Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 9, Number 1, pages 7–11, January 2020

PKI-enabled OSPFv3 for Reliable IP Traceback

T. Oriishi
Graduate School of Advanced Tech. and Science
Tokushima University
Tokushima, Japan
takahiro0914@live.jp

K. Matsuura
Center for Admin. of Info. Tech.
Tokushima University
Tokushima, Japan
ma2@tokushima-u.ac.jp

K. Ohira
Cyber Media Center
Osaka University
Osaka, Japan
ohira@cmc.osaka-u.ac.jp

***Abstract*—In investigations for incidents, it is important to identify a user or a router from source IP addresses. However, in OSPFv3, each router only performs message authentication with pre-sharing key for security. Therefore, each router can claim arbitrary IP prefixes. This point makes difficult to associate a prefix with a router with some authorization. In this paper, we propose a new method that enables reliable IP traceback in OSPFv3 networks. Our proposal is to construct PKI on OSPFv3 and to associate each router with prefixes by the router's certificate. This proposal makes possible to identify the source of a packet from its source IP address directly. In this paper, we implement our proposal and confirm it can associate a router to prefixes.**

***Index Terms*—Dynamic Routing, IP Traceback, OSPFv3, PKI**

I. Introduction

In many cases, an intra-site network is divided into multiple subnets and managed by two or more subnet managers. Most of rules for unauthorized access and information confidentiality are stipulated in the entire intra-site network. However, connections in individual subnets are often left to each administrator. When an incident (for example, illegal access and malware) is detected on the intra-site network, investigators must identify source subnet of packets, for carry out measures to improve security. In similar systems, methods to identify source subnet of packets and administrator are important.

IP traceback is a method for identifying the source of packets. However, no method to managing nor operate subnets on OSPFv3 is standardized. Varying the topology, replacement of subnet managers and some misconfigurations have happened in operations of long term. Thus, routers, subnet managers, subnets, and prefixes are not associated in many cases.

In OSPFv3, only message authentication (AH/ESP) and payload encryption (ESP) are performed between neighbor routers as a security measure [1]. Each router can use arbitrary prefixes. This point makes associating information of a packet to network devices and subnets is difficult.

Following methods are proposed to work out IP traceback. One, each router marks own ID to passing packets, and investigators trace the route of the packet with its mark. Another one, each router logs packet information on own storage, and investigators trace the packet by search on logs of routers. However, these methods require access permission of all router's log or router's ID database. Moreover, investigators need to get help each router or subnet administrator. The routing table is sometimes changed on using OSPFv3. These points make it difficult to identify source subnets from the packet information.

In this study, we propose associating a router to prefixes for more reliable and easy IP traceback. We extend OSPFv3 by construct PKI and Prefix DBs. Associations of a router to prefixes is saved on Prefix DB. Investigators can search a source router from Prefix DB by a source IP address. Every router perform verification on received LSAs with originator's certificate. It prevents falsification by intermediate routers and incursion by unregistered routers. These give legitimacy of prefix information exchanged with OSPFv3. It enables reliable IP traceback. In this paper, we describe our proposal and implement and evaluate Registration Process.

II. Related Works

A. IP Packet Traceback

Some kinds of methods are proposed for IP traceback. Log base traceback is one of IP traceback method. However vast traffic passes on routers. Log data put pressure on the router's storage and make overhead. For solving this problem, each router saves hashed packet logs on storage [2]. Moreover, SPIE (Source Path Isolation Engine) proposes using a bloom filter as a data structure [3]. Another method PPM (Probabilistic Packet Marking) for IP traceback is proposed which is marking the router's ID to the packet and trace the route of it with mark. These methods have the possibility of false-positive which caused by hash collision and bit collision.

There is also HIT (Hybrid IP Traceback) that combines router logs and packet marking [4] [5] for more shrink overhead.

Kang et al. have also proposed a method of building a distributed database in the network and building log data storage outside the router [6].

However, with these methods, it is difficult to manage the association of router to prefixes.

B. PKI on OSPFv2

OSPFv2 only performs message authentication between neighbor routers. Thus, OSPFv2 may be attacked by LSA rewriting by intermediate routers. On this point, OSPFv2 is similar to OSPFv3. RFC2154 extends OSPFv2 by constructing PKI for introducing End-to-End message authentication [7]. PKI associates the router to its certificate. This method can

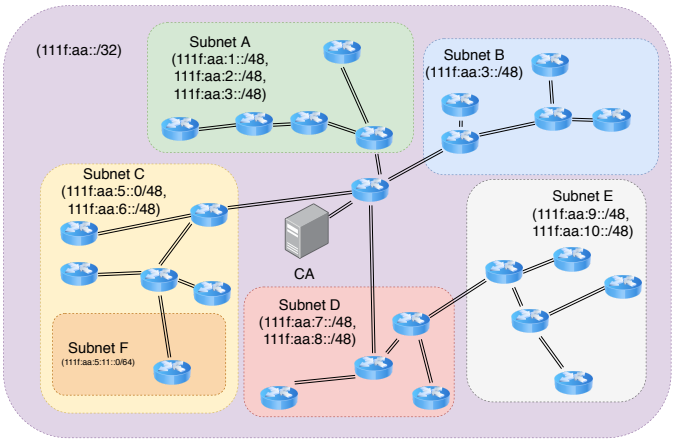

Fig. 1. Target Network Model

associate the LSA to the originating router. However, this method cannot associate CIDRs to the corresponding router.

C. Prefix Auto Assignment

Network managers can make hierarchical prefix management with DHCPv6-PD [8]. In zOSPF, a zOSPF router advertises prefixes to other routers in the subnet by LSA [9]. There is also a method of putting special routers that advertises the setting on OSPFv3 in the site and automatically setting the IPv6 address based on the advertised setting [10].

However, some subnet managers may not have advanced knowledge about IP network and additional operations are put burden to subnet managers. These methods cannot adequately associate the router to prefixes.

III. Proposal

In this study, we construct PKI on OSPFv3 for associating each router's certificate to properly assigned prefixes. Our proposal makes possible to identify the source router of the packet and to prevent using unassigned prefixes.

A. Target Network

The target network model of our proposal is shown in Fig. 1. The network has only a backbone area and is divided into some subnets. Each subnet is assigned some prefixes and can only use them. We propose CA (Certificate Authority) on the network. CA stores certificates of registered routers and records which associating router's certificate to properly assigned prefixes.

B. OSPFv3 Extension and Packet Format

Our proposal authenticates only Intra-Area-Prefix LSA because the target network has only a backbone area.

Routers cannot insert arbitrary data into OSPFv3 packets which are defined in RFC5340 [11]. Extension of LSA format in OSPFv3 which is named TLV (Type Length Value) and some packet formats are defined in RFC8362 [12]. We use E-Router LSA which is defined as new LSA type in RFC8362. In addition, we define the extension of Intra-Area-Prefix LSA for using TLV. Our proposal saves certificates, signatures and other additional information on TLV.

TLV is constructed by three fields which are shown in Table I. The TLV Type field specifies the type of value. The Length field describes byte length of value. The Value filed is for the value.

We define new TLV types which are shown in Table II. LSA signature type specifies the signature of E-Router LSA and Intra-Area-Prefix LSA. Newbie's Router ID, Assigned Prefixes and Newbie's Router Certificate fields are added to E-Router LSA which is generated by CA.

The packet format of E-Router LSA is shown in Table III. In E-Router LSA, some fields defined in RFC5340 are moved and expressed as TLV fields.

The packet format of extended Intra-Area-Prefix LSA is shown in Table IV. It is added TLV Length and TLVs fields. Prefix Entries field includes prefix-list data. Prefix Entry explains an IPv6 prefix. Its format is shown in Table V.

TABLE I
TLV FORMAT

Field	Length (bit)
TLV Type	16
Length	16
Value	Variable

TABLE II
TLV TYPES

Name	Value	LSA
LSA Signature	10	Intra-Area-Prefix and E-Router
Newbie's Router ID	20	E-Router
Assigned Prefixes	21	E-Router
Newbie's Router Certificate	22	E-Router

TABLE III
PACKET FORMAT OF E-ROUTER LSA

Field	Length (bit)
LSA State ID	32
Advertising Router	32
LS Sequence Number	32
LS Checksum	16
Length	16
Options	32
TLVs	Variable

C. Prefix DB

Our proposal puts Prefix DB which stores certificates, properly assigned prefixes, router ID, and associations in storages of CA and each router. Prefix DB is designed with the relational model and its schema is shown in Table VI. Prefix DB is used in Registration Process and Verification Process. We take up Registration Process and Verification Process in the following sections.

TABLE IV
PACKET FORMAT OF INTRA-AREA-PREFIX LSA

Field	Length (bit)
LSA Age	16
LSA Type	16
Link State ID	32
Advertising Router	32
LS Sequence Number	32
LS Checksum	16
Length	16
Number of Prefix Entry	16
Referenced LS Type	16
Referenced Link State ID	32
Referenced Advertising Router	32
TLV Length	16
Prefix Entries	Variable
TLVs	Variable

TABLE V
FORMAT OF PREFIX ENTRY

Field	Length (bit)
Prefix Length	8
Prefix Options	8
Metric	16
Address Prefix	Variable

D. Registration Process

The sequence diagram of Registration Process is shown in Fig. 2. In Registration Process, Newbie participates to the existing network.

- CA and some routers are joined to the existing network.
- Newbie's certificate, router ID, and properly assigned prefixes are not known by CA and other routers.
- Newbie has CA's certificate.
- There is a confidential connection between Newbie and CA.

Newbie issues a private/public key pair, creates a CSR (Certificate Signing Request), and sends the CSR to the CA. The CA issues a new Router ID, assigns it to Newbie, issues a certificate based on the CSR received from Newbie, and registers it in the CA's own Prefix DB in association with the Router ID. The CA then returns the certificate, Router ID, and prefix information issued to Newbie. The CA originates E-Router LSA with Newbie's information and registers to own LSDB. The CA can advertise Newbie's information to other routers after it finished the above tasks.

TABLE VI
ROUTERS TABLE

Field	Type	Options
router_id	varchar	PRIMARY KEY
certificate	varchar	NOT NULL
expiration	varchar	NOT NULL

TABLE VII
PREFIXES TABLE

Field	Type	Options
router_id	varchar	PRIMARY KEY
prefix	varchar	NOT NULL

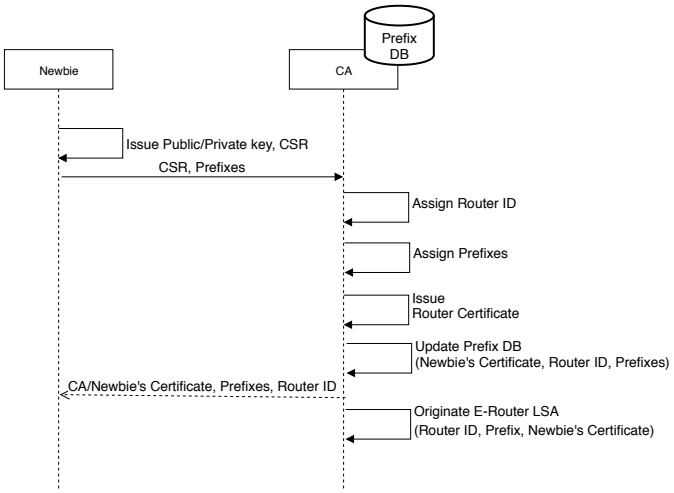

Fig. 2. Sequence Diagram of Registration Process

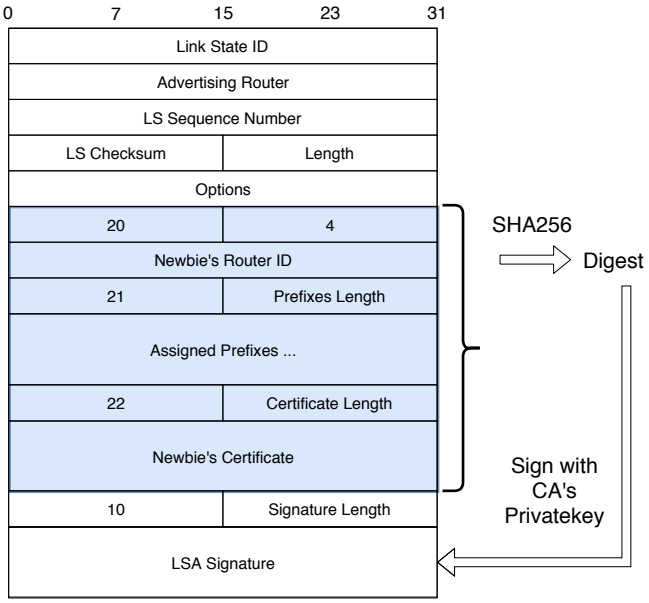

Fig. 3. Example of E-Router LSA

We describe E-Router LSA which is issued at this time. CA adds Newbie's Router ID, Assigned Prefixes and Newbie's Certificate as TLVs to this LSA. As shown in Fig. 3, CA creates the signatures for E-Router LSA and some TLVs . The signature is added to E-Router LSA as TLV. E-Router LSA is flooded when LSR (Link State Request) packet is received from an adjacent router.

E. Verification Process

A router performs the Verification Process when sending and receiving Intra-Area-Prefix LSA from neighboring routers.

The sequence diagram of Verification Process is shown in Fig. 4.

First, the Sender generates an LSA, signs the LSA using its own private key, and stores it in its own LSDB. At this time, as shown in Fig. 5, a signature is created for the LSA excluding the LSA header, stored in the TLV, and added to the LSA. Save the created Intra-Area-Prefix LSA in the LSDB. When Sender receives LSR packet from Receiver, it floods Intra-Area-Prefix LSA.

The receiver performs the following two verifications on the received LSA. Signature verification using the Sender's Router ID and corresponding certificate stored in PrefixDB. Check if the advertised prefix is included in the list of allowed prefixes corresponding to the Sender Router ID stored in PrefixDB. If there is no problem with all verifications, Receiver stores the received LSA in its LSDB.

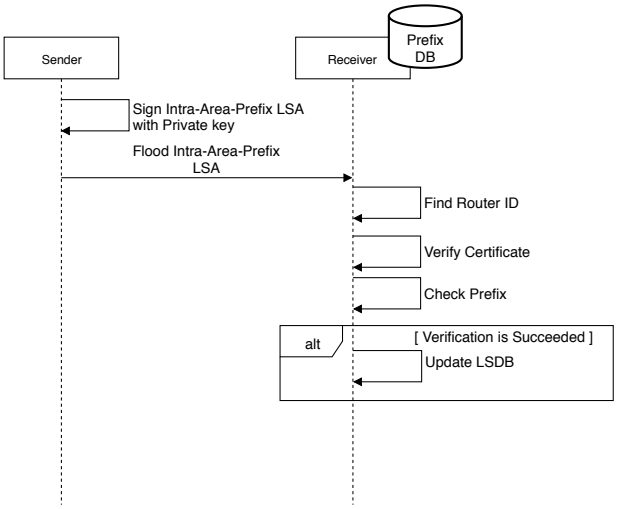

Fig. 4. Sequence Diagram of Verification Process

IV. Implementation and Experiments

A. Implementation

We implement the proposal on software router Quagga[1]. Table VIII shows the software used for the implementation. We use OpenSSL to sign and verify. We use SQLite3[2] to make Prefix DB persistence.

TABLE VIII
Base Software

Kind	Name
Software Router	Quagga 0.99.19
X509	OpenSSL 1.1.0k
RDBMS	SQLite3 3.16.2

[1]"Quagga Routing Suite," https://www.quagga.net/
[2]"SQLite3," https://www.sqlite.org/index.html

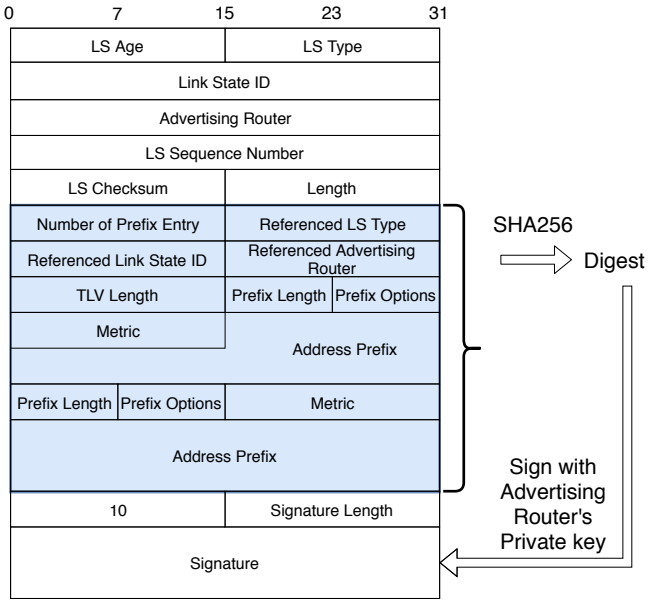

Fig. 5. Example of Intra-Area-Prefix LSA

B. Experiments and Results

Table IX shows that the environment of the experiment. We run the implementation of our proposal on a virtual network constructed by docker container[3] and virtual bridge of Open vSwitch[4].

In this paper, we implement and perform only the Verification Process. For skip Registration Process, all record of router ID, router's certificate, properly assigned prefixes, and associations are saved on a SQLite3 database file beforehand. When the router is activated, it loads all records from the SQLite3 database file to Prefix DB.

Fig. 6 shows network topology for experiments. The network has five routers and all routers are assigned some prefixes shown in Table X.

We have evaluated our proposal from the following two viewpoints.

TABLE IX
Experiments Environment

Kind	Name
CPU	Intel Core i77500U @ 4x 3.5GHz
RAM	15802MiB
Host OS	Manjaro 18.0.4 Illyria
Virtualization	Docker 18.09.7-ce
Virtual Bridge	Open vSwitch 2.11.0-1

1) Associate the Router to Prefixes: We perform the following two scenarios to confirm our proposal can associate the router to prefixes.

A router is configured to use an unassigned prefix, and joins to the network. In this scenario, the other routers fail to verify

[3]"Enterprise Container Platform for High-Velocity Innovation," https://www.docker.com/
[4]"Open vSwitch," https://www.openvswitch.org/

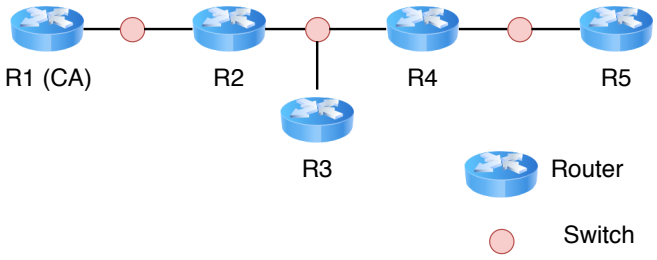

Fig. 6. Network Topology in Experiments

TABLE X
ADEVERISING PREFIXES AND ASSIGNED PREFIXES

Router	Advertising Prefixes	Assigned Prefixes
R1	2001:1:1::/64	2001:1:1::/64, 2001:1:5::/64
R2	2001:1:1::/64, 2001:1:2::/64	2001:1:1::/64, 2001:1:2::/64
R3	2001:1:2::/64	2001:1:2::/64
R4	2001:1:2::/64, 2001:1:3::/64	2001:1:2::/64, 2001:1:3::/64
R5	2001:1:3::/64	2001:1:3::/64

the signature of LSA. These routers did not register LSA to LSDB nor did not flood to neighboring routers.

We configure a router to use a private key which does not correspond to registered certificates on Prefix DB. The router joins to the network. These routers also did not register LSA to LSDB nor did not flood to neighboring routers.

2) Verification Time: The proposed method was run on a virtual network, and the time required for signature creation and signature for one LSA was measured 20 times. We use SHA256 for the message digest hash algorithm, and the generated certificate size was 944 (B). The time required for signature was 1.05 (ms) at minimum, 6.17 (ms) at maximum, and 3.14 (ms) on average. The time required for verification was at minimum of 0.2 (s), at maximum of 0.5 (ms), and 0.32 (ms) on average.

V. CONCLUSION

In this paper, we proposed an extension of OSPFv3. It can identify a source of subnet from IP packet information reliably. In the proposed method, prefixes and routers are associated by PKI-enabled OSPFv3. By using the proposed method on the network, it is possible to identify the source of the router directly from the IP address in the packet header.

Furthermore, we implemented the Verification Process on ospf6d of the routing software Quagga. We confirmed that a prefix and the router were correctly associated. We also confirmed that signature creation and verification with SHA256 can be executed in a relatively small amount of time and that there is no significant overhead for OSPF operations.

In the future, we will implement and evaluate the Registration Process. In addition, we plan to define and implement a certificate renewal and invalidation protocol for each router and operation in multiple areas.

VI. ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19K11943.

REFERENCES

[1] M. Gupta and N. Melam,“Authentication/Confidentiality for OSPFv3, ” Request for Comments 4552, 2006.
[2] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F Tchakountio, B. Schwartz, S.T. Kent and W.T. Strayer, “Single-packet IP traceback, ” IEEE/ACM Transactions on Networking,volume: 10, number: 6, pp. 721 734, 2002.
[3] N. Lu, Y. Wang, F. Yang and M. Xu, “A novel approach for single-packet IP traceback based on routing path, ” Proceedings - 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2012, pp. 253 260, 2012.
[4] C. Gong and K. Sarac, “A More Practical Approach for Single-Packet IP Traceback using Packet Logging and Marking, ” IEEE Transactions on Parallel and Distributed Systems, pp. 253 260, 2012.
[5] Snoeren, Alex C, Partridge, Craig and Sanchez, Luis A. and Jones et al., “Single-packet IP traceback, ” IEEE/ACM Transactions on Networking, pp. 721 734, 2002.
[6] H.S. Kang and S.R. Kim, “A New Logging-based IP Traceback Approach using Data Mining Techniques, ”, Journal of Internet Services and Information Security (JISIS), volume: 3, number: 3, pp. 72 80, 2013.
[7] S. Murphy, M. Badger and B. Wellington, “OSPF with Digital Signatures, ” Request for Comments 2154, 1997.
[8] O. Troan and R. Droms, “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6, ” Request for Comments 3633, 2003.
[9] A. Dimitrelis and A. Williams, “Autoconfiguration of routers using a link state routing protocol, ” Internet Draft (expired), 2002.
[10] K. Ohira, “A Method of IPv6 Auto Address Assignment with OSPFv3 in Multi-Link Site, ” IEICE Technical Report, volume: 113, p. 147 152, 2014.
[11] R. Coltun, A. Roy, D. Ferguson, V.R. Vallem and F. Baker, “OSPF for IPv6, ” Request for Comments 5340, 2008.
[12] A. Lindem, A. Roy, D. Goethals, V.R. Vallem and F. Baker, “OSPFv3 Link State Advertisement Extensibility,” Request for Comments 2154, 2018.