NDN, Technical Report NDN-0050. http://named-data.net/techreports.html
Revision 1: April 29, 2017

NDN Certificate Management Protocol (NDNCERT)

Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, Lixia Zhang

Abstract

NDN certificate management protocol (NDNCERT) enables automatic certificate management in NDN. In Named Data Networking (NDN), every entity should have corresponding identity (namespace) and the corresponding certificate for this namespace. Moreover, entities need simple mechanisms to manage sub-identities and their certificates. NDNCERT provides flexible mechanisms to request certificate from a certificate authority(CA) and, as soon as certificate is obtained, mechanisms to issue and manage certificates in the designated namespace. Note that NDNCERT does not impose any specific trust model or trust anchors. While the primary use case of the developed protocol is to manage NDN testbed certificates, it can be used with any other set of global and local trust anchors.

This specification provides details and packet formats to request certificates, create certificates after one of the validation mechanism, and how the issued certificate is retrieved by the original requester.

1. Introduction

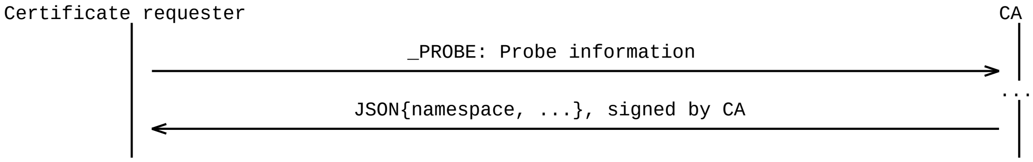
NDNCERT is specifically designed to foster simple yet secure NDN certificate issuance and management on NDN testbed, client machines connected to the testbed, and any other computers that speak NDN. In particular, NDNCERT provides flexible mechanisms to establish certificate authorities (CA) for different NDN namespaces (e.g., NDN Testbed CA for certificates in "`/ndn namespace`", NDN OpenMHealth authority for "`/org/openmhealth`", and others) and to request certificates from the established authorities. Note that the developed NDNCERT protocol does not impose any specific trust model or trust anchors

With NDNCERT, any node can become a certificate authority for a namespace, which is either delegated to this node by a higher-level CA ("`ndn`" -> "`/ndn/edu/ucla`") or self-claimed (self-signed trust anchor), e.g., when using in local environments such as smart homes. These hierarchical names are the main contributors to simplicity to request a certificate. For example, to request a certificate from an NDN Testbed CA, one just needs to send a specially formatted Interest 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 "`/<prefix>`", a node simply needs to start a process that registers "`/<prefix>/CA`" with local NFD. Note that for the node to become a **recognized** CA, its own certificate needs to be trusted by other parties in the network: implicitly if it is issued by a higher-level authority or explicitly for self-signed certificate.

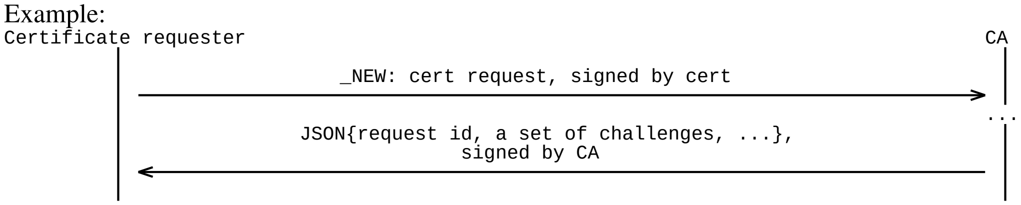
In NDN any node can take a role of CA, managing certificates in the designated namespace. There are mainly two kinds of CA: CA using a self-signed local trust anchor or a CA using delegated certificate for the namespace.

The following diagram illustrates the main process for CA to perform certificate signing. There are mainly three processes in certificate application.

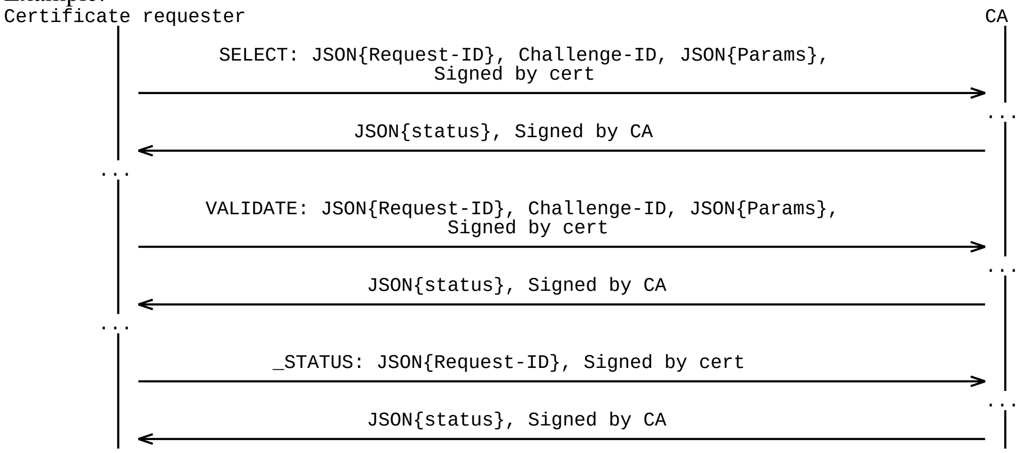
- **Sub namespace assignment**
  The certificate requester first sends an interest called "`_PROBE`" to ask for an available namespace from CA. The CA will reply an available namespace to certificate requester. Notice that whether the certificate requester should perform **sub namespace assignment** step or not depends on CA's policy, which means certificate requester should know CA's policy first. Those CAs for a big number of identities may want to use the sub namespace assignment to avoid name conflicts.
  Requester sends a "`_PROBE`" interest to CA to get itself a namespace. If user wants to get an available namespace, the probe parameter should be set as CA requires. For example, CA may require user to use email address as probe parameter. CA will reply a data packet which contains a JSON file, in which a namespace assigned by CA and available CA information will be included.
  Example:



- **New certificate application**
  With the namespace got from last step, the certificate requester then generate an asymmetric key pairs (user certificate key) with the name and send a NEW interest to CA to apply for a certificate. The NEW interest contains a self-signed certificate (certificate request). When CA receives the NEW command, the CA will collect all available challenge choices according its own policy and reply this challenge list and a reqeust ID to certificate requester.
  Notice that CA will keep certificate requester's request instance at this time point and generate a unique ID for request. After NEW interest, user only need to put this ID to specific interest component and sign the interest with the certificate corresponding key.
  User sends a "_NEW" interest to CA to request a new certificate. The interest command will carry a certificate request (self-signed certificate). CA will answer a data packet which contains a JSON file. A set of validation challenge choices is included. Also CA will store the request instance and put a Request-ID to the json file.
  Example:


Certificate requester
CA
_NEW: cert request, signed by cert
...
JSON{request id, a set of challenges, ...},
signed by CA

- **Challenge selection and validation**
  Certificate requester then selects one challenge and notices CA. CA will perform the challenge for requester. Requester would then follow CA's instructions to finish the challenge. CA will query the out-of-band validation results and answer a data which contains the validation status. Once the challenge result is verified by CA, CA will sign the certificate. To verify the user's identity, at least one **out-of-band validation** is required. For example, email verification code to the requester, or ask requester to obtain secret code out of bind.
  User sends a "_SELECT" **command interest** to CA to inform which challenge have been selected and will be performed. User sends a "_VALIDATE" **command interest** to CA to pass required information to CA to finish the challenge. Challenges should take use of the Challenge-Info component in "_SELECT" and "_VALIDATE". For example, for email challenge, CA can require user to provide email address in this field. CA will reply user with a data packet which contains a JSON file, in which certificate issuing status and challenge validation status will be included. If the challenge is verified successfully, a certificate download name will be provided too.
  Example:


Certificate requester
CA
SELECT: JSON{Request-ID}, Challenge-ID, JSON{Params},
Signed by cert
...
JSON{status}, Signed by CA
...
VALIDATE: JSON{Request-ID}, Challenge-ID, JSON{Params},
Signed by cert
...
JSON{status}, Signed by CA
...
_STATUS: JSON{Request-ID}, Signed by cert
...
JSON{status}, Signed by CA

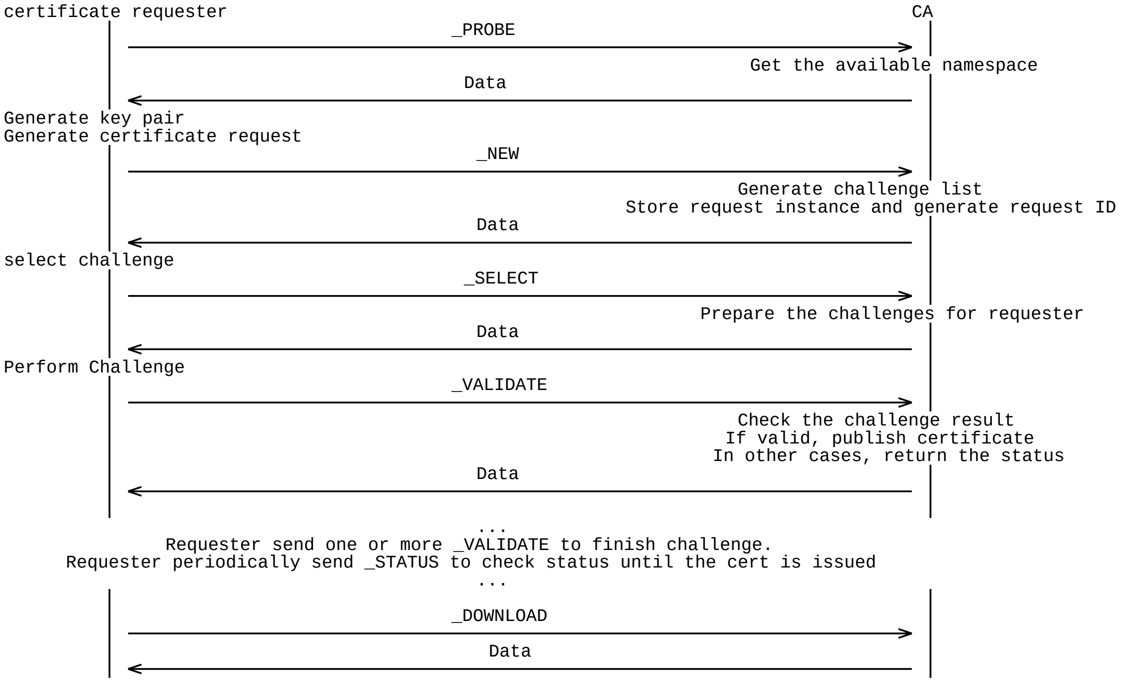
- **Cerificate download**
  Once certificate requester knows the certificate has been signed by CA. Requester can fetch the certificate using the name provided in the data from CA.
  Requester sends "_DOWNLOAD" to fetch the encapsulated certifciate.

Certificate renew is the same as the certificate application. Besides using "_NEW" interest to renew certificate, CA can define other strategy for renewal. For root CA like NDN testbed, it requires end entities to use email address for challenge. In this case, NDN testbed CA can use email to notice end entities whose certificates will expire and provide a button for renewal in email, so that there is no need for user to send NEW interest.

1..1 Concrete Example

As a concrete example, a certificate requester want to apply for a namespace under "`/ndn/CA`".

1) The certificate requester first select NDN testbed CA module as the target CA module. Once the module is selected, the certificate requester gets CA's policy for certificate application. The policy requires that user should use "`_PROBE`" before NEW and NDN testbed CA only accepts email challenge. In the policy, it is also required that email address should be used as the "`_PROBE`" parameter.
2) Certificate requester first send a "`_PROBE`" interest, which contains its email address "`zhiyi@cs.ucla.edu`".
3) Then certificate requester would get the data back. After validating the data signature, from data content, certificate requester gets an available namespace "`/ndn/edu/ucla/zhiyi@cs.ucla.edu`".
4) With the namespace, certificate requester then generates an asymmetric key pair "`/ndn/edu/ucla/zhiyi@cs.ucla.edu/KEY/[key-id]`" and a certificate request (a special formatted self-signed certificate) "`/ndn/edu/ucla/zhiyi@cs.ucla.edu/KEY/[key-id]/self/[cert-version]`" as the certificate request.
5) After that, certificate requester sends the certificate request to NDN testbed CA through "`_NEW`" interest and fetch a Data packet back. After validating the signature of data, requester gets a list of available challenges. In this NDN testbed case, there is only email challenge in the list.
6) Certificate requester then send a "`_SELECT`" interest command noticine CA that Email challenge is selected. In "`_SELECT`" interest, certificate requester puts its email address "`zhiyi@cs.ucla.edu`" to challenge-info part. CA will send a challenge email to this address. Once certificate requester obtains the verification code from the challenge email, requester then sends a "`_VALIDATE`" command to finish the challenge. "`_VALIDATE`" command carries the verification code and signature from requester.
7) If the status in response for the last "`_VALIDATE`" interest is "pending", certificate requester can periodically send "`_STATUS`" interest to check issuing status.
8) Once the status shows "issued", cerificate requester can download the certificate by sending a "`DOWNLOAD`" interest to CA.


certificate requester
CA
_PROBE
Get the available namespace
Data
Generate key pair
Generate certificate request
_NEW
Generate challenge list
Store request instance and generate request ID
Data
select challenge
_SELECT
Prepare the challenges for requester
Data
Perform Challenge
_VALIDATE
Check the challenge result
If valid, publish certificate
In other cases, return the status
Data
...
Requester send one or more _VALIDATE to finish challenge.
Requester periodically send _STATUS to check status until the cert is issued
...
_DOWNLOAD
Data

2. NDNCERT Message Formats

The packet formats in NDNCERT.

2..1 Assign namespace packet exchange (optional)

2..1.1 _PROBE interest packet

- Interest command parameters:
  - Command Type "_PROBE"
  - Probe information
- MustBeFresh field must be set
- Verification
  No requirements.
- Interest naming convention
  The interest name format is as follow:

```
/CA-prefix/_PROBE/<Probe Information>
          \      /\                 /
           --  --  -------  --------
             \/           \/
     Command Type     User provides a name or a email address
```

- Example of "_PROBE" interest
  - If /ndn/edu/ucla/ CA module requires user to use email address for probe, a PROBE interest can be:

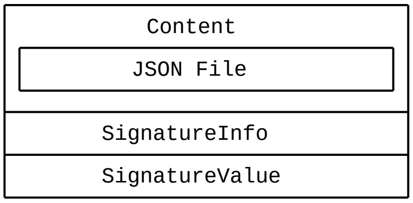
```
/ndn/edu/ucla/_PROBE/zhiyi@cs.ucla.edu
```

  - If /ndn/edu/ucla/cs/zhiyi CA module requires user to use a plain string for probe:

```
/ndn/edu/ucla/cs/zhiyi/_PROBE/my-macbook
```

2..1.2 Data packet answering _PROBE

- Verification
  When received, several parts need to be verified to ensure authentication:
  - The data name is the same with the interest sent
  - CA signature must be verified
- Data structure:



- JSON format
  - Identifier: The namespace assigned by CA to certificate requester based on the hint information
  - CA-info: A link to where CA pubish its statement and policy
- Example of JSON
  The example is as follow:

```
Example #1
{
  "identifier": "/ndn/edu/ucla/cs/zhiyi",
  "ca-info": "/ndn/edu/ucla/cs/www/ca/information.html"
}

Example #2
{
  "identifier": "/ndn/edu/ucla/cs/zhiyi/my-macbook”,
  "ca-info": ""
}
```

2..2 New certificate application packet exchange

2..2.1 _NEW interest packet

- Interest parameters
  - Command Type "`_NEW`"
  - User certificate request
  - a signature signed by old certificate private key: For certificate renewal when user wants to change key pair.
- MustBeFresh field must be set
- Interest Signature
  - Signature must be signed by `user certificate key`
- Verification
  When received, several parts need to be verified to ensure authentication:
  - certificate request signature must be verified
  - For certificate renewal and key pair change at same time
    * Valid signature
    * The signature key matches the certificate name
    * The old certificate still not expires
- Interest naming convention
  The interest name format for new certificate and certificate renewal is as follow

```
/CA-prefix/_NEW/<certificate request>/[signature]
          \    /\                    /
           -  -  --------  ---------
            \/           \/
     Command Type  Certificate component
```

- Example of "`_NEW`" interest
  - To apply a certificate from /ndn/edu/ucla/ CA, a NEW interest can be:

```
/ndn/edu/ucla/_NEW/BvODhQdHCANuZG4...xqt5S2+4=/...
```

2..2.2 Data packet answering _NEW - Verification

When received, several parts need to be verified to ensure authentication:

- The data name is the same with the interest sent
- CA signature must be verified

- JSON format
  - status: The application status.
  - request-id: The unique ID of the request
  - challenges: A challenge list from which the certificate requester can select one to go to next step
  - challenge
    * challenge-type: The identifier of the challenge
- Example of JSON
  The example is as follow:

```
Example #1
{
  "status": "pending",
  "request-id": "1209381203",
  "challenges": [
    {
      "challenge-type": "PIN"
    },
    {
      "challenge-type": "Email"
    }
  ]
}
```

2..3 Challenge selection and validation packet exchange

2..3.1 _SELECT interest packet

- Interest command parameters:
  - Command Type "_SELECT"
  - Request ID in JSON format
  - Challenge ID
  - Challenge related information in JSON format (optional, depends on challenge's requirements)
- MustBeFresh field must be set
- Interest Signature
  - Signature must be signed by user certificate key
- Verification
  - Certificate signature must be verified
  - Signature must match the request ID
- Interest naming convention
  The interest name format is as below:

```
/CA-prefix/_SELECT/{"Request-ID": "1209381203"}}/<ChallengeID>/{"Param1": "<param1>", ...}/
↪[Signature components]
          \    /\                            /\                                        /
           -  -  -------------  ------------   ------------------  --------------------
            \/                \/                                 \/
       Command Type       Request ID          Select a challenge and provide related info
```

- Example of "_SELECT" interest
  - If /ndn/edu/ucla/ CA module requires user to use email address for challenge, a SELECT interest can be:

```
/ndn/edu/ucla/_SELECT/{"request-id":"111"}/EMAIL/{"email-address":"zhiyi@cs.ucla.edu"}/...
                                           |        |
                                           v        |
                   Indicates using email challenge  |
                                                    |
                                                    v
                               User's email address to receive challange email
```

  - If /ndn/edu/ucla/cs/zhiyi CA module can accept PIN challenge and PIN challenge require nothing in SELECT, then a SELECT interest can be:

```
/ndn/edu/ucla/cs/zhiyi/_SELECT/{"request-id":"111"}/PIN/...
                                                  |
                                                  v
                        Indicates using PIN code challenge
```

2..3.2 _VALIDATE interest packet

- Interest command parameters:
  - Command Type "_VALIDATE"
  - Request ID in JSON format
  - Challenge ID
  - Challenge related information in JSON format
- MustBeFresh field must be set
- Interest Signature
  - Signature must be signed by user certificate key
- Verification
  When received, several parts need to be verified to ensure authentication:
  - Certificate signature must be verified
  - Signature must match the request ID
- Interest naming convention
  The interest name format is as below:

```
/CA-prefix/_VALIDATE/{"request-id": "1209381203"}/<ChallengeID>/{"param1": "<param1>", ...}/
↪[Signature components]
          \         /\                           /\                                        /
           ---  ---   -------------  ------------  ------------------  --------------------
              \/                   \/                                \/
         Command Type         Request ID          Select a challenge and provide related info
```

- Example of “_VALIDATE” interest
  - After sending “_SELECT” to select Email challenge, then a following “_VALIDATE” interest can be:

```
/ndn/edu/ucla/_VALIDATE/{"request-id":"111"}/EMAIL/{"code":"1234"}/...
                                             |        |
                                             v        |
                     Indicates using email challenge  |
                                                      |
                                                      v
                                    Verification code obtained from email
```

  - After sending “_SELECT” to select Pin challenge, then a following “_VALIDATE” interest can be:

```
/ndn/edu/ucla/cs/zhiyi/_VALIDATE/{"Request-ID":"111"}/PIN/{"code":"1234"}/...
                                                       |        |
                                                       v        |
                             Indicates using PIN code challenge |
                                                                |
                                                                v
                                              The PIN code obtained out of bind
```

2..3.3 _STATUS interest packet

- Interest command parameters:
  - Command Type “_STATUS”
  - Request ID in JSON format
- MustBeFresh field must be set
- Interest Signature
  - Signature must be signed by `user certificate key`
- Verification
  When received, several parts need to be verified to ensure authentication:
  - Certificate signature must be verified
  - Signature must match the request ID
- Interest naming convention
  The interest name format is as below:

```
/CA-prefix/_STATUS/{"request-id": "1209381203"}/[Signature components]
          \         /\                         /
           ---  ---   -----------  ------------
              \/                 \/
         Command Type         Request ID
```

- Example of “_STATUS” interest
  - After sending “_VALIDATE” to select Email challenge, then a following “_VALIDATE” interest can be:

```
/ndn/edu/ucla/_STATUS/{"request-id":"111"}/...
```

2..3.4 Data packet answering _SELECT, _VALIDATE and _STATUS

- Content format: JSON
  - Request-ID: The request ID
  - Challenge-Type: The challenge module type
  - Status: The application status

  - Certificate: The encapsulated certificate signed by CA. This field is used only when challenge is verified successfully
- Example:

```
Example #1
{
  "request-id": "111",
  "challenge-type": "EMAIL",
  "status": "need-email"
}

Example #2
{
  "request-id": "111",
  "challenge-type": "PIN",
  "status": "success",
  "certificate": "/ndn/edu/ucla/CA/_DOWNLOAD/{"Request-ID":"111"}"
}
```

2..4 Certificate download

2..4.1 _DOWNLOAD interest packet

- Interest command parameters:
  - Command Type "_DOWNLOAD"
  - Request ID in JSON format
- MustBeFresh field must be set
- Interest Signature
  No requirements.
- Verification
  No requirements.
- Interest naming convention
  The interest name format is as below:

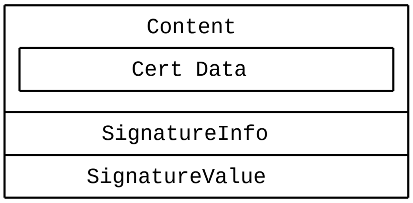
```
/CA-prefix/_DOWNLOAD/{"request-id": "1209381203"}
          \        /\                           /
           ---  --   -------------  ------------
              \/                 \/
       Command Type         Request ID
```

- Example of "_DOWNLOAD" interest
  - After knowing certificate has been issued, a "_DOWNLOAD" interest to download the certificate can be:

```
/ndn/edu/ucla/_DOWNLOAD/{"request-id":"111"}
```

2..4.2 Data packet answering _DOWNLOAD

- Data structure:



2..5 Error information

When there is error caused by certificate requester, CA will reply a error JSON within the data:

- JSON format
  - status: The application status. Here it can only be error.
  - error-info: The error information.

The example is as follow

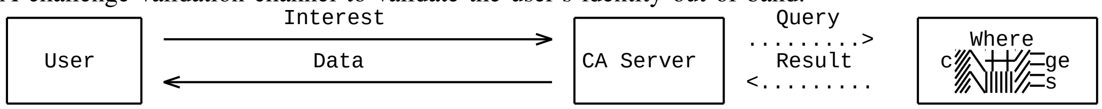
```
Example #1
{
  "status": "error",
  "error-info": "invalid poll packet signature"
}

Example #2
{
  "status": "error",
  "error-info": "invalid certifcate request in new packet"
}
```

3. Security Consideration

3..1 Threat Model

As a service in ndn, the protocol's threat model is different from the Interet threat model. There are two "channels" for a ndn cert management CA to interact with outside.

- The interest-data channel between CA and user.
- A challenge validation channel to validate the user's identity out of band.



The protocol aims to be secure against all kinds of attacks on any channel. The protocol needs to protect from the network layer attacks also the application layer attacks. We need to account for

- Man-in-the-middle attack
- Denial-of-service attack

The Interest and Data exchange safety is ensured by NDN trust schema. Notice that the interest-data exchange can protect the information from interception because the user can check the name of data packet to ensure that interest is not tampered.

The channel is safe when:

- User builds up prior trust to CA
  Prior trust is of vital importance to defense MitM attack. Once build up the trust directly between user and CA, there is no chance for third party MitM attack.
- certificate requester side strictly verifies the signature
- certificate requester side strictly check the Data name with the interest name
- CA side strctly verfies the signature and certificate request

For the out-of-band challenge validation channel, it is CA server's responsibility to design and build up a safe environment.

3..2 Denial of service consideration

As a certificate service running over NDN, requests from user will require CA server to perform a few potentially expensive operations. To mitigate the attacks, CA servers can take the similar strategy as that in TCP/IP network. CA servers can take a proper rate limits. Also CA servers can strictly check the interest request component. An interest request without encryprion or without a valid certificate request can be ignored.

3..3 MitM consideration: Trust CA before certificate management

- User and Root CA

User must have prior trust of Root CA certificate

- Root CA certificate can be pre-installed to user's keychain.
- Root CA certificate can be installed to certificate requester manually.
- User and Sub-namespace CA

It is user's responsibility to ensure that sub-namespace user must have prior trust of CA certificate

- NDNCERT provides a mechanism for sub-namespace user to get sub-namespace certificate
  1) User should indicate a trust repository to fetch CA certificate. e.g. a trust NDNS or a private repo. The default trust repository is NDN testbed root CA.
  2) The certificate requester will send an interest with the sub namespace and link object to fetch the sub-namespace certificate.
  3) Verify the certificate and install.
- Sub-namespace CA certificate can be installed to sub-namespace user manually.

3..4 Signature and Signature Validation

- Signed Interest
- "`_SELECT`" interest command from user must be signed with the `user certificate key`.
- "`_VALIDATE`" interest command from user must be signed with the `user certificate key`.
- "`_STATUS`" interest command from user must be signed with the `user certificate key`.
- Signed Data
- All the data from CA must be signed with the related upper level certificate.
  For example:

```
For the command _NEW to apply for /ndn/edu/ucla/cs/zhiyi,
  the data should be signed by certificate of the namespace /ndn/edu/ucla/cs.
For the command _PROBE to ask for available namespace under /ndn/edu/ucla/remap,
  the data should be signed by certificate of the namespace /ndn/edu/ucla/remap.
For the command `_VALIDATE` to check status of issuing certificate for /ndn/edu/ucla/cs/zhiyi,
  the data should be signed by certificate of the namespace /ndn/edu/ucla/cs.
```

- Signature validation
- All the signed interests and all the data packets must be verified. If signature is invalid, the packet must not be used.

3..5 Encryption and Decryption

Encrption and decryption can be added by CA. Encryption may mainly apply to the safe challenge information part in "`_VALIDATE`" interest.

4. Challenges

This section will introduce how Challenge Module should perform the challenge and in which way should Challange Module collaborate with CA. Also two concrete challenge module examples will be illustrated; these two modules have been implemented in NDNCERT.

4..1 How to perform a challenge

The goal of a challenge is to verify certificate requester's possession of some resources like an email address or ability to contact directly with CA. In the main picture of NDNCERT, after certificate requester send a NEW interest to CA, CA would provide a list of Challenges in reply. User would then select one challenge type from it and notice CA in the following "_SELECT" interest. Challenge module should take use one or more "_VALIDATE" interest and Data exchange to finish the challenge.

As mentioned in the section 2.3.1, there is a component called ChallengeInfo in "_VALIDATE" interest name. This ChallengeInfo is used for certificate requester to provide required information to Challenge Module. Also there is a JSON field called status mentioned in section 2.3.2. This status is used by Challenge Module to give instruction to certificate requester.

With interest name component ChallengeInfo and json field status, Challenge Module can talk directly with certificate requester. certificate requester have no clue what should be put in ChallengeInfo, so it is Challenge Module who should explicitly notice certificate requester.

5. Challenge Examples

Two examples illustrating how NDNCERT performs challenges.

5..1 Email-based Challenge

Email-based Challenge Module (ChallengeEmail) validates claims for the namespace based on email possession verification. The specific namespace that can be claimed based on the email possession depends on the specific CA module and can be directly inferred from the email or be on the first-claim-first-given basis.

Email-based challenge is used by certificate authorities on NDN testbed (/ndn/edu/ucla/CA, /ndn/edu/arizona/CA, and others).

The operations of Email Challenge can be summarized as follows:

- Determining namespace (e.g., using _PROBE command) and generating certificate request
- Email challenge selection
- Informing CA about which email will be used for verification
- Obtaining verification code from email
- Challenge confirmation
- Success or failure

An example of the overall process is shown in the following diagram:

```
   requester                                                                       CA
       |                                                                            |
       |                                                                            |
       |                  _PROBE + <cert> generation (TBD)                          |
       |                                                                            |
       |                                                                            |
       |                                                                            |
       |                                                                            |
       |          Interest for /ndn/edu/ucla/CA/_NEW/<cert>/[signature]             |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                  Validate signature
       |                                                                  using corresponding
       |                                                                     certificate
       |                                                                            |
       |                                                                   Immediately abort
       |                                                                  if signature cannot
       |                                                                     be validated
       |                                                                            |
       |      Data, Content={"request-id":"111", "status":"wait-selection",        |
       |                      "challenges": ["EMAIL"]}, Signed by CA                |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       |                 I N I T I A T I N G   C H A L L E N G E                     |
       |                                                                            |
       |        Interest for /ndn/edu/ucla/CA/_SELECT/{"request-id":"111"}           |
       |                     /EMAIL/{"email":"zhiyi@cs.ucla.edu"}/[signature]        |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |                                                                Generate verification code
       |                                                                  and email to requester
       |                                                                            |
       |         Data, Content={"request-id":"111", "status": "wait-code",          |
       |                        "challenge": "EMAIL"} Signed by CA                  |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       ~                                                                            |
Discover verification code                                                          |
(by checking email)                                                                 |
       ~                                                                            |
       |                                                                            |
       |                  F I N I S H I N G   C H A L L E N G E                     |
       |                                                                            |
       |       Interest for /ndn/edu/ucla/CA/_VALIDATE/{"request-id":"111"}         |
       |                    /EMAIL/{"code":"123"}/[signature]                       |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |      Data, Content={"request-id":"111", "status": "pending",               |
       |                      "challenge": "EMAIL"} Signed by CA                    |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       ~                                                                            ~
       ~                                                                            ~
       |                                                                            |
       |   Interest for /ndn/edu/ucla/CA/_STATUS/{"request-id":"111"}/[signature]   |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                             |
       |      Data, Content={"request-id":"111", "status": "issued",                 |
       |              "cert": "/ndn/edu/ucla/CA/_DOWNLOAD/{"request-id":"111"}",     |
       |              "challenge":"EMAIL"} Signed by CA                              |
       | <-------------------------------------------------------------------------- |
       |                                                                             |
Validate CA signature.                                                               |
Immediately abort if signature cannot be validated                                   |
       |                                                                             |
       ~                                                                             ~
       ~                                                                             ~
       |                R E T R I E V I N G   C E R T I F I C A T E                  |
       |                                                                             |
       |       Interest for /ndn/edu/ucla/CA/_DOWNLOAD/{"request-id":"111"}          |
       | --------------------------------------------------------------------------> |
       |                                                                             |
       |     Data, Content=Certificate(/ndn/edu/ucla/cs/zhiyi/KEY/%01/NDN/%FD01)     |
       | <-------------------------------------------------------------------------- |
```

1) (Pre-knowledge) The requester build up the trust to CA (installed the CA's certificate).
2) (Pre-knowledge) The requester determines from which CA certificate should be requested.

```
/ndn/edu/ucla/CA
```

3) (Pre-knowledge) The requester determines which type of CA (= how namespace is managed).
4) If CA type requires/provides `_PROBE` capabilities, determine the available/allowed namespace for the sub-identity, e.g., `/ndn/edu/ucla/cs/zhiyi`.
5) Generate a key-pair and the certificate request.

```
/ndn/edu/ucla/cs/zhiyi/KEY/%01/cert-request/%00
```

6) Send `_NEW` command to the CA.

```
/ndn/edu/ucla/CA/_NEW/<cert>/[signature]
```

7) Validate the signature of reply data. Get the request ID and available challenge list from data.
8) Send `_SELECT` command, selecting `EMAIL` as a challenge and providing requester's email.

```
/ndn/edu/ucla/CA/_SELECT/{"request-id":"111"}/EMAIL/{"email":"zhiyi@cs.ucla.edu"}/[signature]
```

9) Validate the signature of reply data.
10) (Out-of-band) Obtain verification code by checking email.
11) Send `_VALIDATE` command, confirming the verification code.

```
/ndn/edu/ucla/CA/_VALIDATE/{"request-id":"111"}/EMAIL/{"code":"123"}/[signature]
```

12) Validate the signature of reply data. If response to `_VALIDATE` is an issued certificate ("status": "success"), download it using the provided link to encapsulated cert (or a key bundle). If response is ("status": "pending"), periodically send `_STATUS` requests.

```
/ndn/edu/ucla/CA/_STATUS/{"request-id":"111"}/[signature]
```

5..2 PIN Code-based Challenge

PIN Challenge assumes the certificate requester and certificate authority can agree on a generated PIN in some out-of-band method. This document doesn't define any specific method how this agreement can be reached.

Use case for the challenge: management of sub-namespace certificate. For example, creating a certificate for `/zhiyi/iphone` namespace using `/zhiyi` identity's certificate as a local trust anchor / certificate authority.

The operations of PIN Challenge can be summarized as follows:

- Challenge selection
- Reaching agreement out-of-band on PIN code
- Challenge confirmation
- Success or failure

An example of the overall process is shown in the following diagram:

```
    Requester                                                                      CA
       |                                                                            |
       |                  _PROBE + <cert> generation (TBD)                          |
       |                                                                            |
       |                                                                            |
       |           Interest for /zhiyi/CA/_NEW/<cert>/[signature]                   |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |       Data, Content={"request-id":"111", "status": "wait-selection",       |
       |                      "challenges": ["PIN"]}, Signed by CA                  |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       |                 I N I T I A T I N G   C H A L L E N G E                    |
       |                                                                            |
       |     Interest for /zhiyi/CA/_SELECT/{"request-id":"111"}/PIN/[signature]    |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |                                                                   Generate PIN and record
       |                                                                      it internally
       |                                                                            |
       |         Data, Content={"request-id":"111", "status": "wait-code"},         |
       |               Signed by CA                                                 |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       ~                                                                            |
Discover PIN                                                                        |
(out-of-band, mechanism currently undefined)                                        |
       ~                                                                            |
       |                                                                            |
       |                  F I N I S H I N G   C H A L L E N G E                     |
       |                                                                            |
       |  /zhiyi/CA/_VALIDATE/{"request-id":"111"}/PIN/{"code":"123"}/[signature]   |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |      Data, Content={"request-id":"111", "status": "pending"}               |
       |            Signed by CA                                                    |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       ~                                                                            ~
       ~                                                                            ~
       |      Interest for /zhiyi/CA/_STATUS/{"request-id":"111"}/[signature]       |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |                                                                   Validate signature
       |                                                                   using corresponding
       |                                                                      certificate
       |                                                                            |
       |                                                                    Immediately abort
       |                                                                   if signature cannot
       |                                                                      be validated
       |                                                                            |
       |      Data, Content={"request-id":"111", "status": "issued",                |
       |                     "cert": "/zhiyi/CA/_DOWNLOAD/{"request-id":"111"}"     |
       |            Signed by CA                                                    |
       | <------------------------------------------------------------------------- |
       |                                                                            |
Validate CA signature.                                                              |
Immediately abort if signature cannot be validated                                  |
       |                                                                            |
       ~                                                                            ~
       ~                                                                            ~
       |                R E T R I E V I N G   C E R T I F I C A T E                 |
       |                                                                            |
       |           Interest for /zhiyi/CA/_DOWNLOAD/{"request-id":"111"}            |
       | -------------------------------------------------------------------------> |
       |                                                                            |
       |       Data, Content=Certificate(/zhiyi/subnamespace/KEY/%01/my/%FD01)      |
       | <------------------------------------------------------------------------- |
```

1) (Pre-knowledge) The requester build up the trust to CA (installed the CA's certificate).
2) (Pre-knowledge) The requester determines from which CA certificate should be requested.

```
/zhiyi/CA
```

3) (Pre-knowledge) The requester determines which type of CA (= how namespace is managed).
4) If CA type requires/provides _PROBE capabilities, determine the available/allowed namespace for the sub-identity, e.g., /zhiyi/iphone.
5) Generate a key-pair and the certificate request.

```
/zhiyi/iphone/KEY/%01/cert-request/%00
```

6) Send _NEW command to the CA.

```
/zhiyi/CA/_NEW/<cert>/[signature]
```

7) Validate the signature of reply data. Get the request ID and available challenge list from data.
8) Send _SELECT command, selecting PIN as a challenge.

```
/zhiyi/CA/_SELECT/{"request-id":"111"}/PIN/[signature]
```

9) Validate the signature of reply data.
10) (Out-of-band) Reach agreement on PIN.
11) Send _VALIDATE command, confirming the PIN.

```
/zhiyi/CA/_VALIDATE/{"request-id":"111"}/PIN/{"code":"123"}/[signature]
```

12) Validate the signature of reply data. If the response to `_VALIDATE` is an issued certificate (“status”: “success”), download it using the provided link to encapsulated cert (or a key bundle).
    If response is (“status”: “pending”), periodically send `_STATUS` requests.

```
/zhiyi/CA/_STATUS/{"request-id":"111"}/[signature]
```