An Equivalence Between Secure Network and Index Coding

Lawrence Ong†, Badri N. Vellambi†, Jörg Kliewer†, and Phee Lep Yeoh§
†The University of Newcastle, Australia; ‡New Jersey Institute of Technology, USA; §University of Sydney, Australia

Abstract—We extend the equivalence between network coding and index coding by Effros, El Rouayheb, and Langberg to the secure communication setting in the presence of an eavesdropper. Specifically, we show that the most general versions of secure network-coding setup by Chan and Grant and the secure index-coding setup by Dan, Skachek, and Chee, which also include the randomised encoding setting, are equivalent.

I. INTRODUCTION

Recently, equivalence results in information theory and network coding have been of significant interest in the community. Such reduction results uniquely map one communication problem to another equivalent problem that is potentially easier to study than the original problem. Some of the equivalence results already established include those between instances of multiple-unicast network coding and those of (a) multiple-multicast network coding [1], (b) secure network coding [2], and (c) index coding [3].

In particular, the latter result addresses the equivalence between network coding [4] and index coding [5] in the non-secure setting. Non-secure network coding and index coding with randomised encoding. Specifically, we show that the most general versions of secure network coding [6] and index coding [7] have been studied in the literature so far, an equivalence between these coding approaches for the secure setting does not trivially apply to the secure setting. In particular, we pointed out [11] that equating the eavesdrop per index coding does not trivially apply to the secure setting. In

II. PROBLEM DEFINITION AND NOTATION

For any positive integer n, let [n] = {1, . . . , n}. For a set ξ = {i1, . . . , i|ξ|}, let Xξ = [Xi1, . . . , Xi|ξ|] with an arbitrary but fixed order. The tail and head of an edge (u, v) ∈ E in a directed graph G = (V, E) refer to vertices u, and v, respectively, i.e., u = tail(e) and v = head(e). For a node v ∈ V, we let in(v) be the set of all edges with head(v) = v; similarly, out(v) denotes the set of all edges with tail(e) = v.

A. Secure network coding

1) Network-coding instances: We follow Chan and Grant’s secure network-coding definition [12]. It includes Bhattacharjee and Narayan’s weakly-secure network-coding definition [9] and Cai and Yeung’s strongly-secure network-coding definition [7] as special cases. A secure network-coding instance, denoted by ξ = (G, M, W), is defined as follows:

• G = (V, E) is an acyclic graph with vertex set V and edge set E. Each edge e ∈ E has a capacity given by c_e.

• M = (S, O, D) is the connection requirement. The set S is the collection of source-message indices, where the source messages ξ = {Xs : s ∈ S} are mutually independent and are each distributed on [2^R,s]. For some positive integer n that can be chosen to suit the design of network codes. Here, R_s denotes the rate of the message Xs, s ∈ S. The source-location mapping O : S → V specifies the originating node O(s) for the source message Xs. The destination-location mapping D : S → 2^V specifies the nodes D(s) that require the message Xs.

• W = ((A_r, B_r) : r ∈ R) defines the eavesdropping pattern for |R| eavesdroppers. Each eavesdropper r ∈ R observes the set of links B_r ⊆ E and tries to reconstruct a subset of source messages indexed by A_r ⊆ S, i.e., X_{A_r}. We assume that vertices with no incoming links are originating nodes for some source messages, and vertices with no outgoing links are destinations for some source messages.

2) Deterministic network codes: Given (G, M), a network code (F, G) consists of a collection of encoding functions for the edges F = {f_e : e ∈ E}, and decoding functions for the vertices G = {g_u : u ∈ V} satisfying the following:

This work is supported by ARC grants FT140100219, DE140100420, and DP150100903, and US NSF grants CNS-1526547 and CCF-1439465.
The local encoding function $f_e$ for edge $e$ takes in random variables associated with $\text{in}(\text{tail}(e))$ and source messages originating at node $\text{tail}(e)$, and outputs a random variable associated with link $e$, denoted by $X_e \in [2^{|e|}]$.

Given that $G$ is acyclic, each edge message $X_e$ can be written as a function of source messages originating from its predecessors, denoted by $f_e$. This is known as the global encoding function, and it can be recursively calculated by considering the topology of the graph (using (i) $f_e = f_e$ if $\text{tail}(e)$ has no incoming links, and (ii) $f_e = f_e(f_{e_1}, f_{e_2}, \ldots, f_{e_n})$, where $\{e_1, e_2, \ldots, e_n\} = \text{in}(\text{tail}(e))$.

The decoding function $g_u$ for a node $u \in \mathcal{V}$ takes in random variables associated with links $\text{in}(u)$ and source messages originating at node $u$, and outputs $X_{\{s \in \mathcal{S} : u \in D(s)\}}$. In this paper, we only consider zero-error decoding.

3) Randomised network codes: A network code is said to be randomised if there exists an edge function $f_e$ that is not a deterministic function of the random variables associated with $\text{in}(\text{tail}(e))$ and source messages originating at node $\text{tail}(e)$.

Any randomised function can be generated by an independent random variable $Z_u$ at each node $u \in \mathcal{V}$, where $Z_u$ takes values in an alphabet with size $\prod_{e \in \text{out}(u)} 2^{|e|}$, where out$(u)$ is defined as the set of all outgoing edges from node $u$. These independent random variables are often referred to as random keys.

A randomised network code $(\mathcal{F}, \mathcal{G})$ is similar to a deterministic network code $(\mathcal{F}, \mathcal{G})$, except that each edge encoding function $f'_e$ is a function of (i) random variables associated with $\text{in}(\text{tail}(e))$, (ii) source messages originating at node $\text{tail}(e)$, and (iii) the random key $Z_{\text{tail}(e)}$.

4) Secure network codes: A deterministic or randomised network code $(\mathcal{F}, \mathcal{G})$ for $(G, M)$ is said to be secure against an eavesdropping pattern $\mathcal{W}$ if each eavesdropper $\hat{r}$ gains no information about $\mathcal{X}_A\hat{r}$ it attempts to reconstruct after observing $\mathcal{X}_{B\hat{r}}$ on the links it has access to, i.e.,

$$H(\mathcal{X}_{A\hat{r}} | \mathcal{X}_{B\hat{r}}) = H(\mathcal{X}_{A\hat{r}}), \quad \hat{r} \in \mathcal{R}.$$  

In other words, $(\mathcal{F}, \mathcal{G})$ is a secure network code for the secure network-coding instance $I$.

5) Secure network-coding rates: The network-coding instance is said to be $(\mathcal{R}_{\mathcal{S}}, n)$-feasible if and only if there exists at least one secure network code with the associated source-message rates and block size $n$.

B. Secure index coding

1) Secure index-coding instances: We follow Dau, Skachek, and Chee’s secure index-coding definition [10]. A secure index-coding instance, denoted by $I = (\mathcal{S}, \mathcal{T}, \{\mathcal{W}_i\}, \{\mathcal{H}_i\}, \mathcal{W})$, is defined as follows:

- $\mathcal{S} = [k]$ is the set of indices of $k$ source messages available at a sender.
- The messages $\{\mathcal{X}_s : s \in \mathcal{S}\}$ are mutually independent and for $s \in \mathcal{S}, \mathcal{X}_s$ is distributed on $[2^{R_s}]$, for some non-negative message rate $R_s$ and positive integer $n$ that can be chosen to suit the design of index codes.
- $\mathcal{T} = [\ell]$ is the collection of $\ell$ receiver indices.
- $\mathcal{W}_i$ is the set of the indices of the messages required by receiver $i \in T$.
- $\mathcal{H}_i$ is the set of indices of the messages known a priori to receiver $i \in T$.
- $\mathcal{W} = (\{A_i, B_i\} : \hat{r} \in \hat{\mathcal{R}})$ is the eavesdropping pattern. Each eavesdropper $\hat{r} \in \mathcal{R}$ has access to the codeword broadcast by the sender and a subset of the messages $\mathcal{X}_{B\hat{r}}$, and tries to reconstruct $\mathcal{X}_{A\hat{r}}$, where $A_i, B_i \subseteq \mathcal{S}$.

2) Deterministic index codes: A deterministic index code $(\mathcal{F}, \mathcal{G}) = (\{\hat{y}_i : i \in T\})$ consists of an encoding function by the sender $\mathcal{F} = \mathcal{f}$ which takes in the random variables $\mathcal{X}_\mathcal{S}$ and outputs a random variable $\mathcal{X}_\mathcal{B} \in [2^{R_n}]$, where $\mathcal{R}_n$ is the broadcast rate, and $n$ is the block size of the code. It also consists of a decoding function $\hat{y}_i$ for receiver $i$, which takes in the sender’s codeword $\mathcal{X}_\mathcal{S}$ and its prior messages $\mathcal{X}_{\hat{y}_i}$ and outputs the messages $\mathcal{X}_{\mathcal{W}_i}$ it requires. Similar to the network-coding setup, we only consider zero-error decoding.

3) Randomised index codes: A randomised index code $(\mathcal{F}, \mathcal{G})$ is defined similar to the deterministic index codes except that the sender’s encoding function takes in an independent random key $Z \in [2^{R_n}]$ of some positive rate $\mathcal{R}_n$ in addition to $\mathcal{X}_\mathcal{S}$. Unlike the model by Mojahedian, Aref, and Gohari [8], the randomness allowed in the encoding in our setting is only known to the sender, and is not shared between the sender and the receivers.  

4) Secure index codes: A deterministic or randomised index code $(\mathcal{F}, \mathcal{G})$ is said to be secure against the eavesdropping pattern $\mathcal{W}$ if no eavesdropper $\hat{r}$ gains no information about the message set $\mathcal{X}_{A\hat{r}}$, it tries to reconstruct by observing the sender’s codeword $\mathcal{X}_\mathcal{B}$ and its side information $\mathcal{X}_{\hat{y}_i}$, i.e.,

$$H(\mathcal{X}_{A\hat{r}} | \mathcal{X}_\mathcal{B}, \mathcal{X}_{\hat{y}_i}) = H(\mathcal{X}_{A\hat{r}}), \quad \hat{r} \in \hat{\mathcal{R}}.$$  

Clearly, $\mathcal{A}_\hat{r} \cap \mathcal{B}_\hat{r} = \emptyset$. Specifically, we say that $(\mathcal{F}, \mathcal{G})$ is a secure index code for the secure index-coding instance $I$.

5) Secure index-coding rate: The secure index-coding instance is said to be $(\mathcal{R}_{\mathcal{S}}, n)$-feasible if and only if there exists at least one secure index code with the associated source-message rates $\mathcal{R}_n$, broadcast rate $\mathcal{R}_n$, and block size $n$.

III. MAPPING FROM SECURE INDEX CODING TO SECURE NETWORK CODING

Given an index instance $I = (\mathcal{S}, \mathcal{T}, \{\mathcal{W}_i\}, \{\mathcal{H}_i\}, \mathcal{W})$ of a secure-index-coding problem, where $\mathcal{S} = [k]$ and $\mathcal{T} = [\ell]$, we construct an equivalent secure network-coding instance $\mathcal{I} = (G, M, W)$ using the following rule:

**Index-to-network coding mapping:**

- The graph $G = (\mathcal{V}, \mathcal{E})$ consists of $k + \ell + 2$ vertices labelled as $\mathcal{V} = \{s_1, s_2, \ldots, s_k, t_0, t_1, t_2, \ldots, t_\ell, 1, 2\}$. For each $i \in \mathcal{S}$, vertex $s_i$ has an outgoing link to vertex 1 and to each vertex in $\{t_i : i \in H_i\}$. Each of these links from vertex $s_i$ are of capacity $2^{R_s}$. Vertex 1 has a link of capacity $2^{R_n}$ to vertex 2 and to each vertex in $\{t_i : i \in T\}$.
- The connection requirement $M$ consists of the following: $\mathcal{S} = \mathcal{X}_\mathcal{S}$, and $\mathcal{X}_\mathcal{S}$ has the same distribution as $\mathcal{X}_\mathcal{S}$, which implies $R_t = R_n$. For each message $\mathcal{X}_i, i \in \mathcal{S}$, the source
locations are \(O(i) = s_i\), i.e., the message \(X_i\) originates at vertex \(s_i\), and is destined for \(D(i) = \{t_j : i \in V_i\}\).
- The eavesdropping pattern \(W\) is defined as \(R = \hat{R}, B_r = \{(1 \rightarrow 2), \{\text{out}(s_i) : i \in \hat{B}_r\}\}\), and \(A_r = A_r\).

Note that by construction, for each \(i \in \hat{T}\),
- \(\hat{V}_i = \{j \in S : t_i \in D(j)\}\).
- \(\hat{H}_i = \{j \in S : (s_j \rightarrow t_i) \in E\}\).
- The vertices in \(V \setminus \{t_1, \ldots, t_\ell\}\) are not the destinations of any source message.

Figure 1 depicts an example of such a mapping. With the above construction, we now state an equivalence between these two instances:

**Theorem 1:** Let \(\hat{I}\) and \(\hat{c}_b\) be a secure index-coding instance and a broadcast rate, respectively. Let \(I\) be the corresponding secure network-coding instance using the index-to-network coding mapping. For any \(\hat{R}_\ell\) and \(\hat{c}_b\), the instance \(\hat{I}\) is \((\hat{R}_\ell, \hat{c}_b, n)\)-feasible if and only if \(I\) is \((R_\ell, n)\)-feasible.

**Proof of Theorem 1:**

\[
\hat{I}\ 	ext{is} \ (\hat{R}_\ell, \hat{c}_b, n)\text{-feasible } \Rightarrow \ I\ 	ext{is} \ (R_\ell, n)\text{-feasible}:
\]

Let \((\hat{F}, \hat{G})\) be a secure index code that supports \((\hat{R}_\ell, \hat{c}_b, n)\).

Then,

\[
H(X_{\hat{V}_i} | X_{\hat{b}}, X_{\hat{H}_i}) = 0, \quad \text{for all} \ i \in \hat{T},
\]

\[
H(X_{\hat{A}_i} | X_{\hat{b}}, X_{\hat{B}_i}) = H(X_{\hat{A}_i}), \quad \text{for all} \ i \in \\hat{R},
\]

where \(X_{\hat{b}} = \hat{f}(X_S, \hat{Z})\), where \(\hat{Z} \in [2^{k_n}]\) is a random key independent of the source messages (to account for randomised index coding), and

\[
X_{\hat{V}_i} = \hat{g}(\hat{f}(X_S, \hat{Z}), X_{\hat{H}_i}), \quad \text{for all} \ i \in \hat{T}.
\]

Now, we construct a secure network code as follows:

- Set \(f_e = X_{\text{out}(e)}\) for each outgoing edge \(e\) from each vertex in \(\{s_i : i \in S\}\). This is possible since vertex \(s_i\) is the originating vertex for the message \(X_i\).
- Set \(f_{1,2} = f_e = \hat{f}(X_S, Z_1) \in [2^{k_n}]\) for all \(e \in \text{out}(2)\), where \(Z_1\) is independent of all the source messages \(X_S\) and has the same distribution as \(\hat{Z}\).
- Set \(g_{1,2} = \hat{g}_i\) for all \(i \in \hat{T}\), and \(g_u = 0\) for all other vertices.

For each vertex \(t_i, i \in \hat{T}\), all incoming edges in\(t_i\) originate from vertex 2 and vertices \(\{s_j : j \in H_i\}\). The message on edge \((2, t_i)\) is \(\hat{f}(X_S, Z_1)\), and that on edge \((s_j, t_i)\) is \(f(s_j, t_i) = X_{s_j}\). This means vertex \(t_i, i \in \hat{T}\), can decode

\[
g_{t_i}(\hat{f}(X_S, Z_1), X_{\hat{H}_i}) = \hat{g}_i(\hat{f}(X_S, Z_1), X_{\hat{H}_i})
\]

\[
= X_{\hat{V}_i} = X_{\{j \in S : t_i \in D(j)\}}.
\]

Here, \((6b)\) follows from \((5)\) as \(\hat{Z}\) and \(Z_1\) have the same distribution and both are independent of the respective source messages. Noting that the rest of the vertices are not the destination for any source message, the network code satisfies the decoding requirements of \(I\).

Each eavesdropper \(r \in R = \hat{R}\) has access to messages on the edge set \(\hat{B}_r\) consisting of

- edge \((1, 2)\), which carries \(\hat{f}(X_S, Z)\), and
- edges \(\{\text{out}(s_i) : i \in \hat{B}_r\}\), which carry messages \(X_{\hat{b}}\).

Now,

\[
H(X_{A_r} | X_{\hat{B}_r}) = H(X_{\hat{A}_r} | \hat{f}(X_S, Z), X_{\hat{B}_r})
\]

\[
= H(X_{\hat{A}_r}) = H(X_{A_r}),
\]

where \((7b)\) follows from \((6b)\) with a change of variables (from non-hatted to hatted).

This completes the proof that the network code is also secure against all the eavesdroppers described by \(W\).

\(I\) is \((R_\ell, n)\)-feasible \(\Rightarrow \hat{I}\) is \((\hat{R}_\ell, \hat{c}_b, n)\)-feasible:

Let \((\hat{F}, \hat{G})\) be a secure network code that supports \((\hat{R}_\ell, \hat{c}_b, n)\).

Then,

\[
H(X_{\{s \in S : u \in D(s)\}} | X_{\text{in}(u)}, X_{O^{-1}(u)}) = 0, \quad u \in V,
\]

and

\[
H(X_{A_r} | X_{\hat{B}_r}) = H(X_{A_r} | \{f_e : e \in \hat{B}_r\})
\]

\[
= H(X_{\hat{A}_r}), \quad r \in R.
\]

By the construction of \(I\), we know that \(\{s \in S : u \in D(s)\} = \emptyset\), for any \(u \in \{s_1, s_2, \ldots, s_k, 1, 2\}\), and \(\{s \in S : t_i \in D(s)\} = \hat{W}_i\) for \(i \in \{1, 2, \ldots, \ell\}\). Also, none of the vertices in \(\{t_1, t_2, \ldots, t_\ell\}\) is the originating node for any source message.
where (17b) follows from (13) with a change of variables 

\[
\hat{X}_{\{2,t\}} = f_{2\rightarrow t}(\hat{f}_{\{2,t\}}(X_S, Z), Z_2) 
\]

for \( i \in [\ell] \), where \( \text{in}(t_i) = \{(2, t_i), t_i, t_2, \ldots, t_L\} \), \( L = |H| \) and \( X_{\{2,t\}} = f_{2\rightarrow t}(\hat{f}_{\{2,t\}}(X_S, Z), Z_2) \) 

where \( H_i = \{h_1, \ldots, h_L\}, Z = \{Z_1, \ldots, Z_L\}, Z_2 \) is the collection of all \( \{Z_i\} \) generated by nodes \( \{1, s_1, \ldots, s_{|S|}\} \), and \( Z_2 \) is independent of all messages and \( Z \).

Decoding at all \( t_i \)'s must succeed for any realisation of \( \{Z_i\} \).

Hence, (10) must also hold when all \( Z_{s_i} = 0 \). This gives 

\[
H(X_{W_i})|X'_\{2,t\}, X'_1, \ldots, X'_L) = 0, 
\]

where \( X'_{\{2,t\}} = f_{2\rightarrow t}(\hat{f}_{\{2,t\}}(X_S, Z), Z_2) \) 

\[
X'_{\{t_1, \ldots, t_L\}} = [f_{\{s_1,t_1\}}(X_{h_1}, 0), \ldots, f_{\{s_L,t_L\}}(X_{h_L}, 0)]. 
\]

by setting \( Z_{s_i} = 0 \) for all vertices \( s_i, i = 1, \ldots, k \).

Now, we construct a secure index code as follows where \( \hat{Z} \in 2^{2^{\alpha_m}} \) is a random variable independent of the messages, and has the same distribution as \( Z_1 \).

Now, for each receiver \( i \in T \), 

\[
H(X_{W_i},X_b) = H(X_{W_i}X_h), f_{\{s_1,t_1\}}(\hat{X}_{h_1}, 0), \ldots, f_{\{s_L,t_L\}}(\hat{X}_{h_L}, 0) = 0. 
\]

where (17b) follows from (13) with a change of variables (from hatted to non-hatted), and (17b) follows from the following Markov chains 

\[
\hat{X}_{\{2,t\}} - (\hat{X}_b, \hat{X}_h), f_{\{s_1,t_1\}}(\hat{X}_{h_1}, 0), \ldots, f_{\{s_L,t_L\}}(\hat{X}_{h_L}, 0) - f_{2\rightarrow t}(\hat{X}_b, \hat{Z}_2), 
\]

as \( Z_2 \) is independent of all other random variables, and has the same distribution as \( Z_2 \).

Since conditional entropy is non-negative, it follows from (17b) that each receiver \( i \in T \) can decode the messages \( X_{W_i} \) that it requires, given \( (\hat{X}_b, \hat{X}_h) \).

We now prove the security constraints in \( \hat{I} \). First, consider \( I \). If \( X_i \) is requested by some nodes, then observing all outgoing links from \( s_j \) must enable one to reconstruct \( X_i \). If \( X_i \) is not requested by any node, we assume that observing all outgoing links from \( s_j \) also enables one to reconstruct \( X_i \). The rationale behind this assumption is as follows: If \( R_j = R' \) is in the feasible region, then all \( R_j > R' \) are also in the feasible region by having node \( s_j \) transmitting the first \( nR' \) bits of \( X_j \). Hence, we only need to consider the smallest feasible rate for \( X_j \), denoted by \( R_{\min} \), when all the other rates are kept fixed. Now, if after observing all outgoing links from \( s_j \), one can obtain only \( nR''_j \) bits of information of \( X_j \) (where \( R''_j < R_{\min} \)), then node \( s_j \) could have transmitted \( X_j \) at rate \( R''_j \), which contradicts that \( R_{\min} \) is the smallest feasible rate for \( X_j \).

So, from (9), we have 

\[
H(X_{A_i}) = H(X_{A_i}|X_e : e \in B_c) = (19a) 
\]

\[
H(X_{A_i}) = H(X_{A_i}|f_{\{1,2\}}(X_S, Z_1), X_{\{out(s_i)\} \in B_c} = (19b) 
\]

\[
H(X_{A_i}) = H(X_{A_i}|f_{\{1,2\}}(X_S, Z_1), X_{\{out(s_i)\} \in B_c}, X_{B_c}) \leq (19c) 
\]

\[
H(X_{A_i}) = H(X_{A_i}) \leq (19d) 
\]

where (19c) is derived because observing all outgoing links from \( s_i \) allows one to reconstruct \( X_i \). Thus, it follows that (19d) and (19e) must hold with equality.

Now, consider an eavesdropper \( \hat{r} \in \hat{R} \) in the index-coding equivalence \( \hat{I} \).

\[
H(X_{A_i}|X_b, X_{B_c}) = H(X_{A_i}|f_{\{1,2\}}(X_S, Z), X_{B_c}) = H(X_{A_i}|\hat{X}_h) = H(X_{A_i}), 
\]

where (20a) follows from (19a)–(19e) with a change of variables (from non-hatted to hatted), by noting that \( (X_S, Z) \) and \( (X_S, Z_1) \) have the same distribution.

So, \( \hat{X}_h \) is a secure index code for \( \hat{I} \).

IV. MAPPING FROM SECURE NETWORK CODING TO SECURE INDEX CODING

Given a secure network-coding instance I = (G, M, W), we first construct an augmented secure network-coding instance with deterministic encoding, and then construct an equivalent secure index-coding instance \( I = (S, T, \{W_i\}, \{H_i\}, W) \).

Augmented secure network coding: We construct an augmented secure network-coding instance \( I' = (G', M', W') \) as follows:

- \( G' = (V', E') \subseteq (V, E), \) and \( c'_e = c_e \) for all \( e \in E' \). The vertices, the edges, and the edge capacities remain the same.

- Let \( S = [S] \), where \( S \triangleq |S| \). The connection requirement is augmented as follows: \( S' = S \cup \{S + 1, S + 2, \ldots, S + |V'|\} \), where we introduce an additional source \( X_{S+v} \in \{x^{2^{\alpha_m}}\} \triangleq [2^{k,v}], \) originating at each vertex \( v \in \{V\} \), that takes the role of and has the same distribution as the random key \( Z_e \) used in the randomised encoding at vertex \( v \) in I. So, \( O(S + v) = v \), i.e., \( X_{S+v} \) originates at vertex \( v \), and we define \( D'(S + v) = \emptyset \), i.e., \( X_{S+v} \) is not requested by any vertex. For \( s \in S, X_S \) has the same distribution as \( X_S \), \( O'(s) = O(s) \), and \( D'(s) = D(s) \). Note that \( R'_e = R_e \) for all \( s \in [S] \), and \( R'_{S+v} = k_v \) for all \( v \in [V] \).
Any deterministic or randomised (i.e., using an independent random key $Z_v$ at vertex $v$) secure network code for $I$ is equivalent to a deterministic secure network code for $I'$, where each vertex $v$ gets an additional source $X'_{S+v}$ that is not required to be decoded by any node.

Denote the set of vertices in $I'$ that are destinations for some source messages by $\mathcal{T}': \{ j \in \mathcal{V}' : j \in \mathcal{D}'(i) \text{ for some } i \in S' \}$. Note that $\mathcal{O}'(\cdot)$ can map different source indices to one vertex, and hence, $O'^{-1}(j)$ returns a set of indices of messages originating at vertex $j$.

Network-to-index coding mapping:

- $\tilde{S} = S' \cup \mathcal{E}'$. It consists of one source message $\tilde{X}_s$ for each $s \in S'$ in $I'$, and one $\tilde{X}_e$ for each edge $e \in \mathcal{E}'$ in $I'$. $\tilde{X}_{S'}$ has the same distribution as $X_{S'}$. The rates of the messages are $\hat{R}_s = R'_s$ and $\hat{R}_e = c'_e$.
- $\tilde{\mathcal{T}}' = \{ \tilde{t}_{\tilde{e}} \in \mathcal{E}' \cup \{ \tilde{t}_s \} \}_{e \in \mathcal{E}'}$. This means $\hat{I}$ has $|\mathcal{T}'| + |\mathcal{E}'|$ receivers, one for each destination node in $I'$ and one for each edge in $I'$.

- For each $\tilde{t}_e \in \tilde{\mathcal{T}}$, $\hat{H}_{\tilde{t}_e} = \text{in}(\text{tail}(e)) \cup O'^{-1}(\text{tail}(e))$, and $\hat{W}_{\tilde{t}_e} = \{ e \}$.
- For each $\tilde{t}_s \in \tilde{\mathcal{T}}$, $\hat{H}_{\tilde{t}_s} = \text{in}(e) \cup O^{-1}(e)$, and $\hat{W}_{\tilde{t}_s} = \{ s \in \mathcal{S} : i \in \mathcal{D}(s) \}$.
- The eavesdropper setting $W'$: $\tilde{\mathcal{R}} = \mathcal{R}'$. For each $\tilde{r} \in \tilde{\mathcal{R}}$, $\tilde{B}_e = B'_e$, and $\tilde{A}_e = A'_e$.
- We set the broadcast set $\tilde{\mathcal{A}}$ as $\hat{\mathcal{A}} = \sum_{e \in E'} e'_e$.

Figure 2 depicts an example of such a mapping.

Remark 1: This network-to-index coding mapping is slightly different from that of Effros et al. [3], since we do not require the use of an additional receiver $\tilde{t}_e$ for the corresponding index-coding instance. We will show that omitting this receiver will not affect the equivalence.

Note that unlike the index-to-network mapping, here $X'_{S'}$ and $X'_{E'}$ have different distributions, whereas the latter are functions of $X_{S'}$. For the corresponding secure index-coding instance, we choose $\tilde{X}_e : e \in \mathcal{E}'$ to be mutually independent, of all other messages, and each $\tilde{X}_s$ is uniformly distributed over $[2^{R_s}]$. We will see that using uniformly distributed $\tilde{X}_s$ is the key to ensuring security.

With the above conversion, we now state an equivalence between $I$ and $\hat{I}$ through $I'$:

**Theorem 2:** Let $I$ be a secure network-coding instance and $I'$ be its augmented instance. Let $\tilde{I}$ and $\tilde{c}_e$ be the corresponding secure index-coding instance and a broadcast rate, respectively, obtained using the network-to-index coding mapping from $I'$.

For any $\tilde{R}_S$, the instance $I$ is $(\tilde{R}_S, n)$-feasible if and only if the instance $\hat{I}$ is $(\tilde{R}_S, \tilde{c}_e, n)$-feasible.

**Proof of Theorem 2:**

$I$ is $(\tilde{R}_S, n)$-feasible $\Rightarrow$ $I$ is $(\tilde{R}_S, \tilde{c}_e, n)$-feasible:

Note that if $I$ is $(\tilde{R}_S, n)$-feasible if and only if $I'$ is $(\tilde{R}_S, n)$-feasible using deterministic network encoding functions $\{f'_e\}$ derived from $\{f_e\}$ for $I$, where all the randomness $\{Z_e\}$ in the network code for $I$ is realised using $\{X_{S+v}\}$ in $I'$.

Since the network code for $I'$ is deterministic, we use the same code mapping as that proposed by Effros et al. [3]: The sender’s broadcast message is $X_b = [\hat{X}_b(e) : e \in \mathcal{E}]$, where $\hat{X}_b(e) = \tilde{X}_b + f'_e(\tilde{X}_{S'})$.

Note that $\hat{X}_b, f'_e \in [2^{R_{eh}}] = [2^{R_{eh}}]$.

In $I'$, each vertex $\tilde{v} \in \mathcal{T}'$ can decode all messages that it requires from the message on all incoming edges and messages originating at $e$, meaning that

$$X'_{\{s \in S' : v \in \mathcal{D}'(s)\}} = X'_{\{s \in S : v \in \mathcal{D}(s)\}} = g'_v(\hat{X}_{in}(v))^{O^{-1}(v)} = g'_v(\hat{X}_{in}(v))^{O^{-1}(v)} = g'_v([f'_e(\tilde{X}_{S'})])_{e \in \mathcal{E}(in)}(v), X_{O^{-1}(v)}.$$

As mentioned above, while messages $X'_{O^{-1}(e)}$ and $X_{O^{-1}(e)}$ (with node subscripts) have the same distribution, messages $X'_{in(e)}$ and $X_{in(e)}$ (with edge subscripts) may not. To deal with this issue, consider the broadcast message $\hat{X}_b$ in [21], any receiver that knows $\hat{X}_b$ can obtain $f'_e(\hat{X}_b)$, where $g'_v(\hat{X}_{in}(v))^{O^{-1}(v)} = f'_v(\hat{X}_{S'})_{e \in \mathcal{E}(in)}(v)$ have the same distribution. So, with a change of variables (from non-hatted to hatted), receiver $\tilde{t}_e \in \tilde{\mathcal{T}}$ can decode the messages it requires using

$$X_{\tilde{W}_{\tilde{t}_e}} = X_{\{s \in S' : v \in \mathcal{D}'(s)\}} = g'_v([f'_e(\tilde{X}_{S'})])_{e \in \mathcal{E}(in)}(v), X_{O^{-1}(\tilde{t}_e)}.$$
for each destination vertex 

\ \hat{X} \ \}

\left\{ e \in B'_t \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E' \right\}

\left\{ e \in B'_t \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in B'_t \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}

\left\{ e \in E' \right\}

\left\{ e \in E \right\}