復旦大學
FUDAN UNIVERSITY

An Empirical Study on Fingerprint API Misuse with Lifecycle Analysis in Real-world Android Apps

Xin Zhang, Xiaohan Zhang, Zhichen Liu, Bo Zhao,
Zhemin Yang, Min Yang

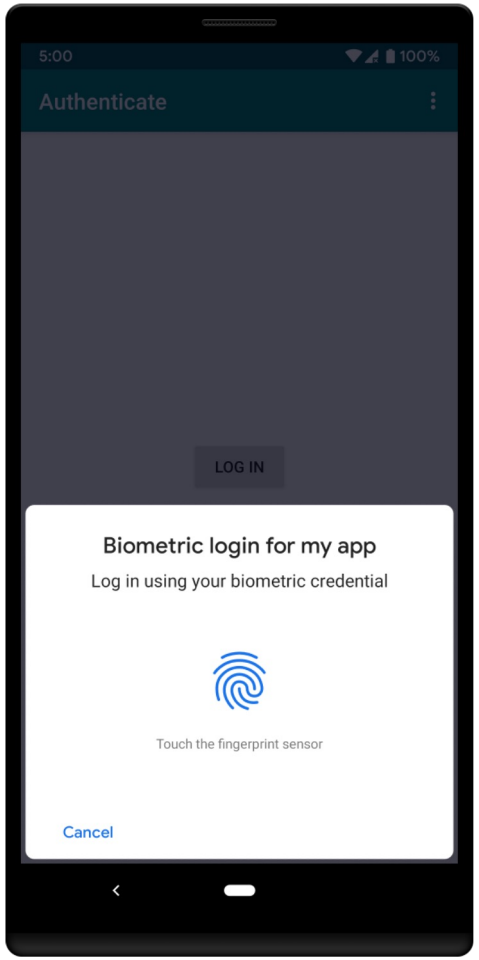
Fudan University

Fingerprint-based Authentication (FpAuth)

- Used commonly by mobile apps

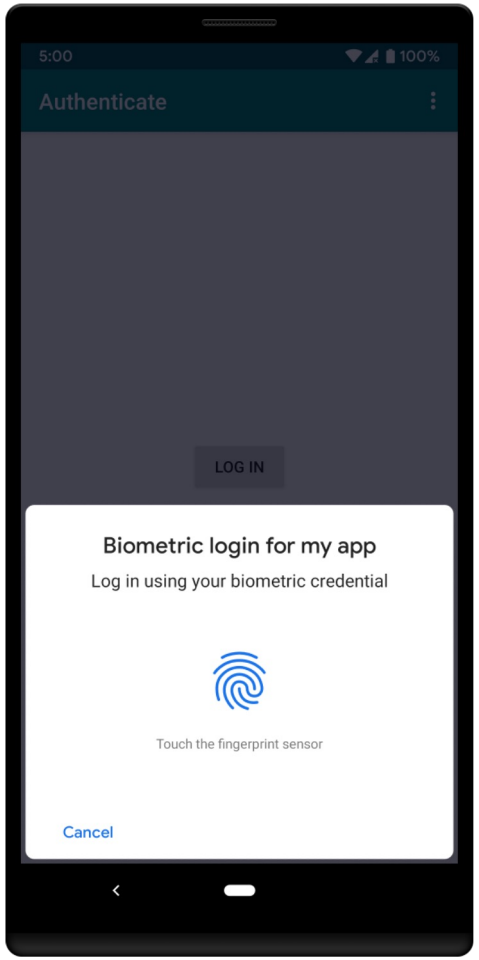
- Used in sensitive scenarios

FpAuth in Android Apps

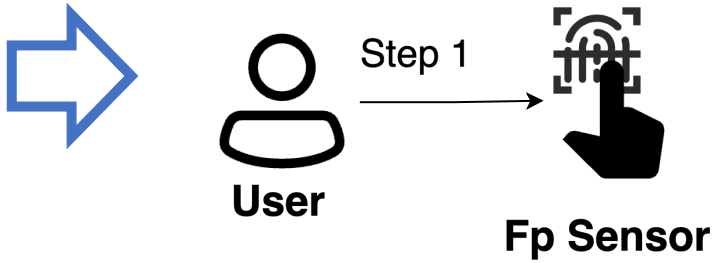


What happens when performing FpAuth?

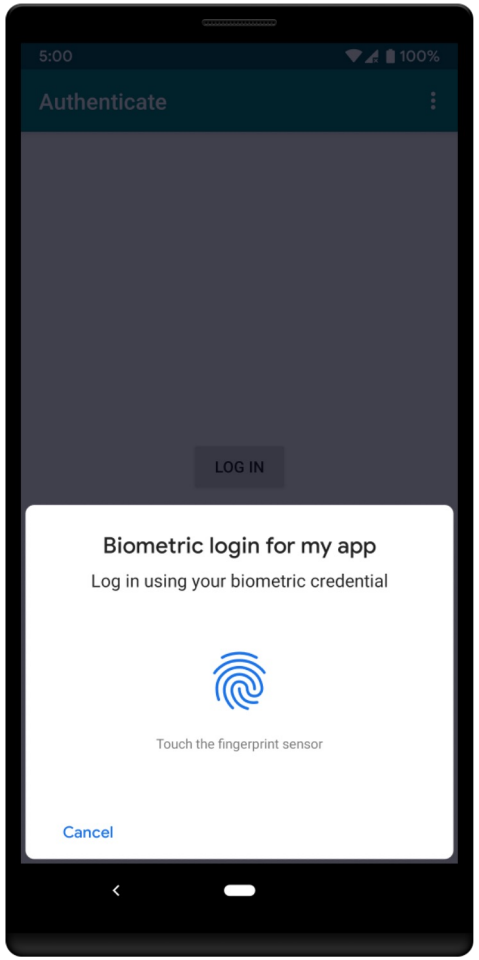
FpAuth in Android Apps



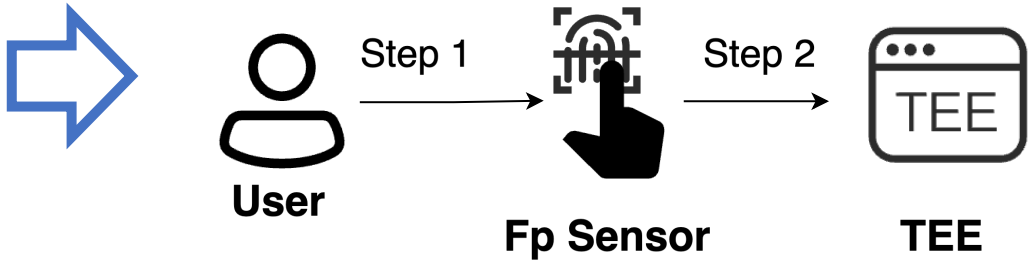
What happens when performing FpAuth?



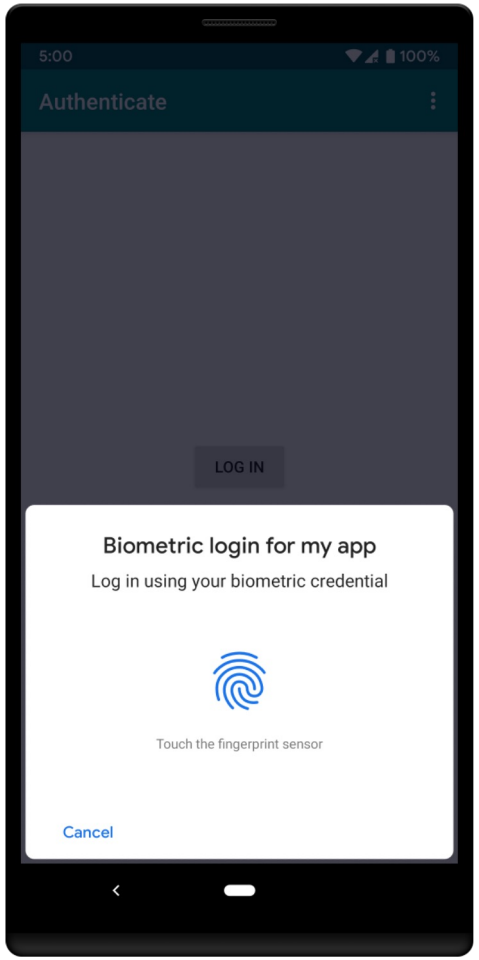
FpAuth in Android Apps



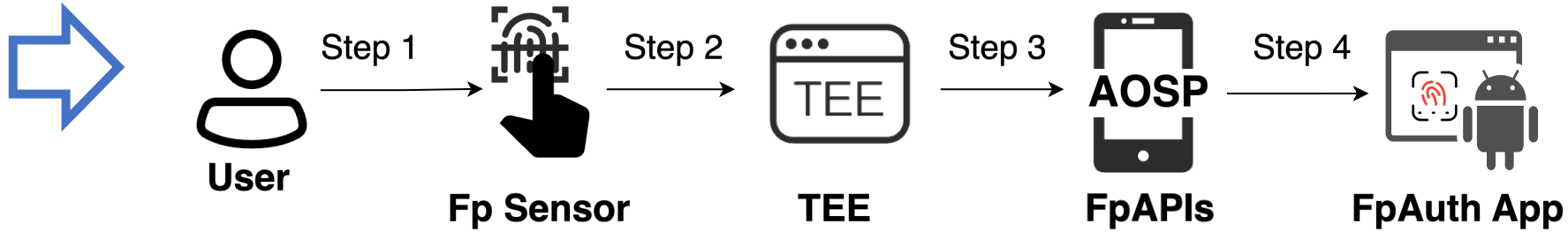
What happens when performing FpAuth?



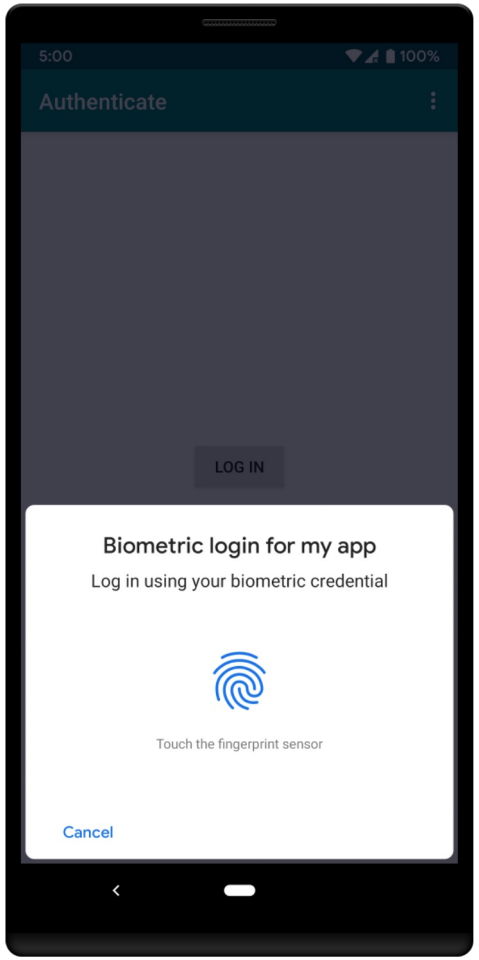
FpAuth in Android Apps



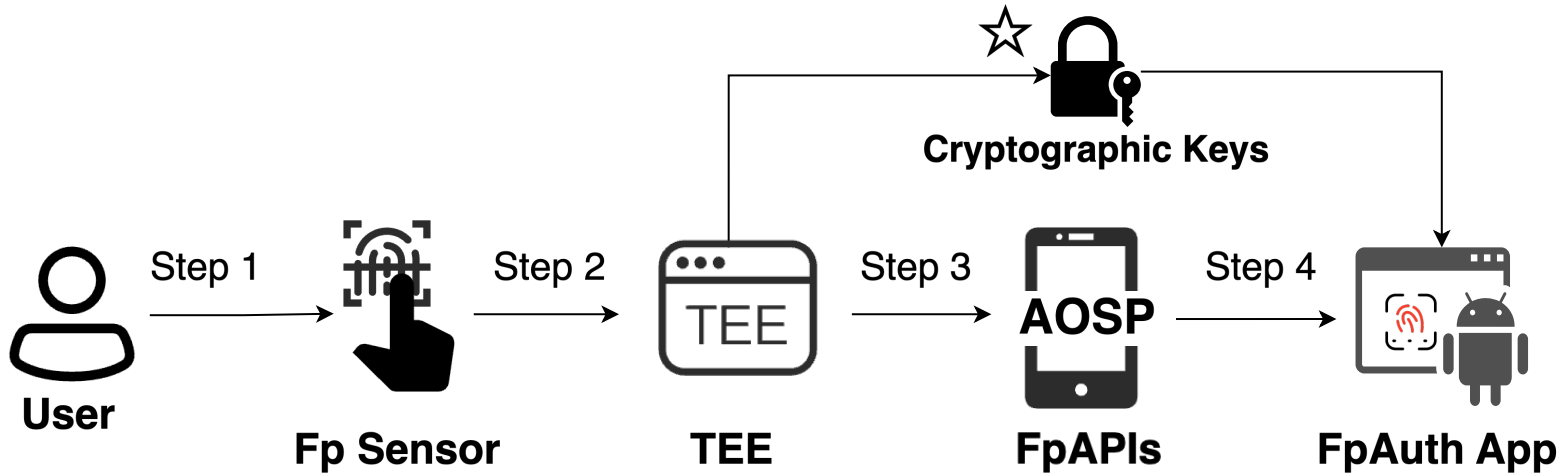
What happens when performing FpAuth?



FpAuth in Android Apps

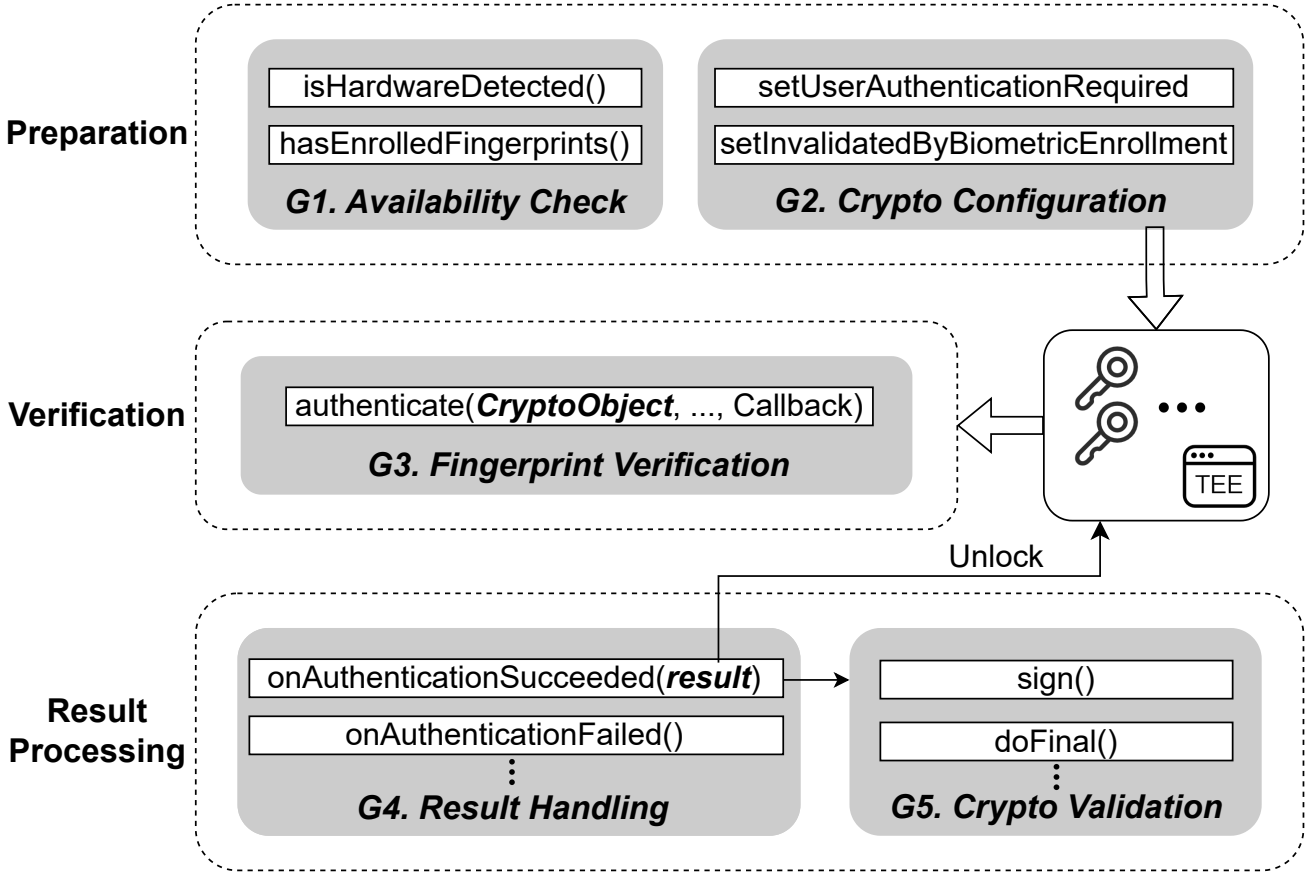


What happens when performing FpAuth?



FpAuth in Android Apps

For developers, how to implement?

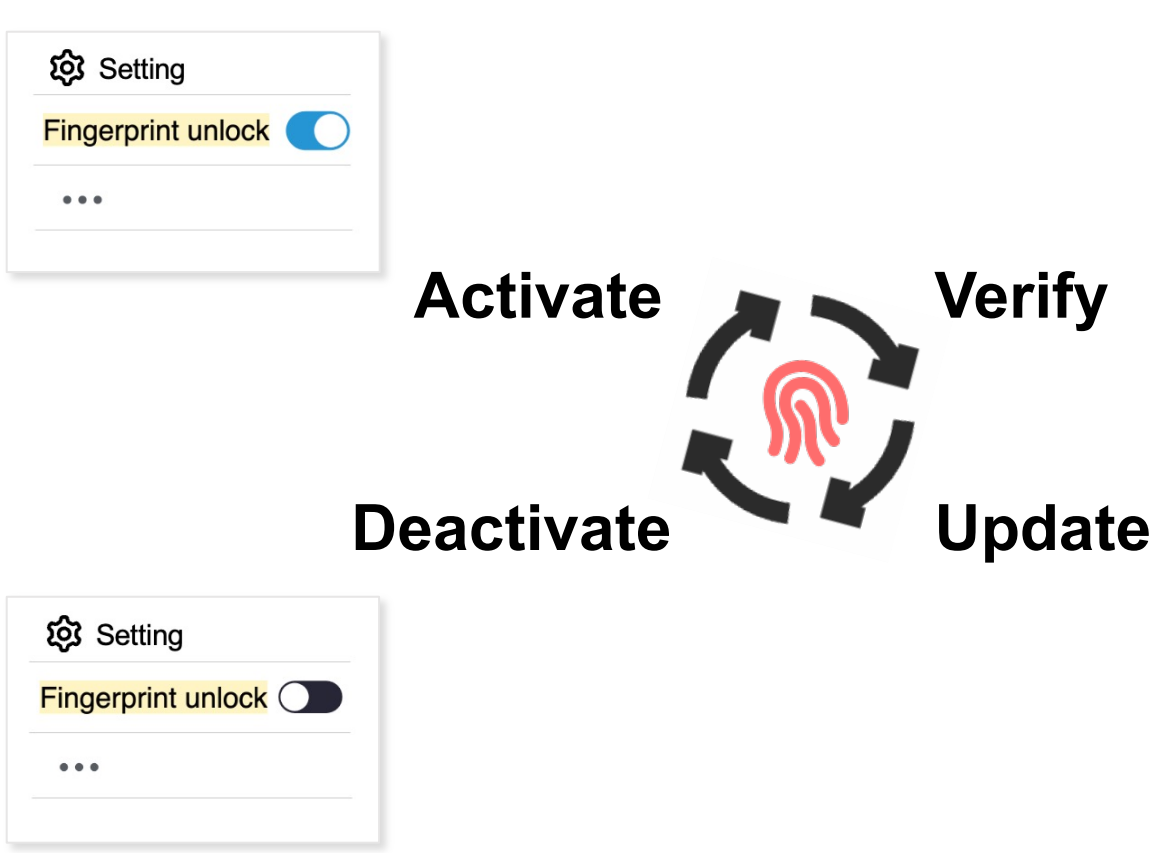


- Android APIs for FpAuth
  - ***Complicated***
    - 5 groups of related APIs
    - Combined with crypto
  - ***Evolving***
    - Issued in 2015
    - Migrated in 2018

Developers are prone to misuse

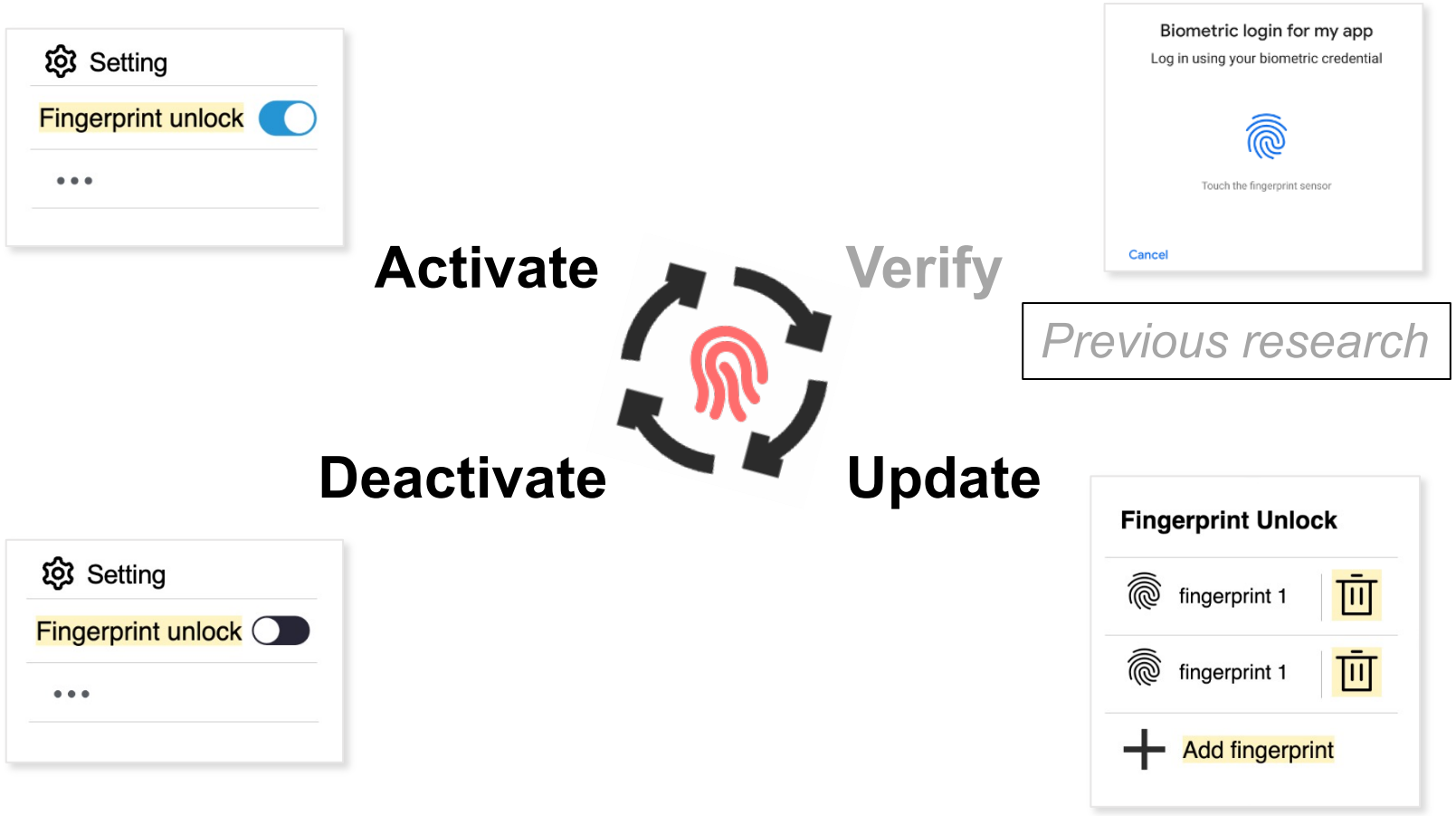
FpAuth Lifecycle Analysis

- Systematic analysis from a new perspective – FpAuth lifecycle



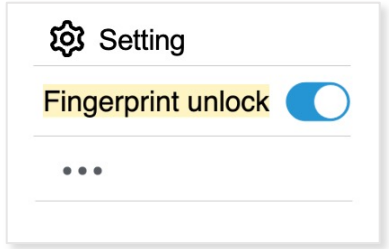
FpAuth Lifecycle Analysis

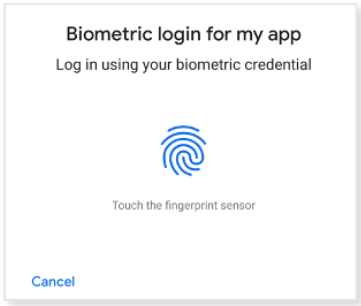
- Systematic analysis from a new perspective – FpAuth lifecycle



FpAuth Lifecycle Analysis

- Systematic analysis from a new perspective – FpAuth lifecycle





Activate

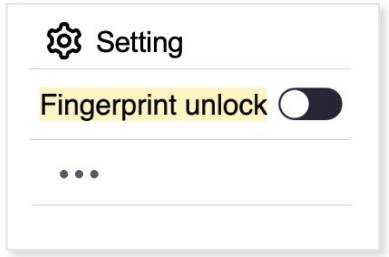
Verify

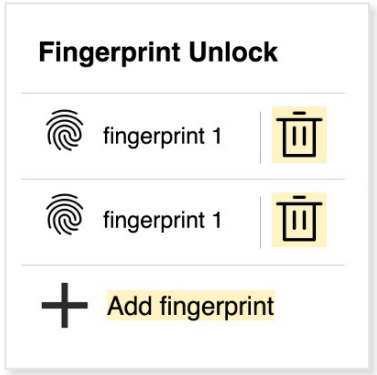
Deactivate

Update

Previous research

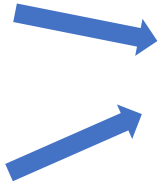
New issues!





Four Types of Misuses

- Misuse #1: Obsolete API Usage
- Misuse #2: Inadequate Cryptographic Validation



Verification

- Misuse #3: Unauthorized Fingerprint Deactivation

Deactivation

- Misuse #4: Mishandled Fingerprint Updates

Update

Fingerprint API Misuse

- **Misuse #1: Obsolete API Usage**
  - Local software-level attacker
  - Using obsolete FpAPI makes it vulnerable to UI attack [PHYjacking-NDSS'22]

[PHYjacking-NDSS'22] Wang, Xianbo, et al. "PHYjacking: Physical Input Hijacking for Zero-Permission Authorization Attacks on Android." NDSS. 2022.
[Broken Fingers-NDSS'18] Bianchi, Antonio, et al. "Broken Fingers: On the Usage of the Fingerprint API in Android." NDSS. 2018.

Fingerprint API Misuse

- **Misuse #1: Obsolete API Usage**
  - Local software-level attacker
  - Using obsolete FpAPI makes it vulnerable to UI attack [PHYjacking-NDSS'22]

- **Misuse #2: Inadequate Cryptographic Validation**
  - Local OS-level attacker
  - Failing to correctly bind crypto → unreliable verification result [Broken Fingers-NDSS'18]

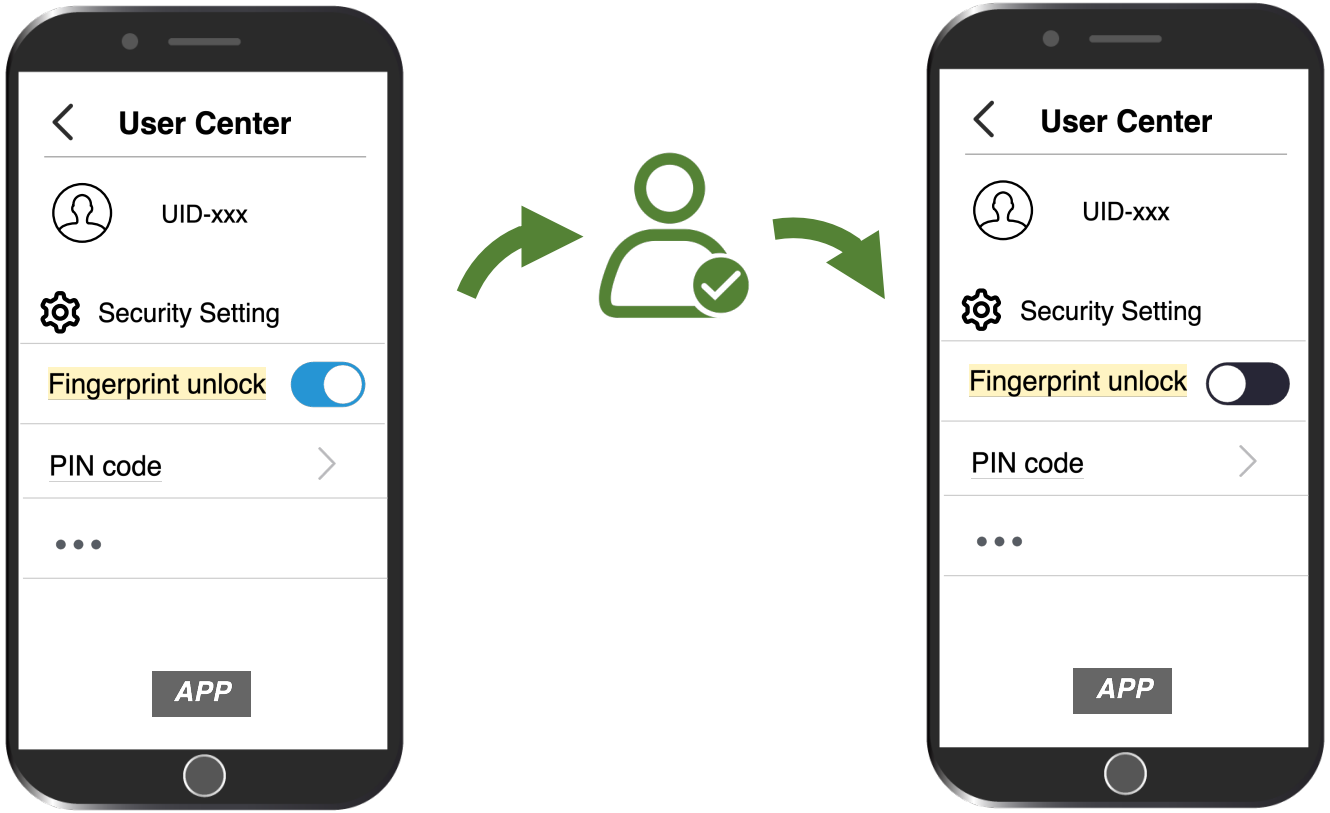
```
    // Pattern 2: Null Cryptographic Object
1   fingerprintmanager.authenticate(null);  // Misuse: Setting parameter crypto to
    null causes FpAuth is not protected by cryptographic key

    // Pattern 3: Failed Key Binding
2   keyGenerator.init(KEY_NAME)
3       ...
4       .setUserAuthenticationRequired(false);  // Misuse: Not explicitly setting this
    API to true causes failure to bind FpAuth with cryptographic key
5   fingerprintmanager.authenticate(CryptoObject(cipher.init(KEY_NAME)), ...);

    // Pattern 4: Failed Result Validation
6   keyGenerator.init(KEY_NAME)
7       ...
8       .setUserAuthenticationRequired(true);
9   fingerprintmanager.authenticate(CryptoObject(cipher.init(KEY_NAME)), ...);

10  public void onAuthenticationSucceeded(result) {
11      log.info("FpAuth succeeded!");  // Misuse: Not validating the result using
    cryptographic key causes the result can be forged
12  }
```

Code example of M2

[PHYjacking-NDSS'22] Wang, Xianbo, et al. "PHYjacking: Physical Input Hijacking for Zero-Permission Authorization Attacks on Android." NDSS. 2022.
[Broken Fingers-NDSS'18] Bianchi, Antonio, et al. "Broken Fingers: On the Usage of the Fingerprint API in Android." NDSS. 2018.

Fingerprint API Misuse

- **Misuse #3: Unauthorized Fingerprint Deactivation**

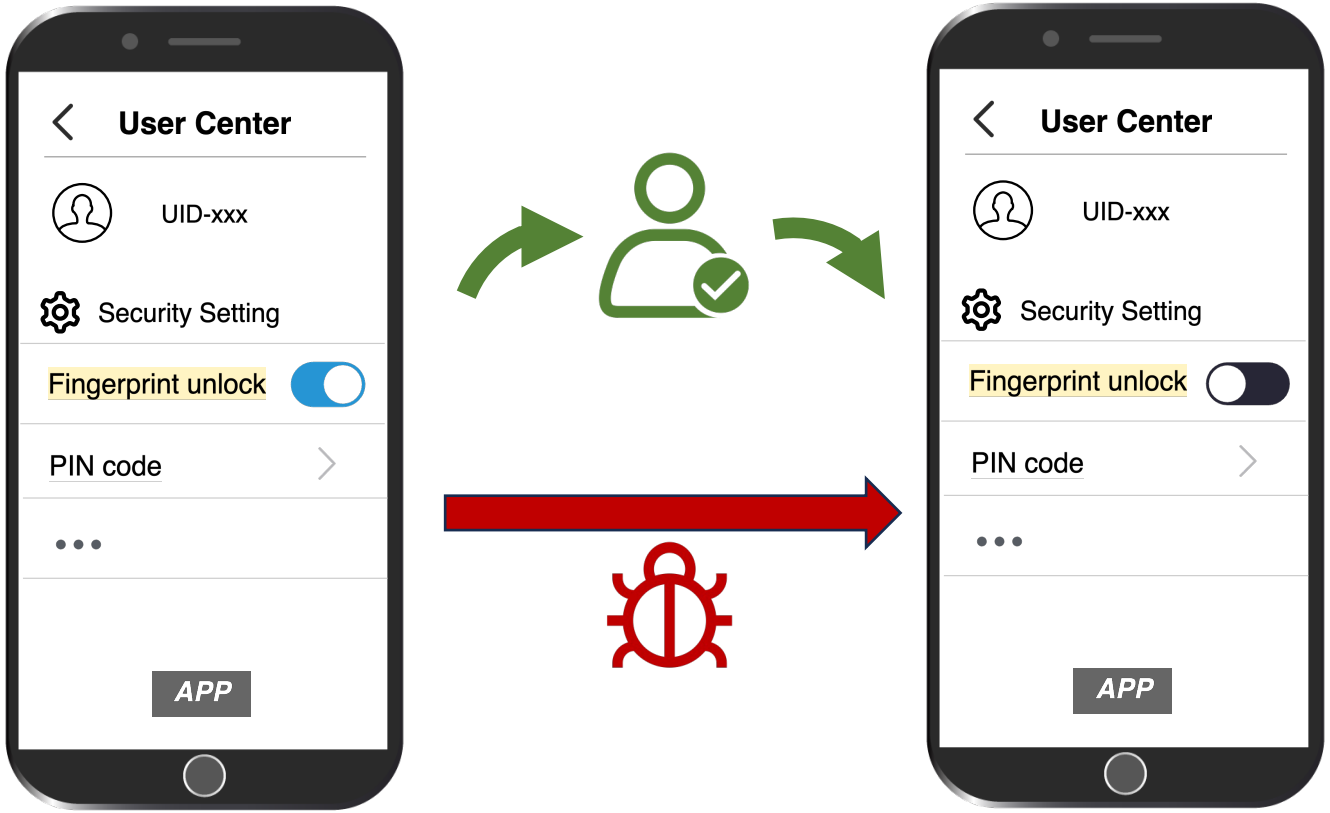


Threat Model

- Physical Curious Attacker
  - Has a one-time opportunity to access
  - Aims for permanent access

Fingerprint API Misuse

- **Misuse #3: Unauthorized Fingerprint Deactivation**



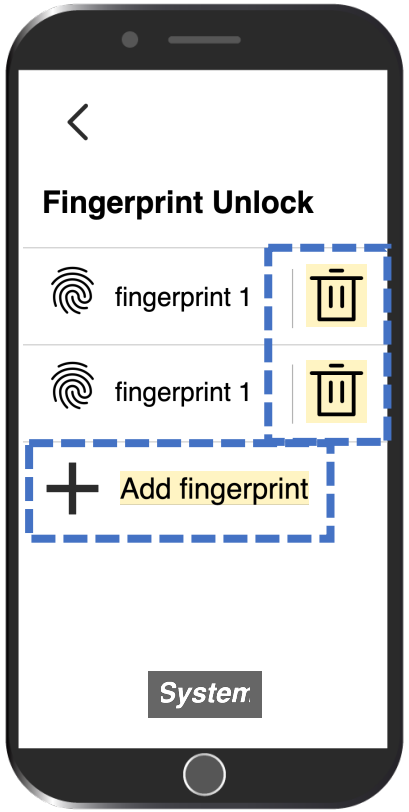
Threat Model

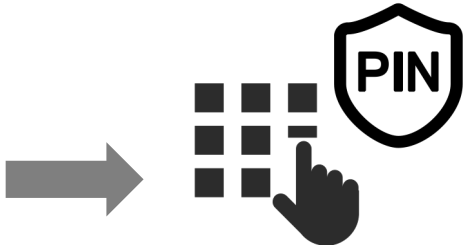
- Physical Curious Attacker
  - Has a one-time opportunity to access
  - Aims for permanent access

FpAuth protection can be disabled, allowing unauthorized access

Fingerprint API Misuse

- **Misuse #4: Mishandled Fingerprint Updates**

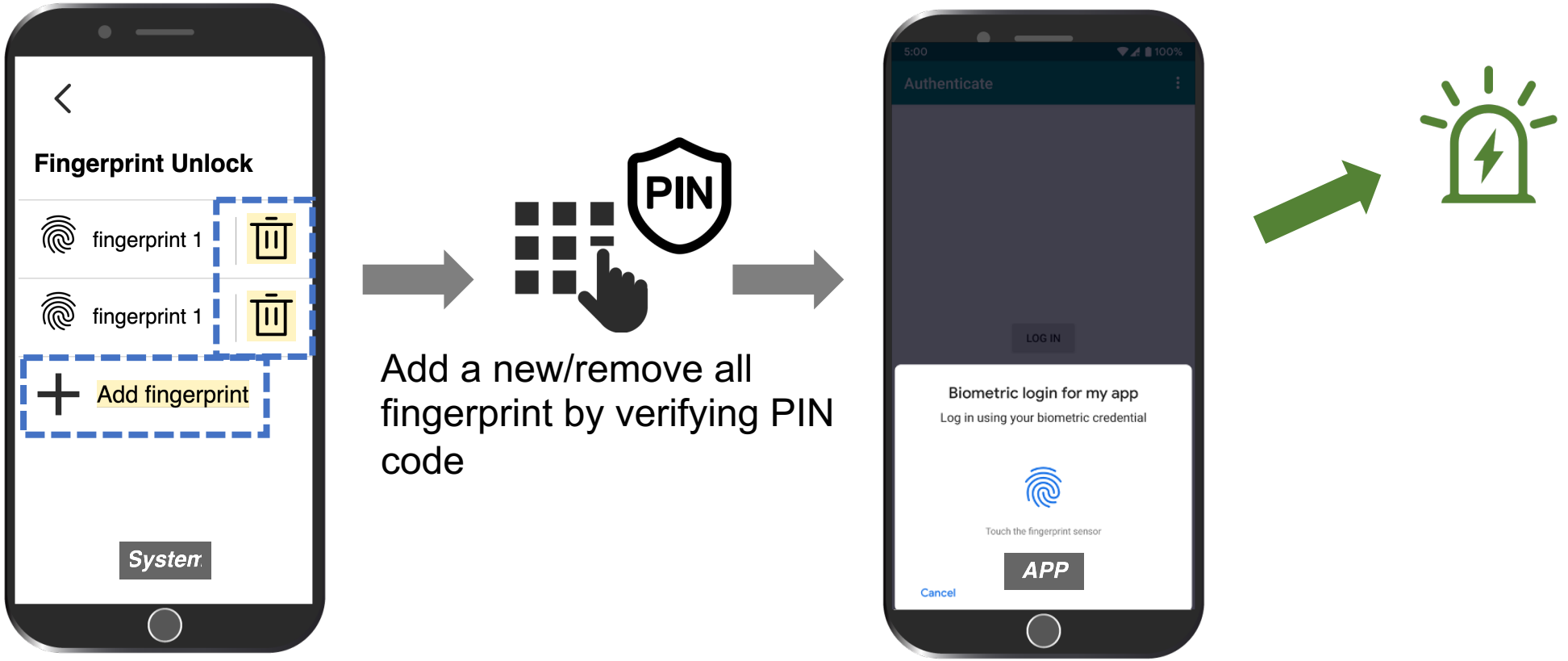




Add a new/remove all fingerprint by verifying PIN code

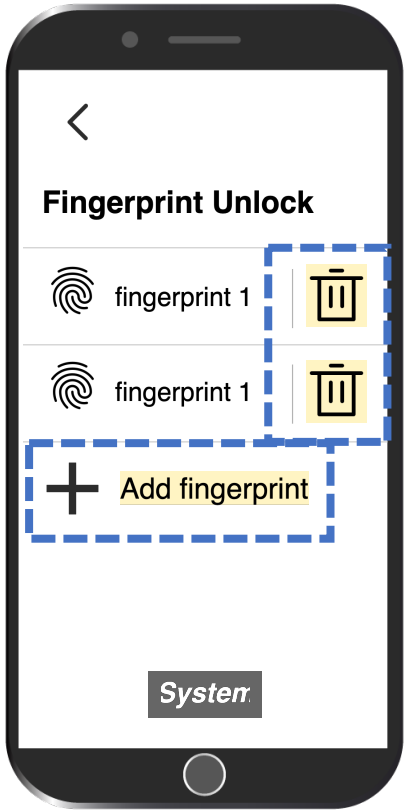
Fingerprint API Misuse

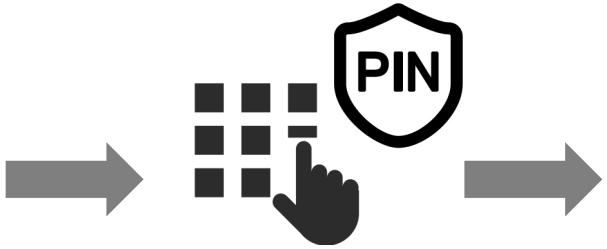
- **Misuse #4: Mishandled Fingerprint Updates**



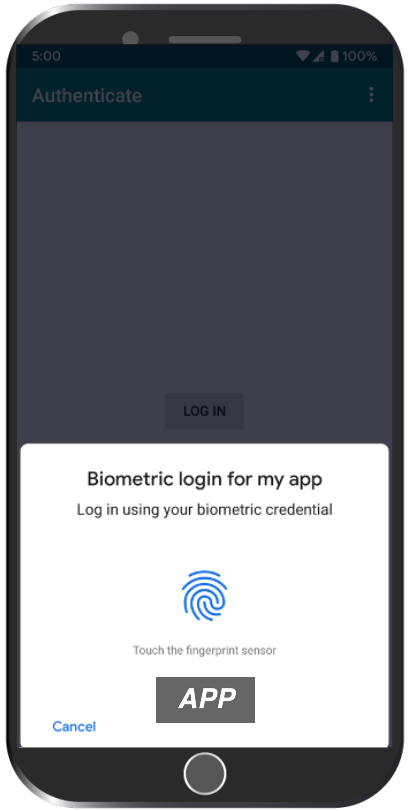
Fingerprint API Misuse

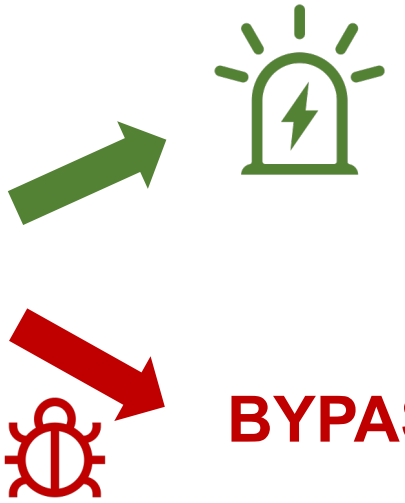
- **Misuse #4: Mishandled Fingerprint Updates**





Add a new/remove all fingerprint by verifying PIN code





BYPASS

Degrade dual protection to single, lowering security level

Fingerprint API Misuse

- **Misuse #4: Mishandled Fingerprint Updates**

Threat Model

- Physical Intimate Attacker
  - Knows locking PIN
  - Aims to bypassing FpAuth

Fingerprint API Misuse

- **Misuse #4: Mishandled Fingerprint Updates**

Threat Model

- Physical Intimate Attacker
  - Knows locking PIN
  - Aims to bypassing FpAuth

PIN is weak!

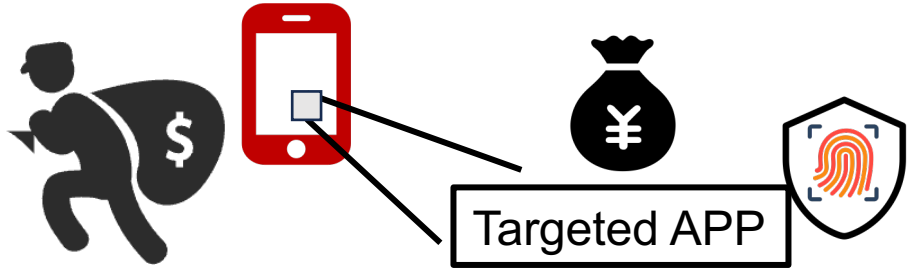
- Shoulder surfing
- Guessing attacks
- ……

Fingerprint API Misuse

- **Misuse #4: Mishandled Fingerprint Updates**

Threat Model

- Physical Intimate Attacker
  - Knows locking PIN
  - Aims to bypassing FpAuth



PIN is weak!

- Shoulder surfing
- Guessing attacks
- ……

Fingerprint API Misuse

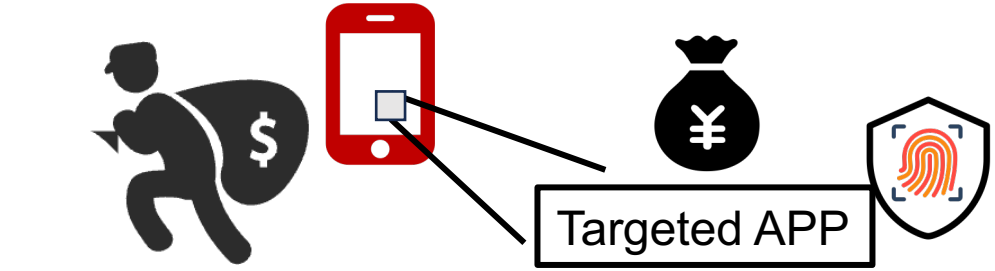
- **Misuse #4: Mishandled Fingerprint Updates**

Threat Model

- Physical Intimate Attacker
  - Knows locking PIN
  - Aims to bypassing FpAuth

PIN is weak!

- Shoulder surfing
- Guessing attacks
- ……



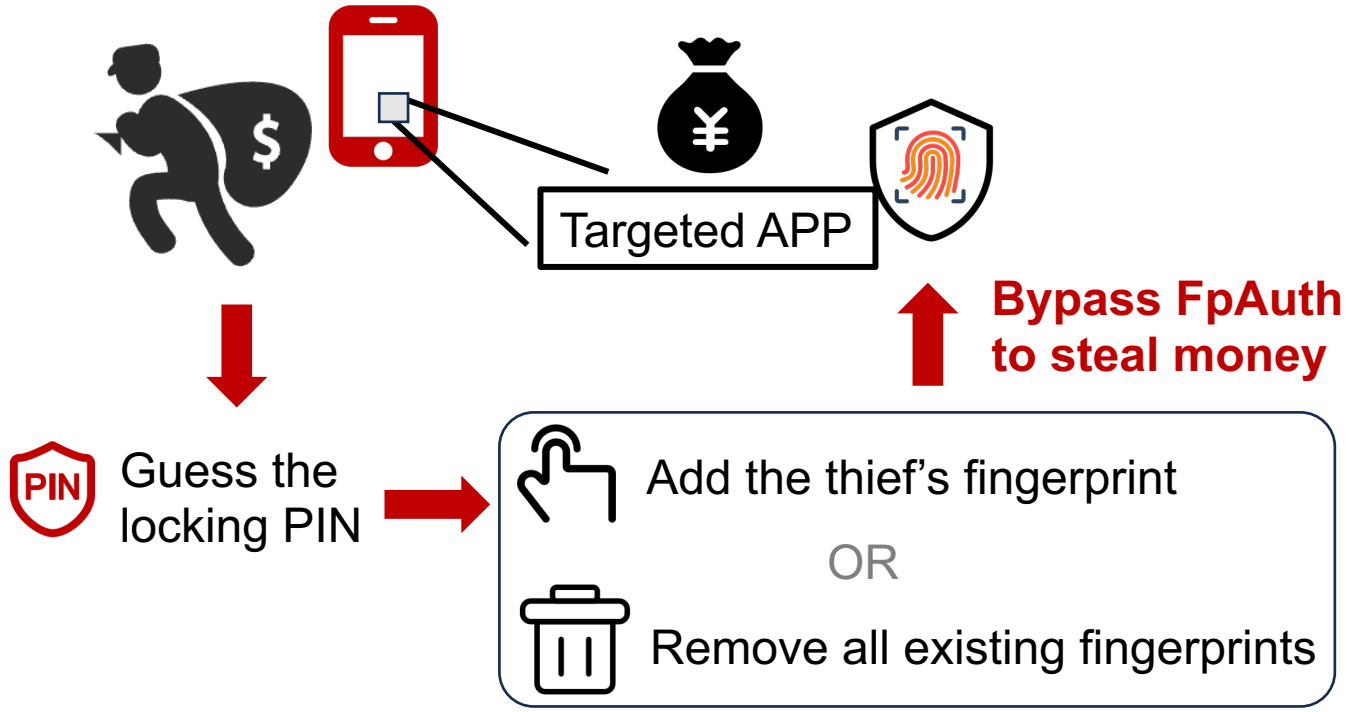
Fingerprint API Misuse

• Misuse #4: Mishandled Fingerprint Updates

Threat Model

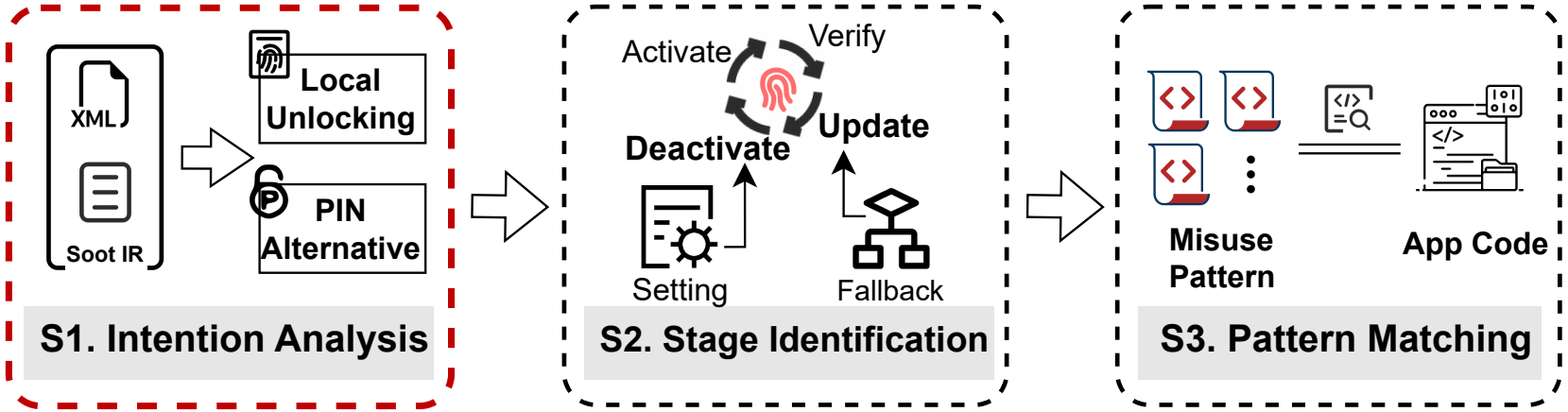
- Physical Intimate Attacker
  - Knows locking PIN
  - Aims to bypassing FpAuth

PIN is weak!
- ➢ Shoulder surfing
- ➢ Guessing attacks
- ➢ ……



Misuse Detection

- Step 1. Intention Analysis



Considering two special app intentions

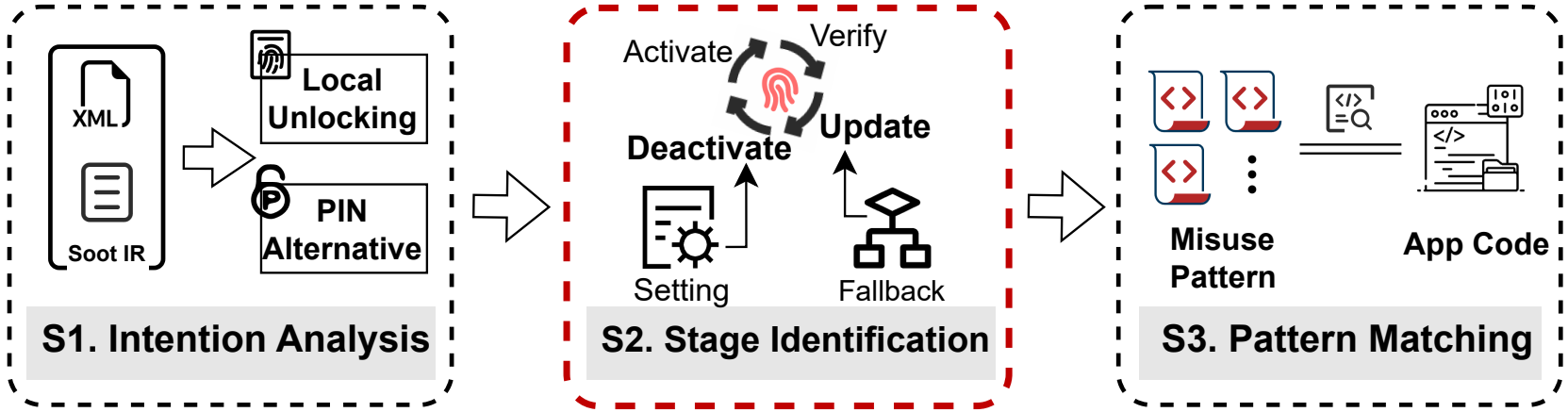
Local Unlocking → Not considering M2. *Inadequate Cryptographic Validation*

PIN Alternative → Not considering M4. *Mishandled Fingerprint Updates*

- Detection
  - UI semantics
  - Code semantics

Misuse Detection

- Step 2. Stage Identification



Identify two special stages

Deactivation

Fingerprint unlock

"Switch" component

Authenticate API

Call chain

onCheckedChanged

Update

Add

setInvalidatedByBiometricEnrollment

Remove

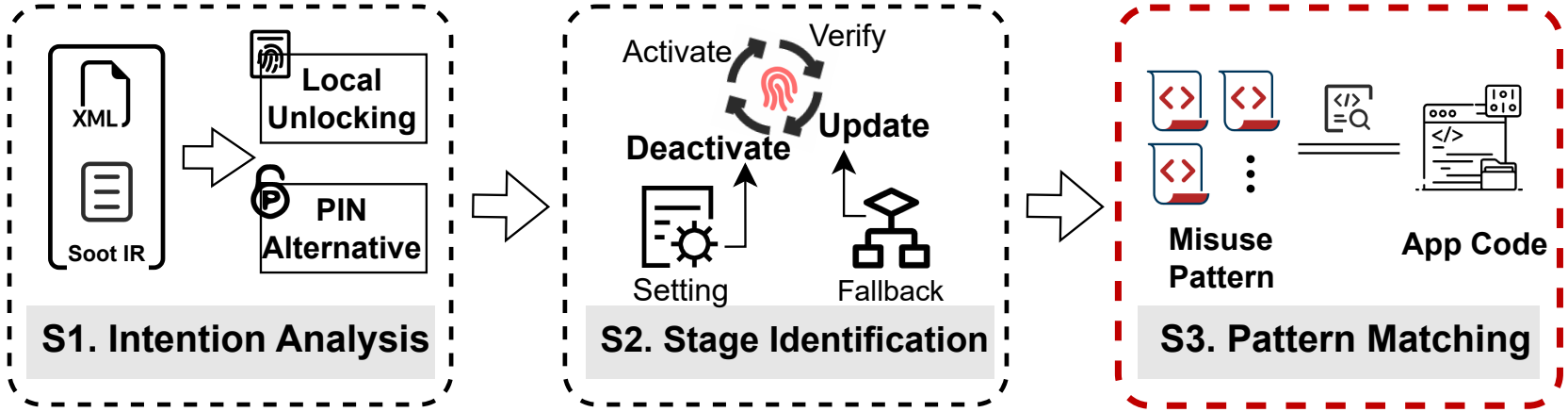
hasEnrolledFingerprints

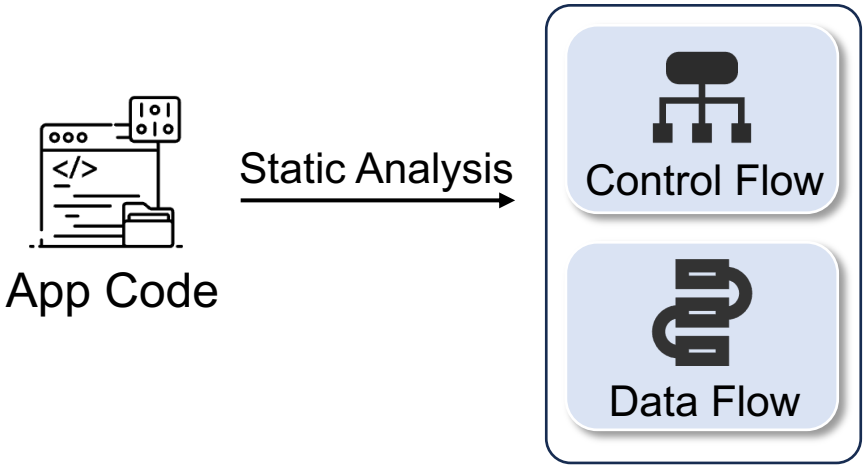
FpAuth

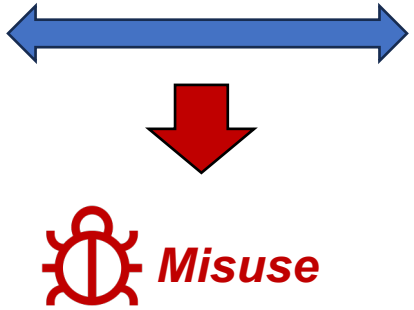
No Auth

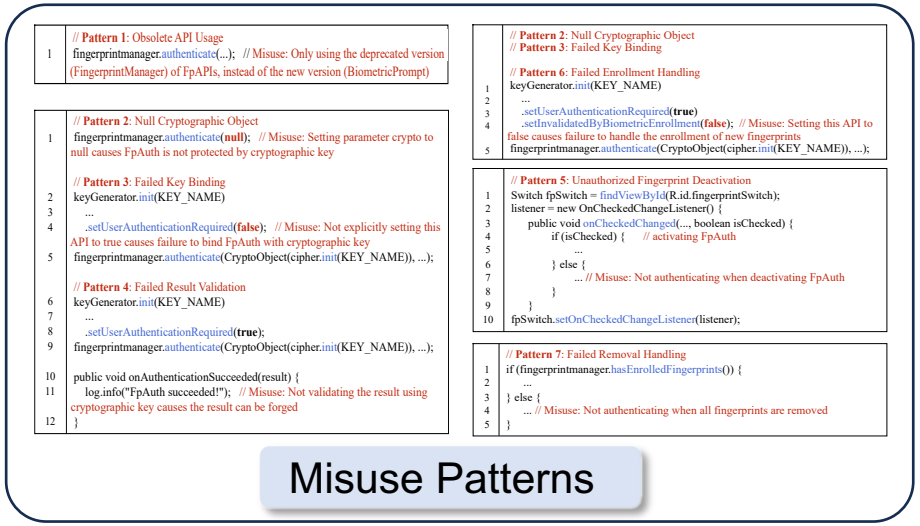
Misuse Detection

- Step 3. Pattern Matching









FpAPI Misuses in Real-world Apps

- Dataset

# Apps	Market	# Categories
29,848	Google Play	37
35,238	Huawei Market	16

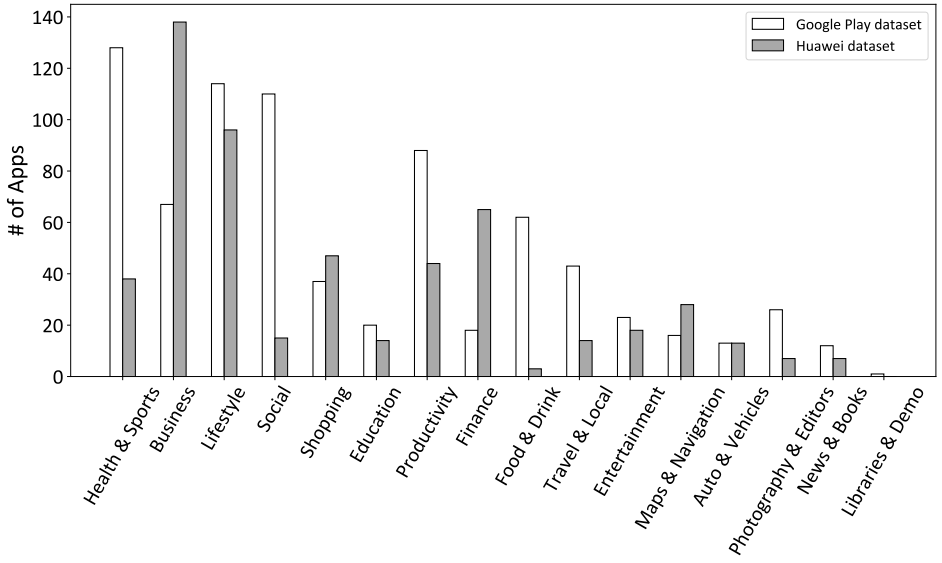
FpAPI Misuses in Real-world Apps

- Dataset

# Apps	Market	# Categories
29,848	Google Play	37
35,238	Huawei Market	16

- Identifies *1,333* FpAuth apps
  - Sensitive categories
  - Collectively over 218 billion installations

FpAuth exhibits significant impact!



FpAPI Misuses in Real-world Apps

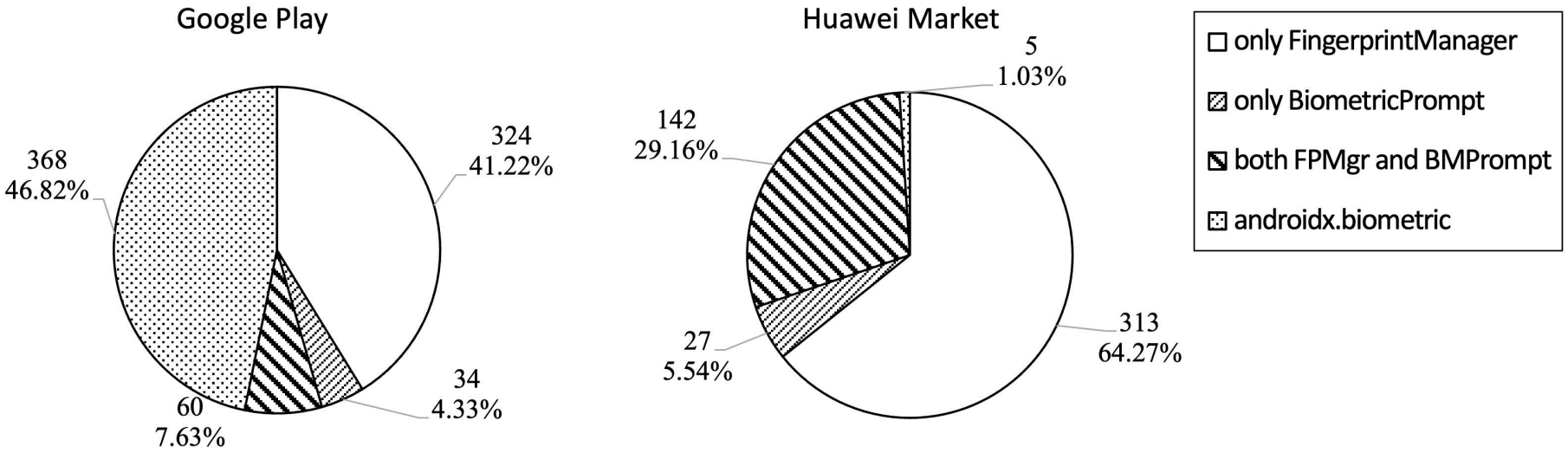
- Overall Result
  - Nearly all FpAuth apps (97.15%) have at least one misuse
  - Each type of misuses are present in nearly or over half of FpAuth apps

Four types of misuses in 1,333 FpAuth apps

Misuse	# Apps (Proportion)	# Misuse	# Apps (Proportion)
M1. Obsolete API	637 (47.79%)	1	179 (13.43%)
M2. Weak Crypto	999 (74.94%)	2	377 (28.28%)
M3. Unauthorized Deactivation	871 (65.34%)	3	488 (36.61%)
M4. Mishandled Updates	894 (67.07%)	4	251 (18.83%)
# Apps without Misuse			38 (2.85%)

FpAPI Misuses in Real-world Apps

- Findings – *Obsolete use of FpAPI (M1)*
  - **Five years after the release of the new secure FpAPI, 47.79% still use the obsolete FpAPI (M1)**



  - *Androidx.biometric* library is recommended but underutilized

FpAPI Misuses in Real-world Apps

- Findings – *Ineffective protection (M3)*

Verification practices when activating/deactivating FpAuth (M3)

	Verified Deactivation	Verified Activation Only	Verified None
# Apps	462 (34.66%)	120 (9.00%)	751 (56.34%)

  - 120 (9%) apps authenticate when activating FpAuth but fail to authenticate when deactivating it ⇨ cannot guarantee the integrity of FpAuth protection

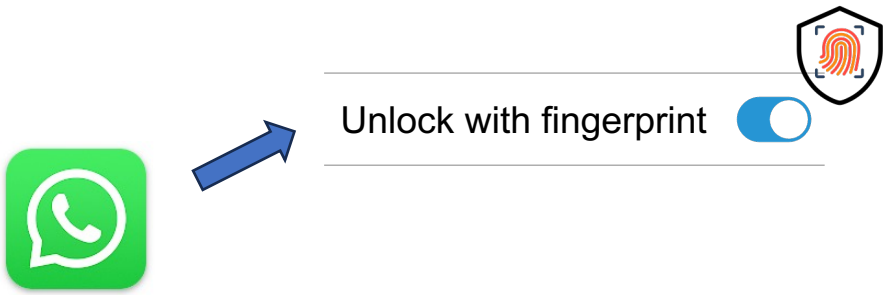
FpAPI Misuses in Real-world Apps

- Findings – *Ineffective protection (M3)*

Verification practices when activating/deactivating FpAuth (M3)

	Verified Deactivation	Verified Activation Only	Verified None
# Apps	462 (34.66%)	120 (9.00%)	751 (56.34%)

  - 120 (9%) apps authenticate when activating FpAuth but fail to authenticate when deactivating it ➡ cannot guarantee the integrity of FpAuth protection



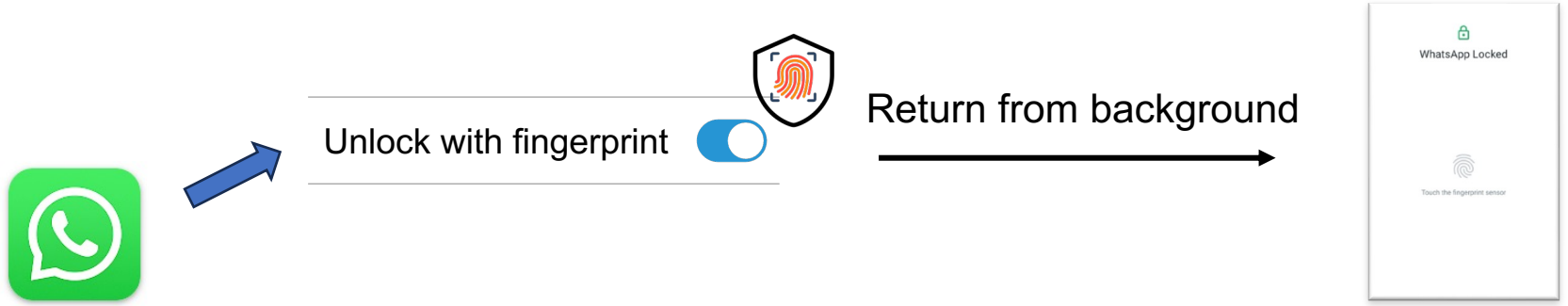
FpAPI Misuses in Real-world Apps

- Findings – *Ineffective protection (M3)*

Verification practices when activating/deactivating FpAuth (M3)

	Verified Deactivation	Verified Activation Only	Verified None
# Apps	462 (34.66%)	120 (9.00%)	751 (56.34%)

  - 120 (9%) apps authenticate when activating FpAuth but fail to authenticate when deactivating it ➡ cannot guarantee the integrity of FpAuth protection



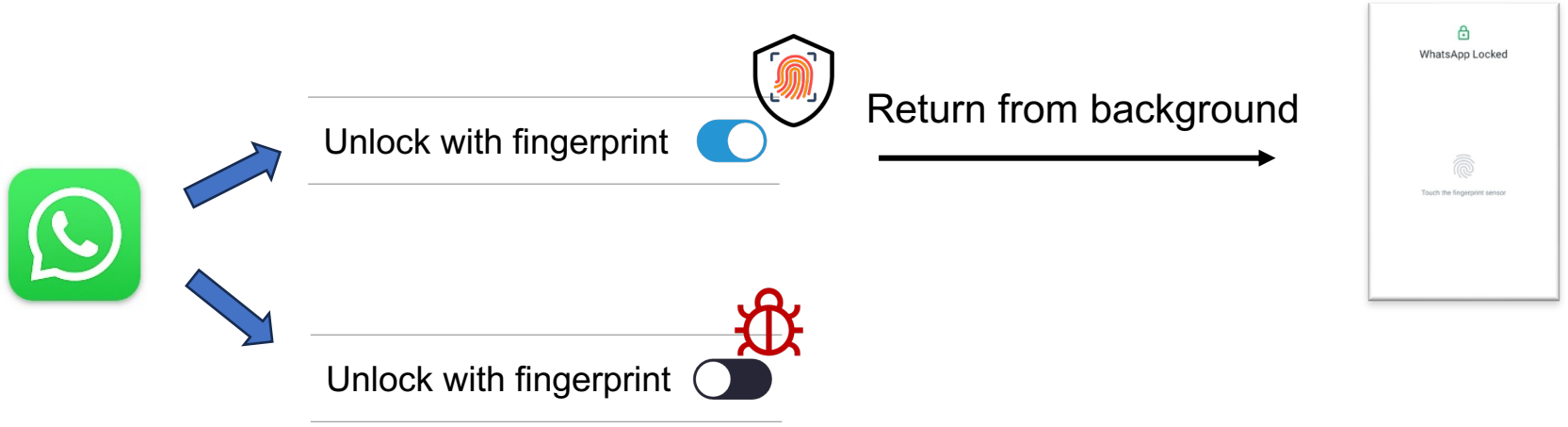
FpAPI Misuses in Real-world Apps

- Findings – *Ineffective protection (M3)*

Verification practices when activating/deactivating FpAuth (M3)

	Verified Deactivation	Verified Activation Only	Verified None
# Apps	462 (34.66%)	120 (9.00%)	751 (56.34%)

  - 120 (9%) apps authenticate when activating FpAuth but fail to authenticate when deactivating it ➡ cannot guarantee the integrity of FpAuth protection



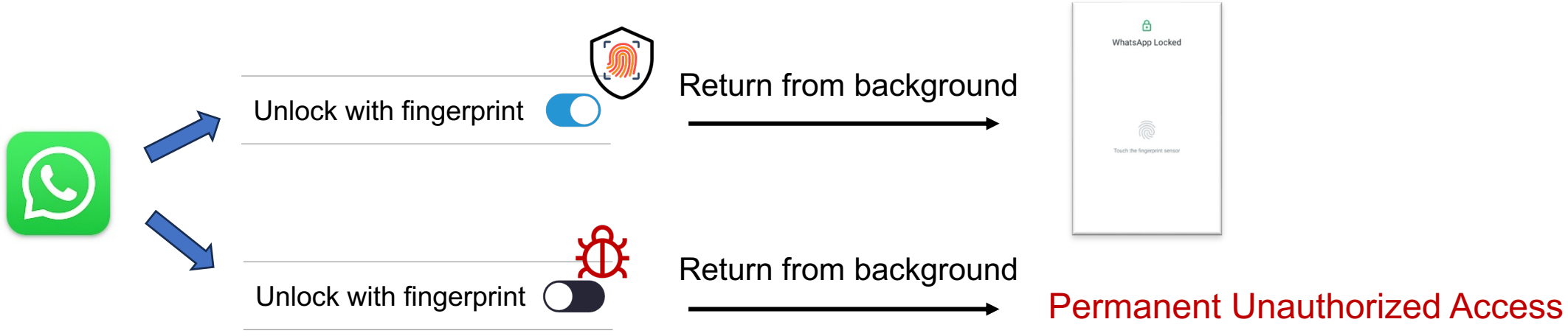
FpAPI Misuses in Real-world Apps

- Findings – *Ineffective protection (M3)*

Verification practices when activating/deactivating FpAuth (M3)

	Verified Deactivation	Verified Activation Only	Verified None
# Apps	462 (34.66%)	120 (9.00%)	751 (56.34%)

  - 120 (9%) apps authenticate when activating FpAuth but fail to authenticate when deactivating it ➡ cannot guarantee the integrity of FpAuth protection



FpAPI Misuses in Real-world Apps

- Findings – *Insufficient security awareness / knowledge (M4)*

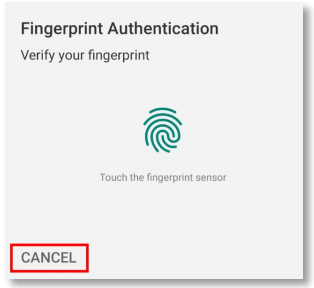
API	Description
setUserAuthenticationRequired	Binds the crypto with FpAuth
setInvalidatedByBiometricEnrollment	Invalidates the crypto when updating fingerprints

FpAPI Misuses in Real-world Apps

- Findings – *Insufficient security awareness / knowledge (M4)*

API	Description
setUserAuthenticationRequired	Binds the crypto with FpAuth
setInvalidatedByBiometricEnrollment	Invalidates the crypto when updating fingerprints

17 apps – Insufficient security awareness



➢ disallow PIN as alternative

Prioritizing functionality over security

setInvalidatedByBiometric Enrollment → False

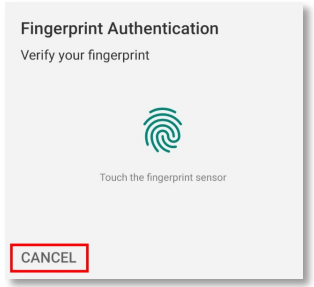
➢ configure to ignore fingerprint updates

FpAPI Misuses in Real-world Apps

- Findings – *Insufficient security awareness / knowledge (M4)*

API	Description
setUserAuthenticationRequired	Binds the crypto with FpAuth
setInvalidatedByBiometricEnrollment	Invalidates the crypto when updating fingerprints

17 apps – Insufficient security awareness



➢ disallow PIN as alternative

Prioritizing functionality over security

setInvalidatedByBiometric Enrollment → False

➢ configure to ignore fingerprint updates

5 apps – Insufficient security knowledge

setInvalidatedByBiometricEnrollment → True ✓

setUserAuthenticationRequired → False ✗

Nullify the security intention

Responsible Disclosure

- Report the issues to vulnerability repositories and developers
  - Apps from Google Play – CVE
  - Apps from Huawei Market – CNVD (Chinese)

	CVE ID	CNVD ID	Developer
Number	184	19	15

Summary

- Comprehensive study from the new perspective – lifecycle of FpAuth in apps
  - New threat models
  - Four common FpAPI misuse types
- Effective static tool to detect FpAPI misuses
- Large-scale measurement on real-world apps to reveal the significant security breaches

Thanks!

Xin Zhang
Fudan University
zhangx22@m.fudan.edu.cn

復旦大學
FUDAN UNIVERSITY