通信情報保護學會論文誌
第 7 卷 第 3 號, 1997. 9

안전한 통신을 위한 비밀 경로 알고리즘의 분석

배 용 근*, 정 일 용*

An Analysis of the Secret Routing Algorithm for Secure Communications[1)]

Yongkeun Bae, Ilyong Chung

요 약

경로 보안은 데이터의 전송을 위해 선택된 경로의 비밀성에 관한 것이다. 만일 경로의 일부분이라도 알려진다면 이 경로를 통해 전달된 데이터가 유출될 확률은 높아지므로 데이터의 전송 경로는 보호되어야 한다. 이를 위해 우리는 중간 노드를 비밀리 선택하여 기존의 최단 거리를 이용하여 데이터를 전송하는 방법 대신에 이 중간 노드를 이용하여 데이터를 목적 노드에 보낸다.

더 나아가 여러 개의 비밀 경로를 이용한다면 한 개의 경로에 모든 데이터를 보내는 대신에 각 경로에 partial 데이터를 보낼 수 있기 때문에 데이터의 보안은 좀 더 강해진다. 본 논문에서는 MRNS 네트워크 상에서 특수한 메트릭스를 응용하여 시간 복잡도가 $O(l)$인 비밀 다중 경로 알고리즘을 설계하고 불확실성의 관점에서 이 알고리즘의 안전도를 분석한다.

Abstract

Routing security is related to the confidentiality of the route taken by the data transmitted over the network. If the route is detected by the adversary, the probability is higher that the data are lost or the data can be intercepted by the adversary. Therefore, the route must be protected. To accomplish this, we select an intermediate node secretly and transmit the data using this intermediate node, instead of sending the data to the destination node using the shortest path.

Furthermore, if we use a number of secret routes from the starting node to the destination node,

---

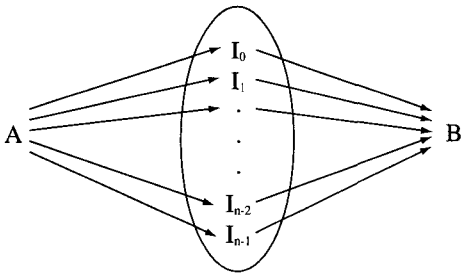
* 조선대학교 전자계산학과 조교수

1) This Study was supported by Factory Automation Center for Parts of vehicles(FACPOV) in Chosun University, Kwangju, Korea. FACPOV is designated as a regional research center of Korea Science and Engineering Foundation(KOSEF) and operated by Chosun University.

data security is much stronger since we can transmit partial data rather than the entire data along a secret route. In this paper, the routing algorithm for multiple secret paths on MRNS(Mixed Radix Number System) Network, which requires $O(l)$ for the time complexity where is the number of links on a node, is presented employing the HCLS(Hamiltonian Circuit Latin Square) and is analyzed in terms of entropy.

1. Introduction

When we transmit the data in a distributed network, we must consider Routing security.[1] Routing security is the security of the route taken by the data transmitted over the network. If the route is detected by the adversary, the probability is high that the data are lost or the data can be intercepted by the adversary. Therefore, the route must be protected. To accomplish this, we select an intermediate node secretly and transmit the data using this intermediate node, instead of sending the data to the destination node using the shortest direct path. The above route consisting of two paths - the first path from the source node to the intermediate node and the second path from that intermediate node to the destination node, is called a secret route. Furthermore, if we use a number of secret routes from the starting node to the destination node, data security is much stronger since we can transmit partial data rather than the entire data along a secret route. To employ the above idea, the data is dispersed into n pieces by the IDAF(Information Dispersal Algorithm using the FFT algorithm).[2] Then, each packet is transmitted simultaneously to the destination along its own secret route in the n-dimensional MRNS(Mixed Radix Number System) network. For this routing, we must find these n secret routes from the starting node to the destination node. Also, all the packets should arrive at the destination in the minimum possible time. Therefore, each secret route should be disjoint from all other secret routes.



A = the starting node B = the destination node
I_j = the i^{th} intermediate node ; $I_i \neq I_j$, if $i \neq j$

Figure 1. A set of secret routes

Finding a set of disjoint paths in a general network is a computationally difficult problem.[3] However, researchers have proved that there exist sets of disjoint paths in specific kinds of networks. From these proofs, they have designed combinatorial algorithms for finding a set of disjoint paths. Unfortunately, these combinatorial algorithms require much time for obtaining these paths. Some approaches have been tried to reduce the time complexity for these algorithms. Rabin[4] has applied an error-correcting code method to the parallel routing algorithms for the hypercube network. Rabin's algorithm employs an (n x n) Hadamard matrix, every two different rows of which differ in exactly n/2 positions. This algorithm is central to the current studies in Routing security and we continue his work below.

We have two methods for designing n secret paths from the starting node to the intermediate nodes. One of them is to select the intermediate nodes secretly, and then to find the disjoint paths from the starting node to the intermediate nodes. The other is to make the disjoint paths secretly and then to randomly select the nodes on these paths as the intermediate nodes. Since these paths are made secretly and the nodes on these paths are selected randomly, each node in this network is chosen secretly. Rabin's algorithm follows the first method. Valiant's algorithm[5], which involves finding one secret path, may be adopted to yield an example of the second method. Our method, as yet unexplained, is designed based on the second method.

We propose an alternative algebraic method for secret routing on the MRNS(Mixed Radix Number System) network.[6] For the first part of the secret routes from the source node to intermediate nodes, our method is to transform a set of vertex-disjoint paths into disjoint sets and then to investigate these disjoint sets. Later, these sets with some constraints are used to construct a special class of matrices, where each set is used as an element of a matrix. We construct a special class of matrices, where each set is used as an element of a matrix. we construct a special latin square called as HCLS(Hamiltonian Circuit Latin Square)[7], and then generate the MMGSP(Modified Matrix for Generating Secret Paths)(see Appendix) applying the HCLS, from which the first part of the secret routes are designed. For the second part of the secret routes from these intermediate nodes to the destination node, the CMEMF(Connection Matrix Employing a Masking Function) (see Appendix) is employed. The reader is referred to[8] for extensive discussion of latin squares and their applications.

This paper is organized of the following three sections. Section 2 describes what MRNS network is. Section 3 gives an application of this matrix to the secret routing algorithms for the MRNS network. Then, the algorithm described is analyzed in Section 4. Finally, concluding remarks appear in Section 5.

2. Description of The MRNS Network

The MRNS network is constructed from the mixed radix number system(MRNS). The routing algorithms of the MRNS network are similar to those of the hypercube network.[9]-[12]

Each algorithm is composed of two phases. The first phase is to transmit the packet to a *randomly chosen intermediate node* through the secret route. The second phase is to send the packet from this intermediate node to the destination node along the secret path. This section provides the definition of the MRNS, gives a description of the MRNS network, and presents two routing algorithms of the MRNS network.

2.1 A Mixed Radix Number System (MRNS)

Let N be the total number of nodes of the MRNS network and let N be represented as a product of m_i's, $m_i > 1$ for $0 \le i \le n-1$.

$$N=m_{n-1}\times m_{n-2}\times \cdots \times m_1\times m_0$$

Then, each node u between 0 and N-1 can be represented as an n-tuple $(u_{n-1}\ u_{n-2}\cdots u_1\ u_0)$ for $0 \le u_i \le (m_i - 1)$. Associated with each ui is a weight wi, such that

$$u=\sum_{i=0}^{n-1} u_i\times w_i \text{ and } w_i=\prod_{j=0}^{i-1} m_j,\ w_0=1$$

Example 1: Let N = 24, m_i and w_i can be computed as follows.

$24=4\times3\times2$.

$m_0=2,\ m_1=3,\ m_2=4$

$w_0=1,\ w_1=2,\ w_2=6$.

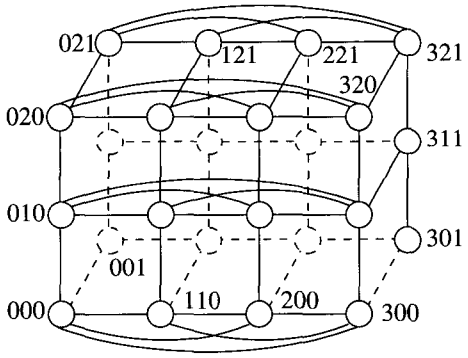


Figure 2. 4×3×2 MRNS Network

Then, $u = (u_2u_1u_0)$, $0 \le u_0 \le 1$, $0 \le u_1 \le 2$, $0 \le u_2 \le 3$ for any u in the range 0 - 23. 010 = (000), 23_{10} = (321) in this mixed number system. Any node can be described in this system between (000) and (321). Node (000) is directly connected to nodes (001), (010), (020), (100), (200) and (300) as shown in Fig. 2. For the sake of clarity, connection is not completed in this figure shown by dotted lines.

2.2 Structure of the MRNS Network

Each node $u = (u_{n-1}u_{n-2}\cdots u_i\cdots u_1)$ is connected to nodes $(u_{n-1}\ u_{n-2}\cdots u'_i\cdots u1)$ for all $1 \le i \le n$, where u'_i can be any integer

from $\{0, 1, \cdots, m_{i-1}\}$ except u_i itself. Given n-dimensions with mi number of nodes in the ith dimension, the following facts are described.

(1) The total number of links per node is $l=\sum_{i=0}^{n-1}(m_i-1)$

(2) The total number of links in the MRNS network is $N/2\times l$ where N is the total number of nodes.

(3) Each dimension is constructed as a complete graph. This means that for the i^{th} dimension, the total number of vertices is m_i and the total number of links is $m_i\times(m_i-1)/2$. Then, the link(p, q) is represented as (i, j), where
$j=\sum_{k=0}^{n-1}(m_i-k-1)+(q-p),\ p<q,\ p,\ q\in Z_{m_i}$

(4) The n-dimensional MRNS is a connected graph of diameter n.

Now, we examine the flexibility of designing the MRNS network. Given N nodes(N prime number), more than one kind of MRNS network can be designed based on considerations such as the dynamic security, the volume of data to be transmitted, and the cost of the hardware. If the network is more secure and has a large volume of data, then the network can be constructed with more links. However, the cost for constructing the network is a primary consideration, so the network should be designed with as few links as possible.

Example 2: Given 24 nodes, four kinds of MRNS networks, NK_1, NK_2, NK_3 and NK_4, can be designed.

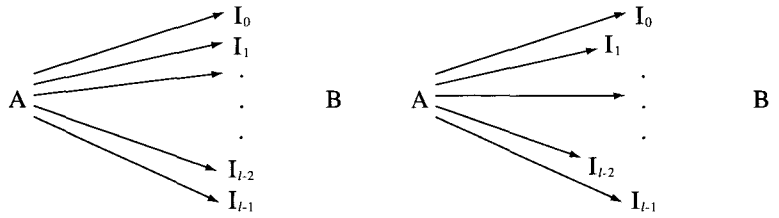
$NK_1 = Z_2 \times Z_{12}$
$NK_2 = Z_3 \times Z_8$
$NK_3 = Z_4 \times Z_6$
$NK_4 = Z_2 \times Z_3 \times Z_4$

From 2.2-(2), the total number of links are 144, 108, 96, 72 for NK_1, NK_2, NK_3, NK_2, respectively.

3. The Design of A Secret Routing Algorithm on the MRNS Network

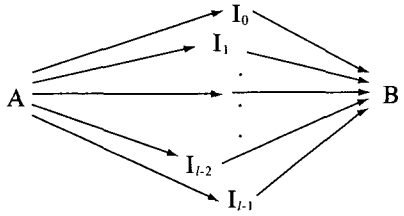
Two special matrices, the MMGSP (Modified Matrix for Generating Secret Paths) and the CMEMF(Connection Matrix Employing a Masking Function), are applied to the secret routing algorithm of the MRNS network. The secret routes consist of two sets of the paths P_1 and P_2; P_1 denotes a set of l paths from the source node to l distinct intermediate nodes, and similarly, P_2 denotes a set of paths from these intermediate nodes to the destination node. A secret routing algorithm therefore is composed of two phases. The first phase involves transmission of data from the starting node to l, randomly chosen, intermediate nodes along the paths belonging to P_1. The second phase involves sending the data, from these intermediate nodes to the destination node along the paths belonging to P_2.

The routing algorithms of the MRNS network are similar to those of the hypercube network. For the MRNS network, the number of channels is determined by the modular number for each dimension, while the modular number of each dimension in the hypercube network is always 2. Considering the structure of the MRNS network, the following two propositions are described and proven in [13].



(a) design a secret set of l node-disjoint paths

(b) an intermediate node is randomly selected on each path



(c) construct P_1 and P_2 via these intermediate nodes

Figure 3. The design of a set of l secret routes

Proposition 1: Let A and B be any two nodes in the MRNS network and assume that H(A, B)[12] < n. Then there are H(A, B) parallel paths of length H(A, B) between the nodes A and B.

Proposition 2: Let A and B be any two nodes of an n-dimensional MRNS network and assume that H(A, B) < n. Then there are l parallel paths between A and B, where $l=\sum_{i=0}^{n-1}(m_i-1)$. The length of each path is at most H(A, B) + 2.

For the design of secret routing algorithm, we describe the special matrix called the HCLS(Hamiltonian Circuit Latin Square).

Definition 1: The HCLS is constructed as follows: Given distinct n points, a Hamiltonian circuit $a_0 \rightarrow a_1 \rightarrow \cdots a_{n-2} \rightarrow a_{n-1} \rightarrow a_0$ is randomly selected. On the circuit each row of the matrix is obtained from the Hamiltonian path, starting at any position $a_j (0 \leq j \leq n-1)$ under the condition that no two rows begin at the same position. If a Hamiltonian path is $a_k \rightarrow a_{k+1} \rightarrow \cdots a_{k-2} \rightarrow a_{k-1}$, then the row obtained from it is $[a_k\ a_{k+1} \cdots a_{k-2}\ a_{k-1}]$

The Algorithm below describes how to construct l parallel routes for the n-dimensional MRNS network, where $l=\sum_{i=0}^{n-1}(m_i-1)$. Unlike other algorithms, this algorithm uses the HCLS to make partitions, each consisting of secret paths. Paths belonging to the same partition utilize the same dimensions at fixed times for the packets, if necessary. Since the source node has l channels, the data is dispersed into l packets and all packets of the data are transmitted to the neighboring nodes. These l channels are determined by two factors - the dimension that the HCLS assigns to each partition, and the link of each

dimension that each partition assigns to the packet. By considering the structure of each dimension in the MRNS network, Proposition 1 and Proposition 2, the second part of the route is determined. To construct disjoint paths in each partition, we use the property of the MMGSP(see Definition 1 in Appendix). But it is hard to check that each element(with represents a link) in the MMGSP is distinct from all other elements. Instead of examining all the elements, we just look at the element in the first and last columns. For the second phase of the route, suppose that all the elements in the first column are distinct. Then, the paths represented by the rows of the matrix, will be disjoint, even if the elements in all columns other than the first are the same. For the second phase of the route, the CMEMF is used. If two elements in the last column are the same, change one of the elements in the first column(in the same row as one of the elements in the last column), and compensate for this by adding an extra step at the end of that path.

SR_MRNS Algorithm

cobegin

(1) Split P_x into l packets P_{x0}, P_{x1}, P_{x2}, ···, P_{l-1} is the data at node x.

(2) Randomly choose a sequence from the n! permutations of <0, 1, 2, ···, n-1>.

(3) Design an (n × n) HCLS, and construct the (l × n) matrix MMGSP by extending the (n × n) HCLS designed above, where an element of the matrix is (m_i, o_j), m_i = the i^{th} dimension, o_j = the j^{th} link selected on that dimension. To accomplish this, a set k_1 of (m_i, o_j)s is randomly chosen from the set of all other (l - 1) sets and $|k_1| = n$.

(4) Select k_2 for the length of each path, $k_2 \in \{1, 2, \cdots, n\}$

(5) Using the (l - n) matrix described and k_2, construct pairwise vertex-disjoint paths D_0, D_1, ···, D_{l-1} from x to R_0, R_1, ···, R_{l-1}, respectively, each length is at most n.

(6) Compute a set of different bit positions of R_i and $\pi(x)$ for the i^{th} packet, $0 \le i \le l$-1.

(7) Randomly choose a sequence from the n! permutations of <0, 1, 2, ···, n-1> and design an (n × n) HCLS, and then construct the (l - n) matrix MMGSP.

(8) From Steps (6) and (7), we design the (l × (n + 1)) matrix CMEMF, and then make dynamical edge-disjoint paths E_0, E_1, ···, E_{l-1} from R_0, R_1, ···, R_{l-1}, respectively, to the destination node $\pi(x)$. Each path E_i has length ≤ n+1.

(9) Attach the i^{th} routing path E_i to (P_{xi}, D_i).

coend.

Given an n-dimensional MRNS network, the routing algorithm presented above describes how to construct l secret routes. The structure of each dimension is described as a complete graph of given nodes. The l secret paths of the MRNS network are more secure than those of the hypercube network, since the choice of the link at each dimension in the MRNS network is flexible.

Theorem 1: The SR_MRNS Algorithm requires $O(l)$ time, where l is the number of links per node.

Proof: The Algorithm is thus fairly straightforward. The time involved in performing all the steps except Step (6) is small. These eight steps of this algorithm does not contribute to an objectionable overhead.

Step (6) Computes different bit positions of intermediate node and destination node for each packet. In Step (1), the data is dispersed into l packets. Therefore, Step (6) needs $O(l)$ time.

The following example will provide a better understanding of the algorithm given above.

Example 3: Let x = (0000) and $\pi(x)$ = (1100), and let $m_i = 3$, where $(0 \le i \le 3)$. The total number of nodes $N = 3^4$, and the total number of links l per node is 8. Then, The Algorithm is executed as follows:

1) Following Steps (1) and (2), the data at node x is dispersed into l packets using the IDAF and select a sequence (2 1 3 0).

2) According to Step (3), design the HCLS and construct the (8 x 4) rectangular matrix.

the (4 x 4) HCLS

$$\begin{bmatrix} 2\,1\,3\,0 \\ 1\,3\,0\,2 \\ 3\,0\,2\,1 \\ 0\,2\,1\,3 \end{bmatrix}$$

(8 x 4) matrix

$$\begin{bmatrix} (2,1)\ (1,2)\ (3,1)\ (0,1) \\ (2,2)\ (1,1)\ (3,2)\ (0,2) \\ (1,1)\ (3,1)\ (0,2)\ (2,2) \\ (1,2)\ (3,2)\ (0,1)\ (2,1) \\ (3,1)\ (0,2)\ (2,2)\ (1,2) \\ (3,2)\ (0,1)\ (2,1)\ (1,1) \\ (0,1)\ (2,1)\ (1,2)\ (3,2) \\ (0,2)\ (2,2)\ (1,1)\ (3,1) \end{bmatrix}$$

3) In Step (4), choose 1,2,4,2,3,2,2,1 for the lengths of the paths, respectively and construct pairwise vertex-disjoint paths $D_0, D_1, \cdots, D_{l-1}$.

D_0 : (0000) → (0100) : (2,1)

D_1 : (0000) → (0200) → (0210) : ((2,2) (1,1))

D_2 : (0000) → (0010) → (1010) → (1012) → (1212) : ((1,1) (3,1) (0,2) (2,2))

D_3 : (0000) → (0020) → (2020) : ((1,2) (3,2))

D_4 : (0000) → (1000) → (1002) → (1202) : ((3,1) (0,2) (2,2))

D_5 : (0000) → (2000) → (2001) : ((3,2) (0,1))

D_6 : (0000) → (0001) → (0101) : ((0,1) (2,1))

D_7 : (0000) → (0002) : (0,2)

For synchronization, we add $(n-|D_i|)$s to the end of D_i, $0 \le i \le 7$

D_0 : ((2,1) s s s)

D_1 : ((2,2) (1,1) s s)

D_2 : ((1,1) (3,1) (0,2) (2,2))

D_3 : ((1,2) (3,2) s s)

D_4 : ((3,1) (0,2) (2,2) s)

D_5 : ((3,2) (0,1) s s)

D_6 : ((0,1) (2,1) s s)

D_7 : ((0,2) s s s)

4) In Step (6), locate the bit positions that differ between R_i and $\pi(x)$.

E_0 : (0100) → (1100) : different positions = ((3,1))

E_1 : (0210) → (1100) : different positions = ((1,1),(2,3),(3,1))

E_2 : (1212) → (1100) : different positions = ((0,2),(1,2),(2,3))

E_3 : (2020) → (1100) : different positions = ((1,2),(2,1),(3,3))

E_4 : (1202) → (1100) : different positions = ((0,2),(2,3))

E_5 : (2001) → (1100) : different positions = ((0,1),(2,1),(3,3))

E_6 : (0101) → (1100) : different positions = ((0,1),(3,1))

E_7 : (0002) → (1100) : different positions = ((0,2),(2,1),(3,1))

5) In Step (7), select a sequence (0 3 1 2) and design the (8 x 5) rectangular matrix.

the (4 x 4) HCLS

$$\begin{bmatrix} 0\,3\,1\,2 \\ 3\,1\,2\,0 \\ 1\,2\,0\,3 \\ 2\,0\,3\,1 \end{bmatrix}$$

(8 x 5) matrix

$$\begin{bmatrix} s & (3,1) & s & s & s \\ (0,1) & (3,1) & (1,1) & (2,3) & (0,1) \\ s & (1,2) & (2,3) & (0,2) & s \\ (3,3) & (1,2) & (2,1) & s & s \\ s & (2,3) & (0,2) & s & s \\ s & (2,1) & (0,1) & (3,3) & s \\ s & (0,1) & (3,1) & s & s \\ (2,2) & (0,2) & (3,1) & s & (2,3) \end{bmatrix}$$

6) In Step (8) and (9),

(P_{x0}, (2,1) s s s s (3,1) s s s)
(P_{x1}, (2,2) (1,1) s s (0,1) (3,1) (1,1) (2,3) (0,1))
(P_{x2}, (1,1) (3,1) (0,2) (2,2) s (1,2) (2,1) (0,2) s)
(P_{x3}, (1,2) (3,2) s s (3,3) (1,2) (2,1) s s)
(P_{x4}, (3,1) (0,2) s s s (2,3) (0,2) s s)
(P_{x5}, (3,2) (0,1) s s s (2,1) (0,1) (3,3) s)
(P_{x6}, (0,1) (2,1) s s s (0,1) (3,1) s s)
(P_{x7}, (0,2) s s s (2,2) (0,2) (3,1) s (2,3))

4. The Analysis of SR_MRNS Algorithm

We now analyze the algorithms above in terms of entropy[7] of how to select a set of secret paths. Intuitively, we realize that the computation of the entropy for it is very difficult, however, the important factors for this computation can be found by examining these routing algorithms. These factors are how to secretly choose l distinct intermediate nodes, and how to generate secret routes from the starting node to the destination node via these intermediate nodes secretly chosen above. The first factor means that if we select l distinct intermediate nodes more secretly, l secret routes designed by using these intermediate nodes, can obtain stronger security. In this paper, secret routes(from the starting node to n randomly chosen intermediate nodes and from these intermediate nodes to the destination node) must be designed under the condition that these routes are disjoint, however, it is impossible to algorithmically generate all the secret routes. In such case, by adding some constraints to the conditions of being secret routes, we can find an algorithm, which produces only small subsets of all the secret routes. From these reasons, we should develop a new method for constructing large subsets of all the secret routes, instead of evaluating the entropy for the second factor. Therefore, we now concentrate on the computation of the entropy of how to select l distinct intermediate nodes. The following three propositions are discussed in sequence.

Proposition 3: The entropy of selection of any l distinct intermediate nodes from N nodes is

$$\sum_{i=1}^{\binom{N}{l}} - \log P_j \times P_j,\ P_j = \frac{1}{\binom{N}{l}}$$

where $N = m_{n-1} \times m_{n-2} \times \cdots m_1 \times m_0$ and $l = \sum_{i=0}^{n-1} (m_i - 1)$

Proof: The number of cases for selecting l nodes fromN is $\binom{N}{l}$. Each case has the same probability of being chosen. Therefore, the case Pj has the following probability, $P_j = \frac{1}{\binom{N}{l}}$.

Proposition 4: If the HCLS is used for constructing node-disjoint paths(see the first phase of the algorithm), the entropy of selection of any node except the starting node is

$$\sum_{i=1}^{n} \sum_{j=1}^{\binom{N}{k}} -\log P_j \times P_j,\ P_j = \frac{1}{n \times w},\ w = \sum_{i=1}^{\binom{N}{k}} \prod_{j=1}^{n} (m_j - 1),$$

where j should be selected $\binom{n-1}{k-1}$ times, $0 \le j \le n-1$.

Proof: The i^{th} intermediate node I_i is determined by the length k of i^{th} path from the starting node x(see Routing algorithm 2; (3) & (4)), that is $d(x, I_i) = k$. The length k is selected from the range[1 ··· n] with the same probability 1/n. Since the number of nodes, each of which has the distance k from node x, is w and each of these nodes has the same probability 1/w of being chosen, the actual

probability of selecting the i^{th} intermediate node, is $1/(n \times w)$, where w is described above.

Using Proposition 4, Proposition 5 will be defined next.

Proposition 5: The entropy of selection of any l distinct intermediate nodes from (N - 1) nodes (except the starting node) is

$$\sum_{i=1}^{\binom{N-1}{l}} -\log P_i \times P_i,\ P_i = \prod_{j=1}^{n} \left(\frac{1}{n \times w}\right)^{a_m},\ \sum_{m=1}^{n} a_m \geq 0$$

Proof: The number of cases for selecting l distinct nodes from (N - 1) nodes, is $\binom{N-1}{l}$. The probability P_i of the i^{th} selection of l distinct nodes is a product of n probabilities, where the k^{th} probability is that a node is chosen as the k^{th} intermediate node, that is $1/(n \times w)$.

5. Conclusion

This paper presents parallel communication and secret routing algorithms in the MRNS(Mixed Radix Number System) network. Two topics are involved in this paper - parallelism and data security. For the aspect of parallelism, we construct a set of vertex-disjoint paths in the MRNS network employing the special matrices. Our algorithm for constructing l parallel paths in the n-dimensional MRNS network requires only $O(l)$ for the time complexity, while other algorithms, such as Rabin's Routing Algorithm, need more than $O(l)$. For the aspect of data security, this paper describes the topic of Routing security. Routing security is the security of the route taken by the data transmitted over the network. If the route is detected by the adversary, the probability is high that the data will be intercepted. In order to receive the data safely at the destination node, the route must be protected. To accomplish this, we select the intermediate nodes secretly and transmit the data via these intermediate nodes to the destination node.

References

[1] Seberry, J. and Pirprzyk, J., An Introduction to Computer Security. Prentice Hall, Englewood Cliffs, NJ,1989

[2] Choi, W., Chung, I. and Lee, S., "The Design of Information Dispersal using the FFT Algorithm(IDAF) for Secure and Fault-Tolerant Communications in Distributed Environment", J. Korea Info. Soc., vol. 23, no. 12, pp.1195 ~1200, 1996. 12.

[3] Knuth, D.E., The Art of Computer Programming, Vol 1: Fundamental Algorithms. Addison-Wesley, Reading, MA, 1983

[4] Rabin, M.O., "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," J. ACM, vol. 36, no. 2, pp. 335-348, Apr. 1989

[5] Valiant, L.G., "A Scheme for Fast Parallel Communication," SIAM J. Comput., vol. 11, no. 2, pp. 350-361, May 1982

[6] Bhuyan, L.N., and Agrawal, D.P., "Generalized Hypercube and Hyperbus Structures for a Computer Network," IEEE Trans. Comput., vol. 33, no. 4, pp. 323-333, Apr. 1984

[7] Chung, I., "Application of the Special Latin Squares to the Parallel Routing Algorithm on Hypercube," Parallel Routing Algorithm on Hypercube," J. Korea Info. Sci., vol. 19, no. 5,

pp. 569-578, Sep. 1992

[8] Denes, J. and Keedwell, A.D., Latin Squares and Their Applications. Academic Press, New York, 1974

[9] Saad, Y., and Schultz, M.H., "Topological Properties of Hypercubes," IEEE Trans. Comput., vol. 37, no. 7, pp. 867-872, July 1988

[10] Ibarra, O.H., and Sohn, S.T., "On Mapping Systolic Algorithm onto the Hypercube," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 1, pp. 48-63, Jan. 1990

[11] Wu, A.Y., "Embedding of Tree Networks into Hypercubes," J. Parallel Distrib. Comput., vol. 2, pp. 238-249, 1985

[12] Johnsson, S.L., and Ho, C.-T., "Optimum Broadcasting and Personalized Communication in Hypercube," IEEE Trans. Comput., vol. 38, no. 9, pp. 1249-1268, Sep. 1989

[13] Choi, W. and Chung, I., "Application of the Special Matrices to the Parallel Routing Algorithm on MRNS Network", Trans. Korea Info. Proc. Soc., vol. 3, no. 1, pp.55~62, 1996. 1.

[14] Kim, S. and Chung, I., "Analysis of Pseudo-randomized Routing Algorithm for Secure Communication", J. Korea Info. Soc., vol. 22, no. 3, pp.478~484, 1995. 3.

Appendix

Definition 1: Call the matrix M^1 as the MMGSP(Modified Matrix for Generating Secret Paths), no two entries in this matrix are the same. This matrix thus satisfied the following conditions.

i) $M^1 = [U_{i,j}]$, $U_{i,j} \subset \{Z_n + s\}\{0 \le i \le n-1, 0 \le j \le m-1, 0 \le l\}$, and where s means: "stay at the current node".

ii) $|U_{i,j}| = j + 1$

iii) $U_{i,j} \neq U_{k,j}$, if $i \neq k$

iv) $U_{i,j} \subset U_{i,j+l}$

v) $U_{i,j+1} = U_{i,j} + \{s\}$, if $s \in U_{i,j}$

Definition 2: Let F be a masking function and M^2 be an $(n \times m)$ matrix, where $M^2 = [u_{i,j}]$; $u_{i,j} \in Z_n$, $0 \le i \le n-1$, $0 \le j \le m-1$.

$$F_{r_0, r_1, \cdots, r_{n-1}}(u_{i,j}) = \begin{cases} u_{i,j}, & \text{if } u_{i,j} \in r_i,\ r_i \subset Z_n \\ s, & \text{otherwise} \end{cases}$$

The Connection Matrix Employing a Masking Function(CMEMF) is constructed according to Definition 1 employing the masking the masking function specified in Definition 2.

Definition 3: Given the MMGSP M^1 and r_i denoting the set of different bit positions between the i^{th} intermediate node and the destination node, this matrix is transformed into partial matrix by the masking function, then this transformed matrix is called the CMEMF(Connection Matrix Employing a Masking Function)

□ 著者紹介

배 용 근

1984년 조선대학교 전산기공학과 졸업(공학사)
1986년 조선대학교 전자공학과 졸업(공학석사)
현재 원광대학교 전자공학과 박사과정
1984년~88년 조선대학교 전자계산소 근무
1988년~현재 조선대학교 전자계산학과 조교수

※ 관심분야:마이크로 프로세서 응용, 병렬처리, 멀티미디어

정 일 용

1983년 한양대학교 공과대학 졸업(공학사)
1987년 미국 City University of New York 전산학과(전산학석사)
1991년 미국 City University of New York 전산학과(전산학박사)
1991년~94년 한국전자통신연구소 선임연구원
1994년~현재 조선대학교 전자계산학과 조교수

※ 관심분야:네트워크 관리, 분산시스템 보안, 코딩이론, 병렬 알고리즘