Journal of Electronic Research and Application, 2024, Volume 8, Issue 5
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Design and Implementation of USB Key System Based on Dual-Factor Identity Authentication Protocol

Jianxin Wang, Zifan Xu*, Xiangze Chang, Chaoen Xiao, Lei Zhang

Beijing Electronic Science and Technology Institute, Beijing 100070, China

***Corresponding author:** Zifan Xu, xuzifan2001@outlook.com

Abstract: With the increasing demand for information security, traditional single-factor authentication technology can no longer meet security requirements. To this end, this paper proposes a Universal Serial Bus (USB) Key hardware and software system based on a two-factor authentication protocol, aiming to improve the security and reliability of authentication. This paper first analyzes the current status and technical principles of USB Key-related research domestically and internationally and designs a two-factor authentication protocol that combines impact/response authentication and static password authentication. The system consists of a host computer and a USB Key device. The host computer interacts with the USB Key through a graphical user interface. The Secure Hash Algorithm 1 (SHA-1) and MySQL database are used to implement the authentication function. Experimental results show that the designed two-factor authentication protocol can effectively prevent replay attacks and information tampering, and improve the security of authentication. If the corresponding USB Key is not inserted, the system will prompt that the device is not found. Once the USB Key is inserted, user identity is confirmed through two-factor verification, which includes impact/response authentication and static password authentication.

Keywords: Information security; USB Key; Impact/response authentication; Static password authentication

Online publication: September 30, 2024

1. Introduction

Data security is a growing concern in modern information society, particularly in scenarios involving sensitive information and key operations. Traditional single-factor authentication methods, relying on passwords or Personal Identification Numbers (PINs), are inadequate due to their vulnerability to social engineering and brute force attacks. Multi-factor authentication, especially Two-Factor Authentication (2FA) using USB Keys, has garnered attention for enhancing security [1–3].

However, existing USB Key-based protocols, such as the Diffie-Kasahara Authenticated Key Exchange Protocol (DKAKEP), face practical challenges like inadequate host program design and insufficient defense against emerging attack methods [4]. To address these issues, this paper proposes a new USB Key software and

hardware system design based on a two-factor authentication protocol [5,6]. This system combines impact/response authentication and static password authentication, aiming to prevent replay attacks and enhance user convenience.

Our study innovatively integrates two authentication methods and verifies their feasibility and superiority through system design and testing. This research provides new solutions for improving USB Key authentication security, promoting its application in real-world scenarios.

2. Related technologies and principles

2.1. USB key

USB Key is a hardware device used for identity authentication, which usually contains a security chip for generating and storing encryption keys [7,8]. Its main technical features include:

(1) USB communication protocol: Defines the data transmission rules between the device and the host. The USB protocol supports multiple transmission types, including control transmission, interrupt transmission, bulk transmission, and synchronous transmission, ensuring the reliability and efficiency of data transmission.

(2) Security chip: The built-in security chip is used to store sensitive information, such as encryption keys and authentication credentials. The security chip usually has tamper-proof features to protect the stored data from physical attacks.

2.2. Identity authentication

Identity authentication technology is an important means to ensure system security and can be classified into three categories: knowledge factors, possession factors, and biological factors. USB Key mainly uses knowledge factors and possession factors for verification during the identity authentication process, including the following methods:

(1) Impact/response authentication: It is a type of possession factor. The impact/response authentication mechanism verifies the user's identity by generating a one-time dynamic password [9]. The dynamic password generated by the USB Key is compared with the expected response of the server in real-time to prevent replay attacks, significantly enhancing the security of identity authentication.

(2) Static password authentication: A type of knowledge factor where static password authentication relies on the user's preset password and verifies the user's identity by comparing the password stored in the server [10]. Its security mainly depends on the complexity and confidentiality of the password. Weak passwords are easy to guess or crack, affecting the overall security of the system.

3. Design of two-factor authentication scheme

This paper designs a two-factor authentication protocol that combines impact/response authentication and static password authentication. The overall scheme design is shown in **Figure 1**.

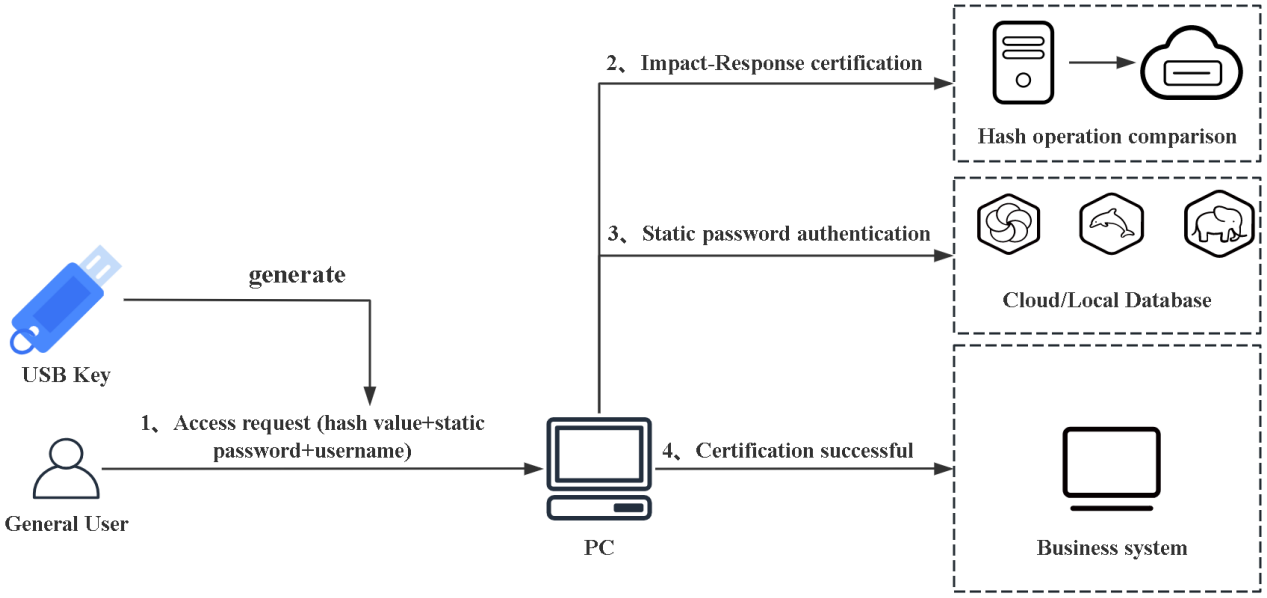

Figure 1. Two-factor authentication overall solution

3.1. Design of shock/response authentication protocol

In this process, the USB Key receives a random number K ("shock") from the host, combines it with its identification (ID) to form plaintext M, and generates a hash value using SHA-1. The hash value is sent back to the host as a response. The specific process is depicted in **Figure 2**.

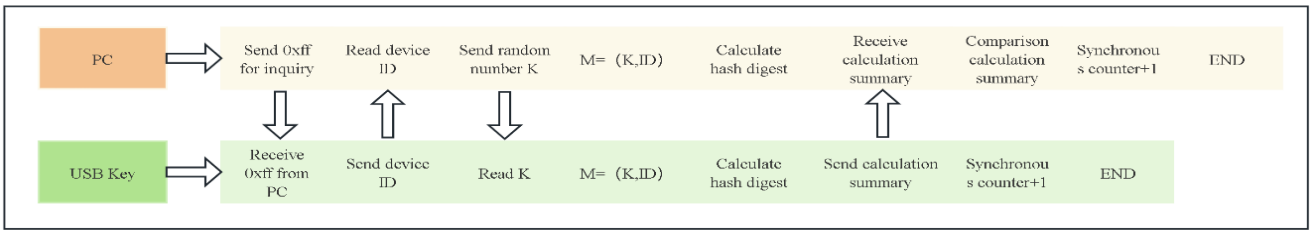

Figure 2. Impact/response to the identity authentication process

3.2. Design of static password authentication protocol

After the impact/response authentication, the user performs static password authentication by entering their username and password into the user interface (UI). The system reads and decrypts the stored encrypted password from the database for comparison. If the credentials match, the static password authentication is successful, enhancing identity verification and preventing identity theft. Failure to match results in authentication failure.

4. Design and implementation of USB Key software and hardware system

Users interact with the host computer through a graphical user interface (GUI), which handles data communication and command exchange with the USB Key via a standardized USB interface. The host computer, as the core application entity, confirms user identity through user interaction. The overall system design is shown in **Figure 3**.

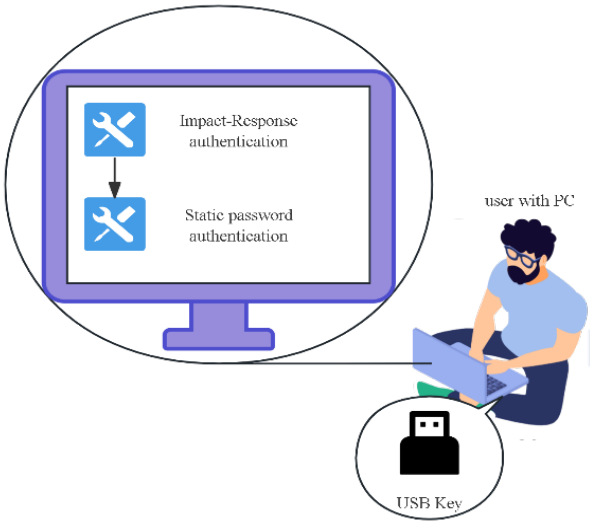

Figure 3. Overall system design

4.1. Host computer platform implementation

This part implements the UI interface and identity authentication scheme on the host computer. The main program logic involves detecting the USB Key insertion, and performing impact/response authentication, followed by static password authentication.

(1) Impact/response identity authentication implementation: Using PySimpleGUI, a simple user interface is created to display prompt information for USB Key authentication. The host calls USB_identification() to read the device identifier (64-bit) from the USB Key. A random number K is generated and combined with the device ID to form plaintext M = (K, ID), which is sent to the USB Key. The host then performs an SHA-1 hash operation on M to compute the 160-bit summary data.

(2) Static password authentication implementation: PySimpleGUI is used to build a login interface. A MySQL database with an auto-increment ID stores usernames and passwords. Users enter their credentials in the login window, and the system retrieves the matching password from the user's table, ensuring security against SQL injection through safe variable insertion.

4.2. USB Key hardware platform implementation

The USB Key system uses the C*core C8000 development board, as shown in **Figure 4**. When inserted, the USB Key monitors and responds to the host's signals. Upon receiving 0xff, it sends the device ID for verification. The USB Key reads the random number K from the data buffer, combines it with the device ID to form plaintext M, calculates the hash result with Dev_SHA1(M), and sends the 20-byte hash result back to the host.

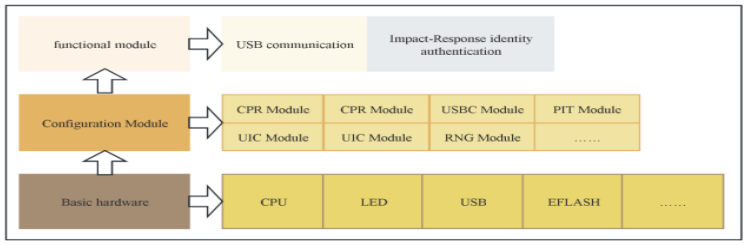

Figure 4. USB Key software and hardware platform

5. Function test

This article is based on the PyCharm 2020 host system and uses the C8000 chip to test various functional modules. It mainly tests the impact/response authentication, static password authentication, and other items.

5.1. Impact/response authentication

When the corresponding USB Key is not inserted, the system fails the ID authentication (**Figure 5**). Due to the presence of dual threads, the system does not exit but instead displays the message: "Waiting for USB Key authentication, Device not found, retrying," indicating that the device is not detected and the USB interface is being continuously monitored. Once the correct USB Key is inserted, the system receives the ID sent by the USB device, confirming its validity. The system then initiates the impact/response authentication process (**Figure 6**), calculates the hash of the combined plaintext, and if the hash comparison is correct, the authentication is successful, and the terminal displays "Authentication successful." If the comparison fails, the program exits.

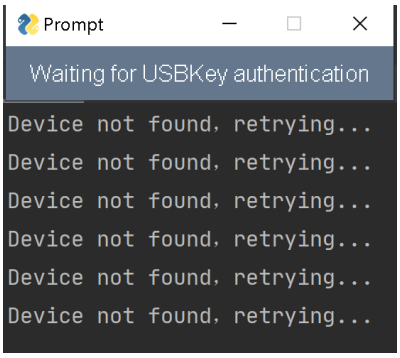

Figure 5. When the USB Key is not inserted

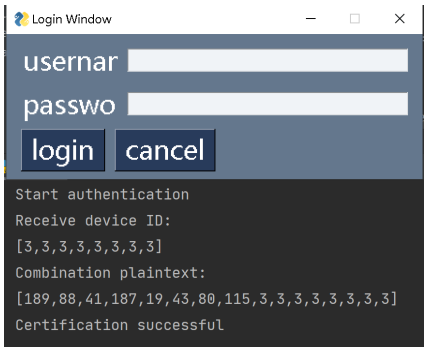

Figure 6. Impact/response authentication is successful

5.2. Static password authentication

In the static password authentication interface that appears, you need to enter the correct password to pass authentication. Clicking the exit key or cancel button in the upper right corner will terminate the system authentication process. If an incorrect password is entered, an error message will be displayed, as shown in **Figure 7**. Similarly, if you enter a user that does not exist in the MySQL database, an error message will be shown, as depicted in **Figure 8**. After receiving an input error prompt, you can continue to enter the correct credentials by clicking OK in the error message box.

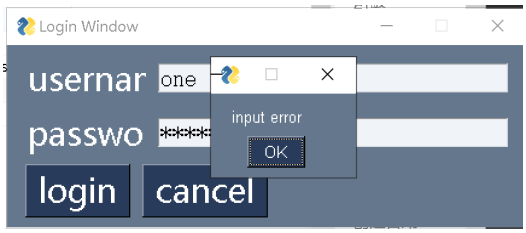

Figure 7. Entering an incorrect username

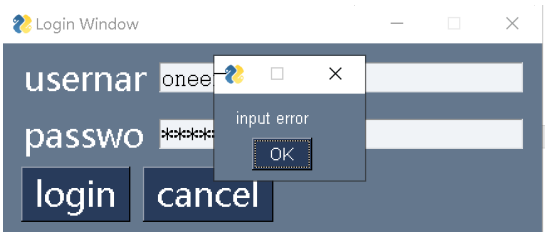

Figure 8. Entering an incorrect password

6. Conclusion

This article designs and implements a USB Key software and hardware system based on a two-factor identity authentication protocol, combining impact/response and static password methods to enhance security. Experimental results demonstrate that the system effectively prevents replay attacks and information tampering, ensuring accurate user verification and ease of operation. When the USB Key is inserted, two-factor authentication ensures user identity accuracy and security. Future research will focus on optimizing system performance, improving user experience, and enhancing cross-platform compatibility, aiming to broaden the system's application and contribute to information security development.

Funding

(1) This research was funded by the College-level Characteristic Teaching Material Project (Project No. 20220119Z0221)
(2) The College Teaching Incubation Project (Project No. 20220120Z0220)
(3) The Ministry of Education Industry-University Cooperation Collaborative Education Project (Project No. 20220163H0211)
(4) The Central Universities Basic Scientific Research Fund (Project No. 3282024009, 20230051Z0114, and 20230050Z0114)
(5) The Beijing Higher Education "Undergraduate Teaching Reform and Innovation Project" (Project No. 20220121Z0208 and 202110018002)
(6) The College Discipline Construction Project (Project No. 20230007Z0452 and 20230010Z0452)

Disclosure statement

The authors declare no conflict of interest.

References

[1] Idrus SZS, Cherrier E, Rosenberger C, Schwartzmann JJ, 2013, A Review on Authentication Methods. Australian Journal of Basic and Applied Sciences, 7(5): 95–107.

[2] Joyce R, Gupta G, 1990, Identity Authentication Based on Keystroke Latencies. Communications of the ACM, 33(2): 168–176.

[3] Wu X, Xu J, Wang J, et al., 2019, Identity Authentication on Mobile Devices Using Face Verification and ID Image Recognition. Procedia Computer Science, 162: 932–9.

[4] Wu Y, Deng L, Xiao D, et al., 2007, A Two-Factor Identity Authentication and Key Exchange Protocol Based on USB Key. Computer Engineering and Science, 29(5): 56–59.

[5] Liu W, Hu J, Liu Y, 2008, Design and Implementation of a Transparent Encryption and Decryption File System Based on USB Key. Computer Science, 35(11): 100–103.

[6] Yu Q, Nan Y, Shi W, 2011, Design of Online Banking Identity Authentication Based on USB Key and Fingerprint Recognition. Science and Technology Communication, 2011(5): 219–221.

[7] Wang S, Chang Z, Wei Y, 2014, USB Key Identity Authentication Scheme Based on Cloud Computing. Computer Applications Research, 31(7): 2130–2134.

[8] Wu P, Cai Q, Wang Q, et al., 2021, Building a Secure Video Conference System with Customized Cryptographic USB Keys. ICC 2021–IEEE International Conference on Communications.

[9] Yang X, Liu W, 2024, Research on Mainstream Security Authentication and Authorization Technology: OAuth2.0. Network Security Technology and Application, 2024(04): 9–12.

[10] Zhou G, 2003, Research on Two-Factor Authentication Technology. Electronics Technology, 2003(22): 37–38, 41.