NDNCERT Protocol 0.3

rev 8a6e75c

Jump to bottom

Zhiyi Zhang edited this page 1 hour ago · 7 revisions

NDN Certificate Management (NDNCERT) Protocol v0.3

Authors

- Zhiyi Zhang (zhiyi@cs.ucla.edu)

Acknowledgement

- Junxiao Shi (junxiao.shi@nist.gov)
- Davide Pesavento (davide.pesavento@nist.gov)
- Members from NDN Dev Call

Note

- NDNCERT protocol v0.3 is for future use and has not been implemented nor deployed. It may subject to future changes after peer reviews.

Changes

Major Changes

- Remove DOWNLOAD phase
- Remove `_` from `_PROBE`, `_NEW`, `_CHALLENGE`
- Separate INFO out of PROBE phase
- Add Redirection extension to PROBE phase to improve scalability and usability
- PROBE phase become purely informational and there is no more bindings between PROBE and later phases
- CA can allow its users to get multiple certificates under the same identity name for different keys
- CA can allow its users to get certificates of a longer name (sub namespace) than the designated identity name

Minor Changes

- Use TLV to replace JSON text used in PROBE, NEW, and CHALLENGE phases
- Use uncompressed ECHD public key in NEW phase

ECDH

Terminologies

- Issuer or Certificate Authority (CA). Issuer and CA are used interchangeably in this document. An issuer or a CA is the party who owns a namespace and can issue certificates to requesters who want to get a sub namespace from the issuer/CA.
- Requester or Client. Requester and client are used interchangeably in this document. A requester or a client is the party who wants to get a sub namespace and its corresponding certificate from an Issuer.
- `<variable>` represents one or more name components in NDN name. For example, `/<ca-prefix>/CA/INFO` refers to `/ndn/edu/ucla/CA/INFO` when CA's name is `/ndn/edu/ucla`.

- `<timestamp>` is one name component which contains the timestamp when the packet is generated. Such a component is a `Generic name component` containing the POSIX time (in second) before Timestamp Name Component is available or a `Timestamp name component` when `Timestamp name component` becomes officially available.
- `<ApplicationParameters_Digest>` is one name component as defined in ndn name.
- `<Request-ID>` is one name component containing a unique ID to identify the application/renewal/revocation request.
- Signed Interest or Interest Signature. All signed Interest packets appear in this document is supposed to follow the format of signed Interest defined in the latest release version of ndn-cxx.
- TLV encoding. In this document, the TLV encoding of integer, string, and bytes all follow NDN TLV encoding.

always use TimeStamp Name Component, it's available in protocal

You are defining protocal, not implementation.

1. Overview

In Named Data Networking (NDN), to generate Data packets with legitimate names and verifiable signatures, an application (producer) needs to obtain a name and an associated certificate for that name. The certificate application can either be accomplished manually or through automated means. NDN certificate management protocol (NDNCERT) aims to enable automatic certificate management in NDN, including

- certificate application,
- certificate renewal
- certificate revocation

All aforementioned management operations will require certain out-of-band or in-band identity verification means.

Furthermore, NDNCERT allows a namespace owner to easily manage its sub-namespaces and corresponding certificates by

- becoming a certificate issuer for the parent namespace

or

- applying for certificates for valid sub namespaces from the same issuer as the one who issued the parent namespace to that entity.

For example, with NDNCERT, Alice (as a requester) can get a namespace `/ndn/edu/ucla/alice` from the issuer `/ndn/edu/ucla` after she successfully proves her identity. After that, by utilizing NDNCERT protocol, Alice can become an issuer for namespace `/ndn/edu/ucla/alice` and issue names/certificates to her devices, e.g., designating name `/ndn/edu/ucla/alice/working-laptop` to her laptop. If issuer `/ndn/edu/ucla` allows, Alice can also directly apply for namespace `/ndn/edu/ucla/alice/working-laptop` for her laptop directly from `/ndn/edu/ucla`.

Note that NDNCERT does not impose any specific trust model or trust anchors.

2. Packet Specification

2.1 INFO phase

State the risk for downloading CA profile from network instead of distributing offline.

2.1.1 Introduction

INFO phase is for a requester to download the profile of a CA. A profile file contains this CA's requirement on name assignment, supported challenges, CA's certificate, etc.

The profile is needed for all the requesters who want to get/renew/revoke certificates from the CA.

2.1.2 Packet Format

Interest format:

Field	Description
Name	`/<CA-Prefix>/CA/INFO`
Can Be Prefix	True
Must Be Fresh	Not required
Signature	Not required

Data format:

Field	Description
Name	`/<CA-Prefix>/CA/INFO/<timestamp>`
Content	TLV of CA's profile
Signature	Signed by CA's identity key

Why not version + segment?

2.1.3 CA Profile

Why not Name in TLV?

The attributes in a CA profile carried by INFO Data packet contains:

- `ca-prefix` , string value, the NDN name of the CA. This name should be reachable to requesters.
- `ca-info` , string value, a brief introduction of the CA.
- `probe` , string value in format of `attribute_1:attribute_2:...:attribute_n` , a list of attributes required by the CA to identify the name for a requester in the PROBE phase.
- `certificate` , bytes value, TLV of the CA's certificate.
- (optional) `probe-encryption-key` , string value, DER encoded RSA public key. This key is used to encrypt values of attributes in the PROBE phase.

Why not multiple "probe" TLV?

Do you mean RSA-OAEP?

@TODO Zhiyi: need discussion. I added the probe-encryption-key to improve privacy because probe info may contain email, UUID, etc., potentially raising privacy issues.

Why not establish session key during probe? RSA-OAEP has no forward secrecy.

An example

```
T:Content, L, V:
  T:ca-prefix, L, V:"/ndn/CA"
  T:ca-info, L, V:"NDN Testbed CA"
  T:probe, L, V:"email:full-name",
  T:certificate, L, V:...
```

T: probe, L, V: "email",
T: probe, L, V: "full-name",
or DescriptionEntry as in certV2

2.1.4 TLV Type Number

Does this follow evolvability guidelines? Why or why not?

Attribute	TLV Type Number
ca-prefix	128
ca-info	129
probe	130

Attribute	TLV Type Number
certificate	131
probe-encryption-key	132

2.2 PROBE phase

2.2.1 Introduction

PROBE is used by the requester to know which name is legitimate with respect to user's identity information. PROBE is useful when the CA needs to keep an association between a sub namespace with the identify of the owner of this sub namespace.

2.2.2 Packet Format

Interest format:

This is called ParametersSha256Digest Component, see protocol

Field	Description
Name	/<CA-Prefix>/CA/PROBE/<ApplicationParameter_Digest>
ApplicationParameters	string in the format of value_1:value_2:...:value_n
Can Be Prefix	False
Must Be Fresh	True
Signature	Not required

Why not TLV? Consider value_n can be binary. (eg. photo)

Data format:

Field	Description
Name	/<CA-Prefix>/CA/PROBE/<ApplicationParameter_Digest>
Content	Name TLV
Signature	Signed by CA's identity key

An example.

```
Interest:
Name: /<CA-Prefix>/CA/PROBE/<ParameterDigest>
ApplicationParameters:
{
  "zhiyi@cs.ucla.edu:zhiyi zhang"
}

Data:
Name: /<CA-Prefix>/CA/PROBE/9A39DC3...
Content:
{
  T:Name, L, V:/ndn/edu/ucla/zhiyi@cs.ucla.edu
}
Signature
```

T: email, L, V: "zhiyi@cs.ucla.edu"
T: FullName, L, V: "zhiyi zhang"

Can CA give out multiple potential names for requester to choose from?

2.2.3 PROBE Extension for Redirection

PROBE can be used for a root CA to redirect its requestors to sub CAs. This extension helps to reduce the workload of the root CA and reduce the out-of-band configuration on requesters side, thus improving system scalability and usability.

In PROBE redirection extension, the Interest packet is the same as the one shown in 2.2.2. However, the Data packet reply has a different content format.

If Alice decides to become sub-CA, how can she inform the parent CA to redirect requests under /ndn/edu/ucla/alice to her sub-CA?

Data format used in PROBE redirection extension:

Field	Description
Name	`/<CA-Prefix>/CA/PROBE/<ApplicationParameter_Digest>`
Content	TLVs of sub CA's information
Signature	Signed by CA's identity key

To be more specific, the content carries following information.

- `ca-prefix` , bytes value, the name TLV of sub CA's prefix.
- `digest-of-cert` , bytes value, the digest of sub CA's certificate.

What digest algorithm?

2.2.4 TLV Type Number

Attribute	TLV Type Number
ca-prefix	128
digest-of-cert	146

2.3 NEW phase

2.3.1 Introduction

NEW is for requester to formally start a certificate application process. The CA will also start to keep the state of the requester.

not requester state, but request state

2.3.2 Packet Format

Interest format:

Field	Description
Name	`/<CA-prefix>/CA/NEW/<ApplicationParameters_Digest>`
ApplicationParameters	TLV format of value as defined in 2.3.3
Can Be Prefix	**False**
Must Be Fresh	**True**
Signature	Signed by the private key whose public key is going to be certified by the CA

Data format:

Field	Description

Field	Description
Name	`/<CA-prefix>/CA/NEW/<ApplicationParameters_Digest>`
Content	TLV format of value as defined in 2.3.3
Signature	Signed by CA's identity key

2.3.3 Interest ApplicationParameters and Data Content

NEW Interest ApplicationParameters field carries following information.

- `ecdh-pub` , bytes value, requester's ECC public key used for Elliptic-Curve Diffie-Hellman key agreement. The key should be encoded as its raw format without compression. Such a ECC public key should be generated with cryptographically secure pseudo random generator for every NDNCERT session.
- `cert-request` , bytes value, the TLV of a self-signed certificate generated by the requester. Such a certificate follows the certificate format defined in ndn certificate.

Importantly, the requester can define the desired validity time of its certificate which will be issued by the CA later. To do this, the requester should specify the validity period of the self-signed certificate in the corresponding field as defined in ndn certificate.

NEW Data content field carries following information.

- `ecdh-pub` , bytes value, the issuer's ECC public key used for Elliptic-Curve Diffie-Hellman key agreement. The key should be encoded as its raw format without compression. Such a ECC public key should be generated with cryptographically secure pseudo random generator for every NDNCERT session.
- `salt` , bytes value, 64 bits or longer random number.
- `request-id` , bytes value, unique ID assignment for this request instance.
- `status` , int value, the application status code.
  - STATUS_BEFORE_CHALLENGE = 0,
  - STATUS_CHALLENGE = 1,
  - STATUS_PENDING = 2,
  - STATUS_SUCCESS = 3,
  - STATUS_FAILURE = 4,
  - STATUS_NOT_STARTED = 5
- `challenges` , bytes value, a list of TLV format challenges from which the requester can select. The child attribute name is `challenge-id` with a string value. E.g., `T:challenge-id, L, V:"Email"` .

2.3.4 TLV Type Number

Attribute	TLV Type Number
ecdh-pub	133
cert-request	134
salt	135
request-id	136

Attribute	TLV Type Number
status	137
challenges	138
challenge-id	139

2.4 RENEW phase

Why separate command?
NEW with old key ⇒ renew

2.4.1 Introduction

RENEW is for a requester to renew its certificate with the CA. Regarding the format, the Interest and Data format in RENEW are almost identical to NEW.

or: NEW with old CA-issued cert (not self-signed) ⇒ renew

2.4.2 Packet Format

Interest format:

Field	Description
Name	/<CA-prefix>/CA/RENEW/<ApplicationParameters_Digest>
ApplicationParameters	TLV format of value as defined in 2.3.3
Can Be Prefix	False
Must Be Fresh	True
Signature	Signed by the private key whose public key is going to be certified by the CA

Data format:

Field	Description
Name	/<CA-prefix>/CA/RENEW/<ApplicationParameters_Digest>
Content	TLV format of value as defined in 2.3.3
Signature	Signed by CA's identity key

2.5 CHALLENGE phase

2.5.1 Introduction

CHALLENGE phase is for the requester to prove his/her identity to the CA. Once approved, the CA will issue the certificate for the requester.

2.5.2 Packet Format

Interest format:

Field	Description
Name	/<CA-prefix>/CA/_CHALLENGE/<Request_ID>/<ApplicationParameters_Digest>
ApplicationParameters	TLV format of value as defined in 2.5.3

Field	Description
Can Be Prefix	False
Must Be Fresh	True
Signature	Signed by the private key whose public key is going to be certified by the CA

Data format:

Field	Description
Name	`/<CA-prefix>/CA/_CHALLENGE/<Request_ID>/<ApplicationParameters_Digest>`
Content	TLV format of value as defined in 2.5.3
Signature	Signed by CA's identity key

2.5.3 Interest ApplicationParameters and Data Content

Where's encryption?

CHALLENGE Interest ApplicationParameters field carries following information.

- `selected-challenge` , string value, the challenge selected by the requester.
- `challenge-parameter` , string value in the format of `parameter_1:parameter_2:...:parameter_n` , other attributes specified by the implementation of the selected challenge.

Why not multiple TLV?
e.g. DescriptionEntry
as in Cert V2 spec

NEW Data content field carries following information.

- `status` : int value, the application status code.
  - STATUS_BEFORE_CHALLENGE = 0,
  - STATUS_CHALLENGE = 1,
  - STATUS_PENDING = 2,
  - STATUS_SUCCESS = 3,
  - STATUS_FAILURE = 4,
  - STATUS_NOT_STARTED = 5

merge with Section 2.3.3, do not repeat

- `challenge-status` , string value, the challenge status code, specified by the selected challenge implementation.
- `remaining-tries` , int value, the remaining times that the requester can send a challenge Interest.
- `remaining-time` , int value, the remaining time for the requester to finish the challenge.

What unit?

- `issued-cert-name` , bytes value, **full** name TLV of the certificate issued by the CA for the requester after the challenge has been successfully accomplished.

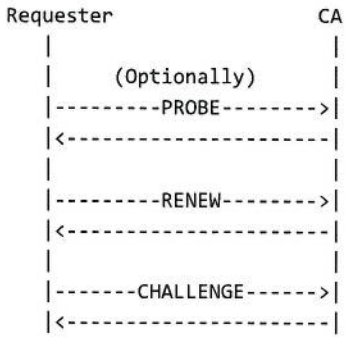
2.5.4 TLV Type Number

Attribute	TLV Type Number
selected-challenge	140
challenge-parameter	141
challenge-status	142
remaining-tries	143

Attribute	TLV Type Number
remaining-time	144
issued-cert-name	145

3. New Certificate Application Protocol

New certificate application contains three steps: PROBE, NEW, and CHALLENGE.



From a requester's perspective:

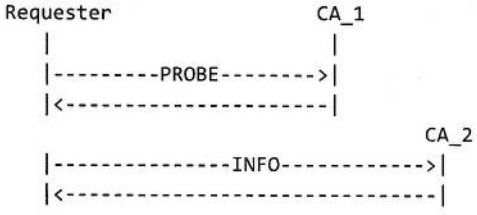
- Optional PROBE. When CA has a name assignment policy, a requester may need the PROBE phase to know the expected name that he/she can obtain based on his/her identity information. Without the PROBE phase, a name request may be rejected by the CA.
- NEW. The requester prepares a pair of asymmetric key (e.g., RSA, ECC), use the private key to sign the public key into a self-signed certificate, and start the application by taking NEW phase.
- CHALLENGE. The requester selects one challenge among available challenges offered by the CA and finish the in-band or out-of-band identity verification. Once the challenge is accomplished, the certificate will be issued.

From a CA's perspective:

- Optional PROBE. When CA has a name assignment policy, the CA needs to explicitly specify the parameters needed for the PROBE in its profile, which can be downloaded through INFO. In the PROBE step, the CA takes the parameters from the requester as input and generate an available name for the requester.
- NEW. The CA verifies the self-signed certificate from the requester and collects all the available challenges back to the requester.
- CHALLENGE. According to the challenge selected by the requester, the CA sets up the challenge and verifies the requester's ownership of the identity.

3.1 Sub CA Redirection

Utilizing PROBE redirection extension, a CA can redirect its requesters to a sub CA.



```
|                                |
|   New Certificate Application  |
```

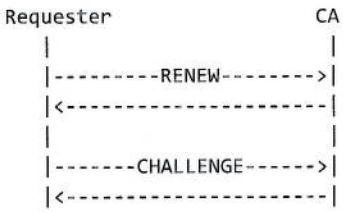
After retrieving `CA_2` 's profile, the requester should

- use the certificate digest obtained from the PROBE Data packet replied by `CA_1` to verify the certificate field in the INFO profile of `CA_2` .
- use the public key in the `CA_2` profile's `certificate` field to verify the signature of the INFO Data packet replied by `CA_2` .

`CA_1` should be aware of the certificate update on `CA_2` so that the digest of `CA_2` 's certificate is always updated.

4. Renewal Protocol

Certificate renewal contains two steps: RENEW and CHALLENGE.



From a requester's perspective: How does this differ from NEW?

- RENEW. The requester prepares a (new) pair of asymmetric key (e.g., RSA, ECC), use the private key to sign the public key into a self-signed certificate, and start the application by taking NEW phase.
- CHALLENGE. The requester selects one challenge among available challenges offered by the CA and finish the in-band or out-of-band identity verification. Once the challenge is accomplished, the certificate will be issued.

From a CA's perspective:

- RENEW. The CA verifies the self-signed certificate from the requester and collects all the available challenges back to the requester.
- CHALLENGE. According to the challenge selected by the requester, the CA sets up the challenge and verifies
  - the requester already owns a certificate issued by the CA or
  - the requester's ownership of the identity.

5. Revocation Protocol

Certificate revocation can be triggered by

- The CA who has issued the certificate
- The owner of the certificate
- Any one who can prove the ownership of the private key that is corresponding to the public key in the certificate.