Security Enhanced Dynamic Routing using Quasigroups

Mr. K.J. Pavithran Kumar[1] and Mr. D. Vivekananda Reddy[2]

[1]M.Tech Student in Computer Science and Engineering, Sri Venkateswara University, Tirupati.

[2]Assistant Professor in Computer Science and Engineering Department, Sri Venkateswara University, Tirupati

Abstract—Security has become one of the major issues for data communication over wired and wireless networks. The dynamic routing algorithm is used to randomize delivery paths for data transmission which is compatible with popular routing protocols, such as the Routing Information Protocol in wired networks and Destination-Sequenced Distance Vector protocol in wireless networks, without introducing extra control messages. This paper proposes the dynamic routing algorithm with cryptography-based system design for more security in data transmission.

***Key words*—Security, Quasigroup cryptography-based system, dynamic routing, RIP, DSDV.**

1. Introduction

In the past decades, various security-enhanced measures have been proposed to improve the security of data transmission over public networks. Existing work on security-enhanced data transmission includes the designs of cryptography algorithms and system infrastructures and security-enhanced routing methods. Their common objectives are often to defeat various threats over the Internet, including eavesdropping, spoofing, session hijacking, etc.

Another alternative for security-enhanced data transmission is to dynamically route packets between each source and its destination so that the chance for system break-in, due to successful interception of consecutive packets for a session, is slim. The intention of security-enhanced routing is different from the adopting of multiple paths between a source and a destination to increase the throughput of data transmission (see, e.g., [1] and [2]). In particular, Lou et al. [3], [4] proposed a secure routing protocol to improve the security of end-to-end data transmission based on multiple path deliveries. The set of multiple paths between each source and its destination is determined in an online fashion, and extra control message exchanging is needed. Bohacek et al. [5] proposed a secure stochastic routing mechanism to improve routing security. Similar to the work proposed by Lou et al. [3], [4], a set of paths is discovered for each source and its destination in an online fashion based on message flooding. Thus, a mass of control messages is needed. Yang and Papavassiliou [6] explored the trading of the security level and the traffic dispersion.

The objective of this work is to explore a security enhanced dynamic routing algorithm based on distributed routing information widely supported in existing wired and wireless networks. We aim at the randomization of delivery paths for data transmission to provide considerably small path similarity (i.e., the number of common links between two delivery paths) of two consecutive transmitted packets. The proposed algorithm should be easy to implement and compatible with popular routing protocols, such as the Routing Information Protocol (RIP) for wired networks [7] and Destination-Sequenced Distance Vector (DSDV) protocol for wireless networks [8], over existing infrastructures. These protocols shall not increase the number of control messages if the proposed algorithm is adopted.

2. Problem Statement

The objective of this work is to explore a security-enhanced dynamic routing algorithm based on distributed routing information widely supported in existing networks. In general, routing protocols over networks could be classified roughly into two kinds: distance-vector algorithms and link-state algorithms [9]. Distance-vector algorithms rely on the exchanging of distance information among neighboring nodes for the seeking of routing paths. Examples of distance-vector-based routing algorithms include RIP and DSDV. Link-state algorithms used in the Open Shortest Path First protocol [10] are for global routing in which the network topology is known by all nodes. Our goal is to propose a distance-vector-based algorithm for dynamic routing to improve the security of data transmission. Before we proceed with further discussions, our problem and system model shall be defined.

A network could be modeled as a graph G = (N, L), where N is a set of routers (also referred to as nodes) in the network, and L is a set of links that connect adjacent routers in the network. A path p from a node s (referred to as a source node) to another node t (referred to as a destination node) is a set oflinks (N_1, N_2) $(N_2, N_3)\ldots(N_i, N_{i+1})$, where s $=N_1$, $N_{i+1}=$ t , $N_j \in N$, and $(N_j, N_{j+1}) \in L$ for $1 \leq j \leq i$. Let Ps;t denote the set of all potential paths between a source node s and a destination node t. Note that the number of paths in Ps;t could be an exponential function of the number of routers in the network, and we should not derive Ps;t in practice for routing or analysis.

Definition 1 (path similarity). Given two paths pi and pj, the path similarity Sim(p_i; p_j) for pi and pj is defined as the number of common links between p_i and p_j:

$$Sim(pi; pj)=|\{(Nx, Ny)|(Nx,Ny) \in p_i \ \wedge (Nx, Ny) \in p_j\}|,$$

Where N*x* and N*y* are two nodes in the network. The path similarity between two paths is computed based on the algorithm of Levenshtein distance [5].

Definition 2 (the expected value of path similarity for anytwo consecutive delivered packets). Given a source node s and a destination node t, the expected value of path similarity of any two consecutive delivered packets is defined as follows:

$$E[Sims;t]= \sum Sim(p_i, p_j) \cdot Prob(p_j|p_i) \cdot Prob(p_i),$$

where Ps;t is the set of all possible transmission paths between a source node s and a destination node t. Prob(pj|pi) is the conditional probability of using pj for delivering the current packet, given that pi is used for the previous packet. Prob(pi) is the probability of using pi for delivering the previous packet.

The purpose of this research is to propose a dynamic routing algorithm to improve the security of data transmission. We define the eavesdropping avoidance problem as follows:

Given a graph for a network under discussion, a source node, and a destination node, the problem is to minimize the path similarity without introducing any extra control messages, and thus to reduce the probability of eavesdropping consecutive packets over a specific link.

3. Security-Enhanced Dynamic Routing

3.1. Notations and Data Structures:

The objective of this section is to propose a distance-vector based algorithm for dynamic routing to improve the security of data transmission. We propose to rely on existing distance information exchanged among neighboring nodes (referred to as routers as well in this paper) for the seeking of routing paths. In many distance-vector-based implementations, e.g., those based on RIP, each node Ni maintains a routing table (see Table 1a) in which each entry is associated with a tuple (t, $W_{Ni,t}$;Nexthop), where t, $W_{Ni,t}$, and Next hop denote some unique destination node, an estimated minimal cost to send a packet to t, and the next node along the minimal-cost path to the destination node, respectively.

With the objective of this work in the randomization of routing paths, the routing table shown in Table 1a is extended to accommodate our security-enhanced dynamic routing algorithm. In the extended routing table (see Table 1b), we propose to associate each entry with a tuple (t,$W_{Ni,t}$, C_t^{Ni},H_t^{Ni}). , C_t^{Ni} is a set of *node candidates* for the nexthop (note that the candidate selection will be elaborated in Procedure 2 of Section 3.2), where one of the nexthop candidates that have the minimal cost is marked. H_t^{Ni} , a set of tuples, records the history for packet deliveries through the node Ni to the destination node t. Each tuple (Nj, h_{Nj}) in H_t^{Ni} is used to represent that N_i previously used the node h_{Nj} as the nexthop to forward the packet from the source node Nj to the destination node t. Let N_{bri} and $w_{Ni,Nj}$ denote the set of neighboring nodes for a node Ni and the cost in the delivery of a packet between Ni and a neighboring node N_j, respectively. Each node Ni also maintains an array (referred to as a link table) in which each entry corresponds to a neighboring node $N_j \in N_{bri}$ and contains the cost $w_{Ni,Nj}$ for a packet delivery. The proposed algorithm achieves considerably small path similarity for packet deliveries between a source node and the corresponding destination node. However, the total space requirement would increase to store some extra routing information. The size of a routing table depends on the topology and the node number of a network under discussions. In the worst case, we have a fully connected network. For each entry in the routing table shown in Table 1b, the additional spaces requirement would increase to store some extra routing information. The size of a routing table depends on the topology and the node number of a network under discussions. In the worst case, we have a fully connected network. For each entry in the routing table shown in Table 1b, the additional spaces required for recording the set of node candidates (as shown in the third column of Table 1b) and for recording the routing history (as shown in the fourth column of Table 1b) are O(|N|). Because there are |N| destination nodes at most in each routing table, the additionally required spaces for the entire routing table for one node are $O(|N|^2)$. Since the provided distributed dynamic routing algorithm (DDRA) is a distance-vector-based routing protocol for intradomain systems, the number of nodes is limited, and the network topology is hardly fully connected. Hence, the increase of the total space requirement is considerably small.

Table I: An Example Of The Routing Table For The Node N_i

Destination Node (t)	Cost $(W_{N_i,t})$	Nexthop
N_1	7	N_6
N_2	8	N_{21}
N_3	9	N_9
⋮	⋮	⋮

(a)

Destination Node (t)	Cost $(W_{N_i,t})$	Nexthop Candidates $(C_t^{N_i})$	History Record for Packet Deliveries to the Destination Node t $(H_t^{N_i})$
N_1	7	$\{\hat{N_6}, N_{20}, N_{21}\}$	$\{(N_2, N_{21}), (N_3, N_6), \cdots, (N_{31}, N_{20})\}$
N_2	8	$\{N_9, \hat{N_{21}}\}$	$\{(N_1, N_9), (N_3, N_9), \cdots, (N_{31}, N_{21})\}$
N_3	9	$\{\hat{N_9}\}$	$\{(N_1, N_9), (N_2, N_9), \cdots, (N_{31}, N_9)\}$
⋮	⋮	⋮	⋮

(b)

(a) The routing table for the original distance-vector-based routing algorithm.

(b) The routing table for the proposed security-enhanced routing algorithm.

3.2. A Secured Distributed Dynamic Routing Algorithm:

The DDRA proposed in this paper consists of two parts:

- Applying the cryptography-based system
- A randomization process for packet deliveries and
- Maintenance of the extended routing table.

3.2.1. Cryptography based system

The cryptography is used to increase the security in dynamic routing algorithm. The data will be encrypted by using the Quasigroup cryptography algorithm. Then the encrypted data is divided into packets. The encrypted packets will send to the destination using distributed dynamic routing algorithm. The cryptography process is as follows:

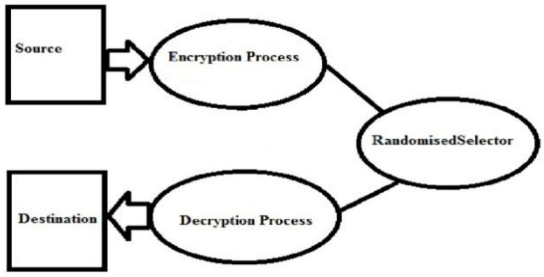

Fig. 1. Cryptography-based System

The cryptography based system encrypts the data and the encrypted data will be sending to randomization process. The randomized process will send the encrypted data to the destination through several paths. The encrypted data will be divided into packets and each packet is send to the destination through different paths. All the packets travelled through different paths will reach the destination and that encrypted data will undergo decryption process. The decryption process will decrypt the data and the destination will get the secure data.

3.2.2. Randomization Process

Consider the delivery of a packet with the destination t at a node Ni. In order to minimize the probability that packets are eavesdropped over a specific link, a randomization process for packet deliveries shown in Procedure 1 is adopted. In this process, the previous nexthop h_s (defined in H_{Ni} t of Table 1b) for the source node s is identified in the first step of the process (line 1). Then, the process randomly picks up a neighboring node in C_t^{Ni} excluding h_s as the nexthop for the current packet transmission. The exclusion of hs for the nexthop selection avoids transmitting two consecutive packets in the same link, and the randomized pickup prevents attackers from easily predicting routing paths for the coming transmitted packets.

Procedure 1 RANDOMIZEDSELECTOR (s, t, pkt)

1: Let h_s be the used nexthop for the previous packet delivery for the source node s.
2: if $h_s \in C_t^{Ni}$ then
3: if $|C_t^{Ni}| > 1$ then
4: Randomly choose a node x from $\{C_t^{Ni} - h_s\}$ as a nexthop, and send the packet pkt to the node x.
5: $h_s \leftarrow x$, and update the routing table of N_i.
6: else
7: Send the packet pkt to h_s.
8: end if
9: else
10: Randomly choose a node y from C_t^{Ni} as a nexthop, and send the packet pkt to the node y.
11: $h_s \leftarrow y$, and update the routing table of N_i.
12: end if

The number of entries in the history record for packet deliveries to destination nodes is |N| in the worst case. In order to efficiently look up the history record for a destination node, we maintain the history record for each node in a hash table. Before the current packet is sent to its destination node, we must randomly pick up a neighboring node excluding the used node for the previous packet. Once a neighboring node is selected, by the hash table, we need $O(1)$ to determine whether the selected neighboring node for the current packet is the same as the one used by the previous packet. Therefore, the time complexity of searching a proper neighboring node is $O(1)$.

3.3.3. Routing table maintenance

Let every node in the network be given a routing table and a link table. We assume that the link table of each node is constructed by an existing link discovery protocol, such as the Hello protocolin [11]. On the other hand, the construction andmaintenance of routing tables are revised based on the well-known Bellman-Ford algorithm [10] and described as follows:

Initially, the routing table of each node (e.g., the node N_i) consists of entries $\{(N_j, W_{Ni,Nj}, C^{Ni}_{Nj} = \{N_j\}, H^{Ni}_{Nj} = \emptyset)\}$, where $Nj \in Nbr_i$ and $W_{Ni,Nj} = \omega_{Ni,Nj}$. By exchanging distance vectors between neighboring nodes, the routing table of N_i is accordingly updated. Note that the exchanging for distance vectors among neighboring nodes can be based on a predefined interval. The exchanging can also be triggered by the change of link cost or the failure of the link/node. In this paper, we consider cases when N_i receives a distance vector from a neighboring node N_j. Each element of a distance vector received from a neighboring nodeN_j includes a destination node t and a delivery cost$W_{Nj,t}$ from the node$N_j$to the destination node t. The algorithm for the maintenance of the routing table of N_i is shown in Procedure 2, and will be described below.

Procedure 2 DVPROCESS(t;$W_{Nj,t}$)
1: if the destination node t is not in the routing table then
2: Add the entry (t, ($\omega_{Ni,Nj}$ + $W_{Nj,t}$), C^{Ni}_t = {N_j}, H^{Ni}_t = $\emptyset$).
3: else if ($\omega_{Ni,Nj}$ + $W_{Nj,t}$) <$W_{Nj,t}$ then
4: $C^{Ni}_t \leftarrow \{N_j\}$ and N_j is marked as the minimal-cost nexthop.
5: $W_{Ni,t} \leftarrow (\omega_{Ni,Nj} + W_{Nj,t})$
6: for each node $N_k \in Nbr_i$ except N_j do
7: if $W_{Nk,t} < W_{Ni,t}$ then
8: $C^{Ni}_t \leftarrow C^{Ni}_t \cup \{N_k\}$
9: end if
10: end for
11: Send (t, $W_{Ni,t}$) to each neighboring node $N_k \in Nbr_i$.
12: else if ($\omega_{Ni,Nj}$ +$W_{Nj,t}$)>$W_{Ni,t}$ then
13: if ($N_j \in C^{Ni}_t$) then
14: if N_j was marked as the minimal-cost nexthop then
15: $W_{Ni,t} \leftarrow MIN\ Nk \in Nbr_i\ (\omega_{Ni,Nk} + W_{Nk,t})$
16: $C^{Ni}_t \leftarrow \emptyset$
17: for each node $N_k \in Nbr_i$ do
18: if $W_{Nk,t} < W_{Ni,t}$ then
19: $C^{Ni}_t \leftarrow C^{Ni}_t \cup \{N_k\}$
20: end if
21: end for
22: Send (t, $W_{Ni,t}$) to each neighboring node $N_k \in Nbr_i$.
23: else if $W_{Nj,t} > W_{Ni,t}$ then
24: $C^{Ni}_t \leftarrow C^{Ni}_t \cup \{N_j\}$
25: end if
26: else if ($Nj \in C^{Ni}_t$) ^ ($W_{Nj,t} < W_{Ni,t}$) then
27: $C^{Ni}_t \leftarrow C^{Ni}_t \cup \{N_j\}$
28: end if
29: end if

First, for the elements that do not exist in the routing table, new entries for the corresponding destination nodes will be inserted (lines 1 and 2). Otherwise, $\omega_{Ni,Nj} + W_{Nj,t}$ is compared with $W_{Nj,t}$ saved in the routing table of N_i, and the following four cases are considered:

1) $\omega_{Ni,Nj} + W_{Nj,t} < W_{Nj,t}$ (lines 3-11). The corresponding minimal cost is updated in the routing table, and N_j is marked as the minimal-cost nexthop. Any neighboring node N_k which has an estimated packet delivery cost from N_k to t (i.e., $W_{Nk,t}$) no more than $\omega_{Ni,Nj} + W_{Nj,t}$ joins the candidate set C^{Ni}_t . It is to aggressively include more candidates for the nexthop to t with reasonable packet delivery cost (i.e., $W_{Nk,t} < W_{Ni,t}$). Compared to the Bellman-Ford algorithm, more than one neighboring node can be selected as the nexthop candidates in this step (lines 6-10) to accommodate multiple packet-delivery paths to the destination node t. Also, the selection policy described above can prevent the algorithm from generating the routing loops.

2) $(\omega_{Ni,Nj} + W_{Nj,t}) > W_{Ni,t}$ and N_j is in the set C_t^{Ni} of nexthop candidates (lines 13-25). Based on whether N_j is marked as the minimal-cost nexthop in the routing table of N_i, the following two cases are further considered. . N_j was marked as the minimal-cost nexthop (lines 14-22). For all neighboring nodes of N_i, the minimal cost to the destination node t is recomputed according to the distance vectors received from the neighboring nodes. Also, the nexthop candidates for the destination node t are reselected, and the selection policy is the same as lines 7-9 for Case 1. . Nj was not marked as the minimal-cost nexthop (lines 23 and 24). If $W_{Nj,t} > W_{Ni,t}$, N_j is removed from C_t^{Ni} .
3) $(\omega_{Ni,Nj} + W_{Nj,t}) > W_{Ni,t}$, and N_j is not in the set C_t^{Ni} of nexthop candidates (lines 26 and 27). If $W_{Nj,t} < W_{Ni,t}$, N_j is inserted into C_t^{Ni}.
4) Otherwise, nothing is done.

When a node Ni receives a distance vector from a neighboring node, Procedure 2 is used to maintain the nexthop candidates for each entry in the routing table of Ni. The time complexity of Procedure 2 maintaining the nexthop candidates is O(|N|). Furthermore, in the routing table of N_i, there are |N| entries in the worst case. Hence, the time complexity of maintaining the routing table is $O(|N|^2)$. Based on Procedures 1 and 2, our security-enhanced dynamic routing can be achieved without modifying the existing distance-vector-based routing protocols such as RIP and DSDV.

4. Conclusion

This paper has proposed a cryptography-based system for security-enhanced dynamic routing algorithm based on distributed routing information widely supported in existing networks for secure data transmission. The proposed algorithm is easy to implement and compatible with popular routing protocols, such as RIP and DSDV, over existing infrastructures. The above procedure will send the data more secure by providing encryption process to the data and the encrypted data will undergo dynamic routing process which is more secure in transferring the data from hop to hop.

References

[1] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichl, "Adaptive Multipath Routing for Dynamic Traffic Engineering," *Proc. IEEEGlobal Telecommunications Conf*. (GLOBECOM), 2003.
[2] C. Hopps, Analysis of an Equal-Cost Multi-Path Algorithm, Request for comments (RFC 2992), Nov. 2000.
[3] W. Lou and Y. Fang, "A Multipath Routing Approach for Secure Data Delivery," *Proc. IEEE Military Comm. Conf.* (MilCom), 2001.
[4] W. Lou, W. Liu, and Y. Fang, "SPREAD: Improving Network Security by Multipath Routing," *Proc. IEEE Military Comm. Conf.* (MilCom), 2003.
[5] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707-710, 1966.
[6] Secure Sockets Layer (SSL), http://www.openssl.org/, 2008.
[7] G. Malkin, Routing Information Protocol (RIP) Version 2 Carrying Additional Information, Request for comments (RFC 1723), Nov. 1994.
[8] C. Perkins and P. Bhagwat, "Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM '94*, pp. 234-244, 1994.
[9] J. F. Kurose and K.W. Ross, Computer Networking—A Top-Down Approach Featuring the Internet. Addison Wesley, 2003.
[10] J. Moy, Open Shortest Path First (OSPF) Version 2, Request for comments (RFC 1247), July 1991.
[11] D. L. Mills, DCN Local-Network Protocols, Request for comments (RFC 891), Dec. 1983.