Field Hopping in Network Coding

Haruko Kawahigashi, Yoshiaki Terashima

Information Technology R&D Center, Mitsubishi Electric Corp.
5-1-1, Ofuna, Kamakura, 247-8501, Japan,
Kawahigashi.Haruko@ah.MitsubishiElectric.co.jp

ABSTRACT

We propose Field Hopping, a technique for making intercept less likely by altering the finite field that is a basis of the network coding operation. The field is altered in a predetermined manner known exclusively among the sender, the receiver, and the intermediate nodes. We study effectiveness of the proposed method on the security of linear network coding through theoretical considerations.

Keywords: Network coding, finite field, security, wireless network

1 INTRODUCTION

The technique of network coding has been successful in many different areas of digital communication in recent years [1]-[8]. The two main advantages of the network coding are saving of the bandwidth through efficient use of the available communication channel and the enhanced data security against stealing of the data at intermediate nodes and channels. In the network coding, each data is transmitted through a channel after linear encoding, so what we see on a channel is an encoded data rather than a raw original data. It is clear that this encoding increases security from a view point of data protection. We propose a new method here to achieve an extra step in this type of data protection.

There have been some studies of the security aspects of linear network coding, although not many [9]–[15]. Our method proposed here takes a different approach from the above studies. Ours does not make wiretapping completely impossible or extremely difficult, but it forces an eavesdropper to make extra efforts for obtaining useful information. Our analysis uses our previous work [9] on a measure of robustness against wiretapping. Our method is independent from the ones in the above papers, and it is possible to apply our proposal simultaneously together with one of the above papers to enhance security.

In the rest of this paper, we give a detailed description of this method of switching the field structures and analyze how much extra safety we gain in this method. At first in Section 2, we present a formulation of the linear network coding and the finite field. Then in Section 3, we describe the proposing method, and evaluate its effectiveness in Section 4. In Section 5, we estimate its effectiveness in different environments. .

2 NETWORK MODELS AND LINEAR NETWORK CODING

We consider linear network coding on finite acyclic graphs consisting of vertices and edges as in [2]—[9]. Our model is characterized as follows.

(1) Multiple edges between a pair of vertices are allowed. We have the distinguished vertex *S* called the source vertex.

(2) Transmission data consist of string of elements in *F*.

(3) The number of independent paths from *S* to each non-source vertex *T* is called the maximum flow from *S* to *T*, and is denoted by maxflow(*T*). Let *d* be the minimum of maxflow(*T*) for all non-source vertices *T* among *V*.

(4) We send out an arbitrary *d*-dimensional column vector *v* over the base field *F* at the source *S* and it is called the information vector.

(5) Each edge is a channel and transmits an element of the base field *F* as transmitted data.

(6) At the source vertex *S*, we have an $s \times d$ matrix M_S, where *s* is the numbers of the outgoing edges from *S*. We have a matrix multiplication $M_S v$ and send out this *s*-dimensional column vector through the edges starting from *S*.

(7) At each vertex *T*, we have an $s \times l$ matrix M_T, where *s* and *l* are the numbers of the outgoing and incoming edges. Each incoming edge carries an element of *F* and they give an *l*-dimensional column vector *v*. We have a matrix multiplication $M_T v$ and send out this *s*-dimensional column vector through the *s* out going edges without any time delay.

(8) At each of a fixed subset of the vertices having maxflow equal to *d*, we want to recover the original information vector. Such a vertex is called a target vertex.

We call the $s \times l$ matrix M_T at *T* the encoding matrix at the vertex *T*. For any *d*-dimensional information vector *v* at the source *S*, the data *z* transmitted through an edge *E*, represented by an element in the base field *F*, depends linearly on *v*, so we have a *d*-dimensional row vector w_E such that $z = w_E v$. We call this vector w_E the encoding vector for the edge *E*.

3 SWITCHING IRREDUCIBLE POLYNOMIALS

Unlike normal linear operation of matrix or vector, all arithmetic operations of linear network coding are performed

in a certain finite field F (Finite set of elements where one can perform addition, subtraction, multiplication and division). It is possible to use any finite field, but we use a field of the size 2^m (m =8) and denote $F(2^m) = F(2^8)$ for simplicity. Then an element in the field is represented with one byte data.

Each element $a = (a_{m-1}, a_{m-2}, \cdots, a_1, a_0)$ of the finite field $F(2^m)$ corresponds to a polynomial $a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1 x + a_0$ of degree *m-1*, where each coefficient takes the value either 0 or 1.

Additions and subtractions of two elements of the finite field $F(2^m)$ are both bitwise operations of "exclusive-or" or modulo 2 addition, and it is easy to perform. Multiplication consists of ordinary multiplication of two polynomials modulo an irreducible polynomial $f(x)$ of degree m. It is known that there are 30 irreducible polynomials of degree 8 (m=8). One such example is $x^8 + x^4 + x^3 + x^2 + 1$. So we have 30 different finite field structures. The number of irreducible polynomials is 56 when m=9, and 99 when m=10.

We now propose changes of irreducible polynomials. We propose to use possibility of this choice as an extra enhancement of data security. That is, we propose to change the field structures depending on the source, destination and the time. With this change, even the same inputs give different outputs, depending on the source, destination and the time (see Fig. 1). This clearly brings extra complicacy for the enemy trying to stealing the data through wiretapping.

Field Hopping is a technique for achieving resistance to eavesdropping, intercept and interference, by altering the finite field that is a basis of the network coding operation.

Note that one flow of data from the source to the targets must use the same finite field, and all the nodes must know which finite field they use at present. The following are methods of announcing the current finite field to the nodes.

(1) Packet attached announcement: The field currently in use can be written in the packet.

(1-a) Explicit announcement: Explicitly write the polynomial in use on the packet e.g. in the header.

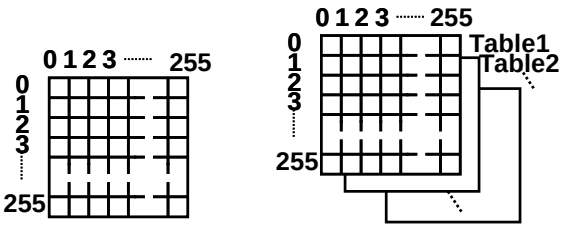
(1-b) Implicit announcement: Alter the polynomial in a predetermined manner shared among the participants, according to the implicitly shared information written in the packet e.g. in the header.

(2) No-announcement: The field is altered in a predetermined manner known exclusively among the participants, i.e., the sender, the receiver, and the intermediate nodes. The resistance to eavesdropping increases since the hopping sequence is shared only among the participants. An example of this method is to switch the polynomial according to time.

4 EFFECTIVENESS OF THE PROPOSED METHOD

We now estimate effectiveness of the above proposed method from a viewpoint of data protection.

We deal with data flows at one node. Suppose that a wiretapper can watch incoming and outgoing data and he wants to find out the encoding matrix. This is the type of security threat we consider here. Suppose that the number of the incoming edges is l and that of the outgoing ones is s. Then we multiply the $s \times l$ encoding matrix M_T and the l-dimensional incoming vector to get an s-dimensional column vector y of outgoing data. First we assume that the wiretapper knows the finite field and consider how many times of wiretapping are necessary in order to find out the encoding matrix M_T. The wiretapper obtains l times of l-dimensional vectors $x_1, x_2, \ldots, x_l$. Multiplication by M_T to these vectors gives l times s-dimensional column vectors $y_1, y_2, \ldots, y_l$. By putting l times of l-dimensional vectors $x_1, x_2, \ldots, x_l$, we obtain an $l \times l$ matrix X. We similarly obtain a $s \times l$ matrix Y by putting l times s-dimensional column vectors $y_1, y_2, \ldots, y_l$. Then we have $M_T X = Y$. If the determinant of X is not zero, one can easily find M_T as YX^{-1}. If X does not have an inverse matrix, and one obviously cannot find M_T.



(a) Single field (Conventional) (b) Multiple filed (Proposed)

Figure 1: Finite field operation multiplication tables

In this case, the wiretapper has to continue to steal more data. So we would like to count how many times one has to wiretaps vectors. We have already considered this mathematical problem in a different context in [9], and the answer is

$$f1 = \sum_{n=1}^{l} \frac{(2^t)^n}{(2^t)^n - 1} = \sum_{n=1}^{l} \frac{256^n}{256^n - 1}, \qquad (1)$$

since the size of the finite field is 256. We called this quantity the *wiretap robustness* (WTR) in [9].

Next we assume that the wiretapper does not know the finite field structure and needs to find it out.

One chooses one of the 30 irreducible polynomials and assumes it gives the finite field structure. One continues as above until determining M_T. Then find another input vector x and compare it with $M_T y$ for the output vector y.

If the above comparison fails, the irreducible polynomial is not the right one. Repeat this process until 29 out of 30 polynomials are excluded. Then the remaining one gives the right answer and we know the encoding matrix.

We now estimate the number of extra times of wiretapping in the above procedure compared with the case we know the finite field from the beginning.

If we wiretap an extra vector, it gives one byte data after multiplications and additions. It gives a correct answer even if we have a wrong finite field structure just by coincidence.

Let p be 1/256, the estimated probability of this coincidence, since one byte gives 256 possible answers. Then $(1-p)^{29}$ is the probability that all the other 29 wrong finite field structures are excluded by this vector. So with probability $1-(1-p)^{29}$, we need one more vector. Similarly, with probability $1-(1-p^2)^{29}$, we need another vector. The total number of vectors we need in this process is as follows.

$$f2 = 1 + (1-(1-p)^{29}) + (1-(1-p^2)^{29}) + \cdots \quad (2)$$

Since p=1/256 is small, the third and later terms are negligible, and we obtain 1.11 approximately. In Figure 2, we have a graph showing this quantity $f2$ for various values of t for the size 2^t of the finite field. In the above setting, l is typically around 2, 3, or 4, so in these cases, this extra number 1.11 gives an increase of 56%, 37%, and 28% respectively. We have Figure 3 showing the increase $f2$ for various values of l and t, where t is as above for the size 2^t of the finite field. In network coding, each node transfers coded data, so having the encoding matrix at one node is usually insufficient for obtaining the entire date flow over the network. So a wiretapper needs to continue the same type of wiretapping at other nodes, and this extra increase of the efforts applies to other nodes, too.

5 OVERHEAD COMARISON TO A GENERAL SECURITY SCHEME

In this section, we compare the overhead of the proposed method to IPsec (Security Architecture for the Internet Protocol), one of general security schemes [16].

In section 3, we described the method of announcing the polynomials to the nodes. The overhead of the announcement is as follows.

(1) Packet attached announcement:

(1-a) Explicit announcement: The overhead is same as the number of the irreducible polynomials, i.e. 1 byte.

(1-b) Implicit announcement: The overhead is zero, since the information already written in the packet header, e.g. source node number, is used.

(2) No-announcement: It is necessary to avoid mixing the packets before and after the change of polynomial in continuous packet stream. We can set intervals to separate the packets before and after the change. We can utilize the existing intervals by setting rules such as prohibition of change during a session. In that case, overhead is not necessary. We can also set a flag on the transmission packets. Then we need 1 flag packet in that case.

We give an overhead comparison of the explicit announcement method and IPsec using AES (Advanced Encryption Standard). The encryption block size of AES is 128 bit (16 byte), and padding information is added to the smaller size information or indivisible transmission information. In IPsec, we assume the addition of tunnel mode header of ESP (Encapsulated Security Protocol) 20 byte, ESP header 8 byte, ESP IV (Initialization Vector) 8 byte, ESP trailer 16 byte, total 52 byte [5]. We focus on security here, and TCP/IP header is not included.

Figure 5 shows that the overhead of the proposed method in explicit announcement is about 1 / 100 of IPsec case.

6 ESTIMATES OF EFFECTIVENESS IN DIFFERENT ENVIRONMENTS

We now estimate effectiveness of our proposed method in different settings of network coding.

(1) Random network coding

Recently technique of random network coding has been studied by many researchers [7]. In this scheme, an encoding matrix at each node is not fixed and is chosen randomly at each time, and the encoding vector is sent within each packet.

We now consider switching finite field structures within this framework. Each node obviously must know which finite field is used, but the wiretapper does not know it. In order for a wiretapper to decode a specific data, one has to collect d times of linearly independent d-dimensional encoding vector. Finding the estimate of the number of data we need to collect in order to obtain linear independence is the same mathematical problem as in the last section, so the number is given by $f1$ where the variable l is now replaced with d, and if one does not know the finite field structure, the estimate of the number of extra vectors we need to find out the finite field is again 1.11 which is given by $f2$ with p=1/256 as above. So the mathematical structures and estimates are the same, and Figures 2-3 apply.

(2) Multi-layer network coding

Suppose that the number l, the number of incoming data, is 3. Then the extra increase we have with our proposed method is 37%, as shown above. Now we consider this effect in the setting of multi-layer network coding. We can apply the network coding scheme also to these data transmissions through intermediate nodes. In this way, we have multiple layers of network coding systems. In the case of double layers, we can apply our proposed method to both layers. Then the estimate of the 37% increase applies to both layers, and a wiretapper has to attack two layers separately. This means that the total increase of the extra efforts for a wiretapper is 88% since 1.37^2=1.88. Figure 4 shows the effect of this multilayer for different values of l.

(3) Multi-party use of the same network

Two or more parties can use the same network with network coding, using different finite fields. Then one party knows only their own data and finite field structure, so they are at the same position as a wiretapper as far as the data of the other party are concerned, and the above analysis applies.

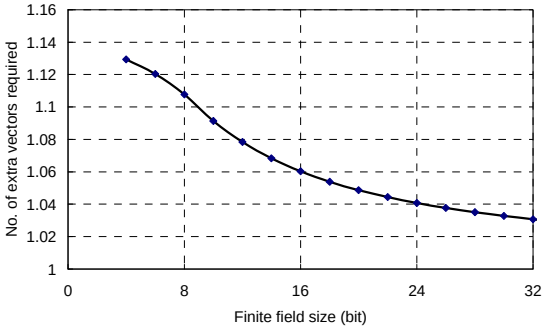


Figure 2. Number of extra vectors required: $f2$ (Eq. (2))

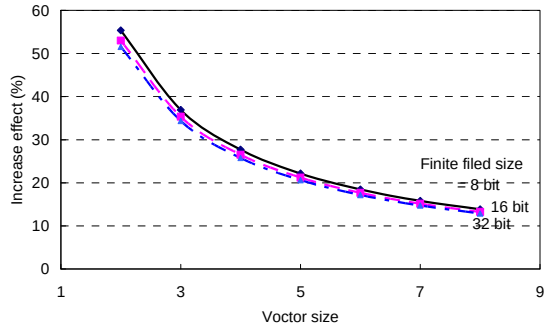


Figure 3. Increase effect in relation to vector size *f*2 / *f*1

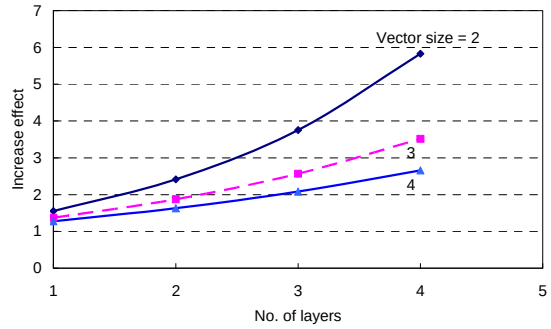


Figure 4. Increase effect in layered coding
(Finite field size: 8bit)

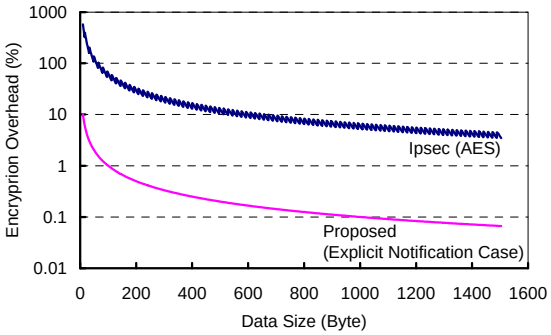


Figure 5. Comparison of the overhead

7 CONCLUSION

In this paper, we have proposed network coding schemes with increased and flexible security by changing finite field structures. Our proposed schemes switch irreducible polynomials used in multiplication of coding operation. This multi-party flexible security mechanism is particularly useful in cases such as Disaster Relief where multi national rescue teams are dispatched to the same field. They can pursue their own individual secret communication by using different finite field structure, and can also start a mutual communication by using the same finite field structure while keeping their data security against the third party.

REFERENCES

[1] R. Ahlswede, N. Cai, S-Y. R. Li, R. and W. Yeung, Network Information Flow, IEEE Trans. on Information Theory, vol. 46, no.4, , pp. 1204 –1216 (July 2000).

[2] S-Y. R. Li, R. W. Yeung, and N. Cai, Linear Network Coding, IEEE Trans. on Information Theory, vol. 49, no.2, pp. 371 –381 (Feb. 2003).

[3] R. Koetter, and M. Medard, An Algebraic Approach to Network Coding, IEEE/ ACM Trans. on Networking, vol. 11, no. 5, pp. 782 –795 (Oct. 2003).

[4] T. Ho, M. Medard, and R. Koetter, An Information-Theoretic View of Network Management, IEEE Trans. on Information Theory, vol. 51, no. 4, pp. 1295 – 1312 (April 2005).

[5] P. Chou, Y. Wu, and K. Jain, Practical Network Coding, Allerton Conference on Communication, Control, and Comput-ing, Monticello, IL (October 2003).

[6] T. Ho, and D. S. Lun, Network Coding, Cambridge University Press (2008).

[7] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, and B. Leong, A random Linear Network Coding Approach to Multi-cast, IEEE Trans. on Information Theory, vol.52, no. 10, pp. 4413—4430 (Oct. 2006).

[8] J.-S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Medard, Code-Cast: A Network-Coding-Based Ad Hoc Multicast Protocol, IEEE Wireless Communication. vol. 13, no. 5, pp.76—81 (Oct. 2006).

[9] H. Kawahigashi, and Y. Terashima, Security Aspects of the Li-near Network Coding, MILCOM 2007, NCS1-6 (Oct. 2007).

[10] N. Cai and R.W. Yeung, Secure network coding, in IEEE International Symposium on Information Theory (July 2002).

[11] K. Jain, Security based on network topology against the wiretapping attack, IEEE Wireless Communications, pp.68—71 (Feb. 2004).

[12] K. Bhattad and K. Narayanan, Weakly secure network coding, in Proc. of the First Workshop on Network Coding, Theory, and Applications (NetCod) (2005).

[13] J. Tan and M. Medard, Secure network coding with a cost criterion, in Proc. of the Second Workshop on Network Coding, Theory, and Applications (NetCod) (2006).

[14] L. Lima, M. Medard, and J. Barros, Random network cod-ing: A free cypher?, in IEEE International Symposium on In-formation Theory (ISIT) (2007).

[15] K. Han, T. Ho, R. Koetter, M. Medard and F. Zhaom On network coding for security, MILCOM 2007 (Oct. 2007).

[16] S. Kent et. Al., “Security Architecture for the Internet Protocol,” RFC 4301, IETF, Dec. 2005.

[17] R. Savarda, et. al.,”Explaining the Gap between Specification and Actual Performance for IPsec VPN Systems,” TISC Insight, vol.3, Issue 9, http://www.tisc-insight.com/newsletters/39.html