

A Key Distribution Scheme for Double Authentication in Link State Routing Protocol

Dijiang Huang Amit Sinha Deep Medhi

Computer Science and Electrical Engineering Department
University of Missouri–Kansas City
Kansas City, Missouri 64110, USA.
{dhuang, asinha, dmedhi}@umkc.edu

Abstract

The Double Authentication (DA) scheme presented in [1] is designed to provide security against impersonation attack to link state routing protocol at a lower computational cost as compared to the existing schemes, such as, digital signature scheme [2]. In this paper, we present a key distribution scheme that can be used for generating and distributing keys to provide DA. This scheme leads to a storage complexity for each router that varies linearly with the number of routers in the network in the worst case (fully connected network with n nodes). Moreover, for router with four or less average number of links, the storage complexity falls below $\log_2 n$. This scheme also increases the security robustness of DA as the subverted routers can collude only if they are neighbors.

1 Introduction

Security is of immense concern for link state routing protocols. One mechanism of providing security is by authenticating the routing information being exchanged by each router. For example, open shortest path first (OSPF) standard (see RFC2328 [3]) has provision for links state routing *packet level authentication* based on a shared-key using a one-way hash function [4]. This scheme does not provide authentication for each Link State Advertisement (LSA) carried in these packets, hence can be very susceptible to impersonation attacks, in which a subverted router impersonates other routers and generates forged network routing information. To overcome this, Murphy et al proposed digital signature (DS) scheme [2] to prevent tampering of LSAs. This scheme uses public-key encryption which suffers from exponential computational cost. The worst case storage complexity varies linearly with the number of nodes. In order to bring down the computational complexity, researchers have proposed hash chain based schemes [5][6] to provide data origin

authentication. These schemes use hash values as a credential or a key used for keyed-hashing for message authentication (HMAC)[7], which is computationally efficient, but it requires a loose synchronous mechanism; thus, it can conflict with the operation mode of many routing mechanism, which may not have a synchronized framework. Furthermore, each router needs to retain a unique pre-computed hash chain and $n - 1$ hash values for other $n - 1$ routers within the link state routing domain.

Since, current link state protocols provide authentication only at the packet level, the LSA can be easily forged, making it difficult to detect a attack. The double authentication (DA) scheme presented in [1] is designed to prevent impersonation attacks. DA is based on a key distribution scheme, and it provides authentication to each LSA carried by the link state routing packets. It requires every router to sign the routing data twice with two different keys which has been provided by a key distribution center (KDC). The first key used is shared by every node except the next-hop¹ in the entire network, while the second key is shared between the sender and its next-hop only. The KDC generates these keys using a one-way hash function, which brings down the computational cost of the scheme as compared to the DS scheme. However, in the worst case, the storage complexity for the scheme grows quadratically with the number of nodes. For a moderate to large size network, this storage complexity forms the bottleneck for the key distribution scheme.

In this paper, we propose a new key distribution scheme that can be used for DA. The new scheme still advocates the use of two different keys by the DA. But here, the first key is shared only by the sender and all

¹Next-hop is the router to which the sender has a direct link. This is the router through which the sender intends to forward the routing packet.

the neighbors of its next-hop, while the second key is shared by the sender and its next-hop. This helps in bringing down the storage requirement from *quadratic* to *linear* in the worst case. Besides this, the new key distribution scheme also improves the security robustness of the DA scheme. Note that, when multiple subverted routers collude to generate the forged routing information, the DA scheme can only provide limited information to identify one of subverted routers. The new key distribution scheme ensures that the subverted routers can collude if and only if they are neighbors, and therefore, making it easier to detect the routers responsible for the attack. In our paper, we assume that the network topology is static and there is no addition or removal of any router/links.

The rest of the paper is organized as follows: In Section 2 we present an overview of the DA scheme. In Section 3 we describe the new key distribution scheme that can be used by DA. Section 4 we present the performance assessment of the key distribution scheme from storage and security point of view. We finally conclude the paper in Section 5.

2 Double Authentication Scheme: Overview

In [1] we proposed a DA scheme to provide security to link state routing protocols against impersonation attack. In this section we provide a brief overview that covers the working of DA and complexity due to the key distribution scheme. For brevity we use, O-DA to represent the DA scheme that uses the old key distribution approach, and N-DA for DA scheme that uses the new key distribution mechanism that we present in this paper.

In the O-DA scheme, LSAs that are 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). Authentication codes are then appended to each individual LSA. 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 reaches the next-hop neighbors.

A brief description of mathematical notations used by DA is given in Table 1.

We assume that all the routers in the link state routing domain belong to the same group². Group

²For example, using OSPF, an area can be defined as a link

Table 1: Notation

V	set, $ V = n$	x	a node, $x \in V$
S	subset, $S \subseteq V$	K	authentication key
Superscript	the source of communication peer(s)		
Subscript	the destination of communication peer(s)		

$V = \{x^i | i = 1, 2, \dots, n; i \in \mathcal{N} \text{ and } n \geq 2\}$, where $|V| = n$. We use x^i to represent router i (group member i) and $S_j^i = \{x^i, x^j\}$ to represent any sub-group communication composed of group member x^i and x^j , where x^i and $x^j \in V$. In other words, in S_j^i , the superscript “ i ” means that x^i originates the sub-group communication S_j^i , the subscript “ j ” represents another sub-group member x^j . Thus, the group members composition of S_j^i and S_i^j are the same, the difference is who generates the sub-group communication. In the O-DA scheme, only two types of sub-group communication are interesting, namely, S_j^i and its complement $S_j^i = \{x^k | k = 1, 2, \dots, n, k \neq j, k \in \mathcal{N}, \text{ and } n \geq 2\}$, where $|S_j^i| = 2$, $|S_j^i| = n - 1$ and $S_j^i \cap S_j^i = \{i\}$. The group members of S_j^i will cover every one except member x^j . For loosely connected networks, using sub-group key of S_j^i , the O-DA’s hop-by-hop authenticating contains much redundant information that are not necessarily involved in group membership validation. We note that the authentication code generated by using sub-group key of S_j^i can be validated only by neighbors of router x^j ; However, it also can be validated by the other routers that are not the neighbors of x^j . This redundancy introduces two problems. First, O-DA requires each group member to retain a set of secrets to generate proper sub-group keys.³ It increases the secrets size stored by a router, which is of the order of $O(n^2)$. Second, the collusion by any routers combinations with the router x^j , is possible, which would go undetected.

3 Key Distribution Approach

The DA by itself is designed to just use the two keys provided to it by the KDC to do the authentication. Hence the onus lies on the key distribution mechanism to bring down the storage cost and also overcome the security hole that remains unplugged in the O-DA. In this section we present the algorithm used by the new key distribution scheme and how the DA can make use of the keys in order to overcome the above issues.

state routing domain. This is because the link state advertisements are only flooded within an area.

³Refer to [8] for technical details.

3.1 Key Generation Algorithm

We use symmetric key scheme, for example, keyed-hashing for message authentication (HMAC) [7], for generating the authentication code and for its verification. We note that proper choice of group key for known link state routing network topology can decrease the number of keys possessed by a router as well as increase the degree of security. Here, we present an algorithm used by a KDC to generate the set of shared keys to be distributed to the routers.

We assume that the network topology is known in advance. We consider a router within the link state routing domain as a vertex x^i of and set containing all the vertices, $V = \{x^i | i \in \mathcal{N}\}$, and $|V| = n$. Γ is the set that contains the sub-groups of routers to which the sub-group keys have been already distributed. The pseudo code of the key generation algorithm is described in Table 2. The complexity of the algorithm varies *linearly* with the number of routers within the link state routing domain.

Table 2: Key Generation Algorithm

Step 1	$V = \{\text{All routers in the link state routing domain}\};$ $\Gamma = \{\phi\};$ distribute $K(S_j^i)$ for each directly connected pair x^i and x^j , where $x^i, x^j \in V$; put each $\{x^i, x^j\}$ in Γ ;
Step 2	if $V \neq \{\phi\}$ and \exists unmarked $x^i \in V$ { select unmarked x^i and all its neighbors; $s_i = \{x^l \text{neighbors of } x^i\}, s_i \neq \{\phi\}, s_i \subset V$; mark x^i ; if $s_i \notin \Gamma$ { distribute $K(\Omega_{s_i}^i)$ to members of s_i ; put s_i in Γ ; } repeat Step 2; } else END;

3.2 Double Authentication using new Key Distribution Scheme

We use the notation Ω to represent all neighbors of a router. Therefore subgroup Ω_j^i represents a set consisting of all neighbors of x^j except x^j , where i is one of j 's neighbor and i initiate the communication. $K(S_j^i)$ is the shared key for sub-group S_j^i and $K(\Omega_j^i)$ is the shared key for Ω_j^i . Note that $K(S_j^i) = K(S_i^j)$, but $K(\Omega_j^i) \neq K(\Omega_i^j)$.

The processing of N-DA is fundamentally same as O-DA. Each LSA is authenticated twice by a router using two sub-group keys discussed above. The first authentication code generated by key $K(S_j^i)$ is used

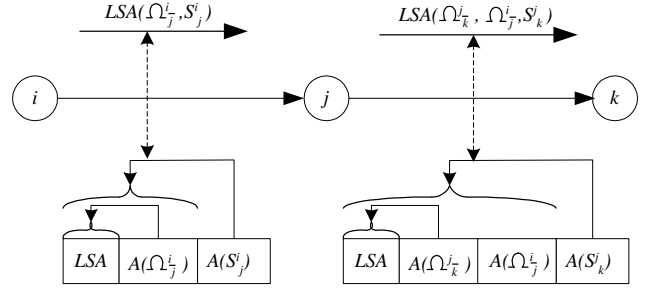


Figure 1: The DA example

by its neighbor to verify the routing information data. The processing is different from O-DA, since the second authentication code generated by key $K(\Omega_j^i)$ is only possessed by the neighbors of router x^j . This authentication code is used to guarantee that the node x^j does not alter the routing data. The sub-group keys $K(S_j^i)$ and $K(\Omega_j^i)$ are distributed in advance to each group member by means of offline or any secure channel from a KDC. It requires that the KDC knows the network topology in advance. We show the DA scheme as follows:

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 $LSA^i(\Omega_j^i, S_j^i)$.
2. When router x^j receives an LSA from its neighbor router x^i , it forwards it to its neighbor x^k as $LSA(\Omega_k^j, \Omega_j^i, S_k^j)$.

We present the working of DA with an example shown in Figure 1. The superscript on LSA identifies the router that floods the LSA. In rule 1, when router x^i creates the LSA, it needs to attach two authentication codes to it. We use Ω_j^i and S_j^i in the bracket to represent authentication codes generated by key $K(\Omega_j^i)$ and $K(S_j^i)$, respectively. Note that authentication code $A(S_j^i)$ should authenticate both the LSA and authentication code $A(\Omega_j^i)$. This can help detect if the LSA and/or $A(\Omega_j^i)$ have been altered. This is why the authentication code $A(S_j^i)$ is at the rightmost side in our representation. This is same for rule 2 where the last authentication code $A(S_k^j)$ covers previous two authentication codes and the LSA.

N-DA processing of intermediate nodes is presented as follows: When x^j receives an LSA, it first detects that the LSA is forwarded by its neighbor x^i . Then, router x^j uses sub-group key $K(S_j^i)$ to verify authentication code $A(S_j^i)$. Once authenticated, router x^j

uses sub-group keys $K(\Omega_k^j)$ and $K(S_k^j)$ to generate authentication codes $A(\Omega_k^j)$ and $A(S_k^j)$. Together with authentication code $A(\Omega_j^i)$ generated by x^i , following rule 2, three authentication codes are attached at the end of LSA that is forwarded to its neighbor x^k . Note that, as described above, authentication code $A(\Omega_j^i)$ is attached after $A(\Omega_k^j)$ as a part of the authenticated data. This can be used by x^k to verify that x^j has not altered the LSA. Authentication code $A(\Omega_j^i)$ needs to be attached after authentication code $A(\Omega_k^j)$. This is because when x^k forwards the LSA to the next hop, authentication code $A(\Omega_j^i)$ is not attached. So, authentication code $A(\Omega_k^j)$ should not authenticate $A(\Omega_j^i)$. However, these two authentication codes should be authenticated by code $A(S_k^j)$. In our scheme only the originator sends the LSA with two authentication codes. The intermediate routers generate two authentication codes but forward three authentication codes.

Finally, router x^j forwards the $LSA(\Omega_k^j, \Omega_j^i, S_k^j)$ to router x^k . When router x^k receives this LSA, it first verifies authentication code $A(S_k^j)$, and then checks code $A(\Omega_j^i)$. If authenticated, it just follows same steps as router x^j does in order to generate the new authentication codes.

4 Performance Assessment

In this section we discuss the storage requirement due to our proposed key management scheme. We then discuss the security robustness of the N-DA versus O-DA scheme.

4.1 Storage complexity

To evaluate storage complexity, we assume there are n routers within a link state routing domain. The OSPF standardization report [9] states that the maximum number of routers in a single OSPF area is about 350. We use $n = 350$ as the maximum number of routers within a single area. We also assume that the probability of having an edge between two routers is p .

Lemma 1 *In the new key distribution scheme, for a network with n nodes, if the probability of a direct link between any two nodes is p , then the average number of keys distributed to a node is $2(n - 1)p$.*

Proof 1 *The average number of keys distributed to a node includes two parts: (1) The total number of direct link from a node x to other nodes is: $\sum_{i=1}^{n-1} p = (n - 1)p$; (2) The total number of sub-groups including x but not the neighbors of x equals to the number of neighbors of x , which is $(n - 1)p$. Thus, the average number of keys for a node is therefore $2(n - 1)p$. \square*

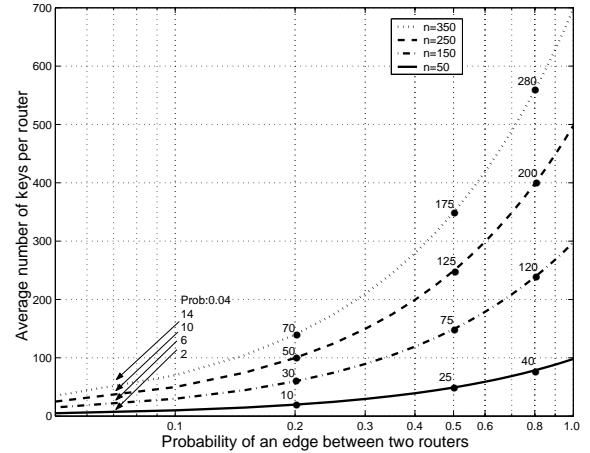


Figure 2: Number of keys needed to be stored by a router with increasing probability

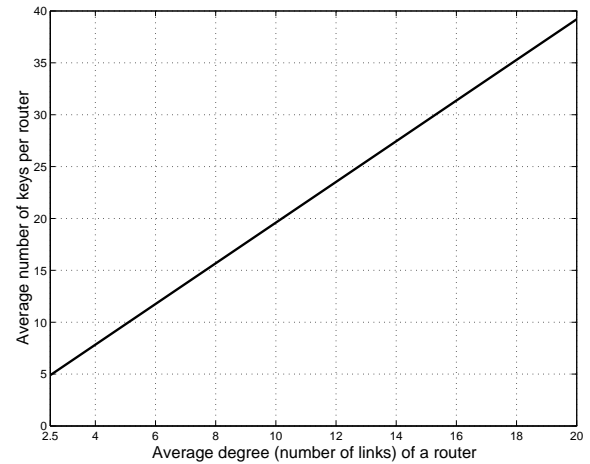


Figure 3: Number of keys needed to be stored by a router with increasing degree

Figure 2 shows the relation between the average number of keys possessed by a node and the probability of a link between two routers. The worst case is $p = 1$, which forms a fully connected network. Based on the Lemma 1, in the worst case, the number of keys possessed by each router is $2(n - 1)$, and the storage complexity is $O(n)$. np is the average degree (number of links connected to a router, which is represented by d) of a router. When $n = 350$, $p = 0.2$, then $d = 75$. Figure 3 shows the relation between average number of keys possessed by a router with the increased degree. We can see that for a very dense network, a router with average degree of 20, the average number of keys possessed is no more than 40. In Figure 4, if the n equals to 150, 250, and 350, and the corresponding p

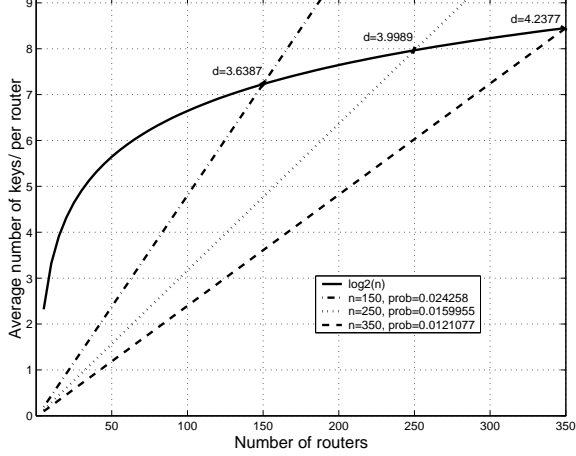


Figure 4: Number of keys needed to be stored by a router with increasing number of routers

is 0.024258, 0.0159955, and 0.0121077, the average degree of a router is 3.6387, 3.9989, and 4.2377, respectively. We note that the complexity of average number keys possessed by a router is restricted by $O(\log_2 n)$.

4.2 Security Robustness

The DA scheme is aimed to guard against impersonation attack. Here, we compare O-DA and N-DA in terms of resilience to impersonation attacks in the following scenarios:

1. There are one or multiple subverted routers and they do not collude.
2. There are multiple subverted routers colluding to impersonate other routers and generate forged routing information. The colluded routers do not partition the network.
3. There are multiple subverted routers. These routers collude and partition the network.

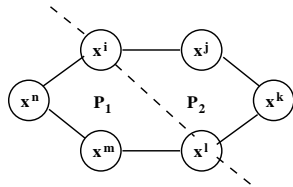


Figure 5: DA robustness analysis example

We analyze the security robustness based on these three scenarios.

First scenario: N-DA provides the same security robustness as O-DA. As shown in Figure 1, if subverted

router x^j compromises LSA received from router x^i , router x^k can immediately identify changes through authentication code $A(\Omega_j^i)$. Router x^j cannot deny it, since authentication code $A(\Omega_k^j)$ proves that router x^k is innocent and authentication code $A(S_k^j)$ proves that only router x^j or x^k can be the subverted router.

Second scenario: O-DA can provide helpful information for the intrusion detection system (IDS) to locate the attack source. A simple example is shown in Figure 5. We assume that the subverted routers are x^j and x^k ; they substitute/spoof the LSA from/of router x^n . As the flooding feature of link state routing protocol, router x^n will eventually receive the forged LSA from router x^m . Hence, router x^n will notice the inconsistency of routing information. Following the routing information propagation path, the authentication chain is set up as follows:

$$\begin{aligned}
 &LSA(V \setminus \{x^l\}, V \setminus \{x^k\}, S_l^k) \rightarrow \\
 &LSA(V \setminus \{x^m\}, V \setminus \{x^l\}, S_m^l) \rightarrow \\
 &LSA(V \setminus \{x^n\}, V \setminus \{x^m\}, S_n^m)
 \end{aligned}$$

These three pieces of routing information are presented by routers x^l , x^m , and x^n , which were received from their neighbors x^k , x^l and x^m , respectively. When x^l presents the evidence of $LSA(V \setminus \{x^l\}, V \setminus \{x^k\}, S_l^k)$, it can help IDS trace back to the source router x^k . As we discussed in the first scenario, router x^k cannot deny the generation of this forged routing information. However, using O-DA, the colluded routers can be any pair (for instance, can be x^j and x^l), which can confuse IDS in locating subverted routers. For the same example, if router x^i is subverted instead of router x^j , x^i can share the $K(V \setminus \{x^k\})$ with another subverted router x^k to forge the authentication code $A(V \setminus \{x^k\})$. In this case, IDS has no clue if router x^i is subverted or not.

In the N-DA scheme, each router only knows keys shared with its next-hop neighbor and keys shared with all the neighbors of its next-hop. Hence, in the above scenario, the authentication chain would be:

$$\begin{aligned}
 &LSA(\Omega_i^k, \Omega_j^i, S_i^k) \rightarrow LSA(\Omega_m^l, \Omega_l^k, S_m^l) \\
 &\rightarrow LSA(\Omega_n^m, \Omega_m^l, S_n^m)
 \end{aligned}$$

Here, only routers x^j and x^l know the shared key $K(\Omega_l^k)$, and no one else. This property restricts the subverted sources staying together, therefore making it easier to locate them.

Third scenario: Network partition leads to two scenarios. First, the partitioning routers do not let the routing information and traffic go through. Second,

the partitioning routers selectively block the routing information and the traffic.

For the first scenario, routers x^i and x^l collude to partition the network into two parts P_1 and P_2 (see Figure 5). If x^i and x^l stop forwarding traffic between these two parts, it is equivalent to the network congestion happening at the network points x^i and x^j . IDS can quickly locate where these problems occur. Thus, x^i and x^l can be easily identified.

For the second scenario, if x^i and x^l forward data traffic as normal and inject false routing information regarding to themselves, i.e., they may either increase or decrease their link metric values. In the first case, increasing metric value, such as delay, the network traffic congestion might occur between two partitions (assuming there are no alternate routes). In the second case, decreasing metric, such as delay, the traffic can be attracted to go through the partitioning routers. This might have a local impact when this partitioned network is sub-network of a comparatively large system.

The attacker's goal is to maximize the consequence of attacks, subverted routers can impersonate a set of routers within a partition and inject forged routing information to another partition. It can attract more traffic to go through partitioned routers.

The O-DA scheme totally fails in the third scenario. For example, if routers x^i and x^l impersonate x^n and x^m in P_1 and inject the forged routing information into P_2 , they can prevent the forged routing information from receiving by x^n and x^m . In case of N-DA, the attackers must be located as neighbors to deploy the attack. Thus, N-DA provides stronger protection than O-DA. An attacker requires not only partitioning of the network, but also that at least two of subverted routers have direct link between them in order to succeed. The reason is the same as present in the second scenario. For example, only x^i 's neighbors can collude with x^i to share the authentication key $K(\Omega_i^*)$.

Besides the three scenarios we had discussed above, N-DA eliminates more uncertainty as compared to O-DA. In O-DA, any combination of sub-group members could forge the routing information. It is very hard for IDS to locate the exact attack source. But in N-DA, only the routers having links between them can forge the routing information; this can quickly help IDS to identify the location of the attack sources.

5 Conclusion

In this paper, we proposed a new key distribution approach for the double authentication scheme to protect from impersonation attacks. This key distribution scheme brings two-fold benefits. First, it reduces the

storage requirement from *quadratic* to *linear* (in the worst case). In the network with average degree of 4, the complexity of number of keys possessed by a router is close to $O(\log_2 n)$. Second, the new key distribution scheme strengthens the robustness of DA scheme by forcing the subverted routers to stay together, which makes it easier in locating the attack source.

References

- [1] D. Huang, A. Sinha, and D. Medhi, "A double authentication scheme to detect impersonation attack in link state routing protocols," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2003, pp. 1723 – 1727.
- [2] S. Murphy, M. Badger, and B. Wellington, "Ospf with digital signatures," *RFC2154*, June 1997.
- [3] J. Moy, "OSPF version 2," *RFC2328*, April 1998.
- [4] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 2003.
- [5] S. Cheung, "An efficient message authentication scheme for link state routing," in *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, 1997, pp. 90–98.
- [6] R. Hauser, T. Przygienda, and G. Tsudik, "Lowering security overhead in link state routing," *Computer Networks*, vol. 31, no. 8, pp. 885–894, 1999.
- [7] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," *RFC2104*, February 1997.
- [8] D. Huang and D. Medhi, "A key-chain based keying scheme for many-to-many secure group communication," *ACM Transactions on Information and System Security*, vol. 7, no. 4, pp. 523 – 552, 2004.
- [9] J. Moy, "OSPF standardization report," *RFC2104*, April 1998.