Communication Efficient Secret Sharing

Wentao Huang, Michael Langberg, *Senior Member, IEEE*, Jörg Kliewer, *Senior Member, IEEE*, and Jehoshua Bruck, *Fellow, IEEE*

***Abstract*— A secret sharing scheme is a method to store information securely and reliably. Particularly, in a *threshold secret sharing scheme*, a secret is encoded into n shares, such that any set of at least t_1 shares suffice to decode the secret, and any set of at most $t_2 < t_1$ shares reveal no information about the secret. Assuming that each party holds a share and a user wishes to decode the secret by receiving information from a set of parties; the question we study is how to minimize the amount of communication between the user and the parties. We show that the necessary amount of communication, termed "decoding bandwidth", decreases as the number of parties that participate in decoding increases. We prove a tight lower bound on the decoding bandwidth, and construct secret sharing schemes achieving the bound. Particularly, we design a scheme that achieves the optimal decoding bandwidth when d parties participate in decoding, universally for all $t_1 \leq d \leq n$. The scheme is based on a generalization of Shamir's secret sharing scheme and preserves its simplicity and efficiency. In addition, we consider the setting of secure distributed storage where the proposed communication efficient secret sharing schemes not only improve decoding bandwidth but further improve disk access complexity during decoding.**

***Index Terms*— Security, secret sharing, communication bandwidth, distributed storage, Reed-Solomon codes.**

I. Introduction

CONSIDER the scenario that n parties wish to store a secret securely and reliably. To this end, a dealer distributes the secret into n shares, i.e., one share for each party, such that 1) (reliability) a collection $\mathcal{A} \subset 2^{\{1,\ldots,n\}}$ of "authorized" subsets of the parties can decode the secret, and 2) (secrecy) a collection $\mathcal{B}$ of "blocked" subsets of the parties cannot collude to deduce any information about the secret. A scheme to distribute the secret into shares with respect to *access structure* $(\mathcal{A}, \mathcal{B})$ is called a secret sharing scheme, initially studied in the seminal works by Shamir [20] and Blakley [3]. A secret sharing scheme is *perfect* if a subset of parties is either authorized or blocked, i.e., $\mathcal{A} \cup \mathcal{B} = 2^{\{1,\ldots,n\}}$. A scheme is often referred to as a *ramp* scheme if it is not prefect [4]. An important application of secret sharing schemes is distributed storage of private data, where each party is a storage node. Besides, secret sharing is a fundamental cryptographic primitive and is used as a building block in numerous secure protocols [1].

We focus on secret sharing schemes for the *threshold* access structure, i.e., $\mathcal{A}$ contains all subsets of $\{1, \ldots, n\}$ of size at least $n - r$, and $\mathcal{B}$ contains all subsets of $\{1, \ldots, n\}$ of size at most z. In other words, the secret can be decoded in the absence of any r parties, and any z parties cannot collude to deduce any information about the secret. Note that a threshold scheme is perfect if $z+r = n-1$ since any set of at least $n-r$ parties are authorized and any set of less parties are blocked. The scheme is a ramp scheme if $z+r < n-1$. The threshold access structure is particularly important in practice, because for this case, space and computationally efficient secret sharing schemes are known. Specifically, Shamir [20] constructs an elegant and efficient perfect threshold scheme using the idea of polynomial interpolation. Shamir's scheme is later shown to be closely related to Reed-Solomon codes [14] and is generalized to ramp schemes in [4], [9], and [24], which allow better space efficiency, i.e., rate, than the original perfect scheme. Shamir's scheme and the generalized ramp schemes achieve optimal usage of storage space, in the sense that fixing the size of the shares, the schemes store a secret of maximum size. The schemes are computationally efficient as decoding the secret is equivalent to polynomial interpolation. An example of Shamir's scheme is shown in Figure 1. Other threshold secret sharing schemes and generalizations of Shamir's scheme may be found in, e.g., [11]–[13] and [25]. Recently, there has been considerable interest in incorporating secrecy into erasure codes for distributed storage, e.g., [15], [18], [19]. These codes can also be viewed as threshold secret sharing schemes.

In addition to space and computational efficiency, this paper studies the *communication efficiency* for secret sharing schemes. Consider the scenario that a user wishes to decode the secret by downloading information from the parties that are available. Referring to the amount of information downloaded by the user as the *decoding bandwidth*, a natural question is to address the minimum decoding bandwidth that allows decoding. It is of practical interest to design secret sharing schemes that achieve a small decoding bandwidth, or in other words, that require communicating only a small amount of information during decoding. In such a case, decoding will be

Manuscript received July 1, 2015; revised May 16, 2016; accepted September 11, 2016. Date of publication October 10, 2016; date of current version November 18, 2016. This work was supported in part by NSF under Grant CCF-1218005, Grant CCF-1321129, Grant CCF-1526771, and Grant CNS-152654, in part by the United States-Israel Binational Science Foundation under Grant 2010075, and in part by the Caltech Lee Center.

W. Huang and J. Bruck are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: whuang@caltech.edu; bruck@caltech.edu).

M. Langberg is with the Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: mikel@buffalo.edu).

J. Kliewer is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (email: jkliewer@njit.edu).

Communicated by A. G. Dimakis, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2016.2616144

Party 1	Party 2	Party 3	Party 4	Party 5	Party 6	Party 7
$f(1) =$	$f(2) =$	$f(3) =$	$f(4) =$	$f(5) =$	$f(6) =$	$f(7) =$
$m_1 + m_2 + k_1$	$m_1 + 2m_2 + 4k_1$	$m_1 + 3m_2 + 9k_1$	$m_1 + 4m_2 + 5k_1$	$m_1 + 5m_2 + 3k_1$	$m_1 + 6m_2 + 3k_1$	$m_1 + 7m_2 + 5k_1$

Fig. 1. Shamir's scheme (generalized ramp version) for $n = 7$, $r = 4$, $z = 1$, with symbols over $\mathbb{F}_{11}$. The scheme stores a secret of two symbols, denoted by m_1, m_2. Let k_1 be a uniformly and independently distributed random variable. $f(x)$ is the polynomial $m_1 + m_2x + k_1x^2$. Note that the share stored by any single party is independent of the secret because it is padded by k_1, and that the secret can be decoded from the shares stored by any three parties by polynomial interpolation.

Party 1	Party 2	Party 3	Party 4	Party 5	Party 6	Party 7
$m_1 + m_2 + k_1$	$m_1 + 2m_2 + 4k_1$	$m_1 + 3m_2 + 9k_1$	$m_1 + 4m_2 + 5k_1$	$m_1 + 5m_2 + 3k_1$	$m_1 + 6m_2 + 3k_1$	$m_1 + 7m_2 + 5k_1$
$m_3 + m_4 + k_2$	$m_3 + 2m_4 + 4k_2$	$m_3 + 3m_4 + 9k_2$	$m_3 + 4m_4 + 5k_2$	$m_3 + 5m_4 + 3k_2$	$m_3 + 6m_4 + 3k_2$	$m_3 + 7m_4 + 5k_2$
$m_5 + m_6 + k_3$	$m_5 + 2m_6 + 4k_3$	$m_5 + 3m_6 + 9k_3$	$m_5 + 4m_6 + 5k_3$	$m_5 + 5m_6 + 3k_3$	$m_5 + 6m_6 + 3k_3$	$m_5 + 7m_6 + 5k_3$

(a) Shamir's Scheme

Party 1	$\cdots$	Party 7
$f(1) = k_1 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6$	$\cdots$	$f(7) = k_1 + 7m_1 + 5m_2 + 2m_3 + 3m_4 + 10m_5 + 6m_6$
$g(1) = k_2 + m_4 + m_5 + m_6$	$\cdots$	$g(7) = k_2 + 7m_4 + 5m_5 + 2m_6$
$h(1) = k_3 + m_3 + m_6$	$\cdots$	$h(7) = k_3 + 7m_3 + 5m_6$

(b) Proposed Scheme

Fig. 2. Two secret sharing schemes for $n = 7, r = 4$ and $z = 1$ over $\mathbb{F}_{11}$. Both schemes store a secret of six symbols (m_1, ..., m_6). In both schemes, k_1, k_2, k_3 are i.i.d. uniformly distributed random variables. Scheme (a) is Shamir's scheme (see Figure 1) repeated three times. In scheme (b), $f(x) = k_1 + m_1x + m_2x^2 + m_3x^3 + m_4x^4 + m_5x^5 + m_6x^6$, $g(x) = k_2 + m_4x + m_5x^2 + m_6x^3$, $h(x) = k_3 + m_3x + m_6x^2$, and party i stores evaluations $f(i)$, $g(i)$ and $h(i)$. Note that in (b), if all 7 parties are available, then the secret can be decoded by downloading only one symbol $f(i)$ from each party i, and then interpolating $f(x)$. If any 4 parties are available, then the secret can be decoded in the following way. Download two symbols $f(i), g(i)$ from each available party i and first interpolate $g(x)$, implying that all coefficients of $f(x)$ of degree larger than 3 are decoded. The remaining unknown part of $f(x)$ is a degree-3 polynomial and so we have enough evaluations of $f(x)$ to interpolate it, hence completely decoding the secret. Similarly, if any 3 parties are available, then the secret can be decoded in the following way. Download all three symbols $f(i), g(i), h(i)$ from each available node i and interpolate $h(x)$, which decodes the degree-3 coefficients of $f(x)$ and $g(x)$. Hence the remaining unknown part of $g(x)$ is a degree-2 polynomial and can be interpolated, which decodes the coefficients of $f(x)$ of degrees $4, 5, 6$. Hence the remaining unknown part of $f(x)$ is a degree-2 polynomial and can be interpolated, decoding the complete secret. This shows that the scheme meets the reliability requirement. In fact, for $d = 3, 4, 7$, scheme (b) achieves the optimal decoding bandwidth when d parties participate in decoding. The secrecy of the scheme derives from the secrecy of Shamir's scheme, as each polynomials $f(x)$, $g(x)$ and $h(x)$ individually is an instance of Shamir's scheme, and we show that combining them still meets the secrecy requirement. The construction is discussed in detail in Section IV. (a) Shamir's Scheme. (b) Proposed Scheme.

completed in a timely manner and the communication resource will be more efficiently utilized.

In many existing secret sharing schemes, e.g., [4], [9], [11]–[15], [18]–[20], [24], [25], a common practice in decoding is that the user will communicate with a minimum set of parties, i.e., exactly $n - r$ parties (even if $d > n - r$ parties are available) and download the whole share stored by these parties. Wang and Wong [23] show that this paradigm is not optimal in terms of communication and that the decoding bandwidth can be reduced if the user downloads only part of the share from each of the $d > n - r$ available parties. Specifically, given d, for any perfect threshold secret sharing scheme, [23] derives a lower bound on the decoding bandwidth when exactly d parties participate in decoding, and designs a perfect scheme that achieves the lower bound. The field size of the scheme is slightly improved in [27]. However, two interesting and important problems remain open: 1) the schemes in [23] and [27] achieve the lower bound on decoding bandwidth when the number of available parties d equals a single specific value, and do not achieve the bound if d takes other values. This raises the question whether the lower bound is uniformly tight, or in other words, if it is possible to design a single scheme that achieves the lower bound universally for all d in the range of $[n-r, n]$. 2) The results in [23] and [27] target the case of perfect secret sharing schemes, i.e., the case of $z + r = n - 1$. It is well known that for this case the size of each share is at least as large as the size of the secret [7], [21], and so the rate of the scheme is at most $1/n$. This raises the question of how to generalize the results and ideas to the case of $z + r \leq n - 1$, so that the parameters and rate of the schemes are more flexible. Both problems are of practical importance as the first problem addresses the flexibility of a scheme in terms of decoding, and the second problem addresses the high-rate case which is a typical requirement in many practical applications such as distributed storage. In this paper we settle both problems and construct (perfect and ramp) schemes of flexible rate that achieve the optimal decoding bandwidth universally. Similar to Shamir's scheme, our schemes are computationally efficient and have optimal space efficiency.

A. Motivating Example

Consider Shamir's ramp scheme in the example of Figure 1, that stores 2 symbols securely and reliably for the setting $n = 7, r = 4$ and $z = 1$. In order to decode the secret, a user needs to download 3 symbols from any 3 parties, and therefore the decoding bandwidth is 3 symbols. Now suppose the same scheme is repeated 3 times in order to store a secret of 6 symbols, as shown in Figure 2a. Then to decode the secret, the decoding bandwidth is 9 symbols.

We propose a new scheme in Figure 2b that also stores a secret of 6 symbols for the same setting, using the same amount of storage space, and over the same field size. In this scheme, if any 3 parties are available, then similar to Shamir's scheme, the secret can be decoded from the 9 symbols stored by the three parties. However, if any 4 parties are available, then the secret can be decoded by downloading 2 symbols from each available party. Therefore, the decoding bandwidth is improved to 8 symbols. If all 7 parties are available, then the secret can be decoded by downloading only 1 symbol from each party and so the decoding bandwidth is further reduced to 7 symbols.

We use the examples in Figure 2 to highlight several ideas to reduce the decoding bandwidth. Firstly, the amount of communication depends on the number of available parties. In fact the necessary amount of communication decreases strictly as the number of available parties increases. Secondly, it is important to distribute multiple subshares (symbols) to a party (essentially using the ideas of array codes [5], [6]). In contrast, Shamir's scheme only distributes one symbol to each party except for trivial repetitions. Thirdly, during decoding it is not always necessary to download the complete share stored by a party. In general, a party can preprocess its share and the user can download a function of the share.

Comparing to the schemes in [23] and [27], the scheme in the example is improved and generalized in the following aspects. 1) The proposed scheme achieves the optimal bandwidth more flexibly. Specifically, the schemes in [23] and [27] achieve the optimal bandwidth for a single specific number of available parties. The proposed scheme can be designed to allow flexibility in the number of available parties d. In the example of Figure 2b the scheme achieves the optimal bandwidth when $d = 3, 4, 7$. In general, we can construct schemes that achieve the optimal bandwidth for all $n - r \leq d \leq n$. 2) The proposed scheme is more flexible in rate. Specifically, the (perfect) schemes in [23] and [27] have rate exactly $1/n$. The proposed scheme in the example has rate $2/7 > 1/n = 1/7$. In general, we can construct schemes of arbitrary rate.

We also remark on an interesting analog between communication efficient secret sharing and the well-studied subject of regenerating codes [8], [16], [22]. Consider a regenerating code of length n that is able to correct $r > 1$ erasures. If only one erasure occurs, then compared to repairing from a minimum set of $n - r$ nodes, repairing from all the $n - 1$ available nodes will significantly reduce the total amount of communication that occurs during the repair. In this sense, for both regenerating codes and communication efficient secret sharing, a key idea is to involve more available nodes/parties than the minimum required set during repair/decoding, for the purpose of reducing the repair/decoding bandwidth.

B. Results

In Section III, we prove a tight information-theoretic lower bound on the decoding bandwidth, given a set of available parties $I \subset \{1, \ldots, n\}$. The bound implies that the decoding bandwidth decreases as $|I|$ increases. The lower bound applies to both perfect and ramp schemes and generalizes the lower bound in [23]. Particularly, we show that the overhead in communication for the case of $|I| = n$ is only a fraction $(n - r - z)/(n - z)$ of the communication overhead when $|I| = n - r$.

In Section IV, we construct efficient secret sharing schemes using the ideas described in Section I-A. Our construction utilizes Shamir's scheme and achieves the optimal decoding bandwidth universally for all $I \in \mathcal{A}$. Additionally, the construction preserves the simplicity of Shamir's scheme and is efficient in terms of both space and computation. Specifically, the scheme achieves optimal space efficiency, and requires the same field size as Shamir's scheme. Encoding and decoding the scheme is also similar to encoding and decoding Shamir's scheme. The scheme shows that our lower bound in Section III is uniformly tight. Interestingly, the scheme also generalizes the construction in a recent independent work [2]. However, the flexibility of our framework allows improved efficiency in terms of computation, decoding delay and partial decoding.

In Section V, we construct another secret sharing scheme from Reed-Solomon codes. The scheme achieves the optimal decoding bandwidth when $|I| = n$ and $|I| = n - r$. The decoder of the scheme has a simpler structure compared to the decoder of the previous scheme, and therefore is advantageous in terms of implementation. The scheme also offers a stronger level of reliability in that it allows decoding even if more than r shares are partially lost.

Finally, in the application of storage where each party is regarded as a disk, it is desirable to optimize the efficiency of disk operations. Our lower bound on the decoding bandwidth is naturally a lower bound on the number of symbol-reads from disks during decoding. In all of our schemes, the number of symbol-reads during decoding equals to the amount of communication. Therefore, our schemes are also optimal in terms of disk operations. In addition, by involving more than the minimum number of disks for decoding, our schemes balance the load at the disks and achieve a higher degree of parallelization.

II. Secret Sharing Schemes

Consider the problem of storing a secret message $\boldsymbol{m}$ securely and reliably into n shares, so that 1) $\boldsymbol{m}$ can be recovered from any $n - r$ shares, and 2) any z shares do not reveal any information about $\boldsymbol{m}$, i.e., they are statistically independent. Such a scheme is called a threshold secret sharing scheme, defined formally as follows. Let $\mathcal{Q}$ be a general Q-ary alphabet, i.e., $|\mathcal{Q}| = Q$. Denote by $[n] = \{1, \ldots, n\}$. For any index set $I \subset [n]$ and a vector $\boldsymbol{c} = (c_1, \ldots, c_n)$, denote by $\boldsymbol{c}_I = (c_i)_{i \in I}$.

Definition 1: An $(n, k, r, z)_{\mathcal{Q}}$ secret sharing scheme consists of a randomized encoding function F that maps a secret $\boldsymbol{m} \in \mathcal{Q}^k$ to $\boldsymbol{c} = (c_1, \ldots, c_n) = F(\boldsymbol{m}) \in \mathcal{Q}^n$, such that

1) *(Reliability) The secret $\boldsymbol{m}$ can be decoded from any $n - r$ shares (entries) of $\boldsymbol{c}$. This guarantees that $\boldsymbol{m}$ is recoverable in the loss of any r shares. Formally,*

$$H(\boldsymbol{m}|\boldsymbol{c}_I) = 0, \quad \forall I \subset [n], \ |I| = n - r. \tag{1}$$

Therefore for any $I \subset [n]$, $|I| = n - r$, there exists a decoding function $D_I^* : \mathcal{Q}^{n-r} \to \mathcal{Q}^k$ such that $D_I^*(\boldsymbol{c}_I) = \boldsymbol{m}$.

2) *(Secrecy) Any z shares of $\boldsymbol{c}$ do not reveal any information about $\boldsymbol{m}$. This guarantees that $\boldsymbol{m}$ is secure if any z shares are exposed to an eavesdropper. Formally,*

$$H(\boldsymbol{m}|\boldsymbol{c}_I) = H(\boldsymbol{m}), \quad \forall I \subset [n],\ |I| = z. \tag{2}$$

Define the *rate* of a scheme to be k/n, which measures the space efficiency. The following proposition gives an upper bound on the rate.

Proposition 1: For any $(n, k, r, z)_{\mathcal{Q}}$ secret sharing scheme, it follows that

$$k \leq n - r - z, \tag{3}$$

and so the rate of the scheme is at most $\frac{n-r-z}{n}$.

Proof: Let the message $\boldsymbol{m}$ be uniformly distributed, then

$$\begin{aligned} k = H(\boldsymbol{m}) &= H(\boldsymbol{m}|\boldsymbol{c}_{[z]}) && (4)\\ &\leq H(\boldsymbol{m}, \boldsymbol{c}_{[n-r]}|\boldsymbol{c}_{[z]}) \\ &= H(\boldsymbol{m}|\boldsymbol{c}_{[n-r]}, \boldsymbol{c}_{[z]}) + H(\boldsymbol{c}_{[n-r]}|\boldsymbol{c}_{[z]}) && (5)\\ &= H(\boldsymbol{c}_{[n-r]}|\boldsymbol{c}_{[z]}) \\ &= H(\boldsymbol{c}_{\{z+1,\ldots,n-r\}}) \leq n - r - z, && (6) \end{aligned}$$

where (4) follows from the security requirement, (5) follows from the chain rule, and (6) follows from the reliability requirement. □

A secret sharing scheme is *rate-optimal* if it achieves equality in (3). Note that the scheme is a *perfect scheme* if $z = n - r - 1$ and is a *ramp scheme* otherwise. Rate-optimal perfect secret sharing schemes are studied in the seminal work by Shamir [20], and are later generalized to ramp schemes [4], [9], [24]. Note that by (3) the rate of any perfect scheme is at most $1/n$ as $k = 1$. Any scheme of a higher rate is necessarily a ramp scheme.

III. Lower Bound on Communication Overhead

Suppose that the n shares of the secret are stored by n parties or distributed storage nodes,[1] and a user wants to decode the secret. By Definition 1, the user can connect to any $n - r$ nodes and download one share, i.e., one Q-ary symbol, from each node. Therefore, by communicating $n - r$ symbols, the user can decode a secret of $k \leq n - r - z$ symbols. It is clear that a communication overhead of z symbols occurs during *decoding*. The question is, whether it is possible to reduce the communication overhead. We answer this question affirmatively in the remaining part of the paper.

There are two key ideas for improving the communication overhead. Firstly, in many practical scenarios and particularly in distributed storage systems, often time more than $n - r$ nodes are available. In this case, it is not necessary to restrict the user to download from only $n - r$ nodes. Secondly, it is not necessary to download the complete share stored by the node. Instead, it may suffice to communicate only a part of the share or, in general, a function of the share. In other words, a node can preprocess its share before transmitting it to the user.

Motivated by these ideas, for any $I \subset [n]$, $|I| \geq n - r$, define a class of *preprocessing functions* $E_{I,i} : \mathcal{Q} \to \mathcal{S}_{I,i}$, where $|\mathcal{S}_{I,i}| \leq |\mathcal{Q}|$, that maps c_i to $e_{I,i} = E_{I,i}(c_i)$. Let $\boldsymbol{e}_I = (e_{I,i})_{i\in I}$, and define a class of *decoding functions* $D_I : \prod_{i\in I} \mathcal{S}_{I,i} \to \mathcal{Q}^k$, such that $D_I(\boldsymbol{e}_I) = \boldsymbol{m}$. For a naive example, consider any I such that $|I| = n - r$. Then for $i \in I$, we can let $\mathcal{S}_{I,i} = \mathcal{Q}$, let $E_{I,i}$ be the identity function, and let D_I be the naive decoding function D_I^* described in Definition 1. In the remaining paper, when I is clear from the context, we will suppress it in the subscripts of $\mathcal{S}_{I,i}$, $E_{I,i}$, $e_{I,i}$ and $\boldsymbol{e}_I$, and denote them by $\mathcal{S}_i$, E_i, e_i and $\boldsymbol{e}$ instead. We now formally define the notion of communication overhead in decoding. Note that all log functions in the paper are base Q.

Definition 2: For any I such that $|I| \geq n - r$, define the communication overhead function to be $\mathrm{CO}(I) = \sum_{i\in I} \log|\mathcal{S}_{I,i}| - k$. Namely, $\mathrm{CO}(I)$ is the amount of extra information, measured in Q-ary symbols, that one needs to communicate in order to decode a secret of k symbols, provided that the set of available shares is indexed by I.

The following result provides a lower bound on the communication overhead function. It generalizes the lower bound in [23] for perfect schemes, i.e., schemes with $z + r = n - 1$.

Theorem 1: For any $(n, k, r, z)_{\mathcal{Q}}$ secret sharing scheme with preprocessing functions $\{E_{I,i}\}_{i\in I, |I|\geq n-r}$ and decoding functions $\{D_I\}_{|I|\geq n-r}$, it follows that

$$\mathrm{CO}(I) \geq \frac{kz}{|I| - z}. \tag{7}$$

Proof: Consider arbitrary $I = \{i_1, \ldots, i_{|I|}\}$ such that $|I| \geq n - r$. Assume without loss of generality that $|\mathcal{S}_{i_1}| \leq |\mathcal{S}_{i_2}| \leq \ldots \leq |\mathcal{S}_{i_{|I|}}|$. Recall that $\boldsymbol{e}_I = (e_{i_1}, \ldots, e_{i_{|I|}})$ is the output of the preprocessing functions.

$$\begin{aligned} H(e_{i_1}, \ldots, e_{i_{|I|-z}}) &\overset{(a)}{\geq} H(e_{i_1}, \ldots, e_{i_{|I|-z}}|e_{i_{|I|-z+1}}, \ldots, e_{i_{|I|}}) \\ &\overset{(b)}{=} H(e_{i_1}, \ldots, e_{i_{|I|-z}}|e_{i_{|I|-z+1}}, \ldots, e_{i_{|I|}}) \\ &\quad + H(\boldsymbol{m}|e_{i_1}, \ldots, e_{i_{|I|}}) \\ &\overset{(c)}{=} H(\boldsymbol{m}, e_{i_1}, \ldots, e_{i_{|I|-z}}|e_{i_{|I|-z+1}}, \ldots, e_{i_{|I|}}) \\ &\geq H(\boldsymbol{m}|e_{i_{|I|-z+1}}, \ldots, e_{i_{|I|}}) \\ &\overset{(d)}{=} H(\boldsymbol{m}) = k, \end{aligned} \tag{8}$$

where (a) follows from conditioning reduces entropy, (b) follows from (1), (c) follows form the chain rule, and (d) follows from (2). Therefore it follows from (8) that

$$\prod_{j=1}^{|I|-z} |\mathcal{S}_{i_j}| \geq Q^{H(e_{i_1},\ldots,e_{i_{|I|-z}})} \geq Q^k,$$

and so

$$\sum_{j=1}^{|I|-z} \log|\mathcal{S}_{i_j}| \geq k. \tag{9}$$

It then follows from $|\mathcal{S}_{i_1}| \leq \ldots \leq |\mathcal{S}_{i_{|I|}}|$ that,

$$\log|\mathcal{S}_{i_{|I|-z}}| \geq \frac{k}{|I| - z},$$

[1] In what follows we do not distinguish between parties and nodes.

and that,

$$\log|\mathcal{S}_{i_{|I|-z+j}}| \geq \log|\mathcal{S}_{i_{|I|-z}}| \geq \frac{k}{|I|-z}, \quad j=1,\ldots,z. \quad (10)$$

Combining (9) and (10) we have,

$$\mathrm{CO}(I) = \sum_{j=1}^{|I|} \log|\mathcal{S}_{i_j}| - k \geq \frac{kz}{|I|-z}.$$

□

The *decoding bandwidth* is defined to be the total amount of Q-ary symbols the user downloads from the nodes, which equals $\mathrm{CO}(I)+k$. Theorem 1 suggests that the communication overhead and the decoding bandwidth decrease as the number of available nodes increases.

For rate-optimal schemes, Theorem 1 implies that if $|I| = n-r$, then the communication overhead is at least z, i.e., the user needs to download the complete share from each available node. The naive decoding function D_I^* in Definition 1 trivially achieves this bound. The more interesting scenario is the regime that $|I| > n-r$. In this case, if (7) is tight, then one can achieve a non-trivial improvement on decoding bandwidth compared to the naive decoder D_I^*. When $k = n-r-z = 1$ (i.e., for perfect schemes) and fixing any $d > n-r$, [23] constructs a rate-optimal scheme that achieves the lower bound (7) for any I such that $|I| = d$. However, several interesting and important questions remain open. Firstly, is the lower bound uniformly tight, or in other words, is it possible to construct a scheme that achieves (7) universally for any I such that $|I| \geq n-r$ (note that the scheme in [23] does not achieve the lower bound when $|I| \neq d$)? Secondly, is the bound tight when $k > 1$ (i.e., for ramp schemes) and how can we design such schemes? We answer these questions in the following section.

IV. Construction From Shamir's Scheme

In this section we construct a rate-optimal scheme that achieves the optimal decoding bandwidth universally for all possible I, i.e., all sets of available nodes. This implies that the lower bound in Theorem 1 is uniformly tight. The scheme is based on a generalization of Shamir's scheme and preserves its simplicity and efficiency. The scheme is flexible in the parameters n, k, r, z and hence is flexible in rate.

We first refer the readers to Figure 2b for an example of the scheme, and use it to describe the general idea of the construction. To construct a scheme that achieves the optimal decoding bandwidth when d nodes are available, for all $d \in \mathcal{D}$, we design a set of polynomials of different degrees. Particularly, for all $d \in \mathcal{D}$, we design a number of polynomials of degree exactly $d-1$, and store one evaluation of each polynomial at each node. For each polynomial, exactly z of its coefficients are independent keys in order to meet the secrecy requirement. The remaining coefficients encode "information": for the highest-degree (e.g., degree $d_{\max}-1$, where $d_{\max} = \max_{d\in\mathcal{D}} d$) polynomials, their coefficients encode the entire message; for other polynomials, say $g(x)$, the information encoded in the coefficients of $g(x)$ is the high-degree coefficients of the polynomials of degree higher than $g(x)$. Such an arrangement of the coefficients enables decoding in a successive manner. Consider decoding when d nodes are available, implying that d evaluations of each polynomial are known and hence all polynomials of degree $d-1$ can be interpolated. Then, roughly speaking, the arrangement ensures that the high-degree coefficients of some higher-degree polynomials are known, so that the remaining unknown parts of these polynomials can be interpolated. This in turn allows to decode coefficients for additional high-degree polynomials and thus to interpolate them. The chain continues until all polynomials of degree higher than $d-1$ are interpolated, implying that the message is decoded. Note that no polynomials of degree smaller than $d-1$ are interpolated, and therefore the keys associated with them are not decoded. This leads to the saving in decoding bandwidth and in fact this amount is the best one can expect to save, so that the scheme achieves the optimal bandwidth. Below we describe the scheme formally.

A. Encoding

Consider arbitrary parameters n, r, z, $\mathcal{D}$ and let $k = n-r-z$. We assume that $n-r \in \mathcal{D}$ since it is implied by the reliability requirement. Choose any prime power $q > n$, the scheme is $\mathbb{F}_q$-linear over share alphabet $\mathcal{Q} = \mathbb{F}_q^b$, where b is the number of ($\mathbb{F}_q$) symbols stored by each node. The message $\boldsymbol{m}$ is a vector over $\mathbb{F}_q$ of length $|\boldsymbol{m}| = kb$. The choice of b is determined by $\mathcal{D}$ in the following way. Let $|\boldsymbol{m}|$ be the least common multiple of $\{d-z : d \in \mathcal{D}\}$, i.e., the smallest positive integer that is divisible by all elements of the set. Note that indeed $|\boldsymbol{m}|$ is a multiple of $k = n-r-z$, and we let $b = \frac{|\boldsymbol{m}|}{k}$. This is the smallest choice of $|\boldsymbol{m}|$ (and thus b) that ensures when $d \in \mathcal{D}$ nodes are available, that the optimal bandwidth, measured by the number of $\mathbb{F}_q$ symbols, is an integer.

We now construct b polynomials over $\mathbb{F}_q$, evaluate each of them at n non-zero points, and let every node stores an evaluation of each polynomial. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$, such that $n \geq d_1 > d_2 > \ldots > d_{|\mathcal{D}|} = n-r$. For $i \in |\mathcal{D}|$, let

$$p_i = \begin{cases} \frac{|\boldsymbol{m}|}{d_1-z} & i=1 \\ \frac{|\boldsymbol{m}|}{d_i-z} - \frac{|\boldsymbol{m}|}{d_{i-1}-z} & i>1 \end{cases} \quad (11)$$

We construct p_i polynomials of degree d_i-1. For all polynomials, their z lowest-degree coefficients are independent random keys. We next define the remaining $d_i - z$ non-key coefficients. We first define them for the highest degree polynomials, and then recursively define them for the lower degree polynomials. For $i=1$, the non-key coefficients of the polynomials of degree d_i-1 are message symbols. Note that there are $|\boldsymbol{m}|$ message symbols and $\frac{|\boldsymbol{m}|}{d_1-z}$ polynomials of degree d_1-1. Each such polynomial has d_1-z non-key coefficients and so there are exactly enough coefficients to encode the message symbols. For $i>1$, the non-key coefficients encode the degree d_i to $d_{i-1}-1$ coefficients of all higher (than d_i-1) degree polynomials. Note that there are $\sum_{j=1}^{i-1} p_j = \frac{|\boldsymbol{m}|}{d_{i-1}-z}$ higher degree polynomials and so the total number of coefficients to encode is $(d_{i-1}-d_i)\frac{|\boldsymbol{m}|}{d_{i-1}-z}$. On the other hand, there are p_i polynomials of degree d_i-1, each of them has

$d_i - z$ non-key coefficients, and so the total number of non-key coefficients is $(d_i - z)\left(\frac{|\boldsymbol{m}|}{d_i - z} - \frac{|\boldsymbol{m}|}{d_{i-1} - z}\right)$. It is trivial to verify that the two numbers are equal and so there is exactly enough coefficients to encode. Note that the specific way to map the coefficients is not important and any 1-1 mapping suffices. Finally, evaluate each polynomial at n non-zero points and store an evaluation of each polynomial at each node. This completes the scheme. Note that indeed the total number of polynomials is $\sum_{i=1}^{|\mathcal{D}|} p_i = \frac{|\boldsymbol{m}|}{d_{|\mathcal{D}|} - z} = \frac{|\boldsymbol{m}|}{k} = b$, implying that the scheme is rate-optimal.

B. Decoding

For any $d_i \in \mathcal{D}$, we describe the decoding algorithm of the scheme when d_i nodes are available. It achieves the optimal decoding bandwidth, and since $d_{|\mathcal{D}|} = n - r$ it implies that the scheme meets the reliability requirement. We first interpolate all polynomials of degree $d_i - 1$. After that for all polynomials of degree $d_{i-1} - 1$, their coefficients of degree larger than $d_i - 1$ are known (as they are encoded in the coefficients of the polynomials of degree $d_i - 1$) and so they can be interpolated. In general, for $j \leq i$, once the polynomials of degree between $d_j - 1$ and $d_i - 1$ are interpolated, then for the polynomials of degree $d_{j-1} - 1$, their coefficients of degree larger than $d_i - 1$ are known by construction and so they can be interpolated. Therefore we can successively interpolate the polynomials of higher degree until the polynomials of degree $d_1 - 1$ are interpolated and so the message symbols are decoded. The total number of $\mathbb{F}_q$ symbols communicated is $d_i \sum_{j=1}^{i} p_j = d_i \frac{|\boldsymbol{m}|}{d_i - z}$. By Theorem 1, the decoding bandwidth is at least

$$|\boldsymbol{m}| + \frac{kbz}{d_i - z} = kb + \frac{kbz}{d_i - z} = kb\left(1 + \frac{z}{d_i - z}\right) = \frac{d_i |\boldsymbol{m}|}{d_i - z}$$

$\mathbb{F}_q$ symbols. Therefore the optimal bandwidth is achieved.

C. Secrecy

We show that the scheme is secure against z eavesdropping nodes. At a high level, each polynomial individually can be viewed as a generalized Shamir's scheme, and so each polynomial individually is secure. The main idea is to show that if these polynomials are combined, the resulting scheme is still secure. We first prove that the generalized Shamir's scheme is indeed a valid secret sharing scheme and so is secure.

Theorem 2: The following is an $(n, n - r - z, r, z)$ rate-optimal secret sharing scheme: For $q > n$, let $m_1, \ldots, m_{n-z-r}$ be message symbols over $\mathbb{F}_q$. Construct a polynomial $f(x)$ of degree $n - r - 1$ over $\mathbb{F}_q$, whose degree 0 to $z - 1$ coefficients are random keys, and the degree z to $n - r - 1$ coefficients are m_1 to m_{n-r-z}. Evaluate $f(x)$ at n distinct non-zero points and assign one evaluation to each party.

Proof: We will prove the theorem assuming a slightly more general $f(x)$. Specifically, we allow $f(x)$ to be any degree-$(n - r - 1)$ polynomial such that z of its coefficients of *consecutive* degrees are random keys, and the remaining coefficients are m_1 to m_{n-r-z}. Such a scheme generalizes Shamir's scheme as a special case when $n - r - z = 1$ and the message m_1 is set to be the constant coefficient. The proof below also follows the same line as the proof of Shamir's scheme [20].

The scheme is reliable because $f(x)$ can be interpolated from any $n - r$ evaluations so that the message symbols can be decoded. To prove the secrecy of the scheme, suppose that the degree i to $i + z - 1$ coefficients of $f(x)$ are random keys $k_1, \ldots, k_z$, and that an adversary observes $C_1, \ldots, C_z$, which are the evaluations of $f(x)$ at z points $x_1, \ldots, x_z$. Then for every candidate value of the message symbols $(m'_1, \ldots, m'_{n-r-z})$, there is one and only one candidate value of the keys $(k'_1, \ldots, k'_z)$ such that the evaluation of the corresponding polynomial meets the observation of the adversary. This is because given $(m'_1, \ldots, m'_{n-r-z})$ and $C_1, \ldots, C_z$, one knows z evaluations of the polynomial $k'_1 x^i + k'_2 x^{i+1} + \ldots + k'_z x^{i+z-1}$, denoted by $D_1, \ldots, D_z$. Then by dividing D_j by x_j^i, $j \in [z]$, z evaluations of the polynomial $k'_1 + k'_2 x + \ldots + k'_z x^{z-1}$ are known. Therefore the unique $(k'_1, \ldots, k'_z)$ are obtained by interpolating this degree-$(z - 1)$ polynomial. By construction, since every value of the keys are equally likely, the adversary cannot deduce any information about the message. □

The following lemma shows that combining two secure schemes is still secure as long as the keys used in the schemes meet certain independence condition.

Lemma 1: Consider random variables M_1, M_2, K_1, K_2 such that K_2 is independent of $\{M_1, K_1\}$. For $i = 1, 2$ Let F_i be a deterministic function of M_i, K_i. If $I(M_1; F_1) = 0$ and $I(M_2; F_2) = 0$, then $I(M_1; F_1, F_2) = 0$. In addition, if K_1 is independent of M_2, then $I(M_1, M_2; F_1, F_2) = 0$.

Proof: We start with the first statement. Since F_2 is a function of K_2, M_2 but K_2 is independent of $\{M_1, K_1, F_1\}$, it follows that F_2 is independent of $\{M_1, K_1, F_1\}$ conditioning on M_2, implying the Markov chain $\{M_1, K_1, F_1\} \to M_2 \to F_2$. Therefore, $I(M_1, K_1, F_1, M_2; F_2) = I(M_2; F_2) = 0$, i.e., F_2 and $\{M_1, K_1, F_1, M_2\}$ are independent. Hence $I(M_1; F_1, F_2) = I(M_1; F_2) + I(M_1; F_1|F_2) \overset{(a)}{=} I(M_1; F_1|F_2) \overset{(b)}{=} I(M_1; F_1) = 0$, where (a) and (b) follows from the fact that F_2 is independent from $\{M_1, F_1\}$.

To prove the second statement, note that since K_1 is independent of M_2 and that F_1 is a function of M_1, K_1, we have the Markov Chain $M_2 \to M_1 \to F_1$, by which it follows that $I(M_1, M_2; F_1) = I(M_1; F_1) = 0$. Similarly because K_2 is independent of $\{M_1, K_1, F_1\}$ and that F_2 is a function of M_2, K_2, we have the Markov Chain $\{M_1, F_1\} \to M_2 \to F_2$. By this chain it follows that $I(M_1, F_1, M_2; F_2) = I(M_2; F_2) = 0$, i.e., $\{M_1, F_1, M_2\}$ is independent of F_2. Therefore $I(M_1, M_2; F_2|F_1) = 0$ and so $I(M_1, M_2; F_1, F_2) = I(M_1, M_2; F_1) + I(M_1, M_2; F_2|F_1) = 0$. □

Suppose that the adversary compromises z nodes and obtains z evaluations of each polynomial. Consider the i-th polynomial in the order that we define them, let $\boldsymbol{f}_i$ denote the adversary's observation of this polynomial, let $\boldsymbol{k}_i$ denote the key coefficients of this polynomial and let $\boldsymbol{m}_i$ denote the non-key coefficients. By Theorem 2 we have

$$I(\boldsymbol{m}_i; \boldsymbol{f}_i) = 0, \quad i = 1, \ldots, b. \tag{12}$$

Consider the first p_1 polynomials which are polynomials of the highest degree $d_1 - 1$. By construction, $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{p_1}$ exactly encode the message $\boldsymbol{m}$. We invoke Lemma 1 by regarding $\boldsymbol{m}_1, \boldsymbol{k}_1, \boldsymbol{f}_1, \boldsymbol{m}_2, \boldsymbol{k}_2$ and $\boldsymbol{f}_2$ as M_1, K_1, F_1, M_2, K_2 and F_2. By the second statement of the lemma it follows that $I(\boldsymbol{m}_1, \boldsymbol{m}_2; \boldsymbol{f}_1, \boldsymbol{f}_2) = 0$. Inductively, for $1 < i < p_1$, suppose that $I(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_i; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_i) = 0$. We regard $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_i\}$ as M_1, $\{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_i\}$ as K_1, $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_i\}$ as F_1, and regard $\boldsymbol{m}_{i+1}, \boldsymbol{k}_{i+1}, \boldsymbol{f}_{i+1}$ as M_2, K_2, F_2. It follows from Lemma 1 that $I(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{i+1}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{i+1}) = 0$. By induction we have $I(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{p_1}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{p_1}) = 0$.

We then regard $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{p_1}\} \triangleq \boldsymbol{m}$ as M_1, $\{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{p_1}\}$ as K_1, $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{p_1}\}$ as F_1, and regard $\boldsymbol{m}_{p_1+1}$, $\boldsymbol{k}_{p_1+1}$, $\boldsymbol{f}_{p_1+1}$ as M_2, K_2, F_2. Then it follows from the first statement of Lemma 1 that $I(\boldsymbol{m}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{p_1+1}) = 0$. Inductively, for $p_1 < i < b$, suppose that $I(\boldsymbol{m}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_i) = 0$. We regard $\boldsymbol{m}$ as M_1, $\{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_i\}$ as K_1, $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_i\}$ as F_1, and regard $\boldsymbol{m}_{i+1}, \boldsymbol{k}_{i+1}, \boldsymbol{f}_{i+1}$ as M_2, K_2, F_2. By Lemma 1 we have $I(\boldsymbol{m}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{i+1}) = 0$. By induction it follows that $I(\boldsymbol{m}; \boldsymbol{f}_1, \ldots, \boldsymbol{f}_b) = 0$, implying that the adversary learns no information about the message $\boldsymbol{m}$. This completes the proof and we have the following theorem.

Theorem 3: Let $\mathcal{D} \subset \{n - r, n - r + 1, \ldots, n\}$, the encoding scheme constructed in Section IV-A is a rate-optimal (n, k, r, z) secret sharing scheme. The scheme achieves the optimal decoding bandwidth when d nodes participate in decoding, universally for all $d \in \mathcal{D}$.

D. Discussion

We remark on some other important advantages and properties of our construction. Firstly, the scheme also achieves the *optimal number of symbol-reads from disks* in decoding. To see this, notice that the lower bound (7) on communication overhead is also a lower bound on the number of Q-ary symbols that need to be read from disks during decoding. The number of symbol-reads in the proposed scheme equals to the amount of communication. Therefore our scheme achieves the lower bound and hence is optimal. Secondly, compared to most existing schemes which decode from the minimum number of $n - r - z$ nodes, our scheme allows all available nodes (or more flexibly, any $d \in \mathcal{D}$ nodes) to participate in decoding and hence can help balance the load at the disks and achieves a higher degree of parallelization. Thirdly, the encoding and decoding of the scheme are similar to that of Shamir's scheme and therefore are efficient and practical. Particularly, the scheme works over the same field as Shamir's scheme. Fourthly, the preprocessing functions only rely on $d = |I|$ instead of I, further simplifying implementation. Fifthly, in practice when a user connects to more nodes, the increased latency may offset the benefit of the reduced bandwidth. One can overcome this issue by avoiding connections to nodes of large latency. If the latency of the nodes are not known a prior, one can start with connecting to all nodes and downloading the evaluations of the polynomials in decreasing order of degrees. If some nodes do not respond in time, consider them as not available and switch adaptively to the mode of decoding from a smaller number of nodes. Finally, the construction is flexible in the parameters, i.e., it works for arbitrary values of n, r and z and $\mathcal{D}$.

1) Connection to Other Schemes: An important idea in our scheme is to construct multiple correlated polynomials of different degrees in order to facilitate decoding when different number of nodes are available. Similar ideas also appear in the schemes in [23] and [27]. The main technique that enables the improvement of our schemes is a more careful and flexible design of the numbers and degrees of the polynomials, as well as the arrangement of their coefficients.

Our scheme maps the high-degree coefficients of the higher degree polynomials into the coefficients of the lower degree polynomials, whereas the specific coefficient mapping is not important and any 1-1 mapping suffices. Additionally, for all polynomials in our scheme, the z lowest degree coefficients are independent keys. However, in general this is not necessary: in any polynomial, we can choose any consecutive z coefficients to be independent keys, and use the remaining coefficients to encode information (i.e., message symbols and coefficients of higher degree polynomials). The resulting scheme is a still valid (see the proof of Theorem 2) and achieves the optimal decoding bandwidth universally. Under this observation, we note that our scheme generalizes the scheme in a recent independent work [2]. Particularly, our scheme is equivalent to the scheme in [2] if we require a specific coefficient mapping and let the z highest (instead of lowest) coefficients of all polynomial to be keys.[2] Refer to the Appendix for more discussion on the connection, where we interpret an example of the scheme in [2] under our framework.

In what follows we discuss the the advantage of our framework which allows the flexibility in choosing the coefficient mapping and the positions of the keys in a polynomial.

2) Coefficient Mapping: As discussed above we can choose the coefficient mapping to be any 1-1 mapping. The flexibility in choosing the specific mapping is helpful in practice. Particularly, it is possible to improve the (computational) encoding complexity of the scheme substantially by choosing a mapping that maintains the order of the coefficients. Refer to Figure 2b for an example. We need to compute $m_4x + m_5x^2 + m_6x^3$ in evaluating $g(x)$, and we can reuse this computation in evaluating $f(x)$, because $f(x)$ contains the same run of consecutive coefficients $m_4x^4 + m_5x^5 + m_6x^6$. This for example will save 2 multiplications and 2 additions.

3) Arrangement of Keys: We remark that choosing the lowest degree coefficients to be keys has several practical advantages. Decoding the scheme involves sequentially interpolating the polynomials through multiple iterations, which can lead to undesirable delay especially when $|\mathcal{D}|$ is large. To mitigate this issue, we wish to decode the message symbols "on the fly" in each iteration. Specifically, if d nodes are available, then each time a polynomial is interpolated, exactly d new message and/or key symbols are decoded. Since the number of symbols decoded in each interpolation, the total number of message symbols and the total number of key symbols to be decoded, are all fixed there is a trade-off between the decoding order of the key and message symbols.

[2]The scheme in [2] also lets a node evaluate all polynomials at the same point, whereas this is not necessary in our framework.

The location of keys in the polynomials plays a crucial role in this trade-off.

Formally, let $\beta(i)$ be the number of message symbols decoded after i iterations, i.e., after i polynomial interpolations. The optimal trade-off is to maximize $\beta(i)$ for every i. We first derive a simple upper bound of $\beta(i)$. Note that by the time that i polynomials have been interpolated, di symbols are decoded and among them there are at least zi keys since each polynomial introduces z independent keys for secrecy. Therefore $\beta(i) \leq (d-z)i$.

Our scheme achieves this upper bound for all i. For example, in Figure 2b, if $d = 3$ nodes are available, then decoding involves 3 iterations and each iteration outputs $d - z = 2$ message symbols. In general, each iteration in decoding our scheme outputs $d - z$ message symbols. This is because by construction, only coefficients of degree higher than $d_{|\mathcal{D}|} = n - r > z$ will be mapped to the coefficients of the lower degree polynomials. Since we place the keys in the z lowest degree coefficients, they are never mapped. Therefore for any polynomial, its coefficients of degree larger than $z-1$ encode message symbols. When a polynomial is interpolated, exactly z keys are decoded and the remaining $d - z$ symbols decoded are message symbols.

Achieving $\beta(i) = (d-z)i$ implies that at any moment during the decoding process, our scheme always decodes the maximum number of message symbols. In other words the decoding delay, measured in the number of iterations, averaged over all message symbols, is minimized. Moreover, the fact that each iteration decodes a fixed number of $d - z$ new message symbols may be helpful for implementation. On the other hand, note that choosing the z highest degree coefficients to be keys implies that the keys will be mapped to the coefficients of lower degree polynomials. Hence the keys will be decoded earlier than necessary (since lower degree polynomials are interpolated earlier) and it is not possible to achieve the optimal trade-off. Consider the example in Figure 2b, if we switch the keys to high degree coefficients, then the polynomials are $f(x) = m_1 + m_2x + m_3x^2 + m_4x^3 + m_5x^4 + m_6x^5 + k_1x^6$, $g(x) = m_5 + m_6x + k_1x^2 + k_2x^3$ and $h(x) = m_4 + k_2x + k_3x^2$. In the case that $d = 4$ nodes are available, only 2 message symbols m_5, m_6 are decoded in the first iteration and the remaining 4 message symbols are decoded in the second (last) iteration. In comparison, the original scheme performs better by decoding 3 message symbols in each iteration. Finally, we remark that decoding the maximum number of message symbols on the fly is also beneficial in terms of partial decoding, i.e., decoding a subset of message symbols. In this case decoding can finish early if all symbols of interest are decoded, and our scheme maximizes the chance of finishing early.

Finally, we remark that using the random coding argument, we can design another secret sharing scheme that achieves the optimal decoding bandwidth universally [10]. However, such a scheme requires a significantly lager field size.

V. Construction From Reed-Solomon Codes

In this section we present another rate-optimal secret sharing scheme that achieves the optimal decoding bandwidth when all n nodes are available. The scheme is flexible in the parameters and hence is flexible in rate. The scheme is directly related to Reed-Solomon codes. Particularly, the encoding matrix of the scheme is a generator matrix of Reed-Solomon codes, and so the scheme can be decoded as Reed-Solomon codes. This is an advantage over the scheme in the previous section, which requires recursive decoding. The scheme also provides a stronger level of reliability in the sense that it allows decoding even if more than r shares are partially erased. On the other hand, unlike the previous scheme, this scheme does not achieve the optimal decoding bandwidth universally, but rather only for $d = n - r$ and $d = n$. However, we remark that the case that the n nodes are available is particularly important because it correspond to the best case in terms of decoding bandwidth and is arguably the most relevant case for the application of distributed storage, where the storage nodes are usually highly available.

A. Encoding

Fix $k = n - r - z$, let $q > n(k + r)$ be a prime power, and let the share alphabet be $\mathcal{Q} = \mathbb{F}_q^{k+r}$. Note that each share is a length $k + r$ vector over $\mathbb{F}_q$. For $j = 1, \ldots, n$, denote the j-th share by $c_j = (c_{1,j}, \ldots, c_{k+r,j})$, where $c_{i,j} \in \mathbb{F}_q$. The secret message $\boldsymbol{m}$ is k symbols over $\mathcal{Q}$ and therefore can be regarded as a length-$k(k+r)$ vector over $\mathbb{F}_q$, denoted by $(m_1, \ldots, m_{k(k+r)})$. The encoder generates keys $\boldsymbol{k} = (k_1, \ldots, k_{kz}) \in \mathbb{F}_q^{kz}$ and $\boldsymbol{k}' = (k'_1, \ldots, k'_{rz}) \in \mathbb{F}_q^{rz}$ independently and uniformly at random. The encoding scheme is linear over $\mathbb{F}_q$, and is described by an encoding matrix G over $\mathbb{F}_q$:

$$(c_{1,1}, \ldots, c_{1,n}, \ldots, c_{k+r,1}, \ldots, c_{k+r,n}) = (m_1, \ldots, m_{k(k+r)}, k_1, \ldots, k_{kz}, k'_1, \ldots, k'_{rz})G. \quad (13)$$

Note that G has $k(k+r)+kz+rz = nk+rz$ rows and has $n(k+r)$ columns. In the following we discuss the construction of G based on a Vandermonde matrix. We start with some notation. Let $\alpha_1, \ldots, \alpha_{n(k+r)}$ be distinct non-zero elements of $\mathbb{F}_q$, and let $v_{ij} = \alpha_j^{i-1}$, $i = 1, \ldots, nk+rz$, $j = 1, \ldots, n(k+r)$, then $V = (v_{ij})$ is a Vandermonde matrix of the same size as G. Suppose $\boldsymbol{f} = (f_0, \ldots, f_i)$ is an arbitrary vector with entries in $\mathbb{F}_q$, we denote by $\boldsymbol{f}[x]$ the polynomial $f_0 + f_1x + \ldots + f_ix^i$ over $\mathbb{F}_q$ with indeterminate x. We construct a set of polynomials as follows:

$$\boldsymbol{f}_i[x] = x^{i-1} \quad i = 1, \ldots, kn, \quad (14)$$

$$\boldsymbol{f}_{kn+i}[x] = x^{i-1} \prod_{j=1}^{kn} (x - \alpha_j) \quad i = 1, \ldots, rz. \quad (15)$$

Let $\boldsymbol{f}_i, i = 1, \ldots, kn+rz$ be the length-$(kn+rz)$ vectors over $\mathbb{F}_q$ corresponding to the polynomials. Stack the $\boldsymbol{f}_i$'s to obtain a square matrix of size $(kn + rz)$:

$$T = \begin{pmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_{kn+rz} \end{pmatrix}$$

Finally, we complete the construction by setting

$$G = TV.$$

Example 1: Consider the setting that $n = 3, r = 1, z = 1$ *and* $k = n - r - z = 1$. *Let* $q = 7$ *and* $\mathcal{Q} = \mathbb{F}_q^2$. *Then* $\boldsymbol{m} = (m_1, m_2)$, $\boldsymbol{k} = (k_1)$ *and* $\boldsymbol{k}' = (k_1')$. *Construct a Vandermonde matrix over* $\mathbb{F}_q$ *as*

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 2 & 4 & 1 \\ 1 & 1 & 6 & 1 & 6 & 6 \end{pmatrix}. \tag{16}$$

Construct polynomials $\boldsymbol{f}_1[x] = 1$, $\boldsymbol{f}_2[x] = x$, $\boldsymbol{f}_3[x] = x^2$ *and* $\boldsymbol{f}_4[x] = (x-1)(x-2)(x-3) = 1 + 4x + x^2 + x^3$.

Therefore,

$$T = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \\ \boldsymbol{f}_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 4 & 1 & 1 \end{pmatrix},$$

and the encoding matrix is given by

$$G = TV = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 2 & 4 & 1 \\ 0 & 0 & 0 & 6 & 3 & 4 \end{pmatrix}.$$

The properties of G are discussed in the following lemma.

Lemma 2: Regard G *as a block matrix*

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix},$$

where G_{11} *has size* $kn \times kn$, G_{12} *has size* $kn \times rn$, G_{21} *has size* $rz \times kn$, *and* G_{22} *has size* $rz \times rn$. *Then,*

(i) *Any* $(n-r)(k+r)$ *columns of* G *are linearly independent.*
(ii) G_{11} *is a Vandermonde matrix.*
(iii) $G_{21} = 0$.
(iv) *Any* rz *columns of* G_{22} *are linearly independent.*

Proof: By construction, the polynomials $\boldsymbol{f}_i[x], i = 1, \ldots, kn + rz$ have distinct degrees and therefore are linearly independent. Therefore the rows of T are linearly independent and so T is full rank. This implies that the row space of G is the same as the row space of V. The row space of V is a linear $(nk + nr, nk + rz)$ Reed-Solomon code because that V is a Vandermonde matrix. Note that $nk + rz = (n-r)(k+r)$, and so the row space of G is a $(nk + nr, (n-r)(k+r))$ Reed-Solomon code. This proves (i).

To prove (ii), note that by (14), the first kn rows of G are exactly the first kn rows of V. Therefore G_{11} is a Vandermonde matrix.

To prove (iii), note that by construction the (i, j)-th entry of G_{21} equals $\boldsymbol{f}_{kn+i}[\alpha_j]$. By (15), α_j is a root of $\boldsymbol{f}_{kn+i}[x]$, for $i = 1, \ldots, rz$, $j = 1, \ldots, kn$. Hence $G_{21} = 0$.

Finally we prove (iv). By construction the (i, j)-th entry of G_{22} equals

$$\boldsymbol{f}_{kn+i}[\alpha_{kn+j}] = \alpha_{kn+j}^{i-1} \prod_{l=1}^{kn} (\alpha_{kn+j} - \alpha_l) = \alpha_{kn+j}^{i-1} \boldsymbol{f}^*[\alpha_{kn+j}], \tag{17}$$

where $\boldsymbol{f}^*[x] = \prod_{l=1}^{kn}(x - \alpha_l)$. Since $\alpha_1, \ldots, \alpha_{(k+r)n}$ are distinct elements, it follows that $\boldsymbol{f}^*[\alpha_{kn+j}] \neq 0$, for $j = 1, \ldots, rn$. Let $1 \leq j_1 < j_2 < \ldots < j_{rz} \leq rn$ and consider the submatrix formed by the j_1-th,…,j_{rz}-th columns of G_{22}. By (17), the l-th column of the submatrix are formed by consecutive powers of α_{kn+j_l}, scaled by $\boldsymbol{f}^*[\alpha_{kn+j_l}]$. Therefore the determinant of the submatrix is $\prod_{l=1}^{rz} \boldsymbol{f}^*[\alpha_{kn+j_l}] \prod_{1 \leq u < v \leq rz} (\alpha_{kn+j_v} - \alpha_{kn+j_u}) \neq 0$. This shows that any rz columns of G_{22} are linearly independent. □

B. Decoding

We describe the decoding procedure for two cases: 1) $|I| = n$, i.e., all nodes are available, and 2) $|I| < n$. First consider the case that $|I| = n$, i.e., $I = [n]$. In order to decode, for this case it suffices to read and communicate the first k symbols over $\mathbb{F}_q$ from each share. Formally, the user downloads $\boldsymbol{e} = (c_{1,1}, \ldots, c_{1,n}, \ldots, c_{k,1}, \ldots, c_{k,n})$. By Lemma 2(ii), G_{11} is invertible. Denote the inverse of G_{11} by G_{11}^{-1}, then the secret can be recovered by

$$\boldsymbol{e} G_{11}^{-1} \overset{(e)}{=} (m_1, \ldots, m_{k(k+r)}, k_1, \ldots, k_{kz}),$$

where (e) follows from (13) and Lemma 2(iii). The decoding process involves communicating kn symbols from $\mathbb{F}_q$. The communication overhead is kz symbols over $\mathbb{F}_q$ or $\frac{kz}{k+r} = \frac{kz}{n-z}$ $\mathcal{Q}$-ary symbols, which achieves the lower bound (7) and therefore is optimal.

Next consider the case that $n - r \leq |I| < n$. Select an arbitrary subset I' of I of size $n - r$, and download the complete share stored by the nodes in I'. Hence, the downloaded information $\boldsymbol{e}$ is a length-$(n-r)(k+r)$ vector over $\mathbb{F}_q$. By Lemma 2(i), it follows that any $(n-r)(k+r)$ columns in G are linearly independent and therefore the submatrix formed by these columns is invertible. The secret $\boldsymbol{m}$ can then be recovered by multiplying $\boldsymbol{e}$ with the inverse. An alternative way to decode the secret is to notice that G is an encoding matrix of a $(nk + nr, nk + rz)$ Reed-Solomon code over $\mathbb{F}_q$. Therefore one may employ the standard decoder of Reed-Solomon code to correct any $r(k + r)$ erasures or $\lfloor r(k + r)/2 \rfloor$ errors of symbols over $\mathbb{F}_q$. Note that when at most r nodes are unavailable, we regard their shares as erased and there are at most $r(k+r)$ erasures of symbols over $\mathbb{F}_q$, and therefore can be corrected. In general, any $r(k + r)$ erasures or $\lfloor r(k + r)/2 \rfloor$ errors are correctable even if they occur to more than r nodes. The decoding process involves communicating $nk + rz$ symbols of $\mathbb{F}_q$. The communication overhead is $(n-r)(k+r) - k(k+r) = z(k+r)$ symbols over $\mathbb{F}_q$, or z symbols over $\mathcal{Q}$, which achieves the lower bound (7) if and only if $|I| = n - r$.

C. Analysis

Theorem 4: The encoding scheme constructed in Section V-A is a rate-optimal (n, k, r, z) *secret sharing scheme. The scheme achieves the optimal decoding bandwidth when* d *nodes participate in decoding, for* $d = n$ *or* $d = n - r$.

Proof: We need to verify that the encoding scheme meets the reliability requirement and the security requirement of a secret sharing scheme, formally defined in Definition 1. Explicit decoding scheme and its communication overhead are discussed in Section V-B and therefore the reliability requirement is met. The scheme is rate-optimal because

$k = n-r-z$. We only need to show that the encoding scheme is secure. To this end, we first show that $H(\boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{c}_I, \boldsymbol{m}) = 0$, for all I such that $|I| = z$. In other words, the random symbols generated by the encoder are completely determined by $\boldsymbol{c}_I$ and the secret. Denote the submatrix formed by the first $k(k+r)$ rows of G by G_{top} and the submatrix formed by the remaining $(k+r)z$ rows of G by G_{low}. Consider any $I = \{i_1, \ldots, i_z\}$, and let $\boldsymbol{c}_I = (c_{1,i_1}, \ldots, c_{1,i_z}, \ldots, c_{k+r,i_1}, \ldots, c_{k+r,i_z})$. It then follows from (13) that

$$\begin{aligned}\boldsymbol{c}_I &= (m_1, \ldots, m_{k(r+k)})G_{\text{top},I} \\ &\quad + (k_1, \ldots, k_{kz}, k'_1, \ldots, k'_{rz})G_{\text{low},I},\end{aligned}$$

where $G_{\text{top},I}$ is the submatrix formed by the subset of columns in $\{i+j | i \in I,\ j = 0, n, \ldots, (k+r-1)n\}$ of G_{top}, and $G_{\text{low},I}$ is the submatrix formed by the same subset of columns of G_{low}. Therefore, written concisely,

$$(\boldsymbol{k}\ \boldsymbol{k}')G_{\text{low},I} = \boldsymbol{c}_I - \boldsymbol{m}G_{\text{top},I}. \tag{18}$$

To study the rank of $G_{\text{low},I}$, note that it is a square matrix of size $(k+r)z$, and we regard it as a block matrix

$$G_{\text{low},I} = \begin{pmatrix} G'_{11} & G'_{12} \\ G'_{21} & G'_{22} \end{pmatrix}, \tag{19}$$

where G'_{11} has size $kz \times kz$, G'_{12} has size $kz \times rz$, G'_{21} has size $rz \times kz$ and G'_{22} has size $rz \times rz$. By Lemma 2(ii), G'_{11} is a block of a Vandermonde matrix and therefore is invertible. By Lemma 2.(iii), $G'_{21} = 0$. Denote $\boldsymbol{c}_I - \boldsymbol{m}G_{\text{top},I}$ by $(u_1, \ldots, u_{(k+r)z})$, then the above two facts together with (18) imply that

$$\boldsymbol{k} = (u_1, \ldots, u_{kz})G'^{-1}_{11} \tag{20}$$

Therefore $\boldsymbol{k}$ is a deterministic function of $\boldsymbol{m}$ and $\boldsymbol{c}_I$. It follows from (18) that

$$\boldsymbol{k}'G'_{22} = (u_{kz+1}, \ldots, u_{(k+r)z}) - \boldsymbol{k}G'_{12}.$$

By Lemma 2(iv), G'_{22} is invertible and therefore

$$\boldsymbol{k}' = \left((u_{kz+1}, \ldots, u_{(k+r)z}) - \boldsymbol{k}G'_{12}\right) G'^{-1}_{22}. \tag{21}$$

This shows that $\boldsymbol{k}'$ is a deterministic function of $\boldsymbol{k}$, $\boldsymbol{c}_I$ and $\boldsymbol{m}$, and so

$$H(\boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{c}_I, \boldsymbol{m}) = 0. \tag{22}$$

It then follows that,

$$\begin{aligned} H(\boldsymbol{m}) - H(\boldsymbol{m}|\boldsymbol{c}_I) &= I(\boldsymbol{m}; \boldsymbol{c}_I) \\ &= H(\boldsymbol{c}_I) - H(\boldsymbol{c}_I|\boldsymbol{m}) \\ &\overset{(f)}{\leq} z - H(\boldsymbol{c}_I|\boldsymbol{m}) \\ &\overset{(g)}{=} z - H(\boldsymbol{c}_I|\boldsymbol{m}) + H(\boldsymbol{c}_I|\boldsymbol{m}, \boldsymbol{k}, \boldsymbol{k}') \\ &= z - I(\boldsymbol{c}_I; \boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{m}) \\ &= z - H(\boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{m}) + H(\boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{c}_I, \boldsymbol{m}) \\ &\overset{(h)}{=} z - H(\boldsymbol{k}, \boldsymbol{k}'|\boldsymbol{m}) \\ &\overset{(i)}{=} z - H(\boldsymbol{k}, \boldsymbol{k}') \\ &\overset{(j)}{=} z - z = 0, \end{aligned} \tag{23}$$

where (f) is due to $|I| = z$; (g) is due to the fact that $\boldsymbol{c}_I$ is a function of $\boldsymbol{m}$, $\boldsymbol{k}$ and $\boldsymbol{k}'$; (h) is due to (22); (i) is due to the fact that $\boldsymbol{k}, \boldsymbol{k}'$ are independent of $\boldsymbol{m}$; and (j) follows from the fact that $\boldsymbol{k}, \boldsymbol{k}'$ are uniformly distributed. Therefore $H(\boldsymbol{m}) = H(\boldsymbol{m}|\boldsymbol{c}_I)$ and the security requirement is met. This completes the proof that the encoding scheme is a valid secret sharing scheme. □

Theorem 4 shows that the proposed secret sharing scheme is optimal in terms of storage usage and is optimal in terms of best-case (i.e., $|I| = n$) communication overhead. Compared to the scheme in the previous section, this scheme has advantages in terms of implementation and error correction because decoding the scheme is equivalent to decoding standard Reed-Solomon codes. The scheme also provides a stronger level of reliability in the sense that it allows decoding even if more than r shares are partially erased. Similar to previous discussion, the scheme achieves the optimal number of symbol-reads from disks when $|I| = n$. Finally, in the scheme all operations are performed over the field $\mathbb{F}_q$, where $q > n(k+r)$. This requirement on the field size can be relaxed in the following simple way. Let β be the greatest common divisor of k and r, then instead of choosing $\mathcal{Q}$ to be $\mathbb{F}_q^{k+r}$, we can let $\mathcal{Q} = \mathbb{F}_q^{\frac{k}{\beta}+\frac{r}{\beta}}$, $\boldsymbol{m} = (m_1, \ldots, m_{\frac{k(k+r)}{\beta}})$, $\boldsymbol{k} = (k_1, \ldots, k_{\frac{kz}{\beta}})$ and $\boldsymbol{k}' = (k'_1, \ldots, k'_{\frac{rz}{\beta}})$. The resulting scheme is a rate-optimal $(n, k, r, z)_{\mathcal{Q}}$ secret sharing scheme with the same communication overhead function as the original scheme. For this modified construction, it is sufficient to choose any field size $q > n\frac{k+r}{\beta}$.

VI. Conclusions

In this paper we study the communication efficiency of secret sharing schemes in decoding. We prove an information-theoretic lower bound on the amount of information to be communicated during decoding, and show that the decoding bandwidth decreases as d, the number of nodes that participate in decoding, increases. We prove that the bound is uniformly tight by designing a secret sharing scheme that achieves the optimal decoding bandwidth universally for all valid d. The scheme is simple and is efficient in both space and computation. We construct another secret sharing scheme that achieves the optimal decoding bandwidth when all nodes are available. The scheme has an advantage in implementation because its codewords form the Reed-Solomon codes. In the application of distributed storage, the proposed communication efficient secret sharing schemes also improve disk access efficiency. There are a number of interesting open problems: 1) in the application of distributed storage, how can one construct codes that are communication efficient in terms of both decoding and repair? We note that very recently, [17], [26] have addressed this problem for certain sets of parameters. 2) how to generalize the results to other (non-threshold) access structures? and 3) is it possible to extend the schemes and ideas in the paper to improve the communication efficiency of other secure protocols that use secret sharing schemes as building blocks?

Appendix

We describe a connection between our scheme in Section IV-A and the Staircase code in [2]. Specifically, we will interpret the Staircase code example in [2, Sec. 5.1]

under our framework described in Section IV-A. The general case follows similarly.

Example of the Staircase code: In [2, Sec. 5.1], Bitar and El Rouayheb describe a construction of a $(n = 4, k = 1, r = 2, z = 1)$ rate-optimal scheme that achieves the optimal decoding bandwidth when d nodes participate in decoding, for $d \in \{2, 3, 4\}$. Specifically, the scheme is over the field $\mathbb{F}_5^6$. Denote the message by $(m_1, \ldots, m_6)$, where each entry is a symbol from $\mathbb{F}_5$. Let $k_1, \ldots, k_6$ be random keys over $\mathbb{F}_5$. The encoder generates a matrix

$$M = \begin{bmatrix} m_1 & m_4 & k_1 & m_3 & m_6 & k_3 \\ m_2 & m_5 & k_2 & k_4 & k_5 & k_6 \\ m_3 & m_6 & k_3 & 0 & 0 & 0 \\ k_1 & k_2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and a Vandermode matrix

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \end{bmatrix}.$$

Note that the 0's in M form a staircase structure, hence its name. Finally, the encoder computes $C = VM$, and assigns the i-th row of C to node i as its share.

We now connect the above scheme to the one described in Section IV-A. Let us regard each column of M as a polynomial: $f_1(x) = m_1 + m_2x + m_3x^2 + k_1x^3$; $f_2(x) = m_4 + m_5x + m_6x^2 + k_2x^3$; $f_3(x) = k_1 + k_2x + k_3x^2$; $f_4(x) = m_3 + k_4x$; $f_5(x) = m_6 + k_5x$; and $f_6(x) = k_3 + k_6x$. Then multiplying the i-th row of V to the j-th column of M is equivalent to evaluating $f_j(x)$ at $x = i$. Therefore, the share assigned to node i is the evaluations of the polynomials $f_j(x)$, $j = 1, \ldots, 6$, at $x = i$. Now we observe that the polynomials $f_j(x)$'s are consistent with the ones defined in Section IV-A: Firstly, the degrees of the polynomials and the number of polynomials with respect to a degree are consistent with (11). Secondly, each polynomial introduces a new key, which is its highest degree coefficient,[3] for security. Finally, the high degree coefficients of the high degree polynomials are mapped to the coefficients of the low degree polynomials. Specifically, the degree-3 coefficients of $f_1(x)$ and $f_2(x)$ are mapped to the coefficients of $f_3(x)$; and the degree-2 coefficients of $f_1(x)$, $f_2(x)$ and $f_3(x)$ are mapped to the coefficients of $f_4(x)$, $f_5(x)$ and $f_6(x)$.

ACKNOWLEDGMENT

The authors thank an anonymous reviewer for pointing them to the paper by Wang and Wong [23].

REFERENCES

[1] A. Beimel, "Secret-sharing schemes: A survey," in *Coding and Cryptology*. Heidelberg, Germany: 2011, pp. 11–46. [Online]. Available: http://link.springer.com/chapter/10.1007%2F9783642209017_2

[2] R. Bitar and S. El Rouayheb, "Staircase codes for secret sharing with optimal communication and read overheads," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1396–1400.

[3] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. AFIPS*, 1979, pp. 313–317.

[4] G. R. Blakley and C. Meadows, "Security of ramp schemes," in *Proc. CRYPTO*, 1984, pp. 242–268.

[5] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 529–542, Mar. 1996.

[6] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[7] R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro, "On the size of shares for secret sharing schemes," *J. Cryptol.*, vol. 6, no. 3, pp. 157–193, 1993.

[8] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[9] M. Franklin and M. Yung, "Communication complexity of secure computation," in *Proc. ACM Symp. Theory Comput.*, 1992, pp. 699–710.

[10] W. Huang, M. Langberg, J. Kliewer, and J. Bruck. (May 2015). "Communication efficient secret sharing." [Online]. Available: https://arxiv.org/abs/1505.07515

[11] E. Karnin, J. Greene, and M. Hellman, "On secret sharing systems," *IEEE Trans. Inf. Theory*, vol. 29, no. 1, pp. 35–41, Jan. 1983.

[12] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k,n)–Threshold secret sharing scheme and its extension," in *Proc. Lect. Notes Comput. Sci.*, vol. 5222. 2008, pp. 455–470. [Online]. Available: http://link.springer.com/chapter/10.1007%2F9783540858867_31

[13] C.-P. Lai and C. Ding, "Several generalizations of Shamir's secret sharing scheme," *Int. J. Found. Comput. Sci.*, vol. 15, no. 2, pp. 455–458, 2004.

[14] R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," *Commun. ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[15] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6734–6753, Oct. 2011.

[16] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[17] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. (Mar. 2016). "Centralized repair of multiple node failures with applications to communication efficient secret sharing." [Online]. Available: https://arxiv.org/abs/1603.04822

[18] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 212–236, Nov. 2013.

[19] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," in *Proc. IEEE Globecom*, Dec. 2011, pp. 1–5.

[20] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[21] D. R. Stinson, "An explication of secret sharing schemes," *Design, Codes Cryptogr.*, vol. 2, no. 4, pp. 357–390, 1992.

[22] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.

[23] H. Wang and D. S. Wong, "On secret reconstruction in secret sharing schemes," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 473–480, Jan. 2008.

[24] H. Yamamoto, "Secret sharing system using (k, l, n) threshold scheme," *Electron. Commun. Jpn. (Part I: Commun.)*, vol. 69, no. 9, pp. 46–54, 1986.

[25] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (t,n) multi-secret sharing scheme," *Appl. Math. Comput.*, vol. 151, no. 2, pp. 483–490, Apr. 2004.

[26] M. Ye and A. Barg. (Apr. 2016). "Explicit constructions of high-rate mds array codes with optimal repair bandwidth." [Online]. Available: https://arxiv.org/abs/1604.00454

[27] Z. Zhang, Y. M. Chee, S. Ling, M. Liu, and H. Wang, "Threshold changeable secret sharing schemes revisited," *Theor. Comput. Sci.*, vol. 418, pp. 106–115, Feb. 2012.

[3] This is slightly different from our scheme: In Section IV-A we let the new keys be the z lowest (instead of highest) degree coefficients to improve the performance of decoding. See the discussion on the arrangement of keys in Section IV-D.

Wentao Huang received his B.Sc. in electrical engineering from Nanjing University of Posts and Telecommunications in 2008, the M.Sc. in communication and information systems from Shanghai Jiao Tong university in 2011, the M.Sc. in electrical engineering from the California Institute of Technology (Caltech) in 2013. He is now a Ph.D student with the Department of Electrical Engineering at Caltech. His current research interests include coding theory, security and information theory.

Michael Langberg (M'07–SM'15) received his B.Sc. in mathematics and computer science from Tel-Aviv University in 1996, and his M.Sc. and Ph.D. in computer science from the Weizmann Institute of Science in 1998 and 2003 respectively. Between 2003 and 2006, he was a postdoctoral scholar in the Electrical Engineering and Computer Science departments at the California Institute of Technology, and between 2007 and 2012 he was in the Department of Mathematics and Computer Science at The Open University of Israel. Prof. Langberg is currently an associate professor in the Department of Electrical Engineering at the State University of New York in Buffalo.

Prof. Langberg's research addresses the algorithmic and combinatorial aspects of information in communication, management, and storage; focusing on the study of information theory, coding theory, network communication and network coding, big data in the form of succinct data representation, and probabilistic methods in combinatorics. Prof. Langberg was an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY during the years 2012-2015 and is currently the Editor of the IEEE INFORMATION THEORY SOCIETY NEWSLETTER.

Jörg Kliewer (S'97–M'99–SM'04) received the Dipl.-Ing. (M.Sc.) degree in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 1993 and the Dr.-Ing. degree (Ph.D.) in electrical engineering from the University of Kiel, Germany, in 1999, respectively.

From 1993 to 1998, he was a research assistant at the University of Kiel, and from 1999 to 2004, he was a senior researcher and lecturer with the same institution. In 2004, he visited the University of Southampton, U.K., for one year, and from 2005 until 2007, he was with the University of Notre Dame, IN, as a Visiting assistant professor. From 2007 until 2013 he was with New Mexico State University, Las Cruces, NM, most recently as an associate professor. He is now with the New Jersey Institute of Technology, Newark, NJ, as an associate professor. His research interests span coding and information theory, graphical models, and statistical algorithms, which includes applications to networked communication and security, data storage, and biology.

Dr. Kliewer was the recipient of a Leverhulme Trust Award and a German Research Foundation Fellowship Award in 2003 and 2004, respectively. He was an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 until 2014, and since 2015 serves as an Area Editor for the same journal. He is also member of the editorial board of the IEEE INFORMATION THEORY SOCIETY NEWSLETTER since 2012.

Jehoshua Bruck (S'86–M'89–SM'93–F'01) is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and its applications to memory systems, including molecular memories (DNA), associative memories (the brain), and nonvolatile memories (flash).

Dr. Bruck received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, in 1989.

His industrial and entrepreneurial experiences include working with IBM Research where he participated in the design and implementation of the first IBM parallel computer; cofounding and serving as Chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that created the first virtualization solution for Network Attached Storage; as well as cofounding and serving as Chairman of XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system.

Dr. Bruck is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award and the IBM Outstanding Technical Achievement Award.