Iraqi Journal of Information and Communication Technology(**IJICT**)
Vol. 5, Issue 3, December 2022

www.ijict.edu.iq

ISSN:2222-758X
e-ISSN: 2789-7362

MOBILE CLOUD COMPUTING SERVICES AUTHENTICATION SCHEME

Safana A. Abdulrahman [1], **M. F. Al-Gailani** [2]
[1,2] College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
safana4auf@gmail.com [1], m.falih@coie-nahrain.edu.iq [2]

Received:31/10/2021, Accepted:28/11/2021
DOI:**10.31987/ijict.5.3.201**

Abstract- **With the rapid growth of cloud computing and the expansion of mobile phone users in recent years, mobile cloud computing has attracted wide attention. Wireless computing networks are the basis of data sharing between mobile devices and cloud services in the mobile cloud. Since air is the communication medium, it must be properly protected; otherwise, it will be subject to a variety of security threats, for example, attacks from middle-man, identity tracking, etc. Furthermore, mobile devices are limited in storage, resources, and computing powers. Hence, designing an efficient and secure balance of authentication schemes is extremely important. First of all, a multi-factor authentication scheme based on biometric (fingerprint information), hash function, and fuzzy vault algorithm is presented in that paper. Secondly, the Validation and Analysis tool of AVISPA Security was approved. Finally, the proposed scheme's security is compared to that of other related schemes.**

keywords: **Mobile cloud computing, Authentication, Biometrics, Fuzzy vault.**

I. Introduction

Due to the rapid development and expansion of wireless communication technologies and the popularity or increasing reliance on portable devices (such as smartphones and tablet computers) , users of these devices can access cloud services on the go. This greatly helps to make everyday life more convenient as many kinds of network services are available wherever and anytime. However, the increasing demand for high-quality services from users requires a great many data to be processed on their mobile devices immediately. However, the resources of mobile devices have limitations and cannot provide the requirements of all users [1], [2]. This weakness has been considered as a performance bottleneck for various mobile device applications. Cloud computing has developed very rapidly in recent years and continues today, as one of the most important networking technologies. Cloud computing provides users with affordable and convenient pay-for-use services via visualization technology [3], [4]. For example, users can get any kind of cloud services like storage services from any service provider (CSPs) like Google and Baidu. An emerging cloud service is developed to track the trend of expanding cloud services and meet the needs of users. Mobile computing has been integrated with cloud computing platforms, and this new service is called Mobile Cloud Computing (MCC). This integration successfully helped address the issues of resource-constrained portable devices. Distributed MCC is used in practical applications increasing the types of MCC services, as many cloud service providers can provide users with various types of cloud services [5], [6]. Fig. 1 shows a typical MCC service architecture. One of the most important challenges in the MCC services environment is the security issues since wireless technology is used to transmit all messages. Ensuring MCC services are available to lawful users only, strong authentication should be used to improve data security while accessing the cloud service. During the last years, 2Factor Authentication (2FA) has been utilized to resist attacks and solve the problem of password difficulty by utilizing an SMS token or biometric information with a smart card. [7]-[9]. Although 2FA schemes are in constant development, they still encounter problems and attacks including fake fingerprint capture from original, insider attacks, and lost/stolen smartcard attacks. The design of an authentication Multi-Factor System (MFA) is therefore important to increase the levels of safety by verifying the user's validity. MFA is a power source for defending the system from unauthorized users and reducing the risk of malicious attacks.

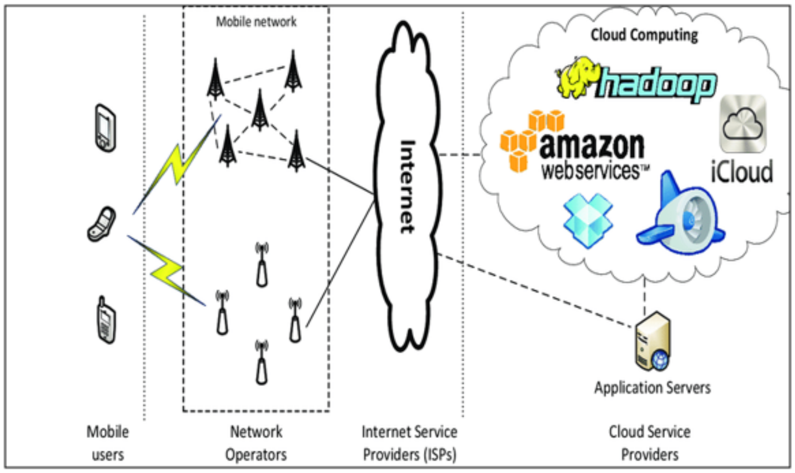


Figure 1: Architecture of MCC services

The outline of this paper is as follows: Section II provides a survey of mobile cloud computing system authentication. Section III illustrates the contributions of the proposed scheme. Section IV introduces the threat model. Section V explains the used notations and the cryptographic concepts. Section VI presents the proposed scheme. Section VII provides an informal security analysis of the proposed scheme. Section VIII presents the formal security analysis. Section IX shows the comparison between the proposed scheme and other related schemes.

II. Literature Survey

Over the last ten years, many authors suggested Various mobile cloud architecture authentication schemes. In this section, the relevant studies in the field of mobile cloud computing and authentication protocol are presented. In 2017 Ziyi Han, Li Yang, and Qiang Liu proposed a two-server Multi-Factor Authenticated Scheme (MTSAS) [10]. Researchers use a username and a password at MTSAS to make the user and the web server authenticated via the authentication server. In addition to using the fingerprint factor, they relied on keeping fingerprint information inside the mobile device within a trusted execution environment and not sharing it with the server. This scheme may seem robust in terms of providing security for fingerprint information, but if an adversary steals or guesses a user's password, he /she can gain access to the account since the scheme does not rely on fingerprint information to achieve verification with the server. In the same year, the Mobile Cloud Computing Environment Framework is suggested by Fati S. A [11]. A mobile device, cloud server, and identity management server are provided in the framework. Mobile devices track user behavior after encryption as login credentials using the homomorphic signature, this signature is then sent to the identity management server for checking. If successful, a new token is generated and delivered to the mobile device and cloud server. The cloud server then coincides with the two tokens. If matches are found, the user has permitted cloud server access. The proposed framework safeguards against adversaries the identity of mobile cloud users and eliminates unauthorized cloud access, but only use 2FA, which cannot be secure enough. In 2018 Dey, S. Q. Ye, and S. Sampalli [12] proposed a message digest, timestamp, and location-based authentication scheme. The scheme is different from traditional authentication schemes, which are based on the use of client passwords and ID as authentication parameters. To achieve mutual authentication between the cloud server and the mobile client, the scheme employs message digest (MD) client, mobile device location (l), and timestamp (tc, ts). Furthermore, asymmetric key encryption is used during the authentication phase and consumes less power on the mobile device than asymmetric key encryption, making it an energy-efficient scheme. The confidentiality of sensitive personal information of mobile phone users is constantly in threat if it is used throughout the Process of authentication while connecting to the network. This is why authors Chean L.T, V. Ponnusamy, and S.M. Fati. [13] Proposed a multi-authentication process that Removes the usage of a password or any important credentials and utilizes the identity that the identity manager and client jointly construct. Identity is produced under both parties' identity management server and the client agreement, depending on the instructions given by the IDM, the identity used for authentication might be updated consecutively or randomly each time. Besides, for the partial identity that the client generates and sends to the cloud, homomorphic encryption is employed. At the end of this scheme, a single identity is utilized during the whole authentication process to transmit information, and the value of identities has subsequently continued to alter with the encryption of homomorphic encryption for each new attempt at authentication. Going beyond the classical authentication model, Zeroual, A, et al. [14] proposed a model that incorporates deep-face recognition for authentication. The model's first step concerns the system's user, who takes a picture of his or her face and transmits it to the cloud. The second portion of the model is on the cloud side, this is where the deep neural network is set up to authenticate. However, there was no picture encryption, making this approach useful for potential image formation attacks. J. Sun et al. [15] proposed a lightweight multi-factor scheme that combines the biometrics of users processed using a fuzzy extractor algorithm and smartphone information to achieve mutual user and remote server authentication. This algorithm uses a lightweight encryption algorithm to reach fast authentication. The scheme does not use an encryption algorithm, so the authors suggested the use of a lightweight encryption algorithm for future work. In 2019, Abuarqoub A. et al. [16] proposed safe and trusted, smart card and password authentication methods through which the user submits his/her data to the server during the registration stage and subsequently sends the user a smart card. The smart card is later used for authentication, which may contain some sensitive and general security parameters. This scheme may ensure that the server can successfully verify the user who has the corresponding password and a valid smart card, but it is unable to solve the problem of malicious users and cancel the lost card. Mo J. et al. proposed an effective and provenly safe authentication of anonymous users [16]. In this scheme, an improved elliptic curve cryptosystem is used to create a secure client/server channel, also a fuzzy extractor algorithm for fingerprint information with the user identity information, and a smartcard used for mutual authentication. Chen H. et al. [18] developed a system based on a three-factor (fingerprint, username, and password) and fuzzy extractor for processing the user's Bio-information with a physical smart cart that uses XOR operations and a hash function.

The three protocols [15]-[18] involved the use of a fuzzy extractor algorithm, which required that one of the generated parameters be kept confidential in the mobile device or smart cart. Since both are vulnerable to lost/stolen attacks. Even if the information inside is irreversible, it may cause the loss of user accounts and disable access to data services. In 2020, Ahmed A. A, et al. [19] proposed the DRmcc protocol. Multifactor authentication, Diffie-Hellman key exchange, and One-Time Password (OTP) are used in the protocol to achieve mutual validation and authentication. The OTP is generated dynamically and does not need to be entered by the user for each new connection. It is a combination of password, username, and international mobile equipment number metrics with an SMS code number. These metrics act as Diffie-Hellman-shared parameters so that the sum of these metrics is the P-value while the G value is their count. In this way, Diffie Hellman's parameters will be calculated in the same way for both parties without sharing any parameters. This will keep them safe from a man-in-the-middle attack; Because DRmcc requires the current session OTP to be stored on both the mobile device and the registration cloud server, this may be vulnerable if the mobile device is lost or stolen. The authors suggested this as future work.

III. Contributions of The Proposed Scheme

1) Design and implement a secure authentication scheme to authenticate legal users in a mobile cloud computing system. This research supports features that previous research hasn't covered, such as avoiding stolen/lost smart mobile devices and card attacks. Furthermore, a secure TLS protocol is used to create a secure private channel with new encryption keys at each phase.
2) Analyze the security of the proposed work using formal analysis and thus Validate the proposed scheme with a well-known and reliable security proofing tool Automated Validation of Internet Security Protocols and Applications (AVISPA).
3) Based on security studies and well-known attack resistance, this paper compares the suggested approach to other relevant works.

IV. Analysis Threat Model

The Dolev-Yao [20] was used as a model of threat in our analysis, and it is described as follows:

1) The adversary can intercept, replay, modify and delete messages that are sent by public channels.
2) If a user's mobile or smart card is stolen, every parameter saved on one of them is taken and utilized in the attachment.
3) A privileged insider attack can access the parameters saved in the cloud server database.

V. Cryptography Concepts

A. Secure Sockets Layer / Transport Layer Security Encryption Protocol

This protocol secures application traffic and enables secure end-to-end communication by operating at the top of the transport layer. The absence of this protocol leaves the communication channel vulnerable to eavesdropping and subsequent modification of information. By exchanging some parameters (such as random number, session ID, cipher suite, compression techniques, etc.) , the SSL / TLS handshake protocol is used to establish a secure session. There are two states each session that is maintained by them. The first is the session state which deals with some parameters such as the session identification, X509 certificate, compression algorithms, specifications for cipher and master key, etc. While the parameter for connection is included the MAC secrets sent to the server and client, initialization vectors, sequence numbers, etc. [21].

B. Fuzzy Vault Algorithm

The fuzzy vault algorithm is used to secure the chosen secret value within a generated set of biometrics. By using Euclidean distance, the fuzzy vault is fault-tolerant (Yu, J, et al.) [22]. It can be thought of as a type of fault-tolerant cryptographic procedure. It generally consists of two stages: The phases of locking and unlocking [23]. The following two algorithms are included in the locking phase:

- The generation algorithm: Gen$(FP, K, P) = LP$

  The user chooses the secret value (K) that provides his or her fingerprint to generate the biometric template (FP). And the algorithm generates the polynomial (P) based on the character number of the secret value K encoded into Pol, and the Pol of every element in B_i is assessed for a set point (LP).
- The encryption algorithm: Enc $(C, LP) = Vu$

  The algorithm generates a lot of randomness chaff points (C) that are not laying on p as a noise point to provide redundancy to the final vault (Vu).

To recover the secret value K, the unlocking phase executes the two algorithms listed below:

- the decryption algorithm:Dec $(FP*i, Vu) = P*$

  Users enter their biometrics to generate a new fingerprint template value $(FP*i)$, then the previously stored value of the vault (Vu) is used to calculate the preceding polynomial value $(P*)$.
- the reconstruction algorithm: Rec $(P*) = K$

  The $(P*)$ value serves as the input to the Rec $(\cdot)$ function to obtain the secret value $(K*)$. Satisfies $(FP*i)$ only: $|FPi - FP*i| \ll \varepsilon$, where $FPi - FP*i = \{b | b \in FPi, b \notin FP*i\}$, is the fuzzy parameter, and the restored $K* = K$. Since $FP*i$ overlaps substantially FPi, $FP*i$ identifies a lot of points LP placed on P and is capable to recover a set of exact points on P, Therefore a set of exact points can be retrieved. The user can then recover the P accurately.

C. Secure Hash Algorithm

SHA is a function in binary data (i.e. , bit strings) whose output length is always constant. SHA3 is a sponge function family that is considered by two different parameters, bitrate (r) and the capacity (c.) the total (r + c) specifies the SHA-3 width, the sponge construction employs a permutation function, where the maximum value is constrained by 1600 bits. The choice of r and c depends on the preferred hash output value. The SHA-3 procedure consists of three stages: the initialization stage defining the all-zero state matrix (A). The absorbing stage, during which each message's r-bit extensive block is XORed with the existing matrix state, and 24 rounds of compression functions are performed. The state matrix is simply squeezed to the desired length of the hash output during the squeezing stage [24].

VI. Proposed Scheme

A robust multi-factor authentication scheme based on a Personal Identification Number (PIN), mobile phone identification number, and user fingerprint is proposed in this section. It is based on the use of an app created for a smart mobile device. The scheme is made up of three major components: authentication server (AS), system users $(U_1....U_i)$, and smart mobile device (SMD_i). The scheme consists of four different phases: setup phase, registration phase, login phase, and account retrieving phase. The phase of registration is carried out only once, while the phase of login is carried out whenever (U_i) wants to access the system, and the account retrieving phase works whenever (U_i) wants to change the smart mobile device or update the PIN. Finally, the setup phase is carried out for all three phases.

A. Setup Phase

This phase is being carried out through the two SMD_i applications and the AS using the TLS/SSL protocol to initiate a secure channel that encrypts all messages between the two parties using the shared secret key generated by the handshake process of the protocol.

B. Registration Phase

At the registration phase, when the user wants to create a new account, he relies on his SMD_i to do this operation using a specific mobile application that connects to the AS. The following steps are executed:

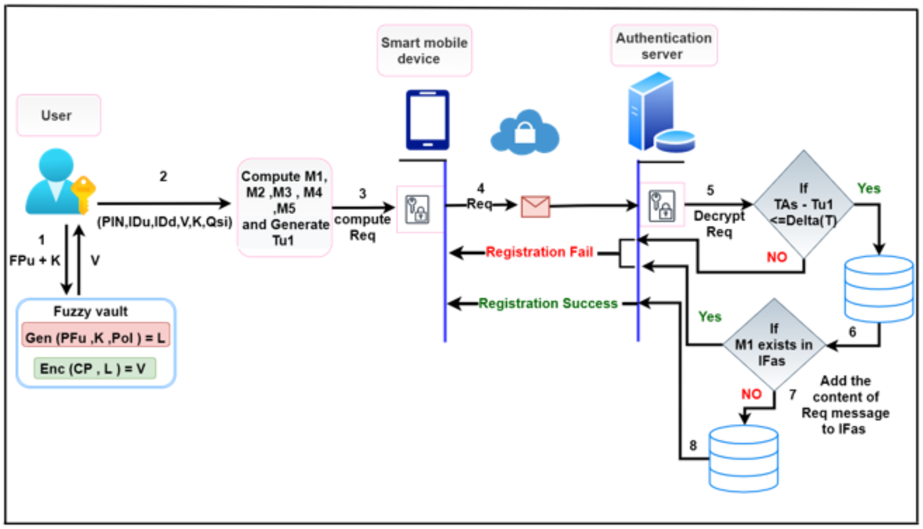


Figure 2: User registration phase

Step 1. The user (U_i) passes his identification number (ID_{ui}), that is, a personal government card, in addition to a 4-digit PIN. The value of the device identity number (ID_{di}) is then automatically extracted from the SMD. All these values are used to calculate the following message:
$M_1 = h(ID_{di}||ID_{ui}||PIN)$

Step 2. The user's fingerprint (FP_{ui}) is captured using an external fingerprint scanner that serves as input to the generator function of the fuzzy vault algorithm with a 6-digit key value(K): Gen$(FP_{ui}, K, Pol) = L$. The output value from the Gen function plus the random chaff points (CP) will act as input to the (Enc) function to produce the encrypted vault: $(CP, L) = V$, then the message is generated:
$M_2 = h(K||ID_{ui})$, $M_3 = V$.

Step 3. Security questions, which are four personal and easy-to-remember questions, that each user (U_i) answers, are used for account retrieval and update. Two messages are generated by taking the hash value of the answers after they are concatenated with PIN and ID_{ui} as follows:
$M_4 = h(Qs_1||Qs_2||Qs_3||Qs_4||PIN||ID_{ui})$.
$M_5 = h(Qs_1||Qs_2||Qs_3||Qs_4||ID_{di}||ID_{ui})$.

Step 4. Finally, SMD_i generates a timestamp value (T_{u1}) to be added to the previously generated messages and sent as a registration request (R_{rq}) over a secure TLS channel encrypted by the shared key (Sk) and user private key (Pr_{ui}) as follows: $R_{rq} = E(Sk, E(Pr_{ui}, (M_1, M_2, M_4, M_3, M_5, T_{u1})))$.

Step 5. Upon receipt a message from the user (U_i), AS carrying out the following:
The AS first decrypts the message (R_{rq}) with the shared key and then uses the user's public key (U_i) to validate the user. Then AS checks the timestamp $(T_{AS} - T_{u1} \leq \Delta T)$ if it meets the conditions, AS checks the value of M_1 in the index file (IF_{As}) and compares it with the received M_1'; if it matches, the user (U_i) has already been registered in the system before and the session will be terminated. Otherwise, the AS creates a new record to save all contents of the message (R_rq) in IF_{As}, and sends R_{rs} = ("Registration success") to the user.

C. Login and Authentication Phase

In this phase, the user's legitimacy is proved with the user's identity, device ID, specified PIN, and biometric information. The procedure is as follows:

Step 1. The user (U_i) passes his identity and PIN through the mobile application, at the same time the device ID is automatically extracted to generate the following value $M_1 = h(ID_{di}||ID_{ui}||PIN)$, which is then encrypted after being concatenated with the current timestamp (T_{u1}) and the nonce value (N_{u1}). The result is sent as a login request message to AS:$L_{rq1} = E(Sk, E(Pr_{ui}, (M_1, T_{u1}, Nu_1)))$.

Step 2. The AS decrypts the message L_{rq1} and checks:

- The timestamp ($T_{AS} - T_{u1} \leq \Delta T$) and the value (N_{u1}), if the equivalent condition is not met, the AS terminates the authentication phase. Otherwise, AS checks the value of M_1' in the IF_{As}, if it does not exist, it means that the user has not registered in the system before and the login request will be rejected. Otherwise, the encrypted value of the locked fuzzy vault (V) with the current time (T_{As1}), new nonce value (N_{As1}) and old nonce value will be sent to the user: $L_{rs1} = E(Sk, E(Pr_{AS}, (V, T_{As1}, N_{As1}, N_{u1})))$.

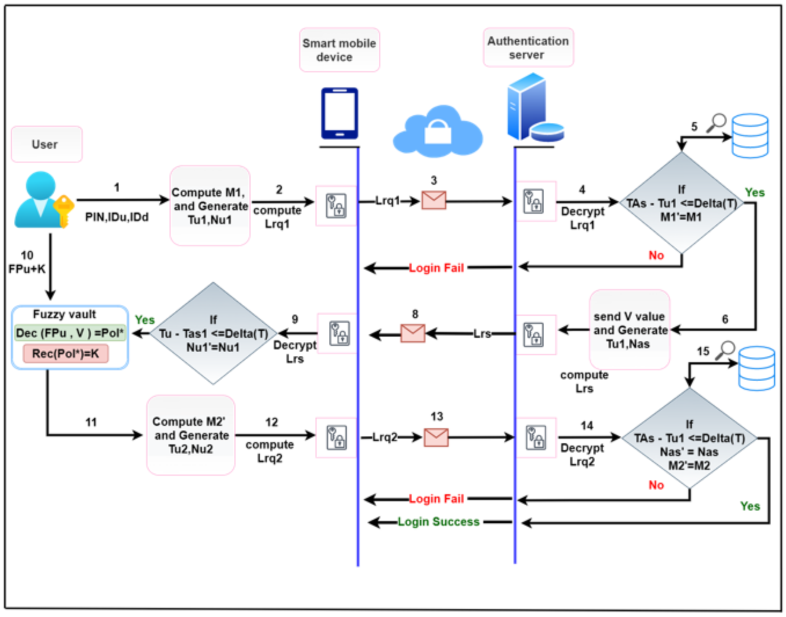


Figure 3: User login phase

Step 3. When the mobile app receives L_{rs1}, it will decrypt the message and

- Check the validity of the timestamp and whether the value of N'_{u1} received equals the value of N_{u1} sent and the value of N_{As1} received.
- Using the current fingerprint and vault value to calculate:
  $Dec(FP'_u, V)$, and $Rec(Pol*) = K$, then generate the value $M_2 = h(K||ID_{ui})$, and send:
  $L_{rq2} = E(Sk, E(Pr_{ui}, (M_2, T_{u2}, Nu_2, N_{As1})))$ to AS.

Step 4. Upon receiving the L_{rq2} message by the AS, the following steps will be taken:

- AS checks the timestamp ($T_{AS} - T_{u2} \leq \Delta T$) and checks whether the reserved value N'_{As2} is equal to the sent value N_{As2} or not.
- Then the AS compares the received M'_2 with the existing M_2, if it matches, the user can access the cloud computing service he wants. Otherwise, access will be denied.

D. Account Retrieve and Update Phase

If the user loses or changes the SMD_i from which he/she is registered, he/she must perform an account retrieve since ID_{di} will change under the new SMD_i :

Step 1. The user (U_i) sends an account to retrieve a request to the AS represented by the message (UP_{rq1}) which is generated by taking first the hash value of the answers to the requested security questions from the user concatenated with the PIN and $ID_{ui}, M_4 = h(Qs_1||Qs_2||Qs_3||Qs_4||PIN||ID_{ui})$. Then it will be sent with the current timestamp and new nonce value (N_{u1}) to the AS after being encrypted with the shared key and user's private key: $UP_{rq1} =$

$E(Sk, E(Pr_{ui}, (M_4, N_{u1}, T_{u1})))$.

Step 2. When AS received the message UP_{rq1}, it will decrypt the message and check:

- The timestamp and the nonce value N_{u1}; if the equivalent condition is not met, AS terminates the phase. Otherwise, it checks the value of M'_4 in IF_{As}, if not exist, it means that the user has not registered in the system and the update request will be rejected. Otherwise, the value of the locked fuzzy vault array will be sent to the user with the current time T_{As1} along with the new and old nonce values: $UP_{rs1} = E(Sk, E(Pr_{AS}, (V, T_2, N_{u1}, N_{As1}, T_{As1})))$.

Step 3. When the mobile app receives UP_{rs1}, it will decrypt the message and then:

- Check the validity of the timestamp and whether the reserved value N'_{u1} is equal to the sent value N_{u1} or not, and check the value of N_{As1}. Then calculate: $Dec(FP'_u, V), Rec(Pol*) = K, M_2 = h(K||ID_{ui})$, the hash of the new ID_{di} concatenated with ID_{ui}, and PIN as $M_1 = h(ID_{di}||ID_{ui}||PIN)$. Which are then sent to the AS with the new generated nonce and timestamp as: $UP_{rq2} = E(Sk, E(Pr_{ui}, (M_2, T_{u2}, N_{As1}, N_{u2}, M'_1)))$.

Step 4. When the AS receives the message UP_{rq2}, the following steps will be performed:

- AS checks the validity of the timestamp and the nonce. Then AS compares the received M'_2 with the existing M_2 if a match is found, the value M_1 will be updated with the new M_1' containing the new value for the new mobile device.

If the user wishes to change the PIN value, then nearly the same procedures are repeated by considering the PIN value instead of ID_{di}.

The update phase occurs to update the vault information but it requires the user to have legal access to his / her account. The system will request to update the vault information after a specified time since the last update. The update steps are as follows:

$UP_{rq1} = E(Sk, E(Pr_{ui}, (M'_2, M'_3, T_{u1})))$. Thus, the next time of login, AS will rely on these new values.

VII. Informal Security Analysis

In this section, the security of the proposed scheme is verified Against known malicious attacks like trafficking and snooping. Note that the proposed scheme contains robust features like shared authentication, user confidentiality, and a protected session key. A comparative analysis of relevant authentication schemes is also provided. As a result, the scheme was validated based on the AVISPA analysis tool, which indicates that the work is safe and secure.

A. *Resist to Stolen /Lost Smart Card or Mobile Device Attack*

In the proposed scheme, the mobile device does not store any sensitive information. Even if the attacker obtains the mobile device ID, still cannot pass the cloud service provider's authentication due to a lack of other authentication information such as biometric information, user ID, and PIN. Besides, the scheme does not support the use of smart carts thus avoiding the lost /steel smart card attacks.

B. *Resistance Against A Replay Attack*

Assume that the message $< Rqi >$ is intercepted and replies to this message to the AS. However, AS will check the validity of the timestamp $(T_{AS} - T_i \leq \Delta T)$. If the reserved values do not meet the conditions, the session will terminate immediately. Similarly, the mobile terminal checks the validity of the timestamp of the replied message $< Rrsi >$ by checking $(T_u - T_i \leq \Delta T)$, if the conditions are not met, the session will also terminate. Thus, the scheme protects against replay attacks.

C. Man-in-The-Middle Attack

The schemes use the hash function, and random fresh nonce in the $< Rqi, Rrsi >$ messages. However, the adversary cannot modify the message to pass the authentication process because the session is encrypted with a secret session key (Sk) generated by the TLS protocol using secret parameters shared by SMD and AS. What makes (Sk) secure is that TLS protocol uses a certified Public-Key. A public-key certificate allows a party to transmit its public key securely while ensuring the authenticity of the recipient's public key. Thus, an adversary cannot change the public key of one party, replace it with his own, and gain access to the entire session. As a result, the suggested scheme is impervious to the in-middle attacker.

D. Privilege Insider Attack

Malicious operators who entrust legal access (i.e, insider) to a system carry out internal attacks. The messages $< M1, M2, M4 >$ are preserved as a hashed value using SHA3, and because of the difficulty of the discrete logarithm problem, inverting the one-way hash function $H(\hat{A}\cdot)$ to retrieve sensitive user information is impossible. While $< M3 >$ that containsthe locked fuzzy vault information is preserved without hashing but remains secure, it requires the expected attempts of the adversary to interpolate the secret with a precision of approximately 1.86×10^9 as calculated in [25]. This corresponds to a 30-bit security level of $1.86 \times 10^9 \approx 2^{30}$.

E. User Impersonation Attack

An impersonation attacker must have the three authentication factors to gain access to the system. Since one of these factors is the biometric factor that only the legal user has, it is difficult to personate that user. Even if the attacker obtains the biometric information, there are still two other factors (ID_{di}, PIN) the attacker does not have.

F. Mutual Authentication

AS validates the legitimacy of U_i in this scheme, by validating the legitimacy of messages $< M1 >$ and $< M3 >$, respectively. For AS authentication, U_i realizes this by verifying the message $< M2 >$. By proving that, the AS knows its hidden message $< M2 >$ by conforming to the message $< M3 >$. As a result, the scheme enables mutual authentication.

G. User Anonymity and Unlinkability

Because ID_{ui} has never been sent on the public channel, the attacker cannot discover the user's identity. Assume the message: $M_1 = h(ID_{di}||ID_{ui}$ is intercepted by the attacker. The attacker is unable to identify the identity of the user. ID_{ui} from M_1 because of the one-way function $h(\cdot)$ protection. As a result, the scheme can protect the user's identity. Moreover, the message $< R_{rqi}, L_{rqi}, UP_{rsi} >$ is transmitted on the private channel, and the Sk used in each new session changes and uses a random number that is not associated with the user, as a result, the adversary cannot detect whether messages are transmitted in numerous sessions by the same user, so this scheme also fulfilled.

H. Perfect Forward and Reverse Confidentiality

In the proposed scheme, a session key Sk is generated using public-private cryptography that uses random numbers, and the Sk is changed by each authentication process. As a result, exposing any session key does not supply the adversary with any information that can be used to calculate the previous and subsequent session keys.

VIII. Formal Security Analysis

The scheme is simulated with AVISPA by running the most important tool, Security-Protocol-Animator (SPAN) 1.6V, on a computer system with Windows 10 Enterprise, OS (64 bit), Intel (R) Core (TM) i5-9400F CPU @ 2.90 GHz, 2.90 GHz processor, and 8.00 GB installed memory (RAM). All are powered by Ubuntu 14.1 on a virtual machine.

A. An Overview of The AVISPA Application Tool

AVISPA tool is a unique approach for analyzing and studying the security of schemes and protocols. It is a powerful software tool for automatic validation (Depending on the push-button method) (See Fig. 4). The scheme is executed in HLPSL (High-Level Protocol Specification Language). Once the scheme is written, the HLPSL2IF translator converts the code to the Intermediate Format (IF) [26]. The steps for the back-ends are as follows:

1) On-the-fly Model-Checker (OFMC).
2) Constraint-logic-based Attack Searcher (CL-AtSe).
3) SAT-based Model-Checker (SATMC).
4) 4. Tree Automated based on Automatic Approximation for the analysis of Security Protocols (TA4SP).

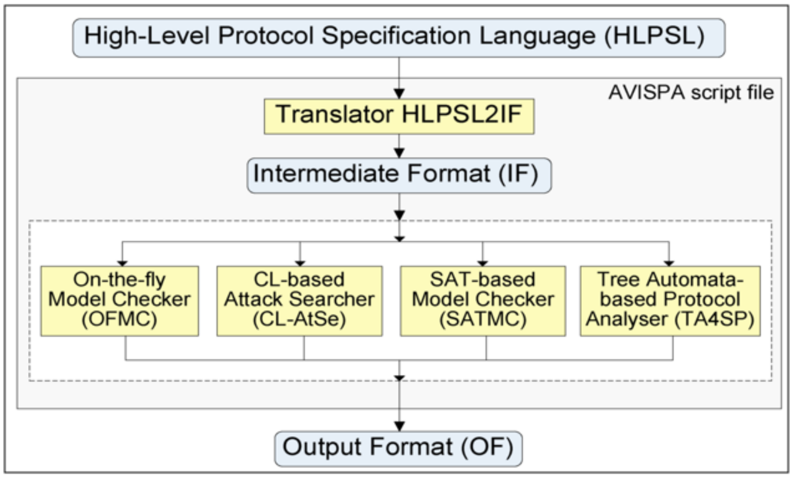

High-Level Protocol Specification Language (HLPSL)
AVISPA script file
Translator HLPSL2IF
Intermediate Format (IF)
On-the-fly Model Checker (OFMC)
CL-based Attack Searcher (CL-AtSe)
SAT-based Model Checker (SATMC)
Tree Automata-based Protocol Analyser (TA4SP)
Output Format (OF)

Figure 4: AVISPA tool architecture

Based on the HLPSL standard, the proposed scheme involves of:

1) Basic role: describes the objects' activities provided in the protocol (e.g., AS and User of a system).

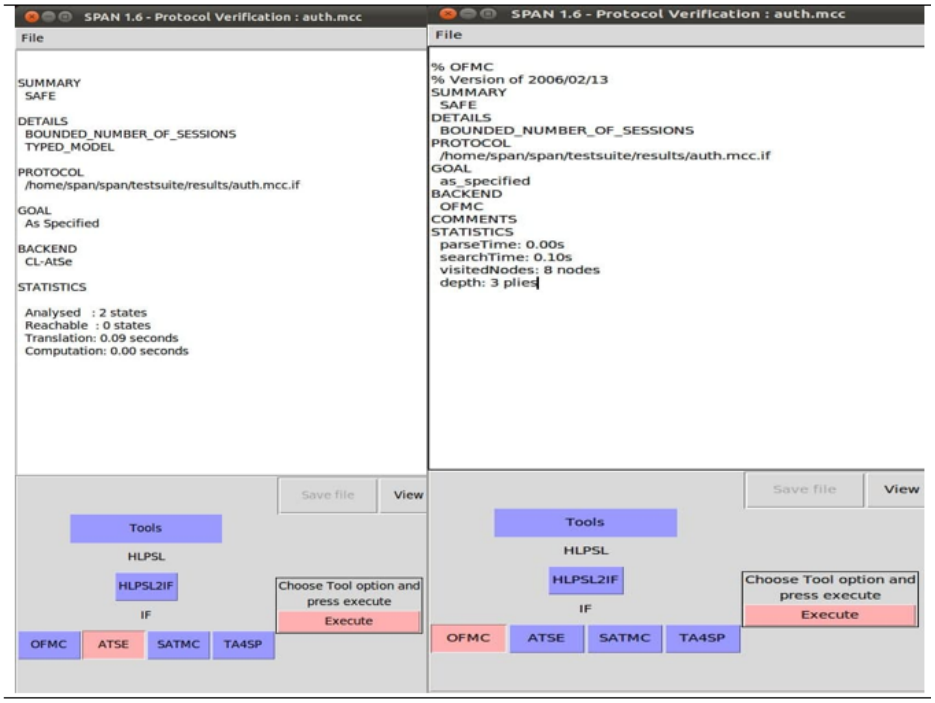
2) Transitions: described in terms of stages where the initial basic role begins with a declaration stating the first steps. This status changes when a message is received only.
3) Composed roles: have one or more fundamental functions to appliance together and designate the sessions engaged in the scheme.
4) Environment: All sessions are held, and the attacker may play various roles as an authorized user.
5) Security goal: outlines the scheme's security aim.

B. The Output's Description of AVISPA Analysis Tool

The result made by the AVISPA analysis tool includes the parts listed below (see Table I):

1) Summary: Defines the dependability of a protocol's security in terms of secure, insecure, or inconclusive states.
2) Details: The outputs describe the setting and environment within which the protocol state needed to be secure, insecure, or inconclusive.
3) Protocol: a name of the protocol, written here which is needed for documentation.
4) Goal: The protocol's recognized security purpose is defined in this section.
5) Backend: One of the four back-ends is discussed in this section.

TABLE I
Security verification results were achieved via the AVISPA tool


SPAN 1.6 - Protocol Verification : auth.mcc
File

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/auth.mcc.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.09 seconds
  Computation: 0.00 seconds
```

Save file
View
Tools
HLPSL
HLPSL2IF
IF
OFMC
ATSE
SATMC
TA4SP
Choose Tool option and press execute
Execute
SPAN 1.6 - Protocol Verification : auth.mcc
File

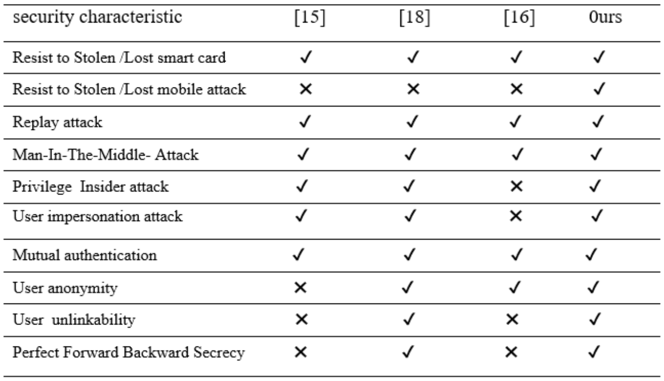
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/auth.mcc.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 8 nodes
  depth: 3 plies
```

IX. Comparison with Other Related Works

A. Security Features

The system's security characteristic is comparable to several prior study systems, as seen in the Table II.

TABLE II
Comparison of security characteristics



security characteristic	[15]	[18]	[16]	0urs
Resist to Stolen /Lost smart card	✓	✓	✓	✓
Resist to Stolen /Lost mobile attack	✕	✕	✕	✓
Replay attack	✓	✓	✓	✓
Man-In-The-Middle- Attack	✓	✓	✓	✓
Privilege Insider attack	✓	✓	✕	✓
User impersonation attack	✓	✓	✕	✓
Mutual authentication	✓	✓	✓	✓
User anonymity	✕	✓	✓	✓
User unlinkability	✕	✓	✕	✓
Perfect Forward Backward Secrecy	✕	✓	✕	✓

X. Conclusion

To maintain cloud computing security, authentication is employed as the first line of defense to check the validity of communication applicants. Hence, before mobile users and cloud servers send information, mutual authentication is necessary. An anonymized bio-based multi-factor mobile cloud Services authentication scheme is suggested in this paper. The hash function and the fuzzy vault processes are used only in the system, making it appropriate for mobile devices with low resources. The proposed scheme is resilient and safe against several well-known threats of attacks, according to informal security analysis and formal verification using AVISPA. Comparisons of the proposed scheme to many relevant current schemes demonstrate that the method not only avoids security vulnerabilities but also functional problems in previous schemes, such as lack of anonymity, vulnerability to impersonation attacks, and lost mobile device/smart cart assaults. Finally, Because of its high efficiency and safety compared to the existing systems, the scheme is an attractive alternative to mobile cloud computing.

References

[1] Xia, Z, et al, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data" , IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 2, pp. 340-352, 2016.

[2] Fu, Z, et al, "Achieving Efficient Cloud Search Services: Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing" , IEICE Transactions on Communications,Vol. E98.B, No. 1: pp. 190-200, 2015.

[3] Armbrust, M, et al, "A View of Cloud Computing" , Commun. ACM, Vol. 53, No. 4, pp. 50-58, 2010.

[4] Lin, A. and N. C. Chen, "Cloud Computing As An Innovation: Percepetion, Attitude, and Adoption" , International Journal of Information Management, Vol. 32, No. 6, pp. 533-540, 2012.

[5] Fu, Z, et al, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement" , IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 9, pp. 2546-2559, 2016.

[6] Ren, Y, et al, "Mutual Verifiable Provable Data Auditing in Public Cloud Storage" , Journal of Internet Technology, Vol. 16, pp. 317-323, 2015.

[7] Bruun, A, K. Jensen, and D. Kristensen, "Usability of Single- and Multi-Factor Authentication Methods on Tabletops: A Comparative Study" , in Human-Centered Software Engineering, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[8] Ometov, A, et al, "Multi-Factor Authentication: A Survey" , Cryptogr. , Vol. 2: pp. 1, 2018.

[9] Sun, J, et al, "Sightless Two-Factor Authentication on Multi-Touch Mobile Devices" , in 2014 IEEE Conference on Communications and Network Security, 2014.

[10] Han, Z, L. Yang, and Q. Liu, "A Novel Multifactor Two-Server Authentication Scheme under The Mobile Cloud Computing" , pp. 341-346, 2017.

[11] Fati, S, "A Coherent Authentication Framework for Mobile Computing Based on Homomorphic Signature and Implicit Authentication" , 2017.

[12] Dey, S, Q. Ye, and S. Sampalli, "AMLT: A Mutual Authentication Scheme for Mobile Cloud Computing" , in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) , 2018.

[13] Chean, L.T, V. Ponnusamy, and S. M. Fati, "Authentication Scheme Using Unique Identification Method with Homomorphic Encryption in Mobile Cloud Computing" , in 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2018.

[14] Zeroual, A, et al, "Deep Authentication Model in Mobile Cloud Computing" , in 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS) , 2018.

[15] Sun, J, et al, "A Lightweight Multi-Factor Mobile User Authentication Scheme" , in IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) , 2018.

[16] Mo, J, et al, "An Efficient and Provably Secure Anonymous User Authentication and Key Agreement for Mobile Cloud Computing" , Wireless Communications and Mobile Computing, Vol. 2019: p. 4520685, 2019.

[17] Abuarqoub, A, "A Lightweight Two-Factor Authentication Scheme for Mobile Cloud Computing" , In Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, Association for Computing Machinery: Paris, France, p. Article 29, 2019.

[18] Chen, H, et al, "An Enhanced Lightweight Biometric-Based Three-Factor Anonymous Authentication Protocol for Mobile Cloud Computing" , in 2019 IEEE 21st International Conference on High Performance Computing and Communications, IEEE 17th International Conference on Smart City, IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) , 2019.

[19] Ahmed, A. A, et al, "Dynamic Reciprocal Authentication Protocol for Mobile Cloud Computing" , IEEE Systems Journal, pp. 1-11, 2020.

[20] Dolev, D. and A. Yao, "On The Security of Public Key Protocols" , IEEE Transactions on Information Theory, Vol. 29, No. 2, pp. 198-208, 1983.

[21] Satapathy, A. and J. Livingston, "A Comprehensive Survey on SSL/ TLS and their Vulnerabilities" , International Journal of Computer Applications, Vol. 153, pp. 31-38, 2016.

[22] Yu, J, et al, "An Efficient Generic Framework for Three-Factor Authentication With Provably Secure Instantiation" , IEEE Transactions on Information Forensics and Security, Vol. 9, No. 12, pp. 2302-2313, 2014.

[23] Juels, A. and M. Sudan, "A Fuzzy Vault Scheme: Designs, Codes and Cryptography" , Vol. 38, No. 2, pp. 237-257, 2006.

[24] Rao, M, T. Newe, and I. Grout, "Secure Hash Algorithm-3 (SHA-3) Implementation on Xilinx FPGAs, Suitable for IoT Applications" , International Journal on Smart Sensing and Intelligent Systems, Vol. 7: pp. 1-6, 2020.

[25] Geng, S, G. Giannopoulou, and M. Kabir-Querrec, "Privacy Protection in Distributed Fingerprint-Based Authentication" , pp. 125-129, 2019.

[26] Hosseini, Z, "Fingerprint Vulnerability: A survey" , pp. 70-77, 2018.