Lux: Enabling Ephemeral Authorization for Display-Limited IoT Devices

Logan Blue
University of Florida

Samuel Marchal
F-Secure Corporation and alto University

Patrick Traynor
University of Florida

N Asokan
University of Waterloo and Alto University

Introduction

- Smart Speakers and Smart hubs – Google Home and Amazon Echo
- Uses are increasing by the day – online service access
- Widespread adoption – Hotels, conference rooms

Key Differences

Permanent Space

Temporary Space

Key Differences

Permanent Space

Temporary Space

Long term ownership
Fully Private Space
One time authorization

Key Differences

Permanent Space

Temporary Space

Long term ownership
Fully Private Space
One time authorization

Short Term Ownership
Semi Private Space
Temporary Authorization

Key Differences

Permanent Space

Issues:
1. No display – Is an issue when we want to authenticate regularly
2. Built keeping long term authorization in mind
3. Not user aware – Hotel staff misusing the authorized hub

Temporary Space

Long term ownership
Fully Private Space
One time authorization

Short Term Ownership
Semi Private Space
Temporary Authorization

Lux : Ephemeral Authorization

- System was designed to improve security in temporary environments like hotels and conference rooms.
- Requirements of such a system:

Lux : Ephemeral Authorization

- System was designed to improve security in temporary environments like hotels and conference rooms.
- Requirements of such a system:
  - Easy first time Authorization
  - Temporarily and Spatially bound on device Authorization
  - Enforce principle of least privilege
  - Deployable

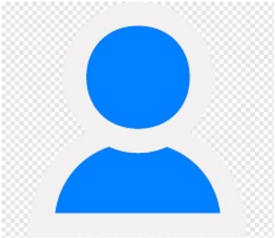
Lux Mechanisms

- Authorization Protocols
  - First Authorization
  - Second Authorization
  - Authorization state machine

Lux Mechanisms

- Authorization Protocols
  - First Authorization
  - Second Authorization
  - Authorization state machine
- Automated Co presence detection
  - Creation of Initial Signature
  - Verification of test signatures

Participants



User

Smartphone

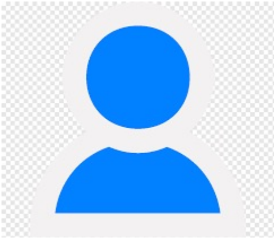
Root Service

Google Home

Online service

Participants

Usually provided by the same company => Easy deployment



User

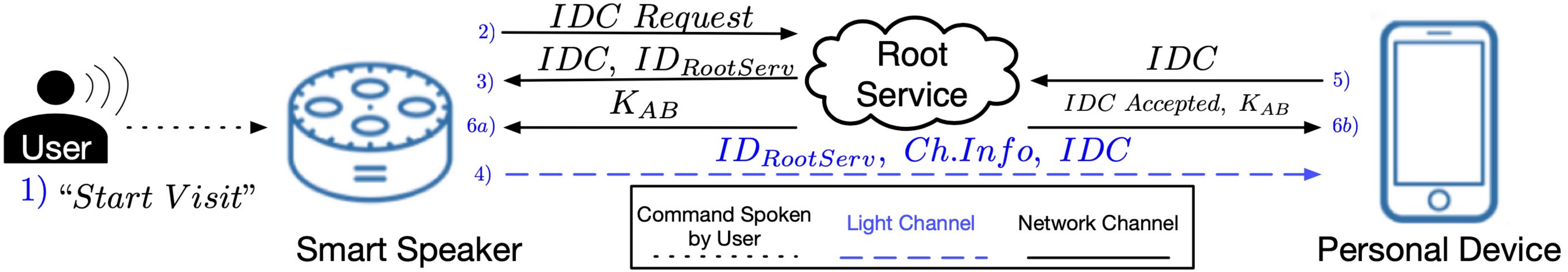
Smartphone

Root Service

Google Home

Online service

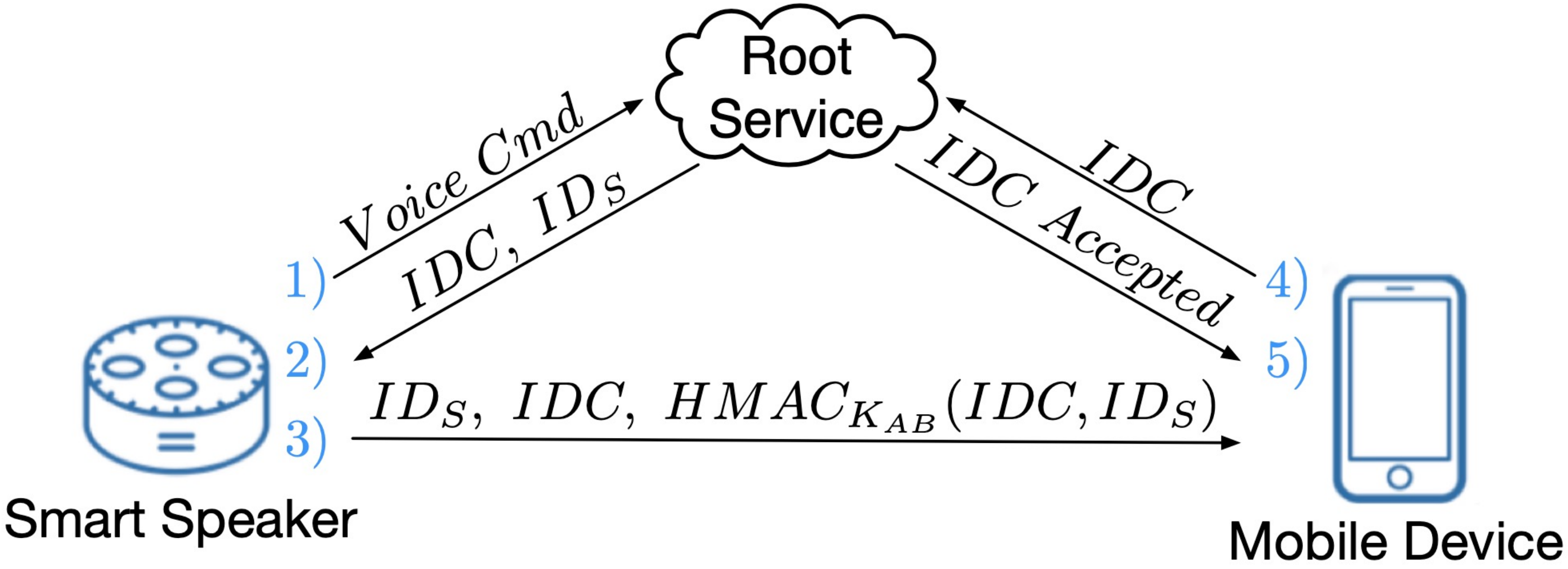
Protocol



First Authorization Step

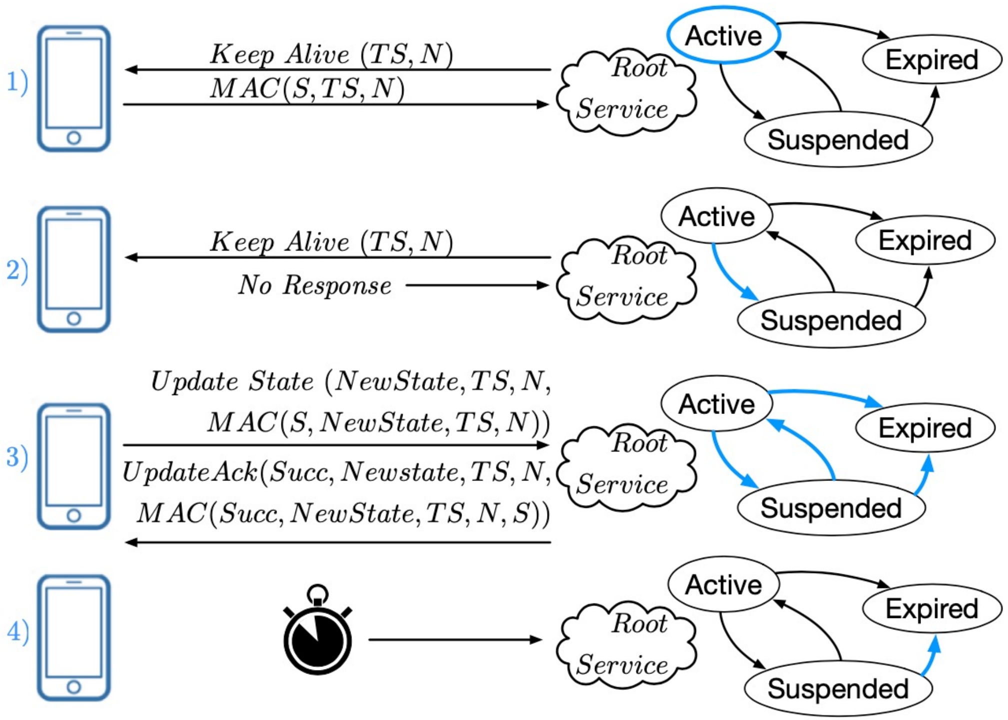
Requirements fulfilled
1. Easy setup
2. Enforce principle of least privilege – only access to user's root service account
3. Deployable – Only using software by one of the companies

Protocol



Second Authorization

Protocol



Permission States and State Machine

Implementation

- Speaker and phone
  - A nexus 6 with android 7
  - Video is offloaded for signal detection and extraction
  - Custom speaker with 18 RGB lights transmitting data over 3 channels.
  - Used a diffuser to deal with white balance

Implementation

- Co Presence detection
  - Use of WiFi access points list to detect co presence
  - A signature is used to for this purpose
    - List of mac and pi
    - Pi is RSS normalized and made positive
      - Has the property of summing up to one
  - Two types of signatures calc by personal device
    - Sigloc and Sigt
  - Pi is a probability distribution and hence we use Hellinger's distance as a measure to determine similarity
  - Threshold is used to classify it as co present.

$$p_i = \frac{RSS_i - dB_{lim} + 10}{\sum_{j=1}^{n} RSS_j - dB_{lim} + 10}$$

Consequently, *Sig* can be seen as a probability distribution:

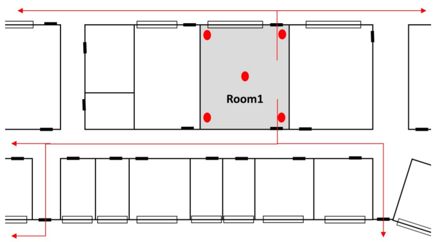
$$Sig = \{(mac_1, p_1), \cdots, (mac_n, p_n)\}, \text{ where } \sum_{i=1}^{n} p_i = 1$$

Evaluation Summary

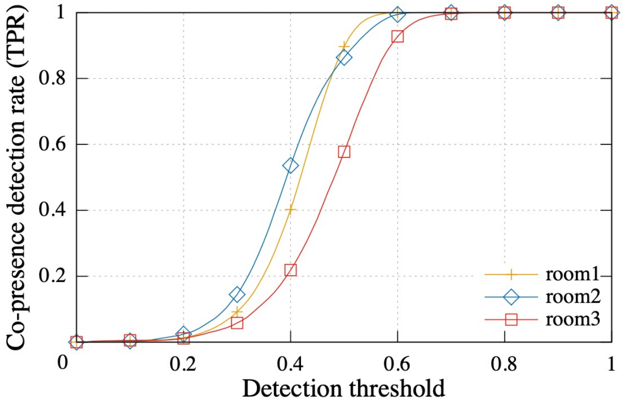
- Various Timings
  - First authorization – 4036 (del 320ms) – encoding and decoding of messages
  - Second authorization – 155ms (del 6.2)
- ProVerif – No leakage in TLS connection – K ab doesn’t leak and hence HMAC cant be forged

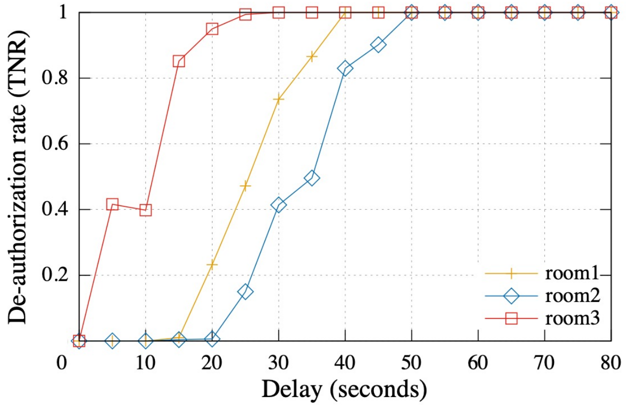
Evaluation Summary

- Deauthorization



Experimental Setup





Accuracy vs Delay Tradeoff

Hyperparameter :

1. dB lim (constant added to normalization eq)
2. Scan rate
3. Threshold for classifier

1 and 2 are found out by minimizing $(H(Sig_{loc}, Sig_t))$

3 is calculated by taking accuracy and delay into account

Can we have an adversary attack?

- It is possible if the adversary simulates 6 AP. A lot of work.
- Need to setup the AP beforehand and then try to simulate signature when the user goes out of scope – basically follow him around with wifi
- Not feasible