Journal of Computer Science

Original Research Paper

A New Secure Passwordless Multi-Server Modified Authenticated Master-Key Agreement Scheme Based on Hardware-Software and Iriscode Identifiers Through SSL/TLS Protocol for E-learning and Similar Web-based Services

[1]**Afshin Zivi** and [2]**Gholamreza Farahani**

[1]*Department of Computer Engineering and IT, Parand Islamic Azad University (PIAU), Parand, I. R. Iran*
[2]*Electrical Engineering and Information Technology Institute,*
Iranian Research Organization for Science and Technology (IROST), Tehran, I. R. Iran

Article history
Received: 10-07-2018
Revised: 14-09-2018
Accepted: 24-10-2018

Corresponding Author:
Afshin Zivi
Department of Computer Engineering and IT, Parand Islamic Azad University (PIAU), Parand, I. R. Iran
Email: afshinzivi@outlook.com

Abstract: There is a growing concern about systems security and how to organize it. This is because the internet which is the backbone of all systems is regarded as unsafe. Also, the internet transmits all connection transactions in the E-learning and similar web-based systems and as a result, intruders and attackers by abusing security holes can compromise the system. The E-learning and other similar systems should be safe against threats and manipulation by intruders and should protect the privacy of users. The purpose of this paper is to provide an authentication model based on hardware, software and iriscode identifiers through the SSL/TLS protocol, in order to significantly improve the security and privacy level, while at the same time, maintain the system performance at an acceptable level. There are major differences between our model and other similar works, such as: no need to password in registration and login phase, using of iriscode identifier, isolation of users profiles based on hardware and software identifiers of relying party, enhancing master secret key exchange phase in the SSL/TLS protocol, no need to password change phase, strong performance in comparison with other approaches because of using SHA-3 function and removing password change phase, capability of providing authentication services over large networks and internet. Also according to conducted studies and tests, the mentioned solution can significantly improve the system security, as well as maintain its function at an acceptable level. Therefore the proposed model easily can be used for immunize E-learning and similar web-based systems that works through internet. The proposed model improves the 32.50% security and 63.58% execution time in comparison to average of five newest methods.

Keywords: Password Less Authentication, E-learning, User Profile Isolation, Iriscode, SHA-3, SSL/TLS

Introduction

Recently, the Internet has had a huge effect on the human community and created a revolution in the 21st century. With the development of information and connection technology, the field of study cannot be isolated from the Internet and connections dispute (Dharmawansa *et al*., 2013). E-learning can be used in different ways, in different sectors and among different individuals. The network of a trading company can use E-learning to educate people in the field of trade. Online education is a fast developing field in education and has been used in many universities for numerous studies (Dharmawansa *et al*., 2013). E-learning has been developed in various fields such as distance learning, online learning and network-based learning and in general learning to promote interactions between students, lecturers and learner communities (Karforma and Ghosh, 2009). This learning method has several advantages compared to traditional learning, especially the ability to learn at any time and place (Richardson and Swan, 2003; Swan *et al*., 2000).

Most innovations in the field of E-learning focus on content development Sharable Content Object Reference Model (SCORM) and posting it, while privacy and security are not considered as a component in this regard. Although there is a growing need for higher levels of confidentiality and privacy in E-learning applications, these security technologies should be designed and implemented in such a way as to cover the existing needs. The perception and knowledge of consumers is on the rise concerning their rights to privacy and new privacy legislation has recently been introduced by various jurisdictions (El-Khatib *et al*., 2003; Zivi *et al*., 2017b).

The position of security and optimality of the system's function in the field of assessment is a special one that has been neglected in comparison with other features of these systems and has been addressed in fewer studies. For example, the study of (Attwell, 2006) on the field of assessing E-learning systems showed that in general, we can improve the way of learning and transferring concepts between the learner and professor; this study, however, has significantly mentioned the effectiveness of the E-learning system. Considering the privacy and security of discussion (Zivi *et al*., 2017b) along with the optimality and integrity of discussion in the assessment of the E-learning system, we cannot rely on this and refer to it during assessment (Zivi *et al*., 2017b).

Compared to other studies such as: (Chansuc and Praneetpolgrang, 2008), (Mustafa and Sharif, 2011), Khedr (2012) as well as (Bentley *et al*., 2012) studies, generally have assessed and improved the E-learning systems in terms of how to learn, the quality of the discussions raised, the quality of the system, the relationship between the professor and student and somehow the discussion of the effectiveness of this learning method. Therefore, according to the mentioned issues, the necessity of assessing the electronic learning system and, generally speaking, e-learning, taking into account the security discussion and covering the existing gaps in this field, as well as the issue of the integrity and optimality of this type of learning system, appears to be more intense than the past.

Despite the fact that the Internet is regarded as a place to get necessary information and knowledge, it has the potential of being turned into a place for doing illegal activities. Information on the Internet is constantly being invaded by security threats. Therefore, the E-learning environment is affected by security threats (Mohd Alwi, 2010).

The aim of this research was to provide an authentication protocol based on hardware, software and iriscode identifier through a SSL/TLS protocol for the E-learning and other similar web-based systems such that specifically, a sample has been assessed and examined in the Islamic Azad University Electronic Unit (Zivi *et al*., 2017b) in order to promote the level of privacy, security in the system, protect the student information and maintain optimal function of the system. The reason for using a SSL/TLS protocol in this authentication model has been explained in detail in a previous research (Zivi *et al*., 2017a). In the previous study, a SSL/TLS protocol was compared with 2 other protocols called IKEv2 and IPsec and in most comparisons the SSL/TLS protocol provided the best result (Zivi *et al*., 2017a).

There are major differences between our model and other similar works, such as: No need to password in registration and login phase, using of iriscode identifier, isolation of users profiles based on hardware and software identifiers of relying party, enhancing master secret key exchange phase in the SSL/TLS protocol, no need to password change phase, strong performance in comparison with other approaches because of using sha3 function and removing password change phase and capability of providing authentication services over large networks and internet.

Literature Review

Remote user authentication is a method for authenticating users who remotely communicate with a server through an insecure network (Jeon *et al*., 2011). Unfortunately, password-based authentication models are still easily hacked by dictionary attacks.

Song (2010) provide an authentication model using smart cards and passwords. But the reasons for the weakness of the this model is actually two. First, even if the model uses smart cards, it is still not easy to create a safe scheme against several attacks, because humans have difficulty remembering long passwords. Therefore, many of the models like (Song, 2010) are vulnerable to Password Guessing attacks using smart cards (Jeon *et al*., 2011). The second reason is that generally, smart card-based authentication models are used in nearby scenarios, such as ATMs and the like, therefore their use in remote scenarios requires a separate infrastructure for card generation and authentication and increases costs significantly. The reason for the increased cost is the discussion valuation of security solutions. Generally, the interdependent cost of security solutions, which is referred to as real cost or real value, is computed in order to be cost-effective. The cost includes all additional costs that are related to the given security solution. For example, for a smart card-based remote authentication system, in addition to provide the necessary equipment, such as a card reader and a generation system for this type of smart card, it should include maintenance and support costs. The costs of integrating it with other systems, hiring a team for startup and support, the cost of purchasing servers and equipment required for the implementation of this system are all a part of the cost of these types of authentication systems. Since it is vulnerable to password-guessing attacks, this method is not necessarily considered as the safest method.

Therefore, (Li and Hwang, 2010), combined a user's biometric characteristics (such as fingerprint, iris and hand geometry) with a password and smart card for designing a remote user authentication model with a higher security level. They used secret keys that had a value of high confusion (Chuang and Chen, 2014).

The main feature of Li and Hwang's model was its biometricity. Li *et al.* (2010) showed that Hwang and Li's scheme presented in 2010 did not provide the correct authentication and is therefore vulnerable to MITM attack (Jeon *et al.*, 2011). Then, Li *et al.* introduced an improved scheme to remove the weaknesses in Hwang and Li's scheme. Afterwards, (Jeon *et al.*, 2011) found that Li *et al.*'s scheme, which was actually an improved version of Hwang and Li's scheme, is vulnerable to the Replay attack, as well as because of the structural weaknesses, the password-changing phase does not work properly and the reason is that when the user gives the system a new password, the system does not compare the old password with the sample saved in the system. Therefore, if the user inadvertently inserts the old password incorrectly, the Hash value will be different from the saved value and so the smart card and authentication process will be in a suspended state (Jeon *et al.*, 2011).

But none of the above models supports multi-server environments and today this is considered as a limitation because there are now a variety of functional servers on the Internet. If the designed authentication model does not support multi-server environments, the user will have to do the registration process frequently and this will not only disturb the user, but also create a significant amount of overload on the server and network (Yang and Lin, 2014).

In the past decade, several multi-server authentication models have been introduced. (Lee *et al.*, 2011) provided a multi-server authentication model and claimed that their model was safer and more efficient than existing models. Later, (Truong *et al.*, 2013) showed that the model of Lee *et al.* introduced in 2011, could not withstand the Impersonation and stolen smart card attacks. Then, Truong *et al.* introduced a revised and modified model to overcome such attacks. But unfortunately, Truong *et al.*'s model was vulnerable to the Insider attack (Mishra *et al.*, 2014).

Then (Sood *et al.*, 2011) also proved the weakness of Hsiang and Shih's model and provided an improved model based on a dynamic identifier for multi-server environments. (However, Li *et al.*, 2012) showed that Sood *et al.*'s model cannot withstand leakage attacks on the Verification Table and the stolen smart card. In addition, they provided an improved smart card-based authentication model for multi-server architecture, which required a control server to provide mutual authentication. The existence of a control server to control the mutual authentication process made the model inefficient (Mishra *et al.*, 2014). Wang and Ma (2013) provided a smart card-based authentication model for multi-server environments. But (He and Wu, 2013) showed that Wang *et al.*'s model was vulnerable to the Privileged Insider, Server Spoofing, Impersonation and Offline Password Guessing attacks. Subsequently, (Pippal *et al.*, 2013) provided a multi-server authentication model using a smart card. They claimed that their model is resistant to Server Spoofing, Impersonation, Insider, Replay, Password Guessing, Stolen Smart Card and Stolen Verifier attacks. But (He *et al.*, 2013) explained that Pippal *et al.*'s model cannot withstand the user impersonation, Server Spoofing, Privileged Insider and Offline Password Guessing attacks. (Tsaur *et al.*, 2012) identified the models that used Timestamp to withstand the Replay attack, because the models to withstand the Replay attacks required clock synchronization between senders and receivers. To overcome this problem, they provided a Self-Verified Timestamp method to prevent the clock synchronization problem in multi-server environments.

Password-based multi-server authentication models use secret keys and passwords to perform the authentication process. But the point here is that passwords may be forgotten or lost and/or even shared with others (Lee and Hsu, 2013). But on the other hand, biometric keys such as the fingerprint, face, iris, hand geometry, palm effect and etc. do not need to be memorized. The unique biometric characteristics have increased its applications in authentication protocols. The advantages of using the biometric keys are as follows (Li and Hwang, 2010; Das, 2011):

- Biometric keys cannot be forgotten or lost
- Biometric keys are highly resistant to forging and distribution
- Biometric keys are very resistant to copying and sharing
- Biometric keys cannot be easily guessed compared to passwords that have less irregularity
- The biometric characteristics of an individual cannot be easily broken

The models of (Yang and Yang, 2010) and (Yoon and Yoo, 2013) suggested biometric multi-server authentication models, but their models did not consider the user's anonymity. Also, Yoon and Yoo model was based on the ECC method, which in general was considered as a safe method with good performance (Chen *et al.*, 2010; 2011). In addition, the model of (Yang and Yang, 2010) was not resistant to the Insider's attack and it operated based on exponential functions, which greatly increased the computational cost (Yang and Yang, 2010), while the model of (Yoon and Yoo, 2013) as shown by (He, 2011) was vulnerable to attacks by the Privileged Insider, Masquerade and stolen smart cards.

Table 1: Examined authentication models

No.	Model	Advantages	Disadvantages	Year
1	(Li and Hwang, 2010)	Using smart card Smart card based verification table Biometric feature Using hash function	MITM attacks	2010
2	(Yang and Yang, 2010)	Using smart card Smart card based verification table Biometric feature Supporting multi-server architecture Discrete logarithm problem Using hash function	Not supporting user anonymity Performance reduction because of using Exponential operation privileged insider attacks password guessing attacks	2010
3	(Song, 2010)	Using Smart Card Smart card based verification table	Using Timestamp If clocks not synchronized, vulnerable to replay attacks performance reduction because of using Exponential Operation	2010
4	(Lee *et al*., 2011)	Using Hash Function Supporting Multi-server Architecture Using Smart Card Smart Card Based Verification Table	Server Spoofing Attacks Impersonation Attack Not Supporting Biometric Feature	2011
5	(Yoon and Yoo, 2011)	Using ECC Asymmetric Encryption System Using Smart Card Smart Card Based Verification Table Biometric Feature	Privileged Insider Attacks Server Spoofing Attacks Password Guessing Attacks	2011
6	(Kim *et al*., 2012)	Using Hash Function Biometric Feature Using Smart Card Smart Card Based Verification Table Supporting Multi-server Architecture	Inefficient Login Phase Inefficient Password Change Phase Not Supporting User Anonymity	2012
7	(Li *et al*., 2012)	Using Smart Card Smart card based verification table biometric feature Improved mode of Hwang-Li Model	Replay Attacks Structural Weakness in Password Change Phase	2012
8	(Yoon and Yoo, 2013)	Using ECC Asymmetric Encryption System Using Smart Card Smart Card Based Verification Table Biometric Feature Supporting Multi-server Architecture	Not Supporting User Anonymity Privileged Insider Attacks Masquerade Attacks	2013
9	(Pippal *et al*. 2013)	Using Hash Function Using Modular Multiplication Operation Using Smart Card Smart Card Based Verification Table Supporting Multi-server Architecture	Impersonation Attack Server Spoofing Attacks Privileged Insider Attacks Offline Password Guessing Attack	2013
10	(Truong *et al*. 2013)	Using Hash Function Supporting Multi-server Architecture Using Smart Card Smart Card Based Verification Table	Privileged Insider Attacks Not Supporting Biometric Feature	2013
11	(Chuang and Chen 2014)	Using Hash Function Supporting Multi-server Architecture Using Smart Card Smart Card Based Verification Table Biometric Feature	Stolen Smart Card Attack Impersonation Attack Server Spoofing Attacks Denial of Service Attack	2014
12	(Mishra *et al*., 2014)	Using Hash Function Supporting Multi-server Architecture Using Smart Card Biometric Feature Smart Card Based Verification Table	Not efficient and secure for internet and web services, because of using vulnerable procedures of remote client side configuration	2014
13	(Amin *et al*. 2015)	User Anonymity Using Smart Card Using ECC Asymmetric Encryption System Smart Card Based Verification Table Supporting Multi-server Architecture	Stolen Smart Card Attack Stolen Verifier Attack Scalability Issues Inefficient Password Change Inefficient Password Recovery	2015
14	(Chaudhry *et al*., 2016)	Supporting Multi-server Architecture Using ECC Asymmetric Encryption System Biometric Feature Using Smart Card Smart Card Based Verification Table User Anonymity	Not efficient and secure for internet and web services, because of using vulnerable procedures of remote client side configuration	2016

Then, (Yoon and Yoo, 2011) provided a biometric authentication key model for multi-server environments. (Kim *et al*., 2012) pointed out that Yoon and Yoo's model could not withstand the Offline Password Guessing attack. Thereafter, Kim *et al*. presented an improved model resistant to Offline Password Guessing attack. But their model was inefficient in identifying password authentication in the login and password change phase (Mishra *et al*., 2014). Their model also could not maintain the user's anonymity exactly like the models of Yang and Yang and Yoon and Yoo (Mishra *et al*., 2014). Subsequently, (Chuang and Chen, 2014) provided a multi-server authenticated key verification model based on smart cards with passwords and biometric characteristics. Their model provides an optimal solution for multi-server environments in which the user can connect with all servers with one sign-up. Their model maintains the user's anonymity and has less computational overload than previous models: Yang and Yang, Yoon and Yoo and Kim *et al*. but, unfortunately, their model cannot withstand a stolen smart card attack and as a result, it leads to Impersonation and Server Spoofing attacks. Their model also cannot withstand the DoS attack (Mishra *et al*., 2014). Also, in 2015, (Amin *et al*., 2015) provided a smart card-based authentication model for hospital information systems that used the ECC encryption system. But then (Chaudhry *et al*., 2016) claimed that the model of Amin *et al*. was vulnerable to Stolen Smart Card and Stolen Verifier attacks and did not properly handle the phase of the change and recovery of the passwords. Table 1 shows the advantages and disadvantages of examined authentication models.

Methodology

The software AVISPA is a collection of tools for building and analyzing official models. This software provides a modular and high level language for determining their protocols and security characteristics and implements a wide range of analysis techniques based on the back-end components (Armando *et al*., 2005). Models are written in HLPSL format or High Level Protocol Specification Language (AVISPA, 2014b). The mode of operation of the AVISPA simulator is shown in Fig. 1 (AVISPA, 2014b).

The HLPSL language is converted to an Intermediate Format (IF) using the Interface Translator, which is the HLPSL2IF translator. IF or Intermediate Format is lower than the HLPSL and is read and implemented directly by AVISPA's back or main modules. It is also worth mentioning that the process of converting HLPSL to IF has always been hidden from the user (AVISPA, 2014b).

The AVISPA tool has four main components, the socalled back-end (AVISPA, 2014b). The most used back-end in the AVISPA simulator is the OFMC (Das *et al*., 2013; Chatterjee *et al*., 2014), which is also used in the given model. It should be noted that the AVISPA simulator (software) can be accessed from three different ways: (1). Through the web interface (AVISPA, 2014d) (2). Through the operating system version of Linux and (3). Through the Mac operating system (AVISPA, 2014c).

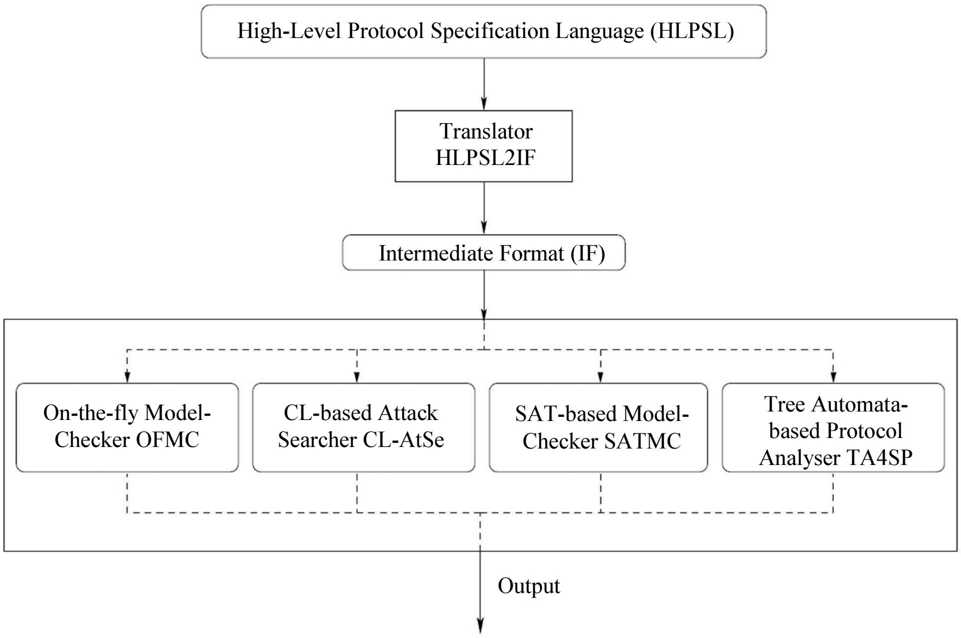

High-Level Protocol Specification Language (HLPSL)
Translator HLPSL2IF
Intermediate Format (IF)
On-the-fly Model-Checker OFMC
CL-based Attack Searcher CL-AtSe
SAT-based Model-Checker SATMC
Tree Automata-based Protocol Analyser TA4SP
Output

Fig. 1: AVISPA Architecture

If you work with the software through the web interface, you can select one of the existing protocols to simulate, in which case you can modify it or change it, or even design your protocol (AVISPA, 2014b).

The user can choose among the four back-ends to check the protocol, or even use them all together and then compare the results (AVISPA, 2014b).

It is also necessary to mention a point about the AVISPA simulator. In the AVISPA simulator, when defining the sides of the relationship, an attribute called DY is used along with the name of Channel. In fact, this attribute refers to the model presented in 1983 by two individuals, (Dolev and Yao, 1983). Under this model, the intruder has complete control over the network, in which all the messages sent by agents are also sent to the intruder. He may copy or analyze the messages and/or change them (of course as soon as he gets the keys) and can send that message to anyone he wants, of course, this is done by the opposite agent who is instead of him (AVISPA, 2014b).

Scenario1: The Model Associated with the SSL/TLS Protocol

This section has been created to provide the authentication operation for securely transmitting the server's secret key and shared key that provides SSO capability. In the SSL/TLS protocol, after the server authentication, the shared key is generated by the user's side software and then transferred to the server after being encrypted with its public key. But in this model, in addition to these, the mutual authentication is carried out for safely transfer of these two important keys. Also in this phase the shared key created using one-time use and randomly generated values in both sides.

Scenario 2: The Model Associated with the Enrollment, Login and user Authentication

In the second scenario, to provide security in the model and prevent attacks such as: MITM and Replay, Nonce is used on the server and the client. There is no need for timestamp in this model and therefore there will be no cost for its use. Some of distinguishing attributes such as: hardware, software, iriscode identifier and passwordless capability implement in this part of the authentication scheme.

Provide, Examine and Assess the Authentication Model

Given the scenario proposed for the Authentication Model in the Methodology Chapter, the authentication model discussed in this study is generally divided into two parts. Given that this authentication model is based on the SSL/TLS security protocol, shown in Fig. 2, the first part of the model was intended to secure and improve the security of the SSL/TLS protocol, which is in perfect harmony with the second part of the model. In other words, this authentication model based on hardware-software and biometric identifiers constitutes two distinct parts that can be used separately. The reason behind the creation of the first part of the model is that the SSL/TLS protocol, as a security protocol, has no efficient and strong mechanism for secure keys' transfer by the protocol.

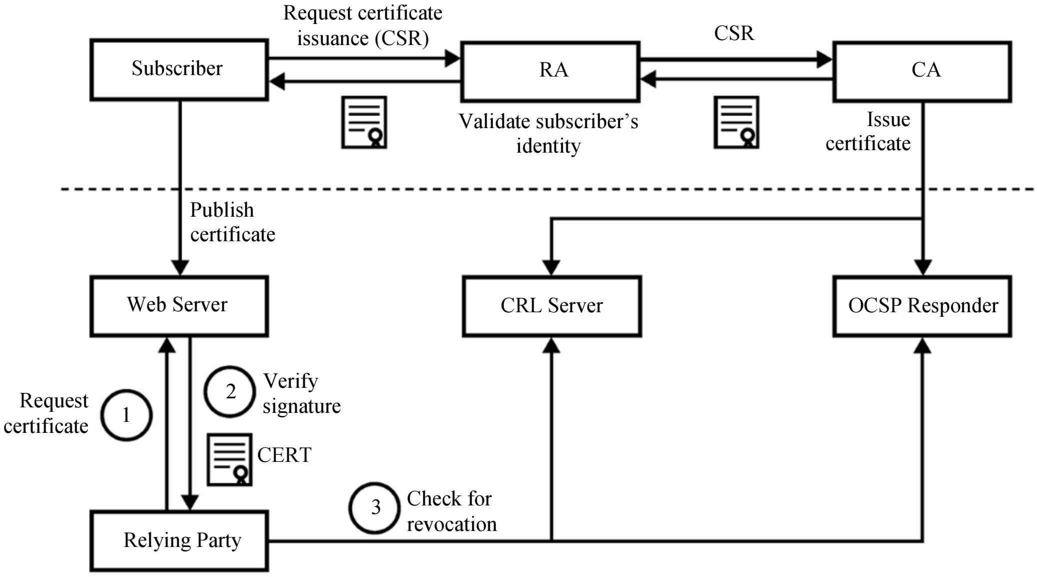


Fig. 2: Internet PKI certificate lifecycle

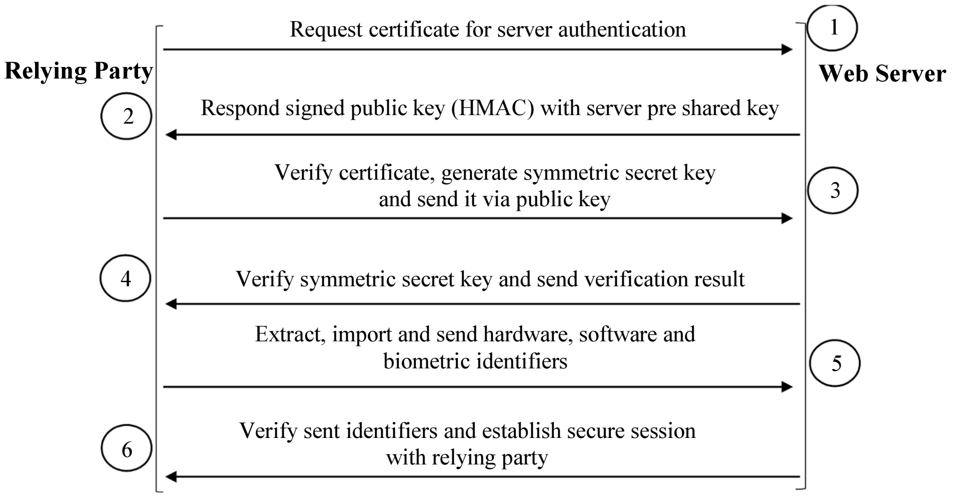

Relying Party
Web Server
1
Request certificate for server authentication
2
Respond signed public key (HMAC) with server pre shared key
3
Verify certificate, generate symmetric secret key and send it via public key
4
Verify symmetric secret key and send verification result
5
Extract, import and send hardware, software and biometric identifiers
6
Verify sent identifiers and establish secure session with relying party

Fig. 3: General Method of Authentication Procedure between User and E-learning Server

The second part of the model is fully in line with the first part of the authentication model, which operates on the basis of the safe SSL/TLS protocol. Also, in this part mutual authentication is fully supported, as well as SSO, which is also available for the user through the utilization of a shared value by the server. Also, this model is completely free of passwords and does not have many weaknesses which are associated with passwords. Using of Iriscode as the most robust and precise biometric method, will increase the security and function of this model. Finally, through the use of hardware- software identifiers, the ability to isolate users in connection with the e-learning and similar systems is created (Fig. 3).

Prerequisites

Prerequisites of the User

Authentication model should be user-friendly, the following were considered in the scheme of the model (Chuang and Chen, 2014):

- Single sign on: In this case, the user only needs to perform the enrollment process on the authentication server and then he is authenticated by all servers and can access various educational content (Chuang and Tseng, 2012)
- Anonymity: The user privacy is one of the most important cases in the authentication process. In fact, an anonymous authentication process requires that the user does not use his associated identifiers, meaning the main identifier is converted to a false one during the process

Security Prerequisites

According to the studies conducted by (He, 2011), (Li and Hwang, 2010), (Yoon and Yoo, 2011), (Yang and Yang, 2010), (Yeh *et al*., 2011), (Yoon and Yoo, 2013), (Lee *et al*., 2012) and (Li *et al*., 2013) an authentication model considers the following security prerequisites (Chuang and Chen, 2014):

- Mutual authentication: This capability points to the fact that in the authentication process, in addition to requiring the user to authenticate the server, the server should also authenticate the user
- Efficiency: Depending on the number of users who communicate with the server overnight, the authentication model should have the needed efficiency, so that the system does not encounter problems such as slowing down, resulting in increased overload on the server
- Independence of the Verification Table: In most applications and models, the central server of the authentication process saves the passwords, which can cause attacks such as Stolen-Verifier attacks. Therefore, the authentication model should be independent of keeping the Verification Table. Many models require a solution to not use the passwords Table, but this capability is fully supported because this model does not require the verification table
- Integrity: The message integrity points to the fact that the data does not change without prior detection and, in the case of unauthorized changes, this issue is quickly detected
- An agreement on the session key: After doing the authentication process, the session key generated between the client and server is exchanged to securely communicate so that the user's relationship with the server always remains secret

Characteristics of the Hash Function

A resistant hash function to one-way collision shown in the form of $h{:}\{0,1\}^{*}\rightarrow\{0,1\}^{n}$ is known as a definite or specific algorithm (Sarkar, 2010); (Stinson, 2006), which receives a binary string to the desired length of $x\in[\{0,1\}]^{*}$ and gives a binary string $x\in[\{0,1\}]^{n}$ as the output with constant length n. A hash function may receive anything as the input, such as a file, a message, or even a data block. The hash functions generally have the following characteristics (Stallings, 2005):

- The $h(.)$ function can be applied to any data size
- For each input x, it is easy to calculate the output $h(x)$ and its implementation is easy as hardware and software. The output length of the hash message $h(x)$ is constant
- Returning the value of x given by the value of y = h(x) and the function $h(.)$ is not computationally cost-effective and is referred to as a one-way attribute
- For each given input x, finding any input y ≠ x such that $h(y) = h(x)$ is not computationally cost-effective and is referred to as the "weak-collision resistant" attribute
- Finding an input pair (x, y), which is $x \neq y$ and $h(x) = h(y)$, is not economically feasible, which is referred to as the strong-collision resistant attribute

The security of this authentication model is provided through the hash function SHA-3, which has a high resistance to Collision, Preimage and 2nd Preimage attacks. This function also has 4 arbitrary lengths in the standard SHA-3 functions, which are 224 bits, 256 bits, 384 bits and 512 bits, respectively. In general, the longer the output of the hash function, the higher the security will be. For the hash function $h(.)$, when the message x is given to it, the calculation of $h(x)$ is simple; however, it is very difficult to calculate x from the $h(x)$ code and it requires high execution time, so in general, it is impossible to calculate it.

The use of hash functions often requires resistance to attacks such as Collision Resistance, Preimage Resistance, or Second Preimage Resistance. These characteristics are shown for SHA-3 and XOF functions in the following Table 2. In producing related and closely related outputs, XOF functions are different from hash functions (Bertoni *et al*., 2015).

As a result obtained from this standard, the security strengths of SHA-3 functions are shown in Table 2. The functions SHA-1 and SHA-2 were placed in the Table 2 for comparison (Bertoni *et al*., 2015).

To resist the Preimage attacks on the message M, the function $L(M)$ is defined as $\left\lceil \log_2 \frac{len(M)}{B} \right\rceil$, so that B is the length of the input block for the function in bits; in other words, for functions SHA-1, SHA-224 and SHA-256, B = 512 and for SHA-512 function, B = 1024 (Bertoni *et al*., 2015).

Four SHA-3 hash functions are an alternative to SHA-2 functions; they have been designed to withstand Collision, Preimage and Second Preimage attacks, which provide more resistance than SHA-2 functions. SHA-3 functions have also been designed to withstand other attacks, such as: Length-extension attacks, which generate more resistance through a random function on the same output length, in other words, depending on the length of the output that can be varied, the same security power is provided for them according to the random function (Bertoni *et al*., 2015; NIST, 2016).

Two SHA-3 XOF functions have been designed to withstand Collision, Preimage and Second Preimage attacks and other attacks triggered by a random function for the output length required increasing the security power of 128 bits for SHAKE128 and 256 bits for SHAKE256. A random function that has a d-bit output length cannot provide more than d/2 security bits for Collision attacks and d security bits for Preimage and Second Preimage attacks, so if d is sufficiently small, according to Table 2, SHAKE128 and SHAKE256 will provide less than 128 and 256 security bits. For example, if d = 224, then SHAKE128 and SHAKE256 provide 112 resistance bits to collision attacks, although they provide a different level of resistance to Preimage attacks: 128 bits for SHAKE128 and 224 bits for SHAKE256 (Bertoni *et al*., 2015; NIST, 2016).

Table 2: Security Strengths of SHA-1, SHA-2 and SHA-3 Hashing Functions (Bertoni *et al*., 2015)

			Security Strengths in Bits	
Function	Output Size	Collision	Preimage	2nd Preimage
SHA-1	160	< 80	160	$160-L(M)$
SHA-224	224	112	224	$min(224,256-L(M))$
SHA-512/224	224	112	224	224
SHA-256	256	128	256	$256-L(M)$
SHA-512/256	256	128	256	256
SHA-384	384	192	384	384
SHA-512	512	256	512	$512-L(M)$
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512
SHAKE128	d	$min(d/2,128)$	$\geq min(d,128)$	$min(d,128)$
SHAKE256	d	$min(d/2,256)$	$\geq min(d,256)$	$min(d,256)$

If $d > r+c/2$, then SHAKE128 and SHAKE256 offer more than 128 and 256 resistance bits to Preimage attacks; in addition, if $d > 1600$, then there may be no Preimage attack (Bertoni *et al*., 2015).

Enrollment and Authentication of the Server

The server enrollment phase is performed on the models of (Chuang and Chen, 2014; Mishra *et al*., 2014) using the IKEv2 protocol, based on the RFC-4306 document (Kaufman, 2005). The method is that initially the server which is intended to be authorized sends the request to the registration center. Then, the registration center verifies the server first and then sends a PSK key using the IKEv2 protocol to the requesting server. It should be noted that the secure transfer key algorithm in the IKEv2 protocol is the Diffie-Hellman algorithm. Then, the server validates legitimate users using the request key. But in the model presented in this study, the enrollment phase of the server as well as the server authentication using the SSL/TLS protocol, are based on the PKI infrastructure and standard X.509 (Ristic, 2015) (Fig. 4 and 5).

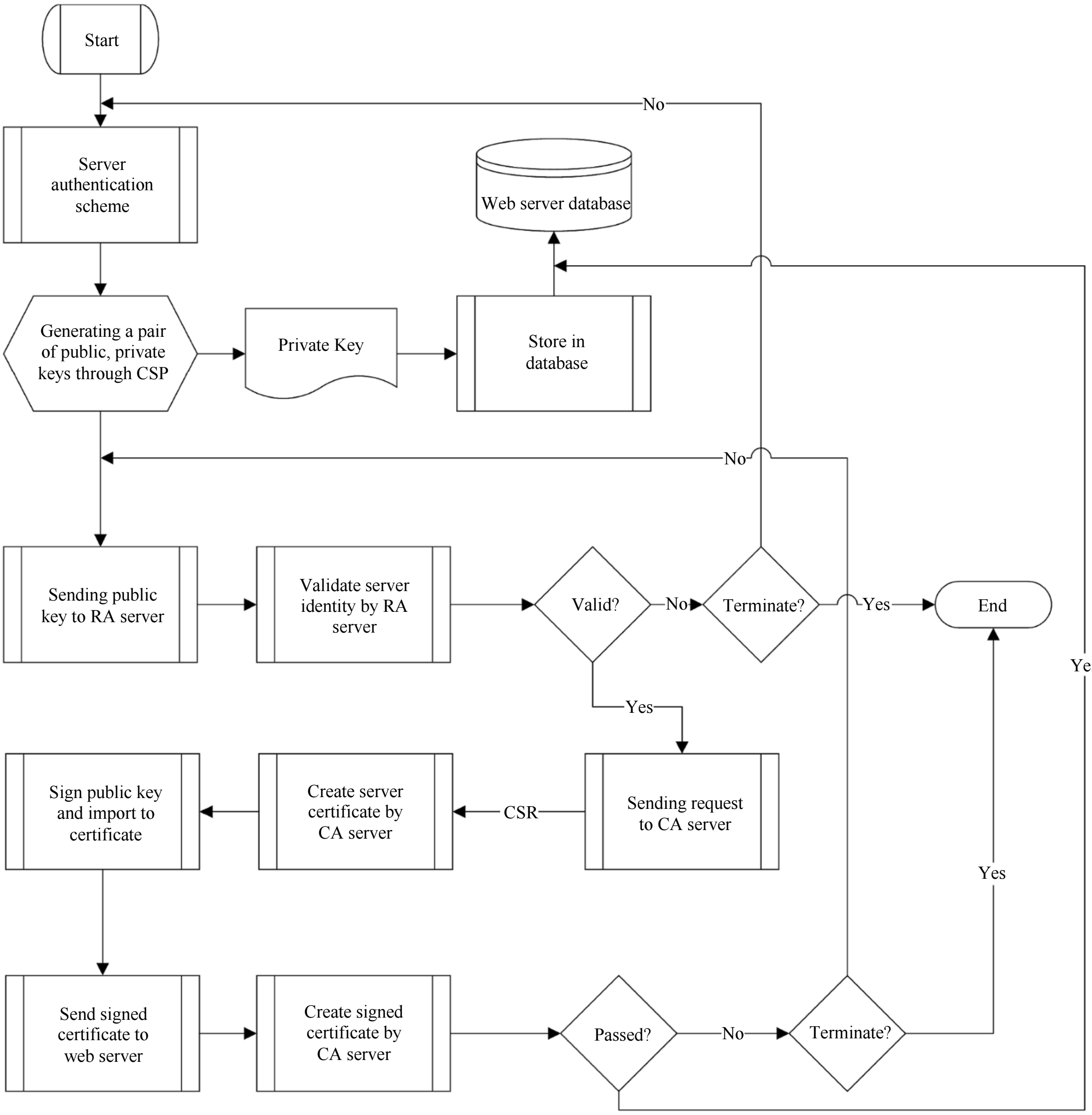

Start
No
Server authentication scheme
Web server database
Generating a pair of public, private keys through CSP
Private Key
Store in database
No
Sending public key to RA server
Validate server identity by RA server
Valid?
No
Terminate?
Yes
End
Yes
Yes
Sign public key and import to certificate
Create server certificate by CA server
CSR
Sending request to CA server
Yes
Send signed certificate to web server
Create signed certificate by CA server
Passed?
No
Terminate?

Fig. 4: Flow diagram of server registration procedure and receiving digital certificate

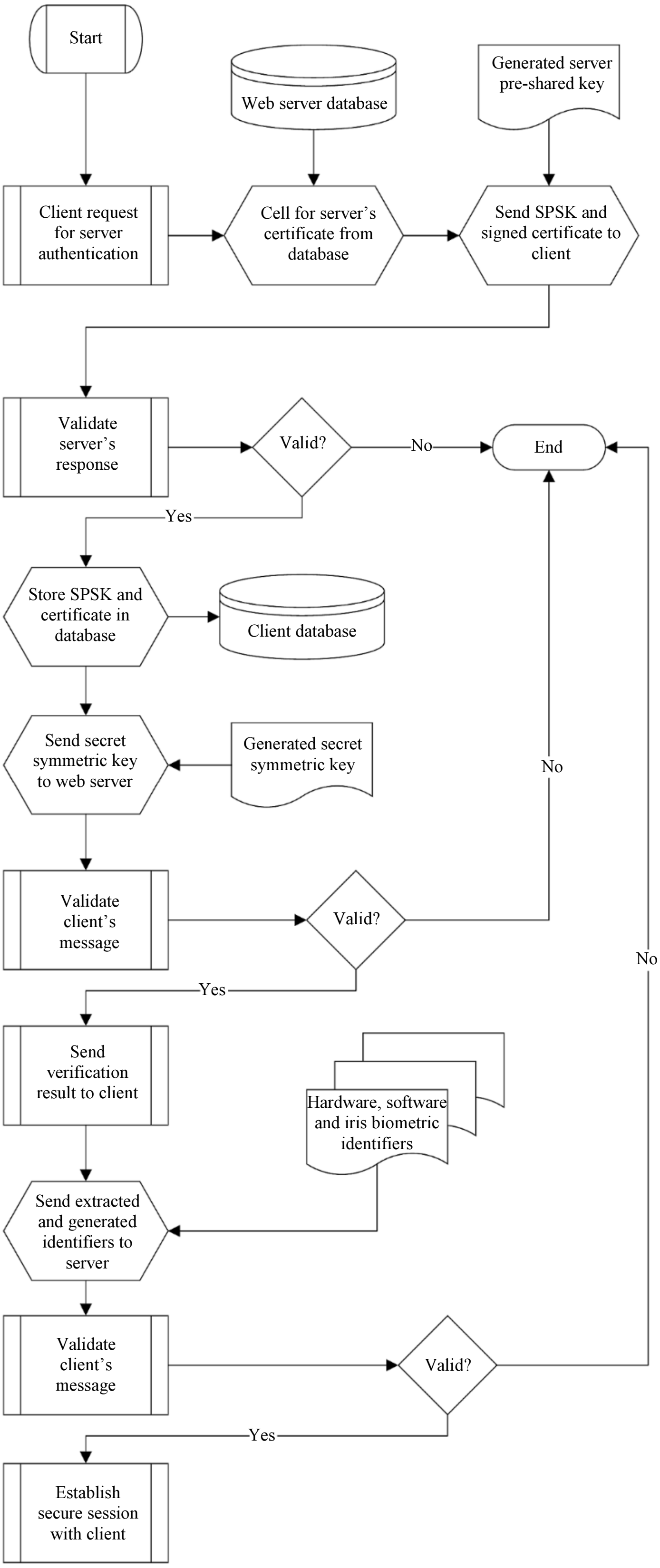

Start
Web server database
Generated server pre-shared key
Client request for server authentication
Cell for server's certificate from database
Send SPSK and signed certificate to client
Validate server's response
Valid?
No
End
Yes
Store SPSK and certificate in database
Client database
Send secret symmetric key to web server
Generated secret symmetric key
No
Validate client's message
Valid?
Yes
No
Send verification result to client
Hardware, software and iris biometric identifiers
Send extracted and generated identifiers to server
Validate client's message
Valid?
Yes
Establish secure session with client

Fig. 5: Flow diagram of server (S_j) and user (U_i) authentication procedure

Table 3: Components used in the proposed authentication model

Syntax (Symbol)	Semantic
U_i	User *i*
S_j	Server *j*
CA	Certification Authority
ID_i	Identifier of User *i*
BID_i	Biometric Identifier of User *i*
$CPUID_i$	CPU Identifier of User *i*
$BBID_i$	Base Board Identifier of User *i*
$MMID_i$	Main Memory Identifier of User *i*
$OSID_i$	Operating System Identifier of User *i*
BA_i	Browser Agent of User *i*
$DCERT_j$	Digital Certificate of Server *j*
$MPSK_j$	Master Pre-Shared Key of Server *j*
$SRVID_j$	Server Identifier of Server *j*
HF3(.)	SHA-3 one-way hash function
$\oplus$	XOR Operation
$\|\|$	Concatenation Operation
N_i	Nonce of User *i* in the user registration phase
N_j	Nonce of Server *j* in the server authentication phase
N_k	Nonce of User *i* in the user login and authentication phase
N_l	Nonce of Server *j* in the user login and authentication phase
N_x	Nonce of User *i* in the server authentication phase
N_y	Nonce of Server *j* in the server authentication phase
T_r	Time of Registration request in the user registration phase
SSKuisj	Secret Symmetric Key that shared between User *i* and Server *j* in the server authentication phase
A_1, A_2, A_i, K_1, K_2, K_3, L_i, M_i, O_i,P_i, Q_i, Q_1, Q_2, R_1, R_2, R_3, R_4, R_5, R_6, X_1, X_2	Temporary Variables

The Authentication Model based on Hardware-Software and Iriscode Identifiers

As earlier mentioned, the authentication model has 2 parts. One of the key characteristics of this model is the lack of Timestamp to prevent Replay or MITM attacks. Using Timestamp characteristic prevents the above attacks, but it should also be noted that if Timestamp should be used, we need Clock Synchronization which actually incurs additional costs involved in the implementation discussion and if the Time Zone is different between the different users the problem will become more evident, even considering the value of offset for the time of sending and receiving packets will not also be effective. The presented model is based entirely on Nonce and while resistant to Replay and MITM attacks, there is no execution time due to the implementation of Timestamp. The Table 3 shows the parameters used in the model.

Enrollment Phase of the Server

- Stage 1: First, the server S_j sends a digital certification request to the CA server to receive a digital certificate. All of these connections are based on the PKI infrastructure and X.509 standard. It should be noted that there is generally a digital certification request using the SCEP protocol. The enrollment process for the server S_j is called Certificate Enrollment
- Stage 2: At this stage, the CA server will first authenticate the server S_j to ensure that the server S_j is the same server that is with the CA server and has the same identifiers validated by the CA server. If the authentication is successful, it goes to the next stage, otherwise the connection will be disconnected
- Stage 3: At this stage, the CA server signs the public key of the server S_j using its private key and creates a digital certificate specifically for it, called $DCERT_j$. Finally, this digital certificate will be sent to the server S_j
- Stage 4: At the final stage, the server S_j checks and validates the digital certificate after receiving it and the process for requesting and obtaining the digital certificate is completed (Fig. 6)

Authentication Phase of the Server

After the server *S_j* sends the digital certification request to the CA server, a digital certificate will be sent to it by performing a series of processes and authentication of the server S_j. The certificate essentially contains a set of parameters that are located alongside the public key of the server S_j and proves its identity in front of all the referring users. When a user U_i is referred to the server S_j, the server S_j is initially authenticated and in this phase the secret key is exchanged between the user and the server. In this phase, in addition to the secret key exchange, mutual authentication is also performed to ensure the correct parties (Fig. 7):

- Stage 1: First, the server Sj sends the $DCERT_j$ digital certificate received from the CA server to the user U_i
- Stage 2: Upon receipt of the $DCERT_j$ certificate, the user U_i verifies the four conditions including: (1). Because the public key of the server S_j has been signed by the private CA server, so the user U_i first tries to decrypt it by the public key of the CA server, if the public key on the side of the user U_i can open it, this stage is successful, otherwise the connection will be disconnected. It should be noted that the value encrypted by the CA server's private key is known as HMAC, which contains the public key of the server S_j. (2). At the next stage, the user U_i compares the name of the certificate recipient with the address bar of his web browser and if it matches, this stage will also be successful, otherwise the connection will be disconnected. (3). Then the user U_i checks the duration of the validity of the $DCERT_j$ certificate, which should not be past the validity period. (4). Finally, the user U_i checks that the $DCERT_j$ certificate issued for the server S_j is not in the list of void certificates or CRLs. If these four conditions are met, the user U_i authenticates the server S_j identity (Fig. 9)
- Stage 3: Then the server S_j generates a single-use or nonce random number, called N_j and using its own identifier i.e., $SRVID_j$, $DCERT_j$, N_j creates a shared key called $MPSK_j$, which is as follows: $MPSK_j = HF3(N_j||SRVID_j||DCERT_j)$. The server S_j then encrypts this key using the U_i public key and sends it to the user U_i
- Stage 4: After receiving the $MPSK_j$ key, the user U_i generates a single-use or nonce random number, called N_x. Then, using this value, generates a message called A_1 in this form: $A_1 = HF3(MPSK_j)\oplus N_x$, encrypts it by the public key of the server S_j and then sends it to the server S_j
- Stage 5: The server S_j first extracts the N_x value through the Eq. $N_x = A_1 \oplus HF3(MPSK_j)$. Then, the server S_j selects a nonce called N_y and using the two values of N_x and N_y generates a message called A_i in this way: $A_i = HF3(N_x)\oplus N_y$. Then, using the value of A_i, a message called A_2 is created in this way: $A_2 = HF3(MPSK_j)\oplus A_i$, encrypts it with the public key of the user U_i and finally sends it to the user U_i
- Stage 6: The user U_i initially extracts the value of Ai from the Eq. $A_i = A_2 \oplus HF3(MPSK_j)$; then, using the value of A_i and by means of the Eq. $N_y = A_i \oplus HF3(N_x)$, N_y is obtained. Finally, using the values of N_x, N_y, A_i, $DCERT_j$ and $MPSK_j$, a secret symmetric key is generated, which is as follows: SSKuisj $= HF3(MPSK_j||DCERT_j||A_i||N_x||N_y)$ and the result of the confirmation is sent to the server S_j as encrypted
- Stage 7: The server S_j, after receiving the confirmation message, calculates the secret symmetric key with the above values as in Stage 6 (Fig. 7)

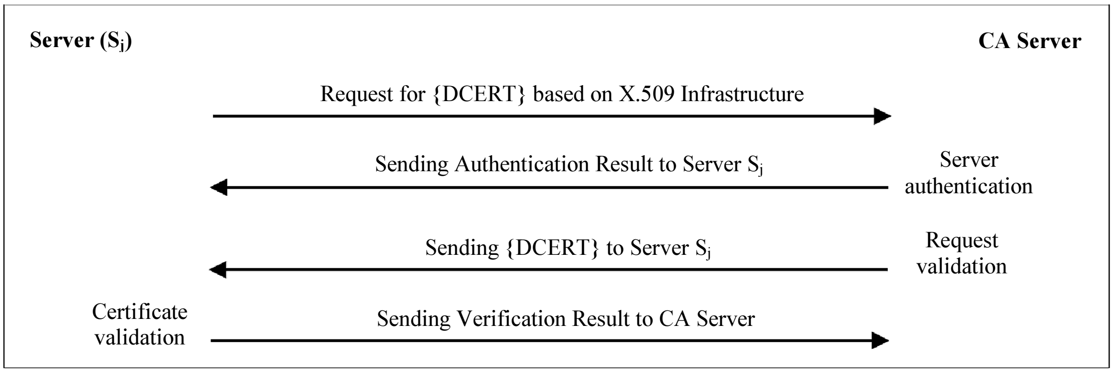

Server (S_j)
CA Server
Request for {DCERT} based on X.509 Infrastructure
Sending Authentication Result to Server S_j
Server authentication
Sending {DCERT} to Server S_j
Request validation
Certificate validation
Sending Verification Result to CA Server

Fig. 6: Server (S_j) registration procedure in CA server

]

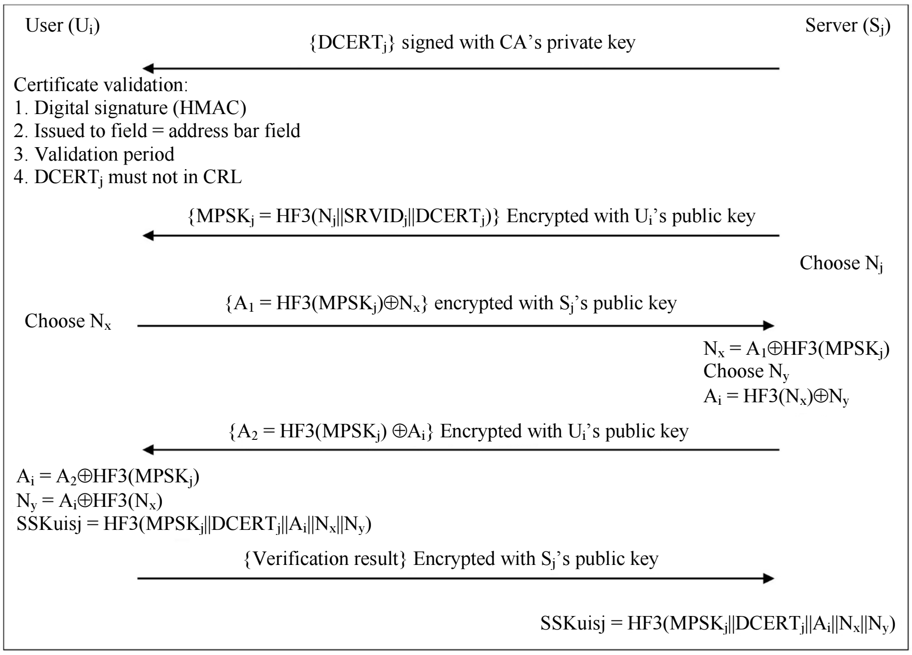

User (U_i)
Server (S_j)
{$DCERT_j$} signed with CA's private key
Certificate validation:
1. Digital signature (HMAC)
2. Issued to field = address bar field
3. Validation period
4. $DCERT_j$ must not in CRL
{$MPSK_j = HF3(N_j||SRVID_j||DCERT_j)$} Encrypted with U_i's public key
Choose N_j
{$A_1 = HF3(MPSK_j) \oplus N_x$} encrypted with S_j's public key
Choose N_x
$N_x = A_1 \oplus HF3(MPSK_j)$
Choose N_y
$A_i = HF3(N_x) \oplus N_y$
{$A_2 = HF3(MPSK_j) \oplus A_i$} Encrypted with U_i's public key
$A_i = A_2 \oplus HF3(MPSK_j)$
$N_y = A_i \oplus HF3(N_x)$
$SSKuisj = HF3(MPSK_j||DCERT_j||A_i||N_x||N_y)$
{Verification result} Encrypted with S_j's public key
$SSKuisj = HF3(MPSK_j||DCERT_j||A_i||N_x||N_y)$

Fig. 7: Server registration procedure and secure transmission of secret key and shared key

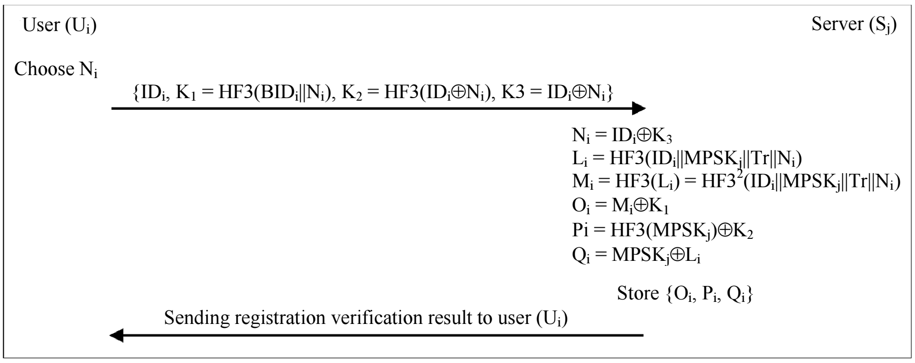

User (U_i)
Server (S_j)
Choose N_i
{ID_i, $K_1 = HF3(BID_i||N_i)$, $K_2 = HF3(ID_i \oplus N_i)$, $K3 = ID_i \oplus N_i$}
$N_i = ID_i \oplus K_3$
$L_i = HF3(ID_i||MPSK_j||Tr||N_i)$
$M_i = HF3(L_i) = HF3^2(ID_i||MPSK_j||Tr||N_i)$
$O_i = M_i \oplus K_1$
$Pi = HF3(MPSK_j) \oplus K_2$
$Q_i = MPSK_j \oplus L_i$
Store {O_i, P_i, Q_i}
Sending registration verification result to user (U_i)

Fig. 8: User (U_i) Registration Procedure in Server (S_j)

Enrollment Phase of the User

After performing the server S_j authentication stage and exchanging the secret session key, all messages are encrypted and decrypted using the *SSKuisj* secret key. In this phase, the user U_i plans to enroll on the server S_j. The user U_i enrollment phase is performed using the parameters of ID_i, biometric identifier BID_i and the single-use random number or nonce N_i. according to Fig. 8, These stages are as follows:

- Stage 1: The user U_i initially generates a nonce called Ni, then using three temporary variables of K1, K2 and K3 creates the following messages: $K_1 = HF3(BID_i||N_i)$, $K_2 = HF3(ID_i \oplus N_i)$ and $K_3 = ID_i \oplus N_i$. In fact, the user U_i enters into the system, his IRIS biometric ID in the 3 messages and also confirms that he has generated and sent his nonce N_i value. Finally, the user U_i sends a message consisting of four components $\{ID_i,K_1,K_2,K_3\}$ to the server S_j
- Stage 2: After receiving messages from the user U_i, the server S_j first calculates the N_i value through the Eq. $N_i = ID_i \oplus K_3$. Then the server S_j begins to make several new messages using the sent values: $L_i = HF3(ID_i||MPSK_j||T_r||N_i)$, $M_i = HF3(L_i) = HF3(ID_i||MPSK_j||T_r||N_i)$, $O_i = M_i \oplus K_1$, $P_i = HF3(MPSK_j) \oplus K_2$ and $Q_i = MPSK_j \oplus L_i$. In this phase, T_r is the time taken to receive the messages $\{ID_i,K_1,K_2,K_3\}$, which the server S_j itself records
- Stage 3: Finally, the server S_j saves a message consisting of three components $\{O_i,P_i,Q_i\}$ and sends a confirmation message to the user U_i

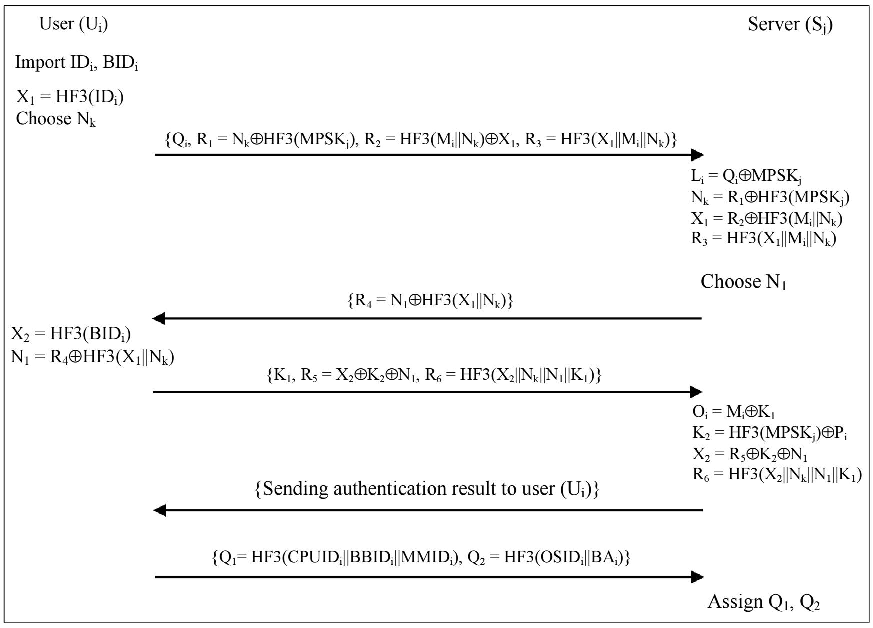

User (U_i)
Server (S_j)
Import ID_i, BID_i
$X_1 = HF3(ID_i)$
Choose N_k
$\{Q_i, R_1 = N_k \oplus HF3(MPSK_j), R_2 = HF3(M_i||N_k) \oplus X_1, R_3 = HF3(X_1||M_i||N_k)\}$
$L_i = Q_i \oplus MPSK_j$
$N_k = R_1 \oplus HF3(MPSK_j)$
$X_1 = R_2 \oplus HF3(M_i||N_k)$
$R_3 = HF3(X_1||M_i||N_k)$
Choose N_l
$\{R_4 = N_l \oplus HF3(X_1||N_k)\}$
$X_2 = HF3(BID_i)$
$N_l = R_4 \oplus HF3(X_1||N_k)$
$\{K_1, R_5 = X_2 \oplus K_2 \oplus N_l, R_6 = HF3(X_2||N_k||N_l||K_1)\}$
$O_i = M_i \oplus K_1$
$K_2 = HF3(MPSK_j) \oplus P_i$
$X_2 = R_5 \oplus K_2 \oplus N_l$
$R_6 = HF3(X_2||N_k||N_l||K_1)$
{Sending authentication result to user (U_i)}
$\{Q_1 = HF3(CPUID_i||BBID_i||MMID_i), Q_2 = HF3(OSID_i||BA_i)\}$
Assign Q_1, Q_2

Fig. 9: User (U_i) Login and Authentication Procedure by Server (S_j)

Login and Authentication Phase of the User

In this phase, the user U_i intends to enter the website of the E-learning system. After performing the authentication process of the server S_j, the following stages are performed as follows, (Fig. 9):

- Stage 1: First, the user U_i will enter into the system his identifier ID_i and then the biometric identifier BID_i
- Stage 2: Then the user U_i takes a single-use or nonce random value, considers $HF3(ID_i)$ and calls them N_k and X_1, respectively. Then, the user U_i calculates 3 messages to this form through these values: $R_1 = N_k \oplus HF3(MPSK_j)$, $R_2 = HF3(M_i||N_k) \oplus X_1$ and $R_3 = HF3(X_1||M_i||N_k)$. Finally, the user U_i sends a message consisting of four components $\{Q_i,R_1,R_2,R_3\}$ to the server S_j
- Stage 3: As soon as the message $\{Q_i,R_1,R_2,R_3\}$ is received, the server S_j using its own dedicated key, $MPSK_j$, extracts L_i value from the Q_i message, as follows: $L_i = Q_i \oplus MPSK_j$. Then, the N_k and X_1 values are extracted into this form: $N_k = R_1 \oplus HF3(MPSK_j)$ and $X_1 = R_2 \oplus HF3(M_i||N_k)$. It is worth mentioning that for the server S_j to obtain the value of M_i, it only needs to calculate the second order hash function of L_i
- Stage 4: Then the server S_j checks the correctness of the Eq. $R_3 = HF3(X_1||M_i|| N_k)$ using the obtained values, i.e. X_1, M_i and N_k. If the Eq. exists, it goes to the next stage, otherwise the connection is immediately disconnected
- Stage 5: At this stage, the server S_j chooses a nonce called N_l, then generates the message $R_4 = N_l \oplus HF3(X_1||N_k)$ and sends it to the user U_i
- Stage 6: The user U_i first obtains the N_l value through the Eq. $N_l = R_4 \oplus HF3(X_1|| N_k)$ and calculates the value of $X_2 = HF3(BID_i)$. Then it computes the messages of $R_5 = X_2 \oplus K_2 \oplus N_l$ and $R_6 = HF3(X_2||N_k||N_l||K_1)$ and finally sends a message consisting of 3 components $\{K_1,R_5,R_6\}$ to the S_j server
- Stage 7: The server S_j first sets up the K_1 message in the Eq. $O_i = M_i \oplus K_1$, calculates the O_i value and compares the new O_i value with the value saved in its memory; if they are equal, they will go to the next stage otherwise the connection is disconnected. Then, the server S_j obtains K_2 and X_2 values, respectively, through the Eqs. $K_2 = HF3(MPSK_j) \oplus P_i$ and $X_2 = R_5 \oplus K_2 \oplus N_l$ respectively. With K_1 and X_2 values, the server S_j verifies the correctness of the Eq. $R_6 = HF3(X_2||N_k||N_l||K_1)$. If the new value matches the value sent by the user U_i, this stage is successful and the user U_i is identified as an authenticated user, otherwise the connection is immediately disconnected
- Stage 8: At the final stage, the user U_i extracts his hardware and software identifiers from the system and sends them to the server S_j in this way: $Q_1 = HF3(CPUID_i||BBID_i||MMID_i)$ and $Q_2 = HF3(OSID_i||BA_i)$ Finally, the server S_j attributes these two messages to the user U_i and uses them to identify the log in usage; that is, if the user U_i intends to be in

place of someone else when he/she enters the E-learning system it will not be possible for him/her to enter, because he/she can only log in once for his hardware and software characteristics. (Fig. 9)

Prove the Authentication Model based on the Analysis of Security Relationships

This section examined several different states from the standpoint of security relationships on the model. Some of these analyses are done by considering a third element i.e., the attacker in the model and others are examined based on the characteristics that are supported in the model.

Insider Attack

In the provided model, the user U_i never sends his/her own identifier data i.e., biometric identifier directly to the server S_j, but it is actually sent in the form of an implicit message, $K_1 = HF3(BID_i||N_i)$. Also, in the destination i.e., the server S_j, the random number or nonce is obtained from the Eq. $N_i = ID_i \oplus K_3$. And virtually none of Ni and BID_i values are explicitly used. As a result, the discussed model is entirely safe against the Privileged Insider attack.

User Anonymity

The combined message $\{Q_i,R_1,R_2,R_3\}$ will be sent during the implementation of the login and authentication phase. The ID_i is virtually protected by $HF3(M_i||N_k)$, while the N_k value is protected by $HF3(MPSK_j)$. To get an ID_i, both M_i and $MPSK_j$ values are required. Because M_i is protected by 2 components of ID_i and BID_i, it is impossible to obtain M_i by the intruder (I). Also, the user's anonymity is supported by hiding ID_i by $HF3(M_i||N_k)$.

Password Guessing Attack

The model in question is also completely free of the use of password. This is because the model only works with ID_i and BID_i, which are biometric identifiers of the Iris type proven in Chapter 2 and is considered as the safest biometric method with the least error. Therefore, none of the password-related attacks are a threat to this model.

Using Randomly Generated and one Time Use Values

Implementing Timestamp mechanism to counteract Replay attacks requires Clock Synchronization between the sender and receiver during login and authentication operations, which will result in execution times as well as development and implementation costs. The model in question is completely based on the non-randomized single-time or nonce and unique random number structure and each time the values of the secret key, shared key, login and authentication messages are changed.

Replay and MitM Attacks

An intruder or attacker with (I) symbol to perform Replay and MitM attacks uses previously sent messages $\{Q_i,R_1,R_2,R_3\}$, $\{R_4\}$, $\{K_1,R_5,R_6\}$ and/or $\{A_1,A_i,A_2\}$. But the model under discussion has the ability to withstand Replay and MitM attacks. The following stages illustrate this issue:

- The intruder (I) replays messages $\{Q_i,R_1,R_2,R_3\}$ to the server S_j, where $R_1 = N_k \oplus HF3(MPSK_j)$, $R_2 = HF3(M_i||N_k) \oplus X_1$ and $R_3 = HF3(X_1||M_i||N_k)$
- Upon receipt of the messages $\{Q_i,R_1,R_2,R_3\}$, the server S_j extracts the values of $L_i = Q_i \oplus MPSK_j$, $N_k = R_1 \oplus HF3(MPSK_j)$ and $X_1 = R_2 \oplus HF3(M_i||N_k)$ and finally approves the message $R_3 = HF3(X_1||M_i||N_k)$. Because all these messages are exactly replayed by the intruder (I), so these stages are done without interruption
- Then the server S_j selects a random and single-use number N_l' and then sends the message $R_4' = N_l' \oplus HF3(X_1||N_k)$ to the user U_i. The intruder (I) will send the message R_4 and then try to replay the server S_j using a suitable message, but for two reasons it does not succeed:

    1. When the intruder (I) tries to replay the server S_j using messages $\{K_1,R_5,R_6\}$, after receiving these messages, the server S_j tries to retrieve the X_2 value and then verifies the correctness of the Eq. R_6, but because $N_l \neq N_l'$, this does not happen and the connection is disconnected i.e., $R_5' \neq R_5$ and $R_6' \neq R_6$
    2. Even if the intruder (I) wants to replay the server S_j using the messages $\{R_5, R_6'\}$, it should be able to compute the values of N_l', N_k, K_2 and X_2. It is known that the values of X_2 and K_2 contain BID_i and ID_i and the intruder (I) cannot actually capture them. Also, the K_2 value is obtained from the combination of ID_i and N_i, so the intruder (I) should also guess the value of N_i, as well as the values N_k and N_l' also have the same procedure. In addition, these values are combined with each other by the XOR operator and SHA-3 hash function. So the intruder (I) cannot use messages $\{R_5',R_6'\}$ for Replay and MitM attacks

It is also true that the server authentication phase is exactly the same as the routine. That is, if the intruder (I) replays the message A_1 to the server S_j, the server S_j first retrieves the N_x value and continues to work without interruption. At the next stage, the server S_j chooses a single-use random value of N_y', sends the message A_i'

and finally A_2' message and sends the message $\{A_2'\}$ to the user U_i. But the intruder (I) takes that message and tries to reply the server S_j which have the encrypted messages. Given that the replay from the intruder (I) is primarily the confirmation of the results, the server S_j calculates the secret key for itself and the intruder (I) has no access. Even if the intruder (I) decides to guess the secret key, this will not be possible because the key is originally protected by N_x, N_y', A_i and $MPSK_j$ values. Also, the $MPSK_j$ value is also protected by the values of $SRVID_j$ and N_j, so the intruder (I) should also guess these values. Therefore, the server authentication phase is also fully resistant to Replay and MitM attacks.

User Impersonation Attack

In this attack, the intruder (I) can put itself in the place of the legal user U_i and enter the server S_j or get the key value in the server's authentication phase. Although our proposed model is quite resistant to this attack, the intruder (I) cannot take advantage of this attack:

- Intruder (I) may want to use a Replay attack on the server Sj, which will not succeed because of the resistance of this model against this attack
- In the latter case, intruder (I) can create artificial messages $\{Q_i,R_1',R_2',R_3'\}$ or $\{A_1\}$using a random and single-use value N_I: $R_1' = N_I \oplus HF3(MPSK_j)$, $R_2' = HF3(M_i||N_I)$ and $R_3' = HF3(X_1||M_i||N_I)$, $A_1' = HF3(MPSK_j)$. The attempt of intruder (I) will fail because these messages cannot be calculated, as a result of the following reasons:

  1. Intruder (I) is not able to calculate the messages A_1' and R_1' because there is a need to know the $MPSK_j$ value and this also requires that intruder (I) knows the values of $SRVID_j$ and N_j which is virtually impossible
  2. Intruder (I) cannot calculate the message R_2'. Because X_1 and M_i information is required to calculate R_2'. X_1 basically contains ID_i which intruder (I) does not actually know, as well as the value of M_i is protected by the values of T_r, ID_i, $MPSK_j$ and N_i and thus the calculation of R_2' message is practically impossible for intruder (I)
  3. Intruder (I) also cannot calculate the R_3' message. To calculate R_3' just like R_2', intruder (I) needs to know X_1 and M_i, which is practically impossible
  4. Also, intruder (I) is not able to get the message M_i. To calculate the value of M_i from the message O_i, intruder (I) should know the value of K_1, which is protected by the values of BID_i and N_i and if it is desired to calculate it directly, it is much harder because the values of ID_i, $MPSK_j$, T_r and N_i should be obtained. This is also impossible

Server Spoofing Attack

Under this attack, intruder (I) can be replaced with the server S_j. This is actually impossible because in this model, the server S_j is always authenticated through the SSL/TLS protocol and in this process the server S_j provides $DCERT_j$ and $MPSK_j$ values which, as stated above, the $MPSK_j$ value is completely secure against this attack. The $DCERT_j$ component is in fact a digital certificate that the server S_j provides to the CA server and it's impossible to rebuild it. Even after these stages, also calculating and transmitting the messages A_1, A_i and A_2 can lead to the safe calculation of the secret key as well as the authentication of the parties, which makes it even more difficult to perform. However, if this impossible assumption becomes possible, it will still not be possible for intruder (I) to complete the login and authentication processes:

- When the user U_i sends the login request $\{Q_i,R_1',R_2',R_3'\}$ to the server S_j, intruder (I) takes this message when sending, which is $R_1' = N_k' \oplus HF3(MPSK_j)$, $R_2' = HF3(M_i||N_k')$ and $R_3' = HF3(X_1||M_i||N_k)$. Intruder ($I$) can resend the combined message $\{R_4\}$ in $\{Q_i,R_1',R_2',R_3'\}$ when replying to the received message. In this case, intruder (I) will not succeed because the R_4 message contains N_I value that was previously used and naturally $N_I \neq N_I'$
- In the latter case, intruder (I) tries to make the message R_4'. In order to make this message, intruder (I) needs to know the values of X_1 and N_k' that it is impossible and thus intruder (I) is not able to calculate the R_4' message

Mutual Authentication

The model presented in this study supports mutual authentication in all phases. Generally, in this model, first the server is authenticated on the basis of PKI infrastructure and the X.509 standard and then on the user's login and authentication phase, the user U_i is authenticated. But in each phase, authentication is also performed when transmitting messages. In the server authentication phase, after providing $\{DCERT_j, MPSK_j\}$ identifiers when transmitting $\{A_1,A_i,A_2\}$ messages, both the user and server are authenticated and this is done by nonce values, the server ID, the shared key and the secret key calculation in both sides. Also, in the login and authentication phase, the user U_i identity and server S_j are re-evaluated by exchanging messages $\{Q_i,R_1,R_2,R_3\}$, $\{R_4\}$ and $\{K_1,R_5,R_6\}$. Therefore, this model fully supports Mutual Authentication.

Secrecy of the Known Key

Assuming that the $SSKuisj$ key between the server S_j and user U_i has been gotten by intruder (I). But the

SSKuisj key does not disclose any information from other connections between the user U_i and server S_j:

- Each SSKuisj secret key has been created using SHA-3 hash function, which has a high resistance to Collision, Preimage and 2nd Preimage attacks. Therefore, no information can be obtained from inside
- Also, each *SSKuisj* secret key is made using variable components of $MPSK_j$, A_i, N_x and N_y, each with different values

Therefore, no information about other connections between the server S_j and user U_i is disclosed and as a result, this model supports the Known Key Secrecy.

Forward Secrecy

Intruder (*I*) can calculate the *SSKuisj* session key by parameters A_i, N_x, N_y and $MPSK_j$. But this is impossible for intruder (*I*) because:

- In order to calculate *SSKuisj* key, the value of A_i is required. This value is always hidden against intruder (*I*) and protected by the values of N_x and N_y
- Also, in order to calculate the *SSKuisj* key, N_x and N_y values are required which intruder (*I*) cannot calculate because they are single-use and do change in each connection
- Finally, intruder (*I*) needs to calculate the *SSKuisj* key to know the $MPSK_j$ value, which is virtually impossible, since this value is also generated using N_j and $SRVID_j$

Therefore, the authentication model fully supports the Forward Secrecy.

Randomly and Temporary Information Attack

The attack also stems precisely from the lack of Forward Secrecy, which means that intruder (*I*) tries to make *SSKuisj* using N_x, N_y, A_i and $MPSK_j$ values. But this is impossible because the values of N_x and N_y are always hidden and even if we assume that these two values are at the hand of intruder (*I*), the $MPSK_j$ value is still hidden and protected by the SHA-3 hash function and the values of N_j and $SRVID_j$. N_x and Ny values are also protected by the SHA-3 hash function and their value changes in each connection. In addition, they are single-use. So the likelihood of the attack is also lost.

Agreement and Verification on the Session Key

It is important to note that the key $SSKuisj = HF3(MPSK_j||DCERT_j||A_i||N_x||N_y)$ is always calculated between the user U_i and server S_j and is not transmitted at all. Also, to calculate this key, its components are always verified by both the user and server. Therefore, Session Key Agreement and Verification is fully supported.

User and Computer Isolation Based on Hardware-Software and Iriscode Identifiers

The capability that distinguishes this authentication model from other models is the use of hardware and software identifiers. Using these identifiers, each user can only log in to the E-learning system through a computer system at the same time and if he wants to enter instead of several people, it will not be available to him through the biometric characteristic as well as hardware and software characteristics. Because his hardware and software characteristics are assigned only once each time. Also, the biometric characteristic will not allow it. Even if several students are willing to perform the biometric authentication, they will not be able to log into the E-learning system through a computer system by assigning hardware and software characteristics as well. Therefore, the model under discussion completely supports this new capability.

Secure Passwordless Scheme

Another unique capability of this model is that there is no need for a password. Passwords may be forgotten, taken, or stolen. But the biometric characteristic does not have these weaknesses and because the model presented in this study uses the Iris biometric characteristic, so this authentication model is in fact the most precise and safest method of authentication.

Efficient SSO Capability

Given that the discussed authentication model uses the sharing key that all servers previously shared among themselves, this model fully supports single authentication.

Assess the Authentication Model using the AVISPA simulator

This section assessed the authentication model using AVISPA simulator. The assessment uses AVISPA and OFMC is the component of the model implementation for performing security or back-end simulations. This assessment is also carried out in the form of two scenarios. The first scenario is, in fact, the first part of the authentication model that acts on the basis of the SSL/TLS protocol, while the second scenario entails enrollment, login and authentication of the user.

Scenario 1: Assess the First Part of the Authentication Model

This section is actually based on how the SSL/TLS protocol works. Given that the SSL/TLS protocol is a security protocol for the server authentication, this is a structure for the secure key transfer and mutual authentication during the key transfer.

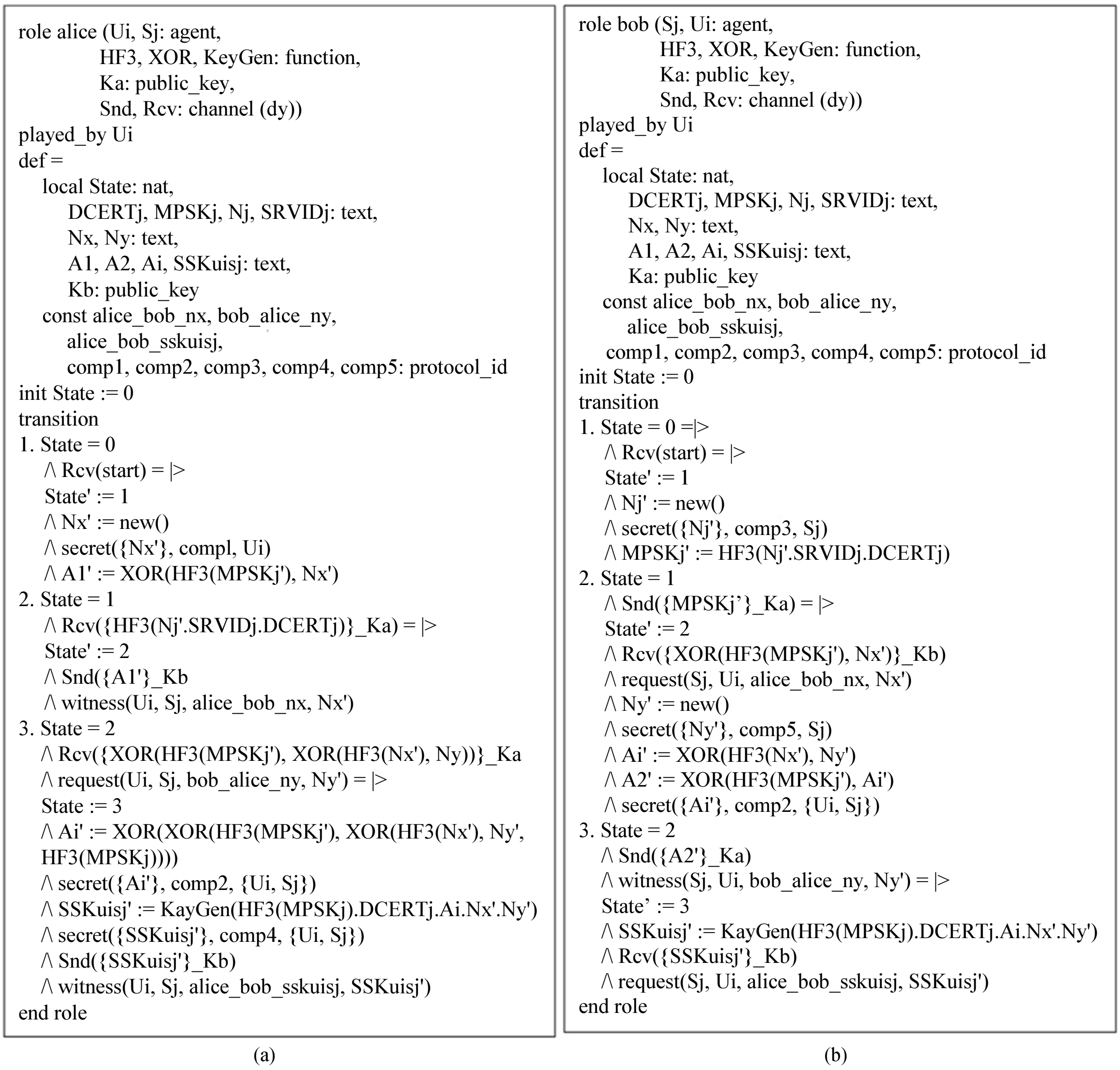
Also, another goal of this phase is, in fact, to secure the transfer of the share key of the S_j server, because this key plays an important role in the second phase and essentially the SSO capability is achieved using this key.

The SSL/TLS protocol by AVISPA is also considered as a secure protocol and its security has been proven according to the simulated codes of this protocol (AVISPA, 2014a) (Fig. 10a, 10b, 11a and 11b).

Generally, each model consists of five parts, the client and connections, the server and its connections, the definition of the connection channels, the intruder or attacker and, finally, the protocol objectives, of course WMF protocols have 6 parts because there is also an interface server.

Scenario 2: Assess the Second Part of the Authentication Model

Based on Figures 10 and 11, the first part of the authentication model is completely safe with respect to the simulation result in the software AVISPA. Now, the second part of the authentication model is assessed. Also based on the Fig. 12a, 12b, 13a and 13b, all parts of this authentication model based on hardware-software identifiers are completely secure. The security of this model was also proven by both the security assumptions and software AVISPA simulation. Therefore, this model or authentication protocol can be used to secure electronic learning connection. Of course, considering that the E-learning system is considered as one of the information systems, this protocol can be used to secure other similar information systems. This model could also be used for most web-based services, as it is fully compatible with the SSL/TLS protocol and moreover it provides high security for web-based services.



```
role alice (Ui, Sj: agent,
            HF3, XOR, KeyGen: function,
            Ka: public_key,
            Snd, Rcv: channel (dy))
played_by Ui
def =
   local State: nat,
       DCERTj, MPSKj, Nj, SRVIDj: text,
       Nx, Ny: text,
       A1, A2, Ai, SSKuisj: text,
       Kb: public_key
   const alice_bob_nx, bob_alice_ny,
       alice_bob_sskuisj,
       comp1, comp2, comp3, comp4, comp5: protocol_id
init State := 0
transition
1. State = 0
    /\ Rcv(start) = |>
    State' := 1
    /\ Nx' := new()
    /\ secret({Nx'}, compl, Ui)
    /\ A1' := XOR(HF3(MPSKj'), Nx')
2. State = 1
    /\ Rcv({HF3(Nj'.SRVIDj.DCERTj)}_Ka) = |>
    State' := 2
    /\ Snd({A1'}_Kb
    /\ witness(Ui, Sj, alice_bob_nx, Nx')
3. State = 2
   /\ Rcv({XOR(HF3(MPSKj'), XOR(HF3(Nx'), Ny))}_Ka
   /\ request(Ui, Sj, bob_alice_ny, Ny') = |>
   State := 3
   /\ Ai' := XOR(XOR(HF3(MPSKj'), XOR(HF3(Nx'), Ny',
   HF3(MPSKj))))
   /\ secret({Ai'}, comp2, {Ui, Sj})
   /\ SSKuisj' := KayGen(HF3(MPSKj).DCERTj.Ai.Nx'.Ny')
   /\ secret({SSKuisj'}, comp4, {Ui, Sj})
   /\ Snd({SSKuisj'}_Kb)
   /\ witness(Ui, Sj, alice_bob_sskuisj, SSKuisj')
end role
```

(a)

```
role bob (Sj, Ui: agent,
            HF3, XOR, KeyGen: function,
            Ka: public_key,
            Snd, Rcv: channel (dy))
played_by Ui
def =
   local State: nat,
       DCERTj, MPSKj, Nj, SRVIDj: text,
       Nx, Ny: text,
       A1, A2, Ai, SSKuisj: text,
       Ka: public_key
   const alice_bob_nx, bob_alice_ny,
       alice_bob_sskuisj,
    comp1, comp2, comp3, comp4, comp5: protocol_id
init State := 0
transition
1. State = 0 =|>
    /\ Rcv(start) = |>
    State' := 1
    /\ Nj' := new()
    /\ secret({Nj'}, comp3, Sj)
    /\ MPSKj' := HF3(Nj'.SRVIDj.DCERTj)
2. State = 1
    /\ Snd({MPSKj'}_Ka) = |>
    State' := 2
    /\ Rcv({XOR(HF3(MPSKj'), Nx')}_Kb)
    /\ request(Sj, Ui, alice_bob_nx, Nx')
    /\ Ny' := new()
    /\ secret({Ny'}, comp5, Sj)
    /\ Ai' := XOR(HF3(Nx'), Ny')
    /\ A2' := XOR(HF3(MPSKj'), Ai')
    /\ secret({Ai'}, comp2, {Ui, Sj})
3. State = 2
   /\ Snd({A2'}_Ka)
   /\ witness(Sj, Ui, bob_alice_ny, Ny') = |>
   State' := 3
   /\ SSKuisj' := KayGen(HF3(MPSKj).DCERTj.Ai.Nx'.Ny')
   /\ Rcv({SSKuisj'}_Kb)
   /\ request(Sj, Ui, alice_bob_sskuisj, SSKuisj')
end role
```

(b)

Fig. 10: Simulated Codes of alice and bob roles in the first part of the authentication model; (a) Simulated codes of the User (U_i), first part of the authentication model; (b) Simulated codes of the server (S_j), first part of the authentication model

```
role session (Ui, Sj: agent,
            Ka, Kb: public_key,
            HF3, XOR, KeyGen: function)
def =
   local SUI, SSJ, RUI, RSJ: channel (dy)
composition
            alice(Ui, Sj, HF3, XOR, KeyGen, Ka, SUI, RUI)
          /\ bob(Sj, Ui, HF3, XOR, KeyGen, Kb, SSJ, RSJ)
end role

role environment()
def =
     const ui, sj: agent,
             ka, kb, ki: public_key,
             hf3, xor, keygen: function,
             alice_bob_nx, bob_alice_ny,
             alice_bob_sskuisj,
             comp1, comp2, comp3, comp4, comp5: protocol_id
   intruder_knowledge = {ui, sj, ka, kb, ki, inv(ki), hf3, xor}
composition
         session(ui, sj, ka, kb, hf3, xor, keygen)
      /\ session(ui, i, ka, ki, hf3, xor, keygen)
      /\ session(i, sj, ki, kb, hf3, xor, keygen)
end role

goal
   secrecy_of comp1
   secrecy_of comp2
   secrecy_of comp3
   secrecy_of comp4
   secrecy_of comp5
   authentication_on alice_bob_nx
   authentication_on alice_bob_sskuisj
   authentication_on bob_alice_ny
end goal

environment()
```

(a)

```
% OFMC
% Version of 2006/02/13
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
   /home/span/span/testsuite/results/Auth_Model50.if
GOAL
   as_specified
BACKEND
   OFMC
COMMENTS
STATISTICS
   parseTime: 0.00s
   searchTime: 0.01s
   visitedNodes: 16 nodes
   depth: 4 plie
```

(b)

Fig. 11: Simulated Codes of session role and final avispa result in the first part of the authentication model; (a) Simulated codes of sessions, first part of the authentication model; (b) Final result in the first part of the authentication model

Assess the Authentication Model in Terms of Execution Times

In this section, the authentication model presented in this study is assessed from the perspective of execution time. This assessment is based on the number of functions and operations used in the model. According to Table 4, the types of parametric execution times are divided into 7 categories.

Also, according to Table 5, the approximate execution time of public key algorithms, digital signatures, secret keys and hash functions are presented in order to compare the algorithms relatively (Chuang and Chen, 2014).

According to the data presented in this section, in order to calculate the execution time of this model, in general the number of algorithms and hash functions are counted in all phases. According to Table 6 and 7, it can be said that the proposed model has a very good function. Given that the discussed model is used in remote scenarios, so the presence of exponential operations resulting from the public key encryption operations and the secret key encryption operations is not only justifiable, but also absolutely necessary. In all similar models, the authentication operations are performed on internal or intranet networks and/or in nearby scenarios such as ATMs. Thus, there is no need for the presence of the public and secret key encryption operations. Therefore, according to the obtained result, it can be said that the proposed model has quite a good performance in all phases of the authentication.

```
role alice (Ui, Sj: agent,
           SSKuisj: symmetric_key,
% HF3 is the SHA-3 one-way Hash Function
           HF3, XOR: function
           Snd, Rcv: channel (dy))
% User Ui Definition
played_by Ui
def =
  local State: nat,
       IDi, BIDi, Ni, Tr, MPSKj: text,
       CPUIDi, BBIDi, MMIDi, WSIDi, BAi: text,
       Nk, Nl: text
const alice_bob_nk, bob_alice_nl,
           alice_bob_ni, bob_alice_tr,
           bob_alice_mpskj,
           comp1, comp2, comp3, comp4, comp5, comp6, comp7: protocol_id
init State := 0
transition
% User Registration Phase
1. State = 0 /\ Rcv(start) =|>
   State' := 1 /\ Ni' := new()
% Sending Registration Request to Server Sj
         /\ secret({BIDi, Ni'}, comop2, Ui)
         /\ secret({IDi}, comp3, {Ui, Sj})
         /\ Snd({IDi.HF3(BIDi.Ni').HF3(XOR(IDi, Ni')).XOR(IDi,
         Ni')}_SSKuisj)
% Strong Authentication over Ni for Sj
         /\ witness(Ui, Sj, alice_bob_ni, Ni')
2. State = 1 /\ Rcv({XOR(HF3(HF3(IDi.MPSKj.Tr'.Ni)), HF3(BIDi.Ni')).
                XOR(HF3(MPSKj), HF3(XOR(IDi, Ni'))).
                XOR(MPSKj, HF3(IDi.MPSKj.Tr'.Ni'))}_SSKuisj)
                /\ request(Ui, Sj, bob_alice_tr, Tr')
                /\ request(Ui, Sj, bob_alice_mpskj, MPSKj') =|>
% Login and Authentication Phase
State' := 2 /\ secret ({Tr'}, comp1, Sj)
              /\ secret({MPSKj'}, comp7, Sj)
              /\ Nk’ := new()
              /\ secret({Nk'}, comp4, Ui)
% Sending Login Message {Qi, R1, R2, R3} to Server Sj
         /\ Snd({XOR(MPSKj, HF3(IDi.MPSKj.Tr'.Ni')).
            XOR(Nk', HF3(MPSKj)).
            XOR(HF3(HF3(HF3(IDi.MPSKj.Tr'.Ni')).Nk'), HF3(IDi)).
            HF3(HF3(IDi).HF3(HF3(IDi.MPSKj.Tr'.Ni')).Nk')}_SSKuisj)
         /\ witness(Ui,Sj, alice_bob_nk, Nk')
% Receiving {R4} Message from Server Sj
3. State = 2 /\ Rcv({XOR(Nl', HF3(HF3(IDi).Nk'))}_SSKuisj) =|>
    State' := 3 /\ Snd({HF3(BIDi.Ni')
                XOR(HF3(BIDi), HF3(XOR(IDi, Ni')), Nl').
                HF3(HF3(BIDi.Nk'.Nl'.HF3(BIDi.Ni)))}_SSKuisj)
              /\ request(Ui, Sj, bob_alice_nl, Nl')
              /\ secret({CPUIDi, BBIDi, MMIDi, WSIDi, BAi}, comp6, Ui)
              /\Snd({HF3(CPUIDi.BBIDi.MMIDi).HF3(WSIDi.BAi)}_SSKuisj)
end role
```

(a)

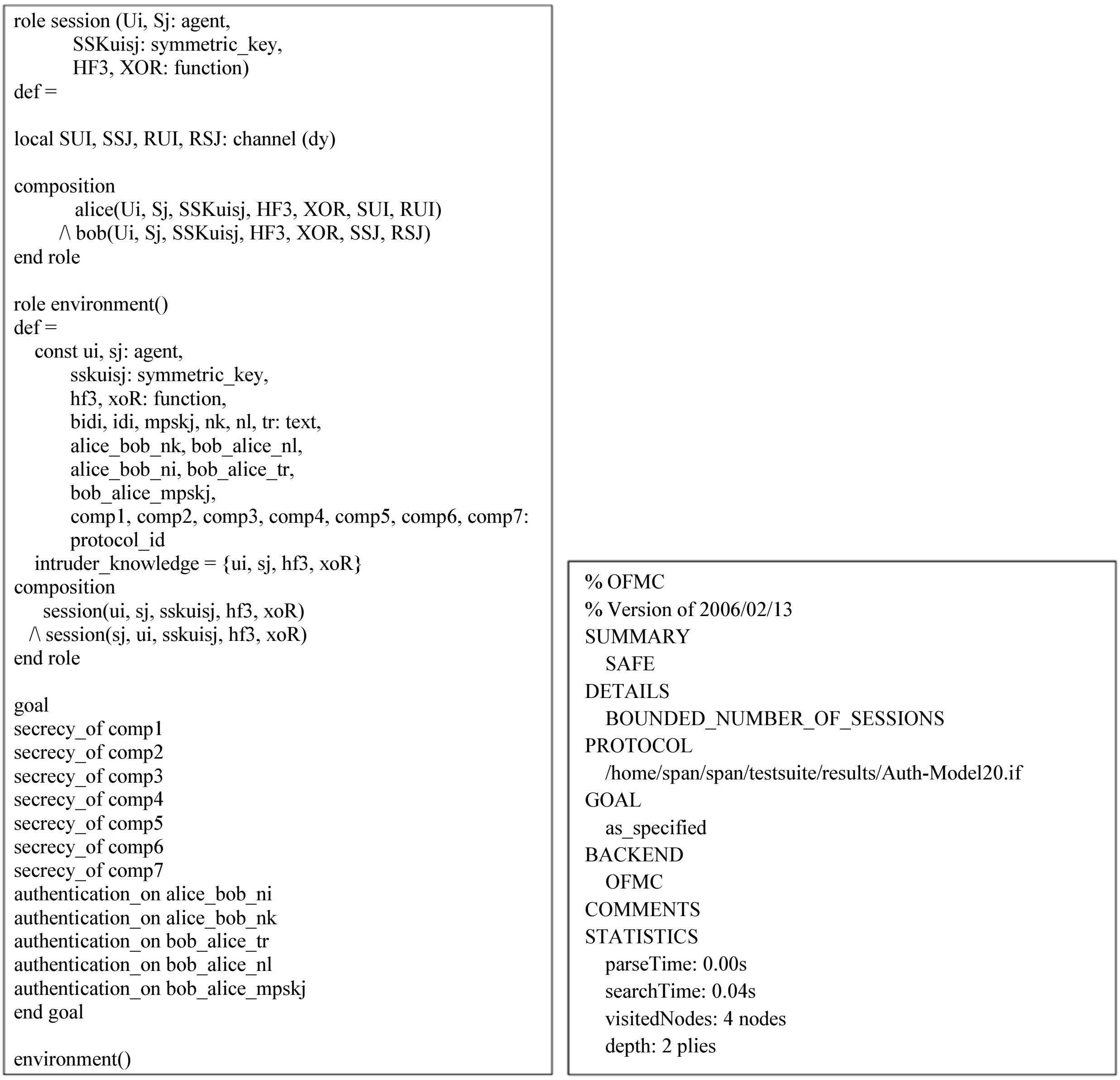
```
role bob (Sj, Ui: agent,
           SSKuisj: symmetric_key,
% HF3 is the SHA-3 one-way Hash Function
           HF3, XOR: function
           Snd, Rcv: channel (dy))
% Server Sj Definition
played_by Sj
def =
  local State: nat,
       IDi, BIDi, Ni, Tr, MPSKj: text,
       CPUIDi, BBIDi, MMIDi, WSIDi, BAi: text,
       Nk, Nl: text
const alice_bob_nk, bob_alice_nl,
           alice_bob_ni, bob_alice_tr,
           bob_alice_mpskj,
           comp1, comp2, comp3, comp4, comp5, comp6, comp7: protocol_id
init State := 0
transition
% User Registration Phase
1. State = 0 /\ Rcv({IDi.HF3(BIDi.Ni).HF3(XOR(IDi, Ni')).XOR(IDi,
Ni)}_SSKuisj
% Sj Request for Ni Value that Ui must have verify that
State' := 1 /\ request(Sj, Ui, alice_bob_ni, Ni')
         /\ Tr' := new()
         /\ MPSKj' := new()
         /\ secret({Tr'}, comp1, Sj)
         /\ secret({MPSKj'}, comp7, Sj)
         /\ secret({BIDi, Ni'}, comp2, Ui)
         /\ secret({IDi}, comp3,{Ui, Sj})
% Sending Registration Acknowledgement to User Ui
         /\ Snd({XOR(HF3(HF3(IDi.MPSKj.Tr'.Ni')), HF3(BIDi.Ni')).
            XOR(HF3(MPSKj), HF3(XOR(IDi. Ni')).
            XOR(MPSKj, HF3(IDi.MSKPj.Tr'.Ni'))}_SSKuisj)
% Server Sj Generate Tr Registration Time for User Ui
         /\ witness(Sj, Ui, bob_alice_tr, Tr')
         /\witness(Sj, Ui, bob_alice_mpskj, MPSKj')
% Login and Authentication Phase
% Receiving Login Request Message {Qi, R1, R2, R3} from User Ui
2. State = 1 /\ Rcv({XOR(MPSKj, HF3(IDi.MPSKj.Tr'.Ni')).
           XOR(Nk', HF3(MPSKj)).
           XOR(HF3(HF3(HF3(IDi.MPSKj.Tr'.Ni')).Nk'), HF3(IDi)).
           HF3(HF3(IDi).HF3(HF3(IDi.MPSKj.Tr'.Ni')).Nk')}_SSKuisj) =|>
State' := 2 /\ Nl' := new()
             /\ secret({Nl'}, comp5, Sj)
% Sending {R4} Message to User Ui
         /\ Snd({XOR(Nl', HF3(HF3(IDi).Nk'))}_SSKuisj)
         /\ witness(Sj, Ui, bob_alice_nl, Nl')
3. State = 2 /\ Rcv({HF3(BIDi.Ni').
               XOR(HF3(BIDi), HF3(XOR(IDi, Ni')), Nl').
               HF3(HF3(BIDi.Nk'.Nl'.HF3(BIDi.Ni)))}_SSKuisj) =|>
State' := 3/\ request(Sj, Ui, alice_bob_nk, Nl')
             /\ secret({CPUIDi, BBIDi, MMIDi, WSIDi, BAi}, comp6, Ui)
             /\Rcv({HF3(CPUIDi.BBIDi.MMIDi).HF3(WSIDi.BAi)}_SSKuisj)
end role
```

(b)

Fig. 12: Simulated Codes of alice and bob roles in the second part of the authentication model; (a) Simulated codes of user (U_i), second part of the authentication model; (b) Simulated codes of the server (S_j), second part of the authentication model

Table 4: Symbols used in evaluation of parametric execution time

Symbol	Description
T_{EXP}	Time Complexity for Execution of Exponential Operations
T_{F}	Time Complexity for Execution of Production/Reproduction fuzzy Extractor Algorithm
T_{EM}	Time Complexity for Execution of Scalar Point Multiplication Operation in Elliptic Curve
T_{h}'	Time Complexity for one way Transformation and one way Secure hash function *h*′(.)
T_{SYM}	Time Complexity for Execution of Symmetric Encryption/Decryption Algorithm
T_{h}	Time Complexity for Execution of one way hash function *h*(.)
–	Without Parametric Execution time



```
role session (Ui, Sj: agent,
        SSKuisj: symmetric_key,
        HF3, XOR: function)
def =

local SUI, SSJ, RUI, RSJ: channel (dy)

composition
         alice(Ui, Sj, SSKuisj, HF3, XOR, SUI, RUI)
      /\ bob(Ui, Sj, SSKuisj, HF3, XOR, SSJ, RSJ)
end role

role environment()
def =
   const ui, sj: agent,
         sskuisj: symmetric_key,
         hf3, xoR: function,
         bidi, idi, mpskj, nk, nl, tr: text,
         alice_bob_nk, bob_alice_nl,
         alice_bob_ni, bob_alice_tr,
         bob_alice_mpskj,
         comp1, comp2, comp3, comp4, comp5, comp6, comp7:
         protocol_id
   intruder_knowledge = {ui, sj, hf3, xoR}
composition
     session(ui, sj, sskuisj, hf3, xoR)
  /\ session(sj, ui, sskuisj, hf3, xoR)
end role

goal
secrecy_of comp1
secrecy_of comp2
secrecy_of comp3
secrecy_of comp4
secrecy_of comp5
secrecy_of comp6
secrecy_of comp7
authentication_on alice_bob_ni
authentication_on alice_bob_nk
authentication_on bob_alice_tr
authentication_on bob_alice_nl
authentication_on bob_alice_mpskj
end goal

environment()
```

(a)

```
% OFMC
% Version of 2006/02/13
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
   /home/span/span/testsuite/results/Auth-Model20.if
GOAL
   as_specified
BACKEND
   OFMC
COMMENTS
STATISTICS
   parseTime: 0.00s
   searchTime: 0.04s
   visitedNodes: 4 nodes
   depth: 2 plies
```

(b)

Fig 13: Simulated Codes of session role and final avispa result in the second part of the authentication model; (a) Simulated codes of sessions, second part of the authentication model; (b) Final result in the second part of the authentication model

It should be noted that 6 exponential execution times and 8 execution times of the hash function in the server authentication phase are related to the SSL/TLS protocol and is not a part of the presented authentication model. But given that it has been used in this model, they have also been mentioned. Another point is that the execution time is not considered for the server enrollment phase, as there are different methods and protocols for this purpose, as well as this phase is rarely done and its execution time is not considered according to the interval of doing it, which is usually each year or every 2 years.

Compare the Proposed Model with Other Models in Terms of the Security and Functional Capabilities

In this section, we compare the model presented in this study with other models in terms of security and function capabilities. According to Table 8, 16 security and functional features have been presented to compare the proposed model with other models having similar function. Given that most of the reviewed models support all security characteristics and capabilities such as: Support for multi-server environments, biometric capabilities, independence from the Verification Table and etc., but as the investigations show, there are structural and security weaknesses. Therefore, the proposed model is completely superior in terms of security and functional features compared to similar models.

Table 5: Execution time for security algorithms with considering of (Chuang and Chen, 2014)

Operation	Microsecond/Operation
RSA-1024 Encryption	3010.00
RSA-1024 Decryption	130.00
RSA-1024 Signature	3020.00
RSA-1024 Verification	130.00
AES-256 Encryption	0.80
AES-256 Decryption	0.80
SHA-1	0.50
SHA-512	0.76
SHA3-256	1.28
SHA3-512	2.28

Table 6: Parametric execution times used in Proposed Authentication Model

Phase	Parametric execution time
Server Registration	–
Server Authentication	$8T_{\text{h}}+6T_{\text{EXP}}+4T_{\text{EXP}}+9T_{\text{h}}$
User Registration	$2T_{\text{SYM}}+5T_{\text{h}}$
User Login and Authentication	$5T_{\text{SYM}}+15T_{\text{h}}$
Total with SSL/TLS	$10T_{\text{EXP}}+7T_{\text{SYM}}+37T_{\text{h}}$
Total without SSL/TLS	$4T_{\text{EXP}}+7T_{\text{SYM}}+29T_{\text{h}}$

Table 7: Execution times considering Table 5 and Table 6

Phase	Execution time (Microsecond)
Server Registration	–
Server Authentication	30138.7
User Registration	13
User Login and Authentication	38.2
Total with SSL/TLS	30189.9
Total without SSL/TLS	12111.7

T_{h}: SHA3-512, T_{SYM}: AES-256, T_{EXP}: RSA-1024 Encryption

Table 8: Comparison of Security and Functional Features with other Similar Models

Security and functional feature	Kim *et al.* (2012)	Yoon and Yoo (2013)	Chuang and Chen (2014)	Mishra *et al.* (2014)	Amin *et al.* (2015)	Our Model
SFF1	×	×	✓	✓	✓	✓
SFF2	✓	×	✓	✓	✓	✓
SFF3	✓	✓	✓	✓	×	✓
SFF4	×	×	×	×	×	✓
SFF5	✓	✓	×	✓	✓	✓
SFF6	✓	✓	✓	✓	✓	✓
SFF7	✓	×	×	✓	✓	✓
SFF8	✓	✓	×	✓	✓	✓
SFF9	✓	✓	×	✓	✓	✓
SFF10	✓	✓	✓	✓	✓	✓
SFF11	✓	✓	✓	✓	✓	✓
SFF12	×	✓	✓	✓	×	✓
SFF13	✓	✓	✓	✓	✓	✓
SFF14	✓	✓	✓	✓	✓	✓
SFF15	×	×	×	×	×	✓
SFF16	×	×	×	×	×	✓

SFF1: User anonymity, SFF2: Insider attack, SFF3: Password guessing attack, SFF4: Maintaining performance and security on the internet and web services, SFF5: Denial of service attack, SFF6: Known session key attack, SFF7: User impersonation attack, SFF8: Server spoofing attack, SFF9: MitM attack, SFF10: Replay attack, SFF11: Mutual authentication,SFF12: Efficient SSO capability, SFF13: Session key agreement and verification, SFF14: Using randomly generated and onetime use values, SFF15: User and computer isolation based on hardware-software identifiers, SFF16: Secure passwordless scheme

Table 9: Security and functional features improvements in percent for proposed model and other models

Models	Security (%)	Number of security functional features	Improvement of Proposed method (%)
Kim *et al.* (2012)	68 75	11.0	31.25
Yoon and Yoo (2013)	62 50	10.0	37.50
Chuang and Chen (2014)	56 25	9.0	43.75
Mishra *et al.* (2014)	81 25	13.0	18.75
Amin *et al.* (2015)	68 75	11.0	31.25
Proposed Model	100	16.0	–
Average of Five Methods	67 50	10.8	32.5

Table 10: Comparison of computational costs with other similar models

Phase	Kim *et al.* (2012)	Yoon and Yoo (2013)	Chuang and Chen (2014)	Mishra *et al.* (2014)	Amin *et al.* (2015)	Our Model
SR	$2T_h+T_h'$	T_h	$2T_h$	$3T_h$	–	–
SA	$5T_h+2T_{EM}$	$5T_h+2T_{EM}$	$8T_h$	$7T_h$	–	$4T_{EXP}+9T_h$
UR	T_h	T_h	T_h	$4T_h$	$T_{EM}+T_h+T_{SYM}$	$2T_{SYM}+5T_h$
ULA	$6T_h+T_h' +2T_{EM}$	$5T_h+2T_{EM}$	$8T_h$	$10T_h$	$11T_{EM}+17T_h+4T_{SYM}$	$5T_{SYM}+15T_h$
PC	$2T_h0 +2T_h$	$2T_h$	$2T_h$	$5T_h$	–	–
RCA	$7T_h$	$7T_h$	–	–	–	–
Total	$4T_{EM}+4T_h0 +23T_h$	$4T_{EM}+21T_h$	$21T_h$	$29T_h$	$12T_{EM}+18T_h+5T_{SYM}$	$4T_{EXP}+7T_{SYM}+29T_h$

SR: Server Registration Phase, SA: Server Authentication Phase, UR: User Registration Phase, ULA: User Login and Authentication Phase, PC: Password Change Phase, RCA: Registration Center Authentication Phase

Table 11: Comparison of execution times with considering (Zivi *et al*., 2017a), (Chuang and Chen, 2014) and (Chaudhry *et al*., 2016) for proposed model and other methods

Models	Execution times (Microsecond)	Improvement of proposed method (%)
Kim *et al.* (2012)	20237.0	67.09
Yoon and Yoo (2013)	20237.0	67.09
Chuang and Chen (2014)	11357.9	−6.22
Mishra *et al.* (2014)	11376.1	−6.07
Amin *et al.* (2015)	35853.5	196.02
Proposed Model	12111.7	–
Average improvement in comparison with all models	19812.3	63.58

In Kim *et al*., Yoon and Yoo, Chuang and Chen, Mishra *et al.* and Amin *et al.* models, five IKEv2 operations for acquiring the same result is considered. This five IKEv2 operation is for client side configuration: (1) request from the server for client side configuration, (2) respond from the user for applying this configuration, (3) sending configuration script or algorithm for receiving validation message, (4) sending system-level execution permission and (5) sending configuration result. Also the IKEv2 task processing time choose from our previous research. This value is average of task processing time in 10%, 30% and 50% of background traffic in 3 states of evaluation (Zivi *et al*., 2017a)

Also Table 9 shows the Security and functional features improvements in percent for proposed model in comparison to other models. To calculate the security percentage of each method, (second column of Table 9), considering 16 security components with a security percentage of 100 (proposed method), with assuming that the security components have an equal importance (Table 8), each security component has a 6.25% impact on overall security. Therefore, based on the third column of Table 9, to calculate the security percentage of each method, the number of security components is multiplied by 6.25% and the second column of Table 9 is created.

Compare the Model Presented with Other Models in Terms of the Execution Time

In this section, we compare the model discussed with other models from the perspective of execution times. Given that the discussed model consists of 4 parts of the server enrollment, server authentication, user enrollment, user login and authentication of the user, so in each section, the numbers of operations performed with other models were compared.

According to Table 10 and 11, the discussed model has a desired and acceptable function. As stated above, the model presented in this study has been designed to secure the E-learning system connection; given that the connection of an E-learning system is done through the Internet connection platform, so not only the presence of the public and secret key encryption operations is justified, but also it is even necessary. It is also worth mentioning that the models of (Kim *et al*., 2012), (Yoon and Yoo, 2013), (Chuang and Chen, 2014) and (Amin *et al*., 2015) had security and structural weaknesses and the model of (Mishra *et al*., 2014) also does not support the user isolation capabilities and Passwordless Scheme. All similar models have been designed to secure the connection of internal networks such as intranets and nearby connection scenarios such as ATMs and naturally do not require the public and secret key encryption operations. Therefore, the performance of the model under consideration is at an acceptable level according to the conducted investigations.

Conclusion

There are a few points to make when using this model. These points are completely related to the model implementation and do not relate to the theoretical issues of this authentication model:

1. Due to the widespread adoption of the TLS 1.2 protocol, there are few scenarios that may still use

SSL version 3.0 or earlier for the connections' security. However, if you use SSL version 3.0, it should be noted that this version has security vulnerabilities. Of course, by using the solution provided by (Joshi *et al*., 2009), the vulnerability can be greatly eliminated

2. Also, when setting up a SSL/TLS protocol service, it is best to consider the following points to ensure the best results. When a CA server issues a certificate based on the DNS structure, there is a series of rules in relation to the issued certificate (Hodges, 2011):

    - The certificate should include DNS-ID for interoperability
    - If a service using a certificate has expanded the technology used for related applications, then the certificate should contain a field called the SRV-ID
    - If a service using a certificate has expanded the technology used for related applications, then the certificate should have a field called URI-ID
    - The certificate may contain applications whose type has been defined before the server name is published, or it may be that the application has a type in which related URI does not exist. Under such conditions, this falls outside the scope of the certificate debate

As its clear, all of the reviewed models such as: (Kim *et al*., 2012), (Yoon and Yoo, 2013), (Chuang and Chen, 2014) and (Amin *et al*., 2015) and (Mishra *et al*., 2014), has security flaws and performance issues that shown in Table 8-11. The authentication model presented in this research provides complete security for an E-learning and similar Information systems or many web services. The function of this model is also very acceptable. But the point is that there are some things to be improved.

Also based on Tables 9 and 11, the proposed model has 32.50% improvement in security and 63.58% improvement in execution time averagely in comparison with five newest methods. Therefore the proposed model completely satisfies all security and performance requirements of E-learning and all information systems and web services.

In general, the authentication model presented in this study has a complete security and very good function in the context of the safety of connections of the E-learning system, which if implemented, in addition to the integrated security of the system; the function will be maintained at an acceptable level. There is an important thing about performance and security of the Authentication model and that is to maintain the balance between these two very important characteristics obtained in this model using optimized and efficient structures and algorithms.

Acknowledgment

The authors of this article acknowledge that all contents of this research paper has been produced and developed by them. This article is a partial conclusion of main research studies of the authors and they are the principal developer and financial supplier for the research.

Author's Contributions

Afshin Zivi: The main responsible author for literature review, Methodology, Analytical results and conclusion sections. Contributed in preparation and writing.

Gholamreza Farahani: The author responsible for review, preparation and improvements, Contributed in preparation, organization and supervision.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Amin, R., S. Islam, G. Biswas, M. Khan and N. Kumar, 2015. An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography. J. Med. Syst., 39: 180-180. DOI: 10.1007/s10916-015-0351-y

Armando, A., D. Basin, Y. Boichut, Y. Chevalier and L. Compagna *et al*., 2005. The AVISPA tool for the automated validation of internet security protocols and applications. Proceedings of the 17th international conference on Computer Aided Verifications, Jul. 06-10, Edinburgh, Scotland, UK, pp: 281-285. DOI: 10.1007/11513988_27

Attwell, G., 2006. Evaluating E-learning: A guide to the evaluation of e-learning. Evaluate Eur. Handbook Series, 2: 46-46.

AVISPA, 2014a. Automated validation of internet security protocols and applications, d6-2.

AVISPA, 2014b. Automated validation of internet security protocols and applications, HLPSL tutorial.

AVISPA, 2014c. Avispa linux and mac os tool.

AVISPA, 2014d. Avispa web tool.

Bentley, Y., H. Selassie and A. Shegunshi, 2012. Design and evaluation of student-focused elearning. Electr. J. e-Learn., 10: 1-12.

Bertoni, G., J. Daemen, M. Peeters and G. Assche, 2015. Sha-3 standard: Permutation-based hash and extendable-output functions, NIST FIPS 202. National Institute of Standards and Technology.

Chansuc, S. and P. Praneetpolgrang, 2008. An empirical study on the effect of organizational culture on the acceptance of elearning in thai higher education. Proceedings of the 5th International Conference on eLearning for Knowledge Based Society, Dec. 11-12, Bangkok, Thailand, pp: 22.1-22.6.

Chatterjee, S., A.K. Das and J. Sing, 2014. An enhanced access control scheme in wireless sensor networks. Adhoc Sensor Wireless Netw., 21: 121-149.

Chaudhry, S., M. Khan, M. Khan and T. Shon, 2016. A multiserver biometric authentication scheme for tmis using elliptic curve cryptography. J. Med. Syst., 40: 230-230. DOI: 10.1007/s10916-016-0592-4

Chen, T., Y. Chen and W. Shih, 2010. An advanced ECC id-based remote mutual authentication scheme for mobile devices. Proceedings of the 7th International Conference on Autonomic and Trusted Computing, Oct. 26-29, IEEE Xplore Press, Xian, Shaanxi, China, pp: 116-120. DOI: 10.1109/UIC-ATC.2010.18

Chen, T., H. Yeh and W. Shih, 2011. An advanced ECC dynamic id-based remote mutual authentication scheme for cloud computing. Proceedings of the 5th FTRA International Conference on Multimedia and Ubiquitous Engineering, Jun. 28-30, IEEE Xplore Press, Loutraki, Greece, pp: 155-159. DOI: 10.1109/MUE.2011.69

Chuang, M. and M. Chen, 2014. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Syst. Applic., 41: 1411-1418. DOI: 10.1016/j.eswa.2013.08.040

Chuang, Y. and Y. Tseng, 2012. Towards generalized id-based user authentication for mobile multi-server environment. Int. J. Commun. Syst., 25: 447-460. DOI: 10.1002/dac.1268

Das, A., 2011. Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. Institut. Eng. Technol., 5: 145-151.

Das, A., A. Massand and S. Patil, 2013. A novel proxy signature scheme based on user hierarchical access control policy. J. King Saudi Univ. Comput. Inform. Sci., 25: 219-228. DOI: 10.1016/j.jksuci.2012.12.001

Dharmawansa, A.D., K. Nakahira and Y. Fukumura, 2013. Detecting eye blinking of a real-world student and introduction to the virtual e-learning environment. Proc. Comput. Sci., 22: 717-726. DOI: 10.1016/j.procs.2013.09.153

Dolev, D. and A. Yao, 1983. On the security of public key protocols. IEEE Trans. Inform. Theory, 29: 198-208. DOI: 10.1109/TIT.1983.1056650

El-Khatib, K., L. Kobra, Y. Xu and G. Yee, 2003. Privacy and security in e-learning. Int. J. Distance Educ., 1: 1-15. DOI: 10.4018/jdet.2003100101

He, D., 2011. Security flaws in a biometrics-based multiserver authentication with key agreement scheme. IACR Cryptography ePrint Archive.

He, D., J. Chen, W. Shi and M. Khan, 2013. On the security of an authentication scheme for multiserver architecture. Int. J. Electronic Security Digital Forens., 5: 288-296. DOI: 10.1504/IJESDF.2013.058669

He, D. and S. Wu, 2013. Security flaws in a smart card based authentication scheme for multi-server environment. Wireless Personal Commun., 70: 323-329. DOI: 10.1007/s11277-012-0696-1

Hodges, J., 2011. Representation and verification of domain-based application service identity within internet Public Key Infrastructure using x.509 (PKIX) certificates in the context of transport layer security. Internet Engineering Task Force (IETF), Request for Comments.

Jeon, I., H. Kim and M. Kim, 2011. Enhanced biometrics-based remote user authentication scheme using smart cards. J. Security Eng., 8: 237-254.

Joshi, Y., D. Das and S. Saha, 2009. Mitigating man in the middle attack over secure sockets layer. Proceedings of the IEEE International Conference on Internet Multimedia Services Architecture and Applications, Dec. 9-11, IEEE Xplore Press, Bangalore, India, pp: 1-5. DOI: 10.1109/IMSAA.2009.5439461

Karforma, S. and B. Ghosh, 2009. On security issues in e-learning system. Institute of Technology, Burdwan University.

Kaufman, C., 2005. Internet Key Exchange (IKEV2) protocol. Internet Engineering Task Force (IETF), Request for Comments.

Khedr, A., 2012. Towards three dimensional analyses for applying e-learning evaluation model: The case of e-learning in Helwan university. Int. J. Comput. Sci., 9: 161-166.

Kim, H., W. Jeon, K. Lee, Y. Lee and D. Won, 2012. Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme. Proceedings of the 12th International Conference on Computational Science and Its Applications, Jun. 18-21, Springer, Salvador de Bahia, Brazil, pp: 391-406. DOI: 10.1007/978-3-642-31137-6_30

Lee, C. and C.W. Hsu, 2013. A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Nonlinear Dynam., 71: 201-211. DOI: 10.1007/s11071-012-0652-3

Lee, C., Y. Lai and C. Li, 2012. An improved secure dynamic id based remote user authentication scheme for multi-server environment. Int. J. Security Applic., 6: 203-210.

Lee, C., T. Lin and R. Chang, 2011. A secure dynamic id based remote user authentication scheme for multiserver environment using smart cards. Expert Syst. Applic., 38: 13863-13870.

Li, C. and M. Hwang, 2010. An efficient biometricsbased remote user authentication scheme using smart cards. J. Netw. Comput. Applic., 33: 1-5. DOI: 10.1016/j.jnca.2009.08.001

Li, X., Y. Xiong, J. Ma and W. Wang, 2012. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. J. Netw. Comput. Applic., 35: 763-769. DOI: 10.1016/j.jnca.2011.11.009

Li, C., C. Lee, C. Weng and C. Fan, 2013. An extended multi-server-based user authentication and key agreement scheme with user anonymity. KSII Trans. Internet Inform. Syst., 7: 119-131. DOI: 10.3837/tiis.2013.01.008

Mishra, D., A. Das and S. Mukhopadhyay, 2014. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst. Applic., 41: 8129-8143. DOI: 10.1016/j.eswa.2014.07.004

Mohd Alwi, N., 2010. E-learning and information security management. Int. J. Digital Society, 1: 148-156. DOI: 10.20533/ijds.2040.2570.2010.0019

Mustafa, Y. and S. Sharif, 2011. An approach to Adaptive E-learning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and evaluation. Int. J. Library Inform. Sci., 3: 15-28.

NIST, 2016. NIST cryptographic algorithm validation program (cavp).

Pippal, R., C. Jaidhar and S. Tapaswi, 2013. Robust smart card authentication scheme for multi-server architecture. Wireless Personal Commun., 72: 729-745. DOI: 10.1007/s11277-013-1039-6

Richardson, C. and K. Swan, 2003. Examining social presence in online courses in relation to students, perceived learning and satisfaction. J. Asynchronous Learn. Netw., 7: 68-84.

Ristic, I., 2015. Bulletproof SSL and TLS. Fiesty Duck.

Sarkar, P., 2010. A simple and generic construction of authenticated encryption with associated data. ACM Trans. Inform. Syst. Security, 13: 1-16. DOI: 10.1145/1880022.1880027

Song, R., 2010. Advanced smart card based password authentication protocol. Comput. Standards Interfaces, 32: 321-325. DOI: 10.1016/j.csi.2010.03.008

Sood, S., A. Sarje and K. Singh, 2011. A secure dynamic identity based authentication protocol for multiserver architecture. J. Netw. Comput. Applic., 34: 609-618. DOI: 10.1016/j.jnca.2010.11.011

Stallings, W., 2005. Cryptography and Network Security: Principles and Practice. 1st Edn., Prentice Hall, ISBN-10: 0131873164, pp: 680.

Stinson, D., 2006. Some observations on the theory of cryptographic hash functions. Designs Codes Cryptography, 38: 259-277. DOI: 10.1007/s10623-005-6344-y

Swan, K., P. Shea, E. Fredericksen, A. Pickett and W. Pelz *et al*., 2000. Building knowledge building communities: Consistency, contact and communication in the virtual classroom. J. Educ. Comput. Res., 23: 359-383. DOI: 10.2190/W4G6-HY52-57P1-PPNE

Truong, T., M. Tran and A.D. Duong, 2013. Robust secure dynamic id based remote user authentication scheme for multi-server environment. Proceedings of the International Conference on Computational Science and its Applications, (CSA' 13), Springer Berlin Heidelberg, pp: 502-515. DOI: 10.1007/978-3-642-39640-3_37

Tsaur, W., J. Li and W. Lee, 2012. An efficient and secure multi-server authentication scheme with key agreement. J. Syst. Software, 85: 876-882. DOI: 10.1016/j.jss.2011.10.049

Wang, B. and M. Ma, 2013. A smart card based efficient and secured multi-server authentication scheme. Wireless Personal Commun., 68: 361-378. DOI: 10.1007/s11277-011-0456-7

Yang, D. and B. Yang, 2010. A biometric password-based multi-server authentication scheme with smart card. International Conference on Computer Design and Applications, Jun. 25-27, IEEE Xplore Press, Qinhuangdao, China, pp: 554-559. DOI: 10.1109/ICCDA.2010.5541128

Yang, J. and P. Lin, 2014. An ID-based user authentication scheme for cloud computing. Proceedings of the IEEE 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Aug. 27-29, IEEE Xplore Press, Kitakyushu, Japan, pp: 98-101. DOI: 10.1109/IIH-MSP.2014.31

Yeh, K., N. Lo and Y. Li, 2011. Cryptanalysis of Hsiang-Shih's authentication scheme for multi-server architecture. Int. J. Commun. Syst., 24: 829-836. DOI: 10.1002/dac.1184

Yoon, E. and K. Yoo, 2011. Cryptanalysis of simple three-party password-based key exchange protocol. Int. J. Commun. Syst., 24: 532-542. DOI: 10.1002/dac.1168

Yoon, E. and K. Yoo, 2013. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. J. Supercomput., 63: 235-255. DOI: 10.1007/s11227-010-0512-1

Zivi, A., G. Farahani and K. Manochehri, 2017a. The role of SSL/TLS, IPSEC and ikev2 protocols in security of e-learning system's communications, a study and simulation approach. Int. J. Comput. Trends Technol., 50: 20-33. DOI: 10.14445/22312803/IJCTT-V50P105

Zivi, A., S. Rezaeian and N. Shahhoseini, 2017b. Analyze e-learning system of Islamic Azad University in perspectives of optimality, privacy and data protection; review, analysis of statistical data and managerial model. Int. J. Comput. Inform. Technol., 5: 17-31.