frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 24 March 2025
DOI 10.3389/fphy.2025.1509626

OPEN ACCESS

EDITED BY
Chengyi Xia,
Tianjin Polytechnic University, China

REVIEWED BY
Dawei Zhao,
Qilu University of Technology, China
Zhigang Li,
Zhengzhou University of Light Industry, China

*CORRESPONDENCE
Jie Hu,
py1909@ynufe.edu.cn

RECEIVED 11 October 2024
ACCEPTED 20 February 2025
PUBLISHED 24 March 2025

CITATION
Hu J and Kang X (2025) Adaptive security protocol for financial management networks in multi-server environments.
Front. Phys. 13:1509626.
doi: 10.3389/fphy.2025.1509626

Adaptive security protocol for financial management networks in multi-server environments

Jie Hu* and Xuan Kang

ZhongHua Vocational College, Yunnan University of Finance and Economics, Kunming, China

Driven by the digital wave, the security and efficiency of financial management networks are key factors determining the competitiveness and sustainable development of enterprises. Faced with complex and ever-changing network threats in multi-server environments, traditional static security strategies are no longer sufficient to meet the security needs of modern enterprises. It is particularly important to develop a security protocol that can adapt to environmental changes and defend against potential threats. Therefore, we propose a lightweight adaptive security protocol for financial management networks in multi-server environments. This protocol uses a hash function to negotiate session keys at low computation and communication overhead, effectively protecting the transmission security of confidential messages. In addition, informal and formal analysis proves that this protocol has high security and can resist various network attack methods. We demonstrate the efficiency of the protocol in practical applications through performance comparisons. It not only has low communication overhead and good computational efficiency but also achieves lightweight message transmission, making it easy to deploy and use in multi-server environments.

KEYWORDS

financial management network, multi-server, security, protocol, lightweight

Highlights

- We propose a lightweight adaptive security protocol for financial management networks in multi-server network environments.
- Informal and formal analysis methods are used to analyze the specific security of the protocol.
- Through performance comparison, it is proven that this scheme has low communication overhead and good computation overhead.

1 Introduction

Financial management is the core link of enterprise operation in today's digital age. Its security and efficiency are directly related to the survival and development of the enterprise [1]. With the rapid advancement of technologies such as cloud computing and the Internet of Things (IoT), enterprise financial management systems are gradually transitioning from traditional single-machine or LAN models to

multi-server, cross-regional, and high-concurrency network environments [2]. This transformation not only greatly enhances the flexibility and real-time performance of financial management but also poses unprecedented challenges to the security of the system. The accompanying network information security issues constantly threaten the privacy and security of information in our communication process [3]. The identity authentication key protocol designed based on cryptography can enable users to securely enjoy the convenience brought by network services and effectively ensure network information security [4].

With the expansion of enterprise scale and the globalization of business, financial management systems become increasingly complex. The amount of data that needs to be processed has exploded, with extremely high requirements for real-time, accurate, and secure data [5]. To address these challenges, enterprises adopt multi-server architectures and improve system stability and reliability through technologies such as load balancing, data redundancy, and disaster recovery backup. However, while a multi-server environment brings convenience, it also exacerbates the difficulty of security management [6]. The financial management network in a multi-server environment faces complex and ever-changing network threats, including but not limited to data breaches, illegal access, service interruptions, and advanced persistent threats [7]. Once these risks become a reality, they cause incalculable economic losses and reputational damage to the enterprise.

Traditional static security strategies are no longer effective in dealing with increasingly complex and ever-changing network attack methods [8]. Therefore, developing a financial management network security protocol that can adapt to environmental changes and intelligently identify and defend against potential threats has become the key to ensuring the security of enterprise assets and promoting sustainable business development. Traditional security protocols are designed for a single server [9]. When a user needs to request network services, providing authentication factors such as identity and password to the single server can obtain the service requested by the server. Due to the rapid development of the network, there are a large number of servers in the current Internet environment [10]. When users want to request services, they need to register with all the requested single servers. Then, users need to remember all the authentication factors, such as identity and password verification, when registering. This is obviously a huge resource burden for users, and there are extensive illegal attacks on the public channel of communication between users and servers [11]. It is very likely that a set of user identity or authentication factors are disclosed and attacked, thus affecting the security of other systems. This is undoubtedly a huge security risk. The factors that must be considered when designing security protocols for different multi-server network environments are also different [12, 13]. Therefore, in the design process of security protocol in a multi-server environment, it is not only necessary to meet the security requirements of the application environment but also to balance computational and communication costs to achieve better performance.

Especially driven by the current global wave of informatization, as the core support for enterprise operations, enterprise financial management systems are undergoing unprecedented changes and challenges. With the expansion of enterprise scale and the globalization of business, traditional financial management models are no longer able to meet the high requirements of modern enterprises for data processing speed, system stability, and information security [14]. In a multi-server environment, financial management systems not only need to handle massive amounts of financial data but must also ensure the security and integrity of these data during cross-regional and cross-network transmission [15]. An adaptive security protocol can automatically adjust security policies according to changes in the network environment to effectively resist various network attacks. It ensures the security and integrity of financial data. At the same time, the protocol can optimize system performance and enhance user experience while ensuring security. Our main contributions are summarized as follows.

(1) Considering the requirements of financial management networks in multi-server network environments, we propose a lightweight adaptive security protocol. In this protocol, both communication parties need to register at the control server and then engage in security negotiations. Through a hash function, this protocol can negotiate session keys with lower computational and communication costs. This protects the transmission of confidential messages and enhances communication security.
(2) This protocol adopts both informal and formal analysis methods to analyze the specific security of the protocol, which strongly demonstrates the high security of this protocol. Through performance comparison, it is proven that this protocol has low communication overhead and good computation overhead. Lightweight message transmission is convenient for practical applications. This protocol achieves security and practicality and is more appropriate for multi-server environments.

The other parts of the article are described. Section II and Section III systematically review the current research status in related fields. Section IV comprehensively introduces the design ideas, specific implementation steps, and key technical details of the security protocol. Section V and Section VI, respectively, focus on the security verification and performance analysis of the protocol. Finally, Section VII is the summary.

2 Literature review

With the significant advancement of communication technology, ensuring the confidentiality and privacy of user information has become particularly important. Therefore, there have been many studies on multi-server authentication protocols both domestically and internationally.

Lamport [16] first proposed a password-based remote identity authentication scheme, which was based on a verification table and password. Subsequently, researchers proposed an increasing number of authentication schemes, but most of them were suitable for single-server environments. However, due to the increasing demand for security, relying solely on verification tables could not guarantee communication security. In a single-server architecture, when users needed to request services from different servers, remembering the identity and password when logging into each server was challenging. To solve the problem of

users needing to remember manage multiple physical passwords and multiple high entropy passwords, an increasing number of identity authentication schemes that could be applied to multi-server environments were proposed. Tsaur et al. [17] introduced the concept of a multi-service model and built an authentication mechanism in a multi-server environment based on the RSA public key cryptosystem and Lagrange interpolation inequality principle. Subsequently, Li et al. [18] integrated neural network technology into a multi-server authentication architecture. Chang et al. [19] proposed a remote authentication scheme that did not require verification table maintenance, and users did not need to remember multi-server passwords, significantly improving the user experience. Yoon et al. [20] used elliptic curve public key encryption technology and designed a three-factor authentication scheme aimed at enhancing security in multi-server environments. However, subsequent research [21] pointed out that this scheme had shortcomings in resisting internal attacks, smart card theft, offline password cracking, and impersonation attacks. In response, Kalra et al. [22] proposed an efficient and cost-optimized multi-server authentication protocol that utilized smart card bidirectional authentication and elliptic curve cryptography technology to achieve higher security. Guo et al. [23] also designed a smart card-based authentication scheme in multi-server architecture, which clearly defined the roles of the registration server, service server, and user. Both users and application servers needed to perform registration once on the registration server. Three-party authentication mode was implemented using the ElGamal public key cryptosystem. Subsequently, the Burrows Abadi Needham logic provided formal proof of the proposed scheme.

Gupta et al. [24] proposed a key exchange authentication scheme that combines biometric cryptography and smart card technology in a distributed multi-client server architecture, particularly for scenarios with multiple registration centers. Subsequently, Li et al. [25] conducted an in-depth analysis of biometric-based identity verification and key negotiation schemes in multi-server environments. They proposed corresponding improvement strategies based on this to further enhance the security and efficiency of the authentication mechanism. Wang et al. [26] reviewed several authentication schemes applicable to multi-server architectures this year and pointed out the security vulnerabilities of the corresponding schemes, proving that the schemes were ineffective in practical applications. Pelaez et al. [27] proposed an enhanced lightweight cloud computing authentication scheme for IoT, which also included a substage called connection attempt evidence. It provided evidence about user and service participation. Unfortunately, the study by Yu et al. [28] revealed significant shortcomings in the security of [27], pointing out that it could not effectively resist impersonation attacks, session key leakage, and replay attacks. They also proposed a secure and lightweight three-factor authentication scheme specifically designed for IoT in cloud computing environments. This scheme innovatively incorporated secret parameters and biometric authentication elements to ensure enhanced mutual authentication mechanisms and user anonymity, effectively addressing various security threats.

Wong et al. [29] focused on the application of 5G wireless sensor networks in electronic health systems and designed a three-factor fast authentication scheme that balances time constraints and user anonymity. This authentication scheme combined a three-factor authentication scheme of biometric, password, and smart card methods to ensure a highly secure communication environment supported by sensors. It maintained user anonymity during the communication process. Tsai et al. [30] proposed a multi-server authentication scheme for online banking transaction environments that used a hash-based multi-server authentication scheme combined with smart cards to authenticate online banking customers and transactions. It provided powerful security features and lower maintenance costs for the online banking platforms of financial institutions. The solution supported interface connection with the banking system, making it easy to integrate the solution into existing banking systems. Sudhakar et al. [31] proposed a multi-server environment-enhanced authentication scheme based on passwords and smart cards by improving the security flaws of [32]. The improved scheme formally proved the security authentication of the scheme using BAN logic and simulated various attacks through Internet security protocol and automatic verification of application tools. The results showed that the improved scheme had better security and performance.

In their research on ensuring authentication security in multi-server environments, Xia et al. [33] introduced the principle of elliptic curve cryptography and designed a three-factor authentication key agreement scheme, significantly enhancing the security of the system. Akram et al. [34] proposed an efficient anonymous authentication key protocol for multi-server infrastructure within the same year. This protocol effectively resisted various security challenges, including impersonation attacks, insider attacks, and password modification attacks. Finally, a formal security analysis of the proposed solution was conducted using a random oracle model. Analysis and comparison showed that this scheme was highly effective for anonymous authentication and key schemes. Wu et al. [35] pointed out the shortcomings of the protocol [36] in terms of fully forward secrecy protection and susceptibility to privileged internal attacks. In response to these security vulnerabilities, they designed a customized authentication key exchange scheme for 5G network multi-server architecture.

Km et al. [37] focused on improving security in multimedia IoT environments and proposed an enhanced multi-factor authentication scheme with provable security. Hsu et al. [38] developed an end-to-end cryptographic authentication key exchange scheme for multi-server architecture in edge computing networks. This scheme allowed end users to use easy-to-remember passwords for initial login and then used external agents to calculate shared keys to achieve secure communication with specific service end users. It was particularly worth mentioning that this scheme provided a high degree of user anonymity protection during the communication process.

3 Preliminaries

3.1 Network model

Figure 1 shows the three main participants in multi-server architecture authentication: the control server, the user, and the application server [22–38]. The control server is the registration center. Each user needs to avoid registering on a specific server that presents a particular service. The registry operates under

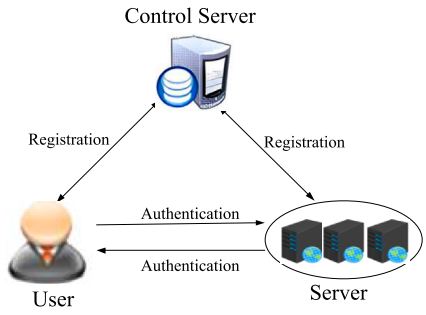

FIGURE 1
System model.

the assumption that the user and the server providing the service trust it. The application server and user first complete the registration process and obtain the corresponding data information authorized by the registration center. The above data information is used for the future mutual authentication process between the user and the server. Distributed application servers can cross geographical boundaries and provide diverse services to remote users. Users only need to complete a one-time registration process through the registration center to obtain access permissions and seamlessly integrate with multiple authorized application servers, thus conveniently obtaining the required resources and services.

3.2 Attacker model

In a multi-server environment, attackers in the security protocol generally possess the following capabilities [30–42]. According to the Dolev Yao model, the attacker's attack on the user is as follows: An attacker can not only eavesdrop on all messages propagated on the public channel during the protocol but also intercept, modify, and forge them before sending.

3.3 Safety objectives

(1) The basic functions that this protocol should implement are bidirectional authentication and session key negotiation. To ensure the legitimacy of the participants in the session key negotiation process, mutual authentication of the identities of the participants should be implemented first. The session key should be jointly negotiated among the participants and cannot be generated and distributed by one party in the negotiation process.
(2) This protocol should resist all sorts of common attacks, such as denial of service attacks, man-in-the-middle attacks, impersonation attacks, offline password guessing attacks, etc.

3.4 One-way hash function

The cryptography one-way hash function can convert the input into a certain length of output, that is $h:\{0,1\}^* \rightarrow \{0,1\}^l$. In detail, the one-way hash function must also meet the following three characteristic conditions [38–41].

(1) For any x ≠ y, its respective hash value $h(x) \neq h(y)$.
(2) For any x and $h(x) = k$, it is not computationally feasible to solve the specific value of x for knowing the k.
(3) For any x, it is computationally infeasible to solve for y with its respective hash values $h(y) = h(x)$.

4 Proposed scheme

4.1 Initialization stage

The control server (CS) selects a hash function $h(\cdot)$. Meanwhile, the CS selects a private key, k. Finally, the CS exposes the security parameters $\{h(\cdot)\}$.

4.2 Server registration stage

The server S_j sends a registration request to the control server, CS. The CS randomly selects a unique identity SID_j and a random number q_j for it and uses the private key k of the CS to generate the key $h(SID_j \| kq_j)$. Then, the CS transmits the value $\{h(SID_j \| kq_j), h(k)\}$ to the S_j via the secure channel. $\{h(SID_j \| kq_j), h(k)\}$ is received and secretly stored.

4.3 User registration stage

1. For U_i to register on the CS, it needs to choose a unique identity ID_i, a random number x_i, a password PW_i, and biometric information B_i.
2. U_i completes the following calculations: $UP_i = h(PW_i \| B_i \| x_i), UD_i = h(ID_i \| PW_i \| x_i)$. U_i sends the $\{UP_i, UD_i\}$ to the CS via the secure channel.
3 After the CS receives $\{UP_i, UD_i\}$, the CS calculation is as follows: $CK_i = h(UD_i || k || UP_i)$, $CD_i = CK_i \oplus h(UP_i)$, $L_i = h(k) \oplus h(UP_i \| UD_i)$, $CG_i = h(CK_i \| h(k) || UP_i || UD_i)$. Then, for each application S_j, the CS completes the corresponding calculation for it. They are $US_{ij} = h(h(SID_i || k || q_j) \| UD_i)$, $UF_{ij} = US_{ij} \oplus h(UP_i \| UD_i \| j)$. Finally, the CS writes the $\{CD_i, L_i, CG_i, (SID_j, UF_{ij}, j)\}$ in the smart card SC_i and sends it to U_i through a secure channel.
4. After U_i receives the smart card, its starting calculation is as follows: $UX_i = h(ID_i PW_i \| B_i) \oplus x_i$. Then, write UX_i into SC_i. Finally, SC_i contains an information value of $\{UX_i, CD_i, L_i, CG_i, (SID_j, UF_{ij}, j)\}$.

4.4 User login stage

When U_i wants to communicate with S_j, U_i needs to insert $\langle ID_i^*, PW_i^*, B_i^* \rangle$ into the smart card and complete the login process. The specific login process is as follows.

SC_i completes the following calculation after receiving the data provided by U_i: $x_i^* = h(ID_i^* PW_i^* \| B_i^*) \oplus UX_i$, $UP_i^* =$

$h(PW_i^* \| B_i^* \| x_i^*)$, $UD_i^* = h(ID_i^* \| PW_i^* \| x_i^*)$, $CK_i^* = h(UP_i^*) \oplus CD_i, h(k) = h(UP_i^* UD_i^*) \oplus L_i, CG_i^* = h(CK_i^* || h(k) || UP_i^* UD_i^*)$. Next, $CG_i^* = CG_i$ is compared to see if it is true. If not true, the user is denied a login.

If the above conditions hold, SC_i extracts the corresponding $US_{ij}^* = UF_{ij} \oplus h(UP_i^* UD_i^* \| j)$ to produce the random numbers m_i and performs the following calculations. They are $UM_i = h(US_{ij}^* \| T_1)$, $UN_i = UM_i \oplus m_i$, $UID_i = UD_i^* \oplus h(h(k) \| j)$, $USM_i = h(UD_i^* \| m_i \| h(k) \| T_1)$, where T_1 represents the current time stamp.

Finally, U_i sends $\{UID_i, UN_i, USM_i, T_1\}$ to S_j through the open channel.

4.5 Mutual authentication and key negotiation stage

U_i and S_j complete mutual authentication and share the session key. The specific steps are described below.

1. When S_j receives the login request from U_i, S_j first checks the timestamp through $T_2 - T_1 \le \Delta T$, where ΔT is the maximum allowed time interval, and T_2 indicates the current timestamp. If the above conditions are m, et, S_j calculates $D_i^* = UID_i \oplus h(h(k) \| j)$, $US_{ij} = h(h(SID_j \| kq_j) \| UD_i^*)$, $UM_i^* = h(US_{ij} \| T_1)$, $m_i^* = UM_i^* \oplus UN_i$, $USM_i^* = h(UD_i^* \| m_i^* h(k) \| T_1)$. S_j tests and calculates whether the USM_i^* is equal to USM_i. If both are equal, the certification process continues.
2. S_j selects a random number u_j, and then the calculations are as follows: $SU_j = u_j \oplus h(US_{ij} \| T_2)$, session key $SK = h(UD_i^* \| SID_j \| m_i^* \| u_j \| US_{ij})$, and $SM_j = h(m_i^* \| u_j \| SK \| T_2)$. Finally, S_j contains $\{SU_j, SM_j, T_2\}$ messages transmitted to U_i in the open channel.
3. After receiving $\{SU_j, SM_j, T_2\}$ from S_j, U_i first checks the timestamp T_2 through $T_3 - T_2 \le \Delta T$, where T_3 indicates the user's current timestamp. If the above conditions are met, U_i calculates the $u_j^* = SU_j \oplus h(US_{ij} \| T_2), SK = h(UD_i^* \| SID_j \| m_i \| u_j^* \| US_{ij}^*), SM_j^* = h(m_i \| u_j^* \| SK \| T_2)$. Finally, U_i tests whether $SM_j^* = SM_j$ holds. If not true, the session is terminated. If true, U_i successfully certifies S_j. Finally, both parties use the session key SK in future interactions to ensure communication security.

5 Protocol security analysis

5.1 Informal analysis

The method of conducting security analysis in this article is to use informal language to provide a detailed introduction to the security of the proposed protocol.

5.1.1 Mutual authentication and key negotiation

During the authentication process, S_j verifies the legitimacy of U_i identity and the integrity of the transmitted message by checking whether the USM_i^* is equal to the received USM_i. U_i verifies the legitimacy of S_j and the integrity of the transmission message by checking whether the condition $SM_j^* = SM_j$ holds. U_i verifies that the received message is not maliciously modified. Two-way authentication between U_i and S_j is realized. At the same time, S_j and U_i negotiate the key SK. By checking whether the SM_j^* is equal to SM_j, U_i verifies the correctness and integrity of the key SK.

5.1.2 Denial of service attack

The login request of U_i is sent to S_j and the login request message $\{UID_i, UN_i, USM_i, T_1\}$ contains the timestamp T_1. When S_j receives the login request, the timestamp is first verified by verifying whether $T_2 - T_1$ is less than or equal to ΔT. Calculating $USM_i^* = h(UD_i^* \| m_i^* h(k) \| T_1)$ determines whether the USM_i^* is equal to the received USM_i. It not only verifies the identity of U_i but also verifies the integrity of the login request message, completely resisting the denial of service attack.

5.1.3 Man-in-the-middle attack

The attacker may capture U_i's message $\{UID_i, UN_i, USM_i, T_1\}$ and try to generate an illegal request. Because the attacker cannot know the secret value h(k) of S_j and the secret value UD_i^* of U_i, a request message cannot be successfully forged. Similarly, the attacker cannot make changes to the message $\{SU_j, SM_j, T_2\}$.

5.1.4 Counterfeit attack

If the attacker A captures U_i message $\{UID_i, UN_i, USM_i, T_1\}$ and obtains the smart card of U_i, then the attacker can get all the information in the smart card $\{UX_i, CD_i, L_i, CG_i, (SID_j, UF_{ij}, j)\}$ through the side channel attack. According to the above analysis in (3), the attacker cannot forge the information sent to S_j only by relying on the information in the smart card. Simultaneously, because $\{SU_j, SM_j, T_2\}$, it involves the $h(SID_j \| kq_j)$ and u_j, so the attacker cannot use the current system time T_2' to forge $\{SU_j, SM_j, T_2\}$ that can be verified by U_i. So, it can completely resist counterfeit attacks.

5.1.5 Replay attack

In this protocol, the timestamp is not only used in the login stage but also plays a major part in the authentication key negotiation stage. It specifies the threshold ΔT for the verification timestamp, so this protocol can resist a replay attack.

5.1.6 User anonymity

First, the attacker is unable to directly steal identity information from the user's smart card, partly because the smart card avoids storing the user's temporary identity within it. On the other hand, even if the attacker causes the message $\{UX_i, CD_i, L_i, CG_i, (SID_j, UF_{ij}, j)\}$ in the smart card to leak through the side channel attack, the attacker cannot get the user's ID_i. The open letter is the dissemination of user identity encrypted information UD_i^* to ensure the anonymity of the user. Therefore, this protocol has very good user anonymity.

Second, for the messages spread in the open letter, there is no similar information in the messages, even if the messages sent by the same user are authenticated with different servers. The attacker cannot track the user's identity. Therefore, this protocol has very good anti-tracking properties.

5.1.7 Forward safety

The key in this protocol is $SK = h(UD_i^* \| SID_j \| m_i \| u_j^* \| US_{ij}^*)$. The m_i and u_j^* are the randomly selected values of the user and

server during the authentication and key negotiation. These values are different in each authentication and key negotiation process. Although the session key is constantly attacked by the attacker, even if the attacker obtains the session key in the authentication process, the session key negotiated before or after cannot be obtained according to the calculation. An attack does not pose a threat to the previous or subsequent communication because each authentication and key negotiation process are independent. An attack would still fail to construct a valid session key. In conclusion, this protocol has a good forward safety profile.

5.1.8 Session key security

In this article, U_i and S_j negotiate to generate a session key $SK = h(UD_i^* \| SID_j \| m_i \| u_j^* \| US_{ij}^*)$ for subsequent secure communication. Among them, the calculation of SK requires a random number m_i generated by U_i and a random number u_j generated by S_j, which will be updated during protocol execution. Therefore, if a session key is compromised, it does not help to recover past or future session keys.

5.2 Analysis of security proof

The tool for verifying protocol security in this article is the random oracle model. Next, we provide a detailed introduction to the security model and inquiry model used for security proof [39].

5.2.1 Security model

The two main parties in this protocol are U_i and S_j. Under this security model, an attacker can eavesdrop or even tamper with all the messages in the open letter in probabilistic polynomial time.

5.2.2 Inquiry model

The attacker's attack capability is simulated by the following five interrogation models.

$Excute(U_i^j, S_j^k)$: This inquiry simulates the passive attack of the attacker; that is, attacker A can capture all the messages spread by the participant in the open letter through this inquiry.

$Send\ (U_i^j, m)$: This inquiry simulates the active attack of the attacker. That is, A can tamper with the message intercepted in the open letter channel and send it to instance U_i^j. After instance U_i^j receives the message, the attacker can also intercept the feedback message generated by the participant U_i.

$Reveal(U_i^j)$: This query simulates that if the instance U_i^j has generated SK, A can get the session key SK. If the instance U_i^j has not generated SK, the attacker cannot get the SK and can only get an invalid identification.

$Corrupt(U_i^j)$: This inquiry simulates that an attacker can obtain its secret credentials on the premise that a participant is corrupted. In this protocol, A can obtain all the information in the smart card of user U_i through this inquiry.

$Test(U_i^j)$: This asks whether the SK used to simulate instance U_i^j is safe. After this, the simulator performs a "coin toss operation." If the result is 1, the correct SK is returned to the attacker. If the result is 0, a random string set it to be the same length as the true session key is returned to the attacker. So, the attacker needs to determine if the return value is a real key or a random equal length string.

Theorem 1: If and only if the attack advantage $Adv_{\mathcal{F}}^{AKE}(A)$ of A in polynomial time is at most one quantity larger than $q_{hash}^2/|Hash| + 2q_{send}/|D|$, it is said the security protocol is semantically secure. The q_{send} is the number of times of A makes Send queries, q_{hash} is the number of times that A makes Hash inquiries, |D| is the dictionary space scale, |Hash| is the Hash query scale, and $\mathcal{F}$ is the protocol proposed in this article, which can be expressed as follows.

$$Adv_{\mathcal{F}}^{SP}(A) \le \frac{q_{hash}^2}{|Hash|} + \frac{2q_{send}}{|D|}.$$

5.2.3 Safety certificate

It is assumed that A can use at most q_{send} times of *Send* queries and q_{hash} times of Hash queries in the time t. We demonstrate that this protocol AKE is safe by using the hybrid experimental games $Game_0, Game_1, Game_2, Game_3$. Among them, $Game_0$ simulates real attacks. With the experimental game, the simulation rules of each advantage are increasingly different. The experimental games end when A gradually fails to distinguish the real session key and a random isolong string. $Pr[Succ_i]$ represents the advantage of A in $Game_1$.

$Game_0$: This experimental game simulates an attack in a real scene. According to the definition of semantic security, it is as follows.

$$Adv_{\mathcal{F}}^{SP}(A) = |2\,Pr[Succ_0] - 1|.$$

$Game_1$: In this experimental game, A begins to add *Execute* inquiries, so A needs to verify whether the SK in the message is the real key SK or a random key of equal length as SK. In this protocol, $SK = h(UD_i^* \| SID_j \| m_i^* \| u_j \| US_{ij})$. If A obtains all the messages, then there is $\{UID_i, UN_i, USM_i, T_1\}$ and $\{SU_j, SM_j, T_2\}$. However, these messages do not help A to get the m_i^* and u_j in the SK, indicating that the eavesdropping attack through $Game_1$ does not increase the advantage. Therefore, $Game_0$ and $Game_1$ are equal, so:

$$Pr[Succ_0] = Pr[Succ_1].$$

$Game_2$: In this experimental game, A adds a Send inquiry and a Hash inquiry, and A can tamper with the message of the participants. If A wants to build a legitimate message, it needs UD_i^*, m_i^*, u_j, and US_{ij}. If those values are not available, the timestamp distinguishes the message. This shows that $Game_2$ and $Game_1$ are the same except for the Send and Hash interrogation advantages. So, according to the birthday paradox,

$$|Pr[Succ_1] - Pr[Succ_2]| \le \frac{q_{hash}^2}{2|Hash|}.$$

$Game_3$: In this experimental game, the *Corrupt* interrogation is increased. A can get all the information stored in the

TABLE 1 Computation overhead comparison.

Protocol	Computation overhead
[43]	$13T_H$
[44]	$19T_H$
[45]	$25T_H$
[46]	$35T_H + 4T_{CM} + T_F$
[47]	$11T_H + 5T_{CM} + 7T_E + T_F$
This protocol	$20T_H$

TABLE 2 Communication overhead comparison.

Protocol	Communication overhead	Number of messages
[43]	1600 bits	4
[44]	3040 bits	4
[45]	2336 bits	4
[46]	2560 bits	4
[47]	1376 bits	3
This protocol	864 bits	2

smart card $\{UX_i, CD_i, L_i, CG_i, (SID_j, UF_{ij}, j)\}$. Because SK = $h(UD_i^* \| SID_j \| m_i^* \| u_j \| US_{ij})$, the information in the smart card cannot get SK. However, in the dictionary password attack, the attack advantage compared with the last increases $q_{send}/|D|$ is as follows.

$$|Pr[Succ_2] - Pr[Succ_3]| \leq \frac{q_{send}}{|D|}.$$

Finally, because A does not know the final result of the simulator coin toss operation, the SK is independently produced independently by U_i and the access server S_j, and

$$Pr[Succ_3] = \frac{1}{2}.$$

According to the above formulas, the following equation can be inferred, which proves Theorem 1.

$$Adv_{\mathcal{F}}^{SP}(A) \leq \frac{q_{hash}^2}{|Hash|} + \frac{2q_{send}}{|D|}.$$

6 Performance analysis

6.1 Computation overhead

Because the main purpose of designing this protocol is to pursue a lightweight identity authentication protocol while ensuring security, only hash and exclusive OR (XOR) operations are involved in the design process. In this section, we compare the computational cost of our scheme with [43–47], as shown in Table 1. This scheme has the lowest computational cost except for [43, 44], where T_H represents hash operation time, T_F represents the fuzzy extractor operation, T_E represents symmetric encryption, and T_{CM} represents the Chebyshev chaotic map. We ignore the time of the XOR operation.

By comparing the computational costs in Table 1, we can see that this protocol has slightly higher computational costs than [43, 44] but lower computational costs than [45–47]. However, [43] cannot perform mutual authentication and [44] cannot resist replay attacks. For security protocols, security attributes are the most important, so it is practical to exchange high security and low communication costs with appropriate computational costs. Therefore, this protocol has reasonable computational overhead and better security, which can better meet the traditional multi-server network environment with higher security requirements.

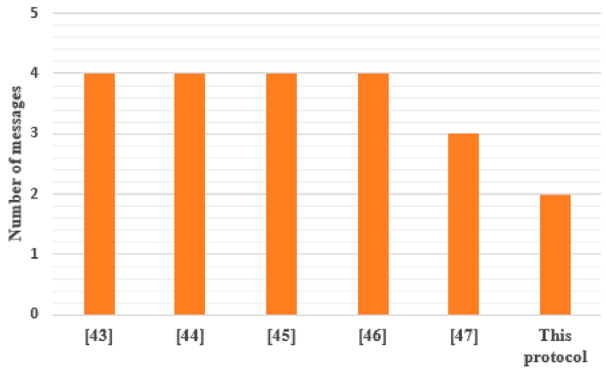

FIGURE 2
Number of messages.

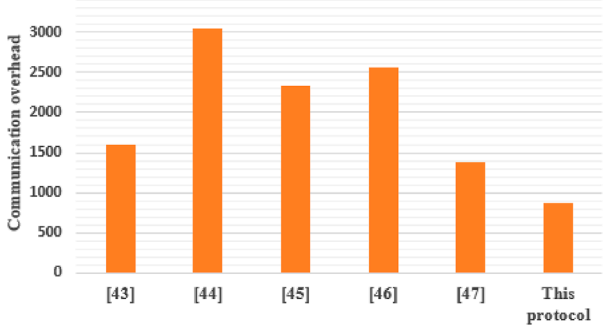

FIGURE 3
Communication overhead.

6.2 Communication overhead

To contrast the communication overhead more intuitively, the identity length is marked as $L_{ID} = 32$ bits. The timestamp length is $L_T = 32$ bits. The output length of the hash function is $L_H = 160$ bits, the output length of symmetric encryption is $L_E = 160$ bits, and the output length of the Chebyshev chaotic map is $L_{CM} = 160$ bits.

Table 2 shows the number of message flow transmissions for the protocols in the table. In Figure 2, we have only two protocol message streams, which is the lowest of the protocols compared. There are also obvious differences in message transmission bytes: [43] has 1600 bits. [44] has 3040 bits, [45] has 2336 bits, [46] has 2560 bits, and [47] has 1376 bits. This protocol transmits two messages in the logon and authentication key negotiation stage. First, U_i sends the request message $\{ UID_i, UN_i, USM_i, T_1\}$ to S_j, and the overhead is $(3 \times 160 + 32) = 512$ bits. Next, S_j sends messages $\{SU_j, SM_j, T_2\}$ to U_i and the overhead is $(2 \times 160 + 32) = 352$ bits. So the total overhead in this protocol is $(512 + 352) = 864$ bits. By comparing the communication cost in Figure 3, it is obvious that this protocol has less communication overhead. Therefore, compared with similar schemes, this protocol has better security attributes, lower communication overhead, and is more practical.

7 Conclusion

This article delves into the security and efficiency challenges faced by enterprise financial management networks in the digital age, particularly in the rapid development of technologies such as cloud computing and IoT. The inevitable trend for financial management systems to transition from single-machine or local area network models to multi-server, cross-regional, and high-concurrency network environments is highlighted. Although this transformation significantly improves the flexibility and real-time performance of financial management, it also makes system security issues increasingly prominent. This becomes a key consideration for the sustainable development and survival of enterprises. The issue of network information security, especially data privacy and communication security, has become an important issue that urgently needs to be addressed. We propose a lightweight, adaptive security protocol for special requirements in multi-server environments. This protocol effectively enhances the identity authentication strength and session key security of both communication parties, reducing the risk of data leakage and illegal access. This article comprehensively evaluates the security of the protocol using both informal and formal analysis, ensuring its robustness in various attack scenarios. In addition, we also fully consider the practicality and performance optimization issues of this protocol. By designing with low computational and communication costs, as well as a lightweight message transmission mechanism, this protocol demonstrates good efficiency and user experience in practical applications.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material; further inquiries can be directed to the corresponding author.

Author contributions

JH: conceptualization, data curation, investigation, methodology, project administration, resources, supervision, validation, writing–original draft, and writing–review and editing. XK: formal analysis, investigation, project administration, resources, supervision, validation, and writing–review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The 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.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

1. Gonzalez-Urango H, Mu E, Ujwary-Gil A, Florek-Paszkowska A. Analytic network process in economics, finance and management: Contingency factors, current trends and further research. *Expert Syst Appl* (2024) 237:121415. doi:10.1016/j.eswa.2023.121415

2. Budiasih Y. The influence of digital technology on financial management. *Account Stud Tax J (Count)* (2024) 1(1):92–100. doi:10.62207/wb6d3c96

3. Song N. Design and development of inclusive finance network security system model based on neural network algorithm. In: 2024 Asia-Pacific Conference on Software Engineering, Social Network Analysis and Intelligent Computing (SSAIC). IEEE (2024). 539–43.

4. Lyu M, Gharakheili HH, Sivaraman V. A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection. *IEEE Access* (2024) 12:89363–83. doi:10.1109/access.2024.3419068

5. Okoye CC, Nwankwo EE, Usman FO, Mhlongo NZ, Odeyemi O, Ike CU. Securing financial data storage: A review of cybersecurity challenges and solutions. *Int J Sci Res Archive* (2024) 11(1):1968–1983. doi:10.30574/ijsra.2024.11.1.0267

6. Atadoga A, Sodiya EO, Umoga UJ, Amoo OO. A comprehensive review of machine learning's role in enhancing network security and threat detection. *World J Adv Res Rev* (2024) 21(2):877–886. doi:10.30574/wjarr.2024.21.2.0501

7. Chen R, Mou Y, Li W. A provably secure multi-server authentication scheme based on Chebyshev chaotic map. *J Inf Security Appl* (2024) 83:103788. doi:10.1016/j.jisa.2024.103788

8. Akinsanya MO, Ekechi CC, Okeke CD. The evolution of cyber resilience frameworks in network security: a conceptual analysis. *Computer Sci & IT Res J* (2024) 5(4):926–949.

9. Miao J, Wang Z, Wu Z, Ning X, Tiwari P. A blockchain-enabled privacy-preserving authentication management protocol for Internet of Medical Things. *Expert Syst Appl* (2024) 237:121329. doi:10.1016/j.eswa.2023.121329

10. Mahmood K, Ghaffar Z, Farooq M, Yahya K, Das AK, Chaudhry SA. A security enhanced chaotic-map based authentication protocol for internet of drones. *IEEE Internet Things J* (2024) 11:22301–9. doi:10.1109/jiot.2024.3379930

11. Miao J, Wang Z, Wang M, Garg S, Hossain MS, Rodrigues JJ. Secure and efficient communication approaches for Industry 5.0 in edge computing. *Computer Networks* (2024) 242:110244. doi:10.1016/j.comnet.2024.110244

12. Shukla S, Patel SJ. A design of provably secure multi-factor ECC-based authentication protocol in multi-server cloud architecture. *Cluster Comput* (2024) 27(2):1559–80. doi:10.1007/s10586-023-04034-6

13. Barman S, Chattopadhyay S, Samanta D. A lightweight authentication protocol for a blockchain-based off-chain medical data access in multi-server environment. *SN Computer Sci* (2024) 5(3):292. doi:10.1007/s42979-024-02660-4

14. Prabhakar NVSS, Talari S, Jangirala S, Vangapa P. Security analysis of two authentication and key agreement protocols based on multi-server architecture. In: 2023 4th International Conference on Intelligent Technologies (CONIT). IEEE (2024). p. 1–7. doi:10.1109/CONIT61985.2024.10626922

15. Lee TF, Chang IP, Huang WJ. *A Privacy-preserving Authenticated Key Agreement Scheme based on Physically Unclonable Functions for Multi-server Architecture*. IEEE Transactions on Services Computing (2024).

16. Lamport L. Password authentication with insecure communication. *Commun Acm* (1981) 24(11):770–2. doi:10.1145/358790.358797

17. Tsaur WJ, Wu CC, Lee WB. A flexible user authentication for multi-server internet services. In: *First International Conference on Networking Colmar:the series Lecture Notes in Computer Science*. France: Springer Berlin Heidelberg (2001). p. 174–83.

18. Li LH, Lin LC, Wang MS. A remote password authentication scheme for multiserver architecture using neural networks. *IEEE Trans Neural Networks* (2001) 12(6):1498–504. doi:10.1109/72.963786

19. Lin IC, Hwang MS, Li LH. A new remote user authentication scheme for multi-server architecture. *Future Generation Computer Syst* (2003) 19(1):13–22. doi:10.1016/s0167-739x(02)00093-6

20. Yoon EJ, Yoo KY. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The J Supercomputing* (2013) 63(1):235–55. doi:10.1007/s11227-010-0512-1

21. Kim H, Jeon W, Lee K, Lee Y, Won D. *Cryptanalysis and improvement of a Biometrics-based multi-server authentication with key agreement scheme*, 451. IACR Cryptology ePrint Archive (2011).

22. Kalra S, Sood S. Advanced remote user authentication protocol for multi-server architecture based on ecc. *J Inf Security Appl* (2013) 18(2-3):98–107. doi:10.1016/j.jisa.2013.07.005

23. Guo D, Wen F. Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. *Wireless Personal Commun* (2014) 78(1):475–90. doi:10.1007/s11277-014-1762-7

24. Gupta PC, Dhar J. Hash based multi-server key exchange protocol using smart card. *Wireless Personal Commun* (2016) 87(1):225–44. doi:10.1007/s11277-015-3040-8

25. Li Y, Zhiming Z, Khurram KM. Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments. *Plos One* (2018) 13(3):e0194093. doi:10.1371/journal.pone.0194093

26. Wang D, Zhang X, Zhang Z, Wang P. Understanding security failures of multi-factor authentication schemes for multi-server environments. *Comput Security* (2020) 88(Jan.):101619–13. doi:10.1016/j.cose.2019.101619

27. Martínez-Peláez R, Toral-Cruz H, Parra-Michel JR, García V, Mena LJ, Félix VG, et al. An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors* (2019) 19(9):2098. doi:10.3390/s19092098

28. Yu SJ, Park KS, Park YHY. A secure lightweight three-factor authentication scheme for IoT in cloud computing environment. *Sensors* (2019) 19:3598. doi:10.3390/s19163598

29. Wong MK, Hsu CL, Le TV, Hsieh MC, Lin TW. Three-factor fast authentication scheme with time bound and user anonymity for multi-server e-health systems in 5g-based wireless sensor networks. *Sensors* (2020) 20(9):2511. doi:10.3390/s20092511

30. Tsai CH, Su PC. The application of multi-server authentication scheme in internet banking transaction environments. In: *Information systems and e-business management* (2020). p. 1–29.

31. Sudhakar T, Natarajan V, Gopinath M, Saranyadevi J. An enhanced authentication protocol for multi-server environment using password and smart card. *Wireless PersonalCommunications* (2020) 115:2779–803. doi:10.1007/s11277-020-07462-4

32. Sahoo SS, Mohanty S, Majhi B. An improved and secure two-factor dynamic ID based authenticated key agreement scheme for multi-server environment. *Wireless Personal Commun* (2018) 101:1307–33. doi:10.1007/s11277-018-5764-8

33. Xia M, Li S, Liu L. A secure three-factor authenticated key agreement scheme for multi-server environment. *Comput Mater Continua* (2020) 64(3):1673–89. doi:10.32604/cmc.2020.010177

34. Akram MA, Ghaffar Z, Mahmood K, Kumari S, Agarwal K, Chen CM. An anonymous authenticated key-agreement scheme for multi-server infrastructure. *Human-centric Comput Inf Sci.* (2020) 10:22. doi:10.1186/s13673-020-00227-9

35. Wu TY, Lee ZY, Obaidat MS, Kumari S, Kumar S, Chen CM. An authenticated key exchange protocol for multi-server architecture in 5G networks. *IEEE Access* (2020) 8:28096–108. doi:10.1109/access.2020.2969986

36. Wu F, Li X, Xu L, Sangaiah AK, Rodrigues JJ. Authentication protocol for distributed cloud computing: An explanation of the security situations for internet-of-things-enabled devices. *IEEE Consumer Electronics Mag* (2018) 7(6):38–44. doi:10.1109/mce.2018.2851744

37. Km A, Wa A, As A, Altaf I, Lodhi MA, Islam SH. An enhanced and provably secure multi-factor authenticationscheme for Internet-of-Multimedia-Things environments. *Comput & Electr Eng* (2020) 88:106888. doi:10.1016/j.compeleceng.2020.106888

38. Hsu CL, Le TV, Lu CF, Lin TW, Chuang TH. A privacy-preserved e2e authenticated key exchange protocol for multi-server architecture in edge computing networks. *IEEE Access* (2020) 8:40791–808. doi:10.1109/access.2020.2976431

39. Miao J, Wang Z, Ning X, Shankar A, Maple C, Rodrigues JJ. A UAV-assisted authentication protocol for internet of vehicles. *IEEE Trans Intell Transportation Syst* (2024) 25(8):10286–97. doi:10.1109/tits.2024.3360251

40. Tanveer M, Chelloug SA, Alabdulhafith M, El-Latif AAA. Lightweight authentication protocol for connected medical IoT through privacy-preserving access. *Egypt Inform J* (2024) 26:100474. doi:10.1016/j.eij.2024.100474

41. Xiao N, Wang Z, Sun X, Miao J. A novel blockchain-based digital forensics framework for preserving evidence and enabling investigation in industrial Internet of Things. *Alexandria Eng J* (2024) 86:631–643. doi:10.1016/j.aej.2023.12.021

42. Chen C, Guo H, Wu Y, Gao Y, Liu J. A novel two-factor multi-gateway authentication protocol for WSNs. *Ad Hoc Networks* (2023) 141:103089. doi:10.1016/j.adhoc.2023.103089

43. Lu Y, Li L, Peng H, Yang X, Yang Y. A lightweight ID based authentication and key agreement protocol for multi-server architecture. *Int J Distributed Sensor Networks* (2015) 11(3):635890. doi:10.1155/2015/635890

44. Xue K, Hong P, Ma C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *J Computer Syst Sci* (2014) 80(1):195–206. doi:10.1016/j.jcss.2013.07.004

45. Li X, Xiong L, Ma J, Wang W. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *J Netw & Computer Appl* (2012) 35(2):763–9.

46. Cui J, Yu J, Zhong H, Wei L, Liu L. Chaotic map-based authentication scheme using physical unclonable function for internet of autonomous vehicle. *IEEE Trans Intell Transportation Syst* (2022) 24(3):3167–81. doi:10.1109/tits.2022.3227949

47. Tanveer M, Alasmary H, Kumar N, Nayak A. SAAF-IoD: secure and anonymous authentication framework for the internet of drones. *IEEE Trans Vehicular Technology* (2024) 73(1):232–44. doi:10.1109/tvt.2023.3306813