Access Your Tesla without Your Awareness: Compromising Keyless Entry System of Model 3

Xinyi Xie*†, Kun Jiang*†, Rui Dai†, Jun Lu†, Lihui Wang†, Qing Li†§✉, Jun Yu†§
†Shanghai Fudan Microelectronics Group Co., Ltd.
§State Key Laboratory of ASIC & System, Fudan University
Email: {xiexinyi, jiangkun, dairui, lujun, wanglihui, liqing, yujun}@fmsh.com.cn

***Abstract*—Tesla Model 3 has equipped with Phone Keys and Key Cards in addition to traditional key fobs for better driving experiences. These new features allow a driver to enter and start the vehicle without using a mechanical key through a wireless authentication process between the vehicle and the key. Unlike the requirements of swiping against the car for Key Cards, the Tesla mobile app's Phone Key feature can unlock a Model 3 while your smartphone is still in a pocket or bag.**

In this paper, we performed a detailed security analysis aiming at Tesla keys, especially for Key Cards and Phone Keys. Starting with reverse engineering the mobile application and sniffing the communication data, we reestablished pairing and authentication protocols and analyzed their potential issues. Missing the certificate verification allows an unofficial Key Card to work as an official one. Using these third-party products may lead to serious security problems. Also, the weaknesses of the current protocol lead to a man-in-the-middle (MitM) attack through a Bluetooth channel. The MitM attack is an improved relay attack breaking the security of the authentication procedures for Phone Keys. We also developed an App named TESmLA installed on customized Android devices to complete the proof-of-concept. The attackers can break into Tesla Model 3 and drive it away without the awareness of the car owner. Our results bring into question the security of Passive Keyless Entry and Start (PKES) and Bluetooth implementations in security-critical applications. To mitigate the security problems, we discussed the corresponding countermeasures and feasible secure scheme in the future.

I. Introduction

Passive Keyless Entry and Start (PKES) is an intelligent automotive system allowing drivers to pull the door directly to enter and start the car. The traditional PKES requires a key fob to provide the legitimacy and the vehicle to verify it. New car models of Tesla, Volvo, Mercedes-Benz, and Lincoln enable car owners to use their smartphones to unlock and activate their cars automatically [81], [85]. To continuously enrich the driving experiences, Tesla supports three types of keys: Phone Keys, Key Cards and key fobs [52] for Model 3, Model X, Model Y, and Model S.

* Both authors contributed equally.

Network and Distributed System Security (NDSS) Symposium 2023
27 February - 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.24082
www.ndss-symposium.org

Key fobs adopted in classical PKES contain a low-frequency (LF) radio frequency identification (RFID) tag and ultrahigh-frequency (UHF) transceiver, and the vehicle equips with an LF receiver and UHF RFID tag [48]. The LF channel is responsible for detecting whether the key fob is within an allowed region, whereas the UHF channel is for challenge-response verification. Relay attack is a widely known vulnerability against this technique used in key fobs [1], [2], [27], [47]. It allows adversaries to open and start the car by distance fraud. Distance bounding technique is commonly proposed to prevent relay attacks [32], [45], [62], [70]. Karani et al. [42] and Lin et al. [49] designed and implemented the PKES based on BLE. This new PKES monitors the BLE received signal strength indicator (RSSI) measurements from the key to estimate users' proximity. Some works insist that considering other techniques or features should be a better solution. They utilize multiple physical features, namely, RSSI, Round-Trip time, Global Position System (GPS) coordinates, and Wi-Fi access point lists, to precisely identify the proximity of a vehicle to its corresponding key fob [20], [63], [83], [84]. A. Ansari et al. [5] transmits the cryptographically secure combined bio-crypto data to enhance the authentication.

Focus on the security issues of Tesla key fobs, considering the first version of the Tesla Model S key fob based on DST40, the research [15] proves that it is susceptible to a brute-force attack. Moreover, this first version misses mutual authentication in the challenge-response protocol. It also has no firmware read-out protection and security partitioning [87]. Further, Wouters et al. [88] target the second version of the key fob based on DST80. Through reverse engineering immobilizer firmware, they recover the key by downgrade attack leading to a reduction of key entropy. They also describe a Denial-Of-Service attack, which can render the key fob unusable. Also, Tesla warned of theft risk through relay attacks [75]. To fix bugs and add features, Tesla introduced over-the-air updates. However, Wouters et al. [78], [86] declared that a hacker could rewrite the firmware of a key fob via Bluetooth connection, lift an unlock code from the fob, and use it to break a Model X just in a few minutes.

Key Cards communicate with the vehicle using RFID signals in standard ISO 14443. When you tap the Key Card on the driver's side pillar, doors can be opened. However, it does not support automatic locking and unlocking. Tesla Model 3 vehicles allow attackers to open a door by leveraging access to a legitimate Key Card and using a Near Field Communications (NFC) relay attack [21]. When an owner enters the car, he needs to swipe the Key Card on the console right by the cup

holder to drive the vehicle. Tesla issued several updates around the Key Cards. These updates allow the owner to start a Model 3/Model Y after unlocking the doors with the digital Key Card, as the key does not need to be placed in the center console to shift out of parking and drive off. Security researchers indicated that it allows a 130-second window between an owner unlocking the door and driving the car. During this time window, new Tesla Key Cards can be added without any authentication required and with no in-car and in-app notification [54].

The developments and widespread applications of Phone Keys follow the trend. Phone Keys supporting PKES communicate with vehicles through Bluetooth Low Energy (BLE) channel. The BLE protocols are specified in an open standard maintained by Bluetooth Special Interest Group (SIG), and its latest version is 5.3 [13]. Recently, the NCC group announced that Tesla Model 3 and Model Y are vulnerable to link-layer relay attacks on BLE [74]. The vehicle is fooled into believing that an authorized driver owns a Key Card or registered smartphone.

This paper demonstrates the detailed security analysis of Tesla keys pairing and authentication processes, especially for Key Cards and Phone Keys. Traditional key fobs susceptible to relay attacks are out of the discussion here. Even though many works have analyzed the security of RFID and BLE techniques, our studies focus on the practical implementation of the Tesla Model 3. We conduct a transparent view of the interactions between keys and vehicles. By analyzing the vulnerabilities of protocols, we find that third parties products that support ISO 14443 and related cryptography can work as official ones. We utilized a customized Java card to prove it. Besides, the authentication procedure of a Phone Key is susceptible to MitM attacks or relay attacks. We also performed an improved relay attack on the authentication process of Phone Keys. Compared to known attacks, it can be performed by only one device and is not a real-time relay. This attack happens silently in the background and does not generate in-car or in-phone notifications to the owner. In summary, we make the following contributions:

- With reverse engineering the Tesla mobile app, we achieved many clues about the algorithm and mechanisms. We captured the communication packets by sniffing pairing and authentication processes in ISO 14443 channel and the BLE channel. Finally, we recovered the pairing and authentication protocols of Key Cards and Phone Keys.
- The weakness in the Key Card pairing protocol allows customized Key Cards to be available. We used a programmable Java card to show that these third-party products would jeopardize the security of the Tesla ecosystem.
- We combined several flaws to craft a practical MitM attack for Tesla Model 3 in the Phone Key authentication process. This attack can fool the vehicle into believing that it has connected to a registered Phone Key resulting in entering and driving a Tesla Model 3 away without any operations on the owner's phone and out of the owner's awareness.

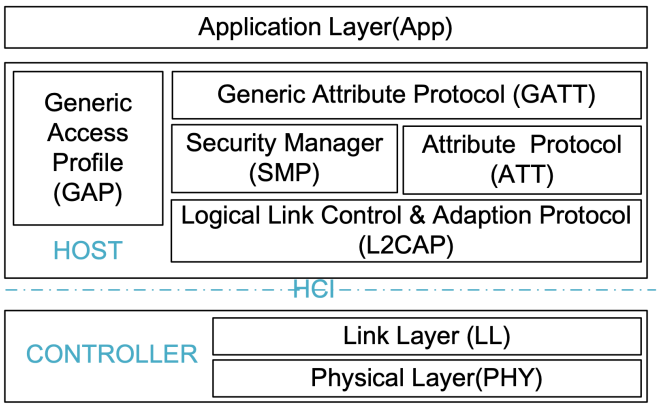

Fig. 1: BLE Architecture

- We demonstrated the feasibility of the MitM attack by implementing it on Android devices. We proposed a novel backdoor firmware and customized framework to force Google Pixel 5A to broadcast by a specific static MAC address. We also developed an Android app named TESmLA to implement the attack.

The rest of the paper is organized as follows. In Section 2, we provide background on BLE and relevant cryptography. In Section 3, we introduce the methodology of reverse engineering. We detail the pairing and authentication protocols in Section 4. The potential problems are described in Section 5. Section 6 shows Key Card threats, while Section 7 reports the MitM attack and its related works in detail. Finally, we discuss mitigations and future works in Section 8 before concluding in Section 9.

II. Background

A. BLE

Bluetooth Low Energy (BLE) is designed for energy-constrained low-cost scenarios that require an intermittent transfer of small amounts of data at a lower speed compared to Bluetooth Classic protocols. The BLE protocol enables a device, acting as a server (e.g., a Tesla Model 3), to efficiently communicate relevant data to another connected device, acting as a client (e.g., a Phone Key). As shown in Figure 1, the architecture of the BLE core system has two parts: a host stack and a controller. The Host Controller Interface (HCI) between them is responsible for transporting commands and events between the host stack and the controller.

The GAP profile provides several procedures for device discovery, connectivity, and related network topology. The primary data exchange method of GAP is a broadcast advertising package. It may be followed by a scan response if a peer device requests. The BLE advertisement header contains the device role, the MAC address, and the message type. Tesla mobile app as the Phone Key scans devices in the BLE range by calling the interface of the GAP. The advertisements from the vehicle can be captured and analyzed to get necessary information by the Phone Key. The GATT

interfaces support discovering, reading, writing, and obtaining indications of characteristics [13]. After the BLE connection has been created, Tesla mobile app as a client can communicate with the vehicle by calling the interfaces of the GATT layers to do operations on characteristics.

B. Relevant Cryptography

ECDH. The Elliptic Curve Diffie Hellman (ECDH) is an key agreement scheme defined in [19], [23]. It allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. This scheme prevents the third party who can eavesdrop on the negotiation process from recovering the shared secret.

AES-GCM. AES with Galois/Counter Mode (AES-GCM) specified in NIST 800-38D [25] provides both authenticated encryption and the ability to check the integrity and authentication of additional data. There are four inputs: secret key, initialization vector(IV)/nonce, plaintext, and optional additional authentication data (AAD). The output consists of the ciphertext, which is the same length as the plaintext, and an authentication tag for integrity check.

III. Methodology

In this section, we describe our work on reverse engineering. Our analysis required a comprehensive understanding of the implemented protocols by Tesla. Unfortunately, no official documentation from Tesla was available regarding the pairing and authentication of keys. Therefore, we reversed the protocols using open-source software and off-the-shelf commodity hardware. We took the official mobile application named Tesla as the entry point because it contains more information compared to the communication data. For example, it may include detailed algorithms and corresponding parameters. So we start with reversing this official application on Android and iOS. We further capture the BLE communication of Phone Keys. The analysis helps us understand protocols of Phone Keys and give clues to further analyze the Key Card. We describe the details as follows.

Mobile App analysis. Nowadays, reverse technology with related tools is widely used for mobile applications. We reversed the Tesla official Android application package (.apk) file with the releases of 4.23 and the iPhone Application (.ipa) file with version 15.4.1. We first made attempts to static analysis. We used open-source JDAX [40] to get the Java Code for Android and IDA Freeware [37] to analyze the iOS application. Although methods and variable names are obfuscated in reversed Java code, the code in reversed iOS gives hints and compensation. The benefit of static analysis is to locate a Java object, which completes many operations related to cryptography, like the AES-GCM and SHA1. These operations must be necessary for the process of pairing and authentication processes. So we dynamically analyzed the app to observe the calling of corresponding functions based on this object. Utilizing the Frida [28], a dynamic instrumentation toolkit, we hooked and observed three particular behavior: First, the login process using a legitimate username and password. Second, the first time for a Phone Key to pair with the vehicle. Third, the authentication of a Phone Key. The dynamic analysis helps us make sure of the role that the specific algorithm plays in these processes.

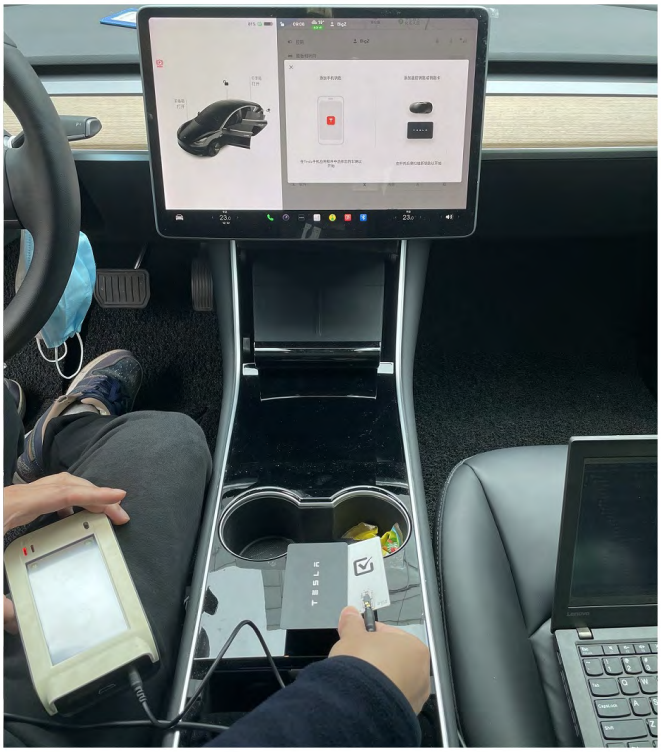

Fig. 2: Key Card Sniffer Scene

BLE data capture and analysis. We also retrieved the HCI Snoop log from Android. This method is stabler and easier to implement. The HCI Snoop log shows the entire interaction between the host and the controller, including the communication data used in pairing and authentication. From this log, we can understand some implementation details, like the advertisement of the vehicle, GATT services, and connection parameters. We analyzed contiguous messages from the advertisement capture to the connection establishment until the end of the communication.

From the above process, we comprehend the pairing and authentication protocols between the Phone Key and the vehicle, including the cryptography and parameters used in these processes. For example, when the first login, the application calls the key generation function to create a key pair based on the NIST P-256 curve. Then the private key is protected by Android KeyStore in case of leakage. KeyStore can protect sensitive materials from unauthorized use in the non-root environment. The information, like algorithms and authentication process, help further analyze the Key Card.

Key Card analysis. Tesla Key Card uses Java Card manufactured by NXP that supports ISO 14443. This kind of card with a high-security level has achieved Common Criteria certification. Many off-the-shelf professional spy devices like Microprocess MP300 [53] and NomadLAB [55] can sniff ISO 14443 signals. As shown in Figure 2, we placed an ISO 14443 spy device named NomadLAB parallel to the Key Card against the card reader on Model 3. And the white Printed Circuit Board (PCB) in the figure is an RFID coil. This coil passes the captured RF signal to NomadLAB, and NomadLAB is responsible for automatically recording APDU

data for further analysis. From these records, We noticed the vehicle's public key, which had already been achieved from the previous analysis. This public key is in uncompressed format with the prefix "0x04". The following 64 byte is the x and y coordinate of the point on the elliptic curve. As we clarified the expression format of public keys, we also found the other three public keys of different Key Cards in APDU data.

According to sniffer data, the authentication of Key Cards is based on a 16-byte challenge and response. We conjectured that it might use a similar method as Phone Keys. It may use the ECDH to negotiate a shared secret and AES-ECB to compute the response. However, the secret key of the Key Card and the vehicle are both securely stored in a secure element(SE). It is hard to verify our conjecture. In this case, we used an Android Phone Key, which is already root and paired with the car. We achieved the key pair of this device by Frida. And we used a programmable Java card and developed a Java applet to implement a challenge-response mechanism. Fortunately, when we send the corresponding response to the vehicle, the authentication success. Based on conjecture and verification, we finally re-established the protocols of Key Cards.

IV. Pairing and Authentication Protocols

The pairing procedure is the process of adding a new Key Card or a Phone Key. First, the vehicle has to confirm the identification of the new key. This process needs a paired key as a credential. Then the car records the information of the new key locally for later authentication. Tesla Model 3 ideally can authenticate and distinguish the legitimate key from malicious signals or interference through identifications from the pairing stage. The authentication requires that two parties are each pre-provisioned with a unique asymmetric key pair. Through the ECDH key agreement, the Phone Key or the Key Card shares a symmetric secret with the vehicle. Key Cards and Phone Keys protocols use different communication channels. Key Cards uses ISO 14443 for data exchange. Phone Keys allow users to use their smartphones as keys in the BLE channel. In the following parts, we clearly illustrate the security mechanisms for Key Cards and Phone Keys, including the pairing and authentication protocols.

A. Key Cards

Pairing. To pair a new card, scan your new Key Card on the card reader located on the top of the center console and follow the on-screen instructions. We have recorded the exchanged data during this process until the success notification shows on the car's screen. We analyzed the communication data and reestablished the pairing procedure in Figure 3.

Before pairing, each Key Card and vehicle has been provisioned its own 256-bit ECDH public-private key pair based on the NIST P-256 curve. For the initial pairing phase, the owner taps an unpaired Key Card against the vehicle card reader on the center console. The connection establishes through the RFID channel. Tesla Model 3 grabs the public key (PK) and unique identifier (UID) (Step 2) of the card. After a paired Key Card is presented to the card reader on the driver's side door pillar or the top of the center console, the car will get its public key (Steps 4 - 6). Then it performs an ECDH key agreement scheme to derive the shared secret with the own private key and the public key of the paired Key Card. The car delivers its public key with a 16 bytes salt as a challenge to the card (Step 9). Once received, The Key Card also performs ECDH key derivation, encrypts the challenge with the shared secret as the key, and returns the encryption result (Steps 10 - 12). By checking this response, Tesla Model 3 verifies whether this key has already been paired. If a paired key passes the verification, the vehicle will insist that the new Key Card belongs to the owner, who has at least one legitimate key. For now, the vehicle has confirmed the identification of the new Key Card.

If the vehicle want to authorize this new card, the car needs to records related information for latter authentication. As a result, Tesla Model 3 adds the public key of this new Key Card into the whitelist. The pairing succeeds.

Authentication. A user can tap a paired Key Card against the card reader on the driver's side door pillar to initiate authentication. Model 3 detects the Key Card and authenticates its legitimacy, and the door unlocks if successful. We found that the authentication protocol is identical to the steps of verification of a paired key (Steps 4 - 13) in the pairing process. Key Cards use a challenge-response authentication based on a shared secret generated by ECDH.

B. Phone Keys

Tesla allows users to use their iPhones or Android phones to unlock and start the car. To enable this function, smartphones must install an official Tesla mobile app, and Bluetooth must be on all the time. The Tesla app running in the background is responsible for managing the procedures of pairing and authentication. By analyzing Bluetooth snoop log and data parsing, we describe the mechanisms of Phone Keys as follows, including pairing and authentication.

Pairing. The pairing procedure is illustrated in Figure 4. The owner touches the "START" button on the phone screen to trigger pairing. The phone scans surroundings in range. The BLE names of Tesla Model 3 always start with the character "S" and end up with the "C". The middle sixteen characters are the first eight bytes of the hash result with an operation on the Vehicle Identification Number (VIN).

According to this specific name, the phone establishes a BLE connection to the vehicle. The app will send a GET_EPHEMERAL_PUBLIC_KEY request to the car by writing the characteristic value on the peer device (Step 7). As a GATT server, the vehicle will respond with its public key with indications of a characteristic (Step 8).

Next, to respond to the request for the whitelist, the vehicle needs to send back a WHITELIST_INFO data consisting of a list of Paired Key IDs, which are equal to the first four bytes of its Public Key's SHA-1 hash result (Step 9). The phone sends its public key and waits for the vehicle to authenticate a valid Key Card. Once finished, Tesla Model 3 adds the smartphone's public key to the whitelist and transmits the relevant data back to the phone. After pairing, the vehicle owns the public key of the Phone Key, and the smartphone records the public key and BLE MAC address of the car BD_ADDR. Adding a Phone Key can be treated as an exchange of public keys based on a valid key authentication.

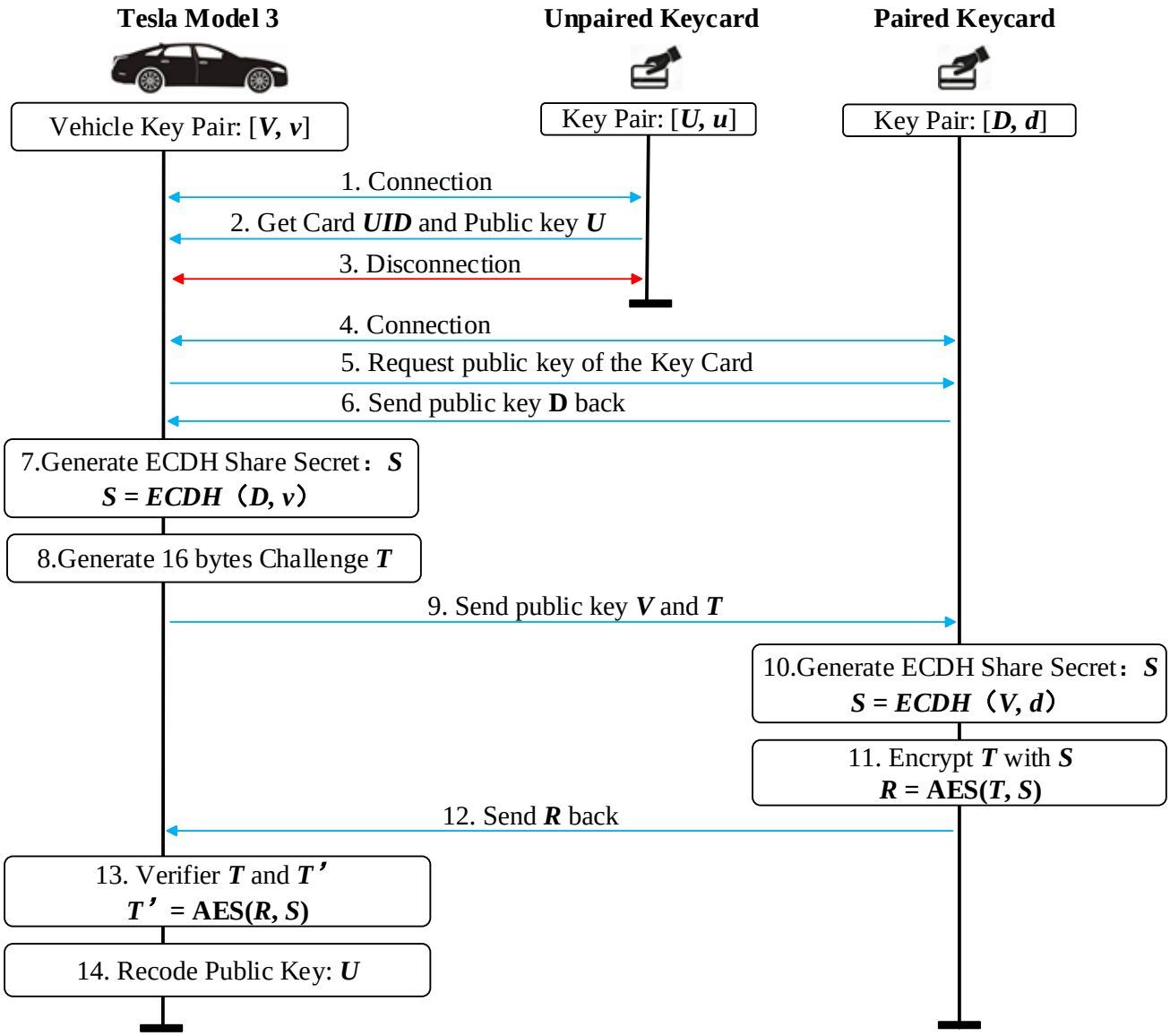

Fig. 3: Key Card Pairing and Authentication Protocols

Authentication. When a paired Phone Key approaches Model 3 from a distance out of the BLE range, the vehicle authenticates its legitimacy, and the door unlocks if successful. We sniffed the BLE channel for this process and reestablished the authentication protocol, as shown in Figure 5.

Phone Key attempts to establish a reconnection with a specific BLE MAC address. It adds the recorded BLE MAC address of Model 3 into the whitelist and sends a connection request that only allows this device to establish the connection. The BLE reconnection is established once the device is in the allowed range. Once the reconnection is created, the Phone Key receives the authentication level indication every second to trigger the authentication. The phone may ask for car status, for example, the lock state of the trunk(Steps 4 - 5). After getting the vehicle's status, the mobile app derives the shared secret by ECDH and generates a serialized java object defined in Appendix A. The app encrypts this Java object by AES encryption in GCM mode using a 4 bytes counter as GCM nonce and a share secret S generated by ECDH as an encryption key (Steps 6 - 7). After encryption, the app sends the first attestation, including the encrypted message (2 bytes ciphertext with 16 bytes GCM tag) and the counter to the vehicle. The format of attestation is shown in Figure 6. Vehicle decrypts and verifies message A with a shared secret S generated by ECDH and responds counter immediately. If verification successes, it continues to appends a 20 bytes token G on the authentication level and responds to the phone.

Afterward, the app increments the counter by one and refresh the Java object (details in Appendix A). After encrypting serialized results in AES-GCM mode involving token G as

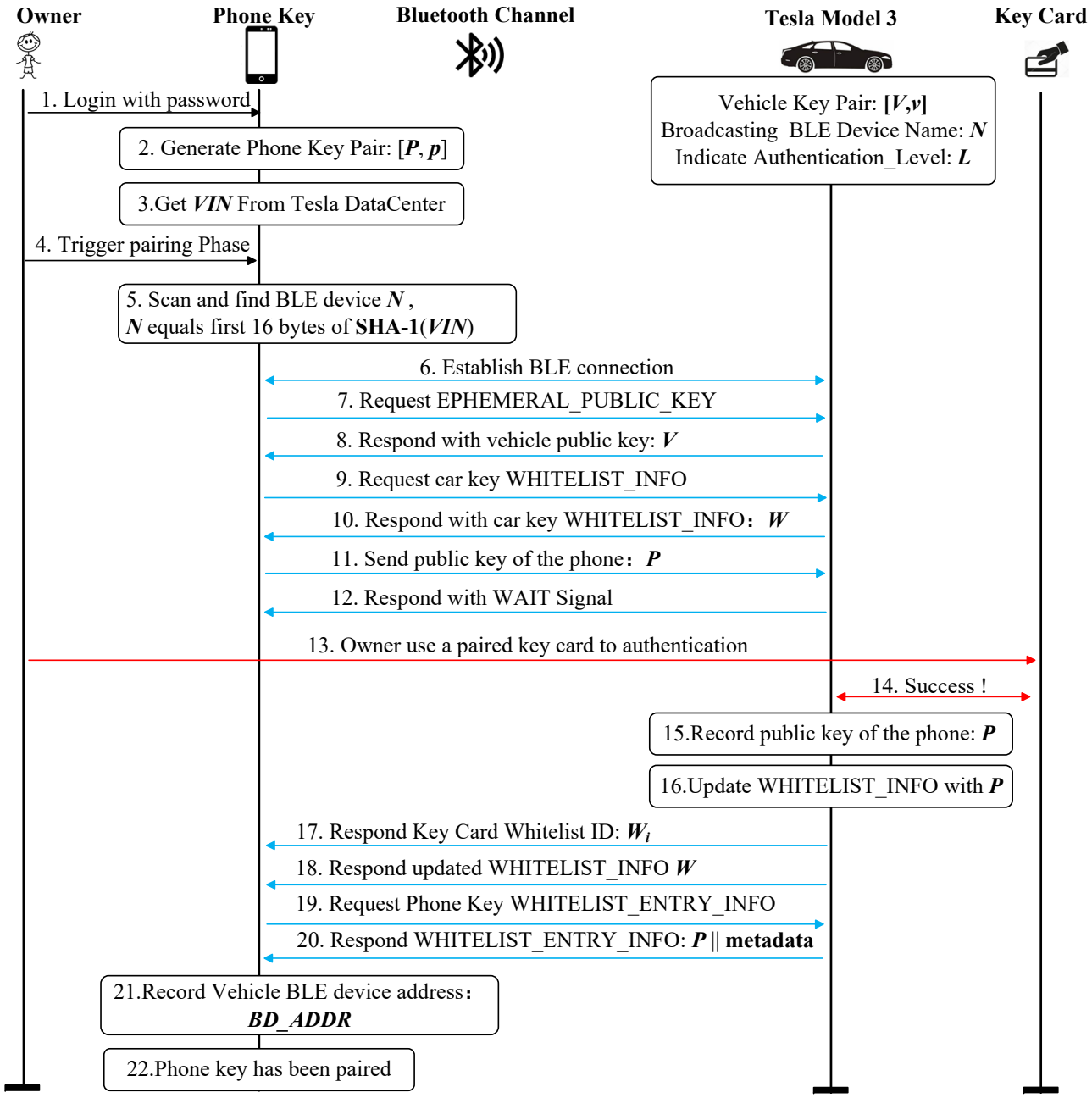

Fig. 4: The Pairing Protocol of a Phone Key

GCM additional authenticated data (Step 13), the app sends the second attestation package, including this encrypted message (4 bytes ciphertext with 16 bytes GCM tag) and the counter to the vehicle.

Finally, the vehicle decrypts and verifies message B with the shared secret S. If verification passes, the car will unlock and can be started.

The core of authentication is two attestations generated by AES-GCM. The shared secret derived from ECDH is directly used as the key to encrypt the message that is known to both sides. Tesla Model 3 uses this shared secret to check these two attestations (Step 9 & 14). Once these two messages pass the verification, the car asserts that the Phone Key is valid.

V. Security Analysis

In this section, we perform a security analysis of the Tesla keys. First, Tesla realizes key secure storage. Key Cards use a SE to manage the secret key. This highly-secure environment protects sensitive information at the hardware level from many known attacks, in particular side-channel attacks.

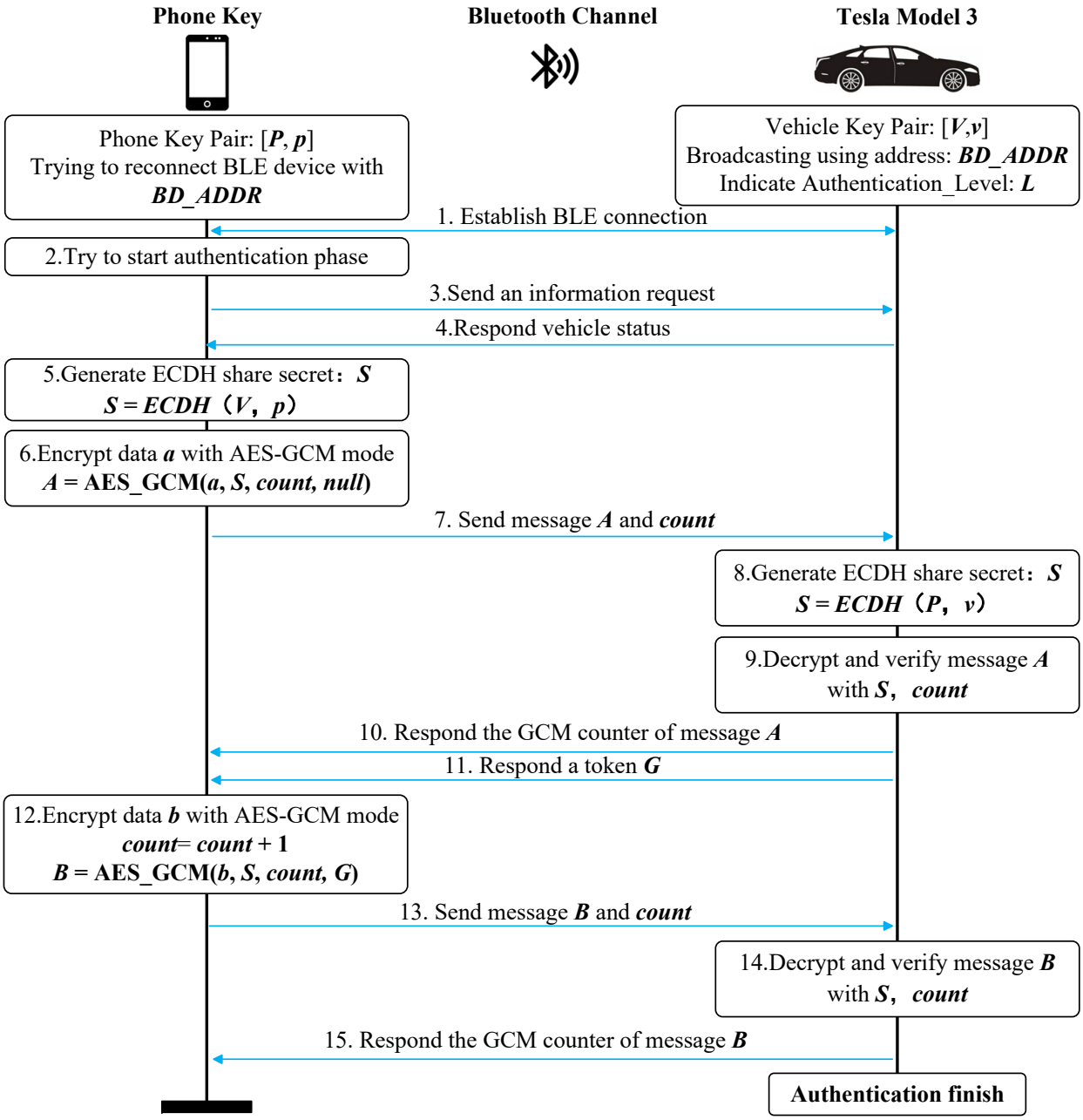

Fig. 5: The Authentication Protocol of a Phone Key

0x00	payload length	0x0a	counter	keyID	ciphertext	GCM tag

Fig. 6: Format of an attestation. The third byte "0x0a" marks this message as the attestation. The ciphertext length indicates this package belongs to the first or the second attestation.

Further, Tesla follows secure pairing and authentication protocols. During the pairing phase, the new Phone Key and the new card need a paired Key Card authentication process to ensure the ownership of the new key. The exchanged data contains the public key and related data without leaking privacy. During the authentication phase, Tesla and its keys perform an ECDH key exchange to derive a shared secret and validate the legitimacy of the keys. The key pair is generated based on the choice of the NIST P-256 curve and is securely stored. Without the private key, attackers cannot calculate the shared secret and generate proper attestations. Tesla pairing and authentication protocols utilize the cryptographic techniques of security algorithms to guarantee security. Further,

Phone Keys involve the counter by AES-GCM, which can resist replay attacks. However, it still has some potential issues.

The vehicle does not verify Key Card certificates during pairing. It might theoretically be possible to make an unofficial product to unlock the Model 3. Customers can purchase new Key Cards from third parties other than Tesla. They may record the provisioned value of the secret key or leave a backdoor. Once these unofficial Key Cards have paired, the authentication protocols guarantee security based on whether you have the private key used in pairing. In this case, you should always assume that whoever sold you Key Cards might also be able to access your car since they may keep or get the private key.

Tesla does not enable link layer encryption of the BLE. According to the sniffing results, the BLE channel is used to transfer data. Tesla leaves all cryptography in the upper layer. It makes the sniffer easier. Also, it offers adversaries opportunities to complete GATT-based BLE relay attacks.

Tesla vehicles use the static BLE MAC address. Anyone receiving its advertisement can spoof the car and establish the reconnection to the Phone Key at a different physical location with the same MAC address.

The update of token value does not depend on the change of connection states. The second attestation from the Phone Key is an AES-GCM result involving the token G. The token remains fixed even though the phone has been connected and disconnected multiple times. In our experiment, it stays fixed for hours.

VI. Key Card Threats

A. Adversary Model

The vehicle adds the new Key Card without verifying its certificates during pairing. It allows third parties products to work, which may introduce several security issues. Any card, which can generate a public-private key pair based on the NIST P-256 and support ECDH and AES operations, can be added as a new key. Indeed, any products that support ISO 14443 could work, but they must be able to complete related algorithms. Once these products have been paired, they can open and drive the Tesla Model 3 as official Key Cards. However, it could introduce serious security problems. Products from third parties may implement the algorithm using the software without a security coprocessor or processing unit. The software implementation is vulnerable to many known attacks. For example, side-channel attacks can easily crack the secret key. Even if the third party uses securely implemented Java cards, the applet may have a malicious backdoor allowing the reading of the key. In this case, you cannot trust any third-party product.

B. Proof-of-concept

We customized a Java card as shown in Figure 7. This card can generate a public-private key pair based on the NIST P-256 curve and do related cryptography. Also, we developed a Java applet to read the key through a non-public APDU command. This customized card was added to the vehicle's key list after using a paired Key Card to complete the pairing process. We found this card can use as an official Key Card to unlock and start Model 3. Since the key is leakage, this card can be easily duplicated.

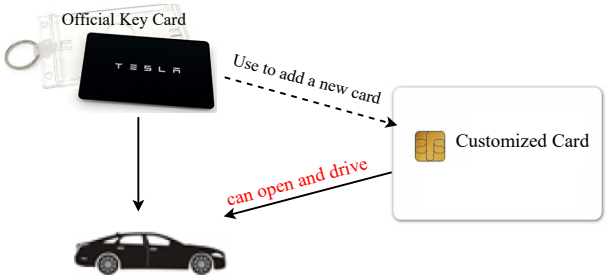

Fig. 7: Adversary model for the customized card.

C. Impact

This attack allows any unofficial product supporting ISO 14443 and corresponding cryptography operations to work as the official Key Card. Even if it is not a strict vulnerability, products from third parties will introduce serious problems. For example, third-party software allows attackers to control Tesla remotely[86]. Missing certificate verification gives opportunities to these unsecured third-party product developers.

VII. Improved Relay Attack

Although the current protocol is mature, it is still susceptible to relay attacks or MitM attacks. The basic relay is to retransmit the analog signal in real time. It needs complex hardware and has distance limits. However, the current protocol does not enable BLE encryption in the link layer. It makes it possible to relay the digital signals from the higher protocol layer, like the GATT layer. The known flaw of the GATT-relay attacks is that they introduce latency in communication. So developers can impose strict GATT response time limits to resist it. The delay of propagation between two devices may lead to failure. The mobile App has no time bound according to our reverse result. However, Tesla Model 3 may monitor the delay time. So latency is the most important influencing factor in GATT-relay attacks.

To eliminate the limitation of latency in relay attacks, we introduced an improved relay attack on the authentication process of Phone Keys. Since we reversed the protocol, we do not need to forward each message immediately. We record messages and transfer multiple messages once to avoid latency detection. In the following sections, we demonstrate this attack in detail.

A. Adversary Model

Suppose that the vehicle to be attacked is in a parking lot. Then the owner locks and leaves the car, then enters another open area out of the BLE range, like a library or a cafe. An attacker uses two malicious devices connecting to the Phone Key and Tesla Model 3 separately through the BLE channel. These two attack devices communicate with each other through another long-distance method like the internet, as shown in Figure 8 upper part. Unlike the real-time relay, we record the continuous messages during one connection period and only forward necessary messages to peer devices. In this case, the attack device will receive multiple data for one transfer. So it

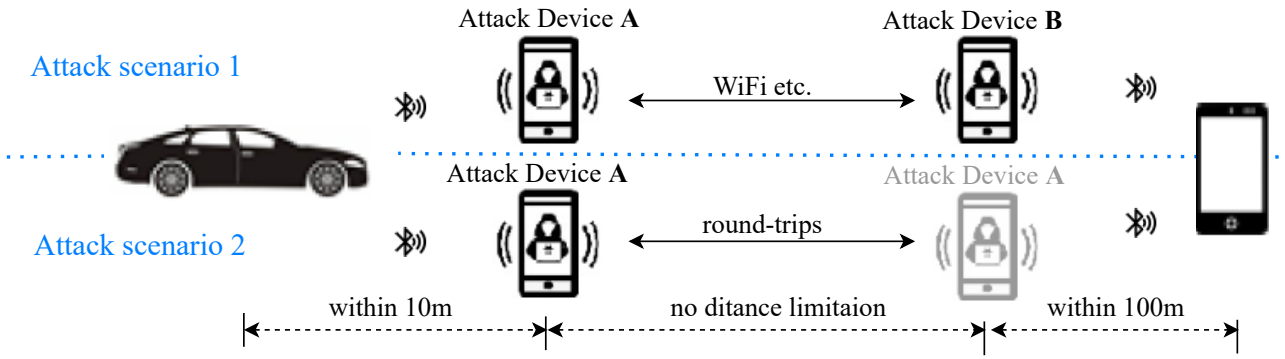

Fig. 8: Attack scenario descriptions. The upper part demonstrates two attack devices scenario. The lower shows one attack device case. In this case, the attacker has to go back and forth between the two parties.

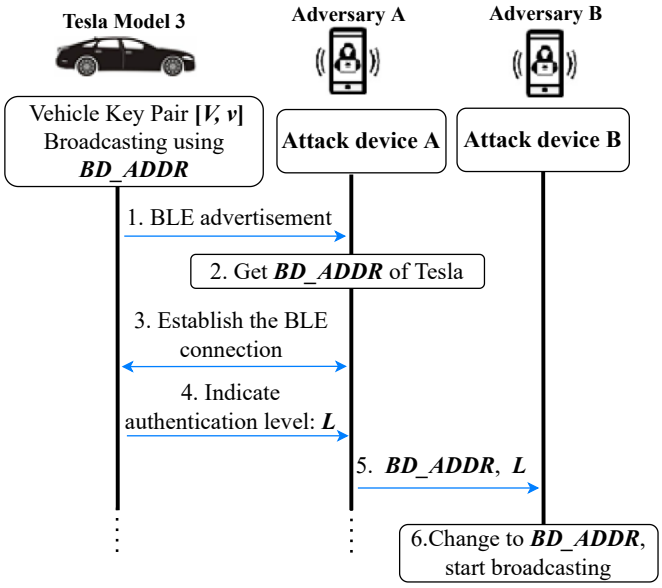

Fig. 9: Adversary Model Phase 1: Setup

can respond immediately. Time-bound of responses are not a defense against this improved type of relay attack.

The attack process can be divided into three phases: The setup phase, the preparation phase, and the attack phase. The setup phase is to complete BLE MAC spoofing. Then prepare two valid attestations during the second phase. Finally, during the attack phase, the attacker unlocks the car door and drives Model 3 away.

Setup: MAC spoof. Figure 9 illustrates the setup phase of the MitM attack. The setup phase is to capture the BLE MAC address from the vehicle advertisements and initial the BLE setting of attack devices. During an initial attack phase, attack device A is close to Model 3 to be attacked. Attack device A initiates a BLE scan to find its surroundings. By parsing the advertisement, the attacker can get the BLE MAC address, the BLE name, and additional data about the car. Once the BLE connection is established, the vehicle indicates the authentication level **L** every second to initiate the authentication. It needs to send the MAC address and the authentication level to attack device B. Then attack device B changes its MAC address and starts broadcasting connectable advertisements identical to the Model 3.

Preparation: attestations capture. Then the preparation phase (Figure 10) begins. Once attack device B has set the specific MAC address and approaches the Phone Key, an automatic connection establishes between the two parties. An authentication level indication is delivered from the attack device to the Phone Key (Step 8). After receiving, Phone Key will generate the shared secret using ECDH. Then the Phone Key sends the first attestation package, including the ciphertext, tag, and counter to the attack device B. So far, the attacker does not own the token value. Attack device B needs to transfer this first attestation to attack device A.

Then attack device A forwards this attestation. It waits for the response involving a token G after the vehicle has verified the first attestation successfully. This token G needs to be forwarded to attack device B. Once received the token G, attack device B retransmits the authentication level (Step 19) and replies with the first attestation with this token to the Phone Key. Attack device B can receive the second attestation from the Phone Key and transfer this pair of attestations to complete the attack (Step 26). For now, the preparation phase can be terminated if you want to open and start the Model 3 just once.

If you want to hijack multiple times, attack device B can get serval pairs of attestations by sending another authentication level and responding, as marked green in Figure 10. Then it can transfer serval pairs to device A. These pairs can be used to unlock the Model 3 multiple times later.

Attack: unlock and access. For the final attack phase shown in Figure 11, attack device A sends the first attestation in the first received pair. Once receiving the response with a token, the attacker delivers the second attestation in this pair. The Model 3 is hijacked now. According to our experiments, the token value does not update frequently. During the token fixed period, attack device A can unlocks doors and drive the car several times using the following attestation pairs.

Furthermore, the token update does not depend on the change of connection state. The token value remains the same

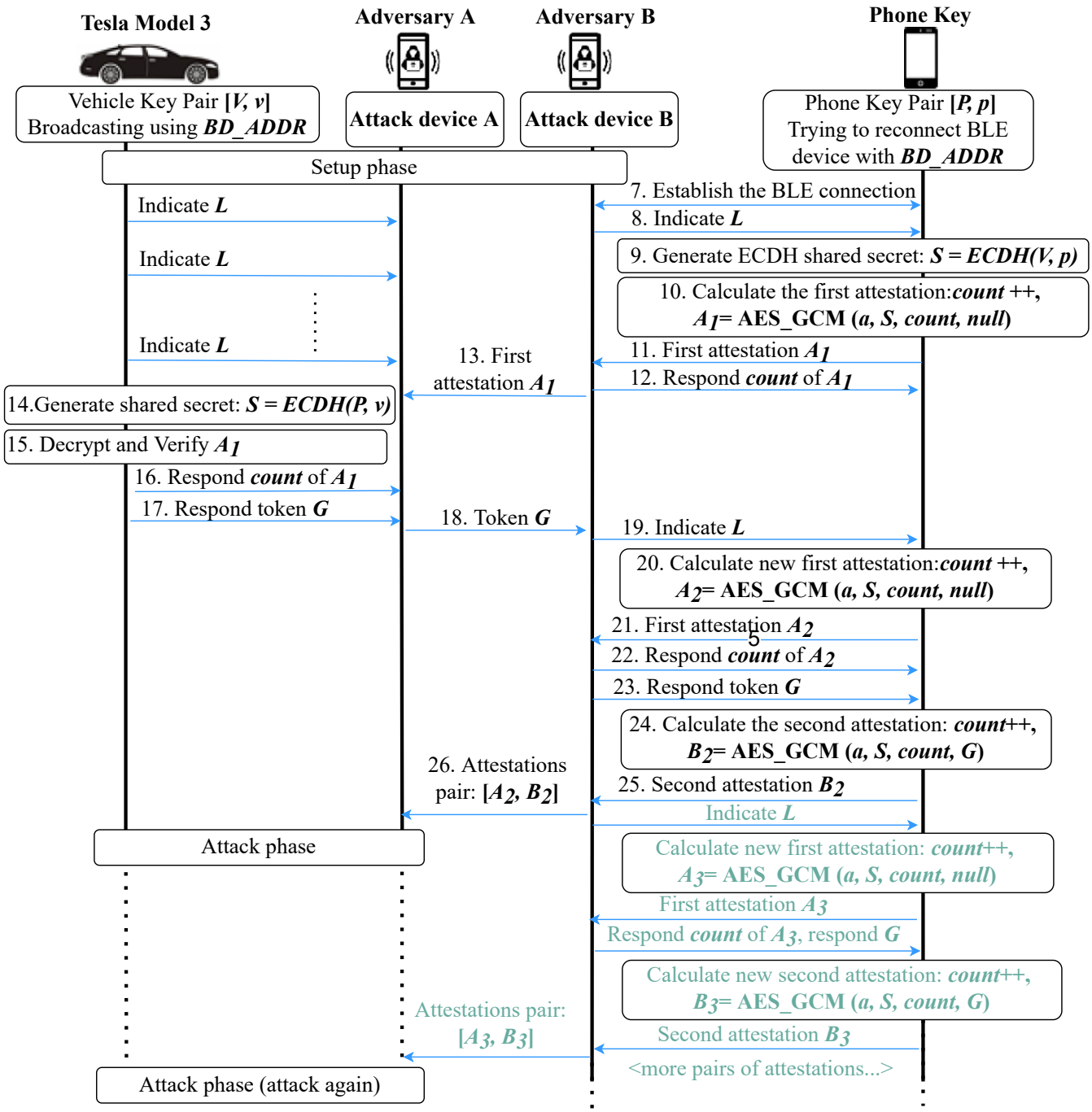

Fig. 10: Adversary Model Phase 2: Preparation

for hours. It gives a long enough attack window. So we can complete this MitM attack with only one attack device, as shown in Figure 8 scenario 2. The detailed process is demonstrated in Appendix Figure 10. Instead of transferring the messages between two attack devices, the attacker needs go back and forth between the vehicle and the Phone Key to connect and communicate to them alternately.

B. Proof-of-concept

In principle, it can be implemented with any off-the-shelf devices that can be programmed to receive and transmit BLE messages. In one attack device scenario, we chose an Android device Google Pixel 5A as the attack device. For the two-devices case, Google Pixel 5A as attack device B connected to the Phone Key, while the Samsung Galaxy S9 played the role of attack device A. This attack includes two key points, physical proximity to the intended target and malicious software on the attack device.

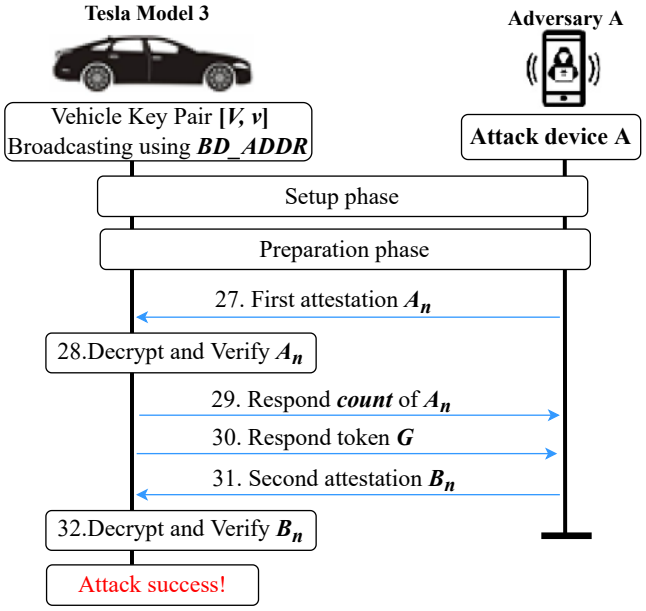

Fig. 11: Adversary Model Phase 3: Attack

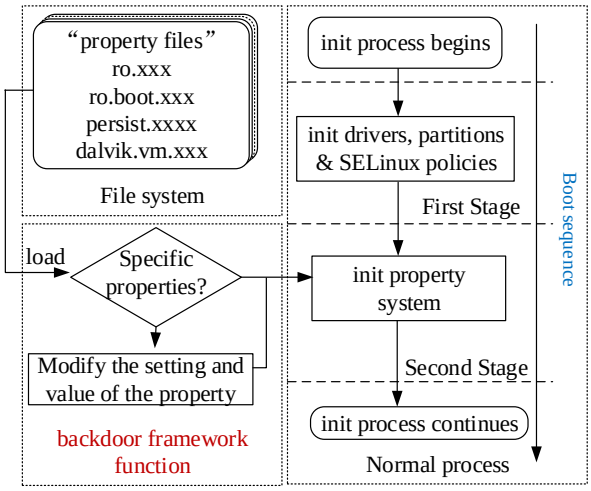

Fig. 12: The modified firmware of the attack device. It monitors the properties loading and changes the settings and value of the specific properties.

TABLE I: Experimental Models. In one attack device scenario, we only used the Google Pixel 5A as the attack device.

Devices	Model	OS version	Software Version
Attack device B	Google Pixel 5A	customized Android 11	TESmLA 2.0
Attack device A	Samsung Galaxy S9	Android 11	TESmLA 2.0
Phone Key	Motorola Edge S	Android 11	Tesla 4.23
	iPhone 12 Pro	iOS 15.4.1	Tesla 4.14.1
Vehicle	Model 3	v11.0(2022.4.5.1)	

BLE MAC Spoof. For physical proximity, the identification of BLE devices is the BLE MAC address. So we need to spoof Model 3 by setting the attack device with the same static MAC address. However, the Android device introduced a new feature called BLE MAC address rotation. It forces Android devices to broadcast with a resolvable address instead of a static address. The resolvable address changes every 15 minutes and cannot be recognized by the Phone Key without pairing.

We customized a specialized Android firmware for Google Pixel 5A to solve this problem. Following the instructions on Google Android source websites, we downloaded Android source codes and modified the Bluedroid definition "BLE LOCAL PRIVACY ENABLE" from True to False. This modification can disable the BLE device address rotation during advertising.

To understand Bluetooth peripheral initialization operation, we analyzed the log information of the boot process through the Android Debug Bridge (ADB). We found that a dynamic link library implemented by Google and compressed in the vendor image file is responsible for initializing the MAC address. We used the disassembly tool IDA Freeware to statically reverse this dynamic link library. As a result, the smartphone first loads a read-only attribute "ro.vendor.bt.bdaddr_path", which refers to the virtual file named "bt_addr" and uses the result as the Bluetooth MAC address. If the read fails, the phone will try to read the Bluetooth MAC address from alternative read-only Android attributes, "ro.vendor.bt.boot.macaddr" or "persist.vendor.service.bdroid.bdaddr". The BLE MAC address has been specified if successful. Otherwise, the smartphone will generate a random address as the Bluetooth MAC address. According to the analysis, we found that the Bluetooth address of the Pixel 5A is closely related to three read-only attributes of Android.

We proposed a novel framework, which can let the application layer software arbitrarily modify the specific MAC address. This new framework monitors the Android init process for the initialization of each Android attribute loading, as illustrated in Figure 12. It hooks the entry of property loading if the input attribute belongs to the three mentioned before. It forces the smartphone to read these attributes as null and change the property of "ro.vendor.bt.boot.macaddr" from a read-only type to a read-write attribute type. Consequently, the smartphone power on with a random MAC address, and the application layer program can use an ADB shell command to set a static specific address of Pixel 5A multiple times later.

Malicious Software In addition, We developed an app called TESmLA, which implements functions as the Phone Key using BluetoothGatt [3] and the Tesla Model 3 using BluetoothGattServer [4]. By sending writing requests to characteristic and receiving indications from the owner's device,

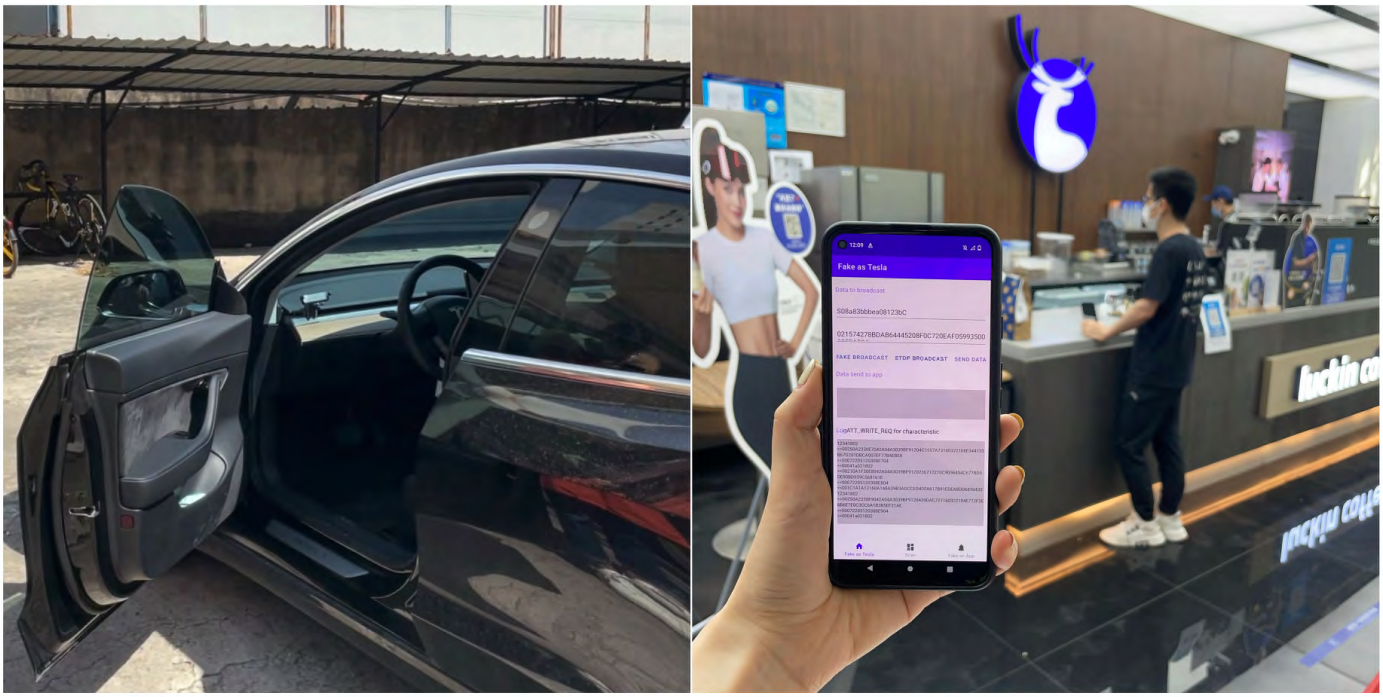

Fig. 13: Attack Results. The owner device maintained screen locked. The attacker approached it about 5 meters away. The owner device connected to the attack device automatically and transferred two attestations. Finally, attacker sends these two attestations to Model 3 to unlock the car.

the app can communicate with the car as the owner's device. As the role of the Tesla Model 3, it also can inform the owner device of indications and obtain the write request from the peer device. The app can parse receiving data byte by byte and respond automatically. By TESmLA, an adversary can complete the attack mentioned before even though they do not understand the cryptography and BLE schemes.

Experimental Evaluation Figure 13 shows the scene of the practical attack to demonstrate the feasibility of our attack. The owner is 500 meters away from The victim's Model 3. The owner's smartphone maintained the screen locked during the whole attack. The distance between the attacker and the owner is about 5 meters. The details of all devices in the experiments are demonstrated in Table I. When attack device A sent two attestations to the vehicle, we heard the unlocked voice of the vehicle. We pulled the car door and found it open. Also, we can start and drive this Model 3. The whole attack can complete within 15s. For one attack device scenario, we only use the Google Pixel 5A as the attack device. Due to the delay of multiple connections, BLE service requests, and operations on TESmLA, the time of the attack extends to 46s without the time of round-trips. The operation system of the Phone Key does not influence the result. Android and iOS are both attacked successfully.

Once the BLE connection is established between the attack device and the trusted parties, the attack will be successful. Due to locking off the car feature, the attack device has to be close to the vehicle enough. The distance limitation between the attack device and the Phone Key depends on many factors, for example, the version of the BLE. We tested the distance between Google Pixel 5A and Phone Key, including iPhone and Motorola Edge S. It can transfer data stably within 100 meters in the underground parking lot. The distance may reduced when you choose devices compatible below BLE 5.0. Besides, this distance would be influenced by inference of environment factors and other radio signals. However, there is no limitation on the distance between two attack devices. So the distance is not the limitation for our attack with two attack devices, while it would extend the time for back and forth in one attack device scenario. If the round-trip were time-consuming even over the token fixed time, the attack would not be successful.

C. Impact and Limitations

All test Tesla vehicles and devices are owned by the authors and their institutions. Due to the limitation of cost and other conditions, we only bought the Model 3 to conduct the practical attack. Model Y should be the same as Model 3. Model S may involve earlier PKES solutions. So it may not be affected by this attack. The presented attack is harmful to car owners who enable their PKES feature and keep their Bluetooth on all the time. The iOS and Android devices are both susceptible. We tested on Android 11 and iOS 15.4.1 and found no differences between these two operating systems.

This attack is easy to implement. The whole process is out of the owner's notice. The attack device, like a smartphone, can blend into common life. The process is conducted in the background, and the owner will not receive any notification from the Tesla mobile app or car screen.

Attackers can access the vehicle's interior and even drive it away. However, there is a time limit on this access. During the

token fixed period, the adversaries can capture multiple pairs of attestations in one connection. These pairs of attestations allow the attacker to access and drive the victim's vehicle serval times. In the scenario of two attack devices, the token update only influences the malicious control period. But in one attack device scenario, due to the long latency of round-trips, the attacker may need more back and forth to capture and transfer the new pair of attestation when the token changes. It may prolong the attack time but not affect the final result.

As a relay attack is a known technique, Tesla advises the user to enable the "PIN to Drive" feature. It forces driver to input four digital numbers into the PIN panel on the vehicle screen to start the vehicle. This feature can protect vehicle from driving but not entering. According to a vote from Tesla owners [82], there are still about 20% of owners have not enabled this feature. So this is still a significant security problem in the real world.

D. Related Works

Relay attacks are a known limitation of the passive entry system. People used two amplifiers close to the trust parties to amplify analog signals. So adversaries can use simple hardware to achieve ultra-low latency with unmodified commnication patterns. For example, P. Staat, etc. [68] design and implement an analog physical-layer relay to amplify the analog signal of the entire 2.4GHz. However, it has a limited relay distance of about 90m and is complicated to handle bi-direction communications on the same frequency. Moreover, this attack can be detected by RSSI. So it has been improved to retransmit the analog signal by a more complex hardware instead of simply amplification. It has the same drawbacks as the former amplification but can circumvent the RSSI and arrival of angle (AoA) triangulation defenses.

Rather than raw analog signals, it can forward data frames at any higher layer of the protocol stack. The first GATT-relay in 2016 [17], [41] forwards GATT requests and responses. The attack is most effective against devices that do not implement Bluetooth security features. However, researches [6], [7], [8], [10] are working on the vulnerabilities of these security features to make the impersonation attack available.

Mirage [18] can transmit HCI messages from standard controllers bypassing standard Bluetooth stacks. It can perform experimental Bluetooth Low Energy attacks using a ButteRFly device (nRF52840 dongle). This new device allows to inject packets into an established connection, hijack the slave role, hijack the master role or perform a Man-in-the-Middle attack [51].

The NCC Group [72] introduces Sniffle Relay, the first link layer relay attack on BLE. They forward the response and add the 8ms latency in the link layer. Since this relay attack operates at the link layer, it can forward encrypted link layer packet data unit (PDU). Neither link layer encryption nor encrypted connection parameter changes (such as changes to the channel map, connection interval, and transmit window offset) are defenses against relay attacks from the link layer [71].

Relay attacks from link layer-relay need to customize the BLE stack. The attacker must be familiar with the details of BLE protocols. Like NCC Group customizes the BLE controller that responds an additional empty PDU to maintain the connection and retransmit the PDU in the link layer. On the contrary, relaying from the GATT layer is much easier. You do not need to care about the timing sequence of data frames. However, It can use time-bound to detect the GATT relay due to long latency. Also, the GATT-relay attack does not support link layer encryption. Our attack is from the GATT layer. Like other GATT-relay, we do not need operations on the baseband and do not need to modify the BLE stack. Also, our attack cannot circumvent the link layer encryption.

However, our improved attack can break the latency detection. Unlike existing relay attacks, our attack is not a real-time relay. It can operate on multiple messages to break the latency bounding and distance limitations by spoofing the trusted party. Besides, it can complete by using only one attack device. And it is not a one-time hijack. Although it is temporary access, the attacker can record multiple pairs of two attestations once. As the token has not been updated, the attacker can access the vehicle serval times by sending these pairs in the sequence of the counter.

VIII. Disscusion

A. Countermeasures

PIN to drive. It allows owners to program a personal identification number. This feature forces the owner to enter these numbers into the screen to drive the car. This multi-factor authentication countermeasure only forbids driving but not entering, as discussed in the previous section. It is worth noting that this feature disobeys the intention of PKES and is not the default setting of Model 3.

Refresh the token frequently. Our results indicate the token value remains fixed in hours. In one device scenario, the attack device needs alternating connect to the vehicle and the Phone Key. If the token changes after recording but before reconnecting to the car, authentication of the second encrypted command on the vehicle's side will fail. If the token updates every time Model 3 establishes a BLE connection, the adversary has to use two attack devices. To a certain degree, refreshing the token fast enough will reduce the attack window.

Enable BLE link layer encryption. BLE supports secure communication according to Bluetooth standards. End devices can negotiate a long-term key during the pairing process and derive a session key for each reconnection. The BLE encrypts the communication between devices by this session key. Enabling BLE encryption will improve the difficulty of analysis and device spoofing. However, it is circumvented by NCC Group, as mentioned in previous related works.

TOF based secure ranging. The PKES system can employ the Time of Flight (TOF) to avoid MitM attacks. Two trust parties can record the timestamp when receiving the messages. Timing is often implemented in the physical layer to eliminate the delay of intra-transmission and increase accuracy. Many vehicle manufacturers [14], [56] announced that UWB technique would be a new solution of PKES. UWB utilizes the TOF technique to measure the distance. However, the spoof messages of measurement or synchronization may modify the propagation time. So the messages require encryption or

signature by a trusted module like SE. Tesla is in the process of combining a new Ultra-Wideband (UWB) technique in its key fobs[76]. Mobile phone manufacturers like Apple[9] and Samsung [57] have already integrated UWB into their mobile devices. It provides simplicity for Tesla to ensemble it directly into a mobile app.

B. Future Directions

Many vehicle manufacturers have announced that UWB technique defined in physical-level standard IEEE 802.15.4z [38], [39] will be a new security solution in the PKES system. UWB has many advantages, like high accuracy, low latency, long distance and high security. The scrambled timestamp sequence (STS) contains a pseudo-random bit sequence for security purposes. Ranging devices can utilize the TOF on the arrival time of the STS, thereby guaranteeing that no external adversary reduces the measured distance by advancing the received signal in time. However, propagation time responded by the receiver should be protected because it determines the distance. The up-level standard [16] recommends encrypting the replied time with a symmetric secret. However, many works [46], [60], [64], [65] have revealed the weaknesses of UWB. The implementation security of UWB in the PKES system will be explored in future works.

IX. Conclusion

Tesla Model 3 promotes new types of Key Cards and Phone Keys for better driver experiences. With reverse-engineering of the protocols and codes, we demonstrate pairing and authentication procedures of these two keys in detail. We find that communication channel like BLE does not employ any security mechanisms. Tesla decides to leave security to up-level cryptography. We demonstrate several design vulnerabilities and implementation flaws. The current protocols allow customized Key Cards to work. Besides, we introduced an improved relay attack against authentication of the Phone Key through the BLE channel. In addition, we developed an app named TESmLA installed on a customized Android device to implement the attack. The attack is performed in the background without the awareness of the car owner and can be extended with any other cheap off-the-shelf device. Our results raise attentions of the insecure implementation of BLE communication since it may be a considerable concern in many other PEPS systems. Our findings have implications for other vehicles supporting Phone Keys. The security analysis of more vehicle manufacturers, even the new technique UWB, leave for pending further investigation.

Responsible Disclosure

We submitted the correlated reverse-engineering results and informed vulnerabilities to Tesla on March 15, 2022. We also indicated the intention to publish research regarding the protocols and BLE relay attacks. Tesla still has not responded to us yet. After over half a year, we decided to disclose and assign CVE-2022-37709.

Acknowledgement

We thank the anonymous reviewers for their constructive and helpful comments and feedback. We also thank Sultan Qasim Khan from the NCC Group for sharing their contributions to BLE Sniffle Relay. We greatly appreciate Ivon, who offers his vehicle for experiments.

References

[1] A. I. Alrabady and S. M. Mahmud, "Analysis of Attacks Against the Security of Keyless-Entry Systems for Vehicles and Suggestions for Improved Designs", *IEEE Transactions on Vehicular Technology*, 54(1), 2005, pp. 41 - 50.

[2] A. I. Alrabady and S. M. Mahmud, "Some attacks against vehicles' passive entry security systems and their solutions", *IEEE Transactions on Vehicular Technology*, 52, 2003, pp. 431 - 439.

[3] Android Developers Reference: BluetoothGatt. [Online]. Available: https://developer.android.google.cn/reference/kotlin/android/bluetooth/BluetoothGatt

[4] Android Developers Reference: BluetoothGattServer. [Online]. Available: https://developer.android.google.cn/reference/kotlin/android/bluetooth/BluetoothGattServer

[5] A. Ansari, P. C. Karthik, D. H. Sharath, M. Aziz, and S. Mukherjee, "Mechanism to Identify Legitimate Vehicle User in Remote Keyless Entry System", *WCX SAE World Congress Experience*, 2022.

[6] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "BIAS: Bluetooth Impersonation AttackS", *In Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2020.

[7] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy", *Transactions on Privacy and Security (TOPS)*, 2020.

[8] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR", *In Proceedings of the USENIX Security Symposium (USENIX Security)*, August 2019.

[9] Apple Inc. Ultra Wideband security in iOS. [Online]. Available: https://support.apple.com/guide/security/ultra-widebandsecurity-sec1e6108efd/web, accessed October 8, 2021.

[10] K. Arai and T. Kaneko, "Formal Verification of Improved Numeric Comparison Protocol for Secure Simple Pairing in Bluetooth Using ProVerif", *In Proceedings of the International Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, 2014.

[11] S. V. D. Beek and F. Leferink, "Vulnerability of Remote Keyless-Entry Systems Against Pulsed Electromagnetic Interference and Possible Improvements", *IEEE Transactions on Electromagnetic Compatibility*, 58(4), 2016, pp. 1-7.

[12] E. Biham and L. Neumann, "Breaking the Bluetooth Pairing–The Fixed Coordinate Invalid Curve Attack", *In International Conference on Selected Areas in Cryptography. Springer*, 2019.

[13] Bluetooth Special Interest Group. Core Specifications 5.3. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-3-feature-enhancements/, accessed 2021.

[14] BMW announces BMW Digital Key Plus with Ultra-Wideband technology coming to the BMW iX. [Online]. Available: https://www.press.bmwgroup.com/global/article/detail/T0324128EN/bmw-announces-bmw-digital-key-plus-withultra-wideband-technology-coming-to-the-bmw-ix, accessed October 8, 2021.

[15] S.C. Bono, M. Green, A. Stubblefield, A. Juels, and M. Szydlo, "Security Analysis of a Cryptographically-Enabled RFID Device", *In USENIX Security Symposium*, vol. 31, 2005, pp. 1–16.

[16] Car Connectivity Consortium. Digital Key Release. [Online]. Available: https://carconnectivity.org/press-release/car-connectivityconsortium-publishes-digital-key-release-3-0/, accessed October 11, 2021.

[17] D. Cauquil, "BtleJuice Framework". [Online]. Available: https://github.com/DigitalSecurity/btlejuice, accesssed 2016.

[18] R. Cayre, "Mirage", [Online]. Available: https://homepages.laas.fr/rcayre/mirage-documentation/index.html, accessed 2019.

[19] C˘etin Kaya Ko˘c. Elliptic Curve Cryptography Fundamentals. [Online]. Available: https://cs.ucsb.edu/koc/ecc/docx/09ecc.pdf, accesed 21 October 2015.

[20] W. Choi, M. Seo, and D.J. Lee, "Sound-Proximity: 2-factor Authentication against Relay Attack on Passive Keyless Entry and Start System", *Journal of Advanced Transportation*, 2018(PT.1), pp.1-13.

[21] CVE-2020-15912. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-15912

[22] Q. Dang, "Secure Hash Standard", *Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology*, Gaithersburg, MD, 2015. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.180-4, accessed June 30, 2022.

[23] Diffie–Hellman Key Exchange. [Online]. Available: https://en.wikipedia.org/wiki/Diffie, accessed 2015

[24] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced Encryption Standard (AES)", *Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD*. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197, accessed June 30, 2022.

[25] M. Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) for Confidentiality and Authentication. *NIST*. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/ocb/gcm.pdf, accessed April, 2006

[26] Elliptic Curve Diffie-Hellman. [Online]. Available: https://en.wikipedia.org/wiki/EllipticcurveDiffie

[27] A. Francillon, B. Danev, and S. Capkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars", *In Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 6–9 February 2011.

[28] Frida, Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers. [Online]. https://frida.re/

[29] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock It and Still Lose It - on the (In)Security of Automotive Remote Keyless Entry Systems", *In Thorsten Holz and Stefan Savage, editors, 25th USENIX Security Symposium, USENIX Security 16*, Austin, TX, USA, August 10-12, 2016, pp. 929–944. USENIX Association, 2016.

[30] K. Greene, D. Rodgers, H. Dykhuizen, Q. Niyaz, and K. A. Shamaileh, "A Defense Mechanism Against Replay Attack in Remote Keyless Entry Systems Using Timestamping and XOR Logic", *IEEE Consumer Electronics Magazine*, 10. 1-1, 10.1109/MCE.2020.3012425.

[31] K. Haataja and P. Toivanen, "Two Practical Man-in-the-middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures", *Transactions on Wireless Communications*, 9(1), 2010, pp. 384–392.

[32] G.P. Hancke and M.G. Kuhn, "An RFID Distance Bounding Protocol", *In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, Athens, Greece, 5–9 September 2005; pp. 67–73.

[33] J. V. D. Herrewegen and F. D. Garcia, "Beneath the Bonnet: A Breakdown of Diagnostic Security", *In Javier López, Jianying Zhou, and Miguel Soriano, editors, Computer Security - 23rd European Symposium on Research in Computer Security*, ESORICS 2018, Barcelona, Spain, September3-7, 2018, Proceedings, Part I, vol. 11098 of Lecture Notes in Computer Science, pages 305–324. Springer, 2018.

[34] C. Hicks, F. Garcia, and D. Oswald, "Dismantling the AUT64 Automotive Cipher", *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2), May 2018, pp.46–69.

[35] K. Hypponen and K. M. Haataja, "Nino: Man-in-the-middle Attack on Bluetooth Secure Simple Pairing", *In Proceedings of the International Conference in Central Asia on Internet*, IEEE 2007, pp. 1–5.

[36] O. Ibrahim, A. M. Hussain, G. Oligeri and R. Pietro, "Key is in the Air: Hacking Remote Keyless Entry Systems", *Security and Safety Interplay of Intelligent Software Systems*, 2019, pp. 125-132.

[37] IDA Freeware. [Online]. Available: https://hex-rays.com/ida-free/

[38] IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, 2020.

[39] IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, 2020, pp. 1–174.

[40] JADX. [Online]. Available: https://github.com/skylot/jadx

[41] S. Jasek, "GATTacker", [Online]. Available: https://github.com/securing/gattacker, accessed 2016.

[42] R. Karani, S. Dhote, N. Khanduri, A. Srinivasan, R. Sawant, G. Gore, and J. Joshi, "Implementation and Design Issues for Using Bluetooth Low Energy in Passive Keyless Entry Systems", *In Proceedings of the 2016 IEEE Annual India Conference (INDICON)*, Bangalore, India, 16–18 December 2016; pp. 1–6.

[43] H. Khalid, S. J. Hashim, S. M. S. Ahmad, f. Hashim and M. A. Chaudhary, Muhammad, *New and Simple Offline Authentication Approach using Time-based One-time Password with Biometric for Car Sharing Vehicles*, 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020.

[44] N. Khatri, "Amplification Attack-Resistant Authentication Scheme for Remote Keyless Entry System", *2020 Fall Conference of the Korean Embedded Engineering Society*, 2020.

[45] C. H. Kim and G. Avoine, "RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks", *In Proceedings of the International Conference on Cryptology and Network Security*, Kanazawa, Japan, 12–14 December 2009, pp. 119–133.

[46] P. Leu, G. Camurati, A. Heinrich, M. Roeschlin, C. Anliker, M. Hollock, S. Capkun, J. Classen, "Ghost Peak: Practical Distance Reduction Attacks Against HRP UWB Ranging", *In Proceedings of the USENIX Security Symposium (USENIX Security)*, 2022.

[47] A. Levi, E. Çetinta¸s, M. Aydos, C. K. Koç, and M. U. Ça˘glayan, "Relay Attacks on Bluetooth Authentication and Solutions", *In Proceedings of the International Symposium on Computer and Information Sciences*, Kemer, Antalya, Turkey, 27–29 October 2004; pp. 278–288.

[48] J. Li, Y. Dong, S. Fang, H. Zhang, and D. Xu, "User Context Detection for Relay Attack Resistance in Passive Keyless Entry and Start System", *Sensors*, 20(16): 4446 (2020).

[49] J.R. Lin, T. Talty, and O.K. Tonguz, "On the Potential of Bluetooth Low Energy Technology for Vehicular Applications", *IEEE Commun. Mag.* 2015, 53, pp. 267–275.

[50] T. Melamed. "An active man-in-the-middle attack on Bluetooth smart devices". *International Journal of Safety and Security Engineering*, 8(2), 2018.

[51] MIRAGE -1.2 [Online]. Available: https://github.com/RCayre/mirage

[52] Model 3 Owner's Manual. [Online]. Available: https://www.tesla.com/ownersmanual/model3/en_us/GUID-E004FAB7-1C71-448F-9492-CACF301304D2.html, accessed 14 April 2022.

[53] MP300 ACL1 Contactless Spy Tool. Available: https://micropross.ni.com/products/range/mp300-acl1/

[54] New Tesla Key Card Vulnerability Lets Hackers Silently Steal Your Ride. [Online]. Available: https://www.reviewgeek.com/120570/new-tesla-key-card-vulnerability-lets-hackers-silently-steal-your-ride/, accessed 9 June 2022.

[55] NomadLAB. [Online]. Available: https://www.keolabs.com/products/services-accessories/nomad-tester

[56] NXP Announces New Automotive Ultra-Wideband Chip Capable of Turning Smartphones into Car Keys. [Online]. Available: https://www.nxp.com/company/about-nxp/nxpannounces-new-automotive-ultra-wideband-chip-capableof-turning-smartphones-into-car-keys:NW-AUTOMOTIVEULTRA-WIDEBAND, accessed October 12, 2021.

[57] NXP Secure UWB deployed in Samsung Galaxy Note20 Ultra Bringing the First UWB-Enabled Android Device to Market. [Online]. Available: https://www.nxp.com/company/about-nxp/nxpsecure-uwb-deployed-in-samsung-galaxy-note20-ultrabringing-the-first-uwb-enabled-android-device-tomarket:NW-SECURE-UWB-SAMSUNG-GALAXY, accessed October 5, 2021.

[58] J. Patel, M. L. Das, and S. Nandi, "On the Security of Remote Key Less Entry for Vehicles", *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2018.

[59] M. Poturalski, M. Flury, P. Papadimitratos, J.Hubaux, and J. Le Boudec, "Distance Bounding with IEEE802.15.4a: Attacks and Countermeasures", *IEEE Trans. Wirel. Commun.*, 10(4), 2011, pp.1334–1344.

[60] M. Poturalski, M. Flury, P. Papadimitratos, J.Hubaux, and J. Le Boudec, "The Cicada Attack: Degradation and Denial of Service in IR Ranging", *In 2010 IEEE International Conference on Ultra-Wideband*, vol.2, 2010, pp.1-4.

[61] Protocol Buffers - Google's data interchange format. [Online]. Available: https://github.com/protocolbuffers/protobuf

[62] J. Reid, J.M.G. Nieto, T. Tang, and B. Senadji, "Detecting Relay Attacks with Timing-based Protocols", *In Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, Singapore, 20–22 March 2007, pp. 204–213.

[63] S. Rizvi, J. Imler, L. Ritchey L, and M. Tokar, "Securing PKES against Relay Attacks using Coordinate Tracing and Multi-Factor Authentication", *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, 2019.

[64] M. Singh, P. Leu, and S. Capkun, "UWB with pulse reordering: Securing ranging against relay and physical-layer attacks", *In 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 2019*, pp. 24-27.

[65] M. Singh, M. Roeschlin, E. Zalzala, P. Leu, and S. Capkun, "Security analysis of IEEE 802.15.4z/HRP UWB time-offlight distance measurement". *In WiSec '21: 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Abu Dhabi, The United Arab Emirates*, 28 June - 2 July 2021, pp. 227–237.

[66] P. Sivakumaran and J. Blasco, "A Study of the Feasibility of Co-located App Attacks against BLE and a Large-scale Analysis of the Current Application-layer Security Landscape", *In Proceedings of the USENIX Security Symposium (USENIX Security)*, August 2019.

[67] D. Spill and A. Bittau, "BlueSniff: Eve Meets Alice and Bluetooth", *In Proceedings of USENIX Workshop on Offensive Technologies (WOOT)*, vol. 7, 2007, pp. 1–10.

[68] P. Staat, K. Jansen, C. Zenger, H. Elders-Boll and C. Paar, "Analog Physical-Layer Relay Attacks with Application to Bluetooth and Phase-Based Ranging", *15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2022.

[69] D. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle Attacks on Secure Simple Pairing in Bluetooth Standard v5. 0 and Its Countermeasure", *Personal and Ubiquitous Computing*, 22(1), 2018, pp. 55–67.

[70] Y. Tao, L. Kong, X. Wei, J. Hu, and C. Zhong, "Resisting Relay Attacks on Vehicular Passive Keyless Entry and start systems", *International Conference on Fuzzy Systems & Knowledge Discovery*, IEEE 2012.

[71] Technical Advisory – Tesla BLE Phone-as-a-Key Passive Entry Vulnerable to Relay Attacks. [Online]. Available: https://research.nccgroup.com/2022/05/15/technical-advisory-tesla-ble-phone-as-a-key-passive-entry-vulnerable-to-relay-attacks/, accessed 15 May 2022.

[72] Technical Advisory – BLE Proximity Authentication Vulnerable to Relay Attacks. [Online]. Available: https://research.nccgroup.com/2022/05/15/technical-advisory-ble-proximity-authentication-vulnerable-to-relay-attacks/, accessed 15 May 2022.

[73] Teen hacker says he's found way to remotely control 25 Tesla EVs around the world. [Online]. Available: https://fortune.com/2022/01/12/teen-hacker-david-colombo-took-control-25-tesla-ev/, accessed 12 January 2022.

[74] Tesla Bluetooth Hack Opens Doors and Start Cars: NCC Group. [Online]. Available: https://researchsnipers.com/tesla-bluetooth-hack-opens-doors-and-start-cars-ncc-group/, accessed 17 May 2022.

[75] Tesla warns of theft risk through relay attacks, and shares 'tips' to help prevent. [Online]. Available: https://electrek.co/2018/07/31/tesla-thefttips-help-prevent-relay-attacks/amp/, accessed 9 October 2021.

[76] Tesla's next car will seamlessly unlock with UWB, FCC leak suggests. [Online]. Available: https://www.theverge.com/2021/2/2/22262996/tesla-uwb-fcc-car-key-ultrawideband-tech, accessed 3 February 2022.

[77] TESmLA. [Online]. Available: https://github.com/fmsh-seclab/TesMla, accessed September 2022

[78] This Bluetooth Attack Can Steal a Tesla Model X in Minutes. [Online]. Available: https://www.wired.com/story/tesla-model-x-hack-bluetooth/, accessed 23 Nov. 2020.

[79] A. Valko, "Relay Attack Resistant Passive Keyless Entry: Securing PKE Systems with Immobility Detection", *thesis for Bachelor of Science*, 2020.

[80] R. Verdult, F. D. Garcia, and B. Ege, "Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer", *In Samuel T. King, editor, Proceedings of the 22nd USENIX Security Symposium*, Washington, DC, USA, August 14-16, 2013, pp. 703–718. USENIX Association, 2015.

[81] Volvo Cars Tests Replacing Keys with Smart Phone App. [Online]. Available: https://www.media.volvocars.com/us/en-us/media/pressreleases/173880/volvo-cars-tests-replacing-keys-with-smart-phone-app, accessed 10 June 2020.

[82] Vote for "you using PIN to drive?" [Online]. Available: https://teslamotorsclub.com/tmc/threads/pin-to-drive.182715/, accessed 22 January 2020.

[83] J. Wang, K. Lounis, and M. Zulkernine, "CSKES: A Context-based Secure Keyless Entry System" *In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Milwaukee, WI, USA, 15–19 July 2019, vol. 1, pp. 817–822.

[84] J. Wang, K. Lounis and M. Zulkernine, "Security Features for Proximity Verification", *In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Milwaukee, WI, USA, 15–19 July 2019, vol. 2, pp. 592–597.

[85] Will Your Smartphone Replace Your Car Key. [Online]. Available: https://www.consumerreports.org/automotivetechnology/will-your-smartphone-replace-your-car-key-virtual-key/, , accessed 10 June 2020.

[86] L. Wouters, B. Gierlichs, B. Preneel, "My Other Car is Your Car: Compromising the Tesla Model X Keyless Entry System", *IACR Trans. Cryptographic Hardware and Embedded Systems- CHES 2021*, vol. 2, pp. 149-172.

[87] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel, "Fast, Furious and Insecure: Passive Keyless Entry and Start Systems in Modern Supercars", *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3), 2019, pp. 66–85.

[88] L. Wouters, J. Van den Herrewegen, F. D Garcia, D. Oswald, B. Gierlichs, and B. Preneel, "Dismantling DST80-based Immobiliser Systems", *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(2), pp. 99–127.

[89] J. Wu, Y. Nan, V. Kumar, D. Tian, and A. Bianchi, "BLESA: Spoofing Attacks against Reconnections in Bluetooth Low Energy", *USENIX WOOT*, 2020.

[90] F. Xu, W. Diao, Z. Li, J. Chen, and K. Zhang, "BadBluetooth: Breaking Android Security Mechanisms via Malicious Bluetooth Peripherals", *In Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

APPENDIX A
ATTESTATION CALCULATION

The first attestation calculates the encryption of an "Authentication Response" Java object with its "Authentication-Level" field set to null, as shown in Listing 1. Serialize this Java object to a 2 bytes hex string with Google-Protobuf framework [61]. Then use AES-GCM to get the encryption result.

```
UnsignedMessage
{
  AuthenticationResponse: AuthenticationResponse
  {
    authenticationLevel: AUTHENTICATION_LEVEL_NONE
  }
}
```

Listing 1: Java object to be serialized for the first attestation

For the second attestation, "Authentication Response" Java object with its "Authentication-Level" field is set to the value received. Normally, if the car owner enables the PKES feature, the "Authentication-Level" field of the "Authentication Response" object will be probably set to AUTHENTICATION_LEVEL_DRIVE.

```
UnsignedMessage
{
  AuthenticationResponse: AuthenticationResponse
  {
    authenticationLevel: AUTHENTICATION_LEVEL_DRIVE
  }
}
```

Listing 2: Java object to be serialized for the second attestation

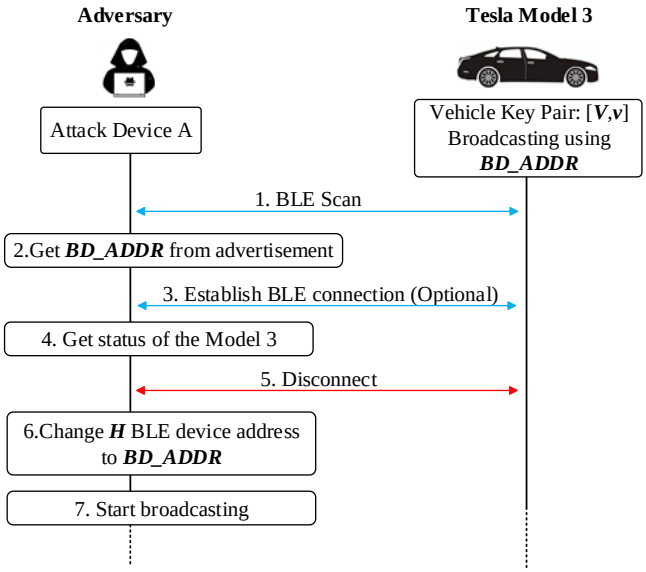

Fig. 14: Attack Model Phase 1: Setup

APPENDIX B
IMPROVED RELAY ATTACK - ONE ATTACK DEVICE

This section describes the scenario of one attack device in Figure 8. The attacker needs to go back and forth between the Model 3 and the car owner to complete the attack. The whole attack can also be divided into three phases as follow.

Setup: MAC spoof (Figure 14). First, the attack device approaches Model 3 and initiates a BLE scan. The attacker can get the BLE MAC address of the car from advertisements (Step 2). Connecting to Model 3 is optional depending on whether the attacker needs to obtain other information, for example, the public key of the vehicle. The adversary sets the attack device with the same BLE MAC address as the car (Step 6). Then the attack device starts to broadcast connectable advertisements identical to the Model 3's broadcasting.

Preparation: attestations capture (Figure 16). Once the attack device with the specific MAC address approached the Phone Key, an automatic connection establishes between the two parties. An authentication level indication of driving is delivered from the attack device to the Phone Key every second (Step 9). After receiving, Phone Key generates the shared secret using ECDH. After encrypting the known serialized java object (Listing 1) in AES-GCM mode using the shared secret as the key, the Phone Key sends the first attestation package (Steps 12 - 14). The attack device records the first attestation (Step 16).

Then the attacker goes back to the Model 3 and connects to the vehicle. The attacker forwards the recorded first attestation to the car and waits for the response involving a token G (Steps 19 - 24).

Next, the adversary approaches the owner again. Similarly, the Phone Key transmits the first attestation for verification. Besides recording this package, the attack device responds to the phone with the counter and the token G recorded before (Steps 30 - 32). Then the second attestation package sends back. The attack device can collect two consecutive attestations with the corresponding token G as a tuple.

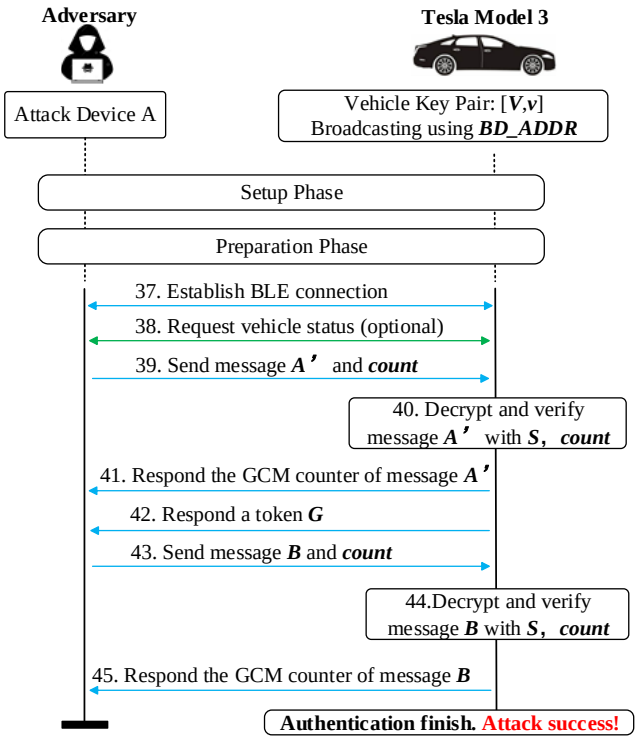

Fig. 15: Attack Model Phase 3: Attack

Attack: unlock and steal (Figure 15). Finally, the attack device reconnects to the car and sends the first attestation in the tuple (Step 39). Once receiving the response with a token, the attacker delivers the second attestation (Steps 42 - 43). If this token is the same as the one in the tuple, the car verifies both attestations successfully. For now, the Model 3 unlocks doors and can be driven away.

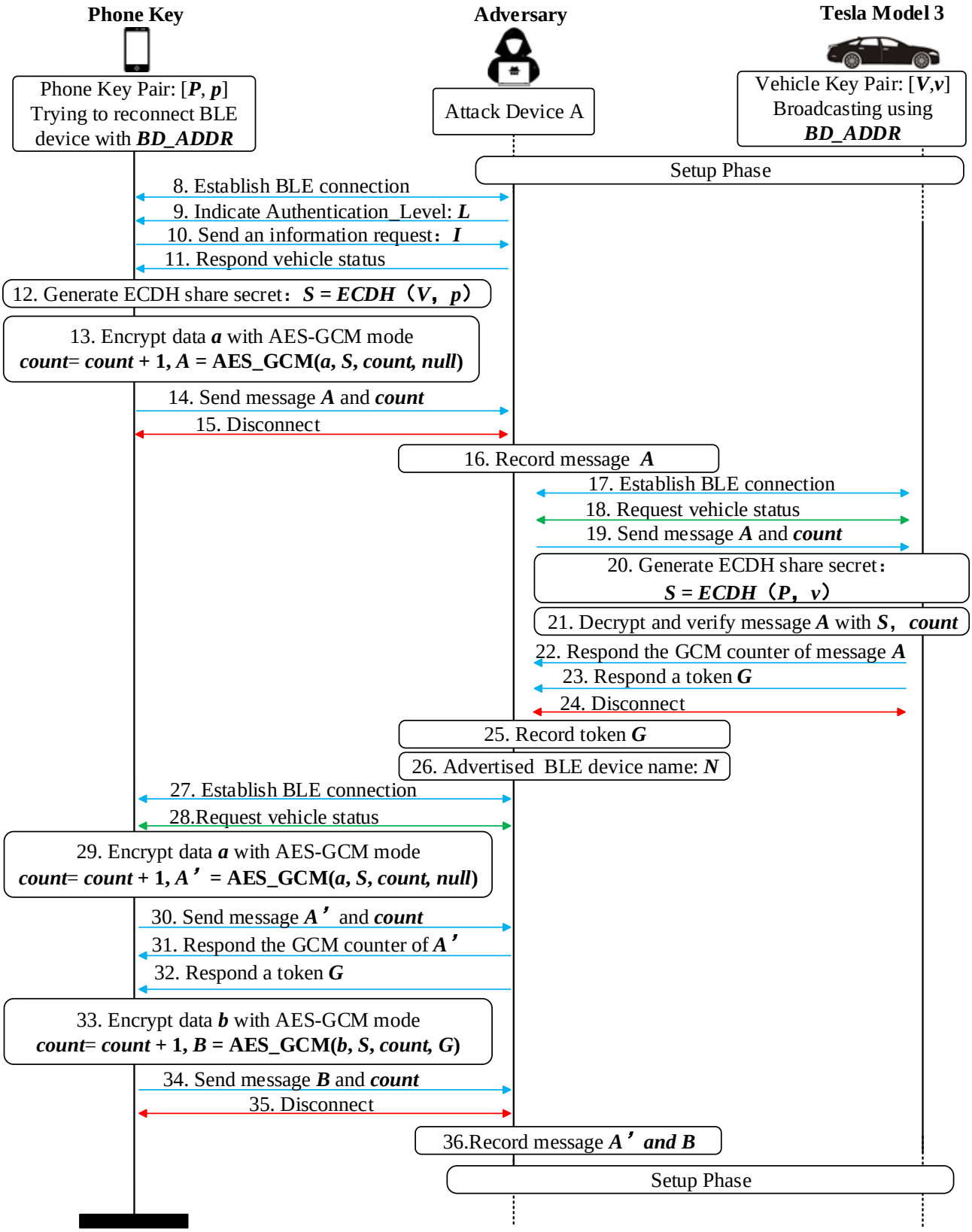

Fig. 16: Attack Model Phase 2: Preparation