Resilient network codes in the presence of eavesdropping Byzantine adversaries

Sidharth Jaggi
Department of Information Engineering
Chinese University of Hong Kong
Shatin, NT, Hong Kong
jaggi@ie.cuhk.edu.hk

Michael Langberg
Computer Science Division
The Open University of Israel
Raanana, 43107, Israel
mikel@openu.ac.il

***Abstract*— Network coding can substantially improve network throughput and performance. However, these codes have a major drawback if the network contains hidden malicious nodes that can eavesdrop on transmissions and inject fake information. In this scenario, even a small amount of information injected by a single malicious hidden node could mix with and contaminate much of the information inside the network, causing a decoding error.**

We improve on previous work by providing a polynomial-time, rate-optimal distributed network code design that functions even in the presence of a Byzantine adversary with substantial eavesdropping capabilities. As long as the sum of the adversary's jamming rate Z_O and his eavesdropping rate Z_I is less than the network capacity C, ($Z_O + Z_I < C$), our codes attain the optimal rate of $C - Z_O$.

The network codes we design are information-theoretically secure and assume no knowledge of network topology. Prior to transmission, no honest node knows the location or strength of the adversary. In our code design, interior nodes are oblivious to the presence of adversaries and implement a classical low-complexity distributed network code design; only the source and destination need to be changed. Finally, our codes work for both wired and wireless networks.

I. Introduction

Network coding, i.e., the mixing of information at internal nodes in a network, was proposed in [1]. Since then, a growing body of literature has demonstrated the considerable advantages of such codes in both wired and wireless networks, such as throughput gains in multicast networks [1], robustness to failures [2], [3] and power efficiency [4].

However, this mixing of information can be catastrophic for networks containing *Byzantine nodes*, i.e., malicious internal nodes that pretend to be routers but instead eavesdrop on transmissions and inject fake packets with the objective of disrupting communications. In this case, even a small amount of corrupted information may be mixed with all the information flowing in the network, causing decoding errors.

Communication via network coding in the presence of Byzantine errors has been addressed in the past. Prior related work has focused primarily on the detection of Byzantine errors [5], non-constructive bounds on the achievable rates [6], [7], and network error-correcting codes [8]. All but the latter address Byzantine adversaries that have unlimited *eavesdropping* capabilities, and analyze the quality of communication as a function of their *jamming* capabilities. Namely, the adversaries studied are assumed to observe all transmissions in the network, but are limited in the amount of errors they inject into the network. In this work we study Byzantine adversaries which are limited in both their eavesdropping and jamming capabilities. Such adversaries were also studied in a previous work of ours [8] (joint with Michelle Effros, Tracey Ho, Dina Katabi, Sachin Katti, and Muriel Médard). The current work substantially improves on one of the main results of [8].

The main results in [8] provide polynomial-time distributed algorithms for two cases. First, a network code is proposed to combat an *omniscient* adversary (one that can observe all transmissions in the network). This code attains the optimal rate of $C - 2Z_O$, where C is the network multicast capacity and Z_O is the rate of the adversary's information generation. Second, a network code is proposed to combat a *limited* adversary (one who can eavesdrop on only a limited amount of information in the network). This code attains the substantially higher rate $C - Z_O$, which can also be shown to be optimal. However, for technical reasons this algorithm requires the fairly restrictive condition that the adversary's eavesdropping rate Z_I is less than $C - 2Z_O$. Thus the presence of some privacy in the network can be leveraged to obtain higher network throughput.

Our results: We improve on this previous work by providing a polynomial-time, rate-optimal distributed network code design that functions even in the presence of a Byzantine adversary with substantial eavesdropping capabilities. As long as the sum of the adversary's jamming rate Z_O and his eavesdropping rate Z_I is less than the network capacity C, ($Z_O + Z_I < C$), our codes attain the optimal rate of $C - Z_O$.

Our code-design algorithm relaxes the restriction on the limited adversary's eavesdropping rate. Namely, our network codes can still achieve the optimal rate $R = C - Z_O$ even if Z_I is almost as large as $C - Z_O$. In other words, as long as $Z_I < R$ our codes' performance is optimal. The difference from the case considered in [8] is substantial. For example the algorithm in [8] is able to achieve a positive rate only for a Byzantine adversary with Z_O at most $C/2$, whereas the algorithm in this paper can function even if Z_O is almost C. The main result of this work can be summarized by the following theorem. Let $\mathbb{F}_q$ denote the field over which all

1-4244-1429-6/07/$25.00 ©2007 IEEE

linear operations in our network code are performed. Let n denote the block length of the packets transmitted over the links of the network (*i.e.* each packet consists of n elements from $\mathbb{F}_q$).

Theorem 1: If $Z_I < C - Z_O$, the "Limited Adversary" algorithm described in Section III achieves a rate of $C - Z_O - \mathcal{O}\left(\frac{C^4 \log q}{n}\right)$ with code-complexity $\mathcal{O}(n\text{poly}(C, \log q))$ and probability of error at most $\mathcal{O}\left(\frac{n^C C + \text{poly}(C) \log q}{q}\right)$.

Setting $q = 2^{\sqrt{n}}$ is a reasonable trade off between rate overhead, probability of error and code complexity. For comparison, in the error free scheme of [2], to obtain an exponential small error probability, the field size q must also be exponential.

Proof techniques: We leverage the following idea in our algorithm. Since more information is being generated by the source than is being intercepted by the adversary ($R > Z_I$), some of the source's information is information-theoretically secret from the adversary. Further, it is known ([8], Theorem 1) that even if an asymptotically negligible amount of information can be transmitted correctly from the source to the destination secretly from the adversary, this secret can be used to *purify* all the rest of the corrupted information reaching the destination. The key challenge arises in the transmission of even this small message secretly and correctly over a network containing Byzantine adversaries. To this end, the source introduces a small amount of redundancy in its information by imposing a few carefully chosen linear constraints on it. By examining its received information the destination is able to determine which linear constraints the source's data satisfies, and the identity of these constraints encodes the secret message that the source wishes to transmit to the destination. However, the adversary is unable to determine anything about these linear constraints since Z_I is asymptotically bounded away from R.

The majority of this work is devoted to presenting an elegant polynomial-time distributed secret sharing scheme that lends itself to classical network code design. As mentioned above once such a scheme is established, we use the results of [8] to obtain Theorem 1.

Secret sharing in the presence of a Byzantine adversary such as ours has been extensively studied in the standard point-to-point communication model (for example see [9]). Our setting does not fall under this standard framework for many reasons. Primarily, several difficulties arise when considering secret sharing under the distributed network coding model. In addition, we note that the errors imposed by the Byzantine adversary in the network coding setting are assumed to be *added* to the original information transmitted on the network (the addition is with respect to the underlying field $\mathbb{F}_q$ used in the linear network code studied). The model in which these errors can *overwrite* the existing information transmitted is an interesting research direction we are currently perusing. We note that there is a difference between these two error models only when the adversary is not aware of the information to be jammed due to eavesdropping limitations (such as the plausible scenario where a wireless node cannot simultaneously transmit and receive).

The design of secret sharing schemes for network coding in the presence of an eavesdropping adversary with no jamming capabilities (*i.e.* $Z_O = 0$) has been addressed in [10], [11]. In some aspects our proof techniques resemble those presented in [11] – however our setting (which includes a jamming adversary) is more general and code design is more demanding.

The focus of this paper is primarily on the effect of the adversary's eavesdropping power. The algorithm that is our main contribution achieves rate-optimal performance in network conditions that are more adverse than previous work. It does so in a polynomial-time, distributed manner, without knowledge of network topology, location or strength of the adversary. Our network codes work even if the adversary is computationally unbounded, and enable the source to share a message with the destination that is secret from the adversary. They work for wired and wireless networks. Finally, internal nodes in our network code are oblivious to the presence of an adversary – they perform a classical network code [2] whose implementation complexity is low.

II. Network model and definitions

We use the general model proposed in [8] that encompasses both wired and wireless networks. To simplify notation, we consider only the problem of communicating from a single source to a single destination. (Similarly to most network coding algorithms, our techniques generalize to multicast problems.)

A. Network Model

There is a source, Alice, and a destination, Bob, who communicate over a wired or wireless network. There is also an attacker (also referred to as an adversary) Calvin, hidden somewhere in the network. Calvin aims to prevent the transfer of information from Alice to Bob, or at least to minimize it. He can observe some of the transmissions, and can inject his own. When he injects his own packets, he pretends they are part of the information flow from Alice to Bob.

Calvin is quite strong. He is computationally unbounded. He knows the encoding and decoding schemes of Alice and Bob, and the network code implemented by the interior nodes. He also knows the exact network realization, and he gets to choose which network links to eavesdrop on and which ones to transmit fake information on. [1]

The network is modeled as a hypergraph [12]. Each packet transmission corresponds to a hyperedge directed from the transmitting node to the set of observer nodes. For wired networks, the hyperedge is a simple point-to-point link. For wireless networks, each such hyperedge is determined by instantaneous channel realizations (packets may be lost due to fading or collisions) and connects the transmitter to all

[1] We consider a model where network links rather than nodes are eavesdropped and jammed as this is more general – control of a node is equivalent to control of links attached to it.

nodes that hear the transmission. The hypergraph is unknown to Alice and Bob prior to transmission.

Source: Alice generates incompressible data that she wishes to deliver to Bob over the network. To do so, Alice encodes her data as dictated by the encoding algorithm (described in subsequent sections). She divides the encoded data into b packets. In what follows, $b = C - Z_O$. A packet contains a sequence of n symbols from the finite field $\mathbb{F}_q$. All arithmetic operations henceforth are done over symbols from $\mathbb{F}_q$. (See the treatment in [13]). Out of the n symbols in Alice's packet, δn symbols are redundancy added by the source. In our setting $\delta = o(1)$ which will imply communication at rate $R \simeq b = C - Z_O$.

Alice organizes her data into a matrix X. We denote the $(i,j)^{th}$ element in the matrix by $x(i,j)$. The i^{th} row in the matrix X is just the i^{th} packet of Alice's data. Similarly to standard network codes [2], some of the redundancy in X is devoted to sending the identity matrix, I. The rest of the redundancy in X is devoted to the secret sharing scheme to be described in Section III. Also as in [2], Alice takes random linear combinations of the rows of X to generate her transmitted packets. As the packets traverse the network, the internal nodes apply a linear transform to the packets. The identity matrix receives the same linear transform. The destination discovers the linear relation between the packets it receives and those transmitted by inspecting how the identity matrix was transformed.

Adversary: A Byzantine adversary Calvin is said to have eavesdropping capacity Z_I if it can view the information transmitted on Z_I of the edges in the underlying graph. It is said to have jamming capacity Z_O if it can inject information into the network in Z_O locations. Throughout, we assume that Calvin has unlimited computational power and that the network topology is known to Calvin. He knows the encoding and decoding schemes of Alice and Bob, and the network code implemented by the interior nodes. (Weakening any of these restrictions can be shown to not increase the achievable rate-region, hence we consider this scenario, the most favourable for Calvin.) However, the random coin tosses made by Alice as part of her encoding scheme are not known to Calvin.

Destination: Analogously to how Alice generates X, the destination Bob organizes the received packets into a matrix Y. The i^{th} received packet corresponds to the i^{th} row of Y. Bob attempts to reconstruct Alice's information, X, using the matrix of received packets Y.

B. Definitions

We define the following concepts. The *network capacity*, denoted by C, is 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. It can be also expressed as *the min-cut from source to destination*. (For the corresponding multicast case, C is defined as the minimum of the min-cuts over all destinations.) The *error probability* is the probability that Bob's reconstruction of Alice's information is inaccurate. The rate, R, is the number of information bits in a batch amortized by the length of a packet in bits. The rate R is said to be *achievable* if for any $\epsilon > 0$, any $\delta > 0$, and sufficiently large n, there exists a block-length-n network code with a redundancy δ and a probability of error less than ϵ.

III. Our main result: proof of Theorem 1

Proof: As the main part of our proof, we describe a *secret-sharing scheme* that can be operated in parallel with the classical distributed network code [2]. This parallel operation requires Alice to transmit an additional m symbols per packet, where m is a parameter of code design. In the event that $Z_I < C - Z_O$, we show that this secret-sharing scheme enables Alice to transmit a bit to Bob secretly from Calvin with probability $1 - \mathcal{O}(m\text{poly}(C)q^{-1})$. To transmit an ℓ-bit secret, the scheme can be repeated in parallel ℓ times.

Once we have designed a secret-sharing scheme, we can use Theorem 1 of [8] to conclude our proof. In Theorem 1 of [8] it is shown that if Alice and Bob have a secret channel that can transmit $C + C^2$ symbols from $\mathbb{F}_q$ (which cannot be corrupted by Calvin and are unknown to him), then there is an efficient *Shared Secret algorithm* with the following property.

Theorem 2: [8] The Shared Secret algorithm achieves rate $C - Z_O$ with code-complexity $\mathcal{O}(nC^2)$ and probability of error $(n^C C + |\mathcal{E}|)/q$. Here $\mathcal{E}$ is the set of links in the graph.

Setting ℓ to $(C + C^2)\lceil \log(q) \rceil$ implies that Bob receives the required number of secret bits to decode with asymptotically negligible probability of error. Setting m to a value that is asymptotically negligible in n ensures that the total redundancy introduced by Alice in each packet (ℓm symbols) is asymptotically negligible in n, and therefore we are done. To obtain the rate, complexity, and probability of error appearing in the statement of Theorem 1 we set m to be $\Theta(C^2)$. We now present the secret-sharing scheme.

A. Overview of Scheme

Let the secret bit that Alice wishes to communicate to Bob be denoted $I \in \{0,1\}$. Let D_0 and D_1 represent two linear hash functions that are part of code design, and hence known in advance to all parties (Alice, Bob, and Calvin). The idea behind Alice's encoding is that her transmission is chosen to satisfy D_I but not $D_{\bar{I}}$ (where $\bar{I}$ is the complement of I). Since Calvin sees fewer transmissions than Alice's rate ($Z_I < R$), based on his eavesdropping he is unable to distinguish between the case when Alice's transmission satisfies D_I and when it satisfies $D_{\bar{I}}$, hence I is secret from Calvin. Further it can be shown that despite Calvin's injecting fake packets into the network, Bob can still, with high probability, distinguish between Alice's use of D_0 and D_1 to generate her transmissions. Hence Alice can secretly and securely transmit bit I to Bob.

B. Alice's Encoder

Let $\alpha = m - b$. Two *parity-check matrices* D_0 and D_1 with αb columns and $b(Z_O + 1)$ rows each are chosen as part of code-design, and are therefore known to each of Alice, Bob and Calvin. Let $r_{i,j}$ for $i \in \{0,1\}$ and $j \in \{1,\ldots,b(Z_O+1)\}$ be $2b(Z_O+1)$ elements of $\mathbb{F}_q$ chosen independently at random during code design. For $I \in \{0,1\}$, the matrix D_I is defined by the $b(Z_O+1)$ elements $\{r_{I,j}\}$. Namely, the jth row of D_I will be the vector $(r_{I,j}, r_{I,j}^2, \ldots, r_{I,j}^{\alpha b})$. Notice that both matrices D_0, D_1 have full row rank with high probability due to well-known properties of *Vandermonde matrices*.

Recall that $I \in \{0,1\}$ is the secret bit that Alice wishes to communicate to Bob. Alice starts by picking a random length-$(b(\alpha - Z_O - 1))$ vector u with scalars from $\mathbb{F}_q$. Alice then computes a length-$b(Z_O+1)$ *padding vector* r such that the vector $D_I(u,r)^T$ equals $\mathbf{0}$. This can always be done, since the length of the vector r (which is $b(Z_O+1)$) is exactly the number of rows of the D_I matrices. Notice that r is a function of u. Finally the vector $(u, r, \text{iden})^T$ (where 'iden' is just the column version of the $b \times b$ identity matrix) is transformed into a $b \times m$ matrix S. The first $m-(Z_O+1)-b$ columns correspond to u, the next Z_O+1 columns correspond to r and the remaining columns correspond to the identity matrix. Alice then encodes S using the encoder defined in Section II-A. In what follows we show that (a) no information regarding the value of I is revealed to Calvin and (b) Bob is able to reconstruct I from the information he receives. This will conclude our proof.

C. Secrecy

As $Z_I < C$, Calvin does not know precisely the value of u and r. Specifically, Calvin knows two things about the set of possible pairs of vectors (u,r) that could have been transmitted by Alice. First, that Alice's transmission satisfies one of the two D_is. Second, the Z_I packets that Calvin eavesdropped on. Let the pairs of vectors (u,r) that are consistent with Calvin's observations of Z_I packets be denoted A. Thus A is an affine subspace of $(\mathbb{F}_q)^{\alpha b}$. This is due to the linearity of the network code constructed by the internal nodes of the network. With probability $1 - q^{-(b-Z_I)}$ (see Claim 5, [8]) over the network code implemented by internal nodes of the network, the subspace A is of dimension $\alpha(b - Z_I) = \alpha(C - Z_O - Z_I) = \delta'\alpha \geq \alpha$. Here we assume that the Z_I eavesdropping edges of Calvin are on a minimum cut between Alice and Bob, otherwise in the above and in what follows we may replace Z_I by a smaller parameter (the latter case makes the communication between Alice and Bob easier). We wish to prove that the affine subspace A, along with the knowledge that Alice's transmission satisfies one of the two D_is, does not reveal any information regarding the secret I; that is, Calvin has no knowledge as to which D_i was chosen by Alice.

Claim 1: With probability at least $1 - (2\alpha b^2(Z_O+1) + 1)q^{-1}$ over the randomness of the network code choices made by internal nodes in the network and the random construction of the matrices D_0, D_1, for any affine subspace A viewed by Calvin: $Pr_u[I=0|A] = Pr_u[I=1|A] = 1/2$.

Proof: Let $\delta' = C - Z_O - Z_I$. In what follows we assume that the network code implemented by the internal nodes of the network and the matrices D_i satisfy: (a) The dimension of A is $\delta'\alpha$, and (b) For each possible value of I, the intersection of A and the null space of D_I is of dimension $\delta'\alpha - b(Z_O+1)$.

As mentioned above, (a) will happen with probability at least $1 - q^{-1}$ over the network code choices of the internal nodes of the network. We claim that (b) will occur with probability at least $1 - \frac{2\alpha b^2(Z_O+1)}{q}$ (over the choice of $r_{i,j}$ defining the D_is). To see this, we begin by noting that any affine subspace A is determined (up to an affine shift) by the code choices of internal nodes of the network. We use the fact that these choices are made independently from the random choices made in the construction of the matrices D_0, D_1. Fix $i \in \{0,1\}$. Let M_0 be a matrix of full row-rank whose null-space equals (an affine shift of) the subspace A. Let the rank of M_0 be some value τ. Let M_j be the matrix consisting of the rows of M_0 and the top j rows of D_i. We show by induction on j that with probability $1 - \frac{\alpha b j}{q}$ the matrix M_j has a rank $\tau + j$. The case $j = 0$ is immediate. Assume the inductive step for j. Since M_j is full-rank, it has $\tau + j$ linearly independent columns. Assume these columns form a square sub-matrix M_j^{sub} of M_j of full rank. (Otherwise, one can rearrange the columns of M_j.) Consider a $\tau + j + 1$ dimension square sub-matrix M_{j+1}^{sub} of M_{j+1} that contains M_j^{sub} as a sub-matrix. Since the $j+1$th row of M_{j+1} comprises entirely of powers of $r_{i,j+1}$ the determinant of M_{j+1}^{sub} can be viewed as a polynomial P_{j+1} in $r_{i,j+1}$. Since the determinant of M_j^{sub} is non-zero, P_{j+1} has at least one non-zero coefficient and is non-zero. Further, it is of degree at most αb, and therefore has at most that many distinct roots. The determinant of M_{j+1} is zero only if $r_{i,j+1}$ is a root of P_{j+1}. But since $r_{i,j+1}$ is chosen uniformly at random from $\mathbb{F}_q$, the probability of this event is $\frac{\alpha b}{q}$. Taking the union bound over the probability of our inductive step being false for each j we have proven our assertion regarding (b).

Now consider conditioning on a network code and matrices D_0, D_1 as above. Conditioning on the network code implies that A can be one of $q^{(b-\delta')\alpha}$ parallel affine subspaces. We show that for any such A: $Pr_u[I=0|A] = Pr_u[I=1|A] = 1/2$. Indeed, by our conditioning on D_i, for each $i \in \{0,1\}$, the number of possible values of (u,r) that lie in A and are in the null space of D_i are exactly $q^{\delta'\alpha - b(Z_O+1)}$. Therefore there is no way for Calvin to distinguish which of the D_is was used by Alice. ■

D. Bob's Decoder

We now show that with high probability Bob is able to reconstruct I. For ease of notation we denote $(u,r)^T$ by x. Using the powerful analytical tools used in the proof of Theorem 2 in [8] we have

Claim 2: Bob can correctly determine a set L of potential messages of size $(\mathbb{F}_q)^{Z_O b}$ in which Alice's transmitted message x lies.

By analyzing the rank of the linear transform in (13) [8], it can be shown that the set L Bob is able to reconstruct is an affine subspace of $(\mathbb{F}_q)^{\alpha b}$ of dimension $Z_O b$ with the form $(x + \mathbf{N})$. Here $\mathbf{N}$ is a subspace of $(\mathbb{F}_q)^{\alpha b}$ whose content is determined by Calvin. Clearly, $x \in (x + \mathbf{N}) = L$. The proof that such a set L can be obtained by Bob is highly nontrivial, and is based on the fact that the information Y received by Bob is of the form $TX+E$. Here X is the information leaving Alice, T is some linear transform and E is a Z_O rank error matrix. A detailed proof can be found in Section 8 of [8].

To obtain an estimate I' to the secret I, Bob performs the following test. For both D_i, Bob checks to see if there is a vector in $x' \in (x+\mathbf{N})$ such that $D_i x' = \mathbf{0}$ (this can be done in poly(mC) time by solving a matrix equation). The test will certainly pass for D_I as $D_I x = \mathbf{0}$ by construction. Thus it suffices to show that with high probability the test will not pass for D_i when $i \neq I$.

Claim 3: With probability at least $1 - (2\alpha b^2(Z_O+1)+1)q^{-1}$, for $i \neq I$, and for every vector $z \in \mathbf{N}$ it holds that $D_i(x+z) \neq 0$. The probability is taken over the choice of u, I, the matrices D_i and the choices of the internal nodes in the network.

Proof: Let $i \neq I$. The objective of Calvin is to construct a subspace $\mathbf{N}$ such that at least one vector $z \in \mathbf{N}$ will satisfy $D_i(x+z) = \mathbf{0}$. However, when constructing $\mathbf{N}$, Calvin only has partial information of the information x transmitted by Alice. Namely, as described previously, Calvin may only identify an affine subspace A (of dimension dim$(A) = \delta'\alpha$) which includes x. Exactly the same subspace would be identified for several values of $x' = (u', r')^T$ transmitted by Alice. Moreover, as in Claim 1, we may assume that the intersection of A with the null space of both D_0 and D_1 is of dimension dim$(A) - 2b(Z_O+1)$ (this happens with the same probability specified in the proof of Claim 1). Conditioning on the matrices D_i as above, the affine subspace A viewed by Calvin, and the secret I; we have for every $v \in \mathbb{F}_q^{b(Z_O+1)}$ that: $Pr_{u'}[D_i(u',r')^T = v] = q^{-b(Z_O+1)}$. Here the probability is over random $x' = (u', r')^T$ which are consistent with A and satisfy $D_I x' = \mathbf{0}$. As Calvin can design $\mathbf{N}$ such that the set $\{D_i z : z \in \mathbf{N}\}$ is at most of size $q^{Z_O b}$, with probability q^{-b} over $x' = (u', r')$ sent by Alice, for every vector $z \in \mathbf{N}$ it holds that $D_i(x+z) \neq \mathbf{0}$. ■

IV. Future work and Conclusions

In this work we present a polynomial-time, rate-optimal distributed network code design that functions in the presence of a Byzantine adversary with substantial eavesdropping capabilities. As long as the sum of the adversary's jamming rate Z_O and his eavesdropping rate Z_I is less than the network capacity C, ($Z_O + Z_I < C$), our codes attain the optimal rate of $C - Z_O$. Our construction strongly builds upon and improves the previous results of Jaggi et al. [8] which study the achievable rate under the more restrictive assumption $2Z_O + Z_I < C$. The crux of our contribution is a secret sharing scheme that enables the sender to communicate a small message to the receiver(s) that is secret from the Byzantine adversary.

The exact rate region as a function of C, Z_O and Z_I remains an intriguing open problem. Clearly, for some cases in which $Z_O + Z_I \geq C$ the achievable rate is zero. Take, for example, the case in which $Z_O = C$. However, for many cases the rate region is positive. This occurs for example when the Byzantine adversary has full eavesdropping capabilities ($Z_I = C$) but limited jamming rate Z_O. In this setting it is shown in [8] that the (optimal) achievable rate is $C - 2Z_O$.

We note that in our secret sharing scheme the matrices D_i designed randomly by Alice are assumed to be known to both Calvin and Bob. This corresponds, for example, to the commonly used setting in which all parties have access to a public source of random bits, and these random bits are used to define D_i.

Acknowledgment

The first author was supported by grant FA9550-06-1-0155 AFOSR.

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network Information Flow. In *IEEE Transactions on Information Theory*, 2000.

[2] T. Ho., R. Koetter, M. Médard, D. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *IEEE International Symposium on Information Theory (ISIT)*, page 442, Yokohama, July 2003.

[3] D. S. Lun, M. Médard, and R. Koetter. Efficient operation of wireless packet networks using network coding. In *International Workshop on Convergent Technologies (IWCT)*, 2005.

[4] Y. Wu, P. A. Chou, and S.-Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. *IEEE Trans. on Communications*, 53(11):1906–1918, Nov. 2005.

[5] T. C. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger. Byzantine modification detection in multicast networks using randomized network coding. In *International Symposium on Information Theory*, 2004.

[6] R. W. Yeung and N. Cai. Network error correction, part 1: Basic concepts and upper bounds. Submitted to Communications in Information and Systems, 2006.

[7] N. Cai and R. W. Yeung. Network error correction, part 2: Lower bounds. Submitted to Communications in Information and Systems, 2006.

[8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard. Resilient network codes in the presence of Byzantine adversaries. Accepted for IEEE INFOCOM 2007, `http://web.mit.edu/jaggi/www/pubs/ncerror_infocom.ps.gz`.

[9] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the Association for Computing Machinery*, 40(1):17–47, January 1993.

[10] N. Cai and R. W. Yeung. Secure Nework Coding. In *International Symposium on Information Theory 2002*.

[11] J. Feldman, T. Malkin, R. Servedio and C. Stein. On the Capacity of Secure Network Coding. In *Allerton 2004*.

[12] Desmond S. Lun, Niranjan Ratnakar, Ralf Koetter, Muriel Médard, Ebad Ahmed, and Hyunjoo Lee. Achieving Minimum-Cost Multicast: A Decentralized Approach Based on Network Coding. In *IEEE INFOCOM, 2005*.

[13] R. Koetter and M. Médard. Beyond routing: An algebraic approach to network coding. In *IEEE INFOCOM, 2002*.

[14] S. Jaggi. *Design and Analysis of Network Codes*. Dissertation, California Institute of Technology, 2005.