A half-done Sigcomm Tutorial App

Group 1: Zhiyi Zhang, Xinyu Ma, Edward Lu, Yu Guan

Group 2: Erynn-Marie Phan, Laqin Fan, Zhiyi Zhang

Contributions

1. Tested that all the application layer support tools in current NDN-Lite works well
2. Add UDP unicast / multicast face into ndn-lite
3. First imple of new Signed Interest support (even before ndn-cxx, nfd)
4. First usable version of Schematized Trust (even before ndn-cxx)
5. Fix several bugs in existing ndn-lite library
    a. Most in Security component
6. A half-done codebase for sigcomm tutorial

Four entities in the network

A consumer

A controller (access control)

A producer who provides Time service: give the current time, like Junxiao’s time server

A producer who provides Print (and even calculation?) service.

Some interesting facts

- It take hours to compile NFD and ndn-cxx on Raspberry Pi 3. **But with ndn-lite, less than one minute**
- The high level APIs really make development life easier
- In our demo, there is totally no manual configuration of network!
- In our demo, there is no pre-shared print service name. So ease of use!
- Packet loss is a big deal in UDP multicast :(
- NDN-Lite really need more doc and a well-writen tutorial page

Trust Schema

Only a signed interest signed by the /ndn/yu/key can print

```
#define yu_prnt_cmd_string "/ndn/SD/Yu/print/hello_world"

#define yu_rule_data_pattern_string "<ndn><SD><Yu><print><><><>"

#define yu_rule_key_pattern_string "<ndn><Yu><KEY><>"


extern ndn_trust_schema_rule_t yu_rule;
```

Service Discovery

Find the service provided by printing service provider automatically

```
ndn_sd_prepare_advertisement(&interest);

ndn_sd_on_advertisement_process(&decoded_interest);

...
```

Access Control

The time returned is encrypted

```
ndn_ac_state_init(&consumer_identity, pub_key, prv_key);

ndn_ac_prepare_key_request_interest(&interest_encoder, &home_prefix,
                                    &component_consumer, 100, prv_key, 0);

ret_val = ndn_data_set_encrypted_content(&data, (uint8_t*)data_string, 32,
                                         &keyid, iv, aes_key);

...
```