Secure RAID Schemes from EVENODD and STAR Codes

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

***Abstract*—We study 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 describe a technique to shorten the secure EVENODD scheme in [6], which can optimally tolerate 2 node failures and 2 eavesdropping nodes. The shortening technique allows us to obtain secure EVENODD schemes of arbitrary lengths, which is important for practical application. We also construct a new secure RAID scheme from the STAR code. The scheme can tolerate 3 node failures and 3 eavesdropping nodes with optimal encoding/decoding and random access complexity.**

I. Introduction

In the RAID architecture [9], information is stored distributively among multiple nodes in a redundant manner that is resilient to individual node failures. Over the past decades, RAID and the fundamental idea of dispersing information to improve reliability have become a ubiquitous principle that lies at the heart of most of today's distributed storage systems.

As the need to store critical and sensitive data increases, the challenge of protecting data privacy becomes imminent. This paper studies the design of schemes to encode and store information distributively so that the system is not only failure-resilient, but also resistant to adversarial eavesdropping of individual nodes. Specifically, we study 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. These schemes can find a wide array of applications including, for example, securing disk arrays [13] (where nodes are disks), securing cloud storage [1] (where nodes are different cloud providers) and securing wireless networks [8] (where nodes are wireless devices).

A well-known secret sharing scheme is Shamir's scheme [14], which is optimal in space. However, in addition to space overhead, the security requirement induces overheads in various other aspects. A series of recent works have focused on modeling and minimizing these overheads, including computational and random access complexity [6], decoding bandwidth [7], [2], repair bandwidth [10], [11] and repair locality [12].

This paper focuses on the aspect of computation and random access. The *secure RAID schemes* proposed in [6], which are schemes with optimal encoding/decoding and random access complexity, can tolerate $r = 2$ failures and $z = 2$ eavesdroppers. The length of the schemes are $n = p + 2$ or $p - 1$, where p is a prime. Two natural and important questions remain open: 1) is it possible to design secure RAID schemes of more flexible lengths? 2) is it possible to design secure RAID schemes that can tolerate more failures and eavesdroppers? In this paper we answer both questions affirmatively.

Specifically, we design a secure RAID scheme of arbitrary length that can tolerate two failures and two eavesdropping nodes by *shortening* the secure EVENODD scheme in [6]. We remark that shortening erasure codes is trivial, i.e., given an arbitrary $[n, k]$ systematic erasure code, one can directly obtain an $[n-s, k-s]$ code of the same distance as the original code, by suppressing s information symbols in the original code and setting them to be 0 [4]. In the contrary, for secure RAID schemes while the same shortening technique will maintain the reliability parameter r, it can reduce the security parameter z. Refer to Figure 1 for an example. However, we show that secure EVENODD has the desirable property that it can be flexibly shortened to arbitrary length without compromising z if the suppressed entries are carefully chosen. This property is particularly important in practice because a specific scheme implemented in a system can be easily adapted to different configurations when the number of nodes varies.

Node 1	Node 2	Node 3	Node 4
$c_1 = u$	$c_2 = m_1 + u$	$c_3 = m_2 + u$	$\sum c_i = m_1 + m_2 + u$

(a) A simple scheme with $n = 4$, $r = 1$, $z = 1$. u is a random key bit and m_1, m_2 are message bits. Security achieved by one-time-pad and reliability achieved by the parity bit.

Node 1	Node 2	Node 3 (suppressed)	Node 4
$c_1 = u$	$c_2 = m_1 + u$	$c_3 = 0$	$\sum c_i = m_1$

(b) Shortened scheme. The bit c_3 is set to be 0 and does not need to be stored. Node 3 acts as a place holder only for the purpose of encoding. The scheme is not secure as Node 4 leaks the message bit.

Fig. 1: An example that naive shortening of a secure RAID scheme will compromise security.

Our second contribution is a new secure RAID scheme that can tolerate $r = 3$ failures and $z = 3$ eavesdroppers. The scheme is XOR-based, optimal in rate, and essentially optimal in encoding/decoding and random access complexity. Specifically, encoding one bit of information on average requires approximately $r + z = 6$ XORs and decoding one bit of information when no erasure occurs on average requires approximately $z = 3$ XORs. The scheme is constructed from the STAR code [5], which is a generalization of the EVENODD code and can optimally tolerate 3 failures. The

construction idea is to use a variant of the dual STAR code for secrecy (key padding) and to use the STAR code for reliability. We integrate this pair of codes into a systematic secure RAID scheme using the framework in [6], so that the scheme preserves the computational efficiency of the codes.

II. Shortening Secure EVENODD

A. Secure RAID schemes

In an (n, k, r, z) secure RAID scheme, a message $\boldsymbol{m} = (m_1, \ldots, m_k)$ of k symbols over some alphabet is encoded into n symbols such that: 1) Reliability: $\boldsymbol{m}$ can be decoded from any subset of encoded symbols of size $\geq n - r$. 2) Secrecy: Any subset of encoded symbols of size $\leq z$ do not reveal information on $\boldsymbol{m}$. Each of the n nodes then stores one encoded symbol. In this paper we focus on the encoding/decoding and random access complexity of secure RAID schemes. The encoding/decoding complexity is the computational complexity of the encoding/decoding algorithm measured in the amount of XORs. The random access complexity is the computational and communication complexity of decoding a single entry of the message $\boldsymbol{m}$.

B. Shortened secure EVENODD

We now discuss the shortening of secure EVENODD. For a prime p, secure EVENODD is a $(n = p + 2, k = p - 2, r = 2, z = 2)$ secure RAID scheme over alphabet $\mathbb{F}_2^{p-1}$ with essentially optimal computational and random access complexity [6]. While the length of the secure EVENODD is restricted to $p + 2$, in practice it is often desirable to obtain schemes with arbitrary length n. For erasure codes, this goal is achieved by the technique of shortening. However, for secure RAID schemes shortening in general can reduce the security parameter z. In this section we show that secure EVENODD has the desirable property that it can be flexibly shortened without compromising z. Namely, from a $(p + 2, p - 2, 2, 2)$ secure EVENODD scheme one can obtain a $(p + 2 - s, p - 2 - s, 2, 2)$ scheme for any $0 < s < p$.

We start with an algebraic description of secure EVENODD. Let p be a prime, and let $M_p(x) = \sum_{i=0}^{p-1} x^i$ be a polynomial over $GF(2)$. Let $\mathcal{R}_p$ be the ring of polynomials of degree less than $p-1$ over $GF(2)$ with multiplication taken modulo $M_p(x)$. We shall use the indeterminate α instead of x to refer to polynomials in $\mathcal{R}_p$. Note that $\alpha^p = 1$, and therefore ring elements of the form α^i always has a multiplicative inverse α^{p-i}, also denoted by α^{-i}. We remark that $\mathcal{R}_p$ is a field if and only if 2 is a primitive element in $GF(p)$. In this section we focus on the case that $\mathcal{R}_p$ is indeed a field. This is not a significant restriction as it is conjectured that 2 is a primitive element in $GF(p)$ for a constant fraction (≈ 0.374) of primes p [3]. Throughout the paper we denote $\{1, \cdots, n\}$ by $[n]$.

Construction 1. (Secure EVENODD) [6] *Let $u_1(\alpha), u_2(\alpha)$ be two key polynomials selected i.i.d. uniformly at random from $\mathcal{R}_p$, and let $m_i(\alpha)$, $i \in [p-2]$ be the message polynomials (each representing $p-1$ bits of information). The key and message polynomials are encoded into $p+2$ codeword polynomials $c_i(\alpha)$, such that $c_i(\alpha)$ represents the $p-1$ bits to be stored on the i-th node. Then $(c_1(\alpha), ..., c_{p+2}(\alpha)) = (u_1(\alpha), u_2(\alpha), m_1(\alpha), ..., m_{p-2}(\alpha))$ G_{pad} G_{EO}, where G_{pad} is a square matrix that pads the key polynomials to the message polynomials, and G_{EO} is the generator matrix for the EVENODD code. Specifically,*

$$G_{\text{pad}} = \left(\begin{array}{cc|ccc} 1 & 1 & 1 & \cdots & 1 \\ 0 & \alpha & \alpha^2 & \cdots & \alpha^{p-1} \\ \hline 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{array}\right) \tag{1}$$

and

$$G_{\text{EO}} = \left(\begin{array}{cccc|cc} 1 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 1 & \cdots & 0 & 1 & \alpha \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & \alpha^{p-1} \end{array}\right). \tag{2}$$

Construction 2. (Shortened Secure EVENODD) *Let $0 < s < p-2$ be an integer. The shortened secure EVENODD of length $p + 2 - s$ and dimension $p - 2 - s$ is encoded by*

$$(u_1(\alpha), u_2(\alpha), m_1(\alpha), ..., m_{p-2-s}(\alpha))\ G'_{\text{pad}}\ G'_{\text{EO}},$$

where $u_1(\alpha)$, $u_2(\alpha)$ are randomly selected key polynomials, $m_1(\alpha), ..., m_{p-2-s}(\alpha)$ are the message polynomials, and G'_{pad} is obtained by deleting the 3-rd to $(s+2)$-th rows and columns from G_{pad}, and G'_{EO} is obtained by deleting the 3-rd to $(s+2)$-th rows and columns from G_{EO}.

Note that the length and dimension of the shortened secure EVENODD is decreased by s compared to the secure EVENODD. Also note that by deleting the rows and columns from the matrices we are essentially suppressing the 3-rd to $(s + 2)$-th entries in the codeword of Construction 1 to be 0. The following theorem shows that the shortened secure EVENODD maintains the security parameter z.

Theorem 1. *If $\mathcal{R}_p$ is a field, then the shortened secure EVENODD is a $(p+2-s, p-2-s, 2, 2)$ secure RAID scheme. Particularly, the scheme has optimal rate.*

Proof. By [7, Proposition 1], the scheme is rate-optimal if it indeed tolerates two erasures and two eavesdroppers. It is easy to see that the shortened scheme maintains the same level of reliability as secure EVENODD, and can tolerate any two erasures. Particularly, the same decoding algorithm can be used, except that the shortened (suppressed) entries in the codeword are set to be 0 by default. It remains to be shown that the shortened scheme is also secure in the presence of two eavesdropping nodes.

By the well known security lemma (e.g., [2, Appendix 7]), the scheme is secure if and only if the following claim is true: let $c_{i_1}(\alpha), c_{i_2}(\alpha)$ be any two entries of the shortened codeword, then $u_1(\alpha)$ and $u_2(\alpha)$ are functions of $c_{i_1}(\alpha), c_{i_2}(\alpha)$ and $m_i(\alpha)$, $i = 1, ..., p-2-s$. To prove the claim, we reformulate it in the context of Construction 1. Note that encoding

Construction 2 is equivalent to encoding Construction 1 and suppressing the 3-rd to $(s+2)$-th entries in the codeword to be 0. Therefore, let $\mathcal{S} = \{3, 4, ..., s+2\}$ be the index set of the shortened entries, then an equivalent claim is: in Construction 1, for any $i_1, i_2 \in [p+2]\backslash\mathcal{S}$, $u_1(\alpha)$ and $u_2(\alpha)$ are functions of $c_{i_1}(\alpha), c_{i_2}(\alpha)$, $\{c_i(\alpha) : i \in \mathcal{S}\}$, and $m_i(\alpha)$, $i \in [p-2]\backslash\mathcal{S}$. In the following we prove this claim by showing that one can recover $u_1(\alpha)$ and $u_2(\alpha)$ from $c_{i_1}(\alpha), c_{i_2}(\alpha)$, $\{c_i(\alpha) : i \in \mathcal{S}\}$, and $m_i(\alpha)$, $i \in [p-2]\backslash\mathcal{S}$. Note that the generator matrix of Construction 1 is

$$G_{\text{pad}}\ G_{\text{EO}} = \left(\begin{array}{cc|ccc|cc} 1 & 1 & 1 & \cdots & 1 & 1 & 0 \\ 0 & \alpha & \alpha^2 & \cdots & \alpha^{p-1} & 1 & 1 \\ \hline 0 & 0 & 1 & \cdots & 0 & 1 & \alpha^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 & \alpha^{p-1} \end{array}\right). \quad (3)$$

We remove the rows corresponding to the message polynomials $m_i(\alpha)$, $i \in [p-2]\backslash\mathcal{S}$, namely the $(3+s)$-th to the p-th rows from (3) to obtain a matrix, denoted by G_s:

$$\left(\begin{array}{cc|ccc|ccc|cc} 1 & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & 0 \\ 0 & \alpha & \alpha^2 & \cdots & \alpha^{s+1} & \alpha^{s+2} & \cdots & \alpha^{p-1} & 1 & 1 \\ \hline 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 1 & \alpha^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 & 1 & \alpha^{s+1} \end{array}\right)$$

Then it suffices to show that column vectors $e_1 = (1, 0, \cdots, 0)$ and $e_2 = (0, 1, 0, \cdots, 0)$ are in the column span of the space generated by the 3-rd to $(s+2)$-th columns plus the i_1-th and i_2-th columns of G_s. Clearly, if both the i_1-th and i_2-th columns are not the last two columns of G_s, then since $\mathcal{R}_p$ is a field, the i_1-th and i_2-th columns span e_1 and e_2. In the remaining part of the proof we focus on the cases that at least one of i_1 and i_2 is equal to $p+1$ or $p+2$. We also need to distinguish the case that s is odd from the case that it is even. We begin with the case that s is odd.

Case 1 ($i_1 = p+1,\ i_2 < p+1$): sum the 3-rd to $(s+2)$-th columns and the i_1-th column to obtain $u = (0, 1 + \sum_{i=2}^{s+1} \alpha^i, 0, \cdots, 0)$. This vector together with the i_2-th column span e_1, e_2.

Case 2 ($i_1 = p+2,\ i_2 < p+1$): for $i = 3, ..., s+2$, scale the i-th column by α^{i-1} and add it to the i_1-th column to obtain the vector $v = (\sum_{j=2}^{s+1} \alpha^j, 1 + \sum_{j=2}^{s+1} \alpha^{2j}, 0, \cdots, 0)$. Now if $i_2 = 1$, then clearly v and the first column spans e_1, e_2. Otherwise, scale the i_2-th column by $\sum_{j=2}^{s+1} \alpha^j$ and add to v to obtain $(0, 1 + \sum_{j=2}^{s+1} \alpha^{j+i_2-1} + \sum_{j=2}^{s+1} \alpha^{2j}, 0, \cdots, 0)$. We only need to show that

$$\rho = 1 + \sum_{j=2}^{s+1} \alpha^{j+i_2-1} + \sum_{j=2}^{s+1} \alpha^{2j} \neq 0. \quad (4)$$

Note that $\alpha^p = 1$ and (4) is trivially true when $s = 1$ or $p = 5$. Now we prove (4) assuming $p > 5$ and $s > 1$. First suppose that $s < \frac{p+3}{2}$ so that the summation $\sum_{j=2}^{s+1} \alpha^{2j}$ includes α^4, α^6 but does not include α^5. $\sum_{j=2}^{s+1} \alpha^{j+i_2-1}$, however, sums consecutive powers of α and therefore if it includes α^5, then it must inlcude either α^4 or α^6 or both. Therefore ρ must either 1) includes both α^4 and α^6 but does not include α^5, or 2) includes α^5 but does not include at least one of α^4 and α^6. In both cases ρ is not zero. Now suppose that $s \geq \frac{p+3}{2}$, then $\sum_{j=2}^{s+1} \alpha^{2j}$ includes α^1, α^3 but does not include α^2. By the same argument as above again it follows that $\rho \neq 0$. This proves (4) and so v and the i_2-th column span e_1, e_2.

Case 3 ($i_1 = p+1, i_2 = p+2$): obtain u as in Case 1 and obtain v as in Case 2. Then u, v span e_1, e_2.

We now turn to the regime that s is even.

Case 1′ ($i_1 = p+1,\ i_2 < p+1$): sum the 3-rd to $(s+2)$-th columns and the i_1-th column to obtain $u' = (1, 1 + \sum_{i=2}^{s+1} \alpha^i, 0, \cdots, 0)$. This vector together with the i_2-th column span e_1, e_2.

Case 2′ ($i_1 = p+2,\ i_2 < p+1$): proof is identical to the proof of Case 2.

Case 3′ ($i_1 = p+1, i_2 = p+2$): Obtain u' as in Case 1′. Add u' to the j-th column to obtain

$$w_j = (0, 1 + \sum_{\substack{k=2 \\ k \neq j-1}}^{s+1} \alpha^k, 0, \cdots, 1, \cdots, 0), \quad j = 3, ..., s+2$$

where the entry of 1 is the j-th entry. Now scale w_j by α^{j-1} and sum all of them to the $(p+2)$-th column to obtain:

$$v' = \left(0, 1 + \sum_{j=2}^{s+1} \left(\alpha^j \left(1 + \sum_{l=2,\ l\neq j}^{s+1} \alpha^l\right)\right), 0, \cdots, 0\right) \quad (5)$$

$$= \left(0, 1 + \sum_{j=2}^{s+1} \alpha^j, 0, \cdots, 0\right). \quad (6)$$

Then u', v' span e_1, e_2. The proof is complete. □

III. Secure STAR

The secure RAID schemes proposed in [6] including the secure EVENODD discussed above are designed to tolerate $r \leq 2$ erasures and $z \leq 2$ eavesdroppers. A natural and important question is how to construct secure RAID schemes that can tolerate more erasures and eavesroppers. In this section we construct an efficient secure RAID scheme based on the STAR code [5], which is a generalization of the EVENODD code. The STAR code is a family of MDS array codes capable of tolerating 3 erasrues with almost optimal encoding complexity. The resulting secure RAID scheme can tolerate $r \leq 3$ erasures and $z \leq 3$ eavesdroppers, with almost optimal encoding and decoding complexity and with efficient random access compleixty. We start with describing the STAR code. Define $M_p(x)$, $\mathcal{R}_p$ and α as in Section II-B.

Construction 3. (STAR code [5]) *Let p be a prime, the STAR code is a $[p+3, p]$ MDS array code over $\mathbb{F}_2^{p-1}$. Specifically, let $m_1(\alpha), ..., m_p(\alpha)$ be p message polynomials each representing $p-1$ message bits. Then the codeword polynomials $(c_1(\alpha), ..., c_{p+3}(\alpha)) = (m_1(\alpha), ..., m_p(\alpha))\ G_{\text{STAR}}$,*

where G_{STAR} is the generator matrix of the STAR code:

$$G_{\text{STAR}} = \left(\begin{array}{cccc|ccc} 1 & 0 & \cdots & 0 & 1 & 1 & 1 \\ 0 & 1 & \cdots & 0 & 1 & \alpha & \alpha^{-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & \alpha^{p-1} & \alpha^{-(p-1)} \end{array}\right) \tag{7}$$

We now describe the secure STAR scheme.

Construction 4. (Secure STAR) *Let $u_1(\alpha), u_2(\alpha), u_3(\alpha)$ be three key polynomials selected i.i.d. uniformly at random from $\mathcal{R}_p$, and let $m_i(\alpha)$, $i \in [p-3]$ be the message polynomials (each representing $p-1$ bits of information). The key and message polynomials are encoded into $p+3$ codeword polynomials $(c_1(\alpha), ..., c_{p+3}(\alpha)) = (u_1(\alpha), u_2(\alpha), u_3(\alpha), m_1(\alpha), ..., m_{p-3}(\alpha))$ G''_{pad} G_{STAR}, where G''_{pad}, defined in (8), is a square matrix that pad the key polynomials to the message and G_{STAR} is defined in (7).*

$$G''_{\text{pad}} = \left(\begin{array}{cc|ccc|c} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{p-2} & \alpha^{p-1} \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(p-2)} & \alpha^{-(p-1)} \\ \hline 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{array}\right) \tag{8}$$

The following result shows that secure STAR is a valid secure RAID scheme.

Theorem 2. *The secure STAR is a $(n = p+3, k = p-3, r = 3, z = 3)$ secure RAID scheme over $\mathbb{F}_2^{p-1}$. Particularly, the scheme has optimal rate.*

Proof. By [7, Proposition 1], the scheme is rate-optimal if it tolerates three erasures and three eavesdroppers. Because the STAR code can tolerate three erasures and the codewords of secure STAR are codewords of the STAR code, secure STAR can also tolerate three erasures. It remains to be shown that the scheme can tolerate three eavesdropping nodes.

By the well known security lemma (e.g., [2, Appendix 7]), it suffices to show that from any three entries of the codeword $c_{i_1}(\alpha)$, $c_{i_2}(\alpha), c_{i_3}(\alpha)$ and $m_i(\alpha)$, $i = 1, ..., p-3$, one can recover $u_1(\alpha), u_2(\alpha)$ and $u_3(\alpha)$. To prove this claim, note that the generator matrix of secure STAR is G''_{pad} $G_{\text{STAR}} =$

$$\left(\begin{array}{cc|ccc|ccc} 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 0 & 0 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{p-2} & \alpha^{p-1} & 0 & 0 & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(p-2)} & \alpha^{-(p-1)} & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & \cdots & 0 & 0 & 1 & \alpha^2 & \alpha^{-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & 1 & \alpha^{p-2} & \alpha^{-(p-2)} \end{array}\right) \tag{9}$$

Let G_{top} be the matrix formed by the first three rows of the matrix in (9), then G_{top} is a systematic parity check matrix of the STAR code if the $(p+2)$-th and $(p+3)$-th columns are swapped. Because the STAR code is MDS, any three columns of its parity check matrix are linearly independent. Therefore any three columns of G_{top} are linearly independent. This proves the claim and the theorem. □

A. Encoding Secure STAR

We analyze the computational complexity of secure STAR. Consider the operation of multiplying a polynomial $f(\alpha) = \sum_{i=0}^{p-2} f_i \alpha^i$ by α^j. Then the resulting polynomial is

$$\alpha^j f(\alpha) = \sum_{\substack{i=0 \\ \langle i+j \rangle \neq p-1}}^{p-2} f_i \alpha^{\langle i+j \rangle} + \sum_{i=0}^{p-2} f_{p-1-j} \alpha^i \tag{10}$$

where $\langle \cdot \rangle$ is the modulo p operator, and we define $f_{p-1} = 0$. Note that the first summation in (10) is simply a cyclic shift of $f(\alpha)$ except that the $(p-1-j)$-th entry becomes 0. Therefore the multiplication in (10) takes at most $p-1$ XORs to compute. Consider the encoding complexity of secure STAR, in the first phase we multiply the key and message polynomials by $G_{\text{pad}''}$. This takes at most $10(p-1)+5(p-3)(p-1)$ XORs. The second phase, which is to encode the standard STAR code, takes at most $3(p-1)^2 + 2(p-2)$ XORs. Therefore the normalized encoding complexity of secure STAR is

$$\frac{10(p-1) + 5(p-3)(p-1) + 3(p-1)^2 + 2(p-2)}{(p-3)(p-1)} \approx 8$$

XORs to encode each bit of message. By [6, Corollary 1], a lower bound on the normalized encoding complexity is $6 + \frac{6}{p-3} \approx 6$ XORs to encode each message bit. Therefore the encoding complexity of secure STAR is almost optimal. In the following we show an improved encoding scheme of secure STAR to further reduce the encoding complexity. The normalized encoding complexity of the improved scheme converges to 6 as p grows, and so is asymptotically optimal.

Specifically, consider the (binary) generator matrix of the STAR code by regarding a polynomial $f(\alpha)$ as a binary row vector of length $p-1$. And so G_{STAR} expands into a $p(p-1)$ by $(p+3)(p-1)$ binary matrix, i.e., each entry in the matrix in (9) expands into a $(p-1)$ by $(p-1)$ block:

$$G'_{\text{STAR}} = \left(\begin{array}{cccc|ccc} I & 0 & \cdots & 0 & I & A_0 & A_0 \\ 0 & I & \cdots & 0 & I & A_1 & A_{\langle -1 \rangle} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & I & I & A_{p-1} & A_{\langle -(p-1) \rangle} \end{array}\right) \tag{11}$$

where I is the identity matrix of order $p-1$, 0 is the zero matrix, and $A_k = (a_{ij}^{(k)})$, $1 \leq i, j \leq p-1$ is defined by:

$$a_{ij}^{(k)} = \begin{cases} 1, & j - i = k \text{ or } i = p - k \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

For example, $A_0 = I$, and for $p = 5$

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{13}$$

Therefore the binary parity check matrix corresponding to the

systematic generator matrix in (11) is :

$$H'_{\text{STAR}} = \left(\begin{array}{cccc|ccc} I & I & \cdots & I & I & 0 & 0 \\ A_0^t & A_1^t & \cdots & A_{p-1}^t & 0 & I & 0 \\ A_0^t & A_{\langle -1 \rangle}^t & \cdots & A_{\langle -p-1 \rangle}^t & 0 & 0 & I \end{array} \right).$$

Consider the complexity of encoding the dual code of the STAR code by multiplying a message vector $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)$ with the matrix H'_{STAR}, where $\boldsymbol{u}_i$ is a binary row vector of length $p-1$. Then multiplying $\boldsymbol{u}_i$ with A_j^t is simply a cyclic shift of $\boldsymbol{u}_i$ (by j entries to the left) except that the $(p-j)$-th entry in the result becomes $u_i^* = \sum_{k=1}^{p-1} u_{ik}$. Therefore the only computation required in multiplying $\boldsymbol{u}_i$ with A_j^t is to compute u_i^*, which only needs to be performed once for each $\boldsymbol{u}_i$.

Now to encode secure STAR, instead of using the padding matrix G''_{pad} in (8), we use the following matrix G'_{pad}:

$$\left(\begin{array}{cc|ccc|c} I & I & I & \cdots & I & I \\ A_0^t & A_1^t & A_2^t & \cdots & A_{p-2}^t & A_{p-1}^t \\ A_0^t & A_{\langle -1 \rangle}^t & A_{\langle -2 \rangle}^t & \cdots & A_{\langle -(p-2) \rangle}^t & A_{\langle -(p-1) \rangle}^t \\ \hline 0 & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \cdots & I & 0 \end{array} \right)$$

Replacing G''_{pad} by G'_{pad} does not affect the security of the scheme. This is because the first three rows of G''_{pad} and of G'_{pad} span the same space, i.e., the space of the dual STAR code, with the last three entries in the codeword deleted.

The improved padding matrix reduce the encoding complexity of the padding phase to at most $2(p-2)+6(p-1)+3(p-3)(p-1)$ XORs. Therefore, the overall normalized encoding complexity of the improved scheme is

$$\frac{4(p-2)+6(p-1)+3(p-3)(p-1)+3(p-1)^2}{(p-1)(p-3)} \approx 6$$

XORs per message bit, which is asymptotically optimal.

B. Decoding Secure STAR

Next we consider the decoding complexity of secure STAR. In general one can decode by multiplying the codeword vector to the inverse of the generator matrix, but matrix inversion is an expensive operation (requiring $O(n^6)$ XORs). Even if the cost of matrix inversion is amortized (as the inverse can be precomputed), matrix multiplication is still expensive (requiring $O(n^4)$ XORs). In the following we show that the construction of secure STAR entails a very efficient decoding algorithm, requiring only $O(n^2)$ XORs in total.

The decoding algorithm can be divided into three steps: First, if any of the first p entries in the codeword is erased, recover them by erasure decoding. Secondly, decode the key polynomials $u_1(\alpha), u_2(\alpha), u_3(\alpha)$ and hence all the key bits from $c_1(\alpha), c_2(\alpha), c_p(\alpha)$. Finally, cancel the keys from $c_i(\alpha)$, $i = 3, ..., p-1$ to obtain the message polynomials. For the first step, since the codewords of secure STAR are codewords of the STAR code, recovering the erased symbols is equivalent to recovering from erasures in the STAR code. A major advantage of the STAR code is that it has a very efficient erasure decoding algorithm [5], requiring at most $O(n^2)$ XORs to recover any three erasures. In the following we focus on the latter two steps that deal with the arguably more important issue of "decrypting" the message, as erasure decoding is needed only when erasures occurred, but "decryption" is always required whenever one wants to retrieve the information.

We first describe the third step of canceling the keys, which is simply to "re-pad" the keys to the codeword in the same way as how they are padded to the messages during the encoding phase. Since the padding scheme G'_{pad} is almost optimal, i.e., most entries in the array are padded by only three key bits, the minimum number of keys to tolerate three eavesdroppers, the complexity of canceling the keys is essentially optimal. Namely, for most entries in the array, recovering the message bit stored in that entry only requires 3 XORs to cancel the keys.

We now describe the second step of decoding the key polynomials. For the ease of notation, denote for short $a_i \triangleq u_{1i}$, $b_i \triangleq u_{2i}$, $c_i \triangleq u_{3i}$, $i = 1, ..., p-1$, and $a_0 \triangleq u_1^*$, $b_0 \triangleq u_2^*$, $c_0 \triangleq u_3^*$ (recall that $u_j^* = \sum_{i=1}^{p-1} u_{ji}$). Then the coefficients of $c_1(\alpha)$ are $a_i + b_i + c_i$, the coefficients of $c_2(\alpha)$ are $a_i + b_{\langle i+1 \rangle} + c_{\langle i-1 \rangle}$ and the coefficients of $c_p(\alpha)$ are $a_i + b_{\langle i-1 \rangle} + c_{\langle i+1 \rangle}$, $i = 1, ..., p-1$. Therefore the coefficients of $c_1(\alpha) + c_2(\alpha)$ are $u_i \triangleq b_i + b_{\langle i+1 \rangle} + c_{\langle i-1 \rangle} + c_i$, and the coefficients of $c_1(\alpha) + c_p(\alpha)$ are $v_i \triangleq b_{\langle i-1 \rangle} + b_i + c_i + c_{\langle i+1 \rangle}$, $i = 1, ..., p-1$.

For $i = 0, ..., p-3$, by XORing $v_{\langle i+1 \rangle}$ and $u_{\langle i+2 \rangle}$ we obtain $w_i = b_i + b_{\langle i+1 \rangle} + b_{\langle i+2 \rangle} + b_{\langle i+3 \rangle}$. Since $b_0 = u_2^* = \sum_{i=1}^{p-1} b_i$, we have $w_0 = \sum_{i=4}^{p-1} b_i$, and $w_{p-3} = \sum_{i=1}^{p-4} b_i$. We consider two cases: Case 1: $p \mod 4 = 1$. Therefore 4 divides $p-5$ and we can combine the w_i's to obtain $\sum_{i=5}^{p-1} b_i$. Canceling it from w_0 we obtain b_4. Similarly, 4 divides $p-9$ and so we can obtain $\sum_{i=6}^{p-4} b_i$. Canceling $\sum_{i=6}^{p-4} b_i$ and w_1 from w_{p-3} we obtain b_5. By symmetric we can also obtain c_4 and c_5. Case 2: $p \mod 4 = 3$. Therefore 4 divides $p-3$ and we can combine the w_i's to obtain $\sum_{i=3}^{p-1} b_i$. Canceling w_0 from it we obtain b_3. Similarly, 4 divides $p-7$ and we can obtain $\sum_{i=4}^{p-4} b_i$. Canceling it from w_{p-3} we obtain $b_1 + b_2 + b_3$. Finally cancel it from w_1 and we obtain b_4. By symmetric we can also obtain c_3 and c_4.

Therefore, there always exists an i so that we can obtain b_i, b_{i+1} and c_i, c_{i+1}. Now cancel b_i, c_i and c_{i+1} from v_i to obtain b_{i-1} and cancel b_{i+1}, c_i and c_{i+1} from u_{i+1} to obtain b_{i+2}. By symmetric we can also obtain c_{i-1} and c_{i+2}. By induction we obtain all b_i, c_i, $i = 1, ..., p-1$. Finally, cancel the b_i's and the c_i's from the coefficients of $c_1(\alpha)$ to obtain a_i, $i = 1, ..., p-1$. This completes the decoding of all key bits.

We summarize the computational complexity of the decoding algorithm when no erasure occurs, i.e., the complexity of the second and third steps of the algorithm. The second step requires no more than $18(p-1)$ XORs and the third step requires no more than $3(p-1) + 3(p-3)(p-1)$ XORs.

Therefore the normalized decoding complexity is

$$\frac{18(p-1)+3(p-1)+3(p-3)(p-1)}{(p-3)(p-1)} \approx 3$$

XORs per message bit. Since every message bit has to be padded by at least three key bits in order to tolerate three eavesdropping nodes, the decoding complexity of the scheme is asymptotically optimal.

REFERENCES

[1] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "Depsky: Dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage*, vol. 9, no. 4, pp. 12:1–12:33, 2013.

[2] R. Bitar and S. El Rouayheb, "Staircase codes for secret sharing with optimal communication and read overheads," *arXiv:1512.02990*, 2016.

[3] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 529–542, 1996.

[4] 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, pp. 192–202, 1995.

[5] 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.

[6] W. Huang and J. Bruck, "Secure RAID schemes for distributed storage," in *2016 IEEE International Symposium on Information Theory (ISIT)*, July 2016, pp. 1401–1405.

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

[8] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," in *International Symposium on Computers and Communications*, 2002.

[9] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *ACM SIGMOD*, vol. 17:3, 1988, pp. 109–116.

[10] S. Pawar, S. E. Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.

[11] 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 Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[12] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal Locally Repairable and Secure Codes for Distributed Storage Systems," *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 212 – 236, Jan. 2014.

[13] J. K. Resch and J. S. Plank, "AONT-RS: blending security and performance in dispersed storage systems," in *USENIX conference on File and stroage technologies (FAST)*, 2011, pp. 191–202.

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.