

# Modelling and Simulation of a Biometric Identity-Based Cryptography

Dania Aljeaid

School of Science and Technology  
Nottingham Trent University  
Nottingham, United Kingdom

Xiaoqi Ma

School of Science and Technology  
Nottingham Trent University  
Nottingham, United Kingdom

Caroline Langensiepen

School of Science and Technology  
Nottingham Trent University  
Nottingham, United Kingdom

**Abstract**—Government information is a vital asset that must be kept in a trusted environment and efficiently managed by authorised parties. Even though e-Government provides a number of advantages, it also introduces a range of new security risks. Sharing confidential and top-secret information in a secure manner among government sectors tends to be the main element that government agencies look for. Thus, developing an effective methodology is essential and it is a key factor for e-Government success. The proposed e-Government scheme in this paper is a combination of identity-based encryption and biometric technology. This new scheme can effectively improve the security in authentication systems, which provides a reliable identity with a high degree of assurance. This paper also demonstrates the feasibility of using finite-state machines as a formal method to analyse the proposed protocols. Finally we showed how Petri Nets could be used to simulate the communication patterns between the server and client as well as to validate the protocol functionality.

**Keywords**—e-Government; identity-based cryptosystem; biometrics; mutual authentication; finite-state machine; Petri net.

## I. INTRODUCTION

e-Government mainly acts as a communication bridge, whether from government to citizen, government to government, or government to business, in efficient and reliable ways through effective use of information technology. The main challenge in e-government is to develop a framework which promotes exchanging data securely among government agencies. While e-government provides a number of advantages, it also introduces a range of new security risks. Sharing confidential and top-secret information in a secure manner among government sectors tend to be the main element that government agencies look for.

When e-Government systems were being designed, Public Key Infrastructure (PKI) seemed to be the best solution for the scheme. PKI is presently deployed in most e-Government implementations, as it is perceived as a mature technology, which is widely supported and can be easily integrated with different systems. Examples of e-government initiatives that apply PKI on a large scale are the US eGov initiative (www.usa.gov) supported by Federal PKI [1] and the Saudi Arabian e-Government Program (yesser.gov.sa) [2].

One of the main issues concerning the security perspective in e-Government is to grant access to authorised users as well as the need to verify that the user is really who they claim to

be. The most common solution to this problem is to deploy a PKI [3] and digital signatures in large-scale e-Government systems. Even though PKI supports strong authentication and digital signature, it has a few disadvantages. For example, users must be pre-enrolled, certificate directories can leak some critical information, key recovery is difficult and costly and boundary services (anti-spam, anti-virus, archiving) integration is very difficult [4].

Thus, to take full advantage of the capabilities of e-Government, end users need robust security solutions to achieve assurance when dealing with e-Government systems. A variant of public key cryptography that derives public keys directly from unique identity information (such as an e-mail address) known by the user is called Identity-Based Cryptography (IBC). This approach has recently received considerable attention from researchers [5, 6, 7, 8, 9], as the development of ID-Based Cryptography offers great flexibility and obviates the requirement for user certificates, since the identity of the user can be transformed into encryption keys and used for authentication.

To develop a new secure cryptosystem for e-Government, several schemes were investigated to determine which protocol would be suitable for the research. We propose a biometric-ID-based scheme using Elliptic Curve Cryptosystem (ECC), which is an improved combination scheme derived from two schemes [10, 11]. The proposed scheme is secure under the Computational Diffie-Hellman Assumption (CDHA) and tackles the security drawbacks of He *et al.*'s scheme and Li and Hwang's scheme. To overcome these, we applied a symmetric key cryptosystem to prevent attackers from altering or gaining any important information in the login and authentication messages.

The structure of this paper is organised as follows. In Section 2, we review related works on ID-Based Cryptography and Biometric authentication and briefly describe both He *et al.*'s and Li and Hwang's schemes. In Section 3, we design the new Biometric-ID-based Authentication Scheme. In Section 4, we model the new protocol with finite-state machines. In Section 5, we model the new protocol with Petri Nets to simulate the communication. We then provide a brief discussion on security analysis and comparisons with related schemes in Section 6. Finally, the conclusion is given in Section 7.

## II. REVIEW OF RELATED WORK

Without a secure and trusted infrastructure, organisations such as governments would leave data electronically unsecured and vulnerable to attacks. Therefore, governments are constantly looking for ways to deliver secure and reliable services. ID-Based Cryptography introduces a lightweight key management and offers encryption for data confidentiality and robust authentication, which are prerequisites for securing high-value transactions.

The idea of ID-based cryptography was originally proposed by Shamir in 1984 [12], but practical ID-based encryption schemes were not developed until recently. In 2001, Boneh & Franklin [5] developed a fully functional ID-based encryption scheme which can be constructed efficiently by using Weil pairing on elliptic curves.

In ID-based cryptosystems, there is a trusted third party called a **Private Key Generator (PKG)** who is responsible for generating the secret keys for all users. As a result, a PKG holds the users' private keys. If a PKG is malicious, it can impersonate any user and therefore decrypt any cipher text or forge signature on any message. This can lead to a problem known as key escrow [13, 14].

There is no question that the **Identity-Based Encryption (IBE)** scheme brings many advantages such as eliminating the need to distribute public keys. The enforcement of the private key generation by the Private Key Generator raises concerns of key escrow and/or privacy surrounding the management of private keys. To address this particular problem of key escrow, an implementation of biometric identification systems can be used as a private key. Biometric technology and verification systems offer a number of benefits to government sectors and users [15, 16].

He *et al.* [10] proposed an ID-based remote mutual authentication with key agreement scheme on ECC. This protocol attempts to cope with many of the well-known security and efficiency problems. However, the scheme has a potential flaw that may lead to man-in-the-middle attack and impersonation attack [17, 18]. It can be seen that, if an attacker  $E$  eavesdrops and listens to the communication between  $S_i$  and  $C_i$ , then  $E$  can intercept a valid login request  $M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$  or  $h(ID_i || Xs)$  and masquerade as a legal user.

Biometric technologies are becoming fundamental elements in ensuring highly secure identification and personal verification solutions [15]. Biometric keys can be extracted from keystroke patterns, the human voice [19], fingerprints [20, 21], handwritten signatures [22], and facial characteristics [23].

Li and Hwang [11] proposed an efficient biometrics-based remote user authentication scheme using smart cards. The security of their scheme is based on one-way hash functions, biometric verification, smart card and it uses a nonce. The scheme is very efficient in computation cost, which has been proved to be relatively low compared with other related schemes [24, 25, 26, 27]. The scheme is composed of four

phases: the registration phase, the login phase, the authentication phase and password change phase.

One of the key characteristics of the cryptographic hash function is that the outputs are very sensitive to small perturbations in their inputs. Hash functions cannot be applied directly when the input data are noisy such as biometrics [28]. Therefore, a secure one-way hash function cannot be used for biometric verification. In the login phase of Li-Hwang's scheme, the user computes  $h(B_i)$  based on a personal biometric template  $B_i$ . Then the biometric authentication process relies on comparing the hash value  $h(B_i)$  with  $f_i$ . However, the scheme does not seem to be able to handle natural variation in the biometrics. For example, when the user logs in, his fresh biometric sample has to match exactly the template recorded during the registration phase, which never happens in practice. Thus, the protocol is fundamentally flawed and does not fulfil the basic objectives of a biometric authentication protocol. As a result, this may prevent a legal user from passing biometric verification at the login phase. So, Li-Hwang's scheme is vulnerable to denial-of-service attack. The scheme is also prone to man-in-the-middle attack and impersonation attack. The attacker can cheat the server by impersonating the user or can impersonate the server to cheat the user without knowing any secret information [29, 30, 31]

Combining ID based cryptography with biometric techniques can effectively improve the security in authentication systems, which provides a reliable identity with a high degree of assurance. The biometric technology is regarded as a powerful solution due to its unique link to an individual identity, which almost impossible to fake. Thus, a biometric identity is an inherent trait, which will always remain with the person all the time. In another words, using biometric techniques in IBE will mean that the person will always have their private key available.

## III. PROPOSED SCHEME

This research will focus on secure e-Government systems and improve their authentication and communication. To guarantee the security of these distributed systems, biometrics verification and ID-based cryptography are used. The proposed protocol is based on the following assumptions:

- We assume that shared secrets in registration phase will never be disclosed.
- We assume that cryptographic algorithms are secure. For example, it is impossible to decrypt a ciphertext without prior knowledge of the secret key.
- We assume that both client and server are able to generate a random number securely.

The security of the proposed scheme is based on the intractability of the following two mathematical problems on elliptic curves [5, 10]:

- (i) **Computational Diffie–Hellman Assumption (CDHA):** Given  $P, xP, yP \in G$ , it is hard to compute  $xyP \in G$ .
- (ii) **Collision Attack Assumption 1 (k-CAA1):** For an integer  $k$ , and  $x \in Z_n^*, P \in G$ , given

$(P, xP, h_0, (h_1, (h_1+x)^{-1}P), \dots, (h_k, (h_k+x)^{-1}P))$ , where  $h_i \in Z_n^*$ , and distinct for  $0 \leq i \leq k$ , it is hard to compute  $(h_0+x)^{-1}P$ .

The proposed scheme consists of four phases: system initialising phase, registration phase, login phase, and authentication phase. The notations used throughout this paper are summarised in Table 1.

TABLE.I. NOTATIONS USED IN THIS PAPER

Symbol	Definition
$C_i$	User/Client /Computer
$S_i$	Server
$R_i$	Registration Centre
$ID_{S_i}$	Identity of Server
$ID_{C_i}$	Identity of user $C$
$PW_{C_i}$	User's password
$Bio_{C_i}$	Biometric template of $C$
Pub_K	Public Key
Pr_K	Private Key
$\parallel$	Message concatenation operation
$p, n$	Two large prime numbers
$F_p$	A finite field
$E$	An elliptic curve over a finite field $F$
$G$	The group of elliptic curve points on $E$
$P$	A point on elliptic curve $E$ with order $n$
$xP$	Denotes point multiplication on elliptic curve
$y$	A piece of secret information maintained by the server
$(x, \text{Pub\_K}_s)$	The server $S$ 's Private/Public key pair, where $\text{Pub\_K}_s = xP$
$r_{C_i}, r_{S_i}$	A random number chosen by the $C_i$ and $S_i$ respectively
$H(.)$	A secure one-way hash function
$\text{MAC}_k(m)$	The secure message authentication code of $m$ under the key $k$
$\oplus$	XOR operation

#### A. System initializing phase

In this phase, we follow the steps in He *et al.*'s scheme where the server  $S_i$  generates parameters of the system.

**Step 1:**  $S_i$  chooses an elliptic curve equation  $E_P(a, b)$ .

**Step 2:**  $S_i$  selects a base point  $P$  with the order  $n$  over  $E_P(a, b)$

**Step 3:**  $S_i$  selects its master key  $x$  and secret information  $y$  and computes public key  $\text{Pub\_K}_s = xP$

**Step 4:** The server chooses four secure one-way hash functions  $H_1(.)$ ,  $H_2(.)$ ,  $H_3(.)$ ,  $H_4(.)$ , where  $H(.)$  is a known hash function that takes a string and assigns it to a point on the elliptic curve, i.e.  $H(A) = QA$  on  $E$ , where  $C$  is usually based on the identity

- $H_1(.)$ : a secure one-way hash function, where  $H_1: \{0, 1\}^* \rightarrow Z_n^*$
- $H_2(.)$ : a secure one-way hash function, where  $H_2: \{0, 1\}^* \rightarrow Z_p^*$
- $H_3(.)$ : a secure one-way hash function, where  $H_3: \{0, 1\}^* \rightarrow Z_p^*$
- $H_4(.)$ : a secure one-way hash function, where  $H_4: \{0, 1\}^* \rightarrow Z_p^*$

The server also chooses a message authentication code  $\text{MAC}_k(m)$ . Then, it keeps  $x$  private and publishes  $\{F_p, E, n, P, \text{Pub\_K}_s, H_1, H_2, H_3, H_4, \text{MAC}_k(m)\}$ .

#### B. Registration Phase

A user  $C_i$  with identifier  $ID_{C_i}$  should be registered first before using the services provided by  $R_i$ . Users may use their employee number as an identity when contacting  $R_i$  for authorisation. In this phase,  $C_i$  needs to perform the following steps.

**Step 1:** User  $C_i$  inputs their  $ID_{C_i}$ , personal biometrics  $Bio_{C_i}$ , on a specific biometric device, and provides the password  $PW_{C_i}$  to  $R_i$  via a secure channel (or to the registration centre in person).

**Step 2:**  $R_i$  reads current timestamp  $T_{S_i}$ , and computes the following:

$$f_i = H_4(Bio_{C_i})$$

$$z_i = H_4(PW_{C_i} \parallel f_i)$$

$$e_i = H_4(ID_{C_i} \parallel y) \oplus z_i$$

**Step 3:**  $R_i$  computes  $C_i$ 's private key using the system private key  $x$  and  $C_i$ 's public key.

$$\text{Pr\_K}_{C_i} = (x + H_4(ID_{C_i}))^{-1} P \in G$$

$$\text{Pub\_K}_{C_i} = H_4((ID_{C_i} + x)P) = H_4((ID_{C_i})P + \text{Pub\_K}_s)$$

**Step 4:**  $R_i$  stores  $\{ID_{C_i}, H_4(.)$ ,  $\text{Enc}\{ \}_a/\text{Dec}\{ \}_a, f_i, e_i, \tau, \text{Pr\_K}_{C_i}\}$  on a secure database and sends it to the user via a secure channel, where  $\text{Enc}\{ \}_a/\text{Dec}\{ \}_a$  is a symmetric encryption with secret key  $a$  and  $\tau$  is a predetermined threshold [28] for biometric verification.

#### C. Login Phase

The user  $C_i$  sends a login request to the server  $S_i$  and performs the following steps:

**Step 1:**  $C_i$  enters the  $ID_{C_i}$  and  $PW_{C_i}$ , and then  $S_i$  verifies the authenticity of client's identity and password.

**Step 2:**  $C_i$  submits the  $Bio_{C_i}$  on specific biometric device, and then verifies the following:

$$\begin{cases} \text{Accept if } d(Bio_{C_i}, Bio_{C_i}^*) < \tau \\ \text{Reject if } d(Bio_{C_i}, Bio_{C_i}^*) \geq \tau \end{cases}$$

**Step 3:** if the above does not hold, it means the biometric information does not match the template

stored in the system. Thus  $C_i$  does not pass the biometric verification process and the authentication scheme is terminated. Otherwise,  $C_i$  passes the biometric verification and computes the following:

$$\begin{aligned} f_i &= H_4(\text{Bio}_{C_i}) \\ z_i &= H_4(\text{PW}_{C_i} \| f_i) \\ M_1 &= e_i \oplus z_i = H_4(\text{ID}_{C_i} \| y) \\ W_1 &= r_{C_i} \cdot P \\ M_2 &= r_{C_i} \cdot \text{Pr\_K}_{C_i} \\ M_3 &= M_1 \oplus r_{C_i} \end{aligned}$$

Where  $r_{C_i} \in Z_n^*$  is a random number generated by the user. For this step, the random value  $r_{C_i}$  is introduced to mask the hash of the secret value  $H_4(\text{ID}_{C_i} \| y)$ .

**Step 4:**  $C_i$  computes  $k = H_2(\text{ID}_{C_i}, T_{C_i}, W_1, M_2)$ , where  $T_{C_i}$  is a timestamp denoting the current time.

**Step 5:** Finally,  $C_i$  encrypts the message  $\{\text{ID}_{C_i}, T_{C_i}, W_1, M_3, \text{MAC}_k(\text{ID}_{C_i}, T_{C_i}, W_1, M_3)\}_a$  and sends it to the server  $S_i$ .

#### D. Authentication Phase

After receiving the request login message,  $S_i$  and  $C_i$  will perform the following steps for mutual authentication.

**Step 1:**  $S_i$  decrypts the message  $\{\text{ID}_{C_i}, T_{C_i}, W_1, M_3, \text{MAC}_k(\text{ID}_{C_i}, T_{C_i}, W_1, M_3)\}_a$ , then checks the validity of  $\text{ID}_{C_i}$  and the freshness of  $T_{C_i}$ . The freshness of  $T_{C_i}$  is checked by performing  $T - T_{C_i} \leq \Delta T$ , where  $T$  is the time when  $S_i$  receives the above message and  $\Delta T$  is a valid time interval. The case where  $\text{ID}_{C_i}$  is not valid or  $T_{C_i}$  is not fresh, then  $S_i$  aborts the current session.

**Step 2:** If Step 1 holds,  $S_i$  computes the following:

$$\begin{aligned} M_2 &= (x + H_1(\text{ID}_{C_i})^{-1} W_1 \\ &= \text{Pr\_K}_{C_i} \cdot r_{C_i} \end{aligned}$$

$$k = H_2(\text{ID}_{C_i}, T_{C_i}, W_1, M_2)$$

$S_i$  checks the integrity of  $\text{MAC}_k(\text{ID}_{C_i}, T_{C_i}, W_1, M_3)$  with the key  $k$ .  $S_i$  will quit the current session if the check produces a negative result.

**Step 3:** If Step 2 holds,  $S_i$  chooses a random number  $r_{S_i} \in Z_n^*$  and computes the following:

$$M_4 = H_4(\text{ID}_{C_i} \| y)$$

$$W_2 = r_{S_i} \cdot P$$

$$K_{S_i} = r_{S_i} \cdot W_1$$

The session key  $sk = H_3(\text{ID}_{C_i}, T_{C_i}, T_{S_i}, W_1, W_2, K_{S_i})$ , where  $T_{S_i}$  is a timestamp denoting the current time

$$M_5 = M_3 \oplus M_4 = r_{C_i}$$

$$M_6 = M_4 \oplus r_{S_i}$$

$$M_7 = H_4(M_3 \| M_5)$$

Where  $M_5$  is the random value  $r_{C_i}$  of the user  $C_i$  and only  $S_i$  can unmask the value because it can compute  $H_4(\text{ID}_{C_i} \| y)$

**Step 4:** Then,  $S_i$  encrypts the message  $\{\text{ID}_{C_i}, T_{S_i}, W_2, M_6, M_7, \text{MAC}_k(\text{ID}_{C_i}, T_{S_i}, W_2, M_6, M_7)\}_a$  and sends it to  $C_i$

**Step 5:** Upon receiving the  $S_i$ 's message,  $C_i$  first decrypts  $\{\text{ID}_{C_i}, T_{S_i}, W_2, M_6, M_7, \text{MAC}_k(\text{ID}_{C_i}, T_{S_i}, W_2, M_6, M_7)\}_a$ , and checks the freshness of  $T_{S_i}$  is by performing  $T - T_{S_i} \leq \Delta T$ , where  $T$  is the time when  $C_i$  receives the above message and  $\Delta T$  is the expected time interval for the transmission delay.

**Step 6:**  $C_i$  verifies whether  $M_7 \stackrel{?}{=} H_4(M_3 \| r_{C_i})$  and checks the integrity of  $\text{MAC}_k(\text{ID}_{C_i}, T_{S_i}, W_2, M_6, M_7)$  with the key  $k$ .  $C_i$  will quit the current session if the check produces a negative result.

**Step 7:** If it holds,  $C_i$  believes that  $S_i$  is authenticated and then computes the following:

$$K_{C_i} = r_{C_i} \cdot W_2$$

The session key  $sk = H_3(\text{ID}_{C_i}, T_{C_i}, T_{S_i}, W_1, W_2, K_{C_i})$

$$M_8 = M_6 \oplus M_1 = r_{S_i}$$

$$M_9 = H_4(M_6 \| M_8)$$

Where  $M_9$  is the random value  $r_{S_i}$  of the server  $S_i$  and only the client  $C_i$ , which know  $M_1 = H_4(\text{ID}_{C_i} \| y)$ , can send back the correct hashed value of  $M_9 = H_4(H_4(\text{ID}_{C_i} \| y) \oplus r_{S_i} \| r_{S_i})$

**Step 8:**  $C_i$  sends the encrypted message  $\{M_9, \text{MAC}_k(M_9)\}_a$  to  $S_i$

**Step 9:** After receiving  $C_i$ 's message,  $S_i$  decrypts  $\text{Enc}\{M_9\}_a$  and check the integrity of  $\text{MAC}_k(M_9)$ . Then,  $S_i$  verifies whether  $M_9 \stackrel{?}{=} H_4(M_6 \| r_{S_i})$

**Step 10:** If the above mentioned holds,  $S_i$  accept  $C_i$ 's login request or otherwise rejects it

#### IV. BEHAVIOUR MODELLING AND STATE MACHINE

Verification is a crucial step in designing security protocols. A Finite-State Machine (FSM) is a powerful tool to simulate software architecture and communication protocols. FSM can only model the control part of a system and consists of a finite number of states, finite number of events, and finite number of transitions. An FSM may be regarded as a five-tuple [32]:  $(Q, \Sigma, \Delta, \sigma, q_0)$ , where:

- $Q$ : finite set of symbols denoting states
- $\Sigma$ : set of symbols denoting the possible inputs
- $\Delta$ : set of symbols denoting the possible outputs
- $\sigma$ : transition function mapping to  $Q \times \Sigma$  to  $Q \times \Delta$
- $q_0 \in Q$ : initial state.

The FSM is used to model the communication channel of proposed protocol between the Client  $C_i$  and the Server  $S_i$ .

Since the exchange of packets follows a pattern defined by a finite set of rules, it will be described by creating three finite-state machines  $FSM_{server}$ ,  $FSM_{register}$  and  $FSM_{client}$ .

#### A. Server FSM

The FSM at the server side represents the various on-going communications with the client at any point of time. It is modelled using 10 states and 22 transitions as detailed below. Fig. 1 shows the transitions diagram for the  $FSM_{server}$ .

1) The  $FSM_{server}$  will loop itself as the server is waiting for clients. The machine advances to the next state once it is triggered by a login/enrol transition accordingly.

2) When the  $FSM_{server}$  is in the state S1, it checks the validity of the received ID. If ID proved to be incorrect,  $S_i$  will request  $C_i$  to enter the valid ID for three times and  $FSM_{server}$  will loop until  $C_i$  enters the valid ID or if the attempts exceed three times. In the latter case, the  $C_i$ 's account will be blocked and  $FSM_{server}$  changes state to S4 from state S1. Generally, three attempts are made through our protocol steps to allow common errors.

3) When the  $FSM_{server}$  is in the state S2, it is triggered by valid ID and it is now waiting for a valid PW. Once  $S_i$  receives PW, it verifies its validity. If PW proved to be wrong,  $S_i$  will request  $C_i$  to enter the valid PW for three times and  $FSM_{server}$  will loop until  $C_i$  enters the valid PW or if the attempts exceed three times. In the latter case, the  $C_i$ 's account will be blocked and  $FSM_{server}$  changes state to S4 from state S2.

4) When the  $FSM_{server}$  is in the state S3, it is triggered by valid PW and it is now waiting for a valid Bio. Once  $S_i$  receives Bio, it verifies its validity by comparing the imprinted Bio with the template stored. If Bio does not match the stored template,  $S_i$  will request  $C_i$  to enter the valid Bio up to three times and the  $FSM_{server}$  will loop until  $C_i$  enters the valid PW or if the attempts exceed three times. In the latter case, the  $C_i$ 's account will be blocked and the  $FSM_{server}$  changes state to S4 from state S3.

5) In state S5, the  $FSM_{server}$  waits until receiving the login request  $SYN = \{ID_{C_i}, T_{C_i}, W_1, M_3, MAC_k(ID_{C_i}, T_{C_i}, W_1, M_3)\}_a$

from the  $FSM_{client}$  to establish a connection by performing three-ways-handshake.

6) While in State S5, the  $FSM_{server}$  checks the validity of ID, freshness of T and the integrity of  $MAC_k$ . Then  $S_i$  generates a random number and timestamp in order to calculate the session key  $sk = H_3(ID_{C_i}, T_{C_i}, T_{S_i}, W_1, W_2, K_{S_i})$ . After that,  $S_i$  replies  $SYN/ACK = \{ID_{C_i}, T_{S_i}, W_2, M_6, M_7, MAC_k(ID_{C_i}, T_{S_i}, W_2, M_6, M_7)\}_a$  to the  $FSM_{client}$ .

7) In state S6,  $FSM_{server}$  waits until receiving ACK from the  $FSM_{client}$ . Once the  $FSM_{client}$  sends  $ACK = \{M_9\}_a$ ,  $FSM_{server}$  verifies  $M_9 \stackrel{?}{=} H_4(M_6 || r_{S_i})$ . At this instance,  $S_i$  authenticates  $C_i$  as a legitimate user.

8) At state S5 and state S6,  $FSM_{server}$  terminates the current session if any of the following situations occurs:

- The client ID is invalid
- The freshness of  $T - T_{C_i} \geq \Delta T$
- Negative result when checking the integrity of  $MAC_k(ID_{C_i}, T_{C_i}, W_1, M_3)$
- If  $M_9 \neq H_4(M_6 || r_{S_i})$

At any stage of  $FSM_{server}$ ,  $FSM_{server}$  aborts the current session and changes to state S9 if the timeout exceeds the defined TIME\_WAIT while waiting for packets. This feature helps to prevent an infinite wait when the  $FSM_{client}$  fails to response.

#### B. Client FSM

The FSM at the client side represents the various on-going transmissions with the server at any point of time. It is modelled using 9 states and 21 transitions as detailed below. Fig. 1 shows the transitions diagram for the  $FSM_{client}$ .

1) First, the  $FSM_{client}$  is in the initial state C0 that is when the request for register/login is initiated by itself. While in state C0, the  $FSM_{server}$  checks whether  $C_i$  is enrolled or not. The next state will

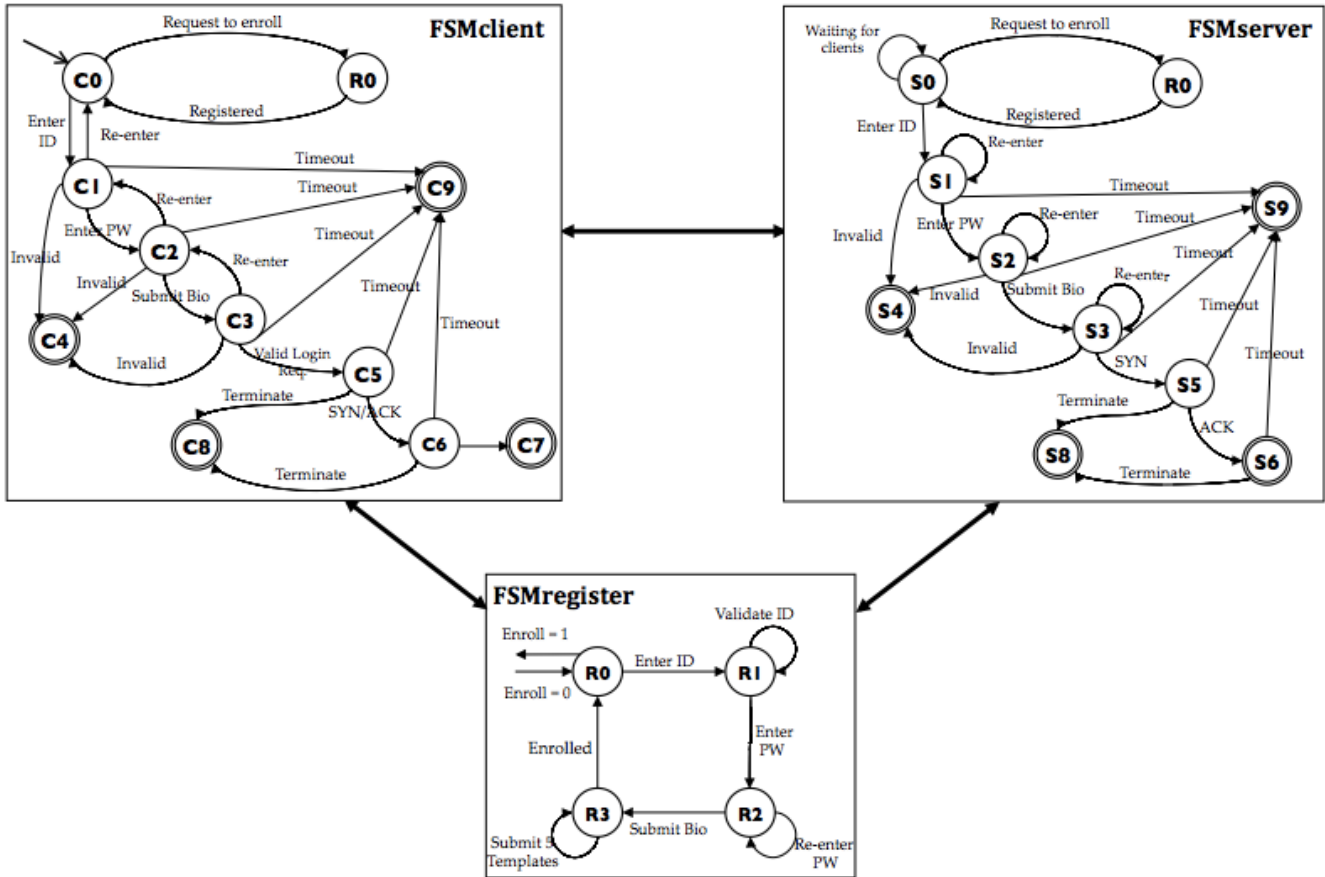


Fig.1. Proposed protocol FSM model

be decided according to the condition  $ClientReg == True$ .

2) In states  $C1, C2, C3$ , the  $FSM_{client}$  is waiting for validating ID, PW, and Bio. Once the client credentials are validated, the  $FSM_{client}$  triggers itself and changes to state  $C5$ .

3) In states  $C1, C2, C3$ , the client may require to re-enter ID, PW, Bio in case if they were incorrect. However, the client's account will be blocked if the number of attempts exceeds three trials, which change the above states to state  $C4$ .

- $ID\_attempt < 3, ID\_attempt = ID\_attempt + 1$
- $PW\_attempt < 3, PW\_attempt = PW\_attempt + 1$
- $Bio\_Attempt < 3, Bio\_attempt = Bio\_attempt + 1$

4) While in state  $C5$ , the  $FSM_{client}$  is waiting for the  $FSM_{server}$  response after sending the login request to establish the connection.

5) In state  $C6$ , the  $FSM_{client}$  is validating the  $FSM_{server}$  response by checking the integrity of  $MAC_k, \Delta T$  and  $M_7 \stackrel{?}{=} H_4(M_4 || r_{C_i})$ . If  $S_i$  is proved to be honest,  $C_i$  authenticates  $S_i$  at this stage.

6) While in state  $C6$ , the  $FSM_{client}$  computes the shared session key  $sk = H_3(ID_{C_i}, T_{C_i}, T_{S_i}, W_1, W_2, K_{C_i})$  and finalises the handshake procedure by sending  $ACK = \{M_9\}_a$  to  $S_i$ .

7) In state  $C7$ , the  $FSM_{client}$  is waiting to be authenticated by  $S_i$ .

8) In state  $C8$ , the client terminates the current session if one of the following occurs:

- Negative result when checking the integrity of  $MAC_k$
- The freshness of  $T' - T_{S_i} \geq \Delta T$
- $M_7 \stackrel{?}{=} H_4(M_4 || r_{C_i})$

At any stage of  $FSM_{client}$ ,  $FSM_{client}$  aborts the current session and changes to state  $C9$  if the timeout exceeds the defined TIME\_WAIT while waiting for packets. This feature helps to prevent an infinite wait when the  $FSM_{server}$  fails to response.

### C. Register FSM

The FSM at Registration side represents the various ongoing transmissions with the server and client at any point of time. It is modelled using 4 states and 7 transitions as detailed below. Fig. 1 shows the transitions diagram for the  $FSM_{register}$ .

1) First, the  $FSM_{register}$  is triggered if the client is not enrolled  $R0$ , that is when the request for register is initiated by

$FSM_{client}$ . While in state  $C0$ , the  $FSM_{server}$  checks whether  $C_i$  is enrolled.

2) When once  $C_i$  enters  $ID$ ,  $FSM_{register}$  changes to state  $R1$  and validates the format of  $ID$ .  $FSM_{register}$  triggers itself. Then  $FSM_{register}$  asks  $C_i$  to enter  $PW$  and changes to state  $R2$ .

3) In state  $R2$ , on receiving  $PW$  for the first time,  $FSM_{register}$  requires  $C_i$  to re-enter  $PW$  for confirmation. Then it triggers and changes to the state  $R3$ .

4) In state  $R3$ ,  $C_i$  is required to submit multiple scans of the biometric data to increase accuracy. Once the acquisition process is complete,  $FSM_{register}$  trigger itself and sends a message to  $R0$ , which indicates that the enrolment is successful.

## V. PROTOCOL MODEL AND PETRI NETS

Due to the unique characteristics possessed by cryptographic protocols, analysis and evaluation tend to be more difficult than normal protocols. Petri Nets (PN) [33] offer a way to simulate the communication patterns between the server and client as well as to validate the protocol functionality.

Petri nets are a finite-state analysis approach that explicitly provides a graphical description for cryptographic protocols. The formal definition of a Petri net is shown in Table 2 [35]. Generally Petri nets focus on specific properties such as liveness, deadlock, livelock, boundedness and safeness [34,35,36]. Typically, a petri net must consist of the following components [35]:

- A set of *places* (drawn as circles in the graphical representation) represent conditions and possible states of the system.
- A set of *transitions* (drawn as rectangles or thick bars) represent a change of state which is caused by events or actions.
- A set of *arcs* (drawn as arrows) connecting a place to a transition and vice versa.
- *Tokens* (drawn as black dots) occupy places to represent the truth of the associated condition.

TABLE.II. FORMAL DEFINITION OF A PETRI NET

<p>A Petri net is 5-tuple, <math>PN=(P,T,F,W,M_0)</math> where:  <math>P=\{p_1, p_2, \dots, p_m\}</math> is a finite set of places,  <math>T=\{t_1, t_2, \dots, t_n\}</math> is a finite set of transitions,  <math>F \subseteq (P \times T) \cup (T \times P)</math> is a set of arcs (flow relations),  <math>W:F \rightarrow \{1, 2, 3, \dots\}</math> is a weight function,  <math>M_0:P \rightarrow \{0, 1, 2, 3, \dots\}</math> is the initial marking,  <math>P \cap T = \emptyset</math> and <math>P \cup T \neq \emptyset</math>.</p> <p>A Petri net structure <math>N=(P, T, F, W)</math> without any specific initial marking is denoted by <math>N</math>.</p> <p>A Petri net with the given initial marking is denoted by <math>(N, M_0)</math>.</p>
---

Our technique involves simulation and verification by using Time-arc Petri nets. Initially, we build a PN model for client-server without intruder using TAPAAL simulation and verification software [37]. Moreover, it is worth to consider the following:

a) Define the places and transitions and declare their functionalities

b) Implement a token passing scheme once the initial marking is set.

c) Assess the model behaviour by examine reachability, boundedness, liveness.

The Petri net model in Fig. 2 represents the proposed protocol. The definitions of the places and transitions used in this model are illustrated in Table 3 and Table 4, respectively.

In our PN model, *places* mostly represent storage for requests, messages, ciphers, or session keys. *Transitions* represent actions that transform a current state to a new one. For example, the following events produce a new state: encryption, decryption, verification, and computations. *Tokens* are modelled in PN as shown in Fig. 2 to represent the key agreement and message exchange between the client and server. During simulation, the token firing rule imitates the three-way handshake procedure.

TABLE.III. DEFINITIONS OF PLACES FOR THE PROPOSED MODEL

Place	Definition	Place	Definition
$P_1$	Client random number	$P_{14}$	Encrypted SYN/ACK
$P_2$	Client timestamp	$P_{15}$	Decrypted SYN/ACK
$P_3$	SYN request	$P_{16}$	Verification message
$P_4$	Login request	$P_{17}$	Rejected request
$P_5$	Encrypted login request	$P_{18}$	Accept request – Server is authenticated
$P_6$	Decrypted login req.	$P_{19}$	Session key
$P_7$	Verification message	$P_{20}$	ACK
$P_8$	Rejected request	$P_{21}$	Encrypted ACK
$P_9$	Accepted request	$P_{22}$	Decrypted ACK
$P_{10}$	Server random number	$P_{23}$	Verification message
$P_{11}$	Server timestamp	$P_{24}$	Rejected request
$P_{12}$	Session Key	$P_{25}$	Accept request – Client is authenticated
$P_{13}$	SYN/ACK		

TABLE.IV. DEFINITIONS OF TRANSITIONS FOR PROPOSED MODEL

Trans.	Definition	Trans.	Definition
$T_1$	Compute login request + SYN	$T_{10}$	Split the packet and verify
$T_2$	Encrypt	$T_{11}$	Drop the packet
$T_3$	Decrypt	$T_{12}$	Accept
$T_4$	Split the packet and verify	$T_{13}$	Compute ACK and session key
$T_5$	Drop the request	$T_{14}$	Encrypt ACK
$T_6$	Accept	$T_{15}$	Decrypt ACK
$T_7$	Compute SYN/ACK and session key	$T_{16}$	Split the packet and verify
$T_8$	Encrypt SYN/ACK	$T_{17}$	Drop the packet
$T_9$	Decrypt SYN/ACK	$T_{18}$	Accept





cryptography during the handshake procedure. Also, it ensures data integrity by applying a Message Authentication Code function (MAC). Typically, the MAC function takes as input a secret key and data block and produces a hash value [38]. The client and server transmit the MAC value during the login and authentication phases. However, both client and server will be aware if an attacker alters the message because the integrity check of MAC value fails. When the communication session between  $C_i$  and  $S_i$  is over, the session key  $sk$  is discarded and a new session key is used in every protocol run to prevent a replay attack.

*Mutual authentication and session key agreement*

Based on FSM model and PN model, we proved that the protocol accomplished mutual authentication and secret session key agreement between a remote client and the server by establishing three-way challenge-response handshake technique. First, the client  $C_i$  sends the login request message  $\{ID_{C_i}, T_{C_i}, W_1, M_3, MAC_k(ID_{C_i}, T_{C_i}, W_1, M_3)\}$  to the server  $S_i$ . Then  $S_i$  verifies the received message by checking the MAC integrity. After validating,  $S_i$  sends a challenge message  $\{ID_{C_i}, T_{S_i}, W_2, M_6, M_7, MAC_k(ID_{C_i}, T_{S_i}, W_2, M_6, M_7)\}$  to  $C_i$ . Next,  $C_i$  check the validity of the received message  $M_7 \stackrel{?}{=} H_4(M_6 \parallel r_{C_i})$  and accept or reject the server request according to the verification result. Finally,  $C_i$  sends a response message  $M_9 = H_4(H_4(ID_{C_i} \parallel y) \oplus r_{S_i} \parallel r_{S_i})$  to  $S_i$ . Upon receiving the message,  $S_i$  verifies if  $M_9 \stackrel{?}{=} H_4(M_6 \parallel r_{S_i})$  holds. If so,  $S_i$  authenticates client  $C_i$  and allows him to get access. During the process, both  $S_i$  and  $C_i$  compute the session key  $sk = H_3(ID_{C_i}, T_{C_i}, T_{S_i}, W_1, W_2, (r_{S_i}, r_{C_i}, P))$  successfully.

*Denial-of-service-attack*

Our scheme can withstand denial-of-service attack, because when the client  $C_i$  imprints personal biometrics  $Bio^*_{C_i}$ , the  $S_i$  will check the validity of  $Bio^*_{C_i}$  with stored template by checking whether  $d(Bio_{C_i}, Bio^*_{C_i}) < \tau$  holds. According to [31], the  $Bio^*_{C_i}$  could pass the verification process even though there is some slight difference between  $Bio_{C_i}, Bio^*_{C_i}$ .

As for the computation cost, the proposed protocol is relatively low cost and efficient since only symmetric encryption; hash operations and XOR operations are required. Moreover, it is based on ECC which has significant advantages over other public-key cryptography. ECC provides the same security level of RSA cryptosystem but with a shorter key length and faster computation [39]

In Table 5, we summarised the performance and demonstrated comparisons between the proposed scheme and other related schemes. The evaluation parameters are defined in Table 6. Even though the number of operations is more than in other schemes, our scheme holds other security properties. The proposed protocol is based on a two-factor user authentication mechanism and it is obvious that it takes few more hash operations and XOR operations for the server and client. Due to the security weaknesses in related schemes, we

applied symmetric encryption and symmetric decryption to ensure the confidentiality and the integrity of transmitted packets. Therefore this feature makes the proposed scheme effective.

TABLE.V. PERFORMANCE COMPARISONS

	He et al.'s Scheme	Li-Hwang's Scheme	Proposed Scheme
Client	$2T_H + 2T_{MAC}$	$3T_H + 3T_X$	$6T_H + 3T_X + 3T_{MAC} + 2T_{SE} + T_{SD}$
Server	$4T_H + 2T_{MAC}$	$4T_H + 2T_X$	$6T_H + 2T_X + 3T_{MAC} + 1T_{SE} + 2T_{SD}$

TABLE.VI. EVALUATION PARAMETERS

Symbol	Definition
$T_X$	Time for executing an XOR operation
$T_H$	Time for executing a one-way hash function
$T_{MAC}$	Time for executing a message authentication code
$T_{SE}$	Time for executing a symmetric encryption operation
$T_{SD}$	Time for executing a symmetric decryption operation

VII. CONCLUSION AND FUTURE WORK

The paper demonstrates how a combination of ID-based encryption with biometrics can be effective and more suited to e-Government environments. Moreover, the new biometric-identity-based scheme can be integrated into e-Government systems as the main authentication method and for secure communication as well. The proposed scheme is aimed to initiate secure authentication and communication between the client and server by building a robust mechanism between communicating government parties. The presented protocol is described as a three-way handshake procedure to establish a reliable connection and ensure secure data sharing. Moreover, we have simulated and validated the behaviour of the proposed protocol by using finite-state machines and Petri nets.

In future, an in-depth security analysis and evaluation will be conducted to thoroughly assess for security vulnerabilities and weaknesses. Furthermore, it is essential to consider using Petri Nets to add an intruder model and implement a token-passing scheme. At this stage, we will examine different attacks, such as impersonation attack, man-in-the-middle attack, and replay attack against the proposed scheme and verify its security.

ACKNOWLEDGMENT

This research has been funded by Saudi Arabian Cultural Bureau in London and King Abdul Aziz University in Saudi Arabia.

REFERENCES

[1] Caloyannides, M., authentication framework and programs. IT professional, , pp. 16-21.

- [2] Sahraoui, S., Gharaibeh, G. And Al-Jboori, A., 2006. E-Government in Saudi Arabia: Can it overcome its challenges, e-Government Workshop 2006.
- [3] Evans, D. And Yen, D.C., 2005. E-government: An analysis for implementation: Framework for understanding cultural and social impact. *Government Information Quarterly*, 22(3), pp. 354-373.
- [4] Voltage Security, 2006. Identity-Based Encryption and PKI Making Security Work. [http://www.voltage.com/pdf/IBE\\_and\\_PKI.pdf](http://www.voltage.com/pdf/IBE_and_PKI.pdf) edn. Voltage Security, Inc.
- [5] Boneh, D. And Franklin, M., 2001. Identity-based encryption from the Weil pairing, *Advances in Cryptology—CRYPTO 2001* 2001, Springer, pp. 213-229.
- [6] Gentry, C. and Silverberg, A., 2002. Hierarchical ID-based cryptography. *Advances in Cryptology—ASIACRYPT 2002*, , pp. 149-155.
- [7] Al-Riyami, S. and Paterson, K., 2003. Certificateless public key cryptography. *Advances in Cryptology-ASIACRYPT 2003*, , pp. 452-473.
- [8] Boneh, D. and Boyen, X., 2004. Efficient selective-ID secure identity-based encryption without random oracles, *Advances in Cryptology-EUROCRYPT 2004* 2004, Springer, pp. 223-238.
- [9] Duffy, A. And Dowling, T., 2004. An object oriented approach to an identity based encryption cryptosystem, *Software Engineering and Applications 2004*, ACTA Press.
- [10] He, D., Chen, J. And Hu, J., 2012. An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security. *Information Fusion*, 13(3), pp. 223-230.
- [11] Li, C.T. And Hwang, M.S., 2010. An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, 33(1), pp. 1-5.
- [12] Shamir, A., 1985. Identity-based cryptosystems and signature schemes, *Advances in cryptology* 1985, Springer, pp. 47-53.
- [13] Liao, J., Xiao, J., Qi, Y., Huang, P. and Rong, M., 2005. ID-based signature scheme without trusted PKG, *Information Security and Cryptology* 2005, Springer, pp. 53-62.
- [14] Yuen, T., Susilo, W. and Mu, Y., 2010. How to construct identity-based signatures without the key escrow problem. *Public Key Infrastructures, Services and Applications*, , pp. 286-301.
- [15] Vacca, J.R., 2007. *Biometric technologies and verification systems*. Oxford: Butterworth-Heinemann.
- [16] Vielhauer, C., 2005. *Biometric user authentication for IT security: from fundamentals to handwriting*. New York ; London: Springer.
- [17] Islam, S.H. and Biswas, G., 2012. An improved ID-based client authentication with key agreement scheme on ECC for mobile client-server environments. *Theoretical and Applied Informatics*, 24(4), pp. 293-312.
- [18] Wang, D. and Ma, C., 2013. Cryptanalysis of a remote user authentication scheme for mobile client-server environment based on ECC. *Information Fusion*, .
- [19] Monrose, F., Reiter, M.K., Li, Q. and Wetzels, S., 2001. Cryptographic key generation from voice, *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on* 2001, IEEE, pp. 202-213
- [20] Seto, Y., 2002. Development of personal authentication systems using fingerprint with smart cards and digital signature technologies, The Seventh International Conference on Control, Automation, Robotics and Vision.
- [21] Clancy T.C., Kiyavash, N. and D.J. Lin, D.J., 2003. *Secure smart card based fingerprint authentication*, Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Application, WBMA.
- [22] Hao, F, Chan, C.W., 2002. Private key generation from on-line handwritten signatures, *Information Management & Computer Security*, Issue 10, No. 2, pp. 159-164.
- [23] Goh, A. and Ngo, D., 2003. Computation of cryptographic keys from face biometrics. *Communications and Multimedia Security.Advanced Techniques for Network and Data Protection*, , pp. 1-13.
- [24] Hwang, M., Lee, C. and Tang, Y., 2002. A simple remote user authentication scheme. *Mathematical and Computer Modelling*, 36(1), pp. 103-107.
- [25] Lin, C. and Lai, Y., 2004. A flexible biometrics remote user authentication scheme. *Computer Standards & Interfaces*, 27(1), pp. 19-23.
- [26] Lee, N. and Chiu, Y., 2005. Improved remote authentication scheme with smart card. *Computer Standards & Interfaces*, 27(2), pp. 177-180.
- [27] Chang, Y., Chang, C. and Su, Y., 2006. A secure improvement on the user-friendly remote authentication scheme with no time concurrency mechanism, *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on* 2006, IEEE, pp. 5 pp.
- [28] Inuma, M., Otsuka, A. and Imai, H., 2009. Theoretical framework for constructing matching algorithms in biometric authentication systems. *Advances in Biometrics*. Springer, pp. 806-815.
- [29] Lu, J., Zhang, S. and QIE, S., 2011. Enhanced Biometrics-based Remote User Authentication Scheme Using Smart Cards. *IACR Cryptology ePrint Archive*, 2011, pp. 676.
- [30] Jeon, S., Kim, H. and Kim, M., 2011. Enhanced biometrics-based remote user authentication scheme using smart cards. *J.of Security Engineering*, 8(2), pp. 237-254.
- [31] Li, X., Niu, J., Ma, J., Wang, W. and Liu, C., 2011. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer*
- [32] Hopcroft, J.E., Je rey D. Ullman. 1979. *Introduction to automata theory, languages, and computation*.
- [33] Petri, C.A., 1962. *Kommunikation mit Automaten*. Ph. D. Thesis, University of Bonn.
- [34] Peterson, J.L., 1981. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall.
- [35] Murata, T., 1989. *Petri nets: properties, analysis and applications*. Proceedings
- [36] Bobbio, A., 1990. *System modelling with Petri nets. Systems reliability assessment*. Springer, pp. 103-143.
- [37] TAPAAL 2.4.3 Petri nets simulation and verification of timed-arc Petri nets. Available ar: [www.tapaal.net](http://www.tapaal.net).
- [38] Stallings, W., 2011. *Cryptography and Network Security, 5/E*. Pearson Education, Inc.
- [39] Yokoyama, V.T.V., 2000. Elliptic curve cryptosystem. *Fujitsu Sci.Tech.J*, 36(2), pp. 140-146.