Computer Networks 99 (2016) 82–98

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

On the optimal design of secure network coding against wiretapping attack☆

Xiangmao Chang[a,f], Jin Wang[b,f], Jianping Wang[c], Kejie Lu[d,e,*], Yi Zhuang[a]

[a] School of Computing Science and technology, Nanjing University of Aeronautics and Astronautics, No.29 Jiangjun Road, Nanjing, 211106, China
[b] Department of Computer Science and technology, Soochow University, No.1 Shizi Road, Suzhou, 215006, China
[c] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
[d] College of Computer Science and Technology, Shanghai University of Electronic Power, No.2588 Changyang Road, Yangpu District, Shanghai, 200090, China
[e] Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico 00681, USA
[f] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, 210023, China

ARTICLE INFO

Article history:
Received 24 April 2015
Revised 9 November 2015
Accepted 29 December 2015
Available online 29 January 2016

Keywords:
Secure network coding
Wiretapping attack
Secure transmission rate
Network topology

ABSTRACT

In this paper, we study the optimal design of *weakly secure linear network coding (WSLNC)* against *wiretapping attack*. Specifically, given a set of wiretapped links, we investigate how to maximize the *weakly secure* transmission rate of multiple unicast streams between a pair of source and destination nodes, and how to minimize the size of the required finite field, over which the WSLNC can be implemented. In our study, we apply a novel approach that integrates the WSLNC design and the transmission topology construction. We first provide theoretical analysis and prove that the problem of finding the optimal transmission topology is NP-hard. We then develop efficient algorithms to find optimal and sub-optimal topologies in different scenarios. With the transmission topology, we design WSLNC schemes and theoretically analyze the relationships between the transmission topology and two important system factors: (1) the size of the finite field, and (2) the probability that a random linear network coding is weakly secure. Based on the relationships, we further improve our algorithms to address the two system factors, while keeping the same maximal *STR*. Extensive simulation results show that the proposed heuristic algorithms can achieve good performance in various scenarios.

1. Introduction

Over the past decade, *network coding* (NC) has attracted significant attention in communications networks [1,2], thanks to its potential of expanding the throughput of a communication network. For instance, Li et al. demonstrated that the *maximum flow* (max-flow) from a source to multiple destinations can be achieved by linear NC with a certain finite field [3]. Because of its simplicity, linear NC has been widely used in practice, and will be applied and investigated in this paper.

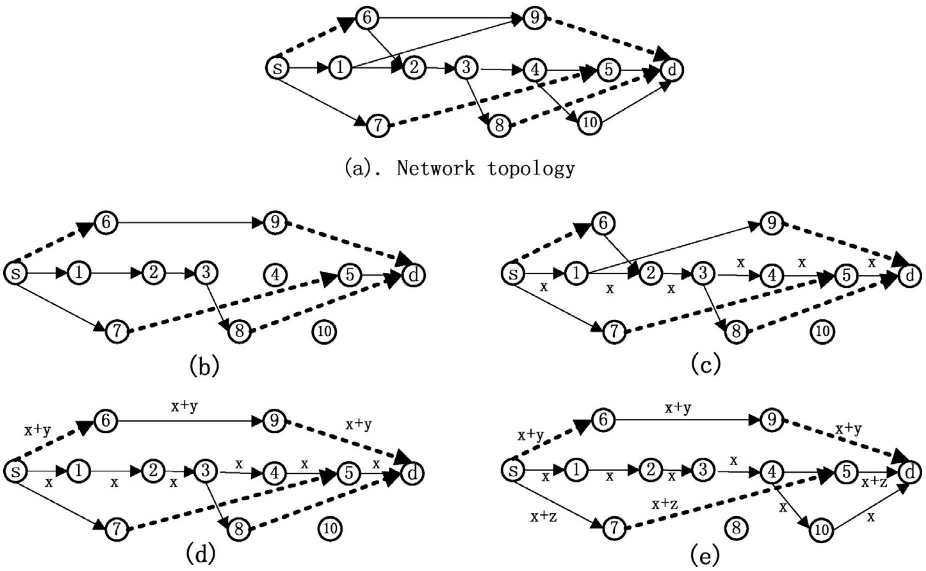
In addition to increasing the transmission data rate, NC can also secure data transmission against different attacks. Based on the attack models, most existing works on secure NC design can be classified into two groups: (1) *active attack* and (2) *passive attack*.

☆ Part of this work has been published in IEEE ICDCS 2010.

* Corresponding author at: Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico 00681, USA.

E-mail addresses: xiangmaoch@nuaa.edu.cn (X. Chang), wjin1985@suda.edu.cn (J. Wang), jianwang@cityu.edu.hk (J. Wang), kejie.lu@upr.edu (K. Lu), zhuangyi@nuaa.edu.cn (Y. Zhuang).

http://dx.doi.org/10.1016/j.comnet.2015.12.012
1389-1286/© 2015 Elsevier B.V. All rights reserved.



(a). Network topology

(b)

(c)

(d)

(e)

Fig. 1. The impact of transmission topology on the *STR*.

Active attacks include entropy attack and Byzantine modification attack, in which an attacker can alter the messages transmitted in the network, and consequently the receiver cannot recover the original data. Roughly speaking, to deal with active attacks, secure NC design requires a data verification scheme to detect and to filter out modified messages, in order to provide *integrity* of data transmission [4–8].

In *passive attacks*, such as wiretapping attack, an attacker may compromise the *confidentiality* of data transmission by wiretapping one or more edges. To against wiretapping attack, the conventional approaches are to encrypt all messages, by using end-to-end encryption or hop-by-hop encryption, which is time-consuming, calculation complex, and requires key distribution infrastructure. To defend against wiretapping attacks without using encryption, Cai and Yeung first demonstrated that NC can securely deliver messages with very powerful performance [9]. Their model includes secret sharing as a special case. In [10], Feldman et al. showed that, if a small amount of capacity is given up, a random code can achieve security by using a much smaller finite field than that in [9]. In [11], Bhattad and Narayanan generalized the model in [9] and defined a *weakly secure* model, which can accommodate the security requirements of a lot of practical applications. In this paper, we will investigate weakly secure NC design against wiretapping attack.

Generally, the implementation of secure NC involves two correlated steps: (1) to select a transmission topology that consists of a subset of nodes and edges from the network, and (2) to design the secure NC scheme based on the selected transmission topology. However, most of the previous work on secure NC against wiretapping attacks has focused on (2), i.e., to design the secure NC scheme based on a given routing topology, or to compute a transformation matrix based on a given NC scheme [9,12,13]. In this paper, we will apply a novel approach to integrate the two steps. Our approach is motivated by the observation that the transmission topology can significantly affect the transmission rate under the weakly secure requirement, a criterion referred to as the *secure transmission rate (STR)*. Our objective is to maximize the *STR*.

In Fig. 1, we illustrate the impact of transmission topology on *STR*. Fig. 1(a) shows a directed acyclic graph where each edge has a unit capacity. Suppose that a wiretapper has compromised a set of edges: {$s \rightarrow 6$, $7 \rightarrow 5$, $8 \rightarrow d$, $9 \rightarrow d$}. Given this information, the source node s wants to transmit messages to the destination node d without leaking meaningful information to the wiretapper. There are four possible topologies, as shown in Fig. 1(b–e), each of which has three disjoint paths between s and d, meaning that the transmission rate from s to d is 3 without considering the weakly secure requirement. However, these transmission topologies can lead to different secure transmission rates, as explained below.

- In Fig. 1(b), all three paths have an edge being wiretapped, which means that the wiretapper can receive and decode the same message as the destination node d does, no matter how NC is designed. In this case, the achievable *STR* can be 0.
- In Fig. 1(c), there exists a path (with all edges marked as "x" and x denotes a message transmitted on the path.) where no edge on that path is wiretapped. Therefore, the message x can be transmitted to the destination d without being wiretapped and the achievable *STR* can be 1.
- In Fig. 1(d), there exists a path (with all edges marked as "x") where no edge on that path is wiretapped. Meanwhile, there is another path (with all edges marked as "$x+y$") where some edges are wiretapped. However, if the second path is used to transmit the coded message $x+y$, then the wiretapper cannot decode the messages x and y because he or she does not

have the message x. Therefore, the achievable *STR* can be 2.

- In Fig. 1(e), there exists a path (with all edges marked as "x") where no edge on that path is wiretapped. Meanwhile, the other two paths, though both of them have some edges wiretapped, can transmit the coded messages $x + y$ and $x + z$, respectively. Again, the wiretapper cannot decode any message because he or she does not have the message x. Therefore, the achievable *STR* can be 3.

The above example clearly demonstrates that the transmission topology can significantly affect the *STR*. Therefore, it is important to consider the transmission topology construction and weakly secure NC design together. However, to the best of our knowledge, such an important issue has been largely overlooked in most of the existing work, except our study in [14–16]. Nevertheless, the attack model we study in this work is different to the ones in [14–16], where all intermediate nodes are trying to acquire data information passing through them but they do not cooperate. In other words, the attack models in [14–16] can be viewed as an insider attack, while in this paper, we address the case of outsider attack. Next, we summarize the novelties and major contributions.

- Theoretical analysis: We theoretically analyze the sufficient and necessary condition that there exists a secure LNC can be designed based on a transmission topology to achieve given *STR*. Based on the sufficient and necessary condition, we prove the maximum *STR* for a given source and destination pair. We also theoretically analyze the computational complexity of the problem.
- Transmission topology algorithm design: We design algorithms to find the optimized transmission topology. Since the problem is NP-hard, when the capacity from the source to destination is small, we design an optimal algorithm to find the optimal transmission topology. For the general case, we design efficient approximate algorithms to find sub-optimal transmission topology that can lead to high *STR*.
- Secure NC design: With the optimized transmission topology, we design secure NC on it. Furthermore, We study the relationship between the transmission topology and two important system factors: (1) the size of the finite field, and (2) the probability that a random NC is weakly secure. At last, we further improve the proposed algorithms to benefit the two system factors mentioned above while does not decrease the archived *STR*.
- Extensive simulations: We conduct extensive simulations under two classical network models to verify the performance of our heuristic algorithms.

The rest of the paper is organized as follows. We first introduce the system models in Section 2. Next, in Section 3, we define the problem to be investigated and analyze its properties, including (1) the conditions that a transmission topology can enable NC design that satisfies the weakly secure requirement, (2) the maximal *STR*, and (3) the complexity of the problem. In Section 4, we design algorithms that can efficiently find optimal or near-optimal transmission topologies in different scenarios. In Section 5, we design secure NC schemes based on the optimized transmission topology, and we further study the relationship between the transmission topology and two important system factors: (1) the size of the finite field, and (2) the probability that a random NC is weakly secure. To evaluate the heuristic algorithms, we conduct extensive simulation experiments and discuss the results in Section 6. Finally, we discuss related work in Section 7, and conclude the paper in Section 8.

Table 1
Notations.

Notations	Meaning
G, V, E	Network topology, node set, edge set
$A, \bar{A}$	Polluted edge set and clean edge set
s, d	Source node and destination node
r, L	Number of data packet streams and coding interval
X	rL original messages that s wants to send to d
$\mathbb{F}_q^{rL}$	rL-dimensional space on $\mathbb{F}_q$
ζ, Γ	Global encoding vector and coding matrix
$\Gamma_{A'}$	A matrix whose rows are all wiretapped coding vectors
Y	Wiretapped information
V_i	A matrix defined in formula (1)
ϑ	The number of wiretapped independent encoding vectors
G_k^m	A subgraph of G, composed by the maximal disjoint paths with k clean paths
c_k^m	The maximal number of disjoint paths in G_k
λ	The maximal number of secure paths
c_{max}	The maximal *STR*
c	The capacity between s and d

2. The system models

In this section, we present the basics for the weakly secure NC design problem, including the network model, the attack model, the NC scheme, the transmission model, and the weakly secure requirement. To facilitate the discussions, we list important notations in Table 1.

2.1. The network and attack models

In this paper, we consider a communications network that is represented by a directed acyclic graph $G = \langle V, E \rangle$, where V and E are the node set and the edge set, respectively. We assume that each edge is noiseless and has a unit capacity. Note that we can always split an edge to several edges with unit capacity if that edge has multiple units of capacity. We further assume that the traffic in the network consists of r unicast data streams that share the same source and destination nodes, denoted by s and d, respectively.

For the attack model, we let $A \subset E$ be the set of edges that are wiretapped. We refer to edges in A as *polluted edges* while edges in $\bar{A} = E - A$ as *clean edges*. A single path is a *clean path* if and only if all the edges on the path are *clean edges*; otherwise, it is a *polluted path*. Two paths are edge (node) disjoint if they do not share any common edges (nodes). In the rest of the paper, we use disjoint paths to refer to edge disjoint paths when there is no ambiguity.

2.2. The NC scheme

We suppose that the encoding (decoding) operations are only done at source (destination) node while let all intermediate nodes simply store and forward messages. This assumption is reasonable because coding operations in intermediate nodes require extra computation capability which may be impossible in many practical scenarios. Moreover, the coding operation at intermediate nodes will lead to computational overhead and transmission delay. For simplify the clarification, we assume that the system is *time division multiplexing* (TDM) based, in which each message occupies one time slot and the message arrival rate of each stream is one.

The following definition is useful throughout the rest of the paper:

Defnition 1. A coding interval L means L time slots during which the source buffer messages from all data streams for encoding.

In general, L can be a constant in an NC scheme. However, it is an important design variable in our scheme, which will be further investigated in Section 3. At the end of each coding interval L, rL buffered messages are encoded at node s, then the encoded messages are forwarded to node d. In the following, we will focus on the transmission in a coding interval L.

We let vector $X(t) = [x_1(t), x_2(t), \dots, x_r(t)]^\top$ be the messages received in the tth slot in a coding interval, where $[x_1(t), x_2(t), \dots, x_r(t)]^\top$ means the transpose of $[x_1(t), x_2(t), \dots, x_r(t)]$ and $x_i(t) \in \mathbb{F}_q$ is the tth message in the ith stream. We let $X = [X(1)^\top, X(2)^\top, \dots, X(L)^\top]^\top$ be the rL original messages that node s wants to send to d. To encode the messages, we will apply linear NC on source node s and destination node d.

For each edge in the transmission topology, the messages transmitted in a time slot can be written as $\zeta \cdot X$, where vector $\zeta \in \mathbb{F}_q^{rL}$ is the *global encoding vector* (GEV) and $\mathbb{F}_q^{rL}$ is a rL-dimensional space on $\mathbb{F}_q$. After each coding interval, the destination node can obtain a set of GEVs, which can be used to construct a matrix. If this matrix is full rank, the destination can decode and obtain the original messages.

For a given transmission topology $G' = \langle V', E' \rangle$, $G' \subset G$ including s and d, let $A' = E' \cap A$ be the wiretapped set of edges in G'. For each edge in A', a wiretapper can acquire both the message and the corresponding GEVs in a given time slot. Therefore, we can use $\Gamma_{A'} \cdot X$ to denote the wiretapped messages, where $\Gamma_{A'}$ represents a matrix whose row vectors are all GEVs of the messages acquired by the wiretapper.

Since the coding operations are only done at the source and destination, and the intermediate nodes only play a relay and forward role, the coded messages must be transmitted in the network through different disjoint paths. Specifically, the messages transmitted on the edges of the same path have the same GEVs. A *coding matrix* can be constructed by all the GEVs of the coded messages as its rows. The NC scheme can be determined uniquely by the coding matrix applied on the source.

Let Γ be a coding matrix with dimension $rL \times rL$ in finite field $\mathbb{F}_q$. In each coding interval, node s sends coded messages $\Gamma \cdot X$ to d. In this way, each coded message sent from s can be written as $\zeta \cdot X$ where ζ is a row vector of Γ. If Γ is a full rank matrix, d can recover X by simply multiplying Γ^{-1} to $\Gamma \cdot X$. Note that, if rL original messages are coded at s, then rL coded messages will be sent to d from s.

2.3. The weakly secure requirement

We define the weakly secure requirement following [11].

Defnition 2. Within a coding interval L, a transmission is weakly security, if

$$I\left(\{x_i(t)\}_{t=1}^L, Y\right) = 0, \forall i$$

where Y is the information that the wiretapper can access within coding internal L, $I(\{x_i(t)\}_{t=1}^L, Y)$ is the mutual information between the ith data stream sent from the source within coding interval L and Y.

Since all messages transmitted in the network are linear combination of messages in X, $I(\{x_i(t)\}_{t=1}^L, Y) = 0$ means that the wiretapper cannot obtain any messages which are linear combination of messages from the same data stream of coding interval L. For example, if the wiretapper accesses an edge containing $x_1(1) \oplus x_2(1)$, where $x_1(1)$ and $x_2(1)$ come from different streams, the wiretapper cannot obtain any meaningful information which implies weakly security. The average number of messages that can be transmitted from s to d under the requirement of weakly secure in unit time slot is referred to as *STR*.

3. Problem definition and analysis

In this section, we first define the weakly secure NC design problem, in which we aim at integrating the weakly secure NC design with the transmission topology construction. We then investigate how a transmission topology can lead to weakly secure NC that satisfies the weakly secure requirement, and the value of the maximal *STR*. Finally, we prove that the problem is NP-hard.

3.1. Problem definition

We now define the *maximizing STR with NC* (*MSTR-NC*) problem as follows.

Defnition 3. Given a network G, the set of wiretapped edges A, and r streams from s to d, The MSTR-NC problem is to find the optimal transmission topology G', on which a weakly secure NC scheme can be designed to achieve the maximal *STR*.

3.2. The impact of the weakly secure requirement

To solve the MSTR-NC problem, we first investigate the impact of the weakly secure requirement on the transmission topology.

Lemma 1. *Given a transmission topology G', r different data streams from s to d, coding interval L and the coding matrix Γ, if the transmission is weakly secure, then the maximal number of independent GEVs obtained by the wiretapper cannot be more than $(r-1)L$.*

Proof. Let α_j be a row vector with length rL where the jth element of the vector is 1 and all other elements are 0. We define matrix V_i as

$$V_i = \begin{bmatrix} \alpha_{(i-1)L+1} \\ \alpha_{(i-1)L+2} \\ \vdots \\ \alpha_{iL} \end{bmatrix}, 1 \le i \le r. \tag{1}$$

Then $V_i \cdot X$ represents all the messages in the ith data stream.

Suppose a wiretapper obtains ϑ independent GEVs. Let ζ_j, $1 \le j \le \vartheta$, be the jth element of them. All ζ_j, $1 \le j \le \vartheta$, form a matrix, $\Gamma_{A'}$, with dimension $\vartheta \times rL$. Then the transmission is weakly secure if and only if by taking any linear combination of $\Gamma_{A'} \cdot X$, the wiretapper cannot recover any message which is a linear combination of $V_i \cdot X$ for all $1 \le i \le r$, which implies: for all nonzero vector η_1, η_2

$$\eta_1 \cdot \Gamma_{A'} \ne \eta_2 \cdot V_i, 1 \le i \le r \tag{2}$$

i.e., all row vectors in $\Gamma_{A'}$ and V_i are linearly independent for $1 \le i \le r$. Since there are totally $\vartheta + L$ row vectors in $\Gamma_{A'}$ and V_i, the length of each row vector is rL, we have $\vartheta + L \le rL$, i.e., $\vartheta \le (r-1)L$. □

Lemma 2. *Given a transmission topology G', r different data streams from s to d, coding interval L, if the maximal number of independent GEVs obtained by the wiretapper is no more than $(r-1)L$, then there exists a coding matrix Γ over $\mathbb{F}_q$ ($q > r$) such that the transmission is weakly secure.*

Proof. Let $rank(\Gamma_{A'}) = \vartheta$, i.e. the wiretapper obtains ϑ independent GEVs, then $\vartheta \le (r-1)L$. In our transmission model, ϑ independent GEVs must be ϑ row vectors in Γ. Without loss of generality, we set the ϑ row vectors to be the first ϑ row vectors in Γ, let these ϑ row vectors form matrix A. Therefore, we only need to show that there exists Γ such that (2) hold.

For $1 \le m \le \vartheta$, it is sufficient to show if vectors in

$$\{\zeta_1, \zeta_2, \ldots, \zeta_{m-1}\} \cup V_i \tag{3}$$

are linearly independent for all $1 \le i \le r$, then it is possible to choose new vector ζ_m such that vectors in

$$\{\zeta_1, \zeta_2, \ldots, \zeta_{m-1}, \zeta_m\} \cup V_i$$

are linearly independent for all $1 \le i \le r$. Specifically, ζ_m is chosen such that it is linearly independent of all vectors in (3) for all $1 \le i \le r$, i.e., we require that

$$\zeta_m \in \mathbb{F}_q^{rL} \setminus \bigcup_{i=1}^{r} \langle \zeta_1, \zeta_2, \ldots, \zeta_{m-1}, V_i \rangle \tag{4}$$

where $\langle \cdot \rangle$ denotes the linear span of a set of vectors. ζ_m exists in $\mathbb{F}_q$ if the set of (4) is nonempty. Since the vectors in (3) are linearly independent, we have

$$\begin{aligned} |\bigcup_{i=1}^{r} \langle \zeta_1, \zeta_2, \ldots, \zeta_{m-1}, V_i \rangle| &\le \sum_{i=1}^{r} |\langle \zeta_1, \zeta_2, \ldots, \zeta_{m-1}, V_i \rangle| \\ &= \sum_{i=1}^{r} q^{L+m-1} = rq^{L+m-1} \end{aligned}$$

Therefore,

$$\begin{aligned} &|\mathbb{F}_q^{rL} \setminus \bigcup_{i=1}^{r} \langle \zeta_1, \zeta_2, \ldots, \zeta_{m-1}, V_i \rangle| \\ &\ge q^{rL} - rq^{L+m-1} = q^{L+m-1}(q^{rL-L-m+1} - r) \end{aligned} \tag{5}$$

Since $m \le \vartheta \le (r-1)L$, when $q > r$, the above inequality is large than zero, i.e., such ϑ vectors exist in $\mathbb{F}_q^{rL}$ which are linearly independent with V_i for all $1 \le i \le r$. After finding the first ϑ row vectors according to (4), we can extend $\{\zeta_1, \zeta_2, \ldots, \zeta_\vartheta\}$ to a basis of $\mathbb{F}_q^{rL}$ which forms a coding matrix Γ. With transmission topology G' and coding matrix Γ, the vectors obtained by wiretapper are the first ϑ row vectors in Γ. Therefore, the transmission is weakly secure. □

Theorem 3. *Given a transmission topology G', r different data streams from s to d, and coding interval L, there exists a coding matrix Γ that makes the transmission weakly secure, if and only if the maximal number of independent GEVs obtained by the wiretapper is no more than $(r-1)L$.*

Proof. According to Lemmas 1 and 2, this theorem holds obviously. □

Theorem 4. *Given a transmission topology G', r different data streams from s to d, and coding interval L, there exists a coding matrix Γ that makes the transmission weakly secure, if and only if the number of different messages transmitted on the clean paths is at least L.*

Proof. In our transmission model, the messages transmitted on the same path have the same GEVs. Since the corresponding GEVs of the messages are the row vectors in Γ and Γ is full rank, the number of independent GEVs obtained by the wiretapper equals to the number of different messages obtained by the wiretapper.

If there exists the coding matrix Γ which makes the transmission weakly secure, according to Theorem 3, the maximal number of independent GEVs obtained by the wiretapper cannot be more than $(r-1)L$. Therefore, the number of different messages obtained by the wiretapper is no more than $(r-1)L$. Thus, the number of different messages transmitted on the clean paths is at least L.

If the number of different messages transmitted on the clean paths is at least L, there exists the coding matrix Γ' such that the maximal number of independent GEVs obtained by the wiretapper cannot be more than $(r-1)L$. According to Theorem 3, there exists the coding matrix Γ which makes the transmission weakly secure. □

3.3. The maximal STR

We now analyze the maximal *STR*. Note that, in our system model, the transmission topology is actually a set of

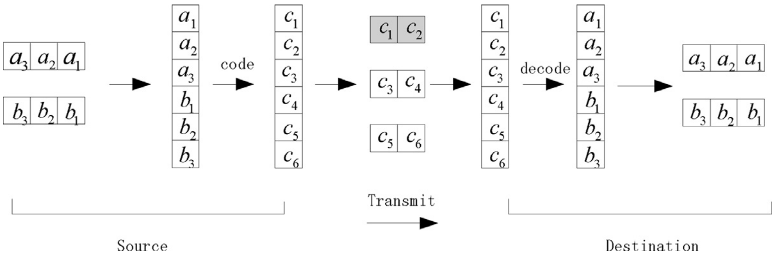

a_3 a_2 a_1
b_3 b_2 b_1
a_1 a_2 a_3 b_1 b_2 b_3
code
c_1 c_2 c_3 c_4 c_5 c_6
c_1 c_2
c_3 c_4
c_5 c_6
c_1 c_2 c_3 c_4 c_5 c_6
decode
a_1 a_2 a_3 b_1 b_2 b_3
a_3 a_2 a_1
b_3 b_2 b_1
Transmit
Source
Destination

Fig. 2. An example of the transmission scheme.

disjoint paths which share the same source node and the same destination node. The set of disjoint paths consists of some clean paths and some polluted paths. Let G_k be an transmission topology which consists of k disjoint clean paths and c_k total disjoint paths. we have the following lemma.

Lemma 5. *Given r different data streams from s to d, the maximal STR from s to d in G_k is $c' = min\{rk, c_k\}$, which can be achieved if and only if $rL \equiv 0\ (mod\ c')$.*

Proof. Let the number of transmission time slots of transmitting rL messages from s to d in G_k be T, where T is an integer. Since the capacity between s and d is c_k and rL messages can be transmitted within T time slots, we have:

$$T \geq \left\lceil \frac{rL}{c_k} \right\rceil \tag{6}$$

According to Theorem 4, there exists the coding matrix Γ which can transmit rL messages from s to d securely if and only if the number of different messages transmitted on the clean paths is no less than L, i.e., $kt \geq L$. Since T is an integer, it is equivalent to:

$$T \geq \left\lceil \frac{L}{k} \right\rceil \tag{7}$$

According to Eqs. (6) and (7), $T \geq max\{\lceil \frac{L}{k} \rceil, \lceil \frac{rL}{c_k} \rceil\}$. Since the *STR* is $\frac{rL}{T}$, to maximize the *STR*, the number of transmission time slots must be:

$$T = max\left\{\left\lceil \frac{L}{k} \right\rceil, \left\lceil \frac{rL}{c_k} \right\rceil\right\} \tag{8}$$

According to Eq. (8), we have the following results.

(1) If $c_k > rk$, then $\lceil \frac{rL}{c_k} \rceil < \lceil \frac{rL}{rk} \rceil = \lceil \frac{L}{k} \rceil$, $T = \lceil \frac{L}{k} \rceil$. The *STR* is $c_1 = \frac{rL}{T} = \frac{rL}{\lceil \frac{L}{k} \rceil}$, $c_1 = rk$ which is maximal if and only if $L \equiv 0 (mod\ k)$.

(2) If $c_k \leq rk$, then $\lceil \frac{rL}{c_k} \rceil \geq \lceil \frac{rL}{rk} \rceil = \lceil \frac{L}{k} \rceil$, $T = \lceil \frac{rL}{c_k} \rceil$. The *STR* is $c_2 = \frac{rL}{T} = \frac{rL}{\lceil \frac{rL}{c_k} \rceil}$, $c_2 = c_k$ which is maximal if and only if $rL \equiv 0\ (mod\ c_k)$.

In summary, the maximal *STR* is $c' = min\{rk, c_k\}$. It can be achieved if and only if $rL \equiv 0\ (mod\ c')$. □

From Lemma 5, a transmission topology G_k from G, such that $c' = min\{rk, c_k\}$ is maximized among all unicast topologies in G, is a transmission topology which can lead to maximal *STR*. Let the maximal number of disjoint clean paths between s and d of G be λ (We can get λ by removing all the polluted edges and calculating the maximal number of disjoint paths between s and d in the remainder graph). Let G_k^m $(1 \leq k \leq \lambda)$ be a subgraph of G which is composed by the maximal number of disjoint paths with k disjoint clean paths. Let the capacity of G_k^m be c_k^m.

Theorem 6. *Given a network topology G and r different data streams from s to d, the maximal STR from s to d is $c_{max} = \max_{1 \leq k \leq \lambda} \min\{kr, c_k^m\}$, which can be achieved if and only if $rL \equiv 0\ (mod\ c_{max})$.*

Proof. For each k, $1 \leq k \leq \lambda$, the maximal *STR* in G_k^m is $min\{kr, c_k^m\}$ if and only if $rL \equiv 0(mod\, c_k^m)$ according to Lemma 5. Therefore, the maximal *STR* from s to d in G is $c_{max} = \max_{1 \leq k \leq \lambda} \min\{kr, c_k^m\}$, if and only if $rL \equiv 0\ (mod\ c_{max})$. □

Upon obtaining the transmission topology with c_k^m disjoint paths such that $\min\{kr, c_k^m\} = c_{max}$, we can determine the coding interval L by the function $rL \equiv 0(mod\ c_{max})$ to achieve the maximal *STR* c_{max}. Though L may have multiple values, considering that the coding matrix Γ is $rL \times rL$ dimension, small Γ means short decoding time interval and low encoding/decoding computational complexity. Therefore, we prefer to have the smallest L. The smallest value of L satisfying $rL \equiv 0(mod\ c_{max})$ can be calculated by $\frac{c_{max}}{(r, c_{max})}$, where (r, c_{max}) means the greatest common divisor of r and c_{max}.

The example in Fig. 2 demonstrates the basic idea of our transmission scheme. The source wants to transmit 2 independent data streams to the destination. Suppose that the obtained transmission topology consists of 3 disjoint paths with 2 clean paths. Thus, the source chooses $L = 3$ and the maximal transmission rate is 3. The source first puts the 6 messages into 1 vector, then encodes them with the proposed coding scheme, and finally transmits them on the transmission topology. The shaded messages can be obtained by the wiretapper. When the destination receives the messages, it first decodes them, then puts them into the buffers of the original 2 data streams.

3.4. The computational complexity of MSTR-NC

In this section, we study the computational complexity of the *MSTR-NC* problem. We will reduce the well-known *SAT* problem [17] to the special case of the *MSTR-NC* problem.

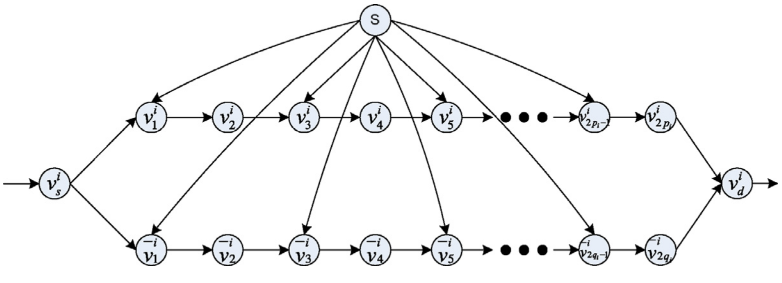


Fig. 3. Acyclic directed lobe for x_i.

According to Theorem 6, a special case of the maximal *STR* is c_k^m, $1 \leq k \leq \lambda$. In this case, the optimal transmission topology required in *MSTR-NC* problem is G_k^m. Thus, a special case of the *MSTR-NC* problem is to find G_k^m. The decision version of finding G_k^m equals to find $c'_k(1 \leq c'_k)$ disjoint paths with k disjoint clean paths from s to d when c'_k is given.

The *SAT* is defined as ([17]):

INSTANCE: Set X of variables, collection D of clauses over X

QUESTION: Is there a true assignment τ for D?

The *SAT* is a well-known NP-hard problem.

For a given instance of the *SAT*, a collection $D = \{D_1, D_2, \ldots, D_{c'_1-1}\}$ of clauses is defined on a finite set $X = \{x_1, x_2, \ldots, x_t\}$. For each variable x_i, a lobe is constructed as shown in Fig. 3, in which p_i is the number of occurrences of x_i in the clauses and q_i is the number of occurrences of $\bar{x}_i$. Then we connect the lobes one by one in series as follows: there is a directed edge from node v_d^i to v_s^{i+1} $(1 \leq i \leq t-1)$, s is connected to v_s^1 and all the nodes v_j^i and $\bar{v}_j^i$ where j is odd. Node v_d^t is connected to d.

In addition, there are nodes $D_1, \ldots, D_{c'_1-1}$ and a directed edge from each of them to d. For the jth occurrence of $x_i(\bar{x}_i)$, there is a directed edge from $v_{2j}^i(\bar{v}_{2j}^i)$ to the D_x in which it occurs.

Output edges of s except $s \rightarrow v_s^1$ and input edges of d except $v_d^t \rightarrow d$ are polluted edges, while other edges are clean edges. It is obvious that this transformation process takes polynomial time.

We next show that the constructed directed graph is acyclic. Assume that there exists a directed cycle in G, e.g., there exists a node v which has a directed path to itself. Note that s has no input edges, d has no output edges, and for each $D_1, \ldots, D_{c'_1-1}$, there is only one output edge to d. So we can only have $v \in \widetilde{V} = V - \{s, d, D_1, \ldots, D_{c'_1-1}\}$. However, from the construction of each lobe, we know that all edges connected to $\widetilde{V}$ are pointed at right as shown in Fig. 3. So there are no cycles in $G\langle\widetilde{V}, E\rangle$ either. Thus the constructed graph is acyclic.

Let the maximal number of disjoint clean paths in G be λ. Since there is only one clean edge in the input edges of d, $\lambda = 1$. Since $0 < k \leq \lambda = 1$, the decision version of finding G_k^m is equals to find $c'_1(1 \leq c'_1 \leq r)$ link-disjoint paths with one clean path from s to d in G.

Lemma 7. *The problem of finding c'_1 link-disjoint paths with one clean path from s to d in G is NP-complete.*

Proof. We first give two claims:

Claim 1: If there are c'_1 edge-disjoint paths with one clean path in the constructed graph G, then the SAT has a YES solution.

We note that the one clean path has to pass through all lobes. Then, for each variable x_i in the SAT, we set it “true” if and only if that clean path passes through the lower part of the ith lobe, otherwise we set it “false”. Since there are $c'_1 - 1$ polluted paths and $c'_1 - 1$ clauses, for each D_m $(1 \leq m \leq c'_1 - 1)$, there exists one polluted path passing through it. That polluted path must pass through a lobe, e.g., the ith lobe. If that polluted path passes through the upper part of the ith lobe which means $x_i \in D_m$, then the clean path must pass through the lower part of ith lobe. Therefore, x_i is set “true” and D_m is “true”. Otherwise, if that polluted path passes through the lower part of ith lobe which means $\bar{x}_i \in D_m$, then the clean path must pass through the upper part of the ith lobe. Therefore, x_i is set “false”, and D_m is also “true”. Thus D_m $(1 \leq m \leq c-1)$ is satisfied. Therefore, we have a YES solution to the SAT.

Claim 2: If the SAT has a YES solution, then there are c'_1 edge-disjoint paths with one clean path in the constructed graph G.

Given the YES solution to SAT, we can have one clean path where we let the path pass through the lower part of the ith lobe if and only if x_i is “true”, otherwise let it pass though the upper part. Since each clause D_m $(1 \leq m \leq c'_1 - 1)$ contains at least one literal x_i or $\bar{x}_i$ which is “true”, the polluted path passes through the upper part of ith lobe if x_i is “true” or lower part of ith lobe if $\bar{x}_i$ is “true”. These c'_1 paths are obviously edge-disjoint.

We can use the example Fig. 4 to demonstrate the proof. The example has a network G corresponding to expression $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. There are $4+1$ disjoint paths (bold) with one clean path (dashed).

The above two claims show that the problem of finding c'_1 edge-disjoint paths with one clean path is NP-complete. □

The above lemma shows the NP-completeness of a special case of the *MSTR-NC* problem. Therefore, we have the following theorem for the general problem.

Theorem 8. The *MSTR-NC* problem is NP-hard.

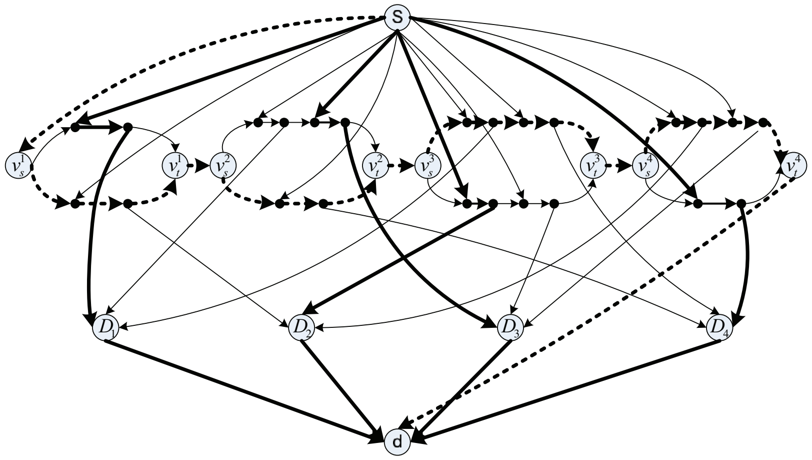


Fig. 4. Satisfiability reduction in a directed acyclic network.

4. The construction of transmission topology

If we can find each G_k^m, $1 \le k \le \lambda$, we can get each c_k^m. By comparing, we can find k', such that $\min\{k'r, c_{k'}^m\} = \max_{1\le k\le\lambda} \min\{kr, c_k^m\}$. From Theorem 6, if $k'r >= c_{k'}^m$, $G_{k'}^m$ is the optimal transmission topology, else, a sub-graph of $G_{k'}^m$ which composed by k' clean disjoint paths and $k'(r-1)$ other disjoint paths is the optimal transmission topology. Thus, the key of finding the optimal transmission topology is to find G_k^m for each k, $1 \le k \le \lambda$. In this section, we give efficient algorithms to find G_k^m.

4.1. An optimal algorithm based on shortest path

In this section, we develop an optimal algorithm to construct the transmission topology. This algorithm is feasible to optimally solve MSTR-NC problem. We name the algorithm as OTCSP, which stands for Optimal Topology Construction based on Shortest Path.

4.1.1. The OTCSP algorithm

Let the capacity between s and d be c. Then the *STR* between these two nodes can be x where $1 \le x \le c$. The OTCSP algorithm will find x disjoint paths with k clean paths if there exist such x paths. By using such an algorithm with all possible x where $1 \le x \le c$, we can find the maximal disjoint paths with k disjoint clean paths between s and d. Algorithm 1 shows the pseudo-code of OTCSP algorithm.

Algorithm 1 The OTCSP Algorithm.

```
Require: $G = \langle V, E\rangle$, k, polluted edge set A.
Ensure: Maximum disjoint paths with k clean paths.
Transform $G$ to $G'$;
for $x = c$ to 1 do
    Transform $G'$ to $\overline{G'_x}$;
    Find the shortest path $\overline{P}$ in $\overline{G'_x}$;
    if the cost of $\overline{P}$ equals to 0 then
        Transform $\overline{P}$ to $x$ node-disjoint paths with $k$ clean paths;
        Transform the $x$ node-disjoint paths with $k$ clean paths in $G'$ to $x$ edge-disjoint paths with $k$ clean paths in $G$;
        return $x$ edge-disjoint paths with $k$ clean paths in $G$;
    else
        x=x-1;
    end if
end for
return Fail;
```

In the OTCSP algorithm, we first transform the original topology G to G', such that the problem of finding edge-disjoint paths in G can be transformed to the problem of finding node-disjoint paths in G'. We then transform G' to $\overline{G'_x}$, such that the problem of finding x node-disjoint paths with k clean paths in G' can be transformed to the problem of finding the shortest path in $\overline{G'_x}$. Finally, we can use the existing algorithm of finding the shortest path to solve the problem. We now give methods to transform G and G' to G' and $\overline{G'_x}$ respectively.

(1) *transform G to G'*: We first add two nodes s' and d' and two clean edges $s' \to s$ and $d \to d'$ to G so that each edge e in G corresponds to a node e' in G'. An edge $e'_i \to e'_j$ is in G' if and only if e_i and e_j share one node in G. $e'_i \to e'_j$ in G' is clean if and only if both e_i and e_j in G are clean. The idea of this method comes from [18]. The difference is [18] doesn't consider security.

(2) *transform* G' *to* $\overline{G'_x}$: We first assign a cost to each edge of G' as follows: if e is clean, $c(e) = 0$, otherwise, $c(e) = 1$. We then relabel nodes with a number l from 1 to $|V|$ to ensure that each edge $u \to v$ in E satisfies $l(u) < l(v)$. For simplicity and without confusion, we still use s and d to represent the source node and the destination node in G'. Without loss of generality, we may assume that $s \to d \notin E$ (if $s \to d$ does exist, we can add a vertex u and replace $s \to d$ by $s \to u$ and $u \to d$). Then, we can transform G' to an acyclic directed graph $\overline{G'_x} = (\overline{V}, \overline{E})$ as follows:

- $\overline{V} = \{\langle v_1, \ldots, v_x\rangle | v_i \in V, i = 1, \ldots, x,$ and $v_i \neq v_j$ when $i \neq j$ unless $v_i = v_j = s$ or $v_i = v_j = d$ $\}$.
- $\overline{E} = \{\langle v_1, v_2, \ldots, v_x\rangle \to \langle u_1, v_2, \ldots, v_x\rangle | v_1 \to u_1 \in E$ and $l(v_1) \leq l(v_i), i = 1, \ldots, x\} \cup \ldots \cup \{\langle v_1, v_2, \ldots, v_x\rangle \to \langle v_1, v_2, \ldots, u_x\rangle | v_x \to u_x \in E$ and $l(v_x) \leq l(v_i), i = 1, \ldots, x-1\}$.

The cost of each edge in $\overline{G'_x}$ is assigned as follows:

- $c(\langle v_1, \ldots, v_i, \ldots, v_x\rangle \to \langle v_1, \ldots, u_i, \ldots, v_x\rangle) = c(v_i \to u_i), i = 1, \ldots, k$.
- $c(\langle v_1, \ldots, v_i, \ldots, v_x\rangle \to \langle v_1, \ldots, u_i, \ldots, v_x\rangle) = 0, i = k+1, \ldots, x$.

The idea of this method comes from [19]. In [19], an algorithm to find two node-disjoint paths from two sources to two destinations is introduced. We generalize this technique to find x node-disjoint paths from one source to one destination. Then, we assign a suitable cost to each edge to find x node-disjoint paths with k clean paths. An example that transform G' to $\overline{G'_2}$ is given in Fig. 5.

4.1.2. The correctness of the algorithm

Before giving the correctness of the algorithm, we give two lemmas as follows.

Lemma 9. *There are x edge-disjoint paths with k clean edge-disjoint paths in G if and only if there are x node-disjoint paths with k clean node-disjoint paths in G'.*

Proof. From [18], we know that any two edge-disjoint paths in G correspond to two node-disjoint paths in G' and vice versa. Thus we only need to show k edge-disjoint paths in G which corresponding to k clean node-disjoint paths in G' are clean and vice versa. From the construction of G', this can be easily verified. □

Lemma 10. *There exist x directed node-disjoint paths from s to d in G' if and only if there exists a directed path $\overline{P}$ from $\langle s, \ldots, s\rangle$ to $\langle d, \ldots, d\rangle$ in $\overline{G'_x}$.*

Proof. The "only if" direction: Let P_i, $i = 1, \ldots, x$, be x node-disjoint paths in G'. The proof is done by induction on $\sum_{i=1}^{x} L(P_i)$, where $L(P_i)$ represents the number of edges in P_i. By our assumption, we know $L(P_i) \geq 2$ for any $1 \leq i \leq c$. So $\sum_{i=1}^{x} L(P_i) \geq 2x$. If $\sum_{i=1}^{x} L(P_i) = 2x$, then there exist x nodes $u_1, \ldots u_x \in V \backslash \{s, d\}$, such that $P_i = (s, u_i, d)$, $i = 1, \ldots, x$ are x disjoint paths. Then $\overline{P} = \langle s, \ldots, s\rangle \to \langle u_1, \ldots, s\rangle \to \ldots \to \langle u_1, \ldots, u_x\rangle \to \langle d, \ldots, u_x\rangle \to \ldots \to \langle d, \ldots, d\rangle$ is the desired path in $\overline{G'_x}$.

Assume that $\sum_{i=1}^{x} L(P_i) > 2x$. Let $P_i = (s = v_i^1, v_i^2, \ldots, v_i^{l_i} = d)$, $i = 1, \ldots, x$. Then $\langle s, s, \ldots, s\rangle \to \langle v_1^2, s, \ldots, s\rangle \to \ldots \to \langle v_1^2, v_2^2, \ldots, v_k^2\rangle$ are the first k edges of $\overline{P}$. The rest edges of P are provided by the inductive hypothesis on paths $P_i' = (v_i^2, \ldots, v_i^{l_i} = d)$, $i = 1, \ldots, x$. This completes the proof of the "only if" direction.

The "if" direction: Let $\overline{P} = (\langle s, \ldots, s\rangle = \langle v_1^1, \ldots, v_x^1\rangle, \ldots, \langle v_1^N, \ldots, v_x^N\rangle = \langle d, \ldots, d\rangle)$, where N is the length of P. Then $P_i = (s = v_i^1, \ldots, v_i^N = d)$ for $i = 1, \ldots, x$ are x directed paths from s to d in G'. We have to prove that for any $1 \leq i \leq x$, $1 \leq j \leq x$ and $i \neq j$, P_i and P_j are node-disjoint. If not, there exists at least one common node except s and d in both P_i and P_j. Without loss of generality, let such two nodes in G' be $v_i^m = v_j^n$ where $m < n$. By the definition of node in $\overline{G'_x}$, the components of node of $\overline{G'_x}$ which are nodes in G' are different except s and d. Therefore, the common node of P_i and P_j in G' must correspond to different nodes in $\overline{G'_x}$. Since $v_i^n \neq v_j^n$, we have $v_i^n \neq v_i^m$. Since nodes $\langle v_1^m, \ldots, v_x^m\rangle$ and $\langle v_1^n, \ldots, v_x^n\rangle$ are in path $\overline{P}$, there exists $m' \in [m, n]$, such that $l(v_i^m) = l(v_i^{m'}) \leq l(v_i^n)$ and $l(v_i^{m'}) \leq l(v_k^{m'})$ for all $k \in [1, x]$. Thus $l(v_i^{m'}) \leq l(v_j^{m'})$. Since $v_i^{m'} \neq v_j^{m'}$, we have $v_j^n \neq v_j^{m'}$, and $l(v_j^{m'}) < l(v_j^n)$. So we have $l(v_i^m) < l(v_j^n)$, which is a contradiction of $v_i^m = v_j^n$. □

Theorem 11. *Let $\overline{P}$ be a minimum-cost path from $\langle s, \ldots, s\rangle$ to $\langle d, \ldots, d\rangle$ in $\overline{G'_x}$. If the cost of $\overline{P}$ is 0, then the x paths $P_1, \ldots, P_x$ corresponding to $\overline{P}$ form x node-disjoint paths from s to d in G' with at least k clean node-disjoint paths.*

Proof. From Lemma 10, the x paths $P_1, \ldots, P_x$ corresponding to $\overline{P}$ are x node-disjoint paths from s to d. According to the way that we assign the cost of edges in $\overline{G'_x}$, if the cost of $\overline{P}$ equals 0, then the cost of $P_i (1 \leq i \leq k)$ equals 0, Thus $P_1, \ldots, P_k$ are k clean paths. □

4.1.3. The complexity of OTCSP

The time complexity of transforming G to $G^{'}$ is $O(|E|)$ (Line 1 in Algorithm 1). Generating $\overline{G'_x}$ takes $O(|\overline{E}|)$ operations. We have $|\overline{E}| \leq P_{|V|-2}^{x}|E|$ since each edge $u \to v \in E$ yields at most $P_{|V|-2}^{x}$ edges in $\overline{G'_x}$. Therefore, generating $\overline{G'_x}$ can be done in $O(|V|^x|E|)$ operations (Line 3 in Algorithm 1). Finding the shortest path in $\overline{G'_x}$ can be done in $O((|V|^x|E|)^2)$ operations (Line 4 in Algorithm 1). Transforming $\overline{P}$ in $\overline{G'_x}$ to x edge-disjoint paths with k clean paths in G can be done in $O(|E|)$ operations (from Line 6 to 7 in Algorithm 1). Thus the complexity of OTCSP is $O(c(|V|^c|E|)^2)$.

4.2. Heuristic algorithms based on k-shortest paths

In the case that the size of the problem is small, we can solve the MSTR-NC problem optimally by using the proposed OTCSP algorithm within a short period of time. However, when the problem size is large, the computational complexity of OTCSP algorithm is considerable large because that the MSTR-NC problem has been proved as NP-hard problem in Section 3.4. Therefore, we develop efficient approximation algorithms in this section to efficiently solve the problem when the problem size grows large. To construct the topology in a general case, we develop a baseline algorithm and two efficient heuristic algorithms based on k-shortest paths to find G_k^m with higher *STR*.

4.2.1. Baseline algorithm

We implement a simple algorithm (Algorithm 2) called BMF (Based on Max-Flow algorithm) as the baseline for performance comparison. BMF algorithm is based on the max-flow algorithm. We first hide all polluted edges and run the max-flow algorithm, which can generate the maximal number of disjoint clean paths because the capacity of

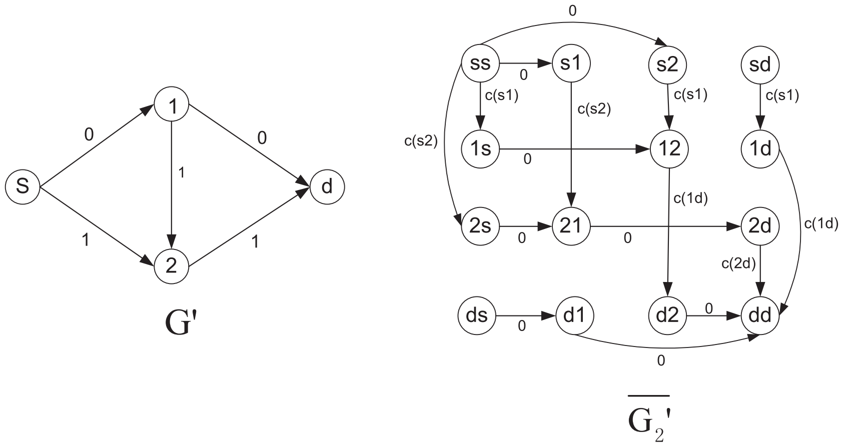


Fig. 5. Transformation from G' to $\overline{G_2'}$.

```
Algorithm 2 The BMF Algorithm.
Require: $G = \langle V, E\rangle$, k, polluted edge set A.
Ensure: Maximum disjoint paths with k clean paths.
Hide all edges in A;
Find the maximal disjoint paths by the max-flow algorithm, then randomly pick out $k$ paths, let them be $P_1, \ldots, P_k$.
Hide $P_1, \ldots, P_k$;
Restore all edges in A;
Find the maximal disjoint paths by the max-flow algorithm: $P_{k+1}, \ldots, P_{k+max}$;
return $P_1, \ldots, P_k, P_{k+1}, \ldots, P_{k+max}$;
```

each edge is 1. We pick out k of them randomly as k disjoint clean paths that we will use, then hide these k disjoint clean paths, restore all polluted edges and find the maximal number of disjoint paths in the remaining graph. These paths with k disjoint clean paths together form G_k^m.

4.2.2. Topology Construction Based On k-Shortest Paths (TCKSP)

From the discussions in Section 3, we can observe that if a clean path uses more edges in a minimum cut, then less number of remaining disjoint paths can be found when we ignore the clean path. Based on such an observation, we can design the following algorithm: Topology Construction based on k-Shortest Paths (TCKSP), as shown in Algorithm 3.

First, we hide all polluted edges, then find k disjoint shortest paths, denoted as $P_1, \ldots, P_k$. Obviously, all of these k disjoint paths are clean paths. Next, we hide all edges that belong to these k clean paths, and we restore all polluted edges. Since each edge have a unit capacity, we can run the max-flow algorithm to find the maximal disjoint paths in the remaining network, denoted as $P_{k+1}, \ldots, P_{k+max}$, where max is the maximal number of disjoint paths in the remaining network. Finally, we can form G_k^m by using $P_1, \ldots, P_k, P_{k+1}, \ldots, P_{k+max}$, which implies that the c_k^m is $k + max$.

```
Algorithm 3 The TCKSP Algorithm.
Require: $G = \langle V, E\rangle$, k, polluted edge set A.
Ensure: Maximum disjoint paths with k clean paths.
Hide all edges in A;
Find k shortest paths: $P_1, \ldots, P_k$;
Hide $P_1, \ldots, P_k$;
Restore all edges of A;
Find the maximal disjoint paths by the max-flow algorithm: $P_{k+1}, \ldots, P_{k+max}$;
return $P_1, \ldots, P_k, P_{k+1}, \ldots, P_{k+max}$;
```

The computational complexity of this algorithm equals to the complexity of finding k shortest paths problem which is $O(|E| + |V|log|V|)$.

4.2.3. Improved TCKSP (iTCKSP)

We now further improve the TCKSP algorithm by carefully selecting k clean disjoint paths, which may lead to more disjoint paths. Algorithm 4 shows the pseudo-code of iTCKSP algorithm.

Let $G' = \langle V, E'\rangle$ and $G'' = \langle V, E''\rangle$ be two subgraphs of G. Initially, we set $E' = E$ and $E'' = \emptyset$. $G'_c = \langle V, E' - A\rangle$ is a subgraph of G', where all edges in G'_c are clean edges. We define a function $C(\cdot)$ to denote the maximal number of disjoint paths between s and d in a graph.

The basic idea of this heuristic algorithm is as follows. Initially, all the edges in G are assigned to G'. In each step, we try to remove one clean path in G' and add it in G'' until $C(G'')$ equals to k. Then there are k clean disjoint paths between s and d in G''. Note that both G' and G'' are subgraphs of G and they have no common edges. If $C(G')$ is maximized, the $c_k^m = C(G') + C(G'')$ disjoint paths with $k = C(G'')$ clean paths in G' and G'' can form G_k^m. With such a property, the key challenge of the algorithm is to design heuristic rules to maximize $C(G')$ and assure $C(G'') \geq k$ at last.

This algorithm works in two stages. In the first stage, we try to maximize $C(G')$ and assure $C(G'') \geq k$. At each step of this stage, we try to remove one clean path from

Algorithm 4 The iTCKSP Algorithm.

```
Require: $G = \langle V, E\rangle$, k, polluted edge set A.
Ensure: Maximum disjoint paths with k clean paths.
1: Stage 1: Let $G' = \langle V, E'\rangle$, $G'' = \langle V, E''\rangle$, $G''' = \langle V, E'''\rangle$, $c' = C(G')$; Set $E' = E$,$E'' = \emptyset$,$E''' = \emptyset$;
2: while $C(G'') \leq k$ do
3:   Get the shortest path of $G'_c$: $P_1$;
4:   if $C(\langle V, E' - P_1\rangle) \geq c' - 1$ and $C(G'_c)$ is decreased only 1 after removing $P_1$ then
5:     Update: $E' = E' - P_1$, $E'' = E'' \cup P_1$, $c' - -$;
6:   else
7:     Get an edge $e^*$ from the intersection of $P_1$ and a minimum cut of $G'$; Update: $E' = E' - \{e^*\}$, $E''' = E''' \cup \{e^*\}$;
8:     for each clean edges $e$ in $G'_c$ except $e^*$ do
9:       if $C(G') = c' - 1$ after removing $e$ then
10:        Update: $E' = E' - \{e\}$, $E''' = E''' \cup \{e\}$;
11:      end if
12:    end for
13:    if $C(G''') \geq 1$ then
14:      Get the shortest path of $G'''$: $P_2$; Update: $E' = E' \cup E'''$;
15:      if $C(G'_c)$ is decreased only 1 after removing $P_2$ then
16:        Update: $E' = E' - P_2$, $E'' = E'' \cup P_2$, $E''' = \emptyset$;
17:      else
18:        Randomly select maximal disjoint paths in $G'_c$;
19:        if there exists path $P_2$, all edges in which belong to $G'''$ then
20:          Update: $E' = E' - P_2$, $E'' = E'' \cup P_2$, $E''' = \emptyset$;
21:        else
22:          $c' - -$; $E''' = \emptyset$; continue;
23:        end if
24:      end if
25:      Update $c' = C(V, E' - E'')$;
26:    else
27:      $c' - -$; $E' = E' \cup E'''$; $E''' = \emptyset$; continue;
28:    end if
29:  end if
30: end while
31: Stage 2:Run the max-flow algorithm in $G'$ and $G''$, pick out all edges which have flow on it. These edges with their endpoints form $G_k^m$. *Note that each edge has a unit capacity.*/
32: return $G_k^m$;
```

G' and add it to G'' which minimizes the decrease of $C(G')$. When $C(G')$ only decreases by one after removing the shortest clean path in it and $C(G'_c)$ only decreases by one too, we assign the clean path to G'' directly (Line 3 to 5 in Algorithm 4). Otherwise, we obtain the set of edges after removing which $C(G')$ only decreases by one and use a new graph $G^{''}{}'$ to store these edges temporarily (Line 7–12 in Algorithm 4). Then, we try to get a clean path in $G^{''}{}'$ and add it to G'', after removing which $C(G'_c)$ only decreases by one too (Line 14–20 in Algorithm 4). If we cannot find such a clean path, then we try to find a clean path in G' after removing which $C(G')$ decreases by 2 (Line 22 and 27 in Algorithm 4). We repeat the above process until k clean paths are found in G''. If there exist k clean paths in the original network, $C(G'')$ will reach k at last. In the second stage, we use the max-flow algorithm to find the maximal number of disjoint paths between s and d in both G' and G'', then combine these disjoint paths together to form G_k^m.

The main computational complexity of this algorithm comes from its repeated invocations of the max-flow algorithm and the shortest path algorithm. When we use Ford–Fulkerson algorithm to compute the maximal flow and use Dijkstra algorithm to find the shortest path, the computational complexity of our heuristic algorithm is $O(k(V^2 + c|E|^2))$.

5. The design of NC schemes

In this section, we present a deterministic NC scheme that can achieve the maximal *STR* on a given transmission topology. We also derive the lower bound of the size of finite field for the constructed linear code to be weakly secure. Finally, we analyze the probability that a random NC is weakly secure.

Algorithm 5 shows the pseudo-code to construct a coding matrix Γ. Let the number of independent GEVs obtained by the wiretapper be ϑ and $\vartheta < (r-1)L$. We first choose a vector ζ_1 from $\mathbb{F}_q^{rL} \backslash \bigcup_{i=1}^{r} \langle V_i\rangle$, where V_i is defined in Eq. (1). Then we choose ζ_j from $\mathbb{F}_q^{rL} \backslash \bigcup_{i=1}^{r} \langle \zeta_1, \ldots, \zeta_{j-1}, V_i\rangle$ for $j = 2, \ldots, \vartheta$. According to Lemma 2, such ζ_j exists when q is sufficient large. Then we extend $\{\zeta_1, \ldots, \zeta_\vartheta\}$ to a basis of $\mathbb{F}_q^{rL}$. According to the proof of Lemma 2, a coding matrix can be formed by vectors in the basis as its rows such that the transmission is weakly secure.

From Lemma 2, we know that we can construct a coding matrix if the finite field is large enough. However, a larger finite field can increase the bandwidth consumptions since more binary bits are needed to represent symbols and encoding coefficients. Moreover, a larger finite

Algorithm 5 Algorithm to find coding matrix Γ.

```
Require: the number of data streams $r$, coding interval $L$, thesize of the base field $q$, the number of wiretapped independentencoding vectors $\vartheta$.
Ensure: coding matrix $\Gamma$.
Choose a vector $\zeta_1 \in \mathbb{F}_q^{rL} \backslash \bigcup_{i=1}^{r} \langle V_i\rangle$; /*$V_i$ comes from (1)*/
for $j = 2$ to $\vartheta$ do
  Choose a vector $\zeta_j \in \mathbb{F}_q^{rL} \backslash \bigcup_{i=1}^{r} \langle \zeta_1, \ldots, \zeta_{j-1}, V_i\rangle$
end for
Extend $\{\zeta_1, \ldots, \zeta_\vartheta\}$ to a basis of $\mathbb{F}_q^{rL}$, which forms a matrix $\Gamma$.
return $\Gamma$;
```

field can increase the computational complexity on encoding and decoding. Such a feature is especially important if the nodes have limited computational capability, in which calculating the transformation matrix is a major challenge. With regard to the lower bound of the size of the finite field, we have the following theorem which shows that the size of the finite field is related to the transmission topology.

Theorem 12. *Given the number of data streams r and a transmission topology, suppose that the STR is c_{max} with k clean paths on the given transmission topology and the coding interval is $L = \frac{c_{max}}{(r,c_{max})}$, a lower bound of the size of the finite field such that there exists a coding matrix Γ is $r^{\frac{1}{\theta}}$, where $\theta = (kr/c_{max} - 1)L + 1$.*

Proof. From Eq. (5), we know that we cannot construct a coding matrix such that the transmission is weakly secure unless

$$q > r^{\frac{1}{(r-1)L-\vartheta+1}}, \tag{9}$$

where ϑ is the number of GEVs obtained by the wiretapper. Given a transmission topology where there are c_{max} disjoint paths with k clean paths, we have rL messages to transmit, so ϑ can be shown as

$$\vartheta = \frac{rL}{c_{max}}(c_{max} - k) = rL(1 - k/c_{max}). \tag{10}$$

Substitute Eq. (10) to Eq. (9), then we have $q > r^{\frac{1}{\theta}}$, where $\theta = (kr/c_{max} - 1)L + 1$. □

According to Theorem 12, when r, c_{max} and L are fixed, we can reduce q by increasing k.

Kapil and Krishna [11] show that random coding can also lead to weakly secure. However, there is only one symbol per data stream in their transmission model. In this section, we give the lower bound of the probability that random coding is weakly secure under the transmission model considered in this paper and analyze the relationship between the lower bound and the transmission topology. Different from the deterministic linear coding, the elements of Γ in random coding are randomly chosen from a finite field $\mathbb{F}_q$ instead of computing with an algorithm. The lower bound of probability of random coding being weakly secure is shown in the following theorem.

Theorem 13. *Given an $rL \times rL$ matrix Γ whose elements are randomly chosen from the finite field $\mathbb{F}_q$, if a wiretapper obtains ϑ independent row vectors in Γ, then the probability that the transmission is weakly secure is no less than $\prod_{j=1}^{\vartheta}(1 - \frac{r}{q^{(r-1)L-j}})$.*

Proof. Let Γ_A be the matrix that consists of ϑ row vectors obtained by the wiretapper. The transmission is weakly secure if and only if row space of Γ_A doesn't include vectors like $\beta_i = [0, \ldots, 0, a_{(i-1)L+1}, a_{(i-1)L+2}, \ldots, a_{iL}, 0, \ldots, 0]$, where $1 \le i \le r$, $a_j \ge 0$, $(i-1)L + 1 \le j \le iL$.

Let σ be the number of matrix Γ_A ($\vartheta \times rL$-dimension), with elements in $\mathbb{F}_q$ such that row space of Γ_A doesn't include β_i, $1 \le i \le r$.

$$\sigma \ge (q^{rL} - q^L r)(q^{rL} - q^{L+1} r) \ldots (q^{rL} - q^{L+\vartheta-1} r)$$

where each term of the product is a lower bound for the number of choices of jth row vector ε_j of Γ_A given $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{j-1}$ such that the span $\{\varepsilon_1, \ldots, \varepsilon_j\}$ doesn't include any β_i, $1 \le i \le r$. The number of such matrices with dimension $\vartheta \times rL$ is $q^{\vartheta rL}$. Let P_s be the probability that a random coding scheme is weakly secure, then

$$P_s \ge \frac{\sigma}{q^{rL\vartheta}} = \prod_{j=1}^{\vartheta}\left(1 - \frac{r}{q^{(r-1)L-j+1}}\right). \tag{11}$$

□

Corollary 1. *Given the number of data streams r and a transmission topology, suppose that the STR is c_{max}, coding interval is $L = \frac{c_{max}}{(r,c_{max})}$, and the number of secure path is k. The lower bound of the probability that random coding is weakly secure is $\prod_{i=1}^{rL(1-k/c_{max})}(1 - \frac{r}{q^{(r-1)L-j+1}})$.*

Proof. The result can be obtained by substitute Eq. (10) to Eq. (11). □

According to Corollary 1, when r, c_{max}, L and q are fixed, we can increase k to increase the lower bound.

Suppose that c_{max} is the maximal number of disjoint paths found by the iTCKSP algorithm. We now propose another heuristic algorithm, referred to as iTCKSP-MCP, which aims to find more clean paths (i.e., a larger k) without sacrificing the maximal *STR*, so that the size of the finite field can be reduced and the lower bound of the probability that random coding is weakly secure can be increased.

iTCKSP with More Clean Paths (iTCKSP-MCP): Suppose that we have obtained $c_{max} = c_k^m$ by the iTCKSP algorithm with G' and G'' respectively, where G'' is composed by k disjoint clean paths and G' is composed by the other part of G. The basic idea of iTCKSP-MCP algorithm is to find $c_{max} - k$ disjoint paths with more clean paths in G'. Specifically, if a clean path satisfies the condition that the capacity of G' is decreased only by 1 when it is taken away, we call it as *augmenting clean path*. We may first find an augmenting clean path in G'. We then try to find another augmenting clean path in G' after removing the found augmenting clean path and so on. This process will continue until no augmenting clean path can be found. Such strategy leads to more clean paths in G without changing the maximal *STR*. We suppose that the number of augmenting clean paths is k', $k' \ge 0$. Since there are k disjoint clean paths in G'', if we can find $c_{max} - k$ disjoint paths with k' disjoint clean paths in G', then we can find c_{max} disjoint paths in G with $k + k'$ disjoint clean paths.

The unicast transmission topology can be formed by c_{max} disjoint paths in G with $k + k'$ disjoint clean paths. Let the maximal number of disjoint clean paths between s and d in G be λ. The computational complexity of iTCKSP-MCP algorithm is $O(\lambda(V^2 + c|E|^2))$.

6. Simulation

To evaluate the proposed TCKSP and iTCKSP algorithms, we conduct extensive simulation experiments. In the rest of this section, we first introduce two different types of network topology model. We then run the proposed algorithms on them and present the simulation results.

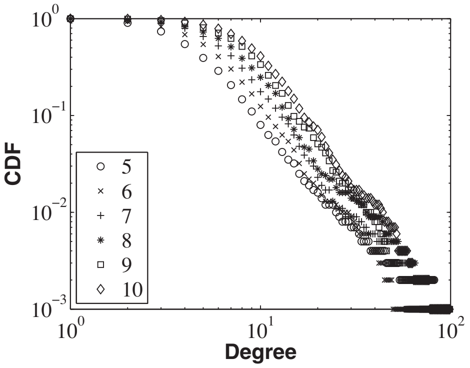


Fig. 6. The CDF of node degree for PA models with different average node degrees on log–log scale.

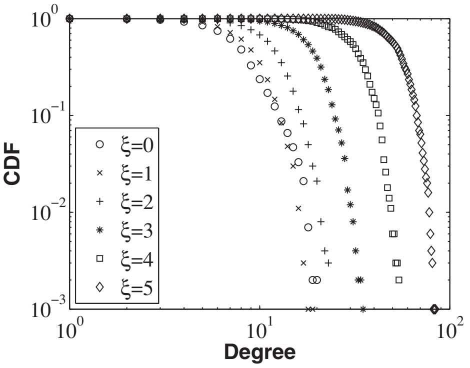


Fig. 7. The CDF of node degree for GRG models with different ξs on log–log scale.

6.1. Network topology models

In [20], it has been shown that network topology generators based on node degree are much better than that based on structure. According to the node degree distribution, network topology models can be classified into two types. The internet topology has been proved to follow power law distribution[21,22], while wireless AdHoc networks have been proved to follow normal distribution[23,24]. Thus, we choose two typical topology models for our simulation, PA model[22] and AdHoc model[23], which follow the power law distribution and the normal distribution respectively.

The PA topology model has an approximate power-law degree distribution and higher likelihood compared with other models in [22]. The process of generating PA topology is as follows: begin with 3 fully connected nodes, then in successive steps add one new node to the graph, such that this new node is connected to an existing node with probability proportional to the current node degree. Then add additional links successively by choosing a node randomly and connecting it to the other nodes with probability proportional to the current node degree. When the number of nodes is fixed, the link density can be decided by the link number or the average node degree of the topology model. Fig. 6 shows the node degree distribution of the PA model with different average node degrees on log–log scale.

The AdHoc topology model is designed for wireless AdHoc networks based on geometric random graph. In this model, a wireless AdHoc network consists of a number of nodes or radio devices spread over a certain geographic area randomly. Two nodes i and j are connected with the probability $p(r_{ij})$, where r_{ij} is the distance between i and j. $p(\hat{r}) = \frac{1}{2}[1 - erf(3.07\frac{ln(\hat{r})}{\xi})]$, $\xi \triangleq \frac{\sigma}{\eta}$, which is derived by the log-normal shadowing model and is shown in expression (3) of [23]. A high value of ξ corresponds with stronger shadowing effects and higher link density. Fig. 7 shows the CDF of degree for AdHoc model with different ξ on log–log scale.

Since directed and acyclic graph is needed in our network model as mentioned in Section 2.1, we assume that the direction in PA and AdHoc is from low id nodes to high id nodes, while each node id is assigned successively when the node is added to the topology. A link is labeled as a polluted link with probability p_p. In the following, we first evaluate the performance of the TCKSP and iTCKSP algorithms on finding G_k^m, after that, we evaluate the STR gain from the secure network coding.

6.2. Performance of the TCKSP and iTCKSP algorithms

We evaluate the performance of the TCKSP algorithm and the iTCKSP algorithm on finding G_k^m by comparing them with the baseline algorithm BMF.

Given a topology G, node pair $< s, d >$, the capacity of G between s and d is an upper bound of the capacity of G_k^m between s and d. In the simulation, we calculate the average of the difference between the capacity of G_k^m for each proposed heuristic algorithm and the capacity of G over the capacity of G, referred to as the *relative error of secure capacity*, which is simplified to *resc*. The smaller the *resc* is, the G_k^m found by the heuristic algorithm is closer to the optimum.

For each combination of the number of secure paths k, the polluted probability p_p and the topology density (average node degree in PA and ξ in AdHoc), we generate 5 topologies, randomly choose 50 node pairs for each topology, and run different algorithms to find G_k^m for each node pair. At last, we calculate the average *resc* to depict the simulation figure. For computation complexity consideration, the node number of the topology is always set to 1000.

In Figs. 8 and 9, we show the variation of *resc* versus k in the PA model and AdHoc model respectively. In these experiments, p_p is set to 10%, the average node degree of PA is set to 9 and the ξ of AdHoc model is set to 2. For fairly comparison, we choose 50 node pairs for each topology model, such that each source-destination pair has no less than 6 maximal clean disjoint paths. Then, we find out $G_k^m(k = 1, 2, \dots, 6)$ by the proposed algorithms for each node pair, and calculate the average *resc* for each k. We can see that the iTCKSP algorithm consistently outperforms TCKSP and BMF algorithms. The increase of k makes it harder to find k optimal disjoint clean paths, and the capacity of the optimal G_k^m decreases as k increase. Thus, the *rescs* of both topology models increase as k increase.

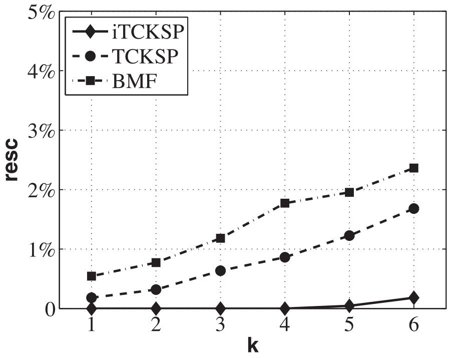


Fig. 8. *resc* vs. *k* in PA model.

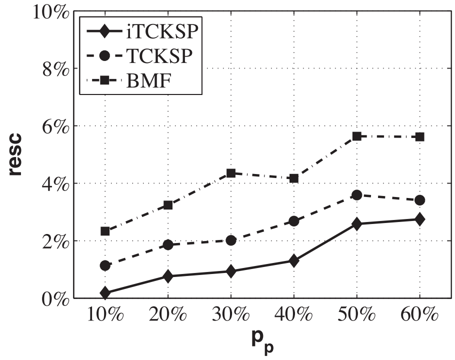


Fig. 11. *resc* vs. p_p in AdHoc model.

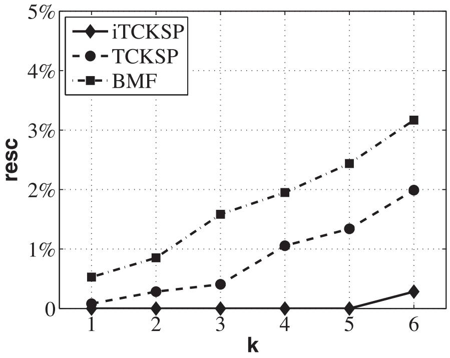


Fig. 9. *resc* vs. *k* in AdHoc model.

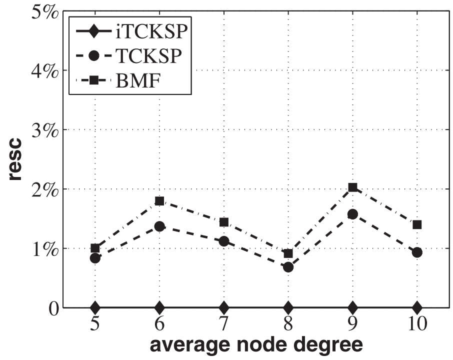


Fig. 12. *resc* vs. *average node degree* in PA model.

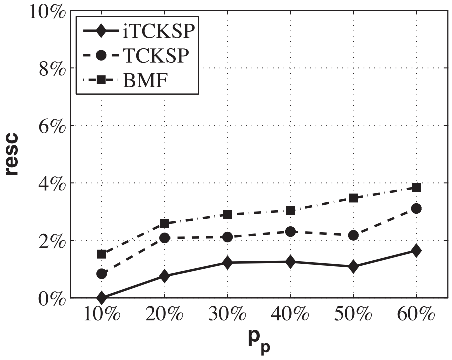


Fig. 10. *resc* vs. p_p in PA model.

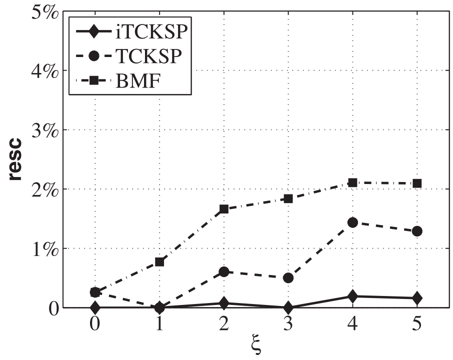


Fig. 13. *resc* vs. ξ in AdHoc model.

The *resc* in the AdHoc model is higher than that in the PA model, this is because the AdHoc model is more complicated than the PA model with similar average node degree, so it is harder to find optimal disjoint clean paths especially for BMF and TCKSP.

In Figs. 10 and 11, we show the variation of *resc* versus p_p in PA model and AdHoc model respectively. In these cases, the average node degree of PA model is set to 9 and ξ of AdHoc model is set to 2. p_p is changed from 10–60%. We choose 50 node pairs for each topology model, such that each pair has more than 3 maximal clean disjoint paths. Then we run the proposed algorithms to find G_3^m for each value of p_p. We can see that the iTCKSP algorithm consistently outperforms TCKSP and BMF algorithms. Moreover, a larger p_p results in more polluted edges, which implies that it is harder to find optimal disjoint clean paths. Thus, the *resc*s of both topology models increase as p_p increase.

In Figs. 12 and 13, we show the variation of *resc* versus the *average node degree* in PA model and ξ in AdHoc model respectively, which demonstrate the relationships between *resc* and the network density. In these cases, p_p is set to 10%. The average node degree of PA model is changed from 5 to 10, and ξ of AdHoc model is changed from 0 to 6. We choose 50 node pairs with more than 3 maximal clean disjoint paths. Then we run the proposed algorithms to find G_3^m for each value of *average node degree* and ξ. We can see that iTCKSP algorithm consistently outperforms TCKSP and

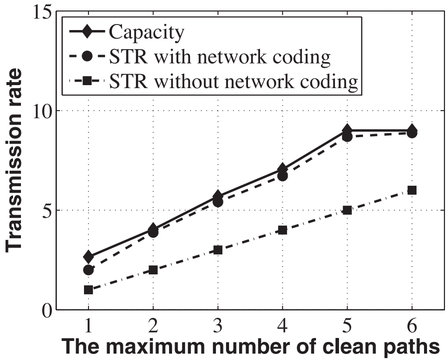


Fig. 14. Coding scheme vs. No coding scheme in PA model.

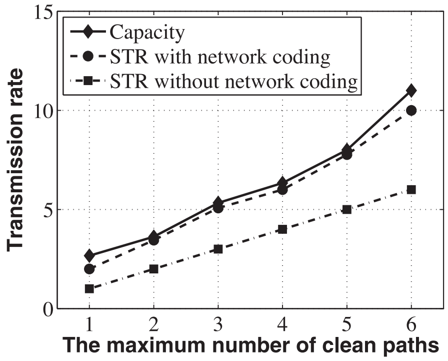


Fig. 15. Coding scheme vs. No coding scheme in AdHoc model.

BMF algorithms. In AdHoc model, the increase of ξ results more complicated AdHoc models, which makes the finding of the optimal disjoint clean paths more difficult. In PA model, due to the power-law node degree distribution, the increase of the average node degree mostly increases the degree of a small number of nodes, thus the complexity of PA model doesn't change too much. Algorithm *iTCKSP* can always find the optimal G_3^m, while *TCKSP* and *BMF* fluctuate due to the random cause.

6.3. *The gain of* STR *from NC*

In this section, we evaluate the performance of achievable secure transmission rate by using approaches with and without NC. In Figs. 14 and 15, p_p is set to 30%, the average node degree of PA model is set to 9 and ξ of AdHoc model is set to 2. For each model, We randomly generate 5 topologies, select 50 node pairs from each topologies, and classify them by the maximum number of clean paths between each node pair. We then use the iTCKSP algorithm to find all G_k^ms and decide the best k to achieve the maximal *STR* according to Theorem 6 for each state of the maximum number of clean paths. From both figures, We can see that the *STR* with NC is very close to the upper bound capacity and much higher than the *STR* without NC.

7. Related work

For secure NC against passive attacks, besides weakly secure, there is another major secure model, namely, *information theoretical secure* (ITS). In the ITS model, a transmission is secure if the attacker cannot obtain *any* information of the original messages. To fulfill such a requirement, random numbers must be included in the coding process. To achieve the ITS requirement, in [9], Cai and Yeung gave a sufficient condition for finding an admissible code to protect the message from being decoded if a set of channels can be accessed by a wiretapper. The requirement is also considered by Fedman et al. in [10], in which they showed that the problem of finding a secure network code is the same as finding a block code with certain distance properties. In this paper, we only focus on the weakly secure model. However, we believe that our approach can be extended to address the ITS model.

From the perspective of traffic pattern, most existing studies addressed multicast [9–11]. As a special case of secure multicast, secure unicast routing is studied in [25], in which the authors considered a single unicast flow over a cyclic network under the information theoretical secure model. In this paper, we consider weakly secure NC for unicast with multiple streams. [14,15,26] studied how to use network coding to deal with the eavesdropping on noncooperative nodes, different with them, this paper studies how to use network coding deal with the eavesdropping on cooperative edges. Several recent studies deal with eavesdropping on cooperative edges by network coding[27–29], however, all of them didn't consider the impacts of transmission topology on STR.

There are previous work that focuses on transmission topology problems to minimize transmission cost [30] or to maximize transmission rate [31,32]. However, these studies do not consider the security requirement.

8. Conclusion

In this paper, we have investigated the optimal design of weakly secure NC under *wiretapping attack*, where we focused on the scenario that there are multiple unicast streams between the same source and destination nodes. Our objectives include (1) maximizing the *STR* under the *weakly secure* requirement, and (2) minimizing the size of the finite field, on which the weakly secure NC is defined. To address the issue, we have applied a novel approach that integrates weakly secure NC design with transmission topology construction. In particular, we first defined the problem and analyzed its behaviors, including the characteristics of an optimal transmission topology, the maximal *STR*, and the NP-hardness of the problem. Based on the understandings of the problem, we developed an optimal algorithm that is practically solvable when the capacity between the source and destination nodes is small, and we developed two efficient heuristic algorithms for general case to achieve the above two objectives. We then devised deterministic and random coding schemes that can achieve the maximal *STR*, given a transmission topology, where we also studied the relationship between the transmission topology and two major system factors: (1) the size of the finite field, and (2) the probability of a random code is weakly secure. Then, another heuristic algorithm is developed for the transmission topology which can reduce the size of the finite field and increase the probability of a random code is weakly secure. Finally, we have conducted extensive simulation experiments and the results show that the the proposed heuristic algorithms achieve good performance in various scenarios.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant no. 61202378, no. 61202350, and no. 61572310. It was also supported in part by China Postdoctoral Science Foundation (nos. 2013M531402, 2012M521080, 2014T70544), the Application Foundation Research of Suzhou of China no. SYG201401, the General Research Fund from Hong Kong Research Grant Council no. 122913, and the Shanghai Oriental Scholar Program.

References

[1] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network information flow, IEEE Trans. Inf. Theory 46 (4) (2000) 1204–1216, doi:10.1109/18.850663.

[2] R. Koetter, M. Medard, An algebraic approach to network coding, IEEE/ACM Trans. Netw. 11 (5) (2003) 782–795, doi:10.1109/TNET.2003.818197.

[3] S. Li, R. Yeung, N. Cai, Linear network coding, IEEE Trans. Inf. Theory 49 (2) (2003) 371–381, doi:10.1109/TIT.2002.807285.

[4] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, D. Karger, Byzantine modification detection in multicast networks using randomized network coding, in: Proceedings of IEEE International Symposium on Information Theory (ISIT), 2004, p. 144, doi:10.1109/ISIT.2004.1365180.

[5] M. Krohn, M. Freedman, D. Mazieres, On-the-fly verification of rateless erasure codes for efficient content distribution, in: Proceedings of IEEE Symposium on Security and Privacy, 2004, pp. 226–240, doi:10.1109/SECPRI.2004.1301326.

[6] C. Gkantsidis, P.R. Rodriguez, Cooperative security for network coding file distribution, in: Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM), 2006, pp. 1–13, doi:10.1109/INFOCOM.2006.233.

[7] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, Resilient network coding in the presence of byzantine adversaries, in: Proceedings of the 26th Conference on Computer Communications (INFOCOM), 2007, pp. 616–624, doi:10.1109/INFCOM.2007.78.

[8] Z. Yu, Y. Wei, B. Ramkumar, Y. Guan, An efficient Signature-Based scheme for securing network coding against pollution attacks, in: Proceedings of the 27th Conference on Computer Communications (INFOCOM), 2008, pp. 1409–1417, doi:10.1109/INFOCOM.2008.199.

[9] N. Cai, R. Yeung, Secure network coding, in: Proceedings of IEEE International Symposium on Information Theory (ISIT), 2002, p. 323, doi:10.1109/ISIT.2002.1023595.

[10] J. Feldman, T. Malkin, C. Stein, R.A. Servedio, On the capacity of secure network coding, in: Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing, 2004.

[11] K. Bhattad, K.R. Narayanan, Weakly secure network coding, in: Proc. of the First Workshop on Network Coding, Theory, and Applications(NetCod), Riva del Garda, Italy, 2005.

[12] N. Cai, R.W. Yeung, A security condition for Multi-Source linear network coding, in: Proceedings of IEEE International Symposium on Information Theory (ISIT), 2007, pp. 561–565, doi:10.1109/ISIT.2007.4557284.

[13] R. Yeung, N. Cai, On the optimality of a construction of secure network codes, in: Proceedings of IEEE International Symposium on Information Theory (ISIT), 2008, pp. 166–170, doi:10.1109/ISIT.2008.4594969.

[14] J. Wang, J. Wang, K. Lu, B. Xiao, N. Gu, Optimal linear network coding design for secure unicast with multiple streams, in: Proceedings of the 29th Conference on Computer Communications (INFOCOM), San Diego, CA USA, 2010.

[15] J. Wang, J. Wang, K. Lu, Y. Qian, B. Xiao, N. Gu, Optimal design of linear network coding for information theoretically secure unicast, in: The 30th IEEE International Conference on Computer Communications (IEEE INFOCOM), 2011, pp. 757–765, doi:10.1109/INFCOM.2011.5935296.

[16] J. Wang, J. Wang, K. Lu, B. Xiao, N. Gu, Modeling and optimal design of linear network coding for secure unicast with multiple streams, IEEE Trans. Parallel Distrib. Syst. 24 (10) (2013) 2025–2035.

[17] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman Co., 1979.

[18] B. Yang, S.-Q. Zheng, S. Katukam, Finding two disjoint paths in a network with min-min objective function., in: Proceedings of the Fifteenth IASTED International Conference on Parallel and Distributed Computing and Systems, vol. 1, 2003, pp. 75–80.

[19] Y. Shiloach, Y. Perl, Finding two disjoint paths between two pairs of vertices in a graph, J.ACM (JACM) 25 (1) (1978) 1–9, doi:10.1145/322047.322048.

[20] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, Network topology generators: Degree-based vs. structural, ACM SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 147–159.

[21] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology, in: Proceedings of ACM SIGCOMM Computer Communication Review, vol. 29, ACM, 1999, pp. 251–262.

[22] L. Li, D. Alderson, W. Willinger, J. Doyle, A first-principles approach to understanding the internet's router-level topology, in: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, in: SIGCOMM '04, ACM, New York, NY, USA, 2004, pp. 3–14, doi:10.1145/1015467.1015470.URL http://doi.acm.org/10.1145/1015467.1015470

[23] P.V.M.R. Hekmat, Degree distribution and hopcount in wireless adhoc networks, in: Proceedings of IEEE International Conference on Networks (ICON), 2003, pp. 603–609, doi:10.1109/ICON.2003.1266257.

[24] R. Hekmat, P. Van Mieghem, Connectivity in wireless ad-hoc networks with a log-normal radio model, Mob. Netw. Appl. 11 (3) (2006) 351–360.

[25] K. Jain, Security based on network topology against the wiretapping attack, IEEE Wirel. Commun. 11 (1) (2004) 68–71, doi:10.1109/MWC.2004.1269720.

[26] J. Xu, B. Chen, Secure coding over networks against noncooperative eavesdropping, IEEE Trans. Inf. Theory 59 (7) (2013) 4498–4509.

[27] Y. Wei, Z. Yu, Y. Guan, Efficient weakly-secure network coding schemes against wiretapping attacks, in: Proceedings of 2010 IEEE International Symposium on Network Coding (NetCod), IEEE, 2010, pp. 1–6.

[28] M. Adeli, H. Liu, On the inherent security of linear network coding, IEEE Commun. Lett. 17 (8) (2013) 1668–1671.

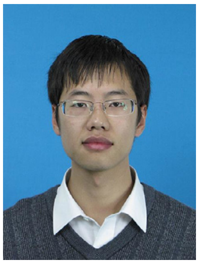
[29] Z. Cao, S. Zhang, X. Ji, L. Zhang, Secure random linear network coding on a wiretap network, AEU-Int. J. Electron. Commun. 69 (1) (2015) 467–472.

[30] D.S. Lun, N. Ratnakar, M. Medard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, F. Zhao, Minimum-cost multicast over coded packet networks, IEEE/ACM Trans. Netw. 14 (SI) (2006) 2608–2623.

[31] Z. Li, B. Li, Efficient and distributed computation of maximum multicast rates, in: Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005. IEEE, vol. 3, 2005, pp. 1618–1628, doi:10.1109/INFCOM.2005.1498444.vol. 3

[32] Z. Li, B. Li, D. Jiang, L.C. Lau, On achieving optimal throughput with network coding, in: Proceedings of the 24th Conference on Computer Communications (INFOCOM), vol. 3, 2005, pp. 2184–2194, doi:10.1109/INFCOM.2005.1498493.vol. 3

Xiangmao Chang received the B.S. degree in mathematics from Liao Cheng University, China, in 2004, the M.S. degree in mathematics from Beijing Jiaotong University, China, in 2007, and the Ph.D. degree in computer science from Beijing University of Posts and Telecommunications, China, in 2011. He is currently an associate professor of computer science with the Nanjing University of Aeronautics and Astronautics. His research interests include network coding and wireless sensor networks.



Jin Wang received the B.S. degree in information and computation science from Ocean University of China in 2006, and the Ph.D. degree in computer science jointly awarded by City University of Hong Kong and University of Science and Technology of China in 2011. He is currently an associate professor at the Department of Computer Science and Technology, Soochow University, China. His research interests include network coding, network security, service-oriented networking and information-centric networking.

Jianping Wang is currently an associate professor in the Department of Computer Science at City University of Hong Kong. She received her B.Sc. and M.Sc. degrees from Nankai University in 1996 and 1999 respectively, and her Ph.D. degree from University of Texas at Dallas in 2003. Her research interests include Dependable Networking, Optical Networking, Service Oriented Wireless Sensor/Ad Hoc Networking.

Kejie Lu received the B.Sc. and M.Sc. degrees in Telecommunications Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Dallas, Richardson, TX, USA, in 2003. In July 2005, he joined the Department of Electrical and Computer Engineering, University of Puerto Rico, Mayagez, PR, USA, where he is currently an Associate Professor. Since January 2014, he has been an Oriental Scholar with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China. His research interests include architecture and protocol design for computer and communication networks, performance analysis, network security, and wireless communications.

Yi Zhuang received the B.S. degree in Computer Science from Nanjing University of Aeronautics and Astronautics, China, in 1981. She is currently a professor and Ph.D. supervisor of computer science with the Nanjing University of Aeronautics and Astronautics. Her research interests include network and distributed computing, and information security.