entropy

MDPI

Article

A Numerical Study on the Capacity Region of a Three-Layer Wiretap Network

Jiahong Wu [1], Nan Liu [1,*] and Wei Kang [2]

[1] National Mobile Communications Research Laboratory, Southeast University, Nanjing 211189, China; jiahongwu@seu.edu.cn
[2] School of Information Science and Engineering, Southeast University, Nanjing 211189, China; wkang@seu.edu.cn
* Correspondence: nanliu@seu.edu.cn

Abstract: In this paper, we study a three-layer wiretap network including the source node in the top layer, N nodes in the middle layer and L sink nodes in the bottom layer. Each sink node recovers the message generated from the source node correctly via the middle layer nodes that it has access to. Furthermore, it is required that an eavesdropper eavesdropping a subset of the channels between the top layer and the middle layer learns absolutely nothing about the message. For each pair of decoding and eavesdropping patterns, we are interested in finding the capacity region consisting of $(N+1)$-tuples, with the first element being the size of the message successfully transmitted and the remaining elements being the capacity of the N channels from the source node to the middle layer nodes. This problem can be seen as a generalization of the secret sharing problem. We show that when the number of middle layer nodes is no larger than four, the capacity region is fully characterized as a polyhedral cone. When such a number is 5, we find the capacity regions for 74,222 decoding and eavesdropping patterns. For the remaining 274 cases, linear capacity regions are found. The proving steps are: (1) Characterizing the Shannon region, an outer bound of the capacity region; (2) Characterizing the common information region, an outer bound of the linear capacity region; (3) Finding linear schemes that achieve the Shannon region or the common information region.

Keywords: wiretap network; secret sharing; Shannon-type inequality; common information; generator matrix

check for updates

Citation: Wu, J.; Liu, N.; Kang, W. A Numerical Study on the Capacity Region of a Three-Layer Wiretap Network. *Entropy* **2023**, *25*, 1566. https://doi.org/10.3390/e25121566

Academic Editors: Jun Chen and Sadaf Salehkalaibar

Received: 29 September 2023
Revised: 10 November 2023
Accepted: 19 November 2023
Published: 21 November 2023

1. Introduction

The general concept of network coding was proposed by Ahlswede et al. [1] in 2000. They investigate the single-source multicast network coding problem where the message generated by the source node is required to be sent to multiple sink nodes through a noiseless network. In addition to routing, the nodes in the network can process the received information to utilize the full capacity of the network. In 2003, Li et al. [2] demonstrated through a vector space approach that linear network coding over a finite alphabet is sufficient for an optimal multicast. Independently, Koetter and Médard [3] developed an algebraic characterization of linear network coding via a matrix approach. A deterministic polynomial time algorithm for constructing a linear network code was later presented by Jaggi et al. [4]. For more background on network coding, a useful source is [5].

Cai and Yeung [6] proposed a wiretap network which incorporates information security with network coding [7–12]. In the wiretap network, a message is sent to possibly more than one legal user and needs to be protected from eavesdroppers, who may tap a set of channels in the network. More specifically, in the wiretap network, it is required that (i) all sink nodes can obtain the message correctly and (ii) the eavesdropper, who can access any one but not more than one eavesdropping set of communication channels, obtains nothing about the message. One solution of the wiretap network is that we send both the message and the random key via a linear scheme. In this way, an eavesdropper can only observe some linear combinations of the message and the random key, which is statistically independent of the message. On the other hand, every legal user can recover the message by canceling the effect of the random key.

The performance of a wiretap network scheme can be measured by the size of the message and the size of the random key. In [6], when the eavesdropper may choose to access any subset of channels of a fixed size, tight bounds were obtained. Some general bounds under arbitrary eavesdropping sets were obtained in [13], but may not be tight in general. Focusing on a simple network topology, Cheng [14] conducted a numerical study and showed the importance of characterizing the entropic region of six linear vector spaces. When focusing on the alphabet size for the existence of secure network codes, Guang and Yeung [15] developed a graph theoretic approach to improve the existing bound. Some variants of the wiretap network include universal secure multiplex network coding [16], secure network code for adaptive and active attacks [17], secure index coding [18], multiple linear combination security network coding [19], a secure network coding for multiple unicast traffic [20] and so on.

In this paper, we focus on a three-layer wiretap network where the source node in the top layer generates the random message and N nodes in the middle layer relay the information sent from the source node to the sink nodes in the bottom layer. The system constraint is that each sink node can recover the message correctly via the middle layer nodes it has access to. Furthermore, the eavesdropper, who can access any one but not more than one eavesdropping set of communication channels between the source and middle layer nodes, obtains nothing about the message. Such a three-layer wiretap network was initially formulated by Cai and Yeung [6] to show that the wiretap network contains secret sharing as a special case. When the eavesdropper may choose to access any subset of channels of *a fixed size*, they had obtained the optimal scheme. But when the eavesdropping pattern is an arbitrary one, the corresponding optimal scheme is unknown. Hence, the aim of our work is to explore arbitrary decoding and eavesdropping patterns and find the corresponding optimal schemes.

The fact that the three-layer wiretap network is a generalization of the secret sharing problem [21,22] can be seen as follows. A secret sharing scheme is a method to share a secret, with the help of random key, among a set of N participants such that the qualified sets of participants can recover the secret, while the forbidden sets of participants can know nothing about the secret. If any subset that is not a qualified set is a forbidden set, then we have the *complete access structure* scenario. The performance of a secret sharing scheme is the (average) information ratio between the size of the share and the size of the secret given an access structure. Since the number of different access structures is finite for a fixed number of participants N, following a case-by-case analysis, the optimal (average) information ratio can be found when $N \leq 4$ [23] for complete access structures. In the converse part, every secret sharing scheme is treated as a discrete probability distribution, thus Shannon-type inequalities, concluded from the non-negativeness of (conditional) entropy and (conditional) mutual information of any probability distribution, are used to provide a lower bound. In terms of achievability, linear schemes, where every codeword corresponds to a distribution of N shares, are sufficient to achieve the converse results.

For the complete access structure, when the number of participants is five, Jackson and Martin [24] had already handled most access structures. Recently, the work was moved further by introducing a new converse for linear schemes [25]. The technique behind this discovery is called the direct use of *common information* [26]. Nevertheless, the general results in the converse are far from tight, as discussed in [27].

Guided by the existing understanding of secret sharing, we let the number of middle layer nodes N be less than or equal to five. Unlike secret sharing, we should consider *incomplete access structures* for the three-layer wiretap network. That is, for some subsets of channels, whether it can obtain some information about the message, is not specified, which may be the circumstance when the eavesdropper has limited eavesdropping resources. In particular, when $N = 5$, there are a total of 74,496 different decoding and eavesdropping pattern pairs that need to be investigated.

Note that the secret sharing problem focuses on the optimal (average) information ratio, which is a scalar. To characterize the optimal (average) information ratio, one bound and one explicit scheme are needed. In the three-layer wiretap network, we consider the scenario that the channels between the top layer and the middle layer are heterogeneous, that is, the capacity of each channel may be different. For a given channel capacity vector, we are interested in the maximum amount of a message that can be securely and correctly transmitted to the sink nodes in the presence of the eavesdropper. To achieve this goal, we need to fully characterize the relationship between the size of the message and channel capacities. Such a relationship in fact formulates the capacity region, whose inner and outer bounds involve several linear schemes and inequalities.

The main contributions of this paper are the numerical results of the capacity region or linear capacity region of the three-layer wiretap network and the techniques we use to find them when the number of middle-layer nodes is no larger than five. We discuss them in detail as follows:

By exhaustive numerical experiments, we draw the conclusion that when the number of middle layer nodes is no larger than four, the capacity region is fully characterized as a polyhedral cone. However, when such a number is 5, there exist 274 decoding and eavesdropping patterns where we only find the linear capacity regions. On the other hand, the capacity regions for the other 74,222 cases are obtained.

The tools and techniques used in obtaining these results are as follows:

(1) Combine an existing bound for secret sharing or a wiretap network, which says that the size of the secret is upper bounded by the sum of the sizes of non-colluding shares, and Benson's algorithm, which is an existing projection algorithm, to obtain the Shannon region, which is the projection of the polyhedral cone formed by Shannon-type inequalities under the system constraints and, therefore, an outer bound of the capacity region;

(2) Modify Benson's algorithm to obtain the common information region, which adds common information for the linear achievability schemes and, therefore, is an outer bound of the linear capacity region;

(3) To obtain good linear schemes of the three-layer wiretap network, we propose the incremental kernel method (IKM), which is based on the existing Marten's method for linear secret sharing schemes but is more memory saving and efficient. However, the essence of the IKM algorithm is still a brute-force search, which fails in two cases. Then, we propose a manual method that uses Gaussian elimination to obtain optimal linear schemes for these two cases.

2. System Model

2.1. Problem Description

We study the model of a three-layer wiretap network, an example of which is shown in Figure 1.

Consider a directed acyclic multigraph with three layers of nodes: the top layer, the middle layer, and the bottom layer. The top layer consists of only one node, the source node, denoted as s. It generates a random message, M, which is uniformly distributed on the message set, $\mathcal{M}$.

The middle layer consists of N nodes, denoted as $u_1, u_2, \cdots, u_N$, and the source node connects to node u_n by an edge $e_n = (s, u_n)$ with capacity r_n, $n \in [1:N]$. On Channel e_n, an index taken from an alphabet $\mathcal{B}^{r_n}$ can be transmitted and is noiselessly received.

We assume that the bottom layer consists of L nodes, denoted as $t_1, \cdots, t_L$, and $\mathcal{D}_l \subseteq [1:N]$ denotes the indices of the nodes in the middle layer to which node t_l is connected. The channels between the middle layer nodes and the bottom layer nodes are of infinite capacity. All nodes in the bottom layer are considered sink nodes, i.e., they want to decode the message M, generated by the source node, without error. Let $\mathcal{A} = \{\mathcal{D}_1, \cdots, \mathcal{D}_L\}$, which we call the *decoding pattern*.

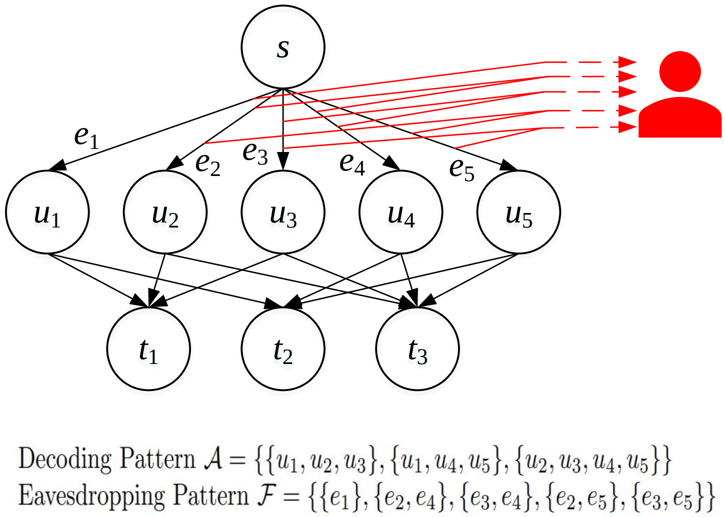
s
e_1
e_2
e_3
e_4
e_5
u_1
u_2
u_3
u_4
u_5
t_1
t_2
t_3
Decoding Pattern $\mathcal{A} = \{\{u_1, u_2, u_3\}, \{u_1, u_4, u_5\}, \{u_2, u_3, u_4, u_5\}\}$
Eavesdropping Pattern $\mathcal{F} = \{\{e_1\}, \{e_2, e_4\}, \{e_3, e_4\}, \{e_2, e_5\}, \{e_3, e_5\}\}$

Figure 1. An example of the system model.

There is also an eavesdropper who can access one of a collection of subsets of channels from the source node to the middle-layer nodes. More specifically, we assume that the *eavesdropping pattern* is $\mathcal{F} = \{\mathcal{E}_1, \cdots, \mathcal{E}_J\}$, and the eavesdropper may access the channels between the source node and the middle layer nodes in $\mathcal{E}_j$ for some $j \in [1{:}J]$. It is required that the eavesdropper knows absolutely nothing about the message M.

For a given $\mathbf{r} := (r_1, r_2, \ldots, r_N)$, we are interested in the maximum value of $H(M)$, i.e., the maximum amount of information that can be securely and correctly transmitted to the sink nodes in the presence of the eavesdropper.

2.2. Arbitrary Scheme and Capacity Region

A scheme for the above three-layer wiretap network consists of a set of random local encoding mapping of the source node, $\phi_n(\cdot)$: $\mathcal{M} \to \mathcal{B}^{r_n}$, which maps the value of the message into an index transmitted on the channel e_n. Note that in order to securely transmit message M in the presence of the eavesdropper, this mapping is random. We denote $Y_n = \phi_n(M)$. We note here that encoding at the middle-layer nodes are not needed, as the output channel is of infinite capacity and, furthermore, not susceptible to eavesdropping. In other words, it is sufficient for the middle-layer node u_n to simply forward Y_n onto its output channels, $n \in [1{:}N]$. The scheme $\{\phi_n(\cdot)$: $n = 1, \cdots, N\}$ must satisfy the following constraints:

1. Transmission constraint: for any $n \in [1{:}N]$, the entropy of Y_n is bounded by the capacity of the channel from the source node to u_n, i.e.,

$$H(Y_n) \le r_n, \quad \forall n \in [1 : N]. \tag{1}$$

2. Security constraint: for $\mathcal{E}_j$ in the eavesdropping pattern $\mathcal{F}$, denote $\{Y_n, e_n \in \mathcal{E}_j\}$ by $Y_{\mathcal{E}_j}$, and given the symbols $Y_{\mathcal{E}_j}$ accessed by the eavesdropper eavesdropping $\mathcal{E}_j$, we have $\Pr(M = m | Y_{\mathcal{E}_j} = y) = \Pr(M = m), \forall m \in \mathcal{M}$, i.e., the eavesdropper can know absolutely nothing about the message M. In other words,

$$H(M | Y_{\mathcal{E}_j}) = H(M), \quad \forall j \in [1 : J], \tag{2}$$

must be satisfied.

3. Decodability constraint: for the bottom layer node t_l, who has access to $Y_{\mathcal{D}_l}$, message M must be decoded without error, i.e.,

$$H(M|Y_{\mathcal{D}_l}) = 0, \quad \forall l \in [1:L]. \tag{3}$$

Since the maximum amount of information that can be correctly and securely transmitted from the source node to the destination nodes, i.e., $H(M)$, depends on the values of the channel capacities $r_n, n \in [1{:}N]$, we define the *capacity region*, denoted as $\mathcal{C}_{\mathcal{A},\mathcal{F}}$, of the three-layer wiretap network as the closure of the set of any $(N+1)$ dimension vector $(H(M), H(Y_1), \dots, H(Y_N))$ corresponding to a scheme that satisfies the transmission, security and decodability constraints.

2.3. Linear Scheme and Linear Capacity Region

We are also interested in linear schemes for the three-layer wiretap network. In defining a linear scheme, we let the alphabet $\mathcal{B}$ be a finite field GF(q), where q is a prime power. In other words, for the edge e_n with capacity r_n, r_n symbols in GF(q) can be transmitted correctly over e_n.

A linear scheme $(r, k, \mathbf{V}_1, \cdots, \mathbf{V}_N)$ consists of the following: (1) for some fixed positive integer r, the message set $\mathcal{M}$ is taken to be $\mathrm{GF}^r(q)$, i.e., message M can be written as r symbols in GF(q), i.e., $M = (M_1, \cdots, M_r)$; (2) for some fixed positive integer k, the randomness introduced by the source node to enable the secure delivery of the message to the destination nodes is denoted as K, which takes values in a uniform fashion in its alphabet $\mathcal{K}$, which is $\mathrm{GF}^k(q)$. This means that the randomness K can be written as k symbols in GF(q) as $K = (K_1, \cdots, K_k)$; (3) the source node performs linear coding, i.e., for each channel e_n, the linear coding coefficient is denoted by the matrix $\mathbf{V}_n$ of size $(r+k) \times r_n$, where each element is in GF(q). Hence, the vector transmitted on channel e_n is $Y_n = \begin{bmatrix} M_1 & \cdots & M_r & K_1 & \cdots & K_k \end{bmatrix} \mathbf{V}_n$, which consists of r_n elements and, therefore, does not exceed the capacity of the edge e_n. Thus, the transmission constraint, i.e., (1), is satisfied. The linear scheme must also satisfy the security constraint and the decodability constraint. Under the assumption of linear schemes, the security constraint (2) becomes

$$\mathrm{rank}\left(\begin{bmatrix} \mathbf{V}_M & \mathbf{V}_{\mathcal{E}_j} \end{bmatrix}\right) = \mathrm{rank}(\mathbf{V}_M) + \mathrm{rank}\left(\mathbf{V}_{\mathcal{E}_j}\right), \quad \forall j \in [1:J],$$

where rank $(\cdot)$ denotes the rank of a matrix, $\mathbf{V}_M$ is the matrix whose column vectors are associated with the message, i.e., $\mathbf{V}_M = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{k \times r} \end{bmatrix}$, and $\mathbf{V}_{\mathcal{E}_j}$ is the juxtaposition of $\mathbf{V}_n, n \in \mathcal{E}_j$. Under the assumption of linear schemes, the decodability constraint (3) becomes

$$\mathrm{rank}\left(\begin{bmatrix} \mathbf{V}_M & \mathbf{V}_{\mathcal{D}_l} \end{bmatrix}\right) = \mathrm{rank}\left(\mathbf{V}_{\mathcal{D}_l}\right), \quad \forall l \in [1:L],$$

where $\mathbf{V}_{\mathcal{D}_l}$ is the juxtaposition of $\mathbf{V}_n, n \in \mathcal{D}_l$.

We define the *linear capacity region*, denoted as $\mathcal{C}^l_{\mathcal{A},\mathcal{F}}$, of the three-layer wiretap network as the closure of the set of any $(N+1)$ dimension vector $(r, r_1, \dots, r_N)$ corresponding to a linear scheme that satisfies the transmission, security and decodability constraints.

3. Preliminaries

In order to characterize the capacity region (linear capacity region) for the three-layer wiretap network, we need to find its inner and outer bounds. For the capacity region, the outer bound we use is found via Shannon-type inequalities, and we call this outer bound the *Shannon region*. For the linear capacity region, the outer bound we use is found via common information, and we call this outer bound the *common information region*. The inner bound is found by explicit linear schemes. To make the paper self-contained, we first present some preliminaries on the Shannon region and the common information region.

3.1. The Shannon Region

The $N+1$ random variables of interest for any scheme for the three-layer wiretap network is $(M, Y_1, \cdots, Y_N)$. Note that for any probability distribution with $N+1$ discrete random variables, we can extract $2^{N+1}-1$ entropies, corresponding to $2^{N+1}-1$ different non-empty combinations of these random variables, and arrange them into a vector $\mathbf{h}$. Denote $\mathcal{H}_{N+1}$ as a $(2^{N+1}-1)$ dimension Euclidean space whose coordinates are labeled by $h_a, \emptyset \neq a \subseteq \mathcal{O} := \{M, Y_1, \ldots, Y_N\}$. The set of all such vectors $\mathbf{h} \in \mathcal{H}_{N+1}$ corresponding to a distribution is called the entropic region [28], denoted as Γ^*, and its closure is a convex cone [29]. In the three-layer wiretap network, the security constraint (2) and the decodability constraint (3) can be handled as homogeneous linear equations involving the coordinates from $\mathcal{H}_{N+1}$ only. More specifically, they can be expressed as

$$\mathcal{C}_1 = \{\mathbf{h} \in \mathcal{H}_{N+1} : h_{M,Y_{\mathcal{E}_j}} - h_{Y_{\mathcal{E}_j}} - h_M = 0, \forall j \in [1:J]\}, \tag{4}$$

$$\mathcal{C}_2 = \{\mathbf{h} \in \mathcal{H}_{N+1} : h_{M,Y_{\mathcal{D}_l}} - h_{Y_{\mathcal{D}_l}} = 0, \forall l \in [1:L]\}, \tag{5}$$

respectively.

It is known that the closure of the entropic region is not a polyhedral cone when the number of random variables is greater than or equal to four [30]. Hence, an easy-to-calculate outer bound is considered, i.e., $\Gamma_{|\mathcal{O}|}$. $\Gamma_{|\mathcal{O}|}$ is a polyhedral cone represented by the intersection of two categories of closed half-spaces, named also as Shannon-type inequalities [31]:

1. Non-decreasing: If $a \subseteq b \subseteq \mathcal{O}$, then $h_a \leq h_b$;
2. Submodular: $\forall a, b \subseteq \mathcal{O}, h_{a\cup b} + h_{a\cap b} \leq h_a + h_b$.

where $h_\emptyset$ is taken to be 0.

Recall the definition of the capacity region, where $N+1$ quantities are of interest, i.e., $(H(M), H(Y_1), \ldots, H(Y_N))$. As for the polyhedral cone $\Gamma_{|\mathcal{O}|} \cap \mathcal{C}_1 \cap \mathcal{C}_2$, we likewise care about the set of $N+1$ coordinates, i.e., $\mathbf{h}_\mathcal{O} := (h_M, h_{Y_1}, \ldots, h_{Y_N})$. To gain a more exact characterization, a suitable concept is illustrated as follows: a *projection* of a region $\mathcal{P}$ in $\mathbb{R}^n = \mathbb{R}^{n_1} \times \mathbb{R}^{n-n_1}$ onto its subspace of the first n_1 coordinates is

$$\text{proj}_{[1:n_1]}(\mathcal{P}) = \{\mathbf{x}_1 \in \mathbb{R}^{n_1} : \exists \mathbf{x}_2 \in \mathbb{R}^{n-n_1}, (\mathbf{x}_1^T, \mathbf{x}_2^T) \in \mathcal{P}\}. \tag{6}$$

After the above preparation, we introduce the concept of the *Shannon region*.

Definition 1 (Shannon Region)**.** *Given the decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$, the Shannon region $\mathcal{R}^s_{\mathcal{A},\mathcal{F}}$ of this three-layer wiretap network is the projection of the polyhedral cone $\Gamma_{|\mathcal{O}|}$ formed by Shannon-type inequalities under the security constraint $\mathcal{C}_1$ and the decodability constraint $\mathcal{C}_2$ onto the set of coordinates $\mathbf{h}_\mathcal{O}$, i.e., proj$_{\mathbf{h}_\mathcal{O}}(\Gamma_{|\mathcal{O}|} \cap \mathcal{C}_1 \cap \mathcal{C}_2)$.*

Any scheme for the three-layer wiretap network will give rise to the corresponding $N+1$ random variables $(M, Y_1, \cdots, Y_N)$, which must satisfy Shannon-type inequalities, the security constraint $\mathcal{C}_1$ and the decodability constraint $\mathcal{C}_2$. Hence, the Shannon region $\mathcal{R}^s_{\mathcal{A},\mathcal{F}}$ is an outer bound on the capacity region $\mathcal{C}_{\mathcal{A},\mathcal{F}}$.

3.2. The Common Information Region

The $N+1$ matrices of interest for any linear scheme for the three-layer wiretap network is $(\mathbf{V}_M, \mathbf{V}_1, \cdots, \mathbf{V}_N)$. In a linear scheme, both security and decodability constraints are related to the ranks of certain matrices. To find the rules that the ranks must obey, we firstly build a framework like for the entropic region, i.e., we extract $2^{N+1}-1$ ranks corresponding to $2^{N+1}-1$ different non-empty combinations of these $N+1$ matrices and arrange them into a vector $\mathbf{h} \in \mathcal{H}_{N+1}$. Then, the set of all such vectors corresponding to $N+1$ matrices is bounded by the so-called *linear rank inequalities* [32].

We note here that each matrix of $(\mathbf{V}_M, \mathbf{V}_1, \cdots, \mathbf{V}_N)$ can be viewed as a subset of a finite-dimension vector space over a finite field, or a set of basis vectors (column-wise) of a vector subspace. In fact, Shannon-type inequalities constrain not only the entropies of discrete random variables but also the ranks of subsets of a vector space. The non-decreasing property holds since one subset is contained within another subset. Furthermore, the submodular property follows by the dimension formula ([33], Appendix A.2), i.e., $\forall a, b \subseteq \mathcal{O}, \dim(\mathcal{V}_a) + \dim(\mathcal{V}_b) - \dim(\mathcal{V}_{a\cup b}) = \dim(\mathcal{V}_a \cap \mathcal{V}_b)$, which is greater than or equal to $\dim(\mathcal{V}_{a\cap b})$. Here, we use the convention that the vector subspace $\mathcal{V}_a$ is spanned by the column vectors of the matrix $\mathbf{V}_a$ and $\dim(\cdot)$ denotes the dimension of a vector subspace.

However, when the number of matrices is greater than or equal to four, there exist other linear rank inequalities, e.g., an Ingleton inequality [34] for the four-matrix case, twenty-four new inequalities for five matrices [32], and the ongoing work for six matrices [35,36]. To the best of our understanding, all of the above new linear rank inequalities can be derived from the tool named as common information, whose definition is given below.

Definition 2 (Common Information). *A random variable Z conveys the common information of the random variables X and Y if $H(Z|X) = H(Z|Y) = 0$ and $H(Z) = I(X;Y)$. We refer to these three equations as the common information constraint.*

In other words, the random variable Z encapsulates the mutual information of random variables X and Y. Unfortunately, given two random variables, it is not always possible to find a third one meeting the common information constraint. Nevertheless, in the context of vector spaces (or the random variables coming from them), common information does exist. More specifically, if X and Y are subspaces of a vector space, let Z be the intersection of X and Y, and Z will have the above three properties with the entropy term replaced by the dimension term. Finally, from the definition of a linear scheme in the three-layer wiretap network, where each random variable $a \in \mathcal{O}$ comes from the vector subspace $\mathcal{V}_a$, we may conclude that common information exists.

In order to obtain new linear rank inequalities besides Shannon-type inequalities for $|\mathcal{O}|$ vector subspaces, we can firstly introduce a new subspace $\mathcal{V}_Z$, which is the intersection of vector subspaces $\mathcal{V}_X, X \subseteq \mathcal{O}$ and $\mathcal{V}_Y, Y \subseteq \mathcal{O}$. Secondly, in the Euclidean space $\mathcal{H}_{|\mathcal{O}|+1}$, we build an intersection of three hyperplanes as follows:

$$\begin{aligned}\mathcal{C}_Z = \{\mathbf{h} \in \mathcal{H}_{|\mathcal{O}|+1} : & h_{Z,X} - h_X = h_{Z,Y} - h_Y = \\ & h_Z - h_X - h_Y + h_{X\cup Y} = 0\},\end{aligned} \tag{7}$$

which corresponds to the common information constraint. Finally, some inequalities constraining the polyhedral cone $\mathrm{proj}_{[1:2^{|\mathcal{O}|}-1]}(\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_Z)$, whose $2^{|\mathcal{O}|} - 1$ coordinates do not involve the letter Z, may be the desired new linear rank inequalities.

Using the above trick to obtain new linear rank inequalities and thus bound the linear capacity region of the three-layer wiretap network better, we introduce an auxiliary random variable Z that is the common information of random variables $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{O}$ and the corresponding intersection of three hyperplanes $\mathcal{C}_Z$. As for the polyhedral cone $\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_{1,2} \cap \mathcal{C}_Z$, where the hyperplanes in $\mathcal{C}_{1,2} := \mathcal{C}_1 \cap \mathcal{C}_2$ are extended in the Euclidean space $\mathcal{H}_{|\mathcal{O}|+1}$, we care about the set of $N+1$ coordinates $\mathbf{h}_{\mathcal{O}}$, similar to the case of the Shannon region. Still using the concept of projection, it follows that the polyhedral cone $\mathrm{proj}_{\mathbf{h}_{\mathcal{O}}}(\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_{1,2} \cap \mathcal{C}_Z)$ is an outer bound of the linear capacity region.

In obtaining the twenty-four new linear rank inequalities for five vector subspaces, it has been shown that different choices of common information lead to different inequalities [32]. Therefore, the polyhedral cone $\mathrm{proj}_{\mathbf{h}_{\mathcal{O}}}(\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_{1,2} \cap \mathcal{C}_Z)$ using a single choice of common information involves only part of the complete list of new linear rank inequalities, and thus may still not be tight for the linear capacity region. To obtain a tighter outer bound, a trivial idea is to build multiple projections corresponding to different choices of common information. Finally, the common information region is defined as the intersection of these projections, which is still an outer bound for the linear capacity region.

Before giving the formal definition of the common information region, some preparation is needed. Recall that $\mathcal{O}$ is the set of random variables essential to the three-layer wiretap network. Let an auxiliary random variable Z be the common information of random variables $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{O}$. We require that X and Y are disjointed, i.e., $X \cap Y = \emptyset$. In this way, we denote the number of different choices of common information for a fixed number of random variables $|\mathcal{O}|$ by $n_{|\mathcal{O}|}$. In particular, $n_6 = 301$. Then, we introduce the auxiliary random variable Z_i as the i-th common information of a choice of random variables $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{O}$ and the corresponding intersection of three hyperplanes is denoted by $\mathcal{C}_{Z_i}$, $i \in [1 : n_{|\mathcal{O}|}]$. Finally, the definition of the common information region is given below.

Definition 3 (The Common Information Region)**.** *Given the decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$, the common information region of this three-layer wiretap network is*

$$\mathcal{R}^c_{\mathcal{A},\mathcal{F}} := \cap_{i \in [1:n_{|\mathcal{O}|}]} proj_{\mathbf{h}_{\mathcal{O}}}(\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_{1,2} \cap \mathcal{C}_{Z_i}),$$

where each projection $_{\mathbf{h}_{\mathcal{O}}}(\Gamma_{|\mathcal{O}|+1} \cap \mathcal{C}_{1,2} \cap \mathcal{C}_{Z_i})$ is the projection of the polyhedral cone $\Gamma_{|\mathcal{O}|+1}$ formed by Shannon-type inequalities under the security constraint $\mathcal{C}_1$, the decodability constraint $\mathcal{C}_2$ and the common information constraint $\mathcal{C}_{Z_i}$ onto the set of coordinates $\mathbf{h}_{\mathcal{O}}$.

From our numerical experiments when the number of middle layer nodes is five, i.e., $|\mathcal{O}| = 6$, the equivalence between the common information region $\mathcal{R}^c_{\mathcal{A},\mathcal{F}}$, formed by single common information only, and the linear capacity region $\mathcal{C}^l_{\mathcal{A},\mathcal{F}}$ holds, and this is established by finding explicit linear schemes corresponding to all extreme directions of $\mathcal{R}^c_{\mathcal{A},\mathcal{F}}$.

4. Main Result

The main result of this paper is the characterization of the capacity region or the linear capacity region when the number of nodes in the middle layer is no larger than 5. It is summarized in the following:

(1) When the number of middle layer nodes $N \leq 4$ for any decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$, the capacity region of the three-layer wiretap network is found. Furthermore, the capacity region is achievable via linear schemes.

(2) When $N = 5$, out of a total of 74,496 different decoding and eavesdropping pattern pairs $(\mathcal{A}, \mathcal{F})$, the capacity region of 74,222 of them is found and achievable via linear schemes. For the remaining 274 $(\mathcal{A}, \mathcal{F})$ pairs, the linear capacity region is found.

(3) The detailed description of the capacity region and the corresponding achievable schemes are provided on GitHub and named SS-WN.

Note that the number of different decoding and eavesdropping pattern pairs is counted after the refinement by permutation, e.g., two pairs $(\mathcal{A} = \{\{1,2\}\}, \mathcal{F} = \{\{1\}\})$ and $(\mathcal{A} = \{\{1,2\}\}, \mathcal{F} = \{\{2\}\})$ are treated as the same one.

Remark 1. *From the converse point of view, we designed the projection algorithms to obtain the Shannon region and the common information region. From the achievability point of view, we proposed an efficient algorithm and a manual method to construct 7087 linear schemes in total. The reason why the number of linear schemes is less than the number of decoding and eavesdropping pattern pairs is because two different pairs may have the subset relationship, and thus a linear scheme for the pair with more restrictions also applies to the other pair.*

Remark 2. *Out of the 274 decoding and eavesdropping pattern pairs in which we only find the linear capacity regions, 17 $(\mathcal{A}, \mathcal{F})$ pairs are complete. Similarly, in the secret sharing problem, when the number of participants is five, optimal schemes that only restricted to the linear sense are proposed for eight complete access structures [25]. Such 8 access structures are included in the 17 $(\mathcal{A}, \mathcal{F})$ pairs.*

Proving the main result consists of the following steps:

1. Characterizing the *Shannon region*.
2. Characterizing the *common information region*.
3. Finding linear schemes that achieve the Shannon region or the common information region.

The methodology of the above three steps are given in Sections 5.1, 5.2 and 6, respectively. In Section 5.1, we combine the existing bounds for secret sharing or the wiretap network, i.e., Set Difference Bound [13,37], and existing projection algorithm, i.e., Benson's algorithm [38], to obtain the Shannon region. In Section 5.2, we modify Benson's algorithm to obtain the intersection of some polyhedral cones, which leads to the construction of the common information region. In Section 6.1, we propose the IKM algorithm to obtain the linear schemes for the three-layer wiretap network, which is more memory saving and efficient than the existing construction of secret sharing schemes [39]. Meanwhile, we design a manual method in Section 6.2 to tackle two cases that the IKM algorithm fails to solve.

5. Obtaining Explicit Forms of the Shannon Region and the Common Information Region

5.1. The Shannon Region

Recall that $\Gamma_{|\mathcal{O}|}$ is a finitely constrained polyhedral cone since the number of Shannon-type inequalities is finite for a fixed number, i.e., $|\mathcal{O}|$, of random variables. According to the Minkowski–Weyl Theorem for Cones ([40], Theorem 2.10), every finitely constrained polyhedral cone has two representations: a H-representation and a V-representation. The H-representation means that a polyhedral cone $\mathcal{P}$ can be represented by a system of m linear inequalities in n variables, e.g.,

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \geq \mathbf{0}\}, \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, which is called an inequality matrix in this paper. Meanwhile, such a polyhedral cone can also be represented by the non-negative linear combinations of t extreme directions, which can be treated as special vectors on the boundary of the cone, e.g.,

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{R}\lambda, \lambda \geq \mathbf{0}\}, \tag{9}$$

where $\mathbf{R} \in \mathbb{R}^{n \times t}$.

Then, we denote the projection of the polyhedral cone $\mathcal{P}$ onto the first n_1 coordinates by $\mathcal{Q}$. To tackle the projection $\mathcal{Q}$ of the original polyhedral cone $\mathcal{P}$ in the H-representation onto a small number of coordinates, one idea is to work directly in the projection space and the projection is incrementally built by successive refinement of an initial approximation $\mathcal{Q}'_0$. The difference in the relationship between the initial approximation and the true projection leads to two different projection algorithms, the Convex Hull Method [41–43] and Benson's algorithm [38,44]. We give an outline of Benson's algorithm in the following.

Benson's algorithm starts with an initial approximation that contains the true projection. For example, we can let some inequalities constraining the true projection constrain the initial approximation. Then, Benson's algorithm gradually adds new inequalities that constrain the true projection to the approximation. The essence of the iteration is to test whether an extreme direction of the approximation also belongs to the true projection, where the negative answer leads to an inequality that will be treated as a new inequality constraining the approximation. Meanwhile, the corresponding V-representation is updated since there are new inequalities. Again, since the dimension of the projection space is small, the conversion from H-representation to V-representation can be carried in practice [45].

Benson's algorithm has already included the method to construct the initial approximation by linear programming (LP). In our three-layer wiretap network, we can actually use some understandings of this problem to build an initial approximation that may be closer to the true projection, thus the number of iterations carried by any of the two algorithms may be smaller. We discuss this special trick according to the converse result in the following.

From the converse point of view, we can build the initial approximation for Benson's algorithm. When considering arbitrary wiretap sets in a general wiretap network, Cheng ([13], Corollary 1) proposed a type of inequality which works in the projection space and conveys the physical meaning that the size of the message is upper bounded by the sum of capacities of non-eavesdropped channels. We call this inequality the Set Difference Bound and illustrate it formally in the following.

Lemma 1 (Set Difference Bound)**.** *Given the decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$, for any decoding set $A \in \mathcal{A}$ and eavesdropping set $F \in \mathcal{F}$,*

$$H(M) \leq \sum_{i \in A - F} H(Y_i). \tag{10}$$

Remark 3. *Recall that the Shannon region is defined as the projection of the polyhedral cone formed by Shannon-type inequalities under the security constraint and the decodability constraint onto the set of coordinates $\mathbf{h}_{\mathcal{O}}$. Moreover, the proof of the Set Difference Bound is also derived from Shannon-type inequalities, the security constraint and the decodability constraint in the same Euclidean space $\mathcal{H}_{N+1}$. Finally, it follows that the Set Difference Bound forms an outer bound of the Shannon region and can be used to initialize Benson's algorithm.*

Remark 4. *In the secret sharing problem ([37], Proposition 2.2.4), the Set Difference Bound conveys the physical meaning that the size of the secret is upper bounded by the sum of sizes of non-colluding shares. In particular, for any complete access structure, the cardinality of the difference between a decoding set A and an eavesdropping set F can be one, so the Set Difference Bound is utilized to prove that the information ratio must be greater than or equal to one.*

In our numerical experiments, we adopt the Set Difference Bound to initialize Benson's algorithm to obtain the explicit forms of the Shannon region. When the number of nodes in the middle layer is less than or equal to five, it turns out that the initial approximation equals the true projection in 64,238 cases, which is nearly 86% of the total number of different decoding and eavesdropping pattern pairs.

The original Benson's algorithm is designed for multi-objective linear programming (MOLP) [38]. Meanwhile, the polyhedral projection problem is equivalent to MOLP, as stated in [46]. The reason is that the projection offers the full information of the sub-system related to the objectives of MOLP. For completeness, we rewrite Benson's algorithm for the polyhedral projection problem in Algorithm 1.

The initial approximation $\mathcal{Q}'_0$ is defined by the Set Difference Bound in the non-negative orthant and the corresponding V-representation is obtained. Since the Set Difference Bound is an outer bound of the Shannon region, we have that $\mathcal{Q}'_0$ contains the true projection $\mathcal{Q}$. We note here that the conversion from H-representation to V-representation can be carried by an existing Python package called pycddlib [45], due to the small size of the corresponding inequality matrix. Furthermore, the LP in Step 2 is solved by an existing commercial solver called Gurobi [47].

Basically, Benson's algorithm gradually contracts $\mathcal{Q}'_0$ by adding new inequalities that constrain the true projection, which are explored in Step 2. In the LP of Step 2, Algorithm 1, the non-negative variable $\mathbf{y}$ can be used to derive an inequality that constrains the original polyhedral $\mathcal{P}$ in the form of $\mathbf{y}^T\mathbf{A}\mathbf{x} \geq \mathbf{0}$. Furthermore, any feasible non-negative solution $\mathbf{y}'$ of the system of linear Equation (12) can be utilized to form an inequality that constrains the true projection $\mathcal{Q}$. More specifically, for any vector $\mathbf{x}_1 \in \mathcal{Q}$, according to the definition of projection (6), there exists a vector $\mathbf{x}_2 \in \mathbb{R}^{n-n_1}$ such that

$$\mathbf{y}'\mathbf{A}(\mathbf{x}_1^T, \mathbf{x}_2^T)^T = \mathbf{y}'^T\mathbf{A}_{[:,1:n_1]}\mathbf{x}_1 \geq 0. \tag{11}$$

Therefore, when the optimal value is less than 0, the inequality $\mathbf{y}'^T\mathbf{A}_{[:,1:n_1]}\mathbf{x}_1 \geq 0$ constraining the true projection can be added to make the intermediate approximation $\mathcal{Q}'_i$ strictly smaller, i.e., $\mathcal{Q}'_{i+1} \subsetneq \mathcal{Q}'_i$. In a polyhedral cone, the optimal value of a linear objective function may be infinitely small, so constraint (14) helps to obtain a bounded solution.

Algorithm 1 Benson's algorithm

```
Input: An initial approximation $\mathcal{Q}'_0$ and the original polyhedral cone $\mathcal{P}$.
Output: The projection $\mathcal{Q}$.
1. Let index $i = 0$.
2. Let the temporary set $\mathcal{S} = \emptyset$. For every extreme direction $\mathbf{d}$ of $\mathcal{Q}'_i$, the following LP is solved:
       $\min_{\mathbf{y}} \quad \mathbf{y}^T\mathbf{A}_{[:,1:n_1]}\mathbf{d}$
       $\text{s.t.} \quad \mathbf{y}^T\mathbf{A}_{[:,n_1+1:n]} = \mathbf{0}$      (12)
       $\mathbf{y} \geq \mathbf{0}$      (13)
       $\mathbf{1}^T\mathbf{y} = 1$      (14)
   If the optimal value is less than 0, the vector $\mathbf{y}^{\star T}\mathbf{A}_{[:,1:n_1]}$ is added to $\mathcal{S}$, where $\mathbf{y}^\star$ is the corresponding optimal solution.
3. If $\mathcal{S} = \emptyset$, the true projection $\mathcal{Q} = \mathcal{Q}'_i$ and the algorithm terminates. Otherwise, a new polyhedral cone $\mathcal{Q}'_{i+1}$ is formed, whose H-representation is the union of vectors in $\mathcal{S}$ and the whole inequality matrix of $\mathcal{Q}'_i$. Meanwhile, the V-representation of $\mathcal{Q}'_{i+1}$ is calculated. Then, let $i = i + 1$ and go back to Step 2.
```

The condition for determining the termination of Benson's algorithm is whether the approximation equals the true projection, where the equivalence means that each extreme direction of the approximation belongs to the true projection. Still based on the definition of projection (6), an extreme direction $\mathbf{d}$ of the approximation $\mathcal{Q}'_i$ is in the true projection if there exists a vector $\mathbf{x}_2 \in \mathbb{R}^{n-n_1}$ such that

$$\mathbf{A}_{[:,n_1+1:n]}\mathbf{x}_2 \geq -\mathbf{A}_{[:,1:n_1]}\mathbf{d}. \tag{15}$$

By Gale's Theorem [40] (Theorem 2.1), the existence of such $\mathbf{x}_2$ means that for any vector $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{y} \geq \mathbf{0}$ and $\mathbf{y}^T\mathbf{A}_{[:,n_1+1:n]} = \mathbf{0}$, the value $-\mathbf{y}^T\mathbf{A}_{[:,1:n_1]}\mathbf{d}$ must be less than or equal to zero. Coupled with the LP in Step 2, when the optimal value is greater than or equal to zero, we can see that the tested extreme direction $\mathbf{d}$ belongs to the true projection and the temporary set $\mathcal{S}$ is not updated.

Therefore, in Step 3, if the optimal value of every LP in Step 2 is greater than or equal to zero, Benson's algorithm terminates and outputs the true projection. Otherwise, the intermediate approximation $\mathcal{Q}'_{i+1}$ may still be strictly bigger than the true projection and thus further refinement is inevitable.

The main cost of Benson's algorithm is the LP in Step 2 and the representation conversion in Step 3. In practice, we run Benson's algorithm on a personal computer with an Intel Core i9-12900K Processor and 128 gigabytes of RAM. A total of 74,880 Shannon regions are obtained within an hour.

5.2. The Common Information Region

Recall that the common information region is defined as the intersection of many polyhedral cones, each of which is the projection of the corresponding original polyhedral cone. Meanwhile, in obtaining the explicit forms of the Shannon region, we have already utilized the existing Benson's algorithm to obtain the projection of the original polyhedral cone. Thus, a straightforward procedure to obtain the explicit forms of the common information region is to run Benson's algorithm with the initialization being the Shannon region multiple times to obtain each projection and finally combine all projections to build the intersection.

Such a procedure constructs the common information region in a parallel fashion since the multiple times of running Benson's algorithm are independent. However, we propose an algorithm that builds the common information region by running Benson's algorithm multiple times in a serial fashion where they are correlated. According to our numerical results, this algorithm is more efficient and is based on the following observation.

Lemma 2. *Let $\mathcal{Q}$ be the projection of the polyhedral cone $\mathcal{P}$. Benson's algorithm takes the initial approximation $\mathcal{Q}'_0$ and the original polyhedral cone $\mathcal{P}$ as an input and then actually outputs the intersection of the initial approximation $\mathcal{Q}'_0$ and the true projection $\mathcal{Q}$, i.e., $\mathcal{Q}'_0 \cap \mathcal{Q}$.*

Proof. Recall that in Benson's algorithm, new inequalities are gradually added to the approximation. We denote the intermediate approximation in the i-th iteration of Benson's algorithm by $\mathcal{Q}'_i$, and it follows that $\mathcal{Q}'_0 \supsetneq \mathcal{Q}'_1 \supsetneq \cdots \supsetneq \mathcal{Q}'_k$, where k is the number of iterations performed until termination. The proof of $\mathcal{Q}'_k = \mathcal{Q}'_0 \cap \mathcal{Q}$ is conducted by showing that the left-hand side (LHS) is inside the right-hand side (RHS) and vice versa.

The reason why the LHS is inside the RHS is that upon the termination of Benson's algorithm, every extreme direction of the polyhedral cone $\mathcal{Q}'_k$ is inside the true projection $\mathcal{Q}$, according to the discussion of (15). Meanwhile, we know that $\mathcal{Q}'_k \subsetneq \mathcal{Q}'_0$, as mentioned above, that is, each extreme direction of $\mathcal{Q}'_k$ also belongs to $\mathcal{Q}'_0$. Hence, we have that $\mathcal{Q}'_k \subseteq \mathcal{Q}'_0 \cap \mathcal{Q}$.

The reason why the RHS is inside the LHS is that in the whole procedure of Benson's algorithm, only inequalities that constrain the true projection $\mathcal{Q}$ are added to the initial approximation $\mathcal{Q}'_0$, according to the discussion of (11). In other words, the inequality matrix of the output $\mathcal{Q}'_k$ consists of the inequalities constraining $\mathcal{Q}'_0$ and some inequalities constraining $\mathcal{Q}$, then we have that $\mathcal{Q}'_0 \cap \mathcal{Q} \subseteq \mathcal{Q}'_k$. □

Remark 5. *Benson's algorithm requires that the initial approximation $\mathcal{Q}'_0$ contains the true projection $\mathcal{Q}$, i.e., $\mathcal{Q} \subseteq \mathcal{Q}'_0$. From the above lemma we can see that since the output of Benson's algorithm is the intersection of the initial approximation and the true projection, the equivalence between the output and the true projection holds.*

Remark 6. *In fact, the initial approximation $\mathcal{Q}'_0$ can be any finitely constrained polyhedral cone, that is, $\mathcal{Q}'_0$ may not contain the true projection $\mathcal{Q}$. In this case, if the rest of Benson's algorithm remains unchanged and when it terminates, the intersection of the initial approximation $\mathcal{Q}'_0$ and the true projection $\mathcal{Q}$ is the output, i.e., $\mathcal{Q}'_0 \cap \mathcal{Q}$.*

Since the common information region is defined as the intersection of many polyhedral cones, each of which is the projection of the corresponding original polyhedral cone, we can still adopt Benson's algorithm to obtain the common information region based on Lemma 2 in a serial fashion, that is, the output of the previous run of Benson's algorithm will be used as the input for the next run of Benson's algorithm. In the following, we use the shorthand BA to denote Benson's algorithm. Then, the formula $\mathcal{Q}' = \mathrm{BA}(\mathcal{Q}'_0, \mathcal{P})$ means that Benson's algorithm takes the initial approximation $\mathcal{Q}'_0$ and the original polyhedral cone $\mathcal{P}$ as an input, then the output is assigned to $\mathcal{Q}'$, which equals $\mathcal{Q}'_0 \cap \mathcal{Q}$ where $\mathcal{Q}$ is the projection of $\mathcal{P}$. We name our algorithm BA-CI, that is, Benson's Algorithm integrated with Common Information, which is illustrated as follows (Algorithm 2):

In the setup, when given the decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$, $n_{|\mathcal{O}|}$, original polyhedral cones $(\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(n_{|\mathcal{O}|})})$ are prepared, each of which is formed by Shannon-type inequalities and the intersection of hyperplanes $\mathcal{C}_{1,2} \cap \mathcal{C}_{Z_i}$ where $\mathcal{C}_{Z_i}$ is determined by the i-th common information, $i \in [1{:}n_{|\mathcal{O}|}]$.

Algorithm 2 BA-CI

```
Input: The Shannon region $\mathcal{R}^s_{\mathcal{A},\mathcal{F}}$ and $n_{|\mathcal{O}|}$ original polyhedral cones $(\mathcal{P}^{(1)}, \dots, \mathcal{P}^{(n_{|\mathcal{O}|})})$.
Output: The common information region $\mathcal{R}^c_{\mathcal{A},\mathcal{F}}$.
1. Let the intermediate polyhedral cone $\mathcal{T}^{(0)} = \mathcal{R}^s_{\mathcal{A},\mathcal{F}}$ and $i = 1$.
2. $\mathcal{T}^{(i)} = \mathrm{BA}(\mathcal{T}^{(i-1)}, \mathcal{P}^{(i)})$.
3. If $i = n_{|\mathcal{O}|}$, we have that $\mathcal{R}^c_{\mathcal{A},\mathcal{F}} = \mathcal{T}^{(i)}$ and the BA-CI algorithm terminates. Otherwise,
   let $i = i + 1$ and go back to Step 2.
```

Then, we run Benson's algorithm in series instead of the parallel implementation. More specifically, in the i-th iteration of the BA-CI algorithm, the information of the existing intersection of projections $\mathcal{R}^s_{\mathcal{A},\mathcal{F}} \cap \mathcal{Q}^{(1)} \cap \cdots \mathcal{Q}^{(i-1)}$ is actually utilized to accelerate the next run of Benson's algorithm, where $\mathcal{Q}^{(j)}$ is the projection of $\mathcal{P}^{(j)}$. The reason is that the initial approximation $\mathcal{T}^{(i-1)}$ is a subset of the Shannon region which is used in the straightforward procedure, and thus more extreme directions of $\mathcal{T}^{(i-1)}$ may already belong to the true projection $\mathcal{Q}^{(i)}$, which may lead to fewer iterations.

In the BA-CI algorithm, we have to implement Benson's algorithm in series to utilize the intermediate result, which seems inferior compared to the straightforward procedure. However, since LP is one of the main costs of Benson's algorithm, one trick is to implement different LPs in Step 2 of Benson's algorithm on different CPU threads concurrently, which also takes the full advantage of the CPU performance.

In practice, it takes us nearly 73 h to obtain the common information region for 74,496 different decoding and eavesdropping pattern pairs $(\mathcal{A}, \mathcal{F})$ when the number of middle layer nodes is five. On the other hand, the time taken by the straightforward procedure is nearly 116 h.

In the pursuit of the entropic region, Csirmaz [44] uses the notion of *Copy Lemma* [28] instead of common information and implements the straightforward procedure to obtain non-Shannon-type inequalities. In this way, different choices of copy strings can be analyzed since each projection is determined exactly. However, we focused on the final result only, i.e., the intersection of many projections, which leads to the discovery of the BA-CI algorithm.

6. Linear Achievable Schemes

Recall that the common information region, a polyhedral cone in the Euclidean space, is an outer bound of the linear capacity region for the three-layer wiretap network. If each extreme direction of the common information region has its corresponding linear scheme $(r, k, \mathbf{V}_1, \cdots, \mathbf{V}_N)$, we claim that the linear capacity region is the same as the common information region. The reason is that the definition of the V-representation of the common information region is consistent with the definition of the linear capacity region. Furthermore, when the Shannon region is identical to the linear capacity region, the capacity region is also obtained since the outer bound and the inner bound meet.

In obtaining the linear scheme for secret sharing, Marten had already proposed a method [39] that can be carried by a computer. Like secret sharing, the three-layer wiretap network also involves the security constraint and the decodability constraint. So, it turns out that Marten's method can also be used to obtain the linear scheme for the three-layer wiretap network. Moreover, we propose the IKM algorithm which shares the same core idea of Marten's method but is more memory saving and efficient. However, two cases remain stuck due to the large complexity that the IKM algorithm cannot handle. To tackle these two cases, we employ a manual method that is based on Gaussian elimination. In the following, we will discuss these two methods, i.e., the IKM algorithm and the manual method, in detail.

6.1. The IKM Algorithm

Note that a linear scheme $(r, k, \mathbf{V}_1, \cdots, \mathbf{V}_N)$ can also be treated as a linear code with generator matrix $\mathbf{V}$, where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_M & \mathbf{V}_1 & \cdots & \mathbf{V}_N \end{bmatrix}. \tag{16}$$

That is, every codeword corresponds to a distribution of the vectors transmitted on the channels between the source node and the middle layer nodes. More specifically, a codeword

$$(M_1, \cdots, M_r, Y_{1,1}, \cdots, Y_{1,r_1}, \cdots, Y_{N,r_N}) \in \mathrm{GF}(q)^{r+\sum_{i=1}^{i=N} r_i}$$

corresponds to a distribution of N vectors where the message is $(M_1, \cdots, M_r) \in \mathrm{GF}(q)^r$, the vector transmitted on channel e_1 is $(Y_{1,1}, \cdots, Y_{1,r_1}) \in \mathrm{GF}(q)^{r_1}$ and so on.

Thus, in a linear scheme, both security and decodability constraints are related to the ranks of submatrices of the generator matrix $\mathbf{V}$. Using the generator matrix formulation, Marten's method is based on the following observations:

(1) Recall that r is the size of the message and J is the cardinality of the eavesdropping pattern. Then, for any eavesdropping set $\mathcal{E}_j$, $j \in [1{:}J]$, consider r special codewords such that the components corresponding to $Y_{\mathcal{E}_j}$ are all-zero and the components corresponding to the message are non-zero. In this way, no matter what linear combinations are adopted, the eavesdropper cannot recover the message. More specifically, we arrange these rJ codewords row-wise into a matrix $\mathbf{G}$ and illustrate it via an example. Assume that the eavesdropping pattern $\mathcal{F} = \{\{1\}, \{2\}, \{3\}\}$ and an extreme direction $(r, r_1, r_2, r_3) = (2, 1, 1, 1)$ is considered, we have that

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & x_1 & x_2 \\ 0 & 1 & 0 & x_3 & x_4 \\ 1 & 0 & x_5 & 0 & x_6 \\ 0 & 1 & x_7 & 0 & x_8 \\ 1 & 0 & x_9 & x_{10} & 0 \\ 0 & 1 & x_{11} & x_{12} & 0 \end{bmatrix}. \tag{17}$$

In addition to the constant part, $\mathbf{G}$ also consists of the variable part that needs to be determined later. It follows that the matrix $\mathbf{G}$ has already satisfied the security constraint if we analyze the corresponding rank terms.

(2) The decodability constraint asks that each column vector of the matrix $\mathbf{V}_M$ corresponding to the message is a linear combination of the column vectors from the matrix $\mathbf{V}_{\mathcal{D}_l}$ where $\mathbf{V}$ is the generator matrix, $\mathcal{D}_l$ is the l-th decoding set of the decoding pattern and $l \in [1{:}L]$. The linear combination coefficients are arranged into a matrix $\mathbf{H} \in \mathrm{GF}(q)^{rL\times(r+\sum_{i\in[1:N]} r_i)}$ such that $\mathbf{V}\mathbf{H}^T = \mathbf{0}$. Note that in $\mathbf{H}$, for any decoding set $\mathcal{D}_l$, the components corresponding to $Y_{[1:N]-\mathcal{D}_l}$ are all-zero and the components corresponding to the message are non-zero. In this way, any sink node in the bottom layer can recover the message successfully via the linear combination. More specifically, we illustrate the matrix $\mathbf{H}$ via an example. Assume that the decoding pattern $\mathcal{A} = \{\{1,2\}, \{2,3\}\}$ and the extreme direction is $(r, r_1, r_2, r_3) = (2, 1, 1, 1)$, we have that

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & y_1 & y_2 & 0 \\ 0 & 1 & y_3 & y_4 & 0 \\ 1 & 0 & 0 & y_5 & y_6 \\ 0 & 1 & 0 & y_7 & y_8 \end{bmatrix}. \tag{18}$$

In addition to the constant part, $\mathbf{H}$ also consists of the variable part that needs to be determined later.

(3) Finally, we build a system of bilinear equations $\mathbf{GH}^T = \mathbf{0}$, where a feasible solution over $\mathrm{GF}(q)$ means that the matrix $\mathbf{G}$ also satisfies the decodability constraint. So, it turns out that the matrix $\mathbf{G}$ in its row echelon form can be treated as a generator matrix.

To discuss the above observations more rigorously, we introduce an index set $\mathcal{I}_0 := \{(i,j){:}i \in [1{:}J], j \in [1{:}r]\}$, where r is the size of the message and J is the cardinality of the eavesdropping pattern. Furthermore, let $\mathbf{e}^i$ be the i-th unit vector in $\mathrm{GF}(q)^r$ where the j-th coordinate equals 1 if $j = i$ and 0 if $j \neq i$. Recall that $\mathcal{E}_i$ is the i-th eavesdropping set of the eavesdropping pattern $\mathcal{F}$. Then, the security constraint leads to rJ codewords $(\mathbf{e}^j, \mathbf{c}^{i,j}), (i,j) \in \mathcal{I}_0$ such that

$$\mathbf{c}^{i,j}_{\mathcal{E}_i} = \mathbf{0}, \quad \forall (i,j) \in \mathcal{I}_0, \tag{19}$$

where $\mathbf{c}^{i,j}_{\mathcal{E}_i}$ is the juxtaposition of $\mathbf{c}^{i,j}_k, k \in \mathcal{E}_i$. On the contrary, every component of each $\mathbf{c}^{i,j}_{[1:N]-\mathcal{E}_i}$ is a variable. Actually, there is an equivalent relationship between the security constraint for a generator matrix and the existence of these rJ codewords. For more details, see ([39], Theorem 4.2).

Similarly, let an index set $\mathcal{I}_1 := \{(i,j){:}i \in [1{:}L], j \in [1{:}r]\}$ where L is the cardinality of the decoding pattern. Recall that $\mathcal{D}_i$ is the i-th decoding set of the decoding pattern $\mathcal{A}$. Then, the decodability constraint leads to a special matrix $\mathbf{H}$ formalized by rL row vectors $(\mathbf{e}^j, \mathbf{c}^{i,j}), (i,j) \in \mathcal{I}_1$ such that

$$\mathbf{c}^{i,j}_{[1:N]-\mathcal{D}_i} = \mathbf{0}, \quad \forall (i,j) \in \mathcal{I}_1. \tag{20}$$

On the contrary, every component of each $\mathbf{c}^{i,j}_{\mathcal{D}_i}$ is a variable. Actually, there is an equivalent relationship between the decodability constraint for a generator matrix and the existence of these rL row vectors. For more details, see ([39], Theorem 4.3).

Finally, a feasible solution of the system of bilinear equations $\mathbf{GH}^T = \mathbf{0}$ leads to a generator matrix that satisfies both the security and decodability constraints, which is summarized formally in ([39], Theorem 6.7).

Note that after finding a feasible solution, some row vectors in $\mathbf{G}$ may be linearly dependent due to exploiting the security constraint in this expanding form. Therefore, we can perform a row-wise Gaussian elimination to obtain a minimal set of basis row vectors of the generator matrix.

Based on Marten's method, if we assign values to the variables of the matrix $\mathbf{G}$, then $\mathbf{G}$ has already satisfied the security constraint. After that, $\mathbf{GH}^T = \mathbf{0}$ can be treated as rL systems of linear equations $\mathbf{GH}^T_{[i,:]}, i \in [1{:}rL]$, which plays the role of checking whether the matrix $\mathbf{G}$ satisfies the decodability constraint. More specifically, if each system of linear equations has a feasible solution, the decodability constraint of $\mathbf{G}$ holds. Otherwise, another choice of the variables in $\mathbf{G}$ needs to be considered.

In a finite field $\mathrm{GF}(q)$, the number of choices of the variables in $\mathbf{G}$ is finite for a fixed prime power q. For example, the matrix $\mathbf{G}$ in (17) has 12 different variables in total, which corresponds to q^{12} different choices of the variables. To prepare every choice, we can build the database row by row. That is, a list $\mathbf{E} = \{E_1, \ldots, E_{rJ}\}$ is introduced such that the i-th element E_i is the set of all choices of the variables in the i-th row vector of $\mathbf{G}$. For example, the first row vector of $\mathbf{G}$ in (17) has two variables, then E_1 has q^2 different two-dimensional arrays where each component is chosen from $\mathrm{GF}(q)$. Moreover, let an index array $\mathbf{J} = \{j_1, \ldots, j_{rJ}\}$ indicate the position in the database $\mathbf{E}$, i.e., j_i indicates the j_i-th array of the set E_i. Note that j_i is not greater than $|E_i|$, which is the cardinality of E_i. So, it turns out that the database $\mathbf{E}$ and the index array $\mathbf{J}$ can also be used to traverse all possible choices of the variables in $\mathbf{G}$, which is more memory saving compared to the tree storing all choices in [39].

Since the above preparation of the choices is row-wise, the procedure to test a choice for the decodability constraint is also carried row by row to avoid some unnecessary cases.

Similar to the database $\mathbf{E}$ for the matrix $\mathbf{G}$, a database $\mathbf{D}^0 = \{D_1^0, \ldots, D_{rL}^0\}$ for the matrix $\mathbf{H}$ is introduced, where each element D_i^0 stores all possible arrays corresponding to the variables in the i-th row vector of $\mathbf{H}$. Basically, the initial few steps are as follows:

(1) The first row vector $\mathbf{G}_{[1,:]}$ is fixed by the first array of the set E_1. Then, solve each linear equation $\mathbf{G}_{[1,:]}\mathbf{H}_{[i,:]}^T$ by exhausting the set D_i^0 of the database $\mathbf{D}^0$. Finally, the corresponding rL solution sets are saved in a new list $\mathbf{D}^1$;

(2) The second-row vector $\mathbf{G}_{[2,:]}$ is fixed by the first array of the set E_2. To solve each new system of linear equations $\mathbf{G}_{[1:2,:]}\mathbf{H}_{[i,:]}^T$, we can actually solve the linear equation $\mathbf{G}_{[2,:]}\mathbf{H}_{[i,:]}^T$ based on the previous solution set D_i^1. Finally, the corresponding rL solution sets are saved in a new list $\mathbf{D}^2$;

(3) If each set in the list $\mathbf{D}^2$ is not empty, i.e., each new system of linear equations $\mathbf{G}_{[1:2,:]}\mathbf{H}_{[i,:]}^T$ is solvable, the procedure continues to the third-row vector of the matrix $\mathbf{G}$;

(4) Otherwise, we assign the second array of the set E_2 to $\mathbf{G}_{[2,:]}$ and solve the corresponding rL linear equations again. Note that any choice of $\mathbf{G}$ consisting of the first array of the set E_1 and the first array of the set E_2 is ignored in the procedure. In this sense, we claim that this procedure can avoid some unnecessary cases.

We name the above procedure as the incremental kernel method, or IKM for short, where the word incremental means that we tackle the system of bilinear equations $\mathbf{G}\mathbf{H}^T = \mathbf{0}$ incrementally and the word kernel means that we actually solve the system of linear equations. The detail of the IKM algorithm is as follows.

In the IKM algorithm, Step 7 means that the choice of the variables in the first i rows of $\mathbf{G}$ is feasible and we will move on to the next row.

If the current choice is not feasible, we need to consider the next choice as in Step 9, which leads to two circumstances depending on the database for the current row vector of $\mathbf{G}$. In the first circumstance, where $j_i \leq |E_i|$, i.e., the set E_i has not been fully explored, we continue to solve rL linear equations for the current row vector. However, in another circumstance, where $j_i > |E_i|$, i.e., the set E_i has already been exhausted, we need to give up the i-th row vector temporarily. More specifically, in Step 12 we restore the index indicating the array for the i-th row vector to the initial position. Furthermore, in Step 13 we move to the previous row vector, for which the next choice is prepared as indicated in Step 14.

Finally, if the IKM algorithm (Algorithm 3) reaches Step 17, it means that the size of the finite field q needs to be larger or tighter converse results need to be found. Otherwise, there is a feasible solution for the matrix $\mathbf{G}$ and thus a linear scheme is constructed successfully.

Remark 7. *Our proposed IKM algorithm is essentially the same as the search algorithm proposed by Marten in [39] (Section 5) in terms of the core idea, since both these algorithms traverse the choices row by row. But, in terms of data structure, these two algorithm are different. That is, the search algorithm walks in the tree storing all possible choices of the matrix* $\mathbf{G}$*, while the IKM algorithm traverses the choices based on the database and the index array, which is more memory saving for a computer.*

Moreover, we propose two improvements as follows:

(1) Parallel computing can be integrated, e,g., split the database set E_1 into m parts and run on m threads of a CPU concurrently. In this way, more choices are explored per unit of time.

(2) We randomize the order of the arrays in each database set. In this way, we will obtain an average performance since we do not know which order is better beforehand.

The time complexity of the IKM algorithm depends on the number of choices of both $\mathbf{G}$ and $\mathbf{H}$. Furthermore, the IKM algorithm is useful in finding the optimal linear achievable scheme in almost all cases of the decoding and eavesdropping pattern pair $(\mathcal{A}, \mathcal{F})$. However, the IKM algorithm is stuck for weeks for two extreme directions due to the large size of $\mathbf{G}$ and the nature of the brute force search of this algorithm. The first case is $\mathcal{A} = \{A_1, A_2, A_3\}, \mathcal{F} = \{F_1, F_2, F_3, F_4, F_5\}$ with the extreme direction $\mathbf{d}_{\mathcal{O}} = (7, 3, 5, 5, 5, 5)$, where $A_1 = \{1,2,3\}, A_2 = \{1,4,5\}, A_3 = \{2,3,4,5\}$, $F_1 = \{1\}, F_2 = \{2,4\}, F_3 = \{3,4\}$, $F_4 = \{2,5\}$ and $F_5 = \{3,5\}$. The second case is $\mathcal{A} = \{A_1, A_2, A_3, A_4\}, \mathcal{F} = \{F_1, F_2, F_3, F_4, F_5, F_6\}$ with the extreme direction $(5,6,6,6,2,5)$, where $A_1 = \{1,2\}, A_2 = \{1,3\}, A_3 = \{2,3,4\}, A_4 = \{1,4,5\}, F_1 = \{1,4\}, F_2 = \{2,4\}, F_3 = \{3,4\}, F_4 = \{2,5\}, F_5 = \{3,5\}$ and $F_6 = \{4,5\}$.

Algorithm 3 Incremental Kernel Method (IKM)

```
Input: Two matrices G and H consisting of the constant part and the variable part, two
    corresponding databases E and $\mathbf{D}^0$ and an index array J for E.
Output: The matrix G full of constants or a warning.
Let each component of the index array J be 1 and $i$=1.
while $1 \le i \le rJ$ do
  if $j_i \le |E_i|$ then
    Assign the $j_i$-th array of the set $E_i$ to the variables of $\mathbf{G}_{[i,:]}$.
    Obtain $\mathbf{D}^i$ from $\mathbf{D}^{i-1}$.
    if $\forall k \in \mathbf{D}^i, k \neq \emptyset$ then
      $i = i + 1$.
    else
      $j_i = j_i + 1$.
    end if
  else
    $j_i = 1$.
    $i = i - 1$.
    $j_i = j_i + 1$.
  end if
end while
if $i = 0$ then
  Raise a warning.
else
  Output the matrix G.
end if
```

For these two cases, we resort to the manual method described below.

6.2. A Manual Method

Recall that a linear scheme $(r, k, \mathbf{V}_1, \cdots, \mathbf{V}_N)$ can be treated as a linear code with generator matrix $\mathbf{V} := \begin{bmatrix} \mathbf{V}_M & \mathbf{V}_1 & \cdots & \mathbf{V}_N \end{bmatrix}$, whose special codewords are utilized in Marten's method. Since Marten's method fails in the two cases mentioned above, we turn our attention to the original generator matrix $\mathbf{V}$. To build a generator matrix that satisfies the security and decodability constraints, two difficulties arise at first glance.

(1) How to choose an appropriate number of randomness, i.e., k;

(2) When k is fixed, the size of the generator matrix is also fixed, i.e., $(r+k) \times (r + \sum_{i \in [1:N]} r_i)$. Then, how to determine each component?

For the first difficulty, we can seek help from the converse part. Take the first case as an example, whose corresponding Shannon region is the same as the common information region. Recall that the Shannon region is the projection of the polyhedral cone $\Gamma_{|\mathcal{O}|}$ formed by Shannon-type inequalities under security constraint $\mathcal{C}_1$ and the decodability constraint $\mathcal{C}_2$ onto the set of coordinates $\mathbf{h}_{\mathcal{O}}$. Then, in the polyhedral cone $\Gamma_{|\mathcal{O}|} \cap \mathcal{C}_1 \cap \mathcal{C}_2$, we extract the integral extreme direction containing the sub-vector $\mathbf{d}_{\mathcal{O}}$ and it turns out to be unique, denoted by $\mathbf{d} = [7,3,10,5,12,8,13,5,12,8,13,10,13,13,13,5,12,8,13,9,16,12,16,9,16,12,16,13,16,16,16,5,12,8,13,9,16,12,16,9,16,12,16,13,16,16,16,10,13,13,13,13,16,16,16,13,16,16,16,16,16,16,16]$, in the usual binary order of $d_M, d_{Y_1}, d_{M,Y_1}, d_{Y_2}, \ldots, d_{\mathcal{O}}$. So the number of randomness can be set to $d_{\mathcal{O}} - d_M$, which is 9 in the first case.

In fact, finding a generator matrix, an arrangement of $N+1$ matrices $\mathbf{V}_M, \mathbf{V}_1, \ldots, \mathbf{V}_N$, whose $2^{N+1}-1$ rank terms correspond to the vector $\mathbf{d}$ is a representable *polymatroid* problem [32] (Section 5). Since the security and decodability constraints are related to the rank terms only, they are both already satisfied in the vector $\mathbf{d}$. Finally, to tackle the second difficulty, we construct the generator matrix corresponding to $\mathbf{d}$ based on the following two ideas:

(1) The $N+1$ matrices are constructed one by one. That is, when constructing the i-th matrix, $i \geq 2$, the actual representations constructed for the $i-1$ matrices are utilized to fulfill the rank terms of $\mathbf{d}_{[2^{i-1}:2^i-1]}$ simultaneously. More specifically, 2^{i-1} values calculated from $\mathbf{d}$ are needed, which are $d_{Y_{i-1}}, d_{M,Y_{i-1}} - d_M, d_{Y_1,Y_{i-1}} - d_{Y_1}, \ldots, d_{M,Y_1,\ldots,Y_{i-1}} - d_{M,Y_1,\ldots,Y_{i-2}}$, and we use the shorthand $d_{Y_{i-1}}, d_{Y_{i-1}|M}, d_{Y_{i-1}|Y_1}, \ldots, d_{Y_{i-1}|M,Y_1,\ldots,Y_{i-2}}$. Note that for any $A \subseteq \{M, Y_1, \ldots, Y_{i-2}\}$, the value $d_{Y_{i-1}|A}$ means that the space spanned by the column vectors of the matrix corresponding to $A \cup \{Y_{i-1}\}$ has $d_{Y_{i-1}|A}$ more basis vectors than the space spanned by the column vectors of the matrix corresponding to A.

(2) Gaussian elimination is exhaustively used to divide the matrix under construction into the constant part and the variable part. The final variable part is handled by human experience or a computer carrying the brute force search.

In Gaussian elimination, there are three types of *elementary row operations* on a matrix that does not alter its rank: swapping two rows, multiplying a row by a nonzero number and adding a multiple of one row to another row. It is similar for *elementary column operations*. For a generator matrix $\mathbf{V}$ of the three-layer wiretap network, we give the following trivial observation:

Lemma 3. *For any generator matrix $\mathbf{V}$ consisting of $N+1$ matrices $\mathbf{V}_M, \mathbf{V}_1, \ldots, \mathbf{V}_N$, there are two operations such that the corresponding $2^{N+1}-1$ rank terms of the changed form $\mathbf{V}'$ are the same as that of the original $\mathbf{V}$:*

1. *Elementary row operations on the whole matrix $\mathbf{V}$.*
2. *Elementary column operations on any matrix $\mathbf{V}_i, i \in \{M, 1, \ldots, N\}$.*

The proof is simple and directly follows from the fact that Gaussian elimination does not alter the rank of the matrix.

Remark 8. *It is known that when using elementary row (column) operations, a matrix can always be transformed into the reduced row (column) echelon form, which is unique and consists of some fixed constants. We use these two operations in Lemma 3 to set some components of the generator matrix to be constants, which makes the later construction easier since we can rely on the existing actual representation.*

Take the first case as an example, we illustrate our construction procedure. Due to space limitation, we only show the first four matrices, which are in (21):

$$\left[\begin{array}{ccccccc|ccc|ccccc|ccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & x_2 & z_1 & z_2 & z_3 & z_4 & z_5 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & x_4 & z_6 & z_7 & z_8 & z_9 & z_{10} \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_5 & x_6 & z_{11} & z_{12} & z_{13} & z_{14} & z_{15} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_7 & x_8 & z_{16} & z_{17} & z_{18} & z_{19} & z_{20} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_9 & x_{10} & z_{21} & z_{22} & z_{23} & z_{24} & z_{25} \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{11} & x_{12} & z_{26} & z_{27} & z_{28} & z_{29} & z_{30} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_{13} & x_{14} & z_{31} & z_{32} & z_{33} & z_{34} & z_{35} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & y_1 & y_2 & a_1 & a_2 & a_3 & b_1 & b_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & y_3 & y_4 & a_4 & a_5 & a_6 & b_3 & b_4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & y_5 & y_6 & a_7 & a_8 & a_9 & b_5 & b_6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]. \tag{21}$$

The first matrix $\mathbf{V}_M$ is always the identity matrix $\mathbf{I}_{d_M}$ stacked vertically with an all-zero matrix, since for any generator matrix meeting the integral extreme direction $\mathbf{d}$, it can be transformed into the reduced row echelon form by elementary row operations. We have $d_{Y_1|M} = 3$ for the second matrix $\mathbf{V}_1$, where the rank of the submatrix formed by the last nine rows must be 3 since $\mathbf{V}_M$ is fixed. Then, by elementary row operations, we have an $\mathbf{I}_3$ and the others are all-zero.

Since the first two matrices are fixed and $d_{Y_2|M,Y_1} = 3$, in the third matrix $\mathbf{V}_2$ the rank of the submatrix formed by the last six rows is 3. We use both elementary row and column operations to get an $\mathbf{I}_3$, while the submatrix above $\mathbf{I}_3$ and the other elements of the last six rows are all zero. As all four values for $\mathbf{V}_2$ need to be filled, from $d_{Y_2|M} = 5$, we have the rank of the last nine rows to be 5, then the y block is full rank and needs to be determined later. Since $d_{Y_2|Y_1} = 5$, it is similar for the x block.

Next, consider the fourth matrix $\mathbf{V}_3$, via $d_{Y_3|M,Y_1,Y_2} = 0$. We leave the last three rows to be all zero and no elementary row operation can be implemented in this matrix since the three matrices constructed before are fixed. Nevertheless, as $d_{Y_3|M,Y_1} = 3$, we use elementary column operations to obtain an $\mathbf{I}_3$ concatenated with an $\mathbf{0}_{3\times 2}$ above the last three rows, and no further operation can be performed in $\mathbf{V}_3$. Since $d_{Y_3|Y_1,Y_2} = 5$, we have the matrix of size 7×7, which is a concatenation of the x block and the z block, that is full rank. We can learn that the z block alone is full rank, as $d_{Y_3} = 5$ and by the non-decreasing property of rank terms, it follows that $d_{Y_3|Y_1} = d_{Y_3|Y_2} = 5$ is already satisfied. From $d_{Y_3|M} = 5$, we need the b block to be full rank. For $d_{Y_3|M,Y_2} = 1$, the matrix which is the concatenation of the y block, the a block and the b block needs to be full rank.

The other matrices are constructed in a similar way, i.e., leaving the final variable part to be determined by trial and error. Sometimes, these variables can be found with the assistance of a computer, which carries out a brute force search.

Remark 9. *The first idea that constructs the matrices one by one is learned from [32] Section 5), where the authors faced the problem of verifying tremendous extreme directions and they handled the i-th matrix by 2^{i-1} values in a combinatorial style without actual numerical vector construction. On the other hand, our method uses the actual matrices constructed before for the matrix under construction. Moreover, by Gaussian elimination, we determine the constant part without loss of generality since the reduced row (column) echelon form is unique. The variable part is decided at last to satisfy all elements of the vector* $\mathbf{d}$ *simultaneously, by hand or computer with brute force search.*

Remark 10. *There is a computational framework provided by [48] where group theoretic techniques for combinatorial generation are utilized. However, we were not able to get any results for weeks. In contrast, we used the manual method to tackle the two cases within four days. Still, when the number of matrices $|\mathcal{O}|$ is larger, we do not think this manual method is efficient due to the number of rank terms growing exponentially. Thus, we are not sure if the manual method would still work for $|\mathcal{O}| \geq 7$, i.e., the number of middle-layer nodes is six.*

7. Conclusions

In this paper, we have studied the capacity region of a three-layer wiretap network that is a generalization of the secret sharing problem. By numerical experiments, we find that the capacity regions are explicit polyhedral cones when the number of middle-layer nodes is less than or equal to four. There are 274 non-tight decoding and eavesdropping pattern pairs when the number of middle-layer nodes is five, where we only obtain the linear capacity regions. The capacity regions for the other 74,222 pairs are found. In obtaining converse results, we combine an existing bound for secret sharing or the wiretap network and Benson's algorithm to obtain the Shannon region, which is an outer bound of the capacity region. Moreover, we modify Benson's algorithm to obtain the common information region, which is an outer bound of the linear capacity region. In achievability, we propose the IKM algorithm and a manual method to obtain the linear schemes.

Author Contributions: Conceptualization, J.W., N.L. and W.K.; methodology, J.W., N.L. and W.K.; formal analysis, J.W., N.L. and W.K.; writing—original draft preparation, J.W.; writing—review and editing, J.W., N.L. and W.K.; supervision, N.L. and W.K.; funding acquisition, N.L. and W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation of China under Grants 61971135 and 62071115, and the Research Fund of National Mobile Communications Research Laboratory, Southeast University (No. 2023A03).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Ahlswede, R.; Cai, N.; Li, S.Y.; Yeung, R. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
2. Li, S.Y.R.; Yeung, R.W.; Cai, N. Linear network coding. *IEEE Trans. Inf. Theory* **2003**, *49*, 371–381. [CrossRef]
3. Koetter, R.; Médard, M. An algebraic approach to network coding. In Proceedings of the 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252), Washington, DC, USA, 29–29 June 2001.
4. Jaggi, S.; Sanders, P.; Chou, P.A.; Effros, M.; Egner, S.; Jain, K.K.; Tolhuizen, L. Polynomial time algorithms for multicast network code construction. *IEEE Trans. Inf. Theory* **2005**, *51*, 1973–1982. [CrossRef]
5. Yeung, R.W. *Information Theory and Network Coding*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
6. Cai, N.; Yeung, R.W. Secure Network Coding on a Wiretap Network. *IEEE Trans. Inf. Theory* **2011**, *57*, 424–435. [CrossRef]
7. Rouayheb, S.Y.E.; Soljanin, E.; Sprintson, A. Secure Network Coding for Wiretap Networks of Type II. *IEEE Trans. Inf. Theory* **2009**, *58*, 1361–1371. [CrossRef]
8. Silva, D.; Kschischang, F.R. Universal Secure Network Coding via Rank-Metric Codes. *IEEE Trans. Inf. Theory* **2008**, *57*, 1124–1135. [CrossRef]
9. Cui, T.; Ho, T.; Kliewer, J. On Secure Network Coding With Nonuniform or Restricted Wiretap Sets. *IEEE Trans. Inf. Theory* **2013**, *59*, 166–176. [CrossRef]
10. Hayashi, M.; Cai, N. Secure Non-Linear Network Code Over a One-Hop Relay Network. *IEEE J. Sel. Areas Inf. Theory* **2020**, *2*, 296–305. [CrossRef]
11. Zhou, H.; Gamal, A.E. Network Information Theoretic Security with Omnipresent Eavesdropping. *IEEE Trans. Inf. Theory* **2021**, *67*, 8280–8299. [CrossRef]
12. Guang, X.; Yeung, R.W.; Fu, F.W. Local-Encoding-Preserving Secure Network Coding. *IEEE Trans. Inf. Theory* **2020**, *66*, 5965–5994. [CrossRef]
13. Cheng, F.; Yeung, R.W. Performance Bounds on a Wiretap Network with Arbitrary Wiretap Sets. *IEEE Trans. Inf. Theory* **2014**, *60*, 3345–3358. [CrossRef]
14. Cheng, F.; Tan, V.Y.F. A Numerical Study on the Wiretap Network with a Simple Network Topology. *IEEE Trans. Inf. Theory* **2016**, *62*, 2481–2492. [CrossRef]
15. Guang, X.; Yeung, R.W. Alphabet Size Reduction for Secure Network Coding: A Graph Theoretic Approach. *IEEE Trans. Inf. Theory* **2018**, *64*, 4513–4529. [CrossRef]
16. Matsumoto, R.; Hayashi, M. Universal Secure Multiplex Network Coding with Dependent and Non-Uniform Messages. *IEEE Trans. Inf. Theory* **2017**, *63*, 3773–3782. [CrossRef]
17. Cai, N.; Hayashi, M. Secure Network Code for Adaptive and Active Attacks with No-Randomness in Intermediate Nodes. *IEEE Trans. Inf. Theory* **2020**, *66*, 1428–1448. [CrossRef]
18. Mojahedian, M.M.; Aref, M.R.; Gohari, A. Perfectly Secure Index Coding. *IEEE Trans. Inf. Theory* **2017**, *63*, 7382–7395. [CrossRef]
19. Bai, Y.; Guang, X.; Yeung, R.W. Multiple Linear-Combination Security Network Coding. *Entropy* **2023**, *25*, 1135. [CrossRef]
20. Agarwal, G.K.; Cardone, M.; Fragouli, C. On Secure Network Coding for Multiple Unicast Traffic. *IEEE Trans. Inf. Theory* **2019**, *66*, 5204–5227. [CrossRef]
21. Blakley, G.R. Safeguarding cryptographic keys. In Proceedings of the 1979 International Workshop on Managing Requirements Knowledge (MARK), New York, NY, USA, 4–7 June 1979; pp. 313–318.
22. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
23. Stinson, D.R. An explication of secret sharing schemes. *Des. Codes Cryptogr.* **1992**, *2*, 357–390. [CrossRef]
24. Jackson, W.A.; Martin, K.M. Perfect Secret Sharing Schemes on Five Participants. *Des. Codes Cryptogr.* **1996**, *9*, 267–286. [CrossRef]
25. Farràs, O.; Kaced, T.; Martín, S.; Padro, C. Improving the Linear Programming Technique in the Search for Lower Bounds in Secret Sharing. *IEEE Trans. Inf. Theory* **2020**, *66*, 7088–7100. [CrossRef]

26. Hammer, D.; Romashchenko, A.; Shen, A.; Vereshchagin, N. Inequalities for Shannon entropies and Kolmogorov complexities. In Proceedings of the Computational Complexity, Twelfth Annual IEEE Conference, Ulm, Germany, 24–27 June 1997; Volume 60, pp. 442–464.
27. Csirmaz, L. The Size of a Share Must Be Large. *J. Cryptol.* **1997**, *10*, 223–231. [CrossRef]
28. Zhang, Z.; Yeung, R. On characterization of entropy function via information inequalities. *IEEE Trans. Inf. Theory* **1998**, *44*, 1440–1452. [CrossRef]
29. Zhang, Z.; Yeung, R. A non-Shannon-type conditional inequality of information quantities. *IEEE Trans. Inf. Theory* **1997**, *43*, 1982–1986. [CrossRef]
30. Matús, F. Infinitely Many Information Inequalities. In Proceedings of the 2007 IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007; pp. 41–44.
31. Yeung, R.W. A framework for linear information inequalities. *IEEE Trans. Inf. Theory* **1997**, *43*, 1924–1934. [CrossRef]
32. Dougherty, R.; Freiling, C.; Zeger, K. Linear rank inequalities on five or more variables. *arXiv* **2010**, arXiv:cs.IT/0910.0284.
33. Strang, G. *Linear Algebra and Its Applications*; Thomson, Brooks/Cole: Belmont, CA, USA, 2006.
34. Ingleton, A.W. Representation of matroids. *Comb. Math. Appl.* **1971**, *23*, 149–167.
35. Dougherty, R. Computations of linear rank inequalities on six variables. In Proceedings of the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014; pp. 2819–2823. [CrossRef]
36. Dougherty, R.; Freiling, C.; Zeger, K. Linrank. Available online: http://code.ucsd.edu/zeger/linrank/ (accessed on 1 January 2023).
37. Padró, C. Lecture Notes in Secret Sharing. *IACR Cryptol. EPrint Arch.* **2012**, *2012*, 674.
38. Benson, H.P. An Outer Approximation Algorithm for Generating All Efficient Extreme Points in the Outcome Set of a Multiple Objective Linear Programming Problem. *J. Glob. Optim.* **1998**, *13*, 1–24. [CrossRef]
39. Dijk, M.V. A Linear Construction of Secret Sharing Schemes. *Des. Codes Cryptogr.* **1997**, *12*, 161–201. [CrossRef]
40. Fukuda, K. Polyhedral Computation. 2020. Available online: https://www.research-collection.ethz.ch/handle/20.500.11850/426218 (accessed on 1 January 2023).
41. Lassez, C.; Lassez, J. Quantifier elimination for conjunctions of linear constraints via a convex hull algorithm. *Symb. Numer. Comput. Artif. Intell.* **1992**, 103–122.
42. Xu, W.; Wang, J.; Sun, J. A projection method for derivation of non-Shannon-type information inequalities. In Proceedings of the 2008 IEEE International Symposium on Information Theory, Toronto, ON, Canada, 6–11 July 2008; pp. 2116–2120. [CrossRef]
43. Apte, J.; Walsh, J.M. Explicit Polyhedral Bounds on Network Coding Rate Regions via Entropy Function Region: Algorithms, Symmetry, and Computation. *arXiv* **2016**, arXiv:1607.06833.
44. Csirmaz, L. Using multiobjective optimization to map the entropy region. *Comput. Optim. Appl.* **2016**, *63*, 45–67. [CrossRef]
45. Mcmtroffaes. Pycddlib. Available online: https://pypi.org/project/pycddlib/2.1.6.html (accessed on 1 January 2023).
46. Löhne, A.; Weißing, B. Equivalence between polyhedral projection, multiple objective linear programming and vector linear programming. *Math. Methods Oper. Res.* **2016**, *84*, 411–426. [CrossRef]
47. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Beaverton, OR, USA, 2022.
48. Apte, J.; Walsh, J.M. Constrained Linear Representability of Polymatroids and Algorithms for Computing Achievability Proofs in Network Coding. *arXiv* **2016**, arXiv:1605.04598.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.