Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081906

ARTICLE

Security Audit of Tuya Smart Lock Using Penetration Testing Methodology

Saken Tleuberdin[1], **Dina Satybaldina**[2,*], **Raikhan Muratkhan**[3] **and Gulsipat Abisheva**[2,*]

[1]School of Cybersecurity, Astana IT University, Astana, Kazakhstan
[2]Scientific Research Institute of Information Security and Cryptology, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
[3]Department of Applied Mathematics and Informatics, Buketov Karaganda National Research University, Karaganda, Kazakhstan

*Corresponding Authors: Dina Satybaldina. Email: satybaldina_dzh@enu.kz;
Gulsipat Abisheva. Email: gulsipat.abisheva@gmail.com

Received: 10 March 2026; Accepted: 04 May 2026; Published: 23 July 2026

ABSTRACT: We perform a cross-layer penetration testing on one of the most popular Wi-Fi smart locks (Tuya 902V). The methodology combines wireless traffic analysis using an Alfa AWUS036AXML adapter, forced re-association via deauthentication to make Wi-Fi Protected Access 2 (WPA2) 4-way Extensible Authentication Protocol over LAN (EAPOL) handshake visible with Airodump/Aireplay, offline dictionary attack with Aircrack-ng, Android app reverse engineering using Apktool, Jadx, and MobSF; denial-of-service experiment (DoS) executed by hping3; Near-Field Communications (NFC)/Radio-Frequency Identification (RFID) key-clone attempt by Flipper Zero. Handshake is empirically captured but no Wi-Fi passphrase found under 14M dictionary entries; DoS test breaks cloud notification and remote control features; MObsF-assisted analysis shows dangerous permissions, many exported components, sensitive data being logged and hardcoded strings even though communication between app and lock is encrypted; Registered MIFARE Classic 1K credential fully read (including both A/B keys) however not able to emulate because of Flipper Unique Identifier (UID)/emulation limitations. This work tackles the issue of having no systematic or experimental method for discovering vulnerabilities in smart lock systems across several layers. We need to investigate how various attacks, for instance, via wireless, reverse-engineering applications, through Denial of Service, or cloning of credentials, could impact the overall security of commercially available products. We summarize the mitigations and check their relevance: enable Protected Management Frames (802.11 w) and move to Wi-Fi Protected Access 3—Simultaneous Authentication of Equals (WPA3-SAE) to reduce de-auth/handshake usefulness, enforce strong Pre-Shared Keys (PSKs), use TLS with cert pinning, rate-limit and make device APIs fault-tolerant, enforce signed/anti-rollback firmware updates, and replace static RFID tokens with Data Encryption Standard Fast Innovative Reliable and Secure (DESFire)/rolling-code credentials. The results show real attack surfaces across the network and application layers and provide actionable hardening guidance for smart-lock vendors and operators.

KEYWORDS: Android app reversing; ethical hacking; Internet of Things (IoT); penetration testing; smart-lock security; threat modeling; Wi-Fi security

1 Introduction

The modern smart locks are technologically based (mobile applications, remotes, cards, biometrics, codes) to allow convenient access and further improve security with such features as event logs and scheduled entry. They are also vulnerable to their wireless functioning: eavesdropping or hacking into the system of the lock is a frequent threat [1], and physical interference (locks are frequently left unlocked) or human mistakes

(leaving to lock out) may undermine security. Strong encryption, hardened hardware, and safe operation training of the users are mitigation measures.

A well-liked multi-factor smart lock (fingerprint, keypad, RFID card, mobile-app, mechanical key) that is designed to mount on an existing door without the need to replace the primary mechanism is the Tuya 902 V. It has built-in encryption, access logs, emergency USB power, and expansive environmental tolerance, thus a convenient option in smart home and small business.

Recent research points out multi-layer IoT threats. Gentile et al. [2] compare the security of IoT overlay networks, and Modesti et al. [3] claim that penetration testing should be conducted on all layers of the IoT systems. Zhukabayeva et al. [4] suggest that machine learning should be used alongside IoT penetration testing, and Tleuberdin et al. [5] assess the wireless IoT attacks in real systems. Ye et al. [6] discovered that there were practical attacks on a commercial smart lock (key leakage, account compromise, DoS (denial-of-service)) because of inadequate cryptography and application design. The smart-home devices and smart locks (Bluetooth Low Energy (BLE)/Wi-Fi/biometric systems) against implementation vulnerabilities are reviewed by Celestine [7]. A combination of these works suggests that smart locks must be thoroughly tested on a device, network, and application level.

We are providing a cross-layer penetration testing approach to a commercial smart lock (Tuya 902V). The wireless channels, mobile firmware, DoS resilience, and credential cloning are experimentally tested. This bridges a gap in real-life smart-lock security tests. The aim will be a systematic security evaluation and realistic guidelines for the IoT access-control systems.

The rest of the paper is organized as follows. Section 2 is a survey on the related literature on smart lock security, IoT penetration testing, and cross-layer vulnerability assessment methodologies. Section 3 has the technical background on the IoT-based smart lock systems, such as the architectural layers, Wi-Fi communication mechanisms, WPA2 security model, and the common threat vectors. Section 4 introduces the suggested penetration testing framework in reference to the PTES (Penetration Testing Performance Standard) methodology and explains the target device, experiment setting, and attack models. Section 5 describes in detail the implemented attack scenarios and experimental results, including changing the operation of Android applications, Wi-Fi handshake interception, denial of service testing, and RFID/NFC cloning attempts. Section 6 discusses the experimental results, and finally, Section 7 concludes the paper with a summary of key importance and contributions.

2 Related Works

Smart locks are a significant type of IoT tools that are utilized in contemporary access control systems. The technologies that enable access to doors by these devices include encrypted keypads, Bluetooth, smartphones, and Wi-Fi bridges. Besides the usual locking capabilities, smart locks also use remote user management and access notifications. This paper is devoted to the smart locks that do not require any modification of the door structure and do not interfere with the original mechanical system.

In the past, the security issues of wireless networks and IoT environments have been emphasized as problematic. An investigative paper published in 2020 revealed that an attack on deauthentication to Wi-Fi devices can create conditions of DoS [8]. The study conducted in [9] found the existence of significant threats to IoT devices that are used within the IEEE 802.11ah wireless networks, such as vulnerabilities in the reinstatement of session keys that can allow the decryption of packets and replay attacks in WPA2 PSK-CCMP. Johnson later discovered that there are more architectural and implementation vulnerabilities in multiple versions of the WPA protocols that can be exploited by attackers to gain access to confidential data by

interacting with the user [10]. Moreover, publicly accessible tools like Wi-Fi Exploitation Framework (WEF), Airgeddon, and Wifiphisher automate the WPA2 network attacks.

In the smart home setting, Allen et al. [11] demonstrated that smart home security systems are prone to network intrusion attacks even with defence mechanisms in place. This shows the necessity to consider smart locks as a part of a larger smart home system instead of individual gadgets. Kassem et al. [12] came up with a smart lock system that used Wi-Fi, and the need to have robust authentication and encryption methods to avoid cases of unauthorized access.

Smart locks are also controlled using mobile applications, which are also a source of security risks. The study in [13] indicated that any lack of encryption between the mobile application and the smart lock will allow hackers to intercept commands, and AES encryption has a significant amount of security enhancements. The analysis of Android applications of IoT security devices on a large scale found general vulnerabilities of insecure credential storage, absence of certificate validation, and undefended application components [14]. In the same manner, Sonamoni et al. [15] have designed an IoT system that consists of a remote door locking system with mobile integration, but the security analysis was minimal.

Other works were on system-level and firmware security. In [16], smart lock firmware de-embedding was not successful, which suggests high protection. The Verisure smart lock was tested in terms of vulnerability testing and was proven to offer an adequate degree of cyberattack protection [17]. Evidence of IoT home security systems IoT forensic analysis indicated the possibility of having important digital evidence recorded on mobile applications, such as user interactions and network communications [18]. Lastly, Ref. [19] research underlines that weak authentication, insecure firmware, and inadequate update mechanisms are a few primary IoT security issues, which explains why extensive security practices should be implemented.

3 Smart Lock Cyber Security Background

The section further discusses why securing the Internet of Things is critical and what factors make it particularly challenging. Possible attack vectors-who the potential attackers are and what mechanisms exist to protect against such threats.

Most approaches break down the system into its components and analyze their interactions to map out the attack surface. Due to the nature of applications and heterogeneity in the IoT landscape, researchers and developers could not agree on a unified architecture to describe IoT devices [20]. This paper presents an approach based on three-layer models: physical, network, and application layers [21]. The attack vectors at each of these layers differ significantly. Therefore, this model is very useful for conducting penetration testing research [22–24].

3.1 Wi-Fi Connection

Wi-Fi refers to a wireless communication technology that is anchored on the IEEE 802.11 standard that allows connecting devices to a local area network. The gadgets that have wireless network interfaces connect to access points via radio channels. Each network is named by a Service Set Identifier (SSID), which becomes visible to the users upon connection. In practice, the notions of SSID and ESSID are used interchangeably to mean network name, whilst the Basic Service Set Identifier (BSSID) is the unique MAC address of the access point (AP) [25]. Most IoT devices use the available Wi-Fi system to contact cloud-based services or other devices in a smart home network [26,27].

3.2 WPA2 Protocol

Wi-Fi supports connections through different generations of encryption protocols. The most recent and secure protocol is WPA version 3 [28]. However, many systems still run on WPA2 with its known vulnerabilities [29]. This protocol made support based on the IEEE 802.11n standard compulsory in 2004 [30]. WPA2 has two modes: Enterprise and Personal (PSK) modes. In WPA2-Personal, a common password is employed in order to verify devices and secure network traffic with the CCMP protocol using AES encryption. The authentication procedure is based on a four-way handshake of EAPOL between the client and the access point, whereby keys to secure communications are established. Though there are newer protocols available, research indicates that the majority of wireless networks continue to use WPA2, and hence, it should be considered as a critical area of security analysis [31].

3.3 Deauthentication Frames

A deauthentication frame is a type of control frame that is used to inform a station about the need to disconnect from the network. Under standard conditions, the client, having received such a frame, automatically initiates a reconnection, starting the authentication process anew. However, if such frames are repeatedly sent from the access point, the client may not be able to reconnect. This poses a serious threat, since in the IEEE 802.11i protocol underlying WPA2, control frames, including deauthentication frames, are not encrypted and authenticated. They are transmitted in unencrypted form, which allows an attacker to easily generate a fake frame, simulating sending on behalf of the access point, and thereby break the client's active connection. Subsequently, the attacker can either intercept traffic during the re-authentication process, or continue sending fake frames, effectively implementing denial of service (DoS), blocking the client's attempts to reconnect to the network.

3.4 Threats and Risks

Hacking an IoT device can impact one of the most important aspects of the confidentiality, integrity, and availability (CIA) model, especially the system availability [32]. IoT is extensively utilized in different industries, such as smart homes, industry, and healthcare. In smart home settings, there is a risk of physical infrastructure access, monitoring system interference, or appliance failure of the compromised devices. Industrial and medical IoT devices may malfunction, e.g., sensors, actuators, medical equipment, etc., to cause operational disruptions or even endanger human life [33].

IoT devices typically handle and store sensitive data, such as personal data, authentication credentials, and network configuration parameters. Unless such data is secured adequately, there is a possibility that the attackers will steal such data and use it in other malicious activities like profiling, phishing, or even unauthorized access to enterprise systems. Research has also indicated that certain IoT objects store network keys, among other sensitive data, on unencrypted memory, and even when the device is not turned on, attackers can access this data [34]. Moreover, hacked IoT devices can become a tool of entry for an additional attack in a network.

Malware and botnets are also commonly used to target IoT devices because of their constant connection and inadequate security measures. Among them, there was the Mirai botnet, which in 2016 infected hundreds of thousands of IoT devices and used them to cause large scale distributed DoS attacks on large scale online service providers [35,36]. These attacks indicate that IoT devices are easily manipulated to initiate malice such as network disruption, data theft, and any other type of cyberattack.

4 Smart Lock Penetration Testing Framework

4.1 Penetration Testing of IoT Device

Penetration testing is a form of security audit that can be performed to ensure an appropriate level of system security [37]. The research method used in this study is penetration testing, which is based on the PTES. The PTES [38,39] was developed in 2009 by an international group of independent experts and enthusiasts in the field of information security. PTES includes communication and justification through the phases of information collection, modeling, and reporting of test results, which makes PTES a reliable foundation as a standard for penetration testing. PTES has consistent and clear steps in conducting penetration testing. We combined the PTES methodology phases with our experiment steps. As the result, we obtained schema for penetration testing as shown in Fig. 1.

Figure 1: Schema of the penetration testing methodology.

The procedure of the penetration test concerning IoT devices is much like that performed on conventional computing systems. All three layers should be covered within the scope of the audit irrespective of the type or kind of target device. However, testing techniques—particularly at network and physical levels—tend to change with a specific characteristic feature attributed to some device among an application's wide range offered by IoT solutions together with involved technologies pertaining to their functioning. Such kind(s) device(s) Smart locks may have specific vulnerabilities due to services used, types of sensors, communication protocols, and other features.

4.2 Target IoT Device

The Tuya 902V is an advanced yet simple-to-use security device supporting multiple methods of authentication fingerprints numeric keypad RFID cards mobile application and mechanical key thereby offering flexible options for access control it comes with a motorized electric drive automatically turning the key installable on both horizontal and vertical locks without replacing the main mechanism features include data encryption access log and emergency power system via USB port capable of working within a wide range of temperatures and humidity conditions this makes it a universal solution to different operating conditions.

4.2.1 Physical Setup

In a system running on Tuya 902V smart lock, between primary components as router, the mobile application, and the smart lock itself. The smart lock connects to the local Wi—Fi network through a router making sure it has been integrated within the smart home infrastructure. A user interacting with the lock using Tuya Smart mobile application sends commands or requests device status over secure Internet connectivity. The application transmits these commands to Tuya cloud platform from where they are routed back down to the smart lock via router. In response to request, the smart lock either executes command (for example open, close, change access rights) or return information about the current state. This approach enables controlling device remotely as well getting notifications in real time.

4.2.2 Attack Models and Use Cases

Smart locks use different methods of authentication: biometrics, radio frequency key, PIN codes and mobile application based. As convenient as these systems can be and however advanced features they may offer, such systems remain vulnerable to various types of cyber-attacks. This paper discusses cybersecurity issues concerning smart locks-mainly those controlled by Android applications installed on smartphones or tablets. Presented further herein is an attack model that could potentially apply to similar devices in addition to specific attacks selected for practical testing checking the security of several technically diverse locks.

This paper conducts three different types of analyses as it may possibly attack smart locks. First is reverse engineering the Android application, checking if there are any possibilities to perform brute force password attacks or implanting malware. The second is a Wi-Fi attack: hacking the wireless network so that traffic can be sniffed and attacking in the sense of denial-of-service to check how tolerant it is to such fault conditions. The last one is physical intervention which includes cloning the key with Flipper Zero. This device reads and reproduces NFC and RFID signals hence emulation attacks using original keys plus bypassing authentication systems.

4.3 Hardware

Apart from the general use of a modern laptop, there are several technical conditions to which the attacker has to conform successfully. Most importantly, the wireless adapter will have to support monitor mode since it is engaged in almost all stages of executing the Python script-from capturing traffic to analyzing packets. Another network card will also be required for carrying out deauthentication attacks. In this experiment, a high-performance and energy-efficient Lenovo ThinkPad P14s Gen 4 based on an AMD Ryzen 7 PRO 7840U processor was used with a built-in dual-band Wi-Fi 6E module (Intel Wi-Fi 6E AX210) supporting three bands: two at 2.4 GHz and one at 5/6 GHz An external Wi-Fi adapter supporting the latest Wi-Fi 6 (802.11ax) standards, also capable of being set into packet monitoring and injection modes, was used. This device is identified as Alfa Network AWUS036AXML. Attacks were implemented using this adapter as part of the wireless security testing on the smart lock.

A laptop running Ubuntu 22.04 LTS was used as the host operating system. Security tools (Aircrack-ng suite, Wireshark, MobSF, hping3) were installed manually. Note that Kali Linux is the Debian-based distribution specialized for penetration testing; Ubuntu is a general-purpose OS. The wordlist for the offline attack (rockyou.txt, ~14M entries) was obtained from a standard wordlists package. During the penetration testing, tools were used to assess the security of the system.

4.4 Software

Almost all software components used in developing the Python script, together with needed libraries, belong to the standard set of Python modules. This indicates that you can begin without acquiring extra dependencies. Below is a summary of principal software tools engaged in the experiment. This list is shown in Table 1.

4.5 System Deployment and Experimental Setup

The experiments were performed on the Tuya 902V smart lock, a commercially available lock that supports Wi-Fi communication, RFID access, and mobile application control with multi-factor authentication in place. The experiments were performed in a realistic smart-home laboratory environment emulated in the laboratory. The smart lock was connected to a typical Wi-Fi network (WPA2-PSK) and paired with a mobile application running on Android and with RFID access.

Table 1: List of tools.

Tool	Description	The Stage Based on PTES Methodology
Wireshark	A traffic analyzer that allows you to intercept and analyze data packets on the network [40].	Intelligence gathering
Mobile Security Framework	Software for analyzing the security of mobile applications [41].	Vulnerability Analysis
Aircrack-ng	The software includes many programs for auditing the security of a Wi-Fi network.	Exploitation
Airodump-ng	A tool for capturing wireless packets.	Exploitation
Aireplay-ng	Software for traffic generation for subsequent use in aircrack-ng for hacking WEP and WPA-PSK keys.	Exploitation
hping3	Software for creating and sending specially generated network packets [42]. It is most often used to simulate DoS attacks.	Exploitation
Flipper Zero	The Flipper Zero is a portable multi-functional hacking device developed for interaction with access control systems [43].	Exploitation

Penetration testing of IoT devices at multiple levels was enabled through various types of hardware and software utilised for different conditions. Wireless traffic capture and packet injection was facilitated using the Alfa AWUS036AXML adapter, which offered support for monitor mode and active attack traffic. Use of Wi-Fi traffic analysis for capture of a handshake and utilisation in deauthentication attacks was conducted using the Aircrack-ng suite of tools in Airodump-ng and Aireplay-ng. For DoS experiments at the network layer with hping3 to evaluate the resilience and robustness of the device and associated cloud services. Physical access testing by utilising the Flipper Zero platform to analyse and silently clone RFID credentials. Moreover, reverse engineering and static security analysis of the Android app were performed using Apktool, Jadx, and MobSF, making it feasible to discover insecure configurations, exposed components, and data leakage. Table 1 in Section 4.3 provides a further description of the hardware and software tools used and in what roles in the experimental workflow.

Remotely performed penetration testing was conducted via a cross-layer approach using various attack vectors on the wireless, application, network, and physical access layer, respectively. At the wireless layer, a deauthentication attack was performed to disconnect the client from the access point forcing it to reconnect so that the WPA2 4-way handshake could be captured offline and attacked via dictionary attack using a dataset of 14 million passwords. At the application layer, the Android mobile application was reverse engineered and subjected to static analysis looking for insecure permissions, exported components and hardcoded sensitive information. At the network layer, DoS conditions were set against the application using packet flooding via hping3, followed by testing for service availability and cloud-based communication. Physical access layer: RFID credential extraction from a MIFARE Classic 1K card with subsequent credential emulation via Flipper Zero, together with an analysis that focuses on attack success

rate, impact on functionality, presence of exploitable vulnerabilities, and overall robustness of the system to their deployed attack.

5 Implemented Attack Scenarios and Experimental Results

This study includes practical attacks on the Tuya 902V smart lock as one of the attempts to check its vulnerability to different types of threats. Testing included network traffic interception, attempts to clone an RFID chip, deauthentication attacks, as well as dictionary attack for cracking of Wi-Fi key. Each of the attacks was performed under controlled conditions by means of tools, allowing the identification of potential vulnerabilities and evaluation of the level of security for the device in real-world operating scenarios.

5.1 Android Application Reversing

Reverse engineering analysis of an Android application allows you to get an idea of its internal structure, operating logic, and ways to interact with external services or devices. In the case of a smart lock management application, this approach can reveal command processing mechanisms, authorization features, and communication with the lock itself. One of the first stages of the analysis is to check the level of obfuscation—how much the source code was intentionally complicated to hide the logic. Besides, one can determine the IP addresses and domains to which connections are being established, which provides an understanding of the network architecture and possible points of interaction with cloud platforms. This information can be used both in the aspect of application's security evaluation and potential vulnerabilities identification.

The Tuya Smart application provides the user with advanced smart lock management capabilities through three main functions, Fig. 2. The Member Management section allows you to add new users with different access levels, which is especially convenient for family or collective use. There are two modes implemented in the Temporary Password category: Online password, recommended with a stable Wi-Fi connection, provides the ability to create a password with an unlimited validity period and Offline password, designed for situations where there is no network, where you can set both a one-time and a permanent code. Besides, Smart Linkage enables creating automation scenarios in which the smart lock interacts with other devices from the Tuya ecosystem-for example-to turn on the lighting when opening the door. This demonstrates flexible access control and extends functional capabilities of the device through intelligent integration.

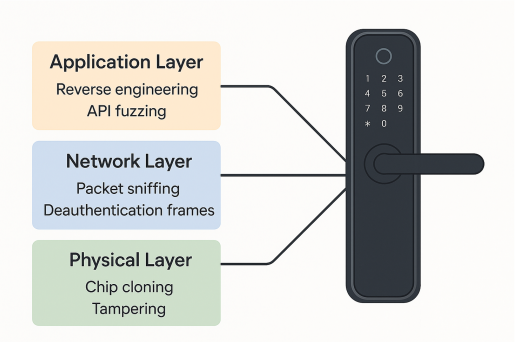

Figure 2: Tuya mobile application.

The attacker has to follow several sequential steps to finally restore the structure of an Android application. First, acquire the APK file of the intended application; this can be done by ripping the installation file from the device using some special utilities or simply downloading the APK file off open sources like

APKPure, AP-KMirror, or APKDownloader. Once acquiring the file, next is decompiling it to check its internal logic. For this, tools are used that make possible conversion of APK into some readable form-for examples, interface source resources and a bytecode proper for consequent analysis.

ApkTool is a popular tool used to decompile Android applications and gain access to the internal code representation. It allows you to disassemble an APK file into components, including interface resources and executable code, presented in smali format. Smali is a format used to display Dalvik bytecode in a form close to assembler, but maintaining a readable structure, including the names of Java and Android API classes and methods. Apktool helps to change the received code and then rebuild the modified application to run on the device. This approach is widely used in analyzing application behavior and vulnerability testing. To start decomplication, you need to use the appropriate command aimed at extracting the contents from the source APK file, Fig. 3.

```
C:\Android-reversing>java -jar apktool.jar d com.tuya.smart.apk -o smart-lock
I: Using Apktool 2.8.1 on com.tuya.smart.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\User\AppData\Local\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes10.dex...
I: Baksmaling classes11.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes4.dex...
```

Figure 3: Reversing the Android application—getting the SMALI code.

A handy tool to analyze Android apps is the Jadx Gui with a graphical interface. It delivers an approximation of the structure and logic of the program to source Java code, Fig. 4. The application can be run both in visual mode and from a command line, thus making it a handy reverse analysis application. While viewing Tuya's application using Jadx Gui, it was observed that though the code is obfuscated, this does not restrict getting good information such as URL addresses being accessed. The application turned out to be rather complicated; more than 20 permissions are required for its correct operation. Also, in AndroidManifest.xml file important information can be found, entry points into an application as well as a list of main components. The application runs encrypted communications however real security can be considered as end-to-end TLS with certificate pinning and no hard-coded secrets. In our MobSF/Jadx review we found dangerous permissions, exported components, logging sensitive data and hardcoded strings.

Table 2 contains the results of the analysis of a selected smart-lock application on Android, based on the mobile security framework. Endpoints were found after downloading and updating the application, which means major changes in its codes: endpoints for access to sensitive information and API keys besides encryption mechanisms used between the smart lock and itself. The report pointed out possible security implications regarding unsecure components and an exchange of unsecure data.

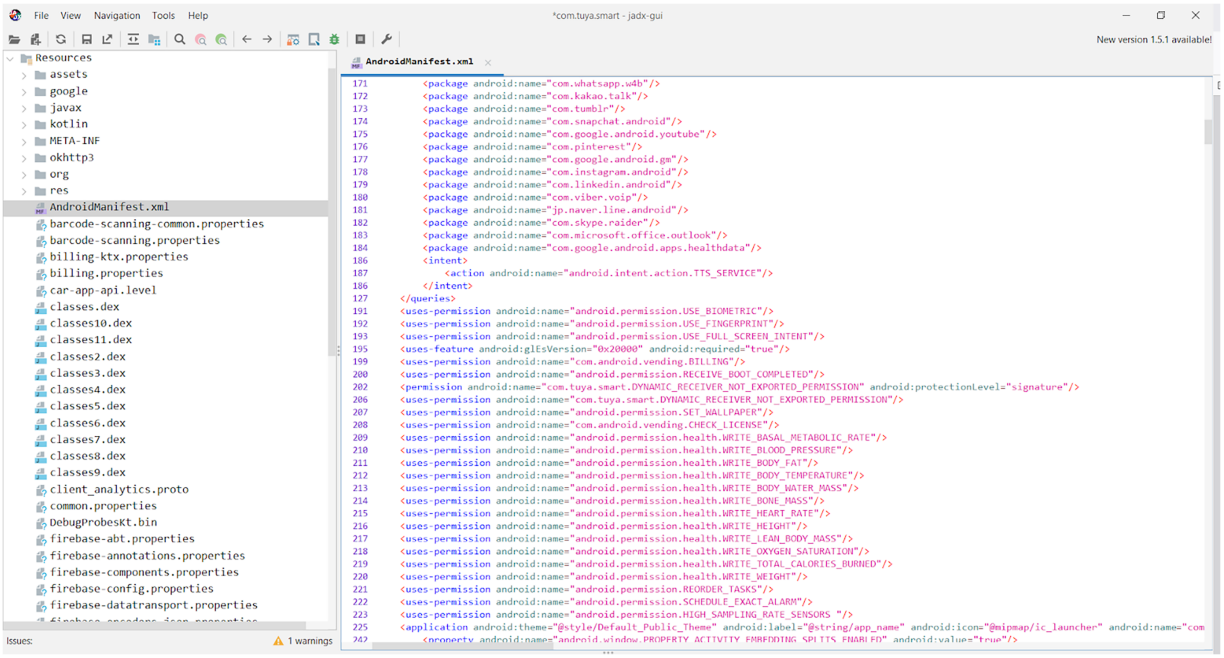

Figure 4: Reversing the Android Jadx Gui application to obtain Java code.

Table 2: Information after decompilation of the mobile application.

No	Threat	Description
1	Dangerous Permissions	The application requests dangerous permissions: microphone, camera, geolocation, Bluetooth access, storage, and network. This increases the attack surface.
2	Exported components	More than 30 exported activities, services, and receivers were found without any kind of protection. This allows other applications to access them and potentially perform attacks.
3	Signature analysis	Using only the old v1 signature makes the application vulnerable to Janus attack (CVE-2017-13156) on Android 5.0–7.0.
4	Logging sensitive data	The application stores sensitive data in logs, which violates the requirements of secure development.
5	Hard-coded data	The code contains hard-coded strings containing possible keys, tokens, and configurations, which makes the application vulnerable to reverse engineering.

5.2 Interception and Hacking of a Handshake on the WPA2-PSK Network

The target is a Wi-Fi password from a network protected by WPA2-PSK. First, fake deauth frames are sent to break the connection between the client device and the access point. The client automatically tries to reconnect, during which again the 4-way EAPOL handshake takes place. This handshake will be captured using Airodump-ng with a wireless card set on monitor mode. Now an offline brute-force attack using a dictionary of common passwords against the captured handshake to find the right key if the password is not very complex then it can be found within a short period of time thus enabling an attacker have access into supposed secured wireless network.

The following MAC addresses were identified on the wireless network as be-longing to different devices during the experiment:

- 00:EB:D8:EF:00:38—MAC address of the access point (router), Fig. 5.
- F8:17:2D:A9:5C:4D—MAC address of the Tuya 902V smart lock, Fig. 6.

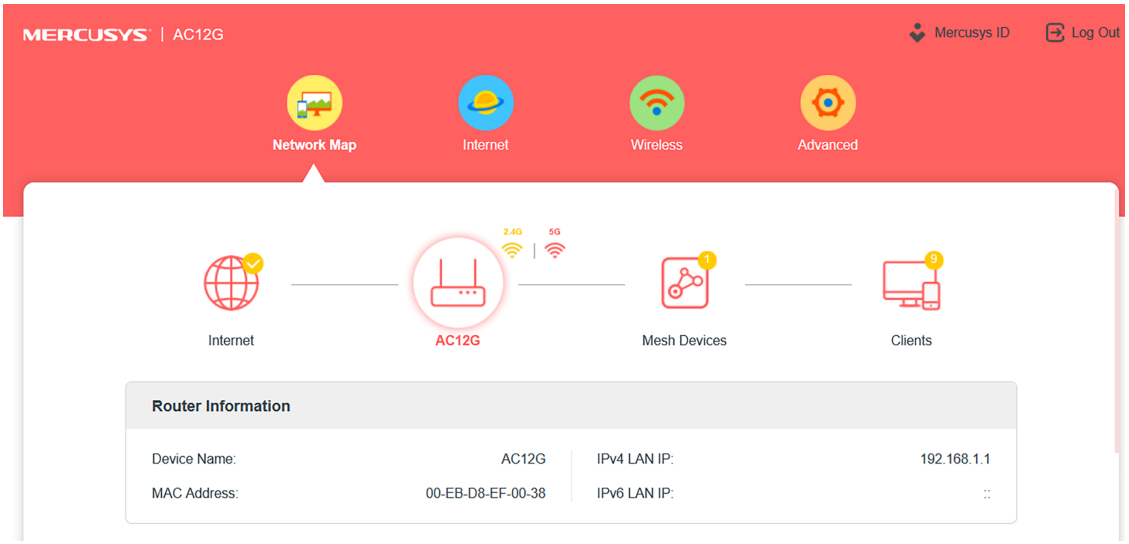

Figure 5: MAC address of the access point.

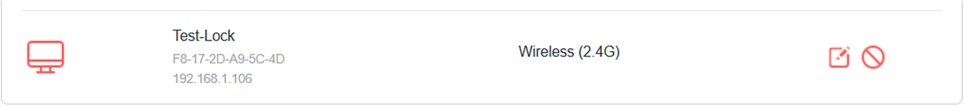

Figure 6: MAC address of the Tuya 902V smart lock.

This facilitated traffic filtering while capturing packets between different device interactions.

Using the Airodump-ng tool, a passive scan was performed to detect running Wi-Fi networks and their connected clients in Fig. 7. The AP parameters detected during the scan are SSID, type of encryption used, signal strength, and MAC addresses of the connected clients. In this phase, a target network was identified together with devices by their MAC addresses which include a router (00:EB:D8:EF:00:38) and Tuya smart lock (F8:17:2D:A9:5C:4D) that later formed basis for subsequent handshake interception and analysis traffic.

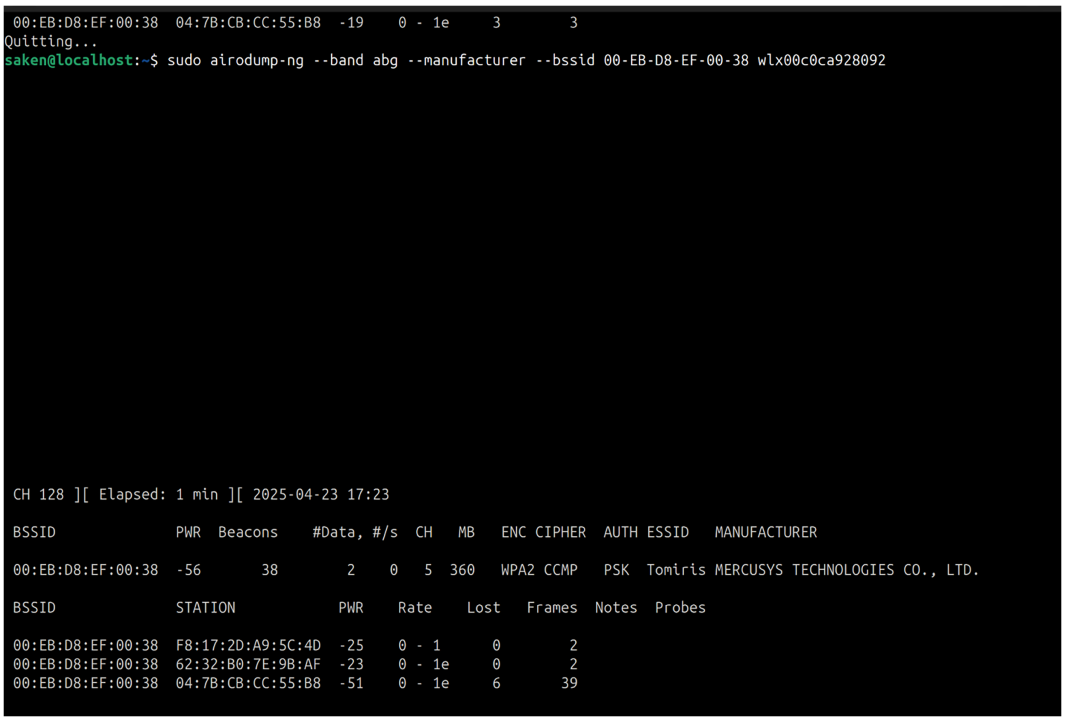

Figure 7: Wi-Fi network scanning.

The Aireplay-ng tool from the Aircrack-ng package was used to perform this deauthentication attack on the smart lock. This tool was used in sending bogus deauthentication frames simulating commands from the access point directed at the Tuya smart lock (00:EB:D8:EF:00:38) (F8:17:2D:A9:5C4D). With such intervention, the connection of the lock with the router is forcibly broken and thus enables an attempt at reconnection which creates suitable conditions for capturing a 4-way EAPOL handshake.

Aireplay-ng was used to carry out deauthentication attacks and again successfully logs the re-authentication of the device. Airodump-ng was able to capture a four-way EAPOL handshake while operating in monitor mode. That means it had been handshaked with Tuya Smart Lock (F8:17:2D:A9:5C:4D), between an Access Point (00:EB:D8:EF:00:38), which is later used for an offline dictionary attack on recovering WPA2-PSK passphrase as shown in Fig. 8.

An Internet attack on the Wi-Fi was carried out in the course of the experiment to make an attempt on recovering the password that the Tuya smart lock used. To scan the wireless environment, Airodump-ng was used to discover the access point BSSID (00:EB:D8:EF:00:38) and the mac address of the smart lock (F8:17:2D:A9:5C:4D). Fig. 8 indicates that the name of the network (SSID/ESSID) is not a MAC address but a readable name that is assigned by a human. Then, the deauthentication attack was done with the help of Aireplay-ng, and the device had to be connected again, which made it possible to capture the 4-way EAPOL handshake (Fig. 8). A dictionary attack was then performed offline with the RockYou wordlist that had over 14 million passwords. But Aircrack-ng tool could not extract the correct password to the captured handshake.

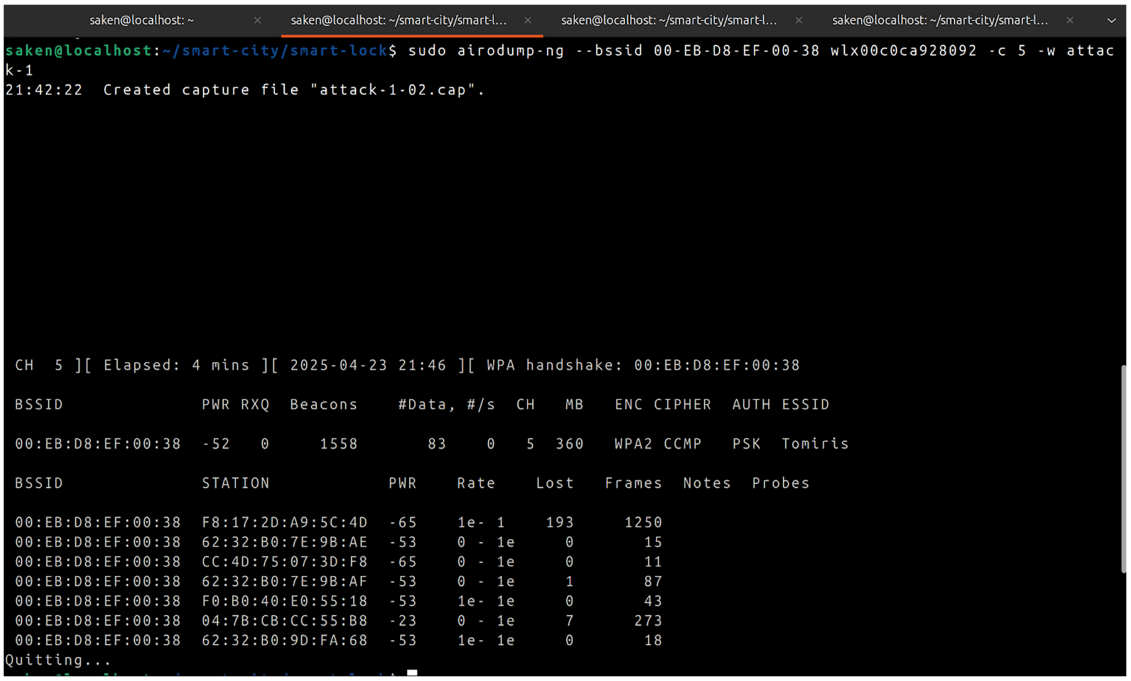

Figure 8: Intercepting the 4-way EAPOL handshake using Airodump-ng after inducing reassociation via Aireplay-ng deauthentication frames.

5.3 DoS Attack

An attack that interrupts the functionality of a smart lock is the so-called denial-of-service (DoS) attack, which is aimed at overloading the network interface or network connection to the cloud server. In case the success of such an attack, the mobile application of the device owner may cease to use the notification about the lock events, including opening and closing, which means that the communication between the device and cloud platform is disrupted.

Wireshark network sniffer was utilized to scan the traffic before the attack in order to monitor the usual behavior of Tuya 902V smart lock. The analysis was based on network packets that were being exchanged in case the lock was communicating to the mobile application. In the regular operating conditions, the device is able to send data via the router to the cloud infrastructure that sends notifications through the latter to the mobile device. Fig. 9 shows the successful transmission of packets under normal operation which will be used to compare this success under the denial-of-service experiment.

The hping3 tool was used to generate large traffic targeting the Tuya 902V smart lock during its denial-of-service attack. The attack was aimed at overloading the lock through successive TCP/UDP/ICMP requests, which was supposed to disrupt its ability to process incoming and outgoing connections. In such an attacking scenario where a single computer tries to flood the victim with packets, most probably, the smart lock will blacklist the attacker's IP address by simply dropping all incoming packets from that particular IP, Fig. 10.

In parallel, network traffic was captured using Wireshark to analyze the network status during the attack. During the monitoring process, it was found that after the attack began, the lock lost its ability to interact correctly with the cloud platform: calls stopped coming to the mobile application, and instead a connection error message appeared, Fig. 11. These observations confirm that the DoS attack successfully disrupted the network functionality of the device, temporarily blocking its integration with the user mobile application.

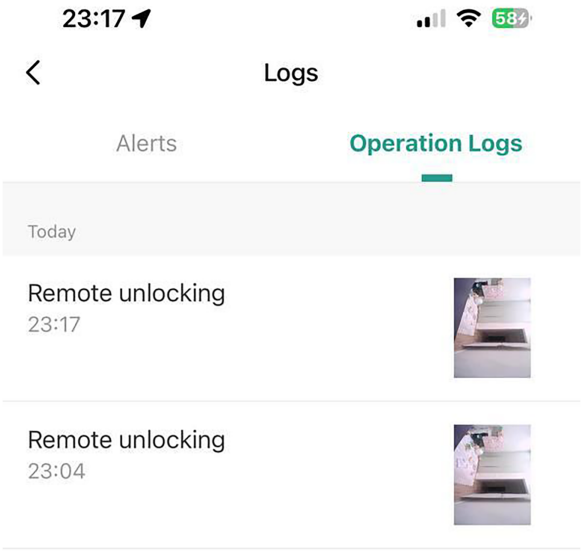

Figure 9: Stable operation during a call via smart lock.

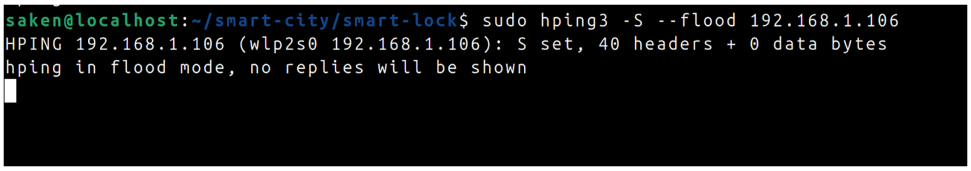

Figure 10: Generating network traffic for DoS attack.

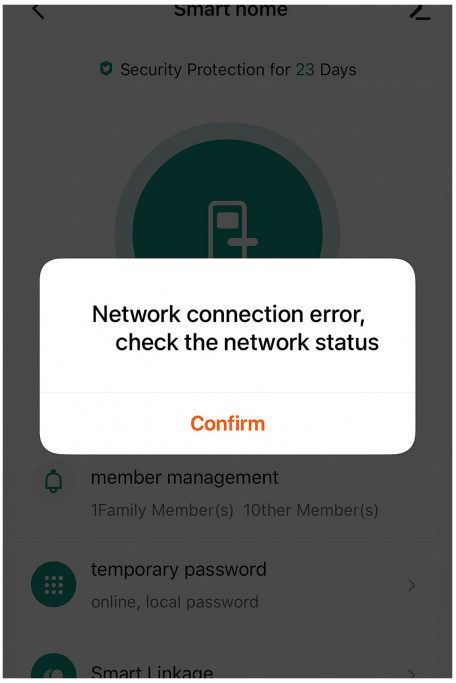

Figure 11: Connection error message.

A denial-of-service attack was performed as part of the experiment directly targeting the Tuya 902V smart lock. The attack was simply flooding the device with too many requests that temporarily breaks its network interaction with the mobile application, Fig. 12.

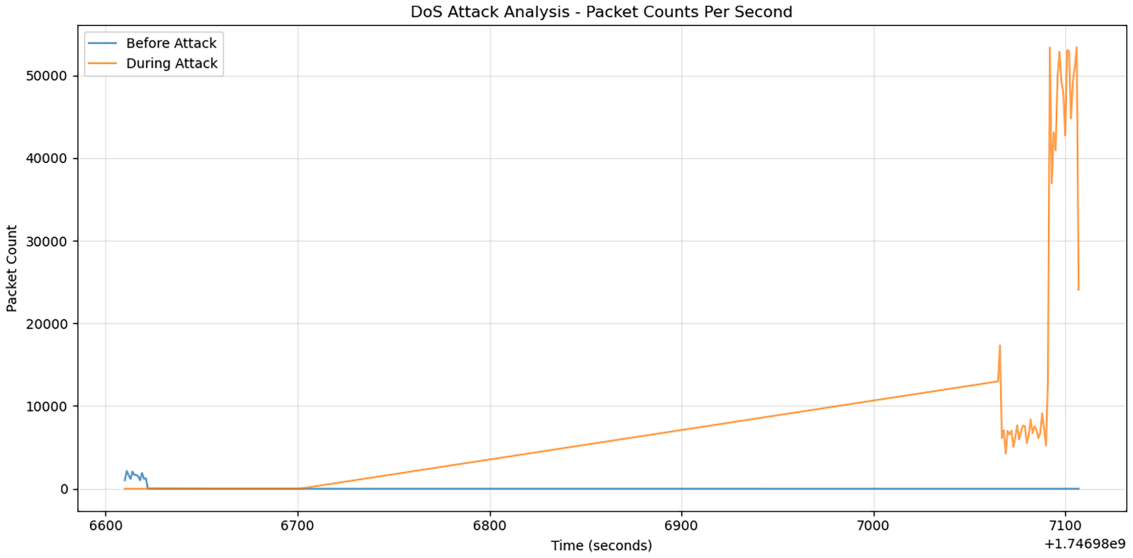

Figure 12: Graph of network traffic during DoS attack.

The lock stays active, but it cannot send any events or signals to the user's mobile application. This is confirmed by the fact that when trying to interact with the lock through the application, notifications and calls signaling actions at the lock were not received. Thus, there was a loss of communication between the device and the management interface, which confirms the successful implementation of the DoS attack and the vulnerability of the device to network denial of service.

5.4 Key Cloning

During the security analysis, RFID/NFC key cloning attack has been tried using Flipper Zero device. The method is aimed at the smart locks which use contactless authentication keys like RFID cards or keychains. The device is able to intercept the radio signal which has the unique identifier of the original credential, save it and then imitate the signal that was captured later to make an attempt of unauthorized access. In the event that the lock is not heavily encrypted or anti-cloned, then it might be able to treat the emulated credential as a legitimate one and give access. The experiment indicates that systems based on immobile and unencrypted RFID/NFC identifiers may be vulnerable and it is necessary to introduce more powerful authentication systems.

In order to test this case, a MIFARE Classic 1K card which is usually used in the contactless access system was registered in the smart lock system (Fig. 13). The test was then done by scanning this card through the Flipper Zero device to determine the possibility of credential cloning.

Flipper Zero successfully dumped the card, all sectors of memory, A/B keys—indicating there is no read protection from the original chip. However, when trying to play the captured card through Flipper Zero, limitations arise: the device cannot emulate end-to-end credentials. This reflects Flipper's limitations with MIFARE Classic (e.g., fixed UID and partial Classic emulation). More capable tools (e.g., Proxmark3–ChameleonMini) may succeed depending on card configuration (ATQA/SAK, sector keys, replay protections). Thus, despite successful readout the attack fails due to hardware limitations of emulation

which indicates that lock is partially resistant to such threat vector when using this type of equipment however when using more advanced & special devices then such attack has higher probability for success.

Figure 13: Reading the chip card using Flipper Zero.

6 Discussion

During testing, we noticed that vulnerabilities could occur precisely when the device is connected to the network. In the absence of certificate verification or protection of traffic from MITM attacks, there is a risk of AP substitution or packet interception. Such scenarios, as practice shows, are real, especially if the lock works in an insecure Wi-Fi environment or uses standard protocols without an additional layer of encryption.

It is also worth noting that with an active DoS attack, the smart lock easily stops responding, and control through the application becomes unavailable. This underlines that protection against network overloads should also be provided. At the same time, more expensive and higher—quality models are usually equipped with additional levels of protection—from dynamic keys to full-fledged TLS protection, and in such cases, hacking becomes much more complicated.

In general, the analysis showed that even if the interface is protected, vulnerabilities may be at the level of protocols or interaction logic. Therefore, when choosing a Wi-Fi lock, it is important to pay attention not only to convenience, but also to how exactly security is implemented inside the device.

7 Conclusion and Future Work

To sum up, the security level of Wi-Fi smart locks mostly depends on specific implementation, besides the stated characteristics. Most modern models do use AES encryption to protect data, especially in the channel between the application and the cloud server. However, this does not always apply to local communication, particularly if the lock is directly connected to the router without any intermediate security mechanisms.

Future work can be directed toward several promising areas to facilitate an in-depth vulnerability analysis of smart lock systems. One such area lies within the comprehensive analysis of application codes and firmware for smart locks. Thorough analyses may uncover coding errors or insecure logic permitting unauthorized access or bypassing security checks intended to be critical. Such findings could serve as a basis for formulating prevention strategies and improving safe development methodologies.

The other possible direction is in creating altered Android applications that can talk to smart locks without going through the usual authentication process. By reverse engineering APK files and recovering network requests, it will be possible to check if tokens or logic are vulnerable to manipulation to use the system.

Finally, hardware attacks-in particular, the cloning of RFID/NFC credentials-constitute an important area for future work. With advanced tools such as Proxmark3 and ChameleonMini supporting more features to capture and copy access tokens, even though some smart locks eliminate this problem by using encryption and anti-cloning protection, there are others that still rely on static identifiers that can be intercepted and replayed. Experimental validation of these attacks across a larger diversity of lock models may help pinpoint systemic weaknesses in credential processing.

Thus, further research is required at both the software and hardware levels of smart lock systems. As these devices are becoming most widely adopted first at home and then in enterprises, ensuring them against new forms of threats, resilience will be most crucial to maintain user trust and, ultimately, physical access control protection.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24992852 "Intelligent models and methods of Smart City digital ecosystem for sustainable development and the citizens' quality of life improvement").

Author Contributions: Conceptualization, Dina Satybaldina; methodology, Dina Satybaldina and Saken Tleuberdin; software, Saken Tleuberdin; validation, Dina Satybaldina, Raikhan Muratkhan and Gulsipat Abisheva; formal analysis, Dina Satybaldina; investigation, Saken Tleuberdin; resources, Gulsipat Abisheva; data curation, Saken Tleuberdin and Raikhan Muratkhan; writing—original draft preparation, Saken Tleuberdin and Raikhan Muratkhan; writing—review and editing, Dina Satybaldina, Raikhan Muratkhan and Gulsipat Abisheva; visualization, Saken Tleuberdin and Gulsipat Abisheva; supervision, Dina Satybaldina; project administration, Dina Satybaldina; funding acquisition, Dina Satybaldina All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, [Dina Satybaldina], upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AES	Advanced Encryption Standard
AP	Access Point
ATQA/SAK	Android application. First, acquire the APK file
BSSID	Basic Service Set Identifier
CIA	Confidentiality–integrity–accessibility
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
DoS	Denial of Service
EAPOL	Extensible Authentication Protocol over LAN
ESSID	Extended Service Set Identifier
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
MAC	Media Access Control
MITM	Man-In-The-Middle
PMK	Pairwise Master Key
PSK	Pre-Shared Key
TLS	Transport Layer Security

IoT	Internet of Things
IP	Internet Protocol
PHY	Physical Layer
PIN	Personal Identification Number
PTES	Penetration Testing Execution Standard
SSID	Service Set Identifier
SQL	Structured Query Language
WLAN	Wireless Local Area Networks
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2
RFID	Radio-Frequency Identification
NFC	Near-Field Communications
WPA3	Wi-Fi Protected Access 3
CVE	Common Vulnerabilities and Exposures
URL	Uniform Resource Locator
CVSS	Common Vulnerability Scoring System
AES	Advanced Encryption Standard
AP	Access Point

References

1. Palle S. Smart locks: exploring security breaches and access extensions [master's thesis]. Stillwater, OK, USA: Oklahoma State University; 2017.
2. Gentile AF, Macrì D, Greco E, Fazio P. IoT IP overlay network security performance analysis with open source infrastructure deployment. J Cybersecur Priv. 2024;4(3):629–49. doi:10.3390/jcp4030030.
3. Modesti P, Golightly L, Holmes L, Opara C, Moscini M. Bridging the gap: a survey and classification of research-informed ethical hacking tools. J Cybersecur Priv. 2024;4(3):410–48. doi:10.3390/jcp4030021.
4. Zhukabayeva T, Ahmad Z, Adamova A, Karabayev N, Mardenov Y, Satybaldina D. Penetration testing and machine learning-driven cybersecurity framework for IoT and smart city wireless networks. IEEE Access. 2025;13(8):86144–66. doi:10.1109/access.2025.3569965.
5. Tleuberdin S, Malakhov K, Tashatov N, Satybaldina D, Yedilkhan D. Reinforcement learning-based adaptive penetration testing framework for wireless communication. Int J Adv Comput Sci Appl. 2026;17(3):355–63. doi:10.14569/IJACSA.2026.0170329.
6. Ye M, Jiang N, Yang H, Yan Q. Security analysis of Internet-of-Things: a case study of august smart lock. In: Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 2017 May 1–4; Atlanta, GA, USA. p. 499–504. doi:10.1109/infcomw.2017.8116427.
7. Celestine D. Smart lock systems: an overview. Int J Comput Appl. 2020;177(37):40–3. doi:10.5120/ijca2020919882.
8. Aggrawal A, Arora I, Giri A. Analysis and rendering of deauthentication attack using IoT technology. In: Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications; 2022 Mar 28–29; Hyderabad, India. Singapore: Springer Nature; 2023. p. 41–51. doi:10.1007/978-981-19-6088-8_5.
9. Yin W, Hu P, Zhou H, Xing G, Wen J. Jamming attacks and defenses for fast association in IEEE 802.11ah networks. Comput Netw. 2022;208(2):108890. doi:10.1016/j.comnet.2022.108890.
10. Vanhoef M. Fragment and forge: breaking {Wi-Fi} through frame aggregation and fragmentation. In: Proceedings of the 30th USENIX Security Symposium (USENIX Security 21); 2021 Aug 11–13; Vancouver, BC, Canada. p. 161–78.
11. Allen A, Mylonas A, Vidalis S, Gritzalis D. Smart homes under siege: assessing the robustness of physical security against wireless network attacks. Comput Secur. 2024;139(10):103687. doi:10.1016/j.cose.2023.103687.

12. Kassem A, El Murr S, Jamous G, Saad E, Geagea M. A smart lock system using Wi-Fi security. In: Proceedings of the 2016 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA); 2016 Jul 13–15; Zouk Mosbeh, Lebanon. p. 222–5. doi:10.1109/actea.2016.7560143.
13. Caballero-Gil C, Álvarez R, Hernández-Goya C, Molina-Gil J. Research on smart-locks cybersecurity and vulnerabilities. Wirel Netw. 2024;30(6):5905–17. doi:10.1007/s11276-023-03376-8.
14. Allen A, Mylonas A, Vidalis S, Gritzalis D. Security evaluation of companion Android applications in IoT: the case of smart security devices. Sensors. 2024;24(17):5465. doi:10.3390/s24175465.
15. Sonamoni JS, Sikdar R, Akib ASMAS, Islam MS, Sourov S, Al Ahasan MA, et al. IoT-based smart remote door lock and monitoring system using an Android application. Eng Proc. 2024;76:85. doi:10.3390/engproc2024076085.
16. Borg A, Francke CA. IoT pentesting: obtaining the firmware of a smart lock [dissertation]. Stockholm, Sweden: KTH, School of Electrical Engineering and Computer Science; 2020.
17. Hassani R. Security evaluation of a smart lock system [dissertation]. Stockholm, Sweden: KTH, School of Electrical Engineering and Computer Science; 2020.
18. Hutchinson S, Stanković M, Ho S, Houshmand S, Karabiyik U. Investigating the privacy and security of the SimpliSafe security system on Android and iOS. J Cybersecur Priv. 2023;3(2):145–65. doi:10.3390/jcp3020009.
19. Bour G, Bosco C, Ugarelli R, Jaatun MG. Water-tight IoT-just add security. J Cybersecur Priv. 2023;3(1):76–94. doi:10.3390/jcp3010006.
20. Ali Jabraeil Jamali M, Bahrami B, Heidari A, Allahverdizadeh P, Norouzi F. Towards the Internet of Things: architectures, security, and applications. Cham, Switzerland: Springer International Publishing; 2020. doi:10.1007/978-3-030-18468-1.
21. Færøy F, Yamin M, Shukla A, Katt B. Automatic verification and execution of cyber attack on IoT devices. Sensors. 2023;23(2):733. doi:10.3390/s23020733.
22. Chu G, Lisitsa A. Penetration testing for Internet of Things and its automation. In: Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS); 2018 Jun 28–30; Exeter, UK. p. 1479–84. doi:10.1109/hpcc/smartcity/dss.2018.00244.
23. Hossain MM, Fotouhi M, Hasan R. Towards an analysis of security issues, challenges, and open problems in the Internet of Things. In: Proceedings of the 2015 IEEE World Congress on Services; 2015 Jun 27–Jul 2; New York, NY, USA. p. 21–8. doi:10.1109/services.2015.12.
24. Rose C. The security implications of the Internet of Things. J Cybersecur Res. 2017;2(1):1–4. doi:10.19030/jcr.v2i1.9931.
25. Hernandez G, Arias O, Buentello D, Jin Y. Smart nest thermostat: a smart spy in your home. In: Proceedings of the Black Hat USA 2015; 2015 Aug 1–6; Las Vegas, NV, USA.
26. Pahlavan K, Krishnamurthy P. Evolution and impact of Wi-Fi technology and applications: a historical perspective. Int J Wirel Inf Netw. 2021;28(1):3–19. doi:10.1007/s10776-020-00501-8.
27. Færøy FL. Expanding the capabilities of cyber range attack agents [master's thesis]. Trondheim, Sweden: NTNU; 2022.
28. Afzal F, Uzair A, Javed MA, Naqvi SAA. An enhanced approach for Wi-Fi Security and authentication protocols: a systematic approach towards WEP, WPA, WPA2, and WPA3. Spectr Eng Sci. 2024;2(5):379–403.
29. Moissinac K, Ramos D, Rendon G, Elleithy A. Wireless encryption and WPA2 weaknesses. In: Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC); 2021 Jan 27–30; Las Vegas, NV, USA. p. 1007–15. doi:10.1109/ccwc51732.2021.9376023.
30. Au E. Introduction to IEEE computer society standards activities. Computer. 2024;57(12):129–32. doi:10.1109/mc.2024.3466328.
31. National Security Agency (NSA). Cybersecurity technical report: WPA3. 2018 Jun (Public Release 2019). [cited 2025 Oct 2]. Available from: https://media.defense.gov/2019/Jul/16/2002158109/-1/-1/0/CTR-CYBERSECURITY-TECHNICAL-REPORT-WPA3.PDF.
32. Samonas S, Coss D. The CIA strikes back: redefining confidentiality, integrity and availability in security. J Inf Syst Secur. 2014;10(3):21–45.

33. Leavitt N. Researchers fight to keep implanted medical devices safe from hackers. Computer. 2010;43(8):11–4. doi:10.1109/mc.2010.237.
34. Bonaventura D, Esposito S, Bella G. Smart bulbs can be hacked to hack into your household. arXiv:2308.09019. 2023.
35. Borys A, Kamruzzaman A, Thakur HN, Brickley JC, Ali ML, Thakur K. An evaluation of IoT DDoS cryptojacking malware and mirai botnet. In: Proceedings of the 2022 IEEE World AI IoT Congress (AIIoT); 2022 Jun 6–9; Seattle, WA, USA. p. 725–9. doi:10.1109/aiiot54504.2022.9817163.
36. Affinito A, Zinno S, Stanco G, Botta A, Ventre G. The evolution of Mirai botnet scans over a six-year period. J Inf Secur Appl. 2023;79(2):103629. doi:10.1016/j.jisa.2023.103629.
37. Office of chief information officer, US department of the interior. 2022 [cited 2025 Oct 7]. Available from: https://www.doi.gov/ocio/customers/penetration-testing/.
38. Sarker KU, Yunus F, Deraman A. Penetration taxonomy: a systematic review on the penetration process, framework, standards, tools, and scoring methods. Sustainability. 2023;15(13):10471. doi:10.3390/su151310471.
39. Wilhelm T. Professional penetration testing: creating and learning in a hacking lab. Amsterdam, The Netherlands: Elsevier; 2025.
40. Bock L. Learn Wireshark: a definitive guide to expertly analyzing protocols and troubleshooting networks using Wireshark. Birmingham, UK: Packt Publishing Ltd.; 2022.
41. Kusreynada SU, Barkah AS. Android apps vulnerability detection with static and dynamic analysis approach using MOBSF. J Comput Sci Eng. 2024;5(1):46–63. doi:10.36596/jcse.v5i1.789.
42. Huraj L, Šimon M, Lietava J. Dataset of DDoS attacks on Fibaro home center 3 for smart home security. Data Brief. 2024;57(5):110991. doi:10.1016/j.dib.2024.110991.
43. Das Dev A, Dr Shyam R. Flipper zero in action: a comparative review of six methods for enhanced cybersecurity tactics. Int J Res Publ Rev. 2024;5(9):1317–21. doi:10.55248/gengpi.5.0924.2525.