Mobile, Biometric Bitlocker

Team sdmay19-23

Yousef Al Absi
Cole Alward
Morgan Anderson
Ammar Khan
Justin Kuhn
Larisa Thys

Advisor / Client

Akhilesh Tyagi

Technical Advisor

Timothy Dee

Problem

Android phones lack the means to safely and securely perform symmetric encryption due to the absence of a Trusted Platform Module (TPM) to store a private key. This unavailability during the encryption process means that the private key must be stored on the device. This presents a security issue that the key could be located, and the data protected by the key could fall into malicious hands.

Solution

Through the use of a pressure-based Physical Unclonable Function (PUF), a mobile device is able to dynamically generate a private key. The private key could be used to authenticate against the public key to access data by simulating the functions of a TPM.

Users

Android users who have information deemed worthy of protecting

Usages

Provide robust security and authentication mechanisms for data protection on mobile devices

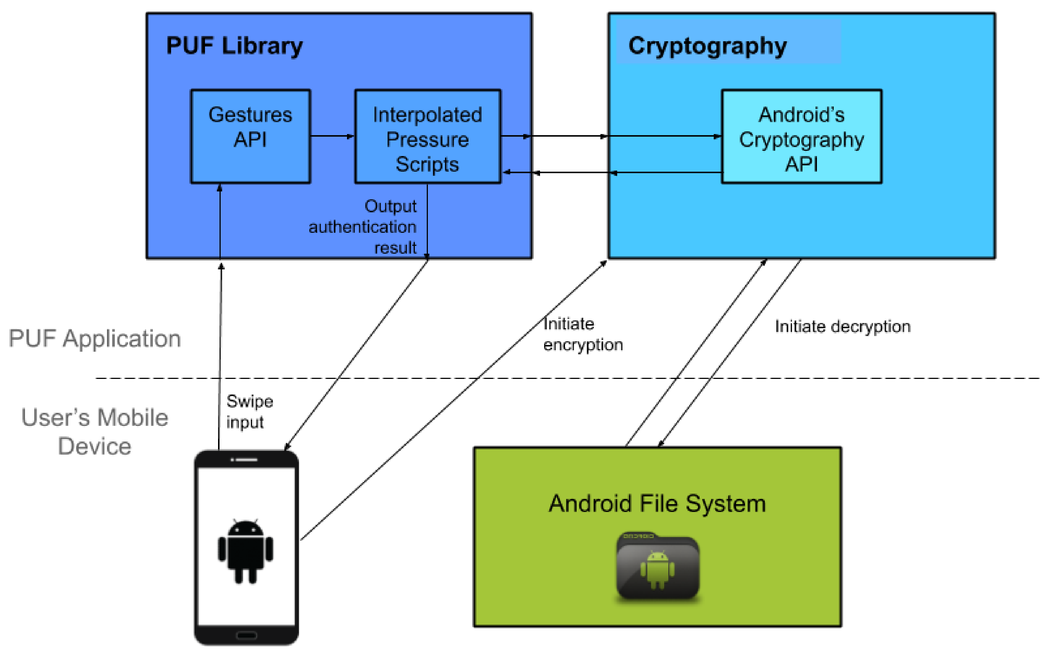
Components

PUF Library

Reads data from user-device gesture interaction and performs statistical analyses on the data to create a user-device pair and profile for authentication

Cryptography

Mimics TPM behavior based on the authentication result provided by the PUF library



Design Requirements

Functional Requirements

Application should create multiple profiles.

Application will complete the encryption or decryption process even if phone is locked.

PUF should encrypt and decrypt user data.

PUF should have an accuracy of at least 80%.

Non-Functional Requirements

Performance:
Response time for authentication should be less than 5 seconds.

Scalability:
Application should have more than 2 profiles.

Maintainability:
Repository should update the application automatically

Security:
Only the proper user can unlock the application.

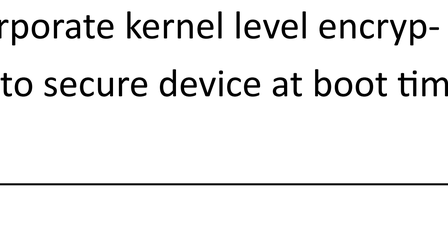
Data Integrity:
Data will be encrypted and decrypted successfully when provided the correct key.

Operational Environment

Android Devices:

SDK 27+

Nexus 7



Testing

Unit Testing
Components tested and simulated using Mockito

Acceptance Testing
Developers ensured all features met desired functional requirements

Manual Testing
Necessary for verification and validation of new UI feature functionalities

Integration Testing
Components tested whenever dependencies surfaced and after integration dependencies

Code Review
Code written in feature branches and reviewed by two other developers before merging to master branch

Test Cases

Test Case 1
User can create multiple profiles

Result: 2 users are authenticated using their profile

Test Case 2
Trace is authenticated within 5 seconds

Result: User is authenticated in under 5 seconds with 80% accuracy

Future Work

Extend Android lock screen API to integrate into PUF library

Create a custom launcher to integrate the PUF library extend lock screen functionality

Incorporate kernel level encryption to secure device at boot time