ACM ICN 2018 Tutorial
Boston, MA, September 21, 2018

NDNSEC Tools and Trust Bootstrapping

Sanjeev Kaushik Ramani (FIU)
Zhiyi Zhang (UCLA)

Contents

- NDNSEC Command Line Tool Suite
- NDN Security Bootstrapping

NDNSEC Introduction

- Command-line toolkit to perform various NDN security management operation
- NDN security data are stored and managed in two places
  - Public Information Base
  - Trusted Platform Module
- Usage:
  - ndnsec <command> [<args>]
  - ndnsec-command [<args>]

Identity, Key, and Certificate in NDN Security

- Keys, certificates and their identities are managed by KeyChain
- Real-world identity can be expressed by a namespace
  - /ndn/edu/ucla/alice
  - /ndn/edu/fiu/MeritLab/PG6142
- Identity: a named entity in NDN
- Key: owned by an identity. Each identity can have more than one keys.
- Certificate: used to certify a key. Each key can have more than one certificates issued by different issuers/authorities

NDNSEC Command Line Tool Suite

```
➜  ~ ndnsec help
\
  help          Show all commands
  version       Show version and exit
  list          Display information in PublicInfo
  get-default   Get default setting info
  set-default   Configure default setting
  key-gen       Generate a Key-Signing-Key for an identity
  sign-req      Generate a certificate signing request
  cert-gen      Generate an identity certificate
  cert-dump     Dump a certificate from PublicInfo
  cert-install  Install a certificate into PublicInfo
  delete        Delete identity/key/certificate
  export        Export an identity package
  import        Import an identity package
  unlock-tpm    Unlock Tpm
```

Example 01: Try ndnsec-ls-identity -c to see your identities, keys, and certificates

```
➜  ~ ndnsec-ls-identity -c
  /ndn/edu/ucla
  +->* /ndn/edu/ucla/KEY/%97%95%D4%CF%3C%F8%B6%16
       +->* /ndn/edu/ucla/KEY/%97%95%D4%CF%3C%F8%B6%16/NA/%FD%00%00%01d%D8%B3%CA%FE

  /ndn/edu/ucla/test
  +->* /ndn/edu/ucla/test/KEY/%7B%E3%E9f%5E%D5%A8%9F
       +->* /ndn/edu/ucla/test/KEY/%7B%E3%E9f%5E%D5%A8%9F/self/%FD%00%00%01e%BD%02I%88
       +->  /ndn/edu/ucla/test/KEY/%7B%E3%E9f%5E%D5%A8%9F/NDNCERT/%FD%00%00%01e%BD%02%A5M

* /ndn/edu/ucla/alice
  +->  /ndn/edu/ucla/alice/KEY/%84%5B%F5%8E%FAz%88%C5
       +->* /ndn/edu/ucla/alice/KEY/%84%5B%F5%8E%FAz%88%C5/self/%FD%00%00%01d%D9%3B%04R
  +->* /ndn/edu/ucla/alice/KEY/%CEj%BA%2F%24%5D%09%19
       +->* /ndn/edu/ucla/alice/KEY/%CEj%BA%2F%24%5D%09%19/self/%FD%00%00%01d%D9%3A%C77
```

Example 02: Try to create a Self-signed Certificate

- ndnsec-key-gen <your identity name>

```
➜  ~ ndnsec-key-gen /icn/tutorial/identity
Bv0Czwc5CANpY24ICHR1dG9yaWFsCAhpZGVudGl0eQgDS0VZCAjVgmddr3XdLAgE
c2VsZggJ/QAAAWXqFodWFAkYAQIZBAA27oAV/QEmMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEA26JIshhXmCoBpdWIxo1WuiqQ/9ux5kloYObDJ+kOr9Yy
bFBqP2txvGhOgCPwo4FsGL5eA4/kV9zZhKkaUMMXyEMji0i4yDFWWtDdQjJvVWZ3
Sx7tFQXr8hpDCaxAv+lL51QmtCpXnIHiEit+W9sqcktaVCmr32MI7lk/mG3qJBC5
wsUPt9mcbUtLpn1bCq5M4obA6ABgTsEM8qsd4iJR9BQvTJSGWYALTK1VEFtVd68V
iXyEhe8Wy/56swj6/UPnsBO7Vjnx62+UgNZb9AI8f8dIDsRcWjyYJW3wozdjQ5uD
veZZjRkGvaIgL9DaC+ehUDPqLkt4WON3onuNTbhdCwIDAQABFlkbAQEcKgcoCANp
Y24ICHR1dG9yaWFsCAhpZGVudGl0eQgDS0VZCAjVgmddr3XdLP0A/Sb9AP4PMTk3
MDAxMDFUMDAwMDAw/QD/DzIwMzgwOTEzVDAwMzEwORf9AQDLh4Xv6C8jWrtdwuRC
+zQHTrALd3nIAXTnu9p/d6d5rcH2zvJZffWDZp69embP0o6fNE/3zO2wA0Q9WRND
FSQMgd706HU/iIqeGEQwxxS5hwA901ydYidqPPdc3m54vFn1X9ngmHL100CJHx9k
aobxb+Vu8wZmgltGu3kSk6+uUJu2Zpw7DSs6Ta4LhyY8hQAA2U35bx3700ruaJVd
7vnHqwnc/yl7zXO5YCRufcgnE0BN56G9A+B+a8RSpbwfvT/7pGqpYrqwwcxQYTcZ
TxnT2C6aRXjMQdMIMcOJXwiE4YgMLBasDnzCG6PQcQMXqCsRMx6cwhw6begdElWZ
Tq2X
```

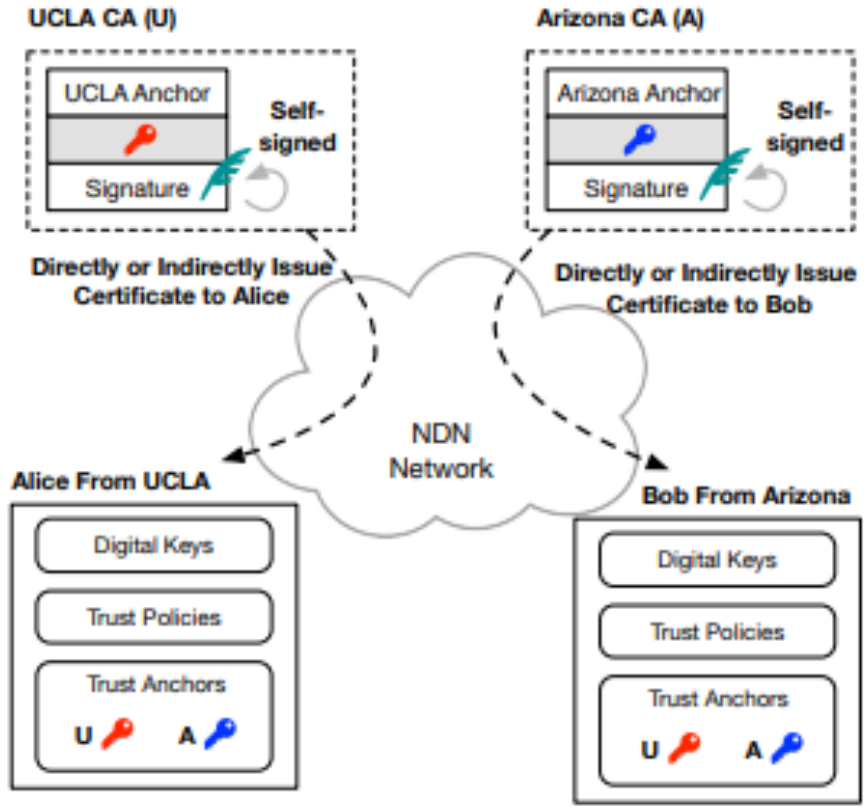
<- Your self-signed certificate

Code and Documentation of NDNSEC

- Manpages of NDNSEC
  - http://named-data.net/doc/ndn-cxx/current/manpages/ndnsec.html
- Source code of NDNSEC (Also a good chance to learn ndn-cxx security APIs)
  - https://github.com/named-data/ndn-cxx/tree/master/tools/ndnsec

Trust Anchor (TA) to Bootstrap Security

- Trusted self-signed certificate
  - Certificate of locally trusted authority
- Trusted non-self-signed certificate
  - Trusting only CS department, but not the whole university
- Necessary option to properly validate packets



Options to Configure Trust Anchor in the Libraries

- Hard-coded in the code ☹
- Configured with trust schema ☺
  - One or multiple
  - Directly defined (base64)
    - Static
  - Indirectly via file name
    - Static
  - Double indirection via directory name
    - Static
    - Dynamic
  - "any" (disabled security)

ValidatorConf syntax

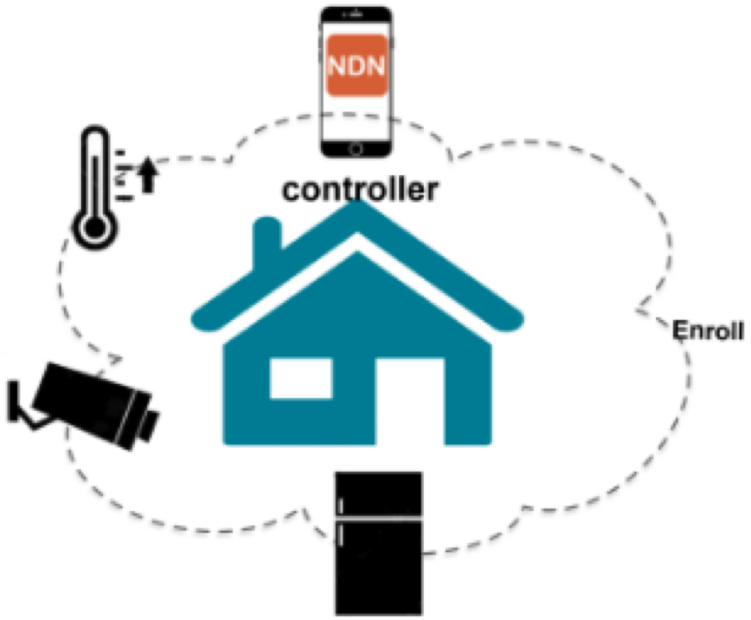
```
trust-anchor
{
  type file
  file-name "trusted-signer.cert"
}
trust-anchor
{
  type base64
  base64-string "Bv0DGwdG...amHFvHIMDw=="
}
trust-anchor
{
  type dir
  dir/usr/local/etc/ndn/keys
}
```

Dynamic Trust Anchor(s)

- A refresh period can be set in cases where the certificate changes during runtime

```
trust-anchor
{
  type dir
  dir /usr/local/etc/ndn/keys
  refresh 1h ; refresh certificates every hour, other units
include m (for minutes) and s (for seconds)
}
```

Automated Bootstrapping in Single Domain Environment (Smart Home Example Scenario)



- Home Controller
  - Android Phone (/home/controller/android-phone)
  - Linux Laptop (/home/controller/linux-laptop)
- IoT Devices
  - Living Room Camera (/home/living/camera)
  - Living Room Television (/home/living/tv)
  - Bed Room Camera (/home/bed/tv)
  - Bed Room Camera (/home/bed/camera)

Automated Bootstrapping Goals and Steps

- Enable device to trust the network with minimal user intervention
  - Obtain the local trust anchor
  - Every home has a global unique name along with a key-pair and a certificate of its public key
  - The certificate serves as the local trust anchor
- Enable device to be trusted by the network
  - Obtain an anchor-signed certificate
  - Device has a key signed by the trust anchor and issued by the controller
  - Device uses this key (referred to as communication key) to sign packets, which can be authenticated by other devices in this network

Establishing Mutual Trust

- High Level Requirements
  - Prevent a device from joining the wrong network
  - Prevent a malicious device from joining network
  - Prevent impersonation / replay attacks by onboarded devices in network
- In the current example:
  - Android Phone / Linux Laptop (controller) would bootstrap the home IoT devices.
  - After bootstrapping, each device can install the trust anchor and obtain an identity certificate.

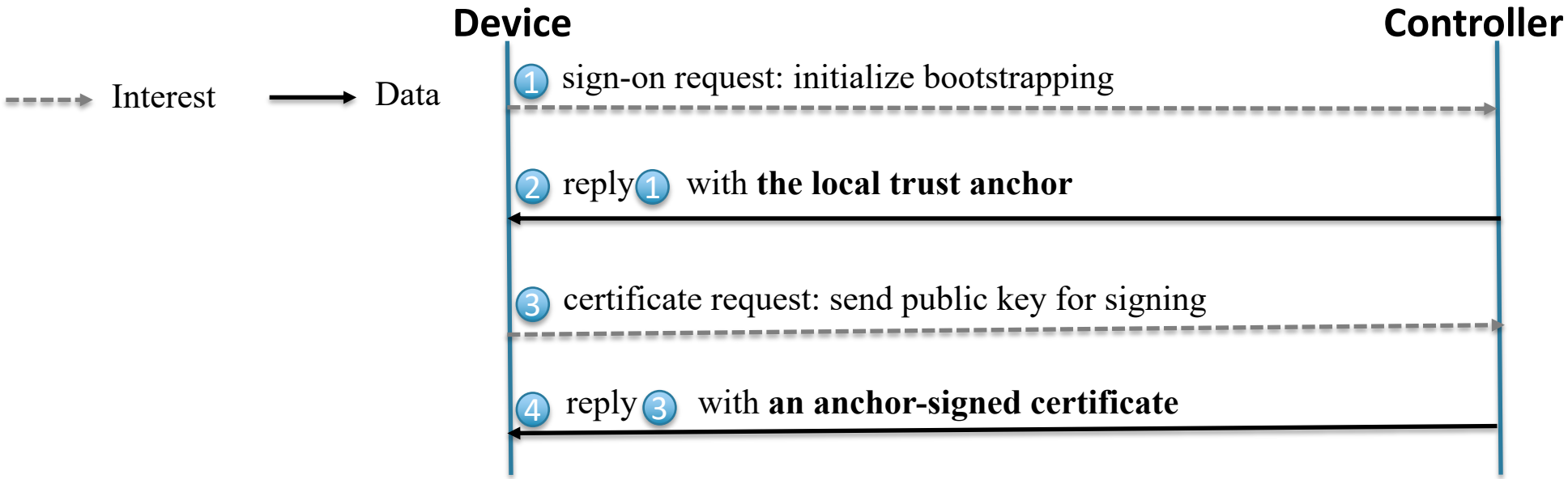
Bootstrapping Assumptions

- Physical connectivity between controller and device exists before bootstrapping
  - Wi-Fi, Bluetooth

- Out-of-band: controller obtains bootstrapping info from device before enrolls it
  - A bootstrapping key (public key) → B_k
  - By other mean like QR code scanning

Bootstrapping Steps

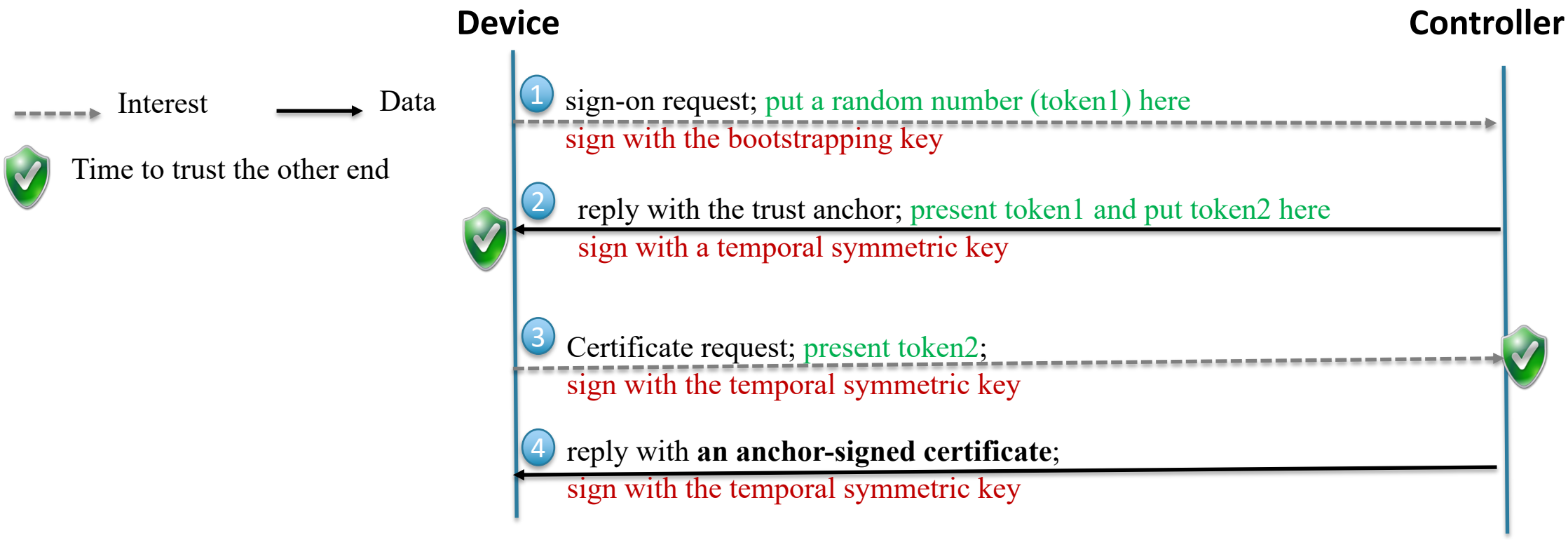
- The device initiates the process by broadcasting a request for bootstrapping; then the controller replies with the local trust anchor
- Then the device generates a key-pair and requests the controller to sign its public key (the communication key); the controller signs that key by the trust anchor and returns the anchor-signed certificate to the device.



Threat Countermeasures

- Device signs the first Interest with the bootstrapping key; then the controller can authenticate the device and thus perceive any fake device.
- Device and controller negotiate a temporal symmetric key, similar to DH, by exchanging two tokens; this key is used to sign following packets to ensure their integrity and both ends' authenticity.
- Meanwhile, for every token, its presence in the next packet prevents replay attacks.

Threat Countermeasures (Contd.)



Becoming Your Own Trust Anchor

- Demo
  - Generate self-signed certificate
  - validator.conf example
  - Proof-of-correctness using simple demo app

Manually Generating Certificates

- ndnsec cert gen demo