SARA:
Secure Android Remote Authorization

Abdullah Imran, Habiba Farrukh, Muhammad Ibrahim, Z. Berkay Celik, Antonio Bianchi
Purdue University
Mobile Devices in Authorization Schemes

Confirm Transfer $100 to John Doe
Confirm Cancel

BANK APP
Threat Model

Attacker Capabilities

- Compromise OS
  - Manipulate IO
  - Alter Program Flow
- Physical Access
  - Sporadic
Threat Model

**Attacker Capabilities**

- **Compromise OS**
- **Physical Access**

**Can we use TrustZone to defeat such a powerful attacker?**
Existing APIs in Android

- Key Storage in TrustZone
- Key Attestation in TrustZone
- TrustZone controlled Secure UI
- Biometric Prompt
Existing APIs in Android

- Key Storage in TrustZone
- Key Attestation in TrustZone
- TrustZone controlled Secure UI
- Biometric Prompt

Biometric Prompt

Use your biometric credentials to authorize the transfer of 9000 USD to Alice

Cancel
Market Analysis

112,886 Apps (Google Play Store)

Android Protected Confirmation
- 0 Apps using

Key Attestation
- 5 Apps using
- All local use cases
API Limitations

Biometric Prompt
- Fake Prompt
- Everlasting Biometric

Android Protected Confirmation
- Illegitimate User
- Overwriting Confirmation
API Limitations

Biometric Prompt
- Fake Prompt
- Everlasting Biometric

Android Protected Confirmation
- Illegitimate User
- Overwriting Confirmation

Biometric Prompt

Biometric Authorization
Use your biometric credentials to authorize the transfer of 9000 USD to Alice

Cancel
API Limitations

Biometric Prompt
- Fake Prompt
- Everlasting Biometric

Android Protected Confirmation
- Illegitimate User
- Overwriting Confirmation
API Limitations

**Biometric Prompt**
- Fake Prompt
- Everlasting Biometric

**Android Protected Confirmation**
- Illegitimate User
- Overwriting Confirmation
API Limitations

- **Biometric Prompt**
  - Fake Prompt
  - Everlasting Biometric

- **Android Protected Confirmation**
  - Illegitimate User
  - Overwriting Confirmation
Goals

Usability Goals
- Easy to use for developers
- Use existing Android APIs

Security Goals
- OS Compromise
  - TEE Usage
  - Key Attestation
  - User Awareness
  - Server Verification
- Physical Attacks
  - User Awareness
  - Physical Authorization
SARA’s Architecture

**TEE-enforced Android APIs**
- App Attestation
- Android Protected Confirmation
- Biometric Prompt

**SARA**
- Authorization Protocol (Biometric Confirmation)
  - Server Module
  - Android Library
SARA’s Process  The User’s Experience

Enable authorization -> One time only process

Keypair generation and Attestation

Authorize Action

Biometric Prompt Displayed

User provides biometric input (i.e., fingerprint)

Prompt gets signed upon user’s valid biometric input

Confirmation Prompt Displayed

User sees Android Protected Confirmation Prompt

User presses hardware button to accept prompt

Second signature takes place and sent to server for verification
SARA’s Process  The User’s Experience

Enable authorization - One time only process

Keypair generation and Attestation

Authorize Action

Biometric Prompt Displayed

User provides biometric input (i.e., fingerprint)

Prompt gets signed upon user’s valid biometric input

Confirmation Prompt Displayed

User sees Android Protected Confirmation Prompt

User presses hardware button to accept prompt

Second signature takes place and sent to server for verification

Biometric Authorization

Use your biometric credentials to authorize the transfer of 9000 USD to Alice

Cancel
<table>
<thead>
<tr>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable authorization -&gt; One time only process</td>
</tr>
<tr>
<td>Keypair generation and Attestation</td>
</tr>
<tr>
<td>Authorize Action</td>
</tr>
<tr>
<td>Biometric Prompt Displayed</td>
</tr>
<tr>
<td>User provides biometric input (i.e., fingerprint)</td>
</tr>
<tr>
<td>Prompt gets signed upon user’s valid biometric input</td>
</tr>
<tr>
<td>Confirmation Prompt Displayed</td>
</tr>
<tr>
<td>User sees Android Protected Confirmation Prompt</td>
</tr>
<tr>
<td>User presses hardware button to accept prompt</td>
</tr>
<tr>
<td>Second signature takes place and sent to server for verification</td>
</tr>
</tbody>
</table>

**Confirmation Prompt**

- **Power Button**: Double-press power to confirm
- **Volume Buttons**: Cancel
- **Prompt**: You are going to transfer 9000 USD to Alice
SARA’s Process  The Developer’s Experience

Android -> Import Library -> Setup Protocol -> Authorize Action

Server -> Import Library -> Setup Protocol -> Authorize Action
**SARA’s Security Evaluation**

**ProVerif Model**
- Model SARA’s authorization protocol in ProVerif’s cryptographic protocol verifier.

**Verify that SARA’s protocol satisfies the following security goals for any action undertaken by a server:**
- The legitimate user sees the action the server performs
- The legitimate user physically authorizes the action the server performs
- Server has a guarantee the that the action has been authorized by the legitimate user

**Attacks on Incomplete Protocols**
- Modeled alternate protocols in ProVerif to show the possible attacks on them due to their limitations
User Study

Comparison between Native API and SARA’s API

Two identical tasks divided into 3 subtasks each

Answers to two questions:

• Does using SARA make it easier for developers to use Android’s TEE-enforced APIs?
• How long does it take for a developer to learn how to use SARA?
## Completion Results

<table>
<thead>
<tr>
<th>Completed after 105 minutes</th>
<th>Native Task</th>
<th>Library Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtask-1: Successfully created a keypair(s) with the requisite properties</td>
<td>0/14</td>
<td>14/14</td>
</tr>
<tr>
<td>Subtask-2: Successfully created the confirmation and biometric prompts</td>
<td>0/14</td>
<td>14/14</td>
</tr>
<tr>
<td>Subtask-3: Successfully attested the keypair(s) on the server</td>
<td>0/14</td>
<td>14/14</td>
</tr>
</tbody>
</table>
# Evaluation Survey Summary

<table>
<thead>
<tr>
<th></th>
<th>SARA</th>
<th>Native API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Experience</td>
<td>93%</td>
<td>0%</td>
</tr>
<tr>
<td>Preference of Usage</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>SUS Score</td>
<td>95.18</td>
<td>11.61</td>
</tr>
</tbody>
</table>
Conclusion

- SARA is easy to use
- SARA uses existing APIs
- SARA provides root resiliency
- SARA even provides resilience against physical attacks
- SARA’s security has been evaluated using ProVerif
- SARA’s usability has been evaluated through a user study
THANK YOU!!!

Any Questions?

imran8@purdue.edu