Secure Routing Using Factual Correctness

Muthusrinivasan Muthuprasanna and Govindarasu Manimaran

Iowa State University
{muthu, gmani}@iastate.edu

Abstract. The routing protocols in use today operate on implicit trust among the different routers. Specifically, the distance vector routing (DVR) protocols compute routing tables in a distributed manner, based on this implicit trust. This trust model however fails to ensure the factual correctness of the routing updates, which is very critical for secure routing. We propose a neighbor update propagation model to ensure factual correctness and detect malicious activity by any subverted router. We also propose a secure DVR protocol based on this model using simple cryptographic primitives, and with minimal operational overhead.

1 Introduction

The research on routing protocols has proceeded along three different lines - Distance Vector (RIP), Path Vector (BGP) and Link State Protocols (OSPF). Malicious attacks, unintended misconfigurations, and simple/byzantine failures can lead to poisoning of the routing tables and result in drastic consequences [1]: AS7007 [2], AS3561 [3] incidents, etc. We focus on the distance vector routing (DVR) protocols here. They are simple efficient distributed algorithms that can be hijacked by modifying, replaying or deleting routing updates using subverted routers and links [4], and can result in sub-optimal routing, network partitioning, DoS attacks, etc. [5]. The desirable properties of secure routing protocols include: quick convergence, scalability, consistency, data integrity, origin authenticity and factual correctness [6]. Our primary focus here has been to ensure the factual correctness of the routing updates and to design a secure DVR protocol.

2 Related Work

Proposals to secure DVR protocols have been along two different lines - light-weight solutions providing limited security guarantees, and computationally-intensive solutions providing much higher security guarantees. The use of sequence numbers [4], consistency check (CC) [4] and PAIR [5] algorithms, digital signatures [7] and Intrusion Detection techniques [8] provide limited security guarantees. On the other hand, S-RIP [9] guarantees minimal factual correctness by using a reputation management framework, while SEAD [10], [11] uses a one-way hash chain to provide update security. In [12], a generic framework to implement secure protocols using a topological map has been proposed.

F. Boavida et al. (Eds.): NETWORKING 2006, LNCS 3976, pp. 1197–1203, 2006.

Our contribution in this paper is two-fold. Firstly, we propose a neighbor update propagation (NUP) model that can ensure factual correctness of the routing updates in an adversarial environment. Secondly, we design a light-weight secure DVR protocol based on the above model. Here, we assume an attack model where the attacker is free to choose any attack technique and/or attack agent.

3 Factual Correctness Concept

The main idea of the proposed model is as follows: a routing update generated by a sender is as usual sent to its one-hop neighbors (receivers), and additionally also to its two-hop neighbors (verifiers); such that a subsequent update triggered and sent by the receiver to its one-hop neighbors (sender, verifiers as above) can easily be evaluated by them for consistency, as they have the precise knowledge of the trigger and the resulting update.

3.1 Neighbor Update Propagation (NUP) Model

The DVR protocol enables every router to determine in a distributed manner, the next hop router for every other destination in the network. This simple protocol takes in as inputs the current routing table and the received routing updates and outputs a new routing table. Thus it is independent of the location/identity of the router and its neighbors; and given the same set of inputs to any other router, it would spew out the same output. We exploit this feature in our NUP model. We now define a group as consisting of a node (sender), a single one-hop neighbor (receiver) and all the two-hop neighbors directly linked to the receiver (verifiers). Thus there exist as many groups as the number of routers in the network. In Fig. 1, node 2 is a receiver in Group 1 and sender in Group 2, while node 4 is a verifier in Group 1 and receiver in Group 2.

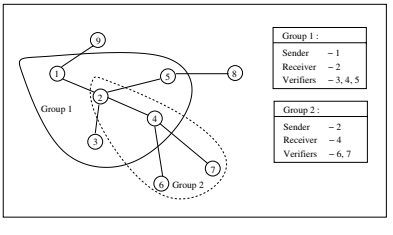

Fig. 1. Groups in NUP model

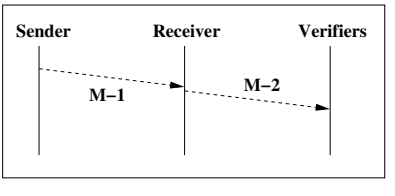

Fig. 2. Messaging in NUP model

The NUP model maintains a snapshot of the receiver's routing table at all its neighbors (verifiers). Any update that a sender (some neighbor) sends to the receiver is then forwarded to all the verifiers, who then maintain an updated cache of the receiver's routing table. The update that is subsequently generated by the receiver is validated by the verifiers for consistency with the cached routing

table. Thus, the basic principle is to provide the verifiers the precise knowledge of what they are supposed to receive, so that they can detect any malicious update sent by the receiver at any time. This, when implemented for every group in the network, provides the necessary factual correctness security guarantees. Consider the sample topology in Fig. 1. The routing table of node 4 is cached by nodes 2, 6 and 7. Now when node 2 sends an update to node 4, it is also propagated to nodes 6 and 7. Nodes 2, 4, 6 and 7 now update (cache of) node 4's routing table appropriately. If node 4 sends any update in future, nodes 2, 6 and 7 verify its validity by comparing it with the cached routing table for consistency. This consistency check ensures the factual correctness of node 4's updates.

3.2 Factual Correctness Guarantee

It is the process by which a router ensures whether the update generated by its neighboring router is actually the same that any trusted or well-behaved router in that place would have generated. This, when performed by every router in the network, can ensure that no router can act maliciously by altering any update in a manner other than it is supposed to.

The two messaging rounds of the NUP model are as shown in Fig. 2. In Round 1, the sender sends a routing update to the receiver ($M1$). In Round 2, the receiver forwards that update to the verifiers (its other one-hop neighbors) ($M2$). Additionally, we identify two types of routing updates sent distinctly, *source updates* and *forwarded updates*. Consider the protocol operation in Group 2 in Fig. 1.

1. A forwarded update is triggered by a previously received routing update, and hence its factual correctness can be trivially verified as explained above.
2. If the source update indicates that a new link is now operational, router 4 (receiver) can employ cryptographic verification (HTC computation, as explained later) to detect that change. eg. new link from router 2 to router 9.
3. If the source update indicates a link weight change, the change will also occur in the update that the other neighbors (sender) send to router 2 (receiver) and router 4 (verifier) in Group 1, and hence can be validated. eg. weight change of link connecting routers 1 and 2.
4. If the source update indicates that a link has gone down, there is no way for router 4 (receiver) to verify it as router 2 (sender) could explicitly drop all packets coming from that neighbor. eg. link failure between routers 2 and 5. Hence the receiver has to trust the claim, but it is not a problem - if it were a malicious router, it would be limiting its own connectivity in the network and hence implicitly checking the spread of the malicious routing information. However, connectivity could be lost in the network and could unavoidably affect routing.

Any malicious activity in a group can be detected by the other routers in that group; and as every router is a sender, receiver, or verifier in every group it belongs to, any single-router hijack is easily identifiable and hence the NUP model is *single-router hijack resistant*. However, the problem of ensuring packet forwarding in accordance with these secured routing tables is beyond the scope of this paper.

4 NUP-Based Secure DVR Protocol

The immediate problem now faced is, how would a sender send a routing update to a verifier to which it is not directly connected, without the receiver maliciously altering it? To ensure data integrity and origin authenticity for these updates, we propose the Hash Table-Chain (HTC) construct, using simple cryptographic primitives - collision-resistant one-way hash functions and symmetric encryption.

4.1 Hash Table-Chain (HTC)

The basic principle here is to morph every message appropriately using a one time pad, generated by the HTC. As the HTC is mutually agreed upon by the sender and a certain verifier, any alteration of the update message by the receiver can be detected. We assume the existence of a Public Key Infrastructure (PKI) in the network. All group members agree upon a pairwise mutually shared key (a random nonce) using public-private keys, say a_1 as in Fig. 3. They then use cryptographic (one-way) collision-resistant non-invertible hash functions H and M to generate a hash chain (Eqn. 1) of length k and a hash table (Eqn. 2) of size n using an offset padding respectively. For the first routing update, the sender uses the hash table corresponding to a_k as a one-time pad to morph the routing update using symmetric encryption. For all subsequent updates, it uses hash tables corresponding to $a_{(k-1)}$, $a_{(k-2)}$, ... ,a_1 respectively. Once the hash chain has been exhausted, the nodes re-negotiate another shared key using public-private keys, and the process continues. The symbols $||$ and $\oplus$, represent the concatenation and symmetric encryption operations respectively.

$$a_1, H(a_1), H^2(a_1), ..., H^k(a_1) \ : \ H^i(a_1) = H(H^{(i-1)}(a_1)), \ H^0(a_1) = a_1 \quad (1)$$

$$M(1||a_i), M(2||a_i), M(3||a_i), ..., M(n||a_i) \quad (2)$$

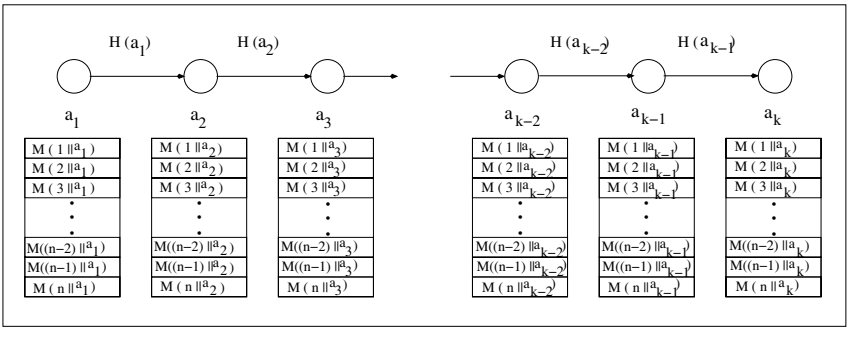

Fig. 3. Cryptographic Hash Table-Chain (HTC)

4.2 NUP-DVR Protocol

Consider the NUP-DVR protocol operation in Group 2 in Fig. 1. Let U_{24}, U_{26}, U_{27} and H_{24}, H_{26}, H_{27} be the DVR updates and the current mutually agreed pairwise hash tables, from node 2 to nodes 4, 6, 7 respectively. In Round 1, node 2 sends an update to node 4, consisting of all the updates (symmetric) encrypted with their respective hash tables (Eqn. 3). In Round 2, node 4 forwards these updates to its neighbors appropriately (Eqns. 4, 5). The symbols S, R, V and Seq represent the sender, receiver, verifiers and a sequence number (to prevent replay attacks) respectively. To avoid the update message size explosion problem, we additionally propose a novel Tree Rotation (TRot) technique employing simple checksum computations, to limit the growth of the DVR update message sizes.

$$M1_{24} = U_{24} \oplus H_{24} || U_{26} \oplus H_{26} || U_{27} \oplus H_{27} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{3}$$

$$M2_{46} = U_{24} \oplus H_{46} || U_{26} \oplus H_{26} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{4}$$

$$M2_{47} = U_{24} \oplus H_{47} || U_{27} \oplus H_{27} || S_2 || R_4 || V_6 || V_7 || Seq_i \tag{5}$$

4.3 Tree Rotation (TRot)

As explained in [5], every DVR update can be viewed as a DVR tree. This DVR tree is constructed as follows: place the sender at the root, then place the routers for which the sender is the predecessor as its children at depth 1. For every router, place it as a child node of its predecessor in the DVR tree. The pathsum metric is defined for every tree node [5], as the sum of its depth in the tree and the pathsum metrics of its immediate children. In Fig. 4, pathsum for nodes 9 and 4 are 2 and 5 respectively. We now formulate Tree Rotation (TRot) based on this pathsum property. Consider a sample DVR update (and its corresponding DVR tree) sent by node 2, for the topology in Fig. 1, as shown in Fig.4. If we now re-orient the tree with node 4 as its root, we get a new DVR tree and hence a new update as shown in Fig. 5. We see that the two routing updates are exactly the same because they both have the same physical underlying connectivity. Thus the DVR tree can be rotated multiple times, each having a different root, and they all would represent the same original DVR tree.

ID	Pred.	Path–Sum
1	2	3
2	2	12
3	2	1
4	2	5
5	2	3
6	4	2
7	4	2
8	5	2
9	1	2

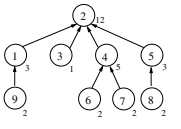

Fig. 4. Original Distance Vector Tree

ID	Pred.	Path–Sum
1	2	5
2	4	13
3	2	2
4	4	15
5	2	5
6	4	1
7	4	1
8	5	3
9	1	3

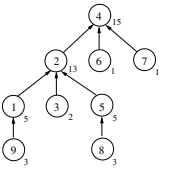

Fig. 5. Rotated Distance Vector Tree

Interestingly, the pathsum values differ in the different DVR trees and we use this feature to provide us the needed security guarantees. To incorporate this feature into the NUP-DVR protocol, we replace the different updates U_{24}, U_{26}, U_{27} with checksums C_{24}, C_{26}, C_{27} respectively, which are calculated as in Eqn. 6. Node 2 rotates the DVR update tree for U_{24}, randomly to some root R in the tree and computes its pathsum metric, R_{ps}, one each for the nodes 6 and 7. The choice of R is unknown to the receiver and serves as the basis of the security of the proposed optimization to the NUP-DVR protocol.

$$C_{24} = U_{24}, \quad C_{26} = R^i || R^i_{ps}, \quad C_{27} = R^j || R^j_{ps} \tag{6}$$

5 Performance and Security Analysis

We compare the NUP-DVR protocol with RIP [13]. In NUP-DVR, the overhead is due to the Round 2 update messages sent to all the verifiers. Thus the number of messages sent on the network is D-fold, where D is the average node degree in the network. Also, as the routers cache the routing tables of all the neighbors, the protocol require D-fold more storage and CPU cycles. As the routing tables used for packet forwarding are updated using Round 1 messages, there is no additional latency involved in the operation of NUP-DVR protocol, and its convergence properties are similar to that of RIP. The loss of Round 1 messages can be handled as in RIP. The Round 2 messages are critical to ensure correctness, and as their loss or explicit dropping would be interpreted as malicious behavior, it would need a retransmission/acknowledgment mechanism as in TCP, to ensure proper, ordered and timely delivery of these messages.

The proposed secure DVR protocol is single-router hijack resistant, as it can detect a single malicious sender, receiver or verifier in any group. However, if two subverted routers share a direct link, the round one messages in that group can be compromised without detection. Additionally, if subverted routers in disjoint groups act in collusion, they could falsely claim a direct link between them by sharing their private keys. A simple solution to the hidden false link problem would be to extend the group concept to depth k clusters. It is to be noted that, to the best of our knowledge, no scheme provides any guarantees against single router compromise, leave alone multiple router or collusion attack scenarios.

Additionally, the problem of verifier discovery by the sender needs to be addressed. This can be easily inferred from the update that the receiver sends to the sender in the corresponding group or more simply by explicit notifications. The sender can validate the verifiers by a simple key exchange protocol (e.g. Diffie-Hellman), and then generate a mutually agreed HTC.

6 Conclusions

It has become imperative to design secure and robust routing protocols in today's Internet, that can operate in a fairly robust manner in the presence of multiple malicious routers. Using novel concepts such as Neighbor Update Propagation

(NUP), Hash Table-Chain (HTC), and Tree Rotation (TRot), we have proposed a secure DVR protocol that is *single-router hijack resistant* and provides limited protection from multi-router collusion attacks. Possible extensions include use of appropriate data anonymizing techniques along with the proposed data morphing techniques to extend these concepts to the path vector protocols (BGP), to embed confidentiality and other policies practiced by network operators.

References

1. S. Bellovin, "Security Problems in the TCP/IP Suite", ACM CCR, pp. 32-48, 1989
2. "NANOG Archives(wow, AS7007!)", http://www.merit.edu/mail.archives/nanog/
3. "NANOG Archives(C&W Routing Instability)", http://www.merit.edu/mail.archives/nanog/
4. Smith, Murthy, Garcia-Luna-Aceves,"Securing Distance Vector Routing Protocols",SNDSS 1997
5. A. Chakrabarti, G. Manimaran, "An Efficient Algorithm for Malicious Update Detection & Detection in Distance Vector Protocols", IEEE ICC 2003
6. K. Bhargavan, D. Obradovic, C. Gunter. "Formal Verification of Standards for Distance Vector Routing Protocols", J. ACM, 49(4): 538-576, 2002
7. K. Zhang, "Efficient Protocols for Signing Routing Messages", NDSS, 1998
8. K. Bradley et. al., "Detecting Disruptive Routers: A Distributed Network Monitoring Approach", IEEE Symp. on Security & Privacy, 1998
9. Wan, Kranakis, Oorschot, "S-RIP: A Secure Distance Vector Routing Protocol", ACNS 2004
10. Y. Hu, D. Johnson, A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless AdHoc Networks", IEEE WMCSA 2002
11. Y. Hu, A. Perrig, Johnson, "Efficient Security Mechanisms for Routing Protocols", NDSS 2003
12. I. Avramopoulos, H. Kobayashi, R. Wang, A. Krishnamurthy, "Highly Secure and Efficient Routing", IEEE INFOCOM 2004
13. C. Hendrik, "Routing Information Protocol", RFC 1058, June 1988