Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9149730, 14 pages
https://doi.org/10.1155/2018/9149730

Research Article

Efficient Multifactor Two-Server Authenticated Scheme under Mobile Cloud Computing

Ziyi Han[1], **Li Yang**[1], **Shen Wang**[2], **Sen Mu**[2] **and Qiang Liu**[3]

[1]*School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2]*Aisino Corporation, Beijing 100195, China*
[3]*Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081, China*

Correspondence should be addressed to Ziyi Han; nnthzy@163.com

Received 28 April 2018; Accepted 9 August 2018; Published 12 September 2018

Academic Editor: Jian Shen

Because the authentication method based on username-password has the disadvantage of easy disclosure and low reliability and the excess password management degrades the user experience tremendously, the user is eager to get rid of the bond of the password in order to seek a new way of authentication. Therefore, the multifactor biometrics-based user authentication wins the favor of people with advantages of simplicity, convenience, and high reliability. Now the biometrics-based (especially the fingerprint information) authentication technology has been extremely mature, and it is universally applied in the scenario of the mobile payment. Unfortunately, in the existing scheme, biometric information is stored on the server side. As thus, once the server is hacked by attackers to cause the leakage of the fingerprint information, it will take a deadly threat to the user privacy. Aiming at the security problem due to the fingerprint information in the mobile payment environment, we propose a novel multifactor two-server authenticated scheme under mobile cloud computing (MTSAS). In the MTSAS, it divides the authentication method and authentication means; in the meanwhile, the user's biometric characteristics cannot leave the user device. Thus, MTSAS avoids the fingerprint information disclosure, protects user privacy, and improves the security of the user data. In the same time, considering user actual requirements, different authentication factors depending on the privacy level of authentication are chosen. Security analysis proves that MTSAS has achieved the authentication purpose and met security requirements by the BAN logic. In comparison with other schemes, the result shows that MTSAS not only has the reasonable computational efficiency, but also keeps the superior communication cost.

1. Introduction

With the vigorous development of the mobile Internet, the cloud computing service based on the mobile terminal has emerged. The mobile cloud computing is the deep fusion of the mobile Internet and the cloud computing, which represents the future trend in the development of the cloud computing [1]. Compared with the traditional cloud services, the mobile cloud service has the characteristics of mobile interconnection, flexible terminal application, and convenient data access [2]. However, the abundant mobile cloud service applications also bring more problems of security and privacy [3]. In the meanwhile, the mobile computing environment also puts forward the new requirements of security, convenience, and privacy protection.

Standing on the user's point of view, the traditional approach of the user authentication is based on the username and the password. In the past, the username and the password are selected to log in their account for the simplicity and the facility. But in recent years, owing to the massive popularity of the mobile terminal, people would prefer to put more and more works in the mobile terminal. In consequence, there exist more and more accounts to be managed. Research result shows that, on average, each person has 25 accounts and 6.5 passwords and logs in eight times one day. The complex password is difficult to remember again, so it is inevitable for people to use a simple and weak password and even share the same password in different network services. In case that the user's password is deceived by the phishing site or intercepted by the virus and Trojan horse, users' personal

information will be compromised. Thereby, it will threat the user account and reveal the user privacy. And also, the password itself has a lot of insurmountable defects. In the static password way, the password leakage (offline dictionary attack, etc.) and the overlapping library will greatly threaten the security of the user data. In the dynamic password way, it is low reliable for short message to authenticate users. In order to reduce the dependence on the password, there appears a mass of alternative strong authentication method [4], such as the authentication method based on the USB key [5] or the security chips [6] and the biometric-based authentication method. These strong authentication methods win the favor of the user for its convenience and strong security. In particular, the unique feature of the biometric-based authentication reinvests it with higher security [7].

In fact, in some critical systems which need the high security level of the authentication [8], such as the mobile financial environment which is related to the user's property safety, a single factor authentication method is insufficient to guarantee the security and the reliability of the authentication. In the circumstances, these factors can be combined together to establish a multifactor authentication method to ensure the strong security. The authentication factors can include the password, the token, and the biological characteristics. The mulitifactor authentication has been widely used in practice. For example, Alibaba, JDcom, and PayPal have fully supported the payment based fingerprint. The FIDO alliance, established in February 2013, also focuses on the multifactor market and devotes to establish a unified multifactor mobile authentication standard.

From the server's side, generally, peoples employ a single server to store user data and authenticate users previously. The user's passwords or the verification data of the passwords are stored in a server [9]. In this way, once the server is captured by an attacker, all passwords and the verification data stored in the server will be stolen by the attacker. Hence, this would be a serious threat to the security of the user data and leads to the leakage of users' privacy [10]. In order to overcome the inherent defect of the single server, the multiserver authentication scheme is proposed [11]. As a result, the risks will be distributed to multiple servers. In this way, the attacker must capture multiple servers in the same time to acquire user data. Consequently, the multifactor can greatly improve the security of the user data. But on the other hand, in the existing multiserver authentication scheme, when the cloud server verifies the user identity, they need to collect user's personal privacy information in the register phase, such as a password and biometric information. Then this personal privacy information are transferred via the link transmission and stored in the server side to verify the user identity. So user's personal privacy information is able to be stolen both in the process of transmission and storage. And some cloud service providers may leak even sell user's privacy information on account of business interests. As a result, the user increasingly mistrusts the cloud. Although the nonrepeatability and the uniqueness of the biometric information such as the fingerprint bring unique safety for the user, it also means that the leakage of the biometric information will have the disastrous consequence for the user to threaten the user privacy seriously. As is known to all, fortune and misfortune are neighbors. The biometric information can uniquely mark the user identity; unfortunately at the same time, it also brings the higher level security risks [12]. Once the server is hacked by attackers to cause the leakage of the biometric information, it will take a deadly threat to the user privacy. It is in urgent need of solving this problem.

Therefore, the user prefers to store the biometric information in the local device rather than in the cloud server to ensure the security of the biometric information. And also, in the current mobile device, it is widely equipped with the security mechanism, such as the trusted execution environment TEE, security chips (TPM chip, SE chips, etc.). Thus the user authentication information such as the biometric information can be stored in the trusted zone to ensure the security of the user privacy. Moreover, fingerprint information has become the most widely used biological feature [13]. And fingerprint authentication function has become the standard practice of mobile phone with one thousand yuan. So the fingerprint is chosen as the authentication factor to strengthen the authentication level in the existing biometric-based authentication schemes [7, 12, 14–16]. And the fingerprint is stored in the server. Absolutely the leakage of the fingerprint information in the server will bring serious consequences.

In allusion to the serious security problem of the fingerprint leakage in the mobile payment environment, we propose a novel multifactor two-server authenticated scheme under mobile cloud computing, shorted as MTSAS [17]. In the MTSAS, the server authenticates the device, and the device verifies the user. In the meanwhile, the user's biometric characteristics are stored locally in the user device and cannot be stolen by attackers [18]. The server side never stores the user's fingerprint information. Specially, the authentication server is deployed by the private cloud [19]. This way can download the security threat that the authentication server may face. Thus, MTSAS avoids the fingerprint information disclosure, protects user privacy, and improves the security of the user data. Moreover, the user requirements are given the full consideration. The different authentication factors depending on the different security levels are selected to make the reasonable use of the server's resources. For the more, MTSAS without introducing a third party is a lightweight protocol. Therefore, MTSAS is more suitable for the mobile payment environment.

Last but not least, the authentication server is applied in the private cloud environment. As a result, the public key of the authentication will not be broadcasted in the whole Internet but just be broadcasted in the private cloud. Consequently, the material's costs will be reduced and MTSAS is easier to be industrialized.

2. Related Work

In order to improve the security of the key exchange protocol, Pointcheval and Zimmer [20] proposed a multifactor authentication protocol with three factors, the password, the security device, and the biological characteristic. They also established

a security model and proved that it is secure in the random oracle security model. Tiwari [8] put forward a multifactor authentication system based on the Transaction Identification Code (*TIC*) and the Short Message Service (*SMS*) for the wireless payment scenario. The protocol is divided into layers so as to provide a highly secure environment, which is easy to use and deploy, and does not need to change any infrastructure or the protocol of the wireless network. Layeghian [21] focused on issues of the customer privacy in the wireless payment protocol. The identity of the customer is hidden through a blind pseudorandom signature certificate and an anonymous bank account. This scheme achieves the anonymity of the customer identity.

With the emergence of the cloud environment, Khan [22] targeted on the multifactor authentication problem in the cloud environment. For the sake of the user's security and privacy, they implemented a verification system. This system is combined with the built-in human factors (handwritten signature biometrics) and the standard knowledge factors (user's specific password) to achieve a high level of security.

A remote login two-factor scheme [23] based on the smart card was proposed in the multiserver architecture. This scheme takes advantages of the smart card as the second factor with the password to verify the user identity together. Li [24] found Chang's scheme [23] cannot resist the smart card lost attack, the check value reveal attack and the session key reveal attack. Specific to this a few security threats, an improved two-factor authentication scheme [24] is further proposed in the multiserver network.

Shen [15] pointed at the multiserver environment in the critical system; they put forward a multifactor authentication scheme with the authentication factors of the password, biological characteristics, and the random numbers. This scheme combines the multiple factors and multiple servers. Later, Li [16] discovered that Shen's scheme in the [15] was vulnerable to the denial of service attacks. The biological templates are directly stored in the smart card with no anonymity. Therefore, the loss of the biological templates will cause the direct threats to the security of the user data. Regarding this defects, an obfuscator is presented to enhance the security of the biological templates for the multiserver environment in the critical system. It is difficult to guarantee the security of the biological templates. Anyway, the biological templates are still stored in the server.

3. BAN

3.1. Basic Principle. The BAN [25] is a type of formal logic analysis method based on knowledge and belief. BAN starts with the initial basic beliefs of the protocol executive, according to every participant's issuing and receiving messages. Then it concludes the participants' eventual beliefs through the formal axioms and the logic reasoning.

When BAN is selected to verify a specific protocol, first of all, we need to idealize protocol messages and transfer them into the formulae in the BAN. Then we carry on the reasonable assumptions on the basis of the specific situation and infer the idealized messages based on the inference rules. Finally, we deduce whether the protocol can achieve the expected goal. If, at the end of the protocol, we are able to build the trust such as sharing communication key with the other identity, it can prove that the protocol is secure; otherwise, the protocol may suffer from security vulnerabilities.

Table 1: Basic concepts of BAN.

Symbols	Meaning
A	The specific main body
k_a	The specific public key of a
$k_{a^{-1}}$	The corresponding private key of a
P, Q, R	Any main body
X, Y	Any sentence
k	Any key
P believes X	P believes X
P sees X	P has received X
P said X	P has sent X
P controls X	P has jurisdiction of X
(X, Y)	X links to Y
fresh(X)	X is fresh
$\{X\}_k$	The result of X encrypted by K
$P \overset{k}{\leftrightarrow} Q$	P and Q communicate with each other by shared key K
$H(X)$	P and Q X is one-way hash function
$\overset{k}{\rightarrow} P$	K is the pubic key of P
$P \overset{X}{\rightleftharpoons} Q$	X is the secret value between P and Q

3.2. Basic Concept. First of all, Table 1 shows several basic concepts of BAN.

3.3. Logic Axioms. BAN has 17 axioms in total. We list some important axioms as follows:

(1) Message meaning rule logic axiom:

$$\frac{P \text{ believes } Q \overset{k}{\leftrightarrow} P,\ P \text{ sees } \{X\}_k}{P \text{ believes } Q \text{ said } X} \tag{1}$$

This logic axiom means that if P believes that K is the shared key between P and Q and P has received the result of X encrypted by K, P believes that Q has sent X.

(2) To public key, there is similar axiom:

$$\frac{P \text{ believes } \overset{k}{\rightarrow} Q,\ P \text{ sees} \{X\}_k{}^{-1}}{P \text{ believes } Q \text{ said } X} \tag{2}$$

(3) Temporary value verification rule logic axiom:

$$\frac{P \text{ believes } fresh(X),\ P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X} \tag{3}$$

This logic axiom means that if P believes that X is fresh and P believes that Q has sent X, P believes that Q believes X.

(4) Jurisdiction rule logic axiom:

$$\frac{P \text{ believes } Q \text{ controls } X,\ P \text{ believes } Q \text{ believes } X}{P \text{ believes } X} \quad (4)$$

This logic axiom means that if P believes that Q has the jurisdiction of X and P believes that Q believes X, P believes X.

(5) Logic axiom:

$$\frac{P \text{ believes } fresh(\mathrm{X})}{P \text{ believes } fresh(\mathrm{X}, \mathrm{Y})} \quad (5)$$

This logic axiom means that if the part of the formula is fresh, the whole formula is fresh.

(6) Logic axiom:

$$\frac{P \text{ believes } \overset{k}{\rightarrow} P,\ P \text{ sees } \{\mathrm{X}\}_{\mathrm{k}}}{P \text{ sees } \mathrm{X}} \quad (6)$$

This logic axiom means that if P receives messages which are encrypted by the public key itself, P is able to decrypt the received messages.

The security of MTSAS is proved from some basic beliefs to final beliefs using above axioms. If final beliefs can meet the security requirement, MTSAS is secure in theory. Unless, there may exit some security loopholes in MTSAS.

4. MTSAS

4.1. Scenario. Mobile payment scenario is a high security level authentication environment involving user's property security. Therefore, the authentication protocol in this scenario should provide high efficiency and strong authentication. The user has registered for an account in the bank by his/her mobile terminal before. When the user wants to login the system through the mobile terminal, there exists the difference of the user's behavior by means of analyzing the user's behavior. The user's behavior can be divided into two groups. One group is that the user logs in the system only to check the account information, not to process the transaction operation. Another group is that the user logs in the system to perform the financial transaction exactly. Therefore, in order to improve the user experience and the user's efficiency, it is essential to provide the authentication with different security levels aiming at this two kinds of user behaviors in the mobile payment scenario.

Because the former user's behavior does not involve the transaction information, this situation has low security requirements of the user authentication. So only providing the weak authentication, namely, the basic authentication is enough. Therefore, the traditional username-password authentication method is suitable to verify the user's identity in this situation.

For the latter type of the user's behavior, the user conducts the financial transaction operation indeed, so the strong authentication must be provided to ensure the security of the user's property and privacy information. Therefore, on the basis of the basic identification, by adding the authentication factors of the Dynamic Verification Code (*DVC*) and the fingerprint (*fp*), the security of the mobile payment process can be strongly enhanced in the way of the multifactor authentication way. At the same time so as to avoid the too concentrated risk problem of the single server and the privacy leakage problem once the single server is compromised, and there is high security requirement in the mobile payment scenario. As a result, it is essential to adopt the two-server way to share risks and ensure the robustness and the stability of the service.

Finally, the user's fingerprint information is stored locally (such as the trusted execution environment based on TPM, SE, and so on), thereby to prevent the potential security hazard on account of the fingerprint information leakage in the server side. The scenario is shown in Figure 1.

4.2. Model Description. In our system, there are two main participating entities, the mobile terminal and the cloud. The user, the fingerprint module and the user agent are located in the mobile terminal. And there are the web server and the authentication server in the cloud. The fingerprint module is embedded in the mobile terminal's chips (TPM chip, SE chip, and so on) and is protected by the trusted execution environment (the ARM TrustZone and so on). As a result, the security issues in the fingerprint module can be guaranteed.

The security model we assumed is as follows. On the one hand, the web server and the user are exposed in the open network environment. They not only should ensure the security of the interaction, but also achieve the purpose of authentication. On the other hand, the authentication server is deployed in the private cloud environment. We consider that the web server and the authentication server have been authenticated before; the authentication problem between them is not considered. And especially, they will not collude. Consequently, we only focus on the security of data transmission between the web server and the authentication server in the MTSAS.

4.3. The Proposed Scheme. The proposed authentication scheme is divided into two phases, the registration phase and the authentication phase. Before we describe the MTSAS in detail, the meanings of the symbols are presented in Table 2. There are some details needed to be described. In the basic authentication phase, *AuthLevel* is low; and *AuthLevel* is high in the transaction authentication phase.

We assume that the mobile terminal has a fingerprint module, and the fingerprint can be stored securely. In other words, there is the secure area in the mobile terminal, such as the trusted execution environment TEE based on the security chip TPM, SE, TrustZone, etc. The MTSAS includes five participants: the user, fingerprint module, the user agent, the web server, and the authentication server. The users, the fingerprint module, and the user agent are located in the mobile terminal. The fingerprint module is responsible for the fingerprint's collection, comparison, and secure storage. The user agent can be the APP or the browser. The web server is in charge of the communication with the user agent, and

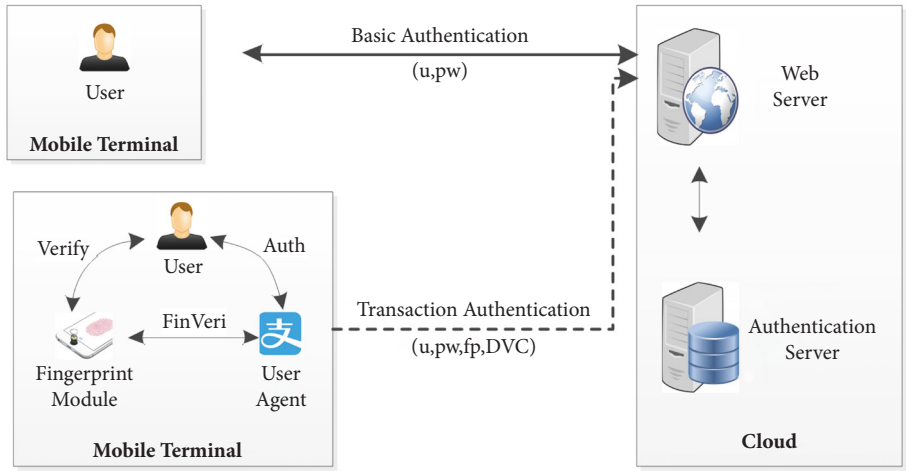

Figure 1: Scenario.

Table 2: Symbols meanings.

Symbols	Meaning
U	User
WS	Web server
AS	Authentication server
u	Username
pw	Password
phn	Phone number
fp	Fingerprint
DVC	Dynamic verification code
n	Temporary number
hpw	The hash of the password
(PK_x, SK_x)	The public/ private key pair of X
K	Session key
$E_K(M)$	Encrypt the message M by the key K
$Sig(M)$	Signature of M
$h()$	Hash function
$AuthLevel$	Authentication Level, can be High/ Low
$BaResult$	Basic authentication result, can be Success/ Failure
$TranResult$	Transaction authentication result, can be Success/ Failure
$LocResult$	Local authentication result, can be Success/ Failure

the forwarding of the authentication data. The authentication server is deployed in the private cloud environment. The authentication is performed by the authentication server. During this process, the user information is stored in the database of the authentication server.

4.3.1. Registration. In the registration phase, the user delivers $\{u, pw, phn\}$ to the web server. The web server forwards the message to the authentication server. The authentication server preserves these data and makes the username u as the index. On the other hand, the user registers locally. That is to say, the fingerprint module registers the user. The user enters the fingerprint to the fingerprint module. If there exists the fingerprint in the fingerprint module, the fingerprint only needs to associate with the user account and stores the hash value of the fingerprint in the trusted execution environment TEE.

For ensuring the security of the communication environment, we assume that the two pairs, the mobile terminal and the web server and the web server and the authentication server both have their own authentication key pair and have obtained the other's public key. The key of the mobile terminal is $\{PK_u, SK_u\}$. The key pair of the web server is $\{PK_{WS}, SK_{WS}\}$. The authentication server also has the key pair $\{PK_{AS}, SK_{AS}\}$. K is the session key between the mobile terminal and the web server.

4.3.2. Authentication. In the authentication phase, we provide the authentication scheme based on the privacy level, which is separated into the basic authentication and the transaction authentication.

When the user just logs in their account to check the account information, we choose the basic authentication scheme of the username-password to provide the authentication of the weak security level. For another, when the user needs to be trade online, we select the transaction authentication scheme by adding the authentication factors-the fingerprint and DVC to support the strong authentication.

(A) Data Transmission. To avoid giving unnecessary details, the data transmission of the user in the mobile terminal, the web server, and the authentication server is abided by the following method.

(1) User ⟷ Web Server. First, the sender encrypts the delivering data by the session key K and signs them with its own private key. Then the sender encrypts the session key K by receiver's public key.

After the receiver accepts, the receiver verifies the signature with the sender's public key. If the verification fails, the authentication fails. Otherwise, the sender passes the verification. Next, the receiver decrypts the session key K by its own private key and then decrypts the delivering data by the session key. As in the following, the receiver obtains the delivering data.

(2) Web Server ⟵⟶ Authentication Server. The sender encrypts the delivering data by the receiver's public key and signs them with its own private key.

After the receiver accepts, the receiver verifies the signature with the sender's public key. If the signature does not pass the verification, the authentication fails. Otherwise, the sender passes the verification. Next, the receiver decrypts the delivering data by its own private key to obtain the delivering data.

(B) Basic Authentication. In this phase, the user log in his account with the method of the basic authentication using the username and his/her password only. The specific process is represented in Figure 2.

A user U enters the username u and the password pw in the user agent of the mobile terminal. The mobile terminal sets the authentication level "*AuthLevel*" to low and generates the temporary number n_u and n_w. And then the mobile terminal sends u, pw, the identifier of the web server *WS*, *AuthLevel*, and n_u, n_w to *WS*. *WS* forwards these data to *AS*. After *AS* receives, *AS* retrieves stored *hpw*, and computes the hash *h(pw)* of the received pw. Then *AS* checks whether $h(pw)$ and hpw are equal. If so, *AS* sets the basic authentication result *BaResult* to "success". Otherwise, *AS* sets *BaResult* to "Failure". Afterwards, *AS* sends BaResult, n_u, and n_w to *WS*. *WS* forwards the data to U to feedback. After the mobile terminal of U receives, the mobile terminal checks the temporary number firstly. The mobile terminal compares the received temporary number with the stored temporary number. If they are not equal, it proves that the data has been expired. Otherwise, the mobile terminal reads *BaResult*. If *BaResult* is "success", the user agent displays "authentication success" to the user. Otherwise, the user agent displays "authentication fail".

(C) Transaction Authentication. When the user has passed the basic authentication and requires the online trading. The strong authentication is needed to verify the user identity. In consequence, the fingerprint and *DVC* are added as the authentication factors to ensure the strong security. When user requires the online trading, the fingerprint module triggers the local verification process and then triggers the *DVC* verification process. Figure 3 shows the specific process.

The mobile terminal sets *AuthLevel* to high and generates the temporary number n_{u1}. Then the mobile terminal triggers the local verification process; that is to say, the user agent delivers the fingerprint verification request to the fingerprint module. After the fingerprint receives, it verifies the user by alerting the user to enter the fingerprint and obtain the fingerprint information $fp*$. Then the fingerprint module retrieves the stored hash *hfp* of the fingerprint from the trusted execution environment TEE. After that the fingerprint module checks whether $h(fp*)$ and *hfp* are equal. If so, the fingerprint module sets the local verification result *LocResult* to "success". Otherwise, the fingerprint sets *LocResult* to "failure". If *LocResult* is "Failure", the fingerprint module repeats the local verification process. On the condition that the local verification fails three times consecutively; the authentication fails and stops the transaction authentication. Or else, the mobile terminal transmits U, *AuthLevel*, *LocResult*, and n_{u1} to *WS*. *WS* forwards the data to *AS*.

After *AS* receives, *AS* retrieves the stored *phn* in the database by the index of u. And then *AS* generates *DVC* and stores it in the database by the index of u. After that, *AS* sends *DVC* and to the mobile terminal by out-of-band way. After U receives, the user agent of the mobile terminal reads n_{u1} and checks whether the received temporary number and the stored temporary number are equal. If they are not equal, it proves that the data has been expired and the transaction authentication fails. Otherwise, the mobile terminal generates the temporary number n_{u2} and allows the user filling the received dynamic verification code $DVC*$ in the user agent. Afterwards, the mobile terminal sends $DVC*$ and n_{u2} to *WS*. *WS* forwards the data to *AS*. After *AS* receives, *AS* retrieves the stored dynamic verification code *DVC* and checks whether *DVC* and $DVC*$ are equal. If so, AS sets the transaction result *TranResult* to "Success". Otherwise, *AS* sets *TranResult* to "failure". At last, *AS* shows the transaction result *TranResult* to the mobile terminal like the basic authentication phase and returns the authentication result to the user.

5. Security Analysis

In our proposed scheme MTSAS, the security of the fingerprint is guaranteed by the TEE in the mobile terminal. And the reliability of the DVC ensured the reliability of the out-of-band transmission. Thus, the security of MTSAS depends on the data transmission way in the authentication phase. In MTSAS, the data transmission the user, the web server and the authentication server are followed by the data transmission way in the part 3. Therefore, we prove the security of the data transmission way by the BAN.

The security model shows that MTSAS has different security requirements for the data transmission between the user and the web server and the data transmission between the web server and the authentication server. For one thing, the data transmission between the user and the web server not only needs to guarantee the security of the data transmission but also meets the authentication needs. For another, the data transmission between the web server and the authentication server just needs to ensure the security of the data transmission. The meanings of symbols are shown in Table 3.

5.1. Modeling. Modeling the above process, we can get the following process. The temporary number and the key are related to the security. Therefore, we idealize them in the following message:

$$U \longrightarrow WS : \{n_u, n_w\}_{K_{uw}}, \{K_{uw}\}_{K_w}, \{n_u, n_w, K_{uw}\}_{Ku^{-1}} \quad (7)$$

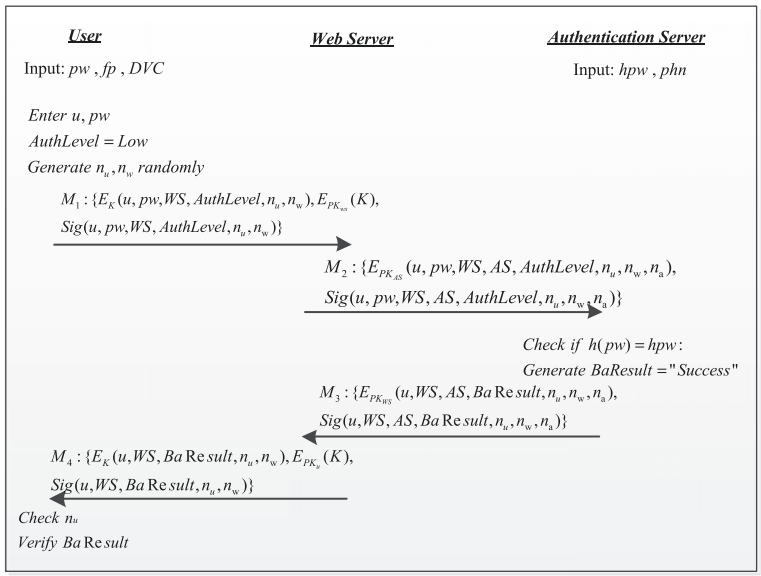

Figure 2: Basic authentication.

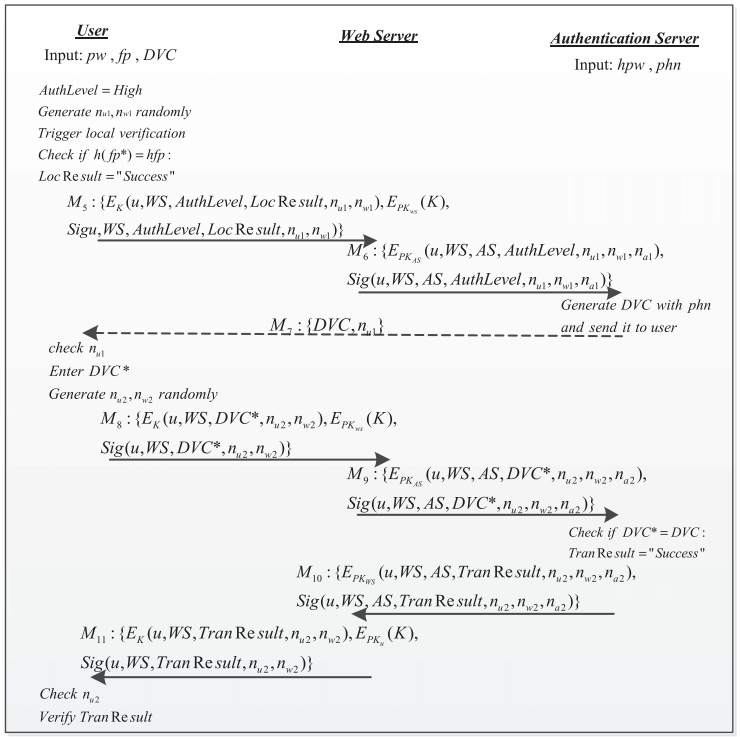

Figure 3: Transaction authentication.

$$WS \longrightarrow AS : \{n_u, n_w, n_a\}_{K_A}, \{n_u, n_w, n_a\}_{K_W{}^{-1}} \quad (8)$$

$$AS \longrightarrow WS : \{n_u, n_w, n_a\}_{K_W}, \{n_u, n_w, n_a\}_{K_A{}^{-1}} \quad (9)$$

$$WS \longrightarrow U : \{n_u, n_w\}_{K_{uw}}, \{K_{uw}\}_{K_u}, \{n_u, n_w, K_{uw}\}_{Kw^{-1}} \quad (10)$$

5.2. Protocol Analysis. The basic beliefs of MTSAS are as follows:

$$U \text{ believes } \xrightarrow{K_W} WS \quad (11)$$

$$WS \text{ believes } \xrightarrow{K_u} U \quad (12)$$

$$WS \text{ believes } \xrightarrow{K_A} AS \quad (13)$$

$$AS \text{ believes } \xrightarrow{K_W} WS \quad (14)$$

$$WS \text{ believes } U \text{ controls } U \xleftrightarrow{K_{uw}} WS \quad (15)$$

$$U \text{ believes } U \xleftrightarrow{K_{uw}} WS \quad (16)$$

Table 3: The meanings of symbols.

Symbol	Meaning
K_i	the public key of i
K_i^{-1}	the private key of i
$\{X\}K$	encrypts X by the key K
$\{X\}_{K_i^{-1}}$	signs X with the public key of i
K_{ij}	the session key between i and j

$$U\ believes\ fresh(n_u) \tag{17}$$

$$WS\ believes\ fresh(n_w) \tag{18}$$

$$AS\ believes\ fresh(n_a) \tag{19}$$

$$WS\ believes\ AS\ controls\{n_a\} \tag{20}$$

$$AS\ believes\ WS\ controls\{n_w\} \tag{21}$$

From beliefs (11), (12), (13), (14), U believes that K_w is the public key of WS, WS believes that K_u is the public key of U, WS believes that K_A is the public key of AS, and AS believes that K_w is the public key of WS. From belief (15), the session is sponsored by U, so WS believes that U has the jurisdiction of the session key K_{uw} between U and WS. From belief (16), U believes that K_{uw} is the session key between U and WS. From belief (17), U believes that n_u is fresh. From belief (18), WS believes that n_w is fresh. From belief (19), AS believes that n_a is fresh.

First, we analyze the data transmission between U and WS. By message (7) and the BAN logic axiom (6), we can obtain the following formula:

$$WS\ sees\{n_u, n_w, K_{uw}\} \tag{22}$$

By message (7), the basic belief (12), and the BAN logic axiom (2), we can receive the following formula:

$$WS\ believes\ U\ said\{n_u, n_w, K_{uw}\} \tag{23}$$

By the basic belief (18) and BAN logic axiom (6), we can gain the following formula:

$$WS\ believes\ fresh(n_u, n_w, K_{uw}) \tag{24}$$

By formula (23), formula (24), and the BAN logic axiom (3), we can get the following formula:

$$WS\ believes\ U\ believes\{n_u, n_w, K_{uw}\} \tag{25}$$

That is the formula

$$WS\ believes\ U\ believes\ U \overset{K_{uw}}{\longleftrightarrow} WS \tag{26}$$

By the basic belief (15), formula (26), and the BAN logic axiom (4), we can obtain the following formula:

$$WS\ believes\ U \overset{K_{uw}}{\longleftrightarrow} WS \tag{27}$$

By the message (10) and the BAN logic axiom (6), formula (28) can be received:

$$U\ sees\{n_u, n_w, K_{uw}\} \tag{28}$$

By message (10), the basic belief (11), and the BAN logic axiom (2), we can gain the following formula:

$$U\ believes\ WS\ said\{n_u, n_w, K_{uw}\} \tag{29}$$

By the basic belief (17) and the BAN logic axiom (5), formula (30) can be gained:

$$U\ believes\ fresh(n_u, n_w, K_{uw}) \tag{30}$$

By formula (28), formula (29), and BAN logic axiom (3), we can get the following formula:

$$U\ believes\ WS\ believes\{n_u, n_w, K_{uw}\} \tag{31}$$

That is formula (32):

$$U\ believes\ WS\ believes\ U \overset{K_{uw}}{\longleftrightarrow} WS \tag{32}$$

By the basic belief (16), we can obtain the following formula:

$$U\ believes\ U \overset{K_{uw}}{\longleftrightarrow} WS \tag{33}$$

Therefore, we can obtain the final beliefs: formulae (26), (27), (32), and (33). The final beliefs (27) and (33) are the beliefs in level 1. The final beliefs (26) and (32) are the beliefs in the level 2. Thus, the communication between the user and the web server not only achieves the purpose of the authentication, but also guarantees the security of the data transmission.

Next, we analyze the communication between WS and AS. By message (8) and the BAN logic axiom (6), we can receive the following formula:

$$AS\ sees\{n_u, n_w, n_a\} \tag{34}$$

By message (8), the basic belief (14), and the BAN logic axiom (2), formula (35) can be obtained:

$$AS\ believes\ WS\ said\{n_u, n_w, n_a\} \tag{35}$$

By the basic belief (19) and the BAN logic axiom (5), we can gain the following formula:

$$AS\ believes\ fresh(n_u, n_w, n_a) \tag{36}$$

By formula (35), formula (36), and the BAN logic axiom (3), formula (37) can be gotten:

$$AS\ believes\ WS\ believes\{n_u, n_w, n_a\} \tag{37}$$

That is formula (38) and formula (39):

$$AS\ believes\ WS\ believes\{n_w\} \tag{38}$$

$$AS\ believes\ WS\ believes\{n_a\} \tag{39}$$

By the basic belief (21), formula (38), and the BAN logic axiom (4), we can receive formula (40):

$$AS\ believes\ \{n_w\} \tag{40}$$

Similar to the analysis process to message (8), we can analyze message (9). Then, we can obtain formula (41) and formula (42):

$$WS\ believes\ AS\ believes\ \{n_w\} \tag{41}$$

$$WS\ believes\ \{n_a\} \tag{42}$$

As a result, we can obtain the final beliefs: formulae (39), (40), (41), and (42).

The final beliefs (40) and (42) are beliefs in level 1. The belief (40) means that *AS* believes that the random number n_w is really from *WS*. In other words, *AS* believes *WS*. And also, the result shows that *WS* believes that *AS* can be obtained similarly in the final belief (42).

The final beliefs (39) and (41) are beliefs in level 2. The belief (39) represents that *AS* believes that *WS* believes that the random number n_a is really from *AS* itself. That is to say, *AS* belief itself is believed by *WS*. And also, Belief (41) explains that *WS* belief itself is believed by *AS* similarly.

Therefore, the security of the data transmission between WS and AS can be ensured; the security of the proposed scheme MTSAS is proved.

6. Performance and Security Comparison

In this part, on the one hand, we compare the performance of the proposed scheme MTSAS with the Shen's scheme in [15] and the Li's scheme in [16] in the authentication stage. On the other hand, the security of three schemes is also compared. Because MTSAS has two types of authentication, there we compare the performance of our proposed scheme in the two aspects of MTSAS1 and MTSAS2. MTSAS1 is the basic authentication and MTSAS2 is the transaction authentication.

6.1. Performance Comparison. The performance of the MTSAS is measured in three aspects, the symmetric encryption/decryption computation time t_{sym}, the public key encryption/ decryption computation time t_{asym}, and the hash computation time t_{hash}.

The public key encryption/decryption computation time involves the exponential computation, which costs the most computational resources. Thus the number of the public key encryption/decryption computation will have crucial influence on the performance of a scheme. Because the symmetric encryption/decryption computation does not involve the exponential computation, typically it is related to addition, subtraction, multiplication, division, or some low order operations and costs less computational resources. The number of the symmetric encryption/decryption computation has less effect on the scheme's performance. The hash computation only refers to the one-way hash function operation and consumes the least resources. Therefore the number of the hash computation has the least impact on the performance. From [26], when the public key encryption/decryption computation adopts the ECC algorithm, the symmetric encryption/decryption computation adopts AES-128, and the hash algorithm adopts the SHA-1 algorithm, $t_{asym} \approx 7.3592ms$, $t_{sym} \approx 0.005ms$, $t_{hash} \approx 0.0004ms$. The detailed performance comparison is shown in Table 4.

Table 4: Performance comparison.

	Computational Overhead (ms)
Shen in [16]	$6t_{asym}+17t_{hash}=44.162$
Li in [25]	$6t_{asym}+22t_{hash}=44.164$
MTSAS1	$4t_{sym}+4t_{asym}+10t_{hash}=29.460$
MTSAS2	$6t_{sym}+6t_{asym}+16t_{hash}=44.284$

Shen's scheme [16] consists of public key encryption/decryption computations with 6 times and hash computations with 17 times in the authentication phase. Li's scheme [25] includes public key encryption/decryption computations with 6 times and hash computations with 22 times in the authentication phase. Our proposed scheme of the basic authentication MTSAS1 consists of public key encryption/decryption computations with 4 times, symmetric encryption/decryption computations with 4 times, and hash computations with 10 times in the authentication phase. Significantly, the number of the public key encryption/decryption computation in MTSAS1 is lower than those schemes of Shen and Li. Therefore, MTSAS1 has lower computational overhead, outstanding performance, and superior user experience. Last but not least, the transaction authentication MTSAS2 concludes public key encryption/decryption computations with 6 times, symmetric encryption/decryption computations with 6 times and hash computations with 16 times in the authentication phase. The MTSAS2 has the same number of the public key encryption/decryption computation as Shen's scheme and Li's scheme, but due to inserting the symmetric encryption/decryption computation. Therefore, the amount of computational time of the MTSAS2 is basically equal and to Shen's scheme and Li's scheme, just slightly higher. But the MTSAS2 mainly pays attention to the high security demand in the mobile payment scenario and enhances the security of the user's biometric characteristic. The MTSAS2 solves the security hidden danger due to the fingerprint information leakage and provides users with the strong authentication. As a result, the MTSAS2 improves the security when the user authenticates the identity. Besides, the computational resources that the symmetric encryption/decryption computation takes are limited. Therefore, the minor performance loss that the MTSAS2 takes can be acceptable

6.2. Security Comparison. In this part, we mainly focus on the security of three schemes. The detailed security comparison is presented in Table 5.

As is shown in Table 5, because these three schemes all adopt the signature to resist forgery; they all satisfy the forward security. So they can guarantee the message's confidentiality and integrity and is able to resist replay attack and key guessing attack. Since the biometrics template is

Table 5: Security comparison.

	Shen in [16]	Li in [25]	MTSAS
Message confidentiality	√	√	√
Message integrity	√	√	√
Resist key guessing attack	√	√	√
Resist replay attack	√	√	√
Forward Security	√	√	√
User anonymity	✗	√	√
Resist biometrics template attack	✗	√	√
Resist data leakage of the server	✗	✗	√
Two servers	✗	✗	√
Divide the security level	✗	✗	√

stored in the user's smart card in Shen's scheme. Therefore, it is vulnerable to biometrics template lost attack that the user's biometrics template can be retrieved if the smart card is stolen by an adversary. Of course, Shen's scheme also cannot meet the requirement of the user anonymity. On the contrary, on account that Li's scheme makes anonymity of the biological templates, it can meet user anonymity and resist biometrics template attack. The biological characteristics are in the trusted execution environment TEE in our proposed scheme. The biological characteristics never leave the local device. And when the server authenticates the user, what the server authenticates is the user device, not the user. The user biometric data never stored any data in the server side. Therefore, MTSAS can also satisfy the requirements of the user anonymity and is able to resist biometrics template attack.

But when the data in the server leaks, the biological characteristics of the user might be captured by the attackers to greatly threat user privacy. Unfortunately, the server stores the data related to the user biometric in the schemes of the Shen and the Li. Therefore, they both cannot resist this attack. Whereas any data related to the biological characteristics is never stored in the server side in MTSAS. So MTSAS is more superior to other two schemes in improving the security of the user data and protecting the user privacy. And compared with the former two schemes, MTSAS adopts the two-server authentication method to disperse security risks. So the stability and the robustness of the MTSAS are greatly improved. At the same time, the MTSAS analyzes user behavior in detail. According to the authentication scenario that the user is in, the MTSAS provides the authentication method with different security level. MTSAS provides the weak authentication in the basic authentication scenario and the strong authentication in the transaction scenario. Our proposed scheme MTSAS stands on the user's point to perfect the user experience and improves the authentication efficiency. Thus, the practicability and the feasibility of our proposed scheme MTSAS are both stronger

7. Experiment

In the experiment part, we realize FREDP based on the FIDO UAF framework. And then we test the performance of basic authentication and transaction authentication, respectively.

FIDO (Fast Identity Online) is an online authentication alliance sponsored by PayPal, Nok Nok, validity, Infineon, AGNITIO, and Lenovo in July 2012. And till September 2016, alliance members have amounted to 252. Specific to the problem of "isolated island" resulting from multiple authentication standards in the mobile identity authentication, FIDO dedicates to unified mobile authentication standard in order to reduce the user dependence to password. FIDO aims to solve the problem of security, convenience, and privacy and provides users with strong authentication. The FIDO architecture has been widely applied in the mobile payment area. Several companies such as Alipay, JD finance, and ICBC have fully supported FIDO. The FIDO framework includes standards of UAF and U2F. UAF (Universal Authentication Framework Protocol) is able to realize without password by binding users' biological characteristics with mobile device. UAF makes the authentication mode, server authenticates device and device authenticates user, come true, which greatly improves the convenience and the reliability of mobile identity authentication. U2F (2nd Universal Framework Protocol) provides the two-factor experience to users by adding the authentication information in hardware devices (such as U shield and mobile hardware information) as the second factor. So FIDO can make the authentication strong.

We implement FREDP based on the FIDO UAF framework. In the mobile terminal, in consideration of current authentication status and user requirements, our proposed scheme provides the authentication way of distinguishing security levels. The basic authentication adopts one-factor authentication way based on password. And the transaction authentication way adopts multifactor authentication way based on password, DVC, and fingerprint. In the server side, in order to guarantee the stability of the server and the security of authentication server, FREDP adopts the two-server mode. The web server is deployed prior to FIDO authentication server. This two servers are both deployed in the cloud environment.

7.1. Experiment Environment. Our experimental environment is as follows.

The mobile terminal adopts ZUK Z2125; the hardware is configured as CPU 2.35HZ ROM 64GB, RAM 6GB, and android version: 6.0.1. The cloud environment is the public cloud service, and the web server and authentication server are deployed on two cloud hosts with the same configuration. The cloud host is configured as Ubuntu 14.04 32-bit server version, with 1GB kernel, 1Mbps bandwidth, and 50G hard disk.

7.2. Experiment Process. In the concrete implementation process, we write code in JAVA language based on FIDO UAF framework. FREDP scheme is realized by FIDO server as authentication server and web server writing by ourselves.

Figure 4: Basic authentication schematic.

During the FREDP operation, in the mobileterminal, we conduct the basic authentication with the user test 03. The basic authentication schematic is shown in Figure 4. The transaction authentication schematic is shown in Figure 5.

In the public cloud, the log of the web server is shown in Figure 6, and the console log of the authentication server is shown in Figure 7.

After the realization of FREDP, we make experiment in four aspects. In order to ensure the reliability and the accuracy of the data, we test three times and average them for each class of data.

7.2.1. Experimental Target: Comparing the Total Authentication Time T_{sum} of Single Factor/Multifactor Authentication Method. We analyze the impact of increasing identity authentication factors under the two-server architecture to the total authentication time. In the FREDP, actually, the basic authentication is the method of the single factor authentication, namely, the password. The transaction authentication is the method of the multifactor authentication, namely, the password, the dynamic verification code, and the fingerprint. Thus, it is necessary to compare the authentication time between the basic identification and the transaction authentication. The test results are shown in Table 6. The timing interval of the authentication time in Table 6 is from the time that the mobile terminal collects all the authentication factors to the time that the mobile terminal receives the authentication result.

Table 6: Comparison of single factor/multifactor authentication.

	Basic Authentication (ms)	Transaction Authentication (ms)
T_{sum1}	252	364
T_{sum2}	277	381
T_{sum3}	283	406
T_{sum}	270.7	383.7

As shown in Table 6, due to the addition of dynamic verification code and the fingerprint as authentication factors, the authentication time of the transaction authentication increases to a certain extent. In the transaction authentication, the comparison inside the mobile terminal would take some time, such as the fingerprint comparison and attestation in the trusted execution environment. And at the same time the increase of the interaction times between the web server and the authentication server may cause the growth of the authentication time. But it is because of these verification and interactive process, the FREDP possesses high the high security that other schemes do not have. Our proposed FREDP avoids the fingerprint information disclosure, realizes fingerprint's local storage and local verification. But luckily, we can see in Table 6, the transaction authentication with multifactor only increases 1 second. It is absolutely acceptable for the user. So FREDP makes a good compromise between the convenience and the security. Therefore, it is feasible and necessary to adopt multifactor authentication in the key system.

7.2.2. Experiment Target: Comparing the Total Authentication Time in the Single Server/Two-Server Environment. In this section, we test the basic authentication and the transaction authentication, respectively. In the single server framework, the mobile terminal directly interacts with the authentication server, and all the forwarding and verification work are deployed in the authentication server side. There are the web server and the authentication server in the two-server framework. The web is responsible for communicating with the mobile terminal and forwarding the information. The authentication server is responsible for the verification and processing of the authentication process data. The test result of the total time T_{bsum} in the basic authentication under the single server/two-server authentication is, respectively, shown in Table 7. The test result of the total time T_{tsum} in the transaction authentication under the single server/two-server authentication is, respectively, shown in Table 8. Similarly, the timing interval of the authentication time in Tables 7 and 8 is, from the time that the mobile terminal collects all the authentication factors to the time that the mobile terminal receives the authentication result.

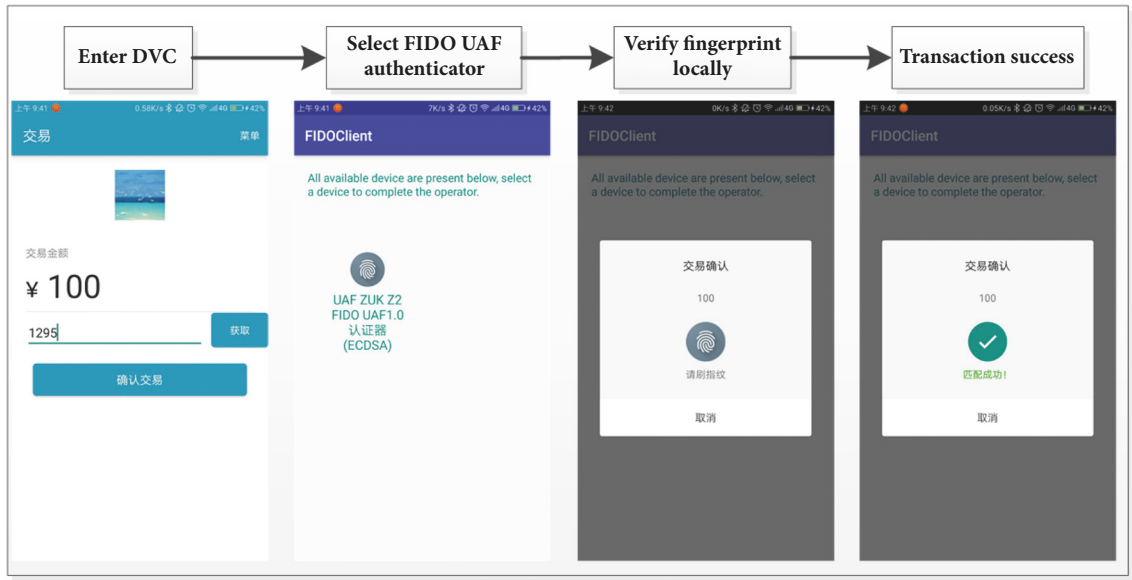

FIGURE 5: Transaction authentication schematic.

登录
2018-04-10 09:39:32,883 INFO [server.modules.user.web.TestLoginController] - username:test03,password:1234
2018-04-10 09:39:38,578 INFO [server.modules.user.web.TestLoginController] - Login OK!
注册请求
2018-04-10 09:40:12,797 INFO [server.modules.user.web.TestLoginController] - Get from client:{"op":"Reg","previousRequest":"test03"}
2018-04-10 09:40:21,784 INFO [server.modules.user.web.TestLoginController] - Get from authentication server:{"statusCode":1200,"uafRe
2018-04-10 09:40:22,003 INFO [server.modules.user.web.TestLoginController] - Send to client:ReturnUAFRequest [statusCode=1200, uafReq
AgU21nbmFOdXJlIEFsZ29yaXRobTogUOhBMjU2d2l0aFJTQSwgTO1EIDOgMS4yLjgOMC4xMTM1NDkuMS4xLjExCgogIEtleTogIFN1biBSUOEgcHVibGljIGtleSwgMjAOOCBi
k6IFtGcm9tOiBUaHUgTWF5IDA0IDEz0jQ5OjU4IENTVCAyMDE3LAogICAgICAgICAgICBUbzogTW9uIE1heSAwNCAxMzo0OTo1OCBDU1QgMjAyMFOKICBJc3N1ZXI6IENOI
AwMDAwMDEwMC5jcmxdCiAgIENSTEl zc3VlcjpbQO49Y2EtMTQ1LCBDPUNOXQpdXQoKWzRdOiBPYmplY3RJZDogMi41LjI5LjE1IENyaXRpY2FsaXR5PW5hbHNlCktleVVzYWdl:
AwNDA6IEI3IDA4IEI2IDg4IEZDIDcxIEQwIDkxICAgNjcgQkMgNDkgMkY gOTY gQTkgOTcgOTkgIC4uLi4uc54uZy5JLy4uLi4NCjAwNTA6IEJDIDI4IENCIDIyIE Q5IEJEIEE4:
ZBIEIyIDE2IDQ3ICAgOEUgOEQgQjgg OTUgMjcgQTAgNjggMjAgIEwuLi5qLi5HLi4uLicuaCANCjAwRjA6IEVEIERGIEVFIDAzIDc3IDAzIEQyIDNBICAgNUEgQTAgMEEgRUY g(
注册响应
2018-04-10 09:40:53,837 INFO [server.modules.user.web.TestLoginController] - Get from client:{"uafResponse":"{\"uafProtocolMessage\":'
3YW5nQG5va25va2xhYnMuY24wWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAAQsCSTWZx3DdOKol3w2HRM6ww_SqWqiaK8XFKGRBDQeW7bmHbFbaLzMdRwuicqTrpG1Q94pmo-hD_
UTRNekl5TVRjNE16WTRPRF16TURjeU9ESTNPVFl5TmpJeE9UY3hNekOOT1RVeU1UVTBORGswTORNME5ETTJNakE1T1RnNU9EQXhPRFkxTmprNU1qVTNOVGN6TURrMk56ZzBNak
WTFPRFkOTWpZNU1qUXdNall5TURRME5ERUtJQOJ3ZFdKc2FXTWdaWGh3YjI1bGJuUTZJRFkxT1RNMONpQWdWbUZzYVdScGRIazZJRnRHY205dE9pQ1VhSFVnVFdGNU1EQTBJRE
NDFMakk1TGpNeElFTnlhWFJwWTJGc2FYUjVQV1poYkhObENrT1NURVJwYzNSeWFXSjFkR2x2YmxCdmFXNTBjeUJiQ21BZ1cwUnBjM1J5YVdKMWRHbHZibEJ2YVc1ME9nb2dJQOI
QTFRUOF4TONBZO1TNHVMaTVoSVMOdUxpNTBMQzVhTGcwSO1EQXhNRG9nTkRNZO5EWWdOVFVnTORZZO1DQWdJQOFnSUNBZO1DQWdJQOFnSUNBZO1DQWdJQOFnSUNBZO1DQWdJQOI
NHVMaTR1TGkOdVNDNE5DakF3TmpBNkIEVTVJRFJDSURkQklFRTRJRGszSURBNElEbEVJRGd4SUNBZ1JVUWdOVVFnUVRZZO5rTWdRamNnTXprZ1JUZ2dOa1FnSUZsTGVpNHVMaTI
VWdOekFnTVRVZO1FWWdRaO1nUTBNZO1DNWJMaTR1VmkOckxtRjFjQzR1TGkOTkNqQXdSVEE2SURSRE1FTTFJRGxCSURFMO1EWkJJRU15SURFMklEUTNJQOFnTOVVZO9FUWdRam
2018-04-10 09:41:03,966 INFO [server.modules.user.web.TestLoginController] - Send to client:{"statusCode":1200,"description":"OPERATI(
认证请求
2018-04-10 09:41:50,826 INFO [server.modules.user.web.AuthenticationController] - Get from client:GetUAFRequest [op=Auth, previousReq
2018-04-10 09:41:51,197 INFO [server.modules.user.web.AuthenticationController] - Get from authentication server:ReturnUAFRequest [st
n\":0,\"attachmentHint\":0,\"tcDisplay\":0,\"authenticatorVersion\":0}]}]}]","additionalData":""}, op=Auth, lifetimeMillis=300000]
2018-04-10 09:41:51,197 INFO [server.modules.user.web.AuthenticationController] - Send to client:{"statusCode":1200,"uafRequest":"{\"
"001A#2121\\\"], \\\"keyIDs\\\":[\\\"A4BR1VZvYmeUW3EfHwKM6uo_1wzQUP_ZA_IC5QHeg9U\\\"], \\\"userVerification\\\":0, \\\"keyProtection\\\":(
4NDMxMjA5NTk3NzYyNzg5MzA5MjYwMTg5ODQwNjA3NzM2MDc2OTY2OTEOMzE5NDY5OTU2NDMxMDk5NDkxMDA1MDc5NzIyNDk4NTMOMjU4NzA3NTQxMzcOMjY2NjE1ODg4ODAyO(
gei4uLgOKXQpdCgpbMlO6IE9iamVjdElkOiAyLjUuMjkuMTkgQ3JpdGljYWxpdHk9ZmFsc2UKQmFraWNDb25zdHJhaW5OczpbCiAgQOE6ZmFsc2UKICBQYXRoTGVuOjIxNDcOO(
gICAgICAgICAgICAgICAgICAgQOZVLgOKXQpdCgpdCiAgQWxnb3JpdGhtOiBbUOhBMjU2d2l0aFJTQVOKICBTaWduYXR1cmU6CjAwMDA6IDkxIEY5IEJEIEZEIEY1IDY1IEVFIDQyI(
5IDE2IEIwICAgMUYgNjggQjEgMTEgREUgNTAgMOIgOUIgIC5KLi4rLi4uLmguLi5QOy4NCjAwQTA6IEIyIEQOIEU1IDc2IEMyIEIwIEFCIDU1ICAgMDYgODUgNDYgRkIgODkgRI
认证响应
2018-04-10 09:42:26,401 INFO [server.modules.user.web.AuthenticationController] - Get from client:{"uafResponse":"{\"uafProtocolMessa
CVGRXSnFaVO4wT21CRFRqMTNkM2N1Wm1sa2IzUmxjM1F1WTI5dExDQkRQVU5FQ2lBZ1UybG5ibUYwZFhKbE1FRnNaMjl5YVhSb2JUb2dVMGhCTWpVMmQybDBhRkpUVVN3Z1QwbI
Jd01ERTNOREUyTURVMU9EWTFNemN3T1RneE1qUTJNakUyT1RjMk1qZ3pNamd5IXpJME5qWXhPVGcyTnpFek9UVTROVEF3TkRrMO1UWTBNVEkOT1Rnek9UZzVNRGN3T1RJMU56W2
FZO1EVWdPVGNnUVVNZO1DQWdJQOFnSUNBZO1DQWdJQOFnSUNBZO1DQWdJQOFnSUNBZO1DQWdlaTR1TGcwS1hRcGRDZ3BiTlYwNkdFOW1hbVZqZEVsaO9pQ
WdWRBb2dJRVJoZEdGZ1JXNWphWEJvWlhKdFpXNTBDaUFnUzJWNVgwRm5jbVZsYldWRBcGRDZ3BiTlYwNkdFOW1hbVZqZEVsaO9pQX1MalV1TWprdU1UUWdRMOpwZEdsallXe
ZZ1FqRWdNRU1nTjBJZO56SWdOak1nSUVBdUxpNHVMbGN1VWk1VOxpNHVjbU1OQ2pBd05EQTZJRUkzSURBNElFSTJJRGc0SUVaRE1EY3hJRVF3SURreE1DQWdOamNuUWtNZO5Ea
KQklEazBJREOOSURWRO1EbEdJQOFnT0RJZO5FWWdSRUVnUmtJZ1EwUWdSalFnTlRNZO5FRWdJQzR1TGkOdU9GOHVMazh1TGkOdVUwbO5DakF3UXpBNklFUXpJRUO:SURCQO1FT
IcE9TR2h4VG1rMA\\\", \\\"upv\\\":{\\\"major\\\":1, \\\"minor\\\":0}}]}\"}
2018-04-10 09:42:26,590 INFO [server.modules.user.web.AuthenticationController] - Send to client:{"statusCode":1200,"description":"OPI

FIGURE 6: Log of the web server.

As shown in Tables 7 and 8, whether in the basic authentication or the transaction authentication, the total authentication time in the two servers takes 1 seconds more than in the single server. It is acceptable for the user. And the mode of the two-server authentication shares the authentication risk, which avoids all verification information leakage because the server is attacked in the single server. We adds a layer of "shelter" to improve the security of the verification information and the reliability of the authentication process. Therefore, the design of the two servers is necessary and reasonable.

8. Conclusion

In allusion to the security problem of the fingerprint leakage in the mobile payment environment, we propose a novel multifactor two-server authentication scheme under mobile

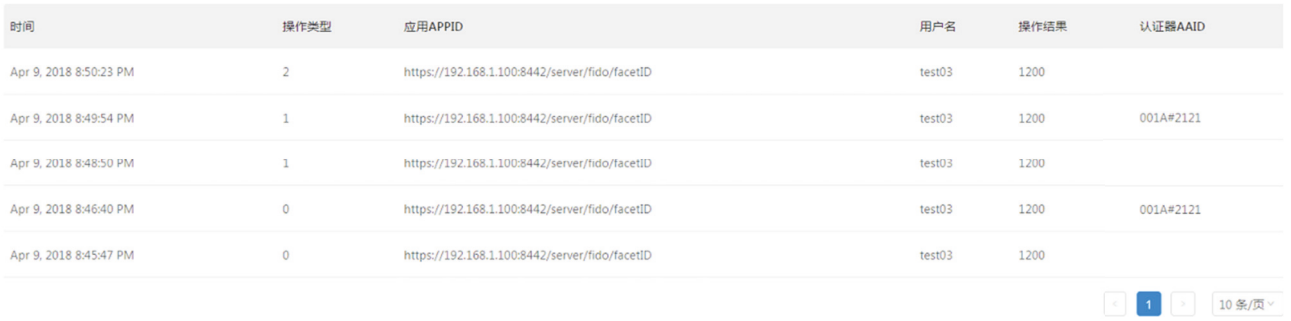

Figure 7: Console log of the web server.

Table 7: Comparison of single factor/multifactor authentication in the basic authentication.

	Single Server (ms)	Two Servers (ms)
T_{bsum1}	191	252
T_{bsum2}	193	277
T_{bsum3}	196	283
T_{bsum}	193.3	270.7

Table 8: Comparison of single factor/multifactor authentication in the basic authentication.

	Single Server (ms)	Two Servers (ms)
T_{tsum1}	288	364
T_{tsum2}	228	381
T_{tsum3}	295	306
T_{tsum}	270.3	383.7

computing MTSAS. In the MTSAS, the user's biometric characteristics cannot leave the user device. And the server side never stores the user's fingerprint information. Particularly, the authentication server is applied by the private cloud. The use of the two servers can obviously lower the security risk of server attack. Moreover, we stand on the user's shoes. MTSAS provides the different authentication factors depending on the privacy level of the authentication. Thus our work has a certain degree of contribution to the mobile payment security.

Unfortunately, the problem of the loss of the user device still troubled us. In this conditions, apparently, it may take the problem of data redundancy in the cloud when the user registration again. We will pay more attention to the authentication problem in this part.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by National Natural Science Foundation of China (61671360 and 61672415), the Key Program of NSFC-Tongyong Union Foundation under Grant U1636209, the National Key Basic Research Program (2017YFB0801805), the Key Program of NSFC Grant U1405255, the Natural Science Basic Research Plan in Shaanxi Province of China (2017JM6082), and the Opening Project of Science and Technology on Communication Networks Laboratory (KX172600024)

References

[1] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013*, pp. 655–659, Sardinia, Italy, July 2013.

[2] W. Song and X. Su, "Review of Mobile cloud computing," in *Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN '11)*, pp. 1–4, Xi'an, China, May 2011.

[3] V. Chang, Y.-H. Kuo, and M. Ramachandran, "Cloud computing adoption framework: A security framework for business clouds," *Future Generation Computer Systems*, vol. 57, pp. 24–41, 2016.

[4] G. Mathew and S. Thomas, "A novel multifactor authentication system ensuring usability and security," *International Journal of Security, Privacy and Trust Management*, vol. 2, no. 5, pp. 21–30, 2013.

[5] D.-J. He, M.-D. Ma, Y. Zhang, C. Chen, and J.-J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Communications*, vol. 34, no. 3, pp. 367–374, 2011.

[6] G. Wang, Q. Liu, J. Zhou, and J. Z. Chen, "A multi-factors identity authentication scheme in classified environment," *Advanced Materials Research*, vol. 765-767, pp. 1734–1738, 2013.

[7] A. Bhargav-Spantzel, A. C. Squicciarini, S. Modi, M. Young, E. Bertino, and S. J. Elliott, "Privacy preserving multi-factor authentication with biometrics," *Journal of Computer Security*, vol. 15, no. 5, pp. 529–560, 2007.

[8] A. Tiwari, S. Sanyal, A. Abraham et al., "A multi-factor security protocol for wireless payment - secure web authentication using mobile devices," *Computer Science*, 2011.

[9] X. Yi, S. Ling, and H. Wang, "Efficient two-server password-only authenticated key exchange," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1773–1782, 2013.

[10] Y. Yang, R. H. Deng, and F. Bao, "A practical password-based two-server authentication and key exchange system," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 2, pp. 105–114, 2006.

[11] C. Lee, T. Lin, and R. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863–13870, 2011.

[12] D. F. L. Souza, A. M. F. Burlamaqui, and G. L. S. Filho, "A multi factor authentication approach based on biometrics, optical interference and chaotic maps," *IEEE Latin America Transactions*, vol. 15, no. 9, pp. 1700–1708, 2017.

[13] H. Saevanee, N. L. Clarke, and S. M. Furnell, "Multi-modal behavioural biometric authentication for mobile devices," *IFIP Advances in Information and Communication Technology*, vol. 376, pp. 465–474, 2012.

[14] P. Baraki and V. Ramaswamy, "Biometric authentication of a user using online dynamic signature," in *Proceedings of the 2nd International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2016*, pp. 576–581, Bangalore, India, July 2016.

[15] H. Shen, C. Gao, D. He, and L. Wu, "New biometrics-based authentication scheme for multi-server environment in critical systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 6, pp. 825–834, 2015.

[16] X. Li, K. Wang, J. Shen, S. Kumari, F. Wu, and Y. Hu, "An enhanced biometrics-based user authentication scheme for multi-server environments in critical systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 3, pp. 427–443, 2016.

[17] Z. Han, L. Yang, and Q. Liu, "A Novel Multifactor Two-Server Authentication Scheme under the Mobile Cloud Computing," in *Proceedings of the 2017 International Conference on Networking and Network Applications (NaNA)*, pp. 341–346, Kathmandu, Nepal, October 2017.

[18] A. Roy, N. Memon, and A. Ross, "MasterPrint: exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2013–2025, 2017.

[19] Y. Xue, Y.-a. Tan, C. Liang, Y. Li, J. Zheng, and Q. Zhang, "RootAgency: A Digital signature-based root privilege management agency for cloud terminal devices," *Information Sciences*, vol. 444, pp. 36–50, 2018.

[20] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *International Conference on Applied Cryptography and Network Security*, vol. 5037 of *Lecture Notes in Computer Science*, pp. 277–295, 2008.

[21] S. Layeghian Javan and A. Ghaemi Bafghi, "An anonymous mobile payment protocol based on SWPP," *Electronic Commerce Research*, vol. 14, no. 4, pp. 635–660, 2014.

[22] S. H. Khan and M. A. Akbar, "Multi-factor authentication on cloud," in *Proceedings of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7, Adelaide, Australia, November 2015.

[23] C.-C. Chang and T.-F. Cheng, "A robust and efficient smart card based remote login mechanism for multi-server architecture," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 8, pp. 4589–4602, 2011.

[24] C. T. Li, C. Y. Weng, and C. I. Fan, "Two-factor user authentication in multi-server networks," *International Journal of Security & Its Applications*, vol. 6, no. 2, 2012.

[25] M. Burrows, M. Abad, and M. Needham, "A logic of authentication," *Proceedings of the Royal Society A Mathematical, Physical and Engineering Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[26] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of Medical Systems*, vol. 39, no. 2, article 10, 2015.