Rateless and Pollution-Attack-Resilient Network Coding

Wentao Huang
California Institute of Technology

Ting Wang, Xin Hu, Jiyong Jang, Theodoros Salonidis
IBM T.J. Watson Research Center

***Abstract*—Consider the problem of reliable multicast over a network in the presence of adversarial errors. In contrast to traditional network error correction codes designed for a given network capacity and a given number of errors, we study an arguably more realistic setting that prior knowledge on the network and adversary parameters is not available. For this setting we propose efficient and throughput-optimal error correction schemes, provided that the source and terminals share randomness that is secret form the adversary. We discuss an application of cryptographic pseudorandom generators to efficiently produce the secret randomness, provided that a short key is shared between the source and terminals. Finally we present a secure key distribution scheme for our network setting.**

I. Introduction

A source wishes to multicast information to a set of terminals over a network. The technique of *network coding*, i.e., allowing routers to mix the information in packets before forwarding them, is able to maximize network throughput, improve robustness against packet losses, and can be efficiently implemented in a distributed manner [1], [2], [3]. However, network coding is vulnerable to packet transmission errors caused by adversarial jamming, as one single corrupted packet may pollute many more others in the process of mixing.

The use of coding approach to correct adversarial errors for network coding systems is introduced by [4], and capacity-achieving code constructions are studied for various adversary and network models, e.g. [5], [6], [7], [8], [9]. However, coding schemes usually need to assume a given capacity of the network and a number of links controlled by the adversary, for the purposes of code design, encoding and decoding. This assumption may be overly restrictive in many practical settings. For example, estimating the network capacity may be costly; the capacity may change over time; and the number of links controlled by the adversary may not be available. To address this issue, *rateless* network error correction codes, i.e., coding schemes that do not require prior knowledge of the network and adversary parameters, are studied in [10]. However, designing low-complexity rateless coding schemes is still an open problem.

This paper proposes simple, efficient and rateless error correction schemes for network coding systems. Our schemes are based on rateless network error correction codes, and do not require priori estimates of the network capacity and the number of errors. The codes can be efficiently encoded and decoded, and are end-to-end in the sense that only the source and the terminals need to participate in error correction. The codes are asymptotically throughput-optimal and are universal in the sense that it is oblivious to the underlying network topology and to the particular linear network coding operations performed at intermediate network nodes.

We design the rateless network error correction code first under the assumption that the source and terminals share random bits which are secret from the adversary. The encoder uses the random bits to hash the message and helps the decoders eliminate injected errors. Over time, the source incrementally sends more linearly dependent redundancy of the message as well as a sequence of short hashes through the network. The terminal amasses both the redundancy and the hashes until it decodes successfully, which happens with high probability once the amount of redundancy received meets the cut-set bound. Then we note that the secret random bits can be efficiently generated by a cryptographic pseudorandom generator, provided that the adversary is computationally bounded, and that the source and terminals share a short key. Finally, we discuss an information-theoretically secure key distribution scheme for the network. The scheme is able to transmit a few bits secretly and reliably if the adversary eavesdrops only on a limited number of links. The key-distribution scheme follows a similar idea as the one proposed in [8], and is augmented to operate in a rateless manner.

Compared to the rateless network error correction code of complexity $O(n^3)$ proposed in [10], where n is the packet length, the complexity of our scheme is $O(n)$. Moreover, the communication overhead (amount of hash transmitted) of our scheme is significantly lower. Finally, we introduce a way to substantially reduce the field size by using universal hashing.

We put our schemes also into perspective with the rich collection of works, e.g., [11], [12], [13], [14], [15], [16], on cryptographic error control schemes for network coding systems, which detect and remove error packets injected by the adversary without the need of information-theoretic error correction. Cryptographic schemes also operate in a rateless manner independently from the network and adversary parameters. However, in order to remove error packets promptly before they contaminate others, frequent cryptographic verification of packets is necessary at intermediate network nodes for these schemes. By contrast, our schemes are end-to-end and do not require any collaboration from intermediate network nodes.

II. Models and Definitions

A. Network Model

A network is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices $\mathcal{V}$ represents network nodes and the set of edges $\mathcal{E}$ represents noiseless network links. The network operates in a synchronized manner and each link can send a symbol from a finite field $\mathbb{F}_q$ per transmission. A source $s \in \mathcal{V}$ wishes to communicate reliably to a terminal $t \in \mathcal{V}$[1]. Denote by C the min-cut (or max-flow) of the network with respect to s and t. The linear network code $\mathcal{C}$ implemented in $\mathcal{G}$ is represented by a set of encoding functions $\mathcal{C} = \{f_e : e \in \mathcal{E}\}$. For $e = (u, v)$, f_e is a linear function taking as input the signals received from all incoming edges of node u, and evaluates to the signal transmitted on e. The source and the terminal have no prior knowledge about C and $\mathcal{C}$. For notational convenience, in the rest of the paper we assume C and $\mathcal{C}$ do not change over time, whereas our results generalize to the case that they do change.

Denote by $\bar{C}$ the number of out-going edges of s, so $\bar{C} \geq C$. To transmit information over the network, the source generates a batch of $\bar{C}$ encoded packets of length n as input to the network, represented by a matrix $X \in \mathbb{F}_q^{\bar{C}\times n}$, where packets are rows. As the packets travel through the network, they undergo linear transforms defined by the network code $\mathcal{C}$. Without loss of generality we assume $\mathcal{C}$ is capacity-achieving[2], i.e., in the absence of adversarial errors, the terminal t will observe a matrix AX, where $A \in \mathbb{F}_q^{C\times \bar{C}}$ is the network transform matrix of rank C. Note that A is not known to the source and the terminal.

B. Adversary Model

The adversary controls $z_e < C$ edges[3] in the network, modeled in the following way. For each compromised edge (u, v), the adversary injects an error packet so that the packet received by v from this edge is the addition (over $\mathbb{F}_q$) of the error packet and the packet originally transmitted on the edge. As the injected error packets travel through the network, they undergo linear transforms defined by the network code $\mathcal{C}$. The terminal receives the sum of the linearly transformed error packets and the linearly transformed X. More precisely, the terminal observes a matrix $Y = AX + BZ$, where $B \in \mathbb{F}_q^{C\times z_e}$ is the network transform matrix (determined by the network code) from the compromised edges to t, and $Z \in \mathbb{F}_q^{z_e\times n}$ are the z_e injected error packets. The adversary may choose Z carefully in order to corrupt the communication between s and t. Note that z_e, Z and B are not known to the source and the terminal.

C. Throughput and Capacity

We call one *stage* as the transmission of one batch of encoded packets, i.e., a matrix X. For a scheme involving N stages, denote by X^N and Y^N the sequences of matrices transmitted by s and received by t. Let M be a source message chosen from an alphabet $\mathcal{M}$. If there exists a scheme that maps M to X^N, and maps Y^N to $\hat{M}$, such that for all $M \in \mathcal{M}$, regardless of the errors injected by the adversary, $\Pr\{M \neq \hat{M}\} \to 0$ as $q \to \infty$, then we say a throughput of $\frac{\log_q |\mathcal{M}|}{Nn}$ is feasible and is achieved by the corresponding scheme. The capacity of the network is the supremum over all feasible throughputs.

III. Rateless Network Error Correction

In this section we describe the rateless network error correction code. We assume that the source and the terminal share secret randomness. Formally, the source and the terminal agree on a sequence of symbols i.i.d. uniformly drawn from $\mathbb{F}_q$. The sequence of symbols are drawn secretly from the adversary, and are independent from the source message M. Non-rateless network error correction under this model is initially studied in [9]. The shared secret randomness are helpful for two reasons. Firstly, it increases the capacity of the network [5]. More precisely, the network capacity is $C - 2z_e$ if the adversary is all-knowing; and it increases to $C - z_e$ if there is a secret that the adversary does not know. Secondly, the shared secret randomness facilitates hashing and verification of the packets, independently from the value of z_e. Particularly, the scheme described in this section needs only a very small amount of secret randomness. In Section IV we will describe another scheme that uses more secret randomness, but operates in a smaller field.

At a high level, in our scheme the encoder incrementally sends more linearly dependent redundancy of the message through multiple stages. The message will be contained in the row space of the received matrices of packets after a number of stages. Additionally, the source sends a sequence of short hashes to facilitate the decoder to pinpoint the message from the row space. Hash transmissions are protected by strong redundancy. Since the hashes are short, the induced overhead is small and is negligible in the packet length. Below we describe the encoder for the source and the decoder for the terminal.

A. Encoder

Suppose the source wishes to transmit a message of b packets, each consisted of n symbols from $\mathbb{F}_q$, represented by a $b\times n$ matrix M over $\mathbb{F}_q$. The communication of M may last for several stages and during stage i, the source draws a random matrix K_i of size $\bar{C}\times b$ with entries i.i.d. uniformly distributed on $\mathbb{F}_q$. The source encodes $X_i = K_i M$, and inputs X_i to the network. Thereafter X_i undergoes the network transform as it travels through the network, as described in Section II-A and II-B.

Next we discuss the construction of the hash. Recall that the vectorization of a matrix is a linear transformation which converts the matrix into a column vector by stacking the columns of the matrix on top of one another. Let the column vector $\boldsymbol{m} \in \mathbb{F}_q^{bn}$ be the vectorized M. Let $\alpha_1 = \bar{C}(b+1) + t$, $t \geq 1$ and $\alpha_i = \bar{C}(b+1)$, $i > 1$ be the length of the hash

[1] For notational convenience we assume there is a single terminal. Our results generalize naturally to multicast problems with multiple terminals.

[2] Otherwise if $\mathcal{C}$ is not capacity-achieving, then some edges are redundant. Remove these edges and $\mathcal{C}$ is capacity achieving in the new network.

[3] We consider a model where edges rather than nodes are corrupted. Corrupting a node is equivalent to corrupting the links outgoing from it.

constructed at the i-th stage. The source draws α_i symbols $\boldsymbol{r}_i = (r_1, ..., r_{\alpha_i})$, and another α_i symbols $\boldsymbol{h}_i = (h_1, ..., h_{\alpha_i})$ uniformly i.i.d. distributed over $\mathbb{F}_q$, from the shared secret randomness. Let $D_i \in \mathbb{F}_q^{\alpha_i \times nb}$ be the matrix whose (u, v)-th entry equals r_u^v, then compute the length-α_i column vector

$$\boldsymbol{l}_i = \boldsymbol{h}_i - D_i \boldsymbol{m}, \tag{1}$$

which is the hash of message $\boldsymbol{m}$. To communicate $\boldsymbol{l}_i$, during the i-th stage, the source draws an random vector $\bar{K}_i \in \mathbb{F}_q^{\bar{C} \times 1}$ with entries i.i.d uniformly distributed over $\mathbb{F}_q$. It then encodes $\bar{X}_i = \bar{K}_i \boldsymbol{l}_i^T$, and inputs $\bar{X}_i$ into the network. Alternatively, the source may include $\bar{X}_i$ as a small header when it sends X_i. This concludes the operations of the encoder. Thereafter $\bar{X}_i$ travels through the network and undergoes the network transform described in Section II-A and II-B.

For stage i, the number of the shared random secret symbols required is $2\alpha_i$, which is asymptotically negligible in n. The communication overhead, i.e., the number of hash-related transmissions, is the length of $\boldsymbol{l}_i$, which equals α_i. Again this is asymptotically negligible in n. The computational cost of encoding is dominated by the operation of computing $\boldsymbol{l}_i$ from (1), which is bounded by $O(\bar{C}b^2 n)$.

We remark that the hashing scheme may be understood in the following way. In (1) if we regard $D_i \boldsymbol{m}$ as the hash of $\boldsymbol{m}$ with respect to a parity check matrix D_i, then $\boldsymbol{l}_i$ is the one-time padded version of $D_i \boldsymbol{m}$.

B. Decoder

During the i-th stage the terminal receives a batch of packets from the network $Y_i = AX_i + B_i Z_i$ corresponding to X_i. The terminal also receives a batch of packets $\bar{Y}_i = \bar{A}\bar{X}_i + \bar{B}_i \bar{Z}_i$ corresponding to $\bar{X}_i$ (alternatively if $\bar{X}_i$ is the header of X_i, then $\bar{Y}_i$ is the header of the received packets), where $\bar{A}$, $\bar{B}_i$ and $\bar{Z}_i$ are defined similarly as A, B_i and Z_i, cf. Section II-A and II-B.

The decoder obtains a matrix $P_{i,1}$ from the shared randomness as:

$$P_{i,1} = \begin{bmatrix} D_1 & I_{\alpha_1} & 0 & ... & 0 \\ D_2 & 0 & I_{\alpha_2} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_i & 0 & 0 & ... & I_{\alpha_i} \end{bmatrix},$$

where I_{α_j} is the identity matrix of order α_j. Denote by $\otimes$ the Kronecker product, the decoder obtains a matrix $P_{i,2}$ from the received packets:

$$P_{i,2} = \begin{bmatrix} (Y^i)^T \otimes I_b & 0 & \cdots & 0 \\ 0 & \bar{Y}_1^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{Y}_i^T \end{bmatrix},$$

where I_b is the identity matrix of order b, and

$$Y^i = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \end{bmatrix}. \tag{2}$$

Let $P_i = P_{i,1} P_{i,2}$ be a matrix of size $t + i\bar{C}(b+1) \times iC(b+1)$. Then the decoder solves the system of equations (3) in variables $\boldsymbol{x}^s$ and $\bar{\boldsymbol{x}}_k^s$, $k = 1, ..., i$, where $\boldsymbol{x}^s$ is a column vector of length ibC and the $\bar{\boldsymbol{x}}_j^s$'s are column vectors of length C.

$$P_i \begin{bmatrix} \boldsymbol{x}^s \\ \bar{\boldsymbol{x}}_1^s \\ \vdots \\ \bar{\boldsymbol{x}}_i^s \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_i \end{bmatrix}. \tag{3}$$

If (3) is not uniquely solvable, i.e., if it has no solution or if there are multiple solutions, the terminal postpones decoding to the next stage so that it will receive more redundancy. If (3) is uniquely solvable, unvectorize $\boldsymbol{x}^s$ into a $b \times iC$ matrix X^s by rearranging every length-b segment of $\boldsymbol{x}^s$ as a column of X^s. Then the source message is recovered as:

$$\hat{M} = X^s Y^i. \tag{4}$$

Note that the size of X^s is much smaller than the size of $\hat{M}$. Therefore, we may regard X^s as a proxy for solving $\hat{M}$ and it improves computational efficiency to first solve X^s from (3) and then obtain $\hat{M}$ from (4). Particularly, the complexity of solving X^s from (3) is bounded by $O(i^3 b^3 \bar{C}^3)$, and is asymptotically negligible in n. The computational cost of decoding is dominated by the matrix multiplication to obtain P_i, which is bounded by $O(i^2 \bar{C}^2 b^2 n)$.

C. Performace

The error performance of the proposed scheme is analyzed in Theorem 1. The proofs of Theorem 1 and the related lemmas are available in Appendix A.

Theorem 1. *Let* i *be the smallest integer such that* $\mathrm{i}(C - z_e) \geq b$*, then the terminal is able to decode* M *correctly after collecting packets for* i *stages, with probability at least* $1 - (\mathrm{i}+1)/(q-1) - \mathrm{i}(nb)^{t+\mathrm{i}(b+1)\bar{C}}/q^t$.

We remark that $\mathrm{i}(C - z_e) \geq b$ is in fact the cut-set bound, i.e., the number of packets received is larger than or equal to the number of message packets plus the number of injected error packets. Therefore Theorem 1 suggests that the scheme is asymptotically throughput-optimal in the sense that the terminal will decode correctly with high probability as $q \to \infty$, after collecting packets for the least possible number of stages.

IV. Computationally Bounded Adversary

The coding scheme described in Section III does not impose restrictions on the computation capability of the adversary. However, it requires shared secret randomness between the source and the terminal. In many practical settings, it may be more reasonable to assume that the adversary is computationally bounded than to assume the availability of perfect shared randomness. For this case, it is natural to replace the shared secret randomness by pseudorandomness generated by a pseudorandom generator. Assuming that the source and the terminal share a short secret key e, they may use a pseudorandom generator to create shared pseudorandomness with e. Then the source and terminal may invoke the scheme in Section III using the generated pseudorandomness.

Definition 1. *A pseudorandom generator with secure parameter ϵ is a deterministic function $PRG : \mathcal{S} \to \{0,1\}^N$, where $\mathcal{S}$ is the set of all keys, such that for every computationally efficient statistical test $E : \{0,1\}^N \to \{0,1\}$, it follows that*

$$|\Pr\{E(PRG(e)) = 1\} - \Pr\{E(x) = 1\}| < \epsilon, \tag{5}$$

where e is uniformly chosen from $\mathcal{S}$ and x is uniformly chosen from $\{0,1\}^N$.

The existence of pseudorandom generators is an open problem and is equivalent to the existence of one-way functions [17]. In practice, a number of candidates of highly efficient pseudorandom generators can be found in the eSTREAM portfolio [18]. For example, with a key of length 256 bits, Salsa20 is conjectured to achieve $\epsilon = 2^{-64}$ for $N = 512$.

Assuming that the adversary is computationally bounded, Theorem 2 shows that the modified scheme is reliable as long as the pseudorandom generator is secure.

Theorem 2. *Let i be the smallest integer such that $\mathrm{i}(C - z_e) \geq b$. If a pseudorandom generator of secure parameter ϵ is employed to generate the shared randomness between the source and terminal, then the terminal is able to decode M correctly after collecting packets for i stages, with probability at least $1 - (\mathrm{i}+1)/(q-1) - \mathrm{i}(nb)^{t+\mathrm{i}(b+1)\bar{C}}/q^t - \epsilon$.*

The proof of Theorem 2 is available in Appendix B.

A. Reducing Field Size

Theorem 1 and 2 suggest that the terminal will decode correctly at the earliest possible stage with high probability provided the field size q is large. Particularly, q has to scale faster than $(nb)^{1+\frac{\mathrm{i}bC}{t}}$. However, from the perspective of practical implementation, a large q is undesirable. In this subsection we describe two approaches to relax the requirement on the field size. The first way is to choose a reasonably large t so that $t > \mathrm{i}b\bar{C}$. Then it suffices to let q be comparable to n^2b^2. For most applications, this field size is reasonably small and is amenable to implementation. In the following, we discuss another approach to further reduce the field size.

The reason that q has to scale up is due to the fact that collisions in hash (1) may occur more frequently as n, b or C grows. Namely, for fixed q and $\boldsymbol{m}$, there exists $\boldsymbol{m}^* \neq \boldsymbol{m}$ such that the probability that the hash of $\boldsymbol{m}^*$ collides with the hash of $\boldsymbol{m}$ will increase as n, b or C grows. The idea therefore is to employ a hashing scheme with a constant collision probability, which may be achieved in the following way. We construct a matrix $\hat{D}_i$ from the shared randomness such that $\hat{D}_i$ is a matrix of size $\alpha_i \times nb$ with entries uniformly i.i.d. distributed over $\mathbb{F}_q$. The hash of message $\boldsymbol{m}$ is then computed as:

$$\boldsymbol{l}_i = \boldsymbol{h}_i - \hat{D}_i\boldsymbol{m}. \tag{6}$$

Proposition 1 shows that the hashing scheme (6) has a collision probability similar to the one of a hashing scheme that assigns truly random hashes to every $\boldsymbol{m}$. Hashing schemes with this property are called universal families of hash functions [19].

Proposition 1. *For $\boldsymbol{m} \neq \boldsymbol{m}^*$, let $\boldsymbol{l}_i = \boldsymbol{h}_i - \hat{D}_i\boldsymbol{m}$ and $\boldsymbol{l}_i^* = \boldsymbol{h}_i - \hat{D}_i\boldsymbol{m}^*$, then $\Pr\{\boldsymbol{l}_i = \boldsymbol{l}_i^*\} \leq 1/q^{\alpha_i}$, where the probability is taken over the distribution of $\hat{D}_i$ and $\boldsymbol{h}_i$.*

The proofs of Proposition 1 and the following Theorem 3 are available in Appendix C.

Theorem 3. *Let i be the smallest integer such that $\mathrm{i}(C - z_e) \geq b$. If the hashes are generated using (6), then the terminal is able to decode M correctly after collecting packets for i stages, with probability at least $1 - (\mathrm{i}+1)/(q-1) - \mathrm{i}/q^t$.*

Theorem 3 shows that the universal hash (6) substantially reduce the probability of error and the requirement on the field size. However (6) needs significantly more shared secret randomness than (1). Hence there is a tradeoff between the size of the field and the size of the shared randomness required. In practice, as there exist efficient pseudorandom generators capable of producing a large amount of pseudorandomness, it may be desirable to employ (6) together with such pseudorandom generators. We note that as shown in Theorem 2, replacing perfect randomness with pseudorandomness will not increase the probability of decoding error by more than ϵ. Finally, we emphasize that although (6) requires more shared randomness, the length of the resulting hash and the induced communication overhead is small as before.

V. Secure Key Distribution

Section IV discusses efficient and rateless error correction for network coding provided that the source and the terminal share a short secret key. The key may be pre-allocated, communicated by a secure side-channel, or communicated over the network by using public key infrastructure while disabling network coding. The question is, if the above options are not available, is it possible to communicate the key via the same network coding system.

In this section we discuss an information-theoretical scheme to communicate a short key secretly and reliably over the network, provided that the adversary has limited eavesdropping capability. Specifically, instead of allowing the adversary to observe all edges in the network, we assume that the adversary can eavesdrop on at most a number of z_w edges in the network, such that $z_e + z_w < C$. We assume that the source has an upper bound on the passive parameter z_w, and assume as before that z_e and C are not known to the source and the terminal.

Our scheme follows a similar key idea as the one proposed in [8], and is augmented to operate in a rateless manner. The idea is that one bit of information is represented by the rank of a matrix transmitted. To send a bit of 0 a low-rank matrix is generated and transmitted; to send a bit of 1 a full-rank matrix is generated and transmitted. The adversary sees only a limited number of edges and therefore cannot distinguish which bit is sent. The adversary has limited capability to change the rank of the transmitted matrix and therefore the terminal, by testing the rank of the received matrix, can distinguish the bits. Below we describe the detailed encoder and decoder to transmit k bits secretly and reliably.

A. Encoder

Denote the k bits to be transmitted by $s_1, ..., s_k$. For each of them a matrix $S_i \in \mathbb{F}_q^{(\bar{C}-z_w)\times\bar{C}(\bar{C}-z_w)}$ is generated. S_i is a zero matrix if $s_i = 0$ or a random matrix if $s_i = 1$. Let $N_i \in \mathbb{F}_q^{z_w\times\bar{C}(\bar{C}-z_w)}$ be a random matrix,

$$W_i = \left[\begin{array}{c} S_i \\ N_i \end{array} \right], \tag{7}$$

and $W = [W_1, ..., W_k]$. We now apply the universal secure network code [20]. Let $Q = q^{\bar{C}}$ and let $\mathbb{F}_Q$ be the degree $\bar{C}$ extension field of $\mathbb{F}_q$. Recall that $\mathbb{F}_Q$ is a vector space over $\mathbb{F}_q$, and let $\phi : \mathbb{F}_q^{1\times\bar{C}} \to \mathbb{F}_Q$ be the vector space isomorphism. Slightly abusing notations we let $\phi(S_i)$ and $\phi(N_i)$ denote the matrices obtained by applying the vector space isomorphism ϕ to each row of S_i or N_i, so that each length-$\bar{C}$ row segment maps to a symbol of $\mathbb{F}_Q$. And we denote $\phi(W)$ as

$$\phi(W) = \left[\begin{array}{ccc} \phi(S_1) & \cdots & \phi(S_k) \\ \phi(N_1) & \cdots & \phi(N_k) \end{array} \right]. \tag{8}$$

Then let $H \in \mathbb{F}_Q^{(\bar{C}-z_w)\times\bar{C}}$ be a parity check matrix of a $(\bar{C}, z_w)$ maximum rank distance (MRD) code over $\mathbb{F}_Q$. Without loss of generality we may assume $H = [I \;\; P]$. The source then computes:

$$X_Q = \left[\begin{array}{cc} I & -P \\ 0 & I \end{array} \right] \phi(W), \tag{9}$$

where X_Q is a $\bar{C} \times k(\bar{C} - z_w)$ matrix over $\mathbb{F}_Q$. Finally, the encoder obtains:

$$X = [I \;\; \phi^{-1}(X_Q)], \tag{10}$$

where X is a $\bar{C} \times (k\bar{C}(\bar{C} - z_w) + \bar{C})$ matrix over $\mathbb{F}_q$ since ϕ^{-1} expends each entry of X_Q into a length-$\bar{C}$ row vector over $\mathbb{F}_q$. Finally, the source inputs X into the network and thereafter it undergoes the network transforms.

B. Decoder

The terminal observes a batch of packets from the network $Y = AX + BZ$. The main objective of the decoder is to obtain from Y a good estimation of X_Q, such that it is close to X_Q in rank distance. The decoder can accomplish this by using the method described in [7], [21]. Specifically, the decoder performs a reduction transformation [21, Section 5.1.2] on Y, which essentially involves a row reduction on Y and an insertion of zero rows. From this the decoder will obtain a matrix $L \in \mathbb{F}_Q^{\bar{C}\times\mu}$; a set of matrices $R_i \in \mathbb{F}_Q^{\bar{C}\times(\bar{C}-z_w)}$, $i = 1, ..., k$; and a set of matrices $E_i \in \mathbb{F}_Q^{\delta\times(\bar{C}-z_w)}$, $i = 1, ..., k$. These matrices are useful because they are related to X_Q in the following way. Divide X_Q into $[X_1 \;\cdots\; X_k]$ where X_i is a $\bar{C} \times (\bar{C} - z_w)$ matrix over $\mathbb{F}_Q$, then by [21, Theorem 5.4], there exist matrices $U_{L,i}$, $U_{E,i}$ and U_i such that $\text{rank}(U_i) \leq z_e - \max\{\mu + C - \bar{C}, \delta\}$, and such that

$$R_i = X_i + LU_{L,i} + U_{E,i}E_i + U_i. \tag{11}$$

To decode, the terminal solves for full rank matrices $J \in \mathbb{F}_Q^{(\bar{C}-z_w-\mu)\times(\bar{C}-z_w)}$ and $K_i \in \mathbb{F}_Q^{(\bar{C}-z_w)\times(\bar{C}-z_w-\delta)}$ such that $JHL = 0$ and $E_iK_i = 0$. Finally, the decoder tests the rank of the matrix JHR_iK_i and decodes $s_i = 1$ if the matrix is full rank or $s_i = 0$ otherwise.

C. Performance

Theorem 4 follows from the property of the universal secure network code, and shows that the adversary, by eavesdropping on any z_w edges, learns no information about the S_i's.

Theorem 4. *[20, Theorem 7] For all $A_w \in \mathbb{F}_q^{z_w\times\bar{C}}$, A_wX is independent from $\{S_i\}_{i=1}^k$.*

Theorem 5 shows that the terminal decodes the bits correctly with high probability.

Theorem 5. *If $s_i = 0$, then JHR_iK_i is not full rank. If $s_i = 1$ then JHR_iK_i is full rank with probability at least $1 - 1/(Q - 1)$.*

The proof of Theorem 5 is available in Appendix D.

VI. Conclusion

This paper proposes simple, efficient and rateless error correction schemes for network coding systems. The schemes do not require priori estimates of the network capacity and the number of errors. The schemes can be efficiently encoded and decoded, and are end-to-end in the sense that only the source and the terminals need to participate in error correction. The schemes are asymptotically throughput-optimal and are universal in the sense that it is oblivious to the underlying linear network code and network topology.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Info. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Trans. Info. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[3] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Info. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[4] R. W. Yeung and N. Cai, "Network Error Correction, I: Basic Concepts and Upper Bounds," *Communications in Information & Systems*, vol. 6, no. 1, pp. 19–35, 2006.

[5] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *IEEE Trans. Info. Theory*, vol. 54, no. 6, pp. 2596–2603, 2008.

[6] R. Koetter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *IEEE Trans. Info. Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[7] D. Silva, F. R. Kschischang, and R. Kotter, "A Rank-Metric Approach to Error Control in Random Network Coding," *IEEE Trans. Info. Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[8] H. Yao, D. Silva, S. Jaggi, and M. Langberg, "Network Codes Resilient to Jamming and Eavesdropping," *IEEE/ACM Transactions on Networking*, vol. 22, no. 6, pp. 1978–1987, 2014.

[9] L. Nutman and M. Langberg, "Adversarial Models and Resilient Schemes for Network Coding," in *IEEE ISIT*, 2008, pp. 171–175.

[10] W. Huang, T. Ho, H. Yao, and S. Jaggi, "Rateless resilient network coding against byzantine adversaries," in *IEEE INFOCOM mini conference*, 2013, pp. 265–269.

[11] C. Gkantsidis and P. Rodriguez, "Cooperative Security for Network Coding File Distribution." *IEEE INFOCOM*, pp. 1–13, 2006.
[12] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a Linear Subspace: Signature Schemes for Network Coding," in *Public Key Cryptography*. Springer Berlin Heidelberg, 2009, pp. 68–87.
[13] F. Zhao, T. Kalker, M. Medard, and K. J. Han, "Signatures for Content Distribution with Network Coding," in *IEEE ISIT*, 2007, pp. 556–560.
[14] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE Authentication for Network Coding," *IEEE INFOCOM*, pp. 1–9, 2010.
[15] F. Oggier and H. Fathi, "An Authentication Code Against Pollution Attacks in Network Coding," *IEEE/ACM Transactions on Networking. on Networking*, vol. 19, no. 6, pp. 1587–1596, Dec. 2011.
[16] X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 33–42, Jan. 2014.
[17] O. Goldreich, *Foundations of Cryptography*. Now Publishers Inc, 2005.
[18] eSTREAM. [Online]. Available: http://www.ecrypt.eu.org/stream/
[19] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," in *annual ACM symposium*. New York, New York, USA: ACM Press, 1977, pp. 106–112.
[20] D. Silva and F. R. Kschischang, "Universal Secure Network Coding via Rank-Metric Codes," *IEEE Trans. Info. Theory*, vol. 57, no. 2, pp. 1124–1135, 2011.
[21] D. Silva, "Error control for network coding," Ph.D. dissertation, University of Toronto.

APPENDIX A
PROOF OF THEOREM 1

In this appendix we analyze the probability of decoding error. If decoding is not successful then either of the following two error events must happen: 1) there does not exists X^s such that $X^sY^i = M$ and $\boldsymbol{x}^s$ is a solution of (3); or 2) there exists X^{s*}, such that $X^{s*}Y^i \neq M$ and $\boldsymbol{x}^{s*}$ is a solution of (3), where $\boldsymbol{x}^{s*}$ is the vectorized X^{s*}.

We first study the probability of the first error event. We start with a useful lemma.

Lemma 1. *Let A be an arbitrary $u \times v$ matrix with rank u and $\boldsymbol{k}$ be a length-v column vector with entries uniformly i.i.d. distributed over $\mathbb{F}_q$, then $A\boldsymbol{k}$ is a random vector with entries uniformly i.i.d. distributed over $\mathbb{F}_q$*

Proof. Consider an arbitrary length-u column vector $\boldsymbol{y} \in \mathbb{F}_q^u$, and the system of equations $A\boldsymbol{x} = \boldsymbol{y}$. Because A has full row-rank, its columns span F_q^u. So $A\boldsymbol{x} = \boldsymbol{y}$ has a solution $\boldsymbol{x}^*$ and the set of solutions is $\{\boldsymbol{x}^* + \boldsymbol{x}_0 : \boldsymbol{x}_0 \in \mathfrak{N}(A)\}$, where $\mathfrak{N}(A)$ is the null space or kernel of A. Therefore the size of the set of solutions is $|\mathfrak{N}(A)| = q^{v-u}$, which does not depend on $\boldsymbol{y}$. So $\Pr\{A\boldsymbol{k} = \boldsymbol{y}\} = q^{v-u}/q^v = 1/q^u$, and $A\boldsymbol{k}$ is uniformly distributed over $\mathbb{F}_q^u$. This proves the lemma. □

Recall that $Y_i = AX_i + B_iZ_i$ and $\bar{Y}_i = \bar{A}\bar{X}_i + \bar{B}_i\bar{Z}_i$. Define for notational convenience that

$$Z^i = \begin{bmatrix} Z_1 \\ \vdots \\ Z_i \end{bmatrix}, \tag{12}$$

and

$$T^i = \left[\begin{array}{c|cccc} AK_1 & B_1 & 0 & \dots & 0 \\ AK_2 & 0 & B_2 & \dots & 0 \\ \vdots & \vdots & & & \\ AK_i & 0 & 0 & \dots & B_i \end{array}\right] = \left[A^i \mid B^i\right]. \tag{13}$$

Then it follows from the network transform that

$$Y^i = T^i \begin{bmatrix} M \\ Z^i \end{bmatrix}. \tag{14}$$

Lemma 2. *If $i(C - z_e) \geq b$, then T^i has full column rank with probability at least $1 - \frac{1}{q-1}$.*

Proof. Without loss of generality we assume all B_k, $k = 1, ..., i$ have full column ranks z_e, otherwise we can select a basis of the column space of B_k and reformulate the problem with a smaller z_e. Therefore B^i has rank iz_e. By Lemma 1, all the entries in A^i are uniformly i.i.d. distributed. Now consider the first b columns of T^i. For $k = 1, ..., b$, the probability that the k-th column of T^i is in the linear span of the $iz_e + b - k$ columns after it equals to q^{iz_e+b-k}/q^{iC}. Therefore, by the union bound,

$$\begin{aligned}
&\Pr\{T^i \text{ is not full column rank}\} \\
&\quad= \Pr\{\bigcup_{k=1}^{b} k\text{-th column in the span of the columns behind}\} \\
&\quad\leq \sum_{k=1}^{b} \Pr\{k\text{-th column in the span of the columns behind}\} \\
&\quad= \sum_{k=1}^{b} \frac{q^{iz_e+b-k}}{q^{iC}} \\
&\quad= \frac{q^{iz_e+b-iC} - q^{iz_e-iC}}{q-1} \leq \frac{1}{q-1}.
\end{aligned}$$

This completes the proof. □

The following result is well-known.

Lemma 3. *A matrix is left-invertible if and only if it has full column rank.*

Corollary 1. *If $i(C - z_e) \geq b$, then with probability at least $1 - 1/(q-1)$, there exists matrix X^s such that $M = X^sY^i$.*

Proof. By Lemma 2, T^i has full column rank with probability at least $1 - 1/(q-1)$. By Lemma 3, if T^i has full column rank, then there exists a matrix V^s such that $V^sT^i = I_{b+iz_e}$. Therefore by (14),

$$V^sY^i = \begin{bmatrix} M \\ Z^i \end{bmatrix}.$$

Let X^s be the first b rows of V^s and we have $X^sY^i = M$. □

Lemma 4 bounds the probability of the first kind of error.

Lemma 4. *If $i(C - z_e) \geq b$, then with probability at least $1 - (i+1)/(q-1)$, there exist a matrix X^s and vectors $\bar{\boldsymbol{x}}_k^s$, $k = 1, ..., i$ such that $M = X^sY^i$ and $\boldsymbol{x}^s$, $\bar{\boldsymbol{x}}_k^s$, $k = 1, ..., i$ is a solution to (3).*

Proof. By Corollary 1, the probability that there does not exist X^s such that $X^sY^i = M$ is upper bounded by $1/(q-1)$. Because $C > z_e$, by the same argument, for $k = 1, ..., i$ the probability that there does not exist $\bar{\boldsymbol{x}}_k^s$ such that $\boldsymbol{l}_k = \bar{Y}_k^T\bar{\boldsymbol{x}}_k^s$ is upper bounded by $1/(q-1)$. We next verify that the above

$\boldsymbol{x}^s$ and $\bar{\boldsymbol{x}}_k^s$, $k = 1, ..., i$, if exists, is a solution to (3). Vectorize $X^s Y^i = M$ we obtain

$$\boldsymbol{m} = (Y^{i^T} \otimes I_b)\boldsymbol{x}^s, \tag{15}$$

Substitue (15) and $\boldsymbol{l}_k = \bar{Y}_k^T \bar{\boldsymbol{x}}_k^s$ into (3) we obtain

$$P_i \begin{bmatrix} \boldsymbol{x}^s \\ \bar{\boldsymbol{x}}_1^s \\ \vdots \\ \bar{\boldsymbol{x}}_i^s \end{bmatrix} = \begin{bmatrix} D_1 & I_{\alpha_1} & 0 & ... & 0 \\ D_2 & 0 & I_{\alpha_2} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_i & 0 & 0 & ... & I_{\alpha_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{l}_1 \\ \vdots \\ \boldsymbol{l}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_i \end{bmatrix} \tag{16}$$

where the second equality follows from (1). Finally, apply the union bound and the lemma is proved. □

Next we study the second kind of error events.

Lemma 5. *For any $(\boldsymbol{m}^*, \boldsymbol{l}_1^*, ..., \boldsymbol{l}_i^*)$ such that $\boldsymbol{m}^* \neq \boldsymbol{m}$, the probability (over the distribution of the $\boldsymbol{r}_i$'s) that*

$$\begin{bmatrix} D_1 & I_{\alpha_1} & 0 & ... & 0 \\ D_2 & 0 & I_{\alpha_2} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_i & 0 & 0 & ... & I_{\alpha_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{m}^* \\ \boldsymbol{l}_1^* \\ \vdots \\ \boldsymbol{l}_i^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_i \end{bmatrix}, \tag{17}$$

is upper bounded by $(nb/q)^{\sum_{k=1}^i \alpha_k}$

Proof. By (1), the event that $(\boldsymbol{m}^*, \boldsymbol{l}_1^*, ..., \boldsymbol{l}_i^*)$ is a solution of (17) is equivalent to the event that

$$\begin{bmatrix} D_1 & I_{\alpha_1} & 0 & ... & 0 \\ D_2 & 0 & I_{\alpha_2} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_i & 0 & 0 & ... & I_{\alpha_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{m}^* - \boldsymbol{m} \\ \boldsymbol{l}_1^* - \boldsymbol{l}_1 \\ \vdots \\ \boldsymbol{l}_i^* - \boldsymbol{l}_i \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{18}$$

Denote

$$\begin{bmatrix} \boldsymbol{m}^* - \boldsymbol{m} \\ \boldsymbol{l}_1^* - \boldsymbol{l}_1 \\ \vdots \\ \boldsymbol{l}_i^* - \boldsymbol{l}_i \end{bmatrix} = (\Delta m_1, ..., \Delta m_{nb}, \Delta l_1, ..., \Delta l_{\sum_{k=1}^i \alpha_k})^T,$$

and denote by d_{j1} the $(j, 1)$-th entry of the matrix $[D_1^T, ..., D_i^T]^T$. Consider the j-th row of (18), by the construction of the D_i's, this row is equivalent to

$$\sum_{k=1}^{nb} \Delta m_k d_{j1}^k + \Delta l_j = 0. \tag{19}$$

Note that (19) is a non-zero polynomial in variable d_{j1} of degree at most nb. By the fundamental theorem of algebra, this polynomial has at most nb roots. By construction, d_{j1} is uniformly distributed over $\mathbb{F}_q$ and so the probability that (19) is true is upper bounded by nb/q. For $j = 1, ..., \sum_{k=1}^i \alpha_k$, because the d_{j1}'s are i.i.d. distributed, the probability that all $\sum_{k=1}^i \alpha_k$ equations in (18) hold is at most $(nb/q)^{\sum_{k=1}^i \alpha_k}$. □

Lemma 6. *The probability (over the distribution of the $\boldsymbol{r}_i$'s) that there exists a solution $(\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_i^{s*})$ of (3) such that $X^{s*}Y^i \neq M$ is upper bounded by $(nb)^{\sum_{k=1}^i \alpha_k} / q^{\sum_{k=1}^i \alpha_k - iC(b+1)}$.*

Proof. Consider a column vector $(\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_i^{s*})$ such that $X^{s*}Y^i \neq M$. Let $(\boldsymbol{m}^*, \boldsymbol{l}_1^*, ..., \boldsymbol{l}_i^*) = P_{i,2}(\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_i^{s*})$, then $\boldsymbol{m}^* \neq \boldsymbol{m}$. It follows that,

$$\begin{aligned} & \Pr\{\bigcup_{\boldsymbol{x}^{s*}:X^{s*}Y^i \neq M} (\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_k^{s*}) \text{ is a solution of (3)}\} \\ & \overset{(a)}{\leq} \sum_{\boldsymbol{x}^{s*}:X^{s*}Y^i \neq M} \Pr\{(\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_k^{s*}) \text{ is a solution of (3)}\} \\ & \overset{(b)}{\leq} q^{ibC} q^{iC} \left(\frac{nb}{q}\right)^{\sum_{k=1}^i \alpha_k} \\ & = \frac{(nb)^{\sum_{k=1}^i \alpha_k}}{q^{-iC(b+1) + \sum_{k=1}^i \alpha_k}}, \end{aligned}$$

where (a) follows from the union bound. To prove the inequality of (b), notice that if $(\boldsymbol{x}^{s*}, \bar{\boldsymbol{x}}_1^{s*}, ..., \bar{\boldsymbol{x}}_i^{s*})$ is a solution to (3), then $(\boldsymbol{m}^*, \boldsymbol{l}_1^*, ..., \boldsymbol{l}_i^*)$ is a solution to (17). Therefore (b) follows from Lemma 5. This completes the proof. □

We are now ready to prove Theorem 1.

Proof (of Theorem 1). By Lemma 6, for stage $k \leq \mathfrak{i}$, the probability of the second kind of error is upper bounded by $(nb)^{t+k(1+b)\bar{C}} / q^{t+k(b+1)(\bar{C}-C)} \leq (nb)^{t+\mathfrak{i}(1+b)\bar{C}} / q^t$. By Lemma 4, at stage $\mathfrak{i}$ the probability of the first kind of error is upper bounded by $1 - (\mathfrak{i}+1)/(q-1)$. The theorem then follows from the union bound. □

Appendix B
Proof of Theorem 2

Proof. Denote for short that $\epsilon_0 = 1 - (\mathfrak{i}+1)/(q-1) - \mathfrak{i}(nb)^{t+\mathfrak{i}(b+1)\bar{C}} / q^t$. Assume for contradiction that the probability of decoding failure at stage $\mathfrak{i}$ is larger than $\epsilon_0 + \epsilon$. We construct a statistical test $E(x)$ that simulates the communication process. Specifically, the test simulates the the encoder and the decoder to execute the scheme by setting x as the shared secret randomness between the source and the terminal. The test simulates all intermediate nodes in the network as well as the attacks by the adversary. The test outputs 0 if the decoder successfully recover the message at stage $\mathfrak{i}$, and outputs 1 otherwise. If x is uniformly i.i.d. distributed, then by Theorem 1, $\Pr\{E(x) = 1\}$ is at most ϵ_0. If x is generated by $PSG(e)$ where e is uniformly distributed over $\mathcal{S}$, then by hypothesis $\Pr\{E(PSG(e)) = 1\} > \epsilon_0 + \epsilon$. Hence $\Pr_{e \in \mathcal{S}}\{E(PSG(e)) = 1\} - \Pr_{x \in \{0,1\}^N}\{E(x) = 1\} > \epsilon$, a contradiction. □

Appendix C
Proof of Proposition 1 and Theorem 3

Proof (of Proposition 1). Let $\Delta \boldsymbol{l} = \boldsymbol{l}_i - \boldsymbol{l}_i^*$, then $\Delta l_k = \sum_{j=1}^{nb} \hat{d}_{kj}(m_j^* - m_j)$. The polynomial in variables $\hat{d}_{kj}$ on the R.H.S. is non-zero because $\boldsymbol{m} - \boldsymbol{m}^* \neq \boldsymbol{0}$. Therefore by by the Schwartz-Zippel Lemma $\Pr\{\Delta l_k = 0\} \leq 1/q$. So $\Pr\{\Delta \boldsymbol{l} = 0\} = \Pr\{\bigcap_{k=1}^{\alpha_i} \Delta l_k = 0\} = \prod_{k=1}^{\alpha_i} \Pr\{\Delta l_k = 0\} \leq 1/q^{\alpha_i}$. □

Proof (of Theorem 3*).* Replacing Lemma 5 by Proposition 1 in the proof of Theorem 1, and then the theorem is proved. □

APPENDIX D
PROOF OF THEOREM 5

Proof. By (11) and the construction of J and K_i, it follows that

$$JHR_iK_i = JHX_iK_i + JHU_iK_i. \tag{20}$$

By (9), it follows that,

$$X_i = \begin{bmatrix} \phi(S_i) - P\phi(N_i) \\ \phi(N_i) \end{bmatrix}.$$

Recall that $H = [I\ P]$, therefore

$$HX_i = [I\ P] \begin{bmatrix} \phi(S_i) - P\phi(N_i) \\ \phi(N_i) \end{bmatrix} = \phi(S_i),$$

and so

$$JHR_iK_i = J\phi(S_i)K_i + JHU_iK_i. \tag{21}$$

Suppose $s_i = 0$, then $\phi(S_i) = 0$ and $JHR_iK_i = JHU_iK_i$. Recall that rank$(U_i) \leq z_e - \max\{\mu + C - \bar{C}, \delta\}$, and because by hypothesis $\bar{C} \geq C > z_w + z_e$, it follows that

$$\text{rank}(U_i) \leq z_e - \mu - C + \bar{C} \leq z_e - \mu < \bar{C} - z_w - \mu,$$

and rank$(U_i) \leq z_e - \delta < \bar{C} - z_w - \delta$. Because the size of JHR_iK_i is $(\bar{C} - z_w - \mu) \times (\bar{C} - z_w - \delta)$, it is not full rank. This proves the first statement of the theorem.

Suppose $s_i = 1$, then the entries of $\phi(S_i)$ by construction are uniformly distributed over $\mathbb{F}_Q$. Because J has full row rank and K_i has full column rank, by Lemma 1, the entries of $J\phi(S_i)K_i$ are uniformly distributed over $\mathbb{F}_Q$. By Theorem 4, the injected errors are independent from S_i and so the error term U_i is independent from X_i. This implies that JHU_iK_i is independent from $J\phi(S_i)K_i$. Hence JHR_iK_i is a random matrix with entries i.i.d. uniformly distributed over $\mathbb{F}_Q$. Then by an argument similar to the proof of Lemma 2, it can be shown that probability that JHR_iK_i is rank-deficient is at most $1/(Q-1)$. This proves the second statement of the theorem. □