applied sciences

MDPI

Article

SPCL: A Smart Access Control System That Supports Blockchain

Jiaxing Wu [1,†], Nian Xue [2,†], Zhen Li [3,4], Xianbin Hong [5], Yilin Zhao [2], Xin Huang [1,*] and Jie Zhang [6,*]

1 School of Computer Science and Technology, Taiyuan University of Technology, Jinzhong 030600, China
2 NYU Tandon School of Engineering, New York University, New York, NY 10012, USA; nx296@nyu.edu (N.X.); yz3505@nyu.edu (Y.Z.)
3 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; lizh0019@gmail.com
4 Shanghai Grandhonor Information Technology Co., Ltd., Shanghai 201210, China
5 Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK; derekgrant01@gmail.com
6 School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China
* Correspondence: xin@huangstudio.org (X.H.); jie.zhang01@xjtlu.edu.cn (J.Z.); Tel.: +86-15190086085 (X.H.)
† These authors contributed equally to this work.

Abstract: The access control system is a critical element in intelligent buildings. In this paper, we present SPCL, an innovative access control system designed to facilitate building entry through the use of mobile phones. Our system aims to provide a secure and convenient solution for building access, capitalizing on the widespread availability and capabilities of mobile devices. Additionally, we propose a lightweight authentication protocol to enhance security. The performance of the protocol is measured for different curves at different frequencies, proving that the protocol is more suitable for door lock systems than the benchmark protocol. In addition, we investigated the security and usability of SPCL. Finally, a comparison of the security of human-lock interfaces for smart locks and blockchain-based payment methods are discussed.

Keywords: access control; protocol; intelligent building; smart lock; blockchain

Citation: Wu, J.; Xue, N.; Li, Z.; Hong, X.; Zhao, Y.; Huang, X.; Zhang, J. SPCL: A Smart Access Control System That Supports Blockchain. *Appl. Sci.* **2024**, *14*, 2978. https://doi.org/10.3390/app14072978

Academic Editor: Gianluca Lax

Received: 8 February 2024
Revised: 27 March 2024
Accepted: 29 March 2024
Published: 1 April 2024

1. Introduction

A lock is the core of an entire access control system. The oldest known key-operated lock can be traced back to ancient Egypt, approximately 4000 years ago [1]. Locks are also indispensable to modern people's daily lives. For example, mechanical locks, which may be one of the most familiar objects to everyone, have been used for hundreds of years [2]. However, as the pace of modern life is accelerating, more features are expected from access control systems, and locks also evolve incessantly, especially in the field of intelligent buildings [3–5]. Consequently, a smart lock [6] is surfacing to improve life quality [7], while also decreasing the security risks and costs effectively [8]. Nowadays, smart locks have become a necessary part of intelligent buildings.

A huge amount of studies have been done in the domain of smart locks over the last couple of years. For instance, Park et al. [9] presented a smart digital door lock solution for home automation. Padmapriya and KalaJames [10] put forward an improved face recognition approach for a vehicle security system. Furthermore, Chang and Jiang studied a binary single-key-lock system [11], and Wu [12] studied a matrix-based lock system in order to enhance security further.

After entering the mobile internet age, more and more smart devices, e.g., smart locks, can be managed by mobile devices easily. For instance, Iftode et al. [13] designed a mobile phone based architecture that enables users to interact with embedded systems by cell phones. In [14], a remote monitoring intelligence system based on fingerprints through wireless transmitting and receiving, was introduced. In [15], the authors developed and implemented a remote lock system utilizing wireless communication on a smart phone by

a dedicated Android application. In 2007, Bauer et al. [16] deployed a demo smartphone-based system in university buildings, aiming to replace existing access control technologies. The authors in [17] proposed an access control system for intelligent buildings in 2016. In 2018, Patil et al. [18] proposed a security protocol between smartphones, smart locks, and cloud servers, and discussed possible security vulnerabilities. In 2021, Taslim et al. [19] proposed a smart home door lock system using security protocol in the Internet of Things (IoT) scenario is proposed, which has the problems of unencrypted data transmission and insufficient authentication methods. Ahmad et al. [20] developed an enhanced access control system for smart locks, using cloud and edge computing to improve authorization, focusing on scalability and performance. Unfortunately, the approach adds complexity in deploying and enforcing policies. In 2023, Guntur et al. [21] designed an IoT-enhanced smart door lock using Radio-Frequency Identification (RFID) for secure access, which is affordable, user-friendly, and includes a fire alarm for additional safety, but it's vulnerable to card loss attacks and unauthorized access. Mehmood et al. [22] proposed a smart disposable door lock system based on invisible touch sensors to prevent the carrying, loss and copying of cards.

Traditional smart lock solutions, while offering convenience, often harbor significant security flaws, including susceptibility to cyber-attacks, data breaches, and unauthorized access. These vulnerabilities primarily stem from their reliance on centralized databases and a lack of robust encryption methodologies. Additionally, the opaque nature of transaction and access logs within these systems has raised concerns regarding their trustworthiness and auditability—critical aspects for both residential and commercial applications. In recent years, there have been some blockchain-based solutions in IoT device scenarios [23–25], but these solutions are difficult to ensure communication security. The integration of blockchain smart contracts with authentication protocol, especially within the context of smart lock systems, signifies a paradigm shift towards more secure, transparent, and efficient operations. This paper presents a smart lock solution that supports blockchain, aimed at addressing the prevalent security gaps and operational inefficiencies inherent in conventional centralized smart lock solutions.

Building on the aforementioned studies, we suggest implementing a smartphone-based smart building access control system. It boasts high security and availability, while ensuring efficiency and user privacy. The main contributions of this paper can be outlined as follows:

(1) A novel access control system, named "Smart-Phone-Controlled-Lock" (SPCL), integrating smart lock technology with mobile phone-based access, has been designed and developed. A lightweight privacy protection protocol called the "SPCL protocol" is proposed and implemented. Its performance works best when the computing performance of the two devices is similar to each other.
(2) The proposed SPCL protocol is formally analyzed using the eCK security model to prove that SPCL achieves the required security goals. Other security features of the protocol are discussed as well.
(3) Survey approaches, including a questionnaire and a focus group, are used to study the attitudes of candidates towards access control systems and several unlocking mechanisms, e.g., biometric identification, password pad and sliding card. It should be a jumping-off point for future studies.
(4) Security comparison regarding human-lock interfaces and blockchain based payment methods for smart locks are discussed.

The remainder of the paper is organized as follows. Section 2 outlines the SPCL system and describes its security mechanism. Section 3 proposes the SPCL protocol and describes its process for establishing authentication. Section 4 provides a proof of the eCK model, as well as an analysis of the security properties of the proposed protocol. Section 5 investigates the performance of SPCL through extensive experiments. Section 6 describes the research methodology used in our survey as well as the survey results and discussion. Section 7 provides a more in-depth discussion of SPCL security mechanisms, popular unlocking

methods, and blockchain-based solutions. Section 8 compares SPCL with several of the most representative related works. Section 9 draws conclusions and outlines the future.

2. SPCL Overview

In this section, we introduce SPCL, a smart lock- and smartphone-based access control system that offers a convenient and secure solution for implementing access control in intelligent buildings. The security mechanism in the system is also studied.

2.1. SPCL Architecture

SPCL is a prototypical system which can be used to control smart locks through mobile applications. It is comprised of five primary parts: cloud server, mobile client, desktop/laptop client, switch and smart devices. The system architecture is shown in Figure 1.

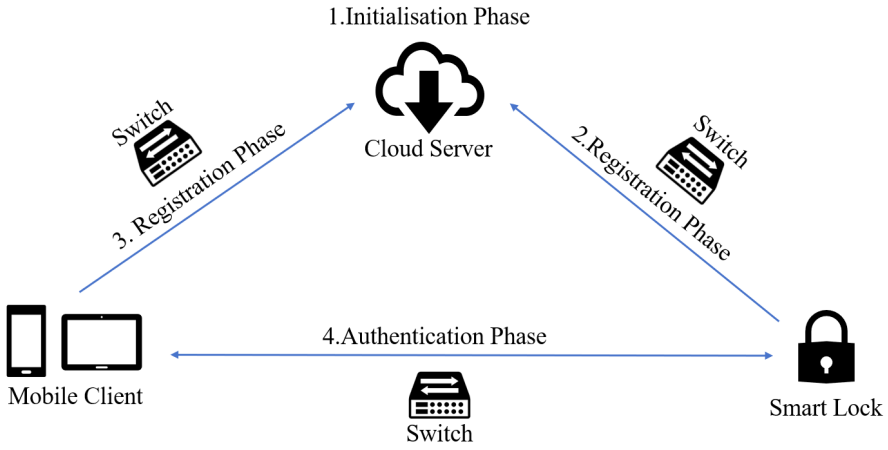

Figure 1. System architecture of SPCL.

A cloud server (CS) acts as a trusted third party. It is used for the registration of smart locks and mobile clients. It also stores registration information and unlocking log data.

A mobile client (MC) is often a smart phone or tablet PC. It can authenticate with the smart lock, after registering with the cloud server. It works like a key for the user to open the door.

Switches forward messages. They work like a bridge between the cloud server and the smart devices.

Smart devices refer to devices such as smart locks (SL), shared charging facilities and smart home appliances. They can connect to the cloud server or mobile client via a wireless channel.

2.2. Security Model

We utilize the eCK security model [26] to analyze our proposed scheme. Let $A, B, C, \cdots$ represent the communicating parties; n represents the total number of parties; sid represents the session identifier of a completed session; and $msg_{\mathcal{A}}, msg_{\mathcal{B}}, \cdots$ represent the messages sent by $A, B, \cdots$. The eCK security experiment is conducted by an adversary $\mathcal{M}$ against an AKE protocol π with a challenger $\mathcal{C}$, which captures real-world attacks through the queries issued by $\mathcal{M}$:

- $Send(A, B, msg_{\mathcal{A}}, msg_{\mathcal{B}})$: This query allows $\mathcal{M}$ to run the protocol by sending message $msg_{\mathcal{B}}$ to the session $(\mathcal{A}, \mathcal{B}, *, *)$. It returns the next message according to the protocol conversation so far.
- $EphemeralKeyReveal(sid)$: This query returns the ephemeral private keys held by the session sid to $\mathcal{M}$.
- $SessionKeyReveal(sid)$: This query return the session key for a session sid to $\mathcal{M}$.

- *LongTermKeyReveal*(A): This query returns the long-term private key of party A to $\mathcal{M}$.
- *Test*(sid) : At some point, $\mathcal{M}$ selects one session sid for a *Test* query. This session sid must be fresh. (defined in Definition 1).

Definition 1 (Fresh Session). *Let sid owned by an honest party $\mathcal{A}$ with peer $\mathcal{B}$, and $\mathcal{B}$ is also honest. Let sid* be the session identifier of the matching session (defined in Definition 2) of sid, if it exists. The session sid is fresh if none of the following conditions hold:*

(1) $\mathcal{M}$ issues SessionKeyReveal(sid) or SessionKeyReveal(sid*) query (if sid* exists);
(2) sid* exists and $\mathcal{M}$ makes one of the following queries:
- *LongTermKeyReveal(A) and EphemeralKeyReveal(sid), or*
- *LongTermKeyReveal(B) and EphemeralKeyReveal (sid*);*

(3) sid* does not exist and $\mathcal{M}$ makes one of the following queries:
- *LongTermKeyReveal(A) and EphemeralKeyReveal(sid), or*
- *LongTermKeyReveal(B)*

Definition 2 (Matching session). *The party executing the session is called the owner of the session and the other party is called the peer. Two sessions: sid owned by A with peer B and sid* owned by B with peer A, are matching sessions if all messages sent (or received) by A are identical to those received (or sent) by B.*

In the eCK security experiment, $\mathcal{M}$ plays with a challenger $\mathcal{C}$ of a hard problem as follows:

- Queries: $\mathcal{M}$ makes queries as defined aforementioned at will.
- Test: $\mathcal{M}$ chooses a fresh session as the *Test* session.
- Queries after test: $\mathcal{M}$ makes queries as in *Queries* step but cannot query the *Test* session.
- Guess: $\mathcal{C}$ flips a fair coin $b \in \{0,1\}$, and returns $\mathcal{M}$ with the session key held by sid if $b = 1$, or a random string of the same length otherwise. $\mathcal{M}$ outputs $b' \in \{0,1\}$.

$\mathcal{M}$ wins the eCK experiment if $b' = b$.

Definition 3 (eCK security). *An AKE protocol π is eCK-secure if the advantage for all probabilistic polynomial time (PPT) adversary $\mathcal{M}$ against π in winning the eCK experiment, denoted by*

$$\mathbf{Adv}_{\pi}^{\mathrm{AKE}}(\mathcal{M}) = Pr[\mathcal{M}\ wins] - \frac{1}{2},$$

is negligible.

2.3. Security Mechanisms

An elliptic curve based on the prime field F_q is defined as follow [27]:

$$y^2 \equiv x^3 + ax + b \pmod{p} \text{ where } 4a^3 + 27b^2 \neq 0$$

Let $E(F_q)$ be an additive group on the elliptic curve and G be a generator of $E(F_q)$. Hardness assumptions are defined as follows [27].

Definition 4 (Discrete Logarithm (DLOG) Problem). *Given $x \times G \in E(F_q)$, where $x \in \mathbb{Z}_n^*$, compute x.*

Definition 5 (Computational Diffie-Hellman (CDH) Problem). *Given $x \times G \in E(F_q)$, $y \times G \in E(F_q)$, where $x, y \in \mathbb{Z}_n^*$, compute $xy \times G$.*

Definition 6 (Decisional Diffie-Hellman (DDH) Problem). *Given $x \times G \in E(F_q), y \times G \in E(F_q), z \times G \in E(F_q)$, where $x, y, z \in \mathbb{Z}_n^*$, determine whether $xy \times G = z \times G$ or not.*

Definition 7 (Gap Diffie-Hellman (GDH) Problem)**.** *Given* $x \times G \in E(F_q), y \times G \in E(F_q)$, *where* $x, y \in \mathbb{Z}_n^*$, *as well as an oracle that solves the DDH problem on* $E(F_q)$, *compute* $xy \times G$.

3. The Proposed Scheme SPCL

This section introduces SPCL protocol, including the initialisation phase, the registration phase and the authentication phase. Notations and corresponding descriptions are given in Table 1.

Table 1. Symbols used in SPCL.

Notations	Description
F_q	Finite field with prime order q
$E(F_q)$	Additive group over finite field F_q
G	Base point of $E(F_q)$
$\mathbb{Z}_n^*$	The additive group of order n
u, v	Curve parameters
d_{SL}	Private key of smart lock
d_{MC}	Private key of mobile client
ID_{MC}, ID_{SL}	Identifiers for mobile client or smart lock
RID_{MC}	Pseudonym identifier for mobile client
P_{SL}	Public key of smart lock
P_{MC}	Public key of mobile client
$H_1(\cdot), H_2(\cdot)$	Cryptographic secure hash functions

3.1. Initialisation Phase

This part introduces the initialisation method [27]. The cloud server (CS) initialises system parameters: $\mathbb{Z}_n^*$, E, G, F_q, u, v, and $H_1 \rightarrow \{0,1\}^a$ and $H_2 \rightarrow \{0,1\}^b$.

3.2. Registration Phase

In the registration phase, the mobile client (MC) and the smart lock (SL) will register with the cloud server (CS).

(1) Smart lock registration.
CS generates an identity identifier for SL and calculates $P_{SL} = d_{SL} \times G$, $N_{SL} = H_1(ID_{SL}||u||v||G||P_{SL})$. CS stores $(ID_{SL}, d_{SL}, P_{SL}, N_{SL})$ in its registry and sends it to SL via secure channel.

$$CS \Longrightarrow SL : M_1 = (ID_{SL}, d_{SL}, P_{SL}, N_{SL})$$

(2) Registration information access.
The user uses MC to scan the QR code on SL and obtains $M_2 = (ID_{SL}, P_{SL}, N_{SL})$ from SL. MC stores (ID_{SL}, P_{SL}) in its registry. The process of scanning the QR is considered a secure channel.

$$SL \Longrightarrow MC : M_2 = (ID_{SL}, P_{SL})$$

(3) Mobile client registration.
① MC generates its unique identifier ID_{MC} and sends $M_3 = (ID_{MC}, ID_{SL}, P_{SL})$ to CS via secure channel.

$$MC \Longrightarrow CS : M_3 = (ID_{MC}, ID_{SL}, P_{SL})$$

② CS checks ID_{MC} and matches MC with ID_{MC}. If there is already MC matching ID_{MC}, CS checks whether ID_{SL} is in its registry; otherwise, CS picks a public-private key pair P_{MC} and d_{MC} for MC: CS picks a random number $d_{MC} \in \mathbb{Z}_n^*$, calculates $P_{MC} = d_{MC} \times G$, $N_{MC} = H_1(ID_{MC}||u||v||G||P_{MC})$ and checks whether ID_{SL} is in its registry.

③ After that, CS stores $(ID_{MC}, d_{MC}, P_{MC}, N_{MC}, ID_{SL}, P_{SL}, N_{SL})$ in its registry and sends it to MC.

$$CS \Longrightarrow MC : M_4 = (ID_{MC}, d_{MC}, P_{MC}, N_{MC}, ID_{SL}, P_{SL}, N_{SL})$$

④ SL receives $(ID_{MC}, P_{MC}, N_{MC})$ from CS and completes the registration phase of MC.

$$CS \Longrightarrow SL : M_5 = (P_{MC}, N_{MC})$$

3.3. Authentication Phase

In the authentication phase, the mobile client (MC) establishes authentication with the smart lock (SL).

(1) Requesting authentication.
① MC picks a random value $r_{MC} \in \mathbb{Z}_n^*$ and computes $R_{MC} = r_{MC} \times G = (x_{MC}, y_{MC})$. MC chooses the identifier of the smart lock you want to open. SL picks a random value $r_{SL} \in \mathbb{Z}_n^*$ and computes $R_{SL} = r_{SL} \times G = (x_{SL}, y_{SL})$.
② MC generates the current time-stamp T_{MC} and computes $RID_{MC} = Enc_{P_{SL}}(ID_{MC} ||N_{SL}||R_{MC})$. MC sends message $M_1 = (RID_{MC}, T_{MC}, R_{MC})$ to SL via wireless channels.

$$MC \rightarrow SL : M_1 = (RID_{MC}, T_{MC}, R_{MC})$$

(2) Responding to the mobile client.
① Upon receiving M_1, SL verifies T_{MC} and decrypts RID_{MC} as $(ID_{MC} ||N_{SL}||R_{MC}) = Dec_{d_{SL}}(RID_{MC})$. SL computes $t_{SL} = (d_{SL} + r_{SL})\ mod\ n$ and $V = t_{SL} \times (P_{MC} + R_{MC}) = (x_V, y_V)$.
② After that, SL calculates $S_{SL} = H_1(y_V||H_1(x_V||N_{SL}||N_{MC}||x_{SL}||y_{SL}||x_{MC}||y_{MC}))$ and the current time-stamp T_{SL}.
③ Then, SL sends message $M_2 = (T_{SL}, R_{SL}, S_{SL})$ to MC.

$$SL \rightarrow MC : M_2 = (T_{SL}, R_{SL}, S_{SL})$$

(3) Accepting the mobile client's session key.
① Upon receiving M_2, SL verifies T_{SL} and R_{SL}. If the verification succeeds, MC calculates $t_{MC} = (d_{MC} + r_{MC})\ mod\ n$ and $U = t_{MC} \times (P_{SL} + R_{SL}) = (x_U, y_U)$.
② Then, MC calculates $S_1 = H_1(y_V||H_1(x_V||N_{SL}||N_{MC}||x_{SL}||y_{SL}||x_{MC}||y_{MC}))$ and checks whether $S_1 = S_{SL}$. If verification is successful, MC computes $S_{MC} = H_1(y_U||H_1(x_U||N_{MC}||N_{SL}||x_{MC}||y_{MC}||x_{SL}||y_{SL}))$ and the session key $SK_{MC} = H_2(x_U\ ||y_U||N_{MC}||N_{SL})$. Then, MC sends message $M_3 = (S_{MC})$ to SL.

$$MC \rightarrow SL : M_3 = (S_{MC})$$

(4) Accepting the smart lock's session key.
Upon receiving M_3, SL computes $S_2 = H_1(y_V||H_1(x_V||N_{MC}||N_{SL}||x_{MC}||y_{MC}||x_{SL}||y_{SL}))$. Then SL checks whether $S_2 = S_{MC}$. If verification is successful, SL calculates the session key $SK_{SL} = H_2(x_V||y_V||N_{MC}||\ N_{SL})$. Thus, MC and SL successfully establish session keys.

4. Security Analysis

4.1. Provable Security of SPCL Protocol

The eCK model [26] is used to conduct a provable security proof for the SPCL protocol. The practical reasons for each step in the proof process are based on the formal proof process followed by the NAXOS protocol.

Theorem 1. *The SPCL protocol eCK-security if $\mathcal{H}_2$ is modeled as independent random oracles and GDH and DLOG problems are hard in G.*

For any PPT adversary $\mathcal{M}$ against SPCL that runs in time at most t, involves at most n honest parties, and activates at most k sessions, there exists a GDH problem solver $\mathcal{S}$ and a DLOG problem solver $\mathcal{T}$ such that

$$\mathbf{Adv}_{\mathcal{M}}^{\mathrm{SPCL}} \leq max\left\{k^2, nk\right\}\left(2\mathbf{Adv}_{\mathcal{S}}^{GDH} + 2n \cdot \mathbf{Adv}_{\mathcal{T}}^{DLOG} + O\left(\frac{k^2}{2^{\lambda}}\right)\right)$$

where $\mathcal{S}$ runs in time $O(tk)$ and $\mathcal{T}$ runs in time $O(t)$.

Proof. Let $\mathcal{M}$ be any AKE adversary against SPCL protocol. The session key of the *Test* session is computed as $sk = H_2(\gamma)$ for a 4-tuple γ. We will demonstrate how to use $\mathcal{M}$'s ability to construct a solver $\mathcal{S}$ for solving the GDH problem. $\mathcal{S}$ first selects the system parameter $params = \left\{E(F_q), G, H_2\right\}$, and then sends *params* to $\mathcal{M}$. As $\mathcal{H}_2$ is a random oracle, $\mathcal{M}$ can distinguish a session key $sk = H_2(\gamma)$ from a random string with a significantly higher probability than $\frac{1}{2}$ in one of the following events:

E_1 Guessing attack: $\mathcal{M}$ correctly guesses the *sk*. The probability of guessing *sk* is $\mathcal{O}\left(\frac{1}{2^{\lambda}}\right)$, which is negligible.

E_2 Key replication: If an adversary $\mathcal{M}$ forces two distinct non-matching sessions to have the same session key, it can select one of the sessions as the *Test* session and query the key of the other session. This is because two non-matching sessions cannot have the same communicating parties and ephemeral public keys. The replication of keys is equivalent to finding a collision for the hash function $\mathcal{H}_2$. Therefore, the probability of **Event** E_2 occurring is $\mathcal{O}\left(\frac{s(\lambda)^2}{2^{\lambda}}\right)$, which is negligible.

E_3 Forging attack: Adversary $\mathcal{M}$ queries $\mathcal{H}_2$ on the value $(N_{\mathcal{A}}, N_{\mathcal{B}}, x_{\mathcal{A}}, y_{\mathcal{A}})$ ($K_{\mathcal{A}} = (x_{\mathcal{A}}, y_{\mathcal{A}})$) in the *Test* session owned by $\mathcal{A}$ communicating with $\mathcal{B}$. We will construct a GDH solver using an adversary $\mathcal{M}$ that succeeds in a forging attack with non-negligible probability. $\mathcal{S}$ simulates the game outlined above. During the game, $\mathcal{S}$ has to answer all queries of the adversary $\mathcal{M}$. The following two sub-events should be considered.

$E_{3.1}$ The *Test* session has a matching session owned by another honest party.

$E_{3.2}$ No honest party owns a session matching with the *Test* session.

□

4.1.1. The Analysis of Event $E_{3.1}$

$\mathcal{S}$ selects matching sessions executed by honest parties $\mathcal{A}$ and $\mathcal{B}$ at random. If two matching sessions are selected, $\mathcal{S}$ proceeds with probability $\frac{2}{k^2}$. $\mathcal{S}$ generates $d_{\mathcal{A}}$, $r_{\mathcal{A}}$ and $d_{\mathcal{B}}$, $r_{\mathcal{B}}$. $\mathcal{S}$ sets $msg_{\mathcal{A}} \leftarrow X_{\mathcal{A}}$ (instead of $(r_{\mathcal{A}} + d_{\mathcal{A}}) \times P$) and $msg_{\mathcal{B}} \leftarrow Y_{\mathcal{B}}$ (instead of $(r_{\mathcal{B}} + d_{\mathcal{B}}) \times P$). With a probability of $\frac{1}{k^2}$, $\mathcal{M}$ chooses one of the selected sessions as a *Test* session and the other as its matching session. If $\mathcal{M}$ wins in this event, $\mathcal{S}$ can solve the CDH problem. The session key *sk* for the selected session is supposed to be $H_2(\gamma)$, where the 4-tuple γ includes the value $CDH(X_{\mathcal{A}}, Y_{\mathcal{B}})$. In order to win, $\mathcal{M}$ must have queried γ to the random oracle $\mathcal{H}_2$.

If the selected session is indeed the *Test* session, $\mathcal{M}$ is permitted to reveal a subset of $\{r_{\mathcal{A}}, r_{\mathcal{B}}, d_{\mathcal{A}}, d_{\mathcal{B}}\}$. But $\mathcal{M}$ is not allowed to reveal both $(r_{\mathcal{A}}, d_{\mathcal{A}})$ or $(r_{\mathcal{B}}, d_{\mathcal{B}})$. The only way for $\mathcal{M}$ to differentiate between this simulated eCK experiment and a real eCK experiment is

by querying $(r_{\mathcal{A}}, d_{\mathcal{A}})$ or $(r_{\mathcal{B}}, d_{\mathcal{B}})$ (this way, $\mathcal{M}$ will discover that $msg_{\mathcal{A}}$ and $msg_{\mathcal{B}}$ were not computed correctly). Probability that $\mathcal{M}$ makes such queries is at most

$$2n \cdot \mathbf{Adv}_{\mathcal{T}}^{\mathrm{DLOG}}$$

Therefore, we have

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{GDH}} \geq \frac{2}{k^2} \cdot \mathbf{Adv}_{\mathcal{M}}^{\mathrm{SPCL}} - 2n \cdot \mathbf{Adv}_{\mathcal{T}}^{\mathrm{DLOG}} - O\left(\frac{k^2}{2^{\lambda}}\right)$$

4.1.2. The Analysis of Event $E_{3.2}$

If $\mathcal{M}$ selects a *Test* session for which no matching session exists, $\mathcal{S}$ modifies the experiment as follows.

$E_{3.2.1}$: $\mathcal{S}$ randomly selects an eCK session in which $\mathcal{B}$ is the owner.

$\mathcal{S}$ picks a random party $\mathcal{B}$, and sets $Q_B \leftarrow X_B$ as its long-term public key. Note that $\mathcal{S}$ doesn't know long-term private key $DLOG(X_B)$ corresponding to X_B. Therefore, it cannot effectively simulate eCK sessions executed by $\mathcal{B}$. $\mathcal{S}$ sets $msg_{\mathcal{B}} = h_B \cdot P$ instead of $(r_{\mathcal{B}} + DLOG(X_B)) \times P$. $\mathcal{S}$ sets a session key sk ($\mathrm{H}_2(CDH(r_{\mathcal{B}} \times P + X_B, msg_{\mathcal{A}}), \cdot, \cdot)$) to be a random value. $\mathcal{S}$ can manage session key and ephemeral secret key reveals by making $SessionKeyReveal(\cdot)$ and $EphemeralKeyReveal(\cdot)$ queries.

Assuming $\mathcal{A}$ is an adversary-controlled party, $\mathcal{M}$ can compute the session key, reveal the session key sk, and detect that it is fake. To mitigate this issue, $\mathcal{S}$ monitors $\mathcal{M}$'s random oracle queries, and if $\mathcal{M}$ ever queries $(x, y, \cdot, \cdot)$ to H_2 (for some $Z \in G$, $Z = (x, y)$), $\mathcal{S}$ checks if $DDH(r_{\mathcal{B}} \times P + X_B, msg_{\mathcal{A}}, Z) = 1$, and if yes, replies with the session key sk. Similarly, while computing sk, $\mathcal{S}$ checks if sk matches any previous response from the random oracle. $\mathcal{M}$ cannot detect that it is in the simulated eCK experiment unless it queries $\mathcal{H}_2$ or reveals $\mathcal{B}$'s long-term private key using the $LongTermKeyReveal(B)$ query. The first event reveals $DLOG(X_B)$, and allows $\mathcal{S}$ to solve the CDH problem, which occurs with probability at most

$$n \cdot \mathbf{Adv}_{\mathcal{T}}^{\mathrm{DLOG}}$$

$E_{3.2.2}$: $\mathcal{S}$ also randomly selects an eCK session in which $\mathcal{B}$ is the peer.

Let the owner of this session be denoted by $\mathcal{A}$. $\mathcal{S}$ normally generates $d_{\mathcal{A}}$, $r_{\mathcal{A}}$ and $d_{\mathcal{B}}$, $r_{\mathcal{B}}$. Then, $\mathcal{S}$ sets $msg_{\mathcal{A}} \leftarrow X_{\mathcal{A}}$ (instead of $(r_{\mathcal{A}} + d_{\mathcal{A}}) \times P$ and $msg_{\mathcal{B}} \leftarrow Y_{\mathcal{B}}$ (instead of $(r_{\mathcal{B}} + d_{\mathcal{B}}) \times P$). With probability at least $\frac{1}{nk}$ (where $\frac{1}{n}$ is to pick the correct party $\mathcal{B}$, and $\frac{1}{k}$ is to pick the correct session), $\mathcal{M}$ picks the selected session as the *Test* session and solves the CDH problem if it wins. $\mathcal{M}$ is not allowed to reveal both $(r_{\mathcal{A}}, d_{\mathcal{A}})$, and cannot corrupt $\mathcal{B}$. In this event, the only way that $\mathcal{M}$ can distinguish this simulated eCK experiment from a true eCK experiment is if $\mathcal{M}$ queries $(r_{\mathcal{A}}, d_{\mathcal{A}})$. By **Event** $E_{3.1}$ it happens with probability at most

$$n \cdot \mathbf{Adv}_{\mathcal{T}}^{\mathrm{DLOG}}$$

Overall, the success probability of $\mathcal{S}$ is at most

$$\mathbf{Adv}_{\mathcal{S}}^{\mathrm{GDH}} \geq \frac{1}{nk} \cdot \mathbf{Adv}_{\mathcal{M}}^{\mathrm{SPCL}} - O\left(\frac{k^2}{2^{\lambda}}\right) - 2n \cdot \mathbf{Adv}^{\mathrm{DLOG}}(\mathcal{T})$$

Therefore, we have:

$$\mathbf{Adv}_{\mathcal{M}}^{\mathrm{SPCL}} \leq max\left\{k^2, nk\right\}\left(2\mathbf{Adv}_{\mathcal{S}}^{GDH} + 2n \cdot \mathbf{Adv}_{\mathcal{T}}^{DLOG} + O\left(\frac{k^2}{2^{\lambda}}\right)\right)$$

Finally, under the GDH assumption, $\mathbf{Adv}_{\mathcal{S}}^{\mathrm{GDH}}$ is negligible. Therefore, $\mathbf{Adv}_{\mathcal{M}}^{\mathrm{SPCL}}$ is negligible, and SPCL protocol has eCK security.

4.2. Security Properties

Regarding security properties, this paper thoroughly examines the security properties of the SPCL protocol and highlights the advantages it offers compared to baseline protocols.

4.2.1. Perfect Forward Security

In our scheme, even if $\mathcal{M}$ can query the long-term secrets of ID_{MC}, d_{SL} and N_{SL}, $\mathcal{M}$ is unable to compute the session key SK_{MC} without the ephemeral secrets of r_{MC} or r_{SL}.

4.2.2. Identity Anonymity

In our scheme, MC uses the pseudonym identifier RID_{MC} so that $\mathcal{M}$ cannot obtain ID_{MC} from the message transmitted by MC. This prevents potential identity leaks and protects user privacy.

4.2.3. Man-in-the-Middle Attack

$\mathcal{M}$ has the ability to impersonate one of the communicating parties as well as eavesdrop on the message. In our scheme, $\mathcal{M}$ is unable to compute (S_{MC}, S_1) or (S_{SL}, S_2), and $\mathcal{M}$ is unable to verify that $S_1 = S_{SL}$ and $S_2 = S_{MC}$ and thus establish communication.

4.2.4. Impersonation Attack

Assume that $\mathcal{M}$ successfully eavesdrops on the long-term key (d_{MC} or d_{SL}). Since the ephemeral secrets cannot be obtained, $\mathcal{M}$ cannot generate the messages (M_1,M_3) or M_2 and pass from checking ($S_1 = S_{SL}$ and $S_2 = S_{MC}$). Therefore, impersonation attacks can be eliminated.

4.2.5. Replay Attack

In our scheme, there are time stamps (T_{MC}, T_{SL}) and random numbers (r_{MC}, r_{SL}). In order to implement a replay attack, $\mathcal{M}$ must forge the timestamp to pass the verification, and still needs to obtain the long-term private keys and temporary secrets of MC and SL. It is extremely difficult for $\mathcal{M}$ to crack and obtain this information, so the probability of a successful replay attack is extremely low and can almost be ignored.

4.2.6. Advantage of SPCL

Firstly, the SPCL protocol provides perfect forward secrecy, in contrast to the Llakep [28] and PSLA [27]. This secrecy ensures the protection of users' ephemeral secrets in scenarios involving smartphone-controlled locks. Secondly, the SPCL protocol surpasses the Llakep [28], ESEAP [29], Xie et al. [30], and SM2 [31] in terms of user anonymity. This highlights the SPCL protocol's greater emphasis on the privacy and security of users. In conclusion, the SPCL protocol is more suitable for the smart phone controlled lock system.

5. Implementation and Evaluation

In the implementation and evaluation section, the performance of the SPCL protocol is evaluated. During our experiment, we used two Raspberry Pis as the smart lock and the mobile client, respectively. The primitive access system is shown in Figure 2 below.

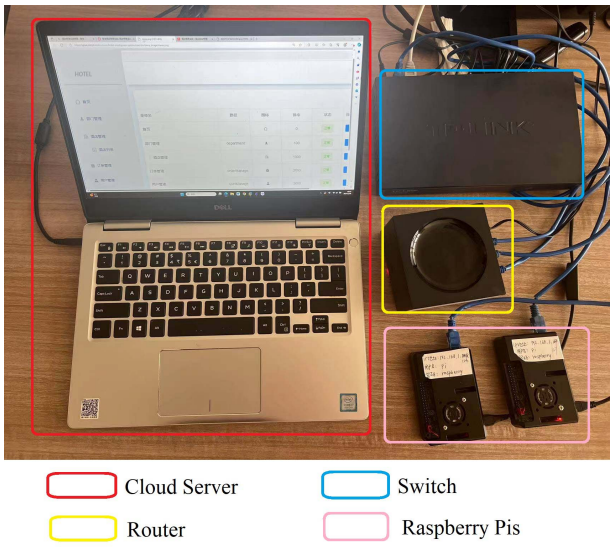

Figure 2. A primitive implementation of the proposed SPCL system.

5.1. System Setup

We have implemented an online hotel reservation system using a combination of Windows, Apache, MySQL, and PHP. Through our system, users can conveniently manage and control smart locks associated with their reservations using mobile clients. Table 2 provides a comprehensive overview of the features available in our experimental system.

Table 2. Features of SPCL system.

Component	Specification	Detail
Cloud Server Remote Controller (Aliyun)	CPU	2.6 GHz (Intel Xeon E5-2650)
	Memory	8G
	Hard Disk	120G
	OS	Windows Server 2008 R2 (64 bit)
	Location	Hangzhou (Internet)
Mobile Client (Raspberry Pi)	CPU	1.2 GHz (ARM Cortex-A53)
	OS	Raspbian GNU/Linux 10
	Memory	1 GB
Smart Lock (Raspberry Pi)	CPU	1.2 GHz (ARM Cortex-A53)
	OS	Raspbian GNU/Linux 10
	Memory	1 GB
Router	TP-link Router	
Switch	TP-link Switch	

5.2. Theoretical Analysis

Let T_{SM}, $T_{E/D}$, T_{KDF}, T_H and T_X represent elliptic curve point scalar multiplication operation time, encryption and decryption time, key derivation function time, hash, bitwise XOR operation time. We run the above operations in a simulated environment. The results of the simulation is that $T_{SM} \approx 33.098$ ms, $T_{E/D} \approx 0.301$ ms, $T_{KDF} \approx 8.930$ ms, $T_H \approx 0.133$ ms and $T_X \approx 0.015$ ms. The performance comparison between the SPCL protocol and the baseline protocol is shown in Table 3.

Table 3 shows that SPCL requires the least computational cost in SL, so the final total running time is the shortest. Regarding the computational cost for MC, SPCL ranks as having the second lowest.

Table 3. Comparison of computational costs and running time.

Schemes	Mobile Client (MC)	Smart Lock (SL)	Running Time
2PAKEP [32]	$3T_{SM} + T_{KDF} + 6T_H + 2T_X$ $\approx$ 107.01 ms	$3T_{SM} + T_{KDF} + 4T_H + 3T_X$ $\approx$ 108.46 ms	$6T_{SM} + 2T_{KDF} + 10T_H + 5T_X$ $\approx$ 254.91 ms
Llakep [28]	$2T_{SM} + T_{KDF} + 6T_H + 2T_X$ $\approx$ 73.88 ms	$4T_{SM} + T_{KDF} + 4T_H + 3T_X$ $\approx$ 141.07 ms	$6T_{SM} + 2T_{KDF} + 10T_H + 5T_X$ $\approx$ 240.38 ms
ESEAP [29]	$3T_{SM} + 2T_{E/D} + 12T_H + 5T_X$ $\approx$ 110.49 ms	$3T_{SM} + 2T_{E/D} + 9T_H + 4T_X$ $\approx$ 103.47 ms	$6T_{SM} + 4T_{E/D} + 21T_H + 9T_X$ $\approx$ 229.08 ms
Xie [30]	$3T_{SM} + 6T_H + 2T_X$ $\approx$ 97.71 ms	$3T_{SM} + 2T_{E/D} + 5T_H + T_X$ $\approx$ 101.13 ms	$6T_{SM} + 2T_{E/D} + 11T_H + 3T_X$ $\approx$ 219.84 ms
SM2 [31]	$2.5T_{SM} + T_{KDF} + 2T_H$ $\approx$ 82.80 ms	$2.5T_{SM} + T_{KDF} + 2T_H$ $\approx$ 84.28 ms	$5T_{SM} + 2T_{KDF} + 4T_H$ $\approx$ 192.84 ms
PSLA [27]	$1.5T_{SM} + T_{KDF} + 5T_H$ $\approx$ 51.03 ms	$3.5SM + T_{KDF} + 5T_H$ $\approx$ 116.61 ms	$5SM + 2T_{KDF} + 10T_H$ $\approx$ 185.58 ms
SPCL	$\mathbf{2T_{SM} + T_{E/D} + 2T_H}$ $\approx$ **66.59** ms	$\mathbf{2T_{SM} + T_{E/D} + 2T_H}$ $\approx$ **67.33** ms	$\mathbf{4T_{SM} + 2T_{E/D} + 4T_H}$ $\approx$ **148.23** ms

5.3. Experiment I

In Experiment I, the frequencies of two Raspberry Pis are set to 1.2 GHz each. During the experiment, the running time and calculation time of each protocol under five elliptic curves were measured ten times and the average values were calculated. The experimental results under the elliptic curve P-256 are shown in Figure 3. We find that the running time and calculation time of SPCL are the lowest among all elliptic curves.

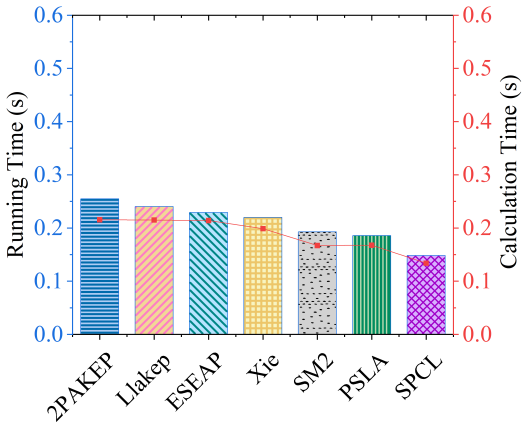

Figure 3. Comparison of average computational and running time under balanced computing power.

5.4. Experiment II

In Experiment II, the frequency of the mobile client Raspberry Pi was changed to 0.6 GHz, 0.7 GHz, 0.8 GHz, 0.9 GHz, 1.0 GHz, and 1.1 GHz, and then each protocol was tested in different elliptic curves. The average values are calculated after ten running times and calculation times. The experimental results on elliptic curve P-256 are shown in Figure 4 and show that SPCL calculation time is the shortest in different curves and different frequencies. SPCL has the shortest running time under all curves when the Raspberry Pi frequency is set to 0.8/1.2 GHz, 0.9/1.2 GHz, 1.0/1.2 GHz, and 1.1/1.2 GHz, respectively. Under all curves, the Raspberry Pi frequency is 0.7/1.2 GHz and 0.6/1.2 GHz, and the running time of SPCL is second only to PSLA.

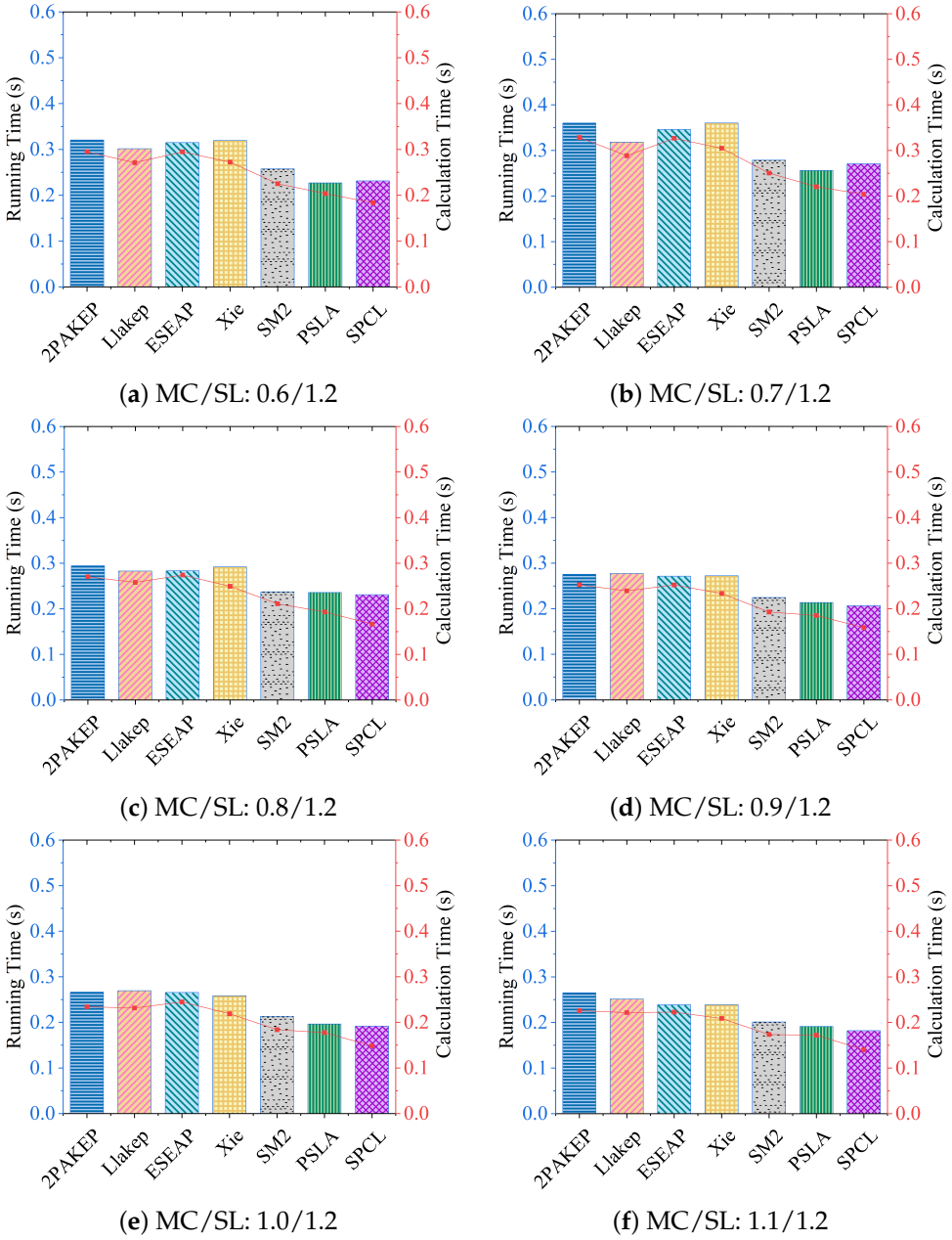

Figure 4. Comparisons of average computational and running time in different curves and different frequencies.

5.5. Summary

Based on experiments and comparisons with other solutions, it can be observed that SPCL exhibits higher performance advantages as the computing power of mobile clients improves. Especially in the scenario where the computing power of both communicating parties is balanced, the SPCL protocol shows the best performance. In common door lock systems, the balanced computing power of the mobile client and the smart lock indicates that the SPCL protocol is more suitable for the hotel reservation system.

6. Survey and Focus Group

The survey mainly focuses on the following research questions:

(i) What are the important features of smart locks?
(ii) Which unlocking method is preferred?

It is hypothesized that:

(i) Security is the most important feature of lock.
(ii) The fingerprint-based unlocking method is the most desirable unlocking method.

6.1. Quantitative Methodology

In order to verify the hypotheses, we utilized a questionnaire as the primary research method. The questionnaire used in the survey is an online one with eighteen questions. The kinds of questions contain multiple choice, demographic, dichotomous and Likert Scale questions. Since the subjects are principally Chinese, in order to prevent the language barrier, we translated the questionnaire into Chinese. On average, it took two minutes for the voluntary participants to answer the questions. Furthermore, all the respondents remained anonymous in order to protect their privacy.

The samples are composed of 45 (37%) females and 77 males (63%), having different backgrounds, such as occupation and education levels. The pie chart below (Figure 5) illustrates the occupational composition of these samples. Table 4 below displays whether they know smart locks and whether they accept smart locks.

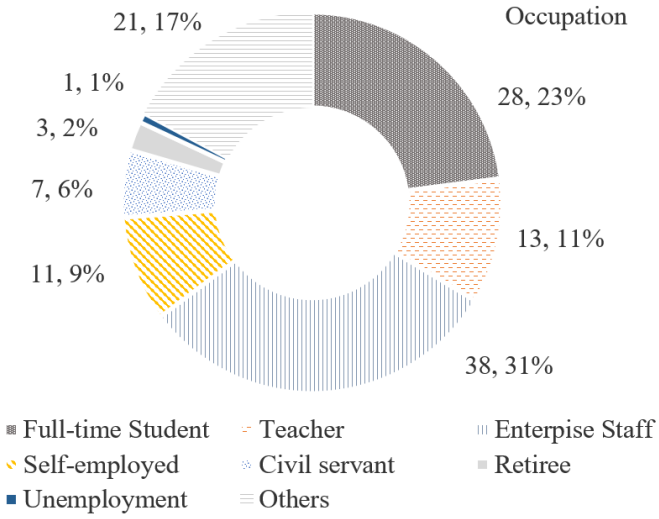

Figure 5. Composition of the participant's occupation.

Table 4. Acceptance of smart locks from different Age Groups.

Group No.	Age Group	#Considering Using Smart Locks (%)	#Respondents
G1	Under 18 years	1 (50%)	2
G2	18∼25 years	15 (62.5%)	24
G3	26∼30 years	17 (70.83%)	24
G4	31∼40 years	30 (66.67%)	45
G5	41∼50 years	19 (90.48%)	21
G6	51∼60 years	3 (60%)	5
G7	Over 60 years	1 (100%)	1

6.2. Qualitative Methodology

We used the focus group as an auxiliary research approach. A focus group is a recognized and valuable qualitative research approach that empowers researchers to establish causal relationships and delve into individuals' subjective perceptions of their own experiences. Eight people (one teacher and seven students) took part in our focus group for about 45 min. Table 5 shows the basic characteristics of these eight candidates in our focus group. We use "♂" to represent male and "♀" to represent female. Our task was to conduct the discussion and develop sub-questions derived from the two basic questions mentioned above. The entire meeting conversation was recorded on a smartphone.

Table 5. Individual background in the focus group.

Group No.	Age	Gender	Occupation
I1	33 years	♂	teacher
I2	34 years	♂	student
I3	27 years	♂	student
I4	22 years	♂	student
I5	22 years	♂	student
I6	28 years	♀	student
I7	22 years	♀	student
I8	22 years	♀	student

6.3. Questionnaire

It is hypothesized that security is the most important factor when people select a lock between traditional and smart ones. Figure 6 below shows a histogram of the factors that will affect informants' choices when they buy new locks. The result illustrates that security (86.89%) is the first important factor of all.

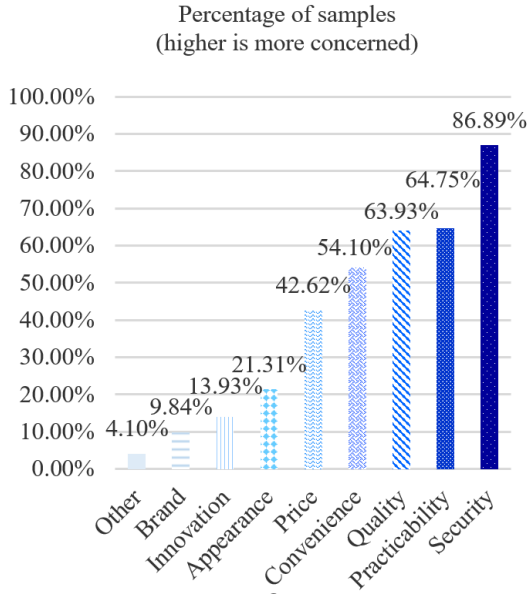

Figure 6. Factors affecting people's choice of smart locks.

In the following assessment questions, informants were asked to give marks (1 lowest, 5 highest) to both smart locks and traditional locks. The average mark of each aspect can be distinctly seen in Figure 7. As expected, smart locks acquire a better mark in the aspects of security.

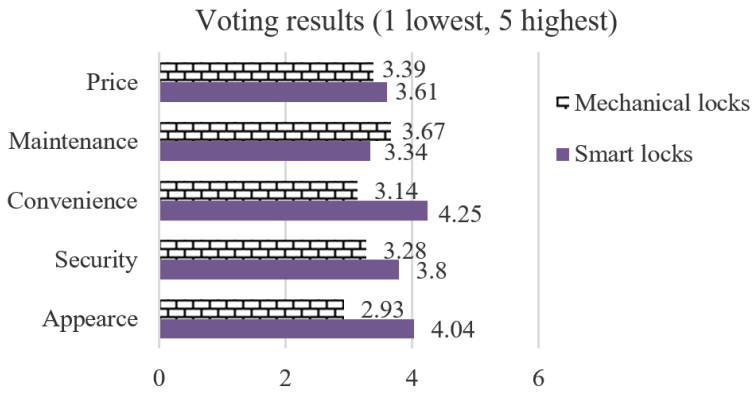

Figure 7. Comparison of smart locks and mechanical locks.

Next, we let the participants give a mark for a series of unlocking styles. Figure 8 provides the average marks for each method. Fingerprint is the only one whose value is over 4. Facial recognition obtains 3.8 points, which is in the second place, followed by smart mobile devices.

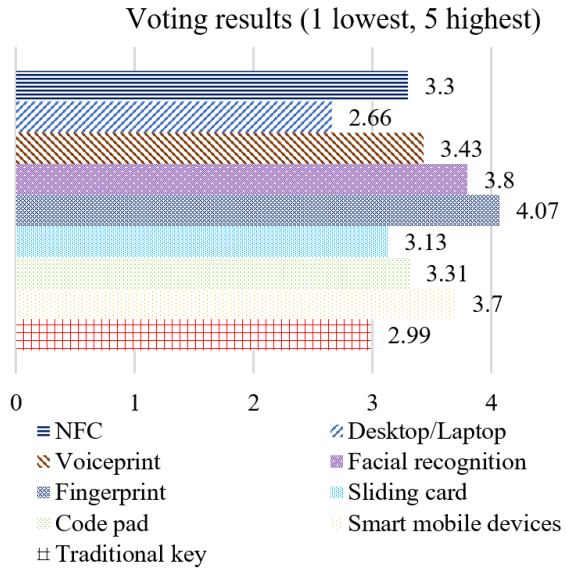

Figure 8. Comparison of various unlocking styles.

Finally, Figure 9 indicates that the biometric unlocking style is the preferable method compared with the other two methods. In other words, people are more prone to using patterns rather than passwords to unlock doors.

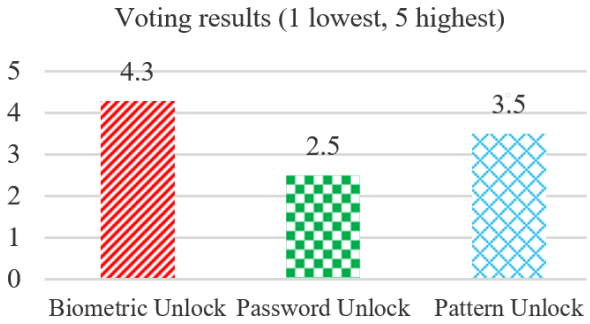

Figure 9. Comparison of different unlocking styles by using smart phone.

6.4. Focus Group and Discussion

All of the group participants' answers to smart locks were very meaningful and positive. A heated debate on the two topics mentioned previously was well organized and developed. The pros and cons of security mechanisms used in smart locks are analysed as well.

The first goal of the research was to wonder if security is significant in the eyes of modern people. Participants in this group totally agreed that security was the most important factor, and they believed that the smart lock had a high level of security.

The second purpose was to discover the most popular unlocking style. The results show that people like fingerprint unlocking style most. However, some focus group members expressed their worries about losing their fingerprints. Losing personal finger information might incur bigger risks rather than losing a key. Furthermore, one participant put forward a viewpoint that a fingerprint can be stolen and duplicated easily in our technologically advanced society.

The last aim of this focus group was to investigate people's opinions towards "smartphone-controlled-lock". Most of the participants were more likely to use smart phones to control the smart locks owing to their expandability and convenience. Based on the survey results, we have proposed the SPCL solution, which utilizes both smartphones and smart locks for authentication. Furthermore, we have taken into consideration the concerns expressed by group members regarding fingerprint authentication and have incorporated a combination of smartphone passcodes and hardware authentication methods. This approach ensures security while enhancing user usability.

In addition, the focus group participants highlighted some detailed cases of security mechanisms on smart locks. For example, the smart lock should send a message to the host as a notification if the password of the smart lock is changed; some advanced smart locks even own a monitor that can take photos or record video. These powerful functionalities further enhance security, and we will prioritize their consideration in future improvements to SPCL.

7. Discussion

Security is vitally essential for access control systems like SPCL. In this section, a deep and detailed discussion of the unlocking methods is analyzed. Additionally, solutions supporting blockchain are introduced in detail.

7.1. Security Comparisons

This section of this paper summarizes a few common security mechanisms of unlocking methods that can be applied to the SPCL system to control smart locks conveniently.

7.1.1. Password

Password is the oldest and most basic encryption scheme, which is still continuing to be used widely. It is very common to set up a series of particular characters that are known by oneself to ensure security. There is no doubt that a long password means high security. However, it would lose the convenience when high security is guaranteed. Security and convenience appear to be conflicted with each other. If the length is short and has limited characters, it will be easy to remember and vice versa. Figure 10 shows a typical password input interface. In terms of the focus group's conclusion, the favorable length of a password is six characters, and people tend to use only numbers to comprise their own passwords. When faced with higher security or more convenience, the user seems to have only one choice.

Figure 10. Password panel.

7.1.2. Pattern

As another prevalent unlocking method, pattern unlock is also welcomed by the public. It largely improves both security and convenience. After all, drawing a pattern is much faster and simpler than inputting a long string of characters. Generally speaking, pattern unlock has found an ideal tradeoff between security and convenience. For another

thing, compared with long passwords consisting of numbers and alphabets, the current pattern cipher strength is weaker and easier to break. For example, in Figure 11 the number of available Android unlock pattern (3×3) is 389,112 [33]. Another potential issue is that the path of a certain pattern is fixed; the pattern (generally nine points) can be traced by the remaining trail on the screen. In spite of weaker security, according to the survey result, people prefer patterns to codes. Perhaps it is easier to remember a simple pattern than a series of characters.

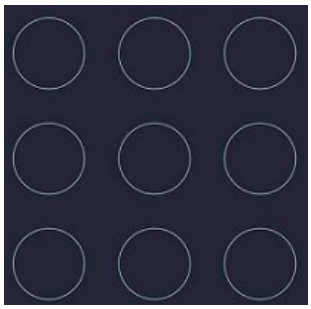

Figure 11. Pattern panel.

7.1.3. Knock Code

Knock code (see Figure 12) refers to allowing users to unlock their devices by tapping the quadrants of the screen in a sequence. The system will automatically record the user's knock sequence as the unlock password. One can knock on any position of the screen, even if the screen is off. This could be another solution that offers both security and convenience if a long enough tapping order is set. This innovative unlock style can also be easily used in our SPCL system to enhance security.

Figure 12. Knock panel.

7.2. Supporting Blockchain

In this section, we proposed a blockchain based smart lock system. Suppose a user wants to pay and stay at a hostel for just one night. For SPCL, smart locks could be updated to support blockchains and crypto-currency. In this case, the user can directly pay to the smart lock without a cashier and open the hostel door.

A typical design is illustrated in Figure 13. The system consists of a Central Unit (CU) storing wallets for cryptocurrencies, smart locks, and the blockchain. The blockchain component is powered by smart contracts that provide functionalities for transferring, recording transactions, and querying transaction records, aimed at ensuring the transparency and immutability of transaction records.

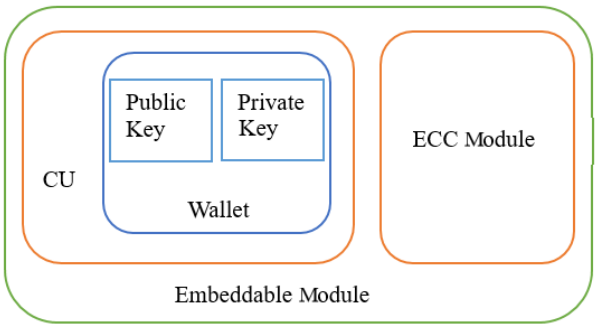

Figure 13. Updated smart lock.

The general process is illustrated in Figure 14:

(1) Authenticate: Users authenticate with the smart lock using their cryptocurrency wallet via the SPCL protocol.
(2) Generation payment message: The smart lock generates a payment message $Msg = Enc_{SK_{SL}}(payment)$. *payment* is the payment information. The smart lock sent the message Msg to the user's wallet.
(3) Sign the payment message: After the user's wallet receives Msg, it verifies the message $payment = Dec_{SK_{MC}}(Msg)$. If the verification is successful, the user signs the message $Sign = Enc_{d_{MC}}(Msg)$.
(4) Send signed payment information: The user wallet sends the signature message $Sign$ to the blockchain. Then, the blockchain verifies the signature $Msg = Dec_{P_{MC}}(Sign)$.
(5) Transaction execution by blockchain: If the verification is successful, the user's identity is confirmed and the transfer operation is performed.
(6) Payment information upload: The smart contract uploads *payment* on the blockchain.
(7) Query the transaction Order : Users or hotel administrators can query *payment* through the smart contract on the blockchain.
(8) Return transaction order: The smart contract returns *payment* to the user or the hotel administrators after querying the message.

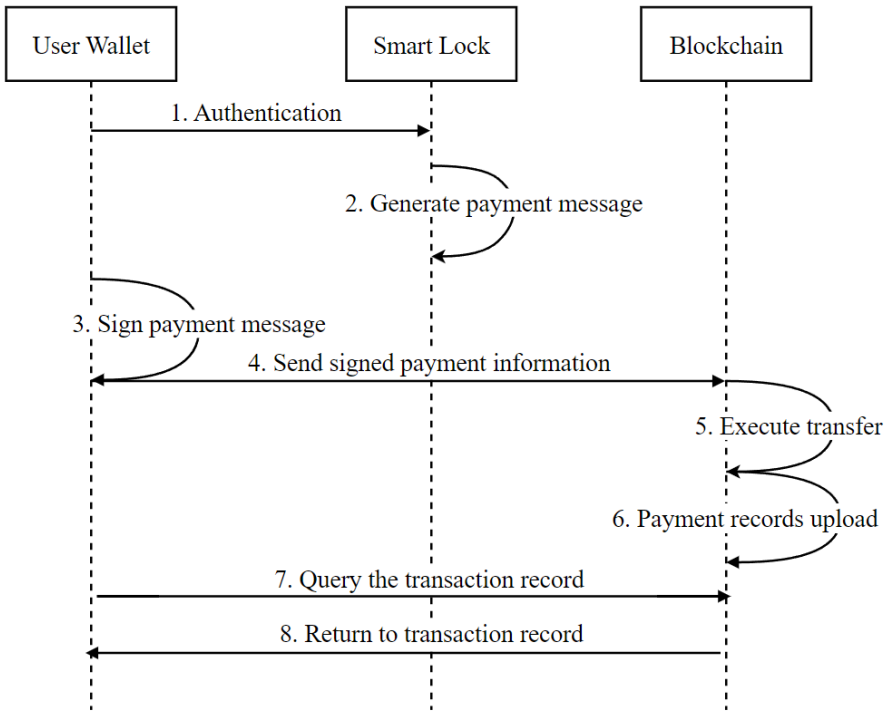

Figure 14. Payment procedure for blockchain based smart lock system.

Our smart contract code can be viewed on this website (https://github.com/wujiax/SPCL.git, accessed on 16 March 2024).

Our blockchain system meticulously balances security with utility, offering enhanced authentication via the SPCL protocol to mitigate unauthorized access risks. Secure payment processing is achieved through encrypted transactions, safeguarding payment details against unauthorized interception. Blockchain's autonomous verification further reduces fraud and error risks, ensuring transactions proceed transparently without third-party interference. In addition, the immutability of blockchain provides permanent verification of all transactions, thereby establishing a safe, reliable and transparent system to manage access and transactions in the hotel reservation system.

8. Related Work

In this section, we have compared and analyzed three categories of related work. The first category is a comparison of several similar intelligent access control solutions proposed previously. The second category is SPCL-related benchmark protocol comparison. The third category is a comparison of the differences in related solutions that support blockchain.

At present, smart phones are becoming considerably popular and ubiquitous in our daily lives [13,34]. With the widespread popularity of smart phones, the smart phone based applications have also gained growing attention. In this paper, we have built a prototype of SPCL, using a smart phone and a cloud server. SPCL can be applied to access control in intelligent buildings to enhance security and convenience. In the architecture presented by Jeong et al. [15], the mobile phone connects with the smart devices directly. Nevertheless, security mechanisms rely on the security of Bluetooth. In addition, the security solution is not quite flexible compared with SPCL. Patil et al. [18] proposed a security protocol for smart lock systems, but this solution did not fully consider user privacy. Guntur et al. [21] used RFID tags to unlock the lock. However, the loss of the card will bring risks to the security of the smart lock. The smart disposable door lock system based on invisible touch sensors by Mehmood et al. [22] is difficult to apply to hotels and other commercial occasions and lacks usability.

In SPCL-related benchmark protocol, Xie et al. [30] introduced an innovative access control scheme leveraging smart cards, albeit this approach has been critiqued for its vulnerability to attacks aimed at compromising long-term secrets. Similarly, the framework proposed by kumari et al. [29], while ambitious, demonstrates limitations in warding off replay attacks and fails to establish mutual authentication, thereby undermining the security integrity of smart lock systems. In an intriguing deviation from conventional methods, Zhang et al. [28] ventured into the metaverse realm, proposing an authentication protocol utilizing smart glasses for server authentication, in contrast to the ubiquitous smartphone-based approaches. However, when benchmarked against the SPCL framework, it becomes evident that SPCL not only offers a more energy-efficient and cost-effective solution but also excels in usability without necessitating the deployment of additional Virtual Reality (VR) equipment. Moreover, the PSLA scheme [27], despite its tailored applicability to environments characterized by disparate computing capabilities, is deemed less conducive for smart door lock scenarios due to its inadequacy in offering perfect forward security and potential exposure of short-term secrets.

Blockchain solutions have garnered widespread attention in recent years. In 2017, Zhang et al. [35] introduced a blockchain data sharing scheme for the healthcare sector. This system utilizes blockchain to store addresses (instead of actual health data) to maintain privacy and efficiency, allowing nodes to securely access and share health data. More recently, Zhang et al. [36] proposed a web3-based academic paper access control system for sharing papers. Due to the lack of blockchain support in current smart lock systems, SPCL provides a blockchain-enabled solution to provide safe and useful credentials for users using online hotel reservation systems.

9. Conclusions and Future Work

Nowadays, the advent of mobile technology, along with cloud computing, has drastically changed the way modern people connect with the world. With the continuous development of the IoT, wireless network control of smart devices has become more important for people. This paper integrates mobile and IoT technologies with cloud computing, offering a smart access control system for intelligent buildings. As a result, the building administrators are able to use mobile phones to control smart locks remotely and securely.

In order to enhance security, we designed and developed the SPCL protocol, providing proof of its security and usability and evaluating the performance of the security protocol. It is crucial to note, however, that while the SPCL protocol offers robust security features, it may not perform optimally in scenarios where computing power is unbalanced. Following that, we conducted an investigation into the issues arising from wireless network-controlled smart devices and discussed and summarized the findings of the investigation. Finally, we compared various popular unlocking methods and proposed a solution supported by blockchain technology.

For future work, more features and extensions will be applied to our system to enhance security and convenience, e.g., Near Field Communication (NFC), facial recognition, and fingerprint functions, which are already integrated into smart phones. In addition, we anticipate the incorporation of multiple authentication factors in future developments, which includes biometric systems and smart card technology. This strategic expansion in authentication methods is expected to significantly augment the system's robustness and user accessibility.

Author Contributions: Conceptualization, J.W., N.X. and X.H. (Xin Huang); methodology, J.W., N.X., Z.L., X.H. (Xianbin Hong) and Y.Z.; software, J.W. and N.X.; validation, Z.L. and X.H. (Xianbin Hong); formal analysis, J.W., N.X. and J.Z.; investigation, J.W. and N.X.; resources, X.H. (Xin Huang) and J.Z.; data curation, J.W., N.X. and Y.Z.; writing—original draft preparation, J.W. and N.X.; writing—review and editing, N.X., Z.L, X.H. (Xianbin Hong) and Y.Z.; visualization, J.W. and Z.L.; supervision, X.H. (Xin Huang) and J.Z.; project administration, X.H. (Xin Huang); funding acquisition, X.H. (Xin Huang) and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Shanxi Scholarship Council of China 2021-038, Applied Basic Research Project of Shanxi Province No. 20210302123130, and the National Natural Science Foundation of China under Grant No. 62002296; the Natural Science Foundation of Jiangsu Province under Grant No. BK20200250; and Xi'an Jiaotong-Liverpool University Research Development Fund under Grant No. RDF-21-02-014.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We would like to extend our heartfelt gratitude to Youjia Zhang for his invaluable contributions to this paper.

Conflicts of Interest: Author Zhen Li was employed by the company Shanghai Grandhonor Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ashley, S. Under lock and key. *Mech. Eng.* **1993**, *115*, 62.
2. McCrie, R.D. A history of security. In *The Handbook of Security*; Palgrave Macmillan: London, UK, 2006; pp. 21–44.
3. Son, J.Y.; Park, J.H.; Moon, K.D.; Lee, Y.H. Resource-aware smart home management system by constructing resource relation graph. *IEEE Trans. Consum. Electron.* **2011**, *57*, 1112–1119. [CrossRef]

4. Zhang, L.; Liu, B.; Tang, Q.; Wu, L. The development and technological research of intelligent electrical building. In Proceedings of the 2014 China International Conference on Electricity Distribution (CICED), Shenzhen, China, 23–26 September 2014; pp. 88–92.
5. Shao, H.; Fu, H. Design and implementation of intelligent building engineering information management system. In Proceedings of the 2014 7th International Conference on Intelligent Computation Technology and Automation, Changsha, China, 25–26 October 2014; pp. 158–161.
6. Bott, E. Help: Working smarter lock, stock and password. *Comput. Secur.* **1995**, *1*, 39.
7. Kaklauskas, A.; Zavadskas, E.K.; Naimavicienė, J.; Krutinis, M.; Plakys, V.; Venskus, D. Model for a complex analysis of intelligent built environment. *Autom. Constr.* **2010**, *19*, 326–340. [CrossRef]
8. Ulusoy, C. Android Library Design and Implementation for Smart Lock Access Control Systems. Master's Thesis, Aalto University, Espoo, Finland, 2015.
9. Park, Y.T.; Sthapit, P.; Pyun, J.Y. Smart digital door lock for the home automation. In Proceedings of the TENCON 2009—2009 IEEE Region 10 Conference, Singapore, 23–26 November 2009; pp. 1–6.
10. Padmapriya, S.; KalaJames, E.A. Real time smart car lock security system using face detection and recognition. In Proceedings of the 2012 International Conference on Computer Communication and Informatics, Chennai, India, 3–5 August 2012; pp. 1–6.
11. Chang, C.K.; Jiang, T.M. A binary single-key-lock system for access control. *IEEE Trans. Comput.* **1989**, *38*, 1462–1466. [CrossRef]
12. Wu, T. A refined key-lock access control system. In Proceedings of the IEEE 1993 National Aerospace and Electronics Conference-NAECON 1993, Dayton, OH, USA, 24–28 May 1993; pp. 583–587.
13. Iftode, L.; Borcea, C.; Ravi, N.; Kang, P.; Zhou, P. Smart phone: An embedded system for universal interactions. In Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems, Suzhou, China, 26–28 May 2004; pp. 88–94
14. Ping, W.; Guichu, W.; Wenbin, X.; Jianguo, L.; Peng, L. Remote Monitoring Intelligent System Based on Fingerprint Door Lock. In Proceedings of the 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, China, 11–12 May 2010; Volume 2, pp. 1012–1014.
15. Jeong, H.D.J.; Lee, W.; Lim, J.; Hyun, W. Utilizing a Bluetooth remote lock system for a smartphone. *Pervasive Mob. Comput.* **2015**, *24*, 150–165. [CrossRef]
16. Bauer, L.; Cranor, L.F.; Reiter, M.K.; Vaniea, K. Lessons learned from the deployment of a smartphone-based access-control system. In Proceedings of the 3rd Symposium on Usable Privacy and Security, Pittsburgh, PA, USA, 18–20 July 2007; pp. 64–75.
17. Xue, N.; Liang, L.; Zhang, J.; Huang, X. An access control system for intelligent buildings. In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, Xi'an, China, 18–20 June 2016; pp. 11–17.
18. Patil, B.; Vyas, P.; Shyamasundar, R. SecSmartLock: An architecture and protocol for designing secure smart locks. In Proceedings of the Information Systems Security: 14th International Conference, ICISS 2018, Bangalore, India, 17–19 December 2018; Proceedings 14; Springer: Berlin/Heidelberg, Germany, 2018; pp. 24–43.
19. Ahmad Taslim, H.; Md Lazam, N.A.; Mohd Yahya, N.A. Development of smart home door lock system. In *Advances in Robotics, Automation and Data Analytics: Selected Papers from iCITES 2020*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 118–126.
20. Ahmad, T.; Morelli, U.; Ranise, S.; Zannone, N. Extending access control in AWS IoT through event-driven functions: An experimental evaluation using a smart lock system. *Int. J. Inf. Secur.* **2022**, *21*, 379–408. [CrossRef]
21. Guntur, J.; Raju, S.S.; Niranjan, T.; Kilaru, S.K.; Dronavalli, R.; Kumar, N.S.S. IoT-Enhanced Smart Door Locking System with Security. *SN Comput. Sci.* **2023**, *4*, 209. [CrossRef]
22. Mehmood, M.Q.; Malik, M.S.; Zulfiqar, M.H.; Khan, M.A.; Zubair, M.; Massoud, Y. Invisible touch sensors-based smart and disposable door locking system for security applications. *Heliyon* **2023**, *9*, e13586. [CrossRef]
23. Song, H.; Tu, Z.; Qin, Y. Blockchain-based access control and behavior regulation system for IoT. *Sensors* **2022**, *22*, 8339. [CrossRef]
24. Zhai, P.; He, J.; Zhu, N. Blockchain-based Internet of Things access control technology in intelligent manufacturing. *Appl. Sci.* **2022**, *12*, 3692. [CrossRef]
25. Hasan, M.R.; Alazab, A.; Joy, S.B.; Uddin, M.N.; Uddin, M.A.; Khraisat, A.; Gondal, I.; Urmi, W.F.; Talukder, M.A. Smart Contract-Based Access Control Framework for Internet of Things Devices. *Computers* **2023**, *12*, 240. [CrossRef]
26. LaMacchia, B.; Lauter, K.; Mityagin, A. Stronger security of authenticated key exchange. In *International Conference on Provable Security*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 1–16.
27. Chai, S.; Yin, H.; Xing, B.; Li, Z.; Guo, Y.; Zhang, D.; Zhang, X.; He, D.; Zhang, J.; Yu, X.; et al. Provably Secure and Lightweight Authentication Key Agreement Scheme for Smart Meters. *IEEE Trans. Smart Grid* **2023**, *14*, 3816–3827. [CrossRef]
28. Zhang, X.; Huang, X.; Yin, H.; Huang, J.; Chai, S.; Xing, B.; Wu, X.; Zhao, L. Llakep: A low-latency authentication and key exchange protocol for energy internet of things in the metaverse era. *Mathematics* **2022**, *10*, 2545. [CrossRef]
29. Kumari, A.; Jangirala, S.; Abbasi, M.Y.; Kumar, V.; Alam, M. ESEAP: ECC based secure and efficient mutual authentication protocol using smart card. *J. Inf. Secur. Appl.* **2020**, *51*, 102443. [CrossRef]
30. Xie, Q.; Wong, D.S.; Wang, G.; Tan, X.; Chen, K.; Fang, L. Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol With Extended Security Model. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1382–1392. [CrossRef]
31. *GM/T 0003-2012*; SM2 Elliptic Curve Public Key Cryptographic Algorithms. Chinese Cryptography Administration: Beijing, China, 2010.

32. Park, K.; Park, Y.; Park, Y.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
33. Aviv, A.J.; Budzitowski, D.; Kuber, R. Is bigger better? Comparing user-generated passwords on 3×3 vs. 4×4 grid sizes for Android's pattern unlock. In Proceedings of the 31st Annual Computer Security Applications Conference, Los Angeles, CA, USA, 7–11 December 2015; pp. 301–310.
34. Ballagas, R.; Borchers, J.; Rohs, M.; Sheridan, J.G. The smart phone: A ubiquitous input device. *IEEE Pervasive Comput.* **2006**, *5*, 70–77. [CrossRef]
35. Zhang, J.; Xue, N.; Huang, X. A secure system for pervasive social network-based healthcare. *IEEE Access* **2016**, *4*, 9239–9250. [CrossRef]
36. Zhang, D.; Wang, C.; Xue, N.; Li, Z.; Zhang, H.; Huang, X. Blockchain papers depository system based on web 3.0. In Proceedings of the International Conference on Computer Application and Information Security (ICCAIS 2022), SPIE, Wuhan, China, 23–24 December 2022; Volume 12609, pp. 623–629.