \widehat{f_n}(\alpha_n) \ ,$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{F}_p^n$.

Proof of Poisson Summation Formula (Lemma 4.16). The proof is a calculation that uses the fact that for any fixed $\vec{\alpha}$, the inner product $\langle \vec{x}, \vec{\alpha} \rangle$, where $\vec{x} \leftarrow C$, is always 0 when $\vec{\alpha} \in C^\perp$ and uniformly random otherwise.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\prod_i f_i(x_i)\right] &= \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\prod_i \sum_{\alpha_i \in \mathbb{F}_p} \widehat{f_i}(\alpha_i)\overline{\chi_{\alpha_i}(x_i)}\right] \\
&= \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\sum_{\vec{\alpha} \in \mathbb{F}_p^n} \prod_i \widehat{f_i}(\alpha_i)\overline{\chi_{\alpha_i}(x_i)}\right] \\
&= \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\sum_{\vec{\alpha} \in \mathbb{F}_p^n} \left(\prod_i \widehat{f_i}(\alpha_i)\right)\omega^{-\langle \vec{x}, \vec{\alpha} \rangle}\right] \\
&= \sum_{\vec{\alpha} \in \mathbb{F}_p^n} \left(\prod_i \widehat{f_i}(\alpha_i)\right) \cdot \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\omega^{-\langle \vec{x}, \vec{\alpha} \rangle}\right] \\
&= \sum_{\vec{\alpha} \in C^\perp} \prod_i \widehat{f_i}(\alpha_i) \ ,
\end{aligned}$$

where the first equality follows from the Fourier Inversion Formula (Theorem 3.9(b)), the third equality follows because $\chi_{\alpha_i}(x) = \omega^{\alpha_i \cdot x}$ and the last equality follows because $\mathbb{E}_{\vec{x} \leftarrow C}\left[\omega^{-\langle \vec{x}, \vec{\alpha} \rangle}\right] = 1$ if $\vec{\alpha} \in C^\perp$ and 0 otherwise. □

Equipped with Lemma 4.16, we now prove Lemma 4.15. This proof primarily specializes the Poisson Summation formula to the specific case of leakage functions. Note that given any output leakage value $\vec{\ell} = (\ell_1, \ldots, \ell_n)$,

$$\Pr_{\vec{x} \leftarrow C}\left[\boldsymbol{\tau}(\vec{x}) = \vec{\ell}\right] = \mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[1_{\ell_1}(x_1) \cdot 1_{\ell_2}(x_2) \cdots 1_{\ell_n}(x_n)\right] \ .$$

This is simply saying that $1_{\ell_i}(x_i)$ indicates whether the leakage from the share x_i is the corresponding value ℓ_i. Hence, we have:

$$\begin{aligned}
\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) &= \frac{1}{2}\sum_{\vec{\ell}}\left|\mathop{\mathbb{E}}_{\vec{x} \leftarrow C}\left[\prod_j 1_{\ell_j}(x_j)\right] - \mathop{\mathbb{E}}_{\vec{x} \leftarrow U_n}\left[\prod_j 1_{\ell_j}(x_j)\right]\right| \\
&= \frac{1}{2}\sum_{\vec{\ell}}\left|\sum_{\vec{\alpha} \in C^{\perp}} \prod_j \widehat{1_{\ell_j}}(\alpha_j) - \mathop{\mathbb{E}}_{\vec{x} \leftarrow U_n}\left[\prod_j 1_{\ell_j}(x_j)\right]\right| \\
&= \frac{1}{2}\sum_{\vec{\ell}}\left|\sum_{\vec{\alpha} \in C^{\perp} \setminus \left\{\vec{0}\right\}} \prod_j \widehat{1_{\ell_j}}(\alpha_j)\right| \ ,
\end{aligned}$$

where the second equality follows from Poisson Summation (Lemma 4.16), the third equality follows from the fact that $\mathbb{E}_{\vec{x} \leftarrow U_n}\left[\prod_j 1_{\ell_j}(x_j)\right] = \prod_j \left(\left|\left(\tau^{(j)}\right)^{-1}(\ell_j)\right|/p\right) = \prod_j \widehat{1_{\ell_j}}(0)$. This completes the proof of Lemma 4.15. □

4.3.1 Proof of Theorem 4.5

We recall Theorem 4.5 below.

Theorem 4.5. *Let $C \subseteq \mathbb{F}_p^n$ be any $[n, t-1, n-t+2]$ linear code. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} < 1$ (when $2^m < p$). Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \tfrac{1}{2} \cdot p^{n-t+1} \cdot c_m^t \ ,$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$ and:

$$\boldsymbol{\tau}(C) = \left\{\left(\tau^{(i)}(x_i)\right)_{i \in [n]} \ : \ \vec{x} \leftarrow C\right\} \quad \textit{and} \quad \boldsymbol{\tau}(U_n) = \left\{\left(\tau^{(i)}(x_i)\right)_{i \in [n]} \ : \ \vec{x} \leftarrow U_n\right\} \ .$$

In Lemma 4.15, we represented the statistical distance $\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n))$ in terms of Fourier coefficients of some characteristic functions of the leakage. Next, in Lemma 4.17, we show bounds on these Fourier coefficients, which then allows us to complete the proof of Theorem 4.5.

Lemma 4.17. *Let m be some positive real number such that 2^m is an integer. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)}$. For any sets $A_1, A_2, \ldots, A_{2^m} \subseteq \mathbb{F}_p$, such that $\sum_{i=1}^{2^m} |A_i| = p$, we have:*

$$\begin{cases}
\displaystyle\sum_{i=1}^{2^m}\left|\widehat{1_{A_i}}(\alpha)\right| \leq c_m & \textit{if } \alpha \neq 0 \ , \\
\displaystyle\sum_{i=1}^{2^m}\left|\widehat{1_{A_i}}(\alpha)\right| = 1 & \textit{if } \alpha = 0 \ ,
\end{cases}$$

where $1_A \colon \mathbb{F}_p \to \{0,1\}$ is the characteristic function of the set $A \subseteq \mathbb{F}_p$ (i.e., $1_A(x) = 1$ if $x \in A$ and 0 otherwise).

Proof. This proof relies on Lemma 3.11 and uses concavity to argue about partitions. The case $\alpha = 0$ follows directly from the following fact:

$$\widehat{1_A}(0) = \mathop{\mathbb{E}}_x\left[1_A(x) \cdot \omega^{0 \cdot x}\right] = |A|/p \ .$$

Let us now focus on the case $\alpha \neq 0$. Recall that $\zeta_k(x) = \frac{\sin(x\pi/k)}{\sin(\pi/k)}$. Let $t_i = |A_i|$. As $\alpha \neq 0$, observe that $\widehat{1_A}(\alpha) = \mathbb{E}_x[1_A(x) \cdot \omega^{\alpha x}] = p^{-1} \cdot \boldsymbol{\omega}^{\alpha A}$, where $\omega = \exp(\frac{2\pi i}{p})$ and $\alpha A = \{\alpha x : x \in A\}$ has the same size as A. We have:

$$\begin{aligned} p \sum_i \left|\widehat{1_{A_i}}(\alpha)\right| = \sum_i \left|\boldsymbol{\omega}^{\alpha A_i}\right| \leq \sum_i \zeta_p(t_i) = \frac{1}{\sin(\pi/p)} \sum_i \sin(\pi t_i/p) \\ \leq \frac{1}{\sin(\pi/p)} \cdot 2^m \cdot \sin(\pi/2^m) = p \cdot \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} = p \cdot c_m \end{aligned}$$

where the first inequality follows from Lemma 3.11, the second inequality follows from the concavity of the $\sin(\cdot)$ function between $[0, \pi]$ and hence the function is maximized when all $t_i = p/2^m$. □

Completing the Proof. At this point, given Lemmas 4.15 and 4.17 we can complete the proof of Theorem 4.5.

Proof of Theorem 4.5. We recall that we abuse notation and define $1_{\ell_j}(x) = 1_{\tau^{-1}(\ell_j)}$. We can express the statistical distance as follows:

$$\begin{aligned} \mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) &= \frac{1}{2} \sum_{\vec{\ell}} \left| \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} \prod_j \widehat{1_{\ell_j}}(\alpha_j) \right| \\ &\leq \frac{1}{2} \sum_{\vec{\ell}} \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} \prod_j \left| \widehat{1_{\ell_j}}(\alpha_j) \right| \\ &= \frac{1}{2} \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} \prod_j \left(\sum_{\ell_j} \left| \widehat{1_{\ell_j}}(\alpha_j) \right| \right) , \end{aligned}$$

where the first equality comes from Lemma 4.15 and the first inequality follows from the triangle inequality. To complete the proof, we bound $\sum_{\ell_j} \left|\widehat{1_{\ell_j}}(\alpha_j)\right|$ using Lemma 4.17 and get:

$$\leq \frac{1}{2} \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} c_m^{\mathsf{HW}(\vec{\alpha})} \leq \frac{1}{2} \sum_{\vec{\alpha} \in C^\perp \setminus \{\vec{0}\}} c_m^t \leq \frac{1}{2} |\mathsf{C}^\perp| \cdot c_m^t \ ,$$

where $\mathsf{HW}(\cdot)$ denotes the Hamming weight. The last inequality follows from the fact that the dual code $C^\perp$ has minimum distance t, as C is a $[n, t-1, n-t+2]$ linear code and is thus MDS. We conclude by remarking that $|\mathsf{C}^\perp| = p^{n-t+1}$.

□

4.3.2 Warm-Up: Proof of Theorem 4.7

Next, we prove stronger bounds on additive secret sharing (Theorem 4.7). This serves as a warm-up to the general result (Theorem 4.6). We start by recalling Theorem 4.7.

Theorem 4.7. *[Additive Secret Sharing] Let $C \subseteq \mathbb{F}_p^n$ be the code generated by* $\mathsf{AddSh}(0)$*. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)} < 1$ (when $2^m < p$). Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \tfrac{1}{2} \cdot 2^m \cdot c_m^{n-2}$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$.

For the proof, we need a bound on Fourier coefficients. Hence, we start by stating and proving the following corollary of Lemma 4.17.

Lemma 4.18. *Let m be some positive real such that 2^m is an integer. Let $c_m = \frac{2^m \sin(\pi/2^m)}{p \sin(\pi/p)}$. For any sets $A_1, A_2, \ldots, A_{2^m} \subseteq \mathbb{F}_p$, such that $\sum_{i=1}^{2^m} |A_i| = p$, we have:*

$$\sum_{i=1}^{2^m} \max_{\alpha \neq 0} \left| \widehat{1_{A_i}}(\alpha) \right| \leq c_m \quad .$$

Proof. We remark that for $\alpha \neq 0$: $\widehat{1_A}(\alpha) = p^{-1} \cdot \boldsymbol{\omega}^{\alpha A}$. Hence, for any $i \in [2^m]$, there exists α_i such that:

$$\max_{\alpha \neq 0} \left| \widehat{1_{A_i}}(\alpha) \right| = \left| \widehat{1_{A_i}}(\alpha_i) \right| = p^{-1} \cdot \boldsymbol{\omega}^{\alpha_i A} = \left| \widehat{1_{\alpha_i A_i}}(1) \right| \quad .$$

We conclude using Lemma 4.17 on $\alpha = 1$ and the sets $\alpha_1 A, \ldots, \alpha_{2^m} A_{2^m}$. □

Proof of Theorem 4.7. We recall that we abuse notation and define $1_{\ell_j}(x) = 1_{\tau^{-1}(\ell_j)}$. We can express the statistical distance as follows thanks to Lemma 4.15:

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) = \frac{1}{2} \sum_{\vec{\ell}} \left| \sum_{\vec{\alpha} \in C^{\perp} \setminus \{\vec{0}\}} \prod_j \widehat{1_{\ell_j}}(\alpha_j) \right|$$

As the dual code of C is the linear code generated by $\vec{1}$ (the all-ones vector), we get that the sum is equivalently,

$$= \frac{1}{2} \sum_{\vec{\ell}} \left| \sum_{\alpha \neq 0} \widehat{1_{\ell_1}}(\alpha) \cdot \widehat{1_{\ell_2}}(\alpha) \cdots \widehat{1_{\ell_n}}(\alpha) \right|$$

Now, we use Cauchy-Shwarz to get that,

$$\leq \frac{1}{2} \sum_{\vec{\ell}} \left\| \widehat{1_{\ell_1}} \right\|_2 \cdot \left\| \widehat{1_{\ell_2}} \right\|_2 \cdot \max_{\alpha \neq 0} \left| \widehat{1_{\ell_3}}(\alpha) \right| \cdots \max_{\alpha \neq 0} \left| \widehat{1_{\ell_n}}(\alpha) \right|$$

$$= \frac{1}{2} \left(\sum_{\ell_1} \left\| \widehat{1_{\ell_1}} \right\|_2 \right) \cdot \left(\sum_{\ell_2} \left\| \widehat{1_{\ell_2}} \right\|_2 \right) \cdot \left(\sum_{\ell_3} \max_{\alpha \neq 0} \left| \widehat{1_{\ell}}(\alpha) \right| \right) \cdots \left(\sum_{\ell_n} \max_{\alpha \neq 0} \left| \widehat{1_{\ell_n}}(\alpha) \right| \right) \quad .$$

To complete the proof we use the following claim.

Claim 4.18.1. *For any $j \in [n]$, $\sum_{\ell_j \in \{0,1\}^m} \left\| \widehat{1_{\ell_j}} \right\|_2 \leq 2^{m/2}$.*

Proof. We have $\left\|\widehat{1_{\ell_j}}\right\|_2 = \left\|1_{\ell_j}\right\|_2 = \sqrt{\Pr_{\alpha\leftarrow\mathbb{F}_p}\left[1_{\ell_j}(\alpha) = 1\right]}$. Furthermore the events $[1_{\ell_j}(\alpha) = 1]$ are pairwise disjoints for $\ell_j \in \{0,1\}^m$, and $\sum_{\ell_j\in\{0,1\}^m} \Pr_{\alpha\leftarrow\mathbb{F}_p}\left[1_{\ell_j}(\alpha) = 1\right] = 1$. Thus:

$$\sum_{\ell_j\in\{0,1\}^m} \left\|\widehat{1_{\ell_j}}\right\|_2 = \sum_{\ell_j\in\{0,1\}^m} \sqrt{\Pr_{\alpha}\left[1_{\ell_j}(\alpha) = 1\right]} \leq 2^m \cdot \sqrt{\frac{1}{2^m} \cdot \sum_{\ell_j\in\{0,1\}^m} \Pr_{\alpha\leftarrow\mathbb{F}_p}\left[1_{\ell_j}(\alpha) = 1\right]} = 2^{m/2} \ ,$$

where the inequality comes from the concavity of $x \mapsto \sqrt{x}$. □

To complete the proof, observe that $\sum_{\ell_j} \max_{\alpha\neq 0}\left|\widehat{1_{\ell_j}}(\alpha)\right| \leq c_m$ by Lemma 4.18. This implies:

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \frac{1}{2} \cdot 2^{m/2} \cdot 2^{m/2} \cdot c_m^{n-2} \ .$$

□

4.3.3 Proof of Theorem 4.6

We turn towards proving Theorem 4.6. The strategy again is to use Cauchy-Schwarz. Now we need a significantly more delicate variant of Lemma 4.18. We start by recalling the theorem.

Theorem 4.6. *Let $C \subseteq \mathbb{F}_p^n$ be any $[n, t-1, n-t+2]$ linear code. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \longrightarrow \{0,1\}^m$. Let $c'_m = \frac{2^m \sin(\pi/2^m + \pi/2^{4m})}{p\sin(\pi/p)}$. Then,*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \frac{1}{2} \cdot 2^{(5m+1)\cdot(n-t)+m} \cdot (c'_m)^{2t-n-2} \ ,$$

where U_n is the uniform distribution on $\mathbb{F}_p^n$.

Let us now prove Theorem 4.6. We start by a lemma that uses the Cauchy-Schwarz inequality to get rid of the sum over all codewords in $C^\perp$ in the proof of Theorem 4.5 and hence remove the dependence on p, at the expense of a factor $2^{m\cdot(n-t+1)-1}$ and a more complicated expression involving some maximum over all codewords in $C^\perp$. Then, we will show a bound on that expression. We begin by describing a property of all MDS Codes.

Proposition 4.19. *An $[n, k, d]$ linear code C is an MDS code if and only if every set of $n-k$ columns of its parity check matrix $H \in \mathbb{F}_p^{(n-k)\times n}$ are linearly independent.*

Proof. The code C exactly consists of all codewords $\vec{x}$ such that $H\vec{x} = 0$. If there exists a set of $n-k$ columns of H that are not linearly independent, then there exists a vector $\vec{v}$ of Hamming weight at most $n-k$ such that $H\vec{v} = 0$. Thus the minimum distance d of C is at most $n-k$ and C is not MDS.

Conversely, if C is not an MDS code, it contains a vector $\vec{v}$ of Hamming weight at most $n-k$ and the set of (at most $n-k$) columns of H corresponding to the non-zero coefficients of $\vec{v}$ are not linearly independent. □

Lemma 4.20. *Let C be an $[n, t-1, n-t+2]$ linear MDS code with parity check matrix H. Partition the indices of the columns of H into $[n] = I_1 \cup I_2 \cup I_3$ where I_1, I_2 have size $n-t+1$ each. Let $\{\vec{h}_j^\intercal\}_{j\in[n]}$ be the family of the columns of H. Let m be a positive integer. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ be any family of* *leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. We abuse notation and define $1_{\ell_j}(x) = 1$ if $\tau^{(j)}(x) = \ell_j$ and 0 otherwise. We then have:*

$$\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \leq \frac{1}{2} \cdot 2^{m\cdot(n-t+1)} \cdot \sum_{\{\ell_j\}_{j\in I_3}} \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| \ ,$$

where $\{\ell_j\}_{j\in I_3} \in \{0,1\}^{2t-n-2}$.

The core of the proof is the following lemma which aims at bounding the Fourier expression

$$\sum_{\{\ell_j\}_{j\in I_3}} \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| \ .$$

Lemma 4.21. *Let $D \subseteq \mathbb{F}_p^k$ be any code of distance at least d. Let $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(k)})$ be any family of leakage functions where $\tau^{(j)} : \mathbb{F}_p \to \{0,1\}^m$. We abuse notation and define $1_{\ell_j}(x) = 1$ if $\tau^{(j)}(x) = \ell_j$ and 0 otherwise. Let $c'_m = \frac{2^m \sin(\pi/2^m+\pi/2^{4m})}{p\sin(\pi/p)}$. We then have:*

$$\sum_{\vec{\ell}\in(\{0,1\}^m)^k} \max_{\vec{\alpha}\in D} \prod_{j=1}^{k} \left|\widehat{1_{\ell_j}}(\alpha_j)\right| \leq 2^{(4m+1)\cdot(k-d)} \cdot c_m'^k \ .$$

We first finish the proof of Theorem 4.6 assuming Lemmas 4.20 and 4.21 and then prove the lemmas.

Proof of Theorem 4.6. Let $k = |I_3| = 2t - n - 2$. Let $D = \{\{x_j\}_{j\in I_3} : \vec{x} \in C^\perp\} \subseteq \mathbb{F}_p^k$. As $C^\perp$ is an $[n, n-t+1, t]$ code, D is a $[k, k', d]$ code, such that $k' \leq n - t + 1$ and $d \geq t - (n-k)$ code (hence, $k - d \leq n - t$). We then use Lemma 4.20 followed by Lemma 4.21 to get:

$$\begin{aligned}\mathrm{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) &\leq \frac{1}{2} \cdot 2^{m\cdot(n-t+1)} \cdot \sum_{\{\ell_j\}_{j\in I_3}} \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| \\ &\leq \frac{1}{2} \cdot 2^{m\cdot(n-t+1)} \cdot 2^{(4m+1)\cdot(k-d)} \cdot c_m'^k \leq \frac{1}{2} \cdot 2^{(5m+1)\cdot(n-t)+m} \cdot c_m'^k \ .\end{aligned}$$

This concludes the proof of Theorem 4.6. □

Next we prove the two lemmas. The first one is applying Cauchy-Schwarz on subsets of coordinates I_1 and I_2 and the second bounds Fourier coefficients.

Proof of Lemma 4.20. By Lemma 4.15, we can express the statistical distance as follows:

$$\begin{aligned}
&\mathsf{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) \\
&= \frac{1}{2}\sum_{\vec{\ell}} \left| \sum_{\vec{\alpha}\in C^{\perp}\setminus\left\{\vec{0}\right\}} \prod_{j} \widehat{1_{\ell_j}}(\alpha_j) \right| \\
&= \frac{1}{2}\sum_{\vec{\ell}} \left| \sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j} \widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle) \right| \\
&= \frac{1}{2}\sum_{\vec{\ell}} \left| \sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \left(\prod_{j\in I_1} \widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right)\cdot\left(\prod_{j\in I_2\cup I_3} \widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right) \right| \\
&\leq \frac{1}{2}\sum_{\vec{\ell}} \sqrt{\sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \left|\prod_{j\in I_1} \widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right|^2} \cdot \sqrt{\sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \left|\prod_{j\in I_2\cup I_3} \widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right|^2} \\
&\leq \frac{1}{2}\sum_{\vec{\ell}} \sqrt{\sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_1} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right|^2} \cdot \sqrt{\sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_2} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right|^2} \cdot \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| ,
\end{aligned}$$

where the first inequality comes from the Cauchy-Schwarz inequality.

Since $\{\vec{h}_j^{\intercal}\}_{j\in I_1}$ is a basis of $\mathbb{F}_p^{(n+t-1)\times 1}$ from Proposition 4.19, the function $\vec{\beta}\in\mathbb{F}_p^{n-t+1}\mapsto\{\langle\vec{\beta},\vec{h}_j\rangle\}_{j\in I_1}\in\mathbb{F}_p^{n-t+1}$ is bijective. We can then write

$$\sum_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_1} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right|^2 = \sum_{\{\vec{\alpha}_j\}_{j\in I_1}\in\mathbb{F}_p^{n-t+1}} \prod_{j\in I_1} \left|\widehat{1_{\ell_j}}(\alpha_j)\right|^2 = \prod_{j\in I_1}\sum_{\alpha\in\mathbb{F}_p} \left|\widehat{1_{\ell_j}}(\alpha)\right|^2 = \prod_{j\in I_1} \left\|\widehat{1_{\ell_j}}\right\|_2^2 .$$

The same holds when I_1 is replaced by I_2 and we thus have:

$$\begin{aligned}
\mathsf{SD}(\boldsymbol{\tau}(C), \boldsymbol{\tau}(U_n)) &\leq \frac{1}{2}\sum_{\vec{\ell}} \prod_{j\in I_1\cup I_2} \left\|\widehat{1_{\ell_j}}\right\|_2 \cdot \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| \\
&= \frac{1}{2}\left(\prod_{j\in I_1\cup I_2}\sum_{\ell_j}\left\|\widehat{1_{\ell_j}}\right\|_2\right)\cdot \sum_{\{\ell_j\}_{j\in I_3}} \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| \\
&\leq \frac{1}{2}\cdot 2^{|I_1\cup I_2|\cdot m/2}\cdot \sum_{\{\ell_j\}_{j\in I_3}} \max_{\vec{\beta}\in\mathbb{F}_p^{n-t+1}\setminus\{\vec{0}\}} \prod_{j\in I_3} \left|\widehat{1_{\ell_j}}(\langle\vec{\beta},\vec{h}_j\rangle)\right| ,
\end{aligned}$$

where the last inequality comes from Claim 4.18.1 and where $|I_1\cup I_2|$ is the cardinal of $I_1\cup I_2$. We conclude by using the fact that $|I_1\cup I_2| = 2\cdot(n-t+1)$. □

We now prove Lemma 4.21.

Proof of Lemma 4.21. We want to bound:

$$\eta = \sum_{\vec{\ell} \in (\{0,1\}^m)^k} \max_{\vec{\alpha} \in D \setminus \{\vec{0}\}} \prod_{j=1}^{k} \left| \widehat{1_{\ell_j}}(\alpha_j) \right| \ .$$

When all the non-zero vectors $\vec{\alpha} \in D$ have no zero coefficients, bounding η is easy, as we can write $\eta \leq \prod_{j=1}^{k} \sum_{\ell_j} \max_{\alpha_j \neq 0} \left| \widehat{1_{\ell_j}}(\alpha_j) \right|$ and proceed as before using Lemma 4.18 as in Theorem 4.7. The issue is that when this is not the case, each term of the sum might be maximized by a vector $\vec{\alpha}$ with different positions of the 0 coefficients. When $\alpha_j = 0$, $\sum_{\ell_j \in \{0,1\}^m} \widehat{1_{\ell_j}}(0) = 1$, hence bounding η requires a more careful analysis.

To handle this, we introduce a different bound for $|\widehat{1_{\ell_j}}(\alpha_j)|$, one that allows us to control for this issue of the positions of the zero coefficients being different for different terms. We introduce the bound ξ below. The key difference between ξ and ζ is that ξ is bounded below. This allows us to bound the multiplicative gap between the case when $\alpha = 0$ and otherwise.

Lemma 4.22. *Let $\xi_p(x) = \max(\zeta_p(x)/p, 2^{-(4m+1)})$. Then ξ has the following properties:*

1. ***Bounds non-zero Fourier coefficients.*** *For every set A of size t and $\alpha \neq 0$,*

$$|\widehat{1_A}(\alpha)| \leq \xi_p(t) \ .$$

2. ***Bounds zero Fourier coefficients multiplicatively.*** *For every set A of size t,*

$$|\widehat{1_A}(0)| \leq 2^{4m+1} \cdot \xi_p(t) \ .$$

3. *ξ_p* ***is bounded over partitions.*** *Let $A_1, A_2, \ldots A_{2^m}$ be any partition of $\mathbb{Z}_p$. Then,*

$$\sum_i \xi(|A_i|) \leq c'_m \ .$$

We first prove Lemma 4.21 assuming Lemma 4.22 and then prove Lemma 4.22. The following calculation proves Lemma 4.21. The key idea in this calculation is that due to the definition of ξ, the max over codewords reduces to counting how many zeros the codeword has, and this is $k - d$. We need some notation: let us set $t_{\ell_j, j} = |\tau_j^{-1}(\ell_j)|$ and indicator $\mathbf{1}_0(\alpha_j)$ equal to 1 when $\alpha_j = 0$ and

0 otherwise.

$$\begin{aligned}
\sum_{\vec{\ell}\in(\{0,1\}^m)^k} \max_{\vec{\alpha}\in D\setminus\{\vec{0}\}} \prod_{j=1}^{k} \left|\widehat{1_{\ell_j}}(\alpha_j)\right| &\leq \sum_{\vec{\ell}\in(\{0,1\}^m)^k} \max_{\vec{\alpha}\in D\setminus\{\vec{0}\}} \prod_{j=1}^{k} \xi(t_{\ell_j,j}) \cdot (2^{4m+1})^{\mathbf{1}_0(\alpha_j)} \\
&= \sum_{\vec{\ell}\in(\{0,1\}^m)^k} \prod_{j=1}^{k} \xi(t_{\ell_j,j}) \cdot \max_{\vec{\alpha}\in D\setminus\{\vec{0}\}} \prod_{j=1}^{k} (2^{4m+1})^{\mathbf{1}_0(\alpha_j)} \\
&\leq \sum_{\vec{\ell}\in(\{0,1\}^m)^k} 2^{(4m+1)\cdot(k-d)} \cdot \prod_{j=1}^{k} \xi(t_{\ell_j,j}) \\
&= 2^{(4m+1)\cdot(k-d)} \cdot \prod_{j=1}^{k} \sum_{\ell_j\in\{0,1\}^m} \xi(t_{\ell_j,j}) \\
&\leq 2^{(4m+1)\cdot(k-d)} \cdot \prod_{j=1}^{k} c'_m \\
&\leq 2^{(4m+1)\cdot(k-d)} \cdot c'^{k}_m \ ,
\end{aligned}$$

where the first inequality follows from using Lemma 4.22 parts (1, 2) with $\alpha \neq 0$ and $\alpha = 0$ respectively; the first equality is a rearrangement; the second inequality follows from observing that D is a code with distance at least d and hence can only have at most $k-d$ zeros; the second equality is a rearrangement; and the third inequality follows from Lemma 4.22 part (3) with the partition, $\left\{\tau_j^{-1}(\ell_j)\right\}_{\ell_j\in\{0,1\}^m}$. □

We now prove Lemma 4.22.

Proof of Lemma 4.22. We prove the three parts in the three claims below. The first two claims follow from the definition easily while the last claim requires a computation similar to Lemma 4.17 involving concavity.

Claim 4.22.1 (Part 1)**.** ***Bounds non-zero Fourier coefficients.*** *For every set A of size t and $\alpha \neq 0$,*

$$|\widehat{1_A}(\alpha)| \leq \xi_p(t) \ .$$

From Lemmas 3.11 and 4.17, we know that $|\widehat{1_A}(\alpha)| = |\boldsymbol{\omega}^{\alpha A}|/p \leq \zeta_p(t)/p$. The claim follows as, $\xi_p(t) = \max(\zeta_p(t)/p, 2^{-4m+1})$.

Claim 4.22.2 (Part 2)**.** ***Bounds zero Fourier coefficients.*** *For every set A of size t and $\alpha \neq 0$,*

$$|\widehat{1_A}(0)| \leq 2^{4m+1} \cdot \xi_p(t) \ .$$

This follows from the observation that $2^{4m+1} \cdot \xi_p(t) \geq 1$ as $\xi_p(t) \geq 2^{-4m-1}$ and that $|\widehat{1_A}(0)| \leq 1$.

Claim 4.22.3. ***ξ_p is bounded over partitions.*** *Let $A_1, A_2, \ldots, A_{2^m}$ be any partition of $\mathbb{Z}_p$. Then,*

$$\sum_i \xi(|A_i|) \leq c'_m$$

Proof. This claim is a consequence of the concavity of the sine function. We start by observing that $\zeta_p(p/2^{4m})/p = \frac{\sin(\pi/2^{4m})}{p\sin(\pi/p)} \geq 2^{-(4m+1)}$. The inequality comes from $\sin(\pi/p) \leq \pi/p \leq 4/p$ and $\sin(\pi/2^{4m}) \geq (2/\pi)\cdot(\pi/2^{4m})$. Hence, $\xi_p(t) = \max(2^{-(4m+1)}, \zeta_p(t)/p) \leq \max(\zeta_p(p/2^{4m})/p, \zeta_p(t)/p) = \zeta_p(\max(t, p/2^{4m}))/p$.

We are now in a position to complete the proof. Let $t_1, t_2, \ldots, t_{2^m}$ be the sizes of the sets $A_1, \ldots A_{2^m}$. Then,

$$\begin{aligned}
\sum_i \xi(|A_i|) &= \sum_i \xi(t_i) \\
&\leq \sum_i \zeta_p(\max(t_i, p/2^{4m}))/p \\
&= \sum_i \frac{\sin(\pi \cdot \max(t_i, p/2^{4m})/p)}{p\sin(\pi/p)} \\
&= \frac{1}{p\sin(\pi/p)} \cdot \sum_i \sin(\pi \cdot \max(t_i, p/2^{4m})/p) \\
&\leq \frac{2^m}{p\sin(\pi/p)} \cdot \sin\left(\frac{\pi}{2^m} \sum_i \frac{\max(t_i, p/2^{4m})}{p}\right) \\
&\leq \frac{2^m}{p\sin(\pi/p)} \cdot \sin\left(\frac{\pi}{2^m} + \frac{\pi}{2^{4m}}\right) = c'_m \quad ,
\end{aligned}$$

where the first inequality was described above, the second inequality comes from the concavity of the sine function in $[0, \pi]$, the third inequality comes from the fact that $\sum_i \max(t_i, p/2^{4m}) \leq \sum_i (t_i + p/2^{4m}) \leq p + 2^m \cdot p/2^{4m}$. □

This concludes the proof of Lemma 4.22 and hence Lemma 4.21. □

5 Leakage Resilience of GMW with preprocessing

In this section, we describe an application of the results on leakage resilience of secret sharing to MPC protocols. Here too, our goal is to show that *natural* MPC protocols that are based on *linear* secret sharing achieve local leakage resilience. Concretely, we show that a variant of the GMW protocol [GMW87] with preprocessing is leakage resilient. We start by defining the notion of MPC protocols with input preprocessing. Then describe our security definitions and our results.

We consider arithmetic circuits over a field $\mathbb{F}$ over a basis $\mathbb{B} = \{+, \times, -1\}$ where the -1 gate negates the input. For convenience, we have input gates that read a field element from the input. The following definition of an MPC protocol is adapted from [GIM+16] (Definition 3).

Definition 5.1 (n-party protocol with encoded input and output). *An n-party protocol for $f : \mathbb{F}^{n_{\text{in}}} \to \mathbb{F}^{n_{\text{out}}}$ is defined by $\Pi = (I, \mathbf{R}, \mathbf{M}, O)$, where:*

- Input Encoder. $I : \mathbb{F}^{n_{\text{in}}} \to (\mathbb{F}^{\hat{n}_{\text{in}}})^n$ *is a randomized* input encoder *circuit, which maps an input $\vec{x}$ for f to a tuple of protocol inputs $\vec{\mathbf{x}} = (\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(n)})$ one for each party.*
- Randomness. $\mathbf{R} = (R^{(1)}, R^{(2)}, \ldots, R^{(n)})$ *are distributions over $\mathbb{F}^{n_r}$ that capture the random inputs of the parties. They are assumed to be correlated due to preprocessing.*

- $\mathbf{M} = (M^{(1)}, M^{(2)}, \dots, M^{(n)})$ *are deterministic* next message functions *where* $M^{(j)}$ *determines the next message sent by party* j *as a function of its input* $\vec{x}^{(j)}$, *random input* $r^{(j)}$, *and the sequence of messages received in the previous rounds. Messages are sent in rounds where each party sends a message to possibly every other party. After a predetermined number of rounds, the function* $M^{(j)}$ *returns a local output* $\vec{y}^{(j)} \in \mathbb{F}^{\hat{n}_{\text{out}}}$ *for party* j.
- $O : (\mathbb{F}^{\hat{n}_{\text{out}}})^n \to \mathbb{F}^{n_{\text{out}}}$ *is a deterministic* output decoder *circuit, which maps a tuple of protocol outputs* $\vec{\mathbf{y}} = (\vec{y}^{(1)}, \dots, \vec{y}^{(n)})$ *to an output* $\vec{y}$ *of* f.

For $\vec{x} \in \mathbb{F}^{n_{\text{in}}}$, *we denote by* $\Pi(\vec{x})$ *the output of* Π *on input* $\vec{x}$, *namely the result of applying the input encoder* I *to* $\vec{x}$, *interacting as specified by* $\mathbf{R}, \mathbf{M}$, *and applying the output decoder* O *to the vector of protocol outputs. We say that* Π correctly computes $f : \mathbb{F}^{n_{\text{in}}} \to \mathbb{F}^{n_{\text{out}}}$ *if for every input* $\vec{x} \in \mathbb{F}^{n_{\text{in}}}$, *we have* $\Pr\left[\Pi(\vec{x}) = f(\vec{x})\right] = 1$.

We denote by $\mathbf{view}(\vec{x})$ *the joint distribution* $(\mathsf{view}^{(1)}(\vec{x}), \dots, \mathsf{view}^{(n)}(\vec{x}))$ *obtained by running* Π *on input* $\vec{x}$, *where* $\mathsf{view}^{(j)}$ *includes the encoded input* $\vec{x}^{(j)}$, *the random input* $r^{(j)}$ *(sampled from* $R^{(j)}$*), and the sequence of messages received by party* j. *(The messages sent by party* j *as well as its output* $\vec{y}^{(j)}$ *are uniquely determined by* $\mathsf{view}^{(j)}$*.)*

We denote by $\mathbf{out}(\vec{x})$ *the joint distribution of the outputs* $\vec{\mathbf{y}}$.

5.1 Security Definitions

The definition we consider uses the simulation paradigm. We only consider an *honest-but-curious* definition, albeit one where the adversary can leak information from the views of the uncorrupted parties. We consider two security notions: private-outputs local leakage resilience and public-outputs local leakage resilience.

In the private-outputs case, the adversary does not learn the local outputs $\vec{y}^{(j)}$ of non-corrupted parties nor the output $\vec{y} = \Pi(\vec{x})$. This would model the setting where a client wants to delegate some computation $f(\vec{x})$ to some leaky parties: the client secret-shares $\vec{x}$ into $\vec{\mathbf{x}}$, sends each share $\vec{x}^{(j)}$ to the party j, the parties run the protocol Π, and each party j sends back its output share $\vec{y}^{(j)}$ to the client.

In the public-outputs case, the adversary learns all the local outputs $\vec{\mathbf{y}}$ of all the parties (and in particular learns the output $\vec{y} = O(\vec{\mathbf{y}}) = \Pi(\vec{x})$). This models a setting where at the end of the computation, the parties would broadcast their local outputs $\vec{y}^{(j)}$ to jointly reconstruct the output $\vec{y}$.

Definition 5.2 (Private-Outputs Local Leakage Resilient Protocol)**.** *We say that* Π *is* (Θ, m, ε)-private-outputs local leakage resilient *for* f *(or* (Θ, m, ε)*-priv-LL-resilient for short) if* Π *correctly computes* f, *and the following security requirement holds. For any family of local leakage functions* $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(n)})$ *where* $\tau^{(j)}$ *is a function that outputs* m *bits, there exists a simulator* $\mathsf{LeakSim}_{\Theta,\vec{\tau}}$ *such that, for any input* $\vec{x} \in \mathbb{F}^{n_{\text{in}}}$, *we have*

$$\mathsf{SD}\big(\mathsf{Leak}_{\Theta,\vec{\tau}}(\mathbf{view}(\vec{x})),\ \mathsf{LeakSim}_{\Theta,\vec{\tau}}()\big) \le \varepsilon.$$

We say that Π *is* (θ, m, ε)-priv-LL-resilient *if* Π *is* (Θ, m, ε)*-LL-resilient for all subsets* $\Theta \subseteq [n]$ *of at most size* θ.

We recall that Leak is defined in Eq. (4) on page 19.

Definition 5.3 (Public-Outputs Local Leakage Resilient Protocol)**.** *We say that* Π *is* (Θ, m, ε)-public-outputs local leakage resilient *for* f *(or* (Θ, m, ε)*-pub-LL-resilient for short) if* Π *correctly computes* f, *and the following security requirement holds. For any family of local leakage functions* $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \dots,$

$\tau^{(n)}$) where $\tau^{(j)}$ is a function that outputs m bits, there exists a simulator $\mathsf{LeakSim}_{\Theta,\vec{\tau}}$ such that, for any input $\vec{x} \in \mathbb{F}^{n_{\mathrm{in}}}$, we have

$$\mathsf{SD}\big((\mathbf{out}(\vec{x}), \mathsf{Leak}_{\Theta,\vec{\tau}}(\mathbf{view}(\vec{x}))),\ \mathsf{LeakSim}_{\Theta,\vec{\tau}}(f(\vec{x}))\big) \leq \varepsilon.$$

*We say that Π is (θ, m, ε)-*pub-LL-resilient *if Π is (Θ, m, ε)-pub-LL-resilient for all subsets $\Theta \subseteq [n]$ of at most size θ.*

Both definitions model a protocol executed in the presence of a real-world adversary $\mathcal{A}$ that may corrupt a subset Θ of the parties. The adversary learns the entire view of corrupted parties (and in the second case, also the output of all parties). As we consider *semi-honest* corruptions, the adversary can only observe their views but does not modify the messages they send. The adversary also leaks independently m bits from each party.

Note that the classical notion of security against semi-honest adversaries corrupting at most θ parties is equivalent to $(\theta, 0, \varepsilon)$-priv-LL-resilient.

5.2 GMW with Shared Product Preprocessing

Notation. Let f be a function computed by a given circuit C. Let G be the set of all gates in C and $G_\times$ be the set of multiplication gates in C. For any input $\vec{x}$, let z_g denote the value at gate $g \in G$ in the circuit C when the input is $\vec{x}$.

In Fig. 3, we describe a variant of the GMW [GMW87] protocol based on the ideas of Beaver triples [Bea91] that we call GMW with shared product preprocessing. The protocol works with any linear secret sharing. We show that if the underlying linear secret sharing is local leakage resilient, then the protocol is pub-LL-resilient and priv-LL-resilient.

Let us first prove correctness.

Proposition 5.4 (Correctness)**.** *The protocol Π in Fig. 3 on any input $\vec{x}$ correctly computes $f(\vec{x})$.*

Proof. To prove correctness, we show that at every gate g, the parties maintain a linear secret sharing of the value z_g. This is easy to verify for the addition, -1 and input gates. We will only do the verification for the multiplication case.

Consider any multiplication gate g with input gates g_1, g_2. Assume that the parties have a valid secret sharing $\mathbf{z}_{g_1}$ and $\mathbf{z}_{g_2}$ of values z_{g_1} and z_{g_2} respectively. Pick any valid Beaver triple $(\mathbf{a}_g, \mathbf{b}_g, (\mathbf{ab})_g)$. We need to show that $\mathbf{z}_g$ as computed is a valid secret sharing of $z_g = z_{g_1} z_{g_2}$. We remark that:

$$\mathbf{z}_g = (z_{g_1} - a_g)(z_{g_2} - b_g) \cdot \mathbf{1} + (z_{g_1} - a_g) \cdot \mathbf{b}_g + \mathbf{a}_g \cdot (z_{g_2} - b_g) + (\mathbf{ab})_g \quad .$$

By linearity $\mathbf{z}_g$ is a secret sharing of:

$$(z_{g_1} - a_g)(z_{g_2} - b_g) \cdot 1 + (z_{g_1} - a_g) \cdot b_g + a_g \cdot (z_{g_2} - b_g) + a_g b_g = z_{g_1} z_{g_2} \quad . \tag{9}$$

This concludes the proof.

□

We have the following security theorems.

Theorem 5.5. *If the linear secret sharing scheme* (Share, Rec) *is (Θ, m, ε)-LL-resilient then the protocol Π in Fig. 3 is (Θ, m, ε)-priv-LL-resilient.*

GMW with Shared Product Preprocessing for computing f with circuit C on field $\mathbb{F}$

Parameters: n the number of parties. $(\mathsf{Share}, \mathsf{Rec})$ a secret sharing scheme for n parties. $\mathbf{1}$ an arbitary sharing of 1.

Input Encoder $I(\vec{x})$:

1. Sample $\vec{\mathbf{x}} \leftarrow \mathsf{Share}(\vec{x})$.
2. Output $\vec{\mathbf{x}}$.

Output Decoder $I(\vec{\mathbf{y}})$:

1. Output $\vec{y} = \mathsf{Rec}(\vec{\mathbf{y}})$

Randomness $R(C)$:

1. For each multiplication gate g in C,
   - (a) Generate $a_g, b_g \leftarrow \mathbb{F}$.
   - (b) Generate $\mathbf{a}_g \leftarrow \mathsf{Share}(a_g)$, $\mathbf{b}_g \leftarrow \mathsf{Share}(b_g)$, and $(\mathbf{ab})_g \leftarrow \mathsf{Share}(a_g \cdot b_g)$.
   - (c) Append to $r^{(j)}$ the tuple $(a_g^{(j)}, b_g^{(j)}, (ab)_g^{(j)})$.
2. Output $\mathbf{r} = (r^{(1)}, r^{(2)}, \ldots, r^{(n)})$.

Protocol run by Party j (defining $M^{(j)}$)

1. Set $\mathsf{state}^{(j)} = (n, C, \vec{x}^{(j)})$.
2. Iterate over gates in C in fixed topological order such that for every gate, its input gates are visited before the gate. And run the subprotocol "Process Gate" below.
3. Output $z_{g_{\text{out}}}^{(j)}$: the share of the output gate g_{out}.

Process Gate g:

1. If gate g is (a) an input gate with input x_i, or, (b) a (-1) gate with input from gate g', or, (c) a $+$ gate with inputs g_1, g_2, then, set $z_g^{(j)}$ as follows:

$$z_g^{(j)} = \begin{cases} x_i^{(j)} & \text{if } g \text{ is an input gate} \\ -z_{g'}^{(j)} & \text{if } g \text{ is a } -1 \text{ gate} \\ z_{g_1}^{(j)} + z_{g_2}^{(j)} & \text{if } g \text{ is a } + \text{ gate} \end{cases}$$

   and append $z_g^{(j)}$ to the list $\mathsf{state}^{(j)}$.

2. If g is a $\times$ gate, with input gates g_1 and g_2, then do the following:
   - (a) Compute $a_g'^{(j)} = z_{g_1}^{(j)} - a_g^{(j)}$ and $b_g'^{(j)} = z_{g_2}^{(j)} - b_g^{(j)}$ and broadcast these values.
   - (b) Receive the corresponding values from other parties.
   - (c) Compute $z_{g_1} - a_g$ and $z_{g_2} - b_g$ from all the values received, using the reconstruction algorithm Rec.
   - (d) Compute $z_g^{(j)} = (z_{g_1} - a_g)(z_{g_2} - b_g) \cdot 1^{(j)} + (z_{g_1} - a_g) \cdot b_g^{(j)} + a_g^{(j)} \cdot (z_{g_2} - b_g) + (ab)_g^{(j)}$, where $1^{(j)}$ is the j-th share of an arbitrary sharing $\mathbf{1}$ of 1.
   - (e) Append $z_g^{(j)}$ and $(a_g^{(j)}, b_g^{(j)}, (ab)_g^{(j)})$ to $\mathsf{state}^{(j)}$.

Figure 3: GMW Protocol with Shared Product Preprocessing

Theorem 5.6. *If the linear secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *is* (Θ, m, ε)*-LL-resilient then the protocol* Π *in Fig. 3 is* (Θ, m, ε)*-pub-LL-resilient.*

Since an (n, t)-secret sharing scheme is $(t, 0, 0)$-LL-resilient, when instantiated with an (n, t)-secret sharing scheme, the protocol is $(t, 0, 0)$-priv-LL resilient and thus secure against a semi-honest adversary corrupting up to t parties.

Before we prove Theorems 5.5 and 5.6, let us state the following lemma.

Lemma 5.7 (Parallel Composition of LL-Resilience)**.** *If* $(\mathsf{Share}, \mathsf{Rec})$ *is a* (Θ, m, ε)*-LL-resilient linear secret sharing scheme, then for any leakage function family* $\boldsymbol{\tau} = (\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(n)})$ *where* $\tau^{(j)}$ *has an* m*-bit output, and for any* $\vec{y}, \vec{y}' \in \mathbb{F}^k$*:*

$$\mathsf{SD}\Big(\big\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\vec{\mathbf{y}}) \ : \ \vec{\mathbf{y}} \leftarrow \mathsf{Share}(\vec{y})\big\}, \big\{\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\vec{\mathbf{y}}') \ : \ \vec{\mathbf{y}}' \leftarrow \mathsf{Share}(\vec{y}')\big\}\Big) \leq \varepsilon.$$

Note that the bound on statistical distance does not degrade with the size of the vectors.

Note that this lemma allows us to avoid using a union bound in our theorems and hence avoid losing a factor of the number of multiplication gates.

Proof. This proof is a reduction showing that if local leakage can distinguish between $\vec{y}$ and $\vec{y}'$ then we can use this to also break the local leakage resilience of the underlying linear secret sharing scheme and distinguish between any two secrets $s \neq s'$. The proof follows from the observation that given shared randomness, the parties can *locally, without interaction* convert shares of s and s' to random shares of vectors $\vec{y}$ and $\vec{y}'$ respectively. This holds for any linear secret sharing scheme.

For contradiction, assume that there exist $\vec{y}, \vec{y}'$ and m-bit leakage functions $\boldsymbol{\tau}$ such that

$$\mathsf{SD}(\mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\vec{\mathbf{y}}), \mathsf{Leak}_{\Theta,\boldsymbol{\tau}}(\vec{\mathbf{y}}')) > \varepsilon.$$

Consider any two secrets $s \neq s' \in \mathbb{F}$. We will show that the scheme $\mathsf{Share}, \mathsf{Rec}$ is not local leakage resilient for these two secrets.

As $s \neq s'$, for every i, there exist constants $\lambda_{i,1}, \lambda_{i,0} \in \mathbb{F}$ such that $\lambda_{i,1} \cdot s + \lambda_{i,0} = y_i$ and $\lambda_{i,1} \cdot s' + \lambda_{i,0} = y'_i$. So, to do a local share conversion, the parties given share $\mathbf{x}$ of either s or s' do the following: Set $\mathbf{y}_i = \lambda_{i,1} \cdot \mathbf{x} + \boldsymbol{\lambda_{i,0}}$ where $\boldsymbol{\lambda_{i,0}} \leftarrow \mathsf{Share}(\lambda_{i,0})$ generated using the shared randomness. That is, party j locally computes the share: $z_i^{(j)} = \lambda_{i,1} x^{(j)} + \lambda_{i,0}^{(j)}$ where $x^{(j)}$ is the input share given to party j and $\lambda_{i,0}^{(j)}$ is the share of $\lambda_{i,0}$ generated using common randomness.

Because of the linearity of the secret sharing scheme, The distribution $\vec{\mathbf{z}}$ locally generated by the parties is identical to the distribution of fresh shares $\vec{\mathbf{y}} \leftarrow \mathsf{Share}(\vec{y})$ if the input $\mathbf{x}$ was a sharing of s or is identical to $\vec{\mathbf{y}}' \leftarrow \mathsf{Share}(\vec{y}')$ if the underlying secret encoded was s'. So, using this reduction gives a local leakage attack to distinguish between the shares of s and s' and hence a contradiction. □

5.3 Proof of Private-Outputs Local Leakage Resilience (Theorem 5.5)

To prove the private-outputs local leakage resilience (Theorem 5.5), we first start with a lemma that characterizes what information the parties see, both individually and jointly. Informally, we show that, when the protocol evaluates the circuit C on input $\vec{x}$, the view of each party (or any subset of parties) can be simulated given a set of common random values and the party's share in a sharing of each output of a multiplication gate. Then, the leakage resilience of the secret sharing scheme allows us to replace the secret sharings used by the simulator by secret sharings of any arbitrary value.

Lemma 5.8. *There exists simulator* S *such that for every input* $\vec{x}$*, the following two distributions are identical.*

$$\mathbf{view}(\vec{x}) \equiv \left\{ \left(\mathsf{S}(j, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}) \right)_{j\in[n]} \;:\; \begin{array}{c} \vec{\mathbf{x}} \leftarrow \mathsf{Share}(\vec{x}) \\ (\mathbf{z}_g \leftarrow \mathsf{Share}(z_g))_{g\in G_\times} \\ (a'_g, b'_g \leftarrow \mathbb{F})_{g\in G_\times} \\ (\mathbf{a}'_g \leftarrow \mathsf{Share}(a'_g))_{g\in G_\times} \\ (\mathbf{b}'_g \leftarrow \mathsf{Share}(b'_g))_{g\in G_\times} \end{array} \right\}.$$

Assuming Lemma 5.8, the proof of Theorem 5.5 (private-outputs-LL-resilience of Π) is immediate.

Proof of Theorem 5.5. Correctness comes from Proposition 5.4, while LL-resilience follows directly by combining Lemma 5.8 with Lemma 5.7: the simulator $\mathsf{LeakSim}()$ samples secret sharings $\vec{\mathbf{x}} \leftarrow \mathsf{Share}(0)$ and $\left(\mathbf{z}_g \leftarrow \mathsf{Share}(0)\right)_{g\in G_\times}$, as well as $\left(\mathbf{a}'_g, \mathbf{b}'_g \leftarrow \mathsf{Share}(a'_g, b'_g)\right)_{g\in G_\times}$ (with $a_g, b_g \leftarrow \mathbb{F}$) and returns

$$\mathsf{Leak}_{\Theta,\tau}\left(\left(\mathsf{S}(j, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}) \right)_{j\in[n]} \right).$$

□

Proof of Lemma 5.8. We describe the simulator and show perfect simulation. Each party's view is described by the internal state state and the messages received. Roughly speaking, for a multiplication gate g with inputs g_1 and g_2, the common vectors $\mathbf{a}'_g$ and $\mathbf{b}'_g$ correspond to the values $\mathbf{z}_{g_1} - \mathbf{a}$ and $\mathbf{z}_{g_2} - \mathbf{b}$ that are publicly broadcast. Given these values and the party's shares of z_{g_1} and z_{g_2}, the simulator can construct the Beaver triple via a simple computation. The simulator proceeds gate by gate reconstructing the views of each party. We describe the simulator in Fig. 4.

We now have to show that the simulator perfectly simulates the view of all n-parties in the protocol Π. We will show that for every possible communication $\left(\mathbf{a}'_g, \mathbf{b}'_g\right)_{g\in G_\times}$ and wire-label sharing obtained in the protocol, there exists a unique set of valid Beaver triples that give rise to this communication and state pattern and vice versa. This proof proceeds by induction. Let $\mathbf{state} = \left(\mathsf{state}^{(j)}\right)_{j\in[n]}$ be the joint distributions of the states of the parties in the protocol. As the base case, observe that before any gate is processed, the state in both the simulator and the actual parties is identical. For each party, it consists of the description of the circuit and the secret shares of the input.

Inductive Step. In the induction step, let us observe the joint state **state** after one more gate is processed. We naturally have two cases: if the gate is not a multiplication gate and if the gate is a multiplication gate.

Case 1. Not a Multiplication Gate. In this case, there is no interaction and each party simply appends the value $z_g^{(j)}$ to their state. As this process is deterministic and both the protocol and the simulator use identical procedures to generate the value, if the distribution of **state** was identical before processing this gate, it stays identical afterwards.

Simulator $\mathsf{S}\Big(j, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g \in G_\times}\Big)$:

1. Set state$^{(j)} = (n, C, \vec{x}^{(j)})$.
2. Iterate over the gates of C in the same order as the protocol. On each gate, do the following:
   (a) If gate g is (a) an input gate with input x_i, or, (b) a (-1) gate with input from gate g', or, (c) a + gate with inputs g_1, g_2, then, set $z_g^{(j)}$ as follows:

$$z_g^{(j)} = \begin{cases} x_i^{(j)} & \text{if } g \text{ is an input gate} \\ -z_{g'}^{(j)} & \text{if } g \text{ is a } -1 \text{ gate} \\ z_{g_1}^{(j)} + z_{g_2}^{(j)} & \text{if } g \text{ is a + gate} \end{cases}$$

   (b) If g is a $\times$ gate, with input gates g_1 and g_2, then do the following:
      i. Set broadcast message to be $(a_g'^{(j)}, b_g'^{(j)})$.
      ii. Set received messages to be $(a_g'^{(j')}, b_g'^{(j')})_{j' \neq j}$.
      iii. Set Beaver triple as: $a_g^{(j)} = z_{g_1}^{(j)} - a_g'^{(j)}$, $b_g^{(j)} = z_{g_2}^{(j)} - b_g'^{(j)}$, and $(ab)_g^{(j)} = z_g^{(j)} - (\sum_{j'} a_g'^{(j')}) \cdot (\sum_{j'} b_g'^{(j')}) \cdot 1^{(j)} - (\sum_{j'} a_g'^{(j')}) \cdot b_g^{(j)} - a_g^{(j)} \cdot (\sum_{j'} b_g'^{(j')})$.

Figure 4: Simulator for Lemma 5.8

Case 2. Multiplication Gate. In this case, the simulator is processing a multiplication gate g with inputs g_1 and g_2. In this case, we need to show that the input shares, the communication, the Beaver triple and the output share are consistently distributed in the actual protocol and the simulation. We remark that in the real world, we have:

$$\begin{aligned} \mathbf{a}'_g &= \mathbf{z}_{g_1} - \mathbf{a}_g & \mathbf{b}'_g &= \mathbf{z}_{g_2} - \mathbf{b}_g \\ \mathbf{z}_g &= a'_g b'_g \cdot \mathbf{1} + a'_g \cdot \mathbf{b}_g + \mathbf{a}_g \cdot b'_g + (\mathbf{ab})_g \end{aligned}$$

where $\mathbf{a}_g$, $\mathbf{b}_g$, and $(\mathbf{ab})_g$ are independent secret sharings of the values a_g, b_g, $a_g b_g$. Thus, by the linearity property of the secret sharing, $\mathbf{a}'_g$, $\mathbf{b}'_g$, and $\mathbf{z}_g$ are independent secret sharings of the values $a'_g = z_{g_1} - a_g$, $b'_g = z_{g_2} - b_g$, and $z_g = z_{g_1} z_{g_2}$ (see Eq. (9) for this latter value). Furthermore, as a_g and b_g are independently and uniformly random, so are a'_g and b'_g.

We conclude by remarking that the simulation sets:

$$\begin{aligned} \mathbf{a}_g &= \mathbf{z}_{g_1} - \mathbf{a}'_g & \mathbf{b}_g &= \mathbf{z}_{g_2} - \mathbf{b}'_g \\ (\mathbf{ab})_g &= -a'_g b'_g \cdot \mathbf{1} - a'_g \cdot \mathbf{b}_g - \mathbf{a}_g \cdot b'_g + \mathbf{z}_g \end{aligned}$$

and these three equations are equivalent to the ones above for the real world.

□

5.4 Proof of Public-Outputs Local Leakage Resilience (Theorem 5.6)

To prove the public-outputs local leakage resilience (Theorem 5.6), we extend Lemma 5.8 to take into account the output shares.

Lemma 5.9. *There exists a simulator S' such that for every input $\vec{x}$, the following two distributions are identical:*

$$
\begin{aligned}
&(\mathbf{out}(\vec{x}), \mathbf{view}(\vec{x}))\\
&\equiv \left\{ \left(\vec{\mathbf{y}},\ \left(\mathsf{S}'(j, \vec{\mathbf{y}}, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times})\right)_{j\in[n]}\right) \ :\ \begin{array}{c} \vec{y} = f(\vec{x});\ \vec{\mathbf{y}} \leftarrow \mathsf{Share}(\vec{y})\\ \vec{\mathbf{x}} \leftarrow \mathsf{Share}(\vec{x})\\ (\mathbf{z}_g \leftarrow \mathsf{Share}(z_g))_{g\in G_\times}\\ (a'_g, b'_g \leftarrow \mathbb{F})_{g\in G_\times}\\ (\mathbf{a}'_g \leftarrow \mathsf{Share}(a'_g))_{g\in G_\times}\\ (\mathbf{b}'_g \leftarrow \mathsf{Share}(b'_g))_{g\in G_\times} \end{array} \right\}.
\end{aligned}
$$

Assuming Lemma 5.9, the proof of Theorem 5.5 (pub-LL-resilience of Π) is immediate.

Proof of Theorem 5.5. Correctness comes from Proposition 5.4, while LL-resilience follows directly by combining Lemma 5.9 with Lemma 5.7: the simulator $\mathsf{LeakSim}(\vec{y})$ samples secret sharings $\vec{\mathbf{y}} \leftarrow \mathsf{Share}(\vec{y})$, $\vec{\mathbf{x}} \leftarrow \mathsf{Share}(0)$, and $\left(\mathbf{z}_g \leftarrow \mathsf{Share}(0)\right)_{g\in G_\times}$, as well as $\left(\mathbf{a}'_g, \mathbf{b}'_g \leftarrow \mathsf{Share}(a'_g, b'_g)\right)_{g\in G_\times}$ (with $a_g, b_g \leftarrow \mathbb{F}$) and returns

$$\vec{\mathbf{y}},\ \mathsf{Leak}_{\Theta,\tau}\left(\left(\mathsf{S}'(j, \vec{\mathbf{y}}, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times})\right)_{j\in[n]}\right).$$

□

Proof of Lemma 5.9. We start by remarking that if each output y_i is an output z_{g_i} of a × gate g_i (and all the outputs correspond to distinct gates), then the simulator S' is straightforward: it just runs the simulator S from Lemma 5.8 where $z_{g_i}^{(j)}$ is replaced by $y_i^{(j)}$ (and the inputs $\mathbf{z}_{g_i}$ are not used by S'), i.e.:

$$\mathsf{S}'\left(j, \vec{\mathbf{y}}, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}\right) = \mathsf{S}\left(j, \vec{x}^{(j)}, (z_g'^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}\right)$$

with

$$z_g'^{(j)} = \begin{cases} y_i^{(j)} & \text{if } g = g_i \text{ for some } i,\\ z_g^{(j)} & \text{otherwise.} \end{cases}$$

However, in general the outputs y_i can be any linear combination inputs x_i or output of × gate z_g ($g \in G_\times$). More formally, we can write $\vec{y} = \Phi(\vec{x}, (z_g)_{g\in G_\times})$, where Φ is a linear map. In a real execution of Π, we also have for all $j \in [n]$: $\vec{y}^{(j)} = \Phi(\vec{x}^{(j)}, (z_g^{(j)})_{g\in G_\times})$. Using Gaussian elimination, we can show that there exists a subset $A \subseteq [n_{\text{in}}]$, a subset $B \subseteq G_\times$, and a linear application Ψ such that, for all $j \in [n]$:

$$\left((x_i^{(j)})_{i\in A}, (z_g^{(j)})_{g\in B}\right) = \Psi\left(\vec{y}^{(j)}, (x_i^{(j)})_{i\in \bar{A}}, (z_g^{(j)})_{g\in \bar{B}}\right),$$

where $\bar{A} = [n_{\text{in}}] \setminus A$ and $\bar{B} = [n_{\text{in}}] \setminus B$ (the elements $\left(x_i^{(j)}\right)_{i\in A}$ and $\left(z_g^{(j)}\right)_{g\in B}$ correspond to the pivots of the system). We can then define S' as follows:

$$\mathsf{S}'\left(j, \vec{\mathbf{y}}, \vec{x}^{(j)}, (z_g^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}\right) = \mathsf{S}\left(j, \vec{x}'^{(j)}, (z_g'^{(j)}, \mathbf{a}'_g, \mathbf{b}'_g)_{g\in G_\times}\right)$$

with

$$\begin{aligned}\Big((x_i'^{(j)})_{i\in A}, (z_g'^{(j)})_{g\in B}\Big) &= \Psi\Big(\vec{y}^{(j)}, (x_i^{(j)})_{i\in\bar{A}}, (z_g^{(j)})_{g\in\bar{B}}\Big),\\ \Big((x_i'^{(j)})_{i\in\bar{A}}, (z_g'^{(j)})_{g\in\bar{B}}\Big) &= \Big((x_i^{(j)})_{i\in\bar{A}}, (z_g^{(j)})_{g\in\bar{B}}\Big).\end{aligned}$$

This concludes the proof. (We remark that the simulator S' does not use $x_i^{(j)}$ for $i \in A$, not $z_g^{(j)}$ for $g \in B$, but instead derive these values from $\vec{\mathbf{y}}$, $x_i^{(j)}$ for $i \in A$, and $z_g^{(j)}$ for $g \in B$.) □

6 On the Impossibility of Local Share Conversion

We start by defining Local Share Conversion. This section has two differences in notation. First, as we will only be dealing with singleton secrets and not vectors, we will use the subscript notation to avoid clutter. That is, when we say $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ we mean that s_i is held by party i. The second change is that because our results concern share conversion on schemes on $\mathbb{F}_p$ and $\mathbb{F}_2$, we will be careful about the ambient field of the secret sharing scheme and write it explicitly, e.g., AddSh_p instead of AddSh as earlier.

Definition 6.1 (Local Share Conversion, adapted from [BIKO12])**.** *Consider two n-party secret sharing schemes $\mathcal{L} = (\mathsf{Share}_{\mathcal{L}}, \mathsf{Rec}_{\mathcal{L}})$ and $\mathcal{L}' = (\mathsf{Share}_{\mathcal{L}'}, \mathsf{Rec}_{\mathcal{L}'})$ be over the domains of secrets $\mathbb{F}$ and $\mathbb{F}'$ respectively, and let $R \subseteq \mathbb{F} \times \mathbb{F}'$ be a relation such that for every secret $s \in \mathbb{F}$, there exists at least one secret $s' \in \mathbb{F}'$ such that $(s, s') \in R$.*

We say that $\mathcal{L}$ is locally convertible *to $\mathcal{L}'$, with error probability ϵ, with respect to R if there exist local conversion functions $g_1, g_2, \ldots, g_k : \mathbb{F} \to \mathbb{F}'$ such that, for every $s \in \mathbb{F}$,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{Share}_{\mathcal{L}}(s)}\Big[(g_1(s_1), g_2(s_2), \ldots, g_n(s_n)) \in \mathsf{Share}_{\mathcal{L}'}(s') \text{ where } (s, s') \in R\Big] > 1 - \epsilon \ ,$$

where $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ is a random secret share of s and $(g_1(s_1), g_2(s_2), \ldots, g_n(s_n)) \in \mathsf{Share}_{\mathcal{L}'}(s')$ indicates that $(g_1(s_1), g_2(s_2), \ldots, g_n(s_n))$ is a valid, not necessarily random, secret sharing of s' under $\mathcal{L}'$.

Note that the definition given is weaker than the definition in [BIKO12] in the sense that we allow the share conversion scheme to be correct "only with high probability" and not "always correct." Because our results are impossibility results on local share conversion, ruling out the aforementioned definition only makes our results stronger. To state our impossibility results, we first define the notion of a non-trivial relation. Roughly speaking, a relation is trivial if it would no matter what secret is shared, it would be acceptable to output a fixed value by each party. We focus on local share conversion problems where the players have to convert secret sharing schemes over $\mathbb{F}_p$ to schemes over $\mathbb{F}_2$.

Definition 6.2. *A relation $R \subseteq \mathbb{F}_p \times \mathbb{F}_2$ is* non-trivial *if it satisfies the following:*

1. *Zero gets mapped to zero. That is, $(0, 0) \in R$ and $(0, 1) \notin R$.*
2. *Some non-zero element does not get mapped to zero. That is, there exists $a \in \mathbb{F}_p$ such that $(a, 0) \notin R$ and $(a, 1) \in R$.*

Note that, in this definition, the requirement that 0 gets mapped to 0 is just for convenience. It would suffice to say that there exists an a that has to be mapped to 0 and b that has to be mapped to 1. We begin by noting that non-trivial share-conversion schemes from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ for n-parties for all $n \geq 2$; and from $\mathbb{F}_p$ to $\mathbb{F}_2$ for *two* parties.

Example 6.3 (Non-Trivial Two-Party Share Conversion). Consider a non-trivial relation R where 0 and 1 have to be mapped to themselves and all other inputs can be arbitrarily mapped. Then the following scheme is a local share conversion from the additive secret sharing AddSh_p over $\mathbb{F}_p$ to the additive secret sharing AddSh_2 over $\mathbb{F}_2$: $g_1(x)$ on input $x \in \mathbb{F}_p$ views x as an integer between 0 to $p-1$ and outputs x mod 2. The function g_2 is defined as $g_2(x) = g_1(-x)$.

This local share conversion scheme works because when sharing 0, the two shares are x and $-x$. Hence the output would be the same. On the other hand, when sharing 1, the two shares are x and $-(x+1)$. Hence, with high probability, the outputs will be different from each other.

Local share-conversion schemes exist for a variety of non-trivial relations over $\mathbb{F}_{2^n}$ for additive secret sharing. This is enabled by the fact that $\mathbb{F}_{2^n}$ as an additive group has many subgroups.

Example 6.4 (Share Conversion over $\mathbb{F}_{2^n}$). Let $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_2$ be an $\mathbb{F}_2$-linear function (looking at $\mathbb{F}_{2^n}$ as a vector space over $\mathbb{F}_2$), i.e., $f(x+y) = f(x) + f(y)$. Consider the relation R where a has to be mapped to $f(a)$ for every a. The following share conversion scheme exists for R: $g_i(x)$ outputs $f(x)$. As f is linear, $\sum_i g_i(s_i) = \sum_i f(s_i) = f(s)$ where $\mathbf{s}$ is an additive secret sharing of s.

The example can also be generalized to Shamir's secret sharing over $\mathbb{F}_{2^n}$.

We now state our results.

On Additive Secret Sharing. We show that any three-party Additive Secret Sharing over $\mathbb{F}_p$ for any prime $p > 2$ is not locally convertible to an additive secret sharing over $\mathbb{F}_2$ for any non-trivial relation R.

Theorem 6.5. *Let $n \geq 3$. For any non-trivial relation R and for any local conversion scheme $g_1, g_2, \ldots, g_n : \mathbb{F}_p \longrightarrow \mathbb{F}_2$, there exists an element $s \in \mathbb{F}_p$ such that,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(s)} \left[(s, \sum_i g_i(s_i)) \notin R \right] \geq \frac{1}{6} .$$

As mentioned in the introduction, this result rules out a possible approach to constructing multiparty Homomorphic Secret Sharing schemes in the spirit Boyle, Gilboa and Ishai [BGI16] where one first obtains a multiplicative secret sharing of a bit b over a DDH group $\mathbb{G}$. That is, $g^b = g^x \cdot g^y$ where the parties hold x and y respectively and then convert the shares *locally* to additive shares of b over $\mathbb{Z}$. The generalized approach to constructing 3-party HSS schemes would also first construct a similar multiplicative sharing of the bit b, but among 3 parties, and then transform it to additive shares.

The proof for this impossibility result is reminiscent of the Fourier analysis proofs of the Blum-Luby-Rubinfeld Linearity test [BLR93, BCLR08].

On Shamir's Secret Sharing. We can show a similar impossibility result for share conversion from Shamir's secret sharing to additive secret sharing as well.

Theorem 6.6. *Let $n \geq 3$. The (n,t)-Shamir's secret sharing scheme, for $(n+3)/2 \leq t \leq n$, over $\mathbb{F}_p$ is not locally convertible to an additive secret sharing over $\mathbb{F}_2$ for any non-trivial relation R. That is, for any non-trivial relation R and local conversion scheme $g_1, g_2, \ldots, g_n : \mathbb{F}_p \longrightarrow \mathbb{F}_2$, there exists $s \in \mathbb{F}_p$ such that,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_p(s)} \left[(s, \sum_i g_i(s_i)) \notin R \right] \geq \frac{1}{\max(6,\ n+1)} .$$

The key technique in this proof is derived from the breakthrough work of Green and Tao [GT10] which involves using Gowers' Uniformity Norm to bound the success probability of the share conversion scheme.

Outline. To prove both the theorems, we first describe some Fourier analysis properties in Section 6.1 and then prove Theorems 6.5 and 6.6 in Sections 6.2 and 6.3 respectively.

6.1 More Fourier Analysis

In the next lemma, we show that any function F from $\mathbb{F}_p$ to $\{-1, 1\}$ cannot be too correlated with any non-zero character. The implication of this lemma is an 'inverse theorem' that if the bias of a function F is greater than $2/3$, then F is highly correlated with the trivial character χ_0 which is always 1.

Lemma 6.7. *Let $F : \mathbb{F}_p \to \{-1, 1\}$ for prime $p > 2$ be a function. Then, $|\widehat{F}(\alpha)| \le \frac{2}{3}$, for all $\alpha \neq 0$.*

This proof relies on the fact that the function F only takes values in $\{-1, 1\}$ while every non-zero character of $\mathbb{F}_p$ takes all the values in the set $\left\{1, \omega, \omega^2, \dots, \omega^{p-1}\right\}$. (Recall that $\omega = \exp(2\pi i/p)$ is a primitive p-th root of unity.) Hence the character and the function F cannot be too correlated.

Proof. This proof uses Lemma 3.11. Let $\gamma = e^{i\cdot\pi/p}$. Then $\gamma^p = -1$ and $\gamma^2 = \omega$. Also, $F(x)\cdot\omega^{\alpha x} \in \left\{\gamma^{2\alpha x}, \gamma^{2\alpha x+p}\right\}$. So, we can bound the Fourier coefficient $\widehat{F}(\alpha)$ as follows:

$$\left|\widehat{F}(\alpha)\right| = \left|\mathop{\mathbb{E}}_{x}[F(x)\cdot\omega^{\alpha x}]\right| \le \max_{\vec{z}\in\{0,1\}^p}\left|\frac{1}{p}\sum_{x}\gamma^{2\alpha x+p\cdot z_x}\right| \le \max_{\substack{A\subseteq\{0,1,\dots,2p-1\}\\ |A|=p}}\left|\frac{1}{p}\sum_{x\in A}\gamma^{\alpha x}\right| ,$$

where the first inequality follows from the fact that $F(x)\cdot\omega^{\alpha x} \in \left\{\gamma^{2\alpha x}, \gamma^{2\alpha x+p}\right\}$ and the second inequality follows from the fact that if $x \neq x' \pmod p$ then the two sets $\{\gamma^{2\alpha x}, \gamma^{2\alpha x+p}\}$ and $\{\gamma^{2\alpha x'}, \gamma^{2\alpha x'+p}\}$ are disjoint and hence no value repeats in the sum. Lemma 3.11 implies that this value is bounded by $p^{-1}\zeta_{2p}(p) = \dfrac{\sin(\pi p/(2p))}{p\sin(\pi/(2p))} = \dfrac{1}{p\sin(\pi/(2p))}$. This value is monotonically decreasing and is $2/3$ for $p = 3$.

□

Lemma 6.8. *Let $F : \mathbb{F}_p \to \{-1, 1\}$ be a function. If $|\widehat{F}(0)| > 1 - \epsilon$, then for every $a \in \mathbb{F}_p$,*

$$\Pr_x[F(x) = F(x + a)] > 1 - \epsilon \ .$$

A balanced function has $\widehat{F}(0) = 0$. When this quantity is large, the function has to be very unbalanced and nearly a constant. The lemma quantifies this.

Proof. Assume that $\widehat{F}(0) > 1 - \epsilon$. The other case is analogous. We use the relationship between $\widehat{F}(0)$ and the expectation to prove the lemma.

$$\begin{aligned}1 - \epsilon < \widehat{F}(0) &= \mathop{\mathbb{E}}_x[F(x)]\\ &= \Pr_x[F(x) = 1] - \Pr_x[F(x) = -1]\\ &= \Pr_x[F(x) = 1] - (1 - \Pr_x[F(x) = 1]) \ ,\end{aligned}$$

where the first equality follows from the definition of $\widehat{F}(0)$. Hence, it holds that, $\Pr_x[F(x) = 1] > 1 - \epsilon/2$. Next, we use the union bound to prove the lemma:

$$\begin{aligned}\Pr_x[F(x) = F(x+a)] &\geq \Pr_x[F(x) = 1 \wedge F(x+a) = 1] \\ &\geq \Pr_x[F(x) = 1] - \Pr_x[F(x+a) \neq 1] \\ &> 1 - \frac{\epsilon}{2} - \frac{\epsilon}{2} = 1 - \epsilon \ \ ,\end{aligned}$$

as $\Pr_x[F(x+a) \neq 1] = \Pr_x[F(x) \neq 1]$. □

6.2 On Additive Secret Sharing: Proof of Theorem 6.5

In this section, we prove Theorem 6.5. We first recall it below.

Theorem 6.5. *Let $n \geq 3$. For any non-trivial relation R and for any local conversion scheme $g_1, g_2, \ldots, g_n : \mathbb{F}_p \longrightarrow \mathbb{F}_2$, there exists an element $s \in \mathbb{F}_p$ such that,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(s)}\left[(s, \sum_i g_i(s_i)) \notin R\right] \geq \frac{1}{6} \ .$$

The main ingredient of this proof is the following 'inverse theorem' style lemma which says that if the $\{g_i\}$ functions locally convert additive shares of 0 over $\mathbb{F}_p$ into additive shares of 0 over $\mathbb{F}_2$, then the function g_1 (or any other g_i) is almost always constant.

Lemma 6.9. *Let $n \geq 3$, $\epsilon \leq 1/6$. Let $g_1, g_2, \ldots, g_n : \mathbb{F}_p \longrightarrow \mathbb{F}_2$ be functions. If,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[\sum_i g_i(s_i) \neq 0\right] < \epsilon, \tag{10}$$

where $\mathbf{s} = (s_1, \ldots, s_n)$, then for every $a \in \mathbb{F}_p$,

$$\Pr_{x \leftarrow \mathbb{F}_p}\left[g_1(x) = g_1(x+a)\right] > 1 - 2\epsilon \ . \tag{11}$$

First, assuming Lemma 6.9 we prove Theorem 6.5. Then we prove Lemma 6.9 itself. To prove Theorem 6.5, we leverage the fact that g_1 is almost always constant to argue that additive shares of any element $s \in \mathbb{F}_p$ will also be converted to additive shares of 0; thus deriving a contradiction to the non-triviality of the relation R.

Proof of Theorem 6.5 assuming Lemma 6.9. Let $\epsilon = 1/6$. Let us assume that the local share conversion algorithms are correct on shares of zero, i.e.,

$$\Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[\sum_i g_i(s_i) \neq 0\right] < \epsilon \ \ . \tag{12}$$

As R is a non-trivial relation, there exists an $s' \in \mathbb{F}_p$ such that $(s', 0) \notin R$ and $(s', 1) \in R$. To prove the theorem, it suffices to show that,

$$\Pr_{\mathbf{s}' \leftarrow \mathsf{AddSh}_p(s')}\left[\sum_i g_i(s'_i) = 0\right] > \epsilon \ \ .$$

Note that the distribution $\{(s_1 + s', s_2, \ldots, s_n) : (s_1, s_2, \ldots, s_n) \leftarrow \mathsf{AddSh}_p(0)\}$ is identically distributed to $\mathsf{AddSh}_p(s')$. Hence,

$$\begin{aligned}
\Pr_{\mathbf{s}' \leftarrow \mathsf{AddSh}_p(s')}\left[\sum_i g_i(s'_i) = 0\right] &= \Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[g_1(s_1 + s') + \sum_{i=2}^{n} g_i(s_i) = 0\right] \\
&\geq \Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[\left(g_1(s_1 + s') = g_1(s_1)\right) \wedge \sum_{i=1}^{n} g_i(s_i) = 0\right] \\
&\geq \Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[g_1(s_1 + s') = g_1(s_1)\right] - \Pr_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}\left[\sum_i g_i(s_i) \neq 0\right] \\
&\geq 1 - 3\epsilon > \epsilon \ ,
\end{aligned}$$

where the second inequality follows from the union bound, the third inequality from Lemma 6.9 and Eq. (12). This gives us the required contradiction. □

We now prove Lemma 6.9. In the proof of Lemma 6.9, we first represent the success probability of the share-conversion scheme in terms of the Fourier spectrum of the functions in the share-conversion scheme. We use this to infer that each of the share-conversion functions has a 'large' Fourier coefficient and use that to deduce that this share-conversion function is almost constant. As mentioned earlier, this analysis is reminiscent of the fourier analytic proof of the Blum, Luby, and Rubinfeld linearity test [BLR93] and group homomorphism testing of Ben-or, Coppersmith, Luby, and Rubinfeld [BCLR08].

Proof of Lemma 6.9. It would be convenient for us to define real-valued functions $G_i : \mathbb{F}_p \to \mathbb{R}$ as $G_i(x) = (-1)^{g_i(x)}$. Restated in terms of G_i's, Eq. (10) is equivalent to,

$$\Lambda(G_1, G_2, \ldots, G_n) = \mathop{\mathbb{E}}_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}[G_1(s_1) \cdots G_n(s_n)] > 1 - 2\epsilon \ .$$

Using Lemma 4.16, and noting that additive shares of 0 form a linear code with the dual code generated by the all-ones vector $\mathbf{1}$, we get that,

$$\begin{aligned}
1 - 2\epsilon &< \mathop{\mathbb{E}}_{\mathbf{s} \leftarrow \mathsf{AddSh}_p(0)}[G_1(s_1) \cdots G_n(s_n)] \\
&= \sum_{\alpha \in \mathbb{F}_p} \widehat{G}_1(\alpha) \cdot \widehat{G}_2(\alpha) \cdots \widehat{G}_n(\alpha) \\
&\leq \|\widehat{G}_1\|_\infty \cdot \|\widehat{G}_2\|_\infty \cdots \|\widehat{G}_{n-2}\|_\infty \cdot \|\widehat{G}_{n-1}\|_2 \cdot \|\widehat{G}_n\|_2 \\
&\leq \|\widehat{G}_1\|_\infty \ ,
\end{aligned}$$

where the first equality follows from Lemma 4.16, the subsequent inequality follows from the Cauchy-Schwarz inequality and the final inequality follows from the fact that: for each $i \in [n]$, $\|\widehat{G}_i\|_2 = \|G_i\|_2 \leq 1$ and $\|\widehat{G}_i\|_\infty \leq 1$.

This implies that $\|\widehat{G}_1\|_\infty > 1 - 2\epsilon \geq 2/3$. Lemma 6.7 implies that for any $\alpha \neq 0$, $|\widehat{G}_1(\alpha)| \leq 2/3$. Hence $|\widehat{G}_1(0)| > 1 - 2\epsilon$. Combining this with Lemma 6.8 shows that, for all $a \in \mathbb{F}_p$:

$$\Pr_x[G_1(x) = G_1(x + a)] > 1 - 2\epsilon \ .$$

This completes the proof as $G_1(x) = G_1(x + a) \iff g_1(x) = g_1(x + a)$. □

Note that this proof breaks down for two parties because using Cauchy-Shwarz does not let us infer that $\|\widehat{G}_1\|_\infty$ is large for either i's. This proof does generalize to other settings for example for share conversion from $\mathbb{F}_p$ to $\mathbb{F}_q$ for $q < p$. Though the error bound degrades with an exponential dependence in q.

6.3 On Shamir's Secret Sharing: Proof of Theorem 6.6

In this section, we prove Theorem 6.6. We recall the theorem below.

Theorem 6.6. *Let $n \geq 3$. The (n, t)-Shamir's secret sharing scheme, for $(n+3)/2 \leq t \leq n$, over $\mathbb{F}_p$ is not locally convertible to an additive secret sharing over $\mathbb{F}_2$ for any non-trivial relation R. That is, for any non-trivial relation R and local conversion scheme $g_1, g_2, \ldots, g_n : \mathbb{F}_p \to \mathbb{F}_2$, there exists $s \in \mathbb{F}_p$ such that,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_p(s)} \left[(s, \sum_i g_i(s_i)) \notin R \right] \geq \frac{1}{\max(6,\, n+1)} \ .$$

This is also a two step proof. The difficult step is proving an inverse theorem and then using it is relatively simple. The inverse theorem was proved in Green and Tao's breakthrough work [GT10]. While Green and Tao prove a more general result, we include for convenience, a proof for the specialized case of Shamir's secret sharing. We state the inverse theorem below.

Lemma 6.10 (Inverse Theorem for Shamir's Secret Sharing)**.** *Let $n, t \geq 3$ be two integers, such that $t \leq n \leq 2t-3$. Let $\epsilon \leq 1/6$. Let $g_1, g_2, \ldots, g_n : \mathbb{F}_p \to \mathbb{F}_2$ be functions such that,*

$$\Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)} \left[\sum_i g_i(s_i) \neq 0 \right] < \epsilon \ , \tag{13}$$

where $\mathbf{s} = (s_1, \ldots, s_n)$, then for every $a \in \mathbb{F}_p$,

$$\Pr_{x \leftarrow \mathbb{F}_p} \left[g_1(x) = g_1(x+a) \right] > 1 - 2\epsilon \ .$$

We will first prove Theorem 6.6 assuming Lemma 6.10 and then prove the lemma.

Proof of Theorem 6.6. Let $\epsilon = 1/\max(6, n+1)$. Let us assume that the local share conversion algorithms are correct on Shamir's shares of zero, i.e.,

$$\Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)} \left[\sum_i g_i(s_i) \neq 0 \right] \leq \epsilon \tag{14}$$

As R is a non-trivial relation, there exists an $s' \in \mathbb{F}_p$ such that $(s', 0) \notin R$ and $(s', 1) \in R$. To prove the theorem, it suffices to show that,

$$\Pr_{\mathbf{s}' \leftarrow \mathsf{ShaSh}_{p,n,t}(s')} \left[\sum_i g_i(s'_i) = 0 \right] > \epsilon \ ,$$

where $\mathbf{s}' = (s'_1, \ldots, s'_n)$. Let $\vec{q} = (q_1, q_2, \ldots, q_n)$ be a secret sharing of s' that has the first $t-1$ shares equal to 0. Such a sharing exists. Because the Shamir's secret shares of s' are a coset of the

Shamir's secret shares of 0, the distribution $\left\{\mathbf{s} + \vec{q} : \mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)\right\}$ is identically distributed to $\mathsf{ShaSh}_{p,n,t}(s')$. Hence,

$$\begin{aligned}
\Pr_{\mathbf{s}' \leftarrow \mathsf{ShaSh}_{p,n,t}(s')}\left[\sum_i g_i(s'_i) = 0\right] &= \Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)}\left[\sum_i^n g_i(s_i + q_i) = 0\right] \\
&\geq \Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)}\left[(\forall i \in \{t, \ldots, n\},\ g_i(s_i + q_i) = g_i(s_i)) \wedge \left(\sum_{i=1}^n g_i(s_i) = 0\right)\right] \\
&\geq 1 - \sum_{i=t}^n \Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)}\left[g_i(s_i + q_i) \neq g_i(s_i)\right] \\
&\qquad - \Pr_{\mathbf{s} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)}\left[\sum_i g_i(s_i) \neq 0\right] \\
&> 1 - (n - t + 1) \cdot (2\epsilon) - \epsilon = 1 + (-2n + 2t - 3) \cdot \epsilon \geq 1 + (-2n + n) \cdot \epsilon \\
&\geq \epsilon \ ,
\end{aligned}$$

where the second inequality follows from the union bound, the third inequality follows from Lemma 6.10 and Eq. (14), and the last inequality follows from the fact that $\epsilon \leq \frac{1}{n+1}$. This concludes the proof. □

6.4 Proof of Lemma 6.10

Proving Lemma 6.10 requires some new notions. In particular, the notion of the Gowers' Uniformity Norm.

6.4.1 Gowers' Uniformity Norm

The Gowers' Uniformity Norm was defined by Gowers in [Gow01] to give an alternate Szemerédi's Theorem. This notion has been very influential in additive combinatorics.

Definition 6.11 (Gowers' U^2 Norm)**.** *Let $f : \mathbb{G} \to \mathbb{C}$ be a function. The* Gowers' U^2 Norm *or the* Uniformity Norm *of f, denoted by $\|f\|_{U^2}$ is defined as follows:*

$$\|f\|_{U^2}^4 = \mathop{\mathbb{E}}_{x,a,b \leftarrow \mathbb{G}}\left[f(x) \cdot \overline{f(x-a)} \cdot \overline{f(x-b)} \cdot f(x-a-b)\right] \ ,$$

We will deal only with real-valued functions and hence usually ignore the conjugates in this paper. Higher-order analogues of the Gowers' Norms can be defined analogously, but we do not need them in the paper.

Before recalling properties of the Gowers' norms, we define the non-standard operator $*$ as in [Gre07].[11] Let $f : \mathbb{G} \to \mathbb{C}$ and $g : \mathbb{G} \to \mathbb{C}$ be two functions. The function $f * g : \mathbb{G} \to \mathbb{C}$ is defined by:

$$(f * g)(y) = \mathop{\mathbb{E}}_{x \leftarrow \mathbb{G}}\left[f(x) \cdot \overline{g(x-y)}\right] \ .$$

We recall the following lemma from [Gow01].

[11] As in [Gre07], we do not need to use the standard convolution, which is normally defined as $f \star g : \mathbb{G} \to \mathbb{C}$, $(f \star g)(y) = \mathbb{E}_{x \leftarrow \mathbb{G}}\left[f(x) \cdot g(y-x)\right]$.

Lemma 6.12. *Let $f : \mathbb{G} \to \mathbb{C}$ and $g : \mathbb{G} \to \mathbb{C}$ be two functions. Then we have:*

$$\widehat{f * g} = \widehat{f} \cdot \overline{\widehat{g}} \ .$$

Proof. We have:

$$\begin{aligned}
(\widehat{f * g})(\alpha) &= \mathop{\mathbb{E}}_{x,y\in\mathbb{G}}\left[f(x)\cdot\overline{g(x-y)}\cdot\omega^{\alpha y}\right] \\
&= \mathop{\mathbb{E}}_{x,y\in\mathbb{G}}\left[f(x)\cdot\omega^{\alpha x}\cdot\overline{g(x-y)\cdot\omega^{\alpha(x-y)}}\right] \\
&= \mathop{\mathbb{E}}_{x,z\in\mathbb{G}}\left[f(x)\cdot\omega^{\alpha x}\cdot\overline{g(z)\cdot\omega^{\alpha z}}\right] \\
&= \widehat{f}(\alpha)\cdot\overline{\widehat{g}(\alpha)} \ .
\end{aligned}$$

□

Theorem 6.13 (Properties of the Gowers' Norms)**.** *Let $f : \mathbb{G} \to \mathbb{C}$ be a function.*

(a) (Alternate Definition of U^2.) *The Gowers' Norm of a function is alternately defined as:*

$$\|f\|_{U^2}^4 = \mathop{\mathbb{E}}_{y,y',z,z'\leftarrow\mathbb{G}}\left[f(y+z)\cdot\overline{f(y+z')}\cdot\overline{f(y'+z)}\cdot f(y'+z')\right]$$

(b) (Connection to Fourier Coefficients.)

$$\|f\|_{U^2}^4 = \|f * f\|_2^2 = \left\|\widehat{f * f}\right\|_2^2 = \left\|\widehat{f}\right\|_4^4$$

(c) (Inverse Theorem for U^2 Norm.) *If $\|f\|_{U^2} \geq \delta$ and $\|f\|_2 \leq 1$, then,*

$$\mathrm{bias}(f) = \left\|\widehat{f}\right\|_\infty \geq \delta^2$$

Proof. These properties are proven for example in the proof of Proposition 1.9 of [Gre07] and in [Gow01].

Proof of Theorem 6.13(a). If we write $x = y+z, a = z-z', b = y-y'$, then: $x = y+z, x-a = y+z'$, $x-b = y'+z$, $x-a-b = y'+z'$. Furthermore, if y, y', z, z' are four independent uniform random variables in $\mathbb{G}$, then x, a, b are three independent uniform random variables in $\mathbb{G}$.

Proof of Theorem 6.13(b). The first equality of the proposition comes from:

$$\begin{aligned}
\|f\|_{U^2}^4 &= \mathop{\mathbb{E}}_{y,y',z,z'\leftarrow\mathbb{G}}\left[f(y+z)\cdot\overline{f(y+z')}\cdot\overline{f(y'+z)}\cdot f(y'+z')\right] \\
&= \mathop{\mathbb{E}}_{z,z'\leftarrow\mathbb{G}}\left[\left|\mathop{\mathbb{E}}_{y\leftarrow\mathbb{G}}\left[f(y+z)\cdot\overline{f(y+z')}\right]\right|^2\right] \\
&= \mathop{\mathbb{E}}_{z,z'\leftarrow\mathbb{G}}\left[\left|\mathop{\mathbb{E}}_{y\leftarrow\mathbb{G}}\left[f(y)\cdot\overline{f(y-z+z')}\right]\right|^2\right] \\
&= \mathop{\mathbb{E}}_{z,z'\leftarrow\mathbb{G}}\left[(f * f)(z-z')\right] = \|f * f\|_2^2 \ .
\end{aligned}$$

The second equality of the proposition comes from Theorem 3.9(a). The third equality of the proposition comes from:

$$\left\|\widehat{f*f}\right\|_2^2 = \left\|\widehat{f}\cdot\overline{\widehat{f}}\right\|_2^2 = \mathop{\mathbb{E}}_{x\leftarrow\mathbb{G}}\left[|\widehat{f}(x)|^{2\cdot 2}\right] = \left\|\widehat{f}\right\|_4^4 \ ,$$

where the first equality comes from Lemma 6.12.

Proof of Theorem 6.13(c). We have:

$$\delta^4 \leq \|f\|_{U^2}^4 = \left\|\widehat{f}\right\|_4^4 \leq \left\|\widehat{f}\right\|_\infty^2 \cdot \left\|\widehat{f}\right\|_2^2 = \left\|\widehat{f}\right\|_\infty^2 \cdot \|f\|_2^2 \leq \left\|\widehat{f}\right\|_\infty^2 \ .$$

□

6.4.2 Proof of Lemma 6.10

We now prove Lemma 6.10. This proof is specialized to the case of Shamir's sharing from the work of Green and Tao's [GT10] which proves a more general result.

Proof of Lemma 6.10. As before, it would be convenient for us to define real-valued functions $G_i : \mathbb{F}_p \to \mathbb{R}$ as $G_i(x) = (-1)^{g_i(x)}$. Restated in terms of G_i's, Eq. (13) is equivalent to,

$$\mathop{\mathbb{E}}_{\mathbf{s}\leftarrow\mathsf{ShaSh}_{p,n,t}(0)}\left[\prod_i G_i(s_i)\right] > 1-2\epsilon \ . \qquad (15)$$

Proof Outline.

1. We will consider the linear code generated by $\mathsf{ShaSh}_{p,n,t}(0)$ (a generalized Reed-Solomon Code). We will write the generator matrix of the code in a suitable 'normalized form.'
2. The Cauchy-Schwartz inequality will enable us to upper-bound the expectation in Eq. (15) by the Gowers' norm of the functions G_i's. Hence implying that G_i's have a high Gowers' norm.
3. Finally, invoking the inverse theorem for Gowers' norm will complete the proof.

Claim 6.13.1. *There exists a matrix $\mathbf{M} \in \mathbb{F}_p^{(t-1)\times n}$ such that the linear code generated by $\mathsf{ShaSh}_{p,n,t}(0)$ is generated by $\mathbf{M}$ i.e.,*

$$\mathsf{ShaSh}_{p,n,t}(0) \equiv \left\{\vec{y}\cdot\mathbf{M} : \vec{y} \leftarrow \mathbb{F}_p^{t-1}\right\},$$

and $\mathbf{M}$ has the following form:

$$\mathbf{M} = \begin{array}{c} \begin{array}{ccc} \overbrace{\quad}^{1} & \overbrace{\qquad\qquad}^{t-2} & \overbrace{\qquad\qquad}^{n-t+1} \end{array} \\ \left(\begin{array}{ccccccc} u_1 & 0 & \dots & 0 & * & \dots & * \\ u_2 & * & \dots & * & 0 & \dots & 0 \\ * & * & \dots & * & * & \dots & * \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ * & * & \dots & * & * & \dots & * \end{array}\right) \end{array} , \qquad (16)$$

where u_1, u_2 are non-zero elements of $\mathbb{F}_p$, and each "$*$" is an element in $\mathbb{F}_p$ (not necessarily all equal).

Proof. Let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be the n distinct evaluation points used in Shamir's secret sharing ($0 \notin A$). Let q_1 and q_2 be the following polynomials:

$$q_1(x) = x \cdot (x - \alpha_2) \cdot (x - \alpha_2) \cdots (x - \alpha_{t-1}) \ ,$$
$$q_2(x) = x \cdot (x - \alpha_t) \cdot (x - \alpha_t) \cdots (x - \alpha_n) \ .$$

The number of factors $(x - \alpha_i)$ in q_1 is $n - t + 1 \leq 2t - 3 - t + 1 = t - 2$. Hence both polynomials q_1 and q_2 have degree at most $t - 1$ and the following vectors are valid Shamir's secret sharing of 0:

$$\mathbf{m}_1 = (q_1(\alpha_1),\, q_1(\alpha_2),\, \ldots,\, q_1(\alpha_n)) \ ,$$
$$\mathbf{m}_2 = (q_2(\alpha_1),\, q_2(\alpha_2),\, \ldots,\, q_2(\alpha_n)) \ .$$

Let us write $u_1 = q_1(\alpha_1) \neq 0$ and $u_2 = q_2(\alpha_2) \neq 0$. The two vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ are of the form:

$$\mathbf{m}_1 = (u_1,\, 0,\, \ldots,\, 0,\, *, \ldots,\, *) \ ,$$
$$\mathbf{m}_2 = (u_1,\, *,\, \ldots,\, *,\, 0, \ldots,\, 0) \ .$$

We conclude the proof by remarking that these two vectors are linearly independent and hence can be completed into a full basis of $\mathsf{ShaSh}_{p,n,t}(0)$. □

Remark 6.14. We remark that the above claim requires $n \leq 2t - 3$. If $n > 2t - 3$, the second row would need to have $n - t + 1 > t - 2$ zeros, which is impossible as not all its coefficients are zero: $u_2 \neq 0$. (Recall that a Shamir's secret sharing of 0 has at most $t - 1$ shares equal to 0, unless all the shares are 0.)

Claim 6.14.1 (Cauchy-Schwarz Argument)**.** *Let $G_1, G_2, \ldots, G_n : \mathbb{F}_p \longrightarrow \mathbb{C}$ such that $\|G_i\|_\infty \leq 1$ for all i. Then,*

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)} \left[\prod_i G_i(x_i) \right] \right| \leq \min_i \|G_i\|_{U^2} \ .$$

Proof. We will prove that the left-hand side is at most $\|G_1\|_{U^2}$. The other cases are true by symmetry. Using the matrix $\mathbf{M}$ from Claim 6.13.1, we write the left-hand side as:

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)} \left[\prod_i G_i(x_i) \right] \right| = \left| \mathop{\mathbb{E}}_{\vec{y} \leftarrow \mathbb{F}_p^{t-1}} \left[\prod_i G_i(\langle \vec{m}^{(i)}, \vec{y} \rangle) \right] \right|$$

where $\vec{m}^{(i)}$ is the i-th column of $\mathbf{M}$. We remark that if we write $\vec{x} = \vec{y} \cdot \mathbf{M}$, then $x_i = \langle \vec{m}^{(i)}, \vec{y} \rangle$.

To prove the claim, it is beneficial to separate the variables y_1 and y_2 from the rest. As a shorthand, we omit the dependence on $y_3, \ldots, y_{t-1}$ and write:

$$h(y_1, y_2) = G_1\Big(\langle \vec{m}^{(1)}, \vec{y} \rangle \Big) \ ,$$
$$b_1(y_1) = \prod_{i=2}^{t-1} G_i\Big(\langle \vec{m}^{(i)}, \vec{y} \rangle \Big) \ ,$$
$$b_2(y_2) = \prod_{i=t}^{n} G_i\Big(\langle \vec{m}^{(i)}, \vec{y} \rangle \Big) \ .$$

We indeed remark that $b_1(y_1)$ and $b_2(y_2)$ do not depend on y_2 and y_1 respectively, by definition of $\mathbf{M}$ (see Eq. (16)). Furthermore, we use b to indicate that these functions are *bounded* by 1.

So, our product can be written as follows:

$$\mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathsf{ShaSh}_{p,n,t}(0)}\left[\prod_i G_i(x_i)\right] = \mathop{\mathbb{E}}_{y_3,\dots,y_{t-1}}\left[\mathop{\mathbb{E}}_{y_1,y_2}\left[h(y_1,y_2)\cdot b_1(y_1)\cdot b_2(y_2)\right]\right] \ .$$

We now link this product to the Gowers' norm of the function h via repeated use of Cauchy-Schwarz inequality. Using Cauchy-Schwarz on y_2, we get:

$$\mathop{\mathbb{E}}_{y_1,y_2}\left[h(y_1,y_2)\cdot b_1(y_1)\cdot b_2(y_2)\right] \le \left(\mathop{\mathbb{E}}_{y_2}\left[\mathop{\mathbb{E}}_{y_1}\left[h(y_1,y_2)\cdot b_1(y_1)\right]^2\right]\right)^{\frac{1}{2}}\left(\mathop{\mathbb{E}}_{y_2}\left[b_2(y_2)^2\right]\right)^{\frac{1}{2}}$$

Boundedness of b_2 implies that $\mathbb{E}_{y_2}\left[b_2(y_2)^2\right] \le 1$. Rearranging the terms, we get that,

$$\le \left(\mathop{\mathbb{E}}_{y_2,y_1,y_1'}\left[h(y_1,y_2)\cdot h(y_1',y_2)\cdot b_1(y_1)\cdot b_1(y_1')\right]\right)^{\frac{1}{2}}$$

Applying Cauchy-Schwarz on y_1, y_1' along with boundedness, we get that,

$$\begin{aligned}
&\le \left(\mathop{\mathbb{E}}_{y_1,y_1'}\left[\mathop{\mathbb{E}}_{y_2}\left[h(y_1,y_2)\cdot h(y_1',y_2)\right]^2\right]\right)^{\frac{1}{4}}\left(\mathop{\mathbb{E}}_{y_1,y_1'}\left[b_1(y_1)^2\cdot b_1(y_1')^2\right]\right)^{\frac{1}{4}}\\
&\le \left(\mathop{\mathbb{E}}_{y_1,y_1'}\left[\mathop{\mathbb{E}}_{y_2}\left[h(y_1,y_2)\cdot h(y_1',y_2)\right]^2\right]\right)^{\frac{1}{4}}\cdot 1\\
&\le \left(\mathop{\mathbb{E}}_{y_1,y_1',y_2,y_2'}\left[h(y_1,y_2)\cdot h(y_1',y_2)\cdot h(y_1,y_2')\cdot h(y_1',y_2')\right]\right)^{\frac{1}{4}}\\
&= \|G_1\|_{U^2} \ .
\end{aligned}$$

where the last equality follows from the fact that h is real-valued, Theorem 6.13(a) and that G_1 and h are related to each other by a linear change of variables. Indeed, for every fixed $y_3,\dots,y_{t-1}$, it holds that $h(y_1,y_2) = G_1\left(u_1y_1 + u_2y_2 + \sum_{i=3}^{t-1} M_{i,1}y_i\right)$, and $u_1, u_2 \neq 0$. Hence,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{y_1,y_1',y_2,y_2'}\left[h(y_1,y_2)h(y_1',y_2)\cdot h(y_1,y_2')\cdot h(y_1',y_2')\right] = \\
&\qquad\qquad \mathop{\mathbb{E}}_{y_1,y_1',y_2,y_2'}\left[G_1(y_1+y_2)\cdot G_1(y_1'+y_2)\cdot G_1(y_1+y_2')\cdot G_1(y_1'+y_2')\right] \ .
\end{aligned}$$

This concludes the proof of Claim 6.14.1. □

We can now prove Lemma 6.10. Claim 6.14.1 and Eq. (15) imply that $\|G_1\|_{U^2} > 1-\epsilon$. We need to relate the Gowers' Norm to the Fourier bias. Using Theorem 6.13(c), we get that,

$$\mathrm{bias}(G_1) \ge (1-\epsilon)^2 > 1-2\epsilon.$$

This implies that $\left\|\widehat{G_1}\right\|_\infty > 1-2\epsilon \ge 2/3$, as $\epsilon \le 1/6$. Lemma 6.7 implies that for any $\alpha \neq 0$, $\left|\widehat{G_1}(\alpha)\right| \le 2/3$. Hence $\left|\widehat{G_1}(0)\right| > 1-2\epsilon$. Combining this with Lemma 6.8 shows that,

$$\Pr_x[G_1(x) = G_1(x+a)] > 1-2\epsilon \ .$$

This completes the proof of Lemma 6.10 as $G_1(x) = G_1(x+a) \iff g_1(x) = g_1(x+a)$. □

References

[AFL^{+}16] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In *CCS*, 2016.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, 2009.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer. Generalized privacy amplification. *IEEE Trans. Information Theory*, 41(6):1915–1923, 1995.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[BCLR08] Michael Ben-Or, Don Coppersmith, Michael Luby, and Ronitt Rubinfeld. Non-Abelian Homomorphism Testing, and Distributions close to their Self-Convolutions. *Random Struct. Algorithms*, 2008.

[BDIR18] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 531–561. Springer, 2018.

[BDL14] Nir Bitansky, Dana Dachman-Soled, and Huijia Lin. Leakage-tolerant computation with input-independent preprocessing. In *CRYPTO*, 2014.

[Bea91] Donald Beaver. Efficient multiparty protocols using circuit randomization. In *CRYPTO*, 1991.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the Circuit Size Barrier for Secure Computation under DDH. In *CRYPTO*, 2016.

[BGK11] Elette Boyle, Shafi Goldwasser, and Yael Tauman Kalai. Leakage-resilient coin tossing. In *Distributed Computing*, 2011.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract). In *STOC*, 1988.

[BIKO12] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Ilan Orlov. Share Conversion and Private Information Retrieval. In *CCC*, 2012.

[BIVW16] Andrej Bogdanov, Yuval Ishai, Emanuele Viola, and Christopher Williamson. Bounded indistinguishability and the complexity of recovering secrets. In *CRYPTO 2016, Part III*, pages 593–618, 2016.

[BKS19] Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. *IACR Cryptology ePrint Archive*, 2019:129, 2019. To appear in Eurocrypt 2019.

[Bla79] G.R. Blakley. Safeguarding cryptographic keys. In *AFIPS National Computer Conference*, 1979.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *J. Comput. Syst. Sci.*, 1993.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, 1988.

[CDH$^+$00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 453–469. Springer, 2000.

[CDI05] Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation. In *TCC 2005*, 2005.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *EUROCRYPT*, 2014.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*, pages 121–137, 2010.

[DF12] Stefan Dziembowski and Sebastian Faust. Leakage-resilient circuits without computational assumptions. In *TCC 2012*, pages 230–247, 2012.

[DHRW16] Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In *CRYPTO 2016, Part III*, pages 93–122, 2016.

[DLZ15] Dana Dachman-Soled, Feng-Hao Liu, and Hong-Sheng Zhou. Leakage-resilient circuits revisited - optimal number of computing components without leak-free hardware. In *EUROCRYPT*, 2015.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, 2007.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, 2008.

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO*, 2012.

[DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 301–324. Springer, 2001.

[FGJI17] Nelly Fazio, Rosario Gennaro, Tahereh Jafarikhah, and William E. Skeith III. Homomorphic secret sharing from paillier encryption. In *ProvSec 2017*, pages 381–399, 2017.

[FRR$^+$10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In *EUROCRYPT*, 2010.

[GIM$^+$16] Vipul Goyal, Yuval Ishai, Hemanta K. Maji, Amit Sahai, and Alexander A. Sherstov. Bounded-Communication Leakage Resilience via Parity-Resilient Circuits. In *FOCS*, 2016.

[GIW17] Daniel Genkin, Yuval Ishai, and Mor Weiss. How to construct a leakage-resilient (stateless) trusted party. In *TCC*, 2017.

[GK18] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *STOC*, 2018.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *STOC 1987*, 1987.

[Gow01] William T Gowers. A new proof of Szemerédi's theorem. *Geometric and Functional Analysis*, 2001.

[GR15] Shafi Goldwasser and Guy N. Rothblum. How to compute in the presence of leakage. *SICOMP*, 2015.

[Gre07] Ben Green. Montréal notes on Quadratic Fourier Analysis. *Additive combinatorics*, 2007.

[GT10] Benjamin Green and Terence Tao. Linear Equations in Primes. *Annals of Mathematics*, 2010.

[GW10] William T Gowers and Julia Wolf. The True Complexity of a System of Linear Equations. *Proceedings of the London Mathematical Society*, 2010.

[GW11a] William T Gowers and Julia Wolf. Linear Forms and Higher-Degree Uniformity for Functions On $\mathbb{F}_n^p$. *Geometric and Functional Analysis*, 2011.

[GW11b] William T Gowers and Julia Wolf. Linear Forms and Quadratic Uniformity for Functions on $\mathbb{F}_n^p$. *Mathematika*, 2011.

[GW17] Venkatesan Guruswami and Mary Wootters. Repairing reed-solomon codes. *IEEE Trans. Information Theory*, 2017.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, 2003.

[KGG$^+$18] Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. *ArXiv e-prints*, January 2018.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, 1999.

[KMS18] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:200, 2018.

[Koc96] Paul C. Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In *CRYPTO*, 1996.

[KOS16] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *CCS*, 2016.

[KP10] Eike Kiltz and Krzysztof Pietrzak. Leakage Resilient ElGamal Encryption. In *ASIACRYPT*, 2010.

[LSG⁺18] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown. *ArXiv e-prints*, 2018.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, 2004.

[NS19] Jesper Buus Nielsen and Mark Simkin. Lower Bounds for Leakage-Resilient Secret Sharing. Cryptology ePrint Archive, Report 2019/181, 2019. https://eprint.iacr.org/2019/181.

[Riv97] Ronald L Rivest. All-or-nothing encryption and the package transform. In *International Workshop on Fast Software Encryption*, pages 210–218. Springer, 1997.

[Rot12] Guy N. Rothblum. How to compute under ${\cal{AC}}^{\sf0}$ leakage without secure hardware. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 552–569, 2012.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 1979.

[SV18] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. *IACR Cryptology ePrint Archive*, 2018:1154, 2018.

[TV06] Terence Tao and Van H Vu. *Additive combinatorics*. Cambridge University Press, 2006.

[Yao86] Andrew Chi-Chih Yao. How to Generate and Exchange Secrets (Extended Abstract). In *FOCS*, 1986.

A Proofs of Useful Bounds

In this section, we prove Proposition 4.14 and the following related bound.

Proposition A.1. *Let $m \geq 1$ and $p \geq 2$ be two integers. Let $c'_m = \frac{2^m \sin(\pi/2^m + \pi/2^{4m})}{p \sin(\pi/p)}$. We have:*

$$\log c'_m \leq -\frac{1}{2^{2m+2}} + \frac{4}{p^2} \quad .$$

To prove these two propositions, we start by studying the function $\eta : \mathbb{R}_{>0} \to \mathbb{R}$ defined by:

$$\eta(x) = \frac{x}{\pi} \sin \frac{\pi}{x} \quad .$$

Claim A.1.1. *For any $x \geq 1$, we have:*

$$\log \eta(x) \leq -\frac{1}{2x^2} \quad .$$

Proof. We have:

$$\eta(x) \leq \frac{x}{\pi}\left(\frac{\pi}{x} - \frac{\pi^3}{6x^3} + \frac{\pi^5}{5!x^5}\right) \leq 1 - \frac{\pi^2}{6x^2} + \frac{\pi^4}{5!x^4} \leq 1 - \frac{1}{2x^2} \quad .$$

We conclude using concavity of $u \mapsto \log(1+u)$, namely the fact that it implies that $\log(1+u) \leq u$. □

Claim A.1.2. *For any $y \geq 2$, we have:*

$$\log \frac{1}{\eta(y)} \leq \frac{4}{y^2} \quad .$$

Proof. We have:

$$\eta(y) \geq \frac{y}{\pi}\left(\frac{\pi}{y} - \frac{\pi^3}{6y^3}\right) \geq 1 - \frac{\pi^2}{6y^2} \quad .$$

Then:

$$\frac{1}{\eta(y)} \leq \frac{1}{1 - \frac{\pi^2}{6y^2}} \leq 1 + \frac{\pi^2}{3y^2}$$

where the last inequality comes from the convexity of $u \mapsto \frac{1}{1-u}$ and the fact that it implies that $\frac{1}{1-u} \leq 4u - 2(u - \frac{1}{2}) = 1 + 2u$ for $0 \leq u \leq 1/2$ (the curve is below its chord). We conclude using again the concavity of $u \mapsto \log(1+u)$. □

We can now prove Propositions 4.14 and A.1.

Proof of Proposition 4.14. Using Claims A.1.1 and A.1.2, we have:

$$\log c_m = \log \frac{\eta(2^m)}{\eta(p)} = \log \eta(2^m) + \log \frac{1}{\eta(p)} \leq -\frac{1}{2^{2m+1}} + \frac{4}{p^2} \quad .$$

This concludes the proof. □

Proof of Proposition A.1. Let us start with the case $m = 1$. We conclude as follows:

$$\log c_1' = \log \frac{2^m/\pi \cdot \sin(\pi/2^m + \pi/2^{4m})}{\eta(p)} = \log(2^m/\pi \cdot \sin(\pi/2^m + \pi/2^{4m})) + \log \frac{1}{\eta(p)} \leq -\frac{1}{2^{2m+1}} + \frac{4}{p^2} \quad ,$$

where the last inequality comes from Claim A.1.2 and the fact that for $m = 1$, $\log(2^m/\pi \cdot \sin(\pi/2^m + \pi/2^{4m})) \approx -0.47$.

Let us now suppose that $m \geq 2$. Let us define:

$$a = \frac{1}{\frac{1}{2^m} + \frac{1}{2^{4m}}} = \frac{2^{4m}}{2^{3m}+1} \leq 2^m \quad .$$

We have:

$$\log c'_m = \log\left(\frac{2^m}{a} \cdot \frac{\eta(a)}{\eta(p)}\right) = \log\left(1 + \frac{1}{2^{3m}}\right) + \log \eta(a) - \log \eta(p) \leq \frac{1}{2^{3m}} - \frac{1}{2a^2} + \frac{4}{p^2} \ ,$$

where the inequality comes from the concavity of $u \mapsto \log(1+u)$ and Claims A.1.1 and A.1.2. We conclude by remarking that:

$$\frac{1}{2^{3m}} - \frac{1}{2a^2} \leq \frac{1}{2^{3m}} - \frac{1}{2 \cdot 2^{2m}} \leq \frac{1}{2^{2m+2}} - \frac{1}{2^{2m+1}} = \frac{1}{2^{2m+2}} \ ,$$

where the second inequality comes from the fact that $2m + 2 \leq 3m$ when $m \geq 2$. □