ACCEPTED FROM OPEN CALL

TSP Security in Intelligent and Connected Vehicles: Challenges and Solutions

Yansong Li, Qian Luo, Jiajia Liu, Hongzhi Guo, and Nei Kato

Abstract

The rapid development of IoT, cloud computing, Artificial Intelligence (AI), big data, and 5G technologies has promoted the transformation of traditional vehicles toward ICVs. Compared to traditional vehicles, ICVs face many security threats introduced by network technologies and intelligent devices, especially in the field of remote wireless communications using Telematics Service Provider (TSP). As the core communication system in ICVs, TSP integrates diverse communication systems, and thus inherits the original vulnerabilities of these systems inevitably. TSP provides various methods for the ICVs to access the Internet, which makes them vulnerable to remote attacks. However, existing auto manufacturers mostly focused on the user experiences of the ICVs, and paid little attention to these potential security risks raised by TSP. Toward this end, in this article we analyze and summarize the TSP security threats in ICVs, and present some attack methodologies. After that, we discuss a practical attack case against an ICV by leveraging the vulnerabilities of TSP, and some countermeasures are proposed to enhance ICV security against TSP attacks.

Introduction

With the development of technologies such as Internet of Things (IoT), cloud computing, Artificial Intelligence (AI), big data, 5G, and so on, Intelligent and Connected Vehicles (ICVs), which are equipped with advanced on-board sensors and controllers, have emerged as the trends of modern vehicles and demonstrated their capabilities in various fields. Compared to traditional vehicles, ICVs can achieve driver assistance, information exchanging/sharing between vehicles and diverse terminal devices, by adopting communication and computing technologies.

In general, ICV communication is mainly composed of two parts, that is, the in-vehicle communication, and the communication between the vehicles and other infrastructures. In particular, as a protocol providing communication services for all electronic components in a vehicle, the controller area network (CAN) bus connects the in-vehicle infotainment (IVI), on-board diagnostic (OBD), and telematics box (T-Box) with each electronic control unit (ECU) to form a complete in-vehicle network environment. Outside the vehicles, ICVs can connect with other vehicles or roadside units (RSUs) by adopting dedicated protocols (e.g., IEEE 802.11p), and also communicate with other devices like smart phones, smart bands, and so on, via Bluetooth/WiFi.

As the core of the external wireless network in the ICVs, the Telematics Service Provider (TSP) builds a bridge for long distance communications between the ICVs and external equipments. On one hand, the ICVs can access the Internet via T-Box and 3G/4G cellular networks to obtain different remote services, such as electronic maps, Web portals, software updates, and so on. On the other hand, the users can use specific apps associated with the ICVs to connect TSP for remote controlling of the vehicles. It is noted that the intelligent networking systems used in ICVs inherit the existing computing and networking architectures, and thus the security flaws of these systems. Moreover, with the introduction of new network architectures, smart devices, wireless interfaces, and new communication protocols, ICVs face increasing security threats.

The authors in [1] pointed out that the potential attack surfaces of the ICVs, including remote keyless entry (RKE), Bluetooth/WiFi, and telematics/Internet/app, can lead to privacy disclosure of the users and even cause the vehicles to be controlled remotely. In [2], the authors simulated a scenario where a number of botnets attack a vehicular network, resulting in traffic jams and possibly even serious traffic accidents. However, most traditional auto manufacturers pay lots of attention to the user experiences of ICVs, and neglect the potential security risks related to property security and life safety. Among these security issues, external wireless system attacks against TSP are the most vulnerable and threatening, since they have a better hiding-place and cover a wide range of areas via cellular network in ICVs, and often cause serious safety hazards and catastrophic consequences for drivers.

Different from traditional cloud, TSP only uses the cloud as a platform to connect ICVs with remote control and management terminals, and it is characterized by a combination of multiple protocols and structures to form a new architecture that is specific to ICVs. Cloud security mainly involves the security of the users' privacy and property, while TSP security is also related to the safety of the users. Therefore, TSP security is not just about cloud services, but includes the supporting terminals and systems (e.g., T-Box, smartphone, and app). Nevertheless, to the best of our knowledge, few works can be found that focus on TSP

Digital Object Identifier:
10.1109/MWC.2019.1800289

Yansong Li, Qian Luo, Jiajia Liu (corresponding author) and Hongzhi Guo are with Xidian University; Nei Kato is with Tohoku University

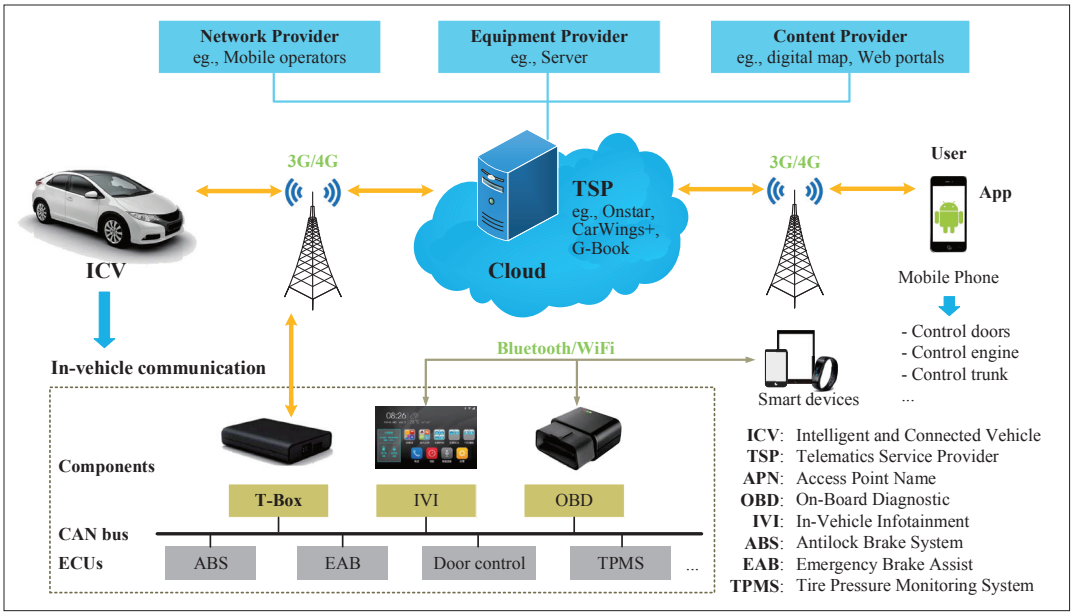

FIGURE 1. The figure demonstrates the typical network architectures of the ICVs including the telematics communication at the top of the figure and in-vehicle communication at the bottom of the figure. Regarding the telematics communication part, the ICVs can obtain telematics services including Network Provider, Equipment Provider, Content Provider, and TSP, via 3G/4G cellular networks. The in-vehicle communication part illustrates that all the electronic components can communicate with each other through CAN bus.

vulnerabilities. In light of this, we analyse and summarize the security threats and attack methodologies in the TSP of ICVs, and present an attack case study against TSP security issues. After that, some countermeasures are proposed for preventing TSP attacks in ICVs.

The remainder of this article is organized as follows. The following section narrates the architectures of ICVs and TSP. Then we summarize the threats and attack methodologies in cloud and TSP. A practical wireless attack against TSP is then presented. On reflection, we propose precautions against TSP attacks, and finally conclude the article in the final section.

Intelligent and Connected Vehicles

Architectures of ICVs

In addition to being embedded with a large number of intelligent electronic devices, most ICVs are equipped with TSP, which aims to supply remote control, third-party services and various information for enhancing driving. For example, the Toyota Prius C has many advanced intelligent devices such as fixed speed cruise, hill-start assist control, and so on, and a TSP named G-Book. The G-Book can provide telematics services including emergency rescue service and guarding against theft. As shown at the top of Fig. 1, ICVs can connect with TSP to obtain telematics services through 3G/4G cellular networks. The bottom of Fig. 1 illustrates a typical architecture of ICVs' in-vehicle communication. From this figure, we can find that the CAN bus and ECUs compose ICVs' in-vehicle network. In particular, the CAN bus establishes the connections among all the electronic components in the ICVs. ECUs are embedded devices consisting of various intelligent systems (such as the antilock brake system (ABS), emergency brake assist (EAB), and door control), and they are usually used to monitor vehicle status and control the car's behavior.

Furthermore, T-Box, which can directly communicate with the CAN bus, is the key device connecting ICVs to TSP. IVI, integrating lots of functions including entertainment, navigation, and driving assistance, is aimed at providing drivers with more comfortable and convenient driving experiences. OBD can record the fault information and offer an interface to connect a specific tool (e.g., OBD-II) for fault diagnosis and CAN bus data collection. Both OBD and IVI can provide the wireless interfaces including Bluetooth and WiFi for ICVs to communicate with outside devices like smart watches, iPads and smartphones.

TSP

As shown at the top of Fig. 1, TSP is mainly supported by three parts, that is, the equipment provider (EP), content provider (CP), and network provider (NP). Specifically, the EP offers TSP hardware and software supports, and the CP provides TSP with services like digital maps, Web portals, multimedia information, and so on. The NP, including mobile operators, fixed-line operators, and satellite operators, is developed to supply communication infrastructures between ICVs and TSP. TSP (e.g., Onstar, CarWings+, and G-Book) is the bridge connecting a vehicle with various terminal devices for information exchange and sharing. By integrating the resources from CPs/EPs and collecting information from intelligent transportation systems, vehicles, and so on, TSP can communicate with T-Box and specific smartphones via cellular networks.

Moreover, TSP can receive the car's status and users' information uploaded by T-Box to prevent access by illegal users. In return, the T-box receives

The CAN bus establishes the connections among all the electronic components in the ICVs. ECUs are embedded devices consisting of various intelligent systems (such as the antilock brake system (ABS), emergency brake assist (EAB), and door control), and they are usually used to monitor vehicle status and control the car's behavior.

Threats	Typical attack methodologies	Details	Platforms	References
T1	DDoS/EDoS	EDoS attacks are the main form of DDoS attacks in the cloud	Amazon EC2, Amazon S2 and Rackspace	2017, [3]
T2	Side-channel attacks based on placement vulnerabilities	High-bandwidth memory bus covert channel	Own testbed server and Amazon EC2	2012, [5]
		L2 cache channel	Amazon EC2	2009, [6]
		L3 cache channel	Unspecified	2016, [8]
		Generic, timing-based side channel	VMware	2013, [9]
T2, T3	MITM	Against Java library	Amazon EC2	2012, [10]
T2, T4	MITC	Against cloud storage	Google Drive, Dropbox, OneDrive and Box	2015, [11]
T2, T3, T6	Traditional Web attacks	Interception, XSS and so on	Eucalyptus, Amazon EC2 and Amazon S3	2011, [12]

TABLE 1. Threats in the traditional cloud platform.

and processes the commands from TSP. Based on this, the operations on ICVs such as opening the doors, stopping the engine, opening the trunk, and so on, can be remotely performed by utilizing specific smartphone apps connected to TSP. TSP can also provide ICV users with third-party services and resources for more convenient and safer driving. For instance, the third-party cloud platform and electronic maps offered by TSP can compute and plan the optimal route for drivers in real time. However, once TSP is compromised, the user's privacy and in-vehicle data may be revealed, and the target vehicle can even be completely controlled.

Threats in the Cloud and TSP

In this section, we first analyse the potential threats in traditional clouds, and then present some representative attack methodologies against the clouds, as illustrated in Table 1. After that, several typical threats and attacks against TSP in ICVs are summarized in Table 2.

Potential Threats for Cyber Attacks in Traditional Clouds

T1: Resource Exhaustion: Resource exhaustion occurs when the cloud computing infrastructures could not meet the users' requirements. Once the resources provided by the cloud service provider are exhausted, the cloud service will be unavailable, and thus the users cannot use shared resources in the cloud platform.

T2: Information Disclosure/Data Loss: Data leakage is endless in the network, which often brings serious economic losses to enterprises, and poses a major challenge to the privacy security of users. Therefore, protecting data integrity and security in the cloud is imminent in this increasingly privacy-conscious era.

T3: Account, Service and Traffic Hijack: When an unauthorized party accesses, changes or hijacks the user's data when sending to the cloud server, illegal interception of data occurs. Through this interception behavior, an attacker has the opportunity to manipulate, block and eavesdrop the data.

T4: Misuse of Privilege and Unauthorized Elevation: Unauthorized privilege escalation means one party accesses service, software, or hardware without permission. When a user accesses the organization's infrastructures, special high-level permissions may be required. In such cases, an attacker can improve his permission level and perform illegal operations by exploiting vulnerabilities.

T5: Insecure System Interfaces and Configuration: Cloud computing providers always expose a set of software interfaces used for managing cloud services and interacting with cloud customers. Based on these interfaces, cloud operators and third parties can provide value-added services for their customers. Nevertheless, these actions inevitably increase the complexity of application program interfaces (APIs), and thus expand the attack surfaces that make the cloud be more vulnerable to the attacks.

T6: Insufficient Preparation: If the maintainer does not have a complete understanding of the content security policy (CSP) environment, insecure applications or services may be placed at the cloud. In this case, cloud users will be caught in some unknown security risks that cannot be predicted.

Representative Attack Methodologies in Traditional Clouds

Distributed Denial of Service (DDoS)/Economic Denial of Sustainability (EDoS): DDoS attacks, which launch attacks through a large number of network nodes by using reasonable requests to overload server resources, always result in the cloud services being unavailable. EDoS is an attack method derived from DDoS attacks specifically targeted at imposing a significant financial burden on victims through DDoS attacks via customized/rented botnets, or skilled and real-time consumption of the victim's (pay-as-you-go) bandwidth. The authors in [3] introduced DDoS attacks and EDoS threats to cloud security in detail and proposed corresponding measures to mitigate these attacks.

Side Channel Attacks: Side channel attacks exploit information leaks from physical facilities such as timing, cache, energy consumption, etc. [4]. In particular, an attacker first ascertains co-residency where the target virtual machine (VM) is likely to reside by adopting a placement vulnerability [5]. This means that an adversary creates a new VM that shares the same physical machine with the victim after determining correlations between the public and private IP addresses. Then, the attacker can extract information from these side channels which are used to provide physical isolation with different VMs.

One of the prior works of cross-VM side channel attacks in the cloud was implemented via an L2 cache channel in Amazon EC2 [6]. There are other forms of channels that can leverage information on the same physical system [7]. For example, some side channels exfiltrated sensitive information of victims by exploiting the memory sharing on the L3 cache [8]. The authors in [9] discussed a generic practical timing side channel attack against

Threats	Entry points	Attack methodologies	Results	References
T1, T7	Routing instructions	Vehicular botnets	Cause serious road congestion	2015, [2]
T2, T8, T9	The same physical machine as the targets	Authentication bypass	Find the geographic location and obtain higher privilege to collect assets	2013, [13]
T3, T4, T5	Mobile operators	Port intrusion	Control IVI and perform a series of dangerous actions	2015, [1]
T5, T10	Talent, Web and SMS	Remote exploitation via a malicious update server	Remotely control safety-critical automobile features (e.g., the brakes)	2015, [14]
T5, T6, T10	Telematics unit	Vulnerability exploitation	Record cabin audio conversations and GPS location	2011, [15]

TABLE 2. New Security challenges in vehicular cloud platform.

Due to the high mobility and short transmission distance of ICVs, the authentication process has to be simplified, which also brings more unsafe factors. Attackers can utilize the loopholes to bypass authentication and log in to the TSP server in the cloud.

the memory management system to deduce the information of the privileged address space layout.

Man-in-the-Middle (MITM): By intercepting normal network communication and building a virtual connection between the two ends of network communication, an attacker alters and relays the communication between two parties who believe they are directly communicating with each other, and this method is called an MITM attack. For instance, the authors in [10] pointed out that the vulnerable softwares including Amazon EC2 Java library and all its cloud clients may cause an MITM attack.

Man in the Cloud (MITC): Cloud storage is a common service in cloud computing, which relies on a token for authentication, and this authentication method makes it increasingly easy to transmit and store large amounts of data. Box, Dropbox, OneDrive and Google Drive are all leaders in this field. Considering that most cloud service systems do not detect whether a token is stolen or not, an MITC attack can easily utilize the token for identity fraud [11]. Once the token is illegally obtained, the attacker can gain access to the cloud accounts, steal data, change file information, and even upload malicious files.

Injection and Cross Site Script (XSS) Based on Web Pages: Injection attacks occur when the user's input data is executed by the database interpreter, which allows the attackers to bypass authentication, access private information, modify data, and even destroy the database. When a user browses the Web pages, an XSS attack may happen. In such an attack, the attackers infiltrate the Web pages with HTML injection and insert malicious scripts (e.g., JavaScript) to control the user's browser. For example, the authors in [12] presented Web page security analysis on a large public cloud (Amazon) control interface and widely used private cloud software (Eucalyptus).

New Threats in TSP

T7: Abuse of TSP Services: Abuse of cloud services from TSP will lead to the deployment of cloud infrastructures being very expensive. Generally, there are mainly three cases that may result in abuse of TSP services, that is, lack of overall coordination of cloud resources, unreasonable usage of multiple cloud providers, and configuration of too many resources for an application.

T8: Insecure Authentication: ICVs tend to simplify the authentication process for connecting TSP due to their fast-moving status. This insecure authentication mechanism such as the shared key specified by the communication partner, may allow the hackers to easily forge identity and invade the TSP.

T9: Limited Storage and Inadequate Battery Life: For TSPs, they have to take both the limitations of the car's data storage and battery power into consideration. Once a power shortage occurs on the car, TSP will not be able to connect to the in-vehicle network and may lose the data stored in the car.

T10: Third-Party Cloud Platform Vulnerabilities: TSP is usually provided through third-party cloud platforms, and these platforms often have their security holes that can be exploited by hackers. Thus, cloud security avoidably affects the TSP security and even ICVs. Further, the third-party cloud platform vulnerabilities exposed to the ICVs have greater security risks than those on the cloud, since ICVs do not relate to the users' property, but have great relations with their life safety.

Typical Attack Methodologies for TSP

Vehicular Botnets: An attacker spreads zombie programs in various ways to infect a large number of hosts on the Internet, and the infected hosts will receive instructions from the attacker through a control channel to form a botnet. The authors in [2] demonstrated a botnet attack, which can cause serious congestion by targeting hot spot road segments, in an autonomous vehicle scenario.

Authentication Bypass: Due to the high mobility and short transmission distance of ICVs, the authentication process has to be simplified, which also brings more unsafe factors. Attackers can utilize the loopholes to bypass authentication and log in to the TSP server in the cloud. Many security challenges and potential privacy threats in vehicles were identified and analyzed in [13] where authentication bypass was discussed.

Port Intrusion: In a port intrusion, as the name suggests, a host is invaded through some open ports. Although port intrusion is a cliche attack technology, it still has strong vitality in the field of ICVs. The most classic wireless remote attack against the cloud platform connecting the in-vehicle network happened in 2015 [1], where Charlie Miller and Chris Valasek used a laptop to invade the ECU of the vehicle by means of the D-Bus service provided via port 6667.

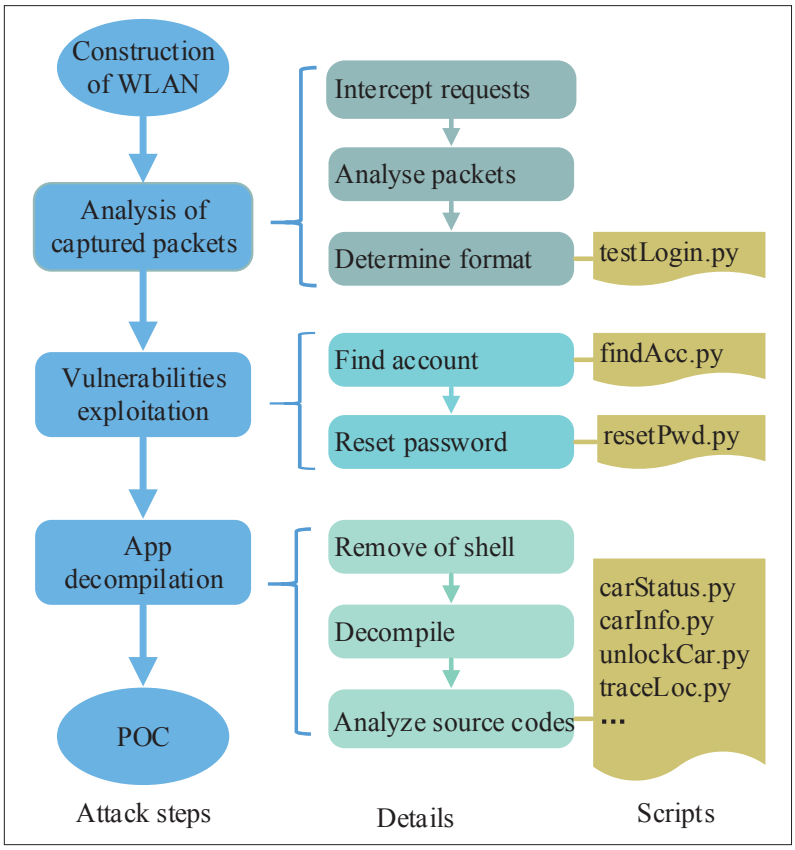

FIGURE 2. The figure demonstrates the process of actual attacks against an ICV. The left part shows the general idea, the center shows the detailed process of each step, and the right shows the scripts we developed.

+86-130****7704	+86-131****2427	+86-131****5234
+86-132****6450	+86-152****0640	+86-152****6434
+86-152****8033	+86-152****8056	+86-155****3374
+86-155****9714	+86-155****9334	+86-156****1348
+86-156****7940	+86-158****1502	+86-184****8334
+86-186****3847	+86-186****8907	+86-187****2944
+86-187****8744	+86-187****9322	+86-187****9619

TABLE 3 .

Malware Implantation: The implantation of malicious codes is also a kind of traditional and common attack method for controlling the target. For instance, the insecure remote update exploited by an attacker often causes malicious codes to be implanted [14].

Practical Attack Experiments

In order to validate that TSP does have vulnerabilities that can be exploited, we conducted a series of safety test experiments in this section, where a Luxgen U5 2017 SUV was adopted. By capturing the communication data between the vehicle's official app and the TSP, we discovered some authorization vulnerabilities and used them to find a number of relevant user names and passwords, which can be adopted to log in to the TSP server without authorization. After that, we controlled the vehicle by sending fake control commands to the TSP.

Detailed Attack Process (Fig. 2)

Construction of the Network Environment: A network scenario was built, where a wireless router was used to set up a wireless local area network (WLAN). In this scenario, a smartphone installing a Luxgen official app and a laptop with packet capture software were placed. After that, we can capture the packets from the app.

Analysis of Captured Packets: The login request uses Security Socket Layer (SSL), which makes it difficult to decrypt the content of the request using our own imported certificate. Here we can only use the analogy to guess the main body of the login request. Considering that not all communications between the app and the TSP are encrypted, we focused on the unencrypted data, and the specific process is as follows.

Step 1: Use Ettercap or Burpsuite to intercept all requests.

Step 2: Analyze unencrypted packets. During our experiments, we found that this app mainly uses different APIs to distinguish diverse instructions, and the request body is basically similar.

Step 3: Write the testLogin.py script to determine the body format of the login request. Based on the existing data, we continuously modified the data format of the login text and sent it to the server with a script until the server returned the correct response code.

Vulnerabilities Exploitation for Cracking User Account and Password: After knowing the correct login format, we wrote two scripts: one script is used for finding the accounts and the other for resetting the passwords. Further, we wrote a piece of code continuously trying to access the server.

Step 1: findAccount.py was used to search for different accounts. In particular, two kinds of response are returned by the server depending on whether the user name is correct or not. The app connecting the TSP uses a phone number as a user name, and we can send the login request to the server with different phone numbers and receive distinct response. After that, a correct user name can be easily found according to the specific response from the server. In this way, we can theoretically find all users who have registered this app. A few registered mobile phone numbers we found are listed in Table 3.

Step 2: resetPassword.py sent a command to the server requesting a Short Message Service (SMS) verification code to the account we just found. Here, we only chose an account that was already bundled with our experimental car. Although we do not know the content of the received message, the script can find the six-digit verification code by adopting a brute force method during the validity period of the verification code. After that, the obtained verification code was sent to the server together with the account and a new password, where the account's password was reset by adopting our script.

Step 3: userLogin.py emulated legitimate users to get the cookies and the tokens by logging into the server.

Android App Decompilation: We can send control instructions to the car through the TSP server when the cookies and tokens are obtained. However, the format of control instructions had not been clarified so far and cannot be inferred

through simple analogy. The only way is to decompile the Luxgen official app. Due to reinforcement and code obfuscation, the difficulty of cracking this app is greatly increased. The details of this process are described as follows.

Step 1: Remove the shell. According to some open source code, by adopting the ptrace() function provided by linux, we modified a shelling tool. With the aid of this tool, we dumped some key dex files of this app.

Step 2: Decompile. We first used dex2jar to decompile the dex files into jar packages, and then adopted jd-gui to view their source codes.

Step 3: Analyze the source codes. The purpose of analyzing the source codes is to better understand the communication mechanisms between the app and the TSP server, and the corresponding instruction format.

Proof of Concept (POC): After the accumulation of the previous steps, we wrote some actual scripts to replace the Luxgen official app so as to remotely control the car without authorization.

Results

During our experiments, we found that the user accounts obtained in the third step exposed the users' privacy and laid the foundation for us to further attack against the TSP. More importantly, we can obtain multiple accounts instead of the account of the ICV owner, which indicates that these TSP vulnerabilities are a common issue in this brand of car. Moreover, with the aid of our scripts, which are designed to replace the official app, we can log in to the TSP server by adopting our obtained accounts and passwords above. After that, more dangerous operations could be performed on the vehicles.

In Fig. 3a, the app shows the current status of our experimental vehicle. As shown in Fig. 3b, we obtained private information of the vehicle through malicious codes, such as frame number, engine number, license plate number, travel distance, T-Box serial number, and so on. Real-time monitoring of the vehicle status information including doors, trunks, engines, batteries, and GPS positioning are also available for us. In particular, Fig. 4a illustrates the positioning of the vehicle, which is obtained from the TSP by adopting the Luxgen official app. However, with the help of our scripts, the real-time location of the vehicle can also be easily tracked, as shown in Fig. 4b. Moreover, we remotely controlled the vehicle behavior involving unlocking/locking the car, turning off the engine, whistle, and so on.

Through the vulnerabilities exposed and the scripts we wrote, sensitive information of the owners and the vehicles may be leaked and more vehicles can be controlled unconsciously, resulting in serious safety hazards to the owners.

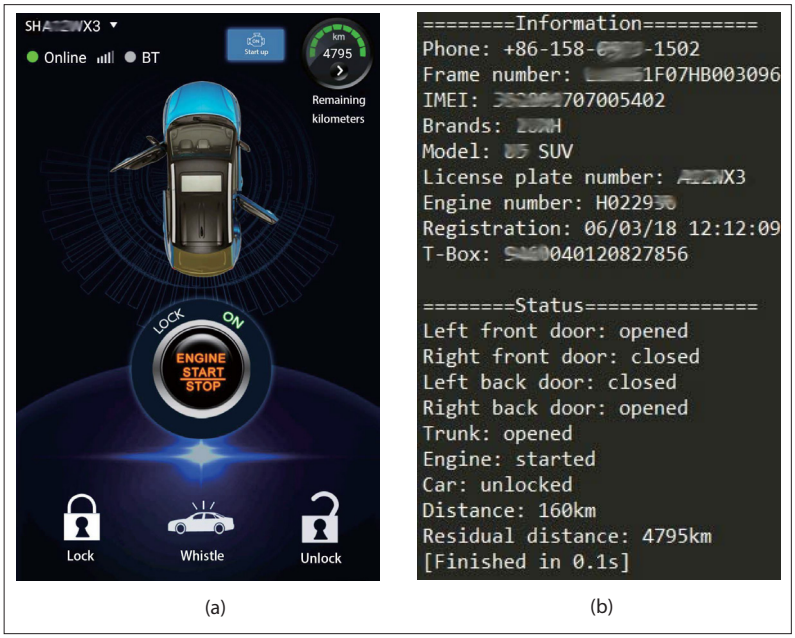

FIGURE 3. The figure shows the result of infiltrating TSP. Privacy information and current operating status of the vehicle are obtained from the malicious scripts we wrote: a) The official app shows the status of the car; b) Privacy exposed by the attack scripts.

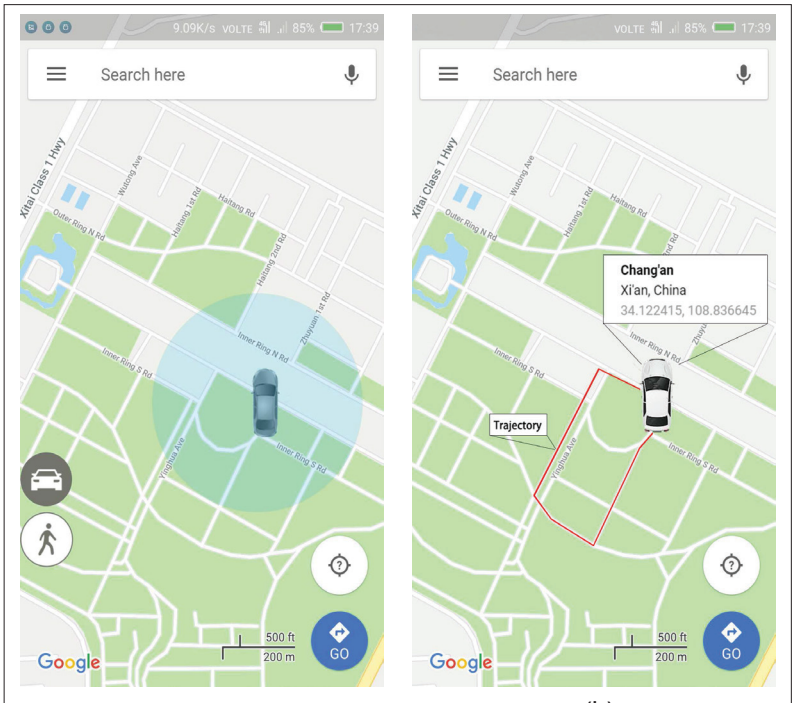

FIGURE 4. The figure shows the positioning of the car which is sent to the user's phone through the TSP: a) The official app shows the current car's location b) Malicious scripts integrate real-time tracking of the car.

Countermeasures

Interface Management and Information Encryption: The remote control system of the Luxgen U5 only encrypts some key data during the transmission process, which makes it possible for an attacker to infer the authentication format of the login request without encryption. In fact, all interfaces have the possibility to expose sensitive information and receive malicious data. Therefore, for safe communications between TSP and ICVs, TSP should either encrypt more information or reduce the coupling between the encrypted data and the plain text as much as possible.

Strong Authentication: During our experiments, we found that TSP has an unreasonable cloud authentication process, for example, too much time required for the authentication codes, and a lack of restrictions on modifying sensitive information, resulting in the leakage of user

TSP should modify the data returned to ICVs. In other words, TSP is better to return the same data regardless of whether the account being used to login is correct or not. Moreover, the blocking of processing HTTP(S) requests for particular IPs can also prevent brute force password attacks.

names and the malicious modification of passwords. Thus, we can forge an identity to log in to the TSP server, which cause serious security risks to ICV owners. To address this issue, TSP should modify the data returned to ICVs. In other words, TSP is better to return the same data regardless of whether the account being used to login is correct or not. Moreover, the blocking of processing HTTP(S) requests for particular IPs can also prevent brute force password attacks.

App Reinforcement and Confusion: Our decompilation of the Luxgen official app had exposed the format of the app's control instructions sent to the TSP, allowing us to further control the vehicle. The reinforcement of the app is not perfect, and TSP should integrate multiple methods to protect the security of the app. On one hand, logic confusion can be used. For instance, by checking the special contents of some files or the running time of the codes, it can be judged whether the app is being debugged. On the other hand, code obfuscation can also be used to reduce the readability of the codes by substituting class and variable names.

Conclusions

In this article, we investigated the security issues in ICVs, especially in the TSP. First, the network architecture of the ICVs was presented. Second, we summarized the security risks and attack methods existing in the cloud and the TSP. After that, a practical attack against the TSP was conducted. The case study on the Luxgen U5 demonstrated that TSP does have vulnerabilities that can be exploited. To prevent such TSP attacks, some countermeasures were proposed as our solutions.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61801360, 61771374, 61771373, and 61601357); in part by the Fundamental Research Fund for the Central Universities (JB181507, JB171501, JB181506, and JB181508); and in part by the China 111 Project (B16037).

References

[1] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," *Proc. DEFCON*, 2015, pp. 1–91.
[2] M. T. Garip *et al.*, "Congestion Attacks to Autonomous Cars Using Vehicular Botnets," *Proc. NDSS*, 2015.
[3] G. Somani *et al.*, "DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions," *Computer Commun.*, vol. 107, 2017, pp. 30–48.
[4] A. O. F. Atya *et al.*, "Malicious Co-Residency on the Cloud: Attacks and Defense," *Proc. IEEE INFOCOM*, 2017, pp. 1–9.
[5] V. Varadarajan *et al.*, "A Placement Vulnerability Study in Multi-Tenant Public Clouds," *Proc. USENIX Security Symposium*, 2015, pp. 913–28.
[6] T. Ristenpart *et al.*, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," *Proc. ACM CCS*, 2009, pp. 199–212.
[7] V. Varadarajan *et al.*, "A Placement Vulnerability Study in Multi-Tenant Public Clouds," *Proc. USENIX Security Symposium*, 2015, pp. 913–28.
[8] M. Kayaalp *et al.*, "A High-Resolution Side-Channel Attack on Last-Level Cache," *Proc. ACM/IEEE DAC*, 2016, pp. 72:1–72:6.
[9] R. Hund, C. Willems, and T. Holz, "Practical Timing Side Channel Attacks Against Kernel Space ASLR," *Proc. IEEE S&P*, 2013, pp. 191–205.
[10] M. Georgiev *et al.*, "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software," *Proc. ACM CCS*, 2012, pp. 38–49.
[11] X. Liang *et al.*, "Man in the Cloud (MITC) Defender: SGX-Based User Credential Protection for Synchronization Applications in Cloud Computing Platform," *Proc. IEEE CLOUD*, 2017, pp. 302–09.
[12] J. Somorovsky *et al.*, "All Your Clouds are Belong to US: Security Analysis of Cloud Management Interfaces," *Proc. ACM CCSW*, 2011, pp. 3–14.
[13] G. Yan *et al.*, "Security Challenges in Vehicular Cloud Computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, 2013, pp. 284–94.
[14] I. D. Foster *et al.*, "Fast and Vulnerable: A Story of Telematic Failures," *Proc. USENIX Workshop on Offensive Technologies*, 2015.
[15] S. Checkoway *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," Proc. USENIX Security, 2011.

Biographies

Yansong Li received his B.S. degree in electronic science in 2016. He is currently working toward the M.S. degree at the School of Cyber Engineering, Xidian University. His research interests include vehicular network security and Android security.

Qian Luo received her B.S. degree in computer science from North University of China in 2016. She is currently working toward the Ph.D. degree at the School of Cyber Engineering, Xidian University. Her research interests include vehicular network security and vehicle navigation security.

Jiajia Liu [S'11, M'12, SM'15] is currently a full professor at the School of Cyber Engineering, Xidian University. His research interests cover wireless mobile communications, IoT, ICV security, and so on. He has published more than 100 peer-reviewed papers in many prestigious IEEE journals and conferences, and currently serves as an associate editor for IEEE TC & TVT, an editors for *IEEE Network* and *IEEE IoT Journal*, and a guest editor for IEEE TETC. He is a Distinguished Lecturer of IEEE ComSoc.

Hongzhi Guo [S'08, M'16] received his B.S. degree in computer science and technology from Harbin Institute of Technology in 2004, and M.S. and Ph.D. degrees in computer application technology from Harbin Institute of Technology Shenzhen, China, in 2006 and 2011, respectively. He is currently a lecturer with the School of Cyber Engineering, Xidian University. His research interests cover a wide range of areas including MEC, UDN, 5G communications, and IoT. He is now serving as an editor of the *International Journal of Multimedia Intelligence and Security*.

Nei Kato [A'03, M'04, SM'05, F'13] is currently a full professor at GSIS, Tohoku University. He currently serves as the Vice-President–Member & Global Activities of IEEE ComSoc, Editor-in-Chief of IEEE TVT, and Associate Editor-in-Chief of *IEEE IoT Journal*. He has also served as the Chair of SSC TC and AHSN TC of IEEE ComSoc. He is a Distinguished Lecturer of IEEE ComSoc and the VTS. He is a fellow of IEICE.