NDNFit Architecture and Progress

Haitao Zhang, Alexander Afanasyev, Jianxun Cao, Euihyun Jung, Jiewen Tan, Jeff Thompson, Yingdi Yu, Jeff Burke, Dan Pei, Christian Tschudin, Lixia Zhang

UCLA Internet Research Lab, UCLA REMAP, Anyang University, Tsinghua University, University of Basel

Introduction

Design Goals
- NDN-based mobile health application — pilot realization of the Open mHealth [2] ecosystem of sensing, storage, analysis, and user interface components
- Open mHealth revisions considering the “data flow” of the application ecosystem, which can be implemented naturally or NDN
- We seek to develop user-centric, privacy-aware data exchange across device, component, and application boundaries

Key Challenges
- Design the application architecture and the namespace
- Design trust relationships between different components
- Search or design a proper algorithm that reconciles consistency as well as handle authorization and authentication
- Design efficient protocols to sync data between different components in the system

Application Architecture and Namespace Design

Architecture (Figure 1)
- NDNFit Capture Application
  - Runs on user’s mobile device
  - Registers identity with NDNFit namespace
  - Acquires data, sign data, timestamped sync data, sign data, sync data with DSU
  - Must auto-register name prefix on edge router in the mobile scenario
- Data Processing Unit (DPU)
  - Constant data flow from DSU and process these data
  - For example, a user can use NFN [2] to work as buffer to calculate the average of buffer walking speed
- Data Processing Unit (DPU)
  - Fetch data from DSUs and process these data
  - For example, a DVU draws and shows a diagram of a user’s walking speed change during a specific hour

Security Design
- The separation of data control and access control
- All the data are stored in user-registered DSUs, which means, data are controlled by DSUs
- The users themselves use mobile devices to generate keys to encrypt data, and decrypt keys to the data consumed in the mobile scenario
- User-configurable trusted relationship
  - Provide default key, sharecenter relationship
  - Allows to configure trust relationship using human-readable config file

Design Development / Implementation – Current Status

Simplified Trust

Trust schema leveraging in named spaces within namespaces names (Figure 2)
- Use exact name prefix to access key chain relationship
- All data are stored in user-registered DSUs, which means, data are controlled by DSUs
- The users themselves use mobile devices to generate keys to encrypt data, and decrypt keys to the data consumed in the mobile scenario

Figure 3: NDNFit Trust Relationship

Data Sync from Mobile Devices to DSUs
- One-directional data sync (captured in DSUs)
  - Introduce one direction of direction – monitor packets – used for DSUs to fetch the names of captured data
  - DSUs send confirmation for monitor packets back to mobile devices
  - Mobile devices have limited storage, thus need to delete data after receiving confirmation

Data sync from DSUs to DPs and DVUs
- One-directional data sync (DSU to DPU/DVU), with a different protocol for storage of results
  - Introduce monitor packets, used for DPU and DVU to fetch the names of stored data from DSUs
  - Introduce update information packets, used for DPU and DVU to fetch new versions of previously fetched data
  - New data may be inserted into previously generated data by mobile devices (because of network disconnection)

Table 1: Three Types of Names in Sync Protocols

| Type | Description | Example
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID manager based on existing NDN cert website</td>
<td></td>
<td>org/lincolnlab/mhealth/PI/10057-900000-10</td>
</tr>
<tr>
<td>Email</td>
<td></td>
<td><a href="mailto:jeff@uh.edu">jeff@uh.edu</a></td>
</tr>
<tr>
<td>NDN namespace (topology-independent)</td>
<td></td>
<td>/org/openmhealth/ndnfit/haitao</td>
</tr>
</tbody>
</table>