Untrusting Network Coding

Yasin Büyükalp*, Ghid Maatouk†, Vinod M. Prabhakaran‡, Christina Fragouli†
* Middle East Technical University, Turkey
E-mail: e162594@metu.edu.tr
† Ecole Polytechnique Fédérale de Lausanne, Switzerland
E-mail: {ghid.maatouk, christina.fragouli}@epfl.ch
‡ Tata Institute of Fundamental Research, India
E-mail: vinodmp@tifr.res.in

***Abstract*—In networks that perform linear network coding, an intermediate network node may receive a much larger number of linear equations of the source symbols than the number of messages it needs to send. For networks constructed by untrusted nodes, we propose a relaxed measure of security: we want to be untrusting, and allow each intermediate node to only learn as much information as the number of independent messages it needs to send. In this paper we formulate this problem and provide sufficient and necessary conditions for classes of combination networks.**

I. Introduction

We consider a set of h non-colocated unit rate sources $S_1, \ldots, S_h$ that would like to multicast information to a set of receivers (with min-cut h) over a network $G = (V, E)$ with untrusted nodes, i.e., where the intermediate network nodes may try to passively eavesdrop and infer the sources' information. As communication networks move towards more flexible, temporary and less controlled structures, communication using such untrusted nodes is common, and we expect it to become even more so.

One approach to dealing with untrusted networks is to impose strict security requirements, namely, to require that the network nodes learn nothing about the source messages. We can indeed use secure network coding designs [1], [2], [3], [4], [5], [6] to protect from a passive eavesdropper, Eve, who has access to at most k edges of the network; in our setting, where Eve has access to a network node, k would be the number of incoming edges in Eve's node. The security comes at the cost of throughput; namely, applying the techniques of [1]-[6], we cannot hope to achieve an information rate higher than $h - k$. Thus, if Eve's node has the same min-cut as a receiver, our throughput becomes zero.

In this paper we take a different approach: we do not want to sacrifice throughput, but we also do not want to be secure, we just want to be 'untrusting.' That is, we do not require that the network nodes learn nothing about the source messages; we just require they do not learn anything they *do not need to know*. We ask that each intermediate node only learns as many independent linear combinations of the source messages as it needs to forward (even if it has a much larger number of incoming edges), so that receivers experience rate h.

To achieve this, we allow the sources to collaborate with each other, i.e., to exchange information among them; we assume they can achieve this through an auxiliary network H that does not use any of the edges in G. Our cost is the bandwidth that is used for the source collaboration.

To summarize, our problem formulation differs from secure network coding in two ways: (i) we do not ask for security guarantees in the usual sense, since the untrusted nodes still learn some information about the sent messages - but this information is the minimum possible, and (ii) our security does not come at the cost of throughput experience for the receivers, but at the cost of bandwidth connecting the sources.

If the in-degree of a node is smaller or equal to the out-degree, our requirement is by default satisfied; the cases that are interesting are when a node has high in-degree and low out-degree. Motivated from this observation, in this paper we started examining combination networks, where the nodes that perform coding have in-degree up to h and out-degree one. For the class of combination networks, it turns out that there exist schemes, described in this paper, that solve our problem by applying the basic principle of one-time pad encryption. Note that these schemes provide unconditional, information-theoretic guarantees bounding the amount of information learned by intermediate nodes.

Our contributions are:

- We provide sufficient and necessary conditions for a family of networks (which we refer to as canonical combination networks) and exactly characterize the associated cost.
- We provide sufficient conditions for a more general family of networks and upper bounds on the associated cost.

The paper is organized as follows. Section II formalizes our problem statement; Section III solves as an example the butterfly network; Section IV looks at canonical combination networks; Section V provides algorithms and bounds for general combination networks and finally Section VI concludes the paper with a short discussion.

G. Maatouk's research was supported by Grant 228021-ECCSciEng of the European Research Council (ERC). V.M. Prabhakaran's research was funded in part by the ERC Project NOWIRE ERC-2009-StG-240317 and a Ramanujan Fellowship from the Department of Science and Technology, Government of India. C. Fragouli's research was funded by the ERC Project NOWIRE ERC-2009-StG-240317.

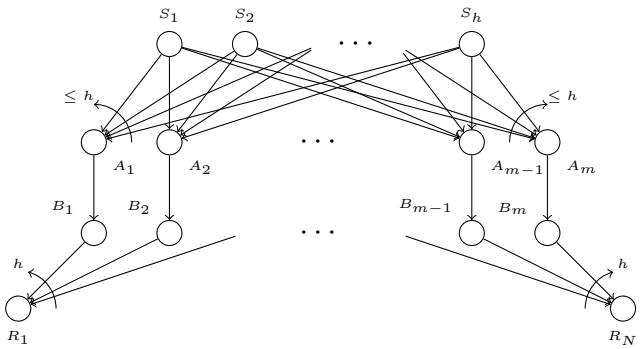


Fig. 1: A combination network with h sources, m bottleneck edges (A_i, B_i), and $N \leq \binom{m}{h}$ receivers, each receiver observing a distinct subset of h B-nodes.

II. Problem statement

We consider a directed acyclic graph $G = (V, E)$ with unit capacity edges, with h not-colocated unit rate sources $S_1, \ldots, S_h$, N receivers, and untrusted intermediate network nodes. Each source S_i wishes to multicast its message X_i to all receivers. We assume that the min-cut to each receiver is h, and that G is a minimal network, in the sense that removing any edge reduces the min-cut for at least one receiver. We also assume that a valid network code over a field of size 2^ℓ, that allows each receiver to decode the h sources, has been designed using any of the methods in the literature [7]. We have at our disposal an auxiliary network H that can be used to provide alternative connections for the source nodes. We want to minimize the number of edges in H that we use, while allowing each intermediate node to learn at most as many linear combinations of the source messages as its out-degree. We assume that intermediate nodes do not collaborate.

In this paper we will focus our attention exclusively on combination networks that we describe below; our goal is to formulate necessary and sufficient conditions to ensure secrecy at every coding point in the combination network, with the additional constraint that no extra network resources should be used, and no decrease in throughput is allowed.

Combination Network: A combination network consists of h unit-rate source nodes, m intermediary nodes or *coding points* $A_1, \ldots, A_m$ controlling m bottleneck links (A_i, B_i), and a maximum of $\binom{m}{h}$ receivers $R_1, \ldots, R_N$, each observing h independent linear combinations of the source messages on a distinct subset of h intermediary nodes B_i, as shown in Fig. 1. Note that coding points are not necessarily connected to all sources.

Definition 1: We describe as *canonical* the set of combination networks that have the following properties: (i) h coding points, say $A_1, \ldots, A_h$, have in-degree 1 and the only incoming edge for coding point A_i comes from source S_i, (ii) $m-h$ coding points are connected to all h sources, and (iii) we have all possible $N = \binom{m}{h}$ receivers.

Note that canonical combination networks are minimal.

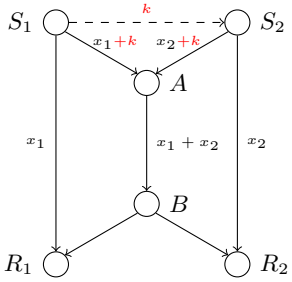


Fig. 2: A secure scheme for the butterfly network.

Security Requirement: The following definition formalizes the secrecy requirement that was described in the introduction.

Definition 2: Assume a network that employs a linear network code over a field $\mathbf{F}_q$, with $q = 2^\ell$. Let $v \in V$ be an intermediary node (i.e., not a source or a receiver node) with in-degree d_1 and out-degree d_2, where the incoming edges are carrying messages $Y_1, \ldots, Y_{d_1}$. We say the secrecy requirement at v is met if

$$I(X_1, \ldots, X_h; Y_1, \ldots, Y_{d_1}) \leq d_2 \log q.$$

Apart from enforcing secrecy at intermediate nodes, we also require that source node S_i should not learn any information about the message X_j of a source S_j, for $i \neq j$.

III. The butterfly network

Here we derive necessary and sufficient conditions for our secrecy requirements to be met over the butterfly network[1] in Fig. 2. Although this network is extremely simple, we will see that the core ideas that allow us to characterize secrecy for more general networks are already at play here.

In Fig. 2, with the standard solution, node A receives values X_1 and X_2 from the two sources and XORs them to produce the value $X_1 + X_2$, which it sends on its outgoing edge. Thus node A learns two independent linear combinations of the source messages, namely, X_1 and X_2, and produces one linear combination of the source messages.

A. Sufficient condition for secrecy: We here claim that it is sufficient to connect the sources with a unit rate edge of arbitrary direction. Indeed, assume this edge points from S_1 to S_2. S_1 can generate a unit rate random key k and share it with S_2. Both sources XOR k to their outgoing messages X_1 and X_2 before sending these to A. The one-time pad k is independent from both X_1 and X_2; thus, the incoming edges of A carry, seperately, no information about either source message. However, by XORing the messages they carry, A will still produce the desired linear combination $X_1 + X_2$.

B. Necessary condition for secrecy: We now show that the sufficient condition for secrecy derived above is optimal.

Assume there exist two unit-rate directed edges between the sources S_1 and S_2. The following lemma provides a lower

[1]Note that the butterfly network is a special case of a canonical combination network, where $h = 2$ and $m = 3$, and where we have replaced two "trivial" coding points by direct links (S_i, R_i).

bound on the communication required between the sources if the secrecy requirement at node A is to be met.

Lemma 1: In the butterfly network, secrecy at the (unique nontrivial) coding point can be ensured iff the directed edges connecting the sources have a sum-rate of at least unity.

Proof: The if-part follows easily from the sufficiency condition[2]. To see the only-if-part, for any pair of nodes V, W in the butterfly network of Fig. 2, denote by Y_{VW} the message transmitted on the edge between V and W. Let $r_i = 1$ be the rate of transmission of source S_i and r_{ij} the rate of the directed link from source S_i to source S_j. The security requirement in this case amounts to

$$I(X_1X_2; Y_{S_1,A}, Y_{S_2,A}) \leq 1. \tag{1}$$

Cut-set bounds for decodability give us

$$r_1 \leq r_{1,2} + I(X_1; Y_{S_1,A}) \tag{2}$$

$$r_2 \leq r_{2,1} + I(X_2; Y_{S_2,A}), \tag{3}$$

where (2) follows from a cut which separates nodes S_1 and R_1 from nodes S_2, A, B, and R_2, and similarly, (3) follows from a cut which separates nodes S_2 and R_2 from nodes S_1, A, B, and R_1. From these we get

$$\begin{aligned} r_1 + r_2 - (r_{1,2} + r_{2,1}) &\leq I(X_1; Y_{S_1,A}) + I(X_2; Y_{S_2,A}) \\ &\leq I(X_1; Y_{S_1,A}, X_2) + I(X_2; Y_{S_2,A}) \\ &= I(X_1; Y_{S_1,A}|X_2) + I(X_2; Y_{S_2,A}) \\ &\leq I(X_1; Y_{S_1,A}, Y_{S_2,A}|X_2) \\ &\quad + I(X_2; Y_{S_1,A}, Y_{S_2,A}) \\ &= I(X_1, X_2; Y_{S_1,A}, Y_{S_2,A}) \leq 1, \end{aligned}$$

where the last inequality follows from (1). This gives us $r_{1,2} + r_{2,1} \geq 1$, since the sources transmit at unit-rate. ∎

IV. Canonical combination networks

In this section, we build on the butterfly network example to derive matching necessary and sufficient conditions for canonical combination networks (see definition 1).

A. Sufficient conditions for secrecy (and an algorithm)

Theorem 1 gives a sufficient condition to ensure secrecy at all coding points in the canonical combination network. We prove this theorem constructively.

Theorem 1: In the canonical combination network, the secrecy requirement can be enforced at all coding points if the auxiliary network contains a unit-rate tree connecting the sources (the edges of the tree may have arbitrary directions).

Proof: At the coding points of in-degree 1, the secrecy requirement is trivially met. Consider a coding point A of in-degree h and suppose there exists a unit-rate directed tree $\mathcal{T}$ connecting the sources.

For edge $e = (S_i, S_j) \in \mathcal{T}$, call S_i the "tail" of the edge and S_j its "head". Assume that the linear combination that A must send on its outgoing edge is $\sum_{i=1}^{h} a_i X_i$, where the a_i are scalars over $\mathbf{F}_{2^\ell}$. The secure coding scheme works as follows:

1) Over each edge $e \in \mathcal{T}$, the tail generates a key k_e and sends it to the head. Keys for different edges are independent.
2) Node S_i sends to A the value $a_i X_i + \sum_{e:S_i \in e} k_e$, i.e., it sends its scaled message $a_i X_i$ added to all keys it sees on its neighboring (whether incoming or outgoing) edges. Let us call the sum of all keys seen by a node its *node key*.
3) Node A receives $\{a_i X_i + \sum_{e:S_i \in e} k_e\}_{i=1}^{h}$, i.e., h linear combinations; it adds them and sends the result to B.

To see that the scheme actually produces the desired linear combination on the outgoing edge of A, note that what A computes is the sum

$$\sum_{i=1}^{h} \left(a_i X_i + \sum_{e:S_i \in e} k_e \right) = \sum_{i=1}^{h} a_i X_i + \left(\sum_{i=1}^{h} \sum_{e:S_i \in e} k_e \right),$$

where the summation $\sum_{i=1}^{h} \sum_{e:S_i \in e} k_e$ performs in fact a double counting of the tree edges.

We have restricted our attention to one coding point A, but it is easy to see that unless the intermediate nodes collude, the same keys can be used for ensuring secrecy with respect to every intermediate node at no additional transmission cost. ∎

Example 1: Fig. 3a shows the application of the tree scheme in a simple combination network.

B. Necessary condition for secrecy

Theorem 2 gives a lower bound on the rate of information exchange that must take place among the sources.

Theorem 2: A necessary condition for secrecy at all points of the canonical combination network is that, in the auxiliary network H that connects the sources, the (undirected) min-cut separating every pair of sources in H has at least unit value.

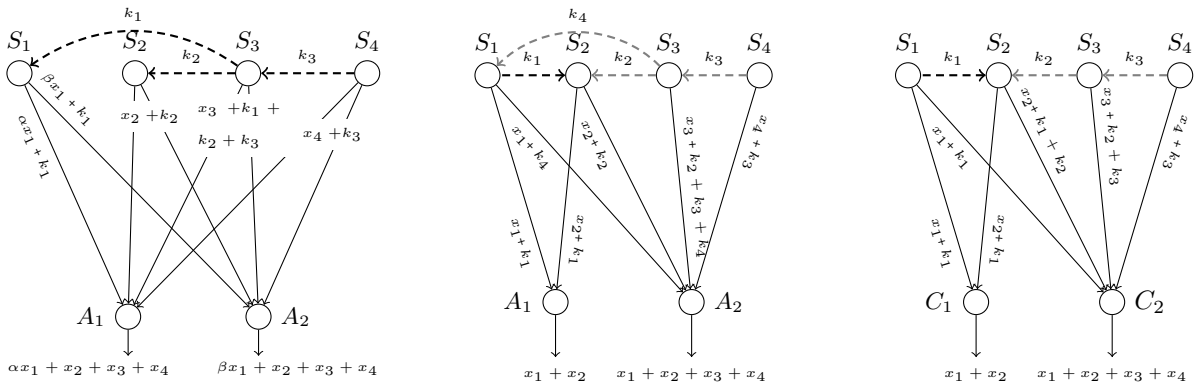
Proof: Suppose to the contrary that there is a cut which partitions the sources into M_1 and M_2 such that the min-cut is less that unity, i.e.,

$$\sum_{(i,j):S_i \in M_1,\ S_j \in M_2} (r_{ij} + r_{ji}) < 1. \tag{4}$$

We may assume without loss of generality that $S_1 \in M_1$ and $S_2 \in M_2$. Let $A_1, \ldots, A_h$ be the h coding points of in-degree 1, each of them controlling the bottleneck link (A_i, B_i) and let A be any coding point of in-degree h controlling the bottleneck link (A, B). We will prove that secrecy at A cannot hold if all receivers decode correctly, thereby leading to a contradiction.

Recall that each of the $\binom{m}{h}$ receivers is connected to a distinct subset of h second-layer intermediary nodes. In particular, consider the receiver that observes the set $\{B_1, B_3, \ldots, B_h, B\}$ (where B_2 does not appear): call this receiver R_1. Also consider the receiver that observes the set $\{B_2, B_3, \ldots, B_h, B\}$ (where B_1 does not appear): call this

[2]We may need to consider an appropriately large blocklength so that the links between the sources can carry integral rate keys. Such details are omitted in the sequel.



(a) A secure scheme for a canonical combination network.

(b) General combination network: disjoint trees.

(c) General combination network: overlapping trees.

Fig. 3: The tree scheme for canonical and general combination networks. In each case, a subset of the coding points is depicted.

receiver R_2. Now we may view the network connecting the sources and the receivers R_1, R_2 as follows: we have two (composite) sources, namely, M_1 and M_2. There are $|M_1|-1$ unit rate links from source M_1 and $|M_2|-1$ unit rate links from source M_2 to receivers R_1 and R_2 (given by the links through $B_3, \ldots, B_h$). There is a unit rate link from M_1 to R_1 and similarly another from M_2 to R_2 (corresponding to B_1 and B_2, respectively). Finally, there is a coding point A with a unit rate edge to B. Let $Y_{M_1,A}$ and $Y_{M_2,A}$ denote, respectively, what the composite sources M_1 and M_2 send to A. Proceeding with the cut-set bounds for decodability at the receivers as in the proof of Lemma 1, we can write

$$|M_1| \leq (|M_1|-1) + \sum_{(i,j):S_i\in M_1,\ S_j\in M_2} r_{ij} + I(M_1; Y_{M_1,A})$$

$$|M_2| \leq (|M_2|-1) + \sum_{(i,j):S_i\in M_1,\ S_j\in M_2} r_{ji} + I(M_2; Y_{M_2,A}),$$

Proceeding as before we get

$$2 - \sum_{(i,j):S_i\in M_1,\ S_j\in M_2} (r_{ij}+r_{ji}) \leq I(S_1,S_2; Y_{S_1,A}, Y_{S_2,A}).$$

But, substituting from (4), we have that the left hand side is strictly greater than one. This violates the secrecy requirement, a contradiction. ∎

Actually, it is easy to verify that the necessary condition in the statement of Theorem 2 is also a sufficient condition. The tree scheme of section IV-A is thus one of many possible schemes that ensure secrecy at all coding points, with the additional property that it minimizes the number of edges of H. Note that an additional attractive feature of the tree scheme is that in a connected network, one can always find a spanning tree in polynomial time.

V. General combination networks

We now discuss sufficient conditions for general combination networks, show how these reduce to a combinatorial problem, and propose a heuristic algorithm for its solution[3].

A. Sufficient conditions as a combinatorial problem

We saw in the previous section that a sufficient condition for canonical networks is to use the auxiliary network H to create a tree (of arbitrary orientation). This tree in canonical networks is used *only once*, although we may have an arbitrary number m of (non-colluding) coding points; intuitively, the tree allows each source to have a "node key" such that the keys of all sources sum to zero, while jointly, the h keys have entropy $h-1$. The same source keys can be used for all coding points since the coding points are not colluding.

In non-canonical combination networks, what changes is that coding points may be connected to any subset of the h sources. A sufficient condition in this case would also be that, if a coding point i is connected to a subset $C_i \subseteq [1:h]$ of the h sources (call it its *parent*) with $|C_i| = k$, say sources $S_1, \ldots, S_k$, these specific k sources are connected in H with a unit rate tree $\mathcal{T}_i$. Indeed, if such a tree exists, we can apply the same algorithm as in the previous section, and create keys so that the node keys of sources $S_1, \ldots, S_k$ sum to zero. However, now the same source keys can only be used for the coding points that have common parents. If our network has multiple coding points j, and the parents of different coding points are different subsets of the source nodes, each such subset needs to be connected directly with a tree T_j, for our desings to apply. Note that the trees T_j do not need to be disjoint; in fact, if we want to minimize the use of resources, we should select the trees T_j so that jointly they use the minimum number of edges, as Example 2 illustrates.

[3]Note that the combinatorial problem defined in this section does not provide necessary conditions for secrecy over general combination networks. Indeed, simple counterexamples show that for some network configurations, the optimal solution to the combinatorial problem is strictly suboptimal in terms of of the number-of-edges/rate-of-information-exchange of the auxiliary network.

Example 2: Consider the network in Fig. 3b and two coding points, the first connected to the subset $C_1 = \{S_1, S_2\}$ of the sources and the second to the set C_2 of all sources. To apply our coding scheme, we can connect the source nodes as in Fig. 3b in two trees, and use the coding scheme shown in the same figure. Alternatively, we can use the more efficient scheme in Fig. 3c: in this second scheme we use edge (S_1, S_2) for both trees.

Note that what interests us now is no longer the number m of coding points, but rather the number n of *types* of coding points, that is, the number of distinct subsets of the sources, of cardinality larger than 1, seen by various coding points[4]. For example, for the canonical combination network, $n = 1$. Typically, n could be much smaller than h, which motivates expressing the minimal connectivity requirements at the sources in terms of n. Thus, our sufficient condition is reduced to the following combinatorial problem.

Combinatorial formulation: Given h nodes and n sets $C_i \subseteq [1:h]$, create a graph H that connects the h nodes with a set of edges E_H of cardinality as small as possible, so that the induced subgraph on each set C_i contains a spanning tree.

This problem can be visualized with the use of Venn diagrams as the following example illustrates.

Example 3: Fig. 4 depicts the case of $n = 3$ sets and 8 source nodes; the constructed graph H has the property that each induced subgraph on C_i contains a spanning tree. Here, connecting the sources through a single tree is not sufficient.

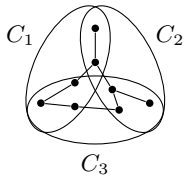


Fig. 4: Venn diagram representation of our requirements. Dots represent sources, and sets C_i correspond to subsets of sources characterizing types of coding points.

B. A method to construct H

As seen above, we may partition the set of h sources into *part*s of the form

$$P_I = \left(\bigcap_{i\in I} C_i\right) \bigcap \left(\bigcap_{j\notin I} \bar{C}_j\right), \text{ where } I \subseteq [1:n].$$

Definition 3: A non-empty part P_I is said to be *active*.

It is clear that for a given valid H, without increasing the number of edges, we may replace the induced subgraph on an active part with any spanning tree on the nodes of that part. Thus, without loss of generality, when convenient we may treat each active part as containing only a single node which stands in for a spanning tree on the nodes of the part.

[4]For coding points seeing exactly one source, the secrecy requirement is trivially satisfied.

Definition 4: The *level* of a part P_I is the cardinality of I.

Definition 5: An active part P_I is said to *lead* an active P_J if $I \supseteq J$. In this case, P_J is said to *follow* P_I.

Definition 6: An active P_I is called a *leader* if it does not follow any other active part. Otherwise, it is called a *follower*.

The following steps will produce a *valid* graph H:

1) Classify active parts into leaders and followers.
2) Form a spanning tree inside each active part.
3) Add an edge between the tree of each follower and the tree of one of the leaders it follows.
4) Connect the trees of the leaders such that all pairs of leaders P_I, P_J with $I \cap J = \{i_1, \ldots, i_k\}$ non-empty have, for each $j = 1, \ldots, k$, a path between them in the subgraph induced by C_j. I.e., we add edges between some pairs of leaders such that the leaders in each set C_j are connected in the subgraph induced by C_j.

Note that while the first step is unambiguous, there are several potential choices for the next three. We already argued that the choices made in step two do not affect the number of edges in the resulting H. The same holds for step three[5]. The choices made in the last step may affect the quality of H produced; we will give upper bounds on the number of edges in Section V-C.

C. Upper bound on the needed number of edges

We will use the lemma below to derive an upper bound on the number of edges produced by the above algorithm.

Lemma 2: The number of leaders is at most $\binom{n}{\lfloor n/2 \rfloor}$.

Proof: Let $N_k = \binom{n}{k}$ be the number of parts of level k. The proof follows from arguing that the largest number of leaders results when all parts of level $\lfloor n/2 \rfloor$ (the level with maximal size) are the only leaders.

If there are, say, m leaders in a level k, we can argue that this will prevent at least m parts from being leaders in a level p, where p is such that $N_p \geq N_k$. Suppose $k > p$. Consider the graph where the parts are the vertices and an edge exists between P_I and P_J if $I \subset J$ or $J \subset I$. The number of edges between a part at level k and the parts at level p is $\binom{k}{p}$, and there are N_k parts at level k, giving rise to $N_k\binom{k}{p}$ edges between the two levels. By symmetry, each part at level p has $N_k\binom{k}{p}/N_p$ edges from level k. The m leaders at level k are responsible for $m\binom{k}{p}$ edges to level p. Hence, the number of parts at level p who are ruled out from being leaders is at least

$$\frac{m\binom{k}{p}}{N_k\binom{k}{p}/N_p} = m\frac{N_p}{N_k} \geq m.$$

[5]The argument is as follows: given any valid H, without increasing the number of edges we may perform the following operations and still keep the graph valid. (i) Every follower who does not have an edge to a leader it follows may have one of its edges removed and replaced by an edge to one of the leaders it follows (note that being a follower implies that it has at least one edge). (ii) Every follower P_I who has an edge to a leader P_J it follows and has another edge to some part P_L (P_L could be another leader it follows) may have this second edge removed and replaced by an edge between P_J and P_L. Note that after these two steps every follower has only one edge which has at the other end one of the leaders it follows. (iii) Now if a follower has more than one leader, then we may remove this single edge and replace it with another edge to any one of the leaders it follows.

A similar counting argument holds for $k < p$.

Thus, if the only active parts fall in two levels, say, levels k_1 and k_2, the number of leaders is at most $\max(N_{k_1}, N_{k_2})$. Now, suppose the active levels are k_1, k_2, k_3 with $N_{k_1} \leq N_{k_2} \leq N_{k_3}$. Suppose there are m_1 leaders in level k_1 and m_2 in level k_2. Then, the m_1 leaders in level k_1 will preclude at least m_1 parts in level k_2 from being leaders. These m_1 parts in level k_2 along with the m_2 leaders in the same level will preclude at least $m_1 + m_2$ parts from being leaders in level k_3. Thus, the largest number of leaders possible is $N_{k_3} = \max(N_{k_1}, N_{k_2}, N_{k_3})$. This clearly extends to any number of active levels and completes the proof. ■

Clearly, a trivial upper bound on the number of edges needed in the graph H is $\binom{h}{2}$, i.e., if H is the complete graph on the h source nodes. For the cases where $n < h$, we can improve on this upper bound using the steps in Section V-B.

Theorem 3: The number of edges in an optimal valid graph H is not larger than

$$h + (n-1)\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor} - n.$$

Proof: We will follow our algorithm. Let h source nodes belong to l leaders and f followers. Then the total number of active parts is $f + l$. Let h_i denote the number of source nodes in part P_i. In step 2, where we create a spanning tree in each part, we add $\sum_{i=1}^{f+l} h_i - 1 = h - (f + l)$ edges. Then, in step 3, we connect each follower to any one of the leaders it follows. As a result, the total number of edges becomes $h - (f + l) + f = h - l$. Now, in step 4, edges are added so that leaders in each set C_i are connected to each other in the subgraph induced by C_i. At worst, this can be done by connecting the leaders of each set C_i, without sharing any edge between different sets. Let l_i represent the number of leaders in set C_i. Thus, in step 4, we will have at most $\sum_{i=1}^{n}(l_i - 1)$ edges. But, $l \geq l_i$ for all $i \in 1, 2, ..., n$ and thus, $l \geq \max(l_i)$. Using these facts, the number of edges in an optimal valid graph H is at most

$$h - l + \sum_{i=1}^{n}(l_i - 1) \leq h + \sum_{i: l_i \neq \max(l_i)}^{n} l_i - n.$$

But l_i is at most the largest number of leaders P_I with $i \in I$, and by Lemma 2,

$$l_i \leq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}.$$

Thus, an upper bound on the number of edges of H is

$$h + (n-1)\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor} - n.$$

■

Note that for the canonical combination network case, where $n = 1$, this bound is tight since it reduces to $h - 1$. Similarly, the bound reduces to $h - 1$ for $n = 2$, so that we know that a tree is sufficient for combination networks with two types of coding points of in-degree larger than 1.

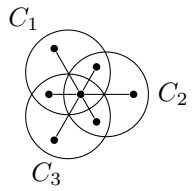


Fig. 5: Tree in the case where $n = 3$ and $P_{[1:3]}$ is active.

The following lemma gives an optimal solution for the case where some coding point sees all h sources.

Lemma 3: If part $P_{[1:n]} = \bigcap_{i=1...n} C_i$ is active, then it is sufficient to use a single tree on the h sources nodes.

Proof: Clearly $P_{[1:n]}$ is the only leader and all other parts are followers, thus step 4 is trivial and for step 3 it suffices to connect with a single edge the tree of each active follower with the tree in $P_{[1:n]}$. Fig. 5 shows an example for $n = 3$. ■

VI. Conclusions and Discussion

In this paper we started exploring a new definition of security relevant for networks that perform linear network coding: can we allow each intermediate node to only learn as many linear combinations as it needs to forward, although it may have a much larger incoming degree? We showed that this is possible for combination networks by providing an auxiliary network H that connects the sources; we derived sufficient and necessary conditions for canonical networks, and showed that for general combination networks, the sufficient conditions reduce to a combinatorial problem.

Combination networks have the special property that coding points directly receive input from the source nodes and not other coding points; it is easy to find examples of networks that do not meet this condition, and where we cannot meet our security conditions by only connecting the source nodes with an auxiliary network H. To deal with such networks, we may need to allow the auxiliary network H to also connect *coding points* inside the network (the source nodes can be viewed as a special class of coding points), or to sacrifice some of the min-cut rate; this is part of our future research.

References

[1] N. Cai and R. W. Yeung, "Secure network coding," IEEE International Symposium on Information Theory (ISIT), p. 323, 2002.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," IEEE Transactions on Information Theory, vol. 46, pp. 1204-1216, Jul. 2000.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai., "Linear network coding," IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371-381, Feb. 2003.

[4] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, "On the capacity of secure network coding," Allerton Conference on Communication, Control, and Computing, Sep. 2004.

[5] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks ii," IEEE International Symposium on Information Theory (ISIT), pp. 551-555, Jun. 2007.

[6] 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, Feb. 2011.

[7] C. Fragouli and E. Soljanin, "Network coding fundamentals," Foundations and Trends in Networking, vol. 2, no. 1, pp. 1133, 2007.