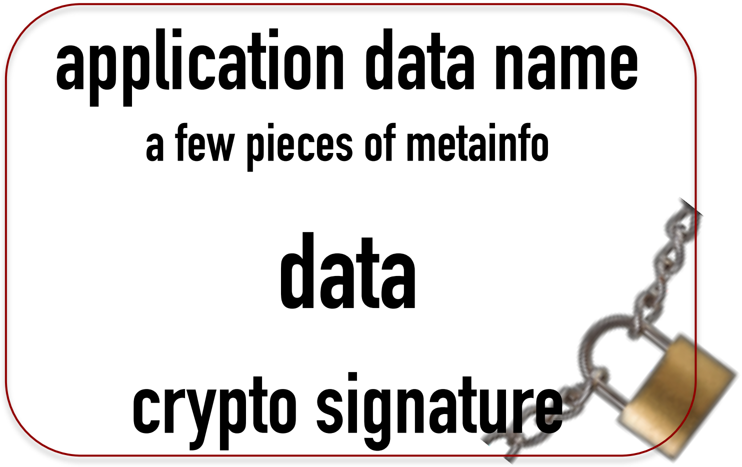
ICN-Enabled Secure Edge Networking with Augmented Reality (ICE-AR): Y2 Progress Overview

JAY MISRA & ALEX AFANASYEV

JULY 16, 2019

Information Centricity

- Information Centric Networking – networking applications
- Communication centered on information → fetching application-named data
- Enabling air-tight security
  - Data secured at generation time
    - Allowing fine granularity of security control



application data name

a few pieces of metainfo

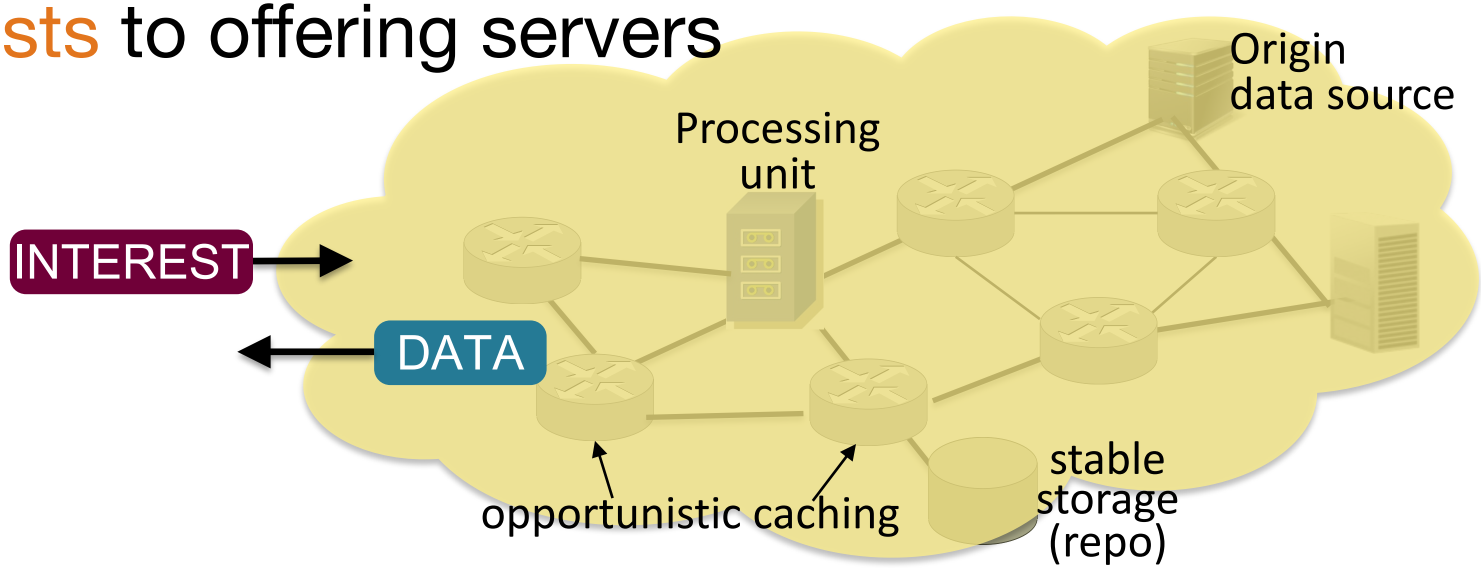
data

crypto signature

Producer binds name to content to create **Data packet**

ICN Enabling Integration

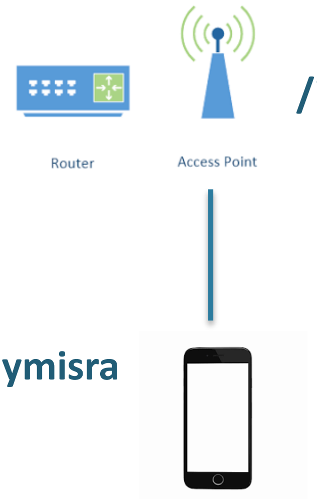
- ◊ Network delivers named, secured data
- ◊ Storage can supply named, secured data equally well
- ◊ So can processing units
  - ○ Processing servers announce their services
  - ○ Clients name the processing results
  - ○ Network forwards requests to offering servers



ICE-AR Security Update

- ◊ Recap of this year's apps work
  - ○ Location-based certificates
  - ○ Attribute-based signatures
- ◊ Plans for next year

Recall – Identity and Location Plurality



/ice-ar/nmsu/cs,ece/SH/AP11

Identity: /ndn/edu/nmsu/cs/fac/jaymisra

Name (hence identity) and certificate tied to content creation can be different!
Devices can get different certificates based on locations or services.

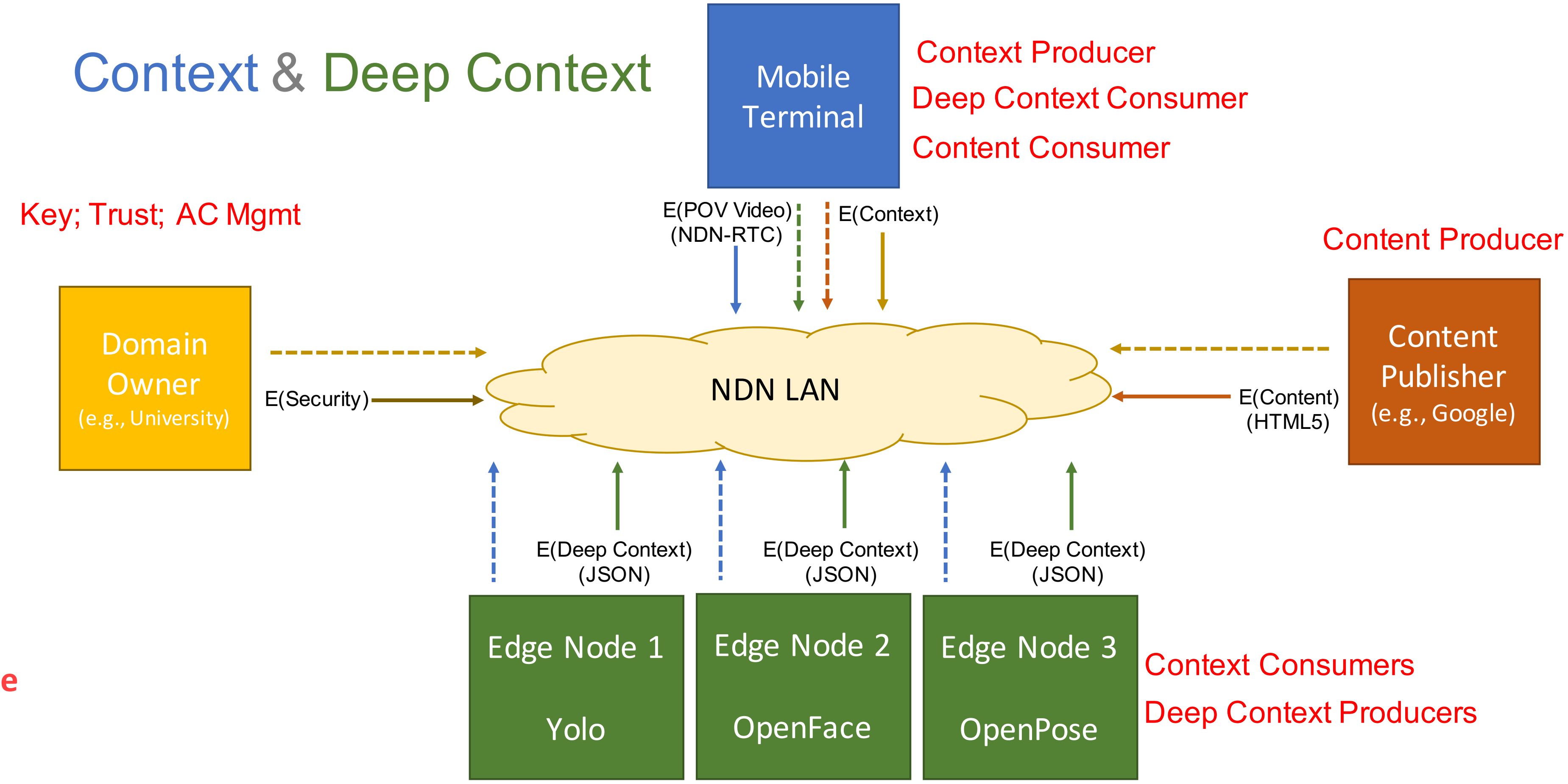
Signature of content vs. signature of location.
Challenges of user privacy vs location privacy.

Things to do:

Extend NDNCert codebase to:

- Automate the process to remove manual user input
- Design a mechanism for the guest to work
- Identify a generic challenge mechanism for the users.
- Integrate with Name-based access control.

Recall: Context + Deep Context



Context & Deep Context

Mobile Terminal

Context Producer

Deep Context Consumer

Content Consumer

E(POV Video) (NDN-RTC)

E(Context)

Key; Trust; AC Mgmt

Domain Owner (e.g., University)

E(Security)

NDN LAN

Content Producer

Content Publisher (e.g., Google)

E(Content) (HTML5)

E(Deep Context) (JSON)

E(Deep Context) (JSON)

E(Deep Context) (JSON)

Edge Node 1

Yolo

Edge Node 2

OpenFace

Edge Node 3

OpenPose

Context Consumers

Deep Context Producers

Owns the namespace

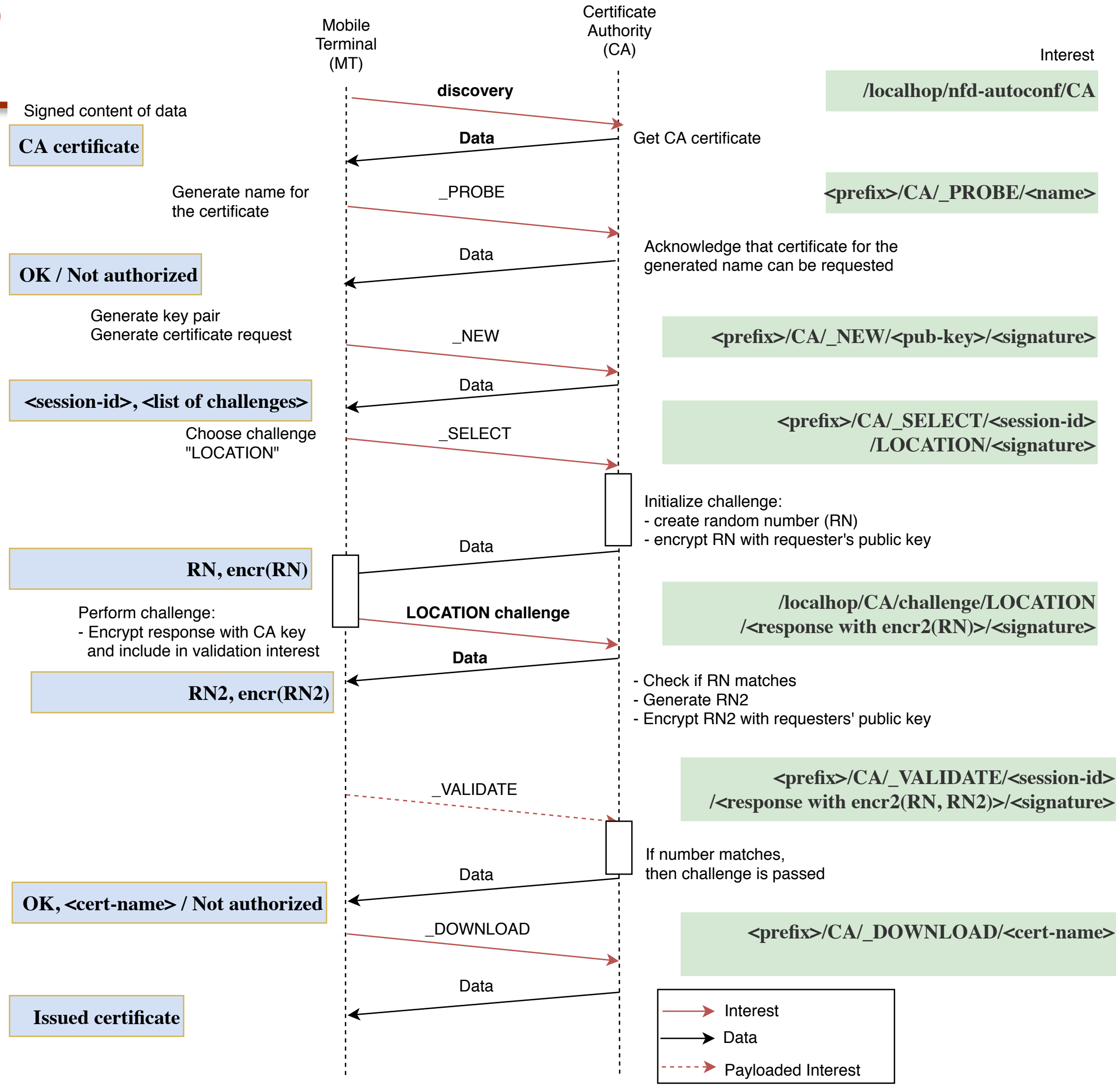
Also, hosts CA (can be separate entity)

Location based Certificates

- Location Discovery:
  - Initiated by MT upon new or changed connectivity

- Certificate Acquisition:
  - Extends NDNCert & adds location challenge
  - CA knows MT is local to its network

- Signature Verification at the Edge
  - Helps verify MT is correctly signing the data
  - Short-lived certs to force location updates



Mobile Terminal (MT)

Certificate Authority (CA)

Interest

Signed content of data

discovery

/localhop/nfd-autoconf/CA

Data

Get CA certificate

CA certificate

Generate name for the certificate

_PROBE

<prefix>/CA/_PROBE/<name>

Data

Acknowledge that certificate for the generated name can be requested

OK / Not authorized

Generate key pair
Generate certificate request

_NEW

<prefix>/CA/_NEW/<pub-key>/<signature>

Data

<session-id>, <list of challenges>

Choose challenge "LOCATION"

_SELECT

<prefix>/CA/_SELECT/<session-id>/LOCATION/<signature>

Initialize challenge:
- create random number (RN)
- encrypt RN with requester's public key

Data

RN, encr(RN)

Perform challenge:
- Encrypt response with CA key and include in validation interest

LOCATION challenge

/localhop/CA/challenge/LOCATION/<response with encr2(RN)>/<signature>

Data

RN2, encr(RN2)

- Check if RN matches
- Generate RN2
- Encrypt RN2 with requesters' public key

_VALIDATE

<prefix>/CA/_VALIDATE/<session-id>/<response with encr2(RN, RN2)>/<signature>

If number matches, then challenge is passed

Data

OK, <cert-name> / Not authorized

_DOWNLOAD

<prefix>/CA/_DOWNLOAD/<cert-name>

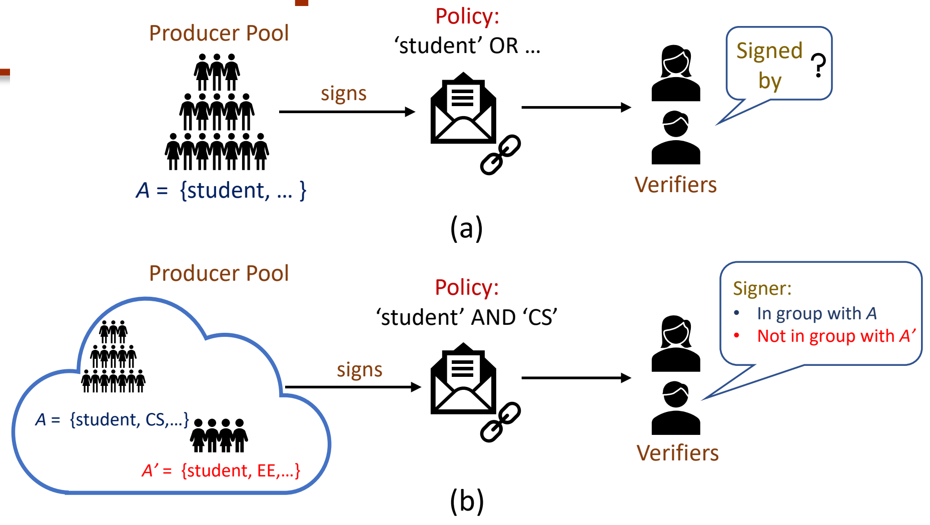
Data

Issued certificate

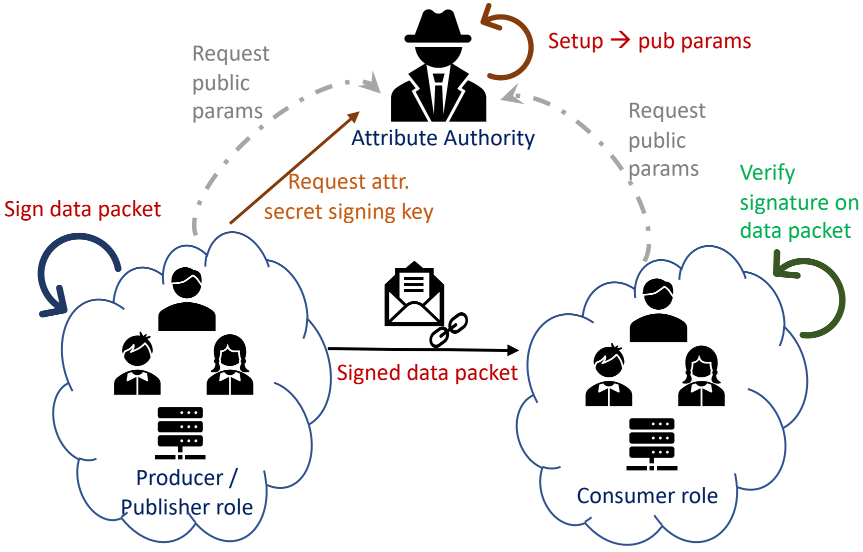
Sequence Diagram for location challenge process between CA and MT

Attributes based signatures design and implementation in NDN

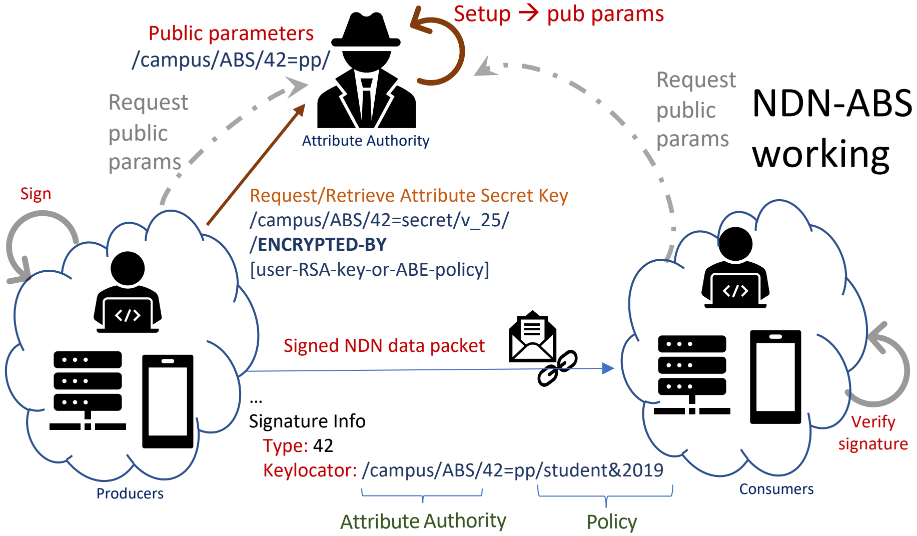
- Ad hoc connectivity implies one may not retrieve all the keys needed to verify retrieved data after retrieving the data; and
- Public key of the signature can be used to identify individual data producers, raising a potential privacy concern.
- ABS can be useful here =>



Conditional privacy using NDN-ABS: (a) Verifier unable to decide the identity of the signer; (b) Verifier can identify the group containing the signer



Overview of NDN-ABS



NDN-ABS naming and working

Challenges we faced so far…

- ◊ Integration of Security with ICE-AR application
  - ○ Applications (e. g., Unity) and OS frameworks (e. g., Android) assume use of IP (in a hardcoded sense) not easy to interface with new paradigms
  - ○ Pulling together different pieces (Android OS, Unity, C#) into a whole requires extensive knowledge and experience (difficult for a research team)
- ◊ IP Focus of Production Platforms (e. g., Android)
  - ○ Difficult to make Android run on link layer directly; had to tunnel over IP, a work-around was a manually configured IP multicast address.
  - ○ Still needed to deploy local DHCP servers on/near WiFi AP, IP needed for Android
  - ○ The auto-connectivity change in Android instead of using MAC address uses IP change to identify connection change (between APs)
- ◊ Lack of ABS Reference Implementations
  - ○ Extensive literature on attribute-based encryption, but significant differences in mathematical details.
  - ○ No complete codebase that realizes the ideas.
  - ○ We guess the lack of practical implementations is the lack of demand. With TCP/IP's point-to-point communication model, today's network security relies on securing connections through encryption, thus ABS is not needed.

Year 3 Plan

- Multi-level Certificates
  - Implement multi-level identity certificates can have guest, limited, and staff credentials in addition to location-based certificates.
  - An MT gets a guest certificate and then gets with authentication to a different identity (e.g., staff).
  - Disruption scenario.
- Extending work on ABS:
  - Our current implementation is in Python and is “slow”
  - Exploring means to speed up: signing manifest, coding in C/C++ and optimizing, evaluating acceleration
- Integration Effort:
  - ICE-AR hackathon planned for the Fall of 2019 to learn about the example applications and library support, then continue the development of our security function demonstration
    - Android client and appliance (Linux-based extension of NDNCERT instance attached to/near access point).
    - We will demonstrate the support for desired application scenarios.
    - We also plan to implement C++ version for attribute-based signature solution and integrate it in the security demonstration.

Thank you!

misra@cs.nmsu.edu; aa@cs.fiu.edu

http://ice-ar.named-data.net