FastShare: Scalable Secret Sharing by Leveraging Locality

Swanand Kadhe, Nived Rajaraman, and Kannan Ramchandran

***Abstract*—Shamir's secret sharing scheme is widely used in many cryptographic protocols such as secure multi-party computation. It allows a secret to be distributed to n parties such that any t parties learn no information about the secret, whereas any $t+1$ parties can recover the secret. However, the *worst-case* recovery guarantees come at the price of $O(n^2)$ computation cost. This quadratic cost limits its scalability to applications involving only a small number of participants.**

This paper presents a framework, called FastShare, for designing secret sharing schemes that ensures *worst-case* security guarantees at a lower computation cost by relaxing the recovery constraint from worst-case to *average-case* in a statistical sense. In particular, FastShare considers the setup where each party takes part in the recovery process with some probability ρ, independently of others. The core idea of FastShare is to construct a 'signal' using the secret and random masks by inserting zeros at judiciously chosen locations, and take its finite field fast Fourier transform (FFT) to generate the shares. We present a scheme designed using FastShare where the judicious zero placement ensures that the shares form a codeword of a locally recoverable code. The locality property along with the FFT allows us to recover the secret with $O(n \log n)$ computational complexity, from a 'random' subset of shares of large enough size. We analyze its security and recovery thresholds, and characterize a trade-off between ρ and the probability of successfully recovering the secret. Further, we carry out numerical simulations to demonstrate the applicability of the proposed scheme for a wide range of values of n.

I. Introduction

Secret sharing, introduced by Shamir [1] and Blakeley [2], is a fundamental primitive that forms the central building block of many cryptograpic protocols including secure multi-party computation [3], threshold cryptography [4], and secure cloud storage [5]. The canonical setup for a secret sharing scheme consists of a source (called a *dealer*) with a secret (from some finite alphabet), a set of n parties, and two collections $\mathcal{A}$ and $\mathcal{B}$ of subsets of parties, called the access-list and block-list, respectively. A secret-sharing scheme allows the dealer to generate n shares from the secret and distribute them to n parties such that: (i) any subset in $\mathcal{A}$ can reconstruct the secret from its shares, and (ii) any subset in $\mathcal{B}$ cannot learn any information about the secret. (See [6] for a survey.)

Shamir's scheme [1] and its variants [7]–[9] are the most widely used class of secret sharing schemes, which realize access- and block-lists with a threshold structure. In particular, for any given n and $t < n$, Shamir's scheme ensures that any t parties cannot learn any information about the secret, whereas any $t+1$ parties can completely recover the secret. However, this *worst-case* recovery guarantee of Shamir's scheme comes at the price of $O(t^2)$ computation cost. This can be a severe bottleneck when t is some constant fraction of n and n is large, as in emerging applications like privacy-preserving federated learning [10], secure network statistics [11], and multi-cloud secure storage [12].

In this paper, we develop a framework for secret sharing that allows one to construct scalable schemes with low computation cost. Our design philosophy is guided by the following key observation. In may applications of secret sharing, such as secure smart-meter aggregation [13], secure aggregation for federated learning [10], secure aggregation for network statistics [11], and secure cloud storage [5], the recovery needs to be performed in a dynamic environment where parties fail or drop out due to natural reasons, e.g., wireless outages. In such applications, it is often sufficient to recover the secret from a *random* set of shares of a certain (large enough) *average* size. At the same time, it is critical to ensure *worst-case* guarantees for security. This is because privacy breaches are often inflicted by malicious participants, who can violate any strong assumptions such as only random sets of parties may collude. Leveraging this observation, FastShare cuts down the computation cost of secret recovery from $O(n^2)$ to $O(n \log n)$ by relaxing *worst-case* requirements to *average-case* requirements, while ensuring *worst-case* security guarantees.

To highlight this point, let us consider the following typical application scenario for secret sharing (see, e.g., [5]). A dealer wants to store a secret file over 10000 servers distributed over multiple cloud networks with a 30% security threshold. The dealer can achieve this by using Shamir's scheme with $t = 2999$. This allows them to recover the secret file from *any* 3000 servers, requiring $O(10^6)$ computations, while being secure if *any* set of fewer than 3000 servers collude. In contrast, when servers have 60% availability (meaning that, for each server, there is a 60% chance that it is online during recovery), FastShare can recover the secret with 99.99% success at $O(10^4)$ complexity, while ensuring the same security guarantee as Shamir's scheme. In fact, FastShare can achieve a trade-off between the probability of successfully recovering the secret and the probability that parties participate during the recovery process (see Fig. 1 on the next page).

It is worth noting that the computation cost of recovering the secret in Shamir's scheme stems from Lagrange's interpolation. Typical algorithms for Lagrange's interpolation take $O(t^2)$ complexity for a polynomial of degree t (see, e.g., [14]). Several practical applications of secret sharing consider the

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA. (E-mails: {swanand.kadhe, nived.rajaraman, kannar}@berkeley.edu)

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518282

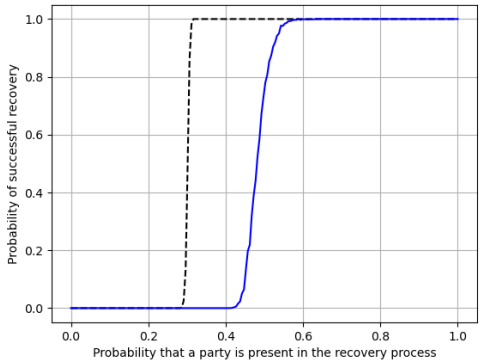

Fig. 1. Dotted black line shows Shamir's scheme and solid blue line show FastShare-LRC for $n = 10000$ and the security threshold of 30%. When $\rho = 0.6$, FastShare-LRC can successfully recover the secret with probability at least 99.99%.

regime of $t = O(n)$, for which the complexity of recovering the secret scales quadratically in the number of parties. There exist asymptotically fast algorithms for polynomial interpolation that allow the recovery in $O(n \log^2 n \log\log n)$ time (see, e.g., [14]–[18]). However, for practical values of n, quadratic time algorithms are faster than the theoretically asymptotically fast algorithms for polynomial interpolation due to moderate-sized constants hidden by the big-Oh notation (see, e.g., [19]). This heavy computation cost limits the scalability of Shamir's scheme to only a small number of participants.

Our Contributions: We develop a novel framework FastShare for designing computationally efficient secret sharing schemes which provide a *worst-case* security guarantee and *average-case* recovery guarantee. In particular, FastShare enables one to design schemes with the following properties: (i) Information-theoretic perfect security of the secret is guaranteed against any set of less than τ fraction of shares, i.e., sets of shares of size less than τn cannot learn any information about the secret. (ii) Suppose each party participates in the recovery process with probability ρ, $\tau \leq \rho \leq 1$, independently of the other parties. Then, FastShare achieves a trade-off between ρ and the probability of successfully recovering the secret.

The core idea of FastShare is to first construct a *signal* using the secret and an appropriate number of random masks (which depends on the security threshold) by inserting zeros at judiciously chosen locations, and then take its finite field fast Fourier transform (FFT) to generate the shares. We present a scheme designed using FastShare, where the shares form a codeword of a locally recoverable code (LRC) (see Sec. II-C for details on LRCs). In particular, we ensure these local parity-check relations by essentially designing the *spectrum* of the shares – our judicious zero placement induces local parity-checks due to the aliasing property of the Fourier transform. This spectral interpretation used in the FastShare framework is inspired by the seminal work of Blahut [20] (which takes a spectral view of algebraic codes) and [21], [22] (which take a spectral view of product and sparse-graph codes).

For this scheme, referred to as FastShare-LRC, we characterize the average-case recovery threshold ρ as a function of n and τ, and show that for any $\rho > \tau$, FastShare-LRC can guarantee recovery with probability $1-O(1/n)$ for sufficiently large n. Moreover, via numerical simulations, we demonstrate that, for a wide range of values of n, FastShare-LRC can guarantee recovery at values of ρ that are useful in practical applications.

We build on our prior work [23], where we proposed a variant of the FastShare framework to design efficient 'multi-secret' sharing schemes for the application of privacy-preserving federated learning. Specifically, the schemes proposed in [23] are designed to share k secrets to n parties for $k = O(n)$, whereas our focus is on sharing a single secret.

Related Work: Several works have considered designing efficient secret sharing schemes where encoding and decoding allow only XOR-operations on binary strings, see e.g., [24]–[26]. However, the number of operations still remain quadratic in the number of shares. Our goal is to reduce the number of computations in encoding and recovery. Security for locally recoverable codes is considered in [27]. Communication efficient secret sharing is considered in [28] (see also [29], [30] and references therein), where the goal is to reduce communication required to recover the secret as a function of the number of parties participating in the process.

II. Preliminaries

Notation: For a prime power q, we will denote the finite field with q elements by $\mathbb{F}_q$. For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$ and $[n^-] = \{0, 1, \ldots, n-1\}$. For a set A, let 2^A denote its power set. For a matrix G, let $\langle G \rangle$ denote the column-span of G. For a vector $C \in \mathbb{F}_q^n$ and a set $A \subseteq [n]$, let C_A denote the projection of C onto the coordinates in A. For random variables $\mathbf{X}$ and $\mathbf{Y}$, let $I(\mathbf{X}; \mathbf{Y})$ denote the mutual information between $\mathbf{X}$ and $\mathbf{Y}$ (see, e.g., [31]).

A. Secret Sharing

In a secret sharing scheme, a dealer takes a secret $S \in \mathbb{F}_q$, generates n shares $C = (C_1, C_2, \ldots, C_n) \in \mathbb{F}_q^n$, and distributes the shares to n parties with one share to each party. The shares are generated such that they satisfy a given access-list $\mathcal{A} \subseteq 2^{[n]}$, which specifies recovery constraints, and a block-list $\mathcal{B} \subseteq 2^{[n]}$, which specifies security constraints. In particular, for any set $A \in \mathcal{A}$, the dealer can recover the secret from the shares C_A; whereas, for any set $B \in \mathcal{B}$, the shares C_B do not contain any information about the secret. A secret sharing scheme is called a *ramp threshold* scheme, if for some $t < r \leq n$, we have $\mathcal{A} = \{A \subseteq [n] : |A| \geq r\}$ and $\mathcal{B} = \{B \subset [n] : |B| \leq t\}$. In other words, in a ramp threshold secret sharing scheme, any r parties can recover the secret, while any t parties cannot learn anything about the secret. In this case, we say that the scheme has a *worst-case* recovery threshold r and a *worst-case* security threshold t.

Shamir's Scheme: Let $q \geq n+1$ be a prime power, and $\alpha_1, \alpha_2, \ldots, \alpha_n$ be n distinct non-zero elements of $\mathbb{F}_q$. To share a secret $S \in \mathbb{F}_q$, the dealer first chooses t random elements K_1, K_2, …, K_t from $\mathbb{F}_q$, independently with uniform distribution. We refer to these as random masks. Then, using these random masks together with the secret, the dealer defines a polynomial

$P(x) = S + \sum_{i=1}^{t} K_i x^i$. The share of party j, $j \in [n]$, is computed as $C_j = P(\alpha_j)$. Using the Lagrange's interpolation theorem, one can show that Shamir's scheme has a worst-case security threshold of t and a worst-case recovery threshold of $t+1$ (see, e.g., [6, Section 3.1]).

B. Worst-Case Security and Average-Case Recovery Model

We are interested in designing ramp secret sharing schemes with an *average-case* recovery threshold and *worst-case* security threshold. In particular, our goal is to recover the secret from a *random* set of shares of a large enough *average* size, when each party fails with a certain probability, independently of others. Formally, we consider the following setup[1].

Definition 1 ((τ, ρ) Ramp Secret Sharing)**.** *Let $\mathbb{F}_q$ be a finite field for a prime power q, n be a positive integer, and τ, ρ be positive constants such that $0 < \tau \leq \rho < 1$. A secret sharing scheme consists of a stochastic encoder* $\textsc{Enc} : \mathbb{F}_q \to \mathbb{F}_q^n$ *and a decoder* $\textsc{Dec} : (\mathbb{F}_q \cup \{\perp\})^n \to \mathbb{F}_q \cup \{\perp\}$*. For every $S \in \mathbb{F}_q$, the stochastic encoder* Enc *outputs a vector of n shares $\mathbf{C} = (\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n) \in \mathbb{F}_q^n$, such that the following conditions are satisfied:*

1) *Security: any set of less than τ fraction of shares does not leak any information about the secret. Specifically, for every $\mathcal{P} \subset [n]$, $|\mathcal{P}| < \tau n$, it holds that*
$$I(\mathbf{S}; \mathbf{C}_{\mathcal{P}}) = 0, \tag{1}$$
where $\mathbf{S}$ denotes the random variable for the secret.
2) *Recovery: It is possible to recover the secret with high probability from a set of shares where each share is selected independently with probability ρ. Specifically, for any $S \in \mathbb{F}_q$ and $\mathbf{C} = \textsc{Enc}(S)$, define the vector $\tilde{\mathbf{C}} \in (\mathbb{F}_q \cup \{\perp\})^n$ as*
$$\tilde{\mathbf{C}}_i = \begin{cases} \mathbf{C}_i & \text{with probability } \rho, \\ \perp & \text{with probability } 1-\rho. \end{cases} \tag{2}$$
Then, it holds that
$$\Pr\left(\textsc{Dec}(\tilde{\mathbf{C}}) = S\right) \geq 1 - O\left(\frac{1}{poly(N)}\right), \tag{3}$$
where the probability is computed over the randomness of generating $\tilde{\mathbf{C}}$ given $\mathbf{C}$.

We refer to τ as the worst-case security threshold and ρ as the average-case recovery threshold of the scheme.

C. Locally Recoverable Codes (LRCs)

Here, we briefly review a class of erasure codes called Locally Recoverable Codes (LRCs) (see [32]–[34], and references therein). Consider an (n, k) code $\mathcal{C}$ with block-length n and dimension k. Consider an ℓ that divides n. The code $\mathcal{C}$ is said to be an LRC with (ℓ, m) locality [33] if the n coordinates can be partitioned into n/ℓ subsets of cardinality ℓ such that the coordinates in each subset form a maximum distance separable (MDS) code of length ℓ and dimension m. We denote such a code as an (n, k, ℓ, m) LRC. An efficient construction of such LRCs is presented in [34, Section V-C].

[1]For simplicity, we assume that the secret as well as each share is an element of a finite field $\mathbb{F}_q$. Indeed, Shamir's scheme as well as our proposed framework operates in this regime. In general, a secret can belong to $\mathbb{F}_{q^l}$ and shares can belong to $\mathbb{F}_{q^m}$, where $m \geq l/(r-t)$, see e.g., [6].

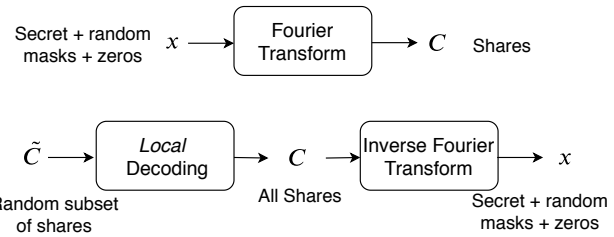

Fig. 2. FastShare framework generates shares by first constructing a 'signal', and then taking the finite field Fast Fourier Transform (FFT). The recovery from a 'random' subset of shares involves obtaining the missing shares via 'local decoding', and then taking the inverse finite field FFT.

III. FastShare Framework

In this section, we present a framework FastShare that allows one to design computationally efficient secret sharing schemes.

To introduce FastShare, we briefly review the basics of the finite field Fourier transform (for details, see, e.g., [35]). Let n be a positive integer such that n divides $(q-1)$ and ω be a primitive n-th root of unity in $\mathbb{F}_q$. The finite field Fourier transform of a signal $x = [x_0\, x_1\, \ldots x_{n-1}]$ is defined as
$$X_j = \sum_{i=0}^{n-1} \omega^{ij} x_i, \quad j = 0, 1, \ldots, n-1. \tag{4}$$
The inverse finite field Fourier transform is given by
$$x_i = \frac{1}{n} \sum_{j=0}^{n-1} \omega^{-ij} X_j, \quad i = 0, 1, \ldots, n-1, \tag{5}$$
where $1/n$ denotes the reciprocal of the sum of n ones in the field $\mathbb{F}_q$. Fast Fourier transform (FFT) algorithms over the complex field can be readily adapted to the case of finite field Fourier transform [35]. These fast algorithms allow one to compute the transform and the inverse in $O(n \log n)$ time.

At a high level, FastShare framework consists of two stages. First, it constructs a length-n 'signal' by taking the secret and (an appropriate number of) random masks, and inserting zeros at judiciously chosen locations. Second, it takes the fast Fourier transform of the signal to generate the shares (see Fig. 2). The shares can be considered as the 'spectrum' of the signal constructed in the first stage. While constructing the signal, the zeros are placed at specific locations such that they induce *local* parity-check constraints on the shares. At the recovery phase, these local parity-check constraints allow us to recover the missing shares in a computationally efficient manner. Then, we can simply take the inverse Fourier transform of the spectrum to obtain the signal, and read off the secret from the appropriate index.

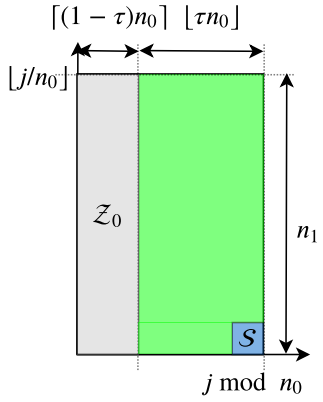

Fig. 3. Indices $j \in [n^-]$ are assigned to a 2-dimensional grid using the mapping $f(j) = (j \mod n_0, \lfloor j/n_0 \rfloor)$. Sets $\mathcal{Z}_0$ and $\mathcal{S}$ are defined in (6), (7), respectively. Set $\mathcal{S}$ is used for the secret, set $\mathcal{Z}_0$ is used for zeros, and the remaining locations are used for random masks (see (8)).

A. FastShare Scheme Inducing an LRC (FastShare-LRC)

Here, we describe a specific signal construction which induces the spectral components (i.e., shares) to be codewords of a locally recoverable code (LRC). We refer to this scheme as FastShare-LRC.

Generate Shares: A dealer is given a composite number n and a prime power q such that n divides $(q-1)$, a constant $0 < \tau < 1$, and a secret $S \in \mathbb{F}_q$. Define $t = \lfloor \tau n \rfloor - 1$. Choose positive integers $n_0 \geq 2$ and n_1, such that $n = n_0 n_1$. Let $f : [n^-] \to [n_0^-] \times [n_1^-]$ be a linear bijection given as $f(j) = (j \mod n_0, \lfloor j/n_0 \rfloor)$. Note that, given $f(j) = (a, b)$, we have $f^{-1}((a,b)) = bn_0 + a \ (= j)$. Define the following sets (see Fig. 3):

$$\mathcal{Z}_0 = \{(a,b) : 0 \leq a \leq \lceil (1-\tau)n_0 \rceil - 1\}, \tag{6}$$

$$\mathcal{S} = (n_0 - 1, 0). \tag{7}$$

The dealer first constructs a length-n signal by placing zeros at indices corresponding to $\mathcal{Z}_0$, placing the secret at the index corresponding to $\mathcal{S}$, and placing a uniform random mask from $\mathbb{F}_q$ (chosen independently of other masks and the secret) at each of the remaining indices.

To formally describe the signal construction, choose an arbitrary bijection $\sigma : [t] \to [n^-] \setminus \{f^{-1}(u) : u \in \mathcal{S} \cup \mathcal{Z}_0\}$. Choose t random masks $K_1, K_2, \ldots, K_t$, from $\mathbb{F}_q$, independently with uniform distribution. For $j \in [n^-]$, define the signal x as:

$$x_j = \begin{cases} S & \text{if } f(j) \in \mathcal{S}, \\ 0 & \text{if } f(j) \in \mathcal{Z}_0, \\ K_{\sigma(j)} & \text{otherwise.} \end{cases} \tag{8}$$

Next, the dealer generates the shares by taking the (fast) Fourier transform of the signal x, i.e., for $j \in [n^-]$, $C_j = \sum_{i=0}^{n-1} \omega^{ij} x_i$, where ω is a primitive n-th root of unity in $\mathbb{F}_q$.

Remark 1. *Consider the case of Shamir's scheme when the n distinct non-zero evaluation points are chosen as the powers of a primitive n-th root of unity in $\mathbb{F}_q$, i.e., $\alpha_i = \omega^{i-1}$, $i \in [n]$. Let t $(< n)$ be the security threshold, S be the secret, and $K_1, \ldots, K_t$ be t random masks. It is easy to see that the shares $C = (C_1, C_2, \cdots, C_n)$ generated by Shamir's scheme is a finite field Fourier transform of the signal $x = (S, K_1, \cdots, K_t, 0, \cdots, 0)$. Observe that, in Shamir's scheme, all the zeros in the signal are grouped together. In contrast, FastShare-LRC judiciously disperses the zeros throughout the signal.*

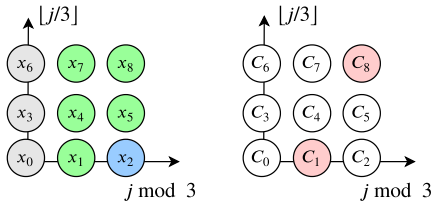

Fig. 4. Example with $n = 9$ and $\tau = 2/3$. Choose $n_0 = n_1 = 3$, and represent $0 \leq j \leq 8$ in a 2D-grid as $(j \mod 3, \lfloor j/3 \rfloor)$. Place zeros at gray locations (i.e., $\mathcal{Z}_0 = \{(0,0), (0,3), (0,6)\}$), and the secret at the blue location (i.e., $\mathcal{S} = \{(2,0)\}$). We consider $\mathbb{F}_{19}$, in which a primitive 9-th root of unity exists. The careful zero placement induces parity checks on the shares in each row due to *aliasing* property of the Fourier transform, i.e., $[C_0 + C_1 + C_2,\ C_3 + C_4 + C_5,\ C_6 + C_7 + C_0] = [0, 0, 0]$. Therefore, any one missing share in a row can be recovered using the parity-check structure. E.g., dropped out red shares can be recovered by *locally decoding* each row.

In the following lemma, we show that the judiciously placed zeros in the signal ensure that the shares form a codeword of a locally recoverable code. The proof essentially follows from the subsampling and aliasing properties of the Fourier transform, and is similar to the proof of [23, Lemma 3.2].

Lemma 1. *For any $S \in \mathbb{F}_q$ and $(K_1, \ldots, K_t) \in \mathbb{F}_q^t$, let $C = (C_1, C_2, \ldots, C_n)$ denote the shares generated by FastShare-LRC. Then, C is a codeword of an $(n, \lfloor \tau n \rfloor, n_0, \lfloor \tau n_0 \rfloor)$ LRC. More specifically, for every $j \in [n_1^-]$, the vector $(C_{jn_0}, C_{jn_0+1}, \cdots, C_{(j+1)n_0-1})$ is a codeword of an $(n_0, \lfloor \tau n_0 \rfloor)$ Reed-Solomon code.*

The above lemma shows that when shares are represented in a 2D-grid using the bijection $f(\cdot)$, each row is a codeword of an $(n_0, \lfloor \tau n_0 \rfloor)$ Reed-Solomon code. These local parity-check constraints on the shares make it possible to *locally* decode missing shares in each row. Note that a codeword of an $(n_0, \lfloor \tau n_0 \rfloor)$ Reed-Solomon code can be viewed as a vector of n_0 evaluations of a polynomial of degree at most $\lfloor \tau n_0 \rfloor - 1$. Therefore, up to $1 - \tau$ fraction of erasures per row can be recovered via polynomial interpolation.

Recover the secret: Given a vector $\tilde{C} \in (\mathbb{F}_q \cup \{\perp\})^n$, the dealer first decodes the Reed-Solomon code corresponding to each row to obtain as many missing shares as possible. If this *local decoding* fails for any row, the dealer outputs $\perp$, and declares failure. Otherwise, it has recovered the vector C. Then, it takes the inverse (finite field) fast Fourier transform of C (using ω) to obtain the signal x. Finally, it outputs the coordinate of x indexed by $\mathcal{S}$ (in the 2D-grid representation) as the secret, i.e., $S = x_{f^{-1}(\mathcal{S})}$. We present a toy example for $n = 9$, $\tau = 2/3$, $n_0 = n_1 = 3$ in Fig. 4.

It is worth noting that FastShare-LRC is one specific scheme realized using the FastShare framework. In general, the Fast-

Share framework allows one to choose the placement of zeros, the secret, and random masks in several ways. How to leverage this flexibility to obtain the *best* possible recovery and security thresholds is an interesting future direction.

IV. Performance Analysis of FastShare-LRC

A. Theoretical Analysis for Large n

In this section, we analyze the security and recovery guarantees of FastShare-LRC for the large-n regime.

Theorem 1. *For a given $0 < \tau < 1$ and any $0 < \epsilon < 1$, for sufficiently large n, there exists a FastShare-LRC scheme that achieves*

1) *The average-case recovery threshold of $\frac{\tau+\epsilon}{1+\epsilon}$ with probability at least $1 - O\left(\frac{1}{n}\right)$ such that the cost of encoding as well as recovery is $O(n \log n)$; and*
2) *The worst-case security threshold of τ.*

Recovery threshold: For some $c \geq 2$, choose $n_0 = \frac{4c}{(1-\tau)\epsilon^2} \log n$. Note that $\rho = (\tau+\epsilon)/(1+\epsilon)$ implies that $(1-\tau)n_0 = (1+\epsilon)(1-\rho)n_0$. Consider the subset of shares corresponding to an arbitrary row in the 2-dimensional grid representation of the shares. Since each share participates in the recovery process independently with probability ρ, by using the Chernoff bound, it is straightforward to show that the probability that the local decoding of the row fails is at most $\exp(-\epsilon^2(1-\tau)/(2(1+\epsilon)))$. Then, by the union bound and using the expression for n_0 in terms of n, it is easy to show that the local decoding fails with probability at most $1/n^{c-1}$.

Computation cost: Encoding consists of constructing the signal and then taking the FFT. This results in $O(n \log n)$ cost due to the FFT. Recovering the secret consists of two steps: local decoding to obtain the missing shares, and taking the inverse FFT. Decoding each local code takes $O(n_0^2) = O(\log^2 n)$ cost, and there are $n_1 = n/O(\log n)$ number of row codes. Therefore, the total cost of local decoding is $O(n \log n)$. The cost of inverse FFT is also $O(n \log n)$. Hence, recovery can be performed in $O(n \log n)$ cost.

Security threshold: We present a sketch of the proof, which builds on the techniques used in [23]. First, observe that the shares (as a length-n column vector) can be written as $\mathbf{C} = G[\mathbf{S}\ \mathbf{K}]^T$, where $\mathbf{S}$ is the secret and $\mathbf{K}$ is the vector of random masks, and $G \in \mathbb{F}_q^{n \times t}$ is the submatrix of the $n \times n$ DFT matrix (i.e. Vandermonde matrix with powers of ω) obtained by removing the columns corresponding to $\mathcal{Z}_0$ and permuting the remaining columns so that the column corresponding to $\mathcal{S}$ is the first column.

Next, we show that the information-theoretic security condition in (1) can be guaranteed by a particular linear algebraic condition on the columns of submatrices of G. In particular, the following lemma is can be proved as a corollary of [36, Lemma 6].

Lemma 2. *Let $\mathbf{S}$, $\mathbf{K}$, and $\mathbf{C} = G[\mathbf{S}\ \mathbf{K}]^T$ be random variables representing the secret, random masks, and shares, respectively. For an arbitrary set $\mathcal{P} \subset [n]$, let $G_{\mathcal{P}}$ denote the sub-matrix of G corresponding to the rows indexed by $\mathcal{P}$. Let G_1 be the first column of $G_{\mathcal{P}}$ and G_2 be its remaining columns. If $\mathbf{K}$ is uniform over $\mathbb{F}_q^t$, then it holds that*

$$G_1 \in \langle G_2 \rangle \implies I(\mathbf{S}; \mathbf{C}_{\mathcal{P}}) = 0. \tag{9}$$

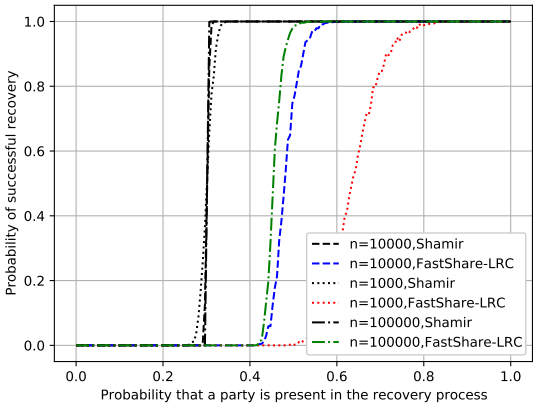

Fig. 5. Probability of success vs ρ for $n = 1000, 10000, 100000$, and the security threshold of 30%. As n increases the ρ with successful recovery approaches τ.

To complete the proof, we show that for any $\mathcal{P} \subset [n]$ of size $\lfloor \tau n \rfloor - 1$, it holds that $G_1 \in \langle G_2 \rangle$. Recall from Sec. III-A that $t = \lfloor \tau n \rfloor - 1$, and observe that G_2 is a square matrix. Now, if G_2 is full-rank for a particular $\mathcal{P}$, then clearly $G_1 \in \langle G_2 \rangle$. The more challenging case is when $\mathcal{P}$ is such that G_2 is rank-deficient. In this case, it is possible to show $G_1 \in \langle G_2 \rangle$ by leveraging the Vandermonde sub-matrix structure of $G_{\mathcal{P}}$. Towards this end, we use similar steps as in [23, Appendix C] which involve iteratively using [23, Lemma C.2].

B. Numerical Simulations

In this section, we numerically evaluate the recovery performance of FastShare-LRC for various values of n. In particular, we fix $\tau = 0.3$, and consider $n = 1000$, 10000, and 100000. For $n = 1000$, we take $n_0 = 10$ and $n_1 = 100$; for $n = 10000$, we take $n_0 = 50$ and $n_1 = 200$; and for $n = 100000$, we take $n_0 = 100$ and $n_1 = 1000$. We plot the probability of successful recovery versus ρ in Fig. 5. For $n = 1000$, 10000, and 100000, we get $\rho = 0.825$, 0.6, and 0.52, respectively, that guarantees successful recovery with probability 99.99%. Note that for $\tau = 0.3$, Shamir's scheme achieves $\rho = 0.3$. One can observe that the ρ with guaranteed successful recovery approaches τ as n grows larger.

To see how FastShare-LRC can be beneficial in practical applications, consider the moderately large value of $n = 10000$, which is typical in federated learning [37]. At this n, FastShare-LRC achieves $\rho \approx 0.6$ with guaranteed recovery, allowing parties to fail with $\sim$40% probability. This works well for federated learning, where it is observed that 6% to 10% of participants fail on average [37]. As discussed in the Introduction, for $n = 10000$ and $\tau = 0.3$, FastShare-LRC cuts down the recovery cost by $100\times$ — from $O(10^6)$ to $O(10^4)$. This demonstrates that the FastShare framework can be used to design schemes that are well suited in practical applications.

REFERENCES

[1] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612–613, Nov. 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*. Los Alamitos, CA, USA: IEEE Computer Society, jun 1979, p. 313. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/AFIPS.1979.98

[3] R. Cramer, I. Damgård, and U. Maurer, "General secure multi-party computation from any linear secret-sharing scheme," in *Advances in Cryptology — EUROCRYPT 2000*, B. Preneel, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 316–334.

[4] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Foundations and Trends in Privacy and Security*, vol. 2, pp. 70–246, 2018.

[5] V. Attasena, J. Darmont, and N. Harbi, "Secret sharing for cloud data security: A survey," *The VLDB Journal*, vol. 26, no. 5, p. 657–681, Oct. 2017. [Online]. Available: https://doi.org/10.1007/s00778-017-0470-9

[6] A. Beimel, "Secret-sharing schemes: A survey," in *Coding and Cryptology*, Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, and C. Xing, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 11–46.

[7] E. Karnin, J. Greene, and M. Hellman, "On secret sharing systems," *IEEE Transactions on Information Theory*, vol. 29, no. 1, pp. 35–41, 1983.

[8] M. Franklin and M. Yung, "Communication complexity of secure computation (extended abstract)," in *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, 1992, pp. 699–710.

[9] M. Cheraghchi, "Nearly optimal robust secret sharing," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2509–2513.

[10] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, 2017, pp. 1175–1191.

[11] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "Sepia: Privacy-preserving aggregation of multi-domain network events and statistics," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10. USA: USENIX Association, 2010, p. 15.

[12] M. Muhil, U. H. Krishna, R. K. Kumar, and E. M. Anita, "Securing multi-cloud using secret sharing algorithm," *Procedia Computer Science*, vol. 50, pp. 421 – 426, 2015, big Data, Cloud and Computing Challenges. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915005128

[13] G. Ács and C. Castelluccia, "I have a dream! differentially private smart metering," in *Proceedings of the 13th International Conference on Information Hiding*, ser. IH'11. Berlin, Heidelberg: Springer-Verlag, 2011, p. 118–132.

[14] J. v. z. Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed. USA: Cambridge University Press, 2013.

[15] E. Horowitz, "A fast method for interpolation using preconditioning," *Information Processing Letters*, vol. 1, no. 4, pp. 157 – 163, 1972. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020019072900506

[16] A. Borodin and R. Moenck, "Fast modular transforms," *Journal of Computer and System Sciences*, vol. 8, no. 3, pp. 366 – 386, 1974. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000074800292

[17] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, 2nd ed. North-holland Publishing Company, 1978.

[18] K. S. Kedlaya and C. Umans, "Fast polynomial factorization and modular composition," *SIAM J. Comput.*, vol. 40, no. 6, p. 1767–1802, Dec. 2011. [Online]. Available: https://doi.org/10.1137/08073408X

[19] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[20] R. E. Blahut, "Transform techniques for error control codes," *IBM Journal of Research and Development*, vol. 23, no. 3, pp. 299–315, 1979.

[21] S. Pawar and K. Ramchandran, "Ffast: An algorithm for computing an exactly k -sparse dft in $o(k \log k)$ time," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 429–450, 2018.

[22] N. T. Janakiraman, S. Emmadi, K. Narayanan, and K. Ramchandran, "Exploring connections between sparse fourier transform computation and decoding of product codes," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 1366–1373.

[23] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "Fast-secagg: Scalable secure aggregation for privacy-preserving federated learning," 2020.

[24] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k,n)-threshold secret sharing scheme and its extension," in *Information Security*, T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 455–470.

[25] Y. Wang and Y. Desmedt, "Efficient secret sharing schemes achieving optimal information rate," in *2014 IEEE Information Theory Workshop (ITW 2014)*, 2014, pp. 516–520.

[26] Y. Wang, "Privacy-preserving data storage in cloud using array bp-xor codes," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 425–435, 2015.

[27] A. Agarwal and A. Mazumdar, "Security in locally repairable storage," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6204–6217, 2016.

[28] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7195–7206, 2016.

[29] R. Bitar and S. E. Rouayheb, "Staircase codes for secret sharing with optimal communication and read overheads," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 933–943, 2018.

[30] U. Martínez-Peñas, "Communication efficient and strongly secure secret sharing schemes based on algebraic geometry codes," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4191–4206, 2018.

[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[32] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *Information Theory, IEEE Transactions on*, vol. 58, no. 11, pp. 6925–6934, Nov 2012.

[33] N. Prakash, G. Kamath, V. Lalitha, and P. Kumar, "Optimal linear codes with a local-error-correction property," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012, pp. 2776–2780.

[34] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *Information Theory, IEEE Transactions on*, vol. 60, no. 8, pp. 4661–4676, Aug 2014.

[35] J. M. Pollard, "The fast Fourier transform in a finite field," *Mathematics of Computation*, vol. 25, pp. 365–374, 1971.

[36] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1124–1135, Feb. 2011.

[37] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proceedings of the 2nd SysML Conference*, 2019.