A Novel Smart Lock Protocol Based on Group Signature

Yonglei Liu[1], Kun Hao[1], Jie Zhao[1], Li Wang[1], and Weilong Zhang[2]

(Corresponding author: Jie Zhao)

School of Computer and Information Engineering, Tianjin Chengjian University, China[1]
26, Jinjing Road, Xiqing District, Tianjin, China
Quality Management Center, Hebei Jiaotong Vocational and Technical College, China[2]
Email: zhaoj@tju.edu.cn

(Received Mar. 2, 2021; Revised and Accepted July 8, 2021; First Online Nov. 13, 2021)

Abstract

Aiming at security vulnerabilities, complex identity token management, and lack of privacy protections in the smart lock protocol, we propose a novel smart lock protocol based on group signature. As a result, two types of cyber-attacks in existing protocols are discovered: Random number attacks and parallel session attacks. With challenge and response, the improved protocol fixes these two security vulnerabilities by group signature and mutual identity authentication. Furthermore, the complexity of unlocking identity tokens is reduced from O(n) to O(1), and the improved protocol can be applied to anonymous unlocking scenarios. Finally, the indistinguishability game proof and analysis show that the proposed smart lock protocol complete anonymous unlock, satisfy other security requirements of smart lock system such as mutual identity authentication and traceability for identity information of the unlocker, and has certain application prospects in the lightweight IoT smart device market.

Keywords: Anonymity; Group Signature; Mutual Identity Authentication; Provable Security; Smart Lock

1 Introduction

With the development of 5G mobile communication technology, big data, artificial intelligence, and other technologies, as well as the development and popularization of information acquisition and processing technologies such as smart sensors and cloud computing, people can obtain more knowledge from massive data and improve the level of human civilization, human society has also entered the Internet of Things [15] era. As an important derivative concept of the Internet of Things, smart devices [20] refer to equipment, appliances or machines that have sensitive and accurate perception functions, correct thinking and judgment functions, and effective executive functions. Smart devices have brought great convenience to our lives, such as smart home devices [5, 18], Internet of Vehicles [23], wearable devices [9], *etc.*, and have quickly penetrated into various fields of economy and society, involving education, logistics, medical care, automobiles, transportation, construction, *etc.* In 2020, smart device market has reached trillions in China, and smart homes, consumer electronics and smart cars are developing rapidly [6]. As a typical smart device, smart locks are gradually replacing traditional locks and are widely used in smart homes, smart cars, shared bicycles [19] and other fields. The user only needs to approach the smart lock and use the smart phone to unlock it. However, because smart lock manufacturers focus more on function realization, lack of security protection research and development capabilities, or considering development time and cost, there are common security risks in smart lock designs [24].

At present, the deep integration of smart devices with cloud computing and big data to form a data market [11] is the main deployment mode. While improving the level of intelligent control of devices, producers and consumers can trade data to discover user habits and preferences to improve the quality of products and services. But this also brings a new attack surface. Hackers can penetrate cloud servers, traverse and break through the access point of smart devices to connect to the Internet of Things, such as smart gateways [22], infect gateways and smart devices through malicious code, and launch distributed denial of service attacks, *etc.*, which can lead to data privacy leakage, malicious control, and serious consequences such as system failure [10]. Compared with other smart devices, the security of smart locks is more important. Once breached, it will cause huge threats to smart devices, family property and even personal safety, such as burglary after breaking the door lock. Therefore, considering the potential security risks of smart locks online for a long time, the difficulty of smart lock wiring in the door, and the energy consumption and cost of hardware modules, most smart locks in the industry cannot directly connect to the Internet through a fixed gateway, and need to use

Bluetooth Low Energy (BLE) Technology to connect to the Internet through smartphones as a mobile gateway, which is called Device-Gateway-Cloud model (DGC) [3]. The discussion of the security of smart locks in this article is also limited to the DGC model. Although the smart lock system using the DGC model can be offline for a long time to mitigate remote attacks and penetration, there are still serious security threats such as malicious unlocking, man-in-the-middle attacks, state consistency attacks, and even undiscovered security vulnerabilities. Furthermore, some anonymous unlocking scenarios, such as entering a self-serving adult product store are not considered. Therefore, based on previous research works, this article has discovered two security vulnerabilities in the original smart lock protocol, and proposed an improved smart lock protocol based on group signatures, which fixes the security vulnerabilities, simplifies the complexity of the certificate, and support anonymous unlocking.

2 Related Work

This section will summarize and review relevant researches on the security of smart locks.

2.1 Traditional Digital Lock

A digital lock system called Grey project is developed for office environments in 2005 [2]. An equipment-based authentication method is proposed in this project by using early smart phone features, including usage of camera to scan a QR code to obtain public keys and network addresses, mobile phone anti-theft, and proof of access rights. However, the system unlocking delay is relatively high, and smart phones are evolving rapidly. Nowadays, mobile phone features and user needs have changed. Although the gray project is outdated, it has certain guidance for developing smarter locks that are safer, more user-friendly, and faster unlocking. The Raspberry Pi is a tiny and affordable computer, Pinjala *et al.* [13] realize a smart lock based on the Raspberry Pi. After the visitor presses the doorbell, the smart lock activates the camera and sends reminders and real-time images to the administrator's smartphone. The administrator views and remotely authenticates visitor to complete the unlocking.

2.2 Smart Lock with Biometric Authentication

Smart locks with biometric authentication solve the problems of traditional digital lock that passwords are easy to lose and difficult to remember, and currently occupy certain markets, such as fingerprint recognition, face recognition, gesture recognition, and voice recognition. Zhu *et al.* [27] use face recognition technology and open source software OpenCV to propose an attribute tracking algorithm and effectively improve the recognition accuracy. Compared with traditional digital locks, the use of biological features improves the security of remembering and storing keys, but increases the complexity of system deployment, additional hardware, and product costs. At the same time, in view of the application background of the Internet of Things, smart locks with biometric authentication usually lack fine-grained authority management and access control mechanisms.

2.3 Smart Lock with Other Auxiliary Technology

Some smart locks use other communication technologies as auxiliary authentication methods [7], such as audio channel, USB, NFC, VLC, *etc.* Among them, visible light communication (VLC) technology uses visible light as the information carrier, and direct transmission of light signals in the air, which can effectively construct a secure information space with anti-interference and low energy consumption. Song *et al.* [14] use visible light recognition technology to realize a smart lock. After the user uses the smart phone to complete the authentication, the LED light will send out a visible light signal, and the receiver on the smart lock receives the signal to complete the decoding, authentication and unlocking. However, smart phones and LED lights should be connected to the same home gateway, which increases the complexity of system deployment.

2.4 Smart Lock with DGC Model

In order to adapt to the Internet of Things environment and reduce system complexity and control costs, most smart lock manufacturers adopt DGC architecture, such as August, Danalock, Okidokeys, and Kevo. Since the smart lock has no fixed network connection under the DGC model, the user's smart phone is required to act as a mobile gateway. Therefore, if a malicious user forcibly offline the phone, it will cause the communication interruption between smart lock and the cloud server, inconsistency of system status, and revocation evasion. Ho *et al.* [3] use eventual consistency to solve the above problems. As long as an honest user approaches the smart lock, the cloud server will update the user permission list of the smart lock. However, this solution has an attack time window. During the time window, malicious users can still use the recovered credentials to unlock the door before the owner reaches home. In addition, in order to solve the location spoofing in the man-in-the-middle attack, wireless body area network technology is used to indicate the unlocking intention and assist the authentication of the unlocker. Patil *et al.* [12] introduce an additional random number segment in the interaction between the smart lock and the cloud server in order to compensate for the attack time window of the state consistency attack. For users whose unlocking authority has been revoked, the server no longer provides encrypted random numbers for users. Therefore, illegal users will be rejected because

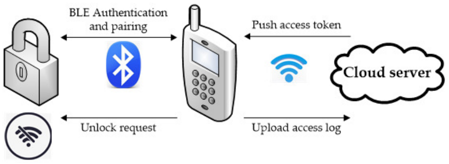

Figure 1: DGC model of smart lock

they cannot provide the random number encrypted by the server within the attack time window. In addition, in order to prevent attackers from maliciously concealing the unlock access logs, additional cameras are used to directly upload the access logs to the cloud server, but the attack surface of the system is undoubtedly increased. Xin *et al.* [21] propose an attribute-based access control mechanism for the problem of cascading deletion of smart lock permissions, using multiple environment attributes to refine access control and support group management, which is applicable to complex family relationships. Bapat *et al.* [1] use steganography to enhance the security of Bluetooth low energy communication between smart phones and smart locks.

In summary, the smart lock with biometric authentication enhances the authentication security to a certain extent, but requires additional hardware and lacks the classification control ability of multiple roles. The auxiliary smart lock also requires additional hardware, which increases the complexity of system deployment. The existing DGC smart locks still have security vulnerabilities. Due to the use of unlock identity token management, as the number of users increases, smart locks have the problem of token storage space overhead, and anonymous unlocking scenarios are not considered, such as entering a self-serving adult product store. Therefore, this article will solve these issues.

3 DGC Model and Security Analysis

This section introduces the DGC-based smart lock system architecture and conducts a security analysis.

3.1 DGC Model

The DGC model consists of three parts: a smart lock installed in the door, a smart phone, and a remote cloud server. Smart locks do not have a direct network connection to cloud services deployed on the Internet, requiring a smart phone to act as a wireless mobile gateway. The smart phone communicates with the smart lock through BLE, and the smart phone communicates with the remote cloud server through mobile communication network such as 5G, as shown in Figure 1.

Users are divided into four categories: owner, resident, recurring guest, and temporary guest. Among them, the owner can unlock the smart lock at any time and use all the administrator functions provided by the smart lock manufacturer, such as granting/recovering permissions, viewing unlock access logs, and updating keys. Residents can unlock the smart lock at any time but cannot use the administrator functions. Recurring guests can unlock the smart lock during the authorized time period, such as housekeepers who come to clean the house every Tuesday from 9 to 11 am. Temporary guest can unlock the smart lock during a temporary period, such as a neighbor visiting from 3 to 4 pm on a sunny day.

3.2 Smart Lock Protocol

The existing smart lock protocol [12] is divided into three phases: Initialization, permission update, and unlocking.

3.2.1 Initialization

- The smart lock (unique identification ID_L) is built with the root key RK_L at the factory, and RK_L is sent to the owner safely and confidentially along with the product manual. At the same time, secure storage (ID_L, RK_L) in the manufacturer's cloud server is processed. And each user has his own public and private key pair (PK_U, SK_U).

- The owner uses RK_L to access into the cloud server and generates the user's unlock identity certificate ($Token_U$), which is stored in the cloud server subsequently. $Token_U = (ID_L, ID_U, SN, Name_U, Type_u, Days_U, Time_U, Dates_U, PK_U)_{RK_L}$, Wherein, $Name_U$ is the user's name; ID_U is the unique user identification, which can be the user's mobile phone number; SN is the sequence number of the unlock identity certificate for permission update; $Type_U$ is the user type such as owner, recurring guest, *etc.*; A combination of $Days_U$ and $Time_U$ describes the authorized unlocking times for recurring guests, $Dates_U$ describes the authorized unlocking times for temporary guests. $Token_U$ is encrypted by RK_L.

- The owner sends $Token_U$ to the user, or after the user accesses the cloud server and passes identity authentication, such as a short message SMS, the user downloads the $Token_U$ in the smart phone.

3.2.2 Permission Update

- The owner access into the cloud server and updates the user's token.

- The owner's smartphone enters the Bluetooth communication range of the smart lock, and the cloud server sends the updated $Token'_U$ to the smart lock.

- The smart lock uses RK_L to decrypt $Token'_U$, verifies whether the SN is fresh, and updates the permission list.

3.2.3 Unlocking

- When the user's smart phone enters the Bluetooth communication range of smart lock, the user sends the $Token_U$ stored in the smart phone to the smart lock.

- The smart lock uses RK_L to decrypt $Token_U$, verifies whether the authority is consistent with the local tokens stored in the smart lock, and obtains the user's public key RK_U, and then (ID_L, N_1) is encrypted by PK_U and is sent to the user.

- The user uses the private key SK_U to decrypt to received (ID_L, N_1), verifies the ID_L and sends N_1 to the smart lock.

- The smart lock verifies N_1 and unlocks. The unlock access log is encrypted by RK_L and is sent to the cloud server.

3.3 Attack Behavior Analysis

Since smart locks with DGC model belong to distributed system, which is inevitably with a network partition. According to the CAP theorem, the availability and consistency of access permission lists, unlock access logs and other data in smart locks and cloud servers cannot be realized at the same time. Therefore, the most serious threat of smart locks is state consistency attacks, such as revocation evasion, access log evasion, *etc.* In addition, due to the characteristics of BLE communication, there are man-in-the-middle attacks and denial of service attacks against the BLE protocol itself. In this section of attack behavior analysis, we analyze the previous research work about state consistency and man-in-the-middle attacks in Section 3.3.1 and 3.3.2. The Sections 3.3.3 and 3.3.4 are our novel work. We discovered random number attacks and parallel session attacks against security vulnerabilities in the original smart lock protocol.

3.3.1 State Consistency Attack

- Revocation evasion. The owner access into the cloud server to withdraw the attacker's unlocking authority, and then the owner with the smart phone approaches to the smart lock to update the permission. However, in the attack time window, the attacker can still unlock the smart lock before the owner reaches home.

- Access log evasion. After the attacker unlocked the smart lock, the malicious smartphone receives the unlock access log and refuses to forward to the cloud server. In this way, the attacker could claim that stealing after unlocking is not convicted because the log was lost.

- Threat mitigation. Ho *et al.* [3] use eventual consistency to solve the above problems, all honest users regardless of owner unlocking the lock, it will trigger the permission update operation, but this action still has an attack time window. In addition, the unlock access log sent by the smart lock should be confirmed by the cloud server's signature, otherwise it will be retransmitted. Therefore, even if the attacker blocks the forwarding, the access log will be eventually uploaded to the cloud server when the honest user unlocks the smart lock. However, the attack time window still exists. Patil *et al.* [12] introduce a random number N_c between the smart lock and the cloud server in order to defend revocation evasion. In the final step of unlocking phase, the user sends N_c encrypted by the cloud server using RK_L. The smart lock cannot be unlocked within the attack time window as the encrypted N_c is not be provided by the cloud server. However, the unlocking process requires the participation of the cloud server, in case of network connection problems or cloud server failure, the smart lock cannot be unlocked and the system availability cannot be guaranteed. In addition, in order to defend access log evasion, an additional camera with permanent network connection is used to directly upload the access log to the cloud server, but this undoubtedly increases the attack surface of the system.

3.3.2 Man-in-the-Middle Attack

- The man-in-the-middle attack of the smart lock is essentially a collusion attack. Attacker A and attacker B have BLE communication channels, and both parties establish a hidden tunnel connection. A is close to the smart lock and is paired through BLE, and B is close to the user's smartphone and is paired through BLE. The user will mistakenly find that there is a smart lock around and sends token to B. B receives the token and forwards to A through the hidden tunnel. A sends the token to the smart lock. The smart lock generates a challenge. A continues to forward the challenge to B through the hidden tunnel, and B sends the challenge to the user. The user generates the challenge response and sends to B, and then B forwards the challenge response to A through the tunnel. Finally, A sends the challenge response to the smart lock, and the unlock succeeds.

- Threat mitigation. Some commercial smart locks use geo-fencing [16] technology to assist the user's mobile phone to determine whether the smart lock is nearby. Therefore, the user's smartphone can recognize that the attacker B is not a real smart lock, and refuses to send token to initiate the session. However, studies have pointed out that geo-fencing technology has security threats such as mistaken unlocking in multiple entrances and exits scenario, and there are also geo-fencing spoofing attacks. The root of the problem lies in auto-unlocking. If the user's unlock intention is asked every time, such as APP pop-up window, SMS, *etc.*, man-in-the-middle attacks can be easily identified. However, it greatly reduces the user experience and brings boredom. Therefore, the existing improvements are mainly to balance user experience and safety. Kevo smart lock combines geo-fencing and touch unlocking technology, the user re-enters the geo-fencing boundary and touches the unlock button to unlock. However, the system cannot verify the identity of the toucher. Ho *et al.* [3] use wireless body area network technology to indicate the unlocking intention. Compared with the Kevo smart lock, after the user touches the unlock button, the smart lock and the user's wearable device will complete the authentication to confirm the user's identity and then unlock.

3.3.3 Random Number Attack

- We find an attack called random number attack. In the last step of the unlocking phase, the user sends N_1 to the smart lock in plaintext. Therefore, the attacker passively listens for all communications between the smart lock and the legitimate user. And then, the attacker initiates a new session and replays the messages acquired during the passive listening. If the N_1 is predictable, the attack can be succeeded, for instance, the N_1 is incremental, the attacker injects $N_1 + 1$ to maliciously unlock.

- Threat mitigation. The random number N_1 is not predictable or is transmitted encrypted. In addition, the impersonation attack is fixed according to Section 3.3.4.

3.3.4 Smart Lock Impersonation Attack

We find a smart lock impersonation attack launched by parallel session attack. The existing smart lock protocols mainly focus on authentication of user rather than the authentication of smart lock, because even if user unlock a fake smart lock, none of assets are lost. For instance, the original smart lock protocol only uses ID_L to authenticate the smart lock, the attacker can easily predict ID_L through manufacturer past product identification number and product information.

- Attack description. Although the PK_U is protected in $Token_U$ and encrypted by RK_L, compared to private key, the PK_U is more easily to disclosed. Therefore, the attacker can predict ID_L and acquire PK_U in order to impersonate a legitimate smart lock. And then, the attacker launches unlocking interaction with legitimate users and obtains all communication traffic for replay attack, offline RK_L cracking and user unlocking behavior analysis. Under the conditions that the random number N_1 is not predictable and is transmitted encrypted, we find a new parallel session attack to make random number attack successful. Firstly, we launch random number attack and acquire $(ID_L,\ N_1)$ encrypted by PK_U. Secondly, without SK_U, we cannot acquire N_1, then launch a parallel session attack to impersonate a legitimate smart lock to ask the oracle of N_1 from a legitimate user. Finally, the response of N_1 is sent to victim of smart lock to complete malicious unlocking. The attack details are showed in Figure 2.

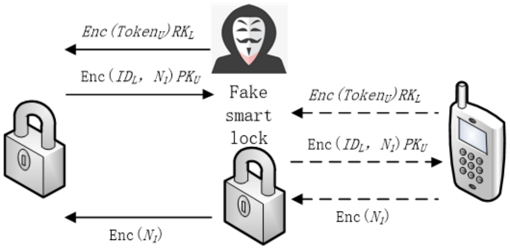

Figure 2: Parallel session attack

- Threat mitigation. The smart lock and the user are authenticated by a secure mutual authentication protocol, such as a challenge-response based mutual authentication protocol.

4 Improved Smart Lock Protocol based on Group Signature

The existing smart lock has high storage and communication overheads of unlocking identity certificate. In addition, in some scenarios, users have a strong desire to protect privacy, such as entering a self-serving adult product store. And as mentioned in Section 3.3, we found random number attacks and parallel session attacks. In response to the above problems, we propose a novel smart lock protocol to improve the smart lock protocol of reference [12] and use group signature with a shorter signature length [4] in our improvement protocol to realize the above security goals, as well as, suitable for lightweight smart home devices [8] such as smart locks. Detailed group signature algorithm can be found in the literature [4]. The improved protocol in this article focuses on anonymity, reducing communication and storage overheads of lengthy unlocking identity credential, and the security feature of the protocol to resist random number attacks and parallel session attacks that we have discovered.

4.1 Group Signature

4.1.1 Setup

The algorithm is given a system safety parameter λ and generates system public parameters $PP = (q, G_1, G_2, G_T, e, P_1, P_2, H(\cdot), n)$, wherein, G_1, G_2, and G_T are cyclic groups of order q (of length λ bits). $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear mapping. P_1 and P_2 are generators of G_1 and G_2 respectively. $H(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^n$is a safe hash function. The random numbers d, s, u are chosen to calculate $D = d \cdot P_1, S = s \cdot P_2, U = u \cdot P_1$ and respectively. The private key of group administrator sk is (d, s); the tracing private key tk is u; the group public key gpk is (D, S, U).

4.1.2 Enroll

According to the PP and the group administrator sk, the group member private key gsk is generated. x_i is randomly select and calculate $Z_i = (d - x_i)(sx_i)^{-1} \cdot P_1$ and $tag_i = H(x_i \cdot Z_i)$. The private key of the group member $gsk_i = (x_i \cdot Z_i)$. The group administrator manages the list of members $list = (GU_i, tag_i)$.

4.1.3 GSign

According to PP, gsk_i, gpk, group members randomly select k and calculate $C_1 = k \cdot P_1, C_2 = x_i \cdot Z_i + k \cdot U$ and $Q = e(U, S)^k$. For the message m to be signed, the group members further calculate $c = H(C_1, C_2, Q, m)$ and $w = kc + x_i$. The signature $GSig(m) = (C_1, C_2, c, w)$.

4.1.4 GVerify

The verifier verifies whether the signature is legal according to m, $GSig(m)$, and gpk. The verifier calculates $Q' = \frac{e(C_2,S) \cdot e(P_1,P_2)^w}{e(c \cdot C_1 + D, P_2)}$and verifies whether $c = H(C_1, C_2, Q', m)$.

4.1.5 GTrace

The group manager determines the signer according to the tracing private key tk and $GSig(m).tag_i = H(C_2 - u \cdot C_1)$ is calculated and search to determine the signer in the list of members $List$.

4.1.6 Revoking

The group administrator adds the revoked member's private key material x_i or $x_i \cdot Z_i$to the revocation list $RList$, the verifier traverses all the x_i or $x_i \cdot Z_i$ in the $RList$, and recognizes the revoked member by whether $e(C_2, S) \cdot e(x_i \cdot P_1, P_2) = e(D, P_2) \cdot Q'$ or $e(C_2 - x_i \cdot Z_i, S) = Q'$ respectively.

4.2 Improved Smart Lock Protocol

Due to space limitations, this section focuses on describing the differences and improvements compared to the original smart lock protocol. Wherein, the details of unlocking are shown in Figure 3.

4.2.1 Initialization

- The smart lock generates its own public-private key pair (PK_L, SK_L). The users are managed by group including permanent user group and guest group, and the token is simplified. The owner/administrator creates a group signature system to generate one permanent user group and n guest groups with GPK, tk, and sk. The owner/administrator uses sk to generate gsk of each user. $Token = (ID_L,\ SN,\ GN,\ Time,\ Cycle,\ gpk)_{RK_L}$. Wherein, GN is the group number, 1 represents the permanent user group, and $2-n$ represents the guest group with overlapping visit times, such as cleaning staff and water and electricity maintenance workers who visit from 8:00 to 9:00 every Tuesday. gpk is group signature public key. Given that guests often visit periodically, such as weekly cleaning and annual house maintenance, the field of the $Cycle$ is made to replace the field of $Days$ and $Dates$ for simplifying and reducing the length of token in original smart lock protocol. For instance, $Cycle$ is 0 for temporary visit once, 1 for every day, 30 for every month, *etc.* And access time determination is whether current time is equal to $Time + nCycle$.

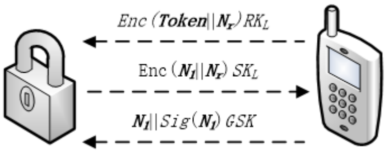
$Enc\ (\boldsymbol{Token}||\boldsymbol{N_r})RK_L$
$\mathrm{Enc}\ (\boldsymbol{N_1}||\boldsymbol{N_r})\ SK_L$
$\boldsymbol{N_1}||Sig(\boldsymbol{N_1})\ GSK$

Figure 3: Improved unlocking protocol

- The owner/administrator securely sends the $Token$ and the corresponding group member private key gsk to each user.

  Compared with the original protocol, all users from one group only store one token on the smart lock and cloud server, where n is the number of users, the complexity is reduced from O(n) to O(1). And the length of the token is also reduced, which is more friendly to lightweight IoT devices such as smart locks.

4.2.2 Unlocking

- When the smart phone of user enters the scope of Bluetooth communication with smart lock, the user sends $Token||N_r$ to the smart lock.

- The smart lock uses RK_L to decrypt $Token$, and determines freshness and updates $Token$ through SN; the smart lock verifies whether the permissions are consistent with the local Token stored in the smart lock; the $Token$ will be stored in the smart lock directly, if a user of the group opens the lock the first time; the smart lock acquires the group signature public key GPK. $(N_1,\ N_r)$ is encrypted by the private key SK_L of smart lock and sent to the user.

- User decrypts (N_1, N_r) by the public key PK_L of the smart lock, verifies the N_r, sends N_1 to the smart lock and signs N_1 with the private key GSK of the group member.

- The smart lock verifies N_1, uses GPK to verify the signature, and judges whether the user's permission has been revoked according to the group member revoking algorithm. If the above verifications pass, the lock will be unlocked, and the unlock access log is sent to the cloud server.

4.2.3 Permission Update and Log Access

- The owner/administrator accesses into the cloud server and generates the RList using the group member revoking algorithm.

- The smart phone of the honest user enters the Bluetooth communication range of the smart lock, and the cloud server sends encrypted RList to the smart lock.$(ID_L, SN, GN, RList)_{RK_L}$

- If the above algorithms are performed or partly performed on the smart door lock or the user's mobile phone, that the smart door lock is directly controlled by the mobile phone, more functions and class libraries related to smart lock hardware should be installed on the smart phone, which increases the complexity of the system and the energy consumption of the smart lock undoubtedly. Moreover, for the equipment manufacturer, the control of the smart lock and the collection of product usage data are lost. Under the DGC model, the wireless communication link such as 5G from the mobile phone to the cloud server has more mature security protection methods, but the BLE link between the mobile phone and the smart door lock is relatively insecure, and there are more security threats such as illegal eavesdropping and man-in-the-middle attacks. Therefore, in the balance between cloud server single point of failure and system security, this paper also continues to use the DGC model for group signature algorithm performed in could server.

- Two methods can be adopted to trace unlocker. The former one is that the smart lock sends unlock access log with group signature to the cloud server, the owner/administrator logins cloud server and use tk to trace unloker. The latter one is that the cloud server sends $(ID_L, SN, GN, RList)_{RK_L}$ to the smart lock in the initialization phase, the smart locks use tk trace unlocker and sends unlock access log with revealed unlocker directly to the cloud server. This approach conflicts with the original intention of anonymity and increases the security risk of the system when the smart lock is physically attacked or captured, so we adopt the former.

4.3 Security Analysis

In this section, using the provable security theory, a security model of our improved smart lock protocol is established, and security analysis is processed.

4.3.1 Adversary Model

The system has smart locks, users, cloud servers, and adversaries A. The ability of the adversary A is consistent with the Dolev-Yao model [17], which can passively listen, steal, forge, and block all communication between users in the channel. Using an oracle to simulate an instance run of the protocol, the attack capabilities of adversary A can be simulated as the following oracle queries:

- Setup-Oracle. A can acquire PP, gpk and public key PK_L of the smart lock by querying the oracle.

- Excute-Oracle. The oracle simulates passive attacks. A can acquire all the messages sent by honest users running the smart lock protocol by querying the oracle.

- Send-Oracle. The oracle simulates active attacks, A sends a message msg to the oracle, and the oracle processes msg according to the protocol rules, and sends the result msg' to A.

- Corrupt-Oracle. The oracle simulates the loss of the unlocking identity credential, and A can acquire the token of any user through query.

- Handle Dispute-Oracle. The oracle simulates signature tracing, and A asks the identity of the signer associated with the N_1. The oracle calculates tag_i through tk and returns tag_i to A.

- Test-Oracle. The oracle does not simulate attack ability of A, but judges the advantage of A in winning the game. After receiving the query request, A randomly selects two unlocking sessions S_0 and S_1 that A has never inquired in the handle dispute oracle. The oracle randomly selects $b \in \{0, 1\}$, and uses tk to trace the message in the session Sb: $N_1||GSig(N_1)$ and calculate tag_b. A gets tag_b and guesses b'. If $b' = b$, A wins, otherwise, A loses. $ADV^A_{SL}(A)$ is defined as the winning advantage of A.

$$ADV^A_{SL}(A) = |Pr(b' = b) - 1/2| \tag{1}$$

4.3.2 Indistinguishability Game Proof

Theorem 1. *The improved smart lock protocol is anonymous, if and only if $ADV^A_{SL}(A)$ is negligible for any polynomial adversary A.*

Proof. Assumes that $ADV^A_{SL}(A)$ is not negligible in distinguishing unlock session S_0 and S_1. Defines adversary B that can break the group signature system, and the attack capabilities of adversary B can be simulated by similar oracles in Section 4.3.1, wherein, in the Test-Oracle, B randomly selected two group signature message $m_1||GSig(m_1)$ and $m_2||GSig(m_2)$. The oracle randomly selects $b \in \{0, 1\}$, and uses tk to trace the message $m_b||GSig(m_b)$ and calculate tag_b. B gets tag_b and guesses b'. If $b' = b$, B wins, otherwise, B

loses. Obviously,$ADV_{SL}^{A}(A) = ADV_{GS}^{A}(B)$ then the adversary B has a non-negligible advantage in breaking the anonymity of group signature, which contradicts with the Theorem that there is no polynomial adversary can attack the group signature anonymity with a non-negligible advantage [4], so the hypothesis is not valid and Theorem 1 is correct. □

4.3.3 Other Security Analysis

In addition to unlocking anonymity, the improved smart lock protocol has other security features.

- Resistance to state consistency and man-in-the-middle attacks. (in Section 3.3.1 and 3.3.2) Aiming at state consistency attacks and man-in-the-middle attacks, we follow the improvements of previous researchers. Due to system complexity considerations, we use the honest user mechanism and log confirmation mechanism of Ho *et al.* [3] to mitigate state consistency attacks, and use the mechanism of disabling auto-unlocking by the advisement of Patil *et al.* [12] to mitigate man-in-the-middle attack in our improved protocol.

- Mutual authentication for resistance to random number and parallel session attack. (in Section 3.3.3 and 3.3.4)The improved smart lock protocol uses challenge N_r and the smart lock private key SK_L to encrypt N_r. Since the attacker cannot obtain SK_L, the smart lock cannot be faked. Simultaneously, a group signature is used for user authentication. Since the attacker cannot obtain the group member's private key gsk_i, the signature of N_1 cannot be constructed for the attacker to impersonate the user. Therefore, the improved smart lock protocol can resist random number attacks and parallel session attacks.

- Token loss and forgery. If the user's Token is illegally leaked, although the attacker can initiate an unlock request, but since the group member's private key gsk_i cannot be obtained, the signature of N_1 cannot be constructed to impersonate the user. In addition, because the attacker cannot obtain the root key RK_L and cannot forge $Token$.

- Resistance to replay attacks. When an attacker performs a replay attack, it will fail because of the freshness of the random number, even if using the Send-Oracle in 4.3.1. For example, the attacker C impersonates a legitimate user to maliciously unlock the lock and replay the unlock request. In last step of the protocol, C requires a group signature on the random number selected by the smart lock, such as N_6. C launches a parallel session attack through Send-Oracle and impersonates a legitimate smart lock to send N_6 to a legitimate user for the answer to the N_6 signature. But since the private key SK_L of the smart lock cannot be obtained, the Send-Oracle cannot be performed. Even if the attacker replays the question N_6 encrypted SK_L, however the N_r of two parallel session are different, the attack is impossible to succeed. Therefore, C can only passively listen to the communication traffic of the legitimate user and smart lock for a long time until C captures the N_6 signature and replays.

  Assumes that $|N_1|$ represents the length of random number N_1. Obviously, the following winning advantage of adversary C can be conducted.

  $$ADV_{SL}^{Replay}(C) \leq 1/2^{|N_1|} \qquad (2)$$

  Under a secure random number length, such as 256 bits, the winning advantage of adversary C can be ignored, and the improved smart lock protocol can resist replay attacks.

- Traceability. When a security accident or property loss occurs, the owner/administrator can use the tracing key tk to find the intruder. Since the attacker cannot obtain tk, the attacker cannot acquire traceability.

5 Conclusions

With the advent of the Internet of Things era, smart devices have gradually entered all aspects of social and economic life. As an important smart device, the security of smart locks has attracted much attention. Two new security vulnerabilities in the existing DGC architecture smart lock protocol are discovered in this article: random number attacks and parallel session attacks. A smart lock protocol based on group signature is proposed, which simplifies the complexity of unlocking identity credentials from O(n) to O(1), resists random number attacks and parallel session attacks, and can be applied to anonymous unlocking scenarios. The indistinguishability game proof and security analysis show that the improved smart lock protocol proposed in this article satisfies the security requirements of smart locks and has certain application prospects. In the next step, we will study the proposed improved smart lock prototype system and explore the combination of other related security technologies, such as machine learning [26] and blockchain [25].

Funding: This research was funded by National Natural Science Foundation of China, grant number NO. 61902273.

References

[1] C. Bapat, S. Inamdar, G. Baleri, *et al.*, "Smartlock security re-engineered using cryptography and steganography," in *International Symposium on Security in Computing & Communication*, pp. 325-336, 2017.

[2] L. Bauer, S. Garriss, J. M. Mccune, *et al.*, ”Device-enabled authorization in the grey system,” in *International Conference on Information Security*, pp. 431-445, 2005.

[3] G. Ho, D. Leung, P. Mishra, *et al.*, ”Smart locks: Lessons for securing commodity internet of things devices,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 461-472, 2016.

[4] T. Ho, L. Yen, C. Tseng, ”Simple-yet-efficient construction and revocation of group signatures,” *International Journal of Foundation of Computer Science*, vol. 26, no. 5, pp. 611-624, 2015.

[5] Q. Huang, Z. Li, W. Xie, *et al.*, ”Edge computing in smart homes,” *Journal of Computer Research and Development*, vol. 57, no. 9, pp. 1800-1809, 2020.

[6] iiMedia Research, *2020 China Smart Hardware Industry Development Panorama Research Report*, 2020. (`https://www.iimedia.cn/c400/70397.html`)

[7] J. Jeong, ”A study on smart door lock control system,” *Cluster Computing*, vol. 19, no. 3, pp. 1-11, 2016.

[8] C. Lin, D. He, N. Kumar, *et al.*, HomeChain: A blockchain-based secure mutual authentication system for smart homes,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818-829, 2020.

[9] Q. Liu, T. Li, Y. Yu, *et al.*, ”Data security and privacy preserving techniques for wearable devices: A survey,” *Journal of Computer Research and Development*, vol. 55, no. 1, pp. 14-29, 2018.

[10] Y. Meng, S. Li, Y. Zhang, *et al.*, ”Cyber physical system security of smart home platform,” *Journal of Computer Research and Development*, vol. 56, no. 11, pp. 2349-2364, 2019.

[11] M. Moniruzzaman, S. Khezr, A. Yassine, *et al.*, ”Blockchain for smart homes: Review of current trends and research challenges,” in *Computers & Electrical Engineering*, vol. 83, 2020. (`https://doi.org/10.1016/j.compeleceng.2020.106585`)

[12] B. Patil, P. Vyas, R. K. Shyamasundar, ”SecSmartLock: An architecture and protocol for designing secure smart locks,” *International Conference on Information Systems Security*, pp. 24-43, 2018.

[13] S. Pinjala, S. Gupta, ”Remotely accessible smart lock security system with essential features,” in *International Conference on Wireless Communications Signal Processing and Networking*, 2019. DOI: 10.1109/WiSPNET45539.2019.9032715.

[14] S. J. Song, H. Nam, ”Visible light identification system for smart door lock application with small area outdoor interface,” *Current Optics and Photonics*, vol. 1, no. 2, pp. 90-94, 2017.

[15] Q. Sun, J. Liu, S. Li, *et al.*, ”Internet of things: Summarize on concepts, architecture and key technology problem,” *Journal of Beijing University of Posts and Telecom*, vol. 33, no. 3, pp. 1-9, 2010.

[16] S. Tang, Y. Yu, R. Zimmermann, *et al.*, ”Efficient geo-fencing via hybrid hashing: A combination of bucket selection and in-bucket binary search,” *ACM Transactions on Spatial Algorithms & Systems*, vol. 1, no. 2, 2015.

[17] Z. Tang, X. Li, ”The formalization description of the dolev-yao intruder model,” *Computer Engineering & Science*, vol. 32, no. 8, pp. 36-45, 2010.

[18] J. Wang, Y. Li, Y. Jia, *et al.*, ”Survey of smart home security,” *Journal of Computer Research and Development*, vol. 55, no. 10, pp. 2111-2124, 2018.

[19] S. Wang, Y. Liu, D. Li, *et al.*, ”A ferrofluid-based planar vibration energy harvester for smart lock of shared bicycle,” *International Journal of Applied Electromagnetics and Mechanics*, vol. 61, no. 2, pp. 293-300, 2019.

[20] Y. Wang, C. Zhang, D. Huo, *et al.*, ”A survey of security threats and defending technologies on IoT smart devices,” *Journal of Cyber Security*, vol. 3, no. 1, pp. 48-67, 2018.

[21] Z. Xin, L. Liu, G. Hancke, ”AACS: Attribute-based access control mechanism for smart locks,” *Symmetry*, vol. 12, no. 6, pp. 1050, 2020.

[22] W. Yan, Z. Wang, H. Wang, *et al.*, ”Survey on recent smart gateways for smart home: Systems, technologies, and challenges,” in *Transactions on Emerging Telecommunications Technologies*, 2020. (`https://doi.org/10.1002/ett.4067`)

[23] F. Yang, S. Wang, J. Li, *et al.*, ”An overview of internet of vehicles,” *China Communications*, vol. 11, no. 10, pp. 1-15, 2014.

[24] M. Ye, N. Jiang, H. Yang, *et al.*, ”Security analysis of internet-Of-things: A case study of August smart lock,” in *IEEE Conference on Computer Communications Workshops*, 2017. DOI: 10.1109/INFCOMW.2017.8116427.

[25] Y. Yuan, F. Wang, ”Blockchain: The state of the art and future trends,” *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481-494, 2016.

[26] L. Zhang, Y. Cui, J. Liu, *et al.*, ”Application of machine learning in cyberspace security research,” *Chinese Journal of Computer*, vol. 41, no. 9, pp. 1943-1975, 2018.

[27] Z. Zhu, Y. Cheng, ”Application of attitude tracking algorithm for face recognition based on openCV in the intelligent door lock,” *Computer Communications*, vol. 154, pp. 390-397, 2020.

Biography

Yonglei Liu is an associate professor in the school of computer and information engineering, Tianjin Chengjian University. In 2014, he received the Ph.D. degree from Tianjin University in computer application technology. His main research interests include computer network performance optimization, wireless network security, network protocol security analysis and network security evaluation.

Kun Hao received her M.S. degree from Tianjin University, Tianjin, China in 2006. In 2010, she received her Ph.D. degree in the School of Computer Science and Technology, Tianjin University. She is an Assoc. Prof. in the School of Computer and Information Engineering at Tianjin Chengjian University. Her research interests lie in underwater sensors network, wireless communications and networking, wireless sensor networks, network protocol and network optimization, and application of VR technology in architecture design.

Jie Zhao is an associate professor in the department of electronic information engineering, Tianjin Chengjian University. In 2015 he received the Ph.D. degree from Tianjin University in information and communication engineering, China. Since 2009, he has been working in the school of computer and information engineering, Tianjin Chengjian University. His current research interests include multimedia information security and computer vision.

Li Wang received PhD degree in optics from Nankai University. She currently is a lecturer of school of computer and information engineering, Tianjin chengjian university. Her research interests include the Internet of Things engineering and optical fiber sensing.

Weilong Zhang is an associate professor in the department of electrical and information engineering, Hebei Jiaotong vocational and technical college, master's degree. His current research interests include multimedia information security and near end wireless network communication technology.