A Double Authentication Scheme To Detect Impersonation Attack In Link State Routing Protocols

Dijiang Huang, Amit Sinha, Deep Medhi
{dhuang,asinha}@conrel.sice.umkc.edu, dmedhi@umkc.edu
School of Interdisciplinary Computing and Engineering
University of Missouri-Kansas City
Kansas City, MO 64110 USA

Abstract—In this paper, we present an authentication scheme to prevent impersonation attack in link state routing protocol. The existing authentication schemes are either simple to compute but vulnerable to attacks or too robust against attacks but has exponential computation cost. We introduce a Double Authentication (DA) scheme which provides authentication to the routing information data carried by the link state routing packets. In this scheme every router needs to sign the routing data twice with two different keys using a group keying scheme, which is based on one-way hash function. Based on our performance assessment, we found that this scheme is simpler to implement, computationally efficient and provides the degree of robustness desired with less communication overhead but has higher memory requirement.

Index Terms—Double Authentication (DA), Insider Attack, Impersonation Attack, Link State Routing.

I. INTRODUCTION

Present link state routing protocols are vulnerable to security attacks. There have been several attempts to address this issue. Perlman proposed a link-state routing protocol that achieves Byzantine Robustness [10]. Her protocol is highly robust but has a high computational overhead associated with the public-key encryption. Open Shortest Path First (OSPF) [8] link state routing protocol has provision to provide links state routing Packet Authentication (PA) based on a shared-key using a one-way hash function [13]. This scheme does not provide authentication for the routing data carried in these packets, hence can be very susceptible to security attacks. To overcome this, Murphy et al proposed digital signature (DS) scheme [9] to prevent tampering of Link-State Advertisements (LSAs). This scheme too uses some form of public-key encryption (different from that of Perlman's) and hence is expensive.

One of the attacks on the network routing protocol can be carried out by the trusted entity involved in the routing information exchange. This entity can be a router that has been misconfigured by a naive network administrator or an outsider who has total control on the entity because of a compromised administrative password. This type of attack is called the "insider attack". We classify insider attacks into two types:

1) A router originates incorrect local routing information: This type of attack can harm the network in two ways. First, the router may minimize the cost of the links associated with it, thereby compelling all the traffic to flow through it. If the network traffic engineering requirements are still fulfilled, there could be passive or active attack on the traffic data. This type of attack is the most difficult to detect and deal with, especially the passive attack. Second, if the router increases the cost of the links associated with it, the network traffic will avoid going through it. In a way the router tends to isolate itself. While we understand the potential threat posed by this attack to the network; dealing with it lies outside the scope of this paper.

2) A router impersonates other routers and generates forged network routing information: We refer to this attack as Impersonation Attack (IA). This is difficult to detect and can cause huge damage to the network. Since, most of the link state protocols provide authentication only at the packet level, the LSA can be easily forged without getting detected.

In this paper, we focus on the second type of attack. We refer to the router(s) responsible for IA as "bad" router(s). In our work, we assume there exists an Intrusion Detection System (IDS) that can sense abnormal network events like the authentication failure, network traffic congestion or router fight back[1] (see [14]). Here, we do not discuss the coordination mechanism between IDS and network routing protocols and how IDS makes decisions from the collected abnormal network events.

Our work is focused on the following goals: (a) present an authentication scheme that does not suffer from the performance issues of the public key scheme, (b) expedite the detection of "bad" routers involved in IA, and (c) provide some network design guidelines for this scheme to be effective.

In order to achieve our set forth goals, we propose a novel Double Authentication (DA) scheme based on a group keying scheme proposed in [4]. This involves authentication of each LSA twice, with two different keys using a one-way hash function. The basic idea behind our approach is to provide a degree of security as desired by a network designer at a relatively low cost. We show that our DA scheme is able to detect (a) a "bad"

---

[1]Fight back phenomenon: Since the link state routing uses reliable flooding to forward link state updates, the original routing information sent by a router can travel back to the sender. As specified in OSPFv2 standard [8], if the router finds the received routing information inconsistent with the one it sent out before, the router will resend the original routing information

router when there is only one of them impersonating routing information, and (b) multiple non-colluding "bad" routers.

We present a comparative assessment for the performance of the PA, DS and DA schemes, based on the memory requirement, CPU usage time and communication overhead. We also analyze the degree of security robustness of the scheme, which we define as the number of "bad" routers in the network that needs to collude for the scheme to fail. In our analysis, we assume all routers are connected using point-to-point links.

The rest of the paper is organized as follows. In Section II we discuss in detail the generation and working of DA. Section III provides a comparison between the existing authentication schemes for link state routing protocol and the DA scheme based on the memory requirement, CPU usage time, security robustness and communication overhead. Finally in Section IV we present the conclusion.

II. DOUBLE AUTHENTICATION

Our DA scheme uses a one-way hash function instead of the asymmetric keying scheme. Moreover unlike DS, DA does not use the end-to-end data origin authentication. Instead, it sets up an authentication chain that follows the LSA flooding path to provide data origin authentication. Here the router only needs to set up trust with its neighbors. This can decrease the number of keys used by symmetric cryptographic scheme. In this section we present the keying scheme, which forms the basis of our proposed authentication scheme. We also discuss the functioning of the scheme in order to provide authentication to the routing information.

A. Keying Scheme

In our scheme, the LSAs that are being flooded are individually authenticated twice by two different keys, i.e., each LSA is signed twice by every router when it floods the LSA to its neighbor(s). Authenticated codes are appended to the individual LSAs. The first authentication code generated by the router can be verified by every other router except the neighbor(s) to which the LSA is being flooded. This can prevent its neighbor(s) from altering the LSA. The second authentication code can be used by its neighbor(s) to check the integrity of both the LSA as well as the first authentication code. This is to ensure that if the LSA and the first authentication code were altered before it reached the neighbor, it can be detected.

The keying scheme for our approach is based on Secure Group Communication Keying Scheme (SGCKS) (see [4] for detailed description of SGCKS). In this scheme, each router has a set of Key Generation Seeds (KGS), which are used to generate the encryption/decryption key. A router uses its KGS to generate the sub-group key that is shared with its neighbor, and the sub-group key that is shared with everyone except its neighbor. The SGCKS scheme provides an efficient way to generate sub-group keys, thereby providing a way to reduce the number of authentication codes required for each LSA.

We assume that all the routers in the routing domain belong to the same KGS group. Group $G = \{x^i | i = 1, 2, ..., n; i \in N \ and \ n \geq 2\}$, where $|G| = n$. The group member is represented by x^i and we use $S_j^i = \{x^i, \ x^j\}$ to represent

any sub-group communication composed of group member x^i and x^j, where x^i and $x^j \in G$. In S_j^i, the superscript "i" means, x^i originates the sub-group communication S_j^i. The subscript "j" represents any sub-group member apart from x^i. In our proposed scheme, we are only interested in two types of sub-group communication, namely, S_j^i and its complement $S_{\bar{j}}^i = \{x^k | k = 1, 2, ..., n, \ k \neq j, \ k \in N, \ and \ n \geq 2\}$. Thus, we have $S_j^i, \ S_{\bar{j}}^i \subseteq G, \ |S_j^i| = 2, \ |S_{\bar{j}}^i| = n - 1, \ S_j^i \cup S_{\bar{j}}^i = G$, and $S_j^i \cap S_{\bar{j}}^i = \{x^i\}$.

In this paper we use x^i to represent router i and it's individual KGS as KGS^i, The individual key is represented as K^i, where $K^i = F(KGS^i)$. Here the function F is called key generation function. It is used to generate certain length of individual key and it is publicly known. F can be a bitwise logical operation or a one-way hash function depending on the implementation. The sub-group key for sub-group S_j^i is represented as K_j^i and the sub-group key for $S_{\bar{j}}^i$ as $K_{\bar{j}}^i$.

We enforce the security beyond the packet level. In our approach, every individual routing information data is authenticated. We use traditional authentication algorithm, for example Keyed-Hashing for Message Authentication (HMAC) [5]. The reason why we use HMAC and not DS is because the former is about a thousand times faster than the DS [15].

The scheme, SGCKS, is suitable for authentication because each router has a secret KGS, which can be used to generate sub-group key K_j^i for S_j^i communication between router x^i and its neighbor x^j. This sub-group key cannot be forged by other group members. Using SGCKS, $K_{\bar{j}}^i$ can be generated, which can be derived by every router in the group except the neighboring router, x^j.

Each LSA is authenticated twice by a router using two sub-group keys discussed above. The first authentication code generated by key K_j^i is used by its neighbor to verify the routing information data. The second authentication code generated by key $K_{\bar{j}}^i$ is used to guarantee that the neighbor does not alter the data. These authentication codes are appended at the end of each LSA. Router x^j forwards the second authentication code to it's neighbors so that they can verify it's integrity. This is a distributed scheme where each router needs to maintain the trust-relation only between its neighbors. The trust is built up when the router is added into the network by assigning a KGS from key distribution center. The trust-relation among network neighbors will tell the router what KGS group its neighbor belongs to. This helps the router to compute the sub-group keys in advance and use them when needed. The trust build-up, KGS generation and updates are out of scope of this paper.

B. Generating the DA

The DA generation rules are listed below:
1) When router x^i creates an LSA, it sends the LSA to its neighbor x^j as $LSA_{(\bar{j}),(j)}^i$.
2) When router x^j receives an LSA from its neighbor router x^i, it forwards it to its neighbor x^k as $LSA_{(\bar{k}),(\bar{j}),(k)}^j$.

We present an example shown in Fig. 1.

The superscript on LSA identifies the router that flooded the LSA. In rule 1, when router x^i creates the LSA, it needs to attach two authentication codes to it. We use subscript $(\bar{j})$ and (j)
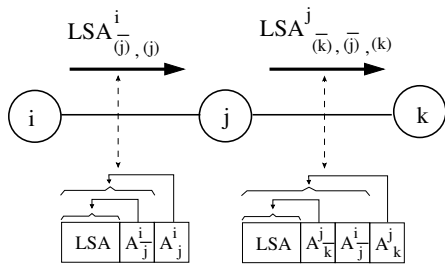
Fig. 1. The DA example

to represent authentication codes generated by key $K_{\bar{j}}^i$ and K_j^i, respectively. Note that authentication code A_j^i should authenticate both the LSA and authentication code $A_{(\bar{j})}^i$. This will help detect if the LSA and $A_{(\bar{j})}^i$ have been altered before reaching the neighbor, as any change in the information will be reflected in A_j^i. This is the reason $_{(j)}$ is at the rightmost side in our representation. This is same for rule 2 where the last authentication code (A_k^j) will cover previous two authentication codes and the LSA.

When x^j receives an LSA, it first detects that the source of LSA is its neighbor x^i. Then, router x^j uses sub-group key K_j^i to verify authentication code A_j^i. Once authenticated, router x^j uses sub-group keys K_k^j and $K_{\bar{k}}^i$ to generate authentication codes for both its neighbor x^k and x^k's neighbors, before forwarding this LSA to x^k. Note, as described above, authentication code $A_{(\bar{j})}^i$ is attached after $A_{(\bar{k})}^j$ as a part of the authenticated data and is forwarded to x^k. This can be used by x^k to verify that x^j has not altered the LSA. Authentication code $A_{(\bar{j})}^i$ needs to be attached after authentication code $A_{(\bar{k})}^j$. This is because when x^k forwards the LSA to further hops, authentication code $A_{(\bar{j})}^i$ is not attached. So, authentication code $A_{(\bar{k})}^j$ should not cover it. But these two authentication codes should be covered by code A_k^j. This will help detect any active attackers who might alter the LSA and authentication codes. In our scheme only the originator sends the LSA with two authentication codes attached to it. The intermediate routers generates two authentication codes but forwards three codes.

Finally, router x^j forwards the $LSA_{(\bar{k}),(\bar{j}),(k)}^j$ to router x^k. When router x^k receives this LSA, it first verifies authentication code A_k^j, and then checks code $A_{(\bar{j})}^i$. If authenticated, it just follows same steps as router x^j does in order to generate new authentication codes before forwarding the LSA to its neighbors.

III. COMPARATIVE ASSESSMENT

The two authentication schemes for link state protocols that have been proposed are, the PA [8] and the DS schemes [9]. The DA scheme is inspired by the DS scheme where authentication is provided for each LSA. In this section we draw a comparison between these schemes based on the memory requirement, CPU usage time, communication overhead and robustness in terms of security that each provides.
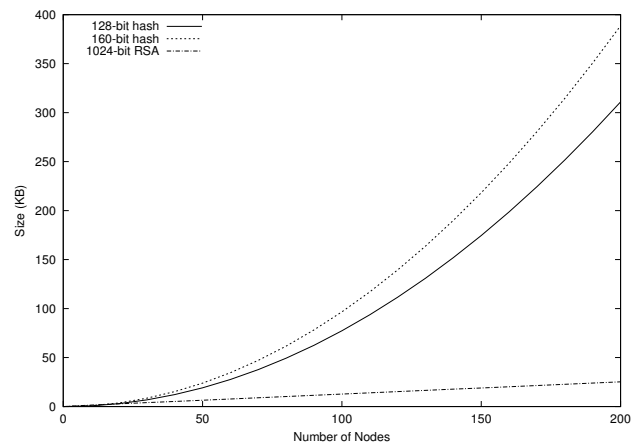

Fig. 2. Memory required to store the keys for various schemes

A. Memory Requirement

In the PA scheme, each router needs to store just one shared key which is used for cryptographic authentication. Hence, the memory requirement is negligible when compared to the DS and DA schemes.

In case of DS scheme, each router needs to store its public and private keys and the public keys of all other routers in the network and that of the trusted entity [2]. Assuming that there are N routers in the network and the size of the key is s, the worst case memory required by each router is equal to $s * (N + 2)$. Thus space complexity for this scheme is $O(N)$. The RSA public key scheme now suggests a 1024-bit key size, since 512-bit size can be vulnerable to attacks [11] [6]. Assuming that 1024-bit RSA is being used for DS scheme, we plot the memory size each router needs to store the keys with varying number of nodes in the network (see Fig. 2).

The worst case space complexity for DA scheme varies as $O(N^2)$ in a network with N nodes (see [3] for details). The memory size equals $s * N * (N-1)/2$ for a key size of s. Fig. 2 presents the memory requirement for a router for both 128-bit and 160-bit hash function with varying network size.

Here, we find that the space complexity for DA scheme is worst when compared to other schemes. Though the analysis presented in [3] suggests dividing the network into subgroups in order to reduce the memory requirement, the DA scheme currently does not support it. This aspect is presently being looked into and will be outlined in future. The higher memory requirement of this scheme, which in fact is below half a megabytes for a 200-node network, may or may not be an issue for currently available routers. For example, Cisco 7500 series router [16] has one 32MBytes (upgradeable to 128 MBytes) DRAM, one 64 MBytes (upgradeable to 256 MBytes) DRAM, one 16 MBytes (upgradeable to 110 MBytes) flash memory. Since we expect future generation routers to have more memory capability as with faster processors, we believe that the additional memory requirement warranted under our approach is not too far fetched.

---

[2] According to Murphy [9] this can be a Certificate Server which certifies the identity of a router. We assume one such entity in the network.

B. CPU Usage Time

We assume that the network is optimally designed in accordance with the guideline provided by Aho and Lee [1]. Therefore, the worst case complexity of the number of LSAs[3] contained in a single link state update packet is $O(N^{4/3})$ for a network with N nodes. For comparing the DS and DA scheme, we have used the benchmark provided in [15] for speeds of the hashing algorithms and the encryption/decryption algorithms. The validity of using this benchmark is discussed in [3].

In the PA scheme, a router authenticates the entire link state packet at once. The other two schemes presented here provides authentication for every LSA contained in the packet. Therefore, the CPU time for PA scheme is negligible as compared to the other schemes.

In DS scheme, a router needs to sign each LSA before flooding it. Each router signs its LSA by first running a one-way hash function on the LSA data and then using a private key to sign the digest [9]. Therefore, the CPU usage time per LSA is the time required by the hashing function to create the message digest and the time needed for encrypting this digest using the router's private key. Now, a router can receive $O(N^{4/3})$ LSAs in the worst case. Hence the router needs to first use the relevant public key to decrypt the message digest for each LSA. It then has to use the keyed one-way hash function on the LSA to authenticate it with the decrypted message digest. We have used the 1024-bit RSA public key scheme and MD5 one-way hash algorithm, to assess the CPU usage time for this scheme. Fig. 3 presents the assessment for the above scenarios, i.e., when a router originates LSAs, and when it receives LSAs.

In DA scheme, a router originating the LSA needs to run one-way hash function twice with different keys, as discussed in Section II. The same process needs to be carried out by the router receiving the LSAs. In Fig. 3 we show the worst case scenario for the same.

We find that the time complexity for DS scheme is worst as compared to the other schemes. This is obvious from the fact that public key scheme has an exponential computation overhead. Computation of the shortest path is known to be the most time consuming process for a router CPU. We have plotted the CPU usage time for shortest path calculation (SPF) by Cisco 7513 router using the formula given by Shaikh and Greenberg [12] in Fig. 3. We observe that the cost for our approach falls below the SPF with increasing number of nodes. Hence implementation of the DA scheme will not require any process upgrades. Whereas, for implementing the DS, the computational time associated with this scheme should decide the processing power needed for the router.

C. Communication Overhead

The communication overhead is the size of the information that has to be carried along with the packet in order to support various schemes.

In the PA scheme, each packet carries the message digest generated by running the one-way hash function over the link
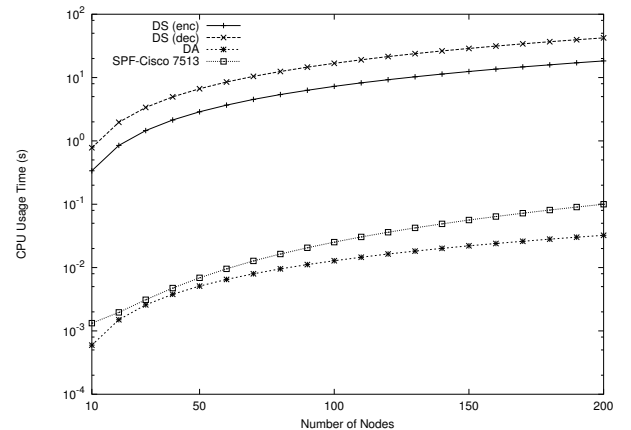

Fig. 3.   CPU processing time required by various schemes

state packet. This message digest of 128 bits is appended to all the link state routing packets [8].

For DS scheme, the signature for each LSA has a variable size. This message size can technically vary from 0 to 1024 bits for 1024-bit RSA scheme. Assuming each size is equally likely, we find that on an average the length of the signature that has to be attached for each LSA is 512 bits.

In case of DA scheme, authentication codes generated by three different hash keys need to be attached for each LSA by the intermediate node. The LSA originator attaches only two authentication codes generated by two different hash keys. Assuming that we use a 128-bit hashing algorithm, the total size of authentication codes per LSA will be 384 bits for intermediate routers, and 256 bits for the originator.

Hence, the communication overhead is maximum for the DS scheme. The overhead associated with PA is negligible.

D. Security Robustness

The PA scheme provides authentication to routing protocol packet but does not provide any authentication to routing data carried in these packets in the form of LSAs. So any faulty or subverted router in between the source and destination router can fiddle with this information and will go undetected.

The DS scheme provides a very strong authentication for each LSA within the link state update packets. The strength of this scheme comes from the asymmetric keying scheme implemented by it. This scheme is so robust that a "bad" router will always get detected unless every single node in the network has been hacked.

In order to analyze the security robustness of the DA scheme, we classify network intrusion into three types depending on the number "bad" routers within the network. The first type is when there is one or multiple "bad" routers within a network which do not collude. The second type of intrusion is when more than one "bad" routers in the network collude to impersonate other routers and generate forged routing information. The third classification is the special case of second when the "bad" routers partition the network. This means the "bad" routers can completely control all communication among the partitioned network. In this case, the forged routing information may not travel back to the originator.

---

[3]Moy in his analysis [7] indicated that the average size of the LSA to be 64 Bytes. We assume the size to have doubled for our analysis.

Our scheme deals with the first class of intrusion efficiently and can immediately detect the "bad" router (see Section II for detail). For second class of intrusion, the DA can provide the helpful information for the IDS to limit the region where the "bad" routers located, by maintaining the record for authentication codes. In this paper, we do not discuss how to trace back to the "bad" router. For the third class of intrusion, the DA can not provide any useful information to intrusion detection system. That means the network is totally hacked.

In the DA scheme the network designer has to decide the minimum number of nodes (k) that partitions the network. The network using the DA scheme will become unsafe from IA when these k routers go "bad" and they collude to harm the network.

IV. CONCLUSION

In this paper we have presented the Double Authentication scheme for link state routing protocols to detect impersonation attack. This scheme provides authentication with two different keys using a one-way hash function. In our approach, a single subverted router can be easily detected by its neighbors. When the network is partitioned by subverted routers, the DA can only be effective within a single partition. So, our DA can be as strong as Digital signature which provides source authentication when there is single or multiple subverted routers that do not work together. In the case where "bad" routers creates multiple partitions in the network, DA can provide security which is as good as PA. We should provide strong security to those network nodes that can easily partition the network. While we believe that link state routing protocols do require a strong authentication scheme, this should not come at the cost of computational efficiency. In this regard, DA scheme fairs well, with good computational efficiency and low communication overhead, though it has higher memory requirement. Therefore, DA lies in the huge gap between the PA and the DS scheme, thereby giving some flexibility in choosing the authentication scheme to be implemented in the link state routing protocols. Reducing the space complexity is one of the intended future work.

REFERENCES

[1] A. V. Aho and D. Lee. "Hierchical Networks and the LSA N-Squared Problem in OSPF Routing", *IEEE GlobeCom*, 2000.

[2] F. Harary, "Graph Theory", *Addison Wesley Publishing Company*, 1969.

[3] D. Huang, A. Sinha and D.Medhi, "On Providing Confidentiality in Network Routing", Technique Report, University Missouri Kansas City, October 2002. Available from http://conrel.sice.umkc.edu/HRP/.

[4] D. Huang and D. Medhi, "A Flat Group Keying Scheme to Support 'Any to Any' Secure Subgroup Communication", technical report, University Missouri Kansas City, October 2002. Available from http://conrel.sice.umkc.edu/HRP/.

[5] H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC2104, February 1997.

[6] A. Lenstra and E. Verheul, "Selecting Cryptographic Key Sizes", *Journal of Cryptology*, 14(4): 255-293, 2001.

[7] J. Moy, "OSPF Protocol Analysis", RFC1245.

[8] J. Moy, " OSPF version 2", RFC2328, April 1998.

[9] S.L. Murphy, M. Badger and W. Wellington, "OSPF with Digital Signatures", RFC2154, June 1997.

[10] R. Perlman, "Network Layer Protocols with Byzantine Robustness", MIT/LCS/TR-429, October 1988.

[11] M. Robshaw, "Security Estimates for 512-bit RSA", Technical Notes and Reports, RSA Laboratories, June 29, 1995

[12] A. Shaikh and A. Greenberg, "Experience with Black-Box OSPF Measurement", *Internet Measurement Workshop*, 2001.

[13] W. Stallings, "Cryptography and Network Security: Principles and Practice, Second Edition", *Prentice Hall*, 1998.

[14] B. Vetter, F. Wang and S.F. Wu,"An Experimental Study of Insider Attack for OSPF Routing Protocol", *IEEE International Conference on Network Protocols*, pp. 293 - 300, October 1997.

[15] http://download.baltimore.com/keytools/docs/v51/pro/j-docs/html/devguide/projdevguide-C.3.html

[16] http://www.cisco.com