Investigation into the Key Technologies of Smart Locks and Analysis of Typical Applications

Jiayi Wu

Communication Engineering institute, Xidian University, Xi'an, 710126, China
21012100060@stu.xidian.edu.cn

Abstract. This paper introduces a proposed Near Field Communication (NFC) based smart access control system. The system utilizes a server built on Spring Boot, an Android-based client, and a hardware lock powered by Raspberry Pi to create a robust smart access control mechanism. Through innovative system design, encryption technologies such as RSA or AES are leveraged to ensure the system's security. Moreover, a mutual authentication process between the user and the server is implemented, reinforcing the legitimacy of both parties involved in the access control process. To validate the system's performance and ascertain its efficiency, a comprehensive test was conducted utilizing three personal computers networked within the same Local Area Network (LAN). The results from the performance testing revealed that our proposed scheme outperforms existing smart access control systems in terms of security and applicability. Our system imposes lower design requirements on users compared to biometric access control systems and radio frequency card access control systems. However, it slightly surpasses the requirements of password-based access control systems. In essence, the NFC-based smart access control system proposed in this paper not only improves the overall security but also enhances the user experience by integrating modern technologies and innovative design elements. This research contributes to the broader discussion on smart lock technologies, paving the way for more secure and user-friendly access control systems in the future.

Keywords: NFC, Android, Door lock control system

1 Introduction

Android smartphones have become an irreplaceable tool of communication in contemporary life, and with ongoing enhancements in communication technology, users are increasingly demanding speedier and more dependable data transmission [1]. Near Field Communication technology, known for its high confidentiality and swift data transmission, has found extensive applications in fields such as mobile payments and smart keys. Given NFC's attributes, an increasing number of smart mobile devices are incorporating this functionality, a trend that is predicted to intensify in the future. Contactless Radio Frequency Identification (RFID) technology holds a significant position in access control systems due to its low cost and ease of

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_77

use. As a rising star in the industry of short-range contactless communication technologies, NFC has seen successful development since its introduction in 2003. It has garnered widespread attention from various industries, significantly contributing to the advancement of NFC technology research and its applications [2].

The integration of smartphones, NFC technology, and smart hardware can lead to a plethora of applications in everyday life, contributing to the creation of safer and smarter work and living environments. For instance, existing laboratory password access control systems could be replaced, allowing laboratory administrators to centralize management, allocate user rights, and assign different rooms to personnel based on their levels [2]. Another example could be the modernization of existing hotel access control systems to permit occupants to access rooms at specific times. The architecture of the door lock control system is illustrated in Fig 1.

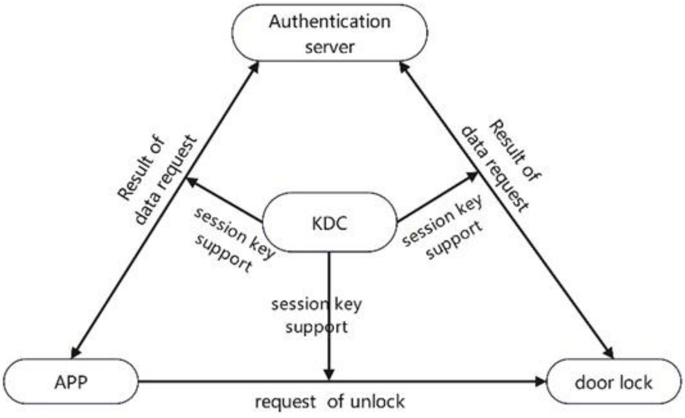

Fig. 1. The architecture of the door control system [3]

2 System Architecture and Implementation

2.1 Software Part

The software part is mainly implemented through an Android-based App. Users who use this App will go through several steps, including registration, login, permission management, and door opening. Before these steps, each application will go through an initialization step [4]. Let's now conduct a detailed analysis. The software component is predominantly executed via an Android-based application. Individuals utilizing this application will undergo various stages, encompassing registration, login, permission administration, and access granting. Preceding these stages, each application will commence with an initialization process. The specific process is shown in the Fig 2.

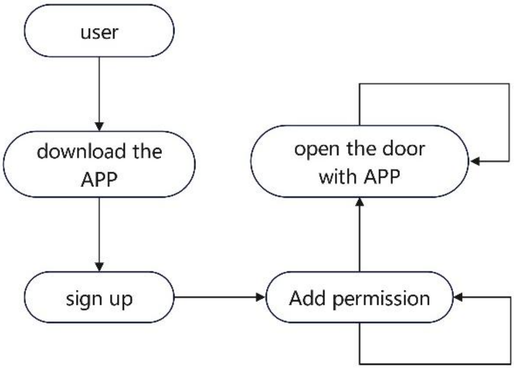

Fig. 2. The architecture of the software system [5]

Below is a specific analysis: Upon initializing the app for the first time, it generates a unique identifier akin to a machine fingerprint, crafting its own public and private keys. The app then requests the Key Distribution Center public key, encrypts it using the RSA algorithm, and signs it using the SHA-1 algorithm. Once it secures the KDC public key, the app registers itself and presents the registration outcome [3].

Users must register before using the service. During this process, they provide a nickname, a login password, and a door opening password. The app automates the rest, generating a user ID that is subsequently revealed to the user. From then on, this user ID serves as their login credential. Users must login every time the app is relaunched or superseded by other applications. They use the user ID and password established during the registration process for this purpose. Once a user account is successfully created, the user can log into the app. Next, they need to be granted access rights to their purchased door. The user-to-door permission relationship falls into three categories: administrator, regular, and temporary permissions. Temporary permissions are contingent on time or usage frequency [5]. To ensure security, each door lock should only have a single user with administrator privileges. However, the number of regular and temporary permissions can be unrestricted. Initially, the default administrative permission for each new door belongs to the management system or manufacturer. Yet, once the door lock is sold, the administrator permission must be promptly transferred to the dealer. After a user buys a door, the dealer uses their account to pass on the door lock's administrative permissions to the user. Once a user has administrative privileges, they can add other users such as family members or friends, assigning them relevant permissions. The app displays the door lock permission information based on the user's permissions. If the user has administrative privileges, they can enter the user ID they want to grant access to and select their permission level. After pressing "OK," the app sends a permission alteration request to the Authorization Server (AS). The AS verifies the user's administrative privileges and the legality of relevant parameters before processing the request. If the user requests administrative permissions, they are transferred to the new user. Regular permissions provide door opening access but no other privileges. Temporary permissions grant time-limited access or frequency-based access, useful for hotels or situations requiring infrequent access, such as household workers [6].

Opening the door is crucial for the access control system. Each door lock comes with a pre-installed basic key, and the door lock control unit keeps track of the door opening count. The door lock compares each key received from a user through NFC to a range outside the total number of times the door has been opened five times. The number of door openings increases by one for each request, regardless of success, but the door lock control does not increment for failed attempts. When the total door openings and the door lock control basic key reach 20, the control unit requests a key change from the AS. If the update fails, the basic key and the opening count remain unchanged. The key update is initiated by the door lock, not the server, so the door won't fail to open due to a key update. Opening a door with administrator or regular privileges follows the same process. The user clicks on the door they want to open, the app prompts for the door opening password, and then the app sends a door opening request to the AS [7]. When a user with temporary permissions attempts to open a door, the server verifies their access rights. For time-based temporary permissions, it checks if the user's access time is still valid. Similarly, for frequency-based temporary permissions, it verifies if the user's access frequency is still within the legal limit.

2.2 Hardware Part

The door lock component of the hardware is mainly composed of four parts: Raspberry Pi, motor drive module, door lock motor module and NFC reading module. The architecture of the hardware is shown Fig 3.

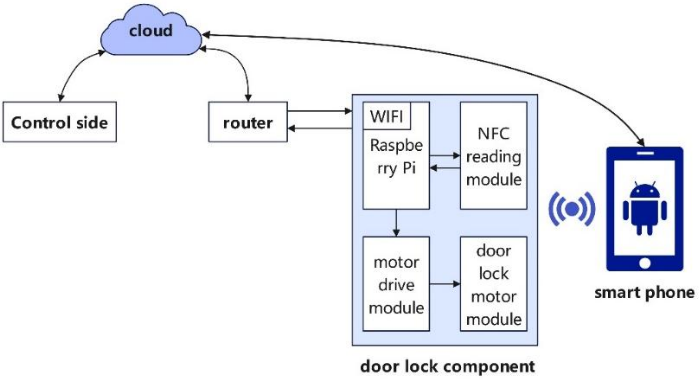

Fig. 3. Architecture of the hardware system [5]

Raspberry Pi is an ARM-based microcomputer motherboard, with SD/MicroSD card as memory hard disk, card motherboard around 1/2/4 USB interface and a

10/100 Ethernet interface (type A no network port), can be connected to keyboard, mouse and network cable, while has video analog signal TV output interface and HDMI high-definition video output interface, the above components are all integrated in a motherboard only slightly larger than a credit card, with all the basic functions of PC just connect to the TV and keyboard, It can perform functions such as spreadsheets, word processing, playing games, playing high-definition videos, and more. In this door lock component, the Raspberry Pi is used to be used to realize the communication between the door lock control and the cloud and the mobile phone, and control the motor drive to realize the unlocking function. The actual design schematic and PCB diagram of the NFC reading module in this door lock control are shown in the Fig 4 and Fig 5.

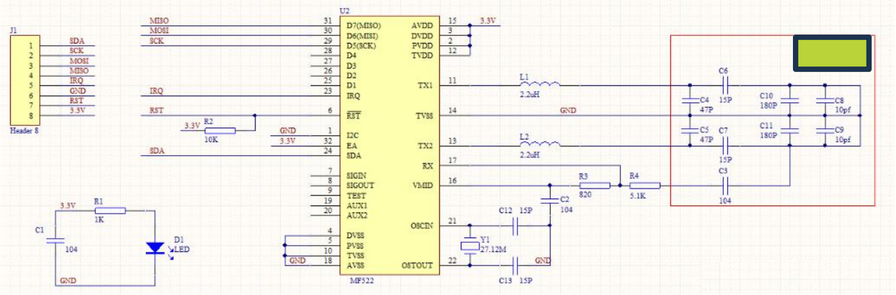

Fig. 4. Rc522 design schematics [6]

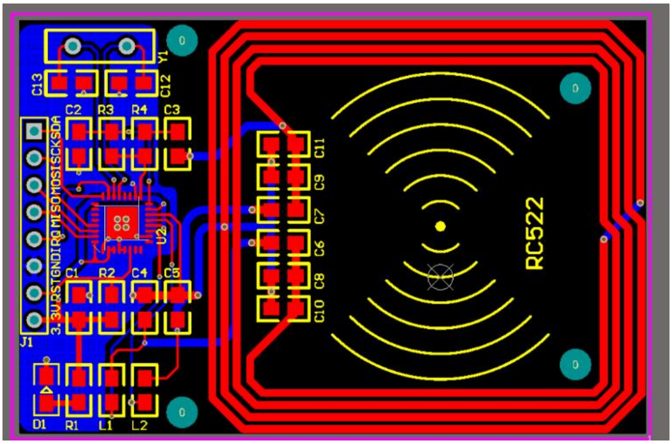

Fig. 5. Rc522 PCB drawing [6]

In this hardware system, in order to realize the door opening function, the user first applies for the unlock password in the cloud, and the administrator issues the key to the eligible user through review, and synchronously issues the verification key to the corresponding door lock. After obtaining the password, the user uses NFC to communicate with the door lock, reads and checks the NFC data through the

Raspberry Party, and controls the motor drive to complete the door opening operation after confirming that it is correct. In this hardware system, the door opening function is achieved through a series of steps [8]. Firstly, the user requests the unlock password from the cloud. The administrator conducts a review process and, if the user is eligible, issues the key to them. Simultaneously, the administrator also provides the corresponding door lock with a verification key. Upon receiving the password, the user employs the Near Field Communication protocol to establish communication with the door lock. Utilizing the Raspberry Party, the user reads and verifies the NFC data, and subsequently governs the motor drive [9]. This ensures the completion of the door opening operation only when the authentication process is successfully validated.

3 System Security Analysis and Design

In this design, two servers are designed: a key distributor and a user authentication server, and the most important thing for both servers is the security of the session and the generation and delivery of keys [10].

3.1 Cryptographic Design

Encryption During APP Initialization. When the app is first launched, the app will generate an identifier similar to the machine fingerprint, uniquely identifying itself.

Encryption Requirements of Registration. In this design, users will be asked to enter three values when registering: user nickname, login password, and door opening password. Encryption requirements of login. Users should log in first when using this design, so whether the user opens a new app or switches from another program to a running APP, they should log in first. Encryption requirements of opening. In this design, the door opening business is the most frequently used business by users. For a user, the door is opened and closed at least twice a day, that is, when there are N users in this design, the minimum number of door openings per day is between N and 2N.

3.2 Key Generation Design

The design discussed in this paper primarily relies on session key requests and employs the Advanced Encryption Standard (AES) 256-bit algorithm, which necessitates a 32-byte long key. Accordingly, the key generation algorithm must be versatile, providing unique keys for varying requests simultaneously and, in highly concurrent scenarios, different keys for various users at different times, as well as for distinct intervals. For the base key of the door lock, a long 16-byte (64-bit) key is required. To facilitate interactions between the user, the door lock, and the shared key between the Authentication Server (AS) and the Key Distribution Centre, a 32-byte (256-bit) long key is necessary. For session key generation, the system design requires a 32-byte long key that can handle high concurrency without conflicts. The ANSI X9.17 pseudo-random number generator, one of the most secure generators of its kind available today, is used for this purpose. As this generator produces an output of 56 random bits, but the session key length in the design is 256 bits, a separate thread is initiated in the KDC design to manage session key generation. In this thread, a key generation factory function is run, which continuously stores randomly generated bits into a key cache queue. When a session key request arrives, the KDC retrieves 256 bits from the queue and returns them to the requester as the session key.

In terms of door lock key generation, the key is intrinsically linked to the door lock base key and the number of door openings. The door lock base key, a 16-byte (64-bit) key, undergoes an SM3 algorithm calculation with each door opening to determine a digest value. Since the digest length of the SM3 algorithm is twice as long as the door lock base key in this design, the resulting SM3 algorithm should be compressed twice. The compression algorithm used in this design is as follows: calculate the sum of the ASCII codes of each character of the incoming string, determine its parity, and, depending on whether it's even or odd, perform a set of calculations and transformations. This process ultimately yields a 16-byte (64-bit) string.

4 System Test Scenario

4.1 The User Registers the Test Scenario

Three computers were used for the test on the LAN, one running KDC and AS and the other two and the one running the NFC-based secure intelligent hotel access control server all running the test code. Twenty threads were created in the main function, each responsible for registering 2500 users, so a total of 150,000 users registered with the KDC from the three computers. The sample space for the random username generation algorithm is {62}^6\approx5.68\times{10}^{10}. For the KDC this is around 180 requests per second and for the AS it is between 120 and 180 requests per second.

4.2 Test Scheme for Random String Generation Function

Write a program that consists of 20 threads, each of which generates 100,000 32-bit long random strings using a random string generation function. From the properties of the random string generation function, the sample space of the function is $62^{32} \approx 2.3 \times 10^{57}$.

4.3 User Registration Test Results

It took 237 minutes for Local Host to register 50,000 users, 240 minutes for Remote Host 1 to register 50,000 users and 241 minutes for Remote Host 2 to register 50,000 users, with no user ID conflicts, but it can be noted that the operational load on the KDC and AS grew rapidly over time resulting in slower processing, but still for each user, with very high registration concurrency, the average The registration time is 5.5 seconds. The scatterplot for the 50,000 native tests is as Fig 6.

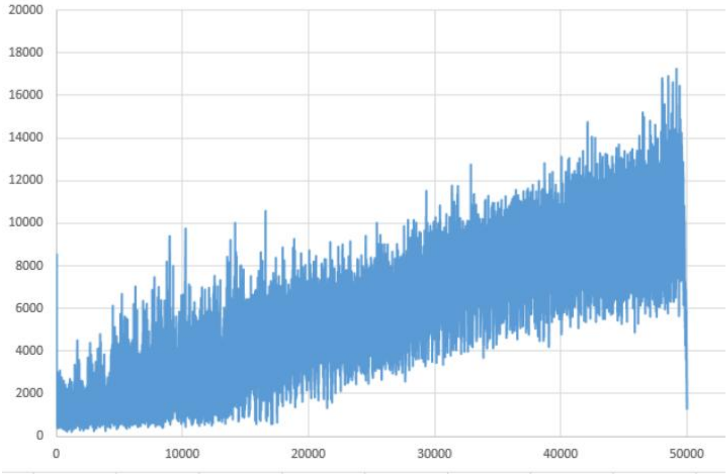

Fig. 6. Scatter plot of local test data (Photo/Picture credit: Original)

4.4 Random String Generation Function Test Results

Table 1. Random string generation function test results

Thread count	Time spent (in seconds)
1	582.12
2	582.45
3	582.32
4	583.12
5	582.13
6	583.14
7	584.12
8	583.22
9	584.48
10	583.12
11	582.59
12	583.46
13	582.56
14	583.205
15	582.92
16	584.19
17	582.05
18	584.47
19	582.75
20	583.98

The main function took 19515.49 seconds. After processing 2 million strings, no collisions were found. At this point the random string generation function can be considered computationally safe. As shown in Table 1.

5 Typical Application Research: Hotel Automation System

This system holds broad applicability, encompassing scenarios such as home, campus, and hotel access control. One specific application within a hotel access control scenario is illustrated here. The management party in the application, in this case the hotel, can audit the customer and issue the corresponding key through the designed interface. Hotel occupants can conveniently access their rooms by swiping their mobile phones, equipped with NFC modules, over the door lock, or even control the lock remotely. This streamlined approach enables the hotel management to effectively maintain information about the hotel and monitor access control status based on the NFC security smart hotel access control system, thereby enhancing security.

The design ensures security by incorporating a network layer communication protocol to protect key transfer and user information. Special modules manage the generation, distribution, storage, and eventual destruction of the NFC key, preserving the security uniqueness of the key. An additional layer of security is provided through a two-factor authentication mechanism. This innovative approach not only optimizes hotel management processes, reducing costs and enhancing security, but also deepens information management. It enables real-time monitoring and remote intelligent management of access control for both the hotel and the user. By offering both security and convenience, the system addresses traditional hotel service process shortcomings and carries substantial practical significance.

6 Conclusion

The paper delineates the conception and execution of a smart access control system that leverages the power of Near Field Communication. An amalgamation of various components such as a server constructed on the Spring Boot framework, an Android-supported client interface, and a Raspberry Pi-based door lock forms the backbone of this innovative system. With a strong focus on autonomy, this setup emerges as an access control solution par excellence. An illustration of its deployment in a hotel access control framework attests to the system's adaptability and resilience. This portrayal shines a light on its capacity to function in high-stake real-life scenarios, reinforcing the system's credibility and practical value. Not merely confined to hospitality industry applications, the robust and flexible nature of this solution invites a spectrum of applications across numerous industries, thereby broadening its horizon.

Performance evaluations underscored the system's supremacy in terms of security and broader applicability. When placed against the backdrop of existing access control mechanisms prevalent in China, the designed system unequivocally outperformed. This marked achievement signifies a substantial stride forward in the realm of intelligent access control solutions. In essence, the system serves as a testament to the transformative potential of NFC technology when coupled with intelligent design and robust security measures. Whether the context is a hotel setting or a broader industrial application, the system exudes versatility, performance, and strength. By transcending traditional limitations, it revolutionizes the landscape of smart access control, pushing the envelope of what is achievable in the domain of access control solutions. This innovation represents a significant milestone, championing a new era in intelligent access control technology.

References

1. Li, B., Tian, M., & Zheng, S. (2019). Design of access control system based on the near field communication (NFC) technology. In International Conference on Insulating Materials, Material Application, and Electrical Engineering.
2. Ji-Yan, L., & Li-Xia, W. (2018). Design and Implementation of Illegal Operating Vehicles' Reporting System Based on NFC and Android. Techniques of Automation and Applications.
3. Klee, S., Roussos, A., Maass, M., et al. (2020). NFCGate: Opening the Door for NFC Security Research with a Smartphone-Based Toolkit.
4. Khabarlak, K., & Koriashkina, L. (2021). Mobile Access Control System Based on RFID Tags And Facial Information.
5. Bao, Y., & Sun, Z. (2020). Ad-Hoc Network Access Control System and Method for Edge NFC Terminal. In International Conference on Machine Learning and Artificial Intelligence.
6. Zhang, B., Pan, W., & Huc, Z. (2017, September). Intelligent Warehouse Management System. In 2017 5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017) (pp. 860-866). Atlantis Press.
7. Zeling, C., & Qi, W. (2018). Research and design of NFC access control system. Information Technology and Network Security.
8. Ruo-Cheng, W. (2019). The Research on Key Technologies of Student Behavior Analysis System in Big Data Environment. Information Technology and Informatization.
9. Mamonov, S., & Benbunan-Fich, R. (2020). Unlocking the smart home: exploring key factors affecting the smart lock adoption intention. Information Technology & People, ahead-of-print(ahead-of-print).
10. Shi, D., Tao, D., Wang, J., et al. (2021). Fine-Grained and Context-Aware Behavioral Biometrics for Pattern Lock on Smartphones. Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies.