Fast Encoding of Polar Codes with Reed-Solomon Kernel

Peter Trifonov, *Member, IEEE,* Vera Miloslavskaya, *Member, IEEE*, Chen Chen, Yuangang Wang

***Abstract*—A low-complexity systematic encoding algorithm for polar codes with Reed-Solomon kernel is presented. The proposed method relies on fast Fourier transform based Reed-Solomon encoding techniques. Application of polar codes in storage systems is considered.**

***Index Terms*—Polar codes, generalized concatenated codes, Reed-Solomon codes.**

I. Introduction

Polar codes were recently shown to be able to achieve the capacity of a wide class of communication channels while having very low encoding and decoding complexity [1]. Furthermore, many applications of polar codes besides communications systems were suggested. On the other hand, numerous improvements to the original Arikan construction of polar codes were proposed. In particular, polar codes over $\mathbb{F}_q$ with an $l \times l$ Reed-Solomon (RS) kernel [2] provide the highest possible polarization rate for the case of $l \leq q$. This results in significant performance gain with respect to polar codes with Arikan kernel [3].

Applications typically require systematic encoding algorithms. However, the original construction of polar codes assumes non-systematic encoding. An efficient systematic encoding algorithm for polar codes with Arikan kernel was introduced in [4]. Besides implementation advantages, systematic polar codes were shown to provide lower symbol error rate compared to non-systematic ones.

In this paper we address the systematic encoding problem for the case of codes with a RS kernel. More specifically, we propose an erasure decoding algorithm, and use it at the encoder side for recovering the values of erased check symbols. The main idea of the proposed method is to treat polar codes as generalized concatenated ones. Then one can employ a multistage decoding algorithm and recursively recover the erasures. Furthermore, we exploit the algebraic structure of the RS kernel in order to reduce the complexity of this algorithm. The proposed approach can be considered as a generalization of the systematic encoding algorithm for polar codes with Arikan kernel [4], and the multidimensional polar encoding and decoding method suggested in [5].

The paper is organized as follows. Section II introduces polar, RS and generalized concatenated codes. Section III presents an erasure decoding algorithm, and some results on correctable erasure patterns. These results are used in Section IV to derive a fast systematic encoding algorithm. Section V considers application of polar codes in storage systems.

P. Trifonov and V. Miloslavskaya are with the Distributed Computing and Networking Department, Saint-Petersburg Polytechnic University, Polytechnicheskaya str., 21, office 103, 194021, Saint-Petersburg, Russia, Email: {petert,veram}@dcn.icc.spbstu.ru.

C. Chen and Y. Wang are with Shannon Cognitive Computing Laboratory, Huawei Technologies Co. Ltd, China. Email: {chenchen9,wangyuangang}@huawei.com

II. Background

A. Polar codes with Reed-Solomon kernel

An $(n = l^m, k)$ polar code over $\mathbb{F}_q$ with $l \times l$ kernel B and the set of frozen symbols $\mathcal{F} \subset \{0, 1, \dots, l^m - 1\}, |\mathcal{F}| = n - k$, is a linear block code with codewords $c_0^{n-1} = u_0^{n-1} A$, such that $u_i = 0, i \in \mathcal{F}$, and $u_j \in \mathbb{F}_q, j \notin \mathcal{F}$. Here a_i^j denotes a vector $(a_i, \dots, a_j)$, $A = P_m B^{\otimes m}$, $\otimes m$ denotes m-times Kronecker product of a matrix with itself, and P_m is a permutation matrix having 1's in positions $(j, R(j))$, where

$$R(j) = \sum_{s=0}^{m-1} j_s l^{m-1-s}$$

is the integer obtained by reversal of digits of integer $j = \sum_{s=0}^{m-1} j_s l^s, j_s \in \{0, \dots, l-1\}$, in the base-$l$ representation. Figure 1 illustrates the non-systematic encoder for polar codes for the case of $l = 3, m = 2$.

If symbols $c_i, 0 \leq i < n$, are transmitted over a memoryless output-symmetric channel $W(y|c)$, then one can define synthetic subchannels with transition probabilities

$$W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i) = \frac{1}{q^{n-1}} \sum_{u_{i+1}^{n-1}} \prod_{j=0}^{n-1} W(y_0^{n-1}|(u_0^{n-1}A)_j).$$

It is possible to show that under some mild conditions on matrix B the capacities of these subchannels converge to 0 or 1 symbols per channel use, and the fraction of subchannels with capacities close to 1 converges to the capacity of $W(y|c)$. This enables one to construct capacity achieving polar codes by selecting $\mathcal{F}$ as the set of indices of subchannels with capacity close to 0 [1], [6]. Alternatively, one can select $\mathcal{F}$ as the set of indices of subchannels with the highest Bhattacharyya parameter. Polar codes can be considered as optimized codes for multistage decoding [7], [8].

A RS kernel is given by the Vandermonde matrix

$$B = \begin{pmatrix} \alpha_0^{l-1} & \alpha_1^{l-1} & \dots & \alpha_{l-1}^{l-1} \\ \alpha_0^{l-2} & \alpha_1^{l-2} & \dots & \alpha_{l-1}^{l-2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^0 & \alpha_1^0 & \dots & \alpha_{l-1}^0 \end{pmatrix},$$

where α_i are some distinct elements of $\mathbb{F}_q$. It can be seen that the κ last rows of this matrix generate an $(l, \kappa, l - \kappa + 1)$ RS

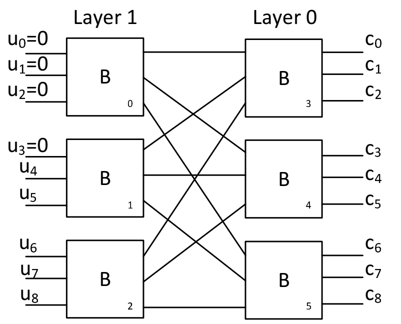


Fig. 1. Non-systematic encoder for polar codes

code. It is possible to show that the RS kernel provides the highest possible polarization rate for $l \leq q$ [2]. Constructions similar to polar codes with RS kernel were also considered in [9], [10], [11].

B. Reed-Solomon codes

There are two equivalent ways to define an $(l, \kappa, l-\kappa+1)$ RS code [12], [13]. The first one is to consider it as a set of vectors $(f(\alpha_0), \ldots, f(\alpha_{l-1})), f(x) = \sum_{i=0}^{\kappa-1} f_i x^i$. The second way is to consider RS code of length $l < q$ as a set of vectors $(c_0, \ldots, c_{l-1})$, which can be represented as

$$c(x) = \sum_{i=0}^{l-1} c_i x^i = a(x) \prod_{i=0}^{l-\kappa-1} (x - \alpha^{b+i})$$

for some $a(x) : \deg a(x) < \kappa$, where α is a primitive element of $\mathbb{F}_q$. These definitions are equivalent if $b = 1$ and $\alpha_i = \alpha^i, 0 \leq i < l = q-1$. In order to obtain a code of length $l = q$ for $\kappa < l$, one can set $\alpha_{l-1} = 0$, so that $c_i = f(\alpha_i)$, and $c_{l-1} = -\sum_{i=0}^{l-2} c_i$.

The second definition enables one to derive an efficient erasure correcting algorithm. Let us consider for the sake of simplicity the case of $\alpha_i = \alpha^i, q = 2^\mu, l < q$, and assume that symbols $c_{e_1}, \ldots, c_{e_t}, 0 \leq t \leq l - \kappa$, are erased. Let $S(x) = \sum_{i=0}^{l-\kappa-1} S_i x^i$ be a syndrome polynomial with coefficients given by

$$S_i = \sum_{j=0}^{l-1} c_j \alpha^{j(i+b)}. \quad (1)$$

Let $\Lambda(x) = \prod_{j=1}^{t} (1 + \alpha^{e_j} x)$ be the erasure locator polynomial, and let

$$\Gamma(x) \equiv S(x)\Lambda(x) \bmod x^{l-\kappa} \quad (2)$$

be the erasure evaluator polynomial. It can be verified [12], [13] that

$$c_{e_j} = \frac{\alpha_{e_j}^{-b} \Gamma(\alpha_{e_j}^{-1})}{\prod_{\substack{s=1 \\ s \neq j}}^{t} (1 + \alpha_{e_s} \alpha_{e_j}^{-1})} = \frac{\alpha_{e_j}^{1-b} \Gamma(\alpha_{e_j}^{-1})}{\Lambda'(\alpha_{e_j}^{-1})}, \quad (3)$$

where $\Lambda'(x)$ is the formal derivative of $\Lambda(x)$.

The standard way to implement these calculations is to use the Horner rule for computing S_i and $\Gamma(\alpha_{e_j}^{-1})$. Fast Fourier transform techniques were suggested to speed up these calculations [14], [15].

C. Generalized concatenated codes

It is convenient to treat polar codes in the framework of generalized concatenated codes [16], [17], [18]. A generalized concatenated code (GCC) over $\mathbb{F}_q$ is given by outer codes $\mathcal{C}_i(N, K_i, D_i), 0 \leq i < l$, and nested inner codes $\mathbb{C}_i(l, l-i, d_i)$. Inner codes can be given by an $l \times l$ matrix B, such that rows $i, \ldots, l-1$ of this matrix generate $\mathbb{C}_i$.

A GCC codeword can be considered as an $l \times N$ matrix C, which is obtained as follows. Partition the data into l blocks of size K_i. Then encode the i-th block with code $\mathcal{C}_i$, and place the obtained codeword into the i-th row of the matrix. Let C' be the obtained matrix. Finally, multiply B^T by each column of C', i.e. compute $C = B^T C'$. It is possible to show that this encoding procedure defines $(Nl, \sum_{i=0}^{l-1} K_i, d \geq \min_{0 \leq i < l} d_i D_i)$ linear code. Observe that by setting i initial data blocks to zero, one obtains that each column of C is a codeword of $\mathbb{C}_i$. This fact is exploited in the multistage decoding algorithm described below, which at stage i essentially subtracts from C the contribution of i initial blocks, so that it can use the decoders of $\mathbb{C}_i$ and $\mathcal{C}_i$.

Observe that encoding of polar codes with kernel B can be represented as $c = uP_m(B^{\otimes(m-1)} \otimes B)$. This implies that polar codes with RS kernel can be considered as GCC with inner codes being $(l, l-i, i+1)$ RS codes, and outer polar codes of length l^{m-1} with RS kernel. The outer codes are given by the sets of frozen symbols $\mathcal{F}^{(i)} = \left\{ j | 0 \leq j < l^{m-1}, (il^{m-1} + j) \in \mathcal{F} \right\}$. By induction, one can show from this that the minimum distance of polar codes with RS kernel is given by

$$d \geq \min_{j \notin \mathcal{F}} \prod_{s=0}^{m-1} (j_s + 1), \quad (4)$$

where $j = \sum_{s=0}^{m-1} j_s l^s, 0 \leq j_s < l$. This statement is obvious for $m = 1$.

The standard way to decode GCC is to employ the multistage decoding algorithm [19]. The successive cancellation decoding algorithm suggested for polar codes can be considered as an instance of multistage decoding.

D. Systematic encoding

By systematic encoding we mean such encoding method, so that the information symbols appear as a part of the obtained codeword. For any (n, k) linear block code one can obtain a generator matrix $G = (I|A)P$, where P is a permutation matrix. The systematic encoder can be implemented as $c = xG$, so that codeword c contains k information symbols x_i and $n-k$ check symbols. The complexity of such implementation is $O(nk)$ due to $O(k)$ operations for each of the $(n-k)$ check symbols. Systematic encoding algorithms with complexity $O(n \log n)$ are known for polar codes with Arikan kernel [4], [20], [21], [22]. The objective of this paper is to present a more efficient systematic encoding algorithm for polar codes with RS kernel.

Note that there may exist many different generator matrices in the systematic form. In order to design an efficient systematic encoding algorithm, we will need to adjust the generator matrix of a polar code, so that it admits application of some standard fast algorithms.

III. Efficient Erasure Decoding for Polar Codes

In this section we present an efficient erasure decoding algorithm for polar codes with RS kernel. It exploits their relationship with generalized concatenated codes, and can be considered as an instance of the multistage decoding algorithm. We show how this algorithm can be related to classical decoding algorithms for RS codes, and exploit their algebraic structure in order to reduce the complexity.

A. An iterative erasure decoding algorithm

Consider an $l\times N$ matrix C corresponding to a codeword of a generalized concatenated code with inner RS codes. Observe that the columns of matrices C' and C are related by the expression $C_{-,i}=B^TC'_{-,i}, 0\leq i<N$, i.e.

$$C_{j,i}=\sum_{t=0}^{l-1}B_{tj}C'_{t,i}=\sum_{t=0}^{l-1}\alpha_j^{l-1-t}C'_{t,i}. \quad (5)$$

This expression is very similar to the one used in the first definition of RS codes.

Lemma 1. *Consider a GCC with inner RS codes $(l,l-j,j+1), 0\leq j<l$. Assume that there are $s\leq l$ erasures in the i-th column of a GCC codeword matrix C. If values $C'_{t,i}, 0\leq t<s$ are known, then these erasures can be recovered.*

Proof. Let E be the set of indices t, such that $C_{t,i}$ are erased. One can rewrite (5) as

$$C_{j,i}=\underbrace{\sum_{t=0}^{l-s-1}\alpha_j^tC'_{l-1-t,i}}_{\tilde{c}_j}+\underbrace{\sum_{t=l-s}^{l-1}\alpha_j^tC'_{l-1-t,i}}_{\widehat{c}_j}.$$

The first term can be considered as the j-th symbol of $(l,l-s,s+1)$ RS code, and the second term depends only on $C'_{t,i}, 0\leq t<s$. One can apply an erasure decoding algorithm to vector $(C_{0,i}-\widehat{c}_0,\dots,C_{l-1,i}-\widehat{c}_{l-1})$, where the values in positions $j\in E$ are replaced with the erasure symbol ϵ, in order to recover $\tilde{c}_j$, and obtain eventually $C_{j,i}$. □

This lemma enables one to use the multistage algorithm shown in Figure 2 for erasure decoding of in an $(Nl,\sum_{i=0}^{l-1}K_i,\geq\min_i D_i(i+1))$ GCC with inner RS codes of length l. Note that similar approach was suggested for the case of interleaved RS polar concatenated codes [23].

In order to prove the correctness of this algorithm, we need to show that step 2 can be always performed.

Lemma 2. *After the multistage erasure decoding algorithm has reached the j-th iteration, each column of C has either 0, or at least $j+1$ erasures.*

Proof. The statement is trivial for $j=0$. Assuming that it is true for some $j<l-1$, consider the operations performed at iteration j. If outer code decoding does not fail at step 3, then values $C'_{t,i}, 0\leq t\leq j$ would be available, so that by Lemma 1 one can correct up to $j+1$ erasures in any column. Hence, at the beginning of iteration $j+1$, each column of C would contain either 0, or more than $j+1$ erasures. □

Input: an $l\times N$ codeword matrix C with erasures
Output: a codeword of the GCC.

1) Let $j\leftarrow 0$.
2) If there are $s>j$ erasures in the i-th column, $0\leq i<N$, of matrix C, then set $C'_{j,i}\leftarrow\epsilon$. Otherwise, compute $C'_{j,i}\leftarrow((B^T)^{-1}C_{-,i})_j$.
3) Perform erasure decoding of $(C'_{j,0},\dots,C'_{j,N-1})$ in the outer code A_j. If this fails, declare decoding error and stop.
4) Use symbols $C'_{t,i}, 0\leq t\leq j$, in order to recover erasures in columns i of C having at most $j+1$ erasures, as described in the proof of Lemma 1.
5) If there are still erasures in matrix C and $j<l-1$, let $j\leftarrow j+1$, go to step 2.

Fig. 2. Multistage decoding in a GCC

Theorem 1. *For a GCC with inner RS codes and outer (N,K_i,D_i) codes $\mathcal{C}_i$, the multistage erasure decoding algorithm can recover any configuration of up to $d-1$ erasures, where $d=\min_i D_i(i+1)$*

Proof. Consider decoding of a codeword with $d-1$ erasures. Lemma 2 implies that for any i at the beginning of iteration i the columns of matrix C have either 0, or at least $i+1$ erasures. Hence, the number of columns with erasures is at most $\left\lfloor\frac{d-1}{i+1}\right\rfloor\leq D_i-1$. This implies that erasure decoding of outer code $\mathcal{C}_i$ at step 3 would be successful, i.e. the algorithm does not declare a decoding failure. After l iterations, no columns with erasures can remain in the codeword being decoded. □

It appears that the multistage erasure decoding algorithm can recover some combinations of more than $d-1$ erasures.

Lemma 3. *Let $E_j\subset\{0,\dots,N-1\}$ be a family of erasure patterns, i.e. the sets of positions of erased codeword symbols, recoverable by outer codes $\mathcal{C}_j, 0\leq j<l$, respectively. If $E_0\supset E_1\supset\cdots\supset E_{l-1}$, then the multistage erasure decoding algorithm for the GCC with outer codes $\mathcal{C}_j$ and inner RS codes of length l can recover any erasure pattern, such that the i-th column of codeword matrix has at most*

$$w(i)=|\{j|i\in E_j\}|, 0\leq i<N,$$

erasures.

Proof. Let $C\in\mathbb{F}_q^{l\times N}$ be a codeword with at most $w(i)$ erasures in each column i. Due to nested property of erasure patterns E_j, one obtains $w(i)=1+\max_{j:i\in E_j}j$. Consider the j-th iteration of the multistage erasure decoding algorithm. At step 2 one obtains vector $(C'_{j,0},\dots,C'_{j,N-1})$, where erasures may occur only in positions given by E_j. Therefore, these erasures can be recovered at step 3, i.e. decoding does not fail. After the algorithm terminates, the columns may have no erasures. □

Input: vector c with erased and non-erased symbols.
Output: a codeword of the polar code.

1) Declare all erased codeword symbols c_i as "unknown", and non-erased ones as "known". Declare all frozen symbols $u_i, i \in \mathcal{F}$, as known.
2) If any node B has $t > 0$ unknown output symbols, then declare its input symbols with indices $0, \dots, t-1$ as "unknown", unless some of these symbols are already declared "known", in which case keep them "known". Furthermore, declare the remaining input symbols as "needed", unless they are declared "known". Apply this operation recursively.
3) Repeat until all codeword symbols become known:
   - A: If all output symbols of a node are "known", compute its "needed" input symbols (i.e. multiply the vector of output symbols of a node by an appropriate submatrix of B^{-1}) and declare them known.
   - B: If a node has t known input symbols, t unknown and $l-t$ known output symbols, recover unknown symbols by local decoding at the node as described in the proof of Lemma 1 (see also Section IV-B below). Declare all recovered output symbols as known.
   - C: If a node at layer m has at least one unknown input symbol, declare decoding failure.

Fig. 3. Iterative erasure decoding in a polar code

Recall, that polar codes can be considered as GCC, where outer codes are also polar ones. This enables one to employ the multistage erasure decoding algorithm recursively for decoding of outer codes, as shown in Figure 3. This algorithm can be considered as an implementation of the multistage erasure decoding algorithm for the case of polar codes. By abuse of notation, we denote by node B a device implementing multiplication of an input vector (shown on the left to it in Figure 1) by matrix B.

It is possible to reduce the complexity of this algorithm by performing some steps of it at the design time. Namely, if all input symbols of a node are frozen, i.e. set to zero, then all its output symbols are also zero, i.e. they can be declared known.

Observe that if the condition of Step C is not satisfied, this algorithm always completes successfully in a finite number of steps.

It can be seen that for each node in the encoding scheme steps A and B are executed at most once, and the number of such nodes is $\frac{n}{l}\log_l n$. The complexity of straightforward implementation of Steps A and B is $O(l^2)$. Hence, the complexity of the proposed algorithm is at most $O(nl\log_l n)$.

The proposed algorithm can be considered as an instance of successive cancellation decoding. It is also similar to the encoding algorithm for generalized error location codes [24].

B. Efficient implementation of Step A

Consider the case of $l = q = 2^\mu$, and assume without loss of generality that $\alpha_i = \alpha^i, 0 \le i < q-1, \alpha_{q-1} = 0$. It can be verified that

$$B = \begin{pmatrix} & & & & 0\\ & & W & & \vdots\\ & & & & 0\\ 1 & 1 & \dots & 1 & 1\end{pmatrix}, B^{-1} = \begin{pmatrix} & & & & 0\\ & & W^{-1} & & \vdots\\ & & & & 0\\ 1 & 0 & \dots & 0 & 1\end{pmatrix},$$

where $W_{ij} = \alpha^{(q-1-i)j}$, so that $Q = (W^{-1})^T$ is the discrete Fourier transform matrix, i.e. $Q_{ij} = \alpha^{ij}$. Hence, Step A, i.e. calculation of $(S'_0, \dots, S'_{\delta-1})^T = \tilde{B}(P_0, \dots, P_{q-1})^T$, where S'_i and P_j are input and output symbols of a node, δ is the number of needed input symbols, and $\tilde{B}$ is the submatrix of $(B^{-1})^T$ consisting of δ bottom rows, can be implemented using the FFT-based syndrome evaluation techniques described in [14] with complexity $O(\delta \log\log l)$ multiplications and at most $O(\delta l/\log l)$ summations, or using the method given in [25] with complexity at most $O(l\log^2 l)$.

IV. Efficient systematic encoding

The above described multistage/iterative erasure decoding algorithm can be used in order to implement systematic encoding. Indeed, one can assume that all $n-k = |\mathcal{F}|$ check symbols are erased, and recover them using erasure decoding. However, one needs to identify a specific set of codeword symbols, which can be used as check ones. We propose a specific symbol assignment method, which enables significant reduction of encoding complexity.

In section IV-A we provide a generalization of the method given in [4]. Section IV-B provides an improved implementation of Step B of the iterative erasure decoding algorithm.

A. Assignment of check symbols

The following theorem essentially states that check symbols can be placed on positions $R(j), j \in \mathcal{F}$, within the codeword, where $R(j)$ is the digit-reversal function. In order to prove that this approach works, we will recursively represent polar codes as generalized concatenated codes. At level τ of the recursion, one starts from a polar code with the set of frozen symbols $\mathcal{F}^{(p)} \subset \{0, \dots, l^{m-\tau}-1\}, p \in \{0, \dots, l-1\}^\tau$, and represents it as a GCC with outer polar codes given by the sets of frozen symbols $\mathcal{F}^{(p,t)} = \left\{i | 0 \le i < l^{m-\tau-1}, (tl^{m-\tau-1}+i) \in \mathcal{F}^{(p)}\right\}, 0 \le t < l$. Here p denotes a τ-dimensional index of the code.

Theorem 2. *Consider a polar code of length $n = l^m$ with RS kernel and the set of frozen symbols $\mathcal{F}$, such that $\mathcal{F}^{(p,t)} \supset \mathcal{F}^{(p,t+1)}$ for any $p \in \cup_{i=0}^{m-2}\{0, \dots, l-1\}^i$, $0 \le t < l-1$. If only codeword symbols $c_{R(j)}$ are erased, where $j \in \mathcal{F}$, then the multistage erasure decoding algorithm can recover all of them.*

Proof. For $m = 1$ the theorem essentially states that a single-layer (l, κ) polar code with the set of frozen symbols $\mathcal{F} = \{0, 1, \dots, l-\kappa-1\}$, i.e. a RS code, can recover a particular combination of $l-\kappa$ erasures. This statement is obviously true.

Assume that the theorem holds for some $m \ge 1$, i.e. that a polar code of length l^m with the set of frozen symbols $\mathcal{F}$ satisfying the above constraints can recover erasure pattern $E = \{R(j) | j \in \mathcal{F}\}$. Consider now a polar code of length

l^{m+1}. It can be represented as a GCC with inner RS codes of length l, and outer polar codes $\mathcal{C}_t$ of length l^m with the sets of frozen symbols $\mathcal{F}^{(t)}, 0 \leq t < l$. Observe that each of the sets $\mathcal{F}^{(t)}$ satisfies the conditions of the theorem.

Any codeword C of the considered polar code can be represented as an $l \times l^m$ matrix. Consider some erased symbol $c_j, j = \sum_{s=0}^{m} j_s l^{m-s}$, where $\sum_{s=0}^{m} j_s l^s \in \mathcal{F}, 0 \leq j_s < l$. In matrix representation of the codeword it is placed in column $j' = \sum_{s=0}^{m-1} j_s l^{m-1-s}$ and row j_m. Hence, row t has erasures in columns given by the set $E_t = \left\{R(j) | j \in \mathcal{F}^{(t)}\right\}, 0 \leq t < l$. By inductive assumption, erasure pattern E_t is correctable by outer code $\mathcal{C}_t$. Since $\mathcal{F}^{(t)} \supset \mathcal{F}^{(t+1)}, 0 \leq t < l-1$, the total number of erasures in column j' is $w(j')$. Hence, Lemma 3 implies that the iterative erasure decoding algorithm can recover all erasures in C as well.

Hence, the multistage erasure decoding algorithm can recover erasure pattern E for a code of length l^{m+1}. □

Example 1. Consider a polar code corresponding to the encoding scheme shown in Figure 1. Here p is an empty vector, $\mathcal{F} = \{0, 1, 2, 3\}$, so that $\mathcal{F}^{(0)} = \{0, 1, 2\}, \mathcal{F}^{(1)} = \{0\}, \mathcal{F}^{(2)} = \emptyset$. These sets satisfy the requirements of Theorem 2, so one can place check symbols on positions $0, 3, 6, 1$.

These check symbols can be recovered as follows. First, observe that the output of the 0-th node is always zero. Therefore, the 0-th input symbols of nodes $3, 4, 5$ are known. Let us declare symbols c_0, c_1, c_3, c_6 unknown, and c_2, c_4, c_5, c_7, c_8 known. Figure 4(a) shows known symbols as black circles, and unknown symbols as white circles. Hence, input symbol 1 of node 3 should be declared unknown.

Now both nodes 4 and 5 have one known input symbol (which is equal to zero) and one unknown output symbol. Hence, c_3 and c_6 can be recovered by erasure decoding of vectors (ϵ, c_4, c_5) and (ϵ, c_7, c_8) in $(3, 2, 2)$ RS code. Now all output symbols of nodes 4 and 5 are known, so one can compute all their input symbols (only input symbol 1 is actually needed) and declare them known, as shown in Figure 4(b).

After this node 1 has one known input symbol, two known and one unknown output symbols. The latter one (i.e. the input symbol 1 of node 3) can be recovered by erasure decoding of $(3, 2, 2)$ RS code, as shown in Figure 4(c).

Eventually, c_0 and c_1 are recovered by erasure decoding of the coset of $(3, 1, 3)$ RS code, which is given by input symbol 1 of node 3. Let us consider the latter operation in more details. The input symbols S'_0, S'_1, S'_2 of node 3 are related to its output symbols via expression $(S'_0, S'_1, S'_2) = (c_0, c_1, c_2)Q$, where $Q = B^{-1}$. Then one obtains

$$S'_0 = c_0 Q_{00} + c_1 Q_{10} + c_2 Q_{20} \quad (6)$$
$$S'_1 = c_0 Q_{01} + c_1 Q_{11} + c_2 Q_{21} \quad (7)$$
$$S'_2 = c_0 Q_{02} + c_1 Q_{12} + c_2 Q_{22}. \quad (8)$$

(6)–(7) can be recognized as syndrome decoding equations for a code with check matrix given by two topmost rows of Q^T. Equation (8) is neither needed, nor usable, since S'_2 cannot be recovered without knowledge of u_6, u_7, u_8, which are not available to the decoder. Given the values of c_2, S'_0, S'_1, one can recover from (6)–(7) the erased symbols c_0, c_1.

The standard way to construct polar codes is to select $\mathcal{F}$ as the set of indices of subchannels $W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i)$ induced by the polarizing transformation with the smallest Bhattacharyya parameters $Z_m^{(i)}$. No techniques are currently known for computing these parameters for the case of RS kernel and arbitrary channel. However, if $W(y|c)$ is a q-ary erasure channel (q-EC), the synthetic subchannels $W_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i)$ are also q-EC, and their Bhattacharyya parameters are simply the erasure probabilities, which can be expressed via the decoding error probability of the corresponding RS codes. It is possible to show that the Bhattacharyya parameter for the i-th subchannel induced by the polarizing transformation can be recursively computed as [2]

$$Z_m^{(i)} = \sum_{t=i''+1}^{l} \binom{l}{t} \left(Z_{m-1}^{(i')}\right)^t \left(1 - Z_{m-1}^{(i')}\right)^{l-t}, \quad (9)$$

where $i = i'' + li'$, $0 \leq i'' < l$, and $Z_0^{(0)}$ is the erasure probability of the underlying channel $W(y|c)$. It can be seen that $Z_v^{(i''+li')} \geq Z_v^{(i''+1+li')}, i'' < l-1$ for any $v > 0$. This together with Lemma 4.7 of [26] implies that polar codes constructed for the q-ary erasure channel satisfy the requirements of Theorem 2. Indeed, the Bhattacharyya parameters of the subchannels derived from $W_v^{(i''+li')}$ cannot be less than the the Bhattacharyya parameters of the corresponding subchannels derived from $W_v^{(i''+1+li')}, i'' < l-1$.

More detailed characterization of the set of recoverable erasure patterns is given by the following theorem.

Theorem 3. *Consider some erasure pattern E correctable by the multistage erasure decoding algorithm. Let $E_{\lambda,j}$ be the set of unknown output symbols of node j at some layer λ after step 2 of the MSD algorithm. If E' is another erasure pattern, such that $|E'_{\lambda,j}| = |E_{\lambda,j}|, 0 \leq j < l^{m-1}$, then E' is also correctable.*

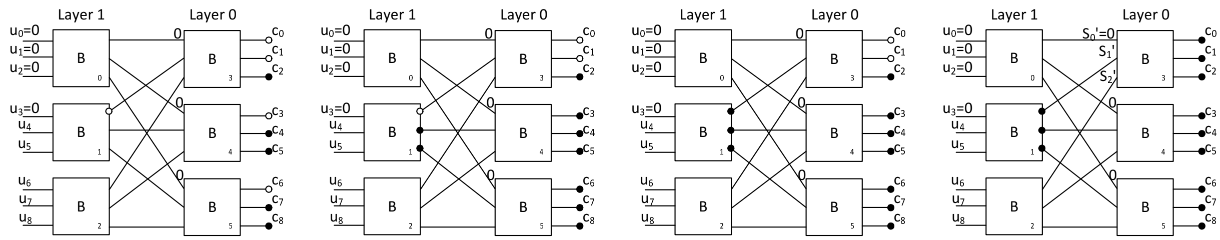
Proof. The set of unknown input symbols of all nodes at layer λ after step 2 is identical for both erasure patterns. Since these symbols are recoverable for E, they are recoverable for E' too. Hence, the condition of step C of the MSD algorithm is never satisfied, and it completes successfully in a finite number of steps. □

In order to implement systematic encoding for an (l^m, k) polar code, one needs to identify the positions of $l^m - k$ check symbols. This can be done using Theorem 2.

B. Efficient implementation of Step B

In this section we provide an efficient implementation of Step B, which becomes possible if the set of erased output symbols to be recovered by the nodes within the encoding scheme can be controlled at the design time.

Let $l = q = 2^\mu$. Let $(S_0, \ldots, S_{l-1})$ and $(P_0, \ldots, P_{l-1})$ be the input and output symbols of a node, respectively, and let $E = \{e_1, \ldots, e_\delta\}, \delta < l$, be the set of indices of unknown



(a) (b) (c) (d)

Fig. 4. An example of erasure recovery

output symbols. Assume that $\alpha_i = \alpha^i, 0 \leq i < q-1, \alpha_{q-1} = 0$. Then one obtains

$$S_i + \underbrace{\sum_{j \notin E} \alpha_j^i P_j}_{S_i'} = \sum_{j=1}^{\delta} \alpha_{e_j}^i P_{e_j}, 0 \leq i < \delta.$$

The values of S_i' can be computed using FFT-based syndrome evaluation techniques. Then erased symbols $P_{e_1}, \ldots, P_{e_\delta}$ can be obtained using the RS decoding techniques described in Section II-B.

If E is a union of a number of cyclotomic cosets $\{a_i, 2a_i, \ldots, 2^{m_i-1}a_i\}$, where $2^{m_i}a_i \equiv a_i \bmod 2^\mu - 1$, and $m_i|\mu$, then $\Lambda(x)$ is a polynomial with binary coefficients. Hence, computing (2) does not require any multiplications. Furthermore, (3) can be computed using the cyclotomic FFT method [25], [27] Indeed, let $\alpha^{-a_i s} = \sum_{t=0}^{m_i-1} A_{its}\gamma_i^{2^t}, A_{its}, \in \mathbb{F}_2$, be an expansion of $\alpha^{-a_i s}$ in a normal basis $\gamma_i, \ldots, \gamma_i^{2^{m_i-1}}$ of $\mathbb{F}_{2^{m_i}}$. Then

$$\Gamma(\alpha^{-a_i 2^j}) = \sum_{t=0}^{m_i-1} \gamma_i^{2^{t+j}} \sum_{s=0}^{l-\kappa-1} A_{its}\Gamma_s.$$

In matrix notation this becomes

$$\begin{pmatrix} \Gamma(\alpha^{-a_i}) \\ \Gamma(\alpha^{-2a_i}) \\ \vdots \\ \Gamma(\alpha^{-2^{m_i-1}a_i}) \end{pmatrix} = L_i A_i \begin{pmatrix} \Gamma_0 \\ \Gamma_1 \\ \vdots \\ \Gamma_{l-\kappa-1} \end{pmatrix},$$

where L_i is a circulant matrix consisting of $\gamma_i^{2^{t+j}}$ values, and A_i is a binary matrix.

Observe also that computing (2) is equivalent to multiplying a binary Toeplitz matrix $\tilde{\Lambda}$ with elements Λ_i by the vector of $S_i + S_i'$ values. Hence, the unknown output elements $P_{e_1}, \ldots, P_{e_\delta}$ of a node can be computed as

$$\begin{pmatrix} P_{e_1} \\ \vdots \\ P_{e_\delta} \end{pmatrix} = \underbrace{\text{diag}(\frac{\alpha_{e_i}}{\Lambda'(\alpha^{-e_i})}, 1 \leq i \leq \delta)}_{D} LA\tilde{\Lambda} \begin{pmatrix} S_0 + S_{0'} \\ \vdots \\ S_{\delta-1} + S_{\delta-1}' \end{pmatrix}, \quad (10)$$

where L is a block-diagonal matrix consisting of circulant matrices L_i. Multiplication by matrices L_i reduces to cyclic convolutions with total complexity $O(\delta \log \mu)$, and multiplication by binary matrix $A\tilde{\Lambda}$ can be implemented using either the recursive algorithm described in [25] with complexity $O(\delta \log^2 \delta)$, or computer-optimized algorithms [28], [29], which require at most $O(\delta^2 / \log \delta)$ operations. For small δ the latter approach turns out to be much more efficient. Observe that the straightforward evaluation of (2) and (3) requires $O(\delta^2)$ operations.

Hence, the complexity of Step B can be significantly reduced if unknown symbols are mapped onto a union of cyclotomic cosets. This can be always be done in the case of $\mu = 2^\tau$. Since $m_i|\mu$, one can construct decomposition $\delta = \delta'\mu + \sum_{i=0}^{\tau-1} \delta_i m_i, \delta_i \in \{0,1\}$, and construct E as a union of cyclotomic cosets of size μ and $m_i : \delta_i = 1$. This requires one to permute the output edges of the corresponding nodes in the encoding scheme, so that the top δ edges become connected to output ports given by set E. Essentially this means that different nodes in the encoding scheme implement multiplication of the input vectors by matrices $B\Pi$ with node-specific permutation matrices Π. It must be recognized that in general introducing such permutations results in a different code. However, this does not affect channel polarization properties, since these permutations are applied to identical synthetic channels.

Example 2. Consider the case of $l = 16, m = 2, \mathcal{F} = \{0, 1, 2, 3, 4, 16, 17, 18\}$, $\alpha_i = \alpha^i, 0 \leq i < 15, \alpha_{15} = 0$. That is, there are 5 and 3 frozen symbols at the input of nodes 0 and 1 at layer 1. It is advantageous to map erased symbols at the output of node 0 to positions $0, 1, 2, 4, 8$, and positions $0, 5, 10$ for node 1. In the case of node 1 one obtains $\Lambda(x) = (1+x)(1+\alpha^5 x)(1+\alpha^{10}x) = 1 + x^3$, $\Lambda'(x) = x^2$, so that D and $\tilde{\Lambda}$ in (10) become identity matrices, and $L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha^5 & 1+\alpha^5 \\ 0 & 1+\alpha^5 & \alpha^5 \end{pmatrix}$. Computing a product of this matrix and a vector requires just 1 multiplication by α^5, while the standard approach to evaluation of (3) would require at least 9 multiplications. In the case of node 0 recovery of 5 output symbols requires 5 multiplications.

Hence, the proposed approach enables one to perform systematic encoding of polar codes with RS kernel with complexity $O(n \log n \log l)$ or $O(nl\frac{\log n}{\log l})$. This is the same as the asymptotic complexity of non-systematic encoding, provided that the latter is implemented using FFT techniques. However, systematic encoding requires one to perform partial FFT, i.e. compute only a few components of the DFT. This has much smaller complexity than the generic FFT algorithm.

Example 3. For the $(256, 248)$ code considered in Example 2, the proposed encoding algorithm requires 89 multiplications. This is much less than the cost of a generic systematic encoding algorithm (1976 multiplications) and a non-systematic encoder, which employs FFT for multiplication by the kernel matrix (512 multiplications).

C. Extensions

The proposed approach can be used in the case of polar codes with kernel of size $l < q$. In order to do this, one should just assume that some $q - l$ output symbols of each node in the encoding scheme are equal to 0. Let Z be the set of such symbols. Technically, in this case the kernel is equal to a submatrix of matrix $B' = WB$, which consists of top l rows, and columns not in set Z, where W is an invertible matrix, such that top l rows of B' have zeroes in columns given by Z.

The proposed approach can be also used in the case of polar codes with mixed kernel [30], i.e. codes with the polarizing transformation given by $A = PB_1 \otimes B_2 \otimes \cdots \otimes B_m$, where B_i are RS kernels of various dimensions, and P is an appropriate permutation matrix.

V. Storage applications

Since polar codes are obtained via a recursive concatenation construction, they can be used to provide protection against hierarchical erasures with arbitrarily many layers in the hierarchy. Hence, as a possible application area of polar codes with RS kernel, we consider a storage system, which includes l_s servers, each server includes l_d storage devices, and each storage device includes l_b blocks[1]. Observe that any server failure implies failure of all the associated storage devices, and failure of any storage device implies failure of all associated blocks. In order to implement protection against block, device and server failures, one can encode the data with an $(n = l_s l_d l_b, k)$ polar code with polarizing transformation given by $A = PB_s \otimes B_d \otimes B_b$, where B_i is a $l_i \times l_i$ RS kernel, so that codeword symbol $c_{j_s l_d l_b + j_d l_b + j_b}$ is stored on block j_b on device j_d within server j_s. The set of frozen symbol indices $\mathcal{F}$ for this code can be designed by combining the following rules, depending on specific application requirements:

- Select $\mathcal{F}$ as the set of subchannel indices j with the highest values of $Z_j^{(3)}$ (see (9)), where $Z_j^{(0)}$ is the probability of block failure within a sufficiently large time interval.
- Include into $\mathcal{F}$ all such $j = j_s + l_s j_d + l_s l_d j_b : (j_b + 1)(j_d + 1)(j_s + 1) < d$. This ensures that the code can recover all combinations of $d - 1$ block failures.
- Include into $\mathcal{F}$ all $j < t l_s l_d$, where t is the number of block failures within each device the system needs to survive. This results in t zero input symbols for nodes at layer 0 in the encoding scheme, so that the iterative erasure decoding algorithm is able to recover up to t block failures on each device by accessing only the information on the same device.
- Include into $\mathcal{F}$ all $j = j_s + j_b l_s l_d, 0 \le j_s < l_s, 0 \le j_b < \phi$, where ϕ is the number of device failures, which can be recovered by the code. This ensures that there are ϕ zero input symbols for each node at layer 1. Observe that ϕ device failures within each server can be recovered by accessing the data only within the corresponding server.

Similar rules can be derived in order to implement protection against server failures.

Hence, in the context of storage systems, polar codes can be considered as a generalization of STAIR codes [31] and sector-disk codes [32]. The most important advantage with respect to these codes is that polar codes enable one to provide protection against failures of any entity (block, device, server, rack, etc) of a storage system. Furthermore, the proposed encoding algorithm in the case of $m = 2, l = l_d = l_b = q$ has complexity $O(n \log n \log l) = O(l^2 \log^2 l)$, while the complexity of the encoding algorithms for STAIR codes is $O(l^3)$. This gain is not only asymptotical, but appears also in practice, since it essentially corresponds to the gain provided by FFT-based RS encoding and decoding techniques [14], [15] with respect to the standard approach, which was shown in [15] to be quite significant even for small values of l.

VI. Conclusions

In this paper a low-complexity systematic encoding algorithm for polar codes with Reed-Solomon kernel was suggested. The proposed method is based on FFT encoding and decoding techniques introduced originally for the case of Reed-Solomon codes.

The proposed method can be used in storage systems, where polar codes with relatively big Reed-Solomon kernel can be used in order to implement protection against block, device and server failures, and enable local recovery of most typical failure configurations.

Acknowledgements

The authors thank K. Ivanov for his help in the implementation of the proposed method.

The authors thank the Associate Editor and the anonymous reviewers for their comments, which have greatly improved the quality of the paper.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2] R. Mori and T. Tanaka, "Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes," in *Proceedings of IEEE Information Theory Workshop*, Dublin, 2010, pp. 1–5.

[3] P. Trifonov, "Binary successive cancellation decoding of polar codes with Reed-Solomon kernel," in *Proceedings of IEEE International Symposium on Information Theory*, Honolulu, 2014, pp. 2972 – 2976.

[4] E. Arikan, "Systematic polar coding," *IEEE Communications Letters*, vol. 15, no. 8, pp. 860–862, August 2011.

[5] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Fast multi-dimensional polar encoding and decoding," in *Proceedings of Information Theory and Applications Workshop*, San-Diego, 2014, pp. 1–5.

[1] The actual number of blocks on a device may be much higher. However, one can assume that the device consists of superblocks, and each superblock contains l_b blocks.

[6] R. Mori and T. Tanaka, "Source and channel polarization over finite fields and Reed-Solomon matrix," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2720–2736, May 2014.

[7] N. Stolte, "Rekursive codes mit der Plotkin-konstruktion und ihre decodierung," Ph.D. dissertation, Technische Universität, Darmstadt, January 2002. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/183/

[8] M. El-Khamy, "Polar codes are OCBM codes," IEEE GLOBECOM Lightning Talk, San Diego, CA, USA, 2015.

[9] J. Massey, D. J. Costello, and J. Justesen, "Polynomial weights and code constructions," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 101–110, January 1973.

[10] G. A. Kabatyanskiy, "Two generalizations of code product," *Doklady Akademii Nauk SSSR*, vol. 232, no. 6, 1977, in Russian.

[11] K. Saints and C. Heegard, "On hyperbolic cascaded Reed-Solomon codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 1993, pp. 291–303.

[12] R. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1984.

[13] T. K. Moon, *Error Correction Coding. Mathematical Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2005.

[14] E. Costa, S. V. Fedorenko, and P. V. Trifonov, "On computing the syndrome polynomial in Reed-Solomon decoder," *European Transactions on Telecommunications*, vol. 15, no. 4, pp. 337–342, May/June 2004.

[15] P. Trifonov, "Low-complexity implementation of RAID based on Reed-Solomon codes," *ACM Transactions on Storage*, vol. 11, no. 1, pp. 1:1–1:25, February 2015.

[16] E. Blokh and V. Zyablov, "Coding of generalized concatenated codes," *Problems of Information Transmission*, vol. 10, no. 3, pp. 45–50, 1974.

[17] M. Bossert, *Channel coding for telecommunications*. Wiley, 1999.

[18] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3221 – 3227, November 2012.

[19] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.

[20] G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Flexible and low-complexity encoding and decoding of systematic polar codes," 2016.

[21] H. Vangala, Y. Hong, and E. Viterbo, "Efficient algorithms for systematic polar encoding," *IEEE Communications Letters*, vol. 20, no. 1, January 2016.

[22] G. T. Chen, Z. Zhang, C. Zhong, and L. Zhang, "A low complexity encoding algorithm for systematic polar codes," *IEEE Communications Letters*, 2016, accepted.

[23] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Performance limits and practical decoding of interleaved Reed-Solomon polar concatenated codes," *IEEE Transactions On Communications*, vol. 62, no. 5, pp. 1406–1417, May 2014.

[24] J. Maucher, V. Zyablov, and M. Bossert, "On the equivalence of generalized concatenated codes and generalized error location codes," *IEEE Transactions On Information Theory*, vol. 46, no. 2, pp. 642–649, March 2000.

[25] P. Trifonov, "On the additive complexity of the cyclotomic FFT algorithm," in *Proceedings of IEEE Information Theory Workshop*, Lausanne, 2012, pp. 537 – 541.

[26] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne, 2009.

[27] P. V. Trifonov and S. V. Fedorenko, "A method for fast computation of the Fourier transform over a finite field," *Problems of Information Transmission*, vol. 39, no. 3, pp. 231–238, July-September 2003, translation of Problemy Peredachi Informatsii.

[28] P. Trifonov, "Matrix-vector multiplication via erasure decoding," in *Proceedings of XI International Symposium on Problems of Redundancy in Information and Control Systems*, Saint-Petersburg, 2007, pp. 104–108.

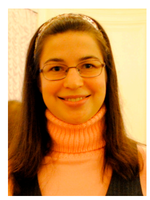
[29] S. Bellini, M. Ferrari, and A. Tomasoni, "On the structure of cyclotomic Fourier transforms and their applications to Reed-Solomon codes," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2110 –2118, August 2011.

[30] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Compound polar codes," in *Proceedings of Information Theory and Applications Workshop*, San-Diego, 2013, pp. 1–6.

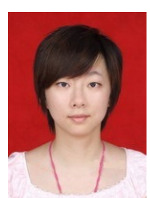
[31] M. Li and P. P. C. Lee, "STAIR codes: A general family of erasure codes for tolerating device and sector failures in practical storage systems," in *Proceedings of 12th USENIX Conference on File and Storage Technologies*, Santa-Clara, February 2014, pp. 147–162.

[32] J. S. Plank and M. Blaum, "Sector-disk (SD) erasure codes for mixed failure modes in RAID systems," *ACM Transactions on Storage*, vol. 10, no. 1, pp. 4:1–4:17, January 2014.

Peter Trifonov (S'02,M'05) was born in St.Petersburg, USSR in 1980. He received the MSc degree in computer science in 2003, and PhD (Candidate of Science) degree from Saint-Petersburg Polytechnic University in 2005. Currently he is an Associate Professor at the Distributed Computing and Networking department of the same university. His research interests include coding theory and its applications in telecommunications and other areas.



Vera Miloslavskaya (S12, M'15) received the B.Sc., M.Sc. and Ph.D. (Candidate of Science) degrees in computer science in 2010, 2012 and 2015, respectively, from Saint-Petersburg Polytechnic University, Russia. Her research interests include coding theory and its applications in telecommunications.



Chen Chen was born in Shannxi, China, in 1987. She received the B.S. degree in Mathematics from Beihang University in 2009, and the Ph.D. degree in Control Theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, in 2014. She is currently a researcher at Huawei Technologies Co. Ltd. Her research interests include complex systems, distributed filters and statistic machine learning.

Yuangang Wang is a Senior Researcher and Team Leader in Cognitive Computing Lab at Huawei Technologies Co. He performs research on storage architecture, file systems and nonvolatile memory.