LAS: An Effective Anti-Spoofing Method Using Existing Information

Baobao Zhang zbb@netarchlab.tsinghua.edu.cn
Jun Bi junbi@tsinghua.edu.cn
Jianping Wu jianping@cernet.edu.cn
1, Institute for Network Sciences and Cyberspace, Tsinghua University, 100084 Beijing, China
2, Department of Computer Science, Tsinghua University, 100084 Beijing, China
3, Tsinghua National Laboratory for Information Science and Technology (TNList) , 100084 Beijing, China

***Abstract*—In today's Internet, users can send packets with forged source IP addresses, called IP Spoofing, which causes many severe problems to the Internet, especially the security problem. Extensive methods have been proposed to prevent IP spoofing. One category of anti-spoofing methods is extra-information-based. The other category of anti-spoofing methods is existing-information-based. The existing-information-based anti-spoofing methods are much easier to deploy than the extra-information-based ones. Thus, we focus on the existing-information-based anti-spoofing methods in this paper. However, the existing existing-information-based anti-spoofing methods perform poorly on preventing IP spoofing. Therefore, in this paper we propose a new existing-information-based anti-spoofing method, named Link-state-based Anti-Spoofing (LAS), which works in a link-state-protocol-based network. LAS provides good effectiveness on preventing IP spoofing as our simulation results show that even if LAS is only deployed on 10% of routers in a link-state-protocol-based network, 80%~99% of spoofing cases in the network can be prevented in general.**

Keywords—Anti-spoofing; Source address validation; DDoS

I. INTRODUCTION

A. The urgency and necessity of preventing IP spoofing

In today's Internet, users can send packets with forged source IP addresses, called IP Spoofing, which causes many severe problems to the Internet, especially the security problem. For example, a NANOG report [1] suggests that IP spoofing is very prevalent in DDoS (Distributed Denial of Service) attacks. It is very urgent to prevent IP spoofing. Furthermore, in the long term, it will be of great significance for the credibility and accountability of the Internet if the truth of the source address can be assured. Thus, it is very necessary and critical for the Internet to prevent IP spoofing.

B. Why is LAS needed?

Extensive methods have been proposed to prevent IP spoofing. One category of anti-spoofing methods is extra-information-based. The extra-information-based anti-spoofing methods prevent IP spoofing through adding extra information to the network. Marking packets and communicating packets are two means of adding extra information. SPM [2], BASE [3], Packet Passport [4] are representative marking-packets-based methods, which embed the signature information into data packets for validating source addresses, leading to expensive authentication cost and computational cost. SAVE [5] is the representative communicating-packets-based method, which makes routers build the mapping table from the source IP addresses to the incoming interfaces by communicating packets, leading to extra communication cost. The other category of anti-spoofing methods is existing-information-based. The existing-information-based anti-spoofing methods prevent IP spoofing by only using the existing information. The existing-information-based anti-spoofing methods need much lower cost than the extra-information-based ones, so the existing-information-based anti-spoofing methods are much easier to deploy than the extra-information-based ones. Thus, we focus on the category of existing-information-based anti-spoofing methods in this paper. However, the existing existing-information-based anti-spoofing methods, Ingress Filtering [6], uRPF [7] and HCF [8], cannot provide good effectiveness on preventing IP spoofing as follows. Ingress Filtering can only filter the spoofing traffic originated from the local router, so the effectiveness of Ingress Filtering on filtering spoofing traffic is not very good. HCF and uRPF may filter legal traffic, which is absolutely intolerable for network operators. SAVO [14] is also an existing-information-based method. The computational complexity of the core algorithm of SAVO is $O(n^3)$, where n is the number of routers in an intra-area network, and is not salable. Therefore, in this paper, we propose a new existing-information-based anti-spoofing method, named Link-state-based Anti-Spoofing (LAS). LAS not only has good effectiveness on filtering spoofing traffic, but also will not filter any legal traffic. Our simulations show that if LAS is only deployed on 10% of routers in a link-state-protocol-based network, 80%~99% of spoofing cases in the network can be prevented by LAS in general. In addition, the computational complexity of the core algorithm of LAS is only $O(n^2)$, which is much lower than that of SAVO.

C. Problem statement

The core task of LAS is to calculate all the valid incoming interfaces for prefixes by fully using the existing LSDB (Link State Data Base) information in the link-state-based-protocol router. An LAS-enabled router filters a packet if it arrives from an invalid interface. The LSDB in a router contains the complete view of the topology information of the routing area where the router is located, so the valid incoming interfaces of the prefixes in the routing area can be accurately computed. However, the difficulty is that how to calculate the valid incoming interfaces for the prefixes outside the routing area under the condition of no complete topology information outside the routing area. The key idea of our solution to that is to calculate the loose valid incoming interfaces instead of the strict ones for the prefixes outside the routing area. Although the loose incoming interfaces may lead to failing to filter a proportion of IP spoofing traffic, they can avoid filtering any legal traffic because it is absolutely intolerable for network operators to filter any legal traffic.

D. Contritutions, limitations and the structure of this paper

The contributions of this paper are three-fold as follows. (a) We design a new existing-information-based anti-spoofing method, named LAS. LAS not only has good effectiveness on filtering spoofing traffic, but also will not filter any legal traffic. (b) We design an optimal strategy of deploying LAS for small-size networks and design a heuristic strategy of deploying LAS for large-size networks under partial deployment. (c) We comprehensively evaluate the effectiveness of LAS on filtering spoofing traffic. In addition, we evaluate the impact of the loose incoming interfaces of LAS on failing to filter spoofing traffic.

The limitation of this paper is that LAS can only work in the pure link-state-protocol-based network, where LAS has considerably good effectiveness on preventing IP spoofing with very low cost.

The structure of the rest of this paper is organized as follows. In Section II, we will introduce the related work. In Section III, we will describe the detailed mechanism of LAS. In section IV, we will describe the strategies of deploying LAS under partial deployment. In Section V, we will comprehensively evaluate LAS on preventing IP spoofing. In Section VI, we conclude this paper and present our future work.

II. Related work

In this paper, we focus on the category of existing-information-based anti-spoofing methods. Ingress Filtering [6], uRPF [7] and HCF [8] are the existing existing-information-based anti-spoofing methods. We describe them as follows.

Ingress Filtering prevents IP spoofing by checking whether the source addresses of packets are from correct ingress ports. Ingress Filtering can only filter IP spoofing traffic originated from local routers, but cannot filter IP spoofing traffic originated from other remote routers, i.e., Ingress Filtering can only prevent the equal proportion of spoofing cases to the proportion of ingress-filtering-enabled routers. Thus, the effectiveness of Ingress Filtering on filtering spoofing traffic is not very good.

uRPF regards the outgoing interfaces as the incoming interfaces for each prefix, so uRPF prevents IP spoofing based on checking incoming interfaces of packets. The incoming interfaces of a prefix may be different from its outgoing interfaces in the asymmetric routing environments, leading to filtering legal traffic, which is absolutely intolerable for network operators. The paper [9] shows that 50% of paths are asymmetric in the Internet.

HCF [8] prevents IP spoofing by building a mapping table from source IP addresses to the hop-count values, i.e., the IP2HC mapping table, which is referred from historical traces. HCF determines whether a packet is spoofed through checking whether the hop count of the packet is consistent with the one the source address of the packet is mapped to in the IP2HC mapping table. However, the referred IP2HC mapping table is very error-prone because the hop count values of not all the source IP addresses can be accurately learned and the IP2HC mapping table may be polluted by the spoofing traffic, etc. Thus, HCF may filter legal traffic. To avoid filtering legal traffic, both uRPF and HCF can only be used in the alarming way instead of the filtering way.

Besides the methodological kind of related work above, there is another kind of architectural work, which is the source address validation architecture (SAVA). SAVA was put forward by our research team in the paper [10] at the first time. Since the year of 2008, SAVA has become a request for comments (RFC 5210) [11] in the Internet Engineering Task Force (IETF). According to SAVA, three levels of source address validation are ought to be needed: the local subnet level, the intra-AS (intra-Autonomous System) level and the inter-AS level. Under the guidance of SAVA, many methods have been devised for the three levels of source address validation. For example, SAVI [12] was designed for the local subnet level of source address validation. SMA [13] was designed for the inter-AS level of source address validation. SAVO [14], an existing-information-based method, was put forward for Intra-AS level of source address validation. In addition, LAS in this paper and Ingress Filtering are also used for the intra-AS level of source address validation. Methods in [2][3][4] are used for the inter-AS level of source address validation. SAVE, HCF and uRPF can be used for both the intra-AS level and inter-AS level of source address validation, but SAVE needs extra communication cost; HCF and uRPF may filter legal traffic. Ingress Filtering will not filter any legal traffic. Our simulation results show that LAS performs much better than Ingress Filtering on preventing IP spoofing. The idea of SAVO is also to calculate valid incoming interfaces for prefixes based on LSDB, but the computational complexity of the core algorithm of SAVO is (n^3) while the computational complexity of the core algorithm of LAS is only $O(n^2)$, where n is the number of routers in an intra-area network.

III. Methodology: LAS

The core task of LAS is to calculate the valid incoming interfaces for prefixes by fully using the existing LSDB (Link State Data Base) information. The LAS-enabled router filters a packet if it arrives from an invalid interface. In this section, we will describe that how LAS generates such an incoming table for prefixes on an OSPF [15] router. OSPF is a link state IGP (Interior Gateway Protocol) protocol, which is widely used in the Internet. In the future, we will expand LAS to support the IS-IS protocol [16], which is another link state IGP protocol. In Subsection A, we give the reverse Dijkstra algorithm. This algorithm is used to calculate the valid incoming ports of the other nodes arriving at a given node on a directed graph. In Subsection B, we will analyze their feasible incoming interfaces for different types of prefixes. In Subsection C, we will describe the detailed LAS algorithm of generating the incoming table. In Subsection D, we will analyze the cost of LAS.

A. The foundation algorithm: the reverse Dijkstra algorithm

In this section, we will introduce how to calculate the valid incoming ports of the other nodes arriving at a given node on a directed graph, which is obtained from the abstract of an intra-area topology in an OSPF network. As is well known, the OSPF protocol uses the shortest path routing. If a source node has two or more equal-cost shortest paths towards a destination node, the source node may use all or any one of these shortest paths towards the destination node. To avoid

filtering legal traffic, if a source node has two or more equal-cost shortest paths towards a destination node, we regard all the incoming ports through these equal-cost shortest paths as the valid incoming ports of the source node to the destination node, which may fail to prevent a proportion of spoofing cases, which is very low as evaluated in Section V. As is well known, the OSPF protocol uses the Dijkstra algorithm [17] to calculate all the least-cost outgoing ports of a given node towards the other nodes. Given a directed graph G and a node A, the reverse Dijkstra algorithm calculates the valid incoming ports of the other nodes arriving at A as follows.

Step 1: Obtain the reverse graph G' of G by reversing the direction of each edge in G.

Step 2: On the reverse graph G', the Dijkstra algorithm is run on Node A to calculate all the least-cost outgoing ports of Node A towards the other nodes. These calculated outgoing ports of Node A towards the other nodes are all the least-cost incoming ports of the other nodes arriving at A, which we will prove in Theorem 1. The least-cost incoming ports of the other nodes arriving at A are regarded as the valid incoming ports of the other nodes arriving at A.

The reverse Dijkstra algorithm has the same computational complexity with the Dijkstra algorithm, which is O(n^2), where n is the number of nodes on the graph.

THEOREM 1. *Given a directed graph G, a node A and one any other node B, we suppose G' is the reverse graph of G. We suppose OGP is the set of all the outgoing ports of A towards B through all the equal-cost shortest paths on G'. We suppose ICP is the set of all the incoming ports of B arriving at A through all the equal-cost shortest paths on G. OGP must be equal to ICP.*

Proof. We first prove that $ICP \subseteq OGP$. $\forall\, i \in ICP$, we suppose that one any shortest path associated with i from B to A on G is p. We obtain p' by reversing the direction of each edge on p. Obviously, p' is path on G'. In addition, p' must be a shortest path from A to B on G'. Otherwise, there must another path s' from A to B on G' that is shorter than p'. We obtain s by reversing the direction of each edge on s'. Obviously, s is a path on G and s is shorter than p from B to A on G, which contradicts with that p is a shortest path from B to A. Therefore, p' must be a shortest path from A to B on G'. Thus, $i \in OGP$. Similarly, we can prove that $OGP \subseteq ICP$. Therefore, $OGP = ICP$.

We now give an example for the reverse Dijkstra algorithm. Fig. 1 is a directed graph G composed of a set of nodes and a set of directed links. The number on each link denotes the weight of the link. We now show how the reverse Dijkstra algorithm calculates the valid incoming ports of the other nodes arriving at a given node A. Firstly, the reverse graph G' of Graph G is obtained as shown in Fig. 2. Secondly, the Dijkstra algorithm is run on the reverse graph G' to calculate all the least-cost outgoing ports of Node A towards the other nodes. The shortest path tree from Node A to the other nodes is shown in Fig. 3. The outgoing ports calculated by the Dijkstra algorithm on G' are shown in Table I. According to Theorem 1, these calculated outgoing ports of Node A towards the other nodes are all the valid incoming ports of the other nodes arriving at A. We now use an example to verify Theorem 1. Fig. 4 is the shortest path tree towards Node A.

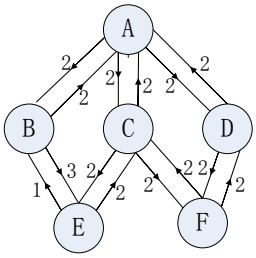

Fig. 1. Original graph

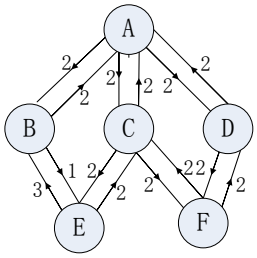

Fig. 2. Reverse graph

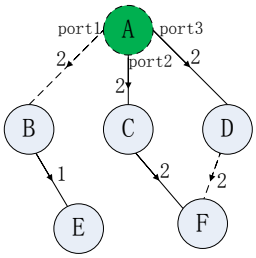

Fig. 3. Shortest path tree from A on the reverse graph

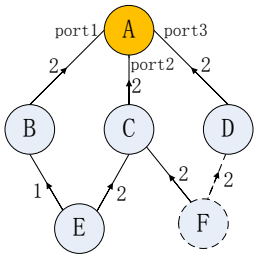

Fig.4. Shortest path tree towards A on the original graph

TABLE I. THE OUTGOING PORTS OF NODE A TOWARDS OTHER NODES ON THE REVERSE GRAPH

Destination	Outgoing ports
B	port1
C	port2
D	port3
E	port1
F	port2, port3

TABLE II. THE INCOMING PORTS OF THE OTHER NODES ARRIVING AT A ON THE ORIGINAL GRAPH

Source	Incoming ports
B	port1
C	port2
D	port3
E	port1
F	port2, port3

We then obtain all the valid incoming ports of the other nodes arriving at A as shown in Table II. We find that the outgoing ports of Node A towards the other nodes on the reverse graph $G'$are the incoming ports of the other nodes arriving at Node A on the original graph G. Theorem 1 is verified.

B. Analysis for feasible incoming interfaces of different types of prefixes

In this section, we will analyze their feasible incoming interfaces for different types of prefixes. Before that, we first give some explanations for some special routers and the different types of prefixes as follows.

ABR (Area-Border Router) refers to an OSPF router that connects to other areas in the OSPF network. An ABR has the complete view of the topology information of the areas where it is located.

ASBR (AS-Boundary Router) refers to an OSPF router that connects to other networks.

Intra-area prefixes with respect to an OSPF router refer to prefixes originated from the areas where the OSPF router is located. Intra-area prefixes are extracted from router-LSAs and network-LSAs for the OSPF protocol.

Inter-area prefix with respect to an OSPF router refer to prefixes originated from the other areas in the link-state-protocol-based network, i.e., the areas where the router is not

located. Inter-area prefixes are extracted from type3-summary-LSAs for the OSPF protocol.

External-AS prefixes with respect to an OSPF router refer to prefixes originated from the outside of the link-state-protocol-based network where the router is located. External-AS prefixes are extracted from external-AS-LSAs for the OSPF protocol.

An OSPF router has the complete view of the topology of the areas where the router is located. Thus, on an OSPF router, the incoming interfaces of the intra-area prefixes with respect to the OSPF router can be accurately computed using the reverse Dijkstra algorithm.

However, for the inter-area prefixes with respect to an OSPF router, the OSPF router only knows the unidirectional cost from the corresponding ABRs to these inter-area prefixes, which are used to calculate the outgoing interfaces towards these inter-area prefixes, but the OSPF router does not know the reverse cost, i.e. the cost from these inter-area prefixes to ABRs. Thus, on an OSPF router, the incoming interfaces of the inter-area prefixes with respect to the OSPF router cannot be accurately computed. Fortunately, it is certain that each inter-area prefix with respect to an OSPF router must traverse one of the ABRs in the areas where the OSPF router is located to reach the OSPF router though it is uncertain that the inter-area prefix traverses which ABR. To avoid filtering legal traffic, we use loose incoming interfaces for inter-area prefixes as follows. On an OSPF router, for the inter-area prefixes with respect to the OSPF router, we regard the incoming interfaces of all the ABRs in the areas where the OSPF router is located as the ones of these inter-area prefixes.

Similarly, for the external-AS prefixes with respect to an OSPF router, the incoming interfaces of these external-AS prefixes cannot be accurately computed. It is certain that an external-AS prefix with respect to an OSPF router must traverse one of the ASBRs and ABRs in the areas where the OSPF router is located to reach the OSPF router. To avoid filtering legal traffic, we also use loose incoming interfaces for external-AS prefixes as follows. On an OSPF router, for the external-AS prefixes with respect to the OSPF router, we regard the incoming interfaces of all the ASBRs and ABRs in the areas where the OSPF router is located as the ones of these external-AS prefixes.

On an OSPF router, the incoming interfaces of the ASBRs and ABRs in the areas where the OSPF router is located can be accurately computed using the reverse Dijkstra algorithm.

As shown in Fig. 5, the OSPF network is composed of three areas. Area 0 is the backbone area. Router A1 and A2 are ABRs that are located in Area 0 and Area 1. Router A4 is an ASBR that is located in Area 1. Router C2 is another ASBR that is located in Area 0. Router A3, B2 and C1 are general routers. With respect to Router A3, N0 and N1 are intra-area prefixes, N2 is an inter-area prefix and N3 is an external-AS prefix. The number at the near end of each edge denotes the weight to the neighbor. The topology view of Router A3 is shown in Fig. 6. On Router A3, the incoming interfaces of N0 and N1 can be accurately computed based on the complete topology knowledge, but the ones of N2 and N3 cannot be accurately computed because of the incomplete topology knowledge. The inter-area prefix N2 must traverse one of the

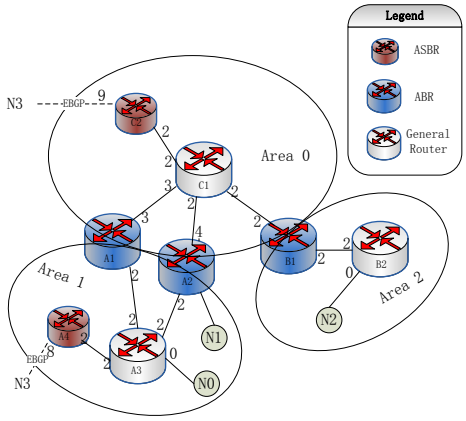

Fig. 5. An OSPF network topology with three areas

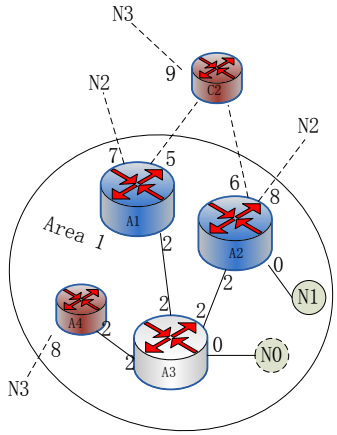

Fig. 6. The topology view of Router A3

ABRs to Router A3. The external-AS prefix N3 must traverse one of the ASBRs and ABRs to Router A3.

C. The LAS algorithm of computing the incoming table

In this subsection, we will describe the detailed LAS algorithm of computing the valid incoming interfaces for all the prefixes arriving at a given OSPF router. There are three steps as follows. In the following three steps, the incoming interfaces refer to the incoming interfaces arriving at the given OSPF router unless otherwise stated.

Step 1: Identifying all the ABRs and ASBRs in the areas where the OSPF router is located. The OSPF router will receive the router-LSAs from all the other intra-area routers. There are two special bits, Bit B and Bit E, in the router-LSA. If bit B=1, then it indicates that the corresponding router is an ABR, otherwise, the corresponding router is not an ABR. Similarly, if bit E =1, it indicates that the corresponding router is an ASBR, otherwise, the corresponding router is not an ASBR.

Step 2: Calculating the incoming interfaces of the intra-area prefixes, ABRs and ASBRs arriving at the OSPF router. Because an OSPF router has the complete view of the topology where the OSPF router is located, the incoming interfaces of the intra-area prefixes, ABRs and ASBRs can be computed using the reverse Dijkstra algorithm.

TABLE III. THE INCOMING TABLE ON ROUTER A3

Prefix	Incoming interfaces
N0	N0
N1	A2
N2	A1,A2
0.0.0.0/0	A4,A1,A2

Step 3: Calculating the incoming interfaces of inter-area and external-AS prefixes arriving at the OSPF router. The incoming interfaces of each inter-area prefix arriving at the OSPF router are the union incoming interfaces of the ABRs in the intra areas where the OSPF router is located. Similarly, the incoming interfaces of each external-AS prefix arriving at the OSPF router are the union incoming interfaces of the ASBRs and ABRs in the intra areas where the OSPF router is located. Because the valid incoming interfaces of each external-AS prefix are the same, all the external-AS prefixes can be aggregated into a default prefix 0.0.0.0/0 in order to reduce the size of the incoming table.

The loose incoming interfaces of external-AS prefixes and inter-area prefixes will lead to failing to filter a proportion of spoofing traffic, which depends on the proportion of ABRs and ASBRs as evaluated in Section V.

The incoming table generated by LAS on Router A3 in Fig. 5 is shown in Table III. We use the corresponding neighbors to denote the corresponding incoming interfaces for simplicity.

D. *Cost analysis for LAS*

We suppose that the number of intra-area routers is n, the number of intra-area prefixes is m and the number of inter-area prefixes is k with respect an OSPF router. The computational complexity of the first step of LAS is O(n). The computational complexity of the second step of LAS is O(n*n+m). The computational complexity of the third step of LAS is O(k). Therefore, the total computational complexity of LAS is O(n*n+m+k). The size of the incoming table is m+k+1, i.e., the number of the prefixes in the OSPF network plus the default prefix 0.0.0.0/0. Therefore, LAS has good scalability in the aspects of the algorithm and the incoming table.

IV. DEPLOYMENT STRATEGIES

In this section, we study strategies of deploying LAS under the graph model. The strategies will be used in the next section. We suppose that the number of nodes on the graph is n. There are n*n*(n-1) spoofing cases in total because the source node in each source-destination pair among all the n*n source-destination pairs can forge any other nodes except for itself. Given the number of deploying nodes k, the task of the LAS deployment strategy is to choose which k nodes should deploy LAS so as to maximize the prevented spoofing cases. In Subsection A, we design an optimal strategy of deploying LAS. This optimal strategy is only fit for small-size networks. With respect to large networks, we design a heuristic strategy of deploying LAS in Subsection B. In Subsection C, we compare the heuristic deployment strategy with the optimal deployment strategy. Our results show that the heuristic deployment strategy performs very close to the optimal one.

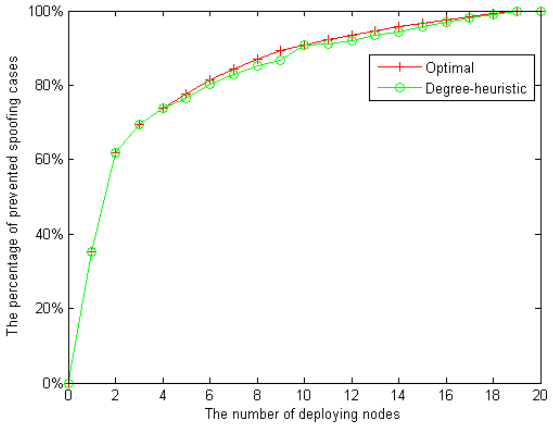

Fig. 7. The effectiveness comparison between the optimal strategy and the degree-heuristic strategy

A. *The optimal strategy of deploying LAS*

The optimal strategy of deploying LAS should make the maximal spoofing cases prevented. The optimal strategy of deploying LAS generates the optimal placement for the k deploying nodes as follows. Firstly, all combinations of placements of deploying k nodes are enumerated. There are C(n, k) combinations in total. The placement that can prevent the maximal spoofing cases among all the combinations of placements is chosen as the optimal placement. The computational complexity of the optimal deployment strategy is exponential order of n, where n is the number of nodes in the network. If n is small, the optimal deployment strategy can compute the optimal placement within acceptable time, but if n is large, the optimal placement cannot be computed within acceptable time. Therefore, we design a heuristic deployment strategy in Subsection B.

B. *The heuristic strategy of deploying LAS*

Intuitively, a node with a high degree has two strengths on filtering spoofing traffic as follows: (a) the node has strong ability of identifying spoofing packets due to many incoming interfaces; (b) there will be large number of flows of traversing the node. Thus, the idea of our heuristic strategy is to prefer the node with the higher degree. The degree-heuristic deployment strategy for LAS works as follows: (a) sort all the n nodes with the decreasing order according to the degree; (2) The top k nodes are chosen as the deploying nodes.

C. *The comparison between the optimal deployment strategy and the heuristic deployment strategy*

We use our campus core network topology to evaluate the two strategies. There are 20 routers and 60 links on the campus topology. We suppose that all the links have equal weights. We increase the number of deploying nodes from 1 to 20 by one. We run the optimal deployment strategy and the heuristic strategy on the campus topology and obverse the percentage of spoofing cases that can be prevented. As shown in Fig. 7, the horizontal axis denotes the number of deploying nodes and the vertical axis denotes the percentage of spoofing cases that can be prevented. We find that the heuristic deployment strategy performs very close to the optimal one.

TABLE IV. THE INFORMATION OF THE ROUTER-LEVEL TOPOLOGIES

The AS where the topology belongs	The number of routers	The number of links	The number of ASBRs
AS 1221	2515	6078	229
AS 1239	7303	19772	840
AS 1755	295	1086	148
AS 2914	4607	15108	1253
AS 3257	411	1306	99
AS 3356	1620	13480	356
AS 3967	353	1640	220
AS 4755	41	136	29
AS 7018	9418	23340	788

TABLE V. THE SIX SITUATIONS OF IP SPOOFING

ID	The location of real source	The location of destination	The location of forged source	The number of spoofing cases
1th	Intra-domain	Intra-domain	Intra-domain	n*n*(n-1)
2th	Intra-domain	Intra-domain	External-AS	n*n*m
3th	Intra-domain	External-AS	Intra-domain	n*m*(n-1)
4th	Intra-domain	External-AS	External-AS	n*m*m
5th	External-AS	Intra-domain	Intra-domain	m*n*n
6th	External-AS	Intra-domain	External-AS	m*n*m

V. EVALUATIONS

In this section, we will evaluate LAS. In Subsection B we comprehensively evaluate the effectiveness of LAS on preventing IP spoofing. In addition, we will compare LAS with Ingress Filtering in Subsection B. In Subsection C, we evaluate the impact of loose incoming interfaces of external-AS prefixes on prevent IP spoofing. In Subsection D, we will evaluate the impact of loose equal-cost incoming interfaces on prevent IP spoofing. In Subsection A, we introduce how we collected data and describe the spoofing cases for evaluations. Now we define two terms that are often used in this section as follows.

The true positive ratio refers to the ratio of spoofing cases that are prevented to the total spoofing cases.

The false negative ratio refers to the ratio of spoofing cases that cannot be prevented to the total spoofing cases.

A. Datasets

We collected the topologies marked with 'ISP map' from the project of rocketfuel [18]. Rocketfuel obtains the ISP-map topologies using traceroute and alias resolution. We regard the routers that connect to the external AS as ASBRs. The summary information of the nine obtained router-level topologies is shown in Table IV. Each of the router-level ISP topologies is named with the corresponding AS number. Table IV shows the number of routers, links and ASBRs of each topology. However, the topologies only have router-level adjacencies, but have no other information that we need, such as weights, the protocols and so on. We assume that routers on each topology run the OSPF protocol with only one single area. All the bidirectional weights are 1. One spoofing case is represented by a triple *<real source, destination, forged source>*, which means that the '*real source*' forges the '*forged source*' to send packets to the '*destination*'. According to the assumption that there is only one OSPF area, there are no ABRs, so the intra-domain is equivalent to the OSPF intra area in our evaluation scenarios. Because the address space in the intra domain and that in the external AS is not an order of magnitude, we classify all the spoofing cases in an intra-domain network into six situations as shown in Table V for evaluation fairness. For the first situation, it means cases where intra-domain hosts forge intra-domain addresses to send packets to intra-domain hosts. The meanings of the other situations are similar. We suppose that the number of the intra-domain routers is n and the number of ASBRs is m. We divide the external-AS nodes into m equivalent classes as follows. We regard the external-AS nodes that traverse an ASBR to reach the intra domain as the equivalent class of the ASBR. The fifth column of Table V shows the number of spoofing cases in each situation. The number of spoofing cases in the first situation is n*n*(n-1) because there are n*n source-destination pairs where each source can forge the addresses of the other n-1 routers. The ones in the other situations are similar. For simplicity, we make the following ideal assumption: (a) the address space for each node is identical; (b) the number of nodes in each equivalent class is identical. According to the two assumptions, each case in the same situation has identical impact factor on IP spoofing.

For the sixth situation, it means cases where external-AS hosts forge external-AS addresses to send packets to intra-domain hosts. Obviously, neither LAS nor Ingress Filtering has the ability of preventing spoofing cases in the sixth situation because spoofing cases in the sixth situation are the inter-AS level of IP spoofing. According to the SAVA architecture, LAS and Ingress Filtering are the intra-AS level of source address validation, and the spoofing cases in the sixth situation should be prevented using the inter-AS level of source address validation methods. Therefore, we only evaluate the top five situations for LAS and Ingress Filtering.

B. The true positive ratio of LAS (reflecting the ability of filtering spoofing traffic)

In this section, we will evaluate the true positive ratio of LAS under partial deployment, which reflects the ability of filtering spoofing traffic. Because of the large number of nodes, we have to use the degree-heuristic deployment strategy. We evaluate the true positive ratios of LAS in terms of the top five situations of spoofing cases in Table V. The results for the first, the third and the fifth situations are similar. We only show the results for the first situation as shown in Fig. 8 due to the space restriction. The results for the second and the fourth situations are similar. We only show the results for the second situation as shown in Fig. 9 due to the space restriction. In Fig. 8 and Fig. 9, the horizontal axis denotes the deployment ratio and the vertical axis denotes the true positive ratio of LAS. We have some findings as follows.

Results for the first, the third and the fifth situations. If LAS is only deployed on 10% of routers, 80%~99% of spoofing cases in the first, the third and the fifth situations can be prevented. It implies that the addresses in the area where LAS is deployed are hard to be forged.

Results for the second and the fourth situations. There are six topologies, in which 80%~99% of spoofing cases in the second and the fourth situations can be prevented if LAS is deployed on 10% of routers. For the other three topologies,

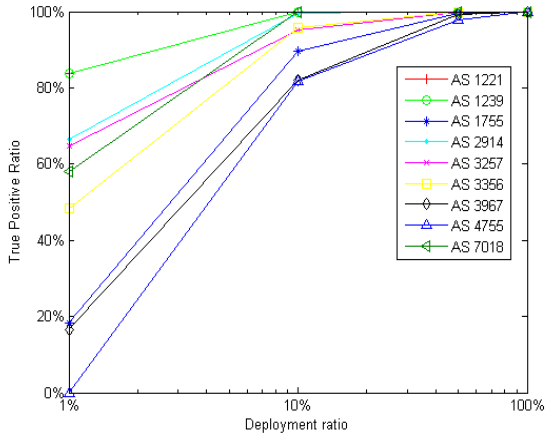

Fig. 8. The true positive ratios under different deployment ratios for the 1th situation. The results for the 3th and 5th situations are similar.

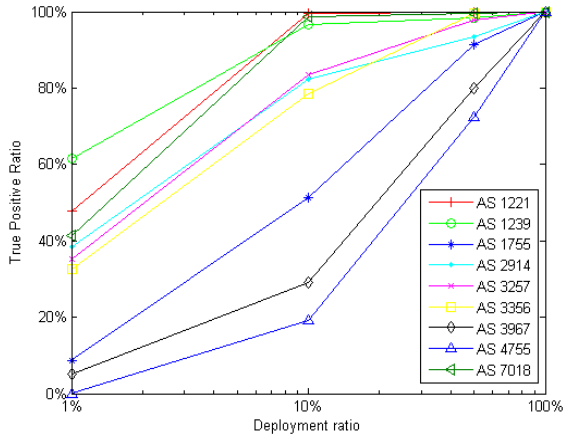

Fig. 9. The true positive ratios under different deployment ratios for the 2th situation. The results for the 4th are similar.

only 10%~50% of spoofing cases in the second and the fourth situations can be prevented when LAS is deployed on 10% of routers. Why does the effectiveness of LAS on preventing spoofing cases in the second and the fourth situations have big difference across different topologies? We speculate that the difference may be related with the proportion of ASBRs and ABRs because we regard all the incoming interfaces of ASBRs and ABRs as the legal incoming interfaces of external-AS prefixes. We will verify our speculation in the next subsection.

Results of comparison between LAS and Ingress Filtering. Ingress Filtering can only prevent spoofing cases originated from the local router, so Ingress Filtering cannot prevent spoofing cases in the fifth situation while LAS can. In addition, For each situation of the top four situations, the ratio of spoofing cases that Ingress Filtering can prevent to the total spoofing cases is equal to the deployment ratio of Ingress Filtering while the ratio of spoofing cases that LAS can prevent to the total spoofing cases is much higher the deployment ratio of LAS as shown in Fig. 8 and Fig. 9. Therefore, LAS performs much better than Ingress Filtering on preventing IP spoofing.

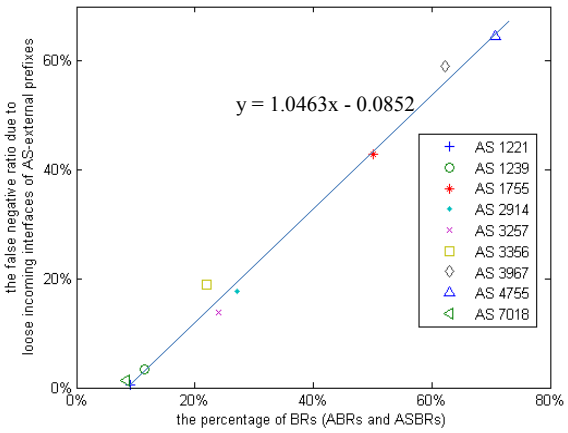

Fig. 10. The relation between the false negative ratio due to loose incoming interfaces of external-AS prefixes and the proportion of ABRs and ASBRs for the 2th situation. The results for the 4th situation are similar.

C. *The false negative ratio due to loose incoming interfaces of external-AS prefixes*

To avoid filtering legal traffic, LAS regards all the incoming interfaces of ASBRs and ABRs as the legal incoming interfaces of the external-AS prefixes, leading to failing to prevent a proportion of spoofing cases, which we call the false negative ratio due to loose incoming interfaces of external-AS prefixes. Specifically, the false negative ratio due to loose incoming interfaces of external-AS prefixes is: the true positive ratio if the accurate incoming interfaces of external-AS prefixes are known minus the true positive ratio under the loose incoming Interfaces of external-AS prefixes. The loose incoming interfaces of external-AS prefixes will not affect spoofing cases in the 1th , 3th and 5th situations to be prevented because these spoofing cases do not forge the external-AS addresses. The results of the false negative ratios due to loose incoming interfaces of external-AS prefixes for the 2th situation are shown in Fig. 10. The results for the 4th situation are similar. We omit them due to space restriction. Fig. 10 is a scatter diagram. Each point in Fig. 10 denotes the result of the corresponding topology. The x-coordinate of each point denotes the ratio of ABRs and ASBRs to the total routers on the topology that the point is corresponding to. The y-coordinate of each point denotes the false negative ratio due to loose incoming interfaces of external-AS prefixes when the deployment ratio is 10% on the topology that the point is corresponding to. We draw a trend line, which is y = 1.0463x - 0.0852, for those points as shown in Fig. 10. We find that the trend line is very close to the linear relation y=x, i.e. the false negative ratio due to loose incoming interfaces of external-AS prefixes under the deployment ratio of 10% is approximately equal to the ratio of ABRs and ASBRs to the total routers. The reason why the effectiveness of LAS on preventing spoofing cases in the second and the fourth situations have big difference across different topologies is obviously clear as shown in Fig. 10. The proportion of ASBRs on the three topologies where LAS performs poorly is high to 50%~70%, leading to 50%~70% of false negative ratio while the proportion of ASBRs on the other six topologies where LAS

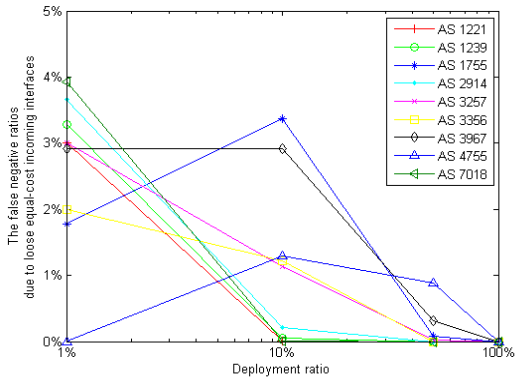

Fig. 11. The false negative ratios due to loose equal-cost incoming interfaces

performs well is only 8%~25%, only leading to 8%~25% of false negative ratio. The reason why the proportions of ASBRs on the three topologies where LAS performs poorly are high is because the three topologies are maybe not so complete. As shown in Table IV, each of these three topologies has less than 300 nodes while each of the other six topologies has more than 1000 nodes. Therefore, if the ratio of ASBRs and ABRs to the total routers in a network is lower than 20%, which holds in general, LAS can also prevent 80%~99% of spoofing cases in the second and the fourth situations.

In addition, the relation between the false negative ratio due to loose incoming interfaces of inter-area prefixes and the percentage of ABRs is likely similar although we have no data of ABRs to verify that. In the future, we will build a model to explain that interesting relation in theory.

D. The false negative ratio due to the loose equal-cost incoming interfaces

When there are multiple equal-cost incoming interfaces for a source node, LAS does not accurately know which will be used by the source node. To avoid the false positive, LAS considers all the equal-cost incoming interfaces legal for the source node. This may bring about false negatives. In this subsection, we will evaluate the false negative ratio due to loose equal-cost incoming interfaces. The false negative ratio due to loose equal-cost incoming interfaces is: the true positive ratio under the situation of knowing which equal-cost incoming interface(s) will be used minus the true positive ratio under the situation of regarding all the equal-cost incoming interfaces as valid incoming interfaces. We suppose that the source node arbitrarily selects one shortest path from all the equal-cost shortest paths for routing. Fig. 11 shows the false negative ratios due to the loose equal-cost incoming interfaces on the nine topologies under different deployment ratios. The horizontal axis denotes the deployment ratio of LAS. The vertical axis denotes the false negative ratio due to the loose equal-cost incoming interfaces. We find that the false negative ratios due to the loose equal-cost incoming interfaces are less than 4%, which is very low.

VI. CONCLUSION AND FUTURE WORK

In this paper, we design a new existing-information-based anti-spoofing method, named LAS. LAS has good effectiveness on preventing IP spoofing as the simulation results show. If the ratio of ASBRs and ABRs to the total routers in a network is lower than 20%, 80%~99% of spoofing cases can be prevented by LAS only with the deployment ratio of 10%. In addition, LAS will not filter any legal traffic. The algorithm of LAS and the incoming table generated by LAS have good scalability. In the pure link-state-protocol-based network, LAS is the best choice among all the existing anti-spoofing methods.

OSPF and IS-IS are two widely used link-state-based protocols. In this paper, we only consider the OSPF situation. In the future, we will expand LAS to support the IS-IS protocol.

ACKNOWLEDGEMENT

This research is supported by Supported by the National High-tech R&D Program ("863" Program) of China(No.2013AA010605), the National Science Foundation of China (No.61161140454), and National Science & Technology Pillar Program of China (No.2012BAH01B01). In addition, we would like to thank Bingyang Liu, Guang Yao and Yangyang Wang for some very good comments on this work. Jun Bi is the corresponding author.

REFERENCES

[1] Craig Labovitz, "Bots, DDoS and, Ground Truth",NANOG50 ,October 5, 2010
[2] A. Bremler-Barr and H. Levy. "Spoofing Prevention Method". In Proceedings of IEEE INFOCOM, Miami, July 2005
[3] Heejo Lee, Minjin Kwon, Geoffrey Hasker and Adrian Perrig, "BASE: An Incrementally Deployable Mechanism for Viable IP Spoofing Prevention", in proc. ASIACCS'07, 2007
[4] Xin Liu and Xiaowei Yang, David Wetherall and Thomas Anderson, "Efficient and Secure Source Authentication with Packet Passports", in proc. SRUTI '06, 2006
[5] Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter L. Reiher, Lixia Zhang: "SAVE: Source Address Validity Enforcement Protocol", INFOCOM 2002
[6] P. Ferguson, D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", Request for Comments: 2827, May 2000
[7] F. Baker, P. Savola,"Ingress Filtering for Multihomed Networks", Request for Comments(RFC): 3704, March 2004
[8] Jin, G., Wang, H., and Shin, K. G. "Hop-count Filtering: An Effective Defense Against Spoofed DDoS Traffic". In Proceedings of the 10th ACM conference on Computer and communication security, 2003
[9] Vern Paxson, "End-to-End Routing Behavior in the Internet",SIGCOMM 1996
[10] Jianping Wu, Gang Ren, Xing Li, "Source Address Validation: Architecture and Protocol Design", ICNP 2007
[11] J. Wu, J. Bi, X. Li, G. Ren, K. Xu, M. Williams, "A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience", Request for Comments (RFC): 5210, June 2008
[12] J. Wu, J. Bi, M. Bagnulo, F. Baker, C. Vogt, Ed., "Source Address Validation Improvement Framework", Internet-Draft, July 25, 2011
[13] Bingyang Liu, Jun Bi, Yu Zhu, "A Deployable Approach for Inter-AS Anti-spoofing", ICNP 2011 FIST Workshop, 2011
[14] Tao, Z., Gong, Z., Lu, Z., Wang, B., Wang, H., Li, S., Liu, Y., Bi, J., and J. Wu, "OSPFv3-based Intra-Domain Source-Address Validation Implementation", draft-tao-savi-savo-01, November 2011.
[15] J. Moy, "OSPF Version 2", Request for Comments (RFC): 2328, April 1998
[16] "End System to Intermediate System Routeing Exchange Protocol for Use in Conjunction with the Protocol for Providing the ConnectionlessMode Network Service (ISO 8473) ", ISO 9542, March 1988
[17] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford, "SteinIntroduction to Algorithms",chapter 24.3,Second Edition
[18] router-level ISP topologies from rocketfuel project: http://www.cs.washington.edu/research/networking/rocketfuel