Reverse Edge Cut-Set Bounds for Secure Network Coding

Wentao Huang and Tracey Ho
California Institute of Technology

Michael Langberg
University at Buffalo, SUNY

Joerg Kliewer
New Jersey Institute of Technology

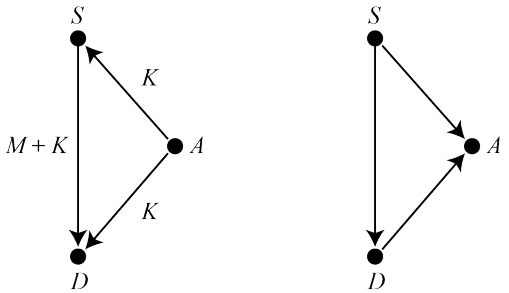
***Abstract*—We consider the problem of secure communication over a network in the presence of wiretappers. We give a new cut-set bound on secrecy capacity which takes into account the contribution of both forward and backward edges crossing the cut, and the connectivity between their endpoints in the rest of the network. We show the bound is tight on a class of networks, which demonstrates that it is not possible to find a tighter bound by considering only cut-set edges and their connectivity.**

I. Introduction

Consider a noise free communication network in which an information source S wants to transmit a secret message to the destination D over the network in the presence of a wiretapper who can eavesdrop a subset of edges. The secure network coding problem, introduced by Cai and Yeung [1], studies the secrecy capacity of such networks. Under the assumptions that 1) all edges have unit capacity; 2) the wiretapper can eavesdrop any subset of edges of size up to z; 3) only S has the ability to generate randomness, [1] shows that the secrecy capacity is $x-z$, where x is the min-cut from S to D. Subsequent works have studied various ways to achieve this capacity with codes on fields of smaller size [2], coset codes [3], and universal codes [4].

Though the secrecy capacity is well understood in this special case, much less is known under a more general setting. In particular, if either edge capacities are not uniform, or the collection of possible wiretap sets is more general (i.e., not characterized by a simple parameter z), Cui et al. [5] show that finding the secrecy capacity is NP-hard. On the other hand, if randomness is allowed to be generated at non-source nodes, Cai and Yeung [6] give an example in which this can be advantageous, and provide a necessary and sufficient condition for a linear network code to be secure. However, for this case [7], [8] show that finding the secrecy capacity is at least as difficult as the long-standing open problem of determining the capacity region of multiple-unicast network coding. To the best of our knowledge, under these general settings, the only known bounds of secrecy capacity are given implicitly in terms of entropy functions/entropic region [9], [10], whereas determining the entropic region is a long standing open problem as well.

This paper gives the first explicit upper bound on secrecy capacity for the secure network coding problem in the case where non-source nodes can generate randomness and the collection of wiretap sets is arbitrary. Our bound has an intuitive graph-theoretic interpretation with the observation that unlike traditional cut-set bounds which only consider forward edges, for the secure network coding problem backward edges may also be helpful in a cut if down-stream (hence non-source) nodes can generate randomness, as shown in Fig. 1-(a). Here the backward edge (A, S) can transmit a random key back to the source to protect the message, and enable secrecy rate 1 to be achieved. However, one should be careful in counting the contribution of backward edges since they are not always useful, such as edge (D, A) in Fig. 1-(b). Notice that the networks of (a) and (b) are identical from the perspective of cuts because they each contain a cut with two forward edges and a cut with one forward edge and one backward edge. Hence to avoid a loose bound we have to see beyond the cut: in this simple example the backward edge in (a) is helpful because it is connected to the forward edge, while the one in (b) is not. More generally, this motivates us to take into account the connectivity from backward edges to forward edges, described by a 0-1 connectivity matrix C. We show that the rank structure of the submatrices of C characterizes the utility of the backward edges, and use this to obtain an upper bound on secure capacity.



(a) Backward edge helpful (b) Not helpful

Fig. 1: Networks with unit capacity edges and $z = 1$.

Finally we show that given any cut in a network, we can construct a corresponding network with the same cut, i.e., the capacity and connectivity of the cut-set edges in the two networks are the same, such that our bound is achievable in the constructed network by random scalar linear codes. Hence it is not possible to find a better bound by merely considering the cut-set edges and their connectivity.

[1]This work has been supported in part by NSF Grant CNS-0905615, CCF-1440001, CCF-1440014 and CCF-1038578.

978-1-4799-5186-4/14/$31.00 ©2014 IEEE

II. Models

Consider a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let $\mathcal{A} \subset 2^{\mathcal{E}}$ be a collection of possible wiretap sets. Since $\mathcal{A}$ is arbitrary (i.e., non-unform), without loss of generality we may assume all edges have capacity one bit per unit time. Any edge of larger capacity can be replaced by multiple parallel edges in both $\mathcal{G}$ and $\mathcal{A}$. In this work we focus on the single source single terminal setting. While seemingly simple, in this setting finding the capacity is as at least as hard as determining the capacity for multiple unicast network coding [7]. Source node S wants to deliver a secret message M uniformly drawn from a finite alphabet set $\mathcal{S}_n$, to sink node D using a code with length n, under perfect secrecy with respect to $\mathcal{A}$. Namely, $\forall A \in \mathcal{A}$, $I(M; \mathfrak{X}^n(A)) = 0$, where $\mathfrak{X}^n(A)$ denotes the signals transmitted on A. Each node $i \in \mathcal{V}$, can generate independent randomness K_i to be used as keys to protect the message. A secrecy rate R_S is feasible if there exists such a code with $|\mathcal{S}_n| = 2^{nR_S}$. The secrecy capacity $\mathfrak{C}$ of the network is defined as the supremum of all feasible secrecy rates.

Consider an arbitrary cut $V \subset \mathcal{V}$ such that $S \in V$ and $D \in V^c$. Denote by $E_V^{\text{fwd}} = \{(i,j) \in \mathcal{E} : i \in V, j \in V^c\}$ the set of forward edges with respect to V, and by $E_V^{\text{bwd}} = \{(i,j) \in \mathcal{E} : i \in V^c, j \in V\}$ the set of backward edges. Letting $|E_V^{\text{fwd}}| = x$ and $|E_V^{\text{bwd}}| = y$, we denote the x forward edges by $e_1^{\text{fwd}}, e_2^{\text{fwd}}, ..., e_x^{\text{fwd}}$, and the y backward edges by $e_1^{\text{bwd}}, e_2^{\text{bwd}}, ..., e_y^{\text{bwd}}$. Let $C_{b\to f,V} = (c'_{ij})$ be an $x \times y$ (0-1) matrix characterizing the connectivity from the backward edges to the forward edges. More precisely,

$$c'_{ij} = \begin{cases} 1 & \text{if } \exists \text{ a directed path from head}(e_j^{\text{bwd}}) \text{ to tail}(e_i^{\text{fwd}}) \\ & \text{that does not pass through any nodes in } V^c \\ 0 & \text{otherwise} \end{cases}$$

III. Cut-set Bound

We derive a cut-set bound on the secure capacity, with respect to a cut V and its connectivity matrix $C_{b\to f,V}$. Define

$$C_V = \begin{pmatrix} C_{b\to f,V} \\ I_y \end{pmatrix} = (c_{ij}),$$

where I_y is the identity matrix of order y. For notational convenience, in the following we will drop the subscript V in $C_{b\to f,V}$ and C_V. Rows in C correspond to edges crossing the cut in $\mathcal{G}$. Denote by $\mathcal{A}_V = \{A \cap (E_V^{\text{fwd}} \cup E_V^{\text{bwd}}) : A \in \mathcal{A}\}$. For $A \in \mathcal{A}_V$, denote by U_A the submatrix of C formed by the rows corresponding to edges in A. Let $\mathcal{U} = \{U_A, A \in \mathcal{A}_V\}$. We can maximize the rank of each submatrix in $\mathcal{U}$ simultaneously subject to the zero constraints in C. The proof is given in [11].

Lemma 1. *There exist field size q and matrix $\bar{C} \in \mathbb{F}_q^{(x+y)\times y} = (\bar{c}_{ij})$ such that: 1) $\bar{c}_{ij} = 0$ if $c_{ij} = 0$; 2) $\forall\ U_A \in \mathcal{U}$, denote the corresponding submatrix of $\bar{C}$ by $\bar{U}_A$, then*

$$\textit{rank}(\bar{U}_A) = \max_{W \in \mathbb{F}_q^{m\times y}, w_{ij}=0 \text{ if } u_{ij}=0} \textit{rank}(W),$$

where $m \times y$ is the size of U_A and u_{ij} are the entries of U_A. In particular, $q > |\mathcal{U}|(x+y)y$ is sufficient.

For $U_A \in \mathcal{U}$, let $\bar{U}_A$ be the corresponding submatrix of $\bar{C}$ as defined in Lemma 1. We now state our main result.

Theorem 1. *The secrecy capacity is bounded by*

$$\mathfrak{C} \le x + \min_{A \in \mathcal{A}_V} \text{rank}(\bar{U}_A) - |A|$$

In the special case of uniform size wiretap sets, Theorem 1 reduces to the following form.

Corollary 1. *For $\mathcal{A} = \{A \subset \mathcal{E} : |A| \le z\}$, define $k_b = \min_{\{\bar{U}:\ z \times y \text{ submatrix of } \bar{C}\}} \text{rank}(\bar{U})$, then*

$$\mathfrak{C} \le x + k_b - z$$

We remark that a simple cut-set bound in the setting of Corollary 1 is $\mathfrak{C} \le x + y - z$. Hence our result can be viewed as an improvement over this bound which takes into account the connectivity of the cut-set edges as expressed by $k_b \le y$.

In the following, we will prove Theorem 1. Given a cut of x forward edges, y backward edges, and the connectivity matrix $C_{b\to f}$, we construct an upper bounding network $\bar{\mathcal{G}}$ as follows: 1) Absorb all nodes downstream of the cut, i.e., all $v \in V^c$, into the sink D. So for all i, j, head(e_i^{fwd})=D, tail(e_j^{bwd})=D. 2) Connect the source to each forward edge with infinite unit capacity edges $(S, \text{tail}(e_i^{\text{fwd}}))$. 3) Connect the backward edges to the forward edges according to $C_{b\to f}$. More precisely, add an infinite number of unit capacity edges $(\text{head}(e_j^{\text{bwd}}), \text{tail}(e_i^{\text{fwd}}))$ if and only if $c'_{ij} = 1$. Finally, in $\bar{\mathcal{G}}$ we only allow S and D to generate independent randomness.

Lemma 2. *The secure unicast capacity of $\bar{\mathcal{G}}$ upper bounds the secure unicast capacity of $\mathcal{G}$.*

Proof Sketch: Note that all connections composed of infinite parallel unit capacity edges are perfectly secure because they can be protected by an infinite number of local keys. Hence any secure code on $\mathcal{G}$ can be simulated on $\bar{\mathcal{G}}$ securely. Only S and D need to generate randomness, because randomness generated at any other node can instead be generated at S and sent to the node through the infinite parallel edges. Refer to [11] for details. ∎

In the following we will consider $\bar{\mathcal{G}}$ instead of $\mathcal{G}$ unless otherwise specified. Note that in $\bar{\mathcal{G}}$ only the edges crossing the cut V are vulnerable, hence we may assume any wiretap set only contains these edges. Therefore $\mathcal{A} = \mathcal{A}_V$ in $\bar{G}$. Let $\mathfrak{f}_1, ..., \mathfrak{f}_x$ be the signals transmitted on edges $e_1^{\text{fwd}}, ..., e_x^{\text{fwd}}$; $\mathfrak{b}_1, ..., \mathfrak{b}_y$ be the signals transmitted on edges $e_1^{\text{bwd}}, ..., e_y^{\text{bwd}}$. For any $e \in \mathcal{E}$, define a set of backward signals w.r.t. to e as

$$\mathfrak{D}_e = \begin{cases} \{\mathfrak{b}_j\} & \text{if } e = e_j^{\text{bwd}} \\ \{\mathfrak{b}_j : c'_{ij} = 1\} & \text{if } e = e_i^{\text{fwd}} \end{cases}$$

and for any $A \in \mathcal{A}$, define $\mathfrak{D}_A = \{\cup \mathfrak{D}_e : e \in A\}$. The following lemma shows a useful property of the rank structure of the submatrices of $\bar{C}$.

Lemma 3. *For any $A \in \mathcal{A}$, there exists a partition $A = A_1 \cup A_2$, such that $|\mathfrak{D}_{A_1}| + |A_2| = \textit{rank}(\bar{U}_A)$.*

Proof: Let $r = \text{rank}(\bar{U}_A)$, so in $\bar{U}_A$ there exist r non-zero entries in different columns and different rows. An entry

in $\bar{U}_A$ can be non-zero only if this entry is 1 in U_A, hence U_A contains r entries of value 1 in different columns and different rows. Perform column and row permutations to move these 1's such that $U_A(r+1-i,i) = 1, \forall 1 \leq i \leq r$, i.e., they become the counter-diagonal entries of the upper-left block formed by the first $r \times r$ entries. See Fig. 2 for an illustration. Note that permutations in U_A are merely reordering of edges, and for notational convenience we denote the matrix after permutations as U_A still. It then follows that $U_A(i,j) = 0, \forall r < i \leq |A|, r < j \leq y$. Otherwise if any entry in this lower right block is non-zero, setting this and the aforementioned r counter-diagonal entries to 1, and all other entries to 0 yields a matrix that satisfies the zero constraint in U_A and it has rank $r+1$. But this is a contradiction because $r = \text{rank}(\bar{U}_A)$ is the maximum rank by Lemma 1. Hence we label this block as *zero*.

The rest of the proof proceeds as follows. We first propose an recursive algorithm that permutes U_A and labels it block-wise. Then we show the labels are correct. Finally we show the labels reveal the structure of U_A from which we can prove the lemma statement conveniently.

The input to the algorithm is a matrix G of arbitrary size $m \times n$ and a positive integer parameter k, such that the upper-left block G_{UL} formed by the first $k \times k$ entries of G has all 1's in its counter-diagonal. The output is a permuted and labeled version of G and an integer t. The algorithm starts with the lower left block $G_{LL} = (g_{ij}), k < i \leq m, 1 \leq j \leq k$. If every column of G_{LL} is non-zero, label this block as *non-zero*, label G_{UL} as *counter-diagonal*, label the block $G_{UR} = (g_{ij}), 1 \leq i \leq k, k < j \leq n$ as *zero**, return $t := 0$ and terminate. If $G_{LL} = \mathbf{0}$ or empty, label this block (if not empty) as *zero*, label G_{UL} as *counter-diagonal*, label G_{UR} as *arbitrary*, return $t := k$ and terminate.

Otherwise G_{LL} contains both zero and non-zero columns. In this case, perform column permutations in G to move all non-zero columns of G_{LL} to the left and zero columns to the right. Assume that there are u such non-zero columns and v zero columns. Label the block $(g_{ij}), k < i \leq m, 1 \leq j \leq u$ as *non-zero* and label the block $(g_{ij}), k < i \leq m, u < j \leq k$ as *zero*. At this point some of the 1's originally in the counter-diagonal of G_{UL} are misplaced due to column permutations, perform row permutations to move them back to the counter-diagonal. Note that only the first k rows need to be permuted and the lower labeled block(s) is not affected. Label the block $(g_{ij}), k-u+1 \leq i \leq k, 1 \leq j \leq u$ as *counter-diagonal*, label the block $(g_{ij}), 1 \leq i \leq k-u, 1 \leq j \leq u$ as *arbitrary*, and label the block $(g_{ij}), k-u+1 \leq i \leq k, k < j \leq n$ as *zero**. Then truncate the first u columns and the last $m-k$ rows from G. Notice that the block formed by the first $v \times v$ entries in the truncated G has all 1's in its counter-diagonal. Now apply the algorithm recursively to the truncated G with parameter $v < k$. The algorithm must terminate because the input parameter is a positive finite integer and cannot decrease indefinitely. Applying the algorithm to the matrix U_A with parameter $k := r$ will permute the rows and columns of U_A and label it completely. Refer to Figure 2 for an example.

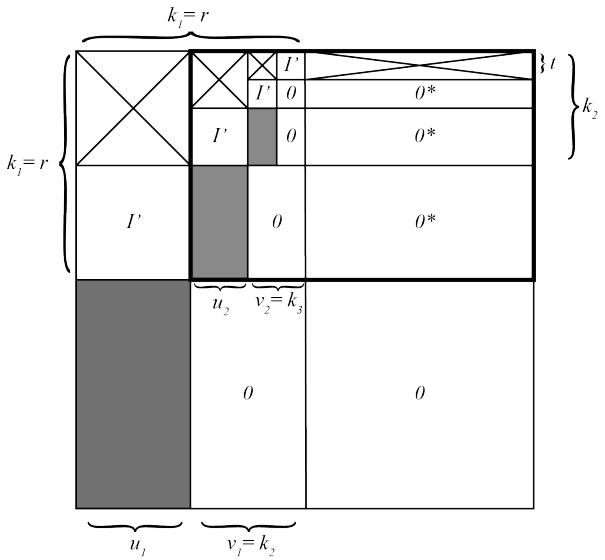


Fig. 2: An example of a labeled U_A. *zero* blocks are indicated by 0; *zero** blocks are indicated by 0*; *counter-diagonal* blocks are indicated by I'; *non-zero* blocks are colored in gray and *arbitrary* blocks are crossed. The algorithm terminates in four iterations and returns t. Key parameters of the first two iterations are illustrated and the subscripts denote iteration numbers. The truncated G after the first iteration is highlighted with bold lines. Note that the first t rows correspond to A_2, and the remaining rows correspond to A_1.

In the next step, we show that the algorithm labels G correctly. Note that the algorithm always labels *counter-diagonal*, *non-zero* and *zero* correctly, i.e., by construction all blocks labeled *counter-diagonal* are square and have 1's in their counter-diagonals (but the off-counter-diagonal entries are arbitrary); all blocks labeled *non-zero* do not contain zero columns; and all blocks labeled *zero* are zero. The only non-trivial label is *zero**, and we will show that a zero*-labeled block is indeed zero. To prove this, notice that all *zero** blocks pile up at the last $y-r$ columns of U_A, and consider any entry α_1 of a *zero** block. By the algorithm the row of α_1 must intersect a unique *counter-diagonal* block. Denote the intersecting counter-diagonal entry of the *counter-diagonal* block as β_1. By the algorithm this intersecting *counter-diagonal* block must lie immediately on top of a *non-zero* block. Therefore the lower *non-zero* block contains a non-zero entry α_2 in the same column as β_1. And again the row of α_2 will intersect a counter-diagonal entry β_2 of a *counter-diagonal* block. In exactly the same way we are able to find a sequence of entries $\alpha_3, \beta_3, \alpha_4, \beta_4 ...$ until we reach the lowest *non-zero* block. Note that all these entries belong to distinct blocks, and because there is a finite number of blocks, the series is finite. In particular, let w be the number of *counter-diagonal* blocks that lie below or intersect the row of α_1, then we can find $\beta_1, ..., \beta_w$ and $\alpha_1, ..., \alpha_{w+1}$, where α_{w+1} lies in the lowest *non-zero* block. Now suppose for the sake

of contradiction that α_1 is non-zero, set $\alpha_1, ..., \alpha_{w+1}$ to 1, set all counter-diagonal entries of all *counter-diagonal* blocks except $\beta_1, ..., \beta_w$ to 1, and set all other entries to 0. This produces a matrix of rank $r+1$ because all $r+1$ 1's appear in distinct columns and rows, contradicting the fact that $\bar{U}_A$ is rank maximized. Hence all zero*-label blocks are zero.

We are ready to prove the lemma statement. After permutations, the block $U_A(i,j), t < i \le |A|, r-t+1 \le j \le y$ is zero. Now partition A into $A_1 \cup A_2$, where A_2 is the subset of edges corresponding to the first t rows of the permuted U_A. So $|A_2| = t$ and $|A_1| = |A| - t$. But the zero constraints in U_A imply that $\mathfrak{D}_{A_1}$ contains $r-t$ of the $\mathfrak{b}_j$'s corresponding to the first $r-t$ columns, hence $|\mathfrak{D}_{A_1}| = r-t$. Finally $|\mathfrak{D}_{A_1}| + |A_2| = r$. ∎

Corollary 2. *Partition A into $A_1 \cup A_2$ as in Lemma 3, and partition A_1 as $A_F \cup A_B$, where $A_F \subset \{e_1^{fwd}, ..., e_x^{fwd}\}$, $A_B \subset \{e_1^{bwd}, ..., e_y^{bwd}\}$, then $H(\mathfrak{D}_{A_F}|\mathfrak{D}_{A_B}) \le$ rank$(\bar{U}_A) - |A_B| - |A_2|$.*

Proof: Note that by definition $\mathfrak{D}_{A_B} = A_B$, $\mathfrak{D}_{A_1}\backslash\mathfrak{D}_{A_B} = \mathfrak{D}_{A_F}\backslash\mathfrak{D}_{A_B}$. So $H(\mathfrak{D}_{A_F}|\mathfrak{D}_{A_B}) \le |\mathfrak{D}_{A_F}\backslash\mathfrak{D}_{A_B}| = |\mathfrak{D}_{A_1}\backslash\mathfrak{D}_{A_B}| = |\mathfrak{D}_{A_1}\backslash A_B| = |\mathfrak{D}_{A_1}| - |A_B|$. The result then follows from Lemma 3. ∎

Due to the cyclic nature of $\mathcal{G}'$, imposing delay constraints on some edges is necessary to avoid stability and causality issues. It suffices to assume there is unit delay on edges $e_1^{\text{fwd}}, ..., e_x^{\text{fwd}}, e_1^{\text{bwd}}, ..., e_y^{\text{bwd}}$. Note that any realistic systems should comply with these minimal delay constraints. Let t be a time index, denote $\mathfrak{f}_i[t]$ and $\mathfrak{b}_j[t]$ as the signals transmitted on edges e_i^{fwd} and e_j^{bwd} during the t-th time step. Consider any secure code that finishes within T time steps. Below we upper bound the rate of this code by $x + \text{rank}(\bar{U}_A) - |A|$, $\forall A \in \mathcal{A}$, as claimed in Theorem 1. We first prove a lemma.

Lemma 4. *Consider arbitrary random variables X, Y, Z, W, if $(Z, W) \to (Y, W) \to X$, then*

$$H(X|Z,W) \ge H(X|W) - I(Y;X|W)$$

Proof: Note that $H(X,Y|W) = H(X|Y,W) + H(Y|W) = H(Y|X,W) + H(X|W)$. So $H(X|Y,W) = H(X|W) + H(Y|X,W) - H(Y|W) = H(X|W) - I(Y;X|W)$. Finally by the Markov chain $H(X|Z,W) \ge H(X|Y,W)$, we prove the claim. ∎

Proof (of Theorem 1): Define $\mathfrak{F}[t] = \{\mathfrak{f}_1[t], ..., \mathfrak{f}_x[t]\}$ as all the forward signals at time t, and $\mathfrak{B}[t] = \{\mathfrak{b}_1[t], ..., \mathfrak{b}_y[t]\}$ as all the backward signals. Let $\underline{\mathfrak{F}} = \{\mathfrak{F}[1], ..., \mathfrak{F}[T]\}$, $\underline{\mathfrak{B}} = \{\mathfrak{B}[1], ..., \mathfrak{B}[T]\}$. Consider any $A \in \mathcal{A}$, partition it into $A_1 + A_2$ as in Lemma 3 and partition A_1 into $A_F + A_B$ as in Corollary 2. Let $\mathfrak{F}_A[t] = \{\mathfrak{f}_i[t] : e_i^{\text{fwd}} \in A_F\}$ denote the signals transmitted on A_F at time t, and likewise let $\mathfrak{B}_A[t] = \{\mathfrak{b}_j[t] : e_j^{\text{bwd}} \in A_B\}$. Let $a = |A_F|$, $b = |A_B|$, $c = |A_2|$. Recall that $\mathfrak{D}_A[t]$ is the set of signals sent by all backward edges to the edges in A at time t, M is the source message, and K_D is all randomness generated by the sink. Now we upper bound the message rate R_s. It follows,

$$\begin{aligned} TR_s = H(M) &\overset{(a)}{=} H(M|K_D) - H(M|\underline{\mathfrak{F}}, \underline{\mathfrak{B}}, K_D) \\ &= I(M; \underline{\mathfrak{F}}, \underline{\mathfrak{B}}|K_D) \\ &= H(\underline{\mathfrak{F}}, \underline{\mathfrak{B}}|K_D) - H(\underline{\mathfrak{F}}, \underline{\mathfrak{B}}|M, K_D), \end{aligned} \quad (1)$$

where (a) is due to the decoding constraint and the fact that K_D is independent from M. We consider the first term in (1). Expanding it according to the chain rule, we have

$$\begin{aligned} &H(\underline{\mathfrak{F}}, \underline{\mathfrak{B}}|K_D) = H(\mathfrak{F}[1], ..., \mathfrak{F}[T], \mathfrak{B}[1], ..., \mathfrak{B}[T]|K_D) \\ &\overset{(b)}{=} \sum_{i=1}^{T} H(\mathfrak{F}[i], \mathfrak{B}[i]|\mathfrak{F}[0...i-1], \mathfrak{B}[0...i-1], K_D) \\ &\overset{(c)}{=} \sum_{i=1}^{T} H(\mathfrak{F}[i]|\mathfrak{F}[0...i-1], \mathfrak{B}[0...i-1], K_D) \\ &\overset{(d)}{\le} \sum_{i=1}^{T} H(\mathfrak{F}[i]\backslash\mathfrak{F}_A[i]|\mathfrak{F}[0...i-1], \mathfrak{B}[0...i-1], K_D) \\ &\qquad + H(\mathfrak{F}_A[i]|\mathfrak{F}[0...i-1], \mathfrak{B}[0...i-1], K_D) \\ &\overset{(e)}{\le} T(x-a) + \sum_{i=1}^{T} H(\mathfrak{F}_A[i]|F_A[0...i-1], \mathfrak{B}_A[0...i-1]) \\ &\overset{(f)}{=} T(x-a) + \sum_{i=1}^{T} H(\mathfrak{F}_A[i]|\mathfrak{F}_A[0...i-1], \mathfrak{B}_A[0...i-1], M) \end{aligned} \quad (2)$$

Here (b) follows from the chain rule; (c) follows from the fact that $\mathfrak{B}[i]$ is a function of the conditions; (d) follows from the chain rule and conditioning reduces entropy; (e) follows from conditioning reduces entropy; and (f) follows from the secrecy constraint, i.e., M is independent from $\mathfrak{F}_A[0...T], \mathfrak{B}_A[0...T]$. Next we deal with the second term in (1).

$$\begin{aligned} &H(\underline{\mathfrak{F}}, \underline{\mathfrak{B}}|M, K_D) \ge H(\mathfrak{F}_A[1...T], \mathfrak{B}_A[1...T]|M, K_D) \\ &= \sum_{i=1}^{T} H(\mathfrak{F}_A[i], \mathfrak{B}_A[i]|\mathfrak{F}_A[0...i-1], \mathfrak{B}_A[0...i-1], M, K_D) \\ &\ge \sum_{i=1}^{T} H(\mathfrak{F}_A[i]|\mathfrak{F}_A[0...i-1], \mathfrak{B}_A[0...i-1], M, K_D) \\ &\overset{(g)}{\ge} \sum_{i=1}^{T} H(\mathfrak{F}_A[i]|\mathfrak{F}_A[0...i-1], \mathfrak{B}_A[0...i-1], M) \\ &\qquad - I(\mathfrak{D}_{A_F}[0...i-1]; \mathfrak{F}_A[i]|\mathfrak{F}_A[0...i-1], \mathfrak{B}_A[0...i-1], M) \end{aligned} \quad (3)$$

where (g) is due to Lemma 4 by regarding $\mathfrak{F}_A[i]$ as X; $\mathfrak{D}_{A_F}[0, ..., i-1]$ as Y; K_D as Z; and $M, \mathfrak{F}_A[0, ..., i-1], \mathfrak{B}_A[0, ..., i-1]$ as W. Note that indeed $\mathfrak{F}_A[i]$ learns everything it can about K_D from $\mathfrak{D}_{A_F}[0, ..., i-1]$. Plug (2) and (3) into (1) yields,

$$\begin{aligned} &TR_s \le T(x-a) \\ &+ \sum_{i=1}^{T-1} I(\mathfrak{D}_{A_F}[1...i]; \mathfrak{F}_A[i+1]|\mathfrak{F}_A[1...i], \mathfrak{B}_A[1...i], M) \end{aligned} \quad (4)$$

Finally we bound the mutual information terms that appear in (4). These terms characterize how the sink generated

keys at times $1, ..., i$ contribute to randomizing (and therefore protecting) the forward signals transmitted at time $i+1$.

$$\begin{aligned}
&\sum_{j=1}^{T-1} I(\mathfrak{D}_{A_F}[1...j]; \mathfrak{F}_A[j+1] | \mathfrak{F}_A[1...j], \mathfrak{B}_A[1...j], M) \\
&\overset{(h)}{=} \sum_{j=1}^{T-1}\sum_{i=1}^{j} I(\mathfrak{D}_{A_F}[i]; \mathfrak{F}_A[j+1] | \mathfrak{F}_A[1...j], \\
&\qquad\qquad \mathfrak{B}_A[1...j], \mathfrak{D}_{A_F}[0...i-1], M) \\
&\overset{(i)}{=} \sum_{i=1}^{T-1}\sum_{j=i}^{T-1} I(\mathfrak{D}_{A_F}[i]; \mathfrak{F}_A[j+1] | \mathfrak{F}_A[1...j], \\
&\qquad\qquad \mathfrak{B}_A[1...j], \mathfrak{D}_{A_F}[0...i-1], M) \\
&\overset{(j)}{\leq} \sum_{i=1}^{T-1} I(\mathfrak{D}_{A_F}[i]; \mathfrak{F}_A[i+1] | \mathfrak{F}_A[1...i], \mathfrak{B}_A[1...i], \mathfrak{D}_{A_F}[0...i-1], M) \\
&\quad + \sum_{i=1}^{T-2}\sum_{j=i+1}^{T-1} I(\mathfrak{D}_{A_F}[i]; \mathfrak{F}_A[j+1], \mathfrak{B}_A[j] \\
&\qquad\qquad |\mathfrak{F}_A[1...j], \mathfrak{B}_A[1...j-1], \mathfrak{D}_{A_F}[0...i-1], M) \\
&\overset{(k)}{=} \sum_{i=1}^{T-1} I(\mathfrak{D}_{A_F}[i]; \mathfrak{F}_A[i+1...T], \mathfrak{B}_A[i+1...T-1] \\
&\qquad\qquad |\mathfrak{F}_A[1...i], \mathfrak{B}_A[1...i], \mathfrak{D}_{A_F}[0...i-1], M) \\
&\overset{(l)}{\leq} \sum_{i=1}^{T-1} H(\mathfrak{D}_{A_F}[i] | \mathfrak{B}_A[i]) \\
&\overset{(m)}{\leq} (T-1)(\text{rank}(\bar{U}_A) - b - c) \qquad (5)
\end{aligned}$$

Here (h) follows from the chain rule for mutual information; (i) follows from changing the order of summation; (j) follows from the fact that $I(X;Y|Z) \leq I(X;Y,Z)$; (k) follows from the chain rule for mutual information; (l) follows from the definition of mutual information and conditioning reduces entropy; and (m) follows from Corollary 2 since $\mathfrak{D}_{A_B}[i] = \mathfrak{B}_A[i]$. Finally substitute (5) into (4) we have

$$\begin{aligned}
R_S &\leq (T(x - a + \text{rank}(\bar{U}_A) - b - c) - \text{rank}(\bar{U}_A) + b + c)/T \\
&= (T(x + \text{rank}(\bar{U}_A) - |A|) - \text{rank}(\bar{U}_A) + b + c)/T \\
&< x + \text{rank}(\bar{U}_A) - |A|
\end{aligned}$$

■

IV. Achievability

In this section we construct a scalar linear code that achieves the upper bound of Theorem 1 in $\bar{\mathcal{G}}$, thereby finding the secrecy capacity of $\bar{\mathcal{G}}$. The achievability result also implies this is the best upper bound one can obtain by looking at a cut and its connectivity matrix. We will build the code on top of $\bar{C}$, the intuition being that $\bar{C}$ is rank maximized and therefore suggests an "optimal" way of using the backward keys (i.e., sink generated randomness) to provide maximum randomization and protection. Hence what remains to be designed are the forward keys (source generated randomness that is independent from the message), and it turns out that $k_f = \max_{A\in\mathcal{A}} |A| - \text{rank}(\bar{U}_A)$ units of forward keys are sufficient in $\bar{\mathcal{G}}$. Therefore a rate of $R_s = x - k_f = x + \min_{A\in\mathcal{A}} \text{rank}(\bar{U}_A) - |A|$ can be achieved.

Let $m_1, ..., m_{R_s}$ be the messages, $K_S^1, ..., K_S^{k_f}$ be the source generated keys, and $K_D^1, ..., K_D^y$ be the sink generated keys, all of them i.i.d. uniformly distributed in $\mathbb{F}_q$. Let $E = (e_{ij}) \in \mathbb{F}_q^{(x+y)\times(x+y)}$ be the encoding matrix, defined by

$$E = \left(\begin{array}{c|c} \dfrac{G}{\mathbf{0}} & \bar{C} \end{array} \right) = \left(\begin{array}{c|c} E_M & E_K \end{array} \right), \qquad (6)$$

where G is a random matrix of size $x \times x$ with entries i.i.d. uniformly chosen from $\mathbb{F}_q$, $\mathbf{0}$ is a zero matrix of size $y \times x$ to ensure causality, E_M are the first R_s columns of E, and E_K are last $k_f + y$ columns of E. The signals on the cut are

$$\begin{pmatrix} F_1 \\ \cdots \\ F_x \\ B_1 \\ \cdots \\ B_y \end{pmatrix} = \left(\begin{array}{c|c} E_M & E_K \end{array} \right) \begin{pmatrix} m_1 \\ \cdots \\ m_{R_s} \\ \hline K_S^1 \\ \cdots \\ K_S^{k_f} \\ K_D^1 \\ \cdots \\ K_D^y \end{pmatrix}$$

Notice that E is a full rank square matrix with high probability since G is generic and the bottom y rows of $\bar{C}$ are linearly independent. Therefore the sink D can decode the message and all key variables. To show the code is secure, note that E_K is designed with appropriate size, i.e., its first k_f columns correspond to source generated keys and the remaining y columns correspond to sink generated keys. Since E_K is generic, with high probability a submatrix of E_K corresponding to any wiretap set A would have full rank $|A|$. This implies that any set of signals that an eavesdropper may see is protected by sufficient random keys and does not reveal information on the message. The following theorem, whose proof we defer to the full length version [11], formalizes this idea.

Theorem 2. *The code E is secure with probability at least $1 - |\mathcal{A}|\frac{k_f(x+y)}{q}$.*

References

[1] N. Cai and R. W. Yeung, "Secure network coding," in *in Proc. IEEE ISIT*, 2002.

[2] J. Feldman, T. Malkin, R. Servedio, and C. Stein, "On the capacity of secure network coding," in *42nd Annual Allerton Conference*, 2004.

[3] S. E. Rouayheb, E. Soljanin, and A. Sprintson, "Secure network coding for wiretap networks of type II," *IEEE Transactions on Information Theory*, vol. 58, pp. 1361 – 1371, 2012.

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

[5] T. Cui, T. Ho, and J. Kliewer, "On secure network coding with nonuniform or restricted wiretap sets," *IEEE Transactions on Information Theory*, vol. 59, pp. 166 – 176, 2013.

[6] N. Cai and R. W. Yeung, "A security condition for multi-source linear network coding," in *Proc. of IEEE ISIT*, Jun. 2007, pp. 561–565.

[7] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "On secure network coding with uniform wiretap sets," in *IEEE NetCod*, 2013.

[8] T. Chan and A. Grant, "Mission impossible: Computing the network coding capacity region," in *Proc. IEEE ISIT*, July 2008, pp. 320 –324.

[9] ——, "Capacity bounds for secure network coding," in *Australian Communications Theory Workshop*, 2008.

[10] S. Jalali and T. Ho, "On capacity region of wiretap networks," 2012, http://arxiv.org/abs/1212.3859.

[11] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "Reverse edge cut-set bounds for secure network coding," http://www.its.caltech.edu/~whuang/ISIT2014-Wentao.pdf, 2014.