Network Codes Resilient to Jamming and Eavesdropping

Hongyi Yao, Danilo Silva, Sidharth Jaggi and Michael Langberg

Abstract—**We consider the problem of communicating information over a network secretly and reliably in the presence of a hidden adversary who can eavesdrop and inject malicious errors. We provide polynomial-time distributed network codes that are information-theoretically rate-optimal for this scenario, improving on the rates achievable in prior work by Ngai *et al*. Our main contribution shows that as long as the sum of the number of links the adversary can jam (denoted by Z_O) and the number of links he can eavesdrop on (denoted by Z_I) is less than the network capacity (denoted by C), (*i.e.,* $Z_O + Z_I < C$), our codes can communicate (with vanishingly small error probability) a single bit correctly and without leaking any information to the adversary. We then use this scheme as a module to design codes that allow communication at the source rate of $C - Z_O - Z_I$, while keeping the communicated message provably secret from the adversary. Indeed, if the secrecy requirements are relaxed, rates up to $C - Z_O$ are achievable. Interior nodes are oblivious to the presence of adversaries and perform random linear network coding; only the source and destination need to be tweaked. We also prove that the rate-region obtained is information-theoretically optimal. In proving our results we correct an error in prior work by a subset of the authors in this work.**

I. Introduction

A source Alice wishes to multicast information to a set of receivers over a network. As shown in an elegant sequence of papers, *network coding*, *i.e.,* allowing internal nodes in a network to perform non-trivial arithmetic operations on incoming information to generate their outgoing information in general strictly increases the achievable rate region. An information theoretic proof of this was provided in [1], and further, the work of [2] demonstrated that linear codes sufficed to achieve such performance. The techniques in [3] demonstrated an explicit procedure to design such codes over finite fields. Efficient code constructions were provided in [4] (deterministic codes) and in [5] (distributed randomized codes). An extensive account of the theory and applications of network coding can be found in [6].

However, if a network with even a single receiver Bob contains a malicious adversary Calvin, there are at least two security challenges – Calvin might eavesdrop on private communications, or he might disrupt communications by injecting fake information into the network, or in general he might do both. In the network coding model this second danger may be even more pronounced since all nodes, including honest ones, mix information. In this case, even a small number of fake packets injected by Calvin may end up corrupting *all* the information flowing in the network, causing decoding errors. In particular, Calvin may use his knowledge of the network topology so as to sit in the bottleneck of the network, and thereby jam communications in a network location where it might do the most damage. Also, Calvin's eavesdropping capabilities have negative security implications in two ways. Firstly, Calvin is able to eavesdrop and thus infer something about Alice's secret message to Bob. Second, Calvin might also be able to use the eavesdropped information to carefully design his jamming pattern so as to make it hard for Bob to correctly decode Alice's message.

In this work we consider the *secrecy* and *error control* issues together. Namely, we design schemes that allow reliable network communications in the presence of an adversary that can both jam and eavesdrop, without leaking information to him. In particular, suppose the network's min-cut from Alice to Bob is C, and Calvin eavesdrops on Z_I links and corrupts Z_O links[1]. We demonstrate schemes that are distributed, computationally efficient to design and implement, and can be used to communicate a *single* bit secretly and without error. We then use this scheme as a tool to improve on prior work [7], and achieve a provably optimal rate of $C - Z_O - Z_I$. In fact, if the secrecy constraint is relaxed, any rate up to $C - Z_O$ can be achieved. In particular, the overall rate of communication thus achievable in the presence of Calvin's *adversarial* jamming is the same as if the jamming behaved like *random noise*.

A preliminary version of the results in this paper was presented in [8].

A. Prior work

Related problems have been considered in the past. Prior results may be classified in the following three categories. An extensive discussion on the field of security problems for networks performing network coding can be found in [6, Chapter 7].

1) *Secrecy:* For networks containing adversaries that only eavesdrop on some links (without jamming transmissions), the work of [9] provided a tight information-theoretic characterization of the *secrecy capacity*, *i.e.*,

The work of D. Silva was partly supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 482131/2010-1.

Hongyi Yao is with the California Institute of Technology.

Danilo Silva is with the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, SC 88040-900, Brazil (e-mail: danilo@eel.ufsc.br).

Sidharth Jaggi is with The Chinese University of Hong Kong.

Michael Langberg is with The Open University of Israel.

[1] We consider a model where network links rather than nodes are eavesdropped and corrupted; eavesdropping on a node is equivalent to eavesdropping on links incoming to it, and corrupting a node is equivalent to corrupting the links outgoing from it.

the optimal rate achievable without leaking any of Alice's information to Calvin. Efficient schemes achieving this performance were proposed by [10]–[12]. Cryptographically (but not information-theoretically) secret schemes for this scenario were also considered in [13].

2) *Error-control:* For networks containing adversaries with unlimited eavesdropping capabilities and limited jamming capabilities, prior related work has focused primarily on the detection of Byzantine errors (*i.e.*, a bounded number of arbitrary – rather than random – errors) [14], non-constructive bounds on the achievable *zero-error* rates [15], [16], and network error-correcting codes [17] (which have high design complexity) and [7], [18]–[20] (which have low design complexity). Results for this setting are also available under cryptographic assumptions [21], [22].

3) *Secrecy + Error-control:* The scenario closest to the one considered in this work, with limitations on both Calvin's eavesdropping power Z_I and his jamming power Z_O, have been considered in [7], [20], [23]–[25]. Here there are two questions one could consider:

   a) What is the maximum rate at which one can communicate reliably (without caring about hiding one's data from Calvin)?

   b) What is the maximum rate at which one could communicate *both* secretly and reliably?

   For Model 3b, under the requirement of *zero* error probability, the maximum rate of secret and reliable communication is given by $C - 2Z_O - Z_I$. Schemes achieving this rate have been proposed in [24], [25] (high design complexity schemes) and [23], [26], [27] (low design complexity schemes). The optimality of such a rate has been shown in [25] for single-letter coding and in [27] for block coding.

   For Model 3a, if the requirement of zero error probability is relaxed to *vanishingly small* error probability, as considered here, then higher rates than $C - 2Z_O - Z_I$ may be achieved. In particular, the work in [7] provided computationally efficient communication schemes at rate $C - Z_O$ as long as the restrictive requirement $C > 2Z_O + Z_I$ was satisfied. Work by a subset of the authors of this paper claimed in [20] to improve this requirement to $C > Z_O + Z_I$. As we demonstrate in Section X the prior proof of the claim was incorrect, and Section II gives a correct proof of the claim.

   Combining these results with the secrecy/"data-hiding" scheme of [12] allows us to obtain the rates (that we prove to be optimal) of $C - Z_O - Z_I$ in Model 3b, and $C - Z_O$ in Model 3a.

To put our results in perspective (in particular to compare with the most relevant prior work [7]), consider the following two scenarios that are specific examples of our general result:

1) Suppose $Z_O = Z_I$ (a "realistic" scenario, corresponding to the adversary being able to eavesdrop on links it can jam) and denote these quantities both by Z. Prior work [7], for either Model 3a or Model 3b, would apply only in the regime where Z is less than a third of the min-cut C. In contrast, our current work demonstrates that communication is possible (in either 3a or 3b) as long as Z is less than half the min-cut C.

2) An even more extreme example is as follows – suppose $Z_I = 0$ (also a "realistic" scenario, corresponding to a "blind" adversary – one who cannot base his jamming function on what is being actually transmitted. In a wireless setting, this may happen perhaps because he has only one antenna). In this scenario, our schemes achieve a positive rate for *any* $Z_O < C$, whereas prior work ([7]) would be restricted to the setting wherein Z_O is less than half the min-cut C.

II. Main Results

The main results of this work are Theorems 1 and 2 below. Let q be the size of the finite field over which the network code operates, and let n be the block-length (number of symbols over $\mathbb{F}_q$) of the packets transmitted over the network.

Theorem 1: If $C > Z_O + Z_I$ then Alice can communicate a single bit correctly to Bob (while keeping it secret from Calvin) using codes of computational complexity $O(\text{poly}(C, \log_2 q))$ and error probability $O(q^{-C})$.

Combining the codes in Theorem 1 with the "shared-secret" codes in [7] then gives us Theorem 2. Roughly, we say that a message at a certain rate is " secretly and robustly achievable" if there exists a communication scheme that keeps Alice's message information-theoretically secure from Calvin, and simultaneously allows Bob to decode Alice's message (with high probability). A more formal definition follows in Section III.

Theorem 2: No rate higher than $C - Z_O - Z_I$ is secretly and robustly achievable. A rate of $C - Z_O - Z_I$ is secretly and robustly achievable with codes of computational complexity $O(n.\text{poly}(C, \log_2 q))$.

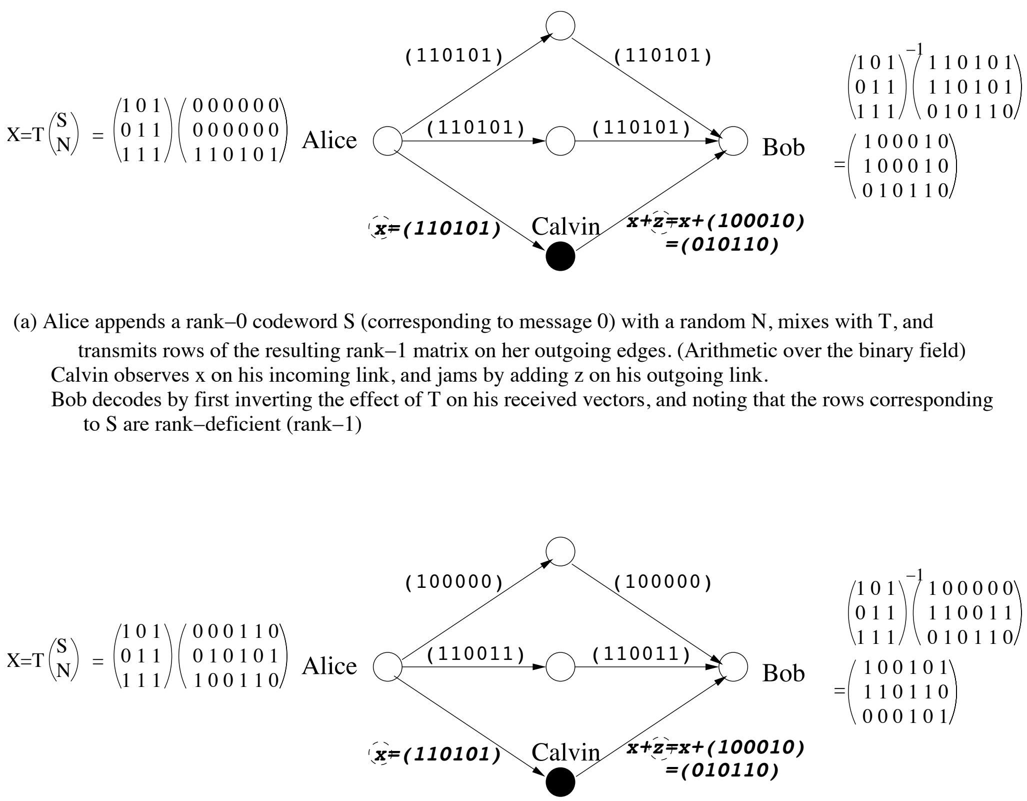
We also say that a message at a certain rate is " robustly achievable" if, roughly, there exists a communication scheme that allows Bob to decode Alice's message (with high probability), but with no guarantees on Calvin's (in)ability to reconstruct Alice's message. A more formal definition follows in Section III. As a direct corollary of our proof techniques in Theorems 1 and 2, we also show

Theorem 3: No rate higher than $C - Z_O$ is robustly achievable. If $C > Z_O + Z_I$, then a rate of $C - Z_O$ is robustly achievable with codes of computational complexity $O(n.\text{poly}(C, \log_2 q))$.

Note: In [25], Ngai *et al.* show that $C - 2Z_O - Z_I$ is an upper bound on the rate, assuming no error events, and single-letter coding (respectively equations (87) and (65) in their proof). Our work achieves higher rates by instead assuming asymptotically negligible probability of error, and block coding.

A. Toy Example

We first begin with a toy example demonstrating the main ideas behind our central result, *viz.* the modular scheme that



(a) Alice appends a rank–0 codeword S (corresponding to message 0) with a random N, mixes with T, and transmits rows of the resulting rank–1 matrix on her outgoing edges. (Arithmetic over the binary field)
Calvin observes x on his incoming link, and jams by adding z on his outgoing link.
Bob decodes by first inverting the effect of T on his received vectors, and noting that the rows corresponding to S are rank–deficient (rank–1)

(b) Alice appends a rank–2 codeword S (corresponding to message 1) with a random N, mixes with T, and transmits rows of the resulting rank–3 matrix on her outgoing edges. (Arithmetic over the binary field)
Calvin observes x on his incoming link, and jams by adding z on his outgoing link.
Bob decodes by first inverting the effect of T on his received vectors, and noting that the rows corresponding to S are full–rank (rank–2)

Fig. 1. **Toy example demonstrating how to share a single bit secretly and robustly**: This example corresponds to the "Secret-sharing layer" referenced later in Figure 4, and thence in Section VII. In this example, $C = 3$, and $Z_O = Z_I = 1$ (hence $C > Z_O + Z_I$). The block-length n'' used in this example equals 6. Alice just want to share a single bit with Bob secretly and reliably – if the bit equals 0, she uses the scheme in (a), else she uses the scheme in (b). Bob decodes by checking the rank of the received matrix. Note that we assume that Alice and Bob know the *value* of Z_O and Z_I in advance, though not Calvin's location – this is analogous to having a prior estimate of channel conditions.

Alice and Bob use to communicate a single bit correctly, and without leaking information to Calvin. In our example network, Alice communicates with Bob over a network that comprises of three parallel paths passing through distinct relays. Honest relays simply forward incoming data. One of the relays is controlled by the adversary Calvin, who therefore knows the transmission on the incoming link, and can corrupt the corresponding outgoing transmission arbitrarily. The identity of the node controlled by Calvin is unknown to Alice and Bob. In a slightly generalized version of this example, Calvin may "control" $Z = Z_I = Z_O$ relays (*i.e.*, observes the packets incoming to Z relays and so $Z_I = Z$, and based on these observations corrupts the packets outgoing from these relays and so $Z_O = Z$), out of a total of C relays.

The scheme demonstrated in Figure 1 is as follows. Let the block-length $n'' > C$ be a design parameter chosen by Alice and Bob so as to guarantee performance.

If Alice wishes to transmit a 0 to Bob, she transmits a "random low-rank codeword" over the parallel links. That is, she appends a $(C - Z_I) \times n''$ zero-matrix S to a random (hence full-rank with high probability) $Z_I \times n''$ matrix N. She "mixes" the rows of the resulting matrix by pre-multiplying it with an invertible matrix T, and transmits the resulting rows over her outgoing links (in Figure 1(a) this corresponds to sending a single vector repeatedly over the different links).

Conversely, if Alice wishes to transmit a 1 to Bob, she transmits a "random high-rank codeword" over the parallel links. That is, she chooses S to be a random (hence full-rank with high probability) $(C - Z_I) \times n$ matrix and appends a random $Z_I \times n''$ matrix N. She again mixes this with T and transmits the resulting row-vectors over her outgoing links (in Figure 1(b) this corresponds to sending three linearly independent vectors over the different links).[2]

Bob's decoding rule is straightforward. He first pre-multiplies the $C \times n''$ matrix with the inverse of T, in an

[2] Calvin knows the matrix T, as does Bob – it is part of code design. For this reason, it is important to use an MDS code's parity-check matrix, so that on observing any Z_I rows/packets, Calvin cannot distinguish between a zero bit-matrix and a high-rank bit-matrix – for instance, if the identity matrix was used, Calvin could "easily" distinguish between the two cases by simply observing the appropriate packet.

attempt to recover S and N. If the resulting row vectors corresponding to the locations of S are "rank-deficient" (of rank strictly less than $C - Z_I$) he decodes to a 0, otherwise he decodes to a 1.

To show that this scheme does not enable Calvin to estimate Alice's message bit, note that from his perspective the distribution over the messages he eavesdrops is identical regardless of Alice's message bit – in both cases, his observed packet is uniformly distributed over all length-n'' vectors. .

To show that the scheme *also* enables Bob to decode Alice's 0 or 1 message correctly, with high probability, *regardless* of Calvin's adversarial jamming action, we proceed as follows. Note that Calvin has no information on what vectors are being transmitted on links not controlled by him. Hence, even though he can transmit arbitrary vectors on the links controlled by him, the probability that these vectors are linearly dependent on the other vectors (on links *not* controlled by him) is quite small (exponentially small in the block-length, and the block-length can be chosen to be large, to guarantee probability of success arbitrarily close to 1). [3]

Refining the ideas presented in this toy example to the general scenario requires several non-trivial extensions. Details of these extensions are in the following sections, but we summarize these here.

Firstly, Alice and Bob need to design a *distributed* scheme that operates even when they are ignorant of network topology prior to communication. This requires that Alice's message bit remains secret from Calvin even if he receives random linear combinations of Alice's transmissions (rather than the specific vectors she injects into the network). It also requires that Calvin's injected jamming vectors be linearly independent of other randomly linearly combined vectors. Since the linear transforms applied by the network need not preserve the uniform probability distribution that Alice imposes on her transmitted vectors, a more delicate analysis is needed.

Secondly, Bob does not in general know the linear transform imposed by the network. To circumvent this problem, the "subspace metric" codes introduced by Kötter *et al.* [19] prove quite useful.

Lastly, we note that the ideas above can really only be used to transmit a "few" bits from Alice to Bob. This is because each use of the scheme requires Alice to send a somewhat bulky matrix simply to communicate a single bit to Bob, and if the scheme is repeated too many times, then the throughput of Alice's message goes down considerably. Fortunately, a "shared-secret" algorithm presented in [7] enables us to guarantee high-rate secret communication from Alice to Bob, as long as Alice can share even just a "few" bits secretly with Bob. We thus use our single-bit sharing scheme as a module for the shared-secret algorithm to obtain the desired result.

III. Network Model and Problem Statement

We use the general model proposed in [7], and pictorially represented in Figure 2 above. To simplify notation we consider only the problem of communicating from a single source to a single destination[4].

A. Network Model

Alice communicates to Bob over a network with an attacker (adversary) Calvin hidden somewhere in it. Calvin aims to disrupt the transfer of information from Alice to Bob and in the meantime to eavesdrop on the information Alice sends. He can observe some of the transmissions, and can inject his own fake transmissions.

Calvin is computationally unbounded, knows the encoding and decoding schemes of Alice and Bob, and the network code implemented by the interior nodes. He also knows the network topology, and he gets to choose which network links to eavesdrop on and which ones to corrupt.

The network is modeled as a directed and delay-free graph whose edges each have capacity equal to one symbol of a finite field of size q, $\mathbb{F}_q$, per unit time[5]. All computations are over $\mathbb{F}_q$. The *network capacity*, denoted by C, is the *min-cut from source to destination*[6].

Each packet contains n symbols from $\mathbb{F}_q$. Alice's message is denoted $W \in \mathcal{S}$. To send this to Bob over the network, Alice encodes W into a matrix $X \in \mathbb{F}_q^{C \times n}$, possibly using a *stochastic encoder*[7]. The i^{th} row in X is Alice's i^{th} packet. As in [5], Alice and internal nodes take random linear combinations of their observed packets to generate their transmitted packets.

Analogously to how Alice generates X, Bob organizes received packets into a matrix Y. The i^{th} received packet corresponds to the i^{th} row of Y. The random linear network code used by Alice and all internal nodes induces a linear transform A from X to Y, such that $Y = AX$ when no error is induced by the adversary[8]. Thus Y is a matrix in $\mathbb{F}_q^{C \times n}$, and $A \in \mathbb{F}_q^{C \times C}$. Hereafter we assume that the matrix A is invertible, which happens with high probability if q is sufficiently large [5].

Calvin can eavesdrop on Z_I edges, and can inject (possibly fake) information at Z_O locations[9], in the network. The matrix received by Bob is then $Y = AX + Z$, where Z corresponds to the information injected by Calvin as seen by Bob. Note that the limitation of Calvin's jamming capacity implies that

[3] Note the following asymmetry – when Alice sends bit 0, Bob never makes an error; he makes an error (with small probability) if and only if bit 1 is sent and the received matrix is not full-rank. The reason for this asymmetry is as follows – if Alice's secret bit is 0, then the rank of the transmitted message is Z_I, and hence the maximum rank of the received message is $Z_I + Z_O < C$. But in this case, by Bob's decoding rule, he (correctly) outputs a 0. On the other hand, if the secret bit is 1, it is possible (though "unlikely") that the packet injected by Calvin is able to lower the rank of the matrix Bob uses to decode.

[4] Similarly to many network coding algorithms, our techniques generalize to multicast problems.

[5] For ease of presentation edges with non-unit capacities are not considered here (as in [7], they may be modeled via block coding and parallel edges).

[6] For the corresponding multicast case, C is defined as the minimum of the min-cuts over all destinations. It is well-known that C also equals the time-average of the maximum number of packets that can be delivered from Alice to Bob, assuming no adversarial interference, *i.e.*, the *max flow*.

[7] The random coin tosses made by Alice as part of her encoding scheme are not known to either Calvin or Bob.

[8] For the ease of notation we assume Bob removes redundant incoming edges so that the number of edges reaching Bob equals the min-cut capacity C from Alice to Bob.

[9] We assume throughout that the information injected into the network by Calvin is *added* to the original information transmitted (here we consider addition over our field $\mathbb{F}_q$).

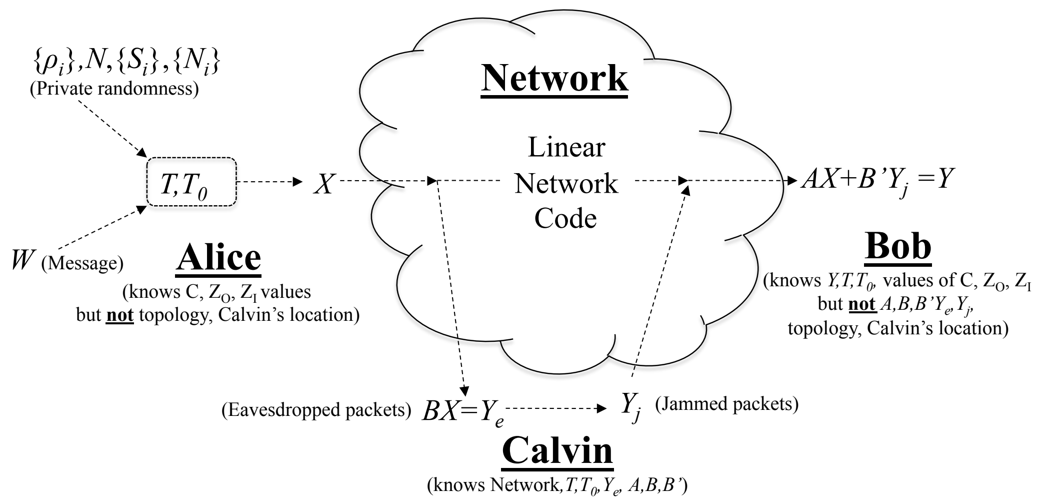

$\{\rho_i\},N,\{S_i\},\{N_i\}$
(Private randomness)
T,T_0
X
W (Message)
Alice
(knows C, Z_O, Z_I values
but not topology, Calvin's location)
Network
Linear
Network
Code
$AX+B'Y_j=Y$
Bob
(knows Y,T,T_0, values of C, Z_O, Z_I
but not $A,B,B'Y_e,Y_j$,
topology, Calvin's location)
(Eavesdropped packets) $BX=Y_e$
Y_j (Jammed packets)
Calvin
(knows Network,T,T_0,Y_e, A,B,B')

Fig. 2. **System diagram**: A pictorial representation of the system model described in Section III. Alice uses private randomness (known only to her) and "mixing" matrices T, T_0 (known to the other parties, Bob and Calvin) to encode her message W (of rate $R = C - Z_I - Z_O$) to X. (This encoding procedure is further detailed in Figures 3 and 4, and in detail in Sections VII and VIII). This X is transmitted over the "Network channel". Calvin overhears packets Y_e, and based on these, attempts to reconstruct W, and also to "jam" transmission to Bob by injecting jamming packets Y_j. In addition to the packets Y_e he overhears, Calvin knows Alice's mixing matrices T, T_0, the network topology, and the network transforms A, B, and B' (respectively from Alice to Bob, Alice to Calvin, and Calvin to Bob) – Calvin's choice of Y_j may be a function of all of these.

$\mathsf{rank}(Z) \leq Z_O$ – in particular, Z can be thought of as $B'Y_j$, where Y_j correspond to the (at most Z_O) packets injected by Calvin, and $B \in \mathbb{F}_q^{C \times Z_O}$ as the linear transform imposed by the network between Calvin and Bob. Similarly, Calvin's observation can be described as a matrix $Y_e = BX$, where $B \in \mathbb{F}_q^{Z_I \times C}$ is the linear transform undertaken by X as seen by Calvin. Both B and B' are known to Calvin *a priori*, but not to Alice or Bob (since Calvin is hidden). Neither Alice nor Bob are assumed to know the network transform, or indeed even the network topology prior to the commencement of communication, though Calvin is allowed to know these. However, Alice and Bob are both assumed to know the *values* of Z_O, Z_I, and C (or good upper bounds on these), since these are critical for them to decide the rates at which to transmit. This is analogous to the encoder/decoder pair needing to "estimate channel conditions" before deciding what rate/code to use.

B. Problem Statement

Alice wishes to communicate with Bob with perfect secrecy and vanishingly small error probability. That is, Alice's scheme is *perfectly secret* if

$$I(\mathbf{W}; \mathbf{Y}_e) = 0 \quad \forall B \in \mathbb{F}_q^{Z_I \times C} \tag{1}$$

i.e., Calvin obtains no information about Alice's message regardless of which Z_I links he eavesdrops. The *error probability* is the probability (over all randomness introduced by Alice and Calvin) that Bob's reconstruction $\hat{W}$ of Alice's information W is inaccurate, *i.e.*, $Pr[\hat{W} \neq W]$. We consider the error probability of the worst-case scenario[10]. Namely, a scheme has error probability less than ϵ if $Pr[\hat{W} \neq W] < \epsilon \quad \forall A, Z$, where A is assumed to be nonsingular, and $\mathsf{rank}(Z) \leq Z_O$. The *rate* R of a scheme is the number of information bits of information Alice transmits to Bob, amortized by the size of a packet in bits, *i.e.*, $R = \frac{1}{n} \log_q |\mathcal{S}|$. The rate R is said to be *secretly and robustly achievable* if for any $\epsilon > 0$, any $\delta > 0$, and sufficiently large n, there exists a perfectly secret block-length-n network code with rate at least $R - \delta$ and a probability of error less than ϵ. The rate R is said to be *robustly achievable* if for any $\epsilon > 0$, any $\delta > 0$, and sufficiently large n, there exists a block-length-n network code (which need not be perfectly secret) with rate at least $R - \delta$ and a probability of error less than ϵ.

IV. High-level overview of proofs and techniques

We first show in Section V that $C - Z_O - Z_I$ is an upper bound on the rate at which a secret message can be correctly transmitted from Alice to Bob, by demonstrating an attack that Calvin can use to successfully disrupt communication if Alice tries to communicate at any higher rate. We then construct efficient codes that essentially achieve rate $C - Z_O - Z_I$. Our codes consist of the data-hiding, error-control, and secret-sharing layers described below. All the three layers are carefully "mixed" and embedded along with Alice's message into her packets and then transmitted through the network using network coding. Hence, though the description of Alice's encoding scheme is somewhat intricate, the network itself is oblivious to it, and can simply do packet-by-packet linear network coding, for instance via the standard codes in [5].

Secret-sharing layer: In Section VII we first prove Theorem 1 by showing how to communicate a single bit secretly and correctly over a network containing adversaries that can jam and eavesdrop, as long as $C > Z_I + Z_O$. This layer is important for the error-control layer described later, and can be implemented via a "small" header appended to each network coded packet. When k secret bits are to be shared, the scheme

[10] Our interest is to design communication schemes that do not rely on the specific network topology or network code used.

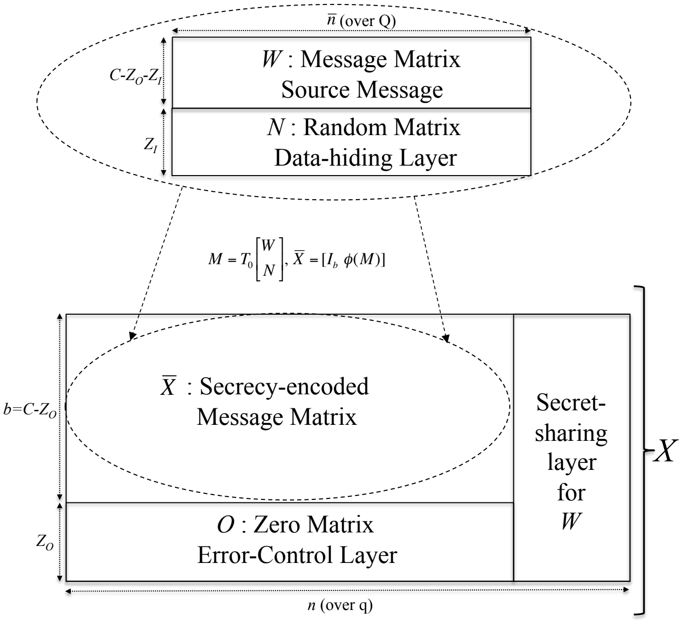


Fig. 3. **Overall encoding**: A pictorial representation of the encoding (described in detail in Section VIII). Alice generates her transmission X as follows. She first mixes the $C - Z_I - Z_O$ packets of her message (written as matrix W) with Z_I random packets (rows of the random matrix N) via the invertible matrix T_0 to obtain her secrecy-encoded/masked message $\bar{X}$ (for technical reasons, she also switches from the field $\mathbb{F}_Q$ to the field $\mathbb{F}_q$ via the isomorphism $\phi(.)$). This matrix $\bar{X}$ has the property that if Calvin observes any Z_I rows of it, or indeed any Z_I linear combinations of its rows, no information about W is leaked to him. To further protect her transmissions from the Z_O jamming packets Calvin may inject into the network, Alice adds redundancy by appending Z_O zero rows. Finally, she appends a "small secret-sharing layer" header, as described in Figure 4.

is repeated k times in each transmitted packet header, for a secret-sharing header of total length $C+kC(C-Z_I)$. Figure 4 shows the components of the secret-sharing layer. The secret-sharing layer consisting of the following components:

1. *Identity matrix:* As standard in random linear network coding [5], [18], the identity matrix I_C is appended to convey to the receiver information about the linear transform induced by the random linear network code.

2. *Bit matrices:* For each secret bit s_i, $i \in \{1, \ldots, k\}$, Alice chooses a corresponding S_i independently and uniformly at random from the set of all $(C - Z_I) \times C(C - Z_I)$ matrices (over $\mathbb{F}_q$). She then sets the ith bit-matrix as s_iS_i, *i.e.*, If the ith secret bit equals 0, the matrix s_iS_i is a zero matrix; otherwise s_iS_i is a uniformly random full-rank matrix. We refer to s_iS_i as a *bit matrix*.

The idea is that the rank of the matrices corresponding to bit 0 is much smaller than the rank of the matrices corresponding to bit 1—due to the limitation on the numbers of packets Calvin can observe or inject, with high probability he cannot change the rank of the corresponding received matrix by too much. Details are given in Lemma 4.

3. *Random matrix:* Alice adapts the scheme of [12] to keep the bit matrices secret from Calvin. That is, for each secret bit i that Alice wishes to communicate to Bob, she combines the bit matrix s_iS_i with a random noise matrix N_i (at rate Z_I). It can be shown that it is impossible for Calvin to glean any useful information (since it can only eavesdrop at rate Z_I).

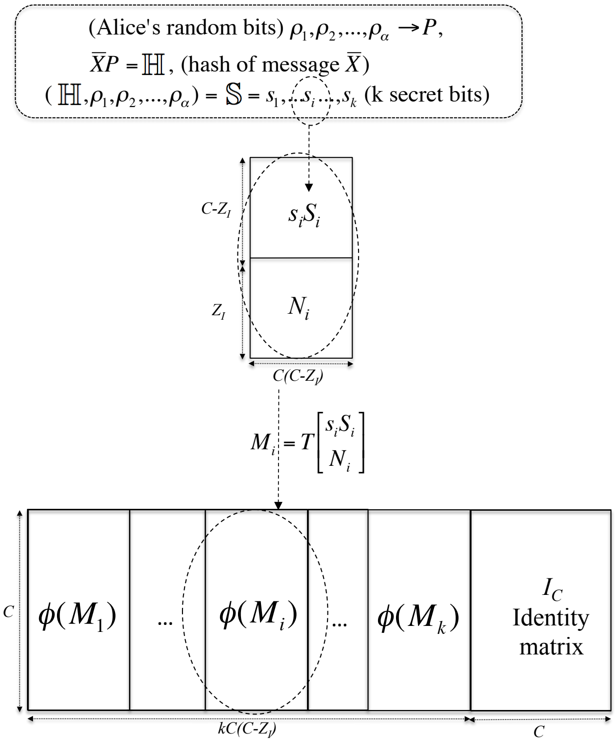

(Alice's random bits) $\rho_1,\rho_2,...,\rho_\alpha \to P$,
$\bar{X}P = \mathbb{H}$, (hash of message $\bar{X}$)
$(\mathbb{H},\rho_1,\rho_2,...,\rho_\alpha) = \mathbb{S} = s_1,...,s_i,...,s_k$ (k secret bits)
C-Z_I
s_iS_i
Z_I
N_i
$C(C$-$Z_I)$
$M_i = T\begin{bmatrix} s_iS_i \\ N_i \end{bmatrix}$
C
$\phi(M_1)$
...
$\phi(M_i)$
...
$\phi(M_k)$
I_C
Identity
matrix
$kC(C$-$Z_I)$
C

Fig. 4. **Secret-sharing layer**: (Described in detail in Section VII.) Alice first generates a secret hash of her message W as follows. She chooses α symbols $\rho_1, \ldots, \rho_\alpha$ uniformly at random from $\mathbb{F}_q$, uses these symbols to generate a random parity-check matrix P, and uses P to generate a hash $\mathbb{H}$ of her secrecy-encoded/masked message $\bar{X}$ (generated as described in Figure 3). The bits of $\mathbb{H}$ and $\rho_1, \ldots, \rho_\alpha$ together comprise her k secret bits $\{s_1, \ldots, s_k\} = \mathbb{S}$. She then uses these bits to encode bit-matrices s_iS_i – if s_i is a 0-bit then s_iS_i is a zero-matrix, else it is a random full-rank matrix. As in the secrecy-encoding in Figure 3, each of these bit-matrices is mixed with a random matrix N_i. The resulting mixed matrices (translated to $\mathbb{F}_q$ via $\phi(.)$), along with the "standard" identity matrix header used in, for instance [5], comprise the secret-sharing layer.

Overall encoding: Section VIII combines the secret-sharing layer with the two other layers described below to complete our code construction. Figure 3 shows the overall encoding procedure Alice follows.

1. *Data hiding layer:* As done with the random matrices N_i in the secret-sharing layer above, a random matrix N is used to preserve the secrecy of the source message W (of rate $C - Z_O - Z_I$), yielding an encoded matrix M (of rate $C - Z_O$).

2. *Error control layer:* In this layer Alice uses the "shared-secret" scheme outlined in Theorem 1 of [7]. That is, Alice first takes a secret linear hash to her secrecy-encoded message M to generate a small hash value. Both the linear hash and the resulting hash value (say k bits in all) are transmitted to Bob using the secret-sharing layer. Alice then combines her data with a zero-value matrix (of rate Z_O), such that Bob can use the secret hash to *distill* Alice's codeword M from the corrupted information reaching the destination.

Vis-a-vis our secret-sharing scheme of Section VII, the work of [20] (by a subset of the authors of this work) claimed to have the same result. However, we show in Section X that the scheme proposed in [20] is incorrect by giving an attack that Calvin can use to ensure that Bob has a significant probability of decoding error.

TABLE I
SUMMARY OF COMMONLY USED NOTATION/PARAMETERS

Notation	Meaning
Network notation/parameters	
C	Min-cut of the network
Z_I	Eavesdropping rate
Z_O	Jamming rate
C'	$C - Z_I$
b	$C - Z_O$
A	Network transform from Alice to Bob
B	Network transform from Alice to Calvin
B'	Network transform from Calvin to Bob
Alice's encoder	
W	Alice's "payload" message
q	Network code field-size
n	Packet length (over $\mathbb{F}_q$)
$Q = q^C$	Extension field size
n'	Message packet-length (over $\mathbb{F}_Q$)
$\mathbb{S} = \{s_1, \ldots, s_k\}$	Alice's "small" secret message
T_0, T	Alice's "Mixing" matrices
α	$bC + 1$
k	$\alpha(b+1)\log_2 q$
ϕ	Mapping from $\mathbb{F}_Q$ to $(\mathbb{F}_q)^C$
n''	Packet length (over $\mathbb{F}_q$) for single-bit scheme

V. CONVERSE FOR THEOREM 2

We start by presenting an attack that Calvin may use to force the achievable rate to at most $C - Z_O - Z_I$, thereby demonstrating that this is indeed an upper bound on the achievable rate. Let $\{e_1, e_2, ..., e_C\}$ be a set of edges that form a cut from Alice to Bob. Calvin jams the edges in $\{e_1, e_2, ..., e_{Z_O}\}$ by adding random errors on them. Further, Calvin eavesdrops on edges in $\{e_{Z_O+1}, e_{Z_O+2}, ..., e_{Z_O+Z_I}\}$. Let $\mathbf{W}$ be the random variable denoting Alice's information. Let $\mathbf{Y}_j$, $\mathbf{Y}_e$, and $\mathbf{Y}_u$ be the random variables denoting the packets carried by the jammed edges $\{e_1, e_2, ..., e_{Z_O}\}$, eavesdropped edges $\{e_{Z_O+1}, e_{Z_O+2}, ..., e_{Z_O+Z_I}\}$, and untouched edges $\{e_{Z_O+Z_I+1}, e_{Z_O+Z_I+2}, ..., e_C\}$ respectively. Let $\mathbf{Y}$ be the random variable denoting the packets received by Bob. Then

$$
\begin{aligned}
nR &= H(\mathbf{W}) = H(\mathbf{W}|\mathbf{Y}) + I(\mathbf{W};\mathbf{Y}) && (2)\\
&\leq 1 + \epsilon nR + I(\mathbf{W};\mathbf{Y}) && (3)\\
&\leq 1 + \epsilon nR + I(\mathbf{W};\mathbf{Y}_j,\mathbf{Y}_e,\mathbf{Y}_u) && (4)\\
&\leq 1 + \epsilon nR + I(\mathbf{W};\mathbf{Y}_e,\mathbf{Y}_u) && (5)\\
&= 1 + \epsilon nR + I(\mathbf{W};\mathbf{Y}_e) + I(\mathbf{W};\mathbf{Y}_u|\mathbf{Y}_e) && (6)\\
&= 1 + \epsilon nR + I(\mathbf{W};\mathbf{Y}_u|\mathbf{Y}_e) && (7)\\
&\leq 1 + \epsilon nR + H(\mathbf{Y}_u) && (8)\\
&\leq n\left[(C - Z_I - Z_O) + \epsilon R + \frac{1}{n}\right]. && (9)
\end{aligned}
$$

Here ϵ refers to the probability of error. Equation (2) follows from the fact that Alice's message is uniformly distributed over $\mathbf{W}$, (3) from Fano's inequality, (4) from the data processing inequality, (5) since in the worst case Calvin adds random noise on the edges he jams and so $\mathbf{Y}_j$ is independent of $(\mathbf{W}, \mathbf{Y}_e, \mathbf{Y}_u)$, (6) by the chain rule for mutual information, (7) from the fact that information-theoretic secrecy is required and so $I(\mathbf{W};\mathbf{Y}_e) = 0$, (8) by the fact that conditioning reduces entropy and the definition of mutual information, and finally (9) by the fact that there are at most $C - Z_I - Z_O$ links corresponding to the random variable $\mathbf{Y}_u$ and the alphabet-size upper bound on entropy. Requiring $\epsilon \to 0$ as $n \to \infty$ gives the required result.

VI. AUXILIARY TOOLS

A. Mapping between finite fields

We first define a mapping commonly used in the network error-correcting code literature (for example, see [12]) that maps between a vector space over a finite field, and a corresponding extension field of the minute field. This mapping helps us translate between the field over which the internal nodes in the network perform network coding ($\mathbb{F}_q$) and the field over which the end-to-end codes operate ($\mathbb{F}_Q$).

Let $\mathbb{F}_Q = \mathbb{F}_q^C$ and let $\mathbb{F}_Q$ be an extension field of $\mathbb{F}_q$. Let $\phi : \mathbb{F}_Q \to \mathbb{F}_q^{1\times C}$ be a vector space isomorphism. In addition, let $\phi_{m,n} : \mathbb{F}_Q^{m\times n} \to \mathbb{F}_q^{m\times Cn}$ be a vector space isomorphism such that the ith row of $\phi_{m,n}(X)$ is given by $\begin{bmatrix}\phi(X_{i,1}) & \cdots & \phi(X_{i,n})\end{bmatrix}$. In other words, we expand each element of $X \in \mathbb{F}_Q^{m\times n}$ as a length-C row vector over $\mathbb{F}_q$ (with the number of columns in matrix increasing accordingly). We will omit the subscript from $\phi_{m,n}$ when the dimensions of the argument are clear from the context.

Throughout, without loss of generality, we assume that n is divisible by C.

We now consider two models of communication, both different from the one we actually consider, that nonetheless are useful modules for our overall construction.

B. Secrecy/Data-hiding Coding

Consider a special case of the problem where the mincut (denoted b) equals $C - Z_O$ (rather than C), Calvin can eavesdrop on $Z_I < b$ packets, but *cannot* jam any packets. Below, we review a construction of a perfectly secret scheme that asymptotically achieves the maximum possible rate (*i.e.*, the secrecy capacity) $b - Z_I$, for this problem. The scheme, proposed in [12], is based on MRD (*Maximum Rank Distance*) codes. (For more details on MRD codes, see [12].)

Let $H \in \mathbb{F}_Q^{(b-Z_I)\times b}$ be the parity-check matrix of a $[b, Z_I]$ linear MRD code over $\mathbb{F}_Q$. Let $T_0 \in \mathbb{F}_Q^{b\times b}$ be an invertible matrix chosen such that the first $b - Z_I$ rows of ${T_0}^{-1}$ are equal to H.

Alice's encoding proceeds as follows. She first generates a random matrix $N \in \mathbb{F}_Q^{Z_I\times\bar{n}}$ uniformly and independently from any other variables. Then, she computes $\bar{X} = \begin{bmatrix}I_b & \phi(M)\end{bmatrix}$, where $M = T_0 \begin{bmatrix} W \\ N \end{bmatrix}$. Alice thus encodes a given message $W \in \mathbb{F}_Q^{(b-Z_I)\times\bar{n}}$, where $\bar{n} = n/C - 1$.

Assuming Bob receives $Y = AX = \begin{bmatrix} A & A\phi(M) \end{bmatrix}$, Bob computes $X = A^{-1}Y$ to recover $M = \phi^{-1}(\phi(M))$. Then, Bob can directly obtain W since, by construction, $W = HM$.

Recall that Calvin's observation is given by $Y_e = BX$, where $B \in \mathbb{F}_q^{Z_I \times b}$. According to Theorem 4 of [12], we have that $I(\mathbf{W}; \mathbf{Y}_e) = 0$ for all $\bar{B}$, and therefore (1) is satisfied. Thus, the scheme is indeed perfectly secret.

The decoding complexity is given by $O(\bar{n}b^2)$ operations over $\mathbb{F}_Q$, which can be done in $O(nC^4)$ operations over $\mathbb{F}_q$.

C. Error Control under a Shared Secret Model

Consider now a second scenario, wherein Calvin can jam $Z_O < C$ packets and eavesdrop *any* number of packets he chooses. However, we posit the existence of a "low-rate side-channel", which Calvin cannot access, that enables Alice to transmit to Bob a "small" secret $\mathbb{S}$ (of size asymptotically negligible compared to Alice's message). We also drop the requirement of secret communication, *i.e.*, all we require is that Bob can decode Alice's transmission correctly, with high probability over the transmissions on the side channel. Below, we review a coding scheme presented in [7] that can asymptotically achieve the maximum possible rate (the so-called *shared-secret capacity*) $C - Z_O$ for this case.

Recall that we use b to denote $C - Z_O$. We first describe how Alice produces the secret bit string $\mathbb{S}$ based on a given message $\phi(M) \in \mathbb{F}_q^{b\times(n-b)}$. To begin with, she generates $\alpha = bC + 1$ symbols $\rho_1, \rho_2, ..., \rho_\alpha \in \mathbb{F}_q$ independently and uniformly at random. Let $P \in \mathbb{F}_q^{n\times\alpha}$ be the matrix given by $P_{(i,j)} = (\rho_j)^i$. Then, she computes a matrix $\mathbb{H} = \bar{X}P \in \mathbb{F}_q^{b\times\alpha}$, where $\bar{X} = \begin{bmatrix} I_b & \phi(M) \end{bmatrix}$. The tuple $(\rho_1, \rho_2, ..., \rho_\alpha, \mathbb{H})$, consisting in total of $\alpha(b+1)$ symbols in $\mathbb{F}_q$, comprises the message "hash" that should be secretly transmitted to Bob. The bit representation of this tuple yields the string $\mathbb{S} \in \{0,1\}^k$, consisting of $k = \alpha(b+1)\log_2 q$ bits. Over the main channel, Alice transmits the $C \times n$ matrix $X = \begin{bmatrix} \bar{X} \\ 0 \end{bmatrix} = \begin{bmatrix} I_b & \phi(M) \\ 0 & 0 \end{bmatrix}$.

Assuming that $(\rho_1, \rho_2, \ldots, \rho_\alpha, \mathbb{H})$ is secretly and correctly received by Bob, let us proceed to the description of Bob's decoder. First, Bob reconstructs the matrix P. Bob obtains $Y = AX + Z$, where $Z \in \mathbb{F}_q^{C\times n}$ has rank at most Z_O. This can also be written as $Y = \tilde{A}\bar{X} + Z$, where $\tilde{A}$ consists of the first b columns of A. Let $\bar{Y}$ be the reduced row echelon form of Y. It is shown in [7] that, with probability at least $1 - O(1/q)$ for any fixed network, $\bar{X}$ can be written as $\bar{X} = U\bar{Y}$ for some $U \in \mathbb{F}_q^{b\times C}$. It is also shown in [7] that, with probability at least $1 - n^\alpha/q$, the system $U\bar{Y}P = \mathbb{H}$ has a unique solution in U. Bob solves this system to find U, computes $\bar{X} = U\bar{Y}$ and finally recovers $\phi(M)$.

Overall, the probability of error of the scheme is at most $n^\alpha/q + O(1/q) = O(n^{C^2}/q)$, while the decoding complexity is $O(nC^3)$ operations in $\mathbb{F}_q$.

VII. Sending a Single Bit Secretly and Reliably

We now give a detailed description of the secret-sharing layer of Figure 4. Let $C' = C - Z_I$. In this section, we show how Alice can transmit a secret bit s_i reliably to Bob when $C > Z_I + Z_O$. We assume that the block-length n'' for this single-bit scheme is $C(1 + C')$, as this is the smallest packet-length required for the scheme to work. Larger block-lengths can be handled by zero-padding the transmitted packets.

Let $T \in \mathbb{F}_Q^{C\times C}$ and $H \in \mathbb{F}_Q^{C'\times C}$ be as given in Section VI-B.

A. Alice's encoder

Initially, Alice chooses a matrix S_i uniformly at random from full-rank $\mathbb{F}_Q^{C'\times C'}$. If her secret bit s_i is 1, s_iS_i is non-zero; otherwise, if s_i is 0, $s_iS_i = 0$. Then, she sends s_iS_i to Bob using the data-hiding scheme described in Section VI-B. More precisely, she transmits $X = \begin{bmatrix} I_C & \phi(M_i) \end{bmatrix}$, where $M_i = T \begin{bmatrix} S_i \\ N_i \end{bmatrix}$ and $N_i \in \mathbb{F}_Q^{Z_I\times C'}$ is a uniformly random matrix chosen independently from S_i.

B. Bob's decoder

Recall that Bob receives a matrix $Y = AX + Z$, where $A \in \mathbb{F}_q^{C\times C}$ is nonsingular and $Z \in \mathbb{F}_q^{C\times C(1+C')}$ has rank at most Z_O. Let $\bar{Y}$ denote the reduced row echelon form of Y. Consider first the case where $\bar{Y} = \begin{bmatrix} I & \phi(r) \end{bmatrix}$, for some $r \in \mathbb{F}_Q^{C\times C'}$. It is possible to show that $Hr = S + E$, where $E \in \mathbb{F}_Q^{C'\times C'}$ is a matrix of rank at most Z_O. As will be shown later, with high probability, Hr is full-rank if and only if Alice's secret bit is 1. Thus, Bob can decode by computing the rank of Hr.

In general, however, $\bar{Y}$ may not have the form described above. Nevertheless, as shown in [18, Proposition 10] and [23, Chapter 5], it is possible to extract from $\bar{Y}$ some matrices $r \in \mathbb{F}_Q^{C\times C'}$, $\hat{L} \in \mathbb{F}_q^{C\times\mu}$ and $\hat{V} \in \mathbb{F}_Q^{\nu\times C'}$ such that

$$r = x + \hat{L}V^1 + L^2\hat{V} + L^3V^3 \tag{10}$$

for some $V^1 \in \mathbb{F}_Q^{\mu\times C'}$, $L^2 \in \mathbb{F}_q^{C\times\nu}$, $L^3 \in \mathbb{F}_q^{C\times f}$ and $V^3 \in \mathbb{F}_Q^{f\times C'}$. The matrices r, $\hat{L}$ and $\hat{V}$ can be obtained by converting $\bar{Y}$ to reduced row echelon form (see [23, Section 5.1.2]) and therefore are *known* to the decoder. The last three terms in (10) may be seen as generalized errors terms, as some of is factors ($\hat{L}$ and $\hat{V}$) are known. Note that a partially known error term is analogous to an erasure in classical coding theory (where the location of the error, but not its value, is known) and has the same effect of enabling the decoder to correct more errors than if such variables were unknown.

Additionally, it is shown in [23, Theorem 5.4] that μ, ν and f (the inner dimensions of the three outer products in (10)) satisfy $\mu, \nu \leq Z_O$ and $f \leq Z_O - \max\{\mu, \nu\}$. Since $Z_O < C'$, it follows that

$$f < C' - \max\{\mu, \nu\}.$$

In possession of r, $\hat{L}$ and $\hat{V}$, Bob is now ready to decode the data-hiding layer that has been applied to x.

We have

$$\begin{aligned} Hr &= Hx + H\hat{L}V^1 + HL^2\hat{V} + HL^3V^3 \\ &= S + \hat{\Lambda}V^1 + \Lambda^2\hat{V} + \Lambda^3V^3 \end{aligned} \tag{11}$$

where $\hat{\Lambda} = H\hat{L}$, $\Lambda^2 = HL^2$ and $\Lambda^3 = HL^3$. Note that $\hat{\Lambda} \in \mathbb{F}_Q^{C' \times \mu}$ and $\hat{V} \in \mathbb{F}_Q^{\nu \times C'}$ are known.

Now, let $J \in \mathbb{F}_Q^{(C'-\mu) \times C'}$ and $K \in \mathbb{F}_Q^{C' \times (C'-\nu)}$ be full-rank matrices such that $J\hat{\Lambda} = 0$ and $\hat{V}K = 0$. Then Bob can further simplify (11) by computing

$$JHrK = JSK + J\Lambda^3 V^3 K.$$

Note that $\mathsf{rank}(J\Lambda^3 V^3 K) \leq f < C' - \max\{\mu, \nu\}$, while $C' - \max\{\mu, \nu\}$ is the maximum possible rank of $JHrK$.

Thus, Bob performs the following test. If $JHrK$ is full-rank, then Bob concludes that bit $s_i = 1$ was sent; otherwise, Bob concludes that bit $s_i = 0$ was sent.

With respect to complexity, computing $\bar{Y}$ takes $O(C^2 n) = O(C^4)$ operations in $\mathbb{F}_q$. Computing J, K, $JHrK$ and the rank of $JHrK$ each take $O(C^3)$ operations in $\mathbb{F}_Q$, which amounts to $O(C^5)$ in $\mathbb{F}_q$. Thus, the overall decoding complexity is $O(C^5)$ operations in $\mathbb{F}_q$.

C. *Probability of error analysis*

When bit 0 is sent, Bob never makes an error; he makes an error if and only if bit 1 is sent and $JHrK$ is not full-rank. Recall that, when bit 1 is sent, S is uniformly distributed over $\mathbb{F}_Q^{C' \times C'}$. Due to the data-hiding encoding, Calvin has no information about S, and therefore S is statistically independent from $\Lambda^3 V^3$. It follows that $S' = S + \Lambda^3 V^3$ is also uniformly distributed over $\mathbb{F}_Q^{C' \times C'}$. Thus, the probability of error when bit 1 is sent is equal to the probability that $JS'K \in \mathbb{F}_Q^{(C'-\mu) \times (C'-\nu)}$ is not full-rank for a uniform S'.

Lemma 4: If $S' \in \mathbb{F}_Q^{C' \times C'}$ is uniformly distributed then, for any $J \in \mathbb{F}_Q^{(C'-\mu) \times C'}$ and any $K \in \mathbb{F}_Q^{C' \times (C'-\nu)}$, the matrix $JS'K$ is full-rank with probability at least $1 - C'/Q$.

Proof: Without loss of generality, assume $\mu \geq \nu$. It suffices to prove the statement for $\mu = \nu$; if $\mu > \nu$, then removing $\mu - \nu$ columns from K cannot possibly increase the rank of $JS'K$.

For any fixed J and K, consider the entries of S' as variables taking values in $\mathbb{F}_Q$. Then each entry of $JS'K$ is a multivariate polynomial over $\mathbb{F}_Q$ with degree at most 1. It follows that $\det(JS'K)$ is a multivariate polynomial over $\mathbb{F}_Q$ with degree at most $C' - \mu \leq C'$. Note that, if $Q \leq C'$, the statement follows trivially, so assume $Q > C'$. From [5, Lemma 4], we have that $P[\det(JS'K) = 0] \leq C'/Q$. ■

Thus, the probability of error of the scheme is upper bounded by $C'/Q \leq C/q^C$, which can be made arbitrarily small by choosing q sufficiently large. This proves Theorem 1.

VIII. Achievability for Theorem 2

We now describe a coding scheme that achieves rate $R = C - Z_I - Z_O$ asymptotically in the packet length n.

As before, assume that n is divisible by C and let $n' = n/C - (1 + kC')$, where $k = (bC + 1)(b + 1)\log_2 q$.

Let $H \in \mathbb{F}_Q^{C' \times C}$ be the parity-check matrix of a $[C, Z_I]$ linear MRD code over $\mathbb{F}_Q$. Let $T \in \mathbb{F}_Q^{C \times C}$ be an invertible matrix such that the first $C - Z_I$ rows of T^{-1} are equal to H.

Similarly, let $H_0 \in \mathbb{F}_Q^{R \times b}$ be the parity-check matrix of a $[b, Z_I]$ linear MRD code over $\mathbb{F}_Q$, and let $T_0 \in \mathbb{F}_Q^{b \times b}$ be an invertible matrix such that the first R rows of T_0^{-1} are equal to H_0.

A. *Alice's encoder*

First, given a message $S \in \mathbb{F}_Q^{R \times n'}$, Alice computes $M = T_0 \begin{bmatrix} S \\ N \end{bmatrix}$, where $N \in \mathbb{F}_Q^{Z_I \times n'}$ is chosen independently and uniformly at random. Then, she generates a string $\mathbb{S} \in \{0,1\}^k$ of k bits according to the scheme described in Section VI-C. Next, for each ith bit s_i of $\mathbb{S}$, Alice produces a matrix $s_i S_i \in \mathbb{F}_Q^{C' \times C'}$ according to the scheme described in Section VII. Then, for each $i = 1, \ldots, k$, she computes $M_i = T \begin{bmatrix} S_i \\ N_i \end{bmatrix}$, where each $N_i \in \mathbb{F}_Q^{Z_I \times C'}$ is chosen uniformly at random and independently from any other variables. Finally, she produces a transmission matrix

$$X = \begin{bmatrix} I_C & \phi(M_1) & \phi(M_2) & \cdots & \phi(M_k) & \begin{bmatrix} \phi(M) \\ 0 \end{bmatrix} \end{bmatrix}.$$

B. *Bob's decoder*

For each $i = 1, \ldots, k$, Bob extracts a submatrix Y_i from Y corresponding to the submatrix $\begin{bmatrix} I_C & \phi(x_i) \end{bmatrix}$ from X (*i.e.*, columns $1, \ldots, C, C + (i-1)C' + 1, \ldots, C + iC'$). He then applies on Y_i the decoder described in Section VII to obtain each $s_i \in \mathbb{S}$.

Similarly, Bob extracts a submatrix Y_0 consisting of the first b and the last $n'C$ rows of Y. Note that $Y_0 = AX_0 + Z_0$, where $X_0 = \begin{bmatrix} I_b & \phi(M) \\ 0 & 0 \end{bmatrix} \in \mathbb{F}_q^{C \times (b + n'C)}$ and Z^0 has rank at most Z_O. Then, Bob applies the decoder described in Section VI-C to obtain M.

Finally, Bob computes $x = \phi^{-1}(M)$ and $S = H_0 x$.

C. *Overall Analysis*

1) Secrecy analysis: The secrecy of the message is guaranteed by the scheme of Section VI-B.

2) Error probability analysis: By the union bound, the probability that Bob makes an error when decoding the k-bit secret $\mathbb{S}$ is at most $kC/q^C \leq C^4(\log_2 q)/q^C = O(\frac{\log_2 q}{q^C})$. Given that the secret is decoded correctly, the probability that Bob makes an error when decoding the message is at most $O(n^{C^2}/q)$. Thus, the overall probability of error is at most $O(n^{C^2}/q)$.

3) Rate analysis: The rate of the scheme is given by $Rn'C/n = R(1 - (1 + kC')C/n) \leq R - RC^5(\log_2 q)/n$. Thus, the rate loss is $O(\frac{\log_2 q}{n})$.

4) Complexity analysis: Decoding all the secret bits takes $O(kC^5) = O(C^8 \log_2 q)$ operations in $\mathbb{F}_q$, while the computational complexity of decoding the message is dominated by the secrecy/data-hiding decoding steps with $O(C^4 n)$ operations over $\mathbb{F}_q$.

Note: Both the rate loss and the error probability can be made asymptotically small by choosing q to grow faster than polynomially but slower than exponentially in n. For instance, we may choose $q = 2^{\lfloor \sqrt{n} \rfloor}$.

IX. Proof Sketch of Theorem 3

The robust-achievability of rate $C - Z_O$ as in Theorem 3 follows directly from the codes used to secretly and robustly achieve rate $C - Z_O - Z_I$ in Theorem 2. In particular, one can use *both* W and N to encode the source's message at rate $C - Z_O$, if one does not require perfectly secret communication schemes (as defined in Section III). As part of the decoding process in Theorem 2, Bob is already guaranteed to be able to recover both W and N.

As to the proof of optimality of the rate, it follows directly from observing that if Calvin may jam Z_O links in the min-cut of the network, no more than $C - Z_O$ rate can be robustly achievable.

X. Errata for [20]

We briefly reprise the scheme of [20] before demonstrating the flaw in the proof. In what follows, all operations are over $\mathbb{F}_q$.

In the scheme of [20] there exist two hash matrices D_0 and D_1 which are chosen independently and uniformly at random $C^2(C - Z_O) \times C^2$ *Vandermonde matrices*, *i.e.*, each column of D_0 and D_1 is of the form $\mathbf{h}(u) = [u, u^2, ..., u^{C^2(C-Z_O)}]^T$, where the generator u is chosen independently and uniformly at random from $\mathbb{F}_q$. Both D_0 and D_1 are publicly known to all parties, including Bob and Calvin.

Alice's Encoder: Alice first chooses a random length-$(C^2(C - Z_O) - C^2)$ row vector $\mathbf{u}$. Let $I \in \{0, 1\}$ be the secret bit that Alice wishes to send to Bob. Alice then constructs the length-1 × C^2 row vector $\mathbf{r}$ such that $[\mathbf{u}, \mathbf{r}]D_I = 0$. Note that such $\mathbf{r}$ exists since the last C^2 rows of D_I form an invertible matrix. Finally the vector $[\mathbf{u}, \mathbf{r}]$ is rearranged into a $(C - Z_O) \times C^2$ matrix which is sent through the network via random linear network coding.

Bob's Decoder: After receiving the $C \times C^2$ matrix Y, for each $I \in \{0, 1\}$ Bob check whether there exists $C - Z_O$ length-C vectors $\{\mathbf{x}_i, i \in [1, C - Z_O]\}$ such that $[\mathbf{x}_1 Y, \mathbf{x}_2 Y, ..., \mathbf{x}_{C-Z_O} Y]D_I = 0$. If so, Bob decodes the secret bit as I. The idea is that if I is Alice's bit, such $\{\mathbf{x}_i, i \in [1, C - Z_O]\}$ exists for D_I with high probability [7].

Calvin's successful attack: When Calvin corrupts $Z_O \geq C - Z_O$ edges, Calvin could mimic Alice's behaviour when she wishes to transmit a particular bit, say 1. As a result Bob would always find length-C row vectors $\{\mathbf{x}_i, i \in [1, C - Z_O]\}$ such that $[\mathbf{x}_1 Y, \mathbf{x}_2 Y, ..., \mathbf{x}_{C-Z_O} Y]D_1 = 0$. In this case Bob cannot determine whether the bit 1 is from Alice or from Calvin.

Even if Calvin can only inject $Z_O < C - Z_O$ errors, if $Z_O + Z_I \geq C - Z_O$, there is another successful attack for Calvin. To see that, without loss of generality let $Z_O + Z_I = C - Z_O$. Since Calvin can eavesdrop on Z_I packets $\{\mathbf{y}_i, i \in [1, Z_I]\}$, he can carefully choose his Z_O injected error packets $\{\mathbf{z}_i, i \in [1, Z_O]\}$ so that $[\mathbf{y}_1, ..., \mathbf{y}_{Z_I}, \mathbf{z}_1, ..., \mathbf{z}_{Z_O}]D_1 = 0$. In this case, Bob also always decodes its bit as 1. Thus the scheme in [20] only works for the case where $C > 2Z_O + Z_I$, which does not improve the result in [7].

Why our scheme works: In our scheme in Section VII, instead of distinguishing the bit by the hash matrices, Alice hides her secret in the rank of the bit matrix she transmits. In particular, there is a rank gap $C - Z_I$ between the bit matrix for bit 0 and the one for bit 1. Thus as long as $C - Z_I > Z_O$, Calvin cannot mimic Alice any more, since he can only inject Z_O errors. As a result Bob can determine Alice's bit by examining the rank of the matrix he decodes.

XI. Conclusion

In this work we considered the problem of communicating information secretly and reliably over a network containing a malicious eavesdropping and jamming adversary. Under the assumptions that vanishingly small probabilities of error and block coding are allowed, we substantially improve on the best achievable rates in prior work [25], and also prove the optimality of our achievable rates. A key component of our code design is a scheme that allows a small amount of information to be transmitted secretly and reliably over the network, as long as the total number of packets that the adversary can either eavesdrop on or jam is less than the communication capacity of the network. In proving this scheme we correct an error in the proof of prior work [20] by a subset of the authors of this work.

References

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

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[5] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[6] M. Médard and A. Sprintson, *Network Coding: Fundamentals and Applications*. Elsevier, 2011.

[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[8] H. Yao, D. Silva, S. Jaggi, and M. Langberg, "Network codes resilient to jamming and eavesdropping," in *Proceedings of the IEEE International Symposium on Network Coding*, Toronto, Canada, Jun. 2010.

[9] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun. 30–Jul. 5, 2002, p. 323.

[10] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, "On the capacity of secure network coding," in *Proc. 42nd Annual Allerton Conf. on Commun., Control, and Computing*, Sep. 2004.

[11] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks II," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 24–29, 2007, pp. 551–555.

[12] D. Silva and F. R. Kschischang, "Security for wiretap networks via rank-metric codes," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, Jul. 6–11, 2008, pp. 176–180.

[13] P. F. Oliveira and J. Barros, "A network coding approach to secret key distribution," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 414–423, 2008.

[14] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2798–2803, 2008.

[15] R. W. Yeung and N. Cai, "Network error correction, part i: Basic concepts and upper bounds," *Commun. Inf. Syst*, vol. 6, no. 1, pp. 19–36, 2006.

[16] N. Cai and R. W. Yeung, "Network error correction, part ii: Lower bounds," *Commun. Inf. Syst*, vol. 6, no. 1, pp. 37–54, 2006.

[17] R. Matsumoto, "Construction algorithm for network error -correcting codes attaining the singleton bound," *arXiv:cs.IT/0610121*, Oct 2006.

[18] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[19] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[20] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping Byzantine adversaries," in *Proc. IEEE Int. Symp. Information Theory*, 24–29 June 2007, pp. 541–545.

[21] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proc. of The 27th Conference on Computer Communications*, 2008.

[22] F. Zhao, T. Kalker, M. Medard, and J. K. Han, "Signatures for content distribution with network coding," in *Proc. of ISIT*, 2007.

[23] D. Silva, "Error control for network coding," Ph.D. dissertation, University of Toronto, Toronto, Canada, 2009.

[24] C.-K. Ngai and S. Yang, "Deterministic secure error-correcting (sec) network codes," in *Proc. IEEE Information Theory Workshop*, Tahoe City, CA, Sep. 2–6, 2007, pp. 96–101.

[25] C.-K. Ngai and R. W. Yeung, "Secure error-correcting (sec) network codes," in *Proc. Workshop on Network Coding Theory and Applications*, Lausanne, Switzerland, Jun. 15-16, 2009, pp. 98–103.

[26] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1124–1135, 2011.

[27] ——, "Universal secure error control schemes for network coding," in *Proc. IEEE Int. Symp. Information Theory*, 2010.