Secure RAID Schemes for Distributed Storage

Wentao Huang and Jehoshua Bruck
California Institute of Technology, Pasadena, USA
{whuang,bruck}@caltech.edu

Abstract

We propose secure RAID, i.e., low-complexity schemes to store information in a distributed manner that is resilient to node failures and resistant to node eavesdropping. We generalize the concept of systematic encoding to secure RAID and show that systematic schemes have significant advantages in the efficiencies of encoding, decoding and random access. For the practical high rate regime, we construct three XOR-based systematic secure RAID schemes with optimal or almost optimal encoding and decoding complexities, from the EVENODD codes and B codes, which are array codes widely used in the RAID architecture. The schemes can tolerate up to two node failures and two eavesdropping nodes. For more general parameters we construct systematic secure RAID schemes from Reed-Solomon codes, and show that they are significantly more efficient than Shamir's secret sharing scheme. Our results suggest that building "keyless", information-theoretic security into the RAID architecture is practical.

I. Introduction

In the RAID (Redundant Array of Independent Disks) architecture [17], [4], information is stored distributively among multiple nodes, such as an array of disks or a cluster of networked computers, in a redundant manner that is resilient to individual node failures. RAID improves the reliability, availability and performance of the system and has seen extensive applications over the decades [4], [9], [7].

Today, as distributed storage systems are increasingly being used to store critical as well as sensitive data, the challenge of protecting data confidentiality is imminent [8]. We propose *secure RAID*, which in addition to being failure-resilient, is also resistant to eavesdroppers compromising individual nodes. Specifically, we address the problem of storing a message among n nodes such that any $n-r$ nodes can decode the message but any coalition of z nodes cannot infer any information about the message. This problem was studied in the literature under the context of secret sharing [1], and rate-optimal schemes (i.e., schemes that store a message of maximum size given parameters n, r, z) are known such as Shamir's scheme [20] and its ramp version [2]. However, application of secret sharing schemes to distributed storage systems has been limited by their high complexities [13], [21], [19], [14]. Particularly, existing secret sharing schemes are significantly more intensive in terms of computation than their erasure code counterparts, such as Reed-Solomon [15] and EVENODD [3] codes, that are extensively employed in practical storage systems, notably for the RAID architecture.

We study the design of low-complexity schemes, termed *secure RAID schemes*, that have similar computational complexities as their erasure code counterparts and as such are suitable for the application of distributed storage. Codes for storage are typically encoded in a *systematic* manner, for better efficiency in encoding, decoding and random access (decoding partial message). In secure RAID, while storing the message in the clear is not allowed due to the secrecy requirement, we generalize the concept of systematic encoding and propose systematic secure RAID schemes. Refer to Fig. 1 for an example of a systematic scheme, which can optimally tolerate two node erasures and two eavesdropping nodes

For general parameters n, r and z, we present a systematic, rate-optimal scheme based on Reed-Solomon (RS) codes, and show that its computational complexity is significantly better than Shamir's scheme, which is also related to RS codes [16] but is not systematic. However, RS codes require computation over finite fields which complicates implementation and affects computational efficiency [3]. Designs of more efficient XOR-based array codes have been extensively researched, e.g., [3], [24], [5], [10]. Specifically, the codeword of an array code is a $t \times n$ array; each node stores a column of the array so erasure and distance are defined column-wise. Well-known families of MDS array codes suitable for RAID include the EVENODD [3] and B [24] codes. The generator matrices of these codes are "low-density" (sparse), and hence encoding them requires an optimal or almost optimal number of XOR operations.

We make several contributions in the design of array-based secure RAID schemes. We study the density of the generator matrix (defined similarly as the generator matrix of linear codes) of secure RAID schemes and prove a lower bound. The density characterizes the number of operations required by encoding. We construct three families of secure RAID schemes based on the B and EVENODD codes. Refer to Fig. 1 for an example. The schemes are XOR-based, rate-optimal, and have low or lowest density generator matrices. Specifically, the schemes can correct $r \leq 2$ node erasures and resist $z \leq 2$ eavesdropping nodes. In these schemes, encoding each bit of a message on average requires slightly more than $r + z = 4$ XORs and decoding each bit of a message when no erasure occurs on average requires $z = 2$ or slightly more XORs. We show that these encoding and decoding complexities are optimal or almost optimal.

Our results suggest that building "keyless", information-theoretic security into the RAID architecture is practical. Particularly, for Reed-Solomon, EVENODD or B coded distributed storage systems, extending them to employ the proposed secure RAID schemes requires only minor modification to the implementation, with small computational and therefore performance overhead.

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
u_1	u_2	u_3	u_4	u_5	u_6
$u_3 \oplus u_5 \oplus m_1$	$u_6 \oplus u_3 \oplus m_2$	$u_2 \oplus u_1 \oplus m_3$	$u_5 \oplus u_6 \oplus m_4$	$u_1 \oplus u_4 \oplus m_5$	$u_4 \oplus u_2 \oplus m_6$
$u_2 \oplus u_6 \oplus m_3 \oplus m_5$	$u_4 \oplus u_5 \oplus m_6 \oplus m_3$	$u_6 \oplus u_4 \oplus m_2 \oplus m_1$	$u_1 \oplus u_3 \oplus m_5 \oplus m_6$	$u_3 \oplus u_2 \oplus m_1 \oplus m_4$	$u_5 \oplus u_1 \oplus m_4 \oplus m_2$

Fig. 1: A secure RAID scheme constructed from the B codes [24]. Symbols are bits and operations are XORs. $m_1, ..., m_6$ are message bits and $u_1, ..., u_6$ are random key bits. The scheme is able to correct two node erasures and is secure against two eavesdropping nodes. The scheme is optimal in several senses. It has optimal rate and optimal field size. It follows a generalized systematic form: all keys are stored uncoded in the first row; all message bits are stored uncoded in the second row, each padded by an optimal number of two keys necessary to defeat two eavesdropping nodes; and the third row is redundant. The systematic form implies optimal decoding complexity as the message bits can be decoded by canceling the least amount of keys. The scheme is also optimal in terms of encoding complexity: every key and message bit is checked by an optimal number of two parities in the redundant (third) row necessary to correct two erasures. Schemes with similar optimal properties are constructed in Section VI-A for any length $p-1$, where $7 \leq p \leq 53$ is prime. Two infinite families of almost optimal schemes, which require only slightly more computation in encoding and decoding, are constructed in Section VI and V-A.

II. Setup and Definitions

We consider the problem of storing a message $\boldsymbol{m}$ in a distributed manner that is reliable against disk failures (i.e., erasures) and secret against eavesdroppers. Namely, for a storage system consisting of n nodes, the message is encoded into n shares, so that 1) $\boldsymbol{m}$ can be decoded from any $n-r$ shares, i.e., the erasure of any r shares can be corrected, and 2) any z shares do not reveal any information about the message, i.e., the shares are statistically independent of $\boldsymbol{m}$. More formally, let $\mathcal{Q}$ be a genric alphabet and let $[n] = \{1, ..., n\}$. For any index set $I \subset [n]$ and a vector $\boldsymbol{c} = (c_1, ..., c_n)$, let $\boldsymbol{c}_I = (c_i)_{i \in I}$. An $(n, k, r, z)_{\mathcal{Q}}$ secure RAID scheme is a randomized encoding function F that maps a secret message $\boldsymbol{m} \in \mathcal{Q}^k$ and a uniformly distributed random vector $\boldsymbol{u} \in \mathcal{Q}^v$, also referred to as *keys*, to the codeword $\boldsymbol{c} = F(\boldsymbol{m}, \boldsymbol{u}) \in \mathcal{Q}^n$, such that:

1) (Reliability) $\forall I \subset [n], |I| \geq n-r : \ H(\boldsymbol{m}|\boldsymbol{c}_I) = 0$, implying a decoding function $D_I : \mathcal{Q}^{|I|} \to \mathcal{Q}^k$ such that $D_I(\boldsymbol{c}_I) = \boldsymbol{m}$.
2) (Secrecy) $\forall I \subset [n], |I| \leq z : \ I(\boldsymbol{m}; \boldsymbol{c}_I) = 0$.

Such schemes are referred to as the threshold ramp secret sharing schemes [2] in the literature. In this paper we focus on designing low-complexity schemes suitable for distributed storage, notably for the RAID architectures, and name them *secure RAID schemes*. We focus on linear schemes, and following the notation of error-correcting codes for distributed storage [3], we study two kinds of linear schemes, namely *scalar* schemes and *array* schemes. For a scalar secure RAID scheme, $\mathcal{Q}$ is a finite field $\mathbb{F}_q$ and the encoding function F is linear over $\mathbb{F}_q$. For an array secure RAID scheme, $\mathcal{Q}$ is a vector space $\mathbb{F}_q^t$ and $\boldsymbol{m}$, $\boldsymbol{u}$ are regarded by the encoding function as vectors over $\mathbb{F}_q$ of length tk and tv, simply by interpreting each symbol of $\mathbb{F}_q^t$ as a block of length t over $\mathbb{F}_q$. When we make this interpretation, $\boldsymbol{m}$ and $\boldsymbol{u}$ are denoted by $\bar{\boldsymbol{m}}$ and $\bar{\boldsymbol{u}}$ to avoid confusion. The encoding function F is linear over $\mathbb{F}_q$, taking $\bar{\boldsymbol{m}}$ and $\bar{\boldsymbol{u}}$ as inputs. The output codeword is viewed as a $t \times n$ array with entries $c_{i,j}$ over $\mathbb{F}_q$, $i = 1, ..., t$, $j = 1, ..., n$. Note that a column of the array corresponds to an entry of $\boldsymbol{c}$ over $\mathbb{F}_q^t$, and that under the array representation erasure and eavesdropping are column-wise. Alternatively, the output codeword is denoted by $\bar{\boldsymbol{c}}$ when regarded as a vector over $\mathbb{F}_q$ of length tn, i.e., $\bar{\boldsymbol{c}} = (c_{1,1}, ..., c_{t,1}, ..., c_{1,n}, ..., c_{t,n})$. Clearly scalar schemes are special cases of array schemes with $t = 1$. Without loss of generality, in the remaining part of the paper it is assumed that the secure RAID schemes are array schemes. An $[n, k]_{\mathbb{F}_q^t}$ array code $\mathcal{C}$ of minimum distance $d_{\min}(\mathcal{C}) = r + 1$, where the Hamming distance is defined with respect to $\mathbb{F}_q^t$, is equivalent to an $(n, k, r, 0)_{\mathbb{F}_q^t}$ secure RAID scheme. Denote the dual code of $\mathcal{C}$ by $\mathcal{C}^\perp$.

In reminiscent of linear codes, we define the *generator matrix* of a linear secure RAID scheme to be a $(v+k)t \times nt$ matrix G over $\mathbb{F}_q$ such that $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{m}})G = \bar{\boldsymbol{c}}$. We refer to the first vt rows of G as the *key rows* which correspond to the keys, and refer to the remaining kt rows as the *message rows* which correspond to the messages. It is useful to note that while two generator matrices with the same row space generates the same linear code, this is not true for secure RAID schemes. Particularly, let G be the generator matrix of a secure RAID scheme, performing elementary row operations on G in general will violate the secrecy condition. And as such the resulting matrix, though has the same row space as G, may not correspond to a valid scheme.

The *rate* of an (n, k, r, z) secure RAID scheme is k/n and characterizes the space efficiency of the scheme. The optimal rate is known to be $\frac{n-r-z}{n}$, namely, the maximum message size is achieved when $k = n-r-z$ [11]. Constructions of rate-optimal schemes are well known, such as Shamir's (ramp) secret sharing scheme [20].

An secure RAID scheme is associated with an encoding algorithm and multiple decoding algorithms. The encoding algorithm is the algorithm of evaluating the encoding function F, and the decoding algorithms are the algorithms of evaluating the decoding functions D_I for $|I| \geq n-r$, referred to as the *systematic decoding algorithm* when $|I| = n$ and the *erasure decoding algorithm* when $|I| < n$. For a secure RAID scheme to be computationally efficient, 1) the encoding/decoding algorithms should take a small number of operations to encode/decode per message symbol, and 2) the field size q should be small. The computational efficiency of secure RAID schemes is of immense practical importance as it is closely related to the read and write performances of the storage systems. In this paper we also address the efficiency of secure RAID schemes in terms of random access, i.e., the operation of decoding partial message. Specifically, we study the computational and communication efficiency of decoding a single arbitrary entry of $\bar{\boldsymbol{m}}$, in the setting that no erasure has occurred.

III. "Lowest Density" Bounds

Define the *density* of a vector or a matrix to be the number of non-zero entries in the vector/matrix. Designing secure RAID schemes with low density generator matrices is important because such scheme requires a small number of operations in encoding. In this section we study lower bounds on the density of the generator matrices of secure RAID schemes. A related question of practical importance is to determine the amount of independent randomness, i.e., the number of keys, required by a scheme. We first address this question. The following lemma is useful.

Lemma 1. *For any rate-optimal $(n,k,r,z)_{\mathbb{F}_q^t}$ secure RAID scheme, and any $J \subset [n]$ such that $|J| = z$, it follows that $H(\boldsymbol{c}_J) = zt$.*

Proof. Let the message $\boldsymbol{m}$ be uniformly distributed and suppose for the sake of contradiction that there exists $J \subset [n]$, $|J| = z$, such that $H(\boldsymbol{c}_J) = zt - \epsilon$ for some $\epsilon > 0$. For the ease of notation, we assume without loss of generality (by permuting the indexes if necessary) that $J = [z]$. By the chain rule, $H(\boldsymbol{c}_J) = \sum_{i=1}^{z} H(\boldsymbol{c}_i|\boldsymbol{c}_{[i-1]}) = zt - \epsilon$, and it follows that there exists $i' \in [z]$ such that $H(\boldsymbol{c}_{i'}|\boldsymbol{c}_{[i'-1]}) \le t - \epsilon'$ for some $\epsilon' > 0$. Hence $H(\boldsymbol{c}_{i'}|\boldsymbol{c}_{[z]\backslash\{i'\}}) \le t - \epsilon'$. Without loss of generality (by permuting the indexes if necessary) let us assume that $i' = 1$. Denote $[i,j] = \{i, i+1, ..., j\}$, it follows that

$$\begin{aligned}
I(\boldsymbol{m}; \boldsymbol{c}_{[2,z+1]}) &\overset{(a)}{=} I(\boldsymbol{m}; \boldsymbol{c}_{[z+1]}) - I(\boldsymbol{m}; c_1|\boldsymbol{c}_{[2,z+1]}) \\
&\overset{(b)}{=} I(\boldsymbol{m}; \boldsymbol{c}_{[z+k]}) - I(\boldsymbol{m}; \boldsymbol{c}_{[z+2,z+k]}|\boldsymbol{c}_{[z+1]}) - I(\boldsymbol{m}; c_1|\boldsymbol{c}_{[2,z+1]}) \\
&\overset{(c)}{=} kt - I(\boldsymbol{m}; \boldsymbol{c}_{[z+2,z+k]}|\boldsymbol{c}_{[z+1]}) - I(\boldsymbol{m}; c_1|\boldsymbol{c}_{[2,z+1]}) \\
&\ge kt - H(\boldsymbol{c}_{[z+2,z+k]}) - I(\boldsymbol{m}; c_1|\boldsymbol{c}_{[2,z+1]}) \\
&\ge kt - (k-1)t - I(\boldsymbol{m}; c_1|\boldsymbol{c}_{[2,z+1]}) \\
&= t - H(c_1|\boldsymbol{c}_{[2,z+1]}) + H(c_1|\boldsymbol{c}_{[2,z+1]}, \boldsymbol{m}) \\
&\ge t - H(c_1|\boldsymbol{c}_{[2,z+1]}) \\
&\ge t - H(c_1|\boldsymbol{c}_{[2,z]}) \\
&\overset{(d)}{\ge} \epsilon',
\end{aligned} \tag{1}$$

where (a) and (b) follow from the chain rule; (c) follows from the fact that the scheme is rate-optimal and so $\boldsymbol{m}$ can be decoded from $\boldsymbol{c}_{[z+k]}$, as $z + k = n - r$; and (d) follows from the hypothesis $H(c_1|\boldsymbol{c}_{[2,z]}) \le t - \epsilon'$. But (1) contradicts the secrecy requirement which implies that $I(\boldsymbol{m}; c_{[2,z+1]}) = 0$. This completes the proof. □

Theorem 1. *A linear rate-optimal $(n,k,r,z)_{\mathbb{F}_q^t}$ secure RAID scheme uses at least zt keys over $\mathbb{F}_q$, and is equivalent to a scheme that uses exactly zt keys.*

Proof. Consider any linear $(n,k,r,z)_{\mathbb{F}_q^t}$ scheme such that $k = n - r - z$. Recall that the keys is a length-v vector $\boldsymbol{u}$ over $\mathbb{F}_q^t$, or equivalently a length-vt vector $\bar{\boldsymbol{u}}$ over $\mathbb{F}_q$. Let the message $\boldsymbol{m}$ be uniformly distributed. We have

$$\begin{aligned}
H(\boldsymbol{u}) &\ge I(\boldsymbol{c}_{[z]}; \boldsymbol{u}|\boldsymbol{m}) \\
&= H(\boldsymbol{c}_{[z]}|\boldsymbol{m}) - H(\boldsymbol{c}_{[z]}|\boldsymbol{u}, \boldsymbol{m}) \\
&\overset{(e)}{=} H(\boldsymbol{c}_{[z]}|\boldsymbol{m}) \\
&\overset{(f)}{=} H(\boldsymbol{c}_{[z]}) \\
&\overset{(g)}{=} zt,
\end{aligned} \tag{2}$$

where (e) follows from the fact that $\boldsymbol{c}_{[z]}$ is a function of $\boldsymbol{u}$ and $\boldsymbol{m}$; (f) follows from the secrecy requirement; and (g) follows from Lemma 1. Equation (2) implies that $v \ge z$ because $H(\boldsymbol{u}) \le vt$. This proves that the scheme uses at least zt keys over $\mathbb{F}_q$. It remains to show that the scheme is equivalent to a scheme that uses exactly zt keys.

Denote the generator matrix of the scheme by G, so G is a $(v+k)t \times nt$ matrix with entries from $\mathbb{F}_q$. Denote by G_1 the submatrix formed by the first vt rows (i.e., the key rows) and the first zt columns of G, denote by G_2 the submatrix formed by the last kt rows (i.e., the message rows) and the first zt columns of G, and denote by $\bar{\boldsymbol{u}}' = \bar{\boldsymbol{u}}G_1$. Then $\bar{\boldsymbol{c}}_{[zt]} = \bar{\boldsymbol{u}}G_1 + \bar{\boldsymbol{m}}G_2 = \bar{\boldsymbol{u}}' + \bar{\boldsymbol{m}}G_2$. Let J be an arbitrary subset of $[nt]$ such that $|J| = (z+k)t$, $[zt] \subset J$ and such that $\boldsymbol{m}$ can be decoded from $\bar{\boldsymbol{c}}_J$.

Clearly, the index set of the symbols stored by the first z nodes plus by any k additional nodes gives a valid J. We have,

$$\begin{aligned} H(\bar{\boldsymbol{c}}_J|\bar{\boldsymbol{m}},\bar{\boldsymbol{u}}') &= H(\bar{\boldsymbol{c}}_J) - I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{m}},\bar{\boldsymbol{u}}') \\ &\stackrel{(h)}{=} H(\bar{\boldsymbol{c}}_J) - I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{m}},\bar{\boldsymbol{c}}_{[zt]}) \\ &\le (z+k)t - I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{m}},\bar{\boldsymbol{c}}_{[zt]}) \\ &\stackrel{(i)}{=} (z+k)t - I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{m}}) - I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{c}}_{[zt]}|\bar{\boldsymbol{m}}) \\ &\stackrel{(j)}{=} zt - I((\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{c}}_{[zt]}|\bar{\boldsymbol{m}}) \\ &= zt - H(\bar{\boldsymbol{c}}_{[zt]}|\bar{\boldsymbol{m}}) + H(\bar{\boldsymbol{c}}_{[zt]}|\bar{\boldsymbol{m}},\bar{\boldsymbol{c}}_J) \\ &\stackrel{(k)}{=} zt - H(\bar{\boldsymbol{c}}_{[zt]}) \\ &\stackrel{(l)}{=} 0, \end{aligned} \tag{3}$$

where (h) follows from $\bar{\boldsymbol{c}}_{[zt]} = \bar{\boldsymbol{u}}' + \bar{\boldsymbol{m}}G_2$; (i) follows from the chain rule; (j) follows from $H(\bar{\boldsymbol{m}}|\bar{\boldsymbol{c}}_J) = 0$ and so $I(\bar{\boldsymbol{c}}_J;\bar{\boldsymbol{m}}) = kt$; (k) follows from $[zt] \subset J$; and (l) follows from Lemma 1. For any $i \in [n]$, since there is a valid J such that $i \in J$, (3) implies $\bar{c}_i$ is a linear function of $\bar{\boldsymbol{m}}$ and $\bar{\boldsymbol{u}}'$. Note that $\bar{\boldsymbol{u}}'$ is a vector of length-zt with entries i.i.d. uniformly distributed over $\mathbb{F}_q$. Hence there exists a matrix G' such that $\bar{\boldsymbol{c}} = (\bar{\boldsymbol{u}}'\ \bar{\boldsymbol{m}})G'$, i.e., G' is the generator matrix of an equivalent scheme that uses exactly zt keys. This completes the proof. □

Theorem 1 shows that for rate-optimal schemes, zt keys are sufficient and necessary. In the remaining part of the paper we assume that a rate-optimal $(n,k,r,z)_{\mathbb{F}_q^t}$ secure RAID scheme uses exactly zt keys, and as such the generator matrix G of the scheme has size $(z+k)t \times nt$. The following theorem lower bounds the density of G.

Theorem 2. *Consider the generator matrix of a rate-optimal $(n,k,r,z)_{\mathbb{F}_q^t}$ secure RAID scheme, then the density of each key row is at least $n-z+1$, and the density of each message row is at least $r+1$.*

Proof. Denote by G the generator matrix. Let the message $\boldsymbol{m}$ be uniformly distributed. Let J be an arbitrary subset of $[n]$ such that $|J| = k+z$, and let Z be an arbitrary subset of J such that $|Z| = z$, then we have

$$\begin{aligned} H(\boldsymbol{c}|\boldsymbol{c}_J) &= H(\boldsymbol{c},\boldsymbol{c}_J) - H(\boldsymbol{c}_J) \\ &= H(\boldsymbol{c}) - H(\boldsymbol{c}_J) \\ &\stackrel{(a)}{\le} (z+k)t - H(\boldsymbol{c}_J) \\ &= (z+k)t - H(\boldsymbol{c}_{J\backslash Z}|\boldsymbol{c}_Z) - H(\boldsymbol{c}_Z) \\ &\stackrel{(b)}{=} (z+k)t - H(\boldsymbol{c}_{J\backslash Z}|\boldsymbol{c}_Z) - zt \\ &\le kt - I(\boldsymbol{m};\boldsymbol{c}_{J\backslash Z}|\boldsymbol{c}_Z) \\ &= kt - H(\boldsymbol{m}|\boldsymbol{c}_Z) + H(\boldsymbol{m}|\boldsymbol{c}_J) \\ &\stackrel{(c)}{=} kt - H(\boldsymbol{m}|\boldsymbol{c}_Z) \\ &\stackrel{(d)}{=} 0, \end{aligned} \tag{4}$$

where (a) follows from Theorem 1; (b) follows from Lemma 1; (c) follows from the fact that $\boldsymbol{m}$ can be decoded from $\boldsymbol{c}_J$; and (d) follows from the secrecy requirement. Equation (4) implies the erasure of any $n-k-z$ entries of $\boldsymbol{c}$ can be corrected, and so that the row space of G is a code of minimum distance $n-k-z+1 = r+1$. Therefore each row of G must have at least $r+1$ non-zero entries.

It remains to lower bound the density of the first zt rows of G. Let Z be an arbitrary subset of $[n]$ such that $|Z| = z$, we have

$$\begin{aligned} H(\boldsymbol{u}|\boldsymbol{c}_Z,\boldsymbol{m}) &= H(\boldsymbol{u}|\boldsymbol{m}) - I(\boldsymbol{c}_Z;\boldsymbol{u}|\boldsymbol{m}) \\ &\stackrel{(e)}{=} zt - I(\boldsymbol{c}_Z;\boldsymbol{u}|\boldsymbol{m}) \\ &\stackrel{(f)}{=} zt - I(\boldsymbol{c}_Z;\boldsymbol{u},\boldsymbol{m}) + I(\boldsymbol{c}_Z;\boldsymbol{m}) \\ &\stackrel{(g)}{=} zt - I(\boldsymbol{c}_Z;\boldsymbol{u},\boldsymbol{m}) \\ &\stackrel{(h)}{=} zt - H(\boldsymbol{c}_Z) \\ &\stackrel{(i)}{=} 0, \end{aligned} \tag{5}$$

where (e) follows from the independence between $\boldsymbol{u}$ and $\boldsymbol{m}$; (f) follows from the chain rule; (g) follows from the secrecy requirement; (h) follows from the fact that $\boldsymbol{c}_Z$ is a function of $\boldsymbol{u}$ and $\boldsymbol{m}$; and (i) follows from Lemma 1. Equation (5) implies that, if $\boldsymbol{m}$ is fixed to $\mathbf{0}$, then the erasure of any $n-z$ entries of $\boldsymbol{c}$ can be corrected as one can first recover $\boldsymbol{u}$ and then compute $\boldsymbol{c}$. Therefore the row space of the submatrix formed by the first zt rows of G is a code of minimum distance $n-z+1$. Therefore the first zt rows of G each has at least $n-z+1$ non-zero entries. This completes the proof. □

From Theorem 2 we obtain a lower bound on the encoding complexity of an XOR-based (i.e., $q=2$) secure RAID scheme.

Corollary 1. *Encoding a rate-optimal (n,k,r,z) secure RAID scheme over $\mathbb{F}_2^t$ requires at least $r+z+\frac{rz-z}{n-r-z}$ XORs per message bit.*

Proof. By Theorem 2, the density of the key rows is at least $n-z+1$ and the density of the message rows is at least $r+1$. By Theorem 1 there are zt key rows. As the scheme is rate-optimal there are $(n-r-z)t$ message rows. Therefore the density of the generator matrix is at least $zt(n-z+1)+(n-r-z)t(r+1)$ and encoding it requires at least $zt(n-z+1)+(n-r-z)t(r+1)-nt$ XORs. Therefore, the number of XORs amortized over the message bits is

$$\frac{zt(n-z+1)+(n-r-z)t(r+1)-nt}{(n-r-z)t} = n+r+\frac{rz-z}{n-r-z} \tag{6}$$

□

IV. Systematic Secure RAID Schemes

Conventional codes for distributed storage are typically encoded in a *systematic* way. Namely, a codeword contains two sets of symbols: the uncoded message symbols that appear "in the clear" which are referred to as the *systematic symbols*, and the set of redundant symbols. Systematic codes have important advantages in terms of computational efficiency. Specifically, encoding systematic codes only requires computing redundant symbols. This is important when the rate of the code is high, i.e., the number of redundant symbols is small compared to the number of systematic symbols, which is the usual case in storage. Decoding of systematic codes is trivial in the usual case that no systematic symbols are erased. Likewise, random accessing a subset of message symbols is efficient for systematic codes. For secure RAID schemes, conventional systematic encoding is forbidden by the secrecy requirement. This motivates us to generalize the concept of systematic encoding under the context of secrecy.

Definition 1. *An $(n,k,r,z)_{\mathbb{F}_q^t}$ secure RAID scheme is systematic if*

(1). *The keys $\bar{\boldsymbol{u}}=(\bar{u}_1,...,\bar{u}_{tv})$ are stored in the uncoded form in tv entries of the codeword $\bar{\boldsymbol{c}}$.*

(2). *The message symbols $\bar{m}_1,...,\bar{m}_{tk}$ are stored in the uncoded form in tk entries of the codeword $\bar{\boldsymbol{c}}$, each padded by a linear function of the keys. Namely, in $\bar{\boldsymbol{c}}$ there is an entry of the form $\bar{m}_i+f_i(\bar{\boldsymbol{u}})$, for $i=1,...,tk$.*

(3). *For $i=1,...,tk$, the padding function $f_i(\bar{\boldsymbol{u}})$ is a function of exactly z keys.*

The tv systematic key symbols and the tk systematic message symbols are referred to as the systematic symbols.

Similar to systematic codes, by requiring the systematic symbols to take the simplest possible form, systematic secure RAID schemes have strong advantages in terms of computational efficiency. Specifically, in Definition 1, (1) ensures that encoding and decoding (when no erasure has occurred) the systematic key symbols are trivial; (2) ensures that encoding and decoding (when no erasure has occurred) the systematic message symbols only require computing the padding functions f_i's; and (3) ensures that the f_i's take the optimal form amenable to computation, in the sense that f_i has to be a function of at least z keys in order to meet the secrecy requirement. Because otherwise if f_i is a function of less than z keys, then an adversary can decode $\bar{m}_i$ by looking at no more than z entries of $\bar{\boldsymbol{c}}$, a contradiction. Systematic schemes also have optimal efficiency in terms of random access, in the sense that decoding a single entry of $\bar{\boldsymbol{m}}$ requires communicating and canceling a minimum number of z keys.

A. Method of Constructing Secure RAID Schemes

We introduce a method to design systematic secure RAID schemes. The method falls under the general framework of coset coding, which dates back to Wyner's work [23] on the wiretap channel. However here we put special emphasis on designing efficient and systematic schemes in the context of secure RAID.

Consider an an $[n,k_1]$ code $\mathcal{C}_1$ and an $[n,k_2]$ code $\mathcal{C}_2$, both over alphabet $\mathbb{F}_q^t$, such that every codeword of $\mathcal{C}_1$ is a codeword of $\mathcal{C}_2$, i.e., $\mathcal{C}_1$ is a *subcode* of $\mathcal{C}_2$. Given such a pair of codes $\mathcal{C}_1$ and $\mathcal{C}_2$, we construct a secure RAID scheme as follows. Encode $\mathcal{C}_2$ systematically and denote the index set of the systematic symbols in the codeword by I_2. Encode $\mathcal{C}_1$ systematically such that the index set I_1 of its systematic symbols satisfies $I_1 \subset I_2$ (which is possible as $\mathcal{C}_1 \subset \mathcal{C}_2$). Alternatively, we can encode $\mathcal{C}_1$ in more flexible ways as long as there is a set of entries I_1 in the codeword such that $I_1 \subset I_2$ and that $\mathcal{C}_1$ can be decoded from the entries in I_1. The secure RAID scheme has 2 steps.

Step 1: Draw tk_1 random keys $\bar{\boldsymbol{u}}$ independently and uniformly from $\mathbb{F}_q$. Encode $\mathcal{C}_1$ by regarding the keys $\bar{\boldsymbol{u}}$ as information symbols to obtain a codeword, and then puncture (delete) all entries in the codeword that is not in I_2. Denote the punctured codeword by $\boldsymbol{d}$. For example, if $I_1 = [tk_1]$ and $I_2 = [tk_2]$, then $\boldsymbol{d}$ is the vector consisting of the first tk_2 entries of the original codeword of $\mathcal{C}_1$.

Step 2: Let $\bar{\boldsymbol{m}} = (\bar{m}_1, ..., \bar{m}_{t(k_2-k_1)})$ be the secret message with entries from $\mathbb{F}_q$, and denote by $\boldsymbol{e} = \boldsymbol{d} + (\boldsymbol{0}, \bar{\boldsymbol{m}})$, where $\boldsymbol{0}$ is a length-tk_1 zero vector corresponding to the entries in I_1 and $\bar{\boldsymbol{m}}$ corresponds to the entries in $I_2 \backslash I_1$. Encode $\mathcal{C}_2$ by regarding $\boldsymbol{e}$ as information symbols to obtain a codeword $\bar{\boldsymbol{c}}$. $\bar{\boldsymbol{c}}$ is a a length-tn vector over $\mathbb{F}_q$, and is the output codeword of the secure RAID scheme. Note that the codeword $\boldsymbol{c}$ as a length-n vector over the original alphabet $\mathbb{F}_q^t$ can be obtained by collapsing each length-t segment in $\bar{\boldsymbol{c}}$ into one symbol over $\mathbb{F}_q^t$.

An illustration of the construction method is shown in Figure 2.

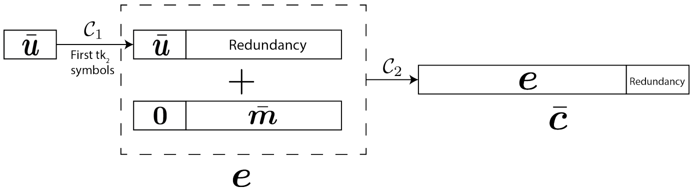

Fig. 2: Construction of SDSS from a pair of erasure codes $\mathcal{C}_1$ and $\mathcal{C}_2$ when $I_1 = [tk_1]$ and $I_2 = [tk_2]$.

Theorem 3. *Let $\mathcal{C}_1$ be an (n, k_1) code and $\mathcal{C}_2$ be an (n, k_2) code, both over $\mathbb{F}_q^t$, such that $\mathcal{C}_1$ is a subcode of $\mathcal{C}_2$. Then the described encoding scheme is an $(n, k_2 - k_1, r, z)$ secure RAID scheme over $\mathbb{F}_q^t$, where $r = d_{\min}(\mathcal{C}_2) - 1$ and $z = d_{\min}(\mathcal{C}_1^{\perp}) - 1$.*

Proof. We need to show that the scheme meets the reliability requirement and the secrecy requirement. Because $\boldsymbol{c}$ is a codeword of $\mathcal{C}_2$, and the minimum distance of $\mathcal{C}_2$ is $r + 1$, it follows that any r erasures of the entries of $\boldsymbol{c}$ can be corrected. Decoding $\boldsymbol{m}$ from $\boldsymbol{c}$ is simple, as one can read the systematic key entries $\bar{\boldsymbol{u}}$ from $\bar{\boldsymbol{c}}$, and then calculate $\boldsymbol{d}$ from $\bar{\boldsymbol{u}}$, and then cancel $\boldsymbol{d}$ from the systematic message entries to obtain $\bar{\boldsymbol{m}}$. This verifies the reliability requirement.

We now prove the security of the scheme. Consider the case that the adversary observes a specific vector $\boldsymbol{c}_I$, where I is the index set of the entries of $\boldsymbol{c}$ that are tapped by the adversary. We assume without loss of generality that $|I| = z$. Recall that $F(\boldsymbol{m}, \boldsymbol{u})$ is the encoding function of the scheme, it suffices to show that $\Pr\{F_I(\boldsymbol{m}, \boldsymbol{u}) = \boldsymbol{c}_I | \boldsymbol{m}\}$ is a constant independent of the choice of $\boldsymbol{m}$, where the probability is taken over the distribution of the keys. Consider the system of linear equations defined by $F_I(\boldsymbol{m}, \boldsymbol{u}) = \boldsymbol{c}_I$ in variables $\boldsymbol{u}$, where $\boldsymbol{m}$ and $\boldsymbol{c}_I$ are fixed, we are interested in finding the number of solutions to this system.

Let G_2 be the $tk_2 \times tn$ generator matrix of $\mathcal{C}_2$ over $\mathbb{F}_q$, such that $(\bar{\boldsymbol{u}}, \bar{\boldsymbol{m}})G_2 = \bar{\boldsymbol{c}}$. Let G_1 be the submatrix formed by the first tk_1 rows of G_2. Then G_1 is a generator matrix of $\mathcal{C}_1$. Denote by $\bar{I}$ the index set of the entries of $\bar{\boldsymbol{c}}$ corresponding to the set of entries indexed by I in $\boldsymbol{c}$, so $|\bar{I}| = tz$. We claim that the set of columns of G_1 indexed by $\bar{I}$ must be linearly independent. To prove the claim, assume for the sake of contradiction that they are linearly dependent and so there exists a length-tn vector $\bar{\boldsymbol{v}}$ such that $G_1 \bar{\boldsymbol{v}}^T = \boldsymbol{0}$, and such that $\bar{\boldsymbol{v}}$ is non-zero only in the entries indexed by $\bar{I}$. Because G_1 is a parity check matrix of $\mathcal{C}_1^{\perp}$, let $\boldsymbol{v}$ be a length-n vector over $\mathbb{F}_q^t$ obtained by collapsing each length-t segment in $\bar{\boldsymbol{v}}$ into a symbol over $\mathbb{F}_q^t$, then $\boldsymbol{v}$ is a codeword of $\mathcal{C}_1^{\perp}$ that is non-zero only in the entries indexed by I. Since $|I| = z$ but $d_{\min}(\mathcal{C}_1^{\perp}) = z + 1$, this is a contradiction.

Denote the submatrix formed by the last tk_2 rows of G_2 by G_3. For $i = 1, 2, 3$, denote by $G_{i,\bar{I}}$ the submatrix formed by columns of G_i indexed by $\bar{I}$. Then $F_I(\boldsymbol{m}, \boldsymbol{u}) = \boldsymbol{c}_I$ is equivalent to $\bar{\boldsymbol{u}} G_{1,\bar{I}} = \bar{\boldsymbol{c}}_{\bar{I}} - \bar{\boldsymbol{m}} G_{3,\bar{I}}$. Since $G_{1,\bar{I}}$ has full column rank, it follows that the system of equations $\bar{\boldsymbol{u}} G_{1,\bar{I}} = \bar{\boldsymbol{c}}_{\bar{I}} - \bar{\boldsymbol{m}} G_{3,\bar{I}}$ in variables $\bar{\boldsymbol{u}}$ always has a solution, and the number of solution is exactly $|\text{Null}(G_{1,\bar{I}})|$, where Null$(A)$ is the left null space of matrix A, i.e., $\{\boldsymbol{x} : \boldsymbol{x}A = \boldsymbol{0}\}$. By the rank-nullity theorem, $|\text{Null}(G_{1,\bar{I}})| = q^{t(k_1 - z)}$. Because $\bar{\boldsymbol{u}}$ is uniformly distributed, we have $\Pr\{F_I(\boldsymbol{m}, \boldsymbol{u}) = \boldsymbol{c}_I | \boldsymbol{m}\} = |\text{Null}(G_{1,\bar{I}})| / q^{tk_1} = q^{-tz}$, which is independent of $\bar{m}$. This completes the proof. □

An $[n, k]$ code $\mathcal{C}$ is MDS (maximum distance separable) if $d_{\min}(\mathcal{C}) = n - k + 1$. An important special case of is that $\mathcal{C}_1$ and $\mathcal{C}_2$ are both MDS codes.

Corollary 2. *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are MDS codes, then the described encoding scheme is an $(n, k_2 - k_1, n - k_2, k_1)$ secure RAID scheme. Particularly, the scheme has optimal rate. Additionally, if the scheme is scalar, then it is systematic.*

Proof. We first state a known fact.

Lemma 2. [24], [15] *A code $\mathcal{C}$ is MDS if and only if $\mathcal{C}^{\perp}$ is MDS.*

Note that the lemma is true for both scalar and array codes. Therefore, $d_{\min}(\mathcal{C}_2) = n - k_2 + 1$ and $d_{\min}(\mathcal{C}_1^{\perp}) = k_1 + 1$. Hence it follows from Theorem 3 that the scheme is an $(n, k_2 - k_1, n - k_2, k_1)$ secure RAID scheme. Clearly the scheme has optimal rate. To see that the scheme is systematic, we only need to verify condition 3 in Definition 1, which is trivially true when $t = 1$ because $|\bar{\boldsymbol{u}}| = k_1 = z$. □

We remark that the construction method can be interpreted under the framework of coset coding in the following way. Denote by $\boldsymbol{f}$ the codeword of $\mathcal{C}_1$ by encoding $\bar{\boldsymbol{u}}$, and denote by $\boldsymbol{g}$ the codeword of $\mathcal{C}_2$ by encoding $(\boldsymbol{0}, \bar{\boldsymbol{m}})$. Because $\mathcal{C}_1$ is a subcode of $\mathcal{C}_2$, $\boldsymbol{f}$ is exactly the codeword of $\mathcal{C}_2$ by encoding $\boldsymbol{d}$ (which is the punctured $\boldsymbol{f}$). Therefore it follows from the linearity of $\mathcal{C}_2$ that $\bar{\boldsymbol{c}} = \boldsymbol{f} + \boldsymbol{g}$. Let H_1 be the systematic parity check matrix corresponding to the systematic generator matrix of $\mathcal{C}_1$ that we employ in the scheme, then $H_1 \boldsymbol{f}^T = \boldsymbol{0}$. And because H_1 is a systematic parity check matrix, we have $H_1 \boldsymbol{g}^T = \bar{\boldsymbol{m}}^T$. Therefore $H_1 \bar{\boldsymbol{c}}^T = H_1(\boldsymbol{f}^T + \boldsymbol{g}^T) = \bar{\boldsymbol{m}}^T$. In this sense, the above encoding scheme can be understood as follows: to encode a secret message $\bar{\boldsymbol{m}}$, the scheme picks a random element from the coset of $\mathcal{C}_1$ whose syndrome is $\bar{\boldsymbol{m}}$.

The construction method results in schemes that are almost systematic, where I_1 is the systematic key symbols, and $I_2 \backslash I_1$ is systematic message symbols. This systematic form connects the computational complexity of the scheme to that of the codes. Specifically, the encoding complexity of the scheme is essentially the complexity of encoding $\mathcal{C}_1$ and $\mathcal{C}_2$. A simple systematic decoding algorithm for the scheme is to compute $\boldsymbol{d}$ by encoding $\mathcal{C}_1$ and then cancel it from $\boldsymbol{e}$ to obtain $\bar{\boldsymbol{m}}$, hence the complexity is dominated by encoding $\mathcal{C}_1$. The erasure decoding algorithm first corrects the erasures by invoking the erasure correction algorithm of $\mathcal{C}_2$, and then invokes the systematic decoding algorithm. So the complexity is essentially the complexity of (erasure) decoding $\mathcal{C}_2$ plus encoding $\mathcal{C}_1$. In words, to construct efficient secure RAID schemes, it suffices to find a pair of MDS codes $\mathcal{C}_1, \mathcal{C}_2$ of appropriate rates such that $\mathcal{C}_1 \subset \mathcal{C}_2$, and that $\mathcal{C}_1$ can be efficiently encoded, and that $\mathcal{C}_2$ can be efficiently encoded and decoded.

The construction method is also promising in terms of the *simplicity of implementation*. Specifically, the encoder of the secure RAID scheme consists of the encoders of $\mathcal{C}_1$ and $\mathcal{C}_2$. The decoder of the scheme consists of the encoder of $\mathcal{C}_1$ (used in systematic decoding) and the decoder of $\mathcal{C}_2$ (used in correcting erasures). Therefore, if $\mathcal{C}_1$ and $\mathcal{C}_2$ are amenable to implementation then so are the secure RAID schemes.

B. Secure RAID from Reed-Solomon Codes

A natural choice of $\mathcal{C}_1$ and $\mathcal{C}_2$ in the construction method described in Section IV-A are the Reed-Solomon codes. In fact, Shamir's scheme can be viewed as based on Reed-Solomon codes [16]. However, we show that a systematic scheme based on Reed-Solomon codes have significant advantage over Shamir's scheme in terms of computational efficiency.

Definition 2. *(*Reed-Solomon Codes [18]*) For any $n > k$, and any prime power $q > n$, let $\mathcal{S} = \{\alpha_1, ..., \alpha_n\}$ be a set of distinct non-zero elements of $\mathbb{F}_q$, the $[n,k]_{\mathbb{F}_q,\mathcal{S}}$ Reed-Solomon code has a generator matrix*

$$G = \begin{pmatrix} 1 & 1 & ... & 1 \\ \alpha_1 & \alpha_2 & ... & \alpha_n \\ \vdots & & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & ... & \alpha_n^{k-1} \end{pmatrix} \tag{7}$$

An equivalent systematic generator matrix G^* can be obtained by performing elementary row operations on G, such that G^* contain an identity submatrix of size k. To construct secure RAID schemes based on Reed-Solomon codes, we let $\mathcal{C}_1$ and $\mathcal{C}_2$ to be Reed-Solomon codes defined on the same $\mathcal{S}$ and such that $\mathcal{C}_1$ has a smaller dimension than $\mathcal{C}_2$.

Theorem 4. *For any integer n, r and z such that $n - r - z > 0$, a systematic, rate-optimal $(n, n-r-z, r, z)$ secure RAID scheme over $\mathbb{F}_q$ can be constructed by choosing $\mathcal{C}_1$ to be an $[n,z]_{\mathbb{F}_q,\mathcal{S}}$ Reed-Solomon code and $\mathcal{C}_2$ to be an $[n, n-r]_{\mathbb{F}_q,\mathcal{S}}$ Reed-Solomon code in the method described in Section IV-A.*

Proof. By Defnition 2, the generator matrix of $\mathcal{C}_1$ is a submatrix of the generator matrix of $\mathcal{C}_2$, and hence $\mathcal{C}_1$ is a subcode of $\mathcal{C}_2$. It is well known that the Reed-Solomon codes are MDS [18], and because Reed-Solomon codes are scalar codes, the assertion follows from Corollary 2. □

Consider an $(n, n-r-z, r, z)$ systematic secure RAID scheme based on Reed-Solomon codes. Encoding the scheme is essentially encoding $\mathcal{C}_1$ and $\mathcal{C}_2$, which takes $O((r+z)(n-r))$ operations (multiplications, divisions or additions) over $\mathbb{F}_q$; systematic decoding the scheme is essentially encoding $\mathcal{C}_1$, which takes $O(z(n-z-r))$; erasure/error decoding the scheme can be accomplished by first erasure/error decoding $\mathcal{C}_2$ using the error-erasure version of the Berlekamp-Massey decoding algorithm [15], which takes $O(rn)$ operations, followed by systematic decoding.

In comparison, an $(n, n-r-z, r, z)$ Shamir's (ramp) scheme can be viewed as the non-systematic version of the proposed scheme. Encoding Shamir's scheme requires evaluating a polynomial of degree $n-r$ at n points which takes $O(n(n-r))$ operations; decoding Shamir's scheme (with or without erasures) requires interpolating the polynomial which takes $O((n-r)^2)$ operations by Lagrange interpolation. The proposed systematic scheme has significantly better computational efficiency than Shamir's scheme. Particularly, in the high rate regime that r and z are fixed and n grows, encoding and systematic decoding the systematic scheme both take $O(n)$ operations, whereas encoding and decoding (with or without erasures) Shamir's scheme both takes $O(n^2)$ operations. We remark that though (asymptotically) efficient $O(n \log n)$ algorithms are known for encoding and decoding Shamir's scheme, they have large overhead factors and are not commonly used in practice [12]. Finally the systematic scheme is also efficient in random access. Decoding one entry of $\boldsymbol{m}$ in the systematic scheme takes $O(z)$ operations and requires communicating $z+1$ symbols. Shamir's scheme, however, does not support random read access and all entries of $\boldsymbol{m}$ need to be decoded together, requiring $O((z+k)^2)$ operations and the communication of $z+k$ symbols, where $k=n-r-z$.

V. Array-based Secure RAID Schemes

Reed-Solomon codes require computation over finite fields which complicates implementation and affects computational efficiency. More efficient XOR-based array codes, e.g., [3], [24], are proposed and widely used in RAID. The generator matrices of these codes are sparse, and hence encoding requires an optimal or almost optimal number of XOR operations. In this section we design XOR-based array secure RAID schemes with optimal or almost optimal computational complexity from the array codes. Particularly, the schemes have low-density generator matrices that achieves or approach the low bound in Section III.

A key idea in our constructions is to design $\mathcal{C}_2$ based on MDS array codes and design $\mathcal{C}_1$ based on their dual codes, in the construction method described in Section IV-A. This is because the array codes and their duals 1) are both MDS, so that the resulting secure RAID scheme is rate-optimal; 2) have high and low rate, respectively, so that the scheme has high rate; 3) both have low or lowest density generator matrices, implying optimal or almost optimal encoding complexity, so that the scheme is efficient. However, array codes and their duals are rarely known to contain each other. Surprisingly, we can often modify the codes appropriately to meet the subcode condition, while not compromising their complexity and distance. We follow this idea to construct three families of optimal and almost optimal schemes in the sequel.

A. Secure RAID from EVENODD Codes

In this subsection we construct a family of low-complexity XOR-based secure RAID schemes from the EVENODD codes [3]. We show that the density of the generator matrix of the scheme approaches the lower bound in Theorem 2, and that the scheme is almost optimal in terms of encoding complexity and systematic decoding complexity.

Let p be a prime, the EVENODD code is a $[p+2,p]$ MDS array code over $\mathbb{F}_2^{p-1}$ of minimum distance 3 and with a low density generator matrix [3]. Refer to Fig. 3 for an example of $p=5$. We describe our construction idea using this example. Denote the code in Fig. 3 by $\mathcal{C}_2$, which corrects 2 column erasures. To build secrecy into $\mathcal{C}_2$, consider its dual $\mathcal{C}_2^{\perp}$, obtained by switching the roles of information and parity bit, i.e., in Fig. 3 an information bit $c_{i,6}$ is checked by (parity) entries labeled by i in the top plot, and $c_{i,7}$ is checked by entries labeled by i and S in the bottom plot. Since $\mathcal{C}_2$ is MDS, so is $\mathcal{C}_2^{\perp}$. $\mathcal{C}_2^{\perp}$ is a $[p+2,2]$ code for secrecy against 2 wiretapped nodes, i.e., if we encode two columns of keys as information bits according to $\mathcal{C}_2^{\perp}$ and pad this key array to a message array, then any two columns in the resulting array reveal no information about the message. Now we have two efficient codes for reliability and secrecy, respectively. The challenge is to combine them into a single scheme that is both reliable and secure. The straightforward approach for combining the two codes typically fails. However, as we show in Section IV-A, we can construct an efficient secure RAID scheme if $\mathcal{C}_1$ (the code for secrecy) is a subcode of $\mathcal{C}_2$ (the code for reliability). In our example, $\mathcal{C}_2^{\perp}$ is not a subcode of $\mathcal{C}_2$. However, switch column 1 and 6 in $\mathcal{C}_2^{\perp}$ to obtain $\mathcal{C}_1$ (encoding described in Fig. 4), then $\mathcal{C}_1$ meets the subcode property. Based on $\mathcal{C}_1$ and $\mathcal{C}_2$ we construct a secure RAID scheme as follows. Generate two columns of random keys; encode the keys by $\mathcal{C}_1$ but skip the last two columns of the codeword; pad message bits to the 3-rd to 5-th columns of the key array; finally complete the last two columns by encoding $\mathcal{C}_2$. Note that the first 2 columns store only keys, the next 3 columns store uncoded message bits padded by keys, and the last two columns are redundant. The encoding of keys is shown in Fig. 4. The scheme corrects 2 erasures, and because $\mathcal{C}_1 \subset \mathcal{C}_2$, the encoding of keys in the last 2 columns is consistent with $\mathcal{C}_1$ (see Fig. 4), implying secrecy against 2 wiretapped nodes. Hence we have the $(7,3,2,2)$ secure EVENODD scheme.

The construction technique can be readily generalized to any prime p. For an integer a, denote by $\langle a\rangle$ the unique integer m, $0 \le m < p$, such that $a \equiv m \pmod{p}$.

Construction 1. (EVENODD Code [3]) *Let p be a prime, and $m_{i,j}$, $i \in [p-1]$, $j \in [p]$ be the message bits. The codewords of EVENODD forms a $(p-1)\times(p+2)$ array, described by the following encoding mapping. The first p columns of the array are the systematic symbols, i.e., for $i \in [p-1]$, $j \in [p]$, $c_{i,j} = m_{i,j}$. The last two columns are redundant symbols, i.e., for $i \in [p-1]$, $c_{i,p+1} = \bigoplus_{l=1}^{p} m_{i,l}$ and $c_{i,p+2} = S + \left(\bigoplus_{l=1}^{p} m_{\langle i+1-l\rangle,l}\right)$, where $S = \bigoplus_{l=2}^{p} m_{\langle 1-l\rangle,l}$, and for the ease of notation we define $m_{0,j}=0$.*

Construction 2. (Secure EVENODD) *Let p be a prime. For $i \in [p-1]$, $j \in [p-2]$ and $l \in [2]$, let $m_{i,j}$ be the message bits, and let $u_{i,l}$ be the uniformly distributed key bits. The codewords of secure EVENODD forms a $(p-1)\times(p+2)$ array, described by the following encoding mapping. The first two columns of the array are the systematic key symbols, i.e., $c_{i,1} = u_{i,1}$ for $i \in [p-1]$, and denote $u_{\Sigma,2} = \bigoplus_{l=1}^{p-1} u_{l,2}$,*

$$c_{i,2} = \begin{cases} u_{i,1} \oplus u_{i+1,2} & i = 1, ..., p-2 \\ u_{i,1} \oplus u_{\Sigma,2} & i = p-1 \end{cases}$$

1	1	1	1	1	c1,6	
2	2	2	2	2	c2,6	
3	3	3	3	3	c3,6	
4	4	4	4	4	c4,6	

1	2	3	4	S		c1,7
2	3	4	S	1		c2,7
3	4	S	1	2		c3,7
4	S	1	2	3		c4,7

Fig. 3: $[7,5]$ EVENODD code. Codeword is a 4×7 array. The first 5 columns store information bits. Parity bit $c_{i,6}$ is the XOR of all entries labeled by i in the top plot. Parity bit $c_{i,7}$ is the XOR of all entries labeled by i and all entries labeled by S in the bottom plot.

1	1	1	1	1	1	
2	2	2	2	2	2	
3	3	3	3	3	3	
4	4	4	4	4	4	

	2	3	4	Σ	1	1
	3	4	Σ	1	2	2
	4	Σ	1	2	3	3
	Σ	1	2	3	4	4

Fig. 4: Encoding of keys in the (7,3,2,2) secure EVENODD, which is exactly the encoding of $\mathcal{C}_1$. $i = 1,...,4$ in the top (or bottom) array represents that a key bit $u_{i,1}$ (or $u_{i,2}$) is added to the corresponding entry in the codeword array; and Σ represents that $\bigoplus_{i=1}^{4} u_{i,2}$ is added. Note that the padding pattern is almost optimal, in the sense that most entries are padded by only two keys and that when more than two keys are padded, Σ only needs to be computed once.

The 3*-rd to* p*-th columns of the array are the systematic message symbols, i.e., for* $j = 3,...,p$,

$$c_{i,j} = \begin{cases} u_{i,1} \oplus u_{\langle i+j-1\rangle,2} \oplus m_{i,j-2} & i+j \neq p+1 \\ u_{i,1} \oplus u_{\Sigma,2} \oplus m_{i,j-2} & i+j = p+1 \end{cases}$$

The last two columns of the array are redundant symbols, which are computed by encoding the EVENODD code described in Construction 1, regarding the first p *columns of the array as information symbols.*

Lemma 3. *In Construction 2,* $c_{i,p+1} = u_{i,1} \oplus u_{i,2} \oplus \left(\bigoplus_{l=1}^{p-2} m_{i,l}\right)$, *and* $c_{i,p+2} = u_{i,2} \oplus S' \oplus \left(\bigoplus_{l=1}^{p-2} m_{\langle i-l-1\rangle,l}\right)$, *for* $i \in [p-1]$, *where* $S' = \bigoplus_{l=1}^{p-2} m_{\langle -l-1\rangle,l}$.

Proof. It follows that

$$\begin{aligned} c_{i,p+1} &\overset{(a)}{=} \bigoplus_{l=1}^{p} c_{i,l} \\ &\overset{(b)}{=} \left(\bigoplus_{l=1}^{p} u_{i,1}\right) + \left(\bigoplus_{\substack{l=2 \\ i+l\neq p+1}}^{p} u_{\langle i+l-1\rangle,2}\right) + \left(\bigoplus_{l=1}^{p-1} u_{l,2}\right) + \left(\bigoplus_{l=3}^{p} m_{i,l-2}\right) \\ &= u_{i,1} \oplus \left(\bigoplus_{\substack{l=2 \\ i+l\neq p+1}}^{p} u_{\langle i+l-1\rangle,2}\right) \oplus \left(\bigoplus_{l=1}^{p-1} u_{l,2}\right) \oplus \left(\bigoplus_{l=3}^{p} m_{i,l-2}\right) \\ &= u_{i,1} \oplus u_{i,2} \oplus \left(\bigoplus_{l=3}^{p} m_{i,l-2}\right) \end{aligned}$$

$$= u_{i,1} \oplus u_{i,2} \oplus \left(\bigoplus_{l=1}^{p-2} m_{i,l} \right),$$

where (a) follows from Construction 1 and (b) follows from Construction 2;. We also have that

$$\begin{aligned}
S &\overset{(c)}{=} \bigoplus_{l=2}^{p} c_{\langle 1-l \rangle, l} \\
&\overset{(d)}{=} \left(\bigoplus_{l=2}^{p} u_{\langle 1-l \rangle, 1} \right) \oplus \left(\bigoplus_{l=2}^{p} \bigoplus_{l'=1}^{p-1} u_{l',2} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle 1-l \rangle, l-2} \right) \\
&= \left(\bigoplus_{l=2}^{p} u_{\langle 1-l \rangle, 1} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle 1-l \rangle, l-2} \right) \\
&= \left(\bigoplus_{l=1}^{p-1} u_{l,1} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle 1-l \rangle, l-2} \right) \\
&= \left(\bigoplus_{l=1}^{p-1} u_{l,1} \right) \oplus \left(\bigoplus_{l=1}^{p-2} m_{\langle -l-1 \rangle, l} \right) \\
&= \left(\bigoplus_{l=1}^{p-1} u_{l,1} \right) \oplus S',
\end{aligned}$$

where (c) follows from Construction 1 and (d) follows from Construction 2. Finally, we have

$$\begin{aligned}
c_{i,p+2} &\overset{(e)}{=} S \oplus \left(\bigoplus_{l=1}^{p} c_{\langle i+1-l \rangle, l} \right) \\
&\overset{(f)}{=} S \oplus \left(\bigoplus_{\substack{l=1 \\ i+1-l \neq 0}}^{p} u_{\langle i+1-l \rangle, 1} \right) \oplus \left(\bigoplus_{\substack{l=2 \\ i+1-l \neq 0}}^{p} u_{i,2} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle i+1-l \rangle, l-2} \right) \\
&= S \oplus \left(\bigoplus_{l=1}^{p-1} u_{l,1} \right) \oplus \left(\bigoplus_{\substack{l=2 \\ i+1-l \neq 0}}^{p} u_{i,2} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle i+1-l \rangle, l-2} \right) \\
&= S' \oplus \left(\bigoplus_{\substack{l=2 \\ i+1-l \neq 0}}^{p} u_{i,2} \right) \oplus \left(\bigoplus_{l=3}^{p} m_{\langle i+1-l \rangle, l-2} \right) \\
&= S' \oplus u_{i,2} \oplus \left(\bigoplus_{l=3}^{p} m_{\langle i+1-l \rangle, l-2} \right) \\
&= S' \oplus u_{i,2} \oplus \left(\bigoplus_{l=1}^{p-2} m_{\langle i-1-l \rangle, l} \right),
\end{aligned}$$

where (e) follows from Construction 1 and (f) follows from Construction 2. □

Theorem 5. *For any prime p, secure EVENODD is a $(p+2, p-2, 2, 2)$ secure RAID scheme over $\mathbb{F}_2^{p-1}$. In particular, the average density of the key rows of the generator matrix is $\frac{3p-1}{2}$, and the average density of the message rows is $\frac{4p-5}{p-1}$.*

Proof. We interpret the scheme using the method described in Section IV-A and apply Corollary 2 to prove the correctness of the scheme. It is clear from the construction that we can regard $\mathcal{C}_2$ as the EVENODD code, the first two columns of the array as systematic key symbols, and the 3-rd to p-th columns as systematic message symbols. Note that though the keys are not stored in the uncoded form, decoding them from the systematic key symbols is trivial. The encoding mapping of $\mathcal{C}_1$ is given by fixing all message bits to be 0 in Construction 2. Specifically, consider encoding $\mathcal{C}_1$ by fixing all m_{ij}'s to be 0 in Construction 2, and then switch the first and $(p+1)$-th column of the obtained array. Denote the resulting code (after permuting the columns) by $\mathcal{C}_1'$ and the resulting array by $C' = (c_{ij}')$, then by Construction 2 and Lemma 3,

$$c_{i,j}' = \begin{cases} u_{i,1} \oplus u_{\langle i+j-1 \rangle,2} & j = 1, ..., p, \ i+j \neq p+1 \\ u_{i,1} \oplus \left(\bigoplus_{l=1}^{p-1} u_{l,2} \right) & j = 1, ..., p, \ i+j = p+1 \\ u_{i,1} & j = p+1 \\ u_{i,2} & j = p+2, \end{cases} \tag{8}$$

where for the ease of notation we define $u_{0,j} = 0$. We present an algebraic description of the encoding mapping (8). Let $M_p(x) = x^{p-1} + x^{p-2} + ... + 1$ be a polynomial of degree $p-1$ over $\mathbb{F}_2$. In terms of a $(p-1) \times (p+2)$ array, we regard each column of the array as a polynomial modulo $M_p(x)$. Namely, let us use the notation $c(\beta) = c_{p-1}\beta^{p-2} + ... + c_2\beta + c_1$, i.e., a polynomial with indeterminate β, to denote a polynomial modular $M_p(x)$, then $c(\beta)$ correspond to the column vector $(c_1, ..., c_{p-1})^T$. Let $c(\beta)d(\beta)$ denote polynomial multiplication modular $M_p(x)$, and note that $\beta c(\beta)$ corresponds to the column vector $(c_{p-1}, c_1 + c_{p-1}, c_2 + c_{p-1}, ..., c_{p-2} + c_{p-1})^T$. Using this polynomial representation, the encoding mapping (8) is equivalent to

$$\left\{ C' = (c_1'(\beta), ..., c_{p+2}'(\beta)) : c_j'(\beta) = u_1(\beta) + \beta^{j-1} u_2(\beta), j = 1, ..., p, c_{p+1}'(\beta) = u_1(\beta), c_{p+2}'(\beta) = u_2(\beta) \right\}$$

And the generator matrix of $\mathcal{C}_1'$ using the polynomial representation is

$$\begin{pmatrix} 1 & 1 & ... & 1 & 1 & 0 \\ 1 & \beta & ... & \beta^{p-1} & 0 & 1 \end{pmatrix} \tag{9}$$

It is easy to see that any 2 columns of the above generator matrix is linearly independent and so the code $\mathcal{C}_1'^{\perp}$ has minimum distance 3 and therefore is MDS. By Lemma 2, $\mathcal{C}_1'$ is MDS, with minimum distance $p+1$. It is interesting to note that (9) is a parity check matrix of the EVENODD code and therefore $\mathcal{C}_1'^{\perp}$ is exactly $\mathcal{C}_2$. Therefore by Corollary 2, Construction 2 is a $(p+2, p-2, 2, 2)$ secure RAID scheme.

We now analyze the density of the generator matrix of secure EVENODD. Recall that we say a key/message bit is checked by c_{ij} if the entry in the generator matrix corresponding to the key/message bit and c_{ij} equals 1. Then by construction, each of the $u_{i,1}$'s is checked for $p+1$ times, and each of the $u_{i,2}$'s is checked for $2(p-1)$ times. Each of the $m_{i,j}$'s, is checked for 3 times if $i+j \neq p-1$, and is checked for $2+p-1 = p+1$ times if $i+j = p-1$. This completes the proof. □

By Theorem 2, a lower bound on the density of the key rows is $p+1$ and a lower bound on the density of the message rows is 3. Therefore the scheme achieves the lower bound within a factor of 3/2 for the key rows and within a factor of $4/3$ for the message rows.

Systematic decoding the scheme is straightforward by first decoding the keys from the first two columns and then canceling them from the 3-rd to p-th columns. In case of erasures/error, the erasure/error decoding algorithm of EVENODD [3] is invoked, followed by systematic decoding. Encoding secure EVENODD according to Construction 2 takes a total number of $4p^2 - 7p + 1$ XORs, or on average $4 + \frac{3}{p-2} + \frac{2}{p-1}$ XORs per message bit. Systematic decoding takes a total number of $2p^2 - 4p + 1$ XORs, or on average $2 + \frac{1}{p-2} + \frac{1}{p-1}$ XORs per message bit. By Corollary 1, encoding each message bit requires at least $4 + \frac{2}{p-2}$ XORs. Moreover, in order to be secure against $z = 2$ eavesdroppers, each message bit has to be padded by at least two keys, and different message bits must not be padded by the same pair of keys, so decoding each message bit requires at least 2 XORs. Therefore secure EVENODD has almost optimal encoding and systematic decoding complexities.

VI. Secure RAID from B Codes

We construct a family of low-complexity XOR-based secure RAID schemes from the B codes [24]. Similar as before, we show that the density of the generator matrix of the scheme approaches the lower bound in Theorem 2, and that the scheme is almost optimal in terms of encoding complexity and systematic decoding complexity.

The B codes are equivalent to perfect one-factorization of complete graphs [24]. For any prime p, the perfect one-factorization of K_{p+1}, the complete graph of $p+1$ vertexes, is known [22], and as such geometrically defines a family of B codes, also equivalent to the codes in [25]. We present a simplified algebraic description of this family of B codes. The algebraic description is useful in later constructions.

We start with the dual B codes which are conceptually simpler. For any prime p, let $t = \frac{p-1}{2}$, the dual B code is a $[p-1, 2]$ MDS array code over $\mathbb{F}_2^t$ of minimum distance $p-2$. We refer the readers to Figure 5 for an example of the dual B code of $p = 7$ and an informal description of the construction. Let a, b be integers, we denote by $\langle \frac{a}{b} \rangle$ by the unique integer m, $0 \leq m < p$, such that $a \equiv bm \pmod{p}$.

Construction 3. (Dual B Code). *Let p be a prime, $t = \frac{p-1}{2}$ and let $m_1, ..., m_{p-1}$ be the message bits. The codewords of the dual B code forms a $t \times (p-1)$ array, described by the following encoding mapping. The first row of the array*

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
m_1	m_2	m_3	m_4	m_5	m_6
$m_2 \oplus m_6$	$m_4 \oplus m_5$	$m_6 \oplus m_4$	$m_1 \oplus m_3$	$m_3 \oplus m_2$	$m_5 \oplus m_1$
$m_3 \oplus m_5$	$m_6 \oplus m_3$	$m_2 \oplus m_1$	$m_5 \oplus m_6$	$m_1 \oplus m_4$	$m_4 \oplus m_2$

Fig. 5: Dual B code of length 6. All symbols are binary bits and all operations are XORs. The code is MDS and is able to correct $6-2=4$ node (column) erasures. Note that each message bit is checked by exactly 4 parities, implying optimal encoding complexity because this is necessary to correct 4 erasures. In general, Dual B codes with similar properties can be constructed for any length $p-1$, where p is prime, in the following simple way: node i stores m_i as well as all sums of the form $m_a \oplus m_b$ such that $\langle a+b \rangle = i$.

consists of the systematic symbols, i.e., $c_{1,j} = m_j$*, for* $j = 1, ..., p-1$*. The 2-nd to* t*-th rows are redundant symbols, i.e.,* $c_{i,j} = m_{\langle i \cdot j \rangle} \oplus m_{\langle (1-i) \cdot j \rangle}$*, for* $i = 2, ..., t$*,* $j = 1, ..., p-1$*.*

Theorem 6. *The dual B codes in Construction 3 are MDS.*

Proof. Note that the dual B codes have dimension $k=2$ because there are $p-1$ message bits and $t = \frac{p-1}{2}$. Therefore it suffices to prove that all message bits can be decoded from any two nodes. Suppose the two nodes are node u and v. To simplify presentation, let us assume that there is an extra bit m_0 which is fixed to 0. Then by construction, for $x = u, v$, node x stores $\{m_a + m_b | a + b = x, 0 \le a, b \le p-1\}$. Let $i = u/2$ and $j = v/2$, where the division is over $\mathbb{F}_p$. We now describe a path, in which vertexes represent the indexes of the message bits, and edges represent the encoded bits stored in either node u or v, i.e., the edge (a,b) represents $m_a + m_b$. The path consists of p vertexes $x_1, ..., x_p$ and $p-1$ edges, defined as follows. Let the first vertex be $x_1 = i$. Let the odd edges (i.e., the 1-st, 3-rd, ..., $(p-2)$-th edges) come from node v, i.e., they are elements of $\{(a,b) | a+b = v = 2j\}$, and let the even edges come from node u, i.e., they are elements of $\{(a,b)|a+b = u = 2i\}$. For example, $x_2 = 2j - i$, since node v stores $m_i + m_{2j-i}$ and stores no other encoded bits involving m_i; and $x_3 = 3i - 2j$, since node u stores $m_{2j-i} + m_{3i-2j}$ and stores no other encoded bits involving m_{2j-i}. By induction, it is straightforward to see that $\{x_1, ..., x_p\} = \{i + 2a(i-j) | a = 0, \pm 1, ..., \pm \frac{p-1}{2}\}$.

We claim that the path is simple, i.e., $|\{x_1, ..., x_p\}| = p$. Suppose $i + 2a(i-j) = i + 2a'(i-j)$, then because $i \neq j$, it follows that $a = a'$, proving the claim. Because $\mathbb{F}_p$ has exactly p elements, it follows that $\{x_1, ..., x_p\} = \{0, ..., p-1\}$. Particularly, the path contains a vertex labeled by 0, whose neighbors on the path are vertexes u and v. Let us cut the path at the vertex 0, obtaining two decoding paths, one starts with vertex u, and the other starts with vertex v. Following the decoding paths, all message bits on the path can be decoded one by one by cancellation, starting with canceling m_u and m_v which are stored in the clear. This completes the proof. □

In the $2t \times (p-1)t$ generator matrix of the dual B code, each row has exactly $p-2$ 1's. This meets the obvious lower bound on the number of 1's (the dual B code has minimum distance $p-2$), and therefore the dual B code has a lowest density generator matrix. This matrix is a (systematic) parity check matrix of the B code, from which we can immediately obtain a generator matrix of the B code, by recalling that $[A \; I_{rt}]$ is a parity-check matrix of an $[n, k = n-r]$ code $\mathcal{C}$ over $\mathbb{F}_q^t$ if and only if $[I_{kt} \; -A^T]$ is a generator matrix of $\mathcal{C}$. We refer the readers to Figure 6 for an example of the B code of $p = 7$ and an informal description of the construction.

Construction 4. (B Code). *Let* p *be a prime,* $t = \frac{p-1}{2}$ *and let* $m_{i,j}$*,* $i \in [t-1]$*,* $j \in [p-1]$ *be the message bits. The codewords of the B code forms a* $t \times (p-1)$ *array, described by the following encoding mapping. The first* $t-1$ *rows of the array consists of the systematic symbols, i.e.,* $c_{i,j} = m_{i,j}$*, for* $i \in [t-1]$*,* $j \in [p-1]$*. The* t*-th row consists of the redundant symbols, i.e.,* $c_{t,j} = \bigoplus_{k=1}^{t-1} \left(m_{k, \langle \frac{j}{k+1} \rangle} \oplus m_{k, \langle -\frac{j}{k} \rangle} \right)$*, for* $j \in [p-1]$*.*

By Lemma 2, the B codes are MDS and can correct 2 node erasures. In the $(p-3)t \times (p-1)t$ generator matrix of the B code, each row has exactly three 1's, meeting the obvious lower bound (the B code has minimum distance 3), and therefore the B code has a lowest density generator matrix.

We now present the $(n = p-1, k = p-5, r = 2, z = 2)$ secure RAID scheme based on the B code.

Construction 5. (Secure B). *Let* p *be a prime and* $t = \frac{p-1}{2}$*. Let* $u_1, ..., u_{p-1}$ *be the uniformly distributed key bits and let* $m_{i,j}$*,* $i \in [t-2]$*,* $j \in [p-1]$ *be the message bits. The codewords of secure B forms a* $t \times (p-1)$ *array, described by the following encoding mapping. The first row of the array consists of the systematic key symbols, i.e.,* $c_{1,j} = u_j \oplus u_{\langle 2 \cdot j \rangle} \oplus u_{\langle -j \rangle}$*,* $j \in [p-1]$*. The 2-nd to* $(t-1)$*-th rows are the systematic message symbols, i.e.,* $c_{i,j} = u_{\langle (i+1) \cdot j \rangle} \oplus u_{\langle -i \cdot j \rangle} \oplus m_{i-1,j}$*, for* $i \in [2, t-1]$*,* $j \in [p-1]$*. The* t*-th row consists of the redundant symbols, which are computed by encoding the B code described in Construction 4, regarding the first* $(t-1)$*-rows of the array as information symbols.*

An example of the scheme is shown in Fig. 7. Similar to previous discussion, the construction idea is to let $\mathcal{C}_2$ be the B code and design $\mathcal{C}_1$ to take a form similar to the dual B code, because it is low rate, MDS, and has optimal encoding

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
$m_{1,1}$	$m_{1,2}$	$m_{1,3}$	$m_{1,4}$	$m_{1,5}$	$m_{1,6}$
$m_{2,1}$	$m_{2,2}$	$m_{2,3}$	$m_{2,4}$	$m_{2,5}$	$m_{2,6}$
$m_{1,4} \oplus m_{1,6} \oplus$	$m_{1,1} \oplus m_{1,5} \oplus$	$m_{1,5} \oplus m_{1,4} \oplus$	$m_{1,2} \oplus m_{1,3} \oplus$	$m_{1,6} \oplus m_{1,2} \oplus$	$m_{1,3} \oplus m_{1,1} \oplus$
$m_{2,5} \oplus m_{2,3}$	$m_{2,3} \oplus m_{2,6}$	$m_{2,1} \oplus m_{2,2}$	$m_{2,6} \oplus m_{2,5}$	$m_{2,4} \oplus m_{2,1}$	$m_{2,2} \oplus m_{2,4}$

Fig. 6: B code of length 6. All symbols are binary bits and all operations are XORs. The code is MDS and is able to correct 2 node (column) erasures. Note that each message bit is checked by exactly 2 parities, implying optimal encoding complexity because this is necessary to correct 2 erasures. In general, B codes of minimum distance 3 and with similar properties can be constructed for any length $p-1$, where p is prime, in the following way: construct the dual B code of length $p-1$ and switch the role of information bits and parity bits. Specifically, the parity bit of node i in the B code corresponds to the information bit of node i in the dual B code, i.e., m_i; in the dual B code, m_i is checked by $n-2$ parities; these $n-2$ parities are regarded as information bits in the B code, where they are exactly the set of information bits check by the parity bit of node i.

complexity. However, the dual B code is not contained in the B code, and we need to design $\mathcal{C}_1$ carefully to meet $\mathcal{C}_1 \subset \mathcal{C}_2$ without compromising complexity.

Note that the way that the keys are padded to the systematic message symbols is similar to the dual B code. With the construction method in Section IV-A in mind, the idea here is that we choose $\mathcal{C}_2$ to be the B code and design $\mathcal{C}_1$ based on the dual B code. Refer to Figure 7 for an example of secure B. Encoding the scheme is straightforward by Construction 5. Algorithm 1 describes the systematic decoding algorithm when no erasure occurs. The correctness of Algorithm 1 is straightforward. In the case of no more than $r = 2$ node erasures, the erasure decoding algorithm of the B code [24] is invoked to correct the erasures, and then Algorithm 1 is invoked to decode the secret message.

Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
$u_1 \oplus u_2 \oplus u_6$	$u_2 \oplus u_4 \oplus u_5$	$u_3 \oplus u_6 \oplus u_4$	$u_4 \oplus u_1 \oplus u_3$	$u_5 \oplus u_3 \oplus u_2$	$u_6 \oplus u_5 \oplus u_1$
$u_3 \oplus u_5 \oplus m_1$	$u_6 \oplus u_3 \oplus m_2$	$u_2 \oplus u_1 \oplus m_3$	$u_5 \oplus u_6 \oplus m_4$	$u_1 \oplus u_4 \oplus m_5$	$u_4 \oplus u_2 \oplus m_6$
$u_\Sigma \oplus u_1 \oplus$	$u_\Sigma \oplus u_2 \oplus$	$u_\Sigma \oplus u_3 \oplus$	$u_\Sigma \oplus u_4 \oplus$	$u_\Sigma \oplus u_5 \oplus$	$u_\Sigma \oplus u_6 \oplus$
$u_4 \oplus m_3 \oplus m_5$	$u_1 \oplus m_6 \oplus m_3$	$u_5 \oplus m_2 \oplus m_1$	$u_2 \oplus m_5 \oplus m_6$	$u_6 \oplus m_1 \oplus m_4$	$u_3 \oplus m_4 \oplus m_2$

Fig. 7: The (6,2,2,2) secure B scheme. $u_\Sigma = \bigoplus_{i=1}^{p-1} u_i$. The first row stores the (relaxed) systematic key bits, the middle row(s) stores the systematic message bits, and the last row is redundant. The scheme is optimal in the middle row(s), because each message bit is padded by exactly two keys necessary for secrecy. Furthermore, the scheme is almost optimal in the last row, because each parity must involve at least two keys for secrecy and two message bits for reliability. Hence a parity involves only one more special key u_Σ, and takes one more XOR than optimal. The scheme is slightly suboptimal in the first row of keys. However encoding this row takes $2(p-1)$ XORs which is insignificant when amortized over the $\frac{p^2-6p+5}{2}$ message bits; and decoding the keys from this row is also efficient, see Algorithm 1.

Lemma 4. *In Construction 5, $c_{tj} = u_\Sigma \oplus u_j \oplus u_{\langle j/2 \rangle} \oplus \left(\bigoplus_{k=2}^{t-1} \left(m_{k-1, \langle \frac{j}{k+1} \rangle} \oplus m_{k-1, \langle -\frac{j}{k} \rangle} \right) \right)$, where $u_\Sigma = \bigoplus_{i=1}^{p-1} u_i$.*

Proof. We have

$$
\begin{aligned}
c_{t,j} &\overset{(a)}{=} \bigoplus_{k=1}^{t-1} \left(c_{k, \langle \frac{j}{k+1} \rangle} \oplus c_{k, \langle -\frac{j}{k} \rangle} \right) \\
&= c_{1, \langle \frac{j}{2} \rangle} \oplus c_{1, \langle -j \rangle} \oplus \left(\bigoplus_{k=2}^{t-1} \left(c_{k, \langle \frac{j}{k+1} \rangle} \oplus c_{k, \langle -\frac{j}{k} \rangle} \right) \right) \\
&\overset{(b)}{=} u_{\langle j/2 \rangle} \oplus u_j \oplus u_{\langle -j/2 \rangle} \oplus u_{\langle -j \rangle} \oplus u_{\langle -2j \rangle} \oplus u_j \oplus \left(\bigoplus_{k=2}^{t-1} \left(c_{k, \langle \frac{j}{k+1} \rangle} \oplus c_{k, \langle -\frac{j}{k} \rangle} \right) \right) \\
&= u_{\langle j/2 \rangle} \oplus u_{\langle -j/2 \rangle} \oplus u_{\langle -j \rangle} \oplus u_{\langle -2j \rangle} \oplus \left(\bigoplus_{k=2}^{t-1} \left(c_{k, \langle \frac{j}{k+1} \rangle} \oplus c_{k, \langle -\frac{j}{k} \rangle} \right) \right) \\
&\overset{(c)}{=} u_{\langle j/2 \rangle} \oplus u_{\langle -j/2 \rangle} \oplus u_{\langle -j \rangle} \oplus u_{\langle -2j \rangle} \oplus \left(\bigoplus_{k=2}^{t-1} \left(u_{\langle \frac{(k+1)j}{k+1} \rangle} \oplus u_{\langle -\frac{kj}{k+1} \rangle} \oplus m_{k-1, \langle \frac{j}{k+1} \rangle} \oplus u_{\langle -\frac{(k+1)j}{k} \rangle} \oplus u_{\langle \frac{kj}{k} \rangle} \oplus m_{k-1, \langle -\frac{j}{k} \rangle} \right) \right) \\
&= u_{\langle j/2 \rangle} \oplus u_{\langle -j/2 \rangle} + \oplus u_{\langle -j \rangle} \oplus u_{\langle -2j \rangle} \oplus \left(\bigoplus_{k=2}^{t-1} \left(u_{\langle -\frac{kj}{k+1} \rangle} \oplus u_{\langle -\frac{(k+1)j}{k} \rangle} \right) \right) \oplus \left(\bigoplus_{k=2}^{t-1} \left(m_{k-1, \langle \frac{j}{k+1} \rangle} \oplus m_{k-1, \langle -\frac{j}{k} \rangle} \right) \right)
\end{aligned}
$$

$$\overset{(d)}{=} u_{\langle j/2\rangle} \oplus u_{\langle -j\rangle} \oplus \left(\bigoplus_{k=1}^{t-1}\left(u_{\langle -\frac{kj}{k+1}\rangle} \oplus u_{\langle -\frac{(k+1)j}{k}\rangle}\right)\right) \oplus \left(\bigoplus_{k=2}^{t-1}\left(m_{k-1,\langle \frac{j}{k+1}\rangle} \oplus m_{k-1,\langle -\frac{j}{k}\rangle}\right)\right)$$

$$\overset{(e)}{=} u_{\langle j/2\rangle} \oplus u_{\langle -j\rangle} \oplus u_{\Sigma} \oplus u_j \oplus u_{\langle -j\rangle} \oplus \left(\bigoplus_{k=2}^{t-1}\left(m_{k-1,\langle \frac{j}{k+1}\rangle} \oplus m_{k-1,\langle -\frac{j}{k}\rangle}\right)\right)$$

$$= u_{\Sigma} \oplus u_j \oplus u_{\langle j/2\rangle} \oplus \left(\bigoplus_{k=2}^{t-1}\left(m_{k-1,\langle \frac{j}{k+1}\rangle} \oplus m_{k-1,\langle -\frac{j}{k}\rangle}\right)\right)$$

where (a) follows from Construction 4; (b) and (c) follows from Construction 5; (d) follows from merging $u_{\langle -j/2\rangle}$ and $u_{\langle -2j\rangle}$ into the summation; and (e) follows from the fact that $\bigoplus_{k=1}^{t-1}\left(u_{\langle -\frac{kj}{k+1}\rangle} + u_{\langle -\frac{(k+1)j}{k}\rangle}\right) = u_{\Sigma} \oplus u_j \oplus u_{\langle -j\rangle}$, which we now prove. Note that $\langle \frac{k}{k+1}\rangle = \langle \frac{k'}{k'+1}\rangle$ implies $\langle k\rangle = \langle k'\rangle$; $\langle \frac{k+1}{k}\rangle = \langle \frac{k'+1}{k'}\rangle$ implies $\langle k\rangle = \langle k'\rangle$; $\langle \frac{k}{k+1}\rangle = \langle \frac{k'+1}{k'}\rangle$ implies that $\langle k+k'\rangle = p-1$, and therefore it follows that in the summation, the $2(t-1) = p-3$ summands are distinct. Denote by J the set of the indexes of the summands, then J contains $1, 2, ..., p-1$ except two elements. Because $\langle \frac{k}{k+1}\rangle \neq 1$ and $\langle \frac{k+1}{k}\rangle \neq 1$, it follows that $\langle -j\rangle \notin J$. Because $\langle \frac{k}{k+1}\rangle = \langle -1\rangle$ and $\langle \frac{k+1}{k}\rangle = \langle -1\rangle$ both imply that $\langle k\rangle = t$, it follows that $j \notin J$. Hence $J = [p-1]\backslash\{j, \langle -j\rangle\}$, implying $\bigoplus_{k=1}^{t-1}\left(u_{\langle -\frac{kj}{k+1}\rangle} \oplus u_{\langle -\frac{(k+1)j}{k}\rangle}\right) = u_{\Sigma} \oplus u_j \oplus u_{\langle -j\rangle}$. This completes the proof. □

Theorem 7. *Secure B is a $(p-1, p-5, 2, 2)$ secure RAID scheme over $\mathbb{F}_2^t$, for any prime p and $t = \frac{p-1}{2}$. In particular, the density of the key rows of the generator matrix is $2p-5$, and the density of the message rows is 3.*

Proof. We interpret the scheme using the method described in Section IV-A and apply Corollary 2 to prove the correctness of the scheme. It is clear from the construction that we can regard $\mathcal{C}_2$ as the B code; $c_{i,j}$, $i \in [2, t-1]$, $j \in [p-1]$ as the systematic message entries; and $c_{1,j}$, $j \in [p-1]$ as the systematic key entries. Note that although $\boldsymbol{u}$ is not stored in the uncoded form, it can be decoded from the systematic key entries (see Algorithm 1). Finally, the encoding mapping of $\mathcal{C}_1$ is given by fixing $\boldsymbol{m}$ to be $\mathbf{0}$ in Construction 5. Specifically, Consider encoding information bits $u_1, ..., u_{p-1}$ using $\mathcal{C}_1$, and denote the codeword by $A = (a_{i,j})$. Then by Construction 5, $a_{1,j} = u_j \oplus u_{\langle 2\cdot j\rangle} \oplus u_{\langle -j\rangle}$, $a_{i,j} = u_{\langle (i+1)\cdot j\rangle} \oplus u_{\langle -i\cdot j\rangle}$, for $i \in [2, t-1]$, $j \in [p-1]$. And by Lemma 4, $a_{t,j} = u_{\Sigma} \oplus u_j \oplus u_{\langle j/2\rangle}$ for $j \in [p-1]$. Consider encoding the same set of information bits $u_1, ..., u_{p-1}$ using the dual B code described in Construction 3, and denote the codeword by $B = (b_{i,j})$. Then for $i \in [2, t-1]$, $j \in [p-1]$, it follows that $a_{1,j} = b_{1,j} \oplus b_{2,j}$, $a_{i,j} = b_{i+1,j}$ and $a_{t,j} = \bigoplus_{l=2}^{t} b_{l,j}$. On the other hand, for $i \in [3, t]$, $j \in [p-1]$, it follows that $b_{1,j} = \bigoplus_{l=1}^{t} a_{l,j}$, $b_{2,j} = \bigoplus_{l=2}^{t} a_{l,j}$, and $b_{i,j} = a_{i-1,j}$. Therefore, $\mathcal{C}_1$ and the dual B code are equivalent, and have the same minimum distance. By Theorem 6, $\mathcal{C}_1$ is MDS. By Corollary 2, it follows that Construction 5 is a $(p-1, p-5, 2, 2)$ secure RAID scheme.

We now analyze the density of G. We say a key u_i or a message bit $m_{i,j}$ is *checked* by $c_{a,b}$ if in G the row corresponding to u_i or $m_{i,j}$ is 1 in the $(at+b)$-th entry (which corresponds to $c_{a,b}$). By construction, u_i is checked by $c_{t,b}$ for $b = 1, ..., p-1$, $b \neq i, \langle 2i\rangle$, and is checked by exactly one element of $\{c_{a,1}, ..., c_{a,t-1}\}$ for $a = 1, ..., p-1$, $a \neq \langle 2i\rangle$. Therefore u_i is checked for exactly $p-2+p-3 = 2p-5$ times. A message bit $m_{i,j}$ is checked by $c_{i+1,j}$, $c_{t,\langle (i+2)j\rangle}$ and $c_{t,\langle -(i+1)j\rangle}$. Therefore $m_{i,j}$ is checked for exactly 3 times. This completes the proof. □

By Theorem 2, a lower bound on the density of the key rows is $p-2$ and a lower bound on the density of the message rows is 3. Therefore for the message rows, the scheme achieves the lowest density. For the key rows, the scheme achieves the lower bound within a factor of 2.

Algorithm 1 $\boldsymbol{m}$ = Dec($\boldsymbol{c}$); Systematic Decoding.

```
for $i \leftarrow 1$ to $t$ do          ▷ Decode keys from $c_{1,j}$, $j \in [p-1]$. Recall that $t = \frac{p-1}{2}$.
    $x \leftarrow c_{1,\langle i/4\rangle} \oplus c_{1,\langle -i/4\rangle}$          ▷ $x = u_{\langle i/2\rangle} + u_{\langle -i/2\rangle}$
    $u_i \leftarrow c_{1,\langle i/2\rangle} \oplus x$
    $u_{-i} \leftarrow c_{1,\langle -i/2\rangle} \oplus x$
end for          ▷ All keys have been decoded.
for $i \leftarrow 2$ to $t-1$ and $j \leftarrow 1$ to $p-1$ do
    $m_{i-1,j} \leftarrow c_{i,j} \oplus u_{\langle (i+1)\cdot j\rangle} \oplus u_{\langle -i\cdot j\rangle}$          ▷ Cancel keys to obtain message bits.
end for
```

Algorithm 1 describes a systematic decoding algorithm for the scheme. In the case of erasures/error, the erasure/error decoding algorithm of the B code [24] is invoked to correct the erasures, and then Algorithm 1 is invoked to decode the secret message. Encoding the scheme according to Construction 5 requires a total number of $2p^2 - 9p + 7$ XORs, or on average $4 + \frac{6}{p-5}$ XORs per message bit. Systematic decoding the scheme according to Algorithm 1 requires a total number of $p^2 - \frac{9}{2}p + \frac{7}{2}$ XORs, or on average $2 + \frac{3}{p-5}$ XORs per message bit. Encoding each message bit requires at least $4 + \frac{2}{p-5}$

XORs by Corollary 1, and decoding each message bit requires at least 2 XORs. Therefore the secure B scheme has almost optimal encoding and systematic decoding complexities.

A. *Optimal secure RAID scheme from B codes*

The secure RAID schemes constructed above are almost optimal in terms of density and computational complexity. It this subsection we describe construction of strictly optimal schemes from the B codes. Particularly, we are able to construct optimal $(p-1, p-5, 2, 2)$ secure RAID schemes over $\mathbb{F}_2^t$, where $t = \frac{p-1}{2}$, for any prime p ranging from 7 to 53.

Definition 3. *Let p be a prime, $t = \frac{p-1}{2}$, and let $\sigma : [t] \to [t]$ be a permutation. We say σ is proper with respect to p if $\sigma(1) \neq t$ and that for every codeword $C = (c_{i,j})$ of the dual B code, $c_{\sigma(i),j}$ is a codeword of the B code.*

Construction 6. *(Optimal Secure B) Let p be a prime, $t = \frac{p-1}{2}$, and let $\sigma : [t] \to [t]$ be a proper permutation with respect to p. Let $u_1, ..., u_{p-1}$ be uniformly distributed key bits. The codewords of optimal secure B forms a $t \times (p-1)$ array. The first $t-1$ rows of the array are the systematic key and message symbols, computed as follows. Denote by $C' = c'_{i,j}$ the codeword of the dual B code computed by encoding the u_j's as information symbols and denote $i^* = \sigma(1)$, then $c_{i^*,j} = c'_{1,j} = u_j$, $j \in [p-1]$; for $i \neq i^*, i \in [t-1]$, $j \in [p-1]$, $c_{i,j} = c'_{\sigma(i),j} \oplus m_{i,j}$, where the $m_{i,j}$'s are the message bits. The t-th row consists of the redundant symbols, which are computed by encoding the B code regarding the first $(t-1)$-rows of the array as information symbols.*

An example of the optimal secure B schemes is shown in Figure 1. The proper permutation (in cycle representation [6]) is $\sigma = (1)(2, 3)$. It would be helpful to compare Figure 1 to Figure 5 and Figure 6 to see the effect of σ.

Theorem 8. *The encoding scheme in Construction 6 is a $(p-1, p-5, 2, 2)$ secure RAID scheme over $\mathbb{F}_2^t$. In particular, the key rows of the generator matrix have optimal density $p-2$, and the message rows have optimal density* 3.

Proof. Similar as before, we interpret the scheme using the method described in Section IV-A. It follows from the construction that $\mathcal{C}_1$ is the dual B code for which the rows of the codeword array is permuted according to σ, and $\mathcal{C}_1$ is the B code. Since both $\mathcal{C}_1$ and $\mathcal{C}_2$ are MDS, by Corollary 2 the scheme is a $(p-1, p-5, 2, 2)$ secure RAID scheme.

By Construction 3, each key bit appears in exactly $p-2$ of the $c_{i,j}$'s, and by Construction 4, each message bit appears in exactly 3 of the $c_{i,j}$'s. Therefore each key row has density $p-2$ and each message row has density 3, meeting the lower bound in Theorem 2 and proving the theorem. □

Encoding Construction 6 requires $4 + \frac{2}{p-5}$ XORs to encode each message bit and achieves the lower bound of Corollary 1. Systematic decoding the scheme by first reading the keys and then canceling them from the systematic message symbols requires 2 XORs to decode each message bit, again achieving the obvious lower bound. Therefore Construction 6 has optimal encoding and systematic decoding complexity.

It remains to address that whether a proper permutation σ exists and how to construct it. We are not aware of a method to construct proper permutations with respect to an arbitrary prime p. However, consider an arbitrary permutation σ, the following result is useful in determining whether σ is proper.

Lemma 5. *Let p be a prime, $t = \frac{p-1}{2}$, and let $\sigma : [t] \to [t]$ be a permutation such that $\sigma(1) = i^* \neq t$. Consider five multisets $A_1 = \{\langle \frac{\sigma^{-1}(i)}{i+1} \rangle : i \in [t-1], i \neq i^*\}$, $A_2 = \{\langle \frac{1-\sigma^{-1}(i)}{i+1} \rangle : i \in [t-1], i \neq i^*\}$, $A_3 = \{\langle -\frac{\sigma^{-1}(i)}{i} \rangle : i \in [t-1], i \neq i^*\}$, $A_4 = \{\langle \frac{\sigma^{-1}(i)-1}{i} \rangle : i \in [t-1], i \neq i^*\}$ and $A_5 = \cup_{i=1}^{4} A_i \cup \{\langle \frac{1}{i^*+1} \rangle, \langle -\frac{1}{i^*} \rangle\}$. Then σ is proper with respect to p if and only if $\sigma^{-1}(t)$ and $\langle 1 - \sigma^{-1}(t) \rangle$ are elements of A_5 with odd multiplicity and all other elements of A_5 have even multiplicity.*

The lemma can be proved by verifying Definition 3 according to Construction 3 and 4. The details are omitted. With Lemma 5 we can easily check whether a given σ is proper or not. Therefore a proper σ with respect to a given p, if exists, can be found by exhaustive search. Proper σ with respect to $7 \leq p \leq 53$ are listed in Table I. While finding a proper σ with respect to p significantly larger than 53 by exhaustive search is prohibitive, we believe that they exist with respect to an infinite sequence of p.

References

[1] A. Beimel, "Secret-Sharing Schemes: A Survey," in *Coding and Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6639, no. Chapter 2, pp. 11–46.

[2] G. R. Blakley and C. Meadows, "Security of ramp schemes," *Advances in Cryptology - CRYPTO*, vol. 196, pp. 242–268, 1985.

[3] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192–202, 1995.

[4] P. M. Chen, E. K. Lee, G. a. Gibson, R. H. Katz, and D. a. Patterson, "RAID: high-performance, reliable secondary storage," *ACM Computing Surveys*, vol. 26, no. 2, pp. 145–185, 1994.

[5] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *USENIX Symposium on File and Storage Technologies (FAST)*, 2004, pp. 1–14.

[6] D. S. Dummit and R. M. Foote, *Abstract Algebra*. Wiley, 2003.

[7] A. Fikes, "Storage architecture and challenges," in *Google Faculty Summit*, 2010.

p	σ
7	(1) (2 3)
11	(1 4 2) (3) (5)
13	(1 5 3) (2) (4) (6)
17	(1) (2 8 3 6 4 7) (5)
19	(1 2) (3 9 8 4) (5 7) (6)
23	(1) (2 11 10 3 4 9 8 7 6 5)
29	(1) (2 14) (3 13 12 11 10 7 5 4) (6) (8 9)
31	(1) (2 15 12 11 6 5) (3 4) (7 10 9 8) (13 14)
37	(1 3 8 5 4 18 17 16 15 14 11 10 9 2) (6 7) (12 13)
41	(1 9 8 7 6 5 4) (2 3) (10 20 17 14 13 12 11) (15 16) (18 19)
43	(1 15 14 13) (2 12 11 10) (3 9 8 7 18 17 16 21 20 19 6 5) (4)
47	(1 17 9 15 5 4 3 2) (6 14 13 12 7) (8 11 10 16) (18 23 22 21 20) (19)
53	(1 5 4 3 18 8 7 15 14 13 12 24 23 10 9 17 16 6 26) (2 25 11 22 21 20 19)

TABLE I: Table of proper permutations with respect to different p. We use the cycle representation of permutations [6]. We note that the proper permutation may not be not unique and this table lists only one of the proper permutation(s) with respect to a specific p.

[8] Gemalto, "2014 Year of mega breaches & identity theft: Findings from the 2014 breach level index," Tech. Rep., 2014.

[9] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," *USENIX Annual Technical Conference (ATC)*, 2012.

[10] C. Huang and L. Xu, "STAR : an efficient coding scheme for correcting triple storage node failures," in *USENIX Conference on File and Storage Technologies (FAST)*, 2005, pp. 197–210.

[11] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *arXiv:1505.07515*, 2015.

[12] D. Knuth, *The Art of Computer Programming*. Addison-Wesley, 1998.

[13] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k, n)-threshold secret sharing scheme and its extension," *ISC*, 2008.

[14] C. Lv, X. Jia, L. Tiany, J. Jing, and M. Sun, "Efficient ideal threshold secret sharing schemes based on Exclusive-OR operations," *Proceedings - 2010 4th International Conference on Network and System Security, NSS 2010*, pp. 136–143, 2010.

[15] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North Holland Publishing, 1977.

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

[17] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *ACM SIGMOD*, vol. 17, no. 3, 1988, pp. 109–116.

[18] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[19] J. K. Resch and J. S. Plank, "AONT-RS: blending security and performance in dispersed storage systems," in *USENIX FAST*, 2011.

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

[21] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti, "POTSHARDS - a secure, recoverable, long-term archival storage system," *ACM Transactions on Storage*, vol. 5, no. 2, pp. 1–35, 2009.

[22] W. D. Wallis, *One-Factorizations*. Norwell, 1997.

[23] A. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, 1975.

[24] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS codes and factors of complete graphs," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1817–1826, 1999.

[25] G. V. Zaitsev, V. A. Zinov'ev, and N. V. Semakov, "Minimum-check- density codes for correcting bytes of errors, erasures, or defects," *Probl. Inform. Transm.*, vol. 19, no. 3, pp. 197–204, 1983.