Name-signature lookup system: A security enhancement to named data networking

Zhicheng Song, Pushpendu Kar

University of Nottingham
UK | CHINA | MALAYSIA

University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo, 315100, Zhejiang, China.

First published 2021

University of Nottingham
UK | CHINA | MALAYSIA

Name-Signature Lookup System: A Security Enhancement to Named Data Networking

Zhicheng Song
School of Computer Science
University of Nottingham
Nottingham, United Kingdom NG7 2RD
Email: scyzs1@nottingham.ac.uk

Pushpendu Kar*
School of Computer Science
University of Nottingham Ningbo China
199 Taikang East Road, Ningbo 315100, China
Email: Pushpendu.Kar@nottingham.edu.cn

***Abstract*—Named Data Networking (NDN) is a content-centric networking, where the publisher of the packet signs and encapsulates the data packet with a name-content-signature encryption to verify the authenticity and integrity of itself. This scheme can solve many of the security issues inherently compared to IP networking. NDN also support mobility since it hides the point-to-point connection details. However, an extreme attack takes place when an NDN consumer newly connects to a network. A Man-in-the-middle (MITM) malicious node can block the consumer and keep intercepting the interest packets sent out so as to fake the corresponding data packets signed with its own private key. Without knowledge and trust to the network, the NDN consumer can by no means perceive the attack and thus exposed to severe security and privacy hazard. In this paper, the Name-Signature Lookup System (NSLS) and corresponding Name-Signature Lookup Protocol (NSLP) is introduced to verify packets with their registered genuine publisher even in an untrusted network with the help of embedded keys inside Network Interface Controller (NIC), by which attacks like MITM is eliminated. A theoretical analysis of comparing NSLS with existing security model is provided. Digest algorithm SHA-256 and signature algorithm RSA are used in the NSLP model without specific preference.**

***Index Terms*—Named Data Networking, Man-in-the-middle attack, Network Interface Controller**

I. Introduction

Initially, Named Data Networking is designed with security features which are sophisticated to be implemented in IP networking with massive overheads [2]. However, to verify encapsulated data packets, the consumer has to iteratively look through *key locator* as declared until an installed trust anchor has been reached. Mostly this verification mechanism works efficiently. However, there are chances that packets for installing a trust anchor are compromised [3].

To eliminate the aforementioned threats, in this paper, we introduce Name-Signature Lookup System (NSLS), a semantically-centralized registry with hardware-level authentication. Tuples *naming zone - public key* are stored in the registry servers. As discussed in Section IV, the NSLS enables the NDN of enhanced security features using standard NDN scheme.

The contribution of this paper is threefold. First, we explain why the safety situation cannot be eliminated by the existing NDN security scheme. Second, we design the NSLS with a protocol to resolve related attacks. Third, we discuss the compatibility of NSLS to NDN. This work on NSLS can draw more inspiration on the combination of hardware and software for cyber-security.

II. Background

A. Named Data Networking

Named Data Networking (NDN) [1], is a content-centric networking where consumer sends out interest packet with a specific name and pulls back the data packet from somewhere in the network with the corresponding name.

Instead of point-to-point communication, NDN avoids establishing channels but to request for a packet with a certain name to the whole network. Any node possessing the packet with matching name respond directly to the consumer. Thus, critical security threats shift from securing the channel to verify the data packet itself by design.

B. Man-in-the-middle Attack (MITM)

Man-in-the-middle attacks (MITM) refers to the safety situation, where a malicious node intercepts a data link between two nodes, generally between an end-user and a router or switch connected to the rest of the network [6]. MITM can intercept every packet that passes through the link and even make a fake one based on meta-information. For an attack on NDN network, MITM can encapsulate a data packet with the same name but different content, and sign the packet with its own key. By intercepting and spoofing the packet containing the genuine public key, the user can never perceive it is under cyber attack and be exposed to MITM entirely.

C. Network Interface Controller (NIC)

Network Interface Controller (NIC) is a hardware component of a networked node which provides low-level connections to any other network. NIC stores crucial details such as MAC address and protocol stacks for data packet exchange with the network. Moreover, these details are transparent to the application level, which in other word means it is difficult to modify any value embedded in NIC through operating systems directly.

D. Digest algorithm and Signature algorithm

A digest algorithm or a hash algorithm is used to produce a message digest which can indicate the integrity of the message. A Signature algorithm typically uses the private key out of a pair of asymmetric keys to sign the hash value of the message, and anyone having the corresponding public key can verify the signature enclosed in the packet.

In terms of NDN, a packet digest comes out by applying a proper hash algorithm to the combination of the name and the content. Then a proper signature algorithm is applied to the hash value to get a signature which eventually is enclosed in the packet.

In the context, digest algorithm SHA-256 and signature algorithm RSA [7] are selected for instance without explicit preference. Other hash algorithms, such as MD5, HMAC, SM3 [13] and signature algorithms such as 3DES, AES, SM2 [12] are also valid. Specific digest algorithm or signature algorithm can be substituted for designated scenarios.

E. Existing Security Scheme for NDN

The data packet is requested to enclose the signature from the producer. The signature is generated by implementing a signature algorithm on the digest of the name and the content [4]. This inherent security scheme can ensure the authenticity and integrity of the data packet by verifying the signature derived from the key locator. The key locator always points to a superior publisher until it reaches to the root, which is the *trust anchor*.

In summery, although existing NDN security schemes use signature and key locator to ensure the authenticity and integrity of a single data packet, there is still the possibility of a MITM blocking the communication from the very beginning. Key locator cannot ensure that the trust anchor is not fabricated since every packet to the so-called reliable source can be intercepted under this extreme circumstance.

III. Problem Statement

In terms of the security, which is authenticity and integrity, of every NDN data packet following a standardized encryption procedure, NDN has realized a higher level of security than IP networking by design. However, as we aforementioned, with the inherence of mobility, an NDN consumer node can join a network without any pre-knowledge, leading to a situation that a MITM can spoof the consumer by not only fabricating the data packet with the same name as the interest packet sent outbound but also providing a legitimate signature. The consumer is not able to perceive the sensitive situation since MITM can intercept any packet going through and thus any kind of hierarchical trust mechanism takes no effect without any pre-knowledge.

Theoretically, MITM cannot be detected directly as stated. Thus, our work mainly focuses on how to eliminate the critical threats caused by an intercepting node, which is specifically spoofing. As a trade-off, privacy especially interest leakage related to eavesdropping or intercepting should be covered by a higher level cryptography algorithm to prevent sensitive content leakage. The Name-Signature Lookup System (NSLS) is carried out to ensure the authenticity and integrity of packets but not secure delivery.

IV. Proposed Solution

To solve the defect of the existing NDN security scheme, we propose a solution with an additional layer of verification, including the support of the NIC hardware and a semantically-centralized server running the Name-Signature Lookup System. The mechanism of conducting the NSLS verification upon the ordinary NDN architecture would be named as *Name-Signature Lookup Protocol* (NSLP).

The prerequisite of NSLP is:

- The NSLS server generates a pair of RSA asymmetric keys and makes it accessible to all the manufacturers of NIC.
- To support a specific NSLS server of NSLP, the manufacturers should embed the up-to-date accessible public key of the NSLS server into the NIC.
- The NDN consumers should equip their network-accessing node with a NIC supporting NSLP.

The integrity of the hardware NIC is crucial to the implementation of NSLP. Thus, the most ideal proposition would be, the profile of NIC is immutable from the perspective of the operating system layer or above, which is similar to Basic Input/Output System (BIOS) on motherboard [10]. The Hardware-level restriction does enhance the reliability of the security model involving asymmetric verification.

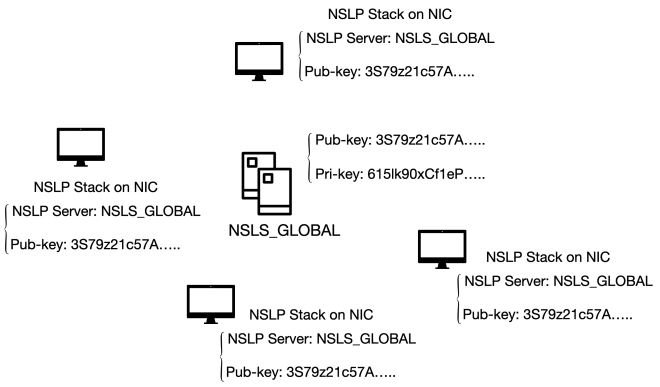

Fig. 1. Illustration of NDN nodes with NSLP

Correspondingly, there would be a cluster of servers running NSLS on the other end. Internally the cluster can form a hierarchy structure similar to DNS [8], [9], while externally it is regarded as an accredited registry where NSLP-enabled publishers can register their NDN-identified name zone with their public key. In other words, a specific packet name only matches at most one unique public key declared by the publisher under NSLP. By this approach, it can also unify the universal NDN namespace. When a consumer attempts to *pull* a packet (with name *pName* for instance), the node sends out two interest packets in parallel. One is the normal

NDN interest packet for *pName*, the other is semantically an NSLP packet. The name of the NSLP packets comprises a prefix pointing to the NSLS server and the name of the interest packet *pName*. Specifically, the name of the NSLP packet is *NSLS/pName* in the context (the name of the NSLS server is assumed as NSLS). The procedure is carried out within the scope of NIC and the actual string of the prefix is directly extracted from NIC. The worker thread shown in Procedure 2 is invoked by Algorithm 1. Both of the *interest* packets go through the standardized NDN procedure, which is, matching the name of the data packet no matter from Content Store of an NDN router or from an NSLS server. A *Content Store* or a built-in cache on NIC for NSLP can be introduced for better performance and this is discussed in the later section. The threads joins after receiving both of the data packets.

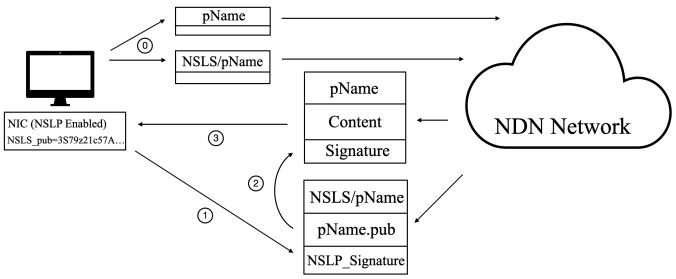

Fig. 2. Overview of how NSLP works in an NDN network

The verification procedure starts from the NSLP data packet first. The signature of the NSLS server can be verified at ease, since the corresponding public key of the specific NSLS server has been embedded into NIC. Any fake signature can be detected at this stage thus preventing further security hazards. If the immutable public key can decrypt and verify the NSLP packet, then the public key on consignment can be retrieved from the content of the NSLP data packet. Unlike normal NDN data packets using key locator to verify the data packet, NSLP uses the public key extracted from the corresponding NSLP data packet to proceed verification. The public key then can be regarded to be authenticated from a universal-trusted registry. Ultimately, any data packet cannot be verified by the *genuine* public key is discarded. On the contrary, content retrieved from an NSLP-verified packet remains authenticity and integrity even from an untrusted NDN network.

The core idea of the NSLP protocol is to make sure every node enabling NSLP has existing knowledge before joining any network. Public keys embedded in the NIC can be regarded as hardware-level trust anchors. Implementation of NSLP and NSLS does not eliminate Man-in-the-middle itself but reduces the impact of being under MITM attacks.

Besides, A NSLP interest and data packet can be regarded as a wrapper based on typical NDN interest and data packet. The only semantic difference is that, the name of NSLP packet explicitly points to an NSLS server. In other words, NSLP has a higher level of abstraction over NDN, which indicates excellent compatibility and scalability.

Algorithm 1 Name-Signature Lookup Protocol

Input:
NSLS ← NSLS server name referred in NIC
NSLS.pub ← corresponding public key of NSLS server
pName ← contentname to fetch from NDN network
Output:
content of data packet *pName* from NDN

```
send i_pkg for pName
threads_fork (invoking Procedure 2)
receive d_pkg pName
threads_join
if RSAVerify(NSLS/pName, NSLS.pub) then
    pName.pub ← content of NSLS/pName
    if RSAVerify(pName, pName.pub) then
        return
    else
        Abort
    end if
else
    Abort
end if
```

Procedure 2 Name-Signature Lookup Protocol worker thread

```
if lookupNICcache(NSLS/pName) then
    return d_pkg NSLS/pName
    threads_join
else
    send i_pkg NSLS/pName
    receive d_pkg NSLS/pName
    threads_join
end if
```

V. Analysis

A. Robustness

The robustness of normal NDN scheme relies on the transmission of the packet containing a trust anchor. This security scheme can be compromised when encountering such extreme MITM attacks in the context.

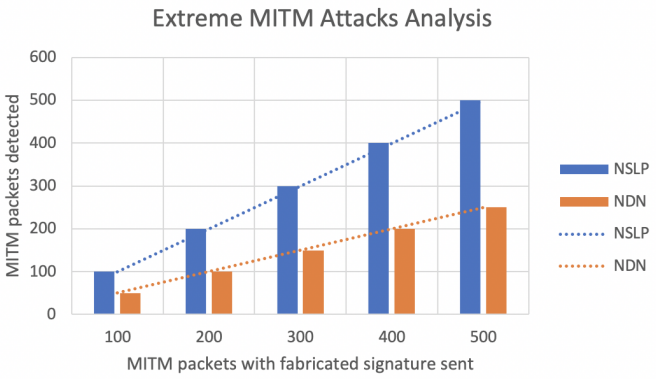

Fig. 3. Analysis when encountering 50% of extreme MITM attacks

On the contrary, the robustness of NSLP depends on the signature algorithm applied. The NSLP remains effective as long as the signature algorithm applied is proved to be robust at the current stage. The implementation of NSLP realizes systematic robustness by design. However, there are Byzantine faults which might lead to immediate cracking to a specific signature algorithm.

Figure 3 shows the expected performance when NSLP and NDN encounter hybrid MITM attacks. Half amount of attacks comprise fabricated signature and no existing trust anchor can verify that. The attacks aforementioned simulate the situation when NDN nodes attempt to install an unknown trust anchor. In contrast, NSLP can theoretically detect fabricated signature attacks with the help of hardware-level pre-installed trust anchor, which is the asymmetric public key in the context.

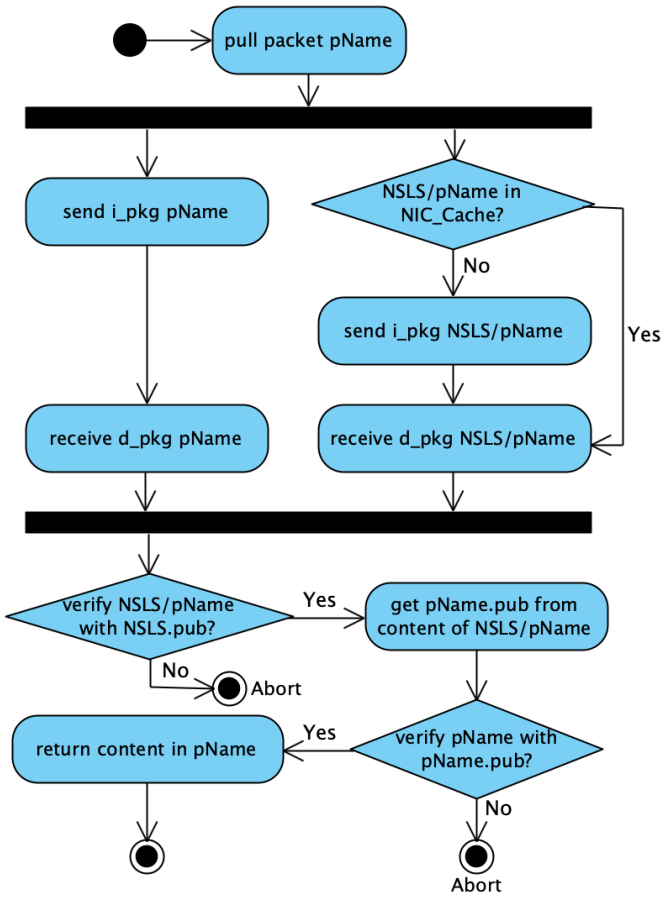

Fig. 4. Name-Signature Lookup Protocol (NSLP) procedure

B. Overheads

Despite confounding factors such as the scattered distribution of data packets, utilization of Content Store and link-layer latency, the overheads can be analyzed by the number of interest packets sent out. The NSLP constantly require two interest packets for a genuine named data packet. However, normal NDN trust management involves a hierarchical key locator. The exact count of interest packets is proportional to the count of naming domain. Take *cs/coursework/user1* for instance. Regarding *cs* as the trust anchor, section *coursework* and publisher *user1* respectively maintain their own key pairs. In other words, the length of the specific key locator is 3.

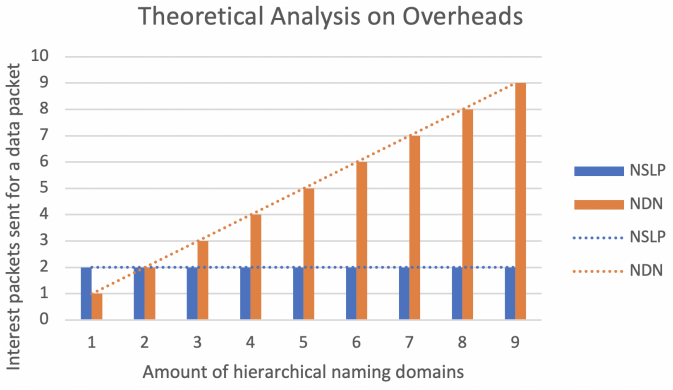

Fig. 5. Theoretical analysis on overheads

It is explicit to conclude from figure 5 that, NSLP has an advantage on overheads over NDN. It is also non-trivial to optimize the naming scheme to avoid unnecessary overheads caused by sophisticated hierarchical trust management scheme.

VI. Further Discussion

A. Scalability

The NSLP can be regarded as an enhanced feature for NDN security. Supporting and running the protocol does not affect the normal NDN procedure. NSLP-supported NIC can be an advanced option for security-sensitive use when building up a PC or server. In the other end, a single centralized server can be scaled up to a cluster of servers. Hierarchical distributed servers such as DNS can be deployed across a nation or the world depending on service type.

B. Compatibility

No alteration of a normal NDN packet is done in order to implement NSLP. In other words, NSLP is entirely transparent to NDN. There are merely NDN interest packets and NDN data packets in the NDN network while NSLS is in service. The existence of NSLP do not affect the normal NDN procedure. Correspondingly, NSLP stack identifies every NSLP process according to packet names.

C. Extension

1) Timeout: A timeout scheme is necessary to be set up since a pair of interest packets are sent out in parallel. There would be no response if an NSLS packet failed to find a matching NSLS entry. Thus, it is confusing to determine whether the NSLS packet is still on its way or already exhausted to find the corresponding record. The actual value of timeout is rather flexible to set in terms of different scenarios.

2) Working modes: A set of parameters can be adjustable in order to fit the diverse need of NSLP.

- parallel/NSLS-first: parallel mode gain higher efficiency while NSLS-first mode does not even make an interest packet out if the name to pull is not registered in NSLS.
- top-sec/up-to-date/cacheable: top-sec mode doesn't allow any NSLS data packet pulled from a CS in routers and even remove local cache in NIC to achieve maximum security. Reversely any NSLS data packet tagged as *top-sec* cannot be cached by the router [11]. Cacheable mode enabling cache in routers is comparatively the most efficient but least secure mode against top-sec mode. Up-to-date mode ignores local cache in NIC in order to always get the *freshest* NSLS data packet.

D. Limitation

The proposed solution can take effect only with the existence of an NSLS server (cluster) and *NSLP-enabled* NIC cards. For the semantically-centralized server, an organization must manage to maintain the server. Moreover, the deputy organization have to keep in tight contact with the NIC manufacturer so as to support the protocol with its public key. Otherwise, it would be sophisticated and potentially risky to update a NIC card to support a specific NSLS server. Thus, to realize the maximum utilization of the NSLS, the ideal situation would be the NIC manufacturer running and maintaining the NSLS servers.

E. Potential Hazards

The only existing security threat would be the leakage of the private key of the NSLS server. The private key in working status should be kept at the top level of defence. Thus, we propose always having a backup pair of private key when using asymmetric signature algorithm. The corresponding backup public key should be embedded in the NIC as well while the backup private key should be kept separately from the primary private key but still in top standard defence. Besides, it is true that there is still the possibility of cracking the asymmetric signature algorithms such as RSA. However, more robust signature algorithms with compatible interfaces can be substituted if necessary. It is reasonable to release new NIC with an up-to-date signature algorithm for an epoch.

VII. Conclusion

In this paper, we described the design of NSLP and NSLS which provide an extra layer of verification in cooperation with network hardware. Details of implementation are needed for further research. So far it has been a conceptual model which theoretically resolved remaining attacks related to authenticity and integrity.

The focus of this paper has always been on an extreme case, which is, what if a malicious node can intercept and fabricate NDN packets and a consumer node just joined and forwarded every interest packet through it? The situation is not pervasive but also non-trivial. The solution to this case highly relies on the robustness of RSA or any other asymmetric signature algorithm. NSLP remains robust and effective as long as the signature algorithm applied remains uncompromised.

In summary, with a semantically-centralized NSLS server and NIC hardware support, NSLP ultimately eliminates impact of any potential spoofing attacks such as MITM. Moreover, it follows the standardized NDN procedure, which indicates high compatibility and resiliency to existing NDN architecture. The NSLP works transparently to the user as long as an NSLP-supported NIC is equipped.

VIII. Acknowledgement

The authors would like to extend sincere thanks to University of Nottingham Ningbo China for supporting this research project under Faculty Inspiration Grant (I01190900047).

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang and B. Zhang, "Named data networking", ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73, July 2014.

[2] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev and L. Zhang, "An Overview of Security Support in Named Data Networking", IEEE Communications Magazine, vol. 56, no. 11, pp. 62-68, November 2018.

[3] B. Hamdane, A. Serhrouchni, A. Fadlallah and S. G. E. Fatmi, "Named-Data security scheme for Named Data Networking", In proceedings of the Third International Conference on The Network of the Future (NOF), pp. 1-6, 2012.

[4] C. Ghali, G. Tsudik and E. Uzun, "Network-Layer Trust in Named-Data Networking", ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 12-19, October 2014.

[5] Y. Yu, A. Afanasyev, D. Clark, K. Claffy, V. Jacobson and L. Zhang, "Schematizing Trust in Named Data Networking", In proceedings of the 2nd ACM Conference on Information-Centric Networking, pp. 177-186, 2015.

[6] B. Bhushan, G. Sahoo and A. K. Rai, "Man-in-the-middle attack in wireless and computer networking — A review", In proceedings of the 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA), pp. 1-6, 2017.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, vol. 21, no. 2, pp. 120–126, February 1978.

[8] P. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, November 1987.

[9] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin and L. Zhang, "NDNS: A DNS-Like Name Service for NDN", In proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1-9, 2017.

[10] G. A. Kildall, CP/M 1.1 or 1.2 BIOS and BDOS for Lawrence Livermore Laboratories, June 1975.

[11] Q. Li, X. Zhang, Q. Zheng, R. Sandhu and X. Fu, "LIVE: Lightweight Integrity Verification and Content Access Control for Named Data Networking", IEEE Transactions on Information Forensics and Security, vol. 10, no. 2, pp. 308-320, February 2015.

[12] "SM2 Elliptic Curve Public-Key Cryptography Algorithm", GM/T 0003-2012.

[13] "SM3 Cryptographic Hash Algorithm", GM/T 0004-2012.