Secure Network Coding with Min-Cut Protecting

Shao-Heng Ko

Department of Electrical Engineering

National Taiwan University

b00901169@ntu.edu.tw

Abstract— **This report summarizes some research on various ways to improve the network capacity on 1-1 secure network coding network. With the idea that one can opt to protect some of the most important part in a wiretap network, the capacity of such a network can be enhanced compared to using basic secure network coding.**

I. Introduction

A network is represented by a directed graph $G=(V,E)$ with nodes correspond to communication units and edges correspond to channels. In this report, like some previous paper in this topic, all channels are assumed to be noiseless and have unit capacity. Messages are generated at a source node S, and to be decoded at a sink node D.

A wiretap network introduced in [1] is a network whose channels are vulnerable to mischievous eavesdroppers. In such a network, an eavesdropper may attack any possible edge sets; the sender need to randomize the message in order to protect it to assure that the wiretapper gains no information after wiretapping.

The model introduced in [1] has 2 assumptions: (1) all channels are vulnerable to be eavesdropped, (2) all the random secret keys are generated at source. The major part of this report, however, is about the probable improvement on capacity upper bound when these assumptions are violated.

This research is motivated by [1], in which that under assumption (1) above, there are no real safe channels in the whole network. If we are interested in strengthen the network, a simple idea is to protect some of the channels from attack. Given limited resource, we may first address the weakest part in the network; that is, the Min-Cut of it.

Once the whole Min-Cut, or at least a part of it, is protected from the wiretapper, we may use it conveniently to transform pure messages without randomness in it. However, in order to pass pure messages through the Min-Cut channels, the message itself needs to be fully decoded before it.

When only the source is able to generate randomness, it still requires randomness in the pass-the-Min-Cut part of the network. Thus we still have to send secret keys across the Min-Cut with regard to safety considerations. This is the motivation for that, in some cases discussed in this report, some of the intermediate nodes in the network have the ability to generate randomness.

Under these two additional condition above, I constructed a model, in which a particular Min-Cut channel set is specified, and there exists no backward flow across the Min-Cut. With the theoretical basis in [1] and [2], I propose an upper bound of the network capacity in this particular wiretap network.

Also, a conceptual algorithm is proposed, aimed to achieve the upper bound above. The bound and the algorithm both suppose that the sender first focus on maximizing the use of the protected part of the Min-Cut, and then the other part. The algorithm is somewhat a greedy algorithm which works on the nodes adjacent to Min-Cut channels. With proper key-recycling and a method to use secret keys, this algorithm achieves the theoretical upper bound at least in some relatively simple networks, while whether the algorithm works in any arbitrary network remains to be further investigated.

II. Secure Network Coding with Protected Channels

The model of this special wiretap network will be presented, and four of special network that I have addressed will follow. In each of the first three cases, I give the optimal network capacity. The fourth and the most complex case will be discussed in the next session.

A. Model

A wiretap network with protected channels consists of the following components:

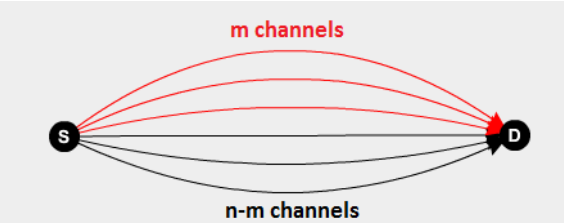
1) *Directed multigraph* $G=(V,E)$, where V and E are respectively the node set and the edge set of G.
2) *Source node s*: *s* is contained in V, where the random message M is generated.
3) *Sink node d*: *d* is contained in V, where the receiver decodes M.
4) *Collection of sets of protected channels* $P\subseteq E$: a collection of those channels that an eavesdropper cannot access to.
5) *Specified Min-Cut* $C\subseteq E$: a Min-Cut between *s* and *d* with no channels from the part conatins *d* to that contains *s*.

This model can be transformed into the model in [1], with the user set being only the sink here, and the collection of sets of wiretap edges simply collects some subsets of $E-P$.

We denote *r* to be the secrecy constraint, which is the largest cardinality of any subsets in C. When the secrecy constraint is set to be *r*, a wiretapper gathers information from at most *r* channels.

In the following text, every model is of secrecy constraint *r*, and C consists of merely every subset of size *r* that contains in $E-P$. This equals to that the wiretapper simply chooses *r* channels to access.

B. Network Case I



In this case, V contains only *s* and *d*, with there are only *n* channels going from *s* to *d*, among which *m* channels are protected. Clearly the Min-Cut value is *n*.

This is the basic network for all my research. First, it is optimal to simply cast *m* messages through the protected channels. From the concept introduced in [2], suppose that $n-m>r$, we can cast more messages through the rest of channels by encoding the messages with *r* secret keys.

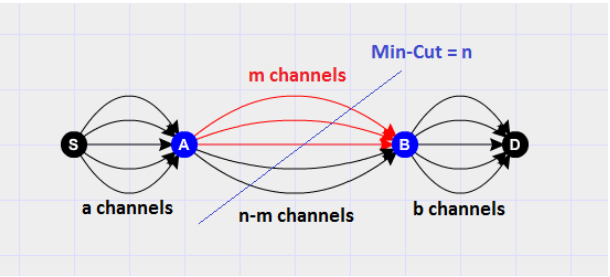
Thus, in this case the optimal channel capacity is

$m+\max\{\mathrm{n}-\mathrm{m}-\mathrm{r},0\}$, or denoted as

$m+(\mathrm{n}-\mathrm{m}-\mathrm{r})^+$

where the + represents taking positive value or 0.

C. Network Case II



In this case, V contains *s*, *d* and two intermediate node *A*, *B*. There are *a* channels from *s* to *A*, *b* channels from *B* to *d*, *n* channels going from *A* to *B*, among which *m* channels are protected. It is assumed that

$\min\{a,n,b\}=n$,

which means the Min-Cut value is also *n*.

Let us consider the case while $\min\{a-r,m,b-r\}=m$. Assume that $n-m>r$. In such case, we can cast m messages through every part of the network. After this, there are $a-r-m$, $n-r-m$, $b-r-m$ channels left respectively at the three stages of network. Since we encoded the messages with *r* keys in the previous step, we can reuse them to encode the other messages as well. Hence we fully use these extra channels to cast more messages.

Notice that $\min\{a,\mathrm{n},\mathrm{b}\}=\mathrm{n}$ by definition. Thus it is obvious that

$$\min\{a-r-m,n-r-m,b-r-m\}$$
$$=\min\{a,n,b\}-r-m$$
$$=n-r-m$$

is the amount of messages we can cast through the unprotected part of network.

If $n-m\le r$, since the *m* protected channels are secure, we can still cast *m* messages to *B*. But the rest part of the network has no more space for any message casting since there are less than *r* channels left in the Min-Cut. Here, if we give *B* the ability to generate randomness as well, *B* doesn't need to receive *r* keys from *s*. Hence the whole capacity is *m*. Consider the other case that *B* cannot generate randomness itself, it will need additional $r-(n-m)$ keys from *s*. These keys will therefore occupy some of the protected channels, lowering the network capacity from *m* to

$m-(r-(n-m))=m+(n-r-m)$

Now we can say that given $\min\{a-r,m,b-r\}=m$, we have capacity of $m+(\mathrm{n}-\mathrm{m}-\mathrm{r})^+$ if B can generate keys, and $m+(\mathrm{n}-\mathrm{m}-\mathrm{r})$ if otherwise.

Finally, it is clear that if $\min\{a-r,m,b-r\}\neq m$, this means $a-r<m$ or $b-r<m$. the network capacity will be merely $\min\{a-r,m,b-r\}$, and since $n<a$, b , $n-r-m<0$.

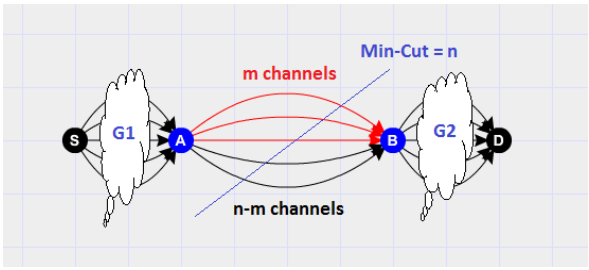
Summarizing all we have, the network capacity is

$\min\{a-r,m,b-r\}+(n-m-r)^+$

if B can generate keys, and

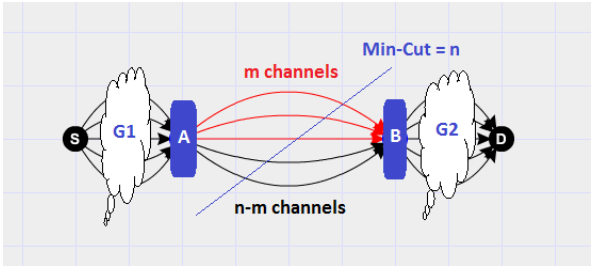
$\min\{a-r,m,b-r\}+(n-m-r)$ if otherwise.

D. Network Case III



This case features two arbitrary sub-network in place of the *a*, *b* channels in the previous case. Given that the Min-Cut valut is *n* here, which means the two sub-network have larger Min-Cut values, it can be shown that, by the theorems introduced in [1] and [2], and the discussion above, the result in this case is the same as above only to substitute *a* with $cut(s;A)$, and *b* with $cut(\mathrm{B};\mathrm{d})$.

E. Network Case IV



The major difference between this last case and case III is that now *A*, *B* are two *node sets* instead of nodes. The selected

Min-Cut has value *n*, and in the whole network, there exists no channels from the right side of the selected Min-Cut to the left side of it. Note that in the previous cases, the Min-Cut is unique and hence there is no need to specify it; but here, there may exist multiple Min-Cut, in which we choose an arbitrary one to strengthen.

This is a far more generous model, and we further investigate this case in the following section. We shall see, although it may seem that Case IV is not that far from Case III, it is in fact far, far more difficult and complex. This case can be called "The Min-Cut protecting wiretap network."

III. Structure of the Min-Cut Protecting Wiretap Network

So far, what we utilize as our strategy is that we try to maximize the use of the protected part of the network, and then see if there remains any possibility to send more messages. We will continue to utilize this strategy in our research of the model. However, it should be noticed that, under the previous three circumstances, this strategy is simply optimal, while in this particular case, it is not known that if this strategy is optimal itself. All the following discussing is limited to this strategy.

A. An Upper Bound of the Capacity of the Protected Part

As briefly mentioned in the introduction, the main problem is that the messages need to be fully decoded right before the Min-Cut. In the three easier cases, there is only one node right before the Min-Cut, so anything arrives there can be decoded. It is not the case now, though, since *A* is now a node set. We begin with some terminology.

1) $A^m \subseteq A$ is defined to be the subset of *A* that consists of all nodes adjacent to the *m* protected channels.
2) $B^m \subseteq B$ is defined likewise.
3) $\alpha(S, A^m)$ denotes the maximum amount of messages that can be cast from *S*, fully decoded and sent across the protected channels at A^m.
4) $\mathrm{P} = \{A_1, A_2, \ldots A_n\}$ is a partition of A^m that satisfies
   $A_i \subseteq A^m \forall 1 \le i \le n,$
   $A_i \cap A_j = \varnothing \forall 1 \le i, j \le n, \mathrm{i} \ne \mathrm{j},$
   $\bigcup_{i=1}^{n} A_i = A^m$
5) $\delta(A_i)$ denotes the number of protected channels in the Min-Cut adjacent with a node in A_i.

The network still consists of three stages. The Min-Cut part is trivial. Now let us consider the third stage, the connections from B^m to *d*. We first prove that given any $cut(B^m, d)$ messages arriving at B^m, they can all be cast to *d* and be fully decoded there given that nodes in B^m has the ability to generate randomness.

Here we cannot directly use the theorem in [2] since the third stage is a multisource network with all the B^m being the sources. However, suppose that there exists some cases that the above argument doesn't hold. Then there exists a subset $B' \subseteq B^m$, with $\delta(B')$ messages coming from the Min-Cut, such that it is impossible to cast all them to destination. Now we reverse all the third stage, making it a single-source network with *d* being the source. Here the theorem in [2] can be used, and we obtain that $cut(B', \mathrm{d}) < \delta(B')$

Now, the Min-Cut has only $n - \delta(B')$ remaining channels ending at $B - B'$. Combining these channels and $cut(B', \mathrm{d})$ makes a cut of our network. But this new cut has size less than *n*, which is the size of Min-Cut. Thus a contradiction occurs, and it implies that such condition is impossible given the second-stage is a Min-Cut.

Until now, we can say that in maximizing the use of the protected part, the capacity of this phase is

$$\min\{\alpha(S, A^m), n, cut(B^m, d)\}.$$

Now we solve $\alpha(S, A^m)$. We claim that for every subset $A' \subseteq A^m$, $\alpha(S, A') = \min\{(cut(S, A') - r)^+, \delta(A')\}$.

The above equality can be verified by checking how many messages can arrive with security, and how many can leave.

Given any partition $\mathrm{P} = \{A_1, A_2, \ldots A_n\}$, we have

$$\alpha(S, A^m) \le \sum_{i=1}^{n} \min\{(cut(S, A_i) - r)^+, \delta(A_i)\}.$$

So we have

$$\alpha(S, A^m) \le \min_{P}\{\sum_{i=1}^{n} \min\{(cut(S, A_i) - r)^+, \delta(A_i)\}\}.$$

This serves as an upper bound for $\alpha(S, A^m)$, and combining it with *n* and $cut(B^m, d)$, we have an upper bound for the capacity of the first phase of the network. In some relatively simple network, the equality is achieved. Now I propose an algorithm to try to solve the network and meet this upper bound.

B. Greedy Algorithm for the Network

First I propose an idea for constructing the residue network on any given existing flow in the network. This technique is determined to enhance key-recycling.

In the concepts in [1], upon pre-coding, any flow that carries *n* messages can be substituted with mixed copies of *n-r* messages and *r* secret keys. That is, when we are looking at the whole network, or a large part of it, we are interested in only the capacity, but not whether a particular channel carries a key or an encoded message.

Given an arbitrary flow, we have a clear idea of what a node "knows" upon the flow, be it a key, a message or some mix of them. Disconnect all the already busy channels from the network. For every packet of information that source has sent in the flow, we create a pseudo-node that corresponds to that particular information, create pseudo-channels connecting source to it, and connecting it to all the nodes that occupy it. The resulting modified residue network is now equivalent to the original network with the best key-storing and key-recycling. Any flow on the residue network can be transformed into one achievable flow by substituting the existing variables into it.

The algorithm works in an arbitrary order on the nodes in A^m. When a node *K* is picked, with the help of existing Secure Network Coding scheme in [1], we cast $\alpha(S,K)$ messages to *K* (with regard to the current residue network, instead of the original one). Then, substitute the existing variables into it.

Keep a counter for the real keys that have been used. If there already exist *r* different secret keys in the network, from the (*r*+1)-th key, it is substituted with summing all the *r* keys and a dummy variable. The ($r+t$) –th key will be

$$\sum_{i=1}^{r} k_i + x_t$$

In the end of the algorithm, it will check if any of the dummies can in fact be fully decoded at destination. If so, it will be substituted with one packet of message.

So far, this algorithm works well on relatively small and simple network. It has not been found any circumstances when the optimizing order does impact the result, and all the results of this algorithm until now have reached the upper bound discussed above.

After the phase when we optimize the use of the protected channels, we simply also construct the residue network, and this time, since there are no secure channels left, it suffices to use Secure Network Coding scheme to determine the rest of the capacity.

IV. Conclusions

In this report, four cases of Min-Cut protected Wiretap Network are discussed. Three easy cases are solved and their network capacity under a particular secrecy constraint is calculated. In the fourth case, an upper bound of capacity of a part of the network is proposed, as also an algorithm trying to meet the upper bound.

These discussions realized the motivation of protecting the Min-Cut, the weakest part in a network, to show that it is possible for real enhancement in all cases. Strengthening the Min-Cut has its effect on the capacity.

Acknowledgment

Thanks to Prof. I-Hsiang Wang for instruction and motivation in the special project course this semester. I appreciate the genuine effort you make in bringing Network Coding and all the interesting topics into our minds.

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.