Network Path Validation for Packets Delivery

Dorine CHAGNON
LIMOS, France
Université Clermont-Auvergne
CNRS, Mines Saint-Étienne

Kévin THIRY-ATIGHEHCHI
LIMOS, France
Université Clermont-Auvergne
CNRS, Mines Saint-Étienne

Gérard CHALHOUB
LIMOS, France
Université Clermont-Auvergne
CNRS, Mines Saint-Étienne

***Abstract*—Path validation that enforces and verifies the path taken by network packets has become an important block for secure network protocols and defence mechanisms. Indeed, an Internet Services Provider (ISP) might decide to redirect your traffic through a less efficient ISP just because it is cheaper. That arises some security issues. This paper exposes the possible problems around the lack of path validation and provides a state-of-the-art of the existing path validation protocols.**

***Index Terms*—network security, path validation, cryptography**

I. Introduction

In the current Internet, routers have full control over packet delivery. The destination address is put in packets and the packets are launched into the network. The network decides the path that the packets take, and the intermediate providers decides if the path passes through them. Sometimes, senders, intermediates routers, and receivers would prefer to control paths of packets. A receiver might request that packets pass through an intrusion detection service or a packet-cleaning service before its arrival. A sender might want to make sure that their packets are carried through friendly countries to avoid packet spying or packet dropping. Path validation aims at allowing end-hosts to enforce the path and to verify that packets have followed the approved path. Several works describe path validation protocols ([2], [5]–[11]) that try to enforce this forwarding process. These protocols manage to thwart skipping attacks or out-of-order-attacks, but they are lacking in detecting addition attacks.

This paper describes what a path validation protocol requires, what kind of attacks it should be able to detect, and possibility, to prevent. Finally, a state-of-the-art of the existing path validation protocols is stated.

II. Problem definition

A. Desired security properties

A network protocol is a path validation protocol if it satisfies both path enforcement and path verification. Path enforcement assures a packet is forwarded on the agreed path over each node en-route in the correct order. Path verification assures that the sender, the intermediate, and the destination routers are able to verify that the forwarded packet has followed the correct path so far.

Chaire de confiance numérique (12LIMO11LIMOS on FCA)

B. Adversary model

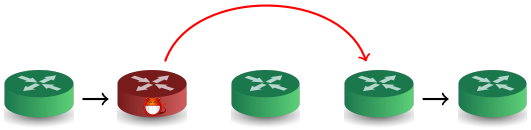
A router that obeys the protocol is called an *honest* node. Some untrustworthy routers may exist due to an attacker. An attacker has the ability to eavesdrop at any point along the path from the source to the destination. An attacker is capable of controlling a subset of the network. The compromised nodes may have the following malicious behaviours, as defined in [1]:

Packet injection: A malicious router can send arbitrary packets toward a destination of its choice.

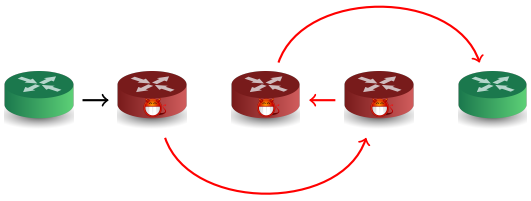
Packet alteration: A malicious router can alter any part of the packet such as the source address, header information or payload data.

Path deviation: A malicious router redirects packets along a path other than the path previously selected by the end-hosts. There are several types of this kind of attacks:

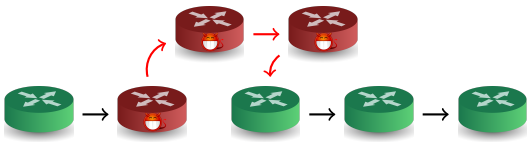
- Skipping attack: A malicious router redirects the packet and skips at least one router on the path. Hence, some router along the intended path does not forward the packet.



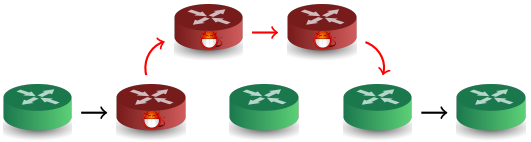
- Out-of-order attack: One or many routers on the intended path are not traversed in the right order, but they are all visited.



- Addition attack : Packets are forwarded to some additional routers that are not on the intended path and eventually return to the expected path. Packets visit one or more routers that are not expected.



- Detour attack: Packets are deviated from some of the routers before arriving to the destination. Not all routers are visited.



III. Related work before path validation protocols

This section exposes solutions that pursue path validation but do not achieve both path enforcement and path verification, which are keys to path validation.

Secure routing aims to find feasible paths between end-hosts while preventing the path-finding process from being attacked. However, routing does not assure that the packet followed the chosen path. It is the forwarding process that determines whether the packet is delivered according to the routing policies.

Secure source routing embeds path directives in packet headers, using cryptographic tools, to inform intermediate routers where to forward packets. Secure source routing achieves path enforcement.

Secure traceroute enables end-hosts to retrieve the path taken by a packet. It achieves path verification.

Nevertheless, secure source routing and secure traceroute are not designed to operate together, it is either infeasible or inefficient to integrate them for path validation [1].

IV. Path validation protocols

A. General design

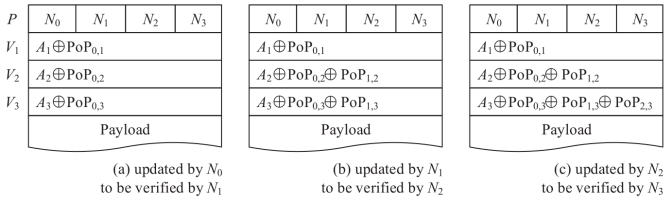
Existing path validation protocols are built on the assumption of a plentiful network bandwidth. That means that they incorporate mechanisms of path validation in headers of packets without caring about their bandwidth consumption [3]. They do path enforcement and path verification by adding path proofs in headers of packets. These proofs should be difficult to forge or alter, and their size can be quite large, on the order of hundreds of bytes [1]. These proofs are used to attest that the packet has traversed intermediate routers in the correct order because each intermediate router should update these proofs.

Mostly, path validation consists of the following steps [1]:

- Path dissemination: the specified path chosen by the end-hosts must be shared with the intermediate routers. It can be done either by putting the path directly in header of packets, or by communicating the path beforehand to the en-route routers.
- Key exchange: routers exchange and share keys that are used to compute the cryptographic proof added to the header of packets for others to verify.
- Proof initialization: the source has to initialize the cryptographic proof(s) that will be used by other routers to verify the path of the packet and to update with their own proof.
- Proof verification: upon receiving a packet, a router first verifies the correctness of the embedded proof to be sure that the packet travelled though the correct path so far.
- Proof update: if the proof verification succeeded, the router continues by updating the embedded proof with its own and then forwards the packet to the next hop according to the path it was given.

B. State-of-the-art

1) ICING [9]: ICING is considered to be the first path validation protocol. ICING uses shared keys for each pair of nodes. The path is embedded inside the header of the packet once each node on this path has acknowledged its agreement to the source with a Proof of Consent. Then the source initializes verifiers V_i for each node i along the chosen path by xoring a Proof of Provenance ($PoP_{0,i}$) and the obfuscated Proof of Consent (A_i). $PoP_{k,i}$ is a cryptographic proof built with the key shared between nodes k and i. A node i only needs to verify its corresponding verifier V_i. To do that, it will recompute A_i xored with the successive $PoP_{k,i}$ shared with its predecessors on the path. Once the verification is done, node i will update the verifiers of its successors k with the $PoP_{i,k}$ shared with them (see Fig. 1).



(a) updated by N_0 to be verified by N_1 (b) updated by N_1 to be verified by N_2 (c) updated by N_2 to be verified by N_3

Fig. 1. ICING packet instance taken from K. Bu et al. [9]

2) OPT [7]: Origin and Path Trace (OPT) is more efficient than ICING because it assumes a higher level of trust. It is assumed that the source is trustworthy, hence each en-route node will only share a key with the source and not with each other as in ICING. The source will initialize an *origin and path validation* verifier, OPV_i, for each en-route node, as well as a general *path validation field*, PVF, that will be updated. The OPV_i contains all the cryptographic proofs of the predecessors of the router i. Upon receiving a packet, the current router will check PVF to be sure that the packet went through all the expected routers. In order to do that, it will compare its OPV_i with the general PVF updated with its cryptographic proof. If the verification is successful, the router i updates PVF with its cryptographic proof. During this protocol, a node i only has to compute one cryptographic proof whereas in ICING, a node has to compute n cryptographic proofs if there are n routers on the path.

3) OSV [2]: Orthogonal Sequences Verification (OSV) is very similar to OPT except that instead of using cryptographic proofs, it uses orthogonal sequences based on Hadamard matrices to build the verifiers. Doing inner product is faster than computing cryptographic proofs.

TABLE I
COMPARISON OF VALIDATION PATH PROTOCOLS FOR A PATH OF n ROUTERS AND A SECURITY LEVEL OF 80 BITS

	Number of known keys by intermediate routers	Proof length	Verifier length (bytes)	Communication overhead (bytes)	Number of computation
ICING	$\mathcal{O}(n)$	$\mathcal{O}(n)$	42	$42n + 13$	$\mathcal{O}(n)$
OPT	$\mathcal{O}(1)$	$\mathcal{O}(n)$	16	$16n + 52$	$\mathcal{O}(1)$
OSV	$\mathcal{O}(1)$	$\mathcal{O}(n)$	2	$2n + 108$	$\mathcal{O}(1)$
PPV	$\mathcal{O}(1)$	$\mathcal{O}(1)$	4	64	$\mathcal{O}(1)$
Atomos	$\mathcal{O}(n)$	$\mathcal{O}(1)$	1415	1415	$\mathcal{O}(1)$
PrivNPV	$\mathcal{O}(1)$	$\mathcal{O}(n)$	16	$16n + 48$	$\mathcal{O}(1)$
EPIC	$\mathcal{O}(1)$	$\mathcal{O}(n)$	5	$5n + 24$	$\mathcal{O}(1)$

4) PPV [11]: PPV uses a flow of packets to validate a path. A router has a certain probability to mark a packet and does not mark all packets. However, a packet in a flow will be marked by two adjacent routers. Collecting all the marks from all the packets enables the forwarding process to be validated. This solution reduces the size of the header of packets.

5) Atomos [5]: Atomos uses asymmetric cryptographic proofs. A router only needs one proof to prove that the packet went through it. This solution aggregates the proofs of all the routers so that there is only one verifier field in the header of a packet. That small header counterbalances the longer computation time of asymmetric cryptography.

6) PrivNPV [10]: Privacy-preserving network path validation enables an en-route router to not know which other nodes are on the path, except for its two neighbours. PrivNPV also states that an en-route router does not know its position on the path, *i.e.* it does not know whether it is in the middle of the path or just before the destination.

7) EPIC [8]: EPIC for "Every Packet Is Checked" is deployed in a stronger attacker model that combines a localized Dolev-Yao adversary [4] with a cryptographic oracle while reducing the size of the header of packets. Since EPIC uses short verifier fields , an attacker can brute force these values for a single packet. In order to model the attacker's ability, EPIC proposes a strong attacker model where a malicious router can obtain valid verifier fields by giving hop information and a packet initialization to an oracle. That means that an attacker can forge individual packets. However, the security is not altered because the forging must happen in a short time frame. Indeed, the verifier fields depend on timestamps. Moreover, the security for end-hosts is not affected because the validation field for the destination is cryptographically strong and can not be forged with this oracle.

C. Comparison

For a path of n routers and a security level of 80 bits, Table I compares the above protocols in terms of overhead of key storage, proof length, verifier length, communication overhead, and computation complexity.

V. CONCLUSION

All validation protocols are able to detect attacks such as packet injection, packet alteration, skipping attack, out-of-order attack, detour attack because the data inside the packet are not going to be the expected one. The proofs embedded inside the packet are used to detect these kinds of behaviour. However, these protocols are not able to detect addition attacks if the additional routers do not do any changes to the packet. Indeed, all the required routers are traversed so that all the required proofs are verified and updated.

ACKNOWLEDGMENT

The authors acknowledge the support of the Chaire de confiance numérique (12LIMO11LIMOS on FCA)

REFERENCES

[1] K. Bu et al., 'Unveiling the Mystery of Internet Packet Forwarding: A Survey of Network Path Validation', ACM Comput. Surv., vol. 53, no. 5, p. 104:1-104:34, Sep. 2020, doi: 10.1145/3409796.

[2] H. Cai and T. Wolf, 'Source Authentication and Path Validation in Networks Using Orthogonal Sequences', in 2016 25th International Conference on Computer Communication and Networks (ICCCN), Aug. 2016, pp. 1–10. doi: 10.1109/ICCCN.2016.7568576.

[3] K. L. Calvert, J. Griffioen, and L. Poutievski, 'Separating routing and forwarding: A clean-slate network layer design', in 2007 Fourth International Conference on Broadband Communications, Networks and Systems (BROADNETS '07), Sep. 2007, pp. 261–270. doi: 10.1109/BROADNETS.2007.4550434.

[4] D. Dolev and A. Yao, 'On the security of public key protocols', IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983, doi: 10.1109/TIT.1983.1056650.

[5] A. He, K. Bu, Y. Li, E. Chida, Q. Gu, and K. Ren, 'Atomos: Constant-Size Path Validation Proof', IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3832–3847, 2020, doi: 10.1109/TIFS.2020.3001669.

[6] W. Jin, E. Kline, T. K. S. Kumar, L. Thurlow, and S. Ravi, 'P3V: Privacy-Preserving Path Validation System for Multi-Authority Sliced Networks'. 2023. Accessed: Jan. 31, 2023. [Online]. Available: https://eprint.iacr.org/2023/053

[7] T. H.-J. Kim, C. Basescu, L. Jia, S. B. Lee, Y.-C. Hu, and A. Perrig, 'Lightweight source authentication and path validation', in Proceedings of the 2014 ACM conference on SIGCOMM, New York, NY, USA, Aug. 2014, pp. 271–282. doi: 10.1145/2619239.2626323.

[8] M. Legner, T. Klenze, M. Wyss, C. Sprenger, and A. Perrig, 'EPIC: every packet is checked in the data plane of a path-aware internet', in Proceedings of the 29th USENIX Conference on Security Symposium, USA, Aug. 2020, pp. 541–558.

[9] J. Naous, M. Walfish, A. Nicolosi, D. Mazières, M. Miller, and A. Seehra, 'Verifying and enforcing network paths with icing', in Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies, New York, NY, USA, Dec. 2011, pp. 1–12. doi: 10.1145/2079296.2079326.

[10] B. Sengupta, Y. Li, K. Bu, and R. H. Deng, 'Privacy-preserving Network Path Validation', ACM Trans. Internet Technol., vol. 20, no. 1, p. 5:1-5:27, Feb. 2020, doi: 10.1145/3372046.

[11] B. Wu et al., 'Enabling Efficient Source and Path Verification via Probabilistic Packet Marking', in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Jun. 2018, pp. 1–10. doi: 10.1109/IWQoS.2018.8624169.