Characterizing the Security Facets of IoT Device Setup

Han Yang, Carson Kuzniar, Chengyan Jiang, Ioanis Nikolaidis, and Israat Haque

DALHOUSIE UNIVERSITY

PINet
Programmable and Intelligent Networking

Security Issues in Smart IoT Devices

- The number of connected IoT devices is projected to reach 29.42 billion by 2030 [1]
- Benefit daily life but also bring security/privacy issues.

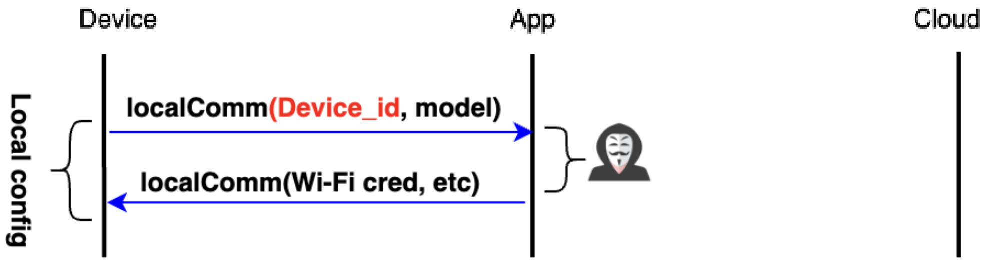
- Research gap in smart home IoT security
  - Many of previous researches' measure IoT security when it's fully deployed.

- Main Research Question
  - **If we were to purchase today a handful of IoT devices for different smart home tasks, what fraction of those involve some degree of leaking sensitive information during the setup?**

https://cheapsslsecurity.com/blog/iot-security-understanding-pki-role-in-securing-internet-of-things/

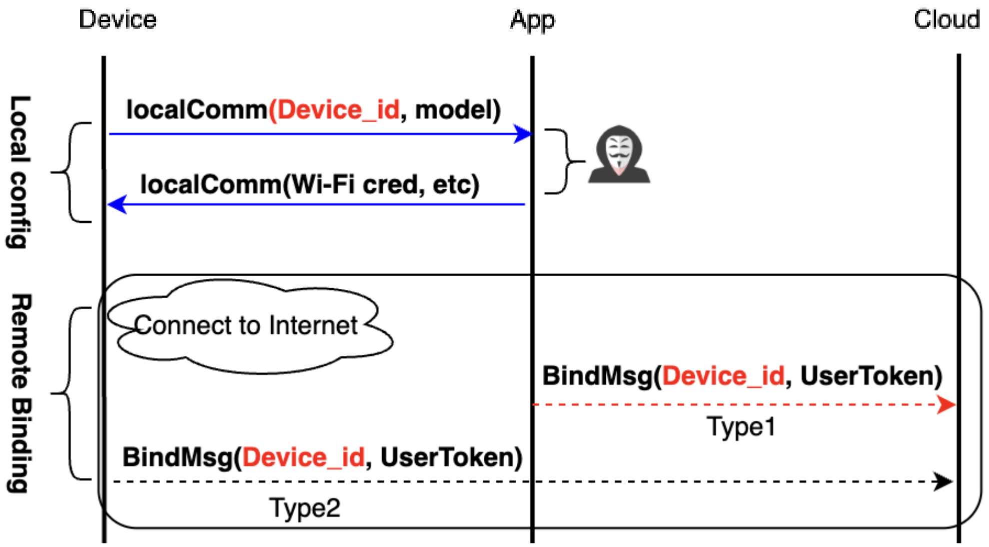
A Typical IoT Platform State Transitions

- Local Config State
  - A **local communication** (LC) is established between the app and device to share information



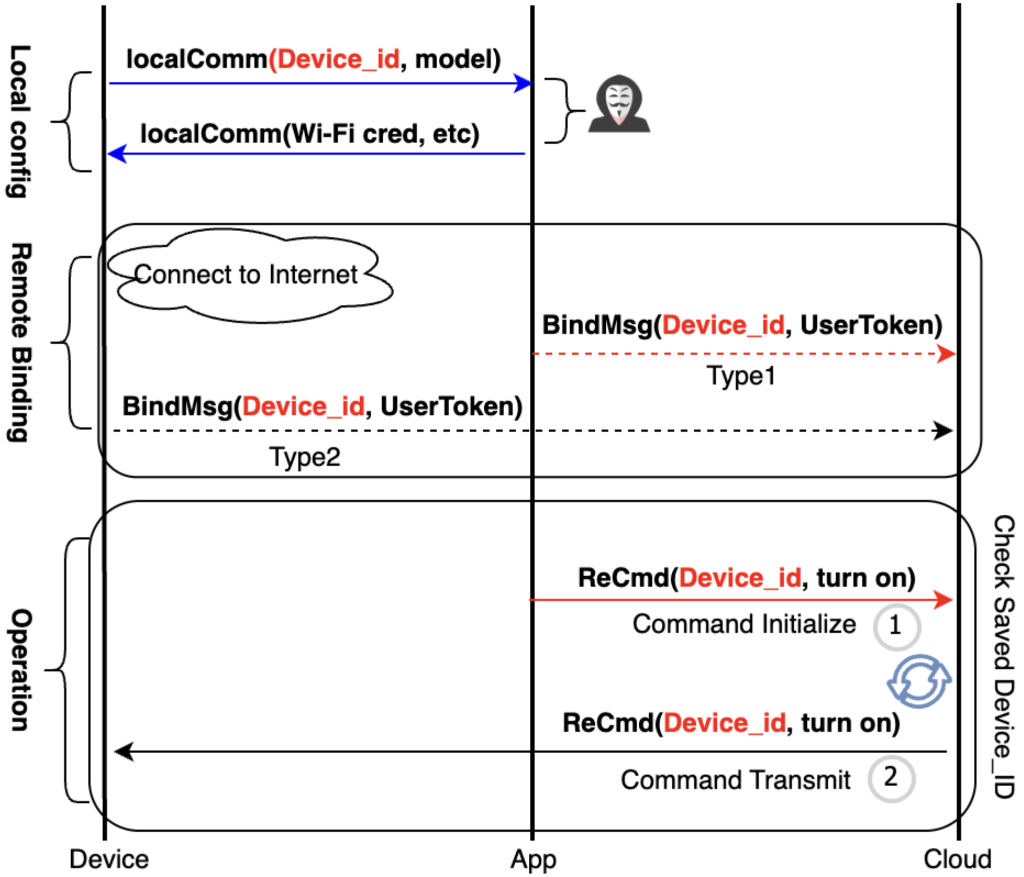
A Typical IoT Platform State Transitions

- Local Config State
  - A **local communication** (LC) is established between the app and device to share information
- Remote Binding State
  - Binding request to register device instances to specific users' account.


Device
App
Cloud
Local config
Remote Binding
localComm(Device_id, model)
localComm(Wi-Fi cred, etc)
Connect to Internet
BindMsg(Device_id, UserToken)
Type1
BindMsg(Device_id, UserToken)
Type2

A Typical IoT Platform State Transitions

- Local Config State
  - A **local communication** (LC) is established between the app and device to share information
- Remote Binding State
  - Binding request to register device instances to specific users' account.
- Operation State
  - State when devices are fully **setup**, can be operated by **remote commands.**


Local config
localComm(Device_id, model)
localComm(Wi-Fi cred, etc)
Remote Binding
Connect to Internet
BindMsg(Device_id, UserToken)
Type1
BindMsg(Device_id, UserToken)
Type2
Operation
ReCmd(Device_id, turn on)
Command Initialize
1
ReCmd(Device_id, turn on)
Command Transmit
2
Check Saved Device_ID
Device
App
Cloud

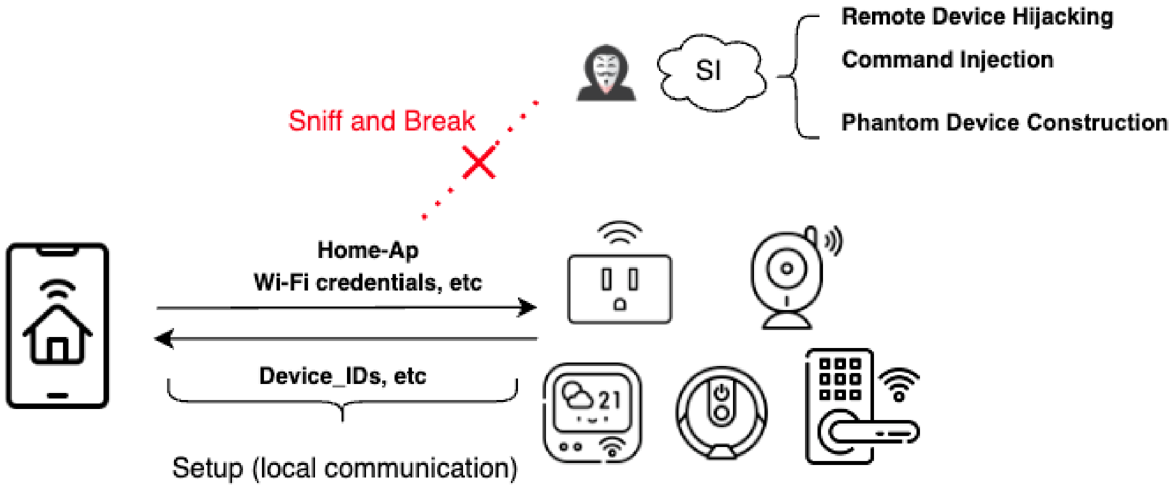
Sensitive Information

Sensitive Information (SI)	Definitions	Why Sensitive
IDs	Device_ID is the unique device identifier that the app, cloud, and device agree upon using a specific device instance authentication.	The steal of the Device_ID might lead to potential attacks such as remote device hijacking, remote command injection and phantom device construction [2].
Credentials	Credentials are any information required to access a system, service or resources.	Credential leakage can have different consequences, such as an attacker's unhindered access to, network (home-AP credentials), data, and app accounts.

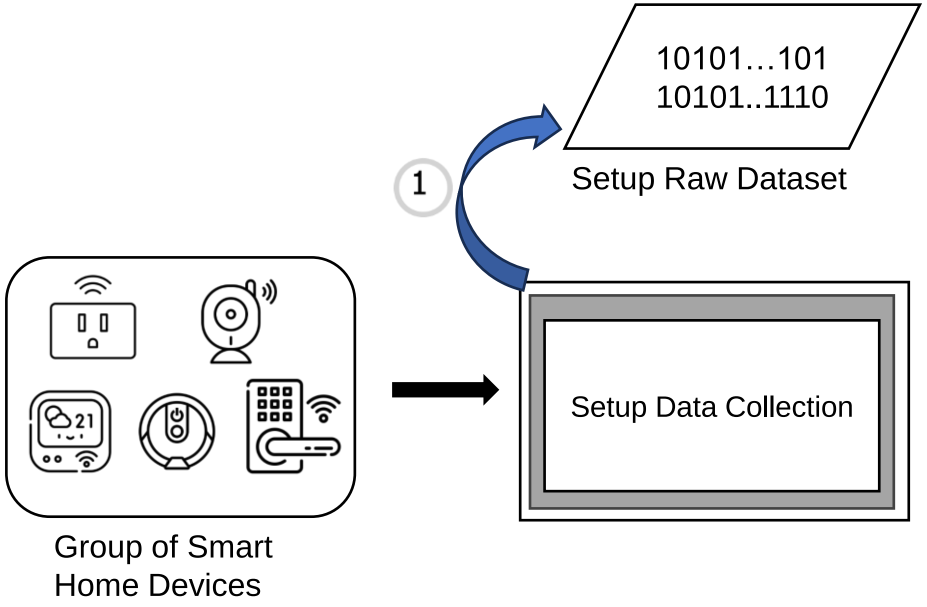
Threat Model

- Attackers Type:
  - An opportunistic attacker passively and continuously sniffs over-the-air (OTA) Wi-Fi and BLE traffic.

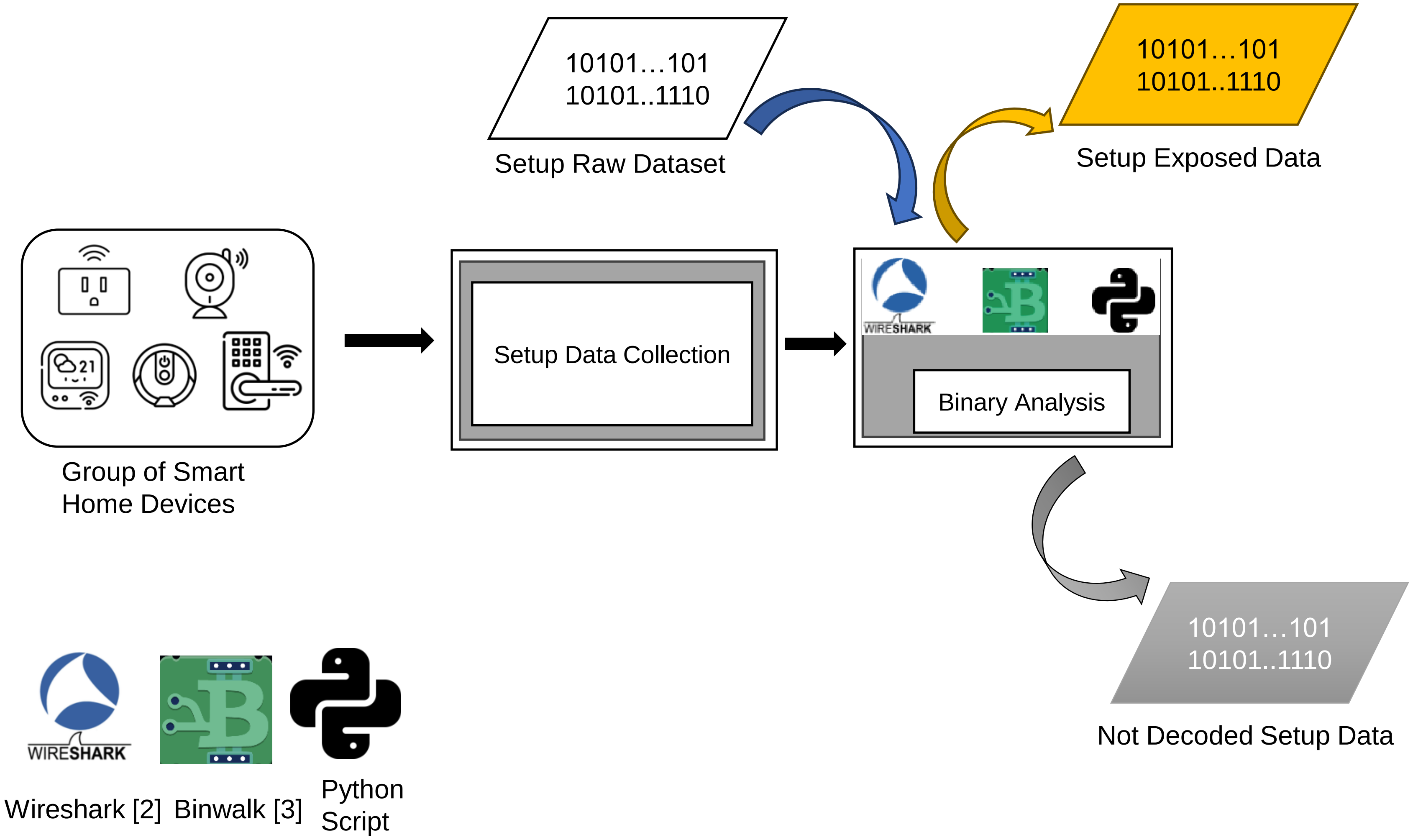
- Attackers' Aim:
  - Harvest sensitive information such as IDs, users' home-AP Wi-Fi credentials, etc. Then once the setup finishes, use the harvested sensitive information holding attacks such as remote device hijacking, command injection, phantom device construction, etc.



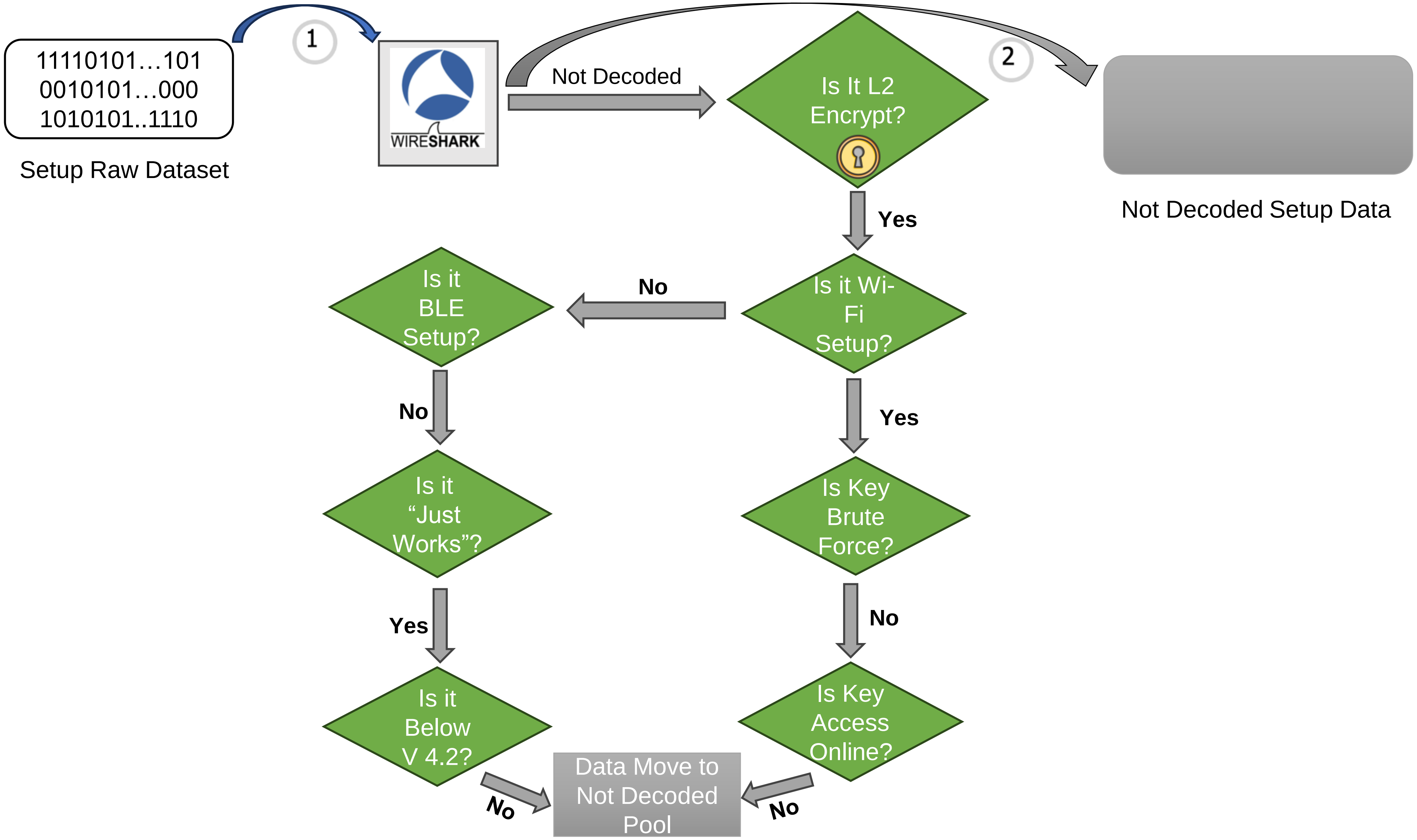
Methodology Overview (Data Collection)



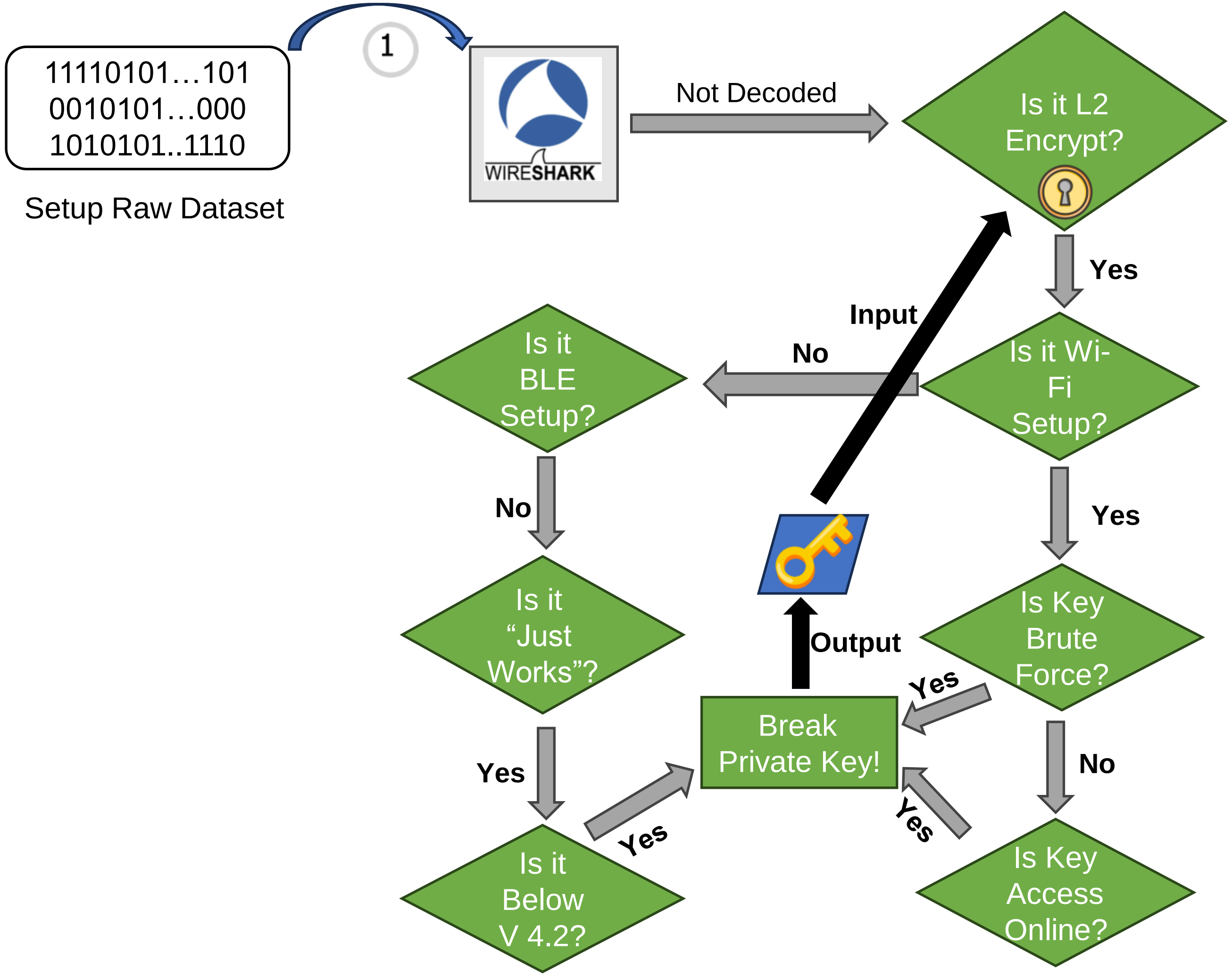
Methodology Steps (Binary Analysis)



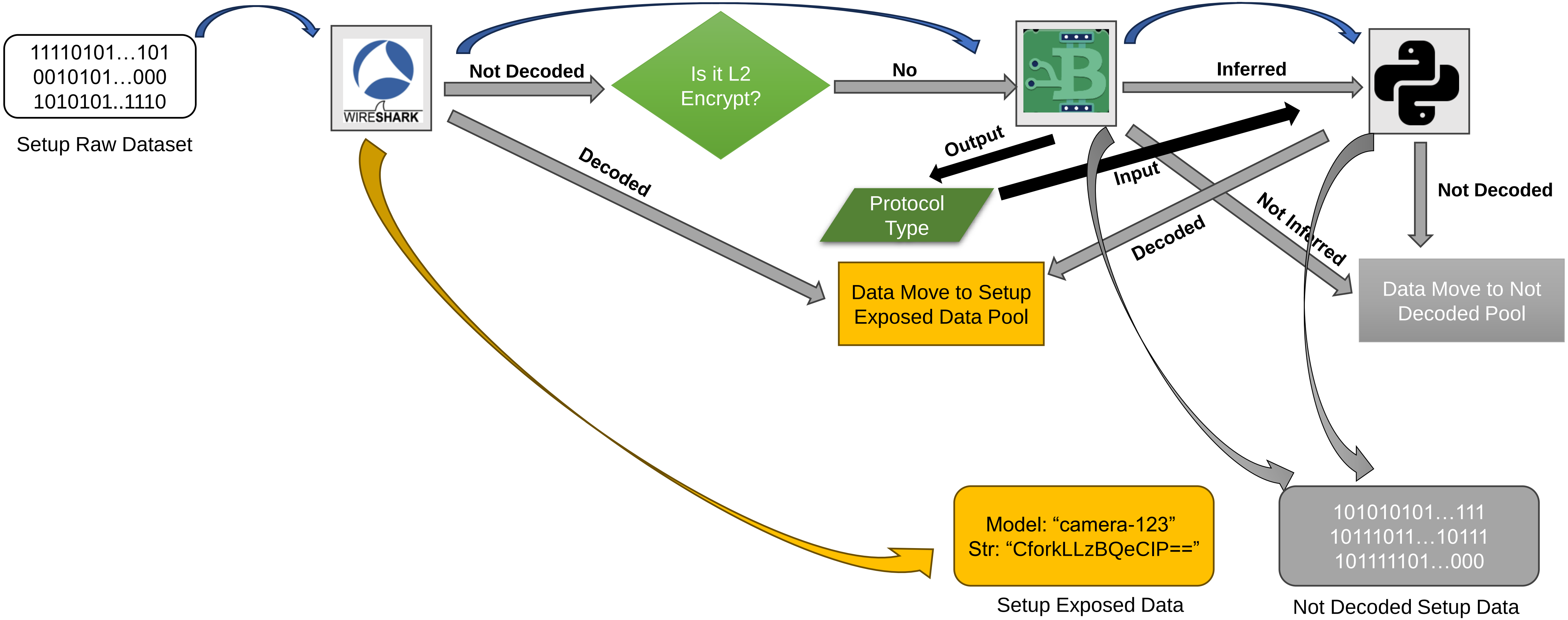
Binary Analysis Example (L2 Encryption, Key Not Derivable)



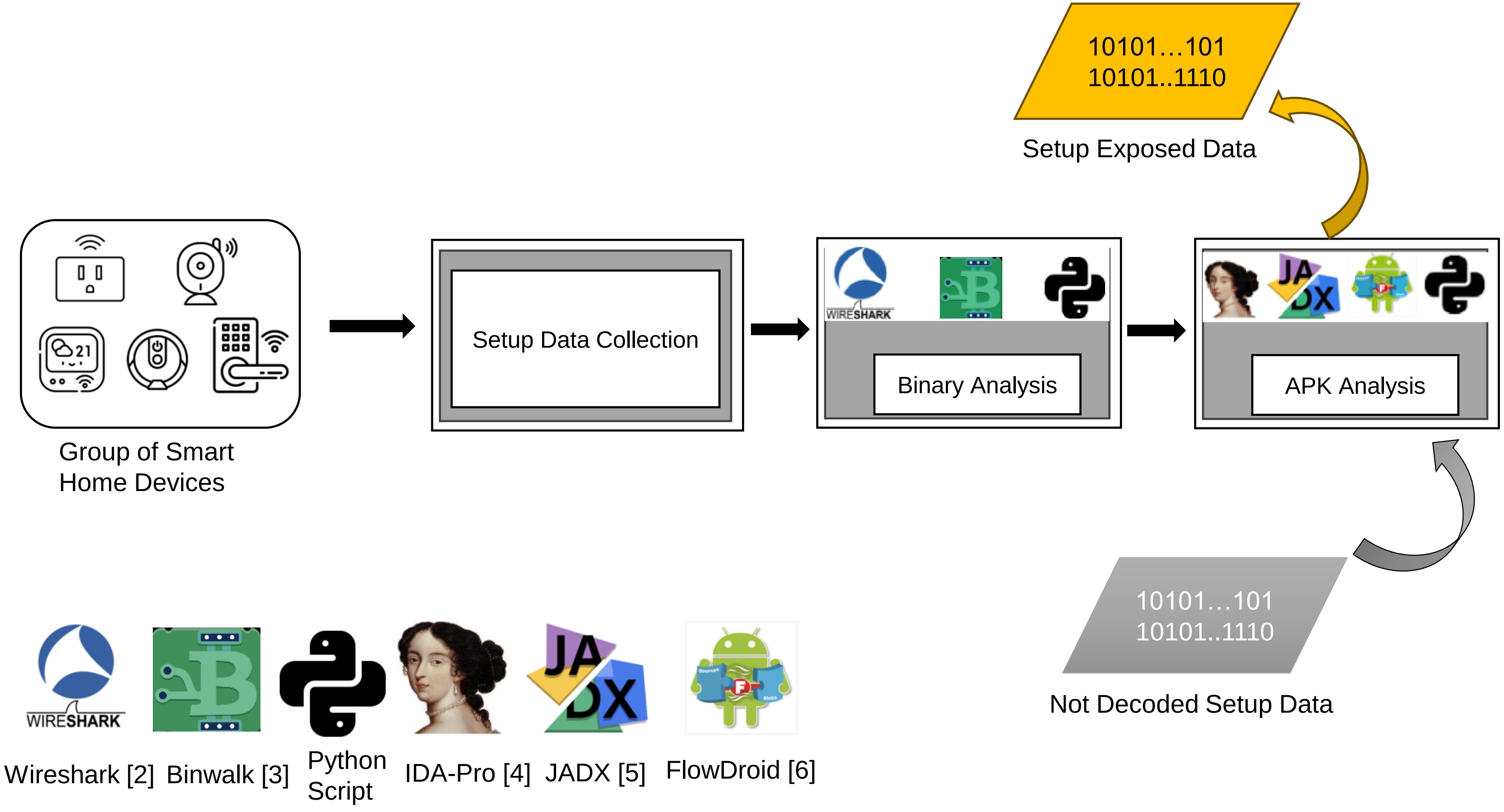
Binary Analysis Example (L2 Encryption, Key Derivable, Break L2 Encryption)



Binary Analysis Example (L2 Encryption Break, Back To No L2 Encryption State)



Methodology Steps (APK Analysis)



10101…101
10101..1110

Setup Exposed Data

Group of Smart Home Devices

Setup Data Collection

Binary Analysis

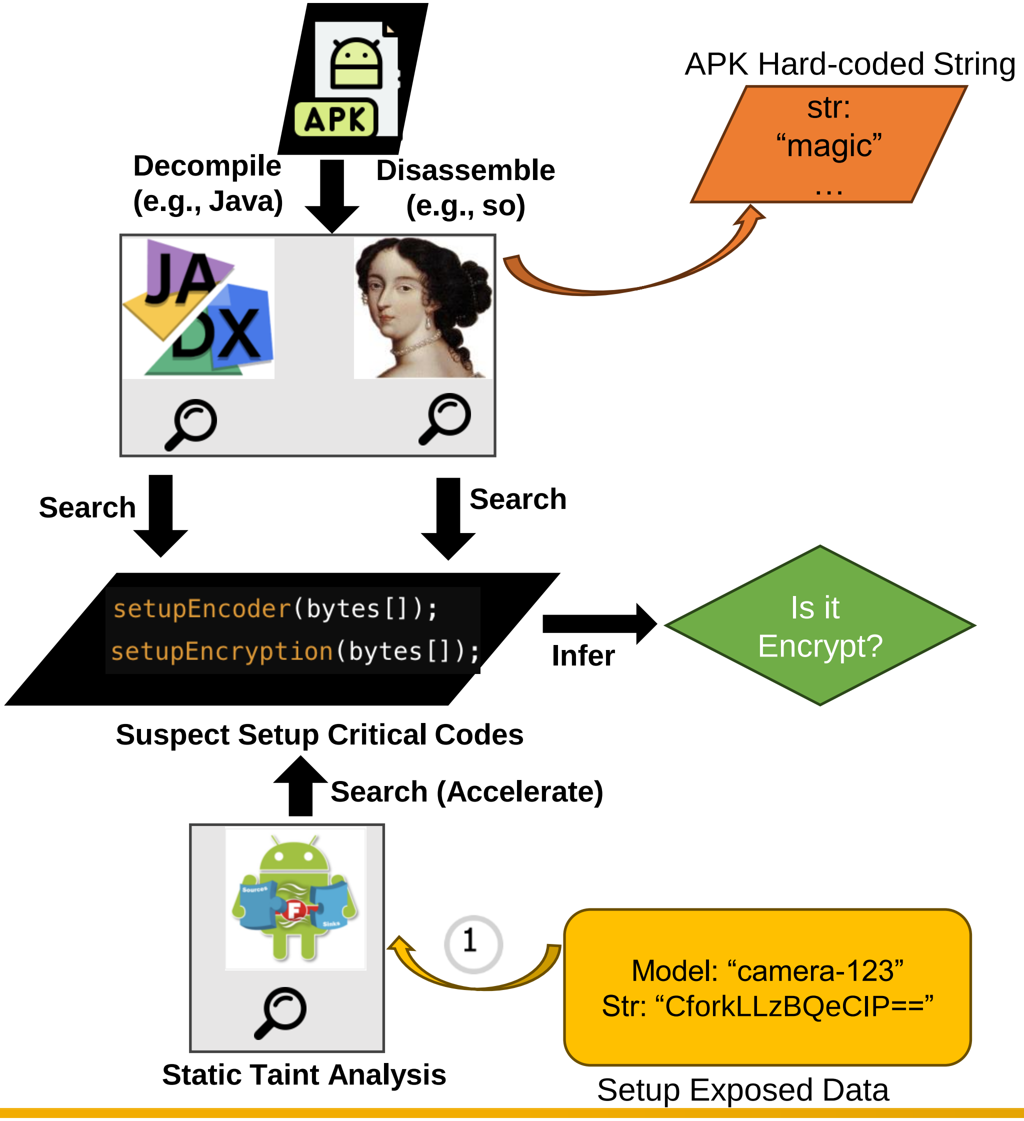
APK Analysis

10101…101
10101..1110

Not Decoded Setup Data

Wireshark [2] Binwalk [3] Python Script IDA-Pro [4] JADX [5] FlowDroid [6]

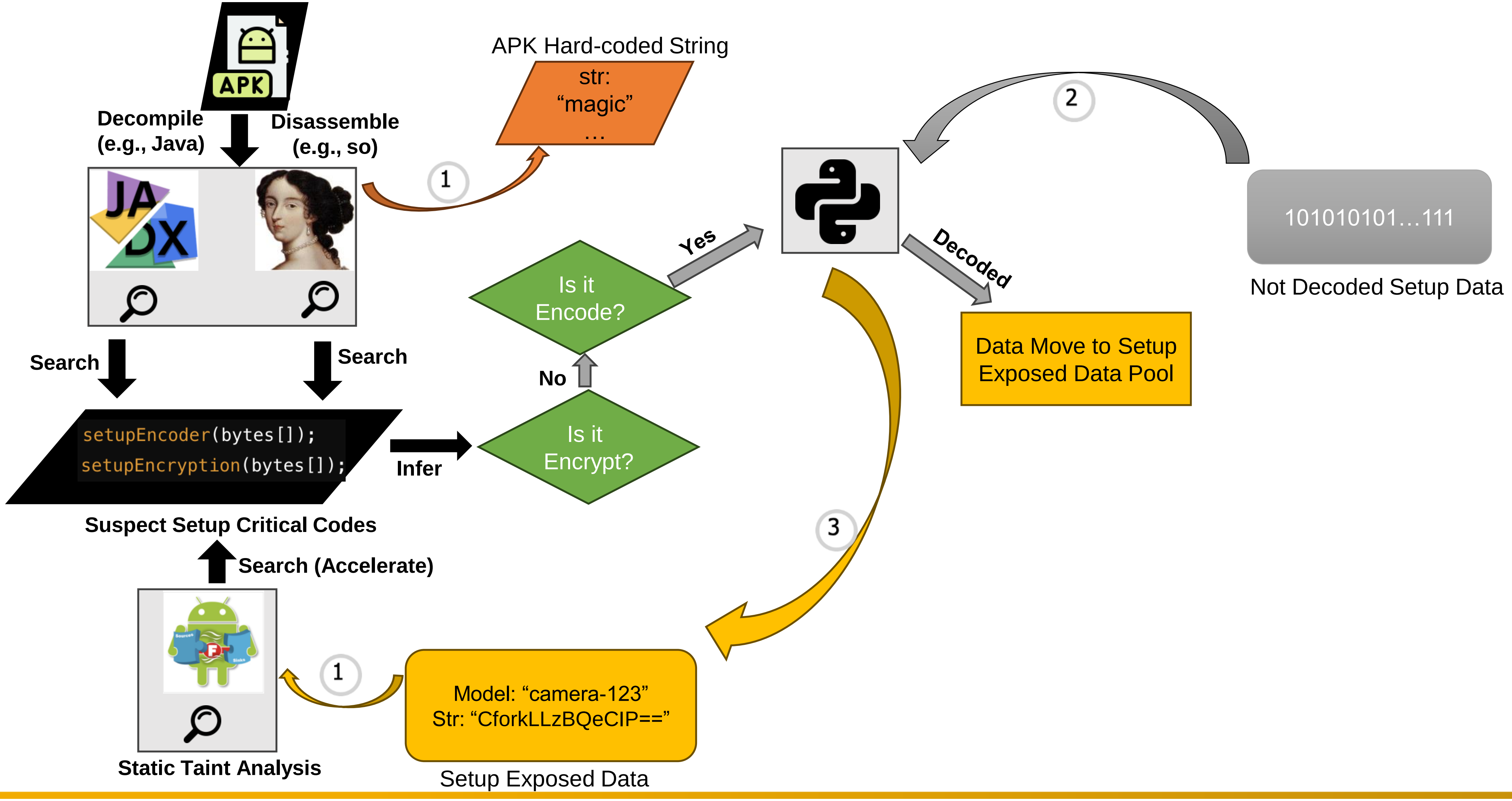
APK Analysis Initial Steps (Search for critical codes)



```
setupEncoder(bytes[]);
setupEncryption(bytes[]);
```

Suspect Setup Critical Codes

APK Analysis Example (No Encryption, Custom Encoding, Python Decoded)



APK Hard-coded String

str:
“magic”
…

Decompile
(e.g., Java)

Disassemble
(e.g., so)

1

2

101010101…111

Not Decoded Setup Data

Yes

Decoded

Is it
Encode?

No

Data Move to Setup
Exposed Data Pool

Search

Search

```
setupEncoder(bytes[]);
setupEncryption(bytes[]);
```

Infer

Is it
Encrypt?

Suspect Setup Critical Codes

3

Search (Accelerate)

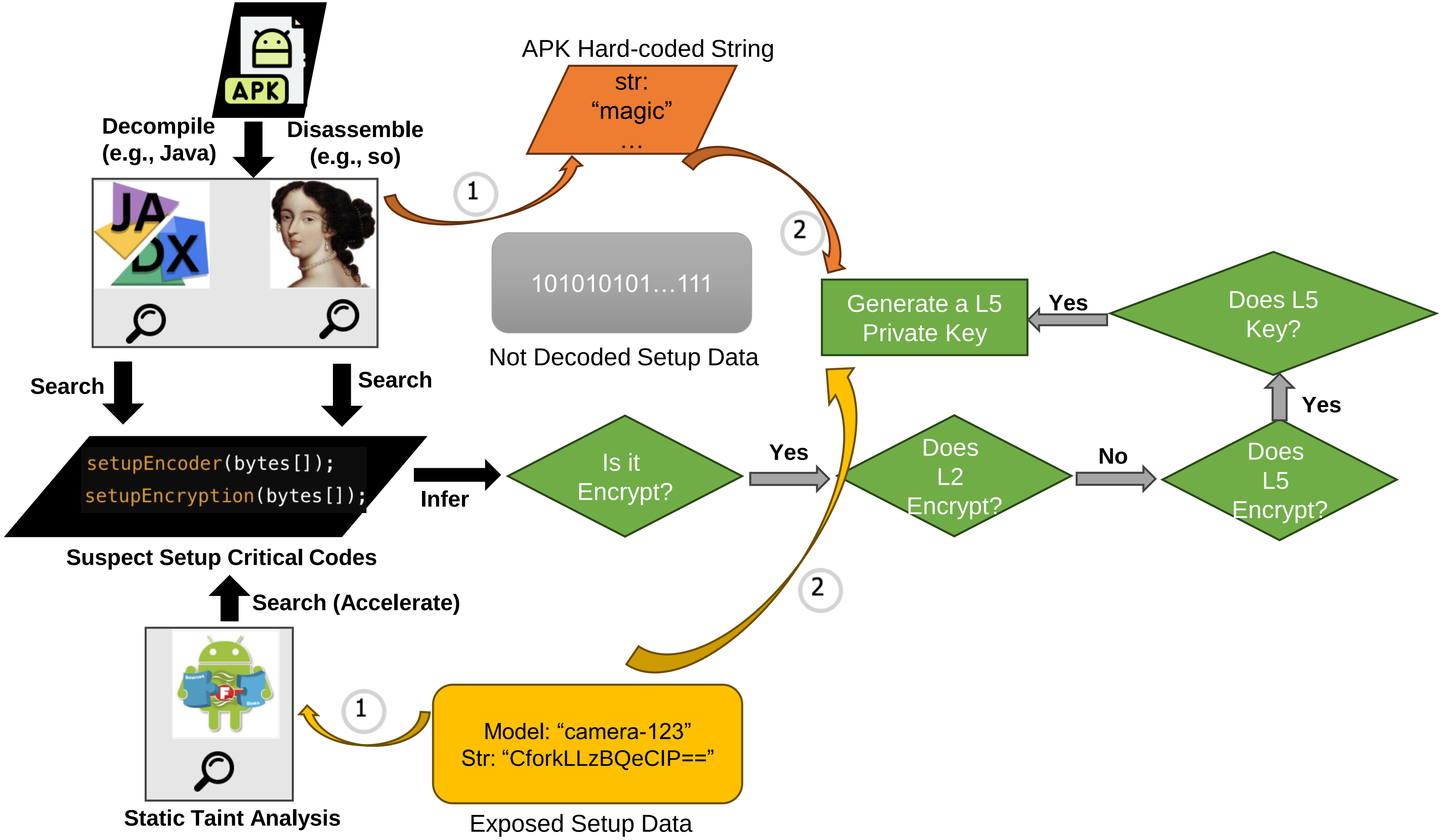
1

Model: “camera-123”
Str: “CforkLLzBQeCIP==”

Static Taint Analysis

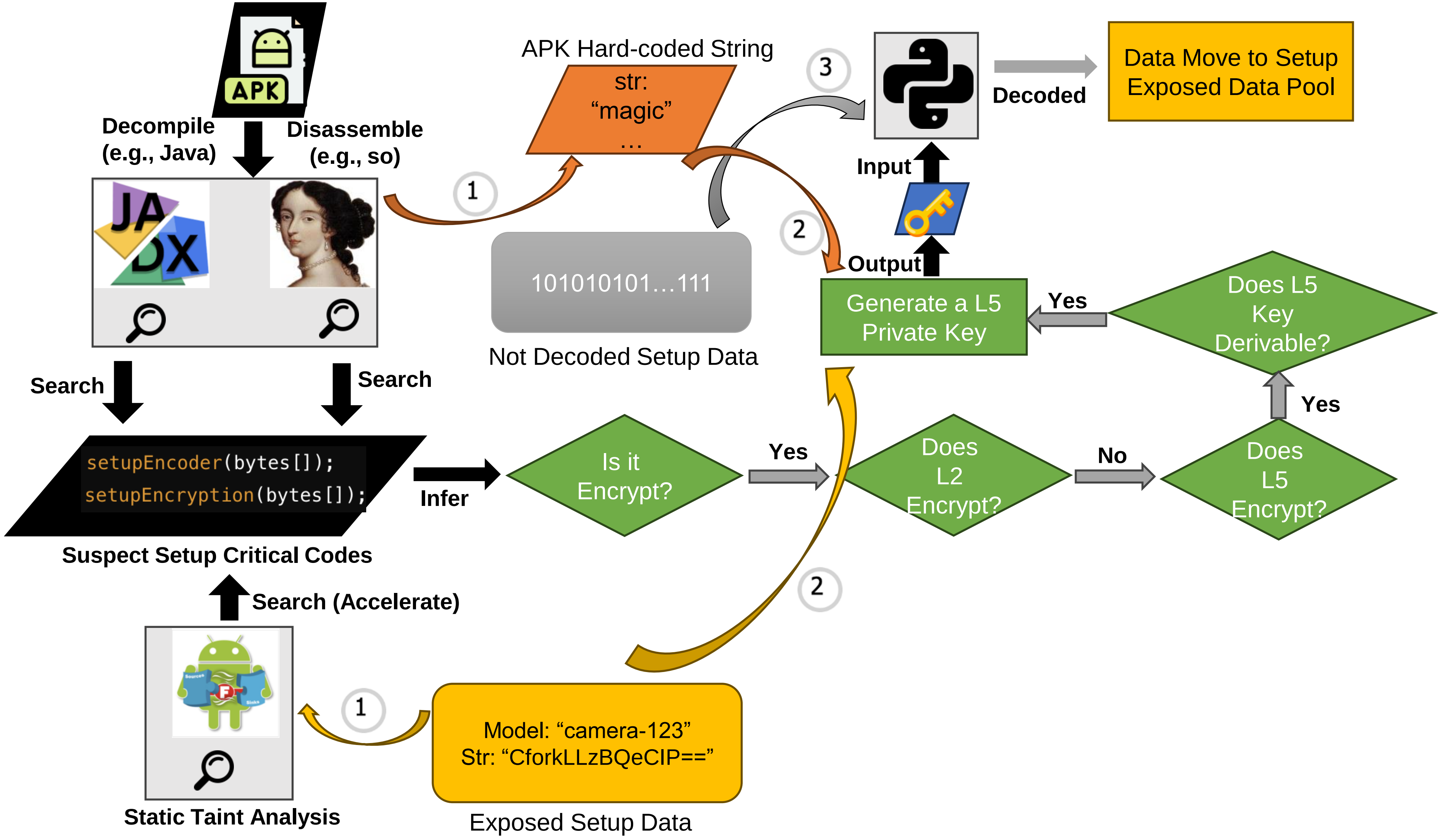
Setup Exposed Data

APK Analysis Example (L5 Encryption, Weak Key, Key Generation)



```
setupEncoder(bytes[]);
setupEncryption(bytes[]);
```

APK Analysis Example (L5 Encryption, Weak Key, Encryption Break, Data to Non-Beaconing Pool)



Decompile (e.g., Java)

Disassemble (e.g., so)

Search

Search

```
setupEncoder(bytes[]);
setupEncryption(bytes[]);
```

Suspect Setup Critical Codes

Search (Accelerate)

Static Taint Analysis

1

Model: “camera-123”
Str: “CforkLLzBQeCIP==”

Exposed Setup Data

2

Infer

Is it Encrypt?

Yes

Does L2 Encrypt?

No

Does L5 Encrypt?

Yes

Does L5 Key Derivable?

Yes

Generate a L5 Private Key

Output

Input

Decoded

Data Move to Setup Exposed Data Pool

1

APK Hard-coded String

str:
“magic”
…

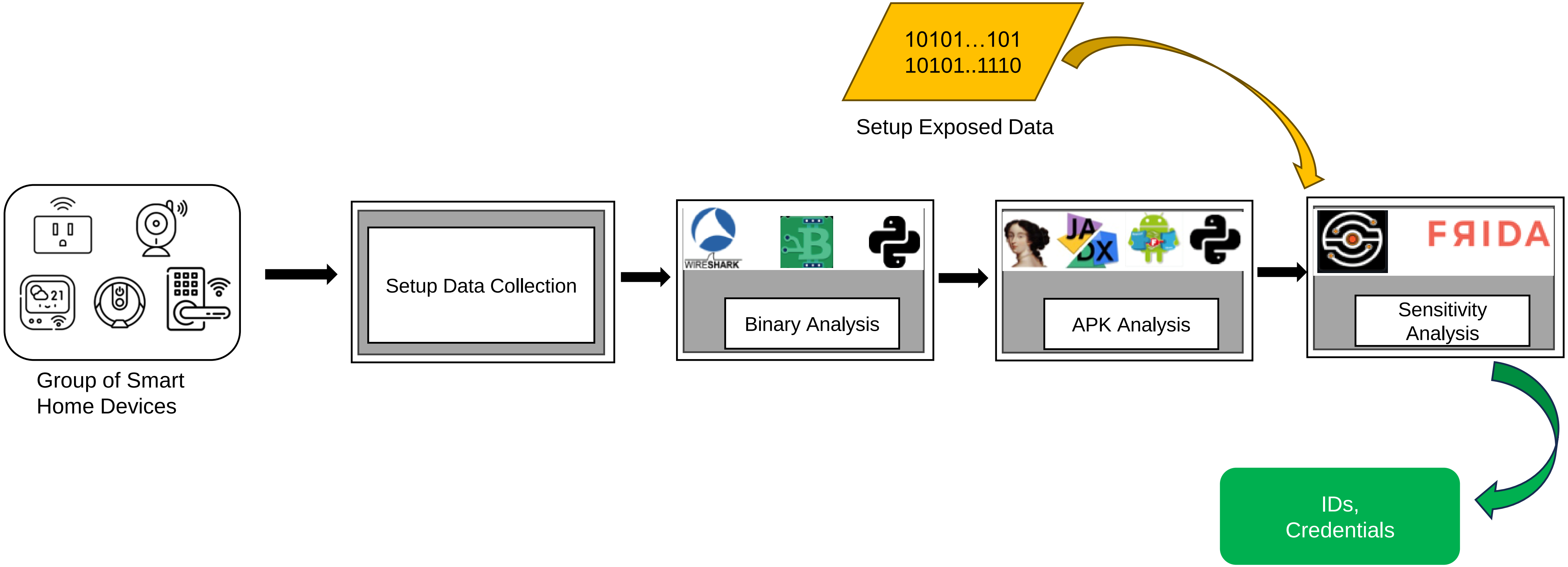
2

101010101…111

Not Decoded Setup Data

3

Methodology Overview



Setup Exposed Data

Group of Smart Home Devices

Setup Data Collection

Binary Analysis

APK Analysis

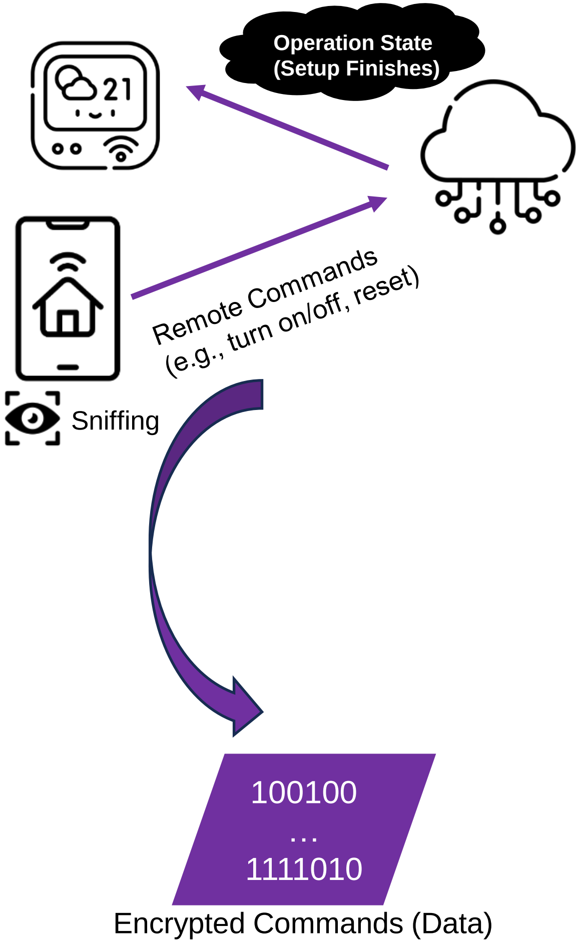
Sensitivity Analysis

IDs, Credentials

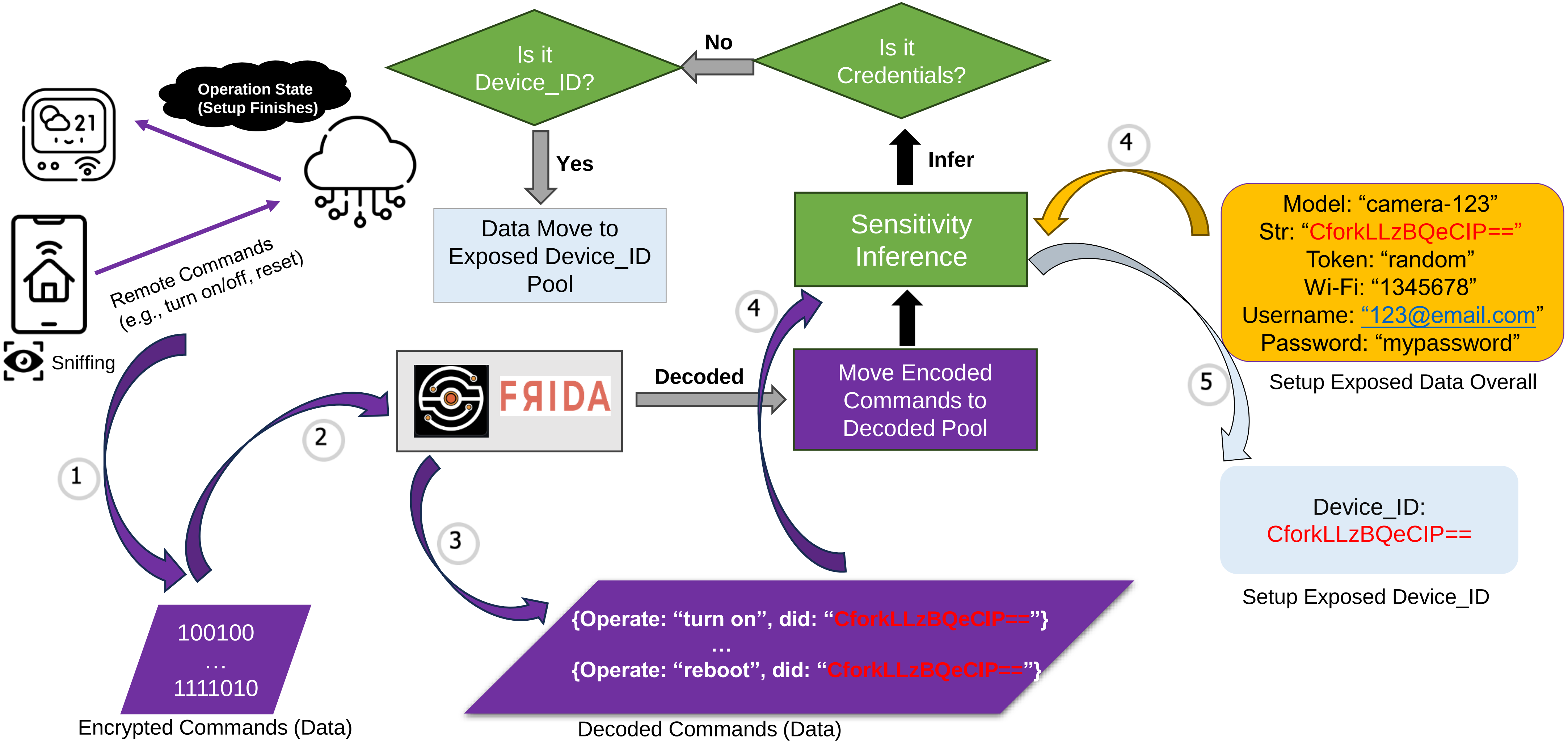
Setup Exposed Sensitive Information (SI)

Wireshark [2] Binwalk [3] Python Script IDA-Pro [4] JADX [5] FlowDroid [6] mitmproxy [7] Frida [8]

Sensitivity Analysis Initial Steps (Remote Commands Generation)



Sensitivity Analysis Example (Remote Commands Decoded, Device_ID Exposing Verified)



Operation State (Setup Finishes)

Remote Commands (e.g., turn on/off, reset)

Sniffing

1

100100
…
1111010

Encrypted Commands (Data)

2

FRIDA

3

{Operate: “turn on”, did: “CforkLLzBQeCIP==”}
…
{Operate: “reboot”, did: “CforkLLzBQeCIP==”}

Decoded Commands (Data)

Decoded

Move Encoded Commands to Decoded Pool

4

Sensitivity Inference

Infer

Is it Credentials?

No

Is it Device_ID?

Yes

Data Move to Exposed Device_ID Pool

4

Model: “camera-123”
Str: “CforkLLzBQeCIP==”
Token: “random”
Wi-Fi: “1345678”
Username: “123@email.com”
Password: “mypassword”

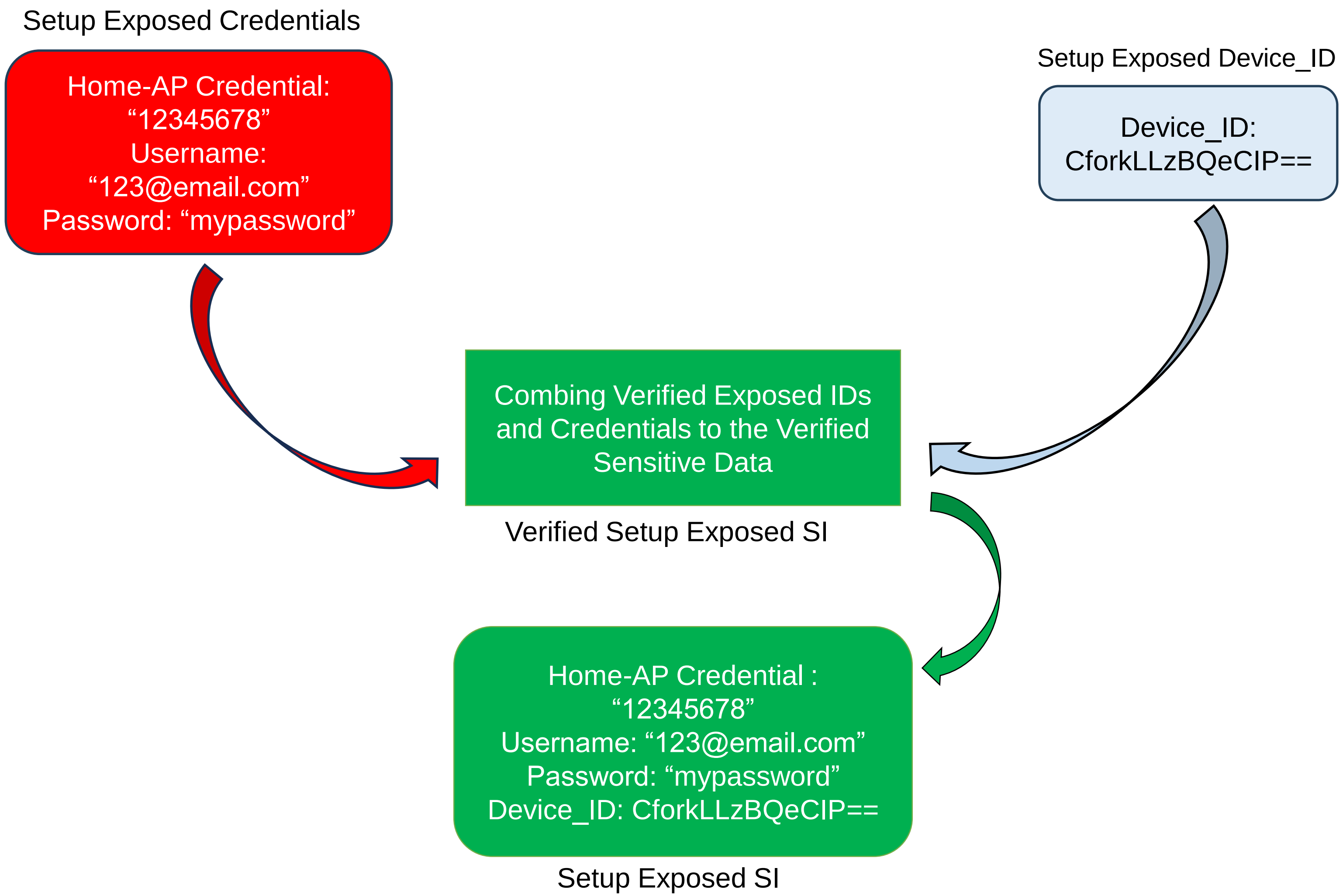
Setup Exposed Data Overall

5

Device_ID:
CforkLLzBQeCIP==

Setup Exposed Device_ID

Sensitivity Analysis Final Step (Combining Exposed ID and Exposed Credentials to Exposed SI)



Setup Exposed Credentials

Home-AP Credential: “12345678”
Username: “123@email.com”
Password: “mypassword”

Setup Exposed Device_ID

Device_ID: CforkLLzBQeCIP==

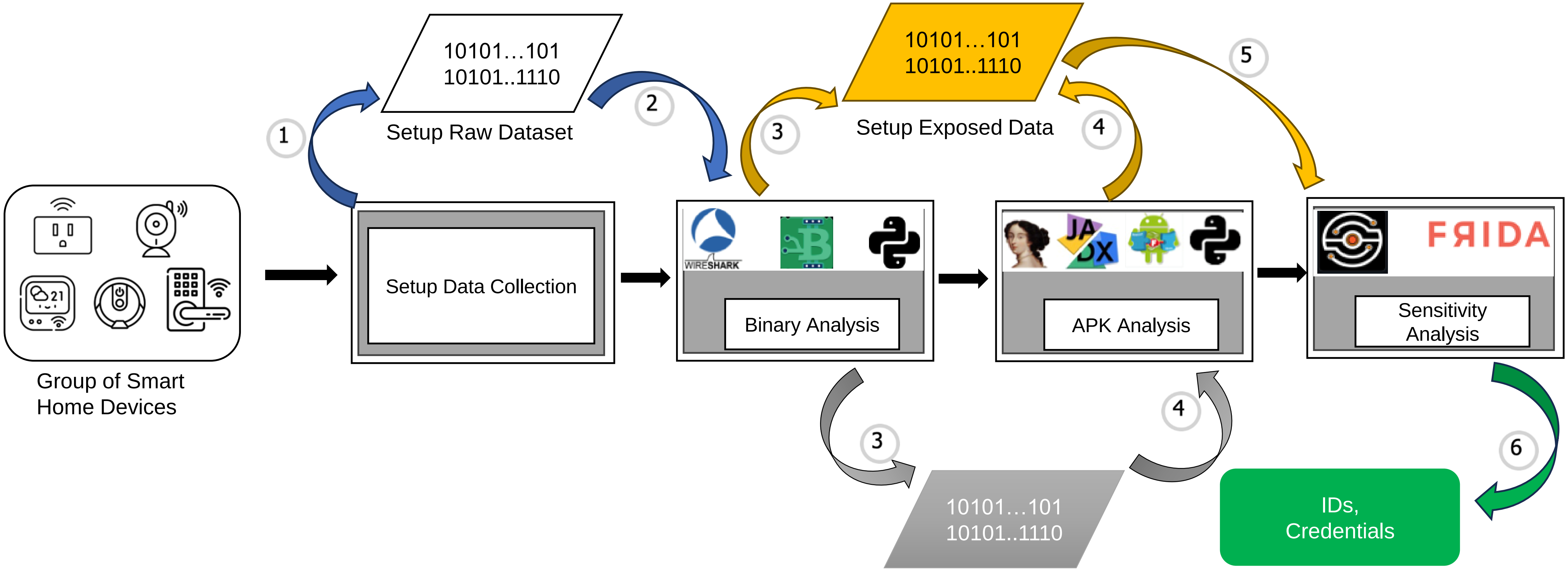
Combing Verified Exposed IDs and Credentials to the Verified Sensitive Data

Verified Setup Exposed SI

Home-AP Credential : “12345678”
Username: “123@email.com”
Password: “mypassword”
Device_ID: CforkLLzBQeCIP==

Setup Exposed SI

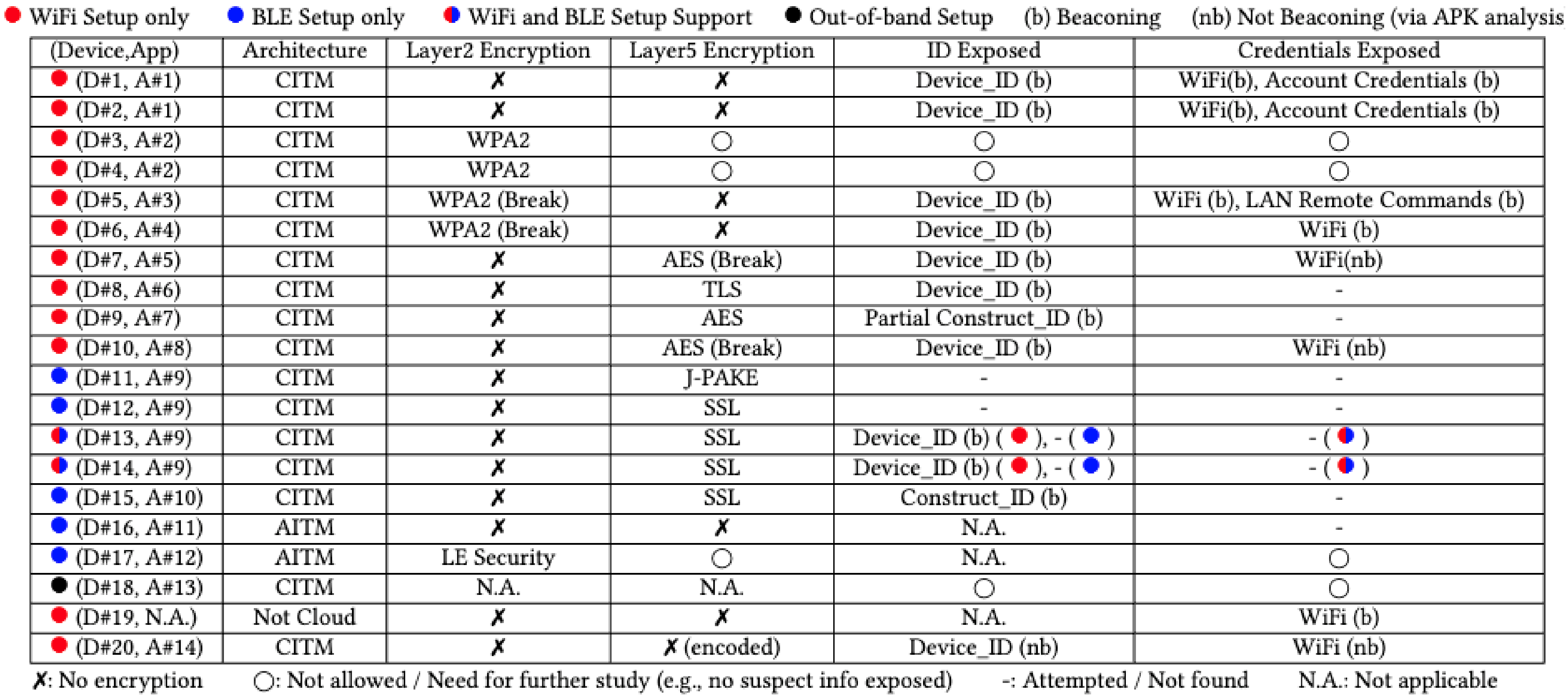
Methodology Overview (Sensitivity Analysis)



Wireshark [2] Binwalk [3] Python Script IDA-Pro [4] JADX [5] FlowDroid [6] mitmproxy [7] Frida [8]

Eval: Results Overview

- After going through the methodology from a sample of 20 devices
  - 7 Safe Devices
  - 11 Devices exposed IDs (Construct and Device)
  - 8 Devices exposed credentials (including home-AP credentials and app login credentials)



🔴 WiFi Setup only 🔵 BLE Setup only ◐ WiFi and BLE Setup Support ● Out-of-band Setup (b) Beaconing (nb) Not Beaconing (via APK analysis

(Device,App)	Architecture	Layer2 Encryption	Layer5 Encryption	ID Exposed	Credentials Exposed
🔴 (D#1, A#1)	CITM	✗	✗	Device_ID (b)	WiFi(b), Account Credentials (b)
🔴 (D#2, A#1)	CITM	✗	✗	Device_ID (b)	WiFi(b), Account Credentials (b)
🔴 (D#3, A#2)	CITM	WPA2	○	○	○
🔴 (D#4, A#2)	CITM	WPA2	○	○	○
🔴 (D#5, A#3)	CITM	WPA2 (Break)	✗	Device_ID (b)	WiFi (b), LAN Remote Commands (b)
🔴 (D#6, A#4)	CITM	WPA2 (Break)	✗	Device_ID (b)	WiFi (b)
🔴 (D#7, A#5)	CITM	✗	AES (Break)	Device_ID (b)	WiFi(nb)
🔴 (D#8, A#6)	CITM	✗	TLS	Device_ID (b)	-
🔴 (D#9, A#7)	CITM	✗	AES	Partial Construct_ID (b)	-
🔴 (D#10, A#8)	CITM	✗	AES (Break)	Device_ID (b)	WiFi (nb)
🔵 (D#11, A#9)	CITM	✗	J-PAKE	-	-
🔵 (D#12, A#9)	CITM	✗	SSL	-	-
◐ (D#13, A#9)	CITM	✗	SSL	Device_ID (b) (🔴), - (🔵)	- (◐)
◐ (D#14, A#9)	CITM	✗	SSL	Device_ID (b) (🔴), - (🔵)	- (◐)
🔵 (D#15, A#10)	CITM	✗	SSL	Construct_ID (b)	-
🔵 (D#16, A#11)	AITM	✗	✗	N.A.	-
🔵 (D#17, A#12)	AITM	LE Security	○	N.A.	○
● (D#18, A#13)	CITM	N.A.	N.A.	○	○
🔴 (D#19, N.A.)	Not Cloud	✗	✗	N.A.	WiFi (b)
🔴 (D#20, A#14)	CITM	✗	✗ (encoded)	Device_ID (nb)	WiFi (nb)

✗: No encryption ○: Not allowed / Need for further study (e.g., no suspect info exposed) -: Attempted / Not found N.A.: Not applicable

Note: We would like to correct ID Exposed for D#6 of the table, we marked as the Device_ID exposed, which is a typo, D#6 should be marked as "Attempted/Not Found" instead.

Eval: Safe Devices (Key Results)

- We could not extract SI from 7 devices.
- We find encryption effective when applied correctly.

(Device, App)	L2/L5 Encryption	Safe Reason
(D#3, A#2) (D#4, A#2) (D16, A#11)	WPA2 WPA2 LE Secure	Strong L2 encryption.
(D#11, A#9) (D#12, A#9)	SSL SSL	Strong L5 encryption and existence of not confirmed information.
(D#16, A#11)	None	Full BLE device does not require home-AP credentials to access the internet, also no defined ID exposure verified.
(D#18, A#13)	N.A./N.A.	No explicit wireless connection on LC.

Eval: ID Exposure (Key Results)

- We could extract IDs (Device or Construct) from in total of 11 devices.
- We find even though the encryption (partial) might take place, many of these devices still expose IDs as plaintext.

(Device, App)	L2/L5 Encryption	ID Exposed Reason
(D#1, A#1) (D#2, A#1)	None	Full Plaintext during LC
(D#5, A#3)	WPA2 (Break)	L2 encryption exists but with a weak WPA passphrase.
(D#8, A#6) (D#9, A#7)	TLS AES (Partial)	Suffix of SSID for the device-AP.
(D#10, A#8) (D#13, A#9) (D#14, A#9) (D#15, A#10) (D#7, A#5)	AES (Partial, Break) SSL (Partial) SSL (Partial) SSL (Partial) AES (Partial, Break)	L5 encryption exists, but ID is transmitted in plaintext
(D#20, A#14)	None	No encryption exists, ID is encoded by a custom protocol (smartCfg).

Eval: Credentials Exposure

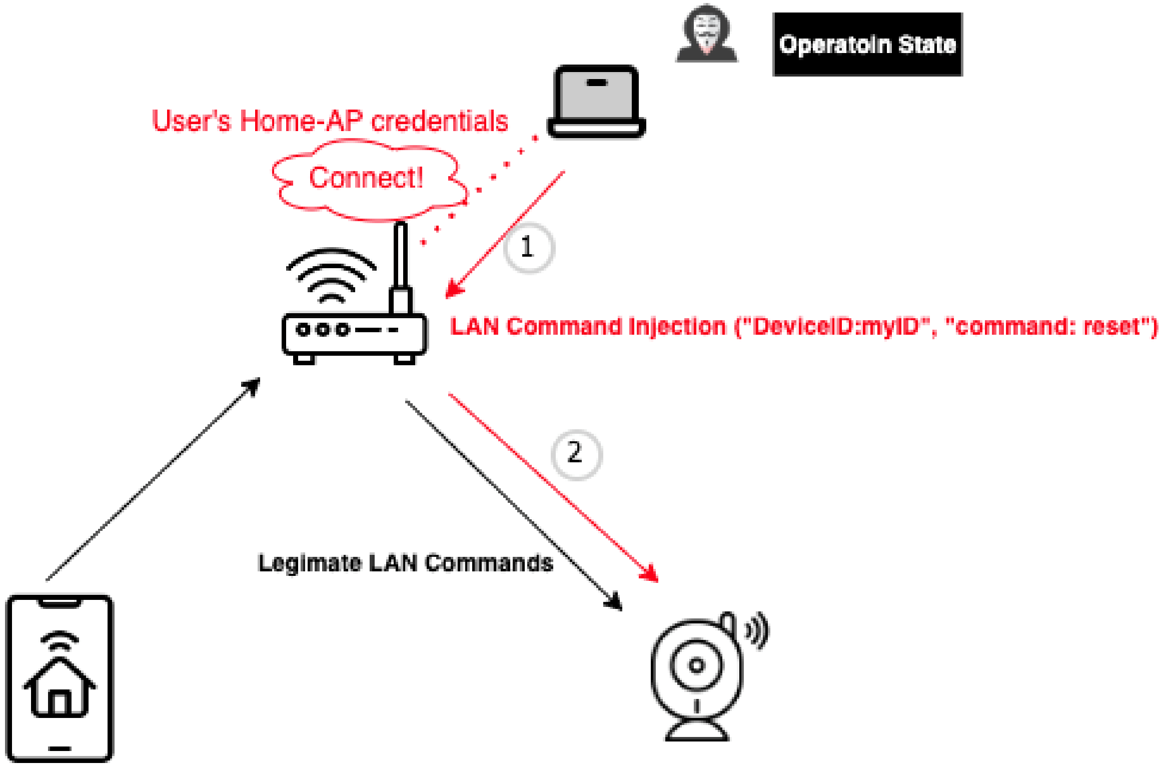
- We could extract credentials from 8 devices.
- Although there are devices that deploy encryption to protect credentials, they do not well guard the private key which we could build the same key to break the setup communication via APK analysis.

(Device, App)	L2/L5 Encryption	Credentials Exposure	ID Exposed Reason
(D#1, A#1) (D#2, A#2)	None	Wi-Fi, app login credentials;	Full Plaintext
(D#5, A#3) (D#6, A#4)	WPA2 (Break) WPA2 (Break)	Wi-Fi, LAN commands Wi-Fi, LAN commands	L2 encryption exists, but with weak WPA passphrase.
(D#7, A#5) (D#10, A#8)	AES (Partial, Break) AES (Partial, Break)	Wi-Fi Wi-Fi	Use L5 AES-based encryption to protect the home-AP Wi-Fi credentials; however, the encryption key is derivable.
(D#19, N.A.)	None	Wi-Fi	Full Plaintext (No cloud-based device, via the captive portal by holding HTTP, sharing information during LC).
(D#20, A#14)	None	Wi-Fi	No encryption exists, credential is encoded by a custom protocol (smartCfg), and decoded via APK analysis.

Eval: Example of Attacks, confirmed

- Chain of Attacks
  - Sensitive information exposing -> LAN command injection -> reset the device -> device hijacking by setting up the reset devices to the attacker' account.

  - Requirement: Device_ID, home-AP credentials, breakable LAN command


Operatoin State
User's Home-AP credentials
Connect!
1
LAN Command Injection ("DeviceID:myID", "command: reset")
2
Legimate LAN Commands

Conclusions

- Examined setup phase of 20 popular smart home IoT devices for potential information leakage
- Two-thirds of the devices indeed exposed sensitive information
- We also successfully executed chain of attacks by some of the compromised IoT device
- This study will assist developers, vendors, and researchers in ensuring IoT setup security

- Prove the ID verification process end-to-end considering device firmware analysis and end-to-end remote attacks verification.
- Extending the current works to test unseen devices in an automated manner

References

[1] Liu, H., Li, J., & Gu, D. (2020). Understanding the security of app-in-the-middle IoT. *Computers & Security*, *97*, 102000.

[2] https://www.wireshark.org/

[3] https://www.kali.org/tools/binwalk/

[4] https://hex-rays.com/ida-pro

[5] https://github.com/skylot/jadx

[6] https://github.com/secure-software-engineering/FlowDroid

[7] https://mitmproxy.org/

[8] https://github.com/frida

Appendix Slides

Local Communication (LC) During Device Setup

- We identify a new possible sensitive information leakage channel here which is the local communication during the setup.
- Local Communication Types:
  - Wi-Fi (Access point (AP) mode, EZ mode).
  - BLE (Standard BLE pairing (central-peripheral mode)).

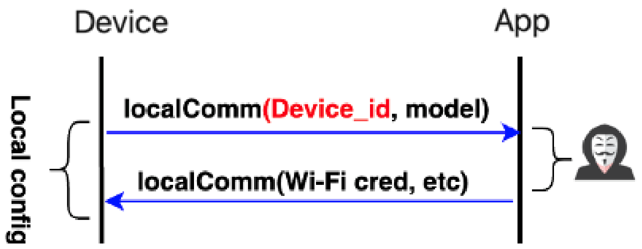
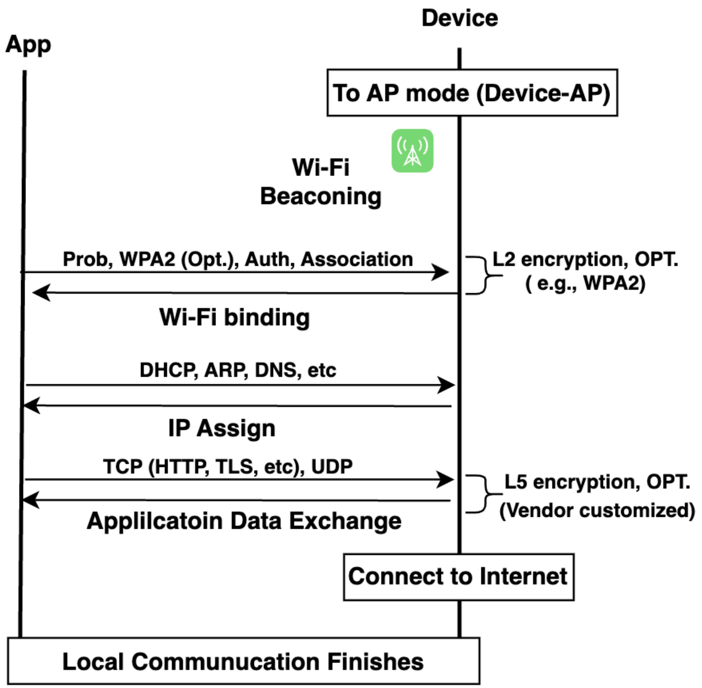
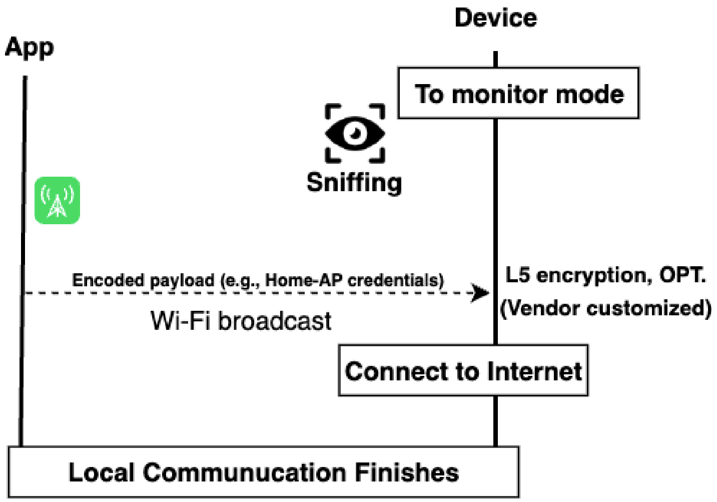


Fig., Sample location communication (LC) exchanged application layer data.



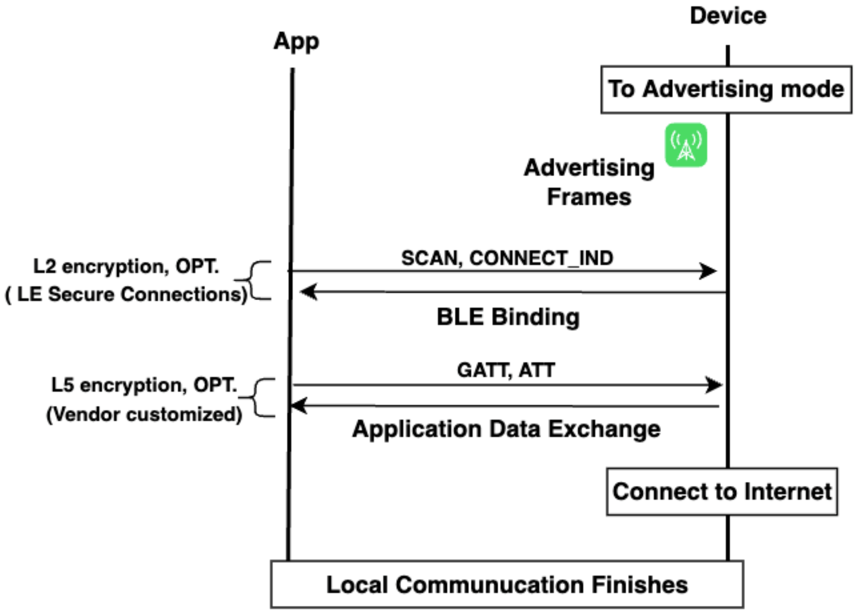
Device
App
To AP mode (Device-AP)
Wi-Fi Beaconing
Prob, WPA2 (Opt.), Auth, Association
L2 encryption, OPT. (e.g., WPA2)
Wi-Fi binding
DHCP, ARP, DNS, etc
IP Assign
TCP (HTTP, TLS, etc), UDP
L5 encryption, OPT. (Vendor customized)
Applilcatoin Data Exchange
Connect to Internet
Local Communucation Finishes

Fig., Sample Location Communication of AP mode setup



Device
App
To monitor mode
Sniffing
Encoded payload (e.g., Home-AP credentials)
L5 encryption, OPT. (Vendor customized)
Wi-Fi broadcast
Connect to Internet
Local Communucation Finishes

Fig., Sample Location Communication of EZ mode setup



Device
App
To Advertising mode
Advertising Frames
L2 encryption, OPT. (LE Secure Connections)
SCAN, CONNECT_IND
BLE Binding
L5 encryption, OPT. (Vendor customized)
GATT, ATT
Application Data Exchange
Connect to Internet
Local Communucation Finishes

Fig., Sample Location Communication of BLE mode setup

Related Work

Sample Related Works	Classification	Contributions	Limitations
[8, 9, 10, 11, 12]	IoT Platform Specific Security Analysis.	Discovered the potential security weakness of a single IoT platform, such as privacy leakage of Alexa [8-10], and authentication design flaws on smartTings [11-12].	1. Only focus on the single IoT platform security. 2. Most of those works are interested in the operation state, setup is less covered.
[6, 7]	Vulnerabilities Due to exposed identity	Provided clear state inferences of IoT platform lifecycle, showing the consequence of the exposing of IDs.	1. Lack of discussion on how attackers can obtain IDs.
[13-16]	IoT Setup Security	Analyze the potential setup security for single protocols, or devices from a single vendor.	1. Lack of work measures the IoT setup confidentiality from sampled devices in the wild. 2. Lack of work has discussed the usage of the information leakage during the setup.
[17]	Most Relevant Work	Showing the consequences of how the SI exposed on setup can harm the whole platform.	1. It's a case study based on three EZ mode devices. 2. They assumed the SI will be exposure during the setup.

Table., Related Work.

Gap: Missing a work to measure the potential setup security of consumer IoT devices on the wild, from crypto implementations to information leakage, also characterizing the usage of those leaked information to discuss their sensitivity.

References

[1] Liu, H., Li, J., & Gu, D. (2020). Understanding the security of app-in-the-middle IoT. *Computers & Security*, *97*, 102000.

[2] https://www.wireshark.org/

[3] https://www.kali.org/tools/binwalk/

[4] https://hex-rays.com/ida-pro

[5] https://github.com/skylot/jadx

[6] https://github.com/secure-software-engineering/FlowDroid

[7] https://mitmproxy.org/

[8] https://github.com/frida