NDNCERT: Universal Usable Trust Management for NDN

Zhiyi Zhang
UCLA
zhiyi@cs.ucla.edu

Alexander Afanasyev
Florida International University
aa@cs.fiu.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

The Named Data Networking (NDN) architecture builds the security primitives into the network layer: all retrieved data packets must be signed to ensure their integrity, authenticity, and provenance. To ensure that these primitives are used in a meaningful way without imposing undue burdens on NDN users, the management of cryptographic keys and certificates needs to work in a simple, secure, and user-friendly way. This poster introduces the NDN Trust Management system (NDNCERT) which is designed to fill this need. NDNCERT provides flexible mechanisms to delegate trust between certificates, either within a single device (managing permissions for local applications on a node to operate under a given namespace) or across devices/entities. NDNCERT features a modular design for security challenges that establish trust through out-of-band means for certificate issuing. Once a node or an application obtains a valid certificate for its namespace (or being configured with a self-signed certificate), it automatically becomes a certificate authority for its namespace, and can use the same NDNCERT protocol to produce certificates for the sub-namespaces.

CCS CONCEPTS

• **Security and privacy → Security protocols**; • **Networks →** *Naming and addressing*;

KEYWORDS

NDN, trust management, certificate

ACM Reference format:
Zhiyi Zhang, Alexander Afanasyev, and Lixia Zhang. 2017. NDNCERT: Universal Usable Trust Management for NDN. In *Proceedings of ICN '17, Berlin, Germany, September 26–28, 2017,* 2 pages.
DOI: 10.1145/3125719.3132090

1 INTRODUCTION

As a proposed Internet architecture, Named Data Networking (NDN) [8] builds the data-centric security into the network layer. All NDN data packets are cryptographically signed at the time of creation, providing integrity and provenance properties regardless where they are stored or how they are retrieved. This requires that each entity in an NDN network possesses proper cryptographic keys and certificates that are signed by global or local trust authorities (or multiple authorities). Therefore, NDN necessitates an easy-to-use, user-friendly system to manage all the certificates.

ICN '17, Berlin, Germany

DOI: 10.1145/3125719.3132090

In this poster, we introduce NDNCERT, the automated and flexible trust management protocol for NDN. NDNCERT defines specialized naming conventions and interest/data exchanges ("`/<prefix>/CA/_<CommandType>`") that empower any namespace owner[1] to easily and securely delegate sub-namespaces and sign the corresponding certificates, and offer ability for users and applications to apply for and obtain such certificates (Figure 1).

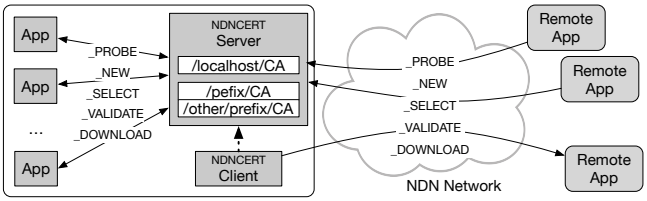

Figure 1: The structure of NDNCERT

The automated certificate management in NDNCERT is based on a set of flexible *security challenges* that rely on out-of-band mechanisms to establish trust relations between local and remote applications and entities. This process is conceptually similar to the automatic certificate management environment (ACME) [4], an inspiration for the NDNCERT, but uses different bases for trust decisions. In the inter-node trust management—between apps on different devices and between different users on the same device—the challenges require proof of control or possession of an externally verifiable element, e.g., an email address or a valid certificate in another namespace. In the intra-node trust management—between local applications—the challenges ensure that the request is received from the legitimate application and from the correct application instance, e.g., through a two-way PIN verification mechanism.

We have implemented an initial prototype of NDNCERT as a generic C++ library and a set of command-line tools [1, 9], and started initial tests of NDNCERT on the NDN testbed.

2 NDNCERT DESIGN

Each node and an application in an NDN network possesses one or multiple certificates. Following the concept proposed in [5], we develop NDNCERT to enable localized trust management—each namespace can have its own trust anchor, that can be signed either by higher level authorities, or by peers to establish trust across namespaces.

With NDNCERT, any node can easily become a certificate authority for the namespace, which is either delegated to this node by a higher-level CA (e.g. "`/ndn`" ⇒ "`/ndn/edu/ucla`") or self-claimed (self-signed trust anchor), e.g., when being used in local environments such as a smart home. NDN's *hierarchical naming structure*

[1] A namespace owner is an entity that possesses the private key and certificate that is allowed by the application's trust schema [7] to publish data under the given namespace.

and well established *naming conventions* are the main contributors to simplicity in certificate management. For example, to request a certificate from the NDN Testbed CA, one simply needs to send a specially formatted Interest packet that starts with "`/ndn/CA`"; for certificate from UCLA site of NDN Testbed, send to "`/ndn/edu/ucla/CA`", etc. Similarly, to become a CA for a local "`/prefix`", a node N simply starts a process that registers "`/prefix/CA`" with its local NDN daemon. Note that for N to become a *recognized* CA, its own certificate must be trusted by other parties in the network. This can be achieved if N's certificate is issued by a recognized higher-level authority, or by others' endorsements of N's self-signed certificate.

The foremost security goal of NDNCERT is the integrity of the whole process. Therefore, NDNCERT requires all interest/data packets must be cryptographically signed by senders and verified by receivers to prevent messages from being altered. Because of the timestamp and the random nonce in each signed interest/data packet, replay attacks can also be prevented. To prevent eavesdropping, all sensitive information (password, PIN code and etc.) must be encrypted in the challenge stage.

The security properties of NDN certificates obtained using the NDNCERT protocol come from the out-of-band validation mechanisms, through so-called *security challenges*. With these challenges, a requester proves to a CA, and others who trust the CA's judgement, that it is a legitimate party to request certificate in a given namespace.

To obtain and to issue certificates, NDNCERT protocol defines the following five steps of interest/data exchanges (Figure 2): discovery of available sub-namespaces (optional "`_PROBE`"), application for the NDN certificate for the assigned/selected namespace ("`_NEW`"), selection an out-of-band challenge to prove legitimacy/ownerships of the namespace ("`_SELECT`"), verification the selected challenge ("`_VALIDATE`"), and checking status and downloading the issued certificate ("`_STATUS`" and "`_DOWNLOAD`").

Figure 2: NDNCERT Protocol Overview

So far, we have defined several challenges that can be used by individual CAs as grounds for certificate approval: email-based, certificate-based, simple PIN, and two-way PIN challenges.

The email-based challenge relies on proof-of-control of an email account: as long as the requester can receive a secret PIN code communicated to the specified email, he/she can be approved for the namespace following CA's rule. *The certificate-based challenge* requires the requester to encrypt a nonce provided by CA with the corresponding private key along with the supplied certificate issued by another CA. The CA will first verify the certificate and then use extracted public key to decrypt the nonce. *The simple PIN challenge* requires an out-of-band secure communication between the requester and CA. After receiving the request, CA simply generates a random PIN, which needs to be delivered to the requester, e.g., in person. *The two-way PIN challenge* is specifically designed to automate intra-node trust decision. This challenge allows users to ensure that the certificate request is coming from a legitimate application (NDNCERT CA verifies that an application is signed by a trusted developer) and from the correct instance of the application through (visual) matching of the PIN code generated by the application and shown in the CA.

3 FUTURE WORK

NDNCERT aims to simplify the work of NDN users and application developers by making trust management flexible, easy, and user-friendly. As the next step, we plan to integrate the client and server parts of the NDNCERT into NDN Control Center [2] and NDN Android [3] to promote a wide adoption and use of the system. We will also look into several aspects of advancing the NDNCERT system. We plan on investigating other types of authentication challenges that can better suite the certificate management in IoT environments [6]. Constrained IoT controllers usually have no user interfaces and require special handling in identity assignment, e.g., leveraging physical possessions of the devices, or physical proximities through light/audio sensors, accelerometers, etc. Furthermore, we plan to integrate NDNCERT as part of the trust schema [7] to automatically generate (request) certificates to sign data packets.

ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation under award CNS-1345318 and CNS-1629922.

REFERENCES

[1] 2017. Codebase of NDN Certificate Management Protocol (NDNCERT). https://github.com/named-data/ndncert. (2017).
[2] 2017. NDN Control Center. https://named-data.net/codebase/applications/ndn-control-center/. (2017).
[3] 2017. NDN on Android. https://github.com/named-data-mobile/NFD-android. (2017).
[4] R. Barnes and others. 2017. Automatic Certificate Management Environment (ACME). Internet Draft, draft-ietf-acme-acme-06. (2017).
[5] Ronald L Rivest and Butler Lampson. 1996. SDSI-a simple distributed security infrastructure.
[6] Wentao Shang and others. 2016. Named data networking of things. In *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*. IEEE, 117–128.
[7] Yingdi Yu, Alexander Afanasyev, David Clark, kc claffy, Van Jacobson, and Lixia Zhang. 2015. Schematizing Trust in Named Data Networking. In *Proc. of ACM ICN*.
[8] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, and others. 2014. Named data networking. *ACM SIGCOMM Comp. Comm. Review* (2014).
[9] Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, and Lixia Zhang. 2017. *NDN Certificate Management Protocol (NDNCERT).* Technical Report NDN-0054. NDN.