科技部補助

大專學生研究計畫研究成果報告

＊ ********* ＊＊ ＊
＊ 計 畫 ＊
＊ ： Interactive Secure Network Coding ＊
＊ 名 稱 ＊
＊ ********* ＊＊ ＊

執行計畫學生： 柯劭珩
學生計畫編號： MOST 103-2815-C-002-063-E
研 究 期 間 ： 103 年 07 月 01 日至 104 年 02 月 28 日止，計 8 個月
指 導 教 授 ： 王奕翔

處理方式： 本計畫可公開查詢

執 行 單 位： 國立臺灣大學電信工程學研究所

中華民國 104 年 03 月 31 日

Interactive Secure Network Coding

Report of the College Student Research Training Fellowship, MST

Shao-Heng Ko
Department of Electrical Engineering
National Taiwan University

***Abstract*—This is a report of the college student research training fellowship sponsored by the Ministry of Science and Technology. This report focus on secure network coding on interactive flow networks with various traffic patterns, such as single-group / multi-group, and two-way / multi-way networks. With the aid of key-distributing, the capacity bound and a complete transmission scheme of the single-group, two-player network is found, and methodology for solving the multi-player problem is proposed. We utilize a key-distribution algorithm to precode symbols to be used in existing secure network coding / network coding algorithms. Also, we present a transmission scheme to a stronger version of problem for the single-group, two-way network, as along as some insights for solving multi-group networks.**

Keywords—network coding, secure network coding, interactive networks, key-recycling, key-distribution.

I. INTRODUCTION

This report bases on the model for secure network coding with uniform wiretap sets presented by Cai and Yeung in [1]. In the model, a flow network is represented by a directed graph $G = (V, E)$ with nodes correspond to communication units and edges correspond to channels. We focus on security issues on such model, and thus all channels are assumed to be noiseless.

Channels in such networks are vulnerable to mischievous eavesdroppers. Given that an eavesdropper may attack any possible edge sets and obtain complete information about the symbols flowing on it, the sender may encode the messages with randomness, in order to assure that the eavesdropper gains no information about the sensitive messages by wiretapping. The model given in [1] has these assumptions:

(1) All channels are vulnerable to be eavesdropped.

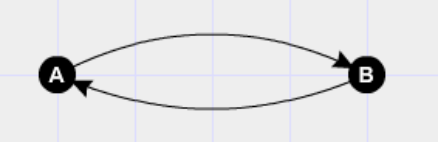
(2) All randomness used to encode messages are generated at the source node.

(3) All channels have unit capacity. This is reasonable since one can partition any given channel by time-division.

(4) No cycle exists in such network.

In [1], under a single-source, single-destination acyclic wiretap network, the possible throughput region from the source to the destination through secure network coding is found. By the assumption of acyclic networks, it is convenient to partially order the whole network, and develop simple algorithms to achieve optimal throughput on them. However, the network in practice is seldom acyclic. Taking into consideration the interaction of user nodes, it may be able to make better use of the secure network coding tool and further improve the performance of a cyclic network.

The idea of key-recycling and key-distribution is rather simple. In a wiretap network, the network resource is divided, conceptually, into those we need to deliver the necessary randomness (secure keys), and the rest to actually deliver the sensitive information. From the nature of a wiretap network problem, the cost of key delivering is inevitable; but once we can make the randomness function more efficiently, we then reduce the resource needed to generate and deliver them, and finally leave more resource available for the data.



Here is a simple example for key-recycling. Consider a wiretap network with only two user nodes A and B, and two single unit channel between them, with the opposite direction. Suppose a wiretapper can compromise at most one channel. With the view of secure network coding in [1], we may see the model as two disjoint acyclic network, but we will also find that no possible secure information flow can happen between the two nodes this way, since each link can be vulnerable to attack.

However, there does exist a quite naïve method to improve from that using key recycling: in time slot T, a secure key K_T is generated at A and directly passed from A to B. A message packet M_T is simultaneously encoded by $M'_T = M_T + K_{T-1}$ and is passed from B to A. We create a constant flow from B to A by 1 unit, obtaining a better result only by utilizing the interaction between the two nodes. Key K_T is of no use at the time it is created; but instead of being abandoned, it is recycled and used at the very next time slot.

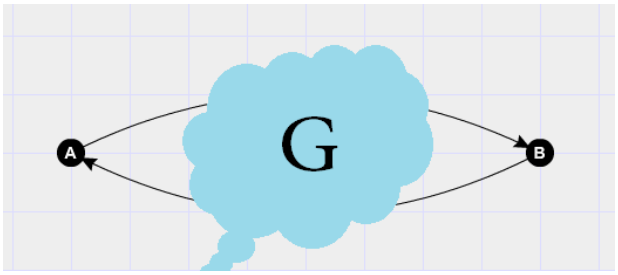
In a multi-way network with more than two user nodes, when desired to recycle the keys and using them for traffic among the nodes, one will encounter the need to pass the keys around the network. Then there comes the idea of key-distribution, which is a problem itself to be solved.

In this work, an improved throughput region for a single-group, two-way wiretap network is found explicitly, with a method to implement the transmission scheme. Also in this type of network, I present a strong implementation that solves even a more difficult condition, in which the wiretapper can

vary its attack destination from time to time. In a single-group, multi-user network, a cost bound for distribution is found, with a proposal of a "distribute-and-recycle" protocol to use on such network.

II. Problem Formulation

A. Single-Group, Two-Way Network



A single-group two-way wiretap network consists of the following components:

1) *Directed multi-graph flow network* $G=(V,E)$, where V and E are respectively the node set and the edge set of G.
2) *User nodes* $A,B\subseteq V$, at which the random messages are only allowed to be generated.
3) *Compromised channels sets* C : $\forall c\in C, c\subseteq E$, where c is a set of compromised channels that a wiretapper can listen to. Which c is the particular one is not known by the designer.
4) *Secrecy constraint* R : $\forall c\in C, |c|\le R$, which means a wiretapper gathers information from at most R channels. Throughout this report, we simply say that all channels in a network are equally like to be wiretapped, which leads to C consisting of merely every subset of size R of E. This equals to that the wiretapper simply chooses R channels to access.

In such a network, we are most interested in two particular throughput values r_{AB} and r_{BA} and their sum. Suppose that A generates message sequence M_A and B generates M_B, and after network coding, A receives $\hat{M}_A$ and B receives $\hat{M}_B$. Denoting the symbol W to be the sequence that the wiretapper obtains, we can then define:

$$\begin{aligned} r_{AB} &= \max\left\{I(M_A;\hat{M}_A)\right\} \\ r_{BA} &= \max\left\{I(M_B;\hat{M}_B)\right\} \\ I(W;M_A) &= I(W;M_B)=0 \end{aligned} \quad , \qquad (1)$$

B. Single-Group, Multi-Way Network

A single-group multi-way wiretap network is much like the two-way version of it. Instead of user nodes $A,B\subseteq V$, now there are k vertices A_1 to A_k in V that qualifies as user nodes. The other settings, including the compromised channel sets and the secrecy constraint, still hold. We are still interested in the total throughput limit here, as well as the throughput limit between any two given user nodes. However in a large multi-way network the throughput between user pairs influence each other, so it is the total throughput that gives most sense.

III. Main Results on Single-Group, Two-Way Networks

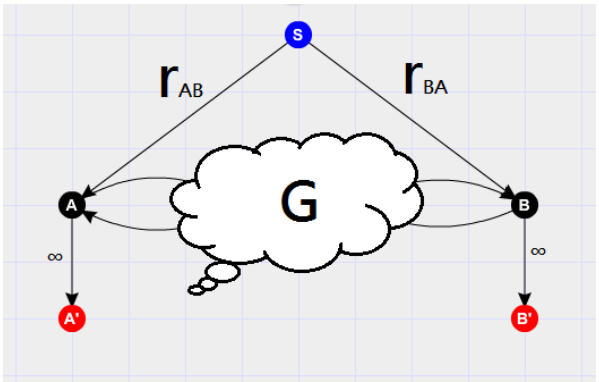
A. An equilavent model and the capacity Region

In this section, we conceptually claim that every arbitrary network in this model is equivalent to a much simpler network with

$$V=\{A,B\}$$

$$E=\{E_{AB}i, E_{BA}j \mid i=1,2,...c(A;B),\ j=1,2,...c(B;A)\}.$$

We start from an arbitrary single-group, two-way network, setting $R=0$, thereby ignoring the security issue.

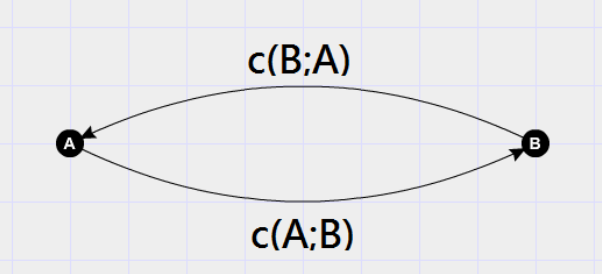


Without altering the already existing network structure, we add a virtual source S and two virtual destinations A',B'. Connect $(S,A),(S,B),(A,A'),(B,B')$, all with infinite capacity. This qualifies as a single-source multicast problem. Suppose when the multicast reaches its maximum, the traffic between S and A is r_{AB}, and that between S and B is r_{BA}. We can claim that r_{AB} is the maximum flow size from A to B through the original network. If it is not, S can multicast more information through A; since: (i) the capacity of (S,A) and (A,A') are infinity, so the path S-A-A' is not fully used; and (ii) by the assumption, S-A-B-B' is not fully used either. The above will imply that r_{AB} is not the maximum flow on the edge from S to A when the multicast is at maximum, a contradiction.

Likewise, r_{BA} is the maximum flow size from B to A.

Now, from the theorem of single-source multicast problem in [2], we obtain the following:

$$r_{AB} \le \min\{c(A;A'), c(A;B')\} = c(A;B)$$
$$r_{BA} \le \min\{c(B;A'), c(B;B')\} = c(B;A)$$
$$r_{AB} + r_{BA} \le c(S;A',B') = c(A,B;A',B') = \infty$$



A simple network like this satisfies the result. With network coding, we can achieve everything as in this simple network, in the case $R = 0$. Also, from [1], when it comes to secure coding, it suffices to precode the message symbols with random keys before the phase of network coding; that is, a wiretap network with secrecy constraint r can be viewed as a coding network with r default messages to send. Given a field size sufficiently large, one can use r secret keys to precode m messages; those encoded messages, along with the keys, are no different from $m+r$ plain messages when we decide if they can pass through the network by network coding.

Thus, any arbitrary network in our model is "equivalent" to a simple network with only direct links with capacity of those Min-cuts in the original network.

Now we are interested in the fundamental limit for throughputs that assures perfect secrecy. Analyzing this equivalent model yields to the capacity region below:

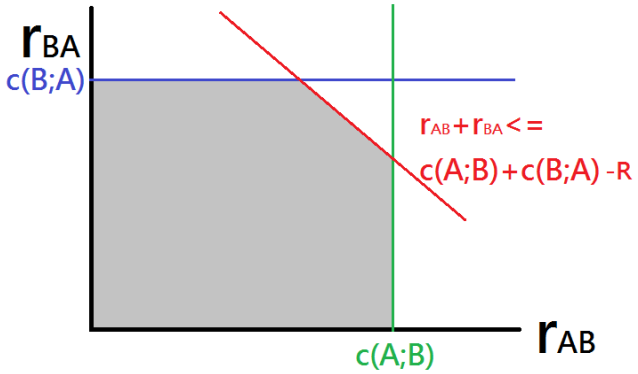
$$r_{AB} \le c(A;B) \quad (2)$$
$$r_{BA} \le c(B;A) \quad (3)$$
$$r_{AB} + r_{BA} \le c(A;B) + c(B;A) - R \quad (4)$$

B. Converse of the Capacity Region

In the above region bound, (2) and (3) are exactly the outer bound from information theory in network coding. A network coding scheme with secrecy certainly does not violate the information bounds.

To see that (4) is the tight outer bound for perfect secrecy, we first observe that $r_{AB} + r_{BA} \le c(A;B) + c(B;A)$ in the case $R = 0$; this is the outer bound from network coding. Now the wiretapper can see up to R channels, so he can obtain R channels that directly begin at one of the two user nodes; therefore the mutual information he gains with respect to the transmitted packets is up to R. If there exists some functions of the transmitted packets that are of perfect secrecy, it cannot has dimension more than $c(A;B) + c(B;A) - R$.

C. Achievability and a Transmission Scheme to Achieve the Capacity Region



In the figure above, the shadowed area shows the achievable throughput region for our model. Consider the two corner points on the boundary, $P_1(c(A;B), c(B;A) - R)$ and $P_2(c(A;B) - R, c(B;A))$; if we can achieve these two particular pairs of throughputs, we will be able to achieve anywhere in the achievable area by time-dividing.

Due to the symmetry of our simplified network, we can assume without loss of generality $c(A;B) \ge c(B;A)$. Three cases follows.

(i) $c(A;B) \ge c(B;A) \ge R$

This is the easiest case to address. At time slot T, A generates R secret keys K_{t1} to K_{tR} and directly send them through the network to B. These R secret keys suffices to encode all the messages that B is to send to A in the next time slot; meanwhile, they are also used to encode all messages that A sends to B, in the same time slot. All the rest part of the network can be utilized to transmit encoded messages, so the throughput pair P_1 is achieved. Also, P_2 can be achieved by letting B generate the keys, and reverse the transmission scheme above.

(ii) $c(A;B) \ge R \ge c(B;A)$

In this case P_1 can be achieved by the same method as in (i). However, B is now unable to generate all of the randomness itself since the lacking of outflow capacity. In addition to $c(B;A)$ keys generated by B per time slot, A has to generate $R - c(B;A)$ more to make a total of R secret keys. Each time slot A precodes the messages with the $R - c(B;A)$ keys generated in the current time slot plus the $c(B;A)$ keys received from B in the previous time slot. This results in $c(A;B) - [R - c(B;A)]$ being the

available throughput from A to B, and no throughput from B to A. This corresponds to P_2.

(iii) $R \geq c(A;B) \geq c(B;A)$

We can directly apply the method to reach P_2 above in (ii). In this case none of the two parties has the ability to offer all the randomness, but once $c(A;B)+c(B;A) \geq R$ – and we will assume it is the case – we can let the two parties coordinate and achieve P_1 and P_2.

After the discussions above, we conclude that P_1 and P_2 are really achievable, therefore any point in the grey area is feasible. The region is bounded by (2), (3) and (4). In all, R secret keys need to be generated each time slot, and it is not important that who does the work. The rest and unused part of the network is then available to transmit messages encrypted by the keys. This phenomenon takes place in all of the network types we consider.

D. Transmission Scheme to a Modified Problem Setting

A slightly different, and difficult, setting is to change c into $c(t)$; that is, the compromised channel set may change from time to time. $|c(t)| \leq R$ still holds, so the amount of compromised channels are still bounded.

Our previous transmission scheme doesn't bode well here. Apply our previous method on this setting, there is a possibility that a wiretapper happens to see a set of secret keys at time slot k and those messages encrypted by them at time slot $k+1$. Suddenly the wiretapper is able to decrypt the messages! We need to slightly adjust our protocol to prevent this kind of information leaking.

Before we continue, we view the original problem in a more conceptual way: information dimensions. In [1], the main result is that if network coding gives some capacity limit C, then given the security constraint R, the throughput $C-R$ is achievable to permit perfect secrecy, by precoding and then random coding. In the previous section, our capacity limit in network coding is totally $c(A;B)+c(B;A)$; then we get the limit $r_{AB}+r_{BA} \leq c(A;B)+c(B;A)-R$. Between the keys and the messages encrypted by them, the eavesdropper can only manage to see R dimensions of data; he is unable to solve the rest. Hence by properly precoding the messages, prefect secrecy is permitted. Now the eavesdropper is able to change its target immediately between time slots; since that a group of keys are only used until the next time slot, now he can see up to $2R$ dimensions of data.

A simple idea is to twice the security constraint to $2R$ and encrypt the messages with two times of secret keys. This method certainly works but narrows the throughput region; the total throughput limit is now $c(A;B)+c(B;A)-2R$. In a network where $R << c(A;B), c(B;A)$ the influence is rather small. But in fact we can maintain the throughput region by adding some shared secrets before the transmission starts.

It is easier to illustrate the idea in an example. Consider a simple network with two unit capacity channels from A to B and one from B to A. Setting $R=1$, the eavesdropper sees one arbitrary channel every time slot. Ignoring the interaction, we first activate only the A to B channels. By the result in [1] $r_{AB}=2-1=1$, it is feasible to send a symbol x from A to B with perfect secrecy, in the first time slot.

Now A and B share a secret symbol x, which the eavesdropper doesn't know anything about; from time slot 2, if we want to achieve throughput pair $(1,1)$, the original transmission scheme will have A send K_2 and $(M_{AB2}+K_2)$ to B. Now A sends $x+K_2$ and $(M_{AB2}+K_2)+x$ to B instead. The eavesdropper cannot solve K_2 now since he lacks x. In time slot 3 B sends back $(M_{BA3}+K_2)$ to A. Even if the eavesdropper sees $x+K_2$ before, he is still unable to solve the message in time slot 3.

Meanwhile, $x+K_3$ is the new key sent by A in time slot 3. The eavesdropper cannot separate x and K_3 again; therefore K_3 is reused in time slot 4 to encrypt M_{BA4}. As time goes on, the eavesdropper never gets to know the secret symbol x. He is then unable to separate any keys alone, and all the messages encrypted by the keys are safe. After time slot T the eavesdropper will have accumulated $T-1$ packages that contain x and the keys; but there are T different symbols x, K_2, K_3 … K_{T-1} among them. The eavesdropper will never solve any one symbol above. After a long enough period of time the average throughput will be $(1,1)$, as we expected.

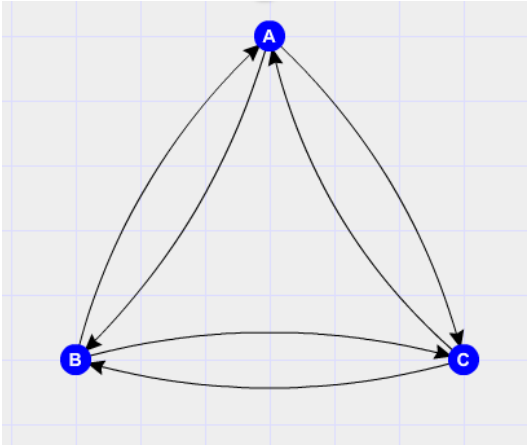
The transmission scheme above can be easily extended to an arbitrary two-user network by accumulating R different shared secret symbols in the beginning. We only need $c(A;B)>R$ or $c(B;A)>R$ to pass the secrets using single-way secure network coding. It is a reasonable assumption, since if more than 50% of capacity of a network can be fully compromised by an attacker, it is unreasonable to require perfect secrecy on it. No matter how long it takes to accumulate the necessary secret symbols, the total throughput can reach $c(A;B)+c(B;A)-R$ in the long term. As we have found, under appropriate transmission schemes, the eavesdropper gains nothing from the ability to change compromised channels from time to time.

IV. Methodology to Single-Group, Multi-Way Networks

In multi-way networks, the first problem arising is that we cannot directly construct a simple equivalent model for arbitrary network. Min-cut values from one point to another is not useful anymore, since there may exist paths go through a third user point. Multiple min-cut paths of multiple user node pairs may share some common channels. In this case, the network topology plays a role in the throughput region.

Without a universal equivalent model, the capacity limit in network coding is now different from case to case. Since secure network coding exploits precoding and network coding schemes, the throughput region for interactive secure network coding on multi-way networks seems not possible to find explicitly. However, we can alternately focus on another problem: given or assumed the capacity region, or the maximum possible throughputs between any node pairs and their total limits in simple network coding, how far away is that of interactive secure network coding? Or, how much is the cost for security in any given network? We begin the search with a simple network structure, the 3-way complete unit capacity network.

A. 3-Way Complete Unit-Capacity Network



A complete unit-capacity network is a multi-way network that has one and only one direct unit-capacity channel from A_i to A_j for $i, j \in 1,2,...k$. In the case $k = 3$, there are 6 channels, each starts and ends at one user node. Since such a network is symmetric, we are interested in (1) the maximum throughput r_{Max} between arbitrary A_i and A_j , and (2) the maximum total throughput r_{total} among all possible communication pairs.

We would still like to utilize key-recycling. Suppose $R = 1$. Say a key K is first generated by A and delivered to B . (If a key is not delivered at all, it is useless.) Will it be wise to utilize K at C ? If no secret key is used among all 3 user nodes, we will need at least one key between all three user node pairs. Each key travels through at least one channel and it follows that $r_{total} \le 6 - 3 = 3$ in this case.

Now we consider what happens if we use K among all 3 nodes. It is obvious that K would travel at least 2 times to arrive at all nodes, although at different time. We say the total cost to spread K is 2. Notice in this case K can be generated anywhere in the network and follows any path to traverse the network. By traveling any two channels to reach all three user nodes, K traverses a spanning tree of the network.

Once K is delivered, it is to be used to encrypt messages. It turns out that one such K each time slot is enough for the entire network. One example is like below.

At time slot t , node A generates K_t and pass K_t to node B . Simultaneously Node B pass $K_{(t-1)}$ to C , which it received one time slot before. Messages from B to A is encrypted with $K_{(t-1)}$, since $K_{(t-1)}$ is the newest common acknowledgements between them. Messages from A to C , C to A or C to B uses $K_{(t-2)}$, the newest secret key that C knows in this moment. In each time slot 2 of the 6 channels are used to deliver keys, and the rest 4 are used to transmit encrypted messages, resulting in $r_{Max} = 2$ and $r_{total} = 4$, which is an improvement. Since the min-cut between any two user nodes is 2, there is no room for improvement on r_{Max} , but we saved one channel's capacity in total by recycling and distributing the secret keys.

From the previous discussion we can easily see that r_{Max} cannot reach 5 in this case. Using different keys is not optimal, as we will see afterwards in about all networks; and using a single key will need at least 2 channels to deliver. The 2 channels needed is just the size of a spanning tree on a 3-way complete unit-capacity network; and the most economical way to flow something to the whole network is to go through a spanning tree, indeed.

B. Defining and the Usage of the Distribution Cost

For illustration of the distribution cost, for now we limit our discussion here to a particular type of multi-way networks. Suppose the user nodes to be A_1 to A_k , and that any two sub-networks between different pairs of user nodes are all disjoint. This equals to say, given any path p_{ij} from A_i to A_j that does not contain another user node, there exists no path from A_i to $A_{j'}$, $A_{j'} \neq A_j$, that does not contain another user node, through any intermediate nodes on p_{ij} . In this case all the subnetworks between two user nodes can be reduced to simple equivalent networks mentioned earlier in the two-user network section. It is now convenient to assume that there are at least one channel directly connect one arbitrary user node to another.

Suppose secrecy constraint R , we introduce the distribution cost $C_{dist} = |T| R$, where $|T|$ is the cardinality of a "user-

spanning tree" that not necessarily goes through all vertices on the network, but all the user nodes on it. It follows that $C_{dist} = (N-1)R$ from the nature of this particular type of networks. This cost function is the average throughput used to broadcast R secret keys per time slot to the whole network, and is the cost for key-distribution in the Newest Acknowledged Key Protocol.

In this protocol, a series (or consequence) of keys are generated anywhere in the network, and is broadcast through the network by a directed "user-spanning tree". Keys come in series of R , and will arrive at any node in at most $N-1$ time slots later. Any traffic between two user nodes is encrypted by the newest shared secret keys between the two nodes. For instance, if node A_i has received secret keys up to K_p , and node A_j has received secret keys up to K_q , they will use keys up to $K_{\min(p,q)}$ to communicate. It follows that

$$r_{total} = c_{total} - c_{dist} = c_{total} - (N-1)R .$$

Note that on a k-way complete unit-capacity network with $R=1$, $r_{total} = c_{total} - c_{dist} = 2C_2^k - (k-1) = (k-1)^2$.

C. An Newest Acknowledged Key Protocol

The protocol above can be expanded to any single-group multi-way network. When it is possible to use just enough amount of secret keys, it is believed to be optimal. Only in very bottlenecked networks that different sets of secret keys make sense, but there is not much we can really do in such networks. The protocol is layered into the following phases:

(i) Key-Distributing Phase. Given a connected network, one of the user nodes is picked to generate randomness. Once the source of the random keys is decided, the distribution problem turns into a broadcast problem. The network designer can then use any technique to establish a broadcast flow from the source node to all other user nodes; this portion of network is seen as removed from the original network and will be responsible for key-distribution throughout the transmission. This broadcast flow exploits network coding and, by the nature of network coding, can reach every user node given the min-cut capacity between it and the source to be larger than the security constraint.

(ii) Handshaking Phase. After a short period of time, every user node starts to receive secret keys. The distance from source varies among every node, so two nodes need to handshake before establishing secure network coding between them. One method is to mark the key number in its header, thus letting user nodes know which keys it have received. Then they can simply send out the newest key number it received to another node, along with the current time; by exchanging these values, they can setup the standard for their secure transmission. This phase needs no secrecy. Most of the time, the two nodes can decide to use the newest acknowledged keys they have received; it is the reason for the protocol name.

(iii) Secure Coding Phase. Already decided the standard of secure coding, two user nodes can deliver messages encrypted accordingly. Using the precoding scheme in [1], the secure work can be done before network coding comes into play. What user nodes do is simply coding received secret keys together with message packets. This can be done in linear time as in [2].

(iv) Network Coding Phase. After the previous phase all packets that are ready for network coding are of perfect secrecy. User nodes exploit network coding to transmit them all around the network. On almost all occasions linear time algorithms suffice.

V. Conclusions And Future Expansions

To deal with the problem of interactive secure network coding, we face the obstacle of the k-unicast problem in network coding. In multi-way networks, the fundamental limits are still unknown. Whether there is security concern, we need network coding to explicitly deliver messages we would like to send, and how good a secure coding scheme can be is most related to the nature of its network model. Once the fundamental limit on a network is known, like the two-way network, we can obtain the limit for secure transmissions, and implement protocols on it. However, almost all the fundamental limits on real networks are never known, but we can still find a good way to do secure coding on them.

By key-recycling, one can save as much resource as he can in any network, security assured, to clear more space for messages transmission; by key-distributing in a large network, the recycling is more systematic and efficient. The Newest Acknowledged Key protocol gives a convenient way to setup various 1-1 secure transmissions in large multi-way networks, while saving capacity and holding the perfect secrecy.

When the network size become very large, it is difficult by any designer to fully obtain the details of the network structure; centralized network coding is unlikely. However, even in localized and distributed coding schemes, the idea of sharing random keys still makes sense. The protocol itself needs only the source to be decided centrally to form a key-sharing network; in practice the process can be done heuristically and key-sharing networks can start small and merge into bigger ones. With key-sharing, security is most efficiently permitted in complex networks.

Acknowledgment

My instructor, Prof. I-Hsiang Wang, guided me into the field of (secure) network coding, and introduced me the idea of interaction. In the process of this work, he was kind in discussion and pointed out the blind spots wisely. His instructions in (network) information theory helps a lot in viewing the problem conceptually.

References

[1] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap Network", *IEEE Transactions on Information Theory*, vol. 57, No. 1, Jan. 2011.

[2] S. R. Li and R. W. Yeung, "Linear Network Coding", *IEEE Transactions on Information Theory*, vol. 49, No. 2, Feb. 2003