نموذج رقم (1)

إقــــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Implementing and comprising of OTP Techniquss (TOTP, HOTP,CRC
to prevent Replay Attack in RADIUS Protocol.

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name: آمنه سلامه محمد ابوجشية اسم الطالب:

Signature: امن التوقيع:

Date: 17-8-2014م التاريخ:

Islamic University of Gaza
Deanery of Higher Studies
Faculty of Information
Technology
Information Technology Program

Implementing and Comprising of OTP Techniques (TOTP,HOTP,CROTP) to Prevent Replay Attack in RADIUS Protocol

Prepared by
Amna S.M Abukeshipa

Supervised by
Dr. Tawfiq SM Barhoom

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology

2014

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا **هاتف داخلي:** 1150

ج س غ/35/
الرقم 2014/06/10م.......... Ref
التاريخ .... Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحثة/ **آمنة سلامة محمد أبوخشيبة** لنيل درجة الماجستير في **كليـة *تكنولوجيـا المعلومـات*** **برنامج تكنولوجيا المعلومات** وموضوعها:

برمجة ومقارنة لتقنيات كلمة المرور لمرة واحدة (TOTP,HOTP, CROTP)
لحماية هجمة الإعادة في بروتوكول RADIUS

Implementing and Comprising of OTP Techniques (TOTP,HOTP, CROTP)
to Prevent Replay Attack in RADIUS Protocol.

وبعد المناقشة العلنية التي تمت اليوم الثلاثاء 12 شعبان 1435هـ، الموافق 2014/06/10م الساعة العاشرة صباحاً بقاعة اجتماعات مبنى اللحيدان ، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. توفيق سليمان برهوم	**مشرفاً ورئيساً**	....
د. إيــاد محمــد الأغــا	**مناقشاً داخلياً**	....
د. سناء وفــا الصايــغ	**مناقشاً خارجيًا**	....

وبعد المداولة أوصت اللجنة بمنح الباحثة درجة الماجستير في **كلية *تكنولوجيا المعلومات*/ برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحها هذه الدرجة فإنها توصيها بتقوى الله ولزوم طاعته وأن تسخر علمها في خدمة دينها ووطنها.

والله ولي التوفيق ،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

١٧/٨/٢٠١٤

أ.د. فؤاد علي العــاجز

شئون البحث العلمي والدراسات العليا
الجامعة الإسلامية - غزة
The Islamic University
Research & Graduate Affairs

بسم الله الرحمن الرحيم

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ
درَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ﴾

سورة المجادلة: الآية 11

Dedication

To My Loving Parents ,,,

To My Dear husband ,,,

To My Loving Children ,,,

To My Dear Brothers ,,,

To My Sincere Sisters,,,

To Faithful My Friends ,,,

Acknowledgment

Thanks to God, the First and the Last. Many thanks and sincere gratefulness goes to my supervisor Dr Tawfiq SM Barhoom . I am indebted to his support, supervision and guidance. I would like to thank my father, mother, dear husband , brothers and sisters for their support which gave me the strength and enthusiasm during the whole period of this thesis.

I am also grateful to Dr. Mayada Mgari , for his help in this thesis , deeply appreciate it. I would like to extend sincere gratitude and appreciation to the people who helped me to accomplish this thesis.

I would like to thank all my academic teachers, staff members and colleagues for their support throughout my study in Islamic university .

الملخص

أمن الشبكة هي مسألة في غاية الأهمية للمنظمات من أجل حماية البيانات الحساسة الخاصة بهم من المهاجمين . عدد من الباحثين وفرت عدد من البروتوكولات الامنية المعتمدة لتعزيز الخصوصية والسرية عبر الشبكات. RADIUS هي واحدة من البروتوكولات الأكثر شعبية المستخدمة في شبكة الاتصالات لمصادقة المستخدم . لسوء الحظ، هناك العديد من نقاط الضعف التي تواجه بروتوكول RADIUS واحدة من نقاط الضعف هذه هي مشكلة إعادة الهجوم وهي تحتاج إلى الى تمنع. قدمت البروتوكولات السابقة عدد من التقنيات للحد من آثار هجوم الاعادة في بروتوكول RADIUS. التقنية OTP هي واحدة من أهم التقنيات التي تستخدم لتعزيز أمن مصادقة المستخدم في بيئات عديدة ولسد الفجوة المحتملة في أمن الشبكات .مع العديد من التقنيات OTP , مساهمتنا هي اختيار ثلاثة تقنيات وهي كلمة المرور لمرة واحدة باستخدام الوقت (TOTP)، كلمة المرور لمرة واحدة باستخدام التجزئة (HOTP) كلمة المرور لمرة واحدة باستخدام التحدي والاستجابة (CROTP) هذا يدفعنا لتقديم ELSBOT نظام التعلم الإلكتروني القائم على كلمة المرور لمرة واحدة لمنع تكرار الهجمات في بروتوكول RADIUS . تقدم هذه الرسالة مقارنة بين هذه التقنيات هذه المقارنة اعتمدت على مجموعة من العوامل مثل منع هجوم الاعادة، ، سرعة الخوارزمية، وقت استجابة الخادم .بعد قياس هذه العوامل من خلال ELSBOT لدينا، فقد بينت النتائج أن التقنيات في OTP قد منعت تكرار الهجوم في بيئة RADIUS، وسرعة خوارزمية في تقنية TOTPهو أعلى في حين أن سرعة خوارزمية في تقنية CROTPأعلى من تقنية HOTP، تقنية TOTP هي الأفضل من حيث وقت استجابة الملقم .أخيرا، من منظور أمني، نجد ان TOTP هو الأسلوب الأكثر أمنا في عملنا لأن OTP صالحة لفترة قصيرة، في حين أن CROTP هو أكثر أمانا من HOTP لأن الملقم يتحدانا مع PIN عشوائي في CROTP. ونتوصل الى ان ELSBOT هو الحل الفعال، حيث يكون أكثر صعوبة للمهاجمين الوصول إلى الخادم.

Abstract

Network security is a very important issue for organizations in order to protect their sensitive data from attackers. Number of researchers have provided a different security solution supported network protocols to enhance data privacy and confidentiality over networks. RADIUS is one of the most popular protocols used in network communication for user authentication. Unfortunately, there are many vulnerabilities facing the security issue in RADIUS network protocol. One of these vulnerabilities is a replay attack problem which is need to be prevented. The previous protocols have presented number of techniques to reduce the effects of replay attack in RADIUS protocol. One time password (OTP) technique is one of the most important techniques which are used to enhance the security of user authentication in numerous environments and to close the potential gap in network security. With several OTP techniques, our contribution are to chosen three techniques namely Time OTP(TOTP), Hash OTP(HOTP) and Challenge Response OTP (CROTP). This motivates us to present the ELSBOT (E-Learning System Based OTP techniques) for implementing the three OTP techniques to prevent the replay attacks in RADIUS protocol. This thesis presents a comparison between these OTP techniques in the ELSBOT. This comparison considers a set of factors like preventing replay attack, CPU overhead, algorithm speed, server response time and OTP duration. After measuring these factors through our ELSBOT, the results show that the three OTP techniques of ELSBOT prevent the replay attack in RADIUS environment, the CPU overhead at TOTP technique is less than the CPU overhead at HOTP and CROTP techniques, the algorithm speed at TOTP technique is the highest while the algorithm speed at CROTP technique is higher than HOTP technique, the TOTP technique is the best in terms of the server response time. Finally, from security perspective, the authors reach that the TOTP is the most secure technique in our work because its OTP is valid for a short time, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP. The ELSBOT is an efficient overall solution and it will be much more difficult for attackers to reach the ELSBOT server.

Keywords: RADIUS protocol , replay attacks ,One Time Password

List of Tables

List of Figures

Abbreviations and Acronyms

AAA	Authentication, Authorization and Accounting
ELSBOT	E-Learning System Based OTP Techniques.
NAS	Network Access Server
AES	Advanced Encryption Standard
TDES	Triple Data Encryption Standard
OTP	One-Time Password
PKI	Public Key Infrastructure
RADIUS	Remote Authentication Dial In User Service
SHA	Secure Hash Algorithm
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPN	virtual private network
IPSec	IP Security
PRNG	pseudorandom number generator
EAB	Extensible Authentication Protocol
PAP	Password Authentication Protocol
CHAP	Challenge-Handshake Protocol
DOS	Denial of service
EKE	Encrypted Key Exchange
EPPKE	Efficient Password-Proven Key Exchange
RSA	Ron Rivest, Adi Shamir and Leonard Adelman
S/Key	Secret-key algorithms
PSK	pre-shared key
PAKE	password-authenticated key exchange
TOTP	Time one time password
HOTP	Hash one time password
CROTP	The OATH Challenge-Response Algorithm
SOAP	Simple Object Access Protocol
JMS	The Java Message Service
PIN	Personal identify number
SMS	Short Message Service
AS	authentication server
CIA	Confidentiality ,Integrity ,Availability
MOE	The margin of error

List of Contents

Chapter 1

Introduction

Internet has become an essential part of our lives, and it is changing every aspect of our life. As Internet is widely being used at most of organizations such as banks, universities, airlines, electronic commerce, hospitals and telecommunications, organizations are required to protect their information from unauthorized access and data during transaction process.

Using authentication is a very important method to secure computer systems. Traditional authentication methods become weak to be used to secure networks as attackers use modern tools to monitor network traffic, so they can analyze data which is being transmitted to intercept passwords.

Many security protocols were developed to provide mechanisms for protecting data during transmitting over networks. Remote Authentication Dial In User Service (RADIUS) is one of the most common protocols to perform distributed services of Authentication , a Authorization, and Accounting (AAA) for dial-up remote access. RADIUS is supported by Virtual Private Network (VPN) servers, wireless access points, authenticating Ethernet switches, Digital Subscriber Line (DSL) access, and other network access types [2].

Recently, new versions of protocols that are interested with AAA have appeared. Diameter is considered more powerful than RADIUS, and it is more secured than RADIUS. However diameter is far complex than RADIUS and it stills a quite new protocol [40], it needs more time to be widely accepted at the market, and there is not much experience and support for it.

Currently, RADIUS is the de-facto standard for remote authentication. It is very prevalent in both new and legacy systems [37]. It is widely used in the industry as many hardware vendors support it, and there is also a lot of free and commercial software written in different programming language [40].

Thus, we decide to choose it as the AAA protocol for dial-up remote access. Unfortunately, there are several vulnerabilities in RADIUS protocol which we should avoid. These vulnerabilities are caused by the protocol itself or are caused by poor client implementation such as: response authenticator based shared secret attack, Denial Of Service (DOS) arising from the prediction of the request authenticator, user-password based password attack and replay attack. A replay attack is a form of network attack which is carried by an adversary who intercepts the data and retransmits it, a valid data transmissions maliciously or fraudulently repeated or delayed. [17].While poor implementation in Pseudo Random Number Generator (PRNG) in RADIUS server can be predicable and more likely to be repeated and violated from replay attack.

Previous studies have provided some suggestions to reduce the effects of replay attacks [5][6][61][69][70]. Some of these suggestions were using IPSec [11,12] and another suggestion was [2] using strong cryptographically request authenticators. Other researchers used smart card such as [26,27,34, 35] and [5] used approaches such as timestamp and sets clock. In [8] they are using hash function and secret key to

prevent replay attack. Other researcher suggested PSK techniques such as [25]. One Time Password (OTP) technique which is used at numerous environments to secure the potential gap in networks [7,8,10,14,29,41,42,54,57,58,61,63]. OTP can be used to provide secure access to remote users using different OTP techniques such as Hash OTP[41,42,43,67,68],Time OTP[41,42,45,67], and CROTP [44,63,67].
Our work is to implement and analyze these techniques in order to prevent replay attacks in RADIUS protocol environment.

1.1 Statement of the Problem

Poor implementation in PRNG be more predicable to guess and repeat, this gives chance of attackers to access to users accounts and steal data , replay attacks one of attacks that face RADIUS protocol . OTP techniques are used to prevent replay attacks . There are several OTP techniques used today , we are going analyze and compare three techniques of OTP's namely TOTP, HOTP and CROTP in RADIUS protocol.

1.2 Objectives

In this section , we present both main and specific objectives of research work. There has been a lot of research done and several techniques have been proposed and reviewed using OTP techniques.

1.2.1 The main objective

The main objective of this research is to implementing and analysis for three OTP techniques in RADIUS protocol environment to prevent replay attack and compare the results.

1.2.2 Specific Objectives:

The specific objectives of this research are:

- Knowing how to secure computers and networks from attacks.
- Clarifying the importance of authentication.
- Discussing RADIUS protocol mechanism and processes
- Discussing the weaknesses and strengths of the methods and techniques that are used to prevent replay attacks.
- Discussing OTP techniques in RADIUS environment.
- Designing and implementing of techniques to integrate RADIUS with OTP.
- Evaluating the factors such as preventing replay attack ,CPU overhead ,algorithm speed and server response time.
- Comparing the results with each other.

1.3 Scope of limitation

- The scope of the research will address and achieve all functionalities and components provided by RADIUS protocol format.
- This study useful according with dial up remote connection side.
- This study does not cover the physical security such as mobile token as solutions to prevent replay attack.
- The cryptograph algorithms is conducted of MD5 as hash function for all implementation in our work.
- This research analysis three OTP namely TOTP,HOTP,CROTP ,other OTP techniques will not be addressed by the our work , because of the time constraint for completing the thesis work.
- Access to the system must be through RADIUS protocol.

1.4 Research difficulties and obstacles.

We encountered a number of difficulties and obstacles while carrying out our work

1. Lack of research dealing with the comparison between the three techniques.
2. Lack in work with RADIUS protocol of local an environmental, so the configuration process of the settings were very difficult.
3. Process of synchronization between the server and the client are almost non-existent in the published research, which took significant time to understand these things.
4. The source code for OTP techniques exist free in the internet world, but unfortunately those codes lacking quite a lot of understanding because it is vague and there is a severe shortage of where they are incomplete as covered by a large number of mistakes, so we decided to start from scratch to create a code applied to the three algorithms .

1.5Thesis structure

The thesis is divided into sex chapters : Introduction , Literature Review, Related Work, Implementing OTP Techniques in RADIUS Protocol to Prevent Replay Attack , Test and Evaluation , Conclusions and Future Works . The main points discussed in the chapters are listed below :

- **Chapter 1** Introduction :gives a short introduction about our work and the thesis problem and objectives.
- **Chapter 2** Literature Review : describes the technical foundations needed for thesis work , we describe three sections , security section describes several authentication mechanisms which are related to the context of this thesis, section two present overview about RADIUS Protocol. It explains packets , work mechanism and vulnerability of RADIUS protocol , overview about OTP present in section three. It explains the definition of OTP, different methods for generation OTP and several techniques that use OTP.

- **Chapter 3** Related Work **:** previous studies in the area of authentication are presented in this chapter. Lots of published papers about RADIUS protocol, replay attacks and OTP have been discussed and reviewed.
- **Chapter 4** Implementing OTP Techniques in RADIUS Protocol to Prevent Replay Attack : Provides implementing for three OTP techniques such as TOTP ,HOTP and CROTP to prevent replay attack in RADIUS environment . We introduced ELSBOT as a case study of our work.
- **Chapter 5** Comparison Results: Presents an test and evaluation of the thesis ,we provide a set of factors to evaluate our work such as preventing replay attack, CPU overhead , algorithm speed and server response time . Then we provide comparison between three OTP techniques and discussed the results.
- **Chapter 6** Conclusions and Future Works **:** Provides concludes the findings and the future work.

Chapter 2

Literature Review

This chapter provides an analysis of the components related to network security, authentication and details about security principles. It is divided into three sections : section one describes explain security issues, section two explains RADIUS protocol, and section three describes OTP techniques and provides an analysis of the components related to OTP techniques.

2.1 Computer Security

Computer security is to prevent attackers from achieving their objectives through unauthorized access or unauthorized use of comuters and networks.

2.1.1 Goals of Security:

- ***Detection***: to detect activities that violate the security policy, detect intruders that sniff network and detect other attacks such as passive attack or active attack[75].
- ***Prevention:*** is ideal, because then there are no successful attacks, to prevent someone from violating security policy.
- ***Recovery***: to stop policy violations to assess and repair damage, ensure availability in presence of an ongoing attack and retaliation against the attacker.

2.1.2 The Components of Security:

1. **Confidentiality:** Keeping data and resources secret or hidden.
2. **Integrity:** Ensuring authorized modifications; and Includes correctness and trustworthiness[75].
3. **Availability:** Ensuring authorized access to data and resources when desired.
4. **Accountability:** Ensuring that an entity’s action is traceable uniquely to that entity.
5. **Security assurance**: Assurance that all four objectives are met.
6. **Authentication:** Identity authentication (a person; organizational entity; software agent; device).

2.1.3 Security Architecture:

Figure 2.1 explains security architecture that includes requirements and policies, information services, and security mechanism which are used to protect information from attackers and intruders.

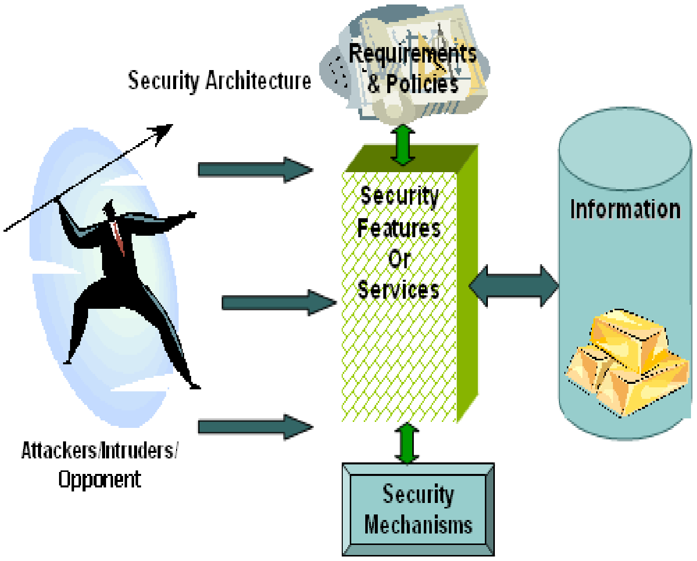


Figure 2.1 : Illustrates the Security Architecture.[15]

2.1.4 Information Security

Information security is well-informed sense of assurance that the information risks and controls are in balance. Information security is the systemic of information and its critical elements, especially the systems that use, store, and transmit that information[75].

It is the protection of the confidentiality, integrity and availability of information during transmission, storage or processing through the application of policy, technology, and education and awareness. It provides security for information during transmission over a network such as e-commerce transactions, online banking, confidential e-mails, file transfers, record transfers, authorization messages, etc.

2.1.5 Types of Attacks:

Attack is to gain unauthorized access to destroy, expose, change or steal assets. Two types of attacks are involved here:

- **Passive attacks:**

Passive attacks mean the monitoring of transmission or traffic analysis. Passive attacks are very difficult to detect because they do not involve any alteration of the data. Such as **Eavesdropping Attack** , Eavesdropping [72] creates the opportunity for adversaries to listen to or possibly extract personal details and information of their victims. Eavesdropping can be carried out through a number of ways. One way is by installing a spyware on the system. Another way is by using a network sniffer on the network to capture and reassemble packets as they are transmitted across the network.

- **Active attacks:**

The second major type of attacks is active attacks. These involve some modification of data stream or the creation of a false stream through stealth, viruses, worms, or Trojan horses. Active attacks result in the disclosure of data files, DOS, or modification of data.

2.1.6 Authentication

Authentication is one of the most important information security objectives. Authentication enables both to authorize (or not) a user to access a resource and to define different users on the system. Remote authentication means any infrastructure in which client and server are connected via some potentially insecure network such as Internet [63].

2.1.7 Authentication vs. Authorization:

Authentication is the process of identifying an individual, usually based on a username and password[75]. It can provide assurance that users (or systems) are who they say they are. While **Authorization** is the process of determining whether a client may use a service, which objects the client is allowed to access, and the type of access allowed for each user. It refers to a user's ability to access resources on a network, usually based on user account rights and privileges. Refer to "Access Control" for details about how authenticated users are allowed to access system resources.

2.1.8 Authentication Functions

Any message authentication or digital signature mechanism can be viewed as having fundamentally two levels. At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of message.

There are three types of functions that may be used to produce an authenticator[75]:

- **Message Encryption*:*** The cipher text of the entire message serves as its authenticator.
- **Message Authentication Code (MAC):** A public function of the message and a secret key that produces a fixed-length value that serves as the authenticator.
- **Hash Function*:*** A public function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

2.1.9 Authentication Methods

A variety of methods are available for performing user authentication, and these methods from the basis for access control systems. The three generally accepted categories of method for verifying the identity of a user are based on something the user knows, such as a Password-Based Authentication, something the user possesses such as an authentication token (Token-Based Authentication), and some physical characteristic of the user such as a fingerprint or voice pattern (Biometrics-Based Authentication) [54].

2.1.10 Sessions

A session provides a way to identify a user across more than one page request or visit to a website and to store information about that user [71]. While that are unique for each client that connects to the server. Illustration of a server with a separate session for each client shown in figure2.2 . Sessions are ideal for storing sensitive data on the server side where it is safe from attacks. It also makes it possible to associate information with individual users making it possible to adapt the data and functionality to the user that is connected and grant users different permissions.

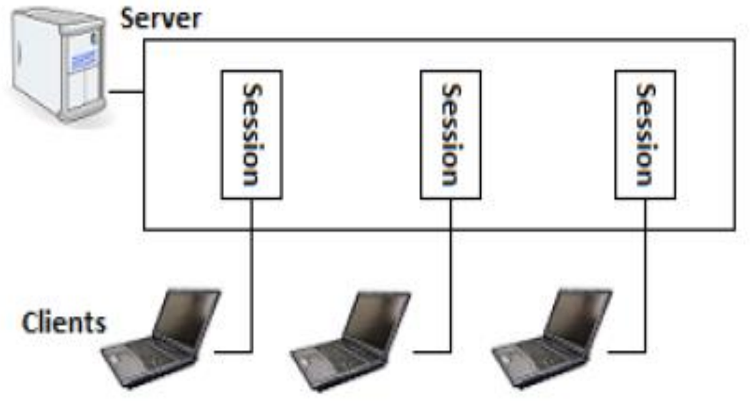


Figure 2.2 : Illustration of a server with a separate session for each client [71].

2.1.11 Authentication Protocols

There are many protocols of authentication in distributed environments, we will introduce a summary for major protocols authentication, as show in table 2.1 .

Table 2.1: Major protocols authentication

Protocol	Acronyms	Features	Protocol Uses
CHAP	Challenge Handshake Authentication Protocol	Uses hashes of passwords and time variant data to avoid straight password transmission.	MS-CHAP, PPP, APC Http, RADIUS
RADIUS	Remote Authentication Dial In User Service	RADIUS provides AAA. - allows user information to be stored on one host. - Minimizing the risk of security loopholes. - RADIUS can be adapted to work with existing security systems and protocols[1,15,37].	Backend for Telnet, SSH, SSL, Front end for Microsoft IAS Server. Typical central authentication method for network devices by UDP.[36]

TACACS+	Terminal Access Controller Access-Control System Plus	TACACS+ support Authentication, Authorization, Accounting, full encryption support,	Cisco protocol, central authentication, some RAS use (Remote Access Service) by TCP
Kerberos	-	Provide service authentication and authorization provide full encryption to data[48].	Kerberos applications like telnet, Microsoft domain authentication service integrated with active directory.[49]

2.2 RADIUS protocol

Authentication protocols designed to prevent unauthorized use of applications or services during communication to ensure integrity and security. There are many protocols used in networks. RASIUS is one of the most common protocols that handles authentication, authorization, and accounting (AAA) for dial-up remote access. Nowadays, RADIUS is commonly used to perform distributed services of transactions of users for network services. RADIUS created by Livingston, and the standard is described in RFC2865 [15].

2.2.1 Overview of RADIUS

Remote Authentication Dial In User Service (RADIUS) is a client/server security protocol that enables Network Access Server (NAS) to use shared authentication server for user authentication and authorization [5]. While user profiles are stored in a central location, known as the RADIUS server.

RADIUS clients communicate with the RADIUS server to authenticate users. The server specifies back to the client what the authenticated user is authorized to do [4]. RADIUS provides (AAA) management for computers to connect. However, this protocol build on top of the User Datagram Protocol (UDP) which was chosen instead of Transmission Control Protocol (TCP) as a transport protocol for technical reasons [12]. RADIUS is mostly used in the internet or any kind of internal networks which can be wired, wireless networks or integrated e-mail services. These networks may integrate modems, access points, DSL, access points, network ports, web servers or VPN‘s [63].Which can be seen in figure 2.3 given below

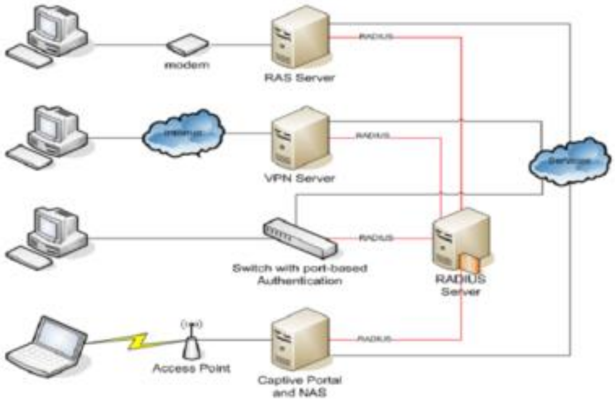
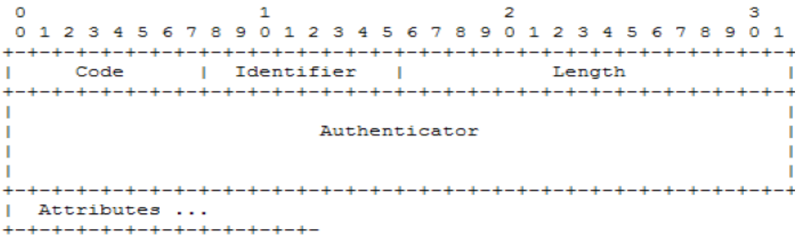


Figure 2.3 : RADIUS Architecture [63].

2.2.2 RADIUS Packet, Attributes, and Authentication Protocols

- **Packet**

The operation of the RADIUS protocol involves the exchange of six types of packets between client and server. RADIUS package as shown in figure 2.4



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2.4 : RADIUS Package [15]

Code is onek2 octet, and identifies the type of RADIUS packet. The code value 1 is used to identify the Access-Request type of packet, 2 for Access-Accept, 3 for Access-Reject, 4 for Accounting-Request and 5 for Accounting-Response[15] .And other used for future purposes.

Identifier is 1 byte long and is used to match the requests to their responses.

Length is two bytes long and specifies the length of the packet including the code, identifier, authenticator and attributes fields. The minimum length is 20 and maximum is 4096.

Authenticator is 16 bytes and is used by a RADIUS client to verify the validity of a RADIUS server's response and used by a RADIUS server for password hiding.

Attributes contain authentication, authorization or configuration information in TLV (Type, Length, and Value) format [15].

- **Authentication Protocols**

RADIUS is made to work with both Password Authentication Protocol (PAP) and Challenge-Handshake Protocol (CHAP). These protocols play an important role in authentication.PAP is a simple method of authentication and requires the user name and password for authentication and password is sent in clear text to the server. In CHAP password is not sent as clear text; encryption make it more secure that PAP [15]. The CHAP Password attribute then transfers both the challenge and the response to the RADIUS server [22].

2.2.3 Work Mechanisms

Operations that occurs between the RADIUS client and RADIUS server are authenticated through a shared key [2]. While user passwords are sent encrypted over the network.

Authentication and authorization: Authentication and authorization plays a key role in ensuring security system over any communication network, while authentication determines the identity of the user and whether the user has appropriate permissions to access the resource [65]. Authorization involves determining whether adequate information was provided to connect and grant services to the user when the user is connected. The RADIUS protocol combines the authentication and authorization processes by sending authorization information in the authentication response message. If the RADIUS server needs to challenge the user for a new password, it sends an access-challenge message to the NAS containing the challenge for the user. The NAS sends this information to the user and then forwards the response to the RADIUS server in an access-request message with the users username and challenge-response. The RADIUS server will reply with an appropriate response (accept, reject, challenge) as shown in figure 2.5

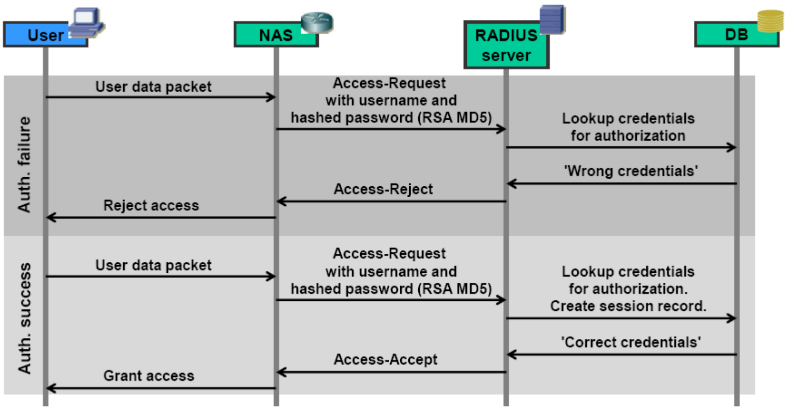

User
NAS
RADIUS server
DB
Auth. failure
User data packet
Access-Request with username and hashed password (RSA MD5)
Lookup credentials for authorization
'Wrong credentials'
Access-Reject
Reject access
Auth. success
User data packet
Access-Request with username and hashed password (RSA MD5)
Lookup credentials for authorization. Create session record.
'Correct credentials'
Access-Accept
Grant access

Figure 2.5: Authentication Flow [39]

- **Accounting**

Accounting involves tracking usage during the lifetime of connection for managerial purposes , billing, and planning. Transaction between objects show in figure 2.6

There are three types of logging for Network Policy Server (NPS):

- Auditing and troubleshooting connection attempts.
- Logging user authentication and accounting requests to a local file for purposes billing and traffic analysis in order to track the activity of a malicious user after an attack.
- Logging user authentication and accounting requests to a Microsoft SQL Server XML-compliant database [63]. It is used to allow multiple servers running NPS to have one data source.

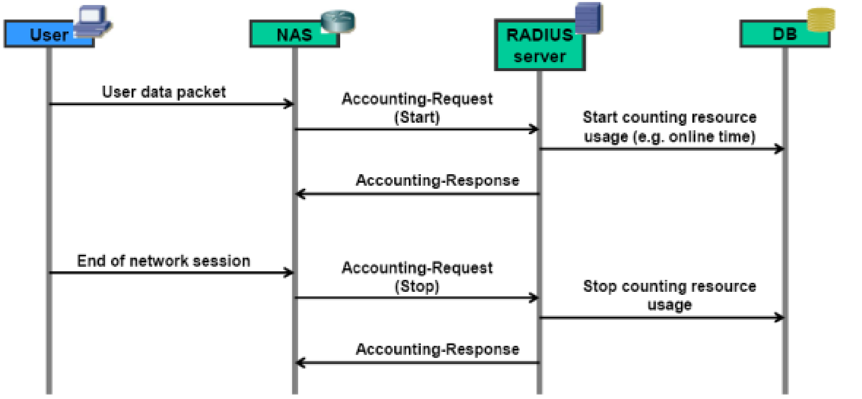

User
NAS
RADIUS server
DB
User data packet
Accounting-Request (Start)
Start counting resource usage (e.g. online time)
Accounting-Response
End of network session
Accounting-Request (Stop)
Stop counting resource usage
Accounting-Response

Figure2.6 : RADIUS Accounting [39]

2.2.4 Pseudo-Random Number Generator (PRNG):

Pseudo Random Number Generator (PRNG) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers [9]. While Security of RADIUS depends on the uniqueness and non-predictable generation of the Request Authenticator .

But poor implementation in PRNG either caused by the protocol or caused by poor client implementation be Repeated Request Authenticator .Some implementations exploit poor PRNGs to performs malicious behavior in protocol . Poor implementation in PRNG through Short cycles for generator be predictable and repeat and guess from by attacks such as dictionary attack birthday attack, and replay attack

2.2.5 The Vulnerability of RADIUS Protocol

RADIUS consistently provides some levels of protection against sniffing and active attacks[2]. Unfortunately, there are several vulnerabilities in RADIUS protocol that are either caused by the protocol or caused by poor client implementation such as:

1. **Offline dictionary attack** on RADIUS shared secret via message-authenticator attribute where attacker .It can attempt offline attack on any packet with a message-authenticator attribute[15].
2. **Online attack against the PAP password** in this attack, RADIUS servers enabling replay of request authenticator (16 octets) and identifier using PAP. Attacker can then try an online dictionary attack against the user password of 16 characters or less[15].
3. **Response authenticator based shared secret attack**, attacker observes a valid Access-Request packet and the associated Access-Accept or Access-Reject packet. They can launch an off-line exhaustive attack on the shared secret [23]. The attacker can pre-compute the MD5 state and then resume the hash once for each shared secret guess.
4. **User-password attribute based shared secret attack**, an attacker can gain information about the shared secret and attempts to authenticate to the client with a known password.
5. **User-password based password attack**, the attacker attempts firstly to authenticate to the client using a valid username, then captures the resulting Access-Request packet and determines the result of the MD5 then it's replay modified Access-Request packets, using the same request authenticator and MD5[23].
6. **DOS arising from the prediction of the request authenticator**, the attacker can predict future values of the request authenticator, then create a dictionary of future request authenticator values. The attacker can then masquerade as the server and respond to the client's requests with valid looking Access-Reject packets then creating a denial of service.
7. **Replay attack** :Attacker can get user password through passive eavesdropping or sniff traffic; an attacker can build a dictionary attack to find patterns and break a cipher, then replay to server with valid login. The adversary records a data transfer and replies it at any time through the network [17]. Replay attack is a method of exploiting a captured packet or packets and resend to user that cause

unexpected or unwanted behavior from the server [6]. If the server does not detect the reused data and accepts the repeatedly transmitted packets, the attack is successful. If an attacker would come across the data from a user that is generated by the JavaScript, it would be possible to login as the user without the server noticing any difference. The data is usually gathered either by listening to the traffic or by installing malicious software on the user computer. Full implementation for replay attack found in appendix B.

2.3 Password credential

In systems , we need to secure channels that access the resources. Remote authentication means any infrastructure in which client and server are connected via some potentially insecure network such as Internet [63]. Authentication needed to a password credential to gain access of resources or assets. Password is a secret word or phrase which is used to authenticate a user to gain access to a resource or assets. So far, researchers have proposed many remote authentication methods, including simple passwords, OTP, PKI sand Biometrics; every method has some advantages and disadvantages.

2.3.1 Static password

With our increasingly dependence on the internet in order to access business systems, the network security perimeter has crumbled at all levels. While the number of users who want network access has grown, traditional password authentication become unsuitable for securing the access requirements of today’s distributed users in the world is. The benefit of static passwords that they are easy to remember. However when you have different passwords for different systems, they start to become very difficult to remember and you might write them down[10], so this make them vulnerable.

Static passwords have many shortcomings [54] as users select passwords based on topics close to themselves such as birthdays, partner names, children’s names, etc.; that makes tradeoff between human memory and powerful password. The alternative method of password management is to change passwords regularly. A shortcoming of changing passwords frequently is that they can be easily forgotten.

2.3.2 OTP techniques

OTP is an instant password, in other words it is a code that changed after every time we use it to authenticate[45]. OTP are passwords that are only valid for a single or small number of transactions. An attacker has a smaller period of time to gain access to resources protected by such password because any previously stolen passwords will likely have become invalid. That means adding some uncertain factors in the procedure of authorization. Every time user logins, the information transmitted over network is different, thus the security is improved [63]. OTP has a characteristic making it impossible to predict the next password from the current password; also they are not vulnerable to replay attacks [3],[5],[6],[7],[26],[69],[35],[70]. OTPs avoid a number of shortcomings that are associated with traditional (static) passwords. On the downside, OTPs are difficult for human beings to memorize[45]. OTP is based on a cryptographic algorithms[53].

Cryptogram=f(k) while key k a cryptographic is generated.

Computing the cryptogram with factors makes the output random and on time, cryptographic algorithms based counter also called even and based time(e.g. seconds)with triple factor, f1: key k a cryptographic is generated, f2:T refers to time factor, f3: c refers to counter factor. While f1..i is number of factor, equation[1] show a cryptographic algorithms of OTP. Figure 2.7 explains OTP generation process.

Cryptogram=f(k , C , T)……………………………[1]

Generation OTP and distribution : The process of the OTP generation consists of
i) Input value.
ii) OTP generation.
iii) OTP extraction.
iv) Time.

The OTP generation algorithm generates an OTP value from an input value (users strong password and secret key). It is based on hash functions for message digest (MD5) and uses the shared input value between the server and the OTP generator [53]. A time value, a counter value and a challenge value that are used as the key and data of the generation algorithm. The extraction algorithm of the OTP value extracts the real OTP value from the output value of the OTP generation algorithm.

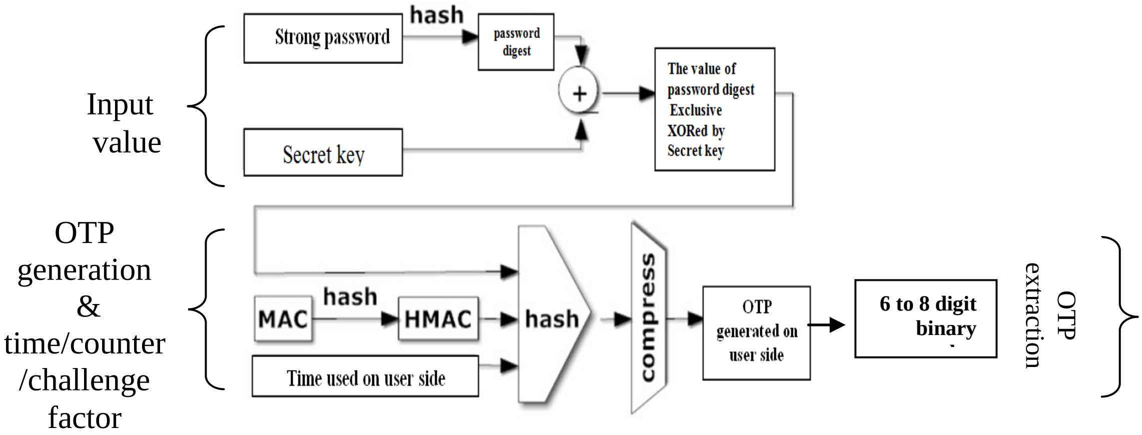


Figure 2.7: OTP generation [67].

2.3.2.1 Justification for OTP

Traditional user authentication mechanisms like public/private key pairs, PSK, IP Sec and PKI certificates. All of these methods suffer from a common weakness; users need to choose a good password contains strong passphrases to protect their privacy and credentials. While users access systems from remote hosts, their credentials are stored at untrusted hosts. While these hosts are infected by malware or phishing and spyware attacks that harvest untrusted key or password as they are entered, these credentials can be reused by an attacker. OTP techniques used to reduce the damage of passwords compromised through many attacks like spyware and replay attacks [53].

- **Various approaches for the generation of OTPs are listed below:**

1. **Time Synchronization** - In this technique, both the client and server will have synchronous time clocks. In this time it is used as a changing factor which changes every 3 minutes. The generation time must be synchronized with the authentication server time. If the authentication server and the user do not keep the same time, then the expected OTP value won't be produced and the user authentication will fail [64].
   With time-synchronized OTPs, the user typically must enter the password within a certain period of time before it's considered expired and another one must be generated.

2. **Event Synchronization** – In this technique, both the client and server will typically have a counter value. Whenever client wants to login, it generates OTP from the counter value and any other input Personal Identifier Number (PIN) and updates the counter. User submits the generated OTP to server. Server also generates the password for using the counter. If both passwords match, the server authenticates the user and updates the counter (increment/ decrement the counter), it may happen that the counter on client and server may drift (due to passwords generated by client but not submitted, passwords submitted by client but does not reach to server due to network failure, etc.).

3. **Challenge – Response technique** – In this technique a random number (PIN) chosen by the authentication server is sent to a user, the user enters PIN value then sends response to the server. This technique based on a challenge.

2.3.2.2 Several techniques that use OTP:

1. Lin OTP: is acronym for Linux One Time Password that uses OTP to increase the security of all types of logon processes.
2. MOTP: is acronym for Mobil One Time Password which deals with synchronization between client and server with period of time usually 3munities; several software downloaded on mobiles support this technology.
3. SMSOTP: SMS OTPs are used as an additional factor in a multi-factor authentication system. Users are required to enter an OTP after logging in with a user name and password [60].
4. HOTP: is acronym for HMACOne Time Password algorithm based on an increase in counter value [45]. Both client and server have a counter value. Server generates the password for using the counter. If both passwords match, the server authenticates the user and updates the counter (increment/ decrement the counter).
5. CROTP: is acronym for The OATH Challenge-Response algorithm based on challenge from authentication server. While server sends random challenge consists of 4 character defined as PIN, the user enters PIN value then sends response to the server.
6. TOTP: is acronym for Time One Time Password is used as an additional factor in a two factor authentication system. Users are required to enter an OTP after logging in with a user name and password to generate OTP in a period of time.

In our research, we discuss three techniques for OTP like TOTP, HOTP and CROTP. These techniques will be implemented under RADIUS protocol environment to prevent replay attacks.

1. HOTP Technique

The authors in [42] define HOTP as an HMAC-based One Time Password technique. The HOTP algorithm is based on an increasing counter value [45]. Both the client and server will typically have a counter value. Server generates the password to use the counter. If both passwords match, the server authenticates the user and updates the counter (increment/ decrement the counter), it may happen that the counter at client and server may drift (due to passwords generated by client but not submitted, or passwords submitted by client but does not reach to server due to network failure, etc.). In this case will response to server with denial service.

In [53], the researchers have provided the output of the HMAC-SHA-1 calculation in 160 bits, they have to truncate this value to a smaller digit so that it can be easily entered.

OTP (K,T) = Truncate(ToHex(HMAC-SHA-1(K,T)))

Where k present secret key. Truncate (***T***) converts the value generated through HMAC-SHA-1to an OTP value.

2. TOTP technique

The researchers of [43] describe TOTP technique that is a time based OTP. Time dynamism is an OTP generation principle widely utilized in two factor authentication schemes; two factor authentication based on PIN identification as first factor and OTP generation (different password) as second factor to increase security. Different password is needed for different time to prevent number of attacks, this schema depends on two periods, static period is designed for user to input the OTP into the login form upon receipt of the dynamic period password, and the length of which can always be customized by the client [67].

We define TOTP as TOTP = HOTP(K, T) , where T is an integer present time , k present secret key. The current time present initial time for the user logs in

The user calculates $T = \left\lfloor \frac{(current\ time - T_0)}{X} \right\rfloor$

X represents the time step in seconds (default value X = 180 seconds) and is a system parameter.

T0 is the Unix time to start counting time steps (default value is
0, i.e., the Unix epoch) and is also a system parameter.

3. CROTP Technique

In [43], CROTP is a generalization of HOTP with variable data inputs that based on an incremented counter and secret key values [45]. It is used in several network protocols such as RADIUS, Kerberos and digest access authentication. The definition of CROTP requires a cryptographic function, a key K and a set of data input parameters [47]. Instead of a shared counter value, a challenge is issued from the validation server to the user. The user would then input this challenge to receive the OTP value. Where CROTP = CryptoFunction(K, DataInput).

While K represents a shared secret key, data input function represents input data values, crypto function represents a function that perform the CROTP. In details inputs that based on challenge response (CR) authentication mechanism consists of the server which present the user with an unpredictable challenge every time the user attempts to authenticate. For every challenge there is an associated response that allows the user to be authenticated if he is able to compute and sends it to the server. As a consequence of the unpredictability of the challenge, this family of mechanisms ensures that the entity being authenticated is active when authentication takes place. Passive attacks are not possible, and even though active attacks remain feasible, they need to be carried out in a real time, which means that the time interval between the moment when the attacker captures the response and when it is used needs to be very short for the user in order to accept it [75].

2.4 Conclusion

In this chapter, we offer different issues about Literature Review . Firstly, we provide concepts about computer security , information security , authorization , authentication , accounting and types of attacks . Secondly, we introduced concepts about RADIUS protocol with mechanism of work , package, and vulnerabilities that facing . Thirdly , we offer OTP techniques and deal with several OTP techniques ,OTP generation and various approaches for the generation of OTPs , also we discussed OTP techniques (TOTP,HOTP , CROTP).

Chapter 3

Related work

In this chapter , different related works are studied and investigated. The related works are introduced and analyzed with respect to the thesis problem to show how far these works address the requirements of our thesis problem.

3.1 RADIUS protocol

A lot of published papers about RADIUS protocol that proposed different approaches to solve various issues of the RADIUS protocol have been discussed and reviewed. Md. ***Hashmathur et.al.*****[2]** have presented different approaches to minimize or resolve problems of the RADIUS protocol using best deployment practices and extensions.These approaches used strong shared-secrets, cryptographic-quality values for the request authenticator, and IPSec to provide data confidentiality for RADIUS messages. Other researchers like ***Ang et.al.*****[3]** have discovered that some weaknesses exist in Encrypted Key Exchange(EKE) protocol that are subjected to replay attacks; also they proposed an improved scheme Efficient Password-Proven Key Exchange (EPPKE) protocol that generates the session keys timestamp against replay attack, and the authenticity of public key without trusted online third parties. Previously, they used timestamp against replay attack that used in logging events to protect messages from manipulations users who use session keys that encrypt all messages in transaction communication session.

In 2012, ***Rohan.***[1] has developed a proposal to improve the communication efficiency between NAS and RADIUS server by allowing the RADIUS server to communicate its state (active/dead) to NAS. Their proposal has effectively helped to improve less CPU utilization in the network. The paper explain how to deal with many of server processes such as closing the session after no response from server side (wait specific time), so this help to reduce time and reduce retransmissions. This paper didn’t consider prevention of replay attack, and required to handle that in a future work in simulation model of interaction between NAS and RADIUS server . *Previously, researchers introduced different approaches to minimize or resolve problems of the RADIUS protocol .They developed schemes against many of attacks such as replay attack, DOS, and phishing attack using best deployment practices and extensions*

3.2 Techniques suggested to reduce from effect the replay attacks

Different techniques have been discussed and reviewed to prevent replay attack in numerous environments such as IPSec , PKI ,PSK ,smart card and OTP techniques. We offer related work about each these techniques

3.2.1 IPSecurity (IPSec) technique

Mikko. [12] has presented the implementation of legacy user authentication into IPSec remote access scenario using the proposed Pre-IKE Credentials Provisioning Protocol(PIC), this without using to analyze the PIC into IPSec to get more security and more speed. The researcher have provided a comparison between his technique and other alternative techniques. The results of this comparison show that his technique is a good interoperability, usability and efficiency.

Chin et.al.[35] they discussed the anti-replay window protocol in IPSec, and potential problem when a long-jump reorder occurs. They used a controlled shift mechanism as addition to the protocol. Controlled shift can reduce the harm caused by long-jump reorder. They presented a modified version of the anti-replay protocol that incorporates the controlled shift mechanism. *Previously, they used IPSec to remote access integrated with PIC to be more secure. While the IPSec transmits small packets, the encryption process of IPSec generates a large overhead, so this diminishes the performance of the network.*

3.2.2 Pre-shared keys (PSK) technique

In 2012, ***Swati et.al.[30]*** have discussed the three authentication security protocols. He used pre-shared keys (PSK) approach against replay attacks. While Wired Equivalent Privacy(WEP) does not prevent replay attack, Wi-Fi Protected Access (WAP) implements anti-replay mechanism that works by keeping track of the sequence numbers in packets as they arrive. Other researchers like ***Hayriye et.al. [31]*** presented two alternative pair-wise key exchange protocols for Robust Security Networks (RSN) in wireless local area networks (WLAN). They used the four-way handshake protocol to provide mutual authentication, their proposed addition was to generate the message integrity check (MIC) code over the whole message containing PSK with sequences counter to prevent replay attack in their schema, this requires further storage space for the counters.

Previously, they used PSK approach. PSK static key is shared between two parties for initiating the communication, and it does not scale. If PSK on a remote access server is changed, a client with a manually configured PSK will be unable to connect to that server until the PSK on the client is changed.

3.2.3 Public key infrastructure (PKI) technique

Mark et. al. [21] used PKI and client certificates as a case study in their work at Digital Certificate Operation in a Complex Environment (DCOCE) project that has been executed at oxford university. As advantages and disadvantages of end user/client digital certificate were explored, the DCOCE project tested an alternative model of PKI, and found that both PKI and client certificates have more security more than their work.

In 2008, ***Hyun et. al. [7]*** have analyzed the problems of vulnerability of authentication mechanisms by using existing S/key authentication mechanism. CROTP and TOTP used public key infrastructure to solve it. His proposed mechanism can prevent spoofing attack in advance by authenticating user with the use of certificate information, and solves the problems of replay attack, Time synchronization and integrity by generating password though applying hash function for label L and random value R which are only used in applicable session. Also, they transferred the generated password by electronically signing with user’s private key.

Jason et.al.[32] have used PKI certificate to establish a Secure Sockets Layer (SSL) connection to mutually authenticate with each other (client and server). He avoided replay attacks by adding the SSL session key into the keyed-hash message authentication code of a cookie like HMAC(user name |expiration name| data| session key, sk) as the keyed-hash message authentication code of each cookie. *Previously, they used PKI for mutual authentication. PKI requires a secure storage of the corresponding private key and trusted distribution of public keys. Also, PKI is complex and not widely deployed.*

3.2.4 Smart card technique

Chun et. al. [29] have presented a new improved unilateral asymmetric authentication protocol that can be used against replay attacks and active attacks. The complexity of this proposed protocol is higher. He used a smart card to protect user authentication. This protocol is useful for applications that are important and critical. ***Misbahuddin et. al. [34]*** have used a password based authentication schemes for authenticating remote users. He proposed an efficient scheme for remote user authentication with smart card, also proposed scheme that achieves mutual authentication which is essential for many applications. Their scheme is secure against ID theft, guessing attack, replay attack, impersonation attack, and reflection attack.

In 2012, other researcher like Tyagi et. al. [27] suggested robust multi-server authentication scheme using smart cards based on cryptographic one-way hash function. This scheme allows remote users to access multiple servers without separately registering with each server. Also, it eliminates the use of verification table, this scheme provides mutual authentication and establishes a common session key between user and the server. The proposed scheme withstands user impersonation attack, server impersonation attack, replay attack, smart card loss attack and stolen verifier attack. *Previously, they used physical techniques for verification of user authentication using smart card and hardware toke, their results gave a good indication to support secure authentication against attacks.*

3.2.5 OTP techniques

Xi Yang. [8] has used OTP techniques to establish "The Generic Security Service Application Program Interface" (GSS-API) security context between two communicating peers. He compared what he proposed with Kerberos and public-key technologies. While OTP techniques provides greater security for user authentication, he used SHA-1 algorithms for integrity message to enhance the security.

In 2013, other researcher like Jonghoon et. al. [60] have introduced a new protocol to assure more secure authentication. His protocol does not only prevent from cloning the OTP generator, it prevents phishing attack through transaction information. The proposed protocol requires the OTP generator by using the OTP generator equipped with keypad. their protocol enhances security and provides more robust authentication method than existing ones. **Other researchers like Kenneth et. al. [13]** has presented a model for the secure use of OTP in password-authenticated key exchange PAKE protocols; considering the idea that such protocols should be secure even if previous or future OTP have been compromised. They have given a generic technique for constructing secure OTP, this construction can be used with pseudo randomly generated OTP, providing greater efficiency in OTP distribution to reduce the damage of many attacks such as replay attack and spyware.

Xuguang et. al. [67] suggested a secure dynamic user authentication scheme that is based on a dynamic OTP with both time and space (location). Their schema used time synchronization to add time factor to OTP and effectively improves two-factor authentication to protect users account against various attacks such as phishing attack, replay attack, and perfect-man-in-the-middle attack. ***Chii-Ren. [39]*** has proposed two potential non-repudiation architectures for financial transactions using CROTP tokens and digital signatures. He compared OTP tokens with digital signature, where OTPs recorded more trust than digital signatures for high-value transactions.

Havardet et. al. [40] have developed protocols that enable individuals to use their mobile phones as OTP generators by a web-based service. Their phones run a Java MIDlet which communicates with a server to generate OTPs. ***Andrew et. al. [41]*** have concluded that OTP approach to enhance security password-based authentication from both a security and usability perspectives, TOTP and HOTP, he regard them as being equally effective. In both cases, he emphasizes the importance of user behavior in the security of the techniques.

Table 3.1 : Advantages and disadvantages of techniques against replay attacks.

Technique	Advantages	Disadvantages
IPSec [11,12]	- IPSec provide security and Scalability - IPSec can be applied in all networks regardless their size[12]. -IPSec used in application Independence -IPSec is not limited to specific applications[11].	During communication over networks , when transmitting small packets, the encryption process of IPSec generates a large overhead[12].. This diminishes the performance of the network. IPSec is very complex. Complexity increases the probability of the presence of a weakness or hole. For example, IPSec is weak against replay attacks [5].
Smart card[27,33,34]	High levels of security, ease of use without need for online connections or via telephone reduce fraud.	It needs a smart card reader **[35]**. It is difficult to be implemented due to needed active directory.
PKI [7,21]	- Provide secure way against many of attacks such as replay attacks and man in the middle attacks.	PKI requires secure storage of the corresponding private key and trusted distribution of public keys [7]. PKI is complex and not widely deployed.

PSK [31,32]	PSK is a static key shared - between two parties for initiating the communication. - **PSK** no required PKI , PSK are simple to configure on a remote access server, and consider relatively simple to configure on a remote access client[31].	PSK does not scale, if the pre-shared key on a remote access server is changed, a client with a mutual configured PSK will be unable to connect to that server until the PSK on the client is changed. If the PSK was distributed to the client within a connection manager profile, that profile must be reissued with the new PSK and reinstalled on the client computer and is only usable on static links[31]
OTP [4,7,8,10,13, 29,40,41,42, 45,53,56,60, 62,66,67]	OTP defeats eavesdropping attacks OTP is instance and not reusable Can be time-limited such as TOTP . Can be used over untrusted communication paths[67]	If the attacker can guess the secret key that was used to generate the list of passwords, there is a chance that he can determine the OTPs that were generated[10].

Previous studies offered five techniques against replay attacks. we summarized the advantages and disadvantage of these techniques in table 3.1.

Previously, we have presented the security models proposed by various researchers in recent and past. We have summarized the security approaches which are used to secure the RADIUS protocol. Most of the proposals have given numerous security techniques at each vulnerabilities. Some researchers have implemented the security techniques to secure the important data from unauthorized access. They suggested several methods to reduce the effect of replay attacks in numerous environments, but there is still a need to completely elimination of replay attacks attack. There are a large number of recent papers that implemented OTP mechanism for user authentication; OTP proved worth and efficiency against attacks are

In this research, we have discuss OTP techniques in RADIUS environment using OTP techniques . This techniques proposed to user authentication by eliminating the risk caused of reused passwords. With different OTP techniques as HOTP[41,42,43,63,64], TOTP [41,42,45,63,64], and CROTP [44,63,64]. Our work is to implement and analyze these techniques in RADIUS protocol environment to prevent replay attacks.

Conclusion

In this chapter, we presented related work for security models proposed by various researchers in the recent past. Numerous security techniques discussed by researchers to prevent replay attacks in RADIUS environment such as PKI Certificates technique, PSK technique, smart cart, IPSec techniques and OTP techniques that included TOTP, HOTP and CROTP. We also introduced a number of researches about each techniques, then we explained advantages and disadvantages of all the discussed techniques.

Chapter 4

Implementing OTP Techniques in RADIUS Protocol to Prevent Replay Attack

This chapter presents the ELSBOT (E-Learning System Based OTP Techniques) for implementing the three OTP techniques (TOTP, CRTOP, HTOP) in the RADIUS environment in order to prevent the replay attack. This ELSBOT is shown as the basic part of our work to protect user data from replay attack. The functional architecture of ELSBOT is provided to draw the main components of our ELSBOT. In this chapter, the authentication procedures are shown to represent the login process for the system user, and then the appropriate environment is prepared for applying this ELSBOT in the RADIUS protocol. The OTP application is introduced to show the OTP generation process at each of the three OTP techniques. Finally, an attacked script is implemented for representing the replay attack process in our work. This script is used to eavesdrop on the user in order to help us that if the ELSBOT is prevent this attack.

We choose three OTP techniques namely TOTP, HOTP,CROTP due to these techniques used in user application and generalized for all , in other word , TOTP and HOTP are used in mobile application MOTP ,mail mailOTP and smsOTP , also CROTP can be used in many application in servers challenges.

The authors are choose the OTP length to be sex character by default, while the OTP techniques dealing with many of application that need to be easy to carry and loading such as Mobil application , SMS application or mail application.

4.1 ELSBOT Architecture

In this section, an integral system is presented to implement three OTP techniques in order to evaluate whether our work prevents the replay attack in RADIUS protocol. The aim of ELSBOT is to find a secure authentication mechanism to prevent the replay attack in the RADIUS environment and protect user information from theft. The ELSBOT is an educational system that offers a set of services for students and staff in a college like add new students and courses, register and withdraw courses, delete students and courses, update students and courses. In general, we provide the functional architecture of our system that has five components namely:

1. **User**

   User is a person who requests access to the system services.
2. **System**

   The System is a website that will design to provide services to end users.
3. **RADIUS server**

The RADIUS server checks the user information (username and password) which is entered in the sign-in form. The RADIUS server has a database to store the user data .

4. **Authentication server (AS)**

   AS is responsible of the second phase in our system. After The system user enters the PIN and the generated OTP in the OTP page, the authentication server will check the secret key, PIN and last OTP and then will send a response to the system.

5. **OTP application**

   OTP application is a software which is downloaded on the user computer at anytime and in anywhere. This application is responsible of the OTP generation process for each of the three techniques.

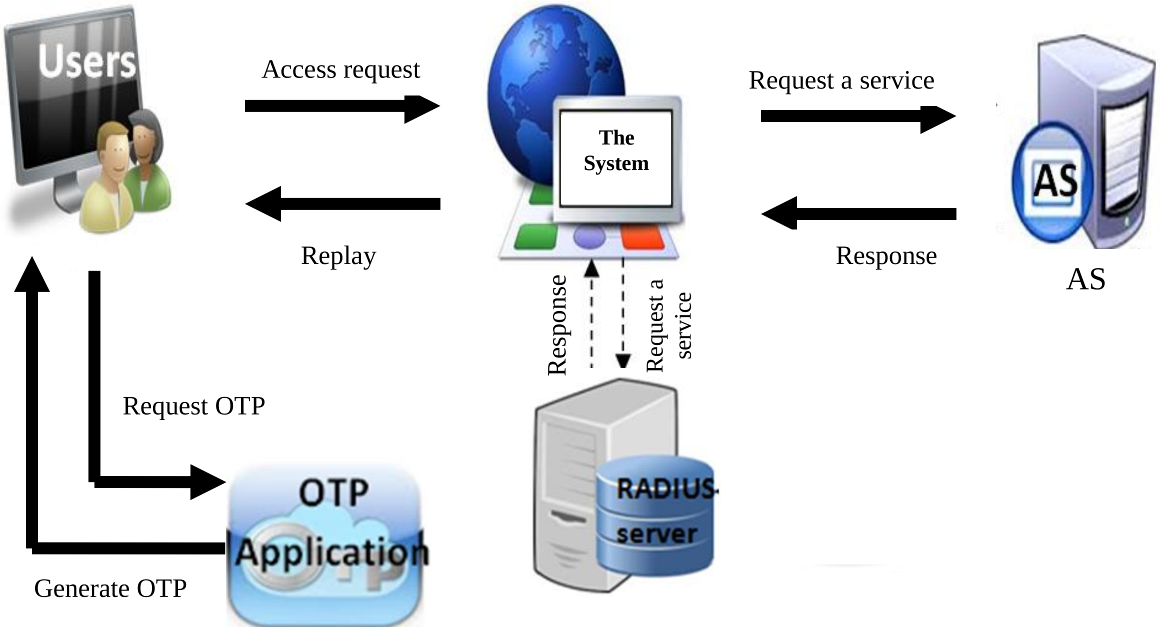


Figure 4.1: Functional Architecture of ELSBOT

4.1.1 TOTP Technique

In this section, the architecture of TOTP technique is offered and then the flowchart of this technique is drawn. The flowchart shows the OTP generation process using the hash algorithm. Flowchart for TOTP display in appendix A:1

- **TOTP Architecture**

The architecture of TOTP is provided as shown in the figure 4.2. In this figure, we show the integral process of TOTP which occurs in our ELSBOT through presenting a set of steps. Table 4.1 describes these steps as shown below:

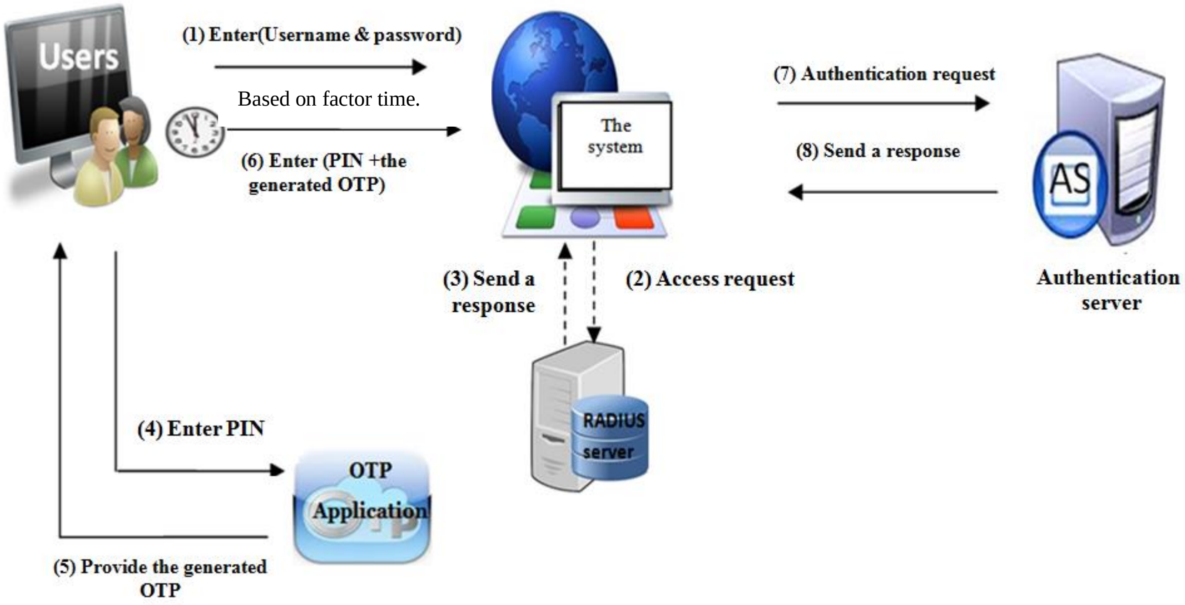


Figure 4.2 : TOTP Architecture

Table 4.1: Illustrates the steps of TOTP architecture

Step #	Description
1.	The user enters the username and password in the login screen. The password should contain more than six characters.
2.	The system sends user data to the RADIUS server to authenticate the user.
3.	The RADIUS server verifies the username and password and sends a response either accept or reject.
4.	The user opens the OTP application and enters the PIN.
5.	The OTP application provides the generated OTP for the system user. This OTP is expired after 3 minutes.
6.	The user enters the PIN and the generated OTP through the system.

7.	The system sends a request to the AS to check the last OTP, the PIN and the secret key of the system user.
.8	The AS verifies user OTP and sends a response to system either accept or reject.

4.1.2 CROTP Technique

The architecture of CROTP technique is presented in this subsection and then the flowchart is shown to draw the OTP generation process using the hash process. Flowchart for CROTP display in appendix A:2

- **CROTP Architecture**

This subsection introduces the architecture of CROTP in the ELSBOT as shown in the figure 4.4 and then presents a set of steps to show the integral process of CROTP which occurs in the ELSBOT. These steps are described in table 4.2

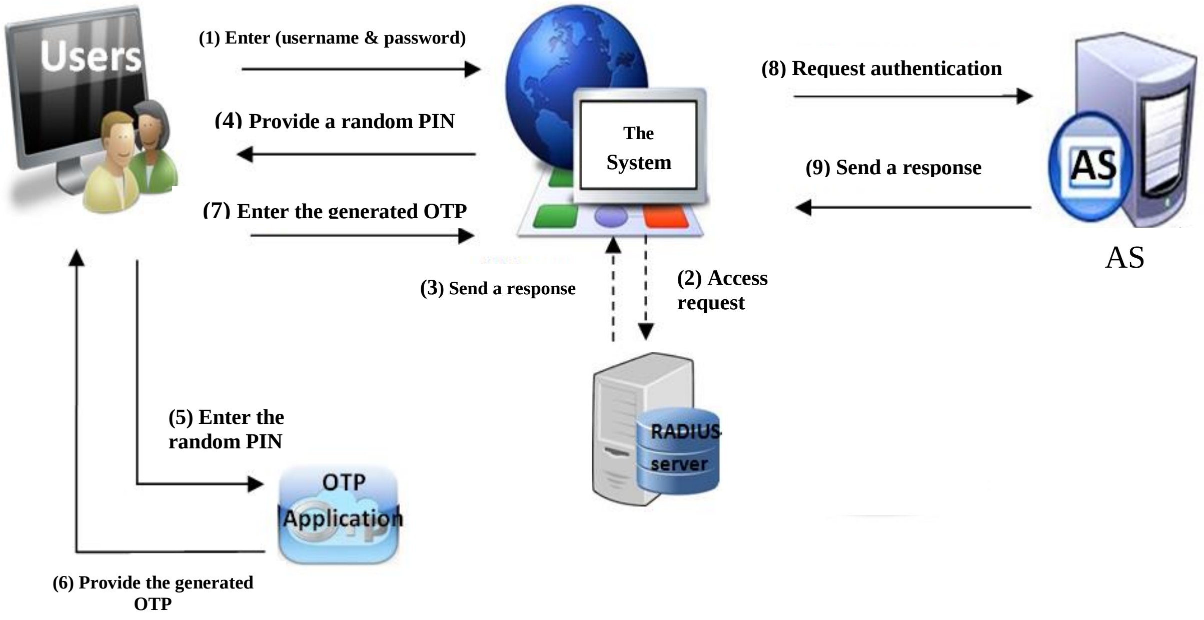


Figure 4.4: CROTP Architecture

Table 4.2: Illustrates the steps of CROTP.

Step#	Description
1.	The user enters the username and password in the login screen. The password should contain more than six characters.
2.	The system sends user data to the RADIUS server to authenticate the user.

3.	The RADIUS server verifies the username and password and sends a response either accept or reject.
4.	The system provides a random PIN to challenge the user .
5.	The user opens the OTP application and enters the system random PIN.
6.	The OTP application provides the generated OTP for the system user.
7.	The user enters the generated OTP through the system.
8.	The system sends a request to the AS to check the last OTP, the random PIN and the secret key of the system user.
9.	The AS verifies the user OTP and sends a response to system either accept or reject.

4.1.3 HOTP Technique

This section presents the architecture of HOTP technique and then offers the flowchart for this technique. The flowchart shows the OTP generation process using the hash algorithm. Flowchart for HOTP display in appendix A:3

- **HOTP Architecture**

The figure 4.6 draws the architecture of HOTP technique. In this HOTP architecture, a set of steps are presented to show the integral process of HOTP for our ELSBOT. These steps are explained in table 4.3.

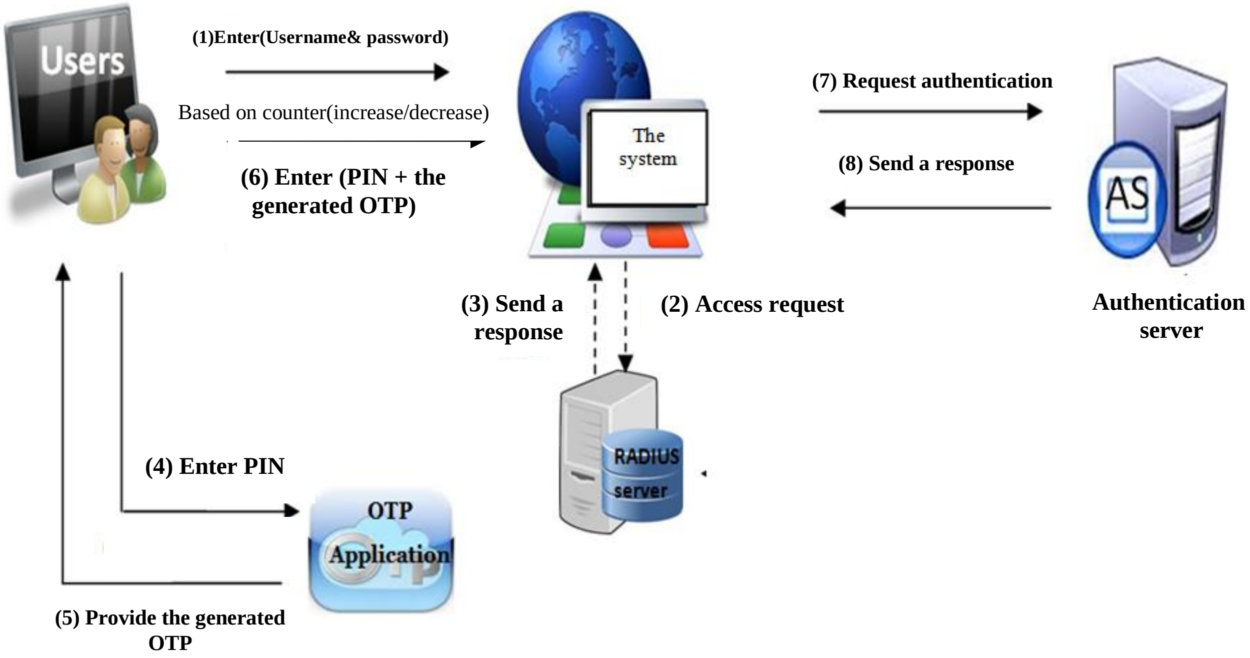

Users
(1)Enter(Username& password)
Based on counter(increase/decrease)
(6) Enter (PIN + the generated OTP)
The system
(7) Request authentication
(8) Send a response
AS
Authentication server
(3) Send a response
(2) Access request
RADIUS server
(4) Enter PIN
OTP Application
(5) Provide the generated OTP

Figure 4.6: HOTP Architecture

Table 4.3: Illustrates the steps of HOTP architecture

Step#	Description
1.	The user enters the username and password in the login screen. The password should contain more than six characters.
2.	The system sends user data to the RADIUS server to authenticate the user.
3.	The RADIUS server verifies the username and password and sends a response either accept or reject.
4.	The user opens the OTP application and enters the PIN.
5.	The OTP application provides the generated OTP for the system user.
6.	The user enters the PIN and the generated OTP through the system.
7.	The system sends a request to the AS to check the last OTP, the PIN and the secret key of the system user.
8.	The AS verifies user OTP and sends a response to system either accept or reject.

4.2 ELSBOT Requirements

This section presents the guaranteed requirements to be taken when applying the ELSBOT. The basic requirements of ELSBOT are categorized in terms of software, hardware and network. These requirement are shown below:

1. Software Requirements:

The table 4.4 provides the software requirements for our work. The most important elements for software requirements are platforms, operating systems, databases, software modules and additional libraries.

Table 4.4: Software Requirements

Platform	Operating Systems	Software	Additional libraries	Modules
RADIUS server	Linux CentOS 6.4.1	FreeRADIUS 2.1.12	SOAP	LAMP
Authentication Server	Windows 7	Apache	SOAP	Xampp
Simple Web site	Windows 7	PHP	jquery	Xampp
OTP application	Windows 7 at user computer	Java language with JDK() and Netbeans IDE 7.3	Hash -MD5	-

2. Hardware Requirements:

- Main Processor : Intel core 2 Due.
- Ram : 2 GB .
- Mother Board : 945 gvm Intel Chipset.
- Hard Disk: 80 GB.
- Mointor:17 Color Monitor.
- Keyboard : Standard 102Keys.
- Mouse: Optical mouse

3. Network Requirements

- Network LAN topology: Star network .
- Connection :Wired.
- Net ID : 10.10.10.
- Subnet mask: 255.0.0.0
- Port : 48.
- Switch : layer 2 managed.
- Router : CISCO
- Cable : STP

4.3 ELSBOT Databases:

Present three databases for our ELSBOT as shown in figure 4.8. In this figure, firstly, the RADIUS server checks that if the user is exist in the radius database which contains all system users. Secondly, the authentication server checks the last OTP, PIN and secret key of the system user. This data is saved in the authentication database. Thirdly, if the user is admitted by the two servers, the user can enter to the system which has a college database. The three databases of ELSBOT are described below:

1. The College Database:

The College database is a data store that contains information about all the users, students, courses, and other information related to our system. All the tables of the system like Student, course_ student, courses tables include .

2. The RADIUS Database:

The RADIUS database checks and verifies from user login process. This database includes two tables: Acct.radius_userstable, and Logs table .

- Acct.radius_users table this table is responsible of checking and verifying from user sessions for entering to system, while it ensures IP, hostname be to the correct session.

- Logs table includes fields such as ID, request time, response time, this table explains basic transactions between user and server where time expressed as time Unix Stamp.

3. The Authentication Database

The authentication database includes two tables, the first called settings table, and the second called logs table:

- Settings table : includes Id of users and type of mechanism (TOTP, HOTP , CROTP) for OTP techniques. Administrator of the system determines the login technique from the three techniques for users, so the user can login to system within one technique.
- System_users table : this table contains Id, username, recent OTP, recent Counter and secret key. Secret key is saved in the database in a cipher manner, implemented by using message digest. The use of server's secret key makes it difficult to implement OTP generation manually even if the attacker gets the hashed password table. So, attackers cannot get any information about the user's static password from the OTP. Through this table, server determines valid user by applying database query.

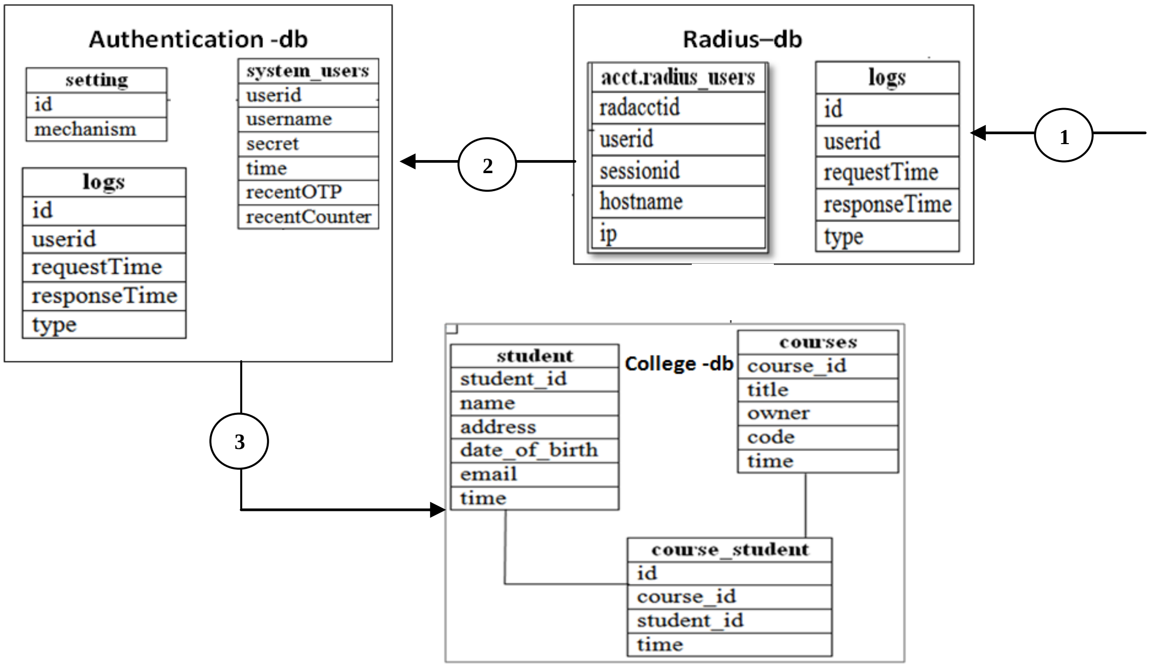


Figure 4.8 : ELSBOT databases

4.4 ELSBOT Authentication Procedures

This section introduces the authentication process in the ELSBOT. The authentication process is divided into three processes: sign-in, sign-up, OTP generation. Three scenarios are described to illustrate the three processes: the first scenario shows how an existing user enters to the system as shown in figure 4.9; the second scenario shows how a new user registers for the first time in the system as shown in figure 4.10; the third scenario shows how the system user gets the OTP that is generated by one of the three OTP techniques in our system as shown in figure 4.11. The three scenarios of authentication procedures for our work are described below:

4.4.1 Sign-in Scenario

In the sign in scenario, the user enters his username and password in login page of ELSBOT. RADIUS server checks the user data and send a response of acceptance if the user already exists. Our ELSBOT transmits this user to OTP page which contains two fields like PIN and the OTP. Thus, the user must open the OTP application in order to generate the OTP, and then copy the generated OTP from the OTP application. In the OTP page of the ELSBOT, the system user enters the PIN and the generated OTP. Finally, the authentication server checks the user PIN and the generated OTP and sends a response either accept or reject .

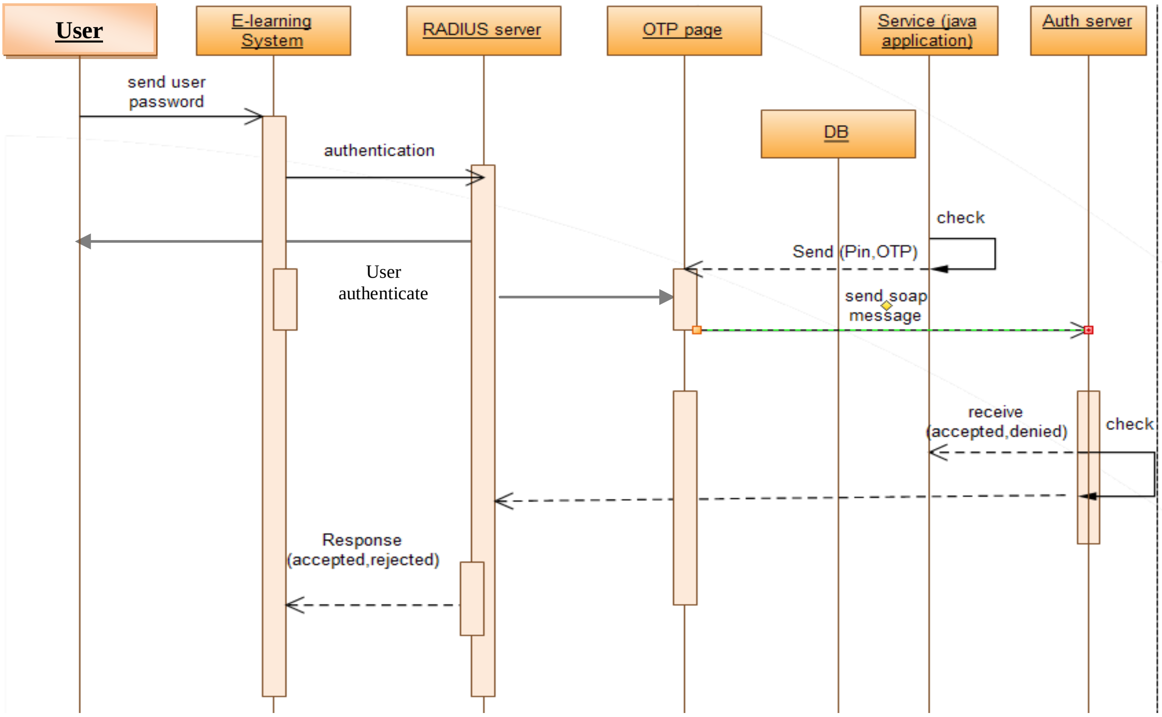


Figure 4.9 : Sequence diagram for Sign-in process

4.4.2 Sign-up Scenario

The figure 4.10 illustrates the sign-up process of our ELSBOT. In this process, the new user registers his information to the system through a set of the required fields like personal name, user name, email, and password. If the user enters all the required fields, the system sends the secret key. The secret key is a key which is given to the user only once when the user registers as a new user in the system. The new user cannot get the OTP without this key. Thus, the user take the secret key to the OTP application in order to get the generated OTP.

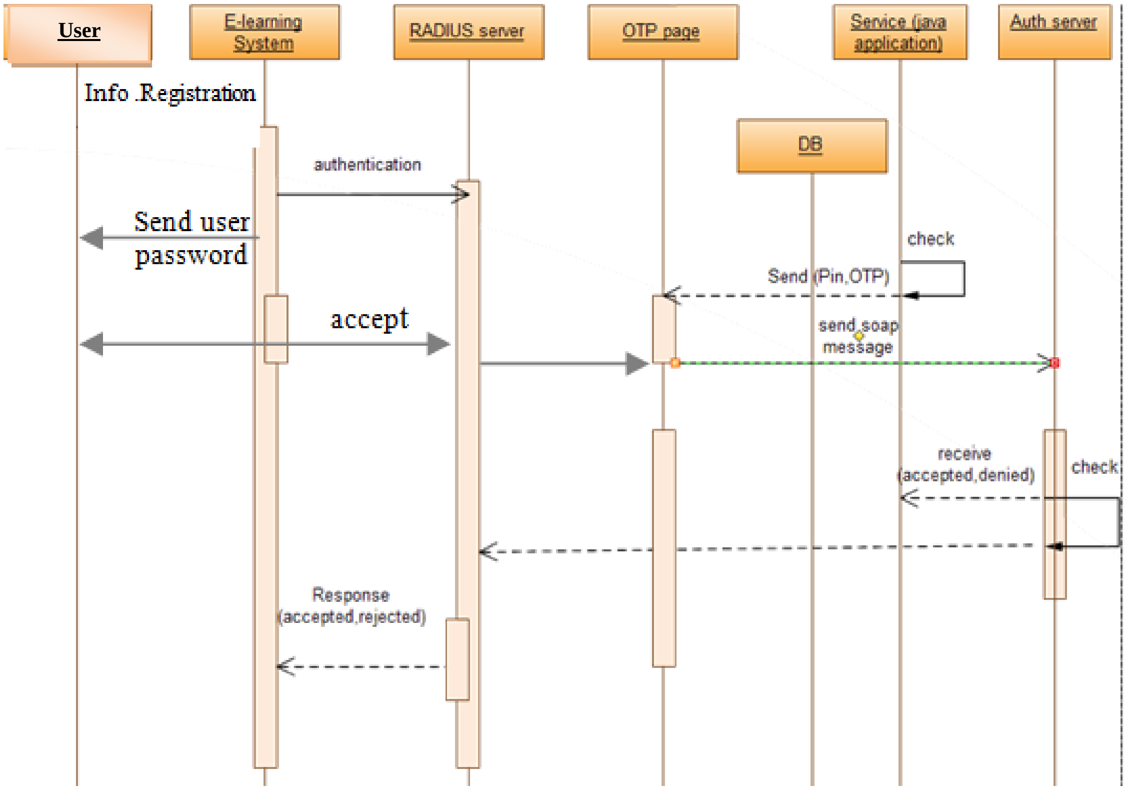


Figure 4.10 : Sequence Diagram for Sign-up process

4.4.3 OTP Generation Scenario

This subsection describes the OTP generation scenario to show how the user OTP is generated through OTP application. The OTP application is a software which is implemented by java language. After the user is authenticated (sign-in process) by RADIUS server, the system transmits the user to the OTP page which is an important page in the login process in the ELSBOT. So the user must open the OTP application and enter the PIN and then get the OTP which is generated by one of the three OTP techniques. Finally, the user takes the OTP and enters the PIN and OTP in the OTP page of the system. After that the system user can login to the system if the matching process is correct by the authentication server.

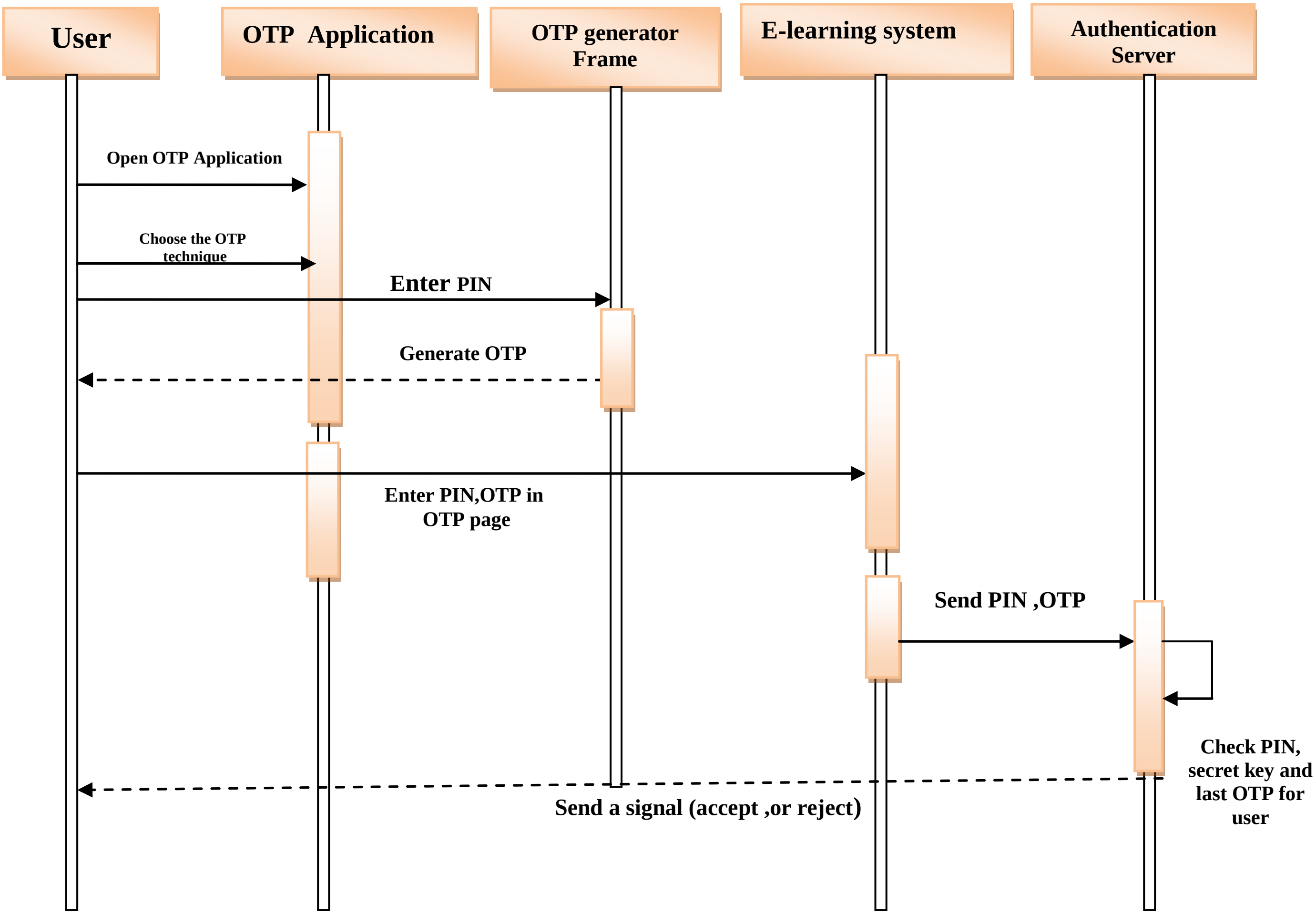


Figure 4.11: Sequence Diagram for OTP generation

4.5 ELSBOT Environment

We prepare the environment of this system through a set of steps. These steps are shown below:

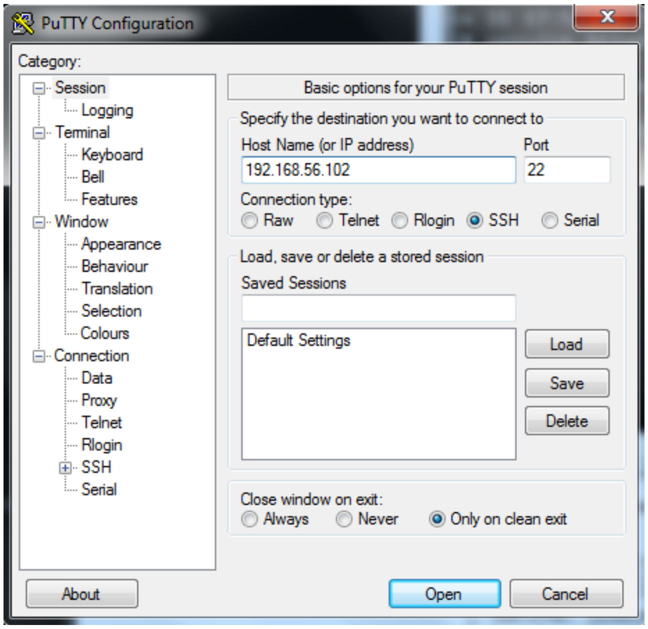
1. We choose Linux operating system with our system because the Linux is a mainstay for deploying web servers and is evolving from handling basic file, print, and utility workloads to running mission-critical applications and databases. As Linux grows in importance in terms of value to the business, the need to manage Linux environments to high standards of service quality , availability, security, and performance becomes an essential requirement for business success [61].

2. We create a virtual machine(VM) with the Kernel-based Virtual Machine (KVM) software that is built into the operating system. We use the Oracle VM Virtualbox which is a server virtualization based on multi operating system, easy and powerful with CentOS Linux.

3. We install the CentOS release 6.4 with kernel 2.6.32-358.14.1.e16.x86-64 Linux operating system.

4. We install free RADIUS 2.1.12 and use putty.exe configuration tool to open two sessions as shown in figure 4.12 and choose the connection type to be SSH on port 22. SSH is a secure shell against eavesdropping and encrypt data during traffic.

5. In putty configuration tool, we put IP address for Centos Linux is 192.168.56.102, and put IP address for Windows is192.168.56.101.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.56.102

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 4.12: Putty configuration

6. We use a set of services with ELSBOT namely:
   - RADIUS service.
   - Apache service
   - Mysql service

   Through LAMP module in CentOS Linux

   - Apache service
   - Mysql service

   Through XAMPP module in Windows

Based on the previous services, we must install the two following modules:

a) XAMPP module for windows. XAMPP is a tool that allows website designers to test the work on the computers without any access to the internet.

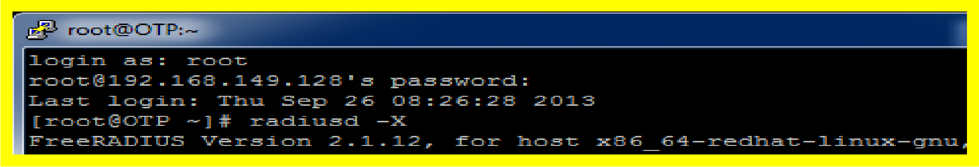
b) LAMP module for CentOs Linux environment, Lamp module is Linux provider service and allows website designers to test the work on the computers without any access to the internet.

4.6 Applying ELSBOT

After preparing the ELSBOT environment, we present three directions to apply our ELSBOT. Firstly, we run the required servers that support the implementation process for the ELSBOT. Secondly, we offer the web platform for our system through showing all web pages in this system such as sign-in, sign-up, OTP and etc. Finally, we introduce the OTP application in this system. Our OTP application provides three OTP techniques to generate the OTP for the system users.

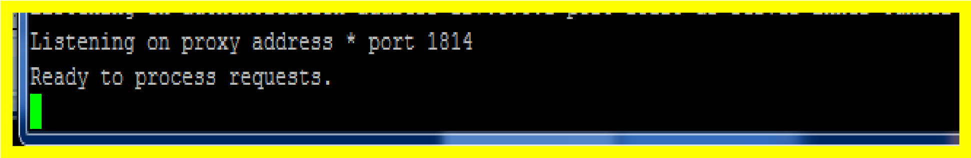
4.6.1 Running Servers

a) To run the RADIUS server, we enter the user name and password for RADIUS and then type the command " *radiusd –x* "as shown in figure 4.13 . Thus, the RADIUS server is a ready to response any request from the user as shown in figure 4.14



```
login as: root
root@192.168.149.128's password:
Last login: Thu Sep 26 08:26:28 2013
[root@OTP ~]# radiusd -X
FreeRADIUS Version 2.1.12, for host x86_64-redhat-linux-gnu,
```

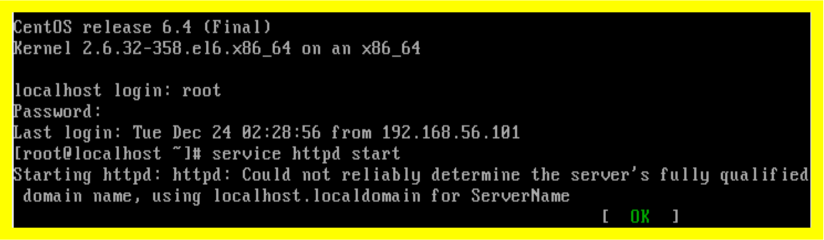
Figure 4.13: Execute RADIUS server with *radiusd–x* command



```
Listening on proxy address * port 1814
Ready to process requests.
```

Figure 4.14: The server waits any request

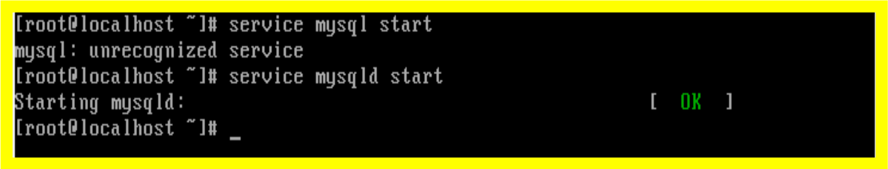
b) We run the LAMP server for CentOS Linux through starting the httpd service as shown in figure 4.15 , and then we start the mysqld service as shown in Figure 4.16



```
CentOS release 6.4 (Final)
Kernel 2.6.32-358.el6.x86_64 on an x86_64

localhost login: root
Password:
Last login: Tue Dec 24 02:28:56 from 192.168.56.101
[root@localhost ~]# service httpd start
Starting httpd: httpd: Could not reliably determine the server's fully qualified
 domain name, using localhost.localdomain for ServerName
                                                           [  OK  ]
```

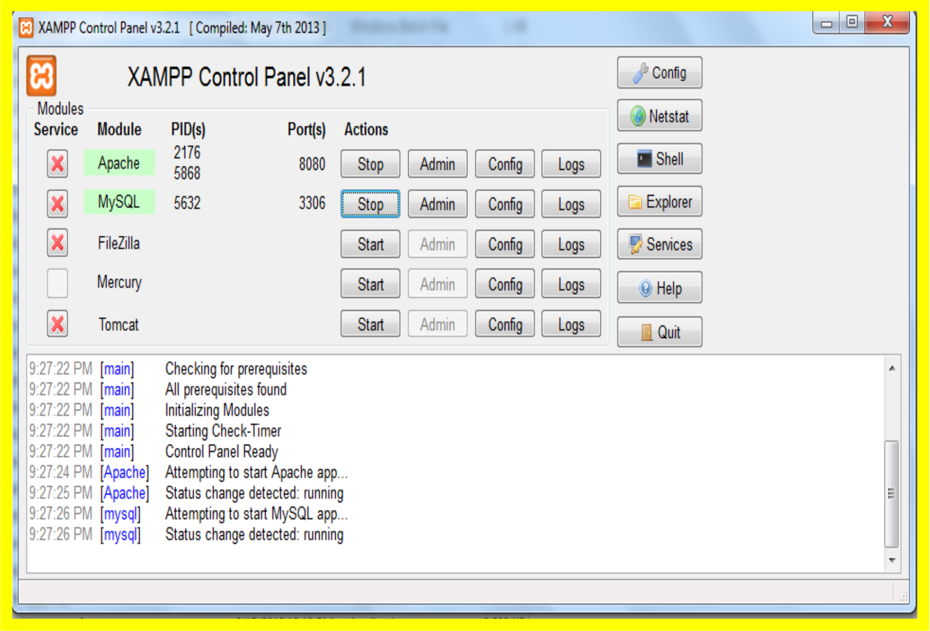
Figure 4.15: Start the *httpd* service of LAMP server



```
[root@localhost ~]# service mysql start
mysql: unrecognized service
[root@localhost ~]# service mysqld start
Starting mysqld:                                           [  OK  ]
[root@localhost ~]# _
```

Figure 4.16: Start the *Mysql* service of LAMP server

c) Finally, we run the XAMPP server version 3.2.1 for windows to start the apache service and the mysql service as shown in figure 4.17



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

Modules

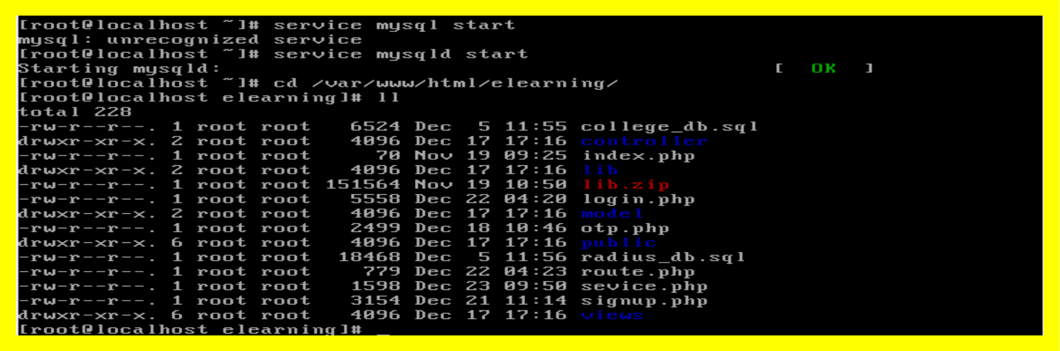
Service	Module	PID(s)	Port(s)	Actions			
☒	Apache	2176 5868	8080	Stop	Admin	Config	Logs
☒	MySQL	5632	3306	Stop	Admin	Config	Logs
☒	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
☒	Tomcat			Start	Admin	Config	Logs

```
9:27:22 PM  [main]    Checking for prerequisites
9:27:22 PM  [main]    All prerequisites found
9:27:22 PM  [main]    Initializing Modules
9:27:22 PM  [main]    Starting Check-Timer
9:27:22 PM  [main]    Control Panel Ready
9:27:24 PM  [Apache]  Attempting to start Apache app...
9:27:25 PM  [Apache]  Status change detected: running
9:27:26 PM  [mysql]   Attempting to start MySQL app...
9:27:26 PM  [mysql]   Status change detected: running
```

Figure 4.17 : Start the *Apache and Mysql* services of XAMPP server

4.6.2 ELSBOT Web-Platform

In this subsection, we present the web pages of ELSBOT. These web pages of this system are designed using PHP language (see Appendix D for the associated source code). We can find the databases and all web pages for our system after writing the command " cd /var/www/html/elearning " and then writing the command " ll "which is display the contents of the elearning directory as shown in figure 4.18

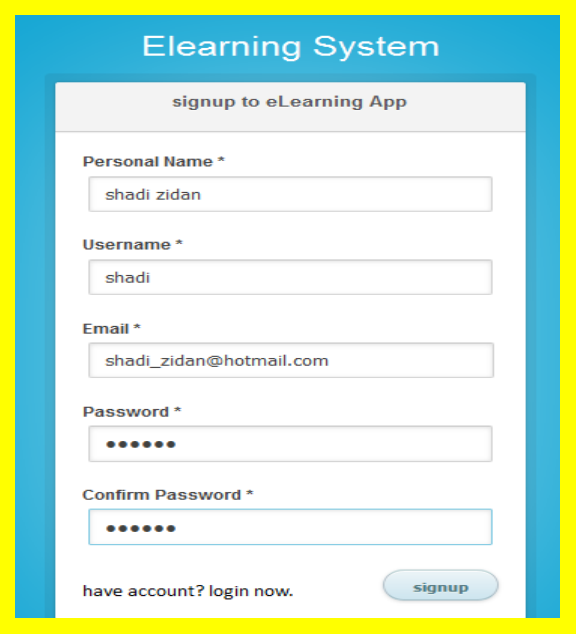


```
[root@localhost ~]# service mysql start
mysql: unrecognized service
[root@localhost ~]# service mysqld start
Starting mysqld:                                           [  OK  ]
[root@localhost ~]# cd /var/www/html/elearning/
[root@localhost elearning]# ll
total 228
-rw-r--r--. 1 root root   6524 Dec  5 11:55 college_db.sql
drwxr-xr-x. 2 root root   4096 Dec 17 17:16 controller
-rw-r--r--. 1 root root     70 Nov 19 09:25 index.php
drwxr-xr-x. 2 root root   4096 Dec 17 17:16 lib
-rw-r--r--. 1 root root 151564 Nov 19 10:50 lib.zip
-rw-r--r--. 1 root root   5558 Dec 22 04:20 login.php
drwxr-xr-x. 2 root root   4096 Dec 17 17:16 model
-rw-r--r--. 1 root root   2499 Dec 18 10:46 otp.php
drwxr-xr-x. 6 root root   4096 Dec 17 17:16 public
-rw-r--r--. 1 root root  18468 Dec  5 11:56 radius_db.sql
-rw-r--r--. 1 root root    779 Dec 22 04:23 route.php
-rw-r--r--. 1 root root   1598 Dec 23 09:50 sevice.php
-rw-r--r--. 1 root root   3154 Dec 21 11:14 signup.php
drwxr-xr-x. 6 root root   4096 Dec 17 17:16 views
[root@localhost elearning]# _
```

Figure 4.18: Command ll displays all pages and databases of ELSBOT

(1) Sign-up page

If the user is not exist in our system, this user will register his information through the signup page of ELSBOT. The sign-up page contains a set of the required text fields such as personal name, username, email, static password should be more than six characters and confirm password as shown in figure 4.19



Elearning System

signup to eLearning App

Personal Name *
shadi zidan

Username *
shadi

Email *
shadi_zidan@hotmail.com

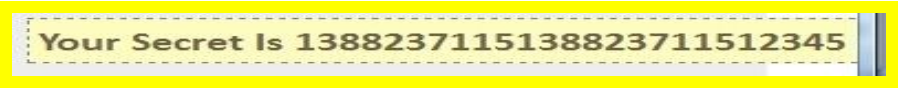
Password *
••••••

Confirm Password *
••••••

have account? login now.

Figure 4.19: Sign-up page

After registering, the system sends the secret key only once to OTP application of user. This secret key consists of 25 characters which is encrypted by MD5 as shown in figure 4.20

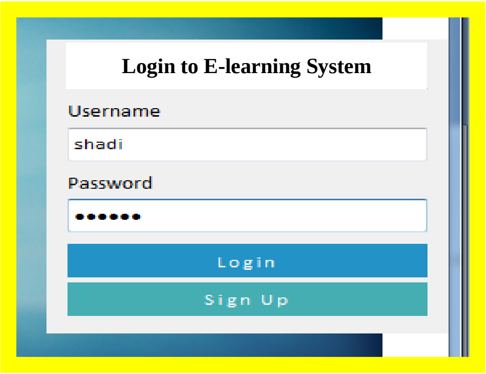


Your Secret Is 1388237115138823711512345

Figure 4.20: A secret Key for example

(2) Sign-in page

In this page, only the system user can login to the system after entering his name and his password as shown in figure 4.21 . If the user data is correct, the system transmits the user to the OTP page that will be prepared in the next.



Login to E-learning System

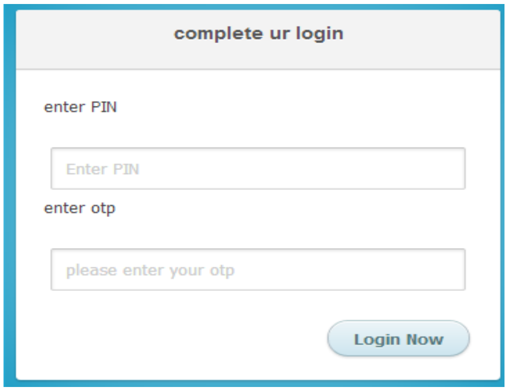
Username: shadi

Password: ●●●●●●

Figure 4.21: Sign-in page

(3) OTP Page

The OTP page of ELSBOT is the most important page in our system because it adds more security and protection on this system. The OTP page as shown in figure 4.22, we show two fields such as PIN and OTP that its values are entered by the system user. The OTP value is taken from the OTP application which generates the OTP based on the type of OTP technique selected by the administrator. The authentication server checks the user PIN and OTP and sends a response either acceptance or rejection.



complete ur login

enter PIN

Enter PIN

enter otp

please enter your otp

Figure 4.22:OTP page

(4) User Page

After checking the user PIN and OTP, the AS sends a response to system. In the case of acceptance, the system transmits the user to his page. The user page contains all operations permitted for this user. For example, The authorized operations for this user are add students and courses, view students and courses, and register courses for students. The figure4.23 shows the page of courses management to add or update the college courses.

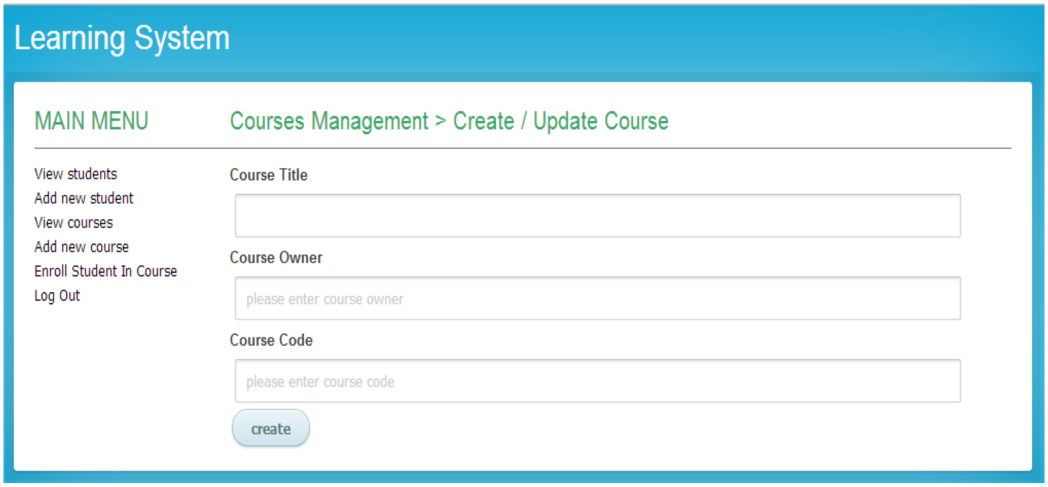
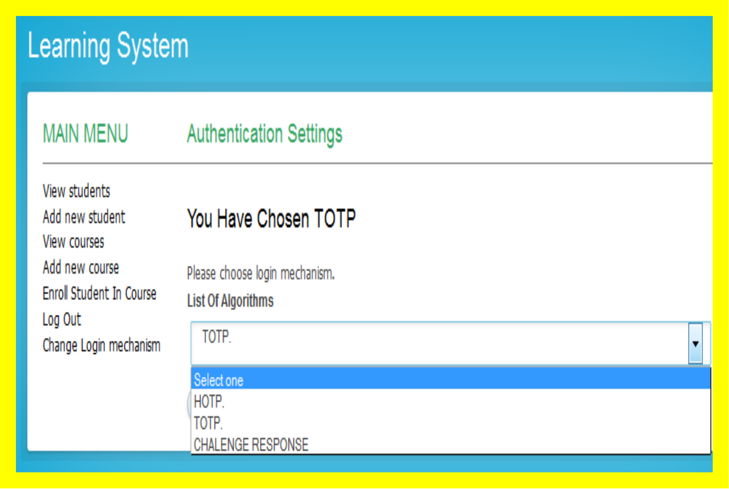


Figure 4.23: Courses management page

(5) Admin Page

Administrator is a user account who can change the login mechanism and perform all system operations. Administrator determines the OTP technique for login process to system users. Thus, the user can login to system after entering the generated OTP of the chosen OTP technique by administrator as shown in figure 4.24



You Have Chosen TOTP

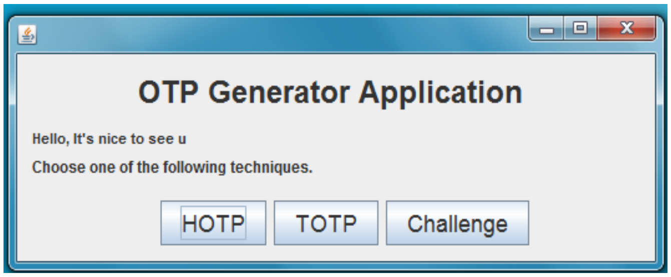
Please choose login mechanism.

List Of Algorithms: TOTP.

Figure 4.24: Admin page

4.6.3 OTP Application

We present the OTP application of ELSBOT as shown in figure 4.25 . This application offers three OTP techniques which generate the OTP in order to provide a secure system to our organizations. We apply this application using Java language because it has a number of features like robust, security and built-in networking (see Appendix A for the associated source code).


OTP Generator Application

Hello, It's nice to see u

Choose one of the following techniques.

Figure 4.25: OTP Application

In this subsection, we show the OTP generation process for each of the three OTP techniques in our application as shown below:

(1) TOTP Technique

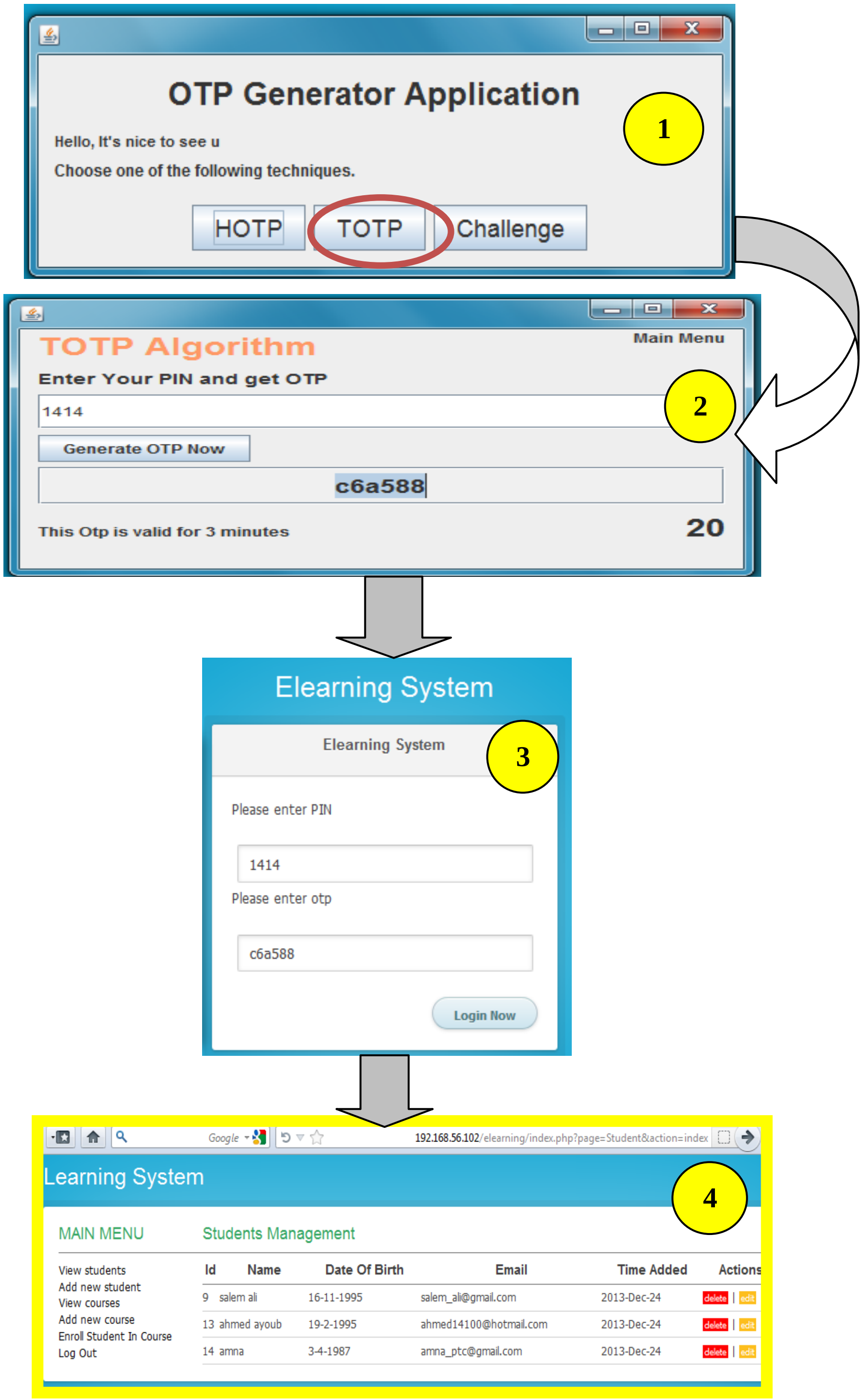
We present a set of steps for TOTP technique in our application. These steps illustrate the OTP generation process in this technique as shown in figure 4.26 . In this technique, the time is the basic factor in the OTP generation process. The generated OTP of this technique effectively remains for up to 3 minutes. Clearly, the user can not enter the generated OTP of this technique in our system if the time of the generated OTP is expired. Thus, the user must generate another OTP in order to login to system. The steps of TOTP technique in OTP application are shown below:

Step 1: The user clicks on the button of TOTP technique in this application in the event that this technique is chosen by the system administrator in the ELSBOT.

Step 2: The user enters the PIN which contains 4 different characters and then clicks on the button “ generate OTP Now”. Thus, the application prints the generated OTP in a text. The user copies the generated OTP from our application and then returns to the OTP page of our system at time 20 minute.

Step 3: In the OTP page, the user enters the generated OTP in the OTP field and then clicks on the button "Login Now".

Step 4: The ELSBOT system checks the values of the elements like the PIN, the secret key and the last OTP. If the elements values are matched by the authentication server, the system will open the specific page for the user. Otherwise, the system will display an error message.



OTP Generator Application

Hello, It's nice to see u

Choose one of the following techniques.

TOTP Algorithm

Enter Your PIN and get OTP

1414

c6a588

This Otp is valid for 3 minutes

20

Elearning System

Elearning System

Please enter PIN

1414

Please enter otp

c6a588

192.168.56.102/elearning/index.php?page=Student&action=index

Learning System

Students Management

Id	Name	Date Of Birth	Email	Time Added	Actions
9	salem ali	16-11-1995	salem_ali@gmail.com	2013-Dec-24	delete | edit
13	ahmed ayoub	19-2-1995	ahmed14100@hotmail.com	2013-Dec-24	delete | edit
14	amna	3-4-1987	amna_ptc@gmail.com	2013-Dec-24	delete | edit

Figure 4.26 : TOTP Steps

(2) CROTP Technique

In the CROTP technique, We provide a set of steps to show the OTP generation process in our OTP application as shown in figure4.27 . The authentication server sends a random PIN to user in the OTP page of our system. This PIN consists of 4 characters randomly. The system user enters this PIN in the OTP application and then work generation for the OTP. The steps of CROTP technique are namely:

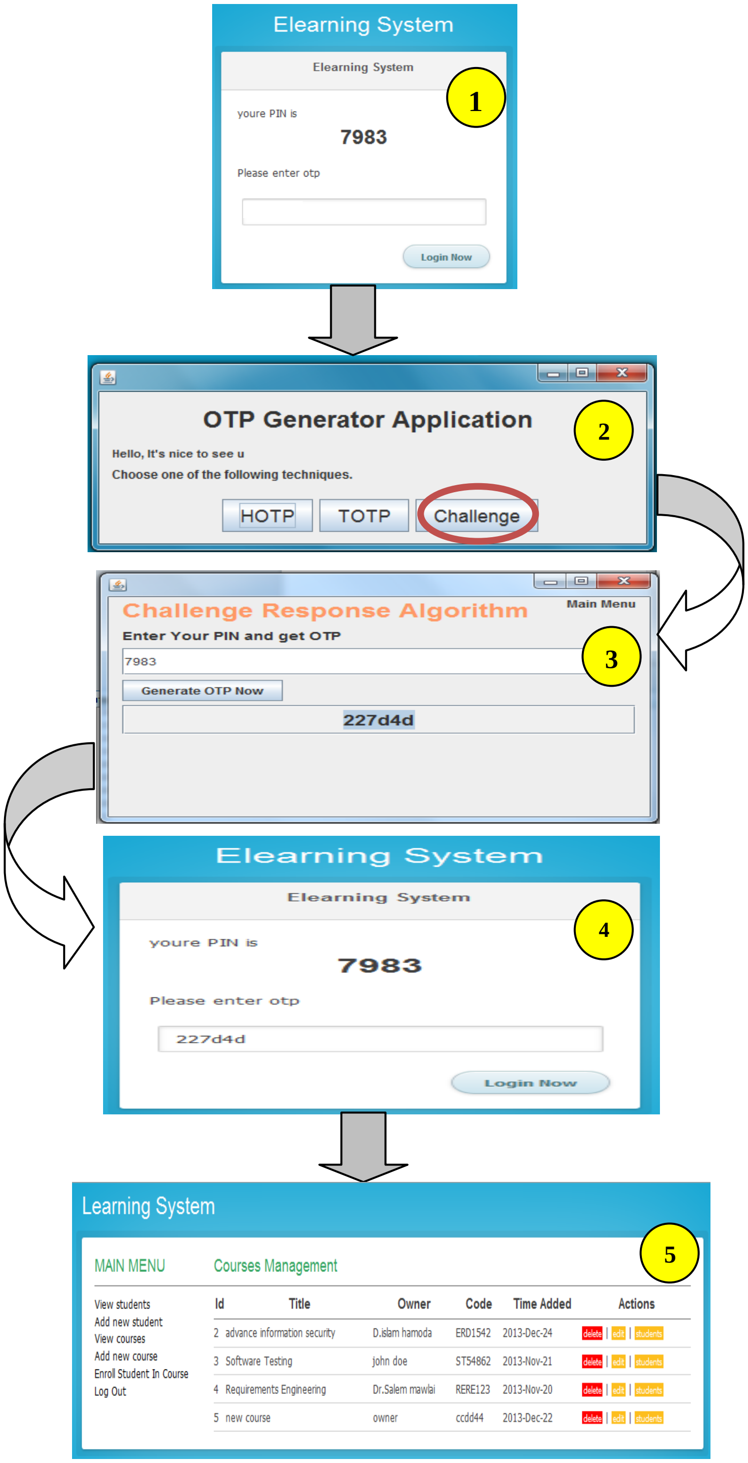
Step 1: Firstly, the system user enters his user name and his password. The system transmits the user to the OTP page which displays the random PIN sending by authentication server. The user copies this PIN and then opens the OTP application in order to get the generated OTP.

Step 2: In this application, the user clicks on the button of CROTP technique in this application in the event that this technique is chosen by the system administrator in the ELSBOT.

Step 3: The user enters the system random PIN and then clicks on the button " generate OTP Now". Thus, the application prints the generated OTP in a text and then the user copies the generated OTP.

Step 4: The user returns to the OTP page of our system and enters the generated OTP in the OTP field and then clicks on the button "Login Now".

Step 5: The ELSBOT system checks the values of the elements like the random PIN, the secret key and the last OTP. If the elements values are matched by the authentication server, the system will open the specific page for the user. Otherwise, the system will display an error message.



Elearning System

youre PIN is

7983

Please enter otp

OTP Generator Application

Hello, It's nice to see u

Choose one of the following techniques.

Challenge Response Algorithm

Enter Your PIN and get OTP

7983

227d4d

Elearning System

youre PIN is

7983

Please enter otp

227d4d

Learning System

MAIN MENU

- View students
- Add new student
- View courses
- Add new course
- Enroll Student In Course
- Log Out

Courses Management

Id	Title	Owner	Code	Time Added	Actions
2	advance information security	D.islam hamoda	ERD1542	2013-Dec-24	delete | edit | students
3	Software Testing	john doe	ST54862	2013-Nov-21	delete | edit | students
4	Requirements Engineering	Dr.Salem mawlai	RERE123	2013-Nov-20	delete | edit | students
5	new course	owner	ccdd44	2013-Dec-22	delete | edit | students

Figure 4.27: CROTP Steps

(3) HOTP Technique

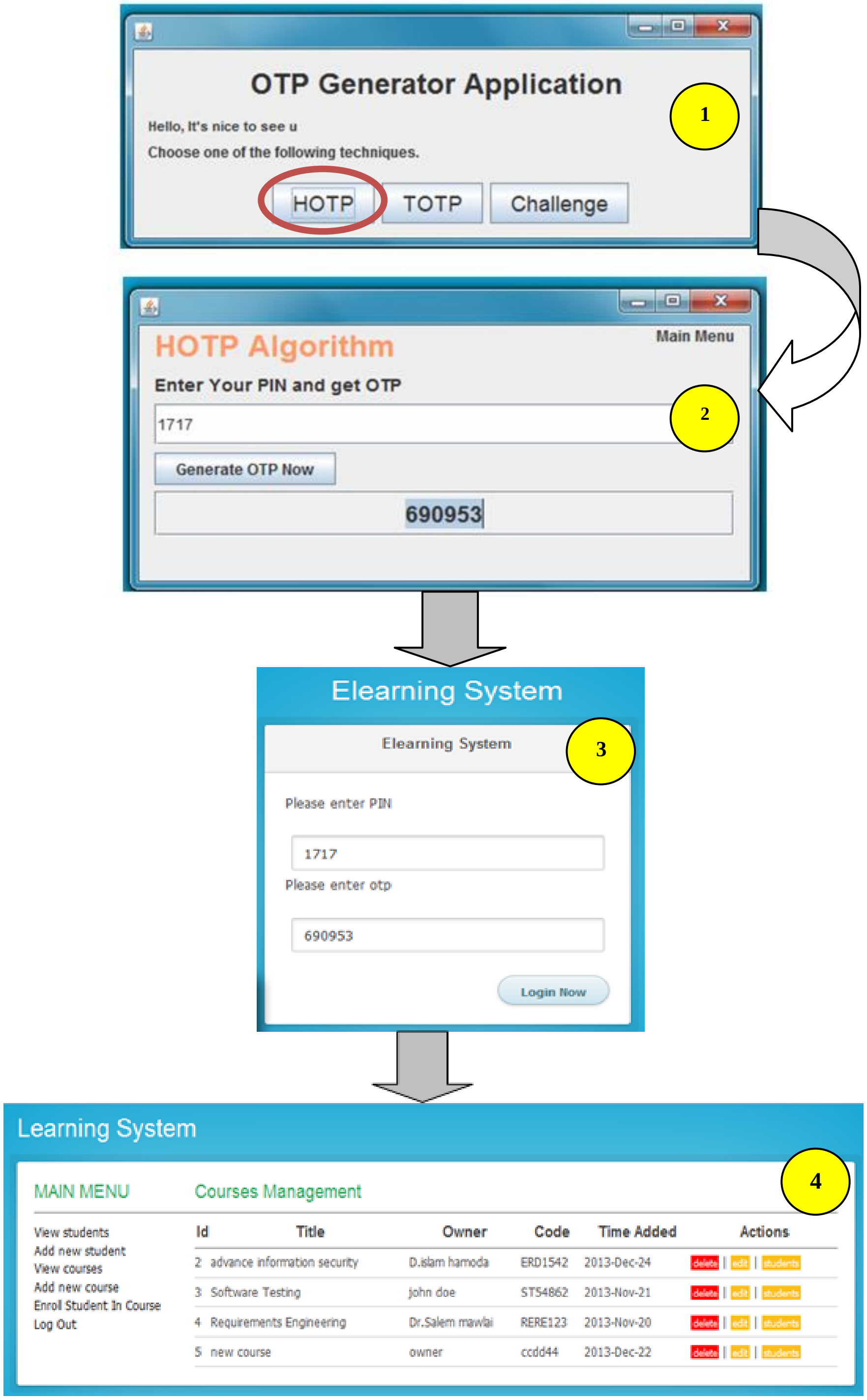
To show the OTP generation process in the HOTP technique in our OTP application, we present a set of steps as shown in the figure 4.28 . In this technique, the count is the basic factor in the OTP generation process. The generated OTP is an encrypted key which is a substring taken from the secret key, PIN and the last count. The steps of HOTP in our application are namely:

Step 1: The user clicks on the button of HOTP technique in this application in the event that this technique is chosen by the system administrator in the ELSBOT.

Step 2: The user enters the PIN which contains 4 different characters and then clicks on the button " generate OTP Now". Thus, the application prints the generated OTP in a text. The user copies the generated OTP from our application and then returns to the OTP page of our system.

Step 3: In the OTP page, the user enters the generated OTP in the OTP field and then clicks on the button "Login Now".

Step 4: The ELSBOT system checks the values of the elements like the PIN, the secret key and the last OTP. If the elements values are matched by the authentication server, the system will open the specific page for the user. Otherwise, the system will display an error message.



Hello, It's nice to see u

Choose one of the following techniques.

Enter Your PIN and get OTP

1717

690953

Please enter PIN

1717

Please enter otp

690953

View students
Add new student
View courses
Add new course
Enroll Student In Course
Log Out

Id	Title	Owner	Code	Time Added	Actions
2	advance information security	D.islam hamoda	ERD1542	2013-Dec-24	delete | edit | students
3	Software Testing	john doe	ST54862	2013-Nov-21	delete | edit | students
4	Requirements Engineering	Dr.Salem mawlai	RERE123	2013-Nov-20	delete | edit | students
5	new course	owner	ccdd44	2013-Dec-22	delete | edit | students

Figure 4.28: HOTP steps

4.7 Replay attack

OTP authentication methods are vulnerable to certain kinds of attacks such as replay attacks [3], [5], [6], [7],[26],[35][69][70]. Replay attacks have been discussed. We generalize the definition of a replay attack as: *replay of messages from a different context into the intended context, thereby fooling the honest participants into thinking they have successfully completed the protocol run* [70] . We have to work on the redirect of service to create scripting for replay attacks , but there were some problems during the service call because browsers like Firefox and Google Chrome did not support redirect service . So , we used the chrome extension application which Working successfully with browsers in order to take feature to work replay attacks. This feature uses malicious JavaScript to take data from server attack and sent back to attacker .

Chrome extension application can listen to any request processed to user system based on jquery library to implement data replay using command *" jQuery.parseJSON(data)"* . Attacker retains data stolen in console log *"console.log(data);*" where chrome console tool continues to operate connect with the user, then the attacker captures all the required data (username/static password)), PIN and OTP " *$("#otp").attr("value",data.otp); $("#pin").attr("value",data.pin); $(".data")*" . Full implementation for replay attack found in appendix C.

4.7.1 Attacker Database

Database for attacker includes two tables, the first table called *users* that includes username of all users with their passwords that have been captured through sniffing on ports or implement malicious script ,etc . The second table is *OTP* table that includes Id PIN and OTP that stolen from entering the users of the system. Attacker database is shown in figure 4.30

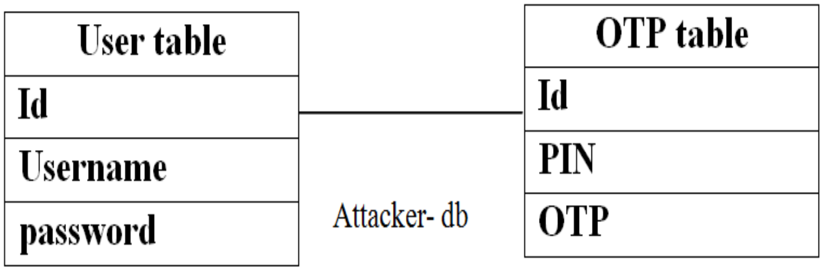


Figure 4.29: Attacker database

4.7.2 Replay attack components

We show describe replay attack architecture in figure 4.30, then we offer set of steps showing in table 4.6 . We provide the functional architecture of replay attack that has five components namely:

1. **User :** User is a person (victim) who requests access to the system services.
2. **System :** The System is a website that will design to provide services to end users.
3. **RADIUS server** : The RADIUS server checks the user information (username and password) which is entered in the sign-in form. The RADIUS server has a database to store the user data .
4. **Authentication server (AS)** AS is responsible of the second phase in our system. After The system user enters the PIN and the generated OTP in the OTP page, the authentication server will check the secret key, PIN and last OTP and then will send a response to the system.
5. **Attacker server** : Where it not send for malicious script else through the server, the attacker confirm with malicious server , where Attacker server sends malicious software script to user to listen on any request processed to system.
6. **Attacker**: is malicious user can sniff to network traffic and steal user data . Attack depends on the reception of incoming requests to the server continuously without interruption.
7. **Internet browser** , through browser can attacker running chrome extension to apply the attack. Table 4.6 show components with their modules and libraries which used in attack.

- We apply OTP techniques of ELSBOT system against replay attack . The work was conducted on a local network (LAN) within the computer science lab at Technical Palestine College (PTC) , scenario applied with six components and we uses set of modules and libraries shown in table 4.5

Table 4.5: Replay attack components

#	Components	Module	library
1	Authentication server	XAMPP	SOAP
2	RADIUS server	LAMPP	SOAP
3	Attacker server	XAMPP	jquery
4	Attacker computer	XAMPP	jquery
5	The user (Victim)	XAMPP/ LAMPP	
6	Internet browser	Google Chrome	Chrome Consol

4.7.3 Replay attack architecture

In this section, we present the replay attack architecture, In the replay attack, the attacker server is able to capture the username and password from user within a malicious script which is recording the private information for users, then attacker server sends the username & password to attacker who is stealing and trying to authenticate username and password in the system. This attacker stills to listen and take any information from user in each traffic over the network . we show describe replay attack architecture in figure 4.30, then we offer set of steps showing in table 4.5

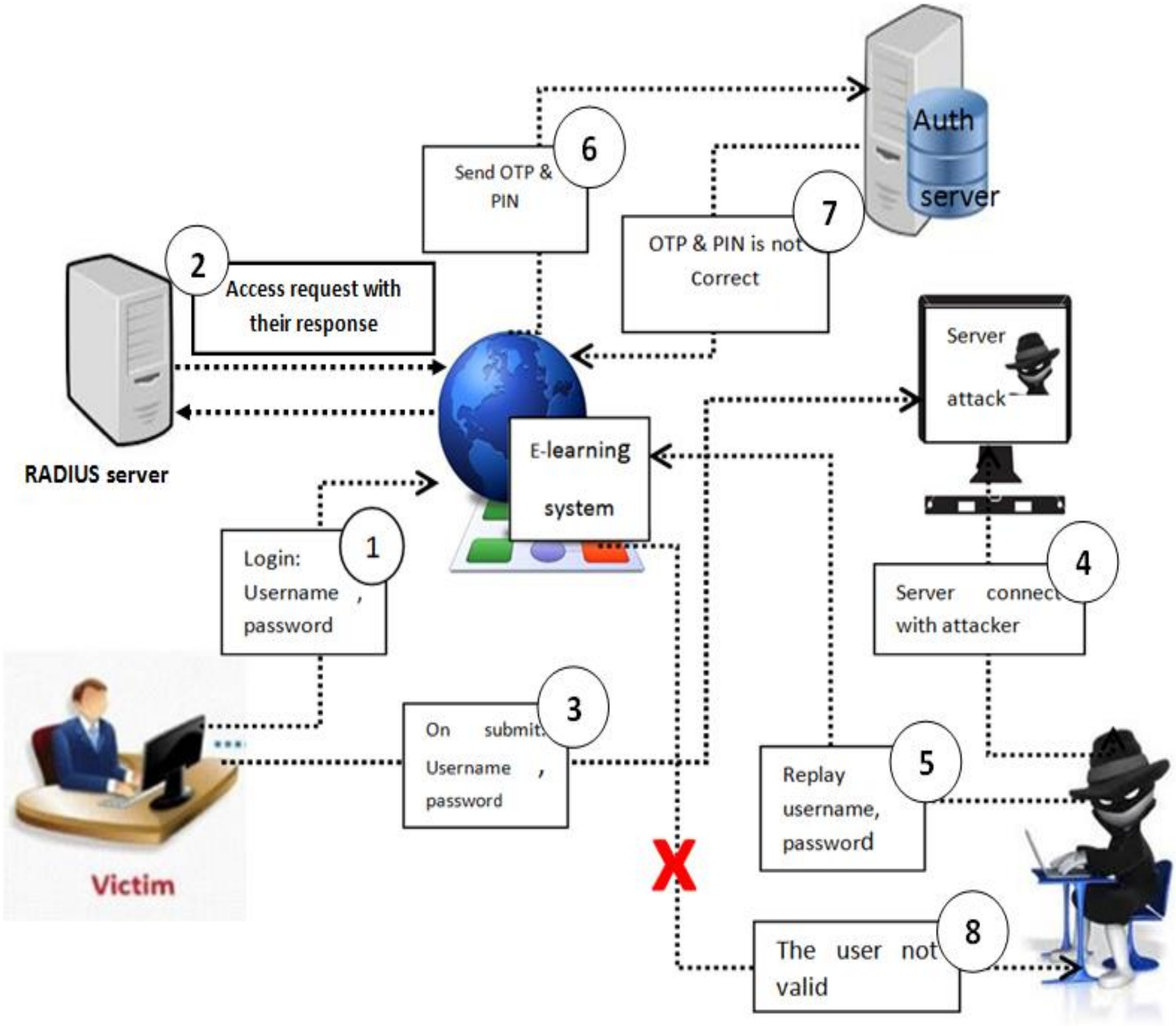

Auth
server
6
Send OTP & PIN
7
OTP & PIN is not Correct
2
Access request with their response
RADIUS server
Server attack
E-learning system
1
Login: Username , password
4
Server connect with attacker
3
On submit: Username , password
5
Replay username, password
Victim
8
The user not valid

Figure 4.30: Replay attack architecture

Table 4.5: Steps for replay attack architecture

Step#	Description
1.	The user enters the username and password in the login screen.
2.	The system sends user data to the RADIUS server to authenticate the user . the RADIUS server verifies the username and password and sends a response either accept or reject.
3.	Attack depends on the reception of incoming requests to the server continuously without interruption.
4.	Attacker server sends malicious software script to user to listen on any request processed to system.
5.	Attacker steals data and last request , then sends these data for ELSBOT.
6.	Attacker returns sending data for ELSBOT through running malicious script.
7.	ELSBOT receives stolen data from attacker and replays for AS.
.8	AS implements DB query and verifies of PIN, OTP, secret key, lastOTP and response with " the user is not authenticate", replay attack scenario for Replay attack architecture shown in figure(4.29).

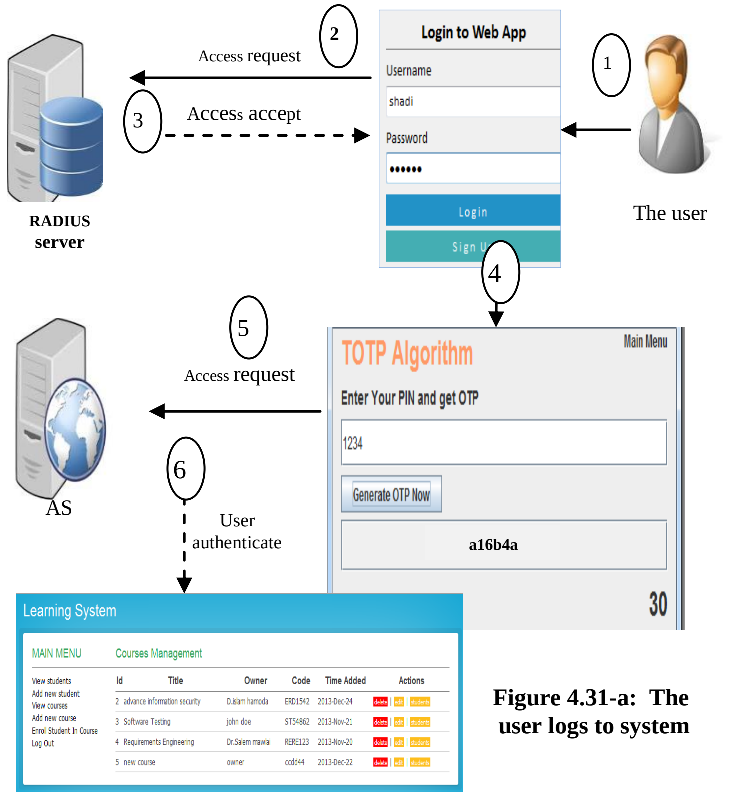
4.7.4 Applying the replay attack on ELSBOT

In this subsection , we present three scenarios to show the whole process of the replay attack on our system. These scenarios are used to apply this attack for each of the three OTP techniques in the ELSBOT .

4.7.4.1 First scenario : TOTP techniques , in this technique , we present two parts , the first part when user login to system with TOTP technique in 30 minute as shown in figure4.31-a . In second part when attacker steal data in 185 minute and replay to system as shown in figure 4.31-b

The user logs to system

(1) The user logs to ELSBOT with username and password
(2) RADIUS protocol is verified of username and password
(3) RADIUS protocol responds with Accepting.
(4) OTP page opens with run application (TOTP technique), while user input PIN and OTP in 30 minute.
(5) The system sends PIN and OTP to The AS.
(6) The AS responds with the user is valid and authenticated ,the user can be access to system.



Username: shadi

Password: ••••••

Enter Your PIN and get OTP: 1234

a16b4a

30

Id	Title	Owner	Code	Time Added	Actions
2	advance information security	D.islam hamoda	ERD1542	2013-Dec-24	delete | edit | students
3	Software Testing	john doe	ST54862	2013-Nov-21	delete | edit | students
4	Requirements Engineering	Dr.Salem mawlai	RERE123	2013-Nov-20	delete | edit | students
5	new course	owner	ccdd44	2013-Dec-22	delete | edit | students

Figure 4.31-a: The user logs to system

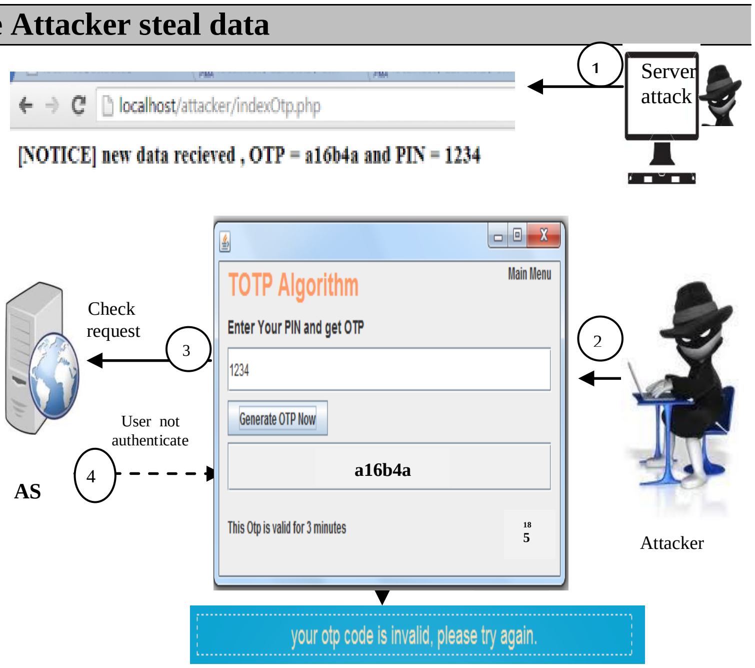
The Attacker steal data

(1)The attacker server runs malicious script on chrome extension to apply chrome console and connect with attacker .

(2)The attacker implements attack on user through chrome console; the attacker in 158 minutes logs to the system with storm account "shadi". The script gets PIN and OTP once the user logs to system.

(3)The AS verifies from own database.

(4)The AS responses with the user is invalid, because the secret key varies constantly. Additionally, it enables authentication without having to expose the secret, while secret + PIN + OTP + last OTP .



[NOTICE] new data recieved , OTP = a16b4a and PIN = 1234

Enter Your PIN and get OTP: 1234

a16b4a

This Otp is valid for 3 minutes 18 5

your otp code is invalid, please try again.

Figure 4.31-b : The attacker logs to system

4.7.4.2 Second scenario : CROTP technique, in this technique , we offer two parts, firstly: when user login to system with CROTP technique shown in figure 4.32-a . Secondly, when the attacker steal user data then replay to system as shown in figure 4.32-b

The user logs to system

(1). The user logs to ELSBOT system with username and password

(2). The system sends request to RADIUS protocol that is verifies from username and password .

(3). RADIUS protocol responds with 'Access-Accept" while logs information match in database.

(4). OTP page opens while server sends random challenge pin to user.

(5). The user sends OTP to AS for verification.

(6). The AS responds with the user is valid and authenticated while PIN and OTP matches in auth_sql.db shown in figure (5.10-a).

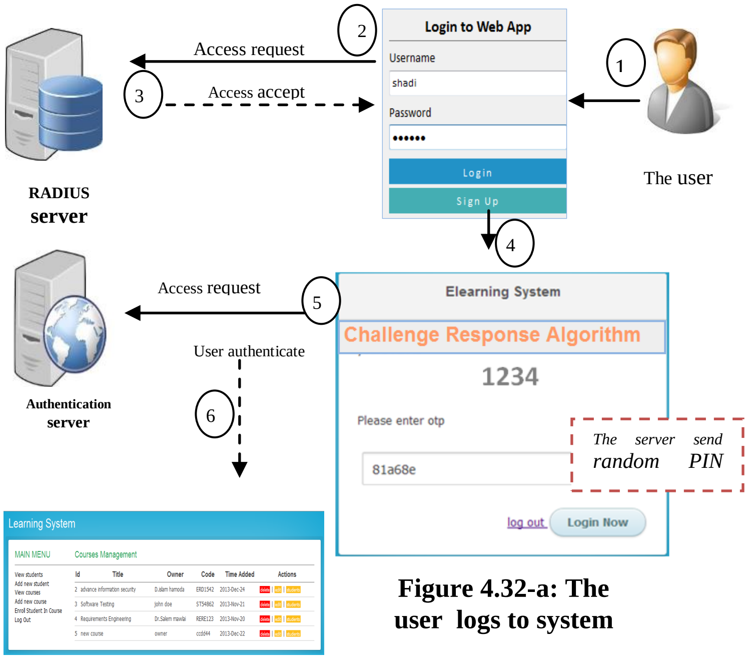


Figure 4.32-a: The user logs to system

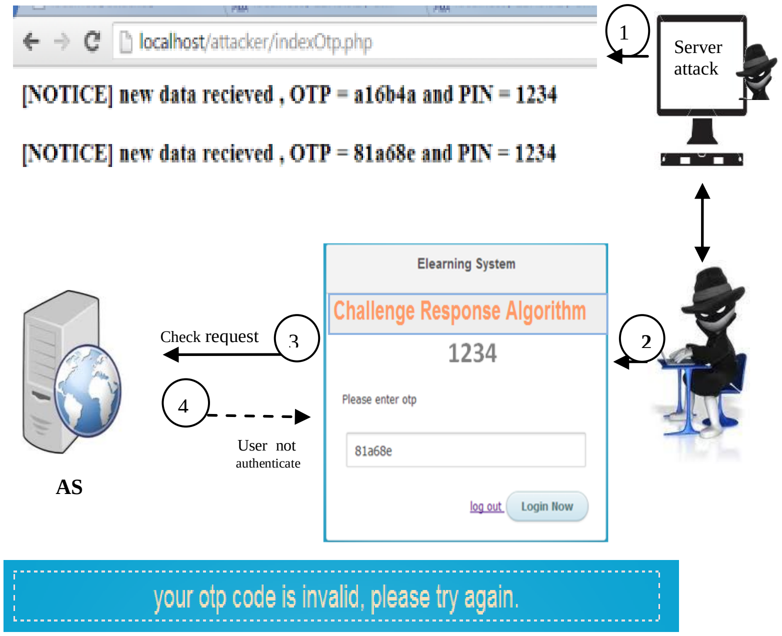
The attacker logs to system

(1) The attacker server runs malicious script on chrome extension where he used chrome console to run script and connects with attacker .

(2) The attacker implements attack and logs in system with storm account "shadi", the script gets PIN and OTP once the user logs to system.

(3) The AS verifies from own database.

(4) The AS response with the user is invalid, because the secret key varies constantly. Additionally, it enables authentication without having to expose the secret. While secret , PIN , OTP and lastOTP be listed in database .



```
[NOTICE] new data recieved , OTP = a16b4a and PIN = 1234

[NOTICE] new data recieved , OTP = 81a68e and PIN = 1234
```

your otp code is invalid, please try again.

Figure 4.32-b: attacker logs to system

4.7.4.3 Third scenario : HOTP techniques ,we offer two parts, firstly when user login to system with HOTP technique shown in figure 4.33-a . Secondly when attacker **runs malicious script** to steal user data then replay to system as shown in figure 4.33-b

The user logs to system

(1) The user logs to ELSBOT system with username and password

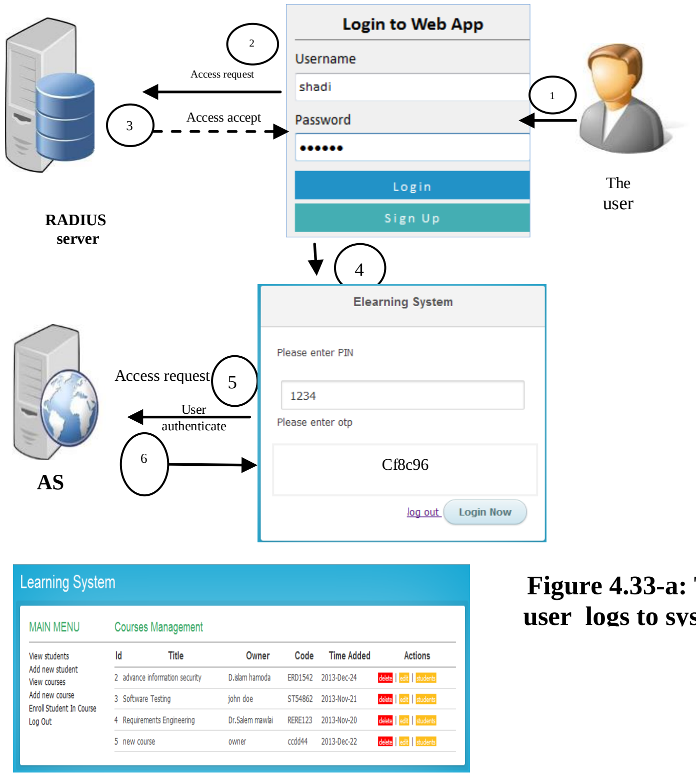
(2) The RADIUS protocol verified from username and password in radius_sql.db

(3) RADIUS protocol responds with Access-Accept .

(4) OTP page opens with run application (HOTP)

(5) The user sends PIN and OTP to AS for verification.

(6) The AS responds with "the user is valid and authenticated" **.**



Login to Web App

Username: shadi

Password: ••••••

Elearning System

Please enter PIN: 1234

Please enter otp: Cf8c96

Learning System

MAIN MENU

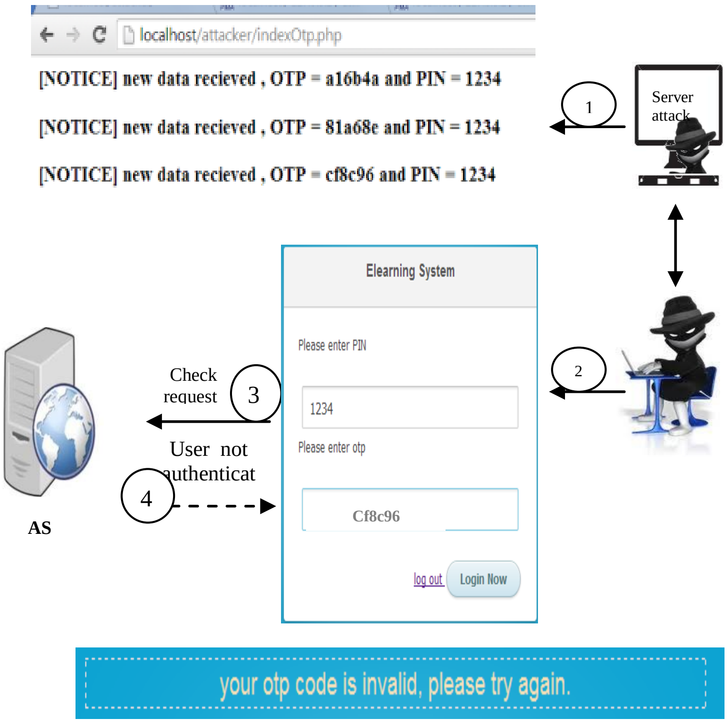
Courses Management

Id	Title	Owner	Code	Time Added	Actions
2	advance information security	D.islam hamoda	ERD1542	2013-Dec-24	delete | edit | students
3	Software Testing	john doe	ST54862	2013-Nov-21	delete | edit | students
4	Requirements Engineering	Dr.Salem mawlai	RERE123	2013-Nov-20	delete | edit | students
5	new course	owner	ccdd44	2013-Dec-22	delete | edit | students

Figure 4.33-a: The user logs to system

The attacker logs to system

1. The attacker server runs malicious script on chrome extension .
2. The aattacker implements attack on user through chrome console and logs to system.
3. The AS verifies from own database.
4. **4.** The AS responses with the user is invalid, because the secret key varies constantly. Additionally, it enables authentication without having to expose the secret, while secret + PIN + OTP + last OTP be listed in database .



localhost/attacker/indexOtp.php

```
[NOTICE] new data recieved , OTP = a16b4a and PIN = 1234

[NOTICE] new data recieved , OTP = 81a68e and PIN = 1234

[NOTICE] new data recieved , OTP = cf8c96 and PIN = 1234
```

Elearning System

Please enter PIN: 1234

Please enter otp: Cf8c96

your otp code is invalid, please try again.

Figure 4.33-b: The attacker logs to system

4.8 Conclusion

This chapter introduced the ELSBOT system for implementing the OTP techniques (TOTP,HOTP,CROTP) in RADIUS protocol to prevent the replay attacks and to increase data privacy and confidentiality. In this chapter, we provided the entire architecture of our system. In detail, we describe the specific architecture for each of the three OTP techniques. These OTP techniques is used to generate of OTP in order to protect user data from attackers. We provided the requirements of the ELSBOT and then prepared the appropriate environment for applying this techniques. In addition, we presented three authentication procedures for ELSBOT such as sign-in process, sign-up process, and OTP generation process. The ELSBOT offered the OTP application which generated the OTP for users to ensure the prevention of the replay attack. Finally, we offered a script for representing the process of replay attack in order to evaluate our ELSBOT against the replay attack.

Chapter 5

Results and Comparison

5.1 Introduction

This chapter draws four directions in order to evaluate our ELSBOT. Firstly, login scenario is presented through a different cases to show the login process phases for the system users. The second direction measures the average response time for authentication server at each OTP technique. The third direction offers the synchronization process at each of the three OTP techniques to measure CPU overhead and algorithm speed. The other important direction of this research evaluate our ELSBOT against the replay attack by applying an attacked script which tries to steal user data. Finally, this chapter draws a comparison between the three OTP techniques. This comparison is presented by considering a set of factors which are mentioned later.

5.2 User login scenario:

The user login scenario offers a set of cases for login process in our ELSBOT. In table 5.1, each case passes the system user in several stages in order to login to the ELSBOT. These cases are namely below:

Case 1 : The valid user

1. A user logins to our system by entering username and password.
2. The RADIUS server sends a response with accepting.
3. The system transmits the user to OTP page, then the user enters PIN and TOTP.
4. The system sends the PIN and OTP to authentication server (AS) which checks PIN, last OTP and secret key and then AS sends an acceptance response to system.
5. The authentication result of this case is succeed and the user can login to system .

Case 2: The missing user name

1. A user attempts to login to the system without entering your user name.
2. The RADIUS server sends a response with rejecting.
3. The system prevent this user from login to the system.

Case 3: Invalid password

1. A user logins to our system by entering valid username and invalid password.
2. The RADIUS server sends a response with rejecting.
3. The system does not response to user

Case 4: Invalid OTP

1. A user logins to our system by entering username and password.

2. The RADIUS server sends a response with accepting.
3. The system transmits the user to the OTP page, then the user enters valid PIN and invalid OTP.
4. The system sends the PIN and the invalid OTP to AS which verifies PIN, last OTP and secret key and then AS sends a response with rejecting.
5. The authentication result of this case is failure.

Case 5: Invalid PIN

1. A user logins to our system by entering valid username and password.
2. The RADIUS server sends a response with accepting.
3. The system transmits the user the OTP page, and the user enters invalid PIN and valid OTP.
4. The system sends the PIN and the OTP to AS which verifies PIN, last OTP and secret key and then AS sends a response with rejecting.
5. The authentication result of this case is failure.

Table 5.1: User Login Cases

	RADIUS server			***OTP Techniques***			***Authentication server***				
Cases	*Authentication request*	*Username*	*Password*	*TOTP*	*HOTP*	*CROTP*	*OTP*	*PIN*	*Last OTP*	*Secret key*	*Authentication Result*
C.1	***V***	***V***	***V***	***V***			***V***	***V***	***V***	***V***	*S*
C.2	***V***	*I*	*Na*	*Na*			*Na*	*Na*	*Na*	*Na*	*F*
C. 3	***V***	***V***	*I*	*I*			-	-	-	-	*F*
C.4	***V***	***V***	***V***	***V***			*I*	***V***	*I*	*I*	*F*
C. 5	***V***	***V***	***V***	***V***			***V***	*I*	*I*	*I*	*F*

V: Valid information; I: Invalid information; Na: Not applicable; S: authentication successful; F: authentication failed.

5.3 Server Response Time :

Server Response Time define with elapsed time between the end request or demand on a computer system by user and the beginning of server response. The (Computer Science) defined Server Response Time " The length of time taken by a system to respond to an instruction by server" [76].

To measure the server response time, ten different cases are offered for each of the three OTP techniques. These cases show the server response time for each request by the system user. Table 5.2 shows the user request time and its response time by the server. While Unix time stamp is a way to track time as a running total of seconds. This count starts at the Unix Epoch on January 1st, 1970 at UTC. Therefore, the unix time stamp is merely the number of seconds between a particular date and the Unix Epoch [74]. For example, the user request time for case 1 is equal to 1386261391 as timestamp. This case is equal to 12/05/13,4:36:31pm as current date.

Table 5.2 : Shows the user request time and server response time in different times.

Case #	Request Time	Response Time	Technique Type	Drift Second
1	1386261391	1386261393	CROTP	2
2	1386261499	138626151	CROTP	2
3	1386261925	1386261926	CROTP	1
4	1387639886	1387639887	CROTP	1
5	1387639978	1387639980	CROTP	2
6	1387642561	1387642563	CROTP	2
7	1387705748	1387705750	CROTP	2
8	1387705796	1387705797	CROTP	1
9	1387705710	1387705711	CROTP	1
10	1387705711	1387705711	CROTP	0
11	1387705812	1387705815	HOTP	3
12	1387806696	1387806697	HOTP	1
13	1387807011	1387807014	HOTP	3
14	1387807056	1387807058	HOTP	2
15	1387807100	1387807101	HOTP	1
16	1387809597	1387809599	HOTP	2
17	1387809598	1387809598	HOTP	0
18	1387870770	1387870772	HOTP	2
19	1387870777	1387870778	HOTP	1
20	1387870790	1387870762	HOTP	2
21	1388519419	1388519420	TOTP	1
22	1388519497	1388519499	TOTP	2
23	1388519572	1388519573	TOTP	1
24	1388519580	1388519582	TOTP	1
25	1388519585	1388519586	TOTP	1
26	1388519592	1388519593	TOTP	1
27	1388519635	1388519636	TOTP	1
28	1388519640	1388519642	TOTP	2
29	1388519670	1388519671	TOTP	2
30	1388519679	1388519680	TOTP	1

After showing these cases, the authors compute the average server response time for each technique. In table 5.3, the results show that the average response time for HOTP is equal to 1.7500 sec., the average response time for TOTP is equal to 1.2500 sec. and the average response time for CROTP is 1.6250 sec. Thus, we find that the average response time of TOTP is lower than CROTP and HOTP. The figure 5.1 illustrates the average server response time for the three OTP techniques.

Table 5.3:Average Response Time

OTP techniques	CROTP	HOTP	TOTP
Average response time	1.6250 sec.	1.7500 sec.	*1.2500 sec.*

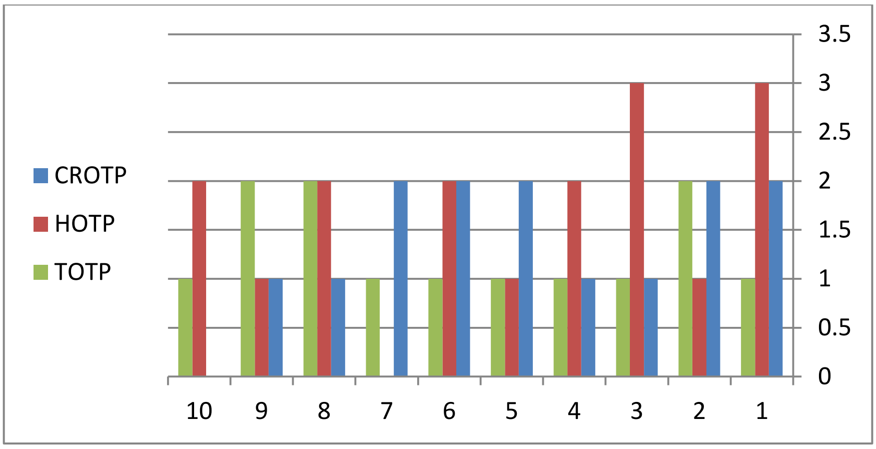


Figure 5.1: Average response time

5.4 Synchronization Challenges:

After applying ELSBOT, some of the synchronization challenges are found at each OTP technique. One of the most important synchronization challenges is the CPU overhead. The CPU overhead challenge is caused by the hash process which is used in the OTP generation process in the ELSBOT. Moreover, if the system performs more hash processes, the system will busy the CPU. Thus, we measure the CPU overhead for each OTP technique in ELSBOT through representing a different cases. The synchronization process is offered for each OTP technique separately as shown below:

5.4.1 TOTP synchronization:

The TOTP synchronization process is provided to synchronize between the server time and the user time. In this technique, the time is the basic factor in the hash process which is used for generating our OTP. In TOTP technique of our ELSBOT, the authors determine 36 rolls (i.e., 36 hash processes) as a worst case to represent the synchronization process for this technique. In the figure (5.2), the 36 rolls of TOTP synchronization process are divided into two halves: (1) The first half of the synchronization process is used when the user time is less than the server time; (2) The second half of the synchronization process is used when the user time is greater than the server time. The system works a hash roll every 10 seconds (i.e., one roll is equal to 10 seconds). Thus, the worst case of the TOTP synchronization process is equal to 36 rolls/hash processes. This case needs 360 seconds . So , we put OTP duration to be 3 minutes to give chance for server synchronization process to match with client synchronization process.

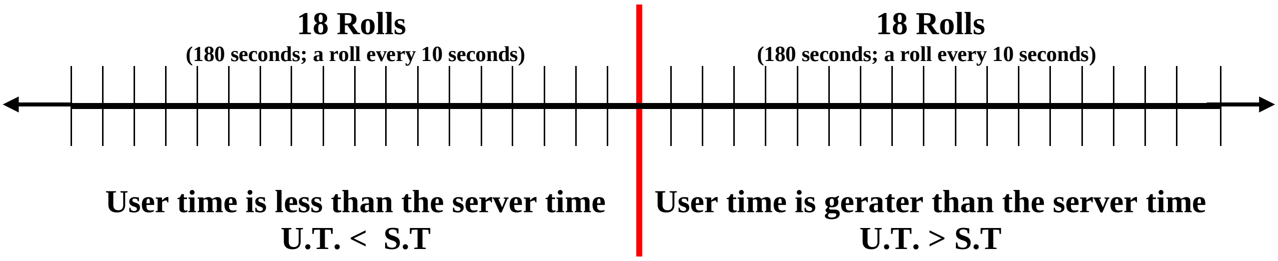

18 Rolls
(180 seconds; a roll every 10 seconds)
18 Rolls
(180 seconds; a roll every 10 seconds)
User time is less than the server time
U.T. < S.T
User time is gerater than the server time
U.T. > S.T

Figure 5.2: Represents the Synchronization process of TOTP

- **TOTP Synchronization Algorithm:**

The TOTP synchronization algorithm is implemented to handle the synchronization process between the server and user. In this algorithm, a set of variables should be defined like interval, timeInUnit, $epoch, and i. The value of the interval variable is equal to 10 (i.e., 10 seconds for each roll). The end value of the iteration of the TOTP synchronization process is equal to 36 (i.e., the maximum number of the hash processes is equal to 36 rolls/hash processes). Figure (5.3) below shows the TOTP synchronization algorithm code.

```
$interval = 10;
$ timeInUnit = floor($time/$interval);
$i = 0;
        $ epoch = $timeInUnit – 18;
        while ($i < 36) {
                $otp = $epoch.$user->getSecret().$pin;
                $otp = md5($otp);
                $epoch++;
                $i++;}
```

Figure 5.3: The Synchronization Algorithm Code for TOTP

- **Synchronization Cases of TOTP:**

To represent the TOTP synchronization process in ELSBOT, a set of different cases are presented to synchronize the time between the server and the user as shown in the table 5.4. If the server clock and the user clock are differ, the system will synchronize the time between the server and the user in order to match between the server clock and the user clock. The synchronization process may require more hash processes by the system. As stated previously, the synchronization process of TOTP technique provides 36 rolls/ hash processes as the worst case. The synchronization cases of TOTP are shown below:

Case 1:

User A requests a response from the server at the time 12:29:10, while the server time is 12:30:00. This case shows that the server time is greater than the user time with 50 seconds. The number of rolls in this case is computed through looking in the first half of the figure [5.1]. Thus, the system needs to 5 rolls /hash processes in this case of synchronization.

Case 2:

User B requests a response from the server at the time 12:25:20, while the server time is 12:27:00. Case 2 shows that the server time is greater than the user time with 100 seconds. The number of rolls in this case is computed through looking in the first half of the figure [5.1]. Thus, the system needs to 10 rolls /hash processes in this case of synchronization.

Case 3:

User C requests a response from the server at the time 12:15:00, while the server time is 12:13:20. This case shows that the user time is greater than the server time with 100 seconds. In this case, firstly, the system works 18 rolls of the first half of the figure [5.1] and works 10 rolls from the second half. Thus, the system needs to 28 rolls /hash processes (280 seconds) in this case of synchronization.

Case 4:

User D requests a response from the server at the time 11:46:00, while the server time is 11:48:00. Case 4 shows that the server time is greater than the user time with 120 seconds. The number of rolls in this case is computed through looking in the first half of the figure [5.1]. Thus, the system needs to 12 rolls /hash processes in this case of synchronization.

Case 5:

User E requests a response from the server at the time 12:32:00, while the server time is 12:29:10. Case 5 shows that the user time is greater than the server time with 170 seconds. In this case, firstly, the system works 18 rolls of the first half of the figure [5.1] and then works 17 rolls from the second half. Thus, the system needs to 35 rolls /hash processes (350 seconds) in this case of synchronization.

Case 6:

User F requests a response from the server at the time 10:30:00, while the server time is 10:32:30. This Case shows that the server time is greater than the user time with 150 seconds. The number of rolls in this case is computed through looking in the first half of the figure [5.1]. Thus, the system needs to 15 rolls /hash processes in this case of synchronization.

Table 5.4: Synchronization Cases of TOTP

Case #	User	IP Address	User Clock	Server Clock	Hash Process (Roll)
1.	A	10.10.10.1	12:29:10	12:30:00	5
2.	B	10.10.10.31	12:25:20	12:27:00	10
3.	C	10.10.10.24	12:15:00	12:13:20	28
4.	D	10.10.10.16	11:46:00	11:48:00	12
5.	E	10.10.10.8	12:32:00	12:29:10	35
6.	F	10.10.10.22	10:30:00	10:32:30	15

5.4.2 CROTP/HOTP synchronization:

The synchronization process is the same in CROTP and HOTP techniques. The two techniques use the counter to synchronize between the server and the user. This counter is the basic factor in the hash process. The server saves the counter value for the last valid login process by the system user (i.e., if the user works five valid login processes, the last counter at server is equal to 5). When the user works invalid login processes as the margin of error, the server will add 1 to its counter in order to match with the user counter. This can lead to the counter value being out of synchronization between the user and server. Thus, the server needs a number of rolls/hash process for the matching process. For two techniques, the authors determine 50 rolls (i.e., 50 hash processes) as a worst case to represent the synchronization process for these techniques.

- **CROTP/HOTP Synchronization Algorithm:**

The synchronization algorithm for each of two techniques is implemented to synchronize the counter between the server and user. In this algorithm, a set of variables are defined like counter, counter client, current counter , MOE, and i. The end value of the iteration is equal to 50 (i.e., the maximum number of the hash processes is equal to 50 rolls/hash processes). Figure5.5 below shows the CROTP/HOTP synchronization algorithms code.

```
$No  beginning of roll =counter client .
$roll end =counter + MOE
Function sync_OTP
{
$ Counter=Counter+1;
$i=0;
$While (i<=50)
$Current counter=counter client –i;
For (i=0;i<i*2;i++)
$Counter++;
$i++;
}
```

Figure 5.5: The Synchronization Algorithm Code for CROTP and HOTP

- **Synchronization Cases of CROTP and HOTP**

To represent the synchronization process at the two techniques, a set of different cases are presented to synchronize between the server counter and the user counter as shown in the table 5.5. If the server counter and the user counter are differ, the server will synchronize its counter with the user counter. The synchronization process may require more hash processes by the system. As stated previously, the synchronization process of two techniques in the ELSBOT provides 50 rolls/ hash processes as the worst case. The synchronization cases for the two techniques are shown below:

Case 1:

User A works invalid login through pressing the login button. Thus, the counter of this user becomes 50 as the margin of error. While the server counter is equal to 5. The system needs 45 rolls /hash processes in this case of synchronization. Case 1 is the worst case in the synchronization process of CROTP and HOTP.

Case 2:

User B works invalid login through pressing the login button. Thus, the counter of this user becomes 45 as the margin of error. While the server counter is equal to 5. The system needs 40 rolls /hash processes in this case of synchronization.

Case 3:

User C works invalid login through pressing the login button. Thus, the counter of this user becomes 15 as the margin of error. While the server counter is equal to 2. The system needs 13 rolls /hash processes in this case of synchronization.

Case 4:

User D works invalid login through pressing the login button. Thus, the counter of this user becomes 22 as the margin of error. While the server counter is equal to 10. The system needs 12 rolls /hash processes in this case of synchronization.

Case 5:

User E works invalid login through pressing the login button. Thus, the counter of this user becomes 15 as the margin of error. While the server counter is equal to 10. The system needs 5 rolls /hash processes in this case of synchronization.

Case 6:

User F works invalid login through pressing the login button. Thus, the counter of this user becomes 6 as the margin of error. While the server counter is equal to 5. The system needs 1 roll /hash process in this case of synchronization. The figure[5.6] illustrates the number of rolls/hash processes for the different cases of the two techniques.

Table 5.5: CROTP/HOTP Cases

Case #	**User**	**IP address**	**User Counter**	**Server Counter**	**Hash Process (Roll)**
1.	A	10.10.10.1	50	5	45
2.	B	10.10.10.31	45	5	40
3.	C	10.10.10.24	15	2	13
4.	D	10.10.10.16	22	10	12
5.	E	10.10.10.8	15	10	5
6.	F	10.10.10.22	6	5	1

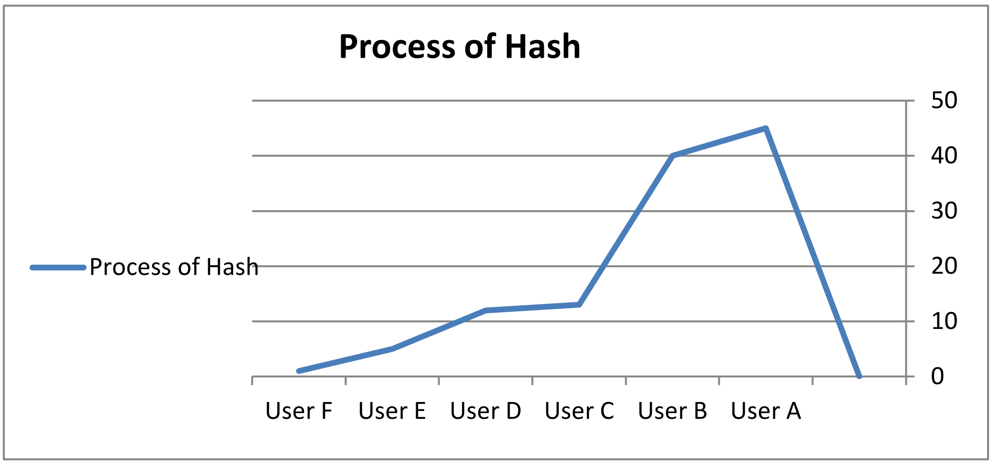


Figure 5.6 : Number of hash processes for cases of CROTP/HOTP techniques

5.5 Measuring ELSBOT against replay attacks:

This section focuses on measuring our work from the security perspective and showing if the ELSBOT prevents the replay attack. The authors present two steps which draw the process of data theft by the attacker. In the first step, the user enters his data (username and password) in the login screen of our system. On the other hand, the attacker snoops and steals the user data to login to the system and then takes a successful authentication from RADIUS server. Thus, the attacker can move to the next screen in the system. In the second step, after entering the user name and password from user, the user enters the PIN and OTP generator. The attacker can capture PIN & OTP from user, but the authentication is failed from the authentication server. Therefore, the attacker cannot enter to our ELSBOT because the authentication server generates a unique number for each connection and checks the secret key, the PIN and the last OTP which is not exist in the attacker side.

The three techniques (TOTP, HOTP, CROTP) of our work are presented against the replay attack as shown in table 5.6. In TOTP technique, firstly, the user enters the system through remote PC with IP address of *10.10.10.4* at the time of *11:20:05* while the server will response at *11:20.35*. The attacker with IP address of *10.10.10.7* replays the username and password of user to RADIUS server at a time of 11:21.39. This attacker gets the succeed authentication from RADIUS server. Secondly, The user enters the PIN and OTP in OTP page of our system. The attacker captures OTP and PIN from user, and then replays it to authentication server at *11:33.25*, but this attacker cannot login to the system because he gets a failed authentication from authentication server.

In second technique (HOTP), the user enters system at *11:40:08, while* the server responses at*11:40.48*, the attacker replays the username and password to RADIUS server at *11:45.30*,while a successful authentication from RADIUS server. Thus, the attacker can move to the next screen in this system after entering the user name and password from user, the user enters the PIN and OTP generator. The attacker can capture PIN & OTP from user at *12:13.22*, while the server responds with authentication failure.

In the CROTP technique, the user enters system at 12:19:02, while the server responses at *12:20.31*, the attacker replays username and password to RADIUS server at *12:28.04*, where the server responds with successful authentication. The attacker captures OTP, PIN and replays to authentication server at *12:40.09*, the server respond with failed authentication.

Table 5.6 : Test replay attack

Algorithm	Step-id	The User		The Server		Privacy data	The Attacker		RADIUS Server	Authentication Server Response
		IP	Time	IP	Time		IP	Time		
TOTP	**1**	10.10.10.4	11:20:05	10.10.10.22	11:20.35	Username password	10.10.10.7	11:21.39	S	-
	2	10.10.10.4	11:30:12	10.10.10.22	11:31:01	PIN / OTP	10.10.10.7	11:33.25	-	F
HOTP	**1**	10.10.10.4	11:40:08	10.10.10.22	11:40.48	Username password	10.10.10.7	11:45.30	S	-
	2	10.10.10.4	12:00:00	10.10.10.22	12:05:11	PIN/ OTP	10.10.10.7	12:13.22	-	F
CROTP	**1**	10.10.10.4	12:19:02	10.10.10.22	12:20.31	Username password	10.10.10.7	12:28.04	S	F
	2	10.10.10.4	12:38:17	10.10.10.22	12:38:59	PIN/ OTP	10.10.10.7	12:40.09	-	F

S: authentication successful; F: authentication failed.

5.6 Results Comparison:

The implementation of our work proves that the ELSBOT is more effective, more powerful, and more able to satisfy the security needs in the RADIUS environment. A comprehensive comparison between the three OTP techniques of ELSBOT is presented in this section. This comparison considers a set of factors. These factors are: preventing replay attack; CPU overhead, Algorithm speed, and server response time.

Firstly, to measure the server response time at each of the three OTP techniques in our ELSBOT, a set of cases are provided for each OTP technique to compute the average server response time for the system user requests as described in the section 5.3. The average server response time is equal to 1.2500 seconds when using TOTP technique, the average server response time is equal to 1.7500 seconds when using HOTP technique, and the average response time is equal to 1.6250 seconds when using CROTP technique. Thus, through measuring the server response time factor, the result shows that the TOTP technique in the ELSBOT is the best because this technique gets less average response time by the server. The OTP techniques of ELSBOT are arranged by less server response time in the order TOTP,CROTP, and HOTP.

The CPU overhead factor is measured through the synchronization process cases for each of the three OTP techniques. These cases show that the TOTP technique needs 36 hash processes as the worst case, while the HOTP and CROTP techniques need 50 hash processes as the worst case. The result of measuring the CPU overhead finds that the CPU overhead at TOTP technique is less than the CPU overhead at HOTP and CROTP techniques. Also, through the synchronization process, the algorithm speed is measured for each OTP technique. Whenever the system works more hash processes, the technique speed is low. The result of measuring the technique speed factor shows that the algorithm speed at TOTP technique is the highest, and the technique speed at CROTP technique is higher than HOTP technique. In the CROTP technique, the system provides the user with a random PIN and then the system does not check this PIN leading to speed up the processor in the system matching process.

As described in the previous section 5.5, in the first phase of login process, after entering user name and password by the system user, the attacker can steal the user data. Thus, the attacker is succeeded in this phase. In the second phase of login process, after entering the PIN and the generated OTP, the attacker can also steal the user PIN and OTP. But, this attacker can not login to system because he doesn’t have the secret key. And then the system matching process at the attacker is failure. We apply this type of attack on the ELSBOT in order to test each of the three OTP techniques against replay attack. According to testing and evaluating these techniques against replay attack, we reach that the three OTP techniques prevent this replay attack in our ELSBOT. Table(5.7) summarizes the comparison among three OTP techniques:

Table 5.7: Three OTP Techniques Comparison

Factor	OTP Techniques		
	HOTP	*TOTP*	*CROTP*
Replay Attack	**Prevent**	**Prevent**	**Prevent**
CPU Overhead	**High (50 rolls/hash processes)**	**Low (36 rolls/ hash processes)**	**High (50 rolls/hash processes)**
Algorithm Speed	**Low**	**High**	**Medium**
Server Response Time	**High**	**Low**	**Medium**

In the ELSBOT, three OTP techniques are implemented for generating the OTP. In TOTP technique, the generated OTP is valid for a short time (i.e., ends after 3 minutes) and this OTP cannot be reused. In HOTP and CROTP techniques, the generated OTP is valid for a long time and is used for an unknown amount of time. Moreover, the TOTP technique is usually used in the applications that need to restrict the time like mobile applications and banking transactions. The HOTP technique is used in the applications that are not concerned time like emails. While the CROTP is used with the complex applications, especially in servers applications. Finally, It is clear from the results obtained in this chapter that the TOTP is the most secure technique, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP.

5.7 Conclusion

This chapter introduced the user login scenario for our ELSBOT through a set cases which show the login process by the system users. The average server response time for each OTP technique was measured. This chapter presented the synchronization process for each of the three OTP techniques to measure CPU overhead and algorithm speed. In order to test our ELSBOT against the replay attack, an attacked script is applied. The aim of the attacked script is to check if this attacker can steal the user data or not. Finally, this chapter summarized the results comparison of our work. This comparison is drawn by considering a set of factors between the three OTP techniques.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Nowadays, user authentication has become the most important issue in the security trend. RADIUS protocol provides remote services for user authentication, but there are some vulnerabilities in RADIUS protocol, where poor implementation in PRNG be more predicable to guess and repeat, this gives chance of attackers to access to users accounts and steal data. Replay attacks one of attacks that face RADIUS protocol, replay attack can replay of package from a different context into the intended context, thereby fooling the honest participants into during request the service. Previous studies are suggested several approaches to reduce the effect of replay attack such as PKI, PSK, IPsec, smartcart, session key, sets clock and OTP techniques. In order to enhance the security in RADIUS environment, three OTP techniques like TOTP, HOTP, CROTP are chosen. In this thesis, we surveyed the literature review related to work provided by various researchers in the past. Our efforts are exerted into two directions: firstly, present the ELSBOT for implementing the three OTP techniques to prevent replay attacks in RADIUS environment; secondly, provide a comparison between these OTP techniques of ELSBOT by considering a set of factors such as preventing replay attack, CPU overhead, algorithms speed, server response time and OTP duration.

After applying the ELSBOT, the comparison factors are measured at each of the three OTP in order to evaluate this work. The result of measuring the first factor shows that the three OTP techniques prevent the replay attack in RADIUS environment. After measuring the two factors: CPU overhead and the algorithm speed. The authors found that the CPU overhead at TOTP technique is less than the CPU overhead at HOTP and CROTP techniques; the algorithm speed at TOTP technique is the highest while the algorithm speed at CROTP technique is higher than HOTP technique. Through measuring the server response time factor, the average server response time at TOTP technique is equal to 1.2500 seconds while the average server response time at HOTP technique is equal to 1.7500 seconds, and the average response time at CROTP technique is equal to 1.6250 seconds. Thus, the TOTP technique in the ELSBOT is the best in terms of server response time. According to measuring the OTP duration factor, the result shows that the TOTP is the most secure technique because its OTP is valid for a short time, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP. Our ELSBOT is a better one than existing solutions from security perspective .

6.2 Future Work

In this section, a set of points are drawn to extend our future work. these points are namely below:

1. Send the secret key to user via the SMS service or email.
2. The length of the generated OTP in our work is 6 characters. The authors propose to increase the length of OTP because the long OTP makes a system more secure.
3. In this work, two fields are used in the authentication process (PIN and OTP). Consider other fields to enhance the security for user authentication that requires additional research.

References

[1] Rohan Deshmukh , "Interactive Remote Authentication Dial In User Service (RADIUS) Authentication Server Model", International Conference on Wireless and Mobile Communications, Page 238, ICWMC 2012.

[2] Hashmathur Rehman1 .Govardhan T. Venkat Narayana Rao ," Design and Implementation of RADIUS An Network Security Protocol" , Global Journal of Computer Science and Technology, Page 48 Vol. 10 Issue 7 Ver. 1.0 September 2010.

[3] Ang Gao, Wei Wei and Wenbo Shi , "Efficient Password-Proven Key Exchange Protocol against Relay Attack on Ad Hoc Networks",International Conference on Advanced Information Networking and Applications, page 8 , 2010 IEEE.

[4] Xuguang Ren , Xin-Wen Wu , Kun Tang ,"TSPass: A Dynamic User Authentication Scheme Based On Time and Space", IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.10, October 2012.

[5] madhuri and ramana lakshmi "attack patterns for detecting and preventing ddos and replay attacks", international journal of engineering and technology vol. 2(9), page 4850-4859, 2010.

[6] Lan Sun, Zhao Luo, Yingjie Wu, Yilei Wang , "A Technique for Preventing Replay Attack in Road Networks", International Conference on Advanced Information Networking and Applications, page 807-810, 2012 IEEE.

[7] Hyun-Chul Kim1, Hong-Woo Lee1, Kyung-Seok Lee 1, Moon-Seog Jun1 ,"A Design of One-Time Password Mechanism using Public Key Infrastructure", page 18-24 , 2008 IEEE.

[8] Xi Yang , "The Use of One-Time Password and RADIUS Authentication in a GSS-API Architecture" master thesis 2006.

[9] http://www.cprogramming.com/debugging/valgrind.html, (February,2013)online

[10] Wen-Bin Hsieh ,Jenq len " Design of a Time and Location Based One-Time Password Authentication Scheme", International Conference on Advanced Information Networking and Applications, page 201-206 , 2011 IEEE.

[11] Daniel Clark , "Vulnerability's of IPSEC: "A discussion of possible weaknesses in IPSEC implementation and protocols" , SANS Institute 2002.

[12] Mikko Saarinen, " Legacy User Authentication with IPSEC" master thesis 2004.

[13] Kenneth G. Paterson, Douglas Stebila " One-time-password-authenticated key exchange (full version) " , Australasian Conference on Information Security and Privacy (ACISP) 2010, Springer 2010.

[14] www.vasco.com , "Transparent, Strong Authentication Using Auto-managed VACMAN® RADIUS Middleware", 2001 VASCO.

[15] C. Rigney, S. Willens, A. Rubens Merit , rfc2865 "Remote Authentication Dial In User Service (RADIUS)", The Internet Society (2000).

[16] Daniel Szilagyi, Arti Sood, and Tejinder Singh, "Radius: A Remote Authentication Dial-In User Service" , JOURNAL, VOLUME 5, NUMBER 2, 2009.

[17] Florian Devic , Lionel Torres, Beno˘t Badrignans," Secure protocol implementation for remote bitstream update preventing replay attacks on FPGA", pages 179-182 , 2010 conference IEEE.

[18] Daniel Granlund, Christer hlund ," A Scalability Study of AAA Support in Heterogeneous Networking Environments with Global Roaming Support " page 488-493 , 2011 conference IEEE.

[19] Gwanyeon Kim1, Chinu Lee1, Sehyun Park, Ohyoung Song, and Byungho Jung ," A Study on Mobile Commerce AAA Mechanism for Wireless LAN", pp. 719-724, Springer 2003.

[20] Il-Gon Kim, Jin-Young Choi ," Formal Verification of PAP and EAP-MD5 Protocols in Wireless Networks :FDR Model Checking", page 264-269 No .2 , 2004 conference IEEE.

[21] Mark Norman, Alun Edwards, Christian Fernau , "Are personal digital certificates really usable and scalable? ", 2003.

[22] http://technet.microsoft.com/en-us/library/cc776961.aspx , (February,2013)online

[23] http://etutorials.org/Networking/Types+of+Authentication, (March ,2013)online

[24] http://www.h3c.com/portal/Technical_Support___Documents/Technical_Documents/Switches/H3C_S5500_Series.htm , (March,2013)online

[25] http://www.aerohive.com/techniques/technology-behind-techniques/simplified-strong-authentication with PSK. (February,2013)online

[26] Saar Drimer , Steven J. Murdoc , "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks",2007

[27] Jitendra Kumar Tyagi , A.K. Srivastava , Pratap Singh Patwal ," Remote User Authentication Scheme in Multi-server Environment using Smart Card ", International Journal of Computer Applications ,Volume 57– No.12, November 2012 .

[28] http://book.soundonair.ru/cisco/ch18lev1sec2.html , (January ,2013)online

[29] Chun-Ying Huang , Shang-PinMaa, Kuan-TaChen" , Using one-time passwords to prevent password phishing attacks ", page 4 , vol .34 , 2011 Elsevier.

[30] Swati Sukhija, Shilpi Gupta , "Wireless network protocol security comperative study", International Journal of Emerging Technology and Advanced Engineering ,ISSN 2250-2459, page 357 , Volume 2, Issue 1, January 2012.

[31] Hayriye Altunbasak and Henry Owen, "Alternative Pair-wise Key Exchange Protocols for Robust Security Networks (IEEE 802.11i) in Wireless LANs", School of Electrical and Computer Engineering Georgia Institute of Technology, page 52, 2004 IEEE.

[32] Alex X. Liu1, Jason Kovacs, Chin-Tser Huang, Mohamed G. Gouda,"A Secure Cookie Protocol" International Conference on Advanced Information Networking and Applications, pages 333-338, 2005 IEEE.

[33] Lanjun Dang , Weidong Kou , Yuxia xiao ,"An Improved Scheme for Unilateral Asymmetric Smart Card Authentication", International Conference on Advanced Information Networking and Applications, pages 265-268 , 2005 IEEE.

[34] Mohammed Misbahuddin, Mohammed Aijaz Ahmed, M.H. Shastri ,"A Simple and Efficient Techniques to Remote User Authentication Using Smart Cards", International Conference on Advanced Information Networking and Applications, ,page 223-228 , 2006 IEEE.

[35] Chin-Tser Huang Mohamed G. Gouda ,"An Anti-Replay Window Protocol with Controlled Shift" ,International Conference on Advanced Information Networking and Applications, 2001 IEEE.

[36] "AAA / RADIUS ",PDF, Available: http://www.indigoo.com , Peter R. Egli 2013 .

[37] Ji Cao," AAA Functionality for Handheld Systems", master thesis 2005.

[38] Gmbh , "DIGIPASS Pack for Remote Authentication ", 2010 VASCO Data Security.

[39] Chii-Ren Tsai, "Non-Repudiation In Practice", Citigroup Information Security Office 2003 .

[40] Havard Raddum, Lars Hopland Nestas, and Kjell Jrgen Hole," Security Analysis of Mobile Phones Used as OTP Generators ", page 324-331, IFIP International Federation for Information Processing 2010.

[41] Andrew Y. Lindell ,"Time versus Event Based One-Time Passwords", the foundation of information security, white paper,2002.

[42] Bellare, Hoornaert, Naccache," HOTP: An HMAC-Based One-Time Password Algorithm" ,RFC4226, 2005

[43] Bellare, Hoornaert, Naccache, "CROTP: OATH Challenge-Response Algorithm", RFC4226, ISSN: 2070-1721,2011.

[44] D. M’Raihi," TOTP: Time-Based One-Time Password Algorithm" , Internet Engineering Task Force (IETF),2011.

[45] Sung-Jae Lee, Jae Seong Lee, Mun-Kyu Lee, Sang Jin Lee, Doo-Ho Choi, and Dong Kyue Kim ,"Low-Power Design of Hardware One-Time Password Generators for Card-Type OTPs", ETRI Journal, Volume 33, Number 4, August 2011.

[46] Christopher Leidigh," Fundamental Principles of Network Security", Schneider Electric’s Data Center Science Center,2013.

[47] Alan Harbitter , and Danile A. Menasce, A Methodology for Analyzing the Performance of Authentication Protocols, George Mason University, ACM Transactions on Information and System Security, Vol. 5, No. 4, November 2002.

[48] Anuradha Gupta," Network Computing, Windows 2000 Kerberos Authentication", December 05, 2000,URL:http://download.microsoft.com/download/1/6/4/16472da8-ce82-4361-94ba-072084d22178/kerberos.doc.

[49] Hani, Ali Al-Darabie , A modified Kerberos Authentication Protocol Utilizing Encrypted Key Exchange Mechanism, Master Thesis, Arab Academy for Science and Technology and Maritime Transport, March 2000, pp. 10-11.

[50] Lubna, M. H., Rajab, Eddeen, A modification of hash visualization technique in user authentication, master thesis, Jordan University, April 2004, pp. 11-14.

[51] Xinyuan, Wang, Internet Security Protocols, George Mason University, URL: http://mason.gmu.edu/~xwangc/teaching/ISA666/ISA666-lecture5-6.pdf , pp 25.

[52] Nathalie Dagorn, Nicolas Bernard, and S¥ebastien Varrette, Practical Authentication in Distributed Environment, CESI, University of Luxembourg, pp. 3-5, URL: http://www-id.imag.fr/~svarrett/download/publis/icsit_2005.pdf

[53] Himika Parmar, Nancy Nainan and Sumaiya Thasee ," Generation Of Secure One-Time Password Based On Image Authentication, Computer Science & Information Technology (CS & IT), 2012, pp. 195–206,

[54] Alireza Beikverdi, Ian K. T. Tan, "Improved Look-Ahead Re-Synchronization Window For Hmac-Based One-Time Password" ,2011.

[55] Collin Mulliner1 Ravishankar Borgaonkar2, Patrick Stewin, and Jean-Pierre Seifert2," SMS-Based One-Time Passwords: Attacks and Defense , Springer- 2013, pp. 150–159,

[56] Ms. E.Kalaikavitha M.C.A., M.Phil., Mrs. Juliana gnanaselvi M.Sc., M.Phil., Ph.D., " Secure Login Using Encrypted One Time Password (Otp) and Mobile Based Login Methodology ", International Journal Of Engineering And Science, Vol.2, Issue 10 (April 2013), Pp 14-17"

[57] Matthew Allan Ezell, "A Framework for Federated Two-Factor Authentication Enabling Cost-Effective Secure Access to Distributed Cyberinfrastructure", Masters Theses 2012.

[58] Usman Habib,"Secure Mobile Authentication for Linux Workstation log on", Norwegian University of Science and Technology,2010

[59] http://technet.microsoft.com/en-us/library/cc726017%28v=ws.10%29.aspx, (sep ,2013 online)

[60] Jonghoon Lee, Jungsoo Park, Seungwook Jung, and Souhwan Jung" Advanced OTP Authentication Protocol using PUFs ",The Fifth International Conference on Evolving Interne, IARIA, 2013, page 48-51, ISBN 978-1-61208-285-1

[61] Hadeel Tariq Al-Rayes, "Studying Main Differences Between Linux & Windows Operating Systems", International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:12 No:04,2012.

[62] Philip Hoyer,"OTP and Challenge/Response algorithms for financial and e-government identity assurance: current landscape and trends", ISSE 2008 Securing Electronic Business Processes, pp 281-290, 2009.

[63] Trupti Hemant Gurav1, Manisha Dhage2,"remote client authentication using mobile phone ", International Journal of Scientific and Research Publications, Volume 2, Issue 5, May 2012,page 5549-5555, ISSN 2250-3153.

[64] Namrata Thakur, Vimmi Pandey" An Approach of Authentication in Public Cloud using Two Step Verification Code ", International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-5), 2013

[65] Humaira Dar1, Wajdi Fawzi Mohammed Al-Khateeb And Mohamed Hadi Habaebi "Secure Scheme For User Authentication And Authorization In Android Environment", Humaira Dar et al. Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 3, Issue 5, Sep-Oct 2013, pp.1874-1882 www.

[66] Adebukola Onashoga1, Adesina Sodiya1 and A. Afolorunso2 ," A One-Time Server-Specific Password Authentication Scheme ", Journal of Computing and Information Technology - CIT 20, 2012, 2, 85–93

[67] Xuguang Ren, Xin-Wen Wu, and Kun Tang, " TSPass: A Dynamic User Authentication Scheme Based On Time and Space", IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.10, October 2012.

[68] Balkis Hamdane ———————————— y , Ahmed Serhrouchni y , Adrien Montfaucony , Sihem Guemara ," Using the HMAC-Based One-Time Password Algorithm for TLS Authentication", , page 1-8 2011 conference IEEE.

[69] Benjamin W. Long , Colin J. Fidge," Formally Analysing a Security Protocol for Replay Attacks ", Australian Software Engineering Conference (ASWEC'06) , 10pp 2006 IEEE.

[70] Gagan Dua , Nitin Gautam , Dharmendar Sharma, Ankit Arora, " replay attack prevention in Kerberos authentication protocol using triple password , International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2, March 2013.

[71] Daniel Nilsson,"Security in behavior driven authentication for web Applications"", Master of Science in Engineering Technology Computer Science and Engineering

[72] Taiwo dayo ajakaiye, karl senanu kudzo krause ,"Online Based Authentication and Secure Payment Methods for M-Commerce Applications", Master of Science Thesis in the Programme Secure and Dependable computer systems, Chalmers University of Technology, University of Gothenburg July 2011.

[73] http://code.google.com/p/mod-authn-otp/wiki/OneTimePasswords, (March ,2014 online)

[74] Xue Liu1, Lui Sha1, Yixin Diao2, Steven Froehlich2, Joseph L. Hellerstein2, and Sujay Parekh2 ," Online Response Time Optimization of Apache , Web Server ",pp. 461–478, 2003. Springer-Verlag Berlin Heidelberg 2003.

[75] James LaPiedra ,"The Information Security Process Prevention, Detection and Response", SANS Institute 2002.

[76] http://www.thefreedictionary.com/response+time (July,2014 online)

Glossary of Terms

- **Attack:** An attack is the deliberate act that exploits vulnerability.

- **An adversary** : A person that is interested in attacking your network; his motivation can range from gathering or stealing information

- **Authentication:** Is the process of identifying an individual, usually based on a username and password.

- **Authenticators:** A record containing information that can be shown to have been recently generated using the session key known only by the client and server.

- **Authorization:** refers to a user's ability to access resources on a network, usually based on user account rights and privileges.

- **AS:** Authentication Server, this service issues Ticket Granting Tickets (TGTs) well for admission to the ticket-granting service in its domain.

- **Shared Secret :** shared secret is a password used between a RADIUS server and a RADIUS client to mutually verify identity. Both the RADIUS server and the RADIUS client must be configured with the same shared secret, it is required for all RADIUS protocol communications. [1]

- **Pseudo Random Number Generator :** Pseudo Random Number Generator (PRNG) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers [9].

- **Biometrics**: Is the measurement of a unique biological feature used to verify the claimed identity of an individual through automated means.

- **Advanced Encryption Standard :** Advanced Encryption Standard (AES) is relatively easy to implement, and requires little memory. AES is currently being deployed on a large scale. That supports a larger range of block and key sizes. As 128 bits and a key size of 128, 192, or 256 bits

- **CHAP:** Challenge Handshake Authentication Protocol, authentication by challenge/response. CHAP is a three way handshake protocol which is considered more secure than PAP Authentication Protocol.

- **DES:** The U.S. Data Encryption Standard.

- **EAP:** Extensible Authentication Protocol (EAP) extension was developed to answer the increasing need for authentication.

- **EKE:** Encrypted Key Exchange, Uses a combination of symmetric and public key cryptography.

- **SETA:** Security Education Training and Awareness programs. The purpose of computer of SETA is to enhance security.

- **Information security:** is well-informed sense of assurance that the information risks and controls are in balance.

- **LDAP:** is the specialization of this norm in a lightweight version adapted to TCP/IP networks.

- **MAC:** Message Authentication Code**,** is a public function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

- **PPP:** The Point-to-Point Protocol technology which allows encapsulating TCP/IP flows on a telephone link.

- **RADIUS:** The RADIUS (Remote Authentication Dial-In User Service) system is an authentication protocol created for STN accesses.

- **Security:** The quality or state of being secure to be free from danger.

- **Smart Card:** A device external to a server which provides user authentication. Smart cards may operate using challenge-response mechanisms, or by providing one-time passwords.

- **Threat:** A threat is a probable violation of security.

- **Password Based Authentication:** The traditional method for authenticating users has been to provide them with a secret password, which they must submit when requesting access to a particular system.

- **Token Based Authentication:** The identity of a human user can be proven by requiring the user to demonstrate possession of a physical object which is unique to that user, or to a group of users. Objects used for this it purpose are known as authentication token.

- **CHAP:** Challenge Handshake Authentication Protocol, authentication by challenge/response. CHAP is a three way handshake protocol which is considered more secure than PAP Authentication Protocol.

- **SOAP message SOAP Acronym** for the Simple Object Access Protocol, is a protocol for the exchange of information in a distributed environment. SOAP messages are encoded as XML document, and can be exchanged using a variety of underlying protocols. this allow to transport SOAP messages reliably JMS subscribers, while JMS used for sending messages between multi clients . It allows the communication between different components of a distributed application .

- **Apache Server** is a web server provides a variety of MultiProcessi Modules (MPMs) which allow apache to run in a process-based.

- **Time Stamping**

  Computing session with time stamps makes it even more difficult for replay attacks to be carried out. This is because the time stamp of the login session will no longer be valid due to the time difference between when the login session was first used and when replayed by the adversary.

- **CIA**

  focusing on the three core goals of confidentiality, integrity and availability of information.

Appendix A:

A1: OTP Application source code designed by Java language platform

1. TOTP Technique

```
public String TOTPAlgorithm(String secret, int pin)
{
        int timez = 10                    // The system works a hash roll every 10 seconds (i.e., one roll is equal to
                                          10 seconds). Thus, the worst case of the TOTP synchronization process is
        Date now = new Date();            equal to 36 rolls/hash processes.
         String epoch = "";
         epoch = "" + (now.getTime() + ((timez - 10) * 3600000));
         epoch = epoch.substring(0, epoch.length() - 4);
         String password = epoch + secret + pin;          //the encrypt length for generator OTP,
         MessageDigest md = null;                         we take it four character to PIN state
         md = MessageDigest.getInstance("MD5");
         md.update(password.getBytes());
         byte byteData[] = md.digest();
         StringBuffer sb = new StringBuffer();
         for (int i = 0; i < byteData.length; i++) {
              sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
           }
         String otp = sb.toString();          //six character is length the otp generator
        return otp.substring(0, 6);           ,we used six character by default.
           if (timer.isRunning() == false) {
              timer = new Timer(1000, actListner);
              timer.start();
           }
}
```

2. HOTP Technique

```
public String HOTPAlgorithm(String secret, int pin, int lastCount)
{
          String password =lastCount +secret + pin;
           MessageDigest md = null;
            md = MessageDigest.getInstance("MD5");
           md.update(password.getBytes());
           byte byteData[] = md.digest();
           StringBuffer sb = new StringBuffer();
           for (int i = 0; i < byteData.length; i++) {
               sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
           }
           String otp = sb.toString();
           return  otp.substring(0, 6);

        }
```

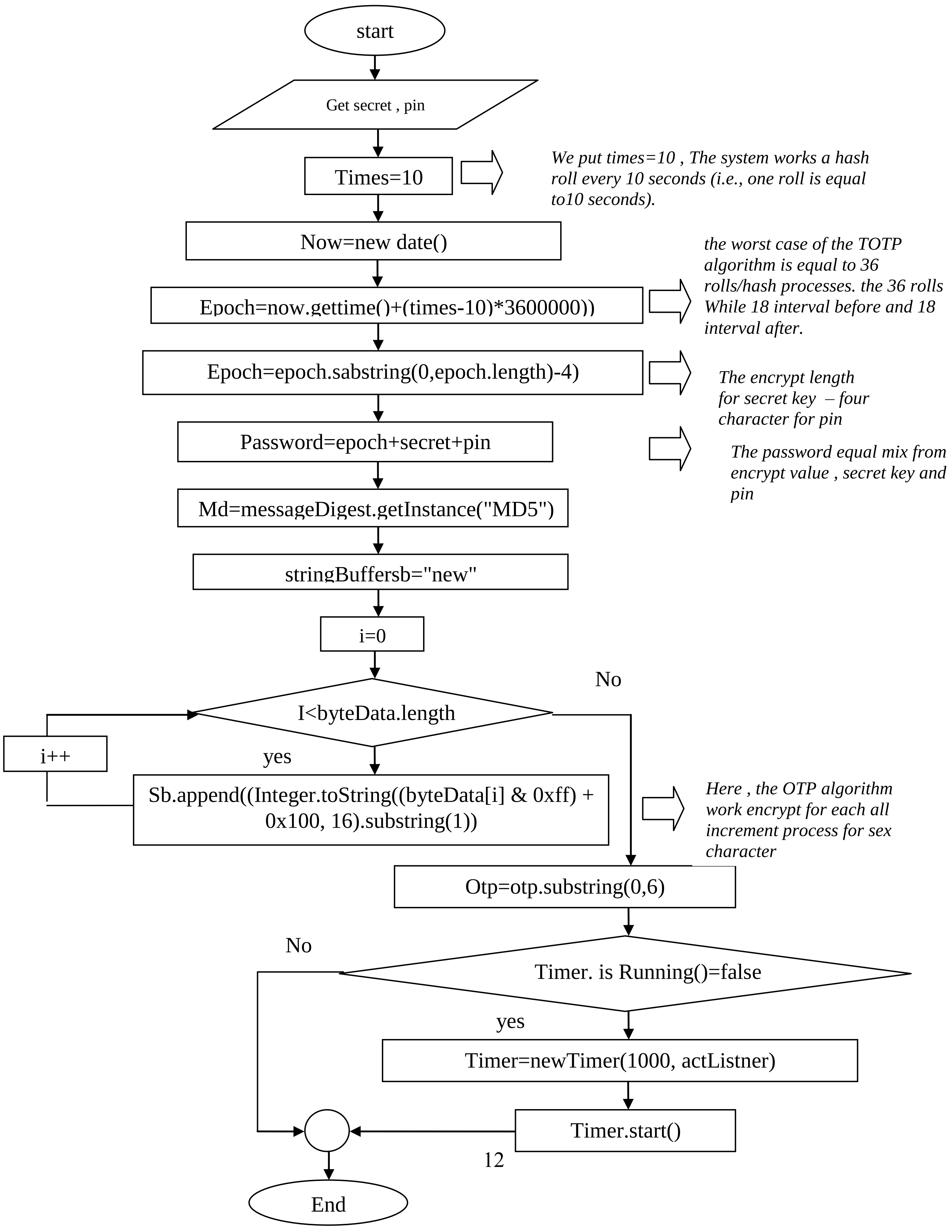
3. Challenge-response Technique

```
public String challengELSBOTAlgorithm(String secret, int randomPin, int lastCount)
{
            String password = lastCount + secret + randomPin;
            MessageDigest md = null;
            md = MessageDigest.getInstance("MD5");
            md.update(password.getBytes());
            byte byteData[] = md.digest();
            StringBuffer sb = new StringBuffer();
            for (int i = 0; i < byteData.length; i++) {
                sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
            }
           String otp = sb.toString();
            return otp.substring(0, 6);
        }
```

Appendix A:

A2: OTP Application flowchart designed by Java language platform

1. **TOTP algorithms flowchart**


start
Get secret , pin
Times=10
Now=new date()
Epoch=now.gettime()+(times-10)*3600000))
Epoch=epoch.sabstring(0,epoch.length)-4)
Password=epoch+secret+pin
Md=messageDigest.getInstance("MD5")
stringBuffersb="new"
i=0
No
I<byteData.length
i++
yes
Sb.append((Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1))
Otp=otp.substring(0,6)
No
Timer. is Running()=false
yes
Timer=newTimer(1000, actListner)
Timer.start()
End

We put times=10 , The system works a hash roll every 10 seconds (i.e., one roll is equal to10 seconds).

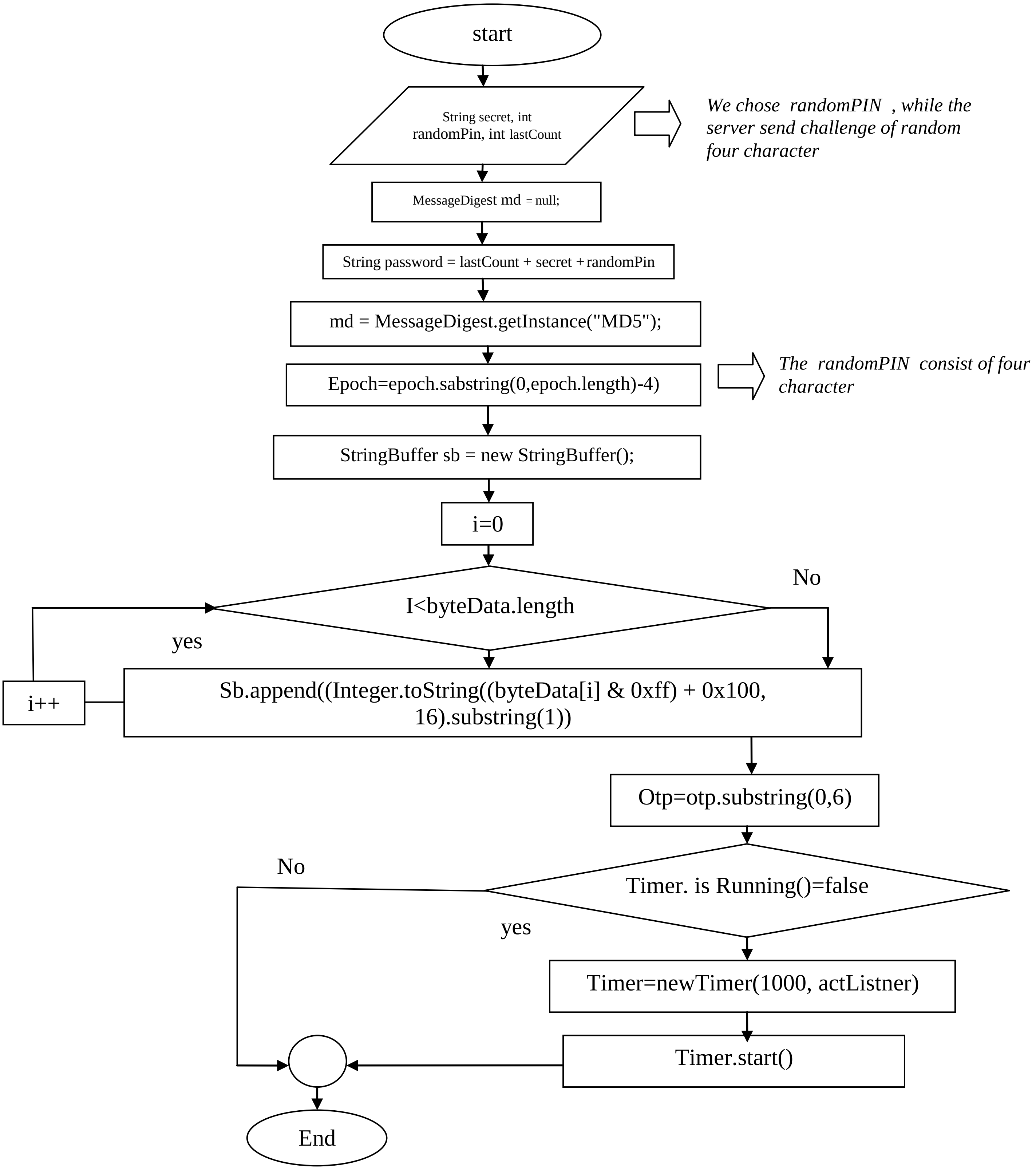
the worst case of the TOTP algorithm is equal to 36 rolls/hash processes. the 36 rolls While 18 interval before and 18 interval after.

The encrypt length for secret key – four character for pin

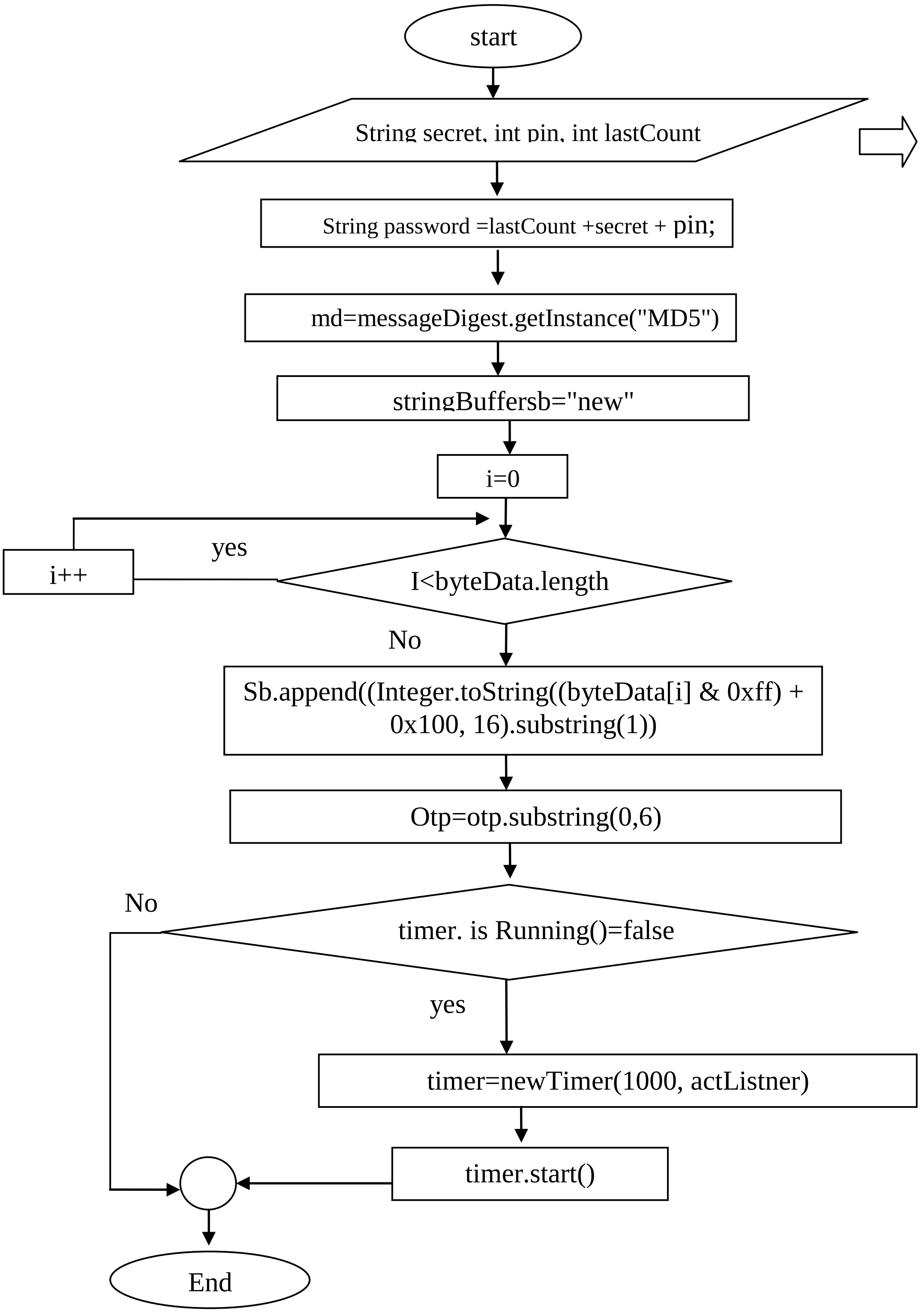
The password equal mix from encrypt value , secret key and pin

Here , the OTP algorithm work encrypt for each all increment process for sex character

2. OCRA algorithms flowchart


start
String secret, int randomPin, int lastCount
We chose randomPIN , while the server send challenge of random four character
MessageDigest md = null;
String password = lastCount + secret + randomPin
md = MessageDigest.getInstance("MD5");
Epoch=epoch.sabstring(0,epoch.length)-4)
The randomPIN consist of four character
StringBuffer sb = new StringBuffer();
i=0
I<byteData.length
No
yes
i++
Sb.append((Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1))
Otp=otp.substring(0,6)
No
Timer. is Running()=false
yes
Timer=newTimer(1000, actListner)
Timer.start()
End

3. HOTP algorithms flowchart


start
String secret, int pin, int lastCount
String password =lastCount +secret + pin;
md=messageDigest.getInstance("MD5")
stringBuffersb="new"
i=0
i++
yes
I<byteData.length
No
Sb.append((Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1))
Otp=otp.substring(0,6)
No
timer. is Running()=false
yes
timer=newTimer(1000, actListner)
timer.start()
End

We compute last Count for HOTP algorithm , so pin & last counter are important in the flowchart.

Appendix B

1. HOTP/CROTP synchronization designed by PHP platform

```
function authenticateHotp($username, $tokenOtp, $pin} (
        $con = connect();
        $sql = "SELECT * from users WHERE username='{$username};
        $result = mysql_query($sql,$con);
        $user = new User(mysql_fetch_row($result);
        $message = "Denied";
        $counter = $user->getRecentCounter;()
        $counter = $counter + 1;
        $i = 0;
        while ($i <= 50}(        // counter equal 50 roll as margin of error
                $otp = "{$counter}".$user->getSecret().$pin;
                $otp = md5($otp);
                $otp = substr($otp,0,6);
                if($otp == $tokenOtp}(
                        $sql = "UPDATE `users` SET `recentCounter`='{$counter}' WHERE
username='{$username'"{                        mysql_query($sql,$con);
                        $message = "Accepted";
                {
                $counter++;
                $i++;
        {
```

2. TOTP synchronization code

// The system works a hash roll every 10 seconds (i.e., one roll is equal to10 seconds). Thus, the worst case of the TOTP synchronization process is equal to 36 rolls/hash processes. the 36 rolls are divided into two halves: The first half of the synchronization process is used when the user time is less than the server time; (2) The second half of the synchronization process is used when the user time is greater than the server time , While 18 interval before and 18 interval after.

```
$time = time;()
    $interval = 10;
    $timeInUnit = floor($time/$interval);
    $i = 0;
    $epoch = $timeInUnit - 18;

    while ($i < 36}(
        $otp = $epoch.$user->getSecret().$pin;
        $otp = md5($otp);
        $otp = substr($otp,0,6);
        if($otp == $tokenOtp)
            $message = "Accepted";
        $epoch++;
        $i++;
     {

    return join(",", array($message)) ;
    mysql_close($con);
{
```

Appendix C

Replay attack implementation

```
$(document).ready(function(){
setInterval(function() {
$.get("getOtp.php",function(data){
data = jQuery.parseJSON(data);
console.log(data);
var text = "<h3 id='"+data.id+"'> [NOTICE] new data recieved , OTP =
"+data.otp+" and PIN = "+data.pin+"</h3>";
var lastH2 = $( "h3" ).last();
if(lastH2.attr("id") != data.id){
$("#otp").attr("value",data.otp);
$("#pin").attr("value",data.pin);
$(".data").append(text);
window.open("http://10.10.10.22/elearning/otp.php?pin="+data.otp+"&otp="
+data.pin+"",'_blank');
var text = "<h4> [LOGIN] login data submited  , OTP = "+data.otp+"
and PIN = "+data.pin+"</h4>";
}
 });
   }, 500);
         });
```

Appendix D

ELSBOT pages, designed by PHP platform

1. Authentication Settings.php

```
<"h1 class="normal-h1>
you
<? ;['have chosen <?php echo $vars['algo
<"form action="" method="post>
<p class="label">list of algorithms</p>
<p>
<"select name="mechanisms>
<option> Select one </option>
option value="1" <?php echo ($vars['data']->mechanism == 1)? "selected" : >
<""; ?> > HOTP. </option
option value="2" <?php echo ($vars['data']->mechanism == 2)? "selected" : >
<""; ?> > TOTP. </option
option value="3" <?php echo ($vars['data']->mechanism == 3)? "selected" : >
<""; ?> > CHALENGE RESPONSE </option
<select/>
<p/>
<";p style="margin-top:10px>
</ "input type="submit"  value="change>
<p/>
<form/>
```

2. Login.php

```
<head>
<"meta charset="utf-8>
<"meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1>
<title>Login Form</title>
<"link rel="stylesheet" href="css/style.css>
if lt IE 9]><script ]--!>
<--[src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif
<head/>
<body>
<"section class="container>
<"div class="login>
<h1>Login to Web App</h1>
<"form method="post" action="index.html>
p><input type="text" name="login" value="" placeholder="Username or >
<Email"></p
p><input type="password" name="password" value="" >
<placeholder="Password"></p
<"p class="remember_me>
<label>
```

```
<"input type="checkbox" name="remember_me" id="remember_me>
Remember me on this computer
<label/>
<p/>
<p class="submit"><input type="submit" name="commit" value="Login"></p>
<form/>
<div/>

<"div class="login-help>
<p>Forgot your password? <a href="index.html">Click here to reset it</a>.</p>
<div/>
<section/>

<body/>
<html/>
```

3. Singnup.php

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Sign up Form</title>
  <link rel="stylesheet" href="css/style.css">
  <!--[if lt IE 9]><script
src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>
<body>
  <section class="container">
    <div class="login">
      <h1>signup to eLearning App</h1>
      <form method="post" action="index.html">
        <p><input type="text" name="username" value="" placeholder="please
enter User name"></p>
<p><input type="text" name="realname" value="" placeholder="please enter
you're Real name"></p>
<p><input type="text" name="date" value="" placeholder="please enter date of
birth."></p>
        <p><input type="password" name="password" value=""
placeholder="please enter Password"></p>
<p><input type="password" name="password" value="" placeholder="please
reconfirm password"></p>
        <p class="submit"><input type="submit" name="commit"
value="signup"></p>
      </form>
    </div>
  </section>
</body>
</html>
```

4. Otp.php

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>OTP FORM</title>
  <link rel="stylesheet" href="css/style.css">
  <!--[if lt IE 9]><script
src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>
<body>
  <section class="container">
    <div class="login">
<h1>OTP FORM</h1>
<div>
please enter the following information to ensure your login data.
</div>
<form method="post" action="index.html">
<p><input type="text" name="login" value="" placeholder="Please enter
PIN"></p>
<p><input type="password" name="password" value="" placeholder="please
enter OTP"></p>
<p class="submit"><input type="submit" name="commit"
value="authenticate."></p>
</form>
    </div>
  </section>
</body>
</html>
```

5. IndexOtp.php

```
<html>
<head>
<script src="jquery.js"></script>
<script>
}()document).ready(function)$
} ()setInterval(function
}(get("getOtp.php",function(data.$
;(data = jQuery.parseJSON(data
;(console.log(data
var text = "<h3 id='"+data.id+"'>
;"<[NOTICE] new data recieved , OTP = "+data.otp+" and PIN = "+data.pin+"</h3
;()var lastH2 = $( "h3" ).last
}(if(lastH2.attr("id") != data.id
;(otp").attr("value",data.otp#")$
;(pin").attr("value",data.pin#")$
;(data").append(text.")$
```

```
window.open("http://192.168.56.102/elearning/otp.php?pin="+data.otp+"&otp="+d
;('ata.pin+"",'_blank
var text = "<h4> [LOGIN]
;"<login data submited  , OTP = "+data.otp+" and PIN = "+data.pin+"</h4
{
;({
;(500 ,{
;({
<script/>
<head/>
<body>
<"div class="form>
form method="post" id="testform" style="display:none" >
<""=action
<"input id="otp" name="otp" type="hidden>
<"input id="pin" name="pin" type="hidden>
<form/>
<div/>
<"div class="data>

<div/>
<body/>
<html/>
```

6. GetOtp.php

```
php?>
;("mysqli = new mysqli("localhost", "root", "", "attack_db$

/* check connection */
} (()if (mysqli_connect_errno
;(()printf("Connect failed: %s\n", mysqli_connect_error
;()exit
{

;"query = "SELECT id, otp, pin FROM `otp` ORDER BY id DESC LIMIT 1$

} ((if ($result = $mysqli->query($query
/* fetch object array */
;()row = $result->fetch_row$
)data = array$
,[id" => $row[0"
,[otp" => $row[1"
,[pin" => $row[2"
;(
;(data = json_encode($data$
/* free result set */
;()result->close$
;echo $data
;exit
```

```
{

/* close connection */
;()mysqli->close$
```

7. Student.php

```
<?php
class Student{
    public $id;
    public $name;
    public $dob;
    public $email;
    public $stime;
    public $con;
    public $attributes;
    function __construct(){
        require_once("Connection.php");
        $connection = new Connection();
        $this->con = $connection->getConnection();
        $this->attributes = array("name" => $this->name,"dob" => $this->dob, "email" => $this->email);
    }
    function setModel($data){
        while (list($key, $value) = each($data)) {
            $this->$key = $value;
            if(array_key_exists($key, $this->attributes))
                $this->attributes[$key] = $value;
        }
    }
        function save(){
        $sql = "INSERT INTO `student`(`id`, ";
        while (list($key, $value) = each($this->attributes)) {
            $sql = $sql . "`$key`,";
        }
        $sql = $sql."`time`) VALUES ( NULL, ";
        foreach ($this->attributes as $key => $value) {
            $sql = $sql . "'$value',";
        }
        $sql = $sql."NULL)";
        $result = mysqli_query($this->con, $sql);
        return $result;
    }
    function update($id){
        $sql = "UPDATE `student` SET ";
        $i = 0;
        while (list($key, $value) = each($this->attributes)) {
            $sql = $sql . " `$key` = '$value'";
            if($i != count($this->attributes)-1)
                $sql = $sql . ",";
            $i++;
        }
        $sql = $sql." WHERE id = $id";
        $result = mysqli_query($this->con, $sql);
        return $result;
    }
    public function findByPk($id){
        $sql = "SELECT * from student WHERE id = {$id}";
        $result = mysqli_query($this->con, $sql);
        $row = mysqli_fetch_row($result);
        $this->id = $row[0];
        $this->name = $row[1];
        $this->dob = $row[2];
        $this->email = $row[3];
        $this->stime = $row[4];
        return $this;
    }
    public function findAll(){
        $sql = "SELECT * from student";
        $result = mysqli_query($this->con, $sql);
        $students = array();
        while($row = mysqli_fetch_array($result)){
            $std = new Student;
            $std->id = $row['id'];
            $std->name = $row['name'];
            $std->dob = $row['dob'];
            $std->email = $row['email'];
            $std->stime = $row['time'];
            array_push($students, $std);
        }
        return $students;
    }
    public function delete($id){
        $sql = "DELETE FROM `college_db`.`student` WHERE
`student`.`id` = {$id}";
        $result = mysqli_query($this->con, $sql);
        return $result;
    }
    public function insert(){
        }
}
```

8. StudentCourse.php

```
 <?php
class StudentCourse{
    public $id;
    public $course_id;
    public $student_id;
    public $stime;
    public $con;
    public $attributes;
    function __construct(){
        require_once("Connection.php");
        $connection = new Connection();
        $this->con = $connection->getConnection();
        $this->attributes = array("course_id" => $this->course_id,
"student_id" => $this->student_id);
    }
    function setModel($data){
        while (list($key, $value) = each($data)) {
            $this->$key = $value;
            if(array_key_exists($key, $this->attributes))
                $this->attributes[$key] = $value;
        }
    }
    function save(){
        $sql = "INSERT INTO `course_student`(`id`, ";
        while (list($key, $value) = each($this->attributes)) {
            $sql = $sql . "`$key`,";
        }
        $sql = $sql."`time`) VALUES ( NULL, ";
        foreach ($this->attributes as $key => $value) {
            $sql = $sql . "'$value',";
        }
        $sql = $sql."NULL)";
        //die($sql);
        $result = mysqli_query($this->con, $sql);
        return $result;
    }
    function update($id){
        $sql = "UPDATE `course_student` SET ";
        $i = 0;
        while (list($key, $value) = each($this->attributes)) {
            $sql = $sql . " `$key` = '$value'";
            if($i != count($this->attributes)-1)
                $sql = $sql . ",";
            $i++;
        }
        $sql = $sql." WHERE id = $id";
        $result = mysqli_query($this->con, $sql);
        return $result;
    }
    public function findByPk($id){
        $sql = "SELECT * from course_student WHERE id = {$id}";
        $result = mysqli_query($this->con, $sql);
        $row = mysqli_fetch_row($result);
        $this->id = $row[0];
        $this->course_id = $row[1];
        $this->student_id = $row[2];
        $this->stime = $row[3];
        return $this;
        }
        public function findAll(){
                $sql = "SELECT * from course_student";
                $result = mysqli_query($this->con, $sql);
                $model = array();
                while($row = mysqli_fetch_array($result)){
                        $mdl = new StudentCourse;
                        $mdl->id = $row['id'];
                        $mdl->course_id = $row['course_id'];
                        $mdl->student_id = $row['student_id'];
                        $mdl->stime = $row['time'];
                        array_push($model, $mdl);
                }
                return $model;
        }
        public function delete($id){
                $sql = "DELETE FROM `college_db`.`course_student` WHERE
`course_student`.`id` = {$id}";
                $result = mysqli_query($this->con, $sql);
                return $result;
        }
                public function insert(){

        }
}
```

9. Students Management.php

```
<h1>Students Management > Create New Student</h1>
<form method="post" action="">
<p class="label">student name</p>
        <p><input type="text" name="name" value="<?php echo $vars['student']-
>name ?> " placeholder="please enter student name"></p>
        <p class="label">student date of birth</p>
        <p><input name="dob" value="<?php echo $vars['student']->dob ?>"
type="date" placeholder="please enter your date of birth"></p>
        <p class="label">student email</p>
        <p><input type="email" name="email" value="<?php echo $vars['student']-
>email ?>" placeholder="please enter student email"></p>
        <p class="submit"><input type="submit" name="commit"
value="create"></p>
</form>
```

10. Course.php

```
?>php
class Course}
  public $id;
  public $title;
  public $owner;
  public $code;
  public $stime;
  public $con;
  public $attributes;
  function __construct}()
          require_once("Connection.php;("
          require_once("Student.php;("
          $connection = new Connection;()
          $this->con = $connection->getConnection;()
          $this->attributes = array("title" => $this->title,"owner" => $this->owner,
"code" => $this->code;(
  {
  function setModel($data}(
          while (list($key, $value) = each($data} ((
                  $this->$key = $value;
                  if(array_key_exists($key, $this->attributes((
                          $this->attributes[$key] = $value;
          {
  {
  function save}()
          $sql = "INSERT INTO `course`(`id;" ,`
          while (list($key, $value) = each($this->attributes} ((
                  $sql = $sql . "`$key;",`
          {
          $sql = $sql."`time`) VALUES ( NULL;" ,
          foreach ($this->attributes as $key => $value} (
                  $sql = $sql . "'$value;",'
          {
          $sql = $sql."NULL;"(
          $result = mysqli_query($this->con, $sql;(
          return $result;
  {
  function update($id}(
          $sql = "UPDATE `course` SET;"
          $i = 0;
          while (list($key, $value) = each($this->attributes} ((
                  $sql = $sql . " `$key` = '$value;"'
                  if($i != count($this->attributes)-1(
                          $sql = $sql;"," .
                  $i;++
          {
          $sql = $sql." WHERE id = $id;"
          $result = mysqli_query($this->con, $sql;(
          return $result;
  {
  public function findByPk($id}(
          $sql = "SELECT * from course WHERE id = {$id;"{
          $result = mysqli_query($this->con, $sql;(
          $row = mysqli_fetch_row($result;(
          $this->id = $row[0;[
          $this->title = $row[1;[
          $this->owner = $row[2;[
          $this->code = $row[3;[
          $this->stime = $row[4;[
          return $this;
  {
  public function findAll}()
          $sql = "SELECT * from course;"
          $result = mysqli_query($this->con, $sql;(
          $courses = array;()
          while($row = mysqli_fetch_array($result}((
                  $crs = new Course;
                  $crs->id = $row['id;['
                  $crs->title = $row['title;['
                  $crs->owner = $row['owner;['
                  $crs->code = $row['code;['
                  $crs->stime = $row['time;['
                  array_push($courses, $crs;(
          {
          return $courses;
  {
  public function findAllStudents($id}(
          $sql = "SELECT *
                     FROM student
                          LEFT      JOIN      course_student      ON      student.id      =
course_student.student_id
                          LEFT JOIN course ON course_student.course_id = course.id
                          WHERE course.id = {$id} GROUP BY student_id;"
          $result = mysqli_query($this->con, $sql;(
          $students = array;()
          while($row = mysqli_fetch_array($result}((
                  $std = new Student;
                  $std->id = $row['id;['
                  $std->name = $row['name;['
                  $std->dob = $row['dob;['
                  $std->email = $row['email;['
                  $std->stime = $row['time;['
                  array_push($students, $std;(
          {
          return $students;
  {

  public function delete($id}(
          $sql   =   "DELETE   FROM   `college_db`.`course`   WHERE   `course`.`id`   =
{$id;"{
          $result = mysqli_query($this->con, $sql;(
          return $result;
  {

  public function insert}()        {{
```

11. CourseManagement.php

```
<h1>Courses Management</h1>
<"table class="tbl>
<thead>
<th>id</th>
<th>title</th>
<th>owner</th>
<th>code</th>
<th>time added</th>
<th>actions</th>
<thead/>
<tbody>
<? :(php foreach($vars['data'] as $course?>
<tr>
<td><?php echo $course->id ?></td>
<td><?php echo $course->title ?></td>
<td><?php echo $course->owner ?></td>
<td><?php echo $course->code ?></td>
td><?php echo date("Y-M-d",strtotime($course->stime)); >
<?></td
<td>
a onclick="return confirm('are you sure ?!');" >
href="http://10.10.10.42/elearning/index.php?page=Course&action=delete&id=<?p
<hp echo $course->id ?>" class="red">delete</a
a >
href="http://10.10.10.42/elearning/index.php?page=Course&action=update&id=<?p
<hp echo $course->id ?>" class="yellow">edit</a
|
a >
href="http://10.10.10.42/elearning/index.php?page=Course&action=students&id=<?
<php echo $course->id ?>" class="yellow">students</a
<td/>
<tr/>
<? php endforeach?>
<tbody/>
<table/>
```

12. Update Course.php

```
<h1>Courses Management > Create / Update Course</h1>
<form method="post" action="">
        <p class="label">course title</p>
        <p><input type="text" name="title" value="<?php echo $vars['course']-
>title ?> " placeholder="please enter course title"></p>
        <p class="label">course owner</p>
        <p><input name="owner" value="<?php echo $vars['course']->owner ?>"
type="text" placeholder="please enter course owner"></p>
        <p class="label">course code</p>
        <p><input type="text" name="code" value="<?php echo $vars['course']-
>code ?>" placeholder="please enter course code"></p>
```

```
		<p class="submit"><input type="submit" name="commit"
	value="create"></p>
	</form>
```

13. Enroll.php

```
	<h1>Students Management > Enroll to Course</h1>
	<form method="post" action="">
		<p class="label">student name</p>
		<p>
			<select name="student_id">
				<?php foreach($vars['students'] as $student): ?>
				<option value="<?php echo $student->id ?>"><?php echo
	$student->name ?></option>
				<?php endforeach ?>
			</select>
		</p>
		<p class="label">course title</p>
		<p>
			<select name="course_id">
				<?php foreach($vars['courses'] as $course): ?>
				<option value="<?php echo $course->id ?>"><?php echo
	$course->title ?></option>
				<?php endforeach ?>
			</select>
		</p>
		<br/>
		<p class="submit"><input type="submit" name="commit" value="enroll
	now"></p>
	</form>
```

14. Layout.php

```
	<head>
	  <meta charset="utf-8">
	  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
	  <title>Sign up Form</title>
	  <link rel="stylesheet" href="public/css/style.css">
	  <!--[if lt IE 9]><script
src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
	</head>
	<body>
	  <section class="wide-container">
		<h1 class="site-logo">
		Learning System
		</h1>
		<?php if(isset($_SESSION['secret'])){ ?>
		<h1 class="secret">
			your secret is <?php echo $_SESSION['secret']; ?>
			<?php unset($_SESSION['secret']); ?>
```

```
        </h1>
        <?php } ?>
        <div class="block">
            <div class="left span3">
                <h1>MAIN MENU</h1>
                <a href="index.php?page=Student&action=index">View
students</a>
                <br/>
                <a href="index.php?page=Student&action=create">Add new
student</a>
                <br/>
                <a href="index.php?page=Course&action=index">View
courses</a>
                <br/>
                <a href="index.php?page=Course&action=create">Add new
course </a>
                <br/>
                <a href="index.php?page=Student&action=enroll">Enroll
Student In Course </a>
                <br/>
                <a href="index.php?page=Student&action=logout"> Log Out
</a>
                <br/>
                <a href="index.php?page=Settings&action=index"> Change
Login mechanism </a>
            </diV>
            <div class="right span7">
                <?php require_once("views/{$viewName}.php"); ?>
            </div>
            <div class="clear"></div>
        </div>
     </section>
    </body>
    </html>
```

15. SessionOpen.php

```
        <?php
    session_start();
    $con=mysqli_connect("localhost","root","root","radius");
    if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    }
    $message = "";
    if(isset($_POST['username'])){
      $output = shell_exec("radtest {$_POST['username']} {$_POST['password']}
    127.0.0.1 0 0123456789012345678901234");
        if (strpos($output, 'Access-Accept') !== false) {
          $_SESSION['name'] = $_POST['username'];
         $_SESSION['firstStep'] = 1;
         header("Location: /elearning/otp.php");
        }else{
         $message = "you are not authorized to login .";
        }
    }
```

SessionClose.php

```
?php
session_start();
$con=mysqli_connect("localhost","root","root","college_db");
if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
$sql = "SELECT * FROM Settings WHERE id = 1";
$result = mysqli_query($con, $sql);
$data = mysqli_fetch_array($result);
$mech = $data['mechanism'];
$givenPin = null;
if($mech == 3){
        $givenPin = rand(0,9999);
}
if(isset($_POST['otp']) && isset($_POST['pin']) && isset($_SESSION['firstStep']) ){
     $username = $_SESSION['name'];
     $otp = $_POST['otp'];
        $pin = $_POST['pin'];
     require_once("sevice.php");
     if($result == "Accepted"){
                $_SESSION['username'] = $username;
        header("Location:index.php?page=Student&action=index");
        }else{
                $_SESSION['autherror'] = "your otp code is invalid, please try again.";
        }
}
?>
```

16. Route.php

```
<?php
//error_reporting(E_ERROR);
 error_reporting(0);
$modelPath = "model/";
 $controllerPath = "controller/";
 $host = $_SERVER['HTTP_HOST'];
 $url = $_SERVER['REQUEST_URI'];
 $page = $_GET['page'];
 $action = $_GET['action'];
 $id = 0;
 if(isset($_GET['id'])){
        $id = $_GET['id'];
 }
 include($controllerPath.$page."Controller.php");
//die($controllerPath.$page."Controller.php");
$class = $page."Controller";
// die($page);
 $object = new  $class;
 $object->$action($id);
?>
```

17. Service.php

```
<?php
error_reporting(E_ERROR);
require_once "lib/nusoap.php";
$client = new nusoap_client("http://192.168.56.101/auth/index.php");
$error = $client->getError();
if ($error) {
    echo "<h2>Constructor error</h2><pre>" . $error . "</pre>";
}
$result = "";

$requestTime = "";
$responseTime = "";
$type = "";
if(isset($secret)){
$result = $client->call("register", array("username" => $username, "secret" =>
$secret));
}
if(isset($otp) && isset($pin) && ($mech == 2) ){
            $requestTime = time();
$type = "TOTP";
$result = $client->call("authenticate", array("username" => $username, "tokenOtp"
=> $otp, "pin" => $pin));
$responseTime = time();
}
if(isset($otp) && isset($pin) && ($mech == 1) ){
        $requestTime = time();
        $type = "HOTP";
$result = $client->call("authenticateHotp", array("username" => $username,
"tokenOtp" => $otp, "pin" => $pin));
        $responseTime = time();
}
if(isset($otp) && isset($pin) && ($mech == 3) ){
        $requestTime = time();
        $type = "CR";
        $result = $client->call("authenticateHotp", array("username" => $username,
"tokenOtp" => $otp, "pin" => $pin));
        $responseTime = time();
}
$sql = "INSERT INTO logs VALUES('', '{$requestTime}', '{$responseTime}',
'{$type}');";
$output = mysqli_query($con, $sql);
```

```
//die(print_r($result));
if ($client->fault) {
   echo "<h2>Fault</h2><pre>";
   print_r($result);
   echo "</pre>";
}
else {
   $error = $client->getError();
   if ($error) {
      echo "<h2>Error</h2><pre>" . $error . "</pre>";
   }
   else {
     //echo $result;
   }
}
```

18. Controller.php

```
<?php
class Controller{

        public function checkUser(){
                if(!isset($_SESSION['username']))
                header("Location: http://10.10.10.42/elearning/login.php");
        }
        public function render($viewName, $vars = array()){
                //equire_once("views/{$viewName}.php");
                //die(print_r($vars));
                require_once("views/layout.php");
        }

}
```

19. SettingController.php

```
<?php
require_once("Controller.php");
class SettingsController extends Controller{
        function __construct(){
                $this->checkUser();
                require_once("model/Setting.php");
        }
                public function index($id = 0){
                $settings = new Setting;
                $data = $settings->findByPk();
                if(isset($_POST['mechanisms'])){
                        $mech = $_POST['mechanisms'];
                        if(is_numeric($mech)){
                                $setting = new Setting;
                                $setting -> mechanism = $mech;
                                $setting -> attributes['mechanism'] = $mech;
                //die(print_r($setting));
                $setting -> update();

    header("location:index.php?page=Settings&action=index");
            }else{
                die();
            }
        }
        $algo = null;
        switch($data->mechanism){
            case 1:
                $algo = "HOTP";
                break;
            case 2:
                $algo = "TOTP";
                break;
            case 3:
                $algo = "Challenge Response";
            break;
            default:
            $algo = "Not Selectec - User Can't Login...";
            break;
        }
        $this->render("Settings/index", array("data" => $data, "algo" =>
$algo));
    }
}
```

20. StudentController.php

```
<?php
require_once("Controller.php");
class StudentController extends Controller{
    public function logout(){
        session_unset();
        header("Location:http://10.10.10.42/elearning/login.php");
    }
    function __construct(){
        $this->checkUser();
        require_once("model/Student.php");
        require_once("model/Course.php");
        require_once("model/StudentCourse.php");
    }
    public function create(){
        $student = new Student;
        if(isset($_POST['name'])){
            $student -> setModel($_POST);
            $student -> save();
            $this->index();
            exit;
        }
        $this->render("student/form",array("student" => $student));
    }

    public function enroll($id){
        $studentCourse = new StudentCourse;
        $model = new Course;
        $courses = $model->findAll();
        if(isset($_POST['student_id'])){
            $studentCourse -> setModel($_POST);
            $studentCourse -> save();
            $this->index();
            exit;
        }
        $model = new Student;
        $students = $model->findAll();
        $this->render("student/enroll",array("students" => $students,
"courses" => $courses));
    }
        public function update($id){
        $model = new Student;
        $student = $model ->findByPk($id);
        if(isset($_POST['name'])){
            $student -> setModel($_POST);
            $student -> update($id);
            $this->index();
            exit;
        }
        $this->render("student/form",array("student" => $student));
    }

    public function delete($id){
        $students = new Student;
        $students->delete($id);
        $data = $students->findAll();
        $this->render("student/index",array("data" => $data));
    }
    public function view(){
        die("Coming Soon");
    }
    public function index(){
        $students = new Student;
        $data = $students->findAll();
        $this->render("student/index",array("data" => $data));
    }
}
```