Single-Unicast Secure Network Coding and Network Error Correction are as Hard as Multiple-Unicast Network Coding

Wentao Huang, Tracey Ho, *Senior Member, IEEE*, Michael Langberg, *Senior Member, IEEE*,
and Jörg Kliewer, *Senior Member, IEEE*

***Abstract*— This paper reduces multiple-unicast network coding to single-unicast secure network coding and single-unicast network error correction. Specifically, we present reductions that map an arbitrary multiple-unicast network coding instance to a unicast secure network coding instance in which at most one link is eavesdropped, or a unicast network error correction instance in which at most one link is erroneous, such that a rate tuple is achievable in the multiple-unicast network coding instance if and only if a corresponding rate is achievable in the unicast secure network coding instance, or in the unicast network error correction instance. Conversely, we show that an arbitrary unicast secure network coding instance in which at most one link is eavesdropped can be reduced back to a multiple-unicast network coding instance. In addition, we show that the capacity of a unicast network error correction instance in general is not (exactly) achievable.**

***Index Terms*— Network coding, equivalence, security, error correction, capacity.**

I. Introduction

IN THE paradigm of *network coding*, a set of source nodes transmits information to a set of terminal nodes over a network with noiseless links; internal nodes of the network may mix received information before forwarding it. This mixing (or encoding) of information has been extensively studied over the last decade (see e.g., [4]–[8], and references therein). In particular, the problems of determining the *capacity* of the network and designing optimal codes achieving the capacity are well understood under the *multicast* setting, where there is a single source node whose information is demanded by all terminal nodes. However, much less is known regarding the *multiple-unicast* setting where there are multiple source nodes, each of them demanded by a single and different terminal node (the more general setting of *multiple-multicast*, where each terminal node may demand the information from an arbitrary subset of source nodes, can be converted to an equivalent multiple-unicast setting using the constructions in [9], [10]). Determining the capacity or the achievability of a rate tuple in a multiple-unicast network coding instance remains an intriguing, central, open problem, e.g., [11]–[13].

This work connects multiple-unicast network coding to two other fundamental network coding problems, namely secure network coding and network error correction, using the idea of reduction. We focus on the *decision problem* of determining whether a rate tuple is achievable in the network, as well as the decision problem of determining whether a rate tuple is in the capacity region of the network. Here a rate tuple is achievable if there exists a code that exactly achieves it, and the capacity region is the closure of the set of all achievable rate tuples. Note that by definition, a rate tuple that is achievable is always in the capacity region, and a rate tuple that is in the capacity region is always asymptotically achievable but may not be (exactly) achievable. We study these problems using the technique of *reduction*. Loosely speaking, we say a class of decision problems $\mathcal{I}$ can be reduced to a class of decision problems $\mathcal{J}$ if there exists a scheme that maps any problem instance in $\mathcal{I}$ to a corresponding problem instance in $\mathcal{J}$, such that the answers to the two instances are the same. In other words, using such a reduction methodology, one may take an instance of $\mathcal{I}$, map it to a corresponding instance of $\mathcal{J}$, solve the decision problem on the instance of $\mathcal{J}$, and deduce a solution for the original instance of $\mathcal{I}$; implying that if there is a way to solve the problems in $\mathcal{J}$, then there is a way to solve the problems in $\mathcal{I}$.

In this paper we construct a reduction, i.e., a mapping, from the problem of multiple-unicast network coding to the problem of unicast secure network coding. Note that the latter problem is asked under the simplest setting of *unicast*, where there is a single source node and a single terminal node in the network. Therefore under the unicast setting the rate tuple degenerates to a scalar, which is referred to as rate. Our reduction addresses the problem of determining whether

Manuscript received January 14, 2016; revised January 6, 2017; accepted February 28, 2018. Date of publication March 29, 2018; date of current version May 18, 2018. This work was supported in part by NSF under Grant CCF-1526771 and Grant CNS-1526547 and in part by the Caltech Lee Center. This paper was presented in part at the 2013 International Symposium on Network Coding [1], the 2014 Allerton Conference on Communication, Control and Computing [2], and the 2015 IEEE International Symposium on Information Theory [3].

W. Huang was with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA. He is now with Snap Inc., Venice, CA 90291 USA (e-mail: whuang@snap.com).

T. Ho was with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA. He is now with Second Spectrum Inc., Los Angeles, CA 90012 USA (e-mail: tracey@secondspectrum.com).

M. Langberg is with the Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260 USA (email: mikel@buffalo.edu).

J. Kliewer is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (email: jkliewer@njit.edu).

Communicated by C.-C. Wang, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2018.2820686

a rate/rate tuple is achievable, as well as the problem of determining whether a rate/rate tuple is in the capacity region. Secondly, we construct another reduction from the problem of multiple-unicast network coding to the problem of unicast network error correction. Note that again the latter problem is asked under the simplest setting of unicast. Surprisingly, we show that our reduction only works for the problem of determining whether a rate/rate tuple is achievable. In contrast, for the problem of determining whether a rate/rate tuple is in the capacity region, we show that the same reduction mapping no longer guarantees that it does not change the answers to the instances. An interesting consequence of this negative result is that the capacity of a unicast network error correction instance in general is not (exactly) achievable.

Compared to the standard network coding problem, secure network coding and network error correction have additional secrecy and reliability requirements, described in the following subsections.

A. Secure Network Coding

Secure network coding is a natural generalization of network communication to networks with eavesdroppers. Specifically, in the secure network coding problem, a subset $A \in \mathcal{A}$ of links may be eavesdropped, where $\mathcal{A}$ is given and is the collection of all possible eavesdropping patterns. A valid code design for the secure network coding problem needs to ensure the secrecy of the source information. Namely, for any choice of A from $\mathcal{A}$, the mutual information between the set of signals transmitted on the links in A and the source messages must be negligible. The secure network coding problem is well studied in the literature and in particular is well understood in the multicast setting under the assumption that 1) all links have equal capacities, and 2) $\mathcal{A}$ is uniform, i.e., $\mathcal{A}$ includes all subsets of links of size z_w, where z_w is the number of wiretapped links, and 3) only the source node can generate randomness, e.g., [14]–[17]. In the cases that either link capacities are not equal, or $\mathcal{A}$ is arbitrary, or non-source nodes may generate randomness, determining the capacity or the achievability of a rate in a secure network coding instance remains an open problem, e.g., [18]–[22].

This paper shows that an arbitrary multiple-unicast network coding instance $\mathcal{I}$ can be reduced to a particular unicast secure network coding instance $\mathcal{I}_s$ that has a very simple setup. Specifically, the reduction mapping ensures that in $\mathcal{I}_s$: a) there is a single source node and a single terminal node in an acyclic network; b) all links have equal capacity; c) there is a single wiretapped link and this link can be any link in the network, namely, $\mathcal{A}$ is uniform with $z_w = 1$; and d) non-source nodes are allowed to generate randomness. The setup of $\mathcal{I}_s$ is simple in the sense that setup a) is the simplest connection requirement, b) is the simplest assumption on link capacities, and c) gives the simplest structure of a non-trivial $\mathcal{A}$. Indeed, under the setup of a) - c) the secure capacity of the network is characterized by the cut-set bound and is achieved by linear codes [14]. In this sense, our reduction suggests that the addition of setup d) is critical; as the secure network coding problem is simple and well understood under setting a) - c), but under setting a) - d) we show that it is as hard as the long standing open problem of multiple-unicast network coding. We remark that allowing non-source nodes to generate randomness is realistic and preferable because this can significantly increase the capacity of the network [21], [23].

Our reduction addresses the problem of determining the achievability of a rate/rate tuple, as well as the problem of determining whether a rate/rate tuple is in the capacity region. Furthermore, our reduction holds for different types of security requirements. Namely, in $\mathcal{I}_s$ we may assume either perfect, strong, or weak security.

Our reduction has an operational aspect. Namely, in our reduction, from a code for $\mathcal{I}_s$ one can construct a code for $\mathcal{I}$. Thus, using our reduction, to solve an instance of the multiple-unicast network coding problem, one may first reduce it to an instance of the unicast secure network coding problem, then solve the latter, and finally use this solution to obtain a solution to the original multiple-unicast problem. We conclude, speaking loosely, that unicast secure network coding under the simple setting described above is at least as hard as multiple-unicast network coding. Our formal results are given in Theorem 1 of Section III.

The hardness of general secure network coding problems are previously studied in [18] and [21]. Specifically, Chan and Grant [18] show that determining the zero-error achievability of a rate in the multicast secure network coding problem with general setup (i.e., arbitrary edge capacities, arbitrary $\mathcal{A}$, and arbitrary nodes may generate randomness) and with perfect security is at least as hard as determining the zero-error achievability of a rate tuple in the multiple-multicast network coding problem. Cui *et al.* [21] show that determining the capacity of a unicast secure network coding problem is NP-hard if either the edge capacities are arbitrary or $\mathcal{A}$ is arbitrary. Our work strengthens the result in [18] by showing that the secure network coding problem under an extremely simple setup (i.e., unicast, equal link capacities, uniform $\mathcal{A}$ with a single wiretap link) is still hard, under various definitions of achievability and security.

B. Network Error Correction

We now turn to the *network error correction* problem, which is a natural generalization of network communication to networks with *adversarial* errors. Specifically, in the network error correction problem a subset $B \in \mathcal{B}$ of links may be erroneous, where $\mathcal{B}$ is given and is the collection of all possible link error patterns. A valid code design for the error correction problem needs to ensure reliable communication between the sources and terminals in the *worst case* no matter which set $B \in \mathcal{B}$ of links are erroneous or which error has been imposed on these links. Namely, in this context, the communication of a message is successful if for any choice of B from $\mathcal{B}$, and for any (error) signals to be transmitted on the links in B, the message is correctly decoded at the terminal node. The network error correction problem is extensively studied and in particular is well understood under the multicast setting with the assumption that 1) all links have equal capacities, and 2) $\mathcal{B}$ is uniform, i.e., $\mathcal{B}$ includes all subsets of links of size z_e, where z_e is the number of erroneous links, e.g., [24]–[29]. In the cases that either link capacities are not equal or $\mathcal{B}$ is arbitrary, determining the capacity of the network or the achievability of a rate remains an open problem, e.g., [30]–[34].

In a flavor similar to the reduction described in Section I.A, we show that an arbitrary multiple-unicast network coding instance $\mathcal{I}$ can be reduced to a particular unicast network error correction instance $\mathcal{I}_c$ that has a very simple setup. Specifically, the reduction mapping ensures that in $\mathcal{I}_c$: a) there is a single source node and a single terminal node in an acyclic network; b) all links have equal capacity; c) there is a single error link; and d) the error link can be any link in the network except a given subset of (well protected) links. The setup of $\mathcal{I}_c$ is simple in the sense that setup a) is the simplest connection requirement, b) is the simplest assumption on link capacities and c) gives the smallest number of error links. Indeed, if the error link can be any one link in the network (namely if $\mathcal{B}$ is uniform), then under the setup of a) - c) the capacity of the network is characterized by the cut-set bounds and is achieved by linear codes [24]. In this sense, our reduction suggests that the addition of setup d), which will result in a non-uniform $\mathcal{B}$, is critical; as a network error correction problem is simple and well understood under setting a) - c), but under setting a) - d) we show that it is as hard as the long standing open problem of multiple-unicast network coding.

Our reduction addresses the problem of determining the achievability of a rate/rate tuple. Namely we show, using our reduction, that one can determine the achievability of a rate tuple in $\mathcal{I}$ by determining the achievability of a corresponding rate in $\mathcal{I}_c$. However, rather interestingly, the same reduction (i.e., the same mapping) does not address the problem of determining whether a rate/rate tuple is in the *capacity* region. Specifically, we show that there exists a multiple-unicast network coding instance $\mathcal{I}$ and a rate tuple, such that the rate tuple is not in the capacity region of $\mathcal{I}$, and that after using our reduction mapping, the corresponding rate is in the capacity region of the corresponding unicast network error correction instance $\mathcal{I}_c$. The fact that our reduction "works" when considering (exact) achievability and does not "work" when considering capacity, implies that in general the capacity of a unicast network error correction instance is not (exactly) achievable, which is a result of separate interest.

Similar to the previous discussion, our reduction has an operational aspect that from a code for $\mathcal{I}_c$ one can construct a code for $\mathcal{I}$. Indeed, using our reduction, to solve an instance of the multiple-unicast network coding problem, one may first reduce it to an instance of the unicast network error correction problem, then solve the latter, and finally use this solution to obtain a solution to the original problem. We conclude, speaking loosely, that unicast network error correction under the simple setting described above is at least as hard as multiple-unicast network coding. Our formal results are given in Theorems 3, 4 and 5 of Section V.

C. Equivalence via Reverse Reduction

The above constructions reduce instances of the multiple-unicast network coding problem to instances of the unicast secure network coding or the unicast network error correction problem. A natural and intriguing question is whether these problems are equivalent, namely, whether it is possible to construct the reverse reductions. Using the technique of A-enhanced networks [35], we show that an arbitrary unicast secure network coding instance in which at most one link is eavesdropped (i.e., $\mathcal{A}$ includes only singleton sets) can be reduced to a multiple-unicast network coding instance, thus implying an equivalence between the two problems. For more complicated $\mathcal{A}$ whether a reverse reduction exists or not remains an open problem. Similarly, the existence of a reverse reduction from unicast network error correction to multiple-unicast network coding remains open.

The paper is organized as follows. In Section II we introduce the models and definitions. Section III reduces multiple-unicast network coding to unicast secure network coding. The reverse reduction is discussed in Section IV. In Section V we reduce multiple-unicast network coding to unicast network error correction. Section VI concludes the paper.

II. Models and Definitions

A. Multiple-Unicast Network Coding

A network is a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices represent network nodes and edges represent links. Each edge $e \in \mathcal{E}$ has a capacity c_e, which is the number of bits that can be transmitted on e in one transmission. An instance $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{T}, B)$ of the *multiple-unicast network coding problem*, includes a network $\mathcal{G}$, a set of source nodes $\mathcal{S} \subset \mathcal{V}$, a set of terminal nodes $\mathcal{T} \subset \mathcal{V}$ and an $|\mathcal{S}|$ by $|\mathcal{T}|$ connection requirement matrix B. The (i, j)-th entry of B equals 1 if terminal j requires the information from source i and equals 0 otherwise. B is assumed to be a permutation matrix so that each source is paired with a single terminal. Denote by $s(t)$ the source required by terminal t. Denote $[n] \triangleq \{1, .., n\}$. Each source $s \in \mathcal{S}$ is associated with an independent message, represented by a random variable M_s uniformly distributed over $[2^{nR_s}]$. A *network code* of length n is a set of encoding functions ϕ_e for every $e \in \mathcal{E}$ and a set of decoding functions ϕ_t for each $t \in \mathcal{T}$. For each $e = (u, v)$, the encoding function ϕ_e is a function taking as input the signals received from the incoming edges of node u, as well as the random variable M_u if $u \in \mathcal{S}$. ϕ_e evaluates to a value in $\{0, 1\}^{nc_e}$, which is the signal transmitted on e. For each $t \in \mathcal{T}$, the decoding function ϕ_t maps the tuple of signals received from the incoming edges of t, to an estimated message $\hat{M}_{s(t)}$ with values in $[2^{nR_{s(t)}}]$.

A network code $\{\phi_e, \phi_t\}_{e\in\mathcal{E}, t\in\mathcal{T}}$ is said to *satisfy* a terminal t under transmission $(m_s, s \in \mathcal{S})$ if $\hat{M}_{s(t)} = m_{s(t)}$ when $(M_s, s \in \mathcal{S}) = (m_s, s \in \mathcal{S})$, namely, terminal t decodes correctly when the message tuple takes the specific value $(m_s, s \in \mathcal{S})$. A network code is said to satisfy the multiple-unicast network coding problem $\mathcal{I}$ with error probability ϵ if the probability that all $t \in \mathcal{T}$ are simultaneously satisfied is at least $1 - \epsilon$. The probability is taken over the joint distribution on $(M_s, s \in \mathcal{S})$. Formally, the network code satisfies $\mathcal{I}$ with error probability ϵ if

$$\Pr_{(M_s, s\in\mathcal{S})} \left\{ \bigcap_{t\in\mathcal{T}} t \text{ is satisfied under } (M_s, s \in \mathcal{S}) \right\} \geq 1 - \epsilon.$$

For an instance $\mathcal{I}$ of the multiple-unicast network coding problem, the rate tuple $(R_s, s \in \mathcal{S})$ is said to be *achievable* if for any $\epsilon > 0$, there exists a network code that satisfies $\mathcal{I}$ with error probability at most ϵ. $(R_s, s \in \mathcal{S})$ is said to be *achievable with zero error* if there exists a network code that satisfies $\mathcal{I}$ with zero error probability. $(R_s, s \in \mathcal{S})$ is said to be *asymptotically achievable* if for any $\delta > 0$, rate tuple $((1-\delta)R_s, s \in \mathcal{S})$ is achievable. Loosely speaking, under our definition, zero-error achievability does not allow any slackness in neither the probability of error nor rate; achievability allows slackness in the probability of error but not in rate; and asymptotic achievability allows slackness in both the probability of error and rate. We remark that asymptotic achievability is the most commonly used definition in the literatures. However, in addition to asymptotic achievability, extra efforts will be placed on exact achievability with vanishing error in Section V because interestingly, the reduction therein allows slackness in the probability of error but does not allow slackness in rate.

Without loss of generality, in the remaining part of the paper we assume that all entries in the rate tuple are unit, i.e., $R_s = 1$, $\forall s \in \mathcal{S}$, because a varying rate source s can be modeled by multiple unit rate sources co-located at s. We say that unit rate is achievable, achievable with zero error, or asymptotically achievable if $R_s = 1$, $\forall s \in \mathcal{S}$ and $(R_s, s \in \mathcal{S})$ is achievable, achievable with zero error, or asymptotically achievable, respectively.

B. Unicast Secure Network Coding

An instance $\mathcal{I}_s = (\mathcal{G}, s, t, \mathcal{A})$ of the *unicast secure network coding problem* includes a network $\mathcal{G}$, a source node s, a terminal node t and a collection of subsets of links $\mathcal{A} \subset 2^{\mathcal{E}}$ susceptible to eavesdropping. Each node $i \in \mathcal{V}$ generates an independent random variable K_i.[1] The source node holds a rate-R_s secret message M uniformly distributed over $[2^{nR_s}]$. A (secure) network code of length n is a set of encoding functions ϕ_e for every $e \in \mathcal{E}$ and a decoding function ϕ_t. For each $e = (u, v)$, the encoding function ϕ_e is a function taking as input the locally generated randomness K_u, the signals received from the incoming edges of node u, and the message M if $u = s$. ϕ_e evaluates to a value in $\{0, 1\}^{nc_e}$, which is the signal transmitted on e. The decoding function ϕ_t maps the tuple of signals received from the incoming edges of t, to an estimated message $\hat{M}$ with values in $[2^{nR_s}]$.

A secure network code $\{\phi_e, \phi_t\}_{e \in \mathcal{E}}$ is said to *satisfy* instance $\mathcal{I}_s$ with error probability ϵ if the probability that $M = \hat{M}$ is at least $1-\epsilon$, where the probability is taken over the distribution on M and K_i, $i \in \mathcal{V}$. For any edge $e \in \mathcal{E}$, denote by $X_i(e)$ the signal transmitted on e during the i-th channel use. For a subset of edges A, denote by $X^n(A) = (X_i(e) : 1 \leq i \leq n, e \in A)$. The network code is said to satisfy the *perfect security* requirement if for all $A \in \mathcal{A}$, $I(M; X^n(A)) = 0$; the *strong security* requirement if for all $A \in \mathcal{A}$, $I(M; X^n(A)) \to 0$ as $n \to \infty$; and the *weak security* requirement if $\forall A \in \mathcal{A}$, $\frac{I(M; X^n(A))}{n} \to 0$ as $n \to \infty$.

For a unicast secure network coding problem $\mathcal{I}_s$, rate R_s is said to be *achievable* with perfect, strong, or weak security if for any $\epsilon > 0$, there exist network codes that satisfy $\mathcal{I}_s$ with error probability at most ϵ and the corresponding security requirement. Rate R_s is said to be *achievable with zero error* and with perfect, strong, or weak security if there exists network codes that satisfy $\mathcal{I}_s$ with zero error probability and the corresponding security requirement. Rate R_s is said to be *asymptotically achievable* with perfect, strong, or weak security if for any $\delta > 0$, rate $(1-\delta)R_s$ is achievable with the corresponding security requirement. The capacity of $\mathcal{I}_s$ under perfect, strong, or weak security is the supremum over all rates that are asymptotically achievable with the corresponding security requirement.

C. Unicast Network Error Correction

An instance $\mathcal{I}_c = (\mathcal{G}, s, t, \mathcal{B})$ of the *unicast network error correction problem* includes a network $\mathcal{G}$, a source node s, a terminal node t and a collection of subsets of links $\mathcal{B} \subset 2^{\mathcal{E}}$ susceptible to errors. An error occurs in a link if the output of the link is different from the input. More precisely, the output of a link e is the addition of the input signal and an error signal $r_e \in \{0, 1\}^{nc_e}$, and an error occurs in link e if and only if r_e is not the zero vector.[2] For a subset B of links, a B-error is said to occur if errors occur in every link in B. The source node holds a rate-R_c message M uniformly distributed over $[2^{nR_c}]$, and the decoder of the terminal outputs an estimated message $\hat{M}$.

Let $\boldsymbol{r} = (r_e)_{e \in \mathcal{E}}$ be the tuple of error signals, called an error pattern. Denote by $\mathcal{R}_{\mathcal{B}}$ the set of all possible error patterns, i.e., $\mathcal{R}_{\mathcal{B}} = \{\boldsymbol{r} :$ non-zero entries in $\boldsymbol{r}$ correspond to B-errors, $B \in \mathcal{B}\}$. A network code $\{\phi_e, \phi_t\}_{e \in \mathcal{E}}$, defined similarly as in Section II-A, is said to *satisfy* $\mathcal{I}_c$ under transmission m if $\hat{M} = m$ when $M = m$, regardless of the occurrence of any error pattern $\boldsymbol{r} \in \mathcal{R}_{\mathcal{B}}$. A network code is said to satisfy problem $\mathcal{I}_c$ with error probability ϵ if the probability that $\mathcal{I}_c$ is satisfied is at least $1 - \epsilon$. The probability is taken over the distribution on the source message M. Note that our model targets the worst-case (or adversarial) scenario, namely the probability of error is upper bounded by ϵ even in the occurrence of the worst case error pattern.

For a unicast network error correction problem $\mathcal{I}_c$, rate R_c is said to be *achievable* if for any $\epsilon > 0$, there exists a network code that satisfies $\mathcal{I}_c$ with error probability at most ϵ. Rate R_c is said to be *achievable with zero error* if there exists a network code that satisfies $\mathcal{I}_c$ with zero error probability. Rate R_c is said to be *asymptotically achievable* if for any $\delta > 0$, rate $(1 - \delta)R_c$ is achievable. The capacity of $\mathcal{I}_c$ is the supremum over all rates that are asymptotically achievable.

[1] We remark that in the secure network coding problem, allowing non-source nodes to generate randomness can significantly increase the capacity [21], [23] and therefore is preferable. In contrast, for simplicity in the multiple-unicast problem and the network error correction problem we do not assume that non-source nodes can generate randomness.

[2] Note that our model and results can be generalized naturally to arbitrary alphabets in which addition may not be defined, as long as we make the convention that $r_e = 0$ if and only if the output of e is identical to the input.

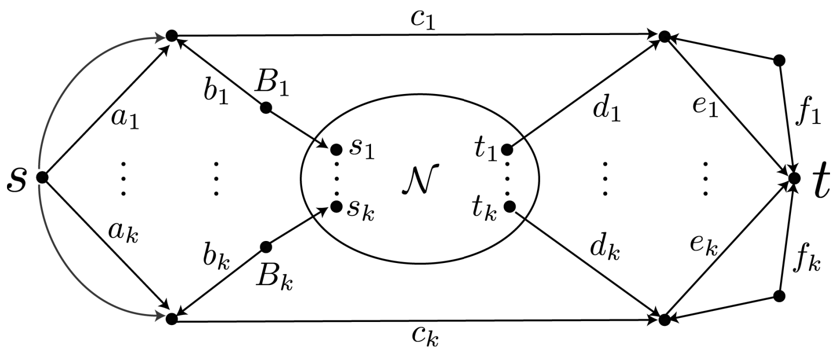


Fig. 1. In the unicast secure network coding problem $\mathcal{I}_s$, the source s communicates with the terminal t. $\mathcal{N}$ is the network on which $\mathcal{I}$ is defined. All links outside $\mathcal{N}$ (i.e., links for which at least one end-point does not belong to $\mathcal{N}$) have unit capacity. The eavesdropper can wiretap on any single link in the network. Namely, $\mathcal{A}$ includes all sets of a single edge. Note that there are k parallel branches in total going from s to t but only the first and the k-th branches are drawn explicitly.

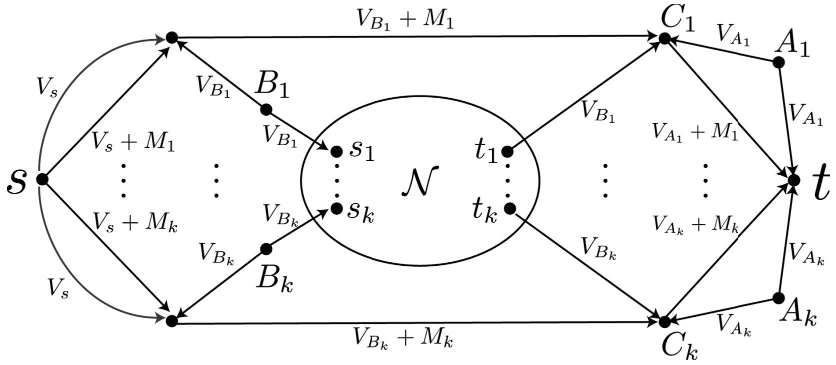


Fig. 2. A scheme of length n that asymptotically achieves rate k in $\mathcal{I}_s$. Fix any $\epsilon > 0$, for node u, let V_u be a length-$(1-\epsilon)n$ random vector generated by node u. The V_u's are independently generated and are uniformly distributed over $\{0,1\}^{(1-\epsilon)n}$. The source message is a k-tuple of i.i.d. uniformly distributed length-$(1-\epsilon)n$ vectors, i.e., $M = (M_i,\ i = 1, \ldots, k)$, where the M_i's are i.i.d. uniformly distributed over $\{0,1\}^{(1-\epsilon)n}$. Since unit rate is asymptotically achievable in $\mathcal{I}$, node t_i obtains $\hat{V}_{B_i}$ such that $\hat{V}_{B_i} = V_{B_i}$ with high probability. $\hat{V}_{B_i}$ is then transmitted to node C_i for key cancellation.

III. Reducing Multiple-Unicast to Secure Unicast

Recall from Section II-A that in a multiple-unicast problem, unit rate is asymptotically achievable if $R_s = 1, \forall s \in \mathcal{S}$ and rate tuple $(R_s, s \in \mathcal{S})$ is asymptotically achievable. The following theorem reduces the problem of determining the asymptotical achievability of unit rate in a general multiple-unicast network coding instance to the problem of determining the asymptotical achievability of a rate in a particular unicast secure network coding instance that has a very simple setup. We remark that although the theorem addresses the achievability of unit rate instead of a general rate tuple in the multiple-unicast network coding problem, this is without loss of generality, because the problem of determining the achievability of an arbitrary rate tuple with rational entries in a multiple-unicast problem can be converted to the problem of determining the achievability of unit rate in a corresponding multiple-unicast problem, by modeling a varying rate source s as multiple unit rate sources co-located at s.

Theorem 1: Given any multiple-unicast network coding problem $\mathcal{I}$ with source-destination pairs $\{(s_i, t_i), i = 1, \ldots, k\}$, a corresponding unicast secure network coding problem $\mathcal{I}_s = (\mathcal{G}, s, t, \mathcal{A})$, in which $\mathcal{A}$ includes all sets of a single edge (i.e., all singletons), can be constructed according to Construction 1, such that unit rate is asymptotically achievable in $\mathcal{I}$ if and only if rate k is asymptotically achievable in $\mathcal{I}_s$ under either perfect, strong, or weak security.

Construction 1: Given any multiple-unicast network coding problem $\mathcal{I}$ on a network $\mathcal{N}$ with source-destination pairs $\{(s_i, t_i), i = 1, \ldots, k\}$, a unicast secure network coding problem $\mathcal{I}_s$ is constructed as specified in Figure 1.

Proof (of Theorem 1): "$\Rightarrow$". In this direction, we show that the asymptotic achievability of unit rate in $\mathcal{I}$ implies the asymptotic achievability of rate k in $\mathcal{I}_s$ under perfect secrecy, which in turn implies the asymptotic achievability of rate k in $\mathcal{I}_s$ under strong and weak secrecy.

The scheme asymptotically achieving rate k is described in Figure 2. Specifically, the rate of the scheme is $(1-\epsilon)k$ if rate $1-\epsilon$ is achievable in $\mathcal{I}$. Let $\epsilon_i = \Pr\{\hat{V}_{B_i} \neq V_{B_i}\}$, then the probability of error in $\mathcal{I}_s$ is upper bounded by $\sum_{i=1}^{k} \epsilon_i$, which can be made arbitrarily small by choosing the ϵ_i's to be small enough. Note that the scheme achieves perfect security, since links in $\mathcal{N}$ are not downstream of s (and therefore the signals transmitted on them are independent of the message), and all other links are one-time padded by uniformly chosen keys.

"$\Leftarrow$". To prove this direction it suffices to show that asymptotic achievability of rate k in $\mathcal{I}_s$ under weak security implies asymptotic achievability of unit rate in $\mathcal{I}$, because asymptotic achievability of rate k in $\mathcal{I}_s$ under perfect or strong security implies asymptotic achievability of the same rate under weak security.

Suppose in $\mathcal{I}_s$ rate k is asymptotically achieved by a code with length n. Let M be the source input message, then $H(M) = kn$. We use the notation of Figure 1. Our objective is lower bound the mutual information between signals $\boldsymbol{b}_i^n$ and $\boldsymbol{d}_i^n$, for $i = 1, \dots, k$. Without loss of generality our analysis will focus on the case of $i = 1$. We start with,

$$\begin{aligned}
&H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(a)}{=} H(M, \boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n) - H(\boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\geq H(M|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n) - H(\boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(b)}{\geq} kn - H(\boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\geq kn - n = (k-1)n, \qquad (1)
\end{aligned}$$

where (a) follows from the chain rule, and (b) follows from our construction which guarantees independence between M and $\{\boldsymbol{d}_1^n,\ \boldsymbol{f}_i^n,\ i = 1, \dots, k\}$. On the other hand,

$$\begin{aligned}
&H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\leq H(M, \boldsymbol{e}_2^n, \dots, \boldsymbol{e}_k^n|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\leq H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n, \boldsymbol{e}_2^n, \dots, \boldsymbol{e}_k^n)\\
&\quad + H(\boldsymbol{e}_2^n, \dots, \boldsymbol{e}_k^n|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(c)}{\leq} n\epsilon_n + H(\boldsymbol{e}_2^n, \dots, \boldsymbol{e}_k^n|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\leq n\epsilon_n + (k-1)n, \qquad (2)
\end{aligned}$$

where $\epsilon_n \to 0$ as $n \to \infty$ and (c) is due to the cut-set $\{\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n, \boldsymbol{e}_2^n, \dots, \boldsymbol{e}_k^n\}$ from s to t and Fano's inequality. We lower bound the entropy of $\boldsymbol{c}_1^n$,

$$\begin{aligned}
H(\boldsymbol{c}_1^n) &\geq H(\boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&= H(M, \boldsymbol{c}_1^n|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\qquad - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\geq H(M|\boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n) - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&= H(M) - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(d)}{\geq} kn - ((k-1)n + n\epsilon_n) = n - n\epsilon_n, \qquad (3)
\end{aligned}$$

where (d) follows from (2). We next lower bound the entropy of $\boldsymbol{d}_1^n$,

$$\begin{aligned}
H(\boldsymbol{d}_1^n) &\geq H(\boldsymbol{d}_1^n|\boldsymbol{c}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&= H(M, \boldsymbol{d}_1^n|\boldsymbol{c}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\qquad - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\geq H(M|\boldsymbol{c}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n) - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(e)}{=} H(M|\boldsymbol{c}_1^n) - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(f)}{\geq} kn - n\delta_n - H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n)\\
&\overset{(g)}{\geq} n - n\epsilon_n - n\delta_n, \qquad (4)
\end{aligned}$$

where $\delta_n \to 0$ as $n \to 0$, (e) follows from the independence between $\{M, \boldsymbol{c}_1^n\}$ and $\{\boldsymbol{f}_i^n,\ i = 1, \dots, k\}$, (f) follows from the weak security requirement, and (g) follows from (2).

By the independence between $\{M, \boldsymbol{c}_1^n, \boldsymbol{d}_1^n\}$ and $\{\boldsymbol{f}_i^n,\ i = 1, \dots, k\}$ we have

$$H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n, \boldsymbol{f}_2^n, \dots, \boldsymbol{f}_k^n) = H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n). \qquad (5)$$

By (1) and (2), the R.H.S of (5) is sandwiched by

$$(k-1)n \leq H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n) \leq n\epsilon_n + (k-1)n. \qquad (6)$$

Now consider the joint entropy of $M,\ \boldsymbol{c}_1^n,\ \boldsymbol{d}_1^n$ and expand it in two ways

$$\begin{aligned}
H(M, \boldsymbol{c}_1^n, \boldsymbol{d}_1^n) &= H(\boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n) + H(M|\boldsymbol{d}_1^n) + H(\boldsymbol{d}_1^n)\\
&= H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n) + H(\boldsymbol{d}_1^n|\boldsymbol{c}_1^n) + H(\boldsymbol{c}_1^n)\\
&\leq (k+1)n + n\epsilon_n,
\end{aligned}$$

where the last inequality holds because of (6) and $H(\boldsymbol{d}_1^n|\boldsymbol{c}_1^n) \leq n,\ H(\boldsymbol{c}_1^n) \leq n$. Therefore

$$\begin{aligned}
H(\boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n) &= H(M, \boldsymbol{c}_1^n, \boldsymbol{d}_1^n) - H(M|\boldsymbol{d}_1^n) - H(\boldsymbol{d}_1^n)\\
&= H(M|\boldsymbol{c}_1^n, \boldsymbol{d}_1^n) + H(\boldsymbol{d}_1^n|\boldsymbol{c}_1^n) + H(\boldsymbol{c}_1^n)\\
&\qquad - H(M|\boldsymbol{d}_1^n) - H(\boldsymbol{d}_1^n)\\
&\overset{(h)}{\leq} (k+1)n + n\epsilon_n - H(M|\boldsymbol{d}_1^n) - H(\boldsymbol{d}_1^n)\\
&\overset{(i)}{=} (k+1)n + n\epsilon_n - kn - H(\boldsymbol{d}_1^n)\\
&\overset{(j)}{\leq} 2n\epsilon_n + n\delta_n, \qquad (7)
\end{aligned}$$

where (h) follows from (6) and $H(\boldsymbol{d}_1^n|\boldsymbol{c}_1^n) \leq n,\ H(\boldsymbol{c}_1^n) \leq n$; (i) follows from the independence between M and $\boldsymbol{d}_1^n$; (j) follows from (4).

Now we have,

$$\begin{aligned}
&H(\boldsymbol{b}_1^n|M, \boldsymbol{c}_1^n)\\
&= H(M, \boldsymbol{b}_1^n, \boldsymbol{c}_1^n) - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&= H(\boldsymbol{c}_1^n|M, \boldsymbol{b}_1^n) + H(M|\boldsymbol{b}_1^n) + H(\boldsymbol{b}_1^n) - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&\leq (k+1)n + H(\boldsymbol{c}_1^n|M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&\overset{(k)}{=} (k+1)n + H(\boldsymbol{c}_1^n|M, \boldsymbol{b}_1^n, \boldsymbol{d}_1^n) - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&\leq (k+1)n + H(\boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n) - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&\overset{(l)}{\leq} (k+1)n + 2n\epsilon_n + n\delta_n - H(M|\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n)\\
&\overset{(m)}{\leq} n + 2n\epsilon_n + 2n\delta_n - H(\boldsymbol{c}_1^n)\\
&\overset{(n)}{\leq} 3n\epsilon_n + 2n\delta_n, \qquad (8)
\end{aligned}$$

where (k) follows from construction, i.e., $H(\boldsymbol{c}_1^n|M, \boldsymbol{b}_1^n) = H(\boldsymbol{c}_1^n|M, \boldsymbol{b}_1^n, \boldsymbol{d}_1^n)$; (l) follows from (7); (m) follows from the weak security requirement; (n) follows from (3). Therefore,

$$\begin{aligned}
H(\boldsymbol{b}_1^n|\boldsymbol{d}_1^n) &\overset{(o)}{=} H(\boldsymbol{b}_1^n|M, \boldsymbol{d}_1^n)\\
&\leq H(\boldsymbol{b}_1^n, \boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n)\\
&= H(\boldsymbol{b}_1^n|\boldsymbol{c}_1^n, M, \boldsymbol{d}_1^n) + H(\boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n)\\
&\leq H(\boldsymbol{b}_1^n|\boldsymbol{c}_1^n, M) + H(\boldsymbol{c}_1^n|M, \boldsymbol{d}_1^n)\\
&\overset{(p)}{\leq} 3n\epsilon_n + 2n\delta_n + 2n\epsilon_n + n\delta_n = 5n\epsilon_n + 3n\delta_n, \qquad (9)
\end{aligned}$$

where (o) follows as M is independent of $\{\boldsymbol{b}_1^n, \boldsymbol{d}_1^n\}$, and (p) follows from (8) and (7). (9) suggests that the signals $\boldsymbol{b}_1^n$ and $\boldsymbol{d}_1^n$ are strongly dependent. Next we need to lower bound $H(\boldsymbol{b}_1^n)$.

$$\begin{aligned} H(\boldsymbol{b}_1^n) &= H(M, \boldsymbol{b}_1^n, \boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{b}_1^n) \\ &= H(\boldsymbol{b}_1^n | M, \boldsymbol{c}_1^n) + H(M|\boldsymbol{c}_1^n) + H(\boldsymbol{c}_1^n) \\ &\quad - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{b}_1^n) \\ &\geq H(M|\boldsymbol{c}_1^n) + H(\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{b}_1^n) \\ &\overset{(q)}{\geq} kn - n\delta_n + H(\boldsymbol{c}_1^n) - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{b}_1^n) \\ &\overset{(r)}{\geq} (k+1)n - n\epsilon_n - n\delta_n - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) - H(M|\boldsymbol{b}_1^n) \\ &\overset{(s)}{\geq} n - n\epsilon_n - 2n\delta_n - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n) \\ &\overset{(t)}{=} n - n\epsilon_n - 2n\delta_n - H(\boldsymbol{c}_1^n | M, \boldsymbol{b}_1^n, \boldsymbol{d}_1^n) \\ &\geq n - n\epsilon_n - 2n\delta_n - H(\boldsymbol{c}_1^n | M, \boldsymbol{d}_1^n) \\ &\overset{(u)}{\geq} n - 3n\epsilon_n - 3n\delta_n, \end{aligned} \tag{10}$$

where (q) follows from the weak security requirement; (r) follows from (3); (s) follows again from the weak security; (t) follows from construction; (u) follows from (7).

Finally, by (9) and (10),

$$I(\boldsymbol{b}_1^n; \boldsymbol{d}_1^n) = H(\boldsymbol{b}_1^n) - H(\boldsymbol{b}_1^n | \boldsymbol{d}_1^n) \geq n - 8n\epsilon_n - 6n\delta_n,$$

The above argument extends to all other paths naturally (by renumbering the notation accordingly), so

$$I(\boldsymbol{b}_i^n; \boldsymbol{d}_i^n) \geq n - 8n\epsilon_n - 6n\delta_n, \quad \forall i = 1, \dots, k. \tag{11}$$

Lemma 1, stated below and proved in the Appendix, shows that (11) implies the asymptotic achievability of unit rate in $\mathcal{I}$, which completes the proof of this direction. Intuitively, since the mutual information between $\boldsymbol{b}_i^n$ and $\boldsymbol{d}_i^n$ is asymptotically n, we can use a random coding argument similar to that used in the proof of the channel coding theorem to show the existence of a network code achieving unit rate asymptotically in $\mathcal{I}$. The reason that we do not use the standard point-to-point channel coding theorem is that there are multiple interacting source-terminal pairs in $\mathcal{I}$ and we need to make sure the existence of a code that is good for all pairs.

Lemma 1: For any $\epsilon > 0$, if there exists a network code of length n for $\mathcal{I}_s$ such that $I(\boldsymbol{b}_i^n; \boldsymbol{d}_i^n) > n(1-\epsilon)$, then unit rate is asymptotically achievable in $\mathcal{I}$.

This completes the proof.

Theorem 1 can be easily adapted to the case of zero-error communication.

Corollary 1: Given any multiple-unicast network coding problem $\mathcal{I}$ with source-destination pairs $\{(s_i, t_i), i = 1, \dots, k\}$, a corresponding unicast secure network coding problem $\mathcal{I}_s = (\mathcal{G}, s, t, \mathcal{A})$, in which $\mathcal{A}$ includes all sets of a single edge (i.e., all singletons), can be constructed according to Construction 1, such that unit rate is achievable with zero error in $\mathcal{I}$ if and only if rate k is achievable with zero error in $\mathcal{I}_s$ under perfect security.

The proof of Corollary 1 follows the same line as the proof of Theorem 1, with the difference that all ϵ and δ become strictly 0. For example, (9) implies that $\boldsymbol{b}_1^n$ is a function of $\boldsymbol{d}_1^n$, and hence that it can be perfectly decoded from $\boldsymbol{d}_1^n$.

We remark that our reduction has an operational aspect that from a code for $\mathcal{I}_s$ one can construct a code for $\mathcal{I}$. Indeed, using our reduction, to solve a multiple-unicast network coding problem, one may first reduce it to a unicast secure network coding problem, then solve the latter, and finally use this solution to obtain a solution to the original multiple-unicast problem.

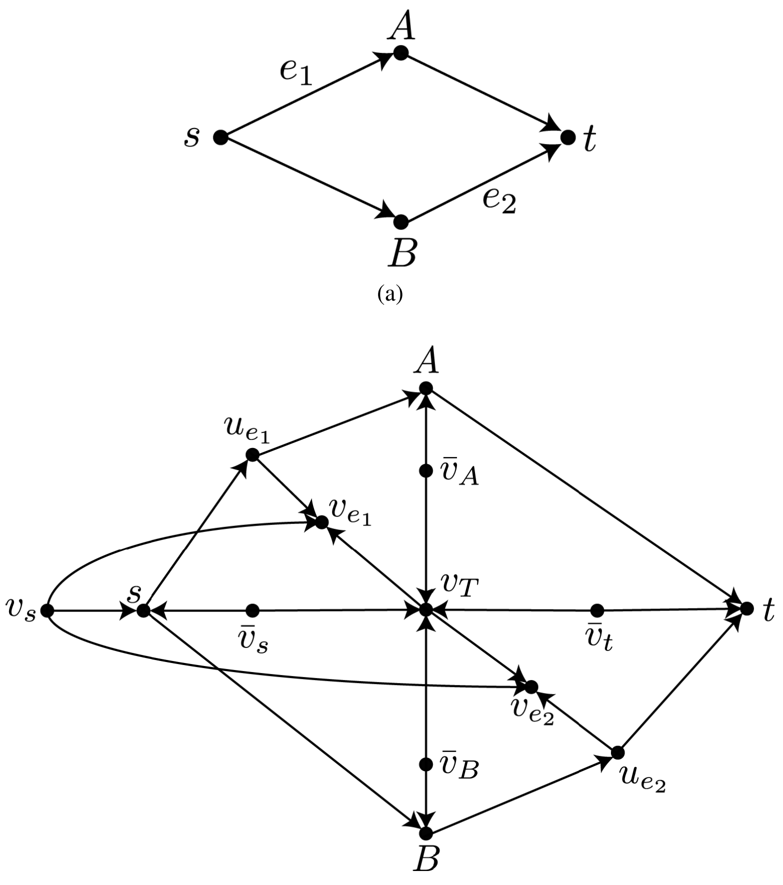
Chan and Grant [18] previously show that determining the zero-error achievability of a rate in a multicast secure network coding problem with general setup (i.e., arbitrary edge capacities, arbitrary $\mathcal{A}$, and arbitrary nodes may generate randomness) and with perfect security is at least as hard as determining the zero-error achievability of a rate tuple in multiple-multicast network coding. The result in this section significantly strengthens the result in [18] by showing that the secure network coding problem under a much simpler setup (i.e., unicast, equal link capacities, uniform $\mathcal{A}$ with a single wiretap link) is still hard, under various definitions of achievability and security.

IV. Reducing Secure Unicast to Multiple-Unicast

In the previous section we have reduced an arbitrary multiple-unicast network coding problem into a particular unicast secure network coding problem with a very simple setup, in which at most one link can be eavesdropped. Conversely, given an arbitrary unicast secure network coding problem where at most one link can be eavesdropped, we can reduce it into a particular multiple-multicast network coding problem without security requirements (which can in turn be reduced into an equivalent multiple-unicast network coding problem [9]). A more general construction was first proposed in [35] for the purpose of lower bounding the capacity of a secure network coding instance by studying a corresponding multiple-multicast network coding instance. In this paper we simplify the construction to address the special case that at most one link can be eavesdropped, i.e., $\mathcal{A}$ comprises only singletons. Loosely speaking, we show that for this special case the multiple-multicast network coding instances not only lower bound but also upper bound the capacity of the secure network coding instances, hence giving a reduction.

Construction 2: Given a unicast secure network coding problem $\mathcal{I}_s$ on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with source s, terminal t, and a collection of wiretap sets $\mathcal{A}$ comprising only singletons. We construct a corresponding multiple-multicast network coding problem $\mathcal{I}$ on an augmented graph $\check{\mathcal{G}} = (\check{\mathcal{V}}, \check{\mathcal{E}})$. We define $\check{\mathcal{E}}$ and $\check{\mathcal{V}}$ from $\mathcal{E}$ and $\mathcal{V}$. In $\mathcal{E}$, denote by c_e the capacity of link e, and by $\mathcal{E}_{out}(i)$ the set of outgoing edges of node i.

1) *For $i \in \mathcal{V}$, add i to $\check{\mathcal{V}}$. For $e \in \mathcal{E}$ such that $\{e\} \notin \mathcal{A}$ (in the remaining part of this subsection we will write $e \notin \mathcal{A}$ or $e \in \mathcal{A}$ instead, because the elements of $\mathcal{A}$ are singletons), add e to $\check{\mathcal{E}}$.*
2) *For $e = (i, j) \in \mathcal{E}$ such that $e \in \mathcal{A}$, create nodes u_e, v_e in $\check{\mathcal{V}}$; create edges (i, u_e), (u_e, j) and (u_e, v_e) in $\check{\mathcal{E}}$, all of capacity c_e.*
3) *Create a key aggregation node v_T in $\check{\mathcal{V}}$. For each node $i \in \mathcal{V}$, create a key source node $\bar{v}_i$ in $\check{\mathcal{V}}$. Create two links $(\bar{v}_i, v_T)$ and $(\bar{v}_i, i)$ in $\check{\mathcal{E}}$, both of*



(a)

(b)

Fig. 3. An example of Construction 2. (a) A unicast secure network coding instance $\mathcal{I}_s$ where s is the source, t is the terminal and $\mathcal{A} = \{\{e_1\}, \{e_2\}\}$. (b) The multiple-multicast network coding instance $\mathcal{I}$ obtained from $\mathcal{I}_s$ according to Construction 2. In this problem, t demands the message originated from v_s; v_{e_1} and v_{e_2} demand the messages originated from v_s, $\bar{v}_s$, $\bar{v}_A$, $\bar{v}_B$ and $\bar{v}_t$.

capacity

$$\check{c}_i = \sum_{e \in \mathcal{E}_{out}(i)} c_e.$$

4) *Create a virtual source node v_s in $\check{\mathcal{V}}$. For all $e \in \mathcal{A}$, create links (v_s, s) and (v_s, v_e) in $\check{\mathcal{E}}$, both of capacity $\sum_{e' \in \mathcal{E}_{out}(s)} c_{e'}$.*
5) *For all $e \in \mathcal{A}$, create a link (v_T, v_e) in $\check{\mathcal{E}}$ of capacity*

$$\sum_{e' \in \mathcal{E}} c_{e'} - c_e.$$

In $\mathcal{I}$, nodes v_s and $\bar{v}_i$, $i \in \mathcal{V}$ are associated with independent messages. Node t demands the message of v_s. For all $e \in \mathcal{A}$, node v_e demands the messages of v_s and all $\bar{v}_i$. Note that in $\mathcal{I}$ there is no security requirement.

Refer to Figure 3 for an example of Construction 2.

Denote for short $\check{\boldsymbol{c}}_{\mathcal{V}} = (\check{c}_i, i \in \mathcal{V})$. Consider any unicast secure network coding instance $\mathcal{I}_s = (\mathcal{G}, s, t, \mathcal{A})$ such that $\mathcal{A}$ comprises only singletons. Let $\mathcal{I}$ be the multiple-multicast network coding instance obtained from $\mathcal{I}_s$ according to Construction 2. The next theorem reveals a connection between $\mathcal{I}_s$ and $\mathcal{I}$.

Theorem 2: Rate R is asymptotically achievable in $\mathcal{I}_s$ subject to the weak or strong security requirement if and only if rate tuple $(R, \check{\boldsymbol{c}}_{\mathcal{V}})$ is asymptotically achievable in $\mathcal{I}$, where R is the rate of the message of v_s, and $\check{\boldsymbol{c}}_{\mathcal{V}}$ is the rate tuple of messages of $\bar{v}_i$, $i \in \mathcal{V}$.

Proof: We first note that by [20], the capacity region of $\mathcal{I}_s$ subject to the weak security requirement is the same as the capacity region subject to the strong security requirement. Therefore, a rate is asymptotically achievable in $\mathcal{I}_s$ subject to weak security if and only if it is asymptotically achievable subject to strong security. In the proof we assume that weak security is imposed in $\mathcal{I}_s$, and the case of strong security follows directly from the equivalence.

"$\Leftarrow$". Assuming that rate tuple $(R, \check{\boldsymbol{c}}_{\mathcal{V}})$ is asymptotically achievable in $\mathcal{I}$, it is proved in [35] that rate R is asymptotically achievable in $\mathcal{I}_s$ under the weak security requirement. Here we give a simplified proof of this fact for completeness. By hypothesis, for any $\boldsymbol{\epsilon} > \boldsymbol{0}$ and $\delta > 0$, there exists a network code ϕ of length n that achieves rate tuple $(R, \check{\boldsymbol{c}}_{\mathcal{V}}) - \boldsymbol{\epsilon}$ in $\mathcal{I}$ with error probability δ. In problem $\mathcal{I}_s$, we simulate the network code ϕ. Specifically, the message originally associated with v_s is replaced by the secret source message, and the messages originally associated with $\bar{v}_i, i \in \mathcal{V}$ are replaced by the independent random keys generated at node i. Let every edge $e = (i, j)$ in $\mathcal{E}$ simulate the encoding function of edge $e \in \check{\mathcal{E}}$ if $e \notin \mathcal{A}$ or the encoding function of (i, u_e) in $\check{\mathcal{E}}$ if $e \in \mathcal{A}$; then the terminal t, by simulating the decoding function, can decode the source message with error probability δ. Therefore decodability is not a problem.

It remains to show that simulating the code ϕ meets the weak security requirement. For an edge $\check{e} \in \check{\mathcal{E}}$, let $X_{\check{e}}$ be the signal transmitted on $\check{e}$ induced by ϕ. We denote by M the message associated with v_s, and by K_i the message associated with $\bar{v}_i$. The for any $e \in \mathcal{A}$, the inequality in (12), as shown at the bottom of the next page follows, where (a) follows from expanding $I(M; X_{(v_s,v_e)}|X_{(u_e,v_e)})$ in two ways; (b) follows from the construction that $X_{(v_s,v_e)}$ is a function of M; (c) follows form the chain rule; (d) follows from the fact that $H(X_{(u_e,v_e)}) \leq nc_{(u_e,v_e)} = nc_e$ and $H(X_{(v_T,v_e)}|X_{(u_e,v_e)}, X_{(v_s,v_e)}) \leq nc_{(v_T,v_e)} = n\sum_{e' \in \mathcal{E}} c_{e'} - nc_e$; (e) follows from the chain rule; (f) follows from $H(M, K_{\mathcal{V}}) \geq nR + n\sum_{e' \in \mathcal{E}} c_{e'} - n\epsilon$, where ϵ is the sum of the entries of $\boldsymbol{\epsilon}$; and (g) follows from Fano's inequality, where $\delta' \to 0$ as $\delta \to 0$. (12) implies that $I(M; X_{(u_e,v_e)}) \leq n(\epsilon + \delta')$. And because $X_{(u_e,v_e)}$ can be viewed as the observation of an adversary eavesdropping on edge e, the weak security requirement is met.

"$\Rightarrow$". Assuming that rate R is asymptotically achievable in $\mathcal{I}_s$ under the weak security requirement, we show that rate tuple $(R, \check{\boldsymbol{c}}_{\mathcal{A}})$ is asymptotically achievable in $\mathcal{I}$. For $e \in \mathcal{E}$, denote by X_e the signal transmitted on edge e. By hypothesis, for arbitrary $\epsilon_R > 0$, $\epsilon_S > 0$ and $\delta > 0$, there exists a network code ϕ with length n that achieves rate $R - \epsilon_R$ weakly securely in $\mathcal{I}_s$ with error probability δ, such that $I(M; X_e) \leq n\epsilon_S$, for all $e \in \mathcal{A}$. For arbitrary $\epsilon_E > 0$, without loss of generality we assume that $H(X_e) \geq n(c_e - \epsilon_E)$, for all $e \in \mathcal{E}$. This is because if $H(X_e) < nc_e$, i.e., if X_e is not "almost" uniform, then we can repeat ϕ for m times and perform a source code of length m (over the supersymbol alphabet $\{0, 1\}^n$) on edge e to compress X_e by encoding only the typical sequences [11, Sec. 3.2], and use the spare capacity of the edge to transmit random bits. Meanwhile, source s and terminal t perform an outer channel code of length m (also over alphabet $\{0, 1\}^n$) to keep the error probability small. For sufficiently large m, the overall concatenated code asymptotically achieves the same rate weakly securely as ϕ does, and such that $H(X_e)/mn$, as desired, is arbitrarily close to c_e due to the asymptotic equipartition property.

Note that in general a node $i \in \mathcal{V}$ will generate an independent random variable K_i as input to the encoding functions. The rate of K_i is upper bounded by $\check{c}_i$, which is the sum capacities of all outgoing edges of i.

We now turn to the problem $\mathcal{I}$. We construct a network code ϕ' of a slightly longer length $n' = (1 + \epsilon_L)n$. In the first n channel uses, ϕ' simulates the operation of ϕ. Specifically, node v_s generates a random variable M uniformly distributed over $[2^{n(R-\epsilon_R)}]$, and transmit it to s via edge (v_s, s). For $i \in \mathcal{V}$, node $\bar{v}_i$ generates a random variable K_i uniformly distributed over $[2^{n\check{c}_i}]$, and transmit it to i via edge $(\bar{v}_i, i)$. The rate of M (over code length n') is $\frac{1}{1+\epsilon_L}(R - \epsilon_R)$ and the rate of K_i is $\frac{1}{1+\epsilon_L}\check{c}_i$. The rates of M and $K_{\mathcal{V}}$ can be made arbitrarily close to R and $\check{\boldsymbol{c}}_{\mathcal{V}}$ by choosing sufficiently small ϵ_R and ϵ_L.

To simulate ϕ, ϕ' performs the same encoding function on edge $e \in \mathcal{E}$ (if e remains in $\check{\mathcal{E}}$) as ϕ did in $\mathcal{I}_s$. Otherwise if e is replaced by (i, u_e) and (u_e, j), then ϕ' performs the same encoding function on $(i, u_e) \in \check{\mathcal{E}}$ as ϕ did on edge $e \in \mathcal{E}$. The induced signal $X_{(i,u_e)}$ is then relayed to edge (u_e, j) and (u_e, v_e). In the extra ϵ_L channel uses, these edges (from $\mathcal{E}$) keep silent (or simply transmit dummy zeros). At terminal node t, by simulating the decoding function, it is able to decode M correctly with probability of error δ.

We next show that $(K_i, i \in \mathcal{V})$ and M can be decoded at v_e, for all $e \in \mathcal{A}$. Note that v_e has three incoming edges, i.e., (u_e, v_e), (v_s, v_e) and (v_T, v_e). The signal $X_{(u_e,v_e)}$ transmitted on (u_e, v_e), as described in the previous paragraph, is the same as the signal X_e on edge e in the secure network coding problem $\mathcal{I}_s$. Let edge (v_s, v_e) transmit M. Now consider a distributed source coding problem with three correlated sources $(K_i, i \in \mathcal{V})$, X_e and M. Note the X_e and M are available at v_e by construction, and the question is that whether v_e can decode $K_{\mathcal{V}}$ from X_e, M and the signal received from edge (v_T, v_e). It follows that,

$$
\begin{aligned}
H(K_{\mathcal{V}}|X_e, M) &= H(X_e, K_{\mathcal{V}}|M) - H(X_e|M) \\
&= H(X_e|K_{\mathcal{V}}, M) + H(K_{\mathcal{V}}|M) - H(X_e|M) \\
&\overset{(a)}{=} H(K_{\mathcal{V}}|M) - H(X_e|M) \\
&\overset{(b)}{=} H(K_{\mathcal{V}}) - H(X_e|M) \\
&\leq n\sum_{e' \in \mathcal{E}} c_{e'} - H(X_e|M) \\
&= n\sum_{e' \in \mathcal{E}} c_{e'} - (H(X_e) - I(M; X_e)) \\
&\overset{(c)}{\leq} n(\sum_{e' \in \mathcal{E}} c_{e'} - c_e) + n(\epsilon_S + \epsilon_E) \\
&\overset{(d)}{<} n(\sum_{e' \in \mathcal{E}} c_{e'} - c_e) + n\epsilon_L(\sum_{e' \in \mathcal{E}} c_{e'} - c_e) \\
&= n'(\sum_{e' \in \mathcal{E}} c_{e'} - c_e), \qquad (13)
\end{aligned}
$$

where (a) follows because X_e is a function of $K_{\mathcal{V}}$ and M; (b) follows from the fact that $K_{\mathcal{V}}$ is independent of M; (c) follows from the weak security requirement and that $H(X_e) \geq n(c_e - \epsilon_E)$; and (d) follows by choosing a sufficiently small ϵ_S and ϵ_E such that $\epsilon_S + \epsilon_E < \epsilon_L(\sum_{e' \in \mathcal{E}} c_{e'} - c_e)$, for all $e \in \mathcal{A}$. Note that $\sum_{e' \in \mathcal{E}} c_{e'} - c_e$ is the capacity of edge (v_T, v_e), and so by (13) and the Slepian-Wolf Theorem of distributed source coding, the rate tuple of the correlated sources are in the achievable region. Therefore by concatenating a (outer) Slepian-Wolf code with the (inner) network code ϕ', for all $e \in \mathcal{A}$, v_e is able to decode $(M, K_{\mathcal{V}})$ with vanishing probability of error. This completes the proof. □

V. Reducing Multiple-Unicast to Unicast Network Error Correction

In this section we reduce instances of the multiple-unicast network coding problem to instances of the unicast network error correction problem. We start with the zero-error case.

A. Zero-Error Achievability

The following theorem reduces the problem of determining the zero-error achievability of a rate in a general multiple-unicast network coding instance to the problem of determining

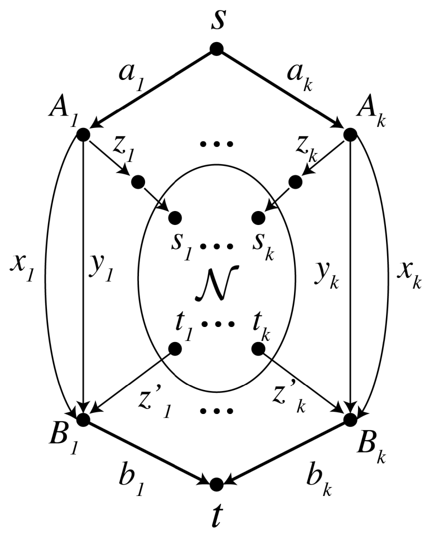


Fig. 4. In the unicast network error correction problem $\mathcal{I}_c$, s is the source and t is the terminal. $\mathcal{N}$ is the network on which $\mathcal{I}$ is defined. All edges outside $\mathcal{N}$ (i.e., edges for which at least one of its end-point does not belong to $\mathcal{N}$) have unit capacity. There is at most one error in this network, and this error can occur at any edge except $\{a_i, b_i, 1 \le i \le k\}$. Namely, $\mathcal{B}$ includes all singleton sets of a single edge in the network except $\{a_i\}$ and $\{b_i\}$, $i = 1, \ldots, k$ (the thickened edges). Note that there are k parallel branches in total going from s to t but only the first and the k-th branches are drawn explicitly.

the zero-error achievability of a rate in a particular unicast network error correction instance that has a very simple setup. Recall from the remark before Theorem 1 that there is no loss of generality in addressing the achievability of a rate instead of a rate tuple in the multiple-unicast network coding problem.

Construction 3: Given any multiple-unicast network coding problem $\mathcal{I}$ on a network $\mathcal{N}$ with source-destination pairs $\{(s_i, t_i), i = 1, \ldots, k\}$, a unicast network error correction problem $\mathcal{I}_c$ is constructed as specified in Figure 4.

Theorem 3: Given any multiple-unicast network coding problem $\mathcal{I}$ with source-destination pairs $\{(s_i, t_i), i = 1, \ldots, k\}$, a corresponding unicast network error correction problem $\mathcal{I}_c = (\mathcal{G}, s, t, \mathcal{B})$ in which $\mathcal{B}$ includes sets with at most one edge can be constructed according to Construction 3, such that unit rate is achievable with zero-error in $\mathcal{I}$ if and only if rate k is achievable with zero-error in $\mathcal{I}_c$.

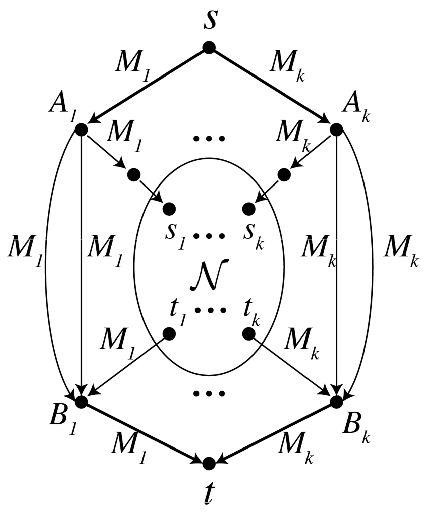


Fig. 5. A scheme to achieve zero-error rate k in $\mathcal{I}_c$ given that unit rate is achievable with zero error in $\mathcal{I}$. $M = (M_1, \ldots, M_k)$ and node B_i performs majority decoding.

Proof: "⇒". We show that the zero-error achievability of unit rate in $\mathcal{I}$ implies the zero-error achievability of rate k in $\mathcal{I}_c$. A constructive scheme is shown in Figure 5. In $\mathcal{I}_c$, the source lets $M = (M_1, \ldots, M_k)$, where the M_i's are i.i.d. uniformly distributed over $[2^n]$. In $\mathcal{N}$, we simulate the network code for $\mathcal{I}$ that achieves unit rate with zero error. Outside $\mathcal{N}$, let the network code be $a_i(M) = x_i(M) = y_i(M) = z_i(M) = z'_i(M) = M_i$, $i = 1, \ldots, k$. Note that the signals on edges x_i, y_i and z'_i form a repetition code and node B_i, by performing majority decoding, can correct any single error to obtain M_i. Hence the scheme ensures that $b_i(M) = M_i$ under all possible error patterns, and so rate k is achievable with zero error in $\mathcal{I}_c$.

$$
\begin{aligned}
H(M|X_{(u_e,v_e)}) &\overset{(a)}{=} H(X_{(v_s,v_e)}|X_{(u_e,v_e)}) + H(M|X_{(v_s,v_e)}, X_{(u_e,v_e)}) - H(X_{(v_s,v_e)}|X_{(u_e,v_e)}, M) \\
&\geq H(X_{(v_s,v_e)}|X_{(u_e,v_e)}) - H(X_{(v_s,v_e)}|X_{(u_e,v_e)}, M) \\
&\overset{(b)}{=} H(X_{(v_s,v_e)}|X_{(u_e,v_e)}) \\
&\overset{(c)}{=} H(X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) - H(X_{(u_e,v_e)}) - H(X_{(v_T,v_e)}|X_{(u_e,v_e)}, X_{(v_s,v_e)}) \\
&\overset{(d)}{\geq} H(X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) - n \sum_{e' \in \mathcal{E}} c_{e'} \\
&\overset{(e)}{=} H(M, K_{\mathcal{V}}, X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) - H(M, K_{\mathcal{V}}|X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) - n \sum_{e' \in \mathcal{E}} c_{e'} \\
&\geq H(M, K_{\mathcal{V}}) - H(M, K_{\mathcal{V}}|X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) - n \sum_{e' \in \mathcal{E}} c_{e'} \\
&\overset{(f)}{\geq} nR - n\epsilon - H(M, K_{\mathcal{V}}|X_{(v_T,v_e)}, X_{(u_e,v_e)}, X_{(v_s,v_e)}) \\
&\overset{(g)}{\geq} nR - n\epsilon - n\delta' \geq H(M) - n(\epsilon + \delta')
\end{aligned} \tag{12}
$$

"$\Leftarrow$". We show that the zero-error achievability of rate k in $\mathcal{I}_c$ implies the zero-error achievability of unit zero-error rate in $\mathcal{I}$.

Suppose rate k is achieved with zero-error in $\mathcal{I}_c$ by a network code with length n, and denote the source message by M, which is uniformly distributed over $[2^{nk}]$. Recall from Section II-C that, $\boldsymbol{r} = (r_e)_{e\in\mathcal{E}}$ is the tuple of additive error signals called an error pattern, and $\mathcal{R_B}$ the set of all possible error patterns, i.e., $\mathcal{R_B} = \{\boldsymbol{r}$: non-zero entries in $\boldsymbol{r}$ correspond to B-errors, $B \in \mathcal{B}\}$. For any edge $e \in \mathcal{E}$, we denote by $e(m, \boldsymbol{r}) : [2^{nk}] \times \mathcal{R_B} \to [2^n]$ the signal received on edge e when the source message equals m and the error pattern $\boldsymbol{r}$ occurs in the network.

Let $\boldsymbol{b}(m, \boldsymbol{r}) = (b_1(m, \boldsymbol{r}), \ldots, b_k(m, \boldsymbol{r}))$, then because the edges $b_1, \ldots, b_k$ form a cut-set from s to t, $\boldsymbol{b}(m, \boldsymbol{r})$ must be injective with respect to m due to the zero error decodability constraint. Formally, for two different messages $m_1 \neq m_2$, it follows from the zero error decodability constraint that $\boldsymbol{b}(m_1, \boldsymbol{r}_1) \neq \boldsymbol{b}(m_2, \boldsymbol{r}_2)$, $\forall \boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathcal{R_B}$. Note that the codomain of $\boldsymbol{b}$ is $[2^n]^k$, which has the same size as the set of messages $[2^{nk}]$. Therefore denote by $\boldsymbol{b}(m) = \boldsymbol{b}(m, \boldsymbol{0})$, then $\boldsymbol{b}(m)$ is a bijective function. This implies that $\boldsymbol{b}(m, \boldsymbol{r}) = \boldsymbol{b}(m)$, $\forall \boldsymbol{r} \in \mathcal{R_B}$, because otherwise if there exist $m_1, \boldsymbol{r}_1$ such that $\boldsymbol{b}(m_1, \boldsymbol{r}_1) \neq \boldsymbol{b}(m_1)$, then there is a $m_2 \neq m_1$ such that $\boldsymbol{b}(m_1, \boldsymbol{r}_1) = \boldsymbol{b}(m_2)$, violating the decodability requirement. Similarly, let $\boldsymbol{a}(m, \boldsymbol{r}) = (a_1(m, \boldsymbol{r}), \ldots, a_k(m, \boldsymbol{r}))$ and $\boldsymbol{a}(m) = \boldsymbol{a}(m, \boldsymbol{0})$, then $\boldsymbol{a}(m)$ is a bijective function and $\boldsymbol{a}(m, \boldsymbol{r}) = \boldsymbol{a}(m)$, $\forall \boldsymbol{r} \in \mathcal{R_B}$.

For any $e \in \mathcal{E}$, denote $e(m) = e(m, \boldsymbol{0})$. We make the following claim:

Claim: For $i = 1, \ldots, k$ and any two messages $m_1, m_2 \in [2^{nk}]$ such that $a_i(m_1) \neq a_i(m_2)$, it follows that $x_i(m_1) \neq x_i(m_2)$, $y_i(m_1) \neq y_i(m_2)$ and $z_i(m_1) \neq z_i(m_2)$.

To prove the claim, suppose for contradiction that there exist m_1, m_2 such that $a_i(m_1) \neq a_i(m_2)$ and that the claim is not true, i.e., $x_i(m_1) = x_i(m_2)$ or $y_i(m_1) = y_i(m_2)$ or $z_i(m_1) = z_i(m_2)$. First consider the case that $x_i(m_1) = x_i(m_2)$. Because of the one-to-one correspondence between m and $\boldsymbol{a}$, there exists a message $m_3 \neq m_1$ and such that $\boldsymbol{a}(m_3) = (a_1(m_1), \ldots, a_{i-1}(m_1), a_i(m_2), a_{i+1}(m_1), \ldots, a_k(m_1))$. Then $x_i(m_1) = x_i(m_3)$ because by construction $x_i(m_3) = x_i(m_2)$, and by hypothesis $x_i(m_1) = x_i(m_2)$. Consider the following two scenarios. In the first scenario, m_1 is transmitted, and an error turns $y_i(m_1)$ into $y_i(m_3)$; in the second scenario, m_3 is transmitted, and an error turns $z_i(m_3)$ into $z_i(m_1)$. Then the cut-set signals $a_1, \ldots, a_{i-1}, x_i, y_i, z_i, a_{i+1}, \ldots, a_k$ are exactly the same in both scenarios, and so it is impossible for t to distinguish m_1 from m_3, a contradiction to the zero error decodability constraint. Therefore $x_i(m_1) \neq x_i(m_2)$. With a similar argument it follows that $y_i(m_1) \neq y_i(m_2)$ and $z_i(m_1) \neq z_i(m_2)$, and the claim is proved.

The claim above suggests that (the signals on) x_i, y_i and z_i, as functions of (the signals on) a_i, are injective. They are also surjective functions because the domain and codomain are both $[2^n]$. Hence there are one-to-one correspondences between a_i, x_i, y_i and z_i.

Next we show that for any two messages m_1, m_2, if $b_i(m_1) \neq b_i(m_2)$, then $z_i'(m_1) \neq z_i'(m_2)$. Suppose for contradiction that there exists $m_1 \neq m_2$ such that $b_i(m_1) \neq b_i(m_2)$ and $z_i'(m_1) = z_i'(m_2)$. Then if m_1 is transmitted and an error $\boldsymbol{r}_1$ turns $x_i(m_1)$ into $x_i(m_2)$, the node B_i will receive the same signals as in the case that m_2 is transmitted and an error $\boldsymbol{r_2}$ turns $y_i(m_2)$ into $y_i(m_1)$. Therefore $b_i(m_1, \boldsymbol{r}_1) = b_i(m_2, \boldsymbol{r}_2)$. But, as shown above, because $b_i(m_1, \boldsymbol{r}_1) = b_i(m_1)$ and $b_i(m_2, \boldsymbol{r}_2) = b_i(m_2)$, it follows that $b_i(m_1) = b_i(m_2)$, a contradiction. This claim suggests that if $z_i'(m_1) = z_i'(m_2)$ then $b_i(m_1) = b_i(m_2)$ and therefore b_i is a function of z_i'. This function is surjective because b_i takes all 2^n possible values. Then since the domain and the codomain are both $[2^n]$, b_i must be a bijective function of z_i'. With the same argument it follows that b_i is also a bijective function of x_i.

Hence z_i is a bijection of a_i, a_i is a bijection of x_i, x_i is a bijection of b_i, and b_i is a bijection of z_i'. Therefore for all $1 \leq i \leq k$, z_i is a bijection of z_i', and therefore unit rate is achievable with zero error in $\mathcal{I}$. □

B. Achievability Allowing Vanishing Error

In this subsection we show that Construction 3 gives a reduction from multiple-unicast network coding to unicast network error correction not only in terms of zero-error achievability, but also in terms of achievability that allows vanishing error. Recall from Section II that we say a rate is achievable if there exists a code achieving the rate with a vanishing error probability.

Theorem 4: Given any multiple-unicast network coding problem $\mathcal{I}$ with source-destination pairs $\{(s_i, t_i), i = 1, \ldots, k\}$, a corresponding unicast network error correction problem $\mathcal{I}_c = (\mathcal{G}, s, t, \mathcal{B})$ in which $\mathcal{B}$ includes sets with at most a single edge can be constructed according to Construction 3, such that unit rate is achievable in $\mathcal{I}$ if and only if rate k is achievable in $\mathcal{I}_c$.

We first prove the forward direction of the theorem, which is simple and is similar to the proof of the zero-error case.

Proof: [Proof ("$\Rightarrow$" part of Theorem 4)] We show that if unit rate is achievable in $\mathcal{I}$, then rate k is achievable in $\mathcal{I}_c$. Again we use the constructive scheme in Figure 5. In $\mathcal{I}_c$, the source lets $M = (M_1, \ldots, M_k)$, where the M_i's are i.i.d. uniformly distributed over $[2^n]$. In $\mathcal{N}$, we simulate the network code for $\mathcal{I}$ that achieves unit rate. Outside $\mathcal{N}$, let the network code be $a_i(M) = x_i(M) = y_i(M) = z_i(M) = M_i$, $i = 1, \ldots, k$. Consider the M_i's as the source messages of the simulated $\mathcal{I}$, and denote by $\hat{M}_i$, $i = 1, \ldots, k$ the outputs of the decoders of $\mathcal{I}$. Let $z_i'(M) = \hat{M}_i$, $i = 1, \ldots, k$, and let node B_i, $i = 1, \ldots, k$ performs majority decoding. The terminal t will not decode an error as long as the multiple-unicast instance $\mathcal{I}$ does not commit an error, i.e., $\hat{M}_i = M_i$. The error probability of $\mathcal{I}$ is negligible, which implies the achievability of rate k in $\mathcal{I}_c$. □

In the remainder of this subsection we prove the other direction of Theorem 4, i.e., that the achievability of rate k in $\mathcal{I}_c$ implies the achievability of unit rate in $\mathcal{I}$.

The main idea of the proof is as follows. In $\mathcal{I}$ the network will simulate the given network code for $\mathcal{I}_c$, and by doing so the terminal t_i will obtain the signal on z_i' in $\mathcal{I}_c$. The main task therefore is to estimate the signal on z_i, which terminal t_i demands, based on the observed signal on z_i'. This task is accomplished by following two steps. In the first step, we construct an explicit decoding function that, taking the signal on z_i' as input, outputs the "most likely" signal to be transmitted on b_i under the given network code for $\mathcal{I}_c$. We then prove (by combinatorial arguments) that the probability of error of this decoding function is small if the probability of error of the given network code is small. In the second step, we construct another explicit decoding function that takes the signal on b_i as input and outputs the "most likely" signal transmitted on a_i. Again we can prove that if the given network code for $\mathcal{I}_c$ has a small probability of error then so does this decoding function. Finally, terminal t_i, by concatenating the two decoding functions, is able to estimate the signal on a_i and therefore the signal on z_i correctly with high probability. We now proceed with the formal proof.

Suppose in $\mathcal{I}_c$ a rate of k is achieved by a network code $\mathcal{C} = \{\phi_e, \phi_t\}_{e\in\mathcal{E}}$ with length n, and with a probability of error ϵ. Recall that M is the source message uniformly distributed over $\mathcal{M} = [2^{kn}]$, and $\hat{M}$ is the output of the decoder at the terminal. Let $\mathcal{M}^{\text{good}} = \{m \in [2^{kn}] : \mathcal{C}$ satisfies $\mathcal{I}_c$ under transmission $m\}$ be the subset of messages that can be decoded correctly under any error pattern $\boldsymbol{r} \in \mathcal{R}_{\mathcal{B}}$. Denote by $\mathcal{M}^{\text{bad}} = \mathcal{M}\backslash\mathcal{M}^{\text{good}}$, then for any $m \in \mathcal{M}^{\text{bad}}$, there exists an error pattern $\boldsymbol{r} \in \mathcal{R}_{\mathcal{B}}$ such that the decoded value $\hat{M}$ differs from M when $M = m$ and $\boldsymbol{r}$ occurs, i.e., a decoding error occurs. Because $\mathcal{C}$ satisfies $\mathcal{I}_c$ with error probability ϵ (recall from Section II-C that the probability of error is taken over the distribution on the message M), it follows that $|\mathcal{M}^{\text{bad}}| \leq 2^{kn}\epsilon$ and thus $|\mathcal{M}^{\text{good}}| \geq (1-\epsilon)\cdot 2^{kn}$.

We introduce some notation needed in the proof. In problem $\mathcal{I}_c$, under the network code $\mathcal{C}$, for $i = 1, \ldots, k$, let $x_i(m, \boldsymbol{r}) : \mathcal{M} \times \mathcal{R}_{\mathcal{B}} \to [2^n]$ be the signal received from channel x_i when $M = m$ and the error pattern $\boldsymbol{r}$ happens. Let $\boldsymbol{r} = \boldsymbol{0}$ denote the case that no error has occurred in the network. Let $x_i(m) = x_i(m, \boldsymbol{0})$, $\boldsymbol{x}(m, \boldsymbol{r}) = (x_1(m, \boldsymbol{r}), \ldots, x_k(m, \boldsymbol{r}))$ and $\boldsymbol{x}(m) = (x_1(m), \ldots, x_k(m))$. We define $a_i, b_i, y_i, z_i, z_i', \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{z}'$ for problem $\mathcal{I}_c$ in a similar way. Note that they are all functions of m and $\boldsymbol{r}$. We use the hat notation to represent a specific value in the range of a function, e.g., $\hat{a}_i$ represents a specific output of a_i. In other words, $\hat{a}_i$ is a specific n-bit signal.

Notice that the set of edges $a_1, \ldots, a_k$ forms a cut-set from s to t, and so does the set of edges $b_1, \ldots, b_k$. Therefore for any $m_1, m_2 \in \mathcal{M}^{\text{good}}$, $m_1 \neq m_2$, it follows from the decodability constraint that $\boldsymbol{a}(m_1) \neq \boldsymbol{a}(m_2)$ and $\boldsymbol{b}(m_1) \neq \boldsymbol{b}(m_2)$. Setting $\mathcal{B}^{\text{good}} = \{\boldsymbol{b}(m) : m \in \mathcal{M}^{\text{good}}\}$, it then follows from $|\mathcal{M}^{\text{good}}| \geq (1-\epsilon)\cdot 2^{kn}$ that $|\mathcal{B}^{\text{good}}| \geq (1-\epsilon)\cdot 2^{kn}$. Setting $\mathcal{B}^{\text{err}} = [2^n]^k\backslash\mathcal{B}^{\text{good}}$, it follows that $|\mathcal{B}^{\text{err}}| \leq 2^{kn}\epsilon$. Similarly, we define $\mathcal{A}^{\text{good}} = \{\boldsymbol{a}(m) : m \in \mathcal{M}^{\text{good}}\}$, and $\mathcal{A}^{\text{err}} = [2^n]^k\backslash\mathcal{A}^{\text{good}}$, then $|\mathcal{A}^{\text{good}}| \geq (1-\epsilon)\cdot 2^{kn}$, $|\mathcal{A}^{\text{err}}| \leq 2^{kn}\epsilon$.

Let $\mathcal{M}(\hat{z}_i', \hat{b}_i) = \{m \in \mathcal{M}^{\text{good}} : z_i'(m) = \hat{z}_i', b_i(m) = \hat{b}_i\}$. Intuitively, $\mathcal{M}(\hat{z}_i', \hat{b}_i)$ represent the set of outcomes (in $\mathcal{I}_c$) that will result in signal $\hat{b}_i$ being transmitted on edge b_i and signal $\hat{z}_i'$ being transmitted on edge z_i'. We define a function $\psi_i : [2^n] \to [2^n]$ as:

$$\psi_i(\hat{z}_i') = \arg\max_{\hat{b}_i} |\mathcal{M}(\hat{z}_i', \hat{b}_i)| \triangleq \hat{b}_{i,\hat{z}_i'} \tag{14}$$

Function ψ_i will be useful later, when we design the network codes in $\mathcal{I}$. Intuitively, in the absence of adversarial errors, ψ_i estimates the signal transmitted on edge b_i given that the signal transmitted on edge z_i' is $\hat{z}_i'$. In the following we analyze how often ψ_i will make a mistake. Define $\mathcal{M}_i^{\psi} = \{m \in \mathcal{M}^{\text{good}} : \psi_i(z_i'(m)) \neq b_i(m)\}$. Notice that $\mathcal{M}_i^{\psi}$ is the set of messages for which when transmitted by the source, ψ_i will make a mistake in guessing the signal transmitted on b_i. Lemma 2 and 3 analyze the size of $\mathcal{M}_i^{\psi}$.

Lemma 2: Let $\mathcal{M}(\hat{z}_i') = \{m \in \mathcal{M}^{good} : z_i'(m) = \hat{z}_i'\}$, then for any $m_1, m_2 \in \mathcal{M}(\hat{z}_i')$ such that $b_i(m_1) \neq b_i(m_2)$, there exists an element of $\mathcal{B}^{err}$ that will be decoded by terminal t to either m_1 or m_2.

Proof:

Consider any $m_1, m_2 \in \mathcal{M}(\hat{z}_i')$ such that $b_i(m_1) \neq b_i(m_2)$. Let $\boldsymbol{r}_1$ be the error pattern that changes the signal on x_i to be $x_i(m_2)$, and let $\boldsymbol{r}_2$ be the error pattern that changes the signal on y_i to be $y_i(m_1)$. Then if m_1 is transmitted by the source and $\boldsymbol{r}_1$ happens, node B_i will receive the same inputs $(x_i(m_2), y_i(m_1), z_i'(m_1) = z_i'(m_2))$ as in the situation that m_2 is transmitted and $\boldsymbol{r}_2$ happens. Therefore $b_i(m_1, \boldsymbol{r}_1) = b_i(m_2, \boldsymbol{r}_2)$, and so either $b_i(m_1, \boldsymbol{r}_1) \neq b_i(m_1)$ or $b_i(m_2, \boldsymbol{r}_2) \neq b_i(m_2)$ because by hypothesis $b_i(m_1) \neq b_i(m_2)$. Consider the first case that $b_i(m_1, \boldsymbol{r}_1) \neq b_i(m_1)$, then the tuple of signals $(b_1(m_1, \boldsymbol{r}_1), \ldots, b_k(m_1, \boldsymbol{r}_1)) = (b_1(m_1), \ldots, b_i(m_1, \boldsymbol{r}_1), \ldots, b_k(m_1))$ will be decoded by the terminal to message m_1 because of the fact that $m_1 \in \mathcal{M}^{\text{good}}$ which is correctly decodable under any error pattern $\boldsymbol{r} \in \mathcal{R}_{\mathcal{B}}$. Therefore this tuple of signals is an element of $\mathcal{B}^{\text{err}}$ since it is not equal to $\boldsymbol{b}(m_1) = (b_1(m_1), \ldots, b_k(m_1))$ and it is not equal to $\boldsymbol{b}(m)$, for any $m \neq m_1$, $m \in \mathcal{M}^{\text{good}}$, because otherwise it will be decoded by the terminal to m.

Similarly in the latter case that $b_i(m_2, \boldsymbol{r}_2) \neq b_i(m_2)$, then $(b_1(m_2, \boldsymbol{r}_2), \ldots, b_k(m_2, \boldsymbol{r}_2)) = (b_1(m_2), \ldots, b_i(m_2, \boldsymbol{r}_2), \ldots, b_k(m_2))$ is an element of $\mathcal{B}^{\text{err}}$ and will be decoded by the terminal to m_2. Therefore in both cases we are able to find an element of $\mathcal{B}^{\text{err}}$ that will be decoded by the terminal to either m_1 or m_2. □

Lemma 3: $|\mathcal{M}_i^{\psi}| \leq 2\epsilon \cdot 2^{kn}$.

Proof: We can partition $\mathcal{M}_i^{\psi}$ as

$$\mathcal{M}_i^{\psi} = \bigcup_{\hat{z}_i'} \left(\mathcal{M}(\hat{z}_i')\backslash\mathcal{M}(\hat{z}_i', \hat{b}_{i,\hat{z}_i'})\right),$$

and so

$$|\mathcal{M}_i^{\psi}| = \sum_{\hat{z}_i'} \left(|\mathcal{M}(\hat{z}_i')| - |\mathcal{M}(\hat{z}_i', \hat{b}_{i,\hat{z}_i'})|\right). \tag{15}$$

Consider an arbitrary $\hat{z}_i'$ and the set $\mathcal{M}(\hat{z}_i')$. We define an iterative procedure as follows. Initialize $\mathcal{W} := \mathcal{M}(\hat{z}_i')$. If there exist two messages $m_1, m_2 \in \mathcal{W}$ such that $b_i(m_1) \neq b_i(m_2)$, then delete both m_1, m_2 from $\mathcal{W}$. Repeat the operation until there does not exist $m_1, m_2 \in \mathcal{W}$ such that $b_i(m_1) \neq b_i(m_2)$.

After the procedure terminates, it follows that $|\mathcal{W}| \leq |\mathcal{M}(\hat{z}_i', \hat{b}_{i,\hat{z}_i'})|$, because otherwise by definition of $\hat{b}_{i,\hat{z}_i'}$ there must exist $m_1, m_2 \in \mathcal{W}$ such that $b_i(m_1) \neq b_i(m_2)$. Therefore at least $|\mathcal{M}(\hat{z}_i')| - |\mathcal{M}(\hat{z}_i', \hat{b}_{i,\hat{z}_i'})|$ elements are deleted from

$\mathcal{M}(\hat{z}_i')$. By Lemma 2, each pair of elements deleted corresponds to an element of $\mathcal{B}^{\text{err}}$. Also by Lemma 2 the elements of $\mathcal{B}^{\text{err}}$ corresponding to different deleted pairs are distinct. Summing over all possible values of $\hat{z}_i'$, it follows that the total number of deleted pairs is smaller than the size of $\mathcal{B}^{\text{err}}$:

$$\begin{aligned}\frac{1}{2}\sum_{\hat{z}_i'}\left(|\mathcal{M}(\hat{z}_i')| - |\mathcal{M}(\hat{z}_i', \hat{b}_{i,\hat{z}_i'})|\right) &\\ \le \sum_{\hat{z}_i'} \#\text{ of pairs deleted from } \mathcal{M}(\hat{z}_i') &\\ \le |\mathcal{B}^{\text{err}}| \le \epsilon \cdot 2^{kn}. & \qquad (16)\end{aligned}$$

Combining (15) and (16) we have $|\mathcal{M}_i^{\psi}| \le 2\epsilon \cdot 2^{kn}$. □

Next, let $\mathcal{M}(\hat{a}_i, \hat{b}_i) = \{m \in \mathcal{M}^{\text{good}} : a_i(m) = \hat{a}_i, b_i(m) = \hat{b}_i\}$. Intuitively, $\mathcal{M}(\hat{a}_i, \hat{b}_i)$ represent the set of outcomes (in $\mathcal{I}_c$) that will result in signal $\hat{b}_i$ being transmitted on edge b_i and signal $\hat{a}_i$ being transmitted on edge a_i. We define a function $\pi_i : [2^n] \to [2^n]$ as:

$$\pi_i(\hat{b}_i) = \arg\max_{\hat{a}_i} |\mathcal{M}(\hat{a}_i, \hat{b}_i)| \triangleq \hat{a}_{i,\hat{b}_i} \qquad (17)$$

Function π_i will be useful later for designing the network codes in $\mathcal{I}$. Intuitively, in the absence of adversarial errors, π_i estimates the signal transmitted on edge a_i given that the signal transmitted on edge b_i is $\hat{b}_i$. In the following we analyze how often will π_i make a mistake. Define $\mathcal{M}_i^{\pi} = \{m \in \mathcal{M}^{\text{good}} : \pi_i(b_i(m)) \ne a_i(m)\}$. Notice that $\mathcal{M}_i^{\pi}$ is the set of messages for which, when transmitted by the source, π_i will make a mistake in guessing the signal transmitted on a_i. Lemma 4 and 5 analyze the size of $\mathcal{M}_i^{\pi}$.

Lemma 4: Define $\mathcal{M}(\hat{a}_i) = \{m \in \mathcal{M}^{good} : a_i(m) = \hat{a}_i\}$. If $|\{b_i(m) : m \in \mathcal{M}(\hat{a}_i)\}| = L$, then there exist $(L-1)|\mathcal{M}(\hat{a}_i)|$ distinct elements of $\mathcal{B}^{err}$ such that each of them will be decoded by terminal t to some message $m \in \mathcal{M}(\hat{a}_i)$.

Proof: Assume for concreteness that $\{b_i(m) : m \in \mathcal{M}(\hat{a}_i)\} = \{\hat{b}_i^{(1)}, \ldots, \hat{b}_i^{(L)}\}$, then there exist L messages $m_1, \ldots, m_L \in \mathcal{M}(\hat{a}_i)$ such that $b_i(m_j) = \hat{b}_i^{(j)}$, $j = 1, \ldots, L$. For $j = 1, \ldots, L$, let $\boldsymbol{r}_j$ be the error pattern that changes the signal on z_i' to be $z_i'(m_j)$. Then if a message $m_0 \in \mathcal{M}(\hat{a}_i)$ is transmitted by the source and $\boldsymbol{r}_j$ happens, the node B_i will receive the same inputs $(x_i(m_0), y_i(m_0), z_i'(m_j))$ as in the situation that m_j is sent and no error happens. Therefore $b_i(m_0, \boldsymbol{r}_j) = \hat{b}_i^{(j)}$, and so $|\{\boldsymbol{b}(m_0, \boldsymbol{r}_j)\}_{j\in[L]}| = |\{b_i(m_0, \boldsymbol{r}_j)\}_{j\in[L]}| = L$. Since $m_0 \in \mathcal{M}^{\text{good}}$, it is correctly decodable under any error pattern $\boldsymbol{r} \in \mathcal{R}_{\mathcal{B}}$, and so all elements of $\{\boldsymbol{b}(m_0, \boldsymbol{r}_j)\}_{j\in[L]}$ will be decoded by the terminal to m_0. Except the element $\boldsymbol{b}(m_0)$, the other $L-1$ elements of $\{\boldsymbol{b}(m_0, \boldsymbol{r}_j)\}_{j\in[L]}$ are elements of $\mathcal{B}^{\text{err}}$. Sum over all $m_0 \in \mathcal{M}(\hat{a}_i)$ and the assertion is proved. □

Lemma 5: $|\mathcal{M}_i^{\pi}| \le 3\epsilon \cdot 2^{kn}$.

Proof: Define $\mathcal{A}_{i,1}^{\pi} = \{\hat{a}_i \in [2^n] : |\mathcal{M}(\hat{a}_i)| \le \frac{1}{2}2^{(k-1)n}\}$, and $\mathcal{A}_{i,2}^{\pi} = \{\hat{a}_i \in [2^n] \backslash \mathcal{A}_{i,1}^{\pi} : |\{b_i(m) : m \in \mathcal{M}(\hat{a}_i)\}| > 1\}$. Then define $\mathcal{M}_{i,1}^{\pi} = \{m \in \mathcal{M}^{\text{good}} : a_i(m) \in \mathcal{A}_{i,1}^{\pi}\}$, and $\mathcal{M}_{i,2}^{\pi} = \{m \in \mathcal{M}^{\text{good}} : a_i(m) \in \mathcal{A}_{i,2}^{\pi}\}$. Notice that by construction $\mathcal{A}_{i,1}^{\pi}$ and $\mathcal{A}_{i,2}^{\pi}$ are disjoint, and $\mathcal{M}_{i,1}^{\pi}$ and $\mathcal{M}_{i,2}^{\pi}$ are disjoint. We claim that,

$$\mathcal{M}_i^{\pi} \subset \mathcal{M}_{i,1}^{\pi} \cup \mathcal{M}_{i,2}^{\pi}. \qquad (18)$$

To prove the claim, consider any $m \in \mathcal{M}^{\text{good}}$ such that $m \notin \mathcal{M}_{i,1}^{\pi} \cup \mathcal{M}_{i,2}^{\pi}$. We will show that $\pi(b_i(m)) = a_i(m)$. Suppose for the sake of contradiction that $\pi(b_i(m)) = \hat{a}_i \ne a_i(m)$, then it follows that

$$\begin{aligned}|\mathcal{M}(\hat{a}_i, b_i(m))| &\overset{(a)}{>} |\mathcal{M}(a_i(m), b_i(m))|\\ &\overset{(b)}{=} |\mathcal{M}(a_i(m))| \overset{(c)}{>} \frac{1}{2}2^{(k-1)n}, \qquad (19)\end{aligned}$$

where (a) is due to the definition of π, (b) is due to the fact that $m \notin \mathcal{M}_{i,2}^{\pi}$ and (c) is due to the fact that $m \notin \mathcal{M}_{i,1}^{\pi}$. Let $\mathcal{M}(\hat{b}_i) = \{m' \in \mathcal{M}^{\text{good}} : b_i(m') = \hat{b}_i\}$, then $\mathcal{M}(\hat{a}_i, b_i(m)) \cup \mathcal{M}(a_i(m)) \subset \mathcal{M}(b_i(m))$. Since $\hat{a}_i \ne a_i(m)$, $\mathcal{M}(\hat{a}_i, b_i(m))$ and $\mathcal{M}(a_i(m))$ are disjoint, and it follows that $|\mathcal{M}(b_i(m))| \ge |\mathcal{M}(\hat{a}_i, b_i(m))| + |\mathcal{M}(a_i(m))| > 2^{(k-1)n}$. However, because $|\{(\hat{b}_1, \ldots, \hat{b}_k) \in [2^n]^k : \hat{b}_i = b_i(m)\}| = 2^{(k-1)n}$, by the pigeonhole principle there must exist two messages $m_1, m_2 \in \mathcal{M}(b_i(m))$ such that $\boldsymbol{b}(m_1) = \boldsymbol{b}(m_2)$. This is a contradiction since the terminal cannot distinguish m_1 from m_2. This proves $\pi(b_i(m)) = a_i(m)$ as well as (18).

We next bound the size of $\mathcal{M}_{i,1}^{\pi}$ and $\mathcal{M}_{i,2}^{\pi}$. For any $\hat{a}_i' \in \mathcal{A}_{i,1}^{\pi}$, by defnition $\{(\hat{a}_1, \ldots, \hat{a}_k) \in [2^n]^k : \hat{a}_i = \hat{a}_i'\} \backslash \{(\boldsymbol{a}(m) : m \in \mathcal{M}(\hat{a}_i')\}$ is a subset of $\mathcal{A}^{\text{err}}$ with size at least $\frac{1}{2}2^{(k-1)n}$. Therefore each element of $\mathcal{A}_{i,1}^{\pi}$ will contribute to at least $\frac{1}{2}2^{(k-1)n}$ distinct elements of $\mathcal{A}^{\text{err}}$. Hence $|\mathcal{A}_{i,1}^{\pi}| \cdot \frac{1}{2}2^{(k-1)n} \le |\mathcal{A}^{\text{err}}| \le \epsilon \cdot 2^{kn}$, and so $|\mathcal{A}_{i,1}^{\pi}| \le 2\epsilon \cdot 2^n$. It then follows that $|\mathcal{M}_{i,1}^{\pi}| \le \frac{1}{2}2^{(k-1)n}|\mathcal{A}_{i,1}^{\pi}| \le \epsilon \cdot 2^{kn}$.

By Lemma 4, each elements of $\mathcal{A}_{i,2}^{\pi}$ will contribute to at least $\frac{1}{2}2^{(k-1)n}$ distinct elements in $\mathcal{B}^{\text{err}}$. Therefore $|\mathcal{A}_{i,2}^{\pi}| \cdot \frac{1}{2}2^{(k-1)n} \le |\mathcal{B}^{\text{err}}| \le \epsilon \cdot 2^{kn}$, and so $|\mathcal{A}_{i,2}^{\pi}| \le 2\epsilon \cdot 2^n$. It then follows that $|\mathcal{M}_{i,2}^{\pi}| \le 2^{(k-1)n}|\mathcal{A}_{i,2}^{\pi}| \le 2\epsilon \cdot 2^{kn}$. Finally, by (18) we have $|\mathcal{M}_i^{\pi}| \le |\mathcal{M}_{i,1}^{\pi}| + |\mathcal{M}_{i,2}^{\pi}| \le 3\epsilon \cdot 2^{kn}$. □

We are now ready to prove Theorem 4. *Proof:* [Proof ("⇐" part of Theorem 4)] We show the achievability of rate k in $\mathcal{I}_c$ implies the achievability of unit rate in $\mathcal{I}$.

Let $\{\phi_e, \phi_t\}_{e\in\mathcal{E}}$ be the network error correction code of length n that achieves rate k in $\mathcal{I}_c$, with probability of error ϵ. We assume that in this code edge z_i simply relays the signal from edge a_i. This is without loss of generality because for any network code that needs to process the signal on edge a_i to obtain the signal to be transmitted on edge z_i, it is equivalent to relay the signal on edge z_i and perform the processing work at the head node of edge z_i.

Let $\mathcal{E}_{\mathcal{N}} \subset \mathcal{E}$ be the set of edges of the embedded graph $\mathcal{N}$. For the multiple-unicast problem $\mathcal{I}$, we define a length-n network code $\{\tau_e, \tau_{t_i} : e \in \mathcal{E}_{\mathcal{N}}, i \ \in [k]\}$ as follows.

$$\begin{aligned}\tau_e &= \phi_e, \qquad \forall e \in \mathcal{E}_{\mathcal{N}}\\ \tau_{t_i} &= \phi_{z_i} \circ \pi_i \circ \psi_i \circ \phi_{z_i'}, \qquad \forall i = 1, \ldots, k.\end{aligned}$$

where $\circ$ denotes function composition; ϕ_{z_i} and $\phi_{z_i'}$ are the encoding functions of edges z_i and z_i' in problem $\mathcal{I}_c$; ψ_i is defined in (14); and π_i is defined in (17). In the following we show that $\{\tau_e, \tau_{t_i} : e \in \mathcal{E}_{\mathcal{N}}, i \ \in [k]\}$ achieves unit rate in $\mathcal{I}$ with probability of error upper bounded by $6k\epsilon$.

In problem $\mathcal{I}$, let M_i be the random message associated with source s_i, then M_i, $i = 1, \ldots, k$ are i.i.d. uniformly distributed over $[2^n]$. Denote for short $\boldsymbol{M} = (M_1, \ldots, M_k)$, then slightly abusing notation we denote by $\tau_{t_i}(\boldsymbol{M})$ the output of the decoder τ_{t_i} under transmission $\boldsymbol{M}$. The probability of decoding error is given by

$$\Pr\{\bigcup_{i=1}^{k} \tau_{t_i}(\boldsymbol{M}) \neq M_i\},$$

where the probability is taken over the joint distribution of the random messages. Let $\boldsymbol{m} = (m_1, \ldots, m_k)$ be the realization of $\boldsymbol{M}$. We claim that if there exists a message m of problem $\mathcal{I}_c$ (not to be confused with $\boldsymbol{m}$, a message of $\mathcal{I}$) such that $m \in \mathcal{M}^{\text{good}}$, $m \notin \mathcal{M}_i^{\psi}$, $m \notin \mathcal{M}_i^{\pi}$ and $\boldsymbol{m} = \boldsymbol{z}(m)$, then $\tau_{t_i}(\boldsymbol{m}) = m_i$. To prove the claim, suppose $\boldsymbol{m} = \boldsymbol{z}(m)$ is transmitted in $\mathcal{I}$. Notice that all edges in $\mathcal{N}$ perform the same coding scheme in $\mathcal{I}$ as in $\mathcal{I}_c$, therefore for terminal node t_i, by invoking the function $\phi_{z_i'}$, it obtains $z_i'(m)$. Then by the definition of $\mathcal{M}_i^{\psi}$, it follows that $\psi_i(z_i'(m)) = b_i(m)$. And by the definition of $\mathcal{M}_i^{\pi}$, it follows that $\pi(\psi_i(z_i'(m))) = a_i(m)$. Finally since $\boldsymbol{m} = \boldsymbol{z}(m)$, it follows that $\phi_{z_i}(\pi(\psi_i(z_i'(m)))) = \phi_{z_i}(a_i(m)) = z_i(m) = m_i$.

Therefore $\tau_{t_i}(\boldsymbol{m}) = m_i$ if $\boldsymbol{m} \in \{\boldsymbol{z}(m) \in [2^n]^k : m \in \mathcal{M}^{\text{good}}, m \notin \mathcal{M}_i^{\psi}, m \notin \mathcal{M}_i^{\pi}\}$. The probability that τ_{t_i} makes an error, i.e., $\Pr\{\tau_{t_i}(\boldsymbol{M}) \neq M_i\}$, is upper bounded by the probability of the union of the following three events.

$$E_1 = \{\boldsymbol{M} = \boldsymbol{m} : \boldsymbol{m} \notin \{\boldsymbol{z}(m) \in [2^n]^k : m \in \mathcal{M}^{\text{good}}\}\}$$
$$E_2 = \{\boldsymbol{M} = \boldsymbol{m} : \boldsymbol{m} \in \{\boldsymbol{z}(m) \in [2^n]^k : m \in \mathcal{M}_i^{\psi}\}\}$$
$$E_3 = \{\boldsymbol{M} = \boldsymbol{m} : \boldsymbol{m} \in \{\boldsymbol{z}(m) \in [2^n]^k : m \in \mathcal{M}_i^{\pi}\}\}.$$

We upper bound the probability of E_1, E_2, E_3, respectively.

$$\begin{aligned}\Pr\{E_1\} &= 1 - \frac{|\{\boldsymbol{z}(m) : m \in \mathcal{M}^{\text{good}}\}|}{2^{kn}} \\ &\overset{(d)}{=} 1 - \frac{|\mathcal{M}^{\text{good}}|}{2^{kn}} \leq 1 - \frac{(1-\epsilon)\cdot 2^{kn}}{2^{kn}} = \epsilon, \quad (20)\end{aligned}$$

where (d) follows from the fact that $\boldsymbol{z}(m) = \boldsymbol{a}(m) \neq \boldsymbol{a}(m') = \boldsymbol{z}(m')$ for any $m, m' \in \mathcal{M}^{\text{good}}$, $m \neq m'$. By Lemma 3,

$$\Pr\{E_2\} = \frac{|\mathcal{M}_i^{\psi}|}{2^{kn}} \leq 2\epsilon. \quad (21)$$

And by Lemma 5, we have

$$\Pr\{E_3\} = \frac{|\mathcal{M}_i^{\pi}|}{2^{kn}} \leq 3\epsilon. \quad (22)$$

Combining (20), (21) and (22), it follows that

$$\Pr\{\tau_{t_i}(\boldsymbol{M}) \neq M_i\} \leq \Pr\{E_1\} + \Pr\{E_2\} + \Pr\{E_3\} \leq 6\epsilon.$$

Finally, by taking the union bound over the k terminals,

$$\Pr\{\bigcup_{i=1}^{k} \tau_{t_i}(\boldsymbol{M}) \neq M_i\} \leq 6\ k\epsilon.$$

Hence the probability of error is arbitrarily small and this establishes the achievability of unit rate in $\mathcal{I}$. □

The proof above suggests that the achievability of rate k with error probability ϵ in $\mathcal{I}_c$ implies the achievability of unit rate with error probability $6k\epsilon$ in $\mathcal{I}$. By setting $\epsilon = 0$, we generalize the result in Theorem 3 regarding the zero-error achievability as a special case.

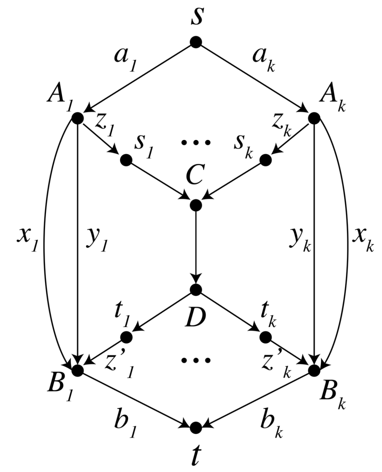


Fig. 6. Construction of $\mathcal{I}_c$ and $\mathcal{I}$. In $\mathcal{I}_c$, the source is s and the terminal is t. $\mathcal{B}$ includes all singleton sets of a single edge except $\{a_i\}$ and $\{b_i\}$, $i = 1, \ldots, k$. In $\mathcal{I}$, the source-destination pairs are $(s_i, t_i), i = 1, \ldots, k$. All edges have unit capacity.

Finally, we remark that our reduction has an operational aspect that from a code for $\mathcal{I}_c$ one can construct a code for $\mathcal{I}$. Indeed, using our reduction, to solve a multiple-unicast network coding problem, one may first reduce it to a unicast network error correction problem, then solve the latter, and finally use this solution to obtain a solution to the original multiple-unicast problem.

C. Asymptotic Achievability and Counter Example

Theorem 3 and 4 show that Construction 3 gives a reduction from multiple-unicast network coding to unicast network error correction in terms of zero-error achievability and in terms of achievability that allows vanishing error. In this subsection we show that the same construction does not provide a reduction in terms of asymptotic achievability (with vanishing error) by presenting a counter-example.

Theorem 5: There exists a multiple-unicast network coding problem $\mathcal{I}$ such that unit rate is not asymptotically achievable in $\mathcal{I}$, but in $\mathcal{I}_c$, which is the unicast network error correction problem constructed from $\mathcal{I}$ according to Construction 3, rate k is asymptotically achievable.

Proof: The construction of $\mathcal{I}$ and the corresponding $\mathcal{I}_c$ are shown in Figure 6. In $\mathcal{I}$, $\{(C, D)\}$ is a cut-set separating all sources from the terminals. Therefore by the cut-set bound, any rate $R > 1/k$ is not achievable in $\mathcal{I}$. This shows that unit rate is not asymptotically achievable in $\mathcal{I}$ if $k > 1$.

We prove the remaining part of the theorem by describing a network code with length n that achieves rate $k - k/n$ in $\mathcal{I}_c$. First divide the source message of rate $k - k/n$ into k pieces $M = (M_1, \ldots, M_k)$, such that M_i, $i = 1, \ldots, k$ are i.i.d. uniformly distributed over $[2^{n-1}]$. We denote $\phi_e : [2^{k(n-1)}] \to [2^n]$ as the encoding function[3] of edge e, which takes the source message M as input, and outputs the signal to be transmitted on e when there is no error in the network.

[3] This is called the *global encoding fuction* in the context of network coding.

For all $i = 1, \dots, k$, we let

$$\phi_{a_i}(M) = \phi_{x_i}(M) = \phi_{y_i}(M) = \phi_{z_i}(M) = \phi_{(s_i,C)}(M) = M_i$$

Furthermore, we let

$$\phi_{(C,D)}(M) = \phi_{(D,t_i)}(M) = \phi_{z'_i}(M) = \sum_{j=1}^{k} M_j, \quad \forall i = 1, \dots, k$$

where the summation is bitwise XOR. Note that the edges $a_i, x_i, y_i, z_i, (s_i, C), (C, D), (D, t_i), z'_i$ each have the capacity to transmit n bits. But we only require each of them to transmit $n - 1$ bits. Hence each edge reserves one unused bit.

Node B_i, by observing the (possibly corrupted) signals received from edges x_i, y_i, z'_i, performs error detection/correction in the following way. If the signal (of $n-1$ bits) received from x_i equals the signal received from y_i, forward the signal to edge b_i, and then transmit one bit of 0 using the reserved bit. Otherwise, forward the signal received from z'_i to b_i, and then transmit one bit of 1 using the reserved bit.

Finally, terminal t recovers the source message in the following way. Note that since there is only one corrupted edge in the network, for $i = 1, \dots, k$, the reserved bit on b_i equals 0 only if both x_i and y_i are not corrupted and it equals 1 only if either x_i or y_i is corrupted. Therefore, if the reserved bit on b_i is 0 then the remaining $n - 1$ bits received from b_i is exactly M_i. If the reserved bit on b_i is 0 for $i = 1, \dots, k$, then $M = (M_1, \dots, M_k)$ is decoded correctly at the terminal.

Otherwise, among $b_1, \dots, b_k$, there is at most one b_l such that the reserved bit on b_l is 1, because there is at most one corrupted edge. The remaining $n - 1$ bits received from b_l is exactly $\sum_{j=1}^{k} M_j$. This is because either x_l or y_l is corrupted and so z_l is not corrupted. Now note that the terminal t can decode M_i, $i \neq l$ correctly from the signals received from b_i, $i \neq l$. And so t can decode M_l correctly by evaluating $\sum_{j=1}^{k} M_j - \sum_{j=1, j\neq l}^{k} M_j = M_l$. Hence $M = (M_1, \dots, M_k)$ is decoded correctly at the terminal. This shows that rate $k - k/n$ is achievable in $\mathcal{I}_c$ and so rate k is asymptotically achievable in $\mathcal{I}_c$, completing the proof. □

Combining Theorem 4 with Theorem 5, it follows that in a unicast network error correction problem, the capacity in general is not achievable. Recall that by the definitions in Section II-C, we mean that the capacity is not *exactly* achievable, and by definition capacity is always *asymptotically* achievable.

Corollary 2: There exists a unicast network error correction problem for which the capacity is not (exactly) achievable.

Proof: The construction of the network error correction problem $\mathcal{I}_c$ is shown in Figure 6. By the cut-set bounds, the capacity of $\mathcal{I}_c$ is upper bounded by k. By Theorem 5, rate k is asymptotically achivable in $\mathcal{I}_c$, and so the capacity of $\mathcal{I}_c$ is k. Also by Theorem 5, unit rate in not achievable in $\mathcal{I}$, and so by Theorem 4, rate k is not achievable in $\mathcal{I}_c$. This shows that the capacity of $\mathcal{I}_c$ is not achievable. □

Corollary 2 suggests that although the (unicast) network error correction capacity is (by definition) asymptotically achievable, in general it is not achievable. This is in contrast to the scenario of network error correction with uniform $\mathcal{B}$, i.e., $\mathcal{B}$ is the collection of all subsets containing z links. In this case the network capacity can be achieved by linear codes. Unachievability of capacity is also studied for multiple-unicast networks [36] and sum networks [37]. For both cases, networks for which the capacity is not achievable are constructed using matriod theory.

VI. Conclusion

This paper presents reductions that map an arbitrary multiple-unicast network coding instance to a unicast secure network coding instance in which at most one link is eavesdropped, or a unicast network error correction instance in which at most one link is erroneous, such that a rate tuple is achievable in the multiple-unicast network coding instance if and only if a corresponding rate is achievable in the unicast secure network coding instance, or in the unicast network error correction instance. Our reductions show that solving seemingly very simple instances of the secure network coding problem or of the network error correction problem are in fact as hard as solving the multiple-unicast problem, a central open problem. Conversely, we show that an arbitrary unicast secure network coding instance in which at most one link is eavesdropped can be reduced back to a multiple-unicast network coding instance, implying an equivalence between the two problems. In addition, we show that the capacity of a unicast network error correction instance in general is not achievable.

Several problems are left open. It would be interesting to study whether the unicast secure network coding problem with more than one eavesdropped links can be reduced to the multiple-unicast network coding problem. Such a reduction, if exists, will imply that the unicast secure network coding problem with only one eavesdropped link is as hard as the general unicast secure network coding problem with an arbitrary number of eavesdropped links. We also leave open the possibility that the unicast network error correction problem can be reduced to the multiple-unicast network coding problem. Similarly, such a reduction would imply an equivalence between the two problems. Finally, it is an interesting fact that in reducing multiple-unicast network coding to unicast network error correction, our construction works for both zero-error achievability and achievability with vanishing error, but not for asymptotic achievability. A natural question is addressing the existence of a reduction for asymptotic achievability.

Appendix

A. Proof of Lemma 1

Our proof follows the same lines as the proof of the standard point-to-point channel coding theorem. The differences are that we need to translate the network code originally designed for $\mathcal{I}_s$ to a code for $\mathcal{I}$, as well as taking care of the multiple source-terminal pairs.

First consider problem $\mathcal{I}_s$. By hypothesis, let $\{\phi_e\}_{e\in\mathcal{E}}$ be the network code for $\mathcal{I}_s$. Denote the edge (B_i, s_i) by b'_i, and denote the signal transmitted on it by $\boldsymbol{b}'^n_i$. By construction we have the Markov chain $\boldsymbol{b}^n_i \to \boldsymbol{b}'^n_i \to \boldsymbol{d}^n_i$. Therefore it follows that $I(\boldsymbol{b}'^n_i; \boldsymbol{d}^n_i) \geq I(\boldsymbol{b}^n_i; \boldsymbol{d}^n_i) > n(1 - \epsilon)$. Denote the distribution of the signal $\boldsymbol{b}'^n_i$ (generated by node B_i) by $p_i(x)$,

$x \in \{0,1\}^n$, $i = 1, \ldots, k$. Note that the set of signals $\{\boldsymbol{b}_i'^n\}_{i=1}^k$ are independent because by construction they are generated by different nodes.

In problem $\mathcal{I}$ we simulate the same network code $\{\phi_e\}_{e \in \mathcal{E}}$ as in $\mathcal{I}_s$, and regard it as the inner code. We also generate the randomness according to the same distribution as in $\mathcal{I}_s$. Let x be a super symbol of n bits (i.e., same length as $\boldsymbol{b}_i'^n$), we independently generate k outer channel codes $\mathcal{C}_1, \ldots, \mathcal{C}_k$, each of 2^{mR} codewords and each codeword consists of m super symbols. Precisely, each of the 2^{mR} codewords of $\mathcal{C}_i$ is generated independently according to the distribution $p_i(x^m) = \prod_{j=1}^m p_i(x_j)$. Namely, each super symbol in the codeword is drawn independently according to the distribution of $\boldsymbol{b}_i'^n$. Notice that the overall length of the codeword is mn and the length of the network code for $\mathcal{I}_s$ is n. k messages $M_1, \ldots, M_k$ are chosen independently according to the uniform distribution: $\Pr\{M_i = w_i\} = 2^{-mR}$, $w_i = 1, \ldots, 2^{mR}$. Then the M_i-th codeword of $\mathcal{C}_i$, denoted by $X_i^m(M_i)$, is transmitted by invoking the inner network code for m times. Let Y_i^m be the output of terminal t_i using the inner code m times on network $\mathcal{N}$. The distribution of Y_i^m is statistically identical to that of $(\boldsymbol{d}_i^n)^m$ obtained by using the communication scheme for problem $\mathcal{I}_s$ m times. Indeed, in both cases $\mathcal{N}$ performs the same network code, generates randomness according to the same distribution, and receives the same input distribution by the codeword construction. In problem $\mathcal{I}$, t_i is a terminal node and it performs jointly typical decoding. Namely, the decoder at t_i declares that the $\hat{w}_i$-th codeword has been sent if: 1) $(X_i^m(\hat{w}_i), Y_i^m)$ is jointly typical, and 2) There is no other index $w' \neq \hat{w}_i$ such that $(X_i^m(w'), Y_i^m)$ is jointly typical. If no such $\hat{w}_i$ exists, an error is declared.

It remains to be shown that the probability of error $\Pr\{M_i \neq \hat{w}_i\}$ vanishes for all $i = 1, \ldots, k$ for an appropriate choice of R. As described above, Y_i^m in $\mathcal{I}$ is statistically the same as $(\boldsymbol{d}_i^n)^m$ in $\mathcal{I}_s$. Therefore it follows that $I(X_i; Y_i) = I(\boldsymbol{b}_i'; \boldsymbol{d}_i) > n(1-\epsilon)$. As a result, we can apply the standard error analysis for jointly typical decoding and standard probabilistic argument (refer to, for example, [11, Chapter 7.7]) to show that for $R = n(1-\epsilon)$ and $\delta > 0$, there exist a large enough length m, such that $\Pr\{M_i \neq \hat{w}_i\} < \delta$, for $i = 1, \ldots, k$. By the union bound, $\Pr\{\cup_{i=1}^k M_i \neq \hat{w}_i\} < k\delta$, which can be made arbitrarily small by choosing a small enough δ.

All in all, combining the inner and outer code, we have shown the existence of a coding scheme of length mn that satisfies the multiple-unicast network coding problem $\mathcal{I}$ with arbitrarily small error probability. The number of codewords for each source-terminal pair is $2^{mR} = 2^{mn(1-\epsilon)}$, and therefore the rate of the scheme (over length mn) is $1 - \epsilon$. This implies that unit rate is asymptotically achievable in $\mathcal{I}$.

REFERENCES

[1] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "On secure network coding with uniform wiretap sets," in *Proc. IEEE NetCod*, Jun. 2013, pp. 1–6.

[2] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "Single-source/sink network error correction is as hard as multiple-unicast," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2014, pp. 423–430.

[3] W. Huang, M. Langberg, and J. Kliewer, "Connecting multiple-unicast and network error correction: Reduction and unachievability," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2015, pp. 361–365.

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

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

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

[7] S. Jaggi *et al.*, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[8] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[9] R. Dougherty and K. Zeger, "Nonreversibility and equivalent constructions of multiple-unicast networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5067–5077, Nov. 2006.

[10] M. F. Wong, M. Langberg, and M. Effros, "On a capacity equivalence between multiple multicast and multiple unicast," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2013, pp. 1537–1544.

[11] T. Cover and J. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 2005.

[12] R. W. Yeung, S. R. Li, N. Cai, and Z. Zhang., *Network Coding Theory*. Breda, The Netherlands: Now Publishers, 2006.

[13] T. Chan and A. Grant, "Dualities between entropy functions and network codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4470–4487, Oct. 2008.

[14] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2002, p. 323.

[15] J. Feldman, T. Malkin, R. A. Servedio, and C. Stein, "On the capacity of secure network coding," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2004, pp. 1–10.

[16] S. Y. El Rouayheb and E. Soljanin, "On wiretap networks II," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 551–555.

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

[18] T. Chan and A. Grant, "Network coding capacity regions via entropy functions," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5347–5374, Sep. 2014.

[19] T. Chan and A. Grant, "Capacity bounds for secure network coding," in *Proc. Austral. Commun. Theory Workshop*, Feb. 2008, pp. 95–100.

[20] S. Jalali and T. Ho. (Dec. 2012). "On capacity region of wiretap networks." [Online]. Available: https://arxiv.org/abs/1212.3859

[21] T. Cui, T. Ho, and J. Kliewer, "On secure network coding with nonuniform or restricted wiretap sets," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 166–176, Jan. 2013.

[22] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "Reverse edge cut-set bounds for secure network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2014, pp. 106–110.

[23] N. Cai and R. W. Yeung, "A security condition for multi-source linear network coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2007, pp. 561–565.

[24] R. W. Yeung and N. Cai, "Network error correction, part I: Basic concepts and upper bounds," *Commun. Inf. Syst.*, vol. 6, no. 1, pp. 19–35, 2006.

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

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

[27] S. Jaggi *et al.*, "Resilient network coding in the presence of byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

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

[29] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, Jan. 2008.

[30] O. Kosut, L. Tong, and D. Tse, "Nonlinear network coding is necessary to combat general byzantine attacks," in *Proc. Allerton Conf. Commun., Control, Comput.*, Oct. 2009, pp. 593–599.

[31] D. Wang, D. Silva, and F. R. Kschischang, "Robust network coding in the presence of untrusted nodes," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4532–4538, Sep. 2010.

[32] O. Kosut, L. Tong, and D. N. C. Tse, "Polytope codes against adversaries in networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 3308–3344.

[33] S. Kim, T. Ho, M. Effros, and A. S. Avestimehr, "Network error correction with unequal link capacities," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1144–1164, Feb. 2011.

[34] P. H. Che, M. Chen, T. Ho, S. Jaggi, and M. Langberg, "Routing for security in networks with adversarial nodes," in *Proc. IEEE NetCod*, Jun. 2013, pp. 1–6.

[35] T. K. Dikaliotis, H. Yao, T. Ho, M. Effros, and J. Kliewer. (Nov. 2012). "Network equivalence in the presence of an eavesdropper." [Online]. Available: https://arxiv.org/abs/1211.4081

[36] R. Dougherty, C. Freiling, and K. Zeger, "Unachievability of network coding capacity," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2365–2372, Jun. 2006.

[37] B. K. Rai and B. K. Dey, "On network coding for sum-networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 50–63, Jan. 2012.

Wentao Huang received his B.Sc. in electrical engineering from Nanjing University of Posts and Telecommunications in 2008, the M.Sc. in communication and information systems from Shanghai Jiao Tong University in 2011, the Ph.D. in electrical engineering from the California Institute of Technology in 2017. He is now a security researcher at Snap Inc..

Tracey Ho received S.B., M.Eng and PhD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology. This work was done while she was with the California Institute of Technology. She is now with Second Spectrum Inc..

Michael Langberg (M'07–S'15) received his B.Sc. in mathematics and computer science from Tel-Aviv University in 1996, and his M.Sc. and Ph.D. in computer science from the Weizmann Institute of Science in 1998 and 2003 respectively. Between 2003 and 2006, he was a postdoctoral scholar in the Electrical Engineering and Computer Science departments at the California Institute of Technology, and between 2007 and 2012 he was in the Department of Mathematics and Computer Science at The Open University of Israel. Prof. Langberg is currently a professor in the Department of Electrical Engineering at the State University of New York in Buffalo.

Prof. Langberg's research addresses the algorithmic and combinatorial aspects of information in communication, management, and storage; focusing on the study of information theory, coding theory, network communication and network coding, big data in the form of succinct data representation, and probabilistic methods in combinatorics. Prof. Langberg was an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY during the years 2012-2015 and Editor of the IEEE Information Theory Society Newsletter during the years 2015-2018.

Jörg Kliewer (S'97–M'99–SM'04) received the Dipl.-Ing. (M.Sc.) degree in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 1993 and the Dr.-Ing. degree (Ph.D.) in electrical engineering from the University of Kiel, Germany, in 1999, respectively.

From 1993 to 1998, he was a research assistant at the University of Kiel, and from 1999 to 2004, he was a senior researcher and lecturer with the same institution. In 2004, he visited the University of Southampton, U.K., for one year, and from 2005 until 2007, he was with the University of Notre Dame, IN, as a Visiting assistant professor. From 2007 until 2013 he was with New Mexico State University, Las Cruces, NM, most recently as an associate professor. He is now with the New Jersey Institute of Technology, Newark, NJ, as an associate professor. His research interests span information and coding theory, graphical models, and statistical algorithms, which includes applications to networked communication and security, data storage, and biology.

Dr. Kliewer was the recipient of a Leverhulme Trust Award and a German Research Foundation Fellowship Award in 2003 and 2004, respectively. He was an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 until 2014, and since 2015 serves as an Area Editor for the same journal. He is also an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY since 2017 and a member of the editorial board of the IEEE Information Theory Newsletter since 2012.