

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



UNIVERSITY OF BLIDA 1



SCIENCE FACULTY

THESIS

Presented for the MASTER diploma in Computer science
Specialty: Information System Security

Strong Authentication Architecture OTP-OATH VPN SSL

Application case: Bank of Algeria

Jury members

Narhimene Boustia	Prof	Univ.Blida	President
Meriem Arkam	Doctor	Univ.Blida	Examiner
Fatima Zohra Adi	Director of training	Algeria Bank	Supervisor

Presented by Ibtissam Elgharbi

College year 2019/2020

ملخص

جعلت الشبكات الخاصة مصادقة الوصول عن بُعد جزءاً مهماً من بنية الشبكة ، من أجل الوصول لبيانات حساسة مع الحماية المناسبة عبر الشبكة العامة للإنترنت من الأطراف المصرح لها. المصادقة هي إحدى الخدمات المهمة لأنها تؤثر على سرية وسلامة البيانات. لم تعد آلية المصادقة الكلاسيكية القائمة على تركيبة اسم المستخدم وكلمة المرور كافية في زمن برامج الاختراق الفعالة، الحل هو إضافة طبقة إضافية من عوامل المصادقة. في هذا المشروع ، هدفنا هو تصميم وتنفيذ بنية مصادقة قوية من خلال الجمع بين استراتيجيتين المصادقة المفتوحة OATH و تقنية VPN SSL.

ABSTRACT

Complex distributed networks have made remote access authentication a significant part of network architecture in order to deliver sensitive data with the appropriate protection over the public internet to the authorized parties. Authentication is one of the important services since it affects the confidentiality and integrity of data. The classic authentication mechanism based on the username/password combination is no longer enough in the time of fast and strong crack software. The solution is to add an extra layer of authentication factors. In this project, our objective is to design and implement a strong authentication architecture by binding two strategies, the open authentication OATH one-time password mechanism, and the VPN SSL technology. For the OATH-OTP, we chose to develop an OATH TOTP applications, and we set up a VPN SSL tunnel using the ASA firewall. As a result, we achieve our purpose, tho the project could not be realized in a real network, we implement it using a simulation tool.

Keywords— OTP, OATH, HOTP, TOTP, ASA , VPN SSL, 2FA, QRcode, Strong authentication.

Les réseaux distribués complexes ont fait de l'authentification d'accès à distance une partie importante de l'architecture réseau, afin de fournir des données sensibles avec la protection appropriée à travers le réseau public d'Internet aux parties autorisées. L'authentification est l'un des services importants car elle affecte la confidentialité et l'intégrité des données. Le mécanisme d'authentification classique basé sur la combinaison nom d'utilisateur et mot de passe ne suffit plus au temps de rapide et fort logiciel de crack. La solution consiste à ajouter une couche supplémentaire de facteurs d'authentification. Dans ce projet, notre objectif est de concevoir et de mettre en œuvre une architecture d'authentification forte par combiner deux stratégies, l'authentification ouverte OATH-mot de passe à usage unique, et la technologie VPN SSL. Pour l'OATH-OTP, nous avons choisi de développer une application OATH TOTP, et nous avons mis en place un tunnel VPN SSL en utilisant le pare-feu ASA. En conséquence, nous atteignons notre objectif, bien que le projet puisse pas être réalisé dans un réseau réel, nous l'implémentons à l'aide d'un outil de simulation.

<i>Introduction</i>	1
<i>1. Information System Security</i>	3
1.1 OSI Security Architecture	3
1.1.1 Security Services	3
1.1.2 Security Mechanisms	5
1.1.3 Security Attack	6
1.2 Cryptography	7
1.2.1 Classes of Cryptographic Systems	7
1.2.2 Advanced Encryption Standard	10
1.2.3 Modes of Operation	12
1.2.4 The RSA Public Key Cryptosystem	14
1.2.5 Secure Hash Algorithm-2	15
1.2.6 Cryptographic Hash Functions and Message Authentication Codes	16
1.2.7 Random Number Generators	17
1.2.8 Public Key Infrastructure	17
1.3 Open Authentication OATH	19
1.3.1 Authentication System	19
1.3.2 One-time Password System	21
1.3.3 HMAC-Based One-Time Password Algorithm	21
1.3.4 Time-Based One-Time Password Algorithm	21
1.4 Virtual Private Network Technology	22
1.4.1 VPN Protocols	23
1.4.2 Remote Access Technologies	24

1.4.3	SSL VPN	25
1.4.4	Security Threats	28
1.4.5	Cisco SSL VPN Product	29
2.	<i>Preliminary Study</i>	31
2.1	Presentation of the Host Organization	31
2.1.1	Description	31
2.1.2	Organization of the Central Bank of Algeria	31
2.2	Specifications Notebook	33
2.2.1	Project Framework	33
2.2.2	Functional Specification	34
2.2.3	Technical Specification	39
2.3	Planning	40
3.	<i>OATH-OTP Analyze and Conception</i>	42
3.1	Analyze	42
3.1.1	Identification of Actors	42
3.1.2	Package Diagram	42
3.1.3	Use Case Diagram-Open Authentication Data Access	44
3.1.4	Use Case Users' Websit	50
3.1.5	Use Case AuthOADA	51
3.2	Conception	53
3.2.1	Open Authentication Data Access Functions	53
3.2.2	Users Authentication	60
3.3	Database Conception	62
3.3.1	Internal Database	62
3.3.2	Mobile Database	62
4.	<i>Strong Authentication Architecture - OTP Mechanism Implementation</i>	63
4.1	Software Used	63
4.2	OADA Administrator Tool Realization	64
4.2.1	Data Server configuration	68
4.2.2	OADA Installer	68
4.3	User Authentication	69
4.3.1	Web Site Realization	69
4.3.2	OTP Generator Application Realization	70

5. <i>Strong Authentication Architecture-Setup VPN SSL Tunnel</i>	72
5.1 Network Topology	72
5.2 Software Used	74
5.3 The GNS3 Environment	74
5.3.1 The GNS3 VM Configuration	74
5.3.2 Establishing Network Topology	75
6. <i>Result And Interpretation</i>	89
6.1 OADA Application	89
6.1.1 OADA Instalation Process	89
6.1.2 OADA Application Examination	91
6.2 VPN SSL Connection	96
6.3 User Authentication	98
6.4 Results Discussion	101
<i>Conclusion</i>	102
<i>Annex</i>	103
<i>Bibliography</i>	107

LIST OF FIGURES

1.1	AES Process.	10
1.2	The AES S-box.	11
1.3	Matrix M	12
1.4	Matrix M^{-1}	12
1.5	ECB Mode.	13
1.6	CBC Mode.	14
1.7	Counter Mode.	14
1.8	Matrix RSA Cryptosystem.	15
1.9	IPsec Site-to-Site VPN Tunnel.	23
1.10	Remote Access VPN Tunnel.	24
1.11	SSL and TCP/IP.	26
1.12	SSL/TLS Protocol Structure.	27
2.1	Source : http://www.bank-of-algeria.dz	33
2.2	Application Icon.	36
2.3	Model-Admin Login.	37
2.4	Model-Admin Manage Users	37
2.5	Model-user Login.	38
2.6	Model-user OTP Authentication.	38
2.7	Model-OTP Generator.	39
2.8	FAST Diagram.	41
3.1	Package Diagram.	43
3.2	Use case Diagram Open Authentication Data Access.	44
3.3	Use case Diagram Users Websit.	50
3.4	Use case Diagram AuthOADA.	51

3.5	Sequence Diagram-Admin Login.	53
3.6	Sequence Diagram-Add User.	53
3.7	Sequence Diagram-Delete User.	54
3.8	Sequence Diagram-Search for User.	54
3.9	Sequence Diagram-Display Users List.	55
3.10	Sequence Diagram-Account Recovery.	55
3.11	Sequence Diagram- Update Password.	56
3.12	Sequence Diagram-Server Configuration.	56
3.13	OADA State Diagram- Main Interface.	57
3.14	OADA Manage Users State Diagram.	58
3.15	OADA Manage Users State Diagram- Edit List.	59
3.16	Sequence Diagram- User Login.	60
3.17	Sequence Diagram-Scan New Code.	61
3.18	Sequence Diagram-Get OTP.	61
3.19	Class Diagram-Internal Database.	62
3.20	Class Diagram-Mobile Database.	62
4.1	Login Interface.	64
4.2	Manage Users Interface.	64
4.3	Add new User Interface.	65
4.4	Delete User Interface.	65
4.5	Search for User Interface.	66
4.6	Recover User account Interface.	66
4.7	Display users list Interface.	67
4.8	Server Configuration Interface.	67
4.9	Document Interface.	68
4.10	Inno-setup File	68
4.11	OADA Launcher.	69
4.12	Login Interface.	69
4.13	OTP Authentication Interface.	70
4.14	OTP Generator Application.	70
4.15	OTP Generator Application Scanner.	71
5.1	The Topology of VPN SSL Tunnel.	73
5.2	GNS3 VM Configuration.	75
5.3	GNS3 VM.	75
5.4	GNS3 Network Topology.	76
5.5	GNS3 ASA Interfaces Configuration.	76

5.6	GNS3 ASA Interfaces Parameters Configuration.	76
5.7	GNS3 ASA Rout Table Configuration.	77
5.8	GNS3 ASA DNS Configuration.	77
5.9	GNS3 Router Configuration.	77
5.10	GNS3 Router Routing Configuration.	78
5.11	GNS3 Cloud Configuration.	78
5.12	GNS3 PcAdmin Configuration.	79
5.13	GNS3 Linux QEMU Configuration.	79
5.14	GNS3 Windows QEMU Configuration.	80
5.15	GNS3 ASA ASDM.	80
5.16	GNS3 ASA ASDM JAVA Instalation.	81
5.17	GNS3 ASA ASDM Connect.	81
5.18	GNS3 ASA ASDM Interface.	82
5.19	GNS3 ASA ASDM ICMP Enabling.	83
5.20	Web Application Deployment.	84
5.21	ASDM Interface	84
5.22	AnyConnect Wizard Interface	85
5.23	AnyConnect Wizard Interface Enable SSL	85
5.24	AnyConnect Wizard Interface AnyConnect Client image	86
5.25	AnyConnect Wizard Interface AAA Server	86
5.26	AnyConnect Wizard Interface SAML Configuration	87
5.27	AnyConnect Wizard Interface AnyConnect Address Pool	87
5.28	AnyConnect Wizard Interface AnyConnect DNS	88
5.29	AnyConnect Profile	88
6.1	Instalation Process.	90
6.2	OADA Application.	91
6.3	OADA Server Configuration Interface.	92
6.4	OADA Application Login Interface.	92
6.5	OADA Application Manage Users Interface.	92
6.6	OADA Application ADD Users Interface.	93
6.7	OADA Application ADD Users Interface Generate QRcode.	93
6.8	OADA Application Users List Interface	94
6.9	OADA Application Find Users- Exist	94
6.10	OADA Application Find Users-Does not Exist	94
6.11	OADA Application Delete Users-Invalid User/Password	95
6.12	OADA Application Delete Users-Valid User/Password	95
6.13	OADA Application About	95

6.14	OADA Application Docs	96
6.15	ASA Connection.	96
6.16	Download AnyConnect.	97
6.17	Instalation of AnyConnect.	97
6.18	VPN SSL Conection.	98
6.19	IP Address.	98
6.20	AuthOADA Android App.	99
6.21	Web Login Interface.	99
6.22	Web TOTP Authentication Interface.	100
6.23	OTP Code.	100
6.24	Web Profile Servlet.	101
6.25	Access Denied Servlet.	101
6.26	AES Key Derivation	103
6.27	AES Encrypt Code.	103
6.28	AES Decrypt Code.	104
6.29	SHA-256 HASH Code.	104
6.30	HMac Code.	104
6.31	Conect to Database.	105
6.32	TOTP Function.	105
6.33	CurrentTime unixEpoch.	105
6.34	Getdecimal Function.	106

LIST OF TABLES

1.1	Secure Hash Algorithm Characteristics.	16
1.2	Structure of X.500.	19
1.3	Remote Access VPN Technologies	25
3.1	Login Case	45
3.2	Add User Case	46
3.3	Delete User Case	47
3.4	Search for User Case	47
3.5	Account Recovery Case	48
3.6	Display Users List Case	49
3.7	Update Password Case	49
3.8	Server Configuration Case	50
3.9	Users Login Case	51
3.10	Scan Case	52
3.11	Get OTP Case	52

LIST OF ABBREVIATIONS

2FA.	TWO-FACTOR AUTHENTICATION
AES.	ADVANCED ENCRYPTION STANDARD
ARP.	ADDRESS RESOLUTION PROTOCOL
ASA.	ADAPTIVE SECURITY APPLIANCE
ASDM.	ADAPTIVE SECURITY DEVICE MANAGER
CA.	CERTIFICATE AUTHORITIES
CBC.	CIPHER BLOCK CHAINING
DNS.	DOMAIN NAME SYSTEM
ECB.	ELECTRONIC CODE BOOK
FTP.	FILE TRANSFER PROTOCOL
GRE.	GENERIC ROUTING ENCAPSULATION
GNS3.	GRAPHICAL NETWORK SIMULATOR
HMAC.	KEYED-HASH MESSAGE AUTHENTICATION CODE
HOTP.	HMAC-BASED ONE-TIME PASSWORD
HTTP.	HYPERTEXT TRANSFER PROTOCOL
HTTPs.	HYPERTEXT TRANSFER PROTOCOL SECURE
IDS.	INTRUSION DETECTION SYSTEM
IETF.	INTERNET ENGINEERING TASK FORCE
IMAP.	INTERNET MESSAGE ACCESS PROTOCOL

IOS.	INTERNETWORK OPERATING SYSTEM
IP.	INTERNET PROTOCOL
IPsec.	INTERNET PROTOCOL SECURITY
ISO.	INTERNATIONAL STANDARDS ORGANISATION
L2F.	LAYER 2 FORWARDING
L2TP.	LAYER 2 TUNNELING PROTOCOL
MAC.	MESSAGE AUTHENTICATION CODE
MFA.	MULTI FACTOR AUTHENTICATION
MPLS.	MULTI PROTOCOL LABEL SWITCHING
NAT.	NETWORK ADDRESS TRANSLATION
OATH.	OPEN AUTHENTICATION
OTP.	ONE-TIME PASSWORD
OSI.	OPEN SYSTEMS INTERCONNECTION
PKI.	PUBLIC KEY INFRASTRUCTURE
POP3.	POST OFFICE PROTOCOL
PPTP.	POINT-TO-POINT TUNNELING PROTOCOL
RNG.	RANDOM NUMBER GENERATORS
RA.	REGISTRATION AUTHORITIES
SHA-2.	SECURE HASH ALGORITHM-2
SMTP.	SIMPLE MAIL TRANSFER PROTOCOL
SSL.	SECURE SOCKET LAYER
TCP.	TRANSMISSION CONTROL PROTOCOL
TOTP.	TIME-BASED ONE-TIME PASSWORD
TLS.	TRANSPORT LAYER SECURITY
UDP.	USER DATAGRAM PROTOCOL
VPN.	VIRTUAL PRIVATE NETWORK

ACKNOWLEDGMENT

First and foremost, I would like to thank God for blessing me to have the strength, knowledge, ability to do this research study, and complete it satisfactorily. Then to my loving parents and my family for their encouragement, moral support, and care.

I would like to thank my supervisor at Algeria bank Mem. Fatima Zohra Adi also the Director of Algeria bank Ms. Ben Bahanne.

I would like to express my deep appreciation to my teachers Mem. Boustia and Mem. Arkam for all the support i got also,i would like to thank Ms.Guesmia Depaetement chef for all the help we got during this year.

I would also like to express my gratitude to Dr. Ben. A Habiba, B.Slimane at Ghardaia University, who have, in their ways, kept me going on my path to success.

I have great pleasure in acknowledging my gratitude to all my teachers at Blida University and especially to my colleagues.

Information security has been among the laboratory's research interests for many years, and as much as information systems data access has changed over time, some challenges have appeared facing authentication authorization and access control. Most enterprises reach for taking control of the organization's security in terms of authentication and remote access workforce.

With the increase of cybersecurity threats nowadays, the process of ensuring that only the right people get access to the proper information has become much more pervasive, complicated, and essential. A standard password is a usual approach to authenticate the user's login. But users often create simple and easy to remember passwords, which makes them vulnerable to a set of cyberattacks and easy to steal, which can be a real security issue for enterprises trying to remote users access their network.

In the early 2000s, the death of the password - as a single measure of account protection - was predicted by *Bill Gates*. The passwords are the oldest single-factor authentication method used with computing systems since 1961 when the first computer system implemented a password login. Over the last few decades, several solutions were provided in order to improve access control, strong authentication has become a popular term in the cybersecurity industry, including the use of open authentication (OATH) and VPNs.

OATH is presenting solutions that allow for trusted, universal authentication, adopting open standards to improve the approval of strong authentication solutions that simplify the use and progress appropriation of two-factor authentication. VPNs allow the creation of a secure connection to another internal network resources across the internet.

Due to the COVID-19 situation - the quarantine imposing for 2020-, many com-

panies are encouraging the work from home, which enforces the need for better remote access policies. In this thesis, we provide a trusted authentication architecture for remote access users by binding the oath OTP (TOTP) solution with VPN SSL.

The principal objective of this project is to strengthen the authenticate to Privileged Access Service by implementing a two-factor authentication java application based on OATH and set up remote access to internal resources using Cisco ASA VPN SSL.

In addition to the general introduction and the general conclusion, we have subdivided this work into six chapters: In the first and second chapter, we cite related literature to our work and the information system security in general, in the third chapter, we present the conception of the strong authentication architecture proposed, and in the fourth and fifth chapters, we indicate the implementation process. Next in the sixth chapter, we will discuss the interpretation of the result we achieve.

1.1 OSI Security Architecture

The IETF Internet Security Glossary; published in RFC 2828; defines the security architecture as "a plan and set of principles that describe (a) the security services that a system is required to provide to meet the needs of its users, (b) the system elements required to implement the services, and (c) the performance levels required in the elements to deal with the threat environment". The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined as:

- Security service: A processing or communication service that improves the security of the data processing systems and the information transfers of an organization.
- Security mechanism: A process (or a device) that is designed to detect, prevent, or recover from a security attack.
- Security attack: Any activity that endangers the security of information owned by an organization.

1.1.1 Security Services

The OSI security architecture divides security services into five classes :

1.1.1.1 Authentication Services

Authentication service provides the authentication of a communicating peer entity or data origin:

1. A peer entity authentication service verifies that the peer entity is what it claims to be, which assurance that an entity is not attempting to masquerade or performing an unauthorized replay.
2. A data origin authentication service verifies the source of data received during data exchange. The data origin authentication service must be complemented with a data integrity service to provide protection against the duplication or modification of data units.

Authentication services are necessary for the management of authorization, access control, and accountability services.

1.1.1.2 Access Control Services

Access control services' purpose is to guard system resources against unauthorized use. Access control services are related to authentication services; a user or process must be authenticated before an access control service can be performed.

1.1.1.3 Data Confidentiality Services

Data confidentiality refers to that data is not accessible or exposed to unauthorized individuals, entities, or processes. Confidentiality services protect data from unauthorized access. There are various forms of such services:

- A connection confidentiality service provides confidentiality for all data exchange across the network.
- A connectionless confidentiality service provides confidentiality for different data units.
- A selective field confidentiality service affords confidentiality for some fields within individual data units or data transmitted in a connection.
- Traffic-Flow confidentiality refers to the protection of the information that might be derived from the observation of traffic flows.

1.1.1.4 Data Integrity Services

Integrity services protect data from unauthorized modification. There are certain forms of such services:

- A connection integrity service with recovery provides integrity for all data transmitted in a connection. If possible, the loss of integrity is recovered.

- A connection integrity service without recovery is similar to a connection integrity service with recovery, except that the loss of integrity is not recovered.
- A selected field connection integrity service provides integrity for specific fields within the data transmitted in a connection.
- A connectionless integrity service provides integrity for individual data units.
- A selected field connectionless integrity service provides integrity for specific fields within individual data units.

1.1.1.5 Nonrepudiation Services

Nonrepudiation services are implemented to prevent an entity involved in communication from later denying having participated in all or part of the communication. There are at least two nonrepudiation services that are relevant in practice:

- A nonrepudiation service with proof of origin provides the recipient of a message with a proof of origin.
- A nonrepudiation service with proof of delivery provides the sender of a message with a proof of delivery.

1.1.2 Security Mechanisms

1.1.2.1 Specific Security Mechanisms

Specific security mechanisms may be incorporated into an appropriate layer to provide some of the security service.

1. Encipherment can be used to protect the confidentiality of data units or to support or complement other security mechanisms.
2. Digital signature can be used to provide an electronic analog of handwritten signatures for electronic documents.
3. Access control can be used to control access to system resources.
4. Data integrity, variety of mechanisms used to assure the integrity of a data unit or stream of data units.
5. Authentication exchange, this mechanism intended to ensure the identity of an entity by means of information exchange.

6. Traffic padding, the insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.
7. Routing control by the selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
8. Notarization, the use of a trusted third party to assure certain properties of data exchange.

1.1.2.2 Pervasive Security Mechanisms

Mechanisms that are not specific to any particular OSI security service or protocol layer.

1. Trusted functionality, that which is perceived to be correct with respect to some criteria.
2. Security label, the marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
3. Event detection, the detection of security-relevant events.
4. Security audit trail, data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
5. Security recovery, deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

1.1.3 Security Attack

1.1.3.1 Passive Attacks

Passive attacks are like eavesdropping on, or monitoring of communications. The goal of the adversary is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

1. In the release of message content, the opponent tries to obtain sensitive or confidential information within the communication.
2. With the traffic analysis, the opponent tries to determine the location and identity of communicating hosts, then observe the frequency and length of

messages being exchanged. This information might be useful in suggesting the nature of communication.

Passive attacks are extremely difficult to detect because they do not involve any data alteration.

1.1.3.2 Active Attacks

Active attacks involve any modification of the data stream or the creation of a false stream and can be divided into four categories: masquerade, replay, modification of messages, and denial of service.

1. Masquerade is identity theft when one entity pretends to be a different entity.
2. Replay includes the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
3. Modification of messages includes the portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized impact.
4. The denial of service prevents or inhibits the normal performance of communications facilities.

Active attacks are detectable but it is quite difficult to prevent. Instead, the goal is to detect active attacks to facilitate recovery from any disruption or delays caused by them.

1.2 Cryptography

Cryptography is a universal term used to define the design and analysis of mechanisms based on mathematical techniques that produce fundamental security services. Cryptography is used in a generic sense, but a more formally correct term is cryptology, which is the scientific art of cryptography (the design of such mechanisms) and cryptanalysis (the analysis of such mechanisms), providing solutions for the need of reinforcing data integrity and authentication due to the rapidly increasing of electronic communication.

1.2.1 Classes of Cryptographic Systems

There are three classes of cryptographic systems.

Definition 1.2.1: (**Unkeyed cryptosystem**) *An unkeyed cryptosystem is a cryptographic system that uses no secret parameter.*

Representatives of unkeyed cryptosystems are one-way functions (hash functions).

Definition 1.2.2: (**Secret key cryptosystem**) *A secret key cryptosystem is a cryptographic system that uses secret parameters that are shared between the participating entities.*

Definition 1.2.3: (**Public key cryptosystem**) *A public key cryptosystem is a cryptographic system that uses secret parameters that are not shared between the participating entities.*

1.2.1.1 Unkeyed Cryptosystems

Definition 1.2.4: (**One-way function**) *A function $f : X \rightarrow Y$ is one way if $f(x)$ can be computed efficiently for all $x \in X$, but $f^{-1}(y)$ cannot be computed efficiently for any randomly chosen $y \in Y$.*

Definition 1.2.5: (**Trapdoor function**) *A one-way function $f : X \rightarrow Y$ is a trapdoor function (or a trapdoor one-way function, respectively) if there exists some extra information (i.e., the trapdoor) with which f can be inverted efficiently, that is, $f^{-1}(y)$ can be computed efficiently for any randomly chosen $y \in Y$.*

Definition 1.2.6: (**Hash function**) *Let Σ_{in} be an input alphabet and Σ_{out} be an output alphabet. Any function $h : \Sigma_{in} \rightarrow \Sigma_{out}$ that can be computed efficiently is said to be a hash function. It generates hash values of length n .*

Definition 1.2.7: (**Cryptographic hash function**) *A hash function $h : \Sigma_{in} \rightarrow \Sigma_{out}$ is cryptographic if it is one way or collision resistant.*

In cryptography, we are interested in hash functions with the following characteristics:

- A hash function h is *one-way or preimage resistant* if it is computationally infeasible to find an input word $x \in \Sigma_{in}^*$ in with $h(x) = y$ for any given (and randomly chosen) output word $y \in \Sigma_{out}^n$.
- A hash function h is *second-preimage resistant* or *weak collision resistant* if it is computationally infeasible to find a second input word $x' \in \Sigma_{in}^*$ in with $x' \neq x$ and $h(x') = h(x)$ for any given (and randomly chosen) input word $x \in \Sigma_{in}^*$.

- A hash function h is *collision resistant* or *strong collision resistant* if it is computationally infeasible to find two input words $x, x' \in \Sigma_{in}^*$ with $x' \neq x$ and $h(x') = h(x)$.

1.2.1.2 Secret Key Cryptosystems

Definition 1.2.8: (**Symmetric encryption system**) *A symmetric encryption system or cipher consists of the following five components:*

- a plaintext message space \mathcal{M} ;
- a ciphertext space \mathcal{C} ;
- a key space \mathcal{K} ;
- a family $E = E_k : \{k \in \mathcal{K}\}$ of (deterministic or probabilistic) encryption functions $E_k : \mathcal{M} \rightarrow \mathcal{C}$;
- a family $D = \{D_k : k \in \mathcal{K}\}$ of (deterministic) decryption functions $D_k : \mathcal{C} \rightarrow \mathcal{M}$.

For every key $k \in \mathcal{K}$ and every message $m \in \mathcal{M}$, the functions D_k and E_k must be inverse to each other, that is, $D_k(E_k(m)) = m$.

Definition 1.2.9: (**Block Ciphers**) *A block cipher operates on fixed-length groups of bits (i.e., blocks) with an unvarying transformation (determined by the key).*

Definition 1.2.10: (**Stream Cipher**) *A stream cipher operates on individual bits or bytes, and the actual transformation varies during the encryption process.*

1.2.1.3 Public Key Cryptosystems

Definition 1.2.11: (**Asymmetric encryption system**) *An asymmetric encryption system consists of the following three efficiently computable algorithms:*

- Generate (1^n) is a probabilistic key generation algorithm that takes as input a security parameter 1^n and generates as output a public key pair (consisting of a public key k and a corresponding private key k^{-1}).
- Encrypt (k, m) is a deterministic or probabilistic encryption algorithm that takes as input a public key k and a plaintext message m , and that generates as output a ciphertext c (i.e., $c = \text{Encrypt}(k, m)$).
- Decrypt (k^{-1}, c) is a deterministic decryption algorithm that takes as input a private key k^{-1} and a ciphertext c , and that generates as output a plaintext message m (i.e., $m = \text{Decrypt}(k^{-1}, c)$).

1.2.2 Advanced Encryption Standard

Operation of AES

The Advanced Encryption Standard (AES) ciphers are the most popular and widely used symmetric encryption algorithms. It comprises rounds of combined procedures occurs in blocks of 16 characters, or, equivalently, 128 bits, the number of rounds in AES is variable and depends on the length of the key. AES does 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. The AES ciphers algorithms includes three fundamental steps shown in Figure 1.1:

1. encryption;
2. decryption;
3. the key expansion.

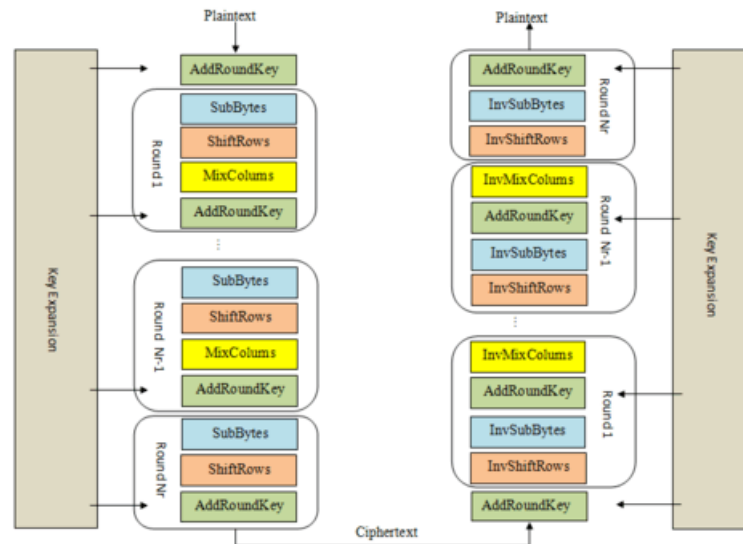


Fig. 1.1: AES Process.

The S-box

The S-box is used in the Advanced Encryption Standard to transform a given byte into another byte. The AES S-box contains 16 rows and 16 columns, each labeled with the hexadecimal representations of the decimal numbers 0 through 15, in order. To use the S-box, first convert the input byte from binary into an input two-digit hexadecimal number, then find an output two-digit hexadecimal number as the entry in the S-box where the row marked with the first digit of the input hexadecimal number, and the column marked with the second digit of the

input hexadecimal number. Finally, we convert this output two-digit hexadecimal number into binary. Figure 1.2.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Fig. 1.2: The AES S-box.

AES Encryption

The plaintext are formatted in blocks of 128 bits (4×4 plaintext matrix). AES ciphers use a variety of different steps and types of operations for encryption:

1. Byte Substitution (ByteSub): The entries in an input matrix are transformed using the S-box.
2. ShiftRow: The entries in an input matrix are shifted to the left with the entries at the left wrapping to the right.
 - (a) the first row is not shifted;
 - (b) the second row is shifted one position to the left;
 - (c) the third row is shifted two positions to the left;
 - (d) the fourth row is shifted three positions to the left.
3. MixColumn: An input matrix is multiplied on the left by a fixed matrix M . Figure 1.3.
4. AddRoundKey: this procedure uses the XOR operation to add to the input matrix the key given by the key schedule for the specific round.

$$M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Fig. 1.3: Matrix M .

AES Decryption

To decrypt a ciphertext that was created utilizing the AES encryption process, we must use the inverses of the encryption operations:

1. InvByteSub: The entries in an input matrix are transformed using the inverse of the S-box.
2. InvShiftRow: The entries in an input matrix are shifted to the right by zero, one, two, or three positions, with the entries at the right wrapping to the left.
3. InvMixColumn: An input matrix is multiplied on the left by a fixed matrix M^{-1} : Figure 1.4.
4. AddRoundKey is its own inverse.

$$M^{-1} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix}$$

Fig. 1.4: Matrix M^{-1} .

The key Expansion

The key Expansion process is used to generate a key schedule. The key expansion specifies how ExpandedKey is derived from the cipher key. The total number of bits in ExpandedKey is equal to the block length multiplied by the number of rounds plus 1.

1.2.3 Modes of Operation

Block ciphers are extremely varied cryptographic primitives that can be used to afford plenty of different security properties. There are three different modes, ECB, CBC, and Counter Mode .

1.2.3.1 Electronic Code Book Mode

In the first mode called the Electronic Code Book (ECB), we take the first block of plaintext and encrypt it with the key to produce the first block of ciphertext. We then take the second block of plaintext and encrypt it with the key to produce the second block of ciphertext, and so on. Figure 1.5.

The origin of the name for this mode of operation comes from the fact that, once the key is determined, encryption could (at least in theory) be conducted using an enormous codebook that is consulted to attain which ciphertext block replaces which plaintext block.

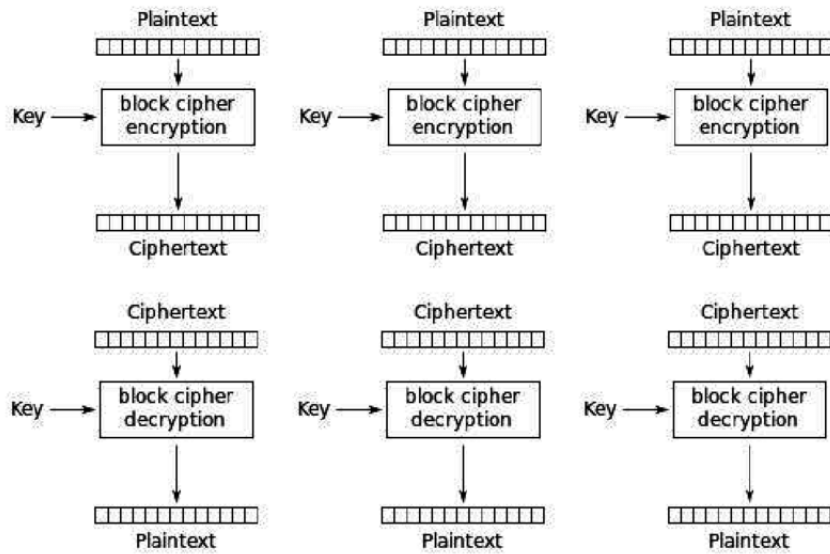


Fig. 1.5: ECB Mode.

1.2.3.2 Cipher Block Chaining Mode

In the Cipher Block Chaining (CBC) mode, each ciphertext block, as well as being sent to the receiver, is used as an input into the encryption process of the following plaintext block. As a result, all the ciphertext blocks are computationally 'chained' together. Figure 1.6.

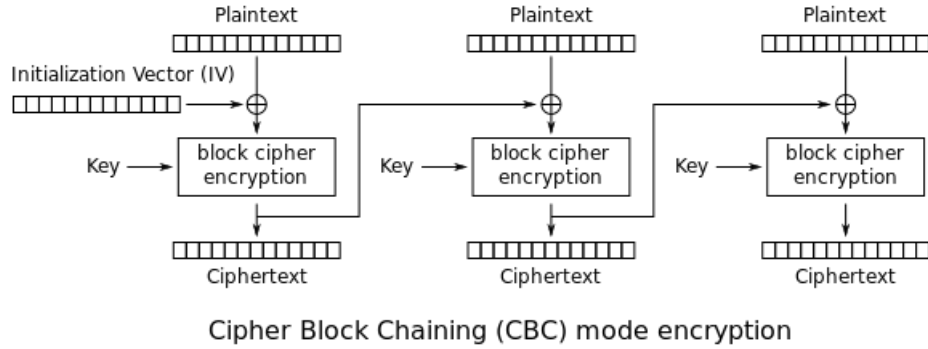


Fig. 1.6: CBC Mode.

1.2.3.3 Counter Mode

In this mode, we assume both the sender and receiver have access to a reliable counter, which computes a new shared value every time a ciphertext block is exchanged, both sides must keep the counter synchronized. In this mode, it no longer about encrypting the message but encrypting a counter that is used once, The counter will be encrypted using a secret key to produce a ciphertext that will be XORed with the original message blocks. Figure 1.7.

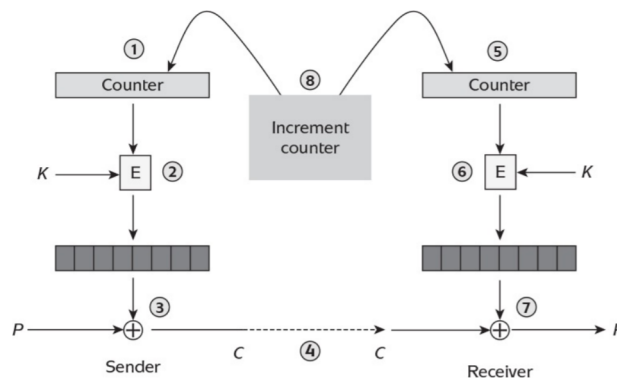


Fig. 1.7: Counter Mode.

1.2.4 The RSA Public Key Cryptosystem

The RSA public key cryptosystem was designed by Ron Rivest, Adi Shamir, and Len Adleman in 1977. It was the first viable implementation of the approaches developed by Diffie and Hellman in the previous year. As such, the RSA public key cryptosystem allows both an asymmetric encryption system and a digital signature system. The RSA asymmetric encryption system steps are:

- The RSA generate algorithm first randomly selects two appropriately sized

prime numbers p and q and computes the RSA modulus $n = pq$, then it randomly selects an integer $1 < e < \phi(n)$ with $\gcd(e, \phi(n)) = 1$ and computes another integer $1 < d < \phi(n)$ with $de \equiv 1 \pmod{\phi(n)}$ using, for example, the *Extended Euclid* algorithm. d then represents the multiplicative inverse of e modulo $\phi(n)$. The output of the algorithm is a public key pair that consists of a public key (n, e) and a corresponding private key d .

- The RSA encrypt algorithm is deterministic. It takes as input a public key (n, e) and a plaintext message $m \in \mathbb{Z}_n$, and it generates as output the ciphertext $c = m^e \pmod{n}$.
- The RSA decrypt algorithm is deterministic, too. It takes as input a private key d and a ciphertext c , and it generates as output the corresponding plaintext message $m = c^d \pmod{n}$.

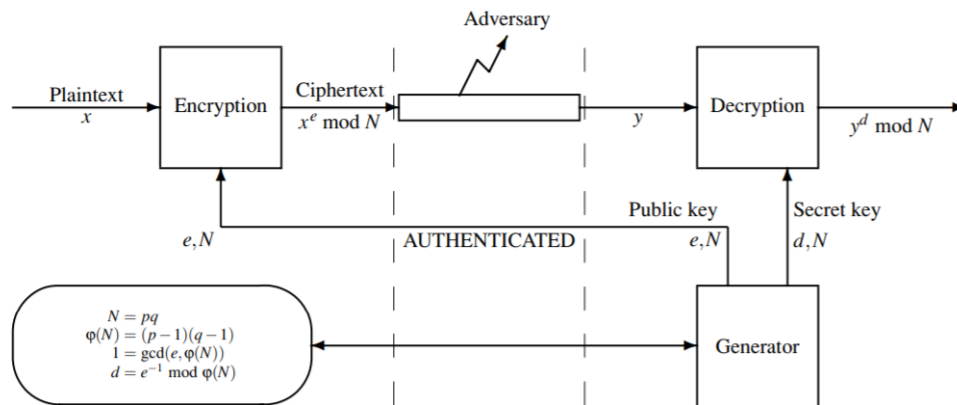


Fig. 1.8: Matrix RSA Cryptosystem.

1.2.4.1 Advantages of RSA

RSA presents a few improvements that have helped in the performance of information security management. These advantages are:

- simplification of the problem of key management;
- enhanced security of the transactions.

1.2.5 Secure Hash Algorithm-2

The SHA-2 hash standard specifies four secure hash algorithms, SHA-224, SHA-256, SHA-384, and SHA-512. All four of the algorithms are iterative, one-way hash functions that can treat a message to compose a hashed representation called a *message digest*. Each algorithm can be described in two steps:

1. Preprocessing includes preparing the message through the padding, parsing the padded message into m -bit blocks, and arranging any initialization values to be used in the hash generation.
2. The hash computation generates a message schedule from the padded message which is used, along with functions, constants, and word operations, to iteratively generate a series of hash values.

The final hash value generated by the hash computation is used to determine the *message digest*. A message M of length l to be hashed is processed by blocks of m bits. Each block is subdivided into 16 w -bit words for computation, the word-size w depending on the algorithm. The most significant difference between the four algorithms is the size of *the message digest*. Additionally, the algorithms differ in terms of the size of the blocks and words of data that are used during hashing. Table 1.1.

Algorithm	Word (w)	Message size (l)	Block (m)	Digest	Security
SHA-224	32	$< 2^{64}$	512	224	112
SHA-256	32	$< 2^{64}$	512	256	128
SHA-384	32	$< 2^{128}$	1024	384	192
SHA-512	32	$< 2^{128}$	1024	512	256

All sizes are given in bits

Tab. 1.1: Secure Hash Algorithm Characteristics.

1.2.6 Cryptographic Hash Functions and Message Authentication Codes

Message authentication code (MAC) is a cryptographic checksum that is used to ensure the integrity of the message during transmission. To generate a MAC, we use either an encryption algorithm or a hashing algorithm which is generally much faster than encryption algorithms. Furthermore, to provide confidentiality, it's required to encrypt the message. The MAC can be appended to the message before encryption.

HMAC uses a hash function and a symmetric-key encryption algorithm to generate authentication codes. The basic idea of HMAC is to embed the secret information of the key into the data and then compute a hash value from it.

The design of the HMAC is very simple. Let h be a hash function, and let K_1 and K_2 be two symmetric keys. Then the MAC on message M is computed as follows:

1. Compute the hash of K_2 concatenated with the message.

2. Compute the hash of K_1 concatenated with the output of step 1.

$$h(k_1 || h(k_2 || M))$$

The security of the message authentication mechanism depends on cryptographic characteristics of the hash function used.

1.2.7 Random Number Generators

Randomness is found everywhere in cryptography; in the generation of secret keys, in encryption schemes, and even in the attacks on cryptosystems. Without randomness, cryptography would be impracticable because the whole operations would become predictable, and therefore insecure.

Using RNGs is the key to making cryptography efficient and secure. The randomness comes from the environment, which is analog, chaotic, uncertain, and so unpredictable. Randomness can't be generated by computer-based algorithms only. In cryptography, randomness usually comes from random number generators (RNGs), which are software or hardware components that leverage entropy in the analog world to produce unpredictable bits in a digital system.

RNGs can also harvest the entropy in a running operating system by drawing from attached sensors, I/O devices, network or disk activity, system logs, running processes, and user activities such as keypresses and mouse movement. Such system- and human-generated activities can be a good source of entropy, but they can be fragile and manipulated by an attacker. Also, they're slow to yield random bits.

1.2.8 Public Key Infrastructure

Cryptographic algorithms are deployed in the network applications by using Public-key cryptography to distribute secret keys over the open networks. PKIs consist of one or more management entities responsible for generating, distributing, and providing ongoing support for public-key certificates. This involves a sort of different types of entity, including:

1. CAs who are responsible for generating public key certificates under a defined certification practice statement, and so that the certificates can be interpreted subject to a defined certificate policy.
2. Registration authorities (RAs), who are bound for verifying the identities of individuals demanding the generation of a certificate by a CA.

3. Certificate repositories, which holds and make available public key certificates.
4. Certificate status servers, who afford on-line information about the current status of a certificate.

The term digital certificate has been introduced to cover a larger class of objects than just public-key certificates. A digital certificate confirms that a particular public key belongs to a particular user.

1.2.8.1 Digital Certificates

A major task of a PKI is to provide authenticity proofs for public keys. Essential tools that are used in such proofs are certificates.

1.2.8.2 Digital Certificates Authentication

Digital Signature Authentication

The user authenticates himself to the server by signing the challenge that sent by the server then send it back, so the server can verify the signature with the user's certificate.

Exchanging Keys Authentication

In this protocol, the server encrypts a random produced secret key using the user's public key defined in his certificate. The user decrypts the challenge with the private key and gets the secret key, then the user encrypts an agreed value and sends this back to the server. The server decrypts the response and verifies the result. If it is correct, the authentication was successful.

Mutual Authentication

In this protocol, both user and server must authenticate themselves to each other using the protocols just described.

1.2.8.3 X.509 Certificates

The best-known standard for directory services is ITU-T X.500, X.500. X.500 is a typical ISO standard, it provides many functions, but is consequently very complex.

Structure of X.500

Field	Description
Version	Specifies the X.509 version being used (in this case V3).
Serial Number	Unique identifier for the certificate.
Signature	Digital signature algorithm used to sign the certificate.
Issuer	Name of the creator of the certificate.
Validity	Dates and times between which the certificate is valid.
Subject	Name of the creator of the certificate.
Public-key Info	Public-key value; Identifier of public-key algorithm.
Issuer ID	Optional identifier for certificate creator.
Subject ID	Optional identifier for certificate owner.
Extensions	Key usage (specifies usage restrictions); Location of revocation information; Identifier of policy relating to certificate; Alternative names for owner.

Tab. 1.2: Structure of X.500.

1.3 Open Authentication OATH

The Open Authentication Reference Architecture (OATH) initiative is a group of companies working together to help drive the adoption of open strong authentication technology across all networks. The OATH Purposes are:

- Expand secure and safe online transactions for consumers and business users with strong 2-factor authentication.
- Leverage existing standards and create an open reference architecture for strong authentication which users and service providers can rely upon, and leverage to interoperate.
- Reduce the cost and complexity of adopting strong authentication solutions.

1.3.1 Authentication System

1.3.1.1 Authentication Factors

Knowledge-based

Commonly referred to as “something you know.” This method based on authenticating the user access by his provided account key, which is known only by the owner, to gain access to a service. This category includes traditional passwords.

Possession-based

In this method, identity will be verified by using an item that only the user would have. Such as a physical key card, or personal email account, to gain access to the desired resources.

Inheritance-based

Referred to as “something you are.” Typically biometric characteristics, such as a fingerprint, facial scan, or voice recognition, will be used to verify the user’s identity.

1.3.1.2 Authentication Techniques**Single Factor Authentication (1FA)**

Basically is username/password authentication, the single factor is something you know; your password (Knowledge Authentication). In most cases, the authentication server store only the hash of the password, rather than storing the password in plaintext. Single factor authentication methods such as the basic username/password combination are no longer sufficient enough, the hashes are vulnerable due to weak passwords because they are easy to guess, so faster password cracking tools will match the hash quickly. Complex passwords would solve this problem, but they are also too difficult to remain in memory.

Two-Factor Authentication (2FA)

Two-factor authentication (2FA) is the second layer of protection, an extra step beyond username and password by adding a second authentication factor, users must go through two layers of the authentication process, before gaining access to an account or system. 2FA is designed to prevent unauthorized access with just a stolen password.

Multi Factor Authentication (MFA)

Multi-factor authentication can combine two or three different factors, while two-factor authentication is always limited to two factors. Cybersecurity community adopting MFA due to the security threats evolution nowadays.

1.3.2 One-time Password System

The one-time password system OTP protects the secured system from external attacks on its authentication subsystem. The OTP system authenticates the user using a challenge. This challenge consists of the iteration and the seed. These values are specific to each user and were configured when the user was added to the OTP database. OTP is calculated through a hashing algorithm to obtain the single-use password, consequently, OTP is not vulnerable to either eavesdropping or password replay, or to theft or password file attacks.

The process of the OTP system requires two sides: the client and the host. On the client-side, the appropriate one-time password must be produced. On the host side, the server must verify the one-time password and allow the secure changing of the user's secret passphrase.

1.3.3 HMAC-Based One-Time Password Algorithm

The HOTP algorithm is based on an increasing counter value and static symmetric key known only to the token and the validation service. It uses the HMAC-SHA to generate the HOTP code. The HOTP formula is as follows:

$$HOTP(K, C) = Truncate(HMAC - SHA - X(K, C))$$

- C: 8-byte counter value, the moving factor. This counter MUST be synchronized between the HOTP generator (client) and the HOTP validator (server).
- K: Shared secret between client and server; each HOTP generator has a different and unique secret K.

The output of the HMAC-SHA calculation is a long suit of bits, we must truncate this value to something that can be easily entered by a user by using this formula:

$$Finalcode = OTP \bmod 10^d$$

- d is the maximum number of digits that the OTP contains.

1.3.4 Time-Based One-Time Password Algorithm

TOTP is a variant of the HOTP algorithm that specifies the calculation of one-time password value, based on a representation of the counter as a time factor.

The HOTP uses a counter shared by the server and the user. In this case, the

problem is that the generated password is valid until it is used, besides keeping the synchronization of the counter. TOTP adds a restriction to the generated code that can only be used for a limited period and the synchronization based on the Unix time epoch. The TOTP formula is:

$$TOTP = HOTP(PrivateKey, CurrentTime)$$

- **PrivateKey**: Randomly generated password known only by server and client.
- **CurrentTime**: Current time in Unix time.

However, the time changes every second, which makes it impossible for a user to transmit the code to the application. One second would be enough for the TOTP to be no longer valid and for the server to generate a new value. This is the reason why we use another formula to calculate the **CurrentTime** :

$$CurrentTime = \text{floor}((\text{unixtime}(\text{now}) - \text{unixtime}(T0)) / T1)$$

- **Unixtime(now)**: The current moment in unix time.
- **Unixtime (T0)**: The Unix time at time T0 from which the count is made(01/01/1970 at midnight).
- **T1**: Interval in which the TOTP is valid, generally 30 seconds.
- **Floor**: Function allowing to round the calculated value to an integer number.

The security and strength of this algorithm depend on the properties of the underlying building block HOTP, which is a construction based on HMAC using a hash function.

1.4 Virtual Private Network Technology

Since the advent of the Internet, network administrators have searched for ways to leverage this low-cost to transporting data and protecting data integrity and confidentiality while providing transparency to the end-user. This reproduced the concept of Virtual Private Networks (VPN). A VPN is:

- **Virtual**, because there is no actual direct network connection between the two (or more) communication entities.
- **Private**, because only the members of the group connected by the VPN Software are allowed to read the data transferred.

The IETF defined a number of VPN protocols, including Point-to-Point Tunneling Protocol (PPTP), Layer 2 Forwarding (L2F) Protocol, Layer 2 Tunneling Protocol (L2TP), Generic Routing Encapsulation (GRE) Protocol, Multiprotocol Label Switching (MPLS) VPN, Internet Protocol Security (IPsec), and Secure Socket Layer VPN (SSL VPN). A secure VPN satisfies the following basic requirements:

- authentication;
- confidentiality;
- message integrity.

1.4.1 VPN Protocols

Site-to-site VPN

Site-to-site protocols provide secure connections between two or more offices to send traffic back and forth over a shared medium such as the Internet. These connections can also be used to connect the private or semiprivate networks of an organization with the private or semiprivate networks of another organization across the shared medium. This excludes the need for dedicated leased lines to connect the remote offices to the organization's network. IPsec, GRE, and MPLS VPN are commonly used site-to-site VPN protocols. Figure 1.9 shows a simple IPsec VPN topology.

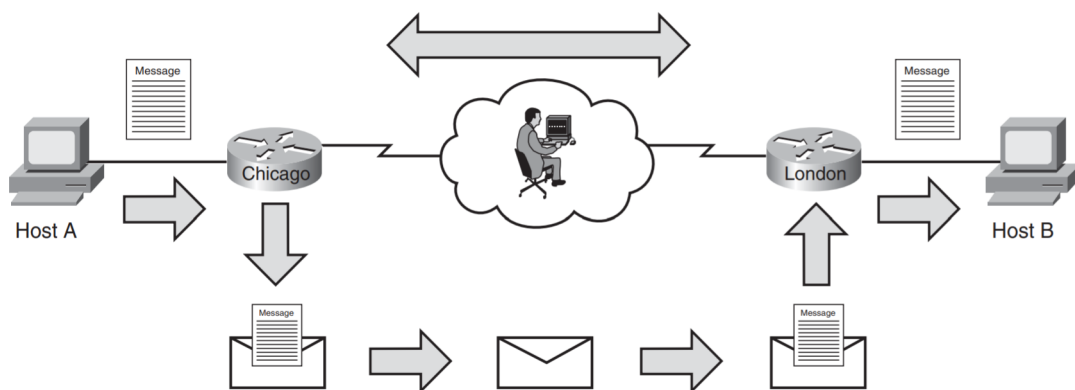


Fig. 1.9: IPsec Site-to-Site VPN Tunnel.

Remote Access VPN

The remote access protocols enabling the work from remote locations for mobile users. Organizations do not need to maintain a huge pool of modems and access

servers to provide remote users. Some commonly used remote access VPN protocols are SSL VPN, IPsec, L2TP, L2TP over IPsec, and PPTP. Figure 1.10 shows a deployment model for using remote access VPN technologies.

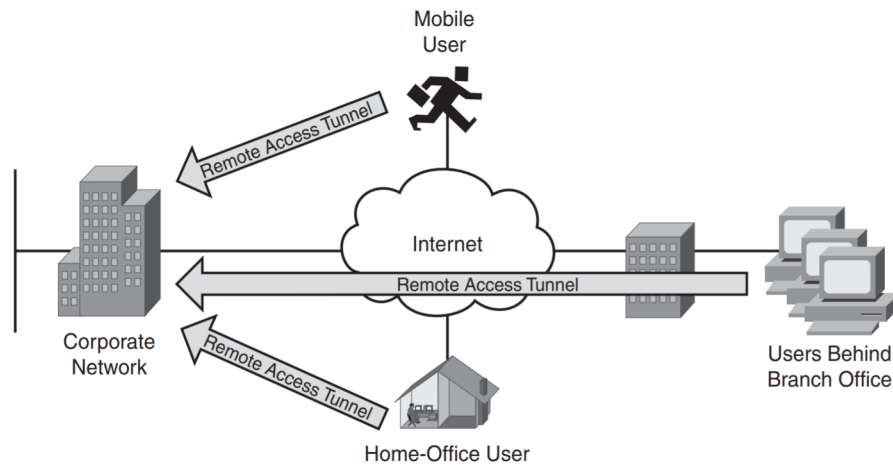


Fig. 1.10: Remote Access VPN Tunnel.

1.4.2 Remote Access Technologies

Nowadays organizations are reducing costs by leveraging modern technology in their actual network infrastructure. Besides the growth of the Internet require to provide their employees with 24/7 access to organizational resources. The increasing amount of mobile workers and telecommuters is a significant factor in the exponential growth of remote access technologies for giving users the illusion of being directly connected to the corporate LAN. Table 1.3 represents several remote access technologies exist such:

1. IPsec

IPsec is a widely used VPN technology. Because it provides protection at the IP level (Layer 3), it can be deployed to secure communication between a pair of gateways, a pair of host computers, or even between a gateway and a host computer.

2. SSL VPN

Secure Socket Layer (SSL) VPN sits between the transport and application layers of the OSI model, therefore it provides secure connectivity to the internal resources through a web browser or a dedicated client.

3. L2TP

Layer 2 Tunneling Protocol combines features from Layer 2 Forwarding

(L2F) from Cisco Systems and PPTP from Microsoft to add security features and improved encapsulation that satisfies the emerging industry requirements. It packages data within the Point-to-Point Protocol (PPP) and uses registered User Datagram Protocol (UDP) port 1701 for both tunnel negotiations and data encapsulation.

Functionality	IPsec	L2TP	SSL VPN
VPN client	Requires a third-party client	Built into newer Windows OSs	VPN client is optional
Encryption	DES, 3DES, AES	MPPE	DES, 3DES, RC4-128, RC4-40, AES
Deployment	Extensively used	Rarely used	Steady growth

Tab. 1.3: Remote Access VPN Technologies

1.4.3 SSL VPN

Secure Socket Layer (SSL) Virtual Private Network (VPN) technology has been deployed rapidly in recent years because it provides universal access and low-cost deployment and management. The SSL protocol was developed by Netscape to promote e-commerce sites that required data encryption and user authentication. No official standards exist for SSL VPN technologies; various vendors use different implementations. Secure Socket Layer (SSL) VPN is the emerging remote access technology that implements secure connectivity to the internal corporate resources through a web browser or a dedicated client. The SSL VPN solution can be customized to meet any business necessity. VPN SSL provides:

- Secure communication employing cryptographic algorithms: It offers confidentiality, integrity, and authentication.
- Ubiquitousness: The universality of SSL/TLS offers the possibility for VPN users to remotely access internal resources from anywhere.
- Low management cost.
- Effective operation with a firewall and NAT: SSL VPN operates on the same port as HTTPS (TCP/443). Most Internet firewalls, proxy servers, and NAT devices have been set up to handle TCP/443 traffic well. So there is no need for any special consideration to transport SSL VPN traffic over the network.

1.4.3.1 SSL TLS Protocol

OSI Layer Placement and TCP/IP Protocol Support

SSL is a platform-independent and application-independent protocol that is used to secure TCP-based applications. It sits on top of the TCP layer, below the application layer figure 1.11 , and acts like sockets connected by TCP connections.

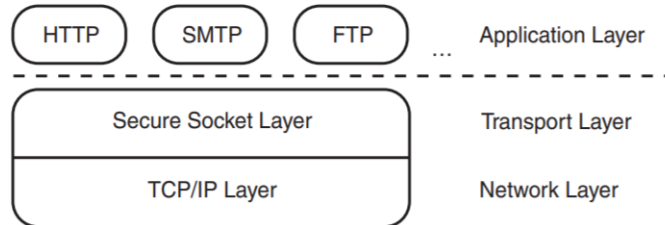


Fig. 1.11: SSL and TCP/IP.

For the most common applications defined in the TCP/IP suite, such as HTTP and Simple Mail Transfer Protocol (SMTP), measures have been set for all the technical details to be used for SSL to secure the communications. The two well-known examples are:

1. HTTP over SSL: Securing the web was the main driveway for designing SSL, and HTTP is the first application-layer protocol secured by SSL.
2. Email over SSL: E-mail protocols such as SMTP, POP3, IMAP can be supported by SSL.

SSL Record Protocol and Handshake Protocols

An SSL connection is established in two main phases:

1. The handshake phase negotiates cryptographic algorithms, authenticates the server, and establishes keys for data encryption and MAC.
2. The secure data transfer phase is under the protection of an established SSL connection.

SSL is a layered protocol. At the lowest layer is the SSL record protocol. The record protocol consists of several message types or protocols carrying out different tasks. Figure 1.12.

- Record protocol: Is essentially an encapsulation protocol. It transmits various higher-level protocols and application data. The record protocol takes

messages to be transmitted from upper-client protocols; performs the necessary tasks such as fragmentation, compression, applying MAC, and encryption; and then transmits the final data. It also performs the reverse actions—decryption, verification, decompression, and reassembly—to the receiving data. The record protocol consists of four upper-layer client protocols: Handshake Protocol, Alerts Protocol, Change Cipher Spec Protocol, and Application Data Protocol.

- Handshake protocols: Are responsible for setting and resuming SSL sessions. Three subprotocols exist:
 - Handshake protocol negotiates the security attributes of an SSL session.
 - Alerts protocol is a housekeeping protocol that is utilized to send alert messages between the SSL peers. The alert messages contain errors, exception conditions such as a bad MAC or decryption failure, or notification such as a closure of the session.
 - Change cipher spec protocol is applied to signal transitions in cipher strategies in the subsequent records.
- Application data protocol controls the transmission of upper-layer application data.

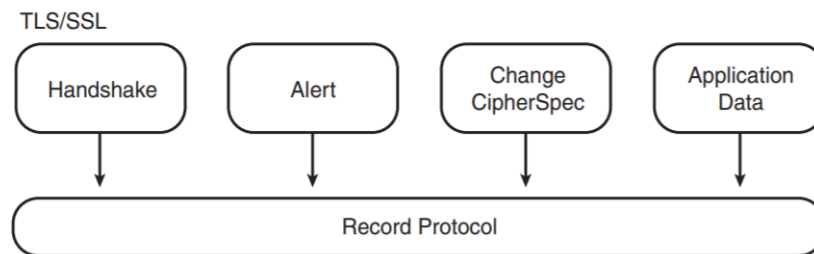


Fig. 1.12: SSL/TLS Protocol Structure.

1.4.3.2 SSL Connection

Handshake protocols are utilized for the SSL client and server to establish the connection:

1. Negotiate security capabilities:
 - protocol version: The protocol version field defines the highest SSL version that the client supports;

- client random: Used to calculate the master secret and to prevent replay attacks;
 - client cipherSuite: The cipher suite specifies a collection of cryptographic algorithms that are used during the SSL connection;
 - compression method: Determines the compression methods supported by the client.
2. Authentication: The client authenticates the server. Optionally, the server can also authenticate the clients.
 3. Key exchange: The two parties exchange keys or information that is needed to generate the master keys.
 4. Key derivation: The two parties derive the master secret that is next used to produce keys used for data encryption and MAC.

1.4.4 Security Threats

SSL VPNs support users coming from any computer on the Internet, that is not controlled by the corporate IT department, this poses a major threat to security.

1.4.4.1 Data Theft

Various types of security threats lead to data theft or password theft:

1. Browser's cache, web browsers reserve the various web objects that users downloaded during browsing to improve the browsing experience. The browser cache files are physically stored on the user's computer in predefined directories. Attackers can easily use the computer and collect the browser cache to retrieve sensitive information.
2. The browser histories expose the user activities and internal web server structure. Similar to the browser cache, browser histories saved on unmanaged computers that are vulnerable to data theft.
3. Documents and other types of sensitive data left on the unmanaged computers are vulnerable to data theft.
4. Key loggers or Trojan horse programs.

1.4.4.2 Man-in-the-Middle Attacks

In this attack, the attacker uses the Address Resolution Protocol (ARP) spoofing attack or Domain Name System (DNS) spoofing attack to the SSL VPN user. The SSL traffic then will be redirected to the attack host that is configured with SSL proxy software. The attacking host then acts as the destination web server by establishing an SSL connection with the user on one side and another SSL connection with the true destination web server on the other side.

1.4.4.3 Split Tunneling Attack

In a remote access VPN deployment, split tunneling gives the user direct access to a public network and VPN access to a private network concurrently. Attackers have possibilities to compromise the computer from the Internet and access to the internal network through the VPN tunnel.

1.4.5 Cisco SSL VPN Product

1.4.5.1 Cisco ASA

Cisco¹ offers the SSL VPN functionality in some of its product offerings, including the Cisco ASA 5500 series. Cisco ASA integrates all the firewalls, IDS, and VPN capabilities of its current products. This provides an all-in-one solution for your network. Incorporating all these solutions into Cisco ASA secures the network without the need for extra overlay equipment or network alterations. Cisco ASA was developed to respond to the many Cisco customers and network professionals who requested the type of integration that ASA supplies in a security product. The ASA offers two SSL VPN modes:

1. clientless mode that does not need a VPN client to be installed on the user's computer;
2. full tunnel mode that allows full network access.

1.4.5.2 Cisco Adaptive Security Device Manager (ASDM)

Cisco Adaptive Security Device Manager also provides an easy-to-navigate and simple graphical interface to set up and manage the different features that the security appliances provide. It is bundled with a variety of administration and

¹Cisco Systems, an American technology company known for its computer networking products.

monitoring tools to check the health of the appliance and the traffic traversing through it.

1.4.5.3 Cisco AnyConnect VPN client

Cisco AnyConnect is a software application that can be pushed to or installed on a user machine. The AnyConnect client is used to provide full network access to corporate resources after the SSL VPN tunnel has been negotiated.

2.1 Presentation of the Host Organization

2.1.1 Description

The Bank of Algeria have been established on December 12, 1962, was endowed with all the statutes of an issuing institution, in order to create the conditions favorable to an orderly development of the national economy. The Central Bank exercises the functions of issuing fiat money, directing and supervising credit, as well as managing foreign exchange reserves.

The mission of the Algeria Bank is to maintain in the area of currency , credit, and exchange, the most favorable conditions for the orderly development of the economy.

The main objectives are:

- High quality of consulting services.
- Safeguarding long term liquidity.
- Security of information.

2.1.2 Organization of the Central Bank of Algeria

The Bank of Algeria is organized at the central level in:

1. Seven (7) General Directorates dealing with the departments of studies, inspection, and banking activities:
 - General Directorate of Studies.

- General Directorate of General Inspection.
- Directorate General of Credit and Banking Regulations.
- General Directorate of Foreign Exchange Control.
- General Management of the General Fund.
- General Directorate of External Financial Relations.
- General Management of the Network.
- Two (2) of the General Directorates managing specific aspects related to the issuance of banknotes and banking training, these are:
 - The General Management of the Hôtel des Monnaies.
 - The General Directorate of the Higher School of Banking, which is responsible for staff training and retraining function for the entire banking sector.

2. Two (2) General Directorates responsible for administrative management and resources of the Bank, these are:

- The General Directorate of Human Resources.
- The General Directorate of Resources Administration.

It also has a network of 48 agencies and branches, ensuring an effective presence in each of the country's wilayas: the agencies and branches are coordinated by three regional offices located in the cities of Algiers, Oran, and Annaba. A staff of nearly 3,000 agents contributes, at all levels, to the achievement of Bank objectives.

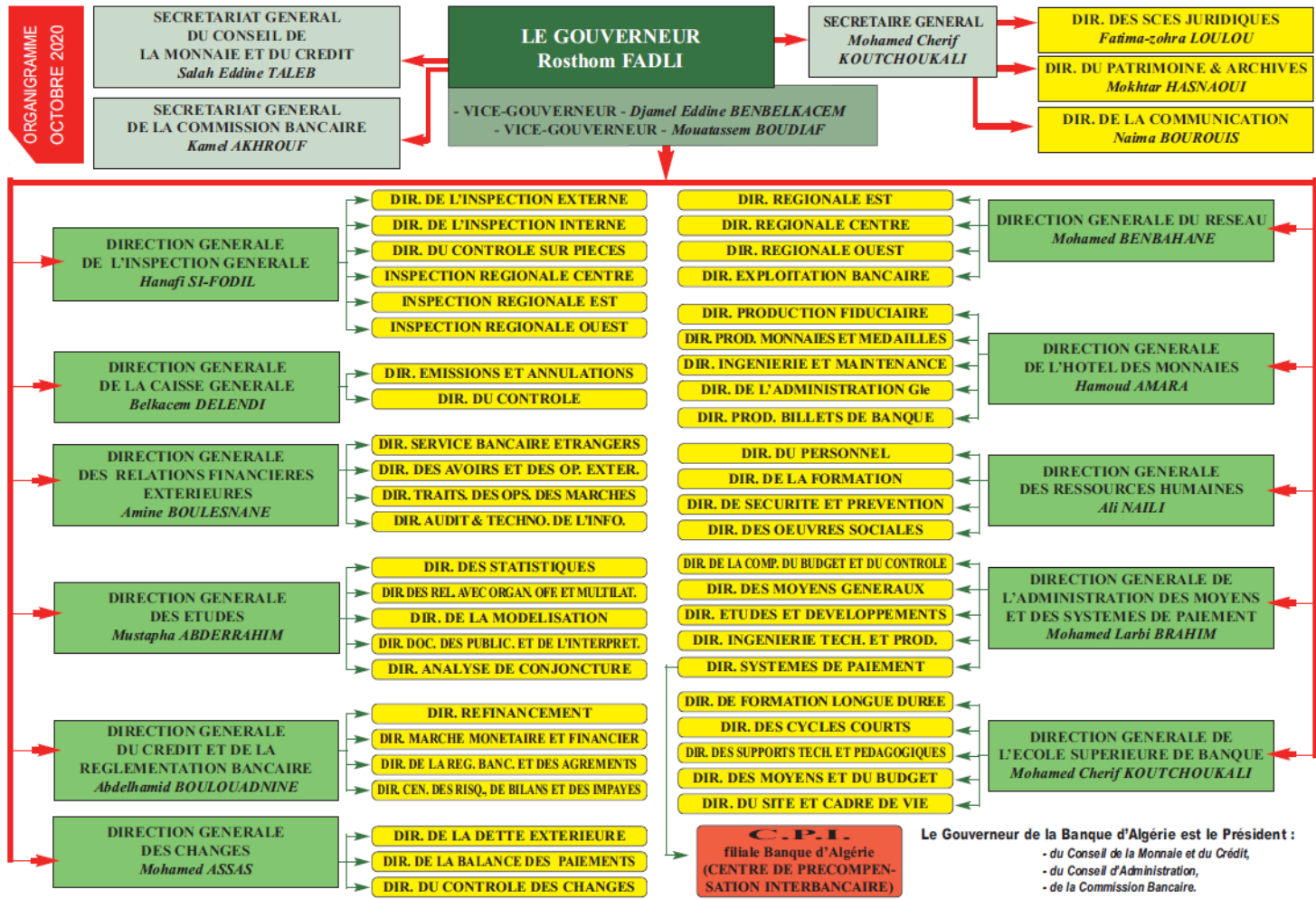


Fig. 2.1: Source : <http://www.bank-of-algeria.dz>.

2.2 Specifications Notebook

The bank agrees on the functional specification paper that we suggest as we will present it as we go.

2.2.1 Project Framework

2.2.1.1 Presentation

Strong authentication architecture is the binding of two security mechanisms to provide 2FA for the remote users connected through a secure tunnel.

2.2.1.2 Context

This internship is part of an end-of-studies project to obtain a Master's degree in information systems security. Our training was performed within the Algeria national Bank under the title of Strong authentication architecture. The purpose of our project is to enable universal access to internal resources through a strong remote access authentication architecture.

This architecture is destined essentially for the companies that want to remote access their job, therefore the user accesses the internal resources remotely over a secure tunnel then being authenticated through two-factor authentication.

2.2.1.3 Objectives

The objectives of this project are:

- Improve the remote access authentication.
- Provide full remote access to the internal resources through a secure tunnel.
- Create a web interface that authenticates users with 2FA.
- Provide an administration tool to manage and control the users' account.

2.2.1.4 Target

The organization that wants to improve remote access work security.

2.2.1.5 Responsibility

The administrator is responsible for managing the users' accounts (add, delete, recover the non-access accounts, search and display the existed ones).

The administrator will manage the secure tunnel.

2.2.2 Functional Specification

2.2.2.1 Functional requirements

The functional requirements are presented as follows:

- Secure full remote access to the organization's resources.
- Manage the users' accounts with an administrator tool.
- Users inscription.

- Authenticate users through two factors authentication.
- Recover the users' accounts.

1. ***Secure full remote access***

Our architecture gives the users full access to the organization's resources- in case the users need to use different resources- remotely through a secure tunnel.

2. ***Manage the users' accounts***

In this architecture we create an administrative tool to help the admin managing the user's account, the admin is able to add, delete, display the user's list, search for user and, recover a user's account.

3. ***Users inscription***

The administrator is responsible for adding a new user by providing the necessary information through the administrative tool. The user will be given a secrets key that, will be used in the two factors authentication.

4. ***Two factors authentication***

The users' authentication passes through two factors:

- Factor one: Knowledge-based.
- Factor two: Possession-based.

5. ***Recover the users' accounts***

The administrator is responsible for recover the user account -in case a user lost access to his account- by proffering a new secret key.

2.2.2.2 Non Functional requirements

Non-functional requirements are important because they indirectly affect the proposed solution performance, which means that they should not be neglected, for this the following requirements must meet:

1. ***reliability*** The application should operate consistently with satisfaction and, without errors.
2. ***Errors*** Error messages are organized to properly facilitate the use of the different parts of this architecture.

3. ***Ergonomics and interface*** The architecture is adapted to the user without making any effort (clear and easy to use) in terms of administration or users' authentication process.
4. ***Security*** Our proposed architecture objective is to improve authentication while keeping the integrity and confidentiality of data.
5. ***Maintenance and reuse*** The proposed solution should conform to a standard and clear architecture allowing its maintenance and reuse.
6. ***Compatibility and portability*** The solution approach is easy to configure and adaptable with most internal networks.

2.2.2.3 Graphic Design

Icon Design

We design the Icon shown in Figure 2.2 to use it in the following applications.



Fig. 2.2: Application Icon.

Model

1. ***Administration tool***

The login interface will be as follows:

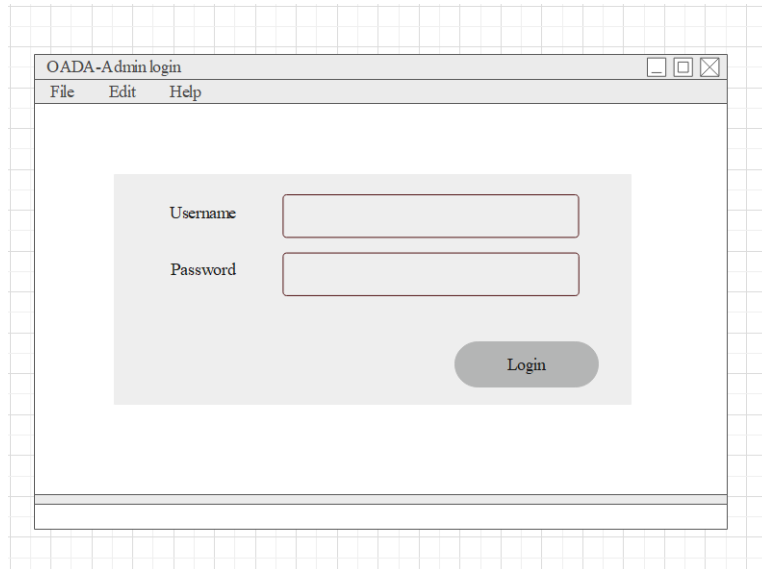


Fig. 2.3: Model-Admin Login.

The manage users interface will be as follows:

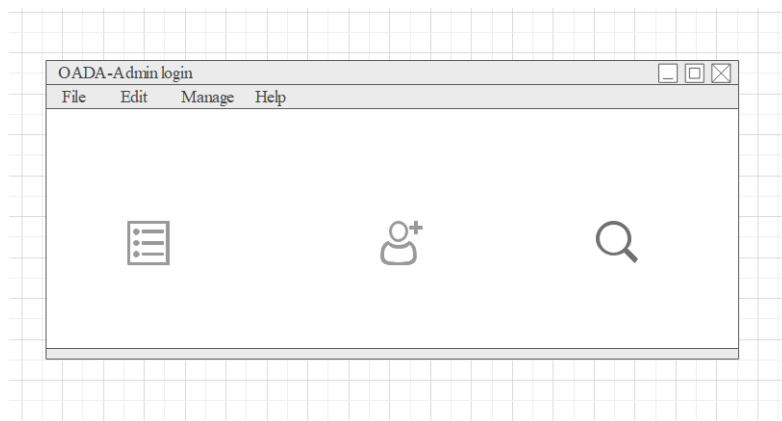


Fig. 2.4: Model-Admin Manage Users .

2. *Web interface*

The web interface to authenticate users with the username/ password will be as follows:

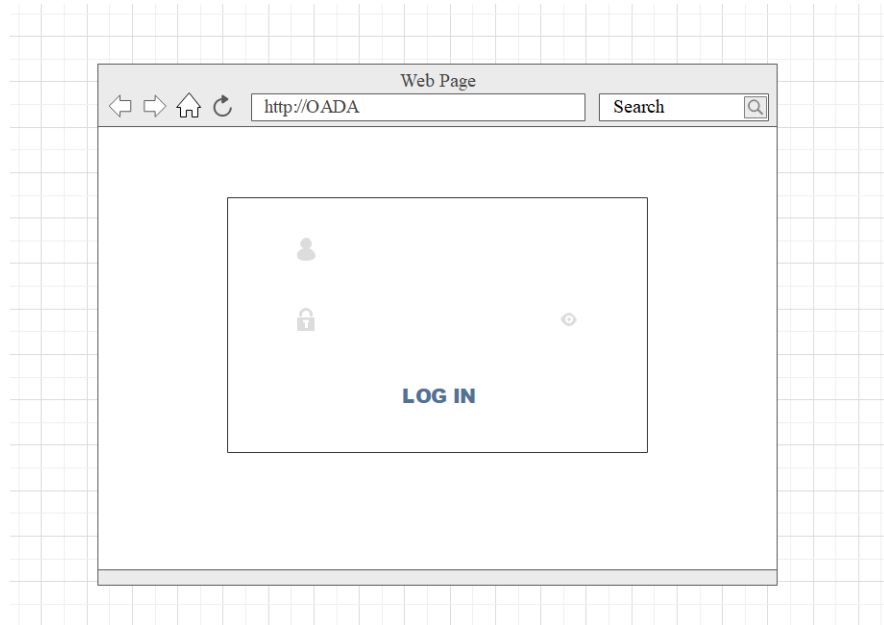


Fig. 2.5: Model-user Login.

The web interface to authenticate users with the one time password will be as follows:

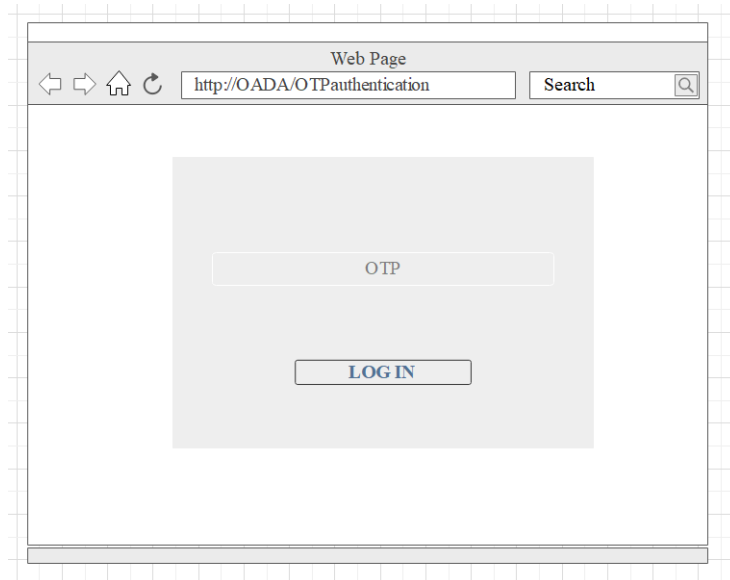


Fig. 2.6: Model-user OTP Authentication.

3. *OTP Generator*

The interface of the one time password generator will be as follows:

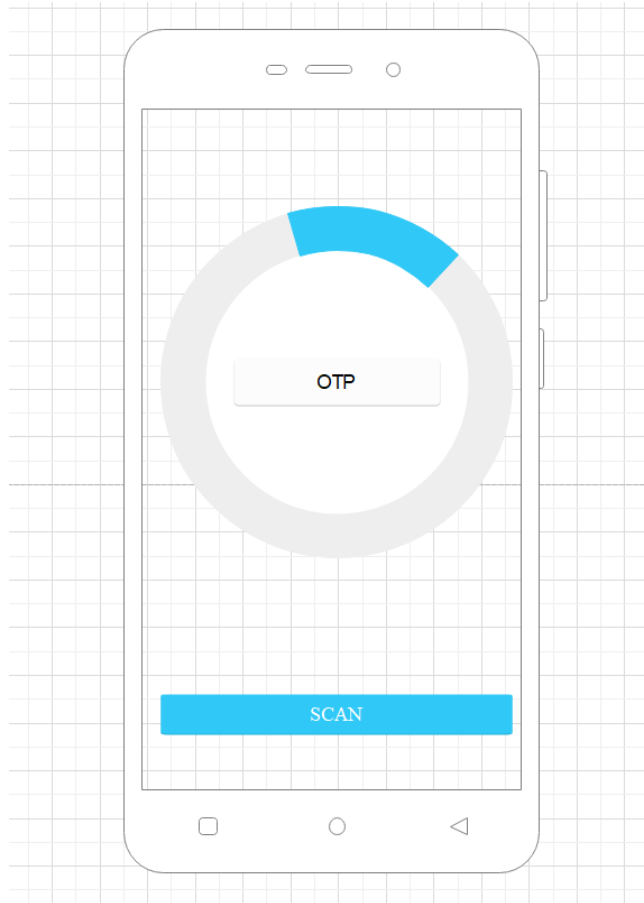


Fig. 2.7: Model-OTP Generator.

2.2.3 Technical Specification

2.2.3.1 Technologies Choice

- We chose to set up a VPN SSL tunnel.
- For the administration tool, we develop a desktop application.
- For the users authentication through the web interface, we chose those factors:
 - Knowledge based by the usual username-password combination.
 - Possession based by using a One-time password. We chose to develop a mobile application to generate the second password. For the OTP algorithm, we will be based on a timed one-time password (TOTP).

2.2.3.2 Data base

- The admin and users' data will be stored in an internal data server.

- The light database will be used for the mobile application to generate the one time password.

2.2.3.3 Equipment

- Cisco ASA firewall.
- Data server.
- Web server.
- Cisco routers and switches.
- An Internal and external computers.
- User smart phone.

2.2.3.4 Security

- SSL VPN uses SSL protocol and its successor the Transport Layer Security (TLS) in order to provide a secure connection between the remote users and the internal network resources.
- TOTP algorithm is based on a hash function To generate the one time password.
- The OTP code generated by the TOTP algorithm is valid for a limited period (30 seconds) therefore is not vulnerable to the phishing attack.
- We use AES encryption to encrypt the shared secret keys within the internal database and the light database in the mobile database.
- We use the SHA256 hash function to hash the sensitive data of the users.

2.3 Planning

To better understanding the proposed solution we use the FAST (function analysis system technique) diagram, showed in figure 2.8 ,to explain the required function besides the technique solutions.

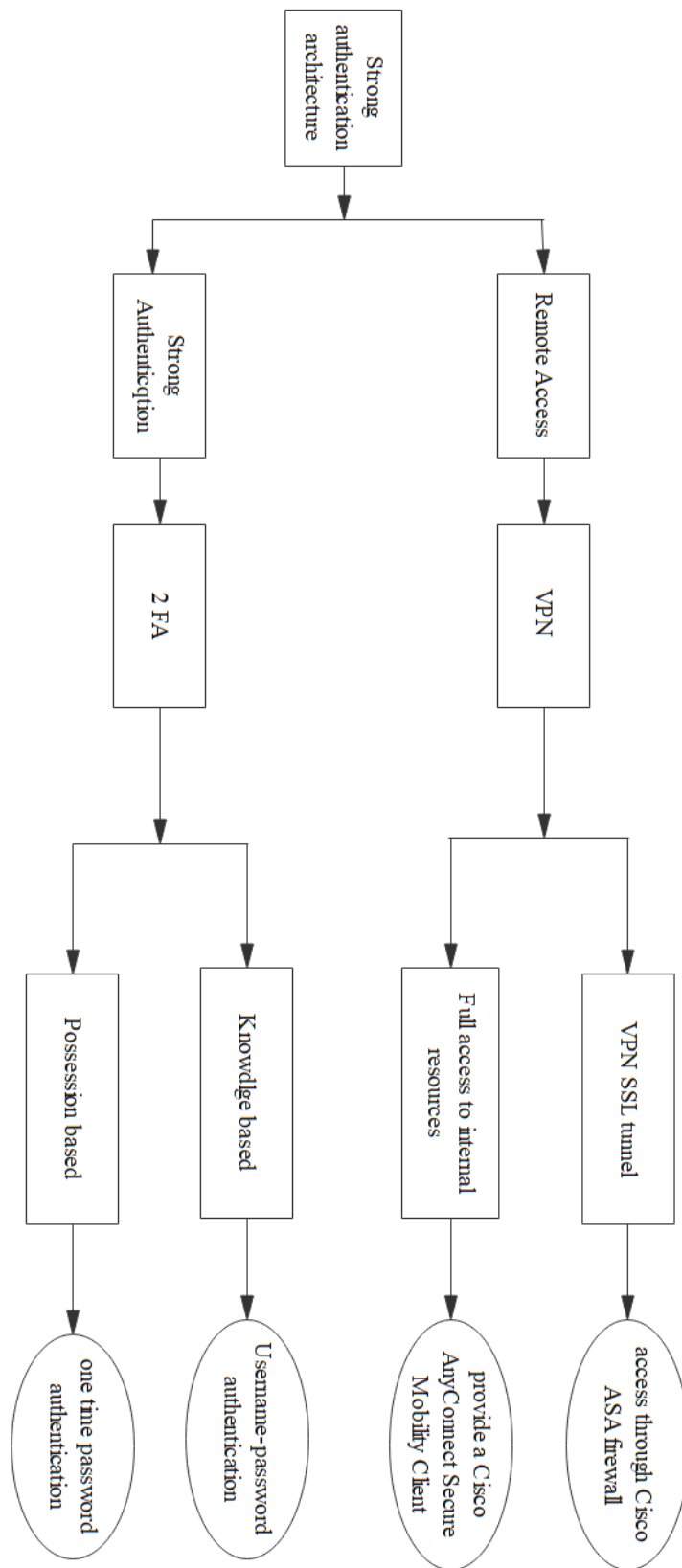


Fig. 2.8: FAST Diagram.

the design of our information system will be done by using the UML language. UML is the acronym for "Unified Modeling Language". UML is a visual language composed of a set of diagrams, which give a different vision of the project.

3.1 Analyze

3.1.1 Identification of Actors

1. **Administrator:** This is the role responsible for managing the users' accounts. Adding, deleting, searching, display the existed users list, and recover the inaccessible accounts.
2. **User:** A person who is already registered on our site, and has been authenticated through two-factor authentication.

3.1.2 Package Diagram

- Package « Open Authentication Data Access »

In order to give the administrator the ability to manage the users account over the network and control them, we design the Open Authentication Data access application.

- Package « Users Authentication Process »

In order to authenticate the users through the 2F process we need:

- Package « Users' Website »

We design a basic web interface to authenticate the user through 2F Authentication as:

- * knowledge-based by using the username and password combination;
 - * possession-based by using the second password generated by the possessed application.
- Package \ll AuthOADA \gg

This is a One-time-password-generator application that generates and displays a one-time password based on time and secret key values every 30 seconds.

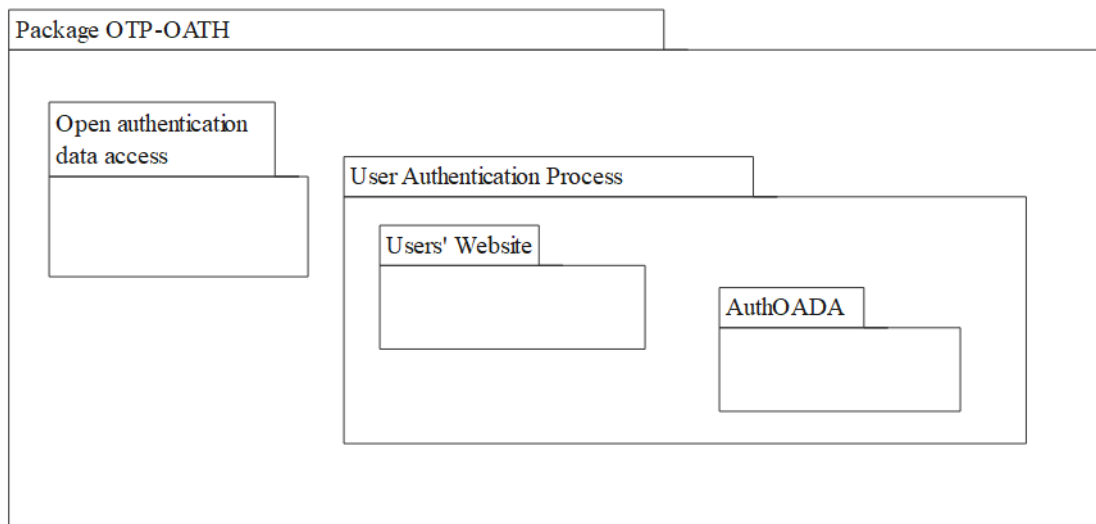


Fig. 3.1: Package Diagram.

3.1.3 Use Case Diagram-Open Authentication Data Access

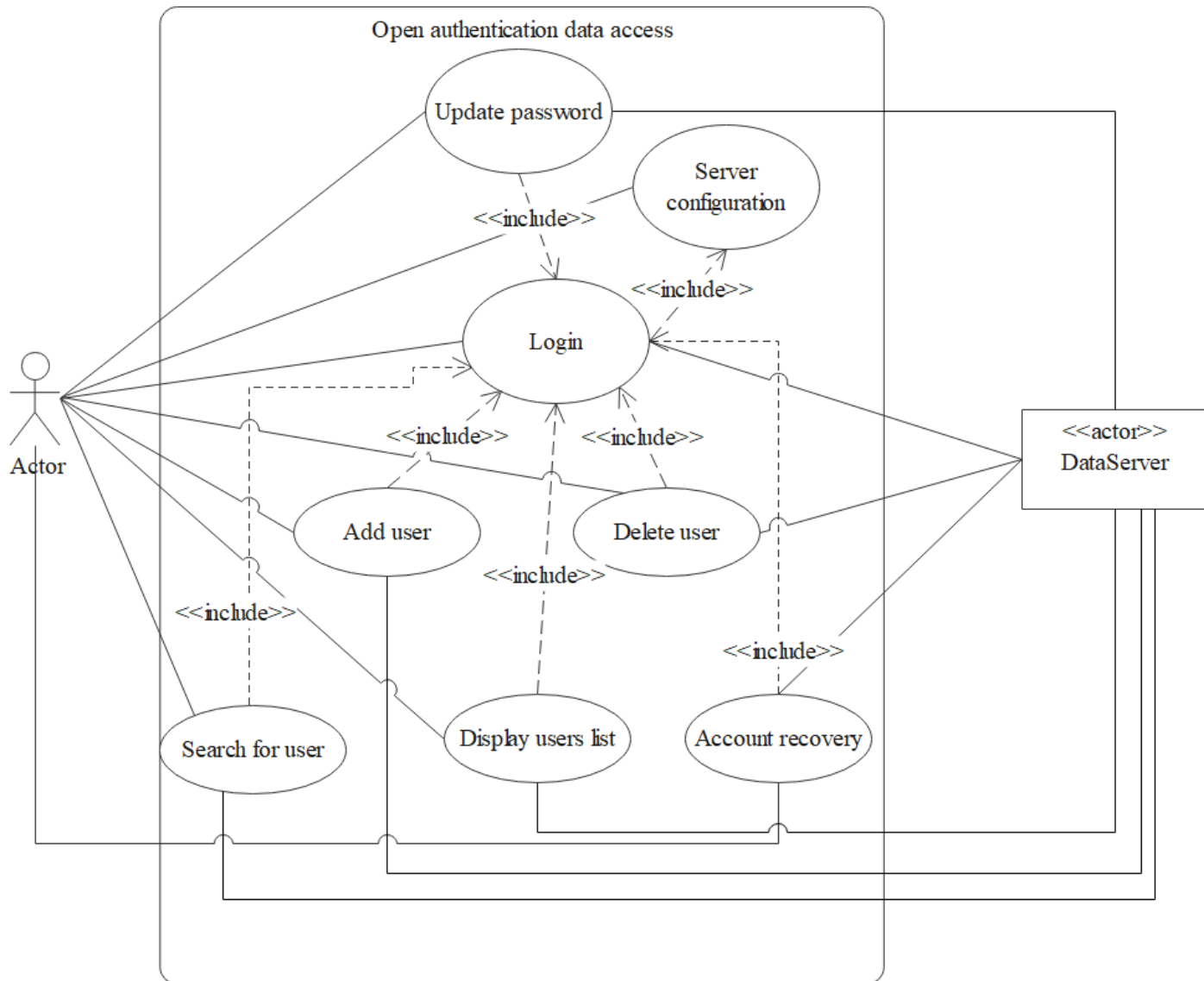


Fig. 3.2: Use case Diagram Open Authentication Data Access.

3.1.3.1 Description of Use Cases

Use case	Login
Actor	Administrator
Objective	Authentication.
Abstract	The administrator will be authenticated before using the features of the application by providing the username and password combination.
Precondition	The application should be configured to connect to the remote database. The administrator should exist in the database.
Scenario	<ol style="list-style-type: none"> 1. The system displays the login interface. 2. In case that no configuration was set to connect to the remote database, the admin needs to set the necessary details. 3. The admin types the username and password in order to log in. 4. The system will verify the previous combination in case of a successful login, the main interface will be displayed.

Tab. 3.1: Login Case

Use case	Add User
Actor	Administrator
Objective	Create new user account.
Abstract	The user registration process start with an user' order to access the internal resource, the admin supervisor the creation of the new account for the user. The user types the necessaire information, then get QR code that hold the secret key used to generate the OTP, the user must scan the QR code using the OTP generator app on his smartphone.The secret key is shared between the user and the the authentication serevr.
Precondition	The user needs to have installed the one-time password generator on his smart phone.
Scenario	<ol style="list-style-type: none"> 1. The system displays the Add user interface. 2. The admin fills the fields with the user information. 3. The system generates the QR code that holds the shared secret key. 4. The user scans the QR code using the phone application. 5. The admin adds the user to the database.

Tab. 3.2: Add User Case

Use case	Delete User
Actor	Administrator
Objective	Delete an existing user account.
Abstract	The admin is able to delete a user account, if the user is willing to, by providing the user's username and password combination.
Precondition	Accord of the user.
Scenario	<ol style="list-style-type: none"> 1. The system displays the Delete user interface. 2. The admin fills the fields with the user information(username and password of the user). 3. The system verifies the input information then displays a confirmation dialogue in case of confirmation the user will be removed from the database.

Tab. 3.3: Delete User Case

Use case	Search for User
Actor	Administrator
Objective	Find user' accounts.
Abstract	In order to check if the user exists or not , the admin can find the user' account based on the user's username.
Scenario	<ol style="list-style-type: none"> 1. The system displays the search for user interface. 2. The admin fills the field with the user' username. 3. The system displays "exist" if the user was found in the database.

Tab. 3.4: Search for User Case

Use case	Account Recovery
Actor	Administrator
Objective	Recover a user account.
Abstract	In case that the user lost possession of the OTP generator, the administrator will supervise the recovery process by providing the user' necessary information besides assigning the user to a new shared secret key.
Precondition	The user needs to re-install the OTP generator on the smartphone.
Scenario	<ol style="list-style-type: none"> 1. The system displays the account recovery interface. 2. The admin fills the fields with the user' information (username and recover code) . 3. The system verifies the inputs then generates a QR code that holds the new shared secret key 4. The user uses the app on his smartphone to scan the QR code 5. The system modifies the secret key field related to the user account in the database.

Tab. 3.5: Account Recovery Case

Use case	Display users list
Actor	Administrator
Objective	Display the existed users' accounts.
Abstract	The administrator is able to display all the usernames of the users registered on the database.
Scenario	<ol style="list-style-type: none"> 1. The admin open the users list interface. 2. The admin refresh the users' list. 3. The system displays the final users' list.

Tab. 3.6: Display Users List Case

Use case	Update password
Actor	Administrator
Objective	Change the admin password.
Abstract	The admin can update the account password.
Scenario	<ol style="list-style-type: none"> 1. The system displays the update password interface. 2. The administrator provide the old password beside the username and the new password. 3. The system will verify the previous combination(username,old password). 4. The system update the old password.

Tab. 3.7: Update Password Case

Use case	Server configuration
Actor	Administrator
Objective	Configure the connection to the remote database.
Abstract	In order to connect to the remote data server is necessary to provide a configuration file.
Scenario	<ol style="list-style-type: none"> 1. The system displays the server configuration interface. 2. The administrator provides the necessaire information 3. The system saves the connection information.

Tab. 3.8: Server Configuration Case

3.1.4 Use Case Users' Websit

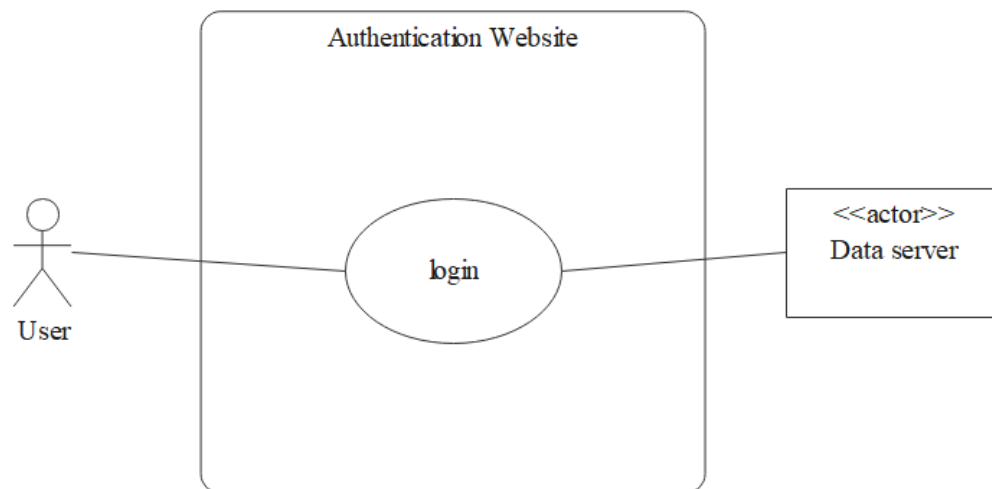


Fig. 3.3: Use case Diagram Users Websit.

3.1.4.1 Description of Use Cases

Use case	Login
Actor	User
Objective	Authentication.
Abstract	The user will be authenticated through 2F authentication.
Precondition	Possession of the OTP generator that is associated with this account.
Scenario	<ol style="list-style-type: none"> 1. The user connects to the web server and provides the user-name/password combination. 2. The web server verifies the user credentials. In case the user is authenticated, the TOTP authentication process will start. 3. The user types the TOTP code shown in the TOTP generator, if it's valid code the user will gain access to the account.

Tab. 3.9: Users Login Case

3.1.5 Use Case AuthOADA

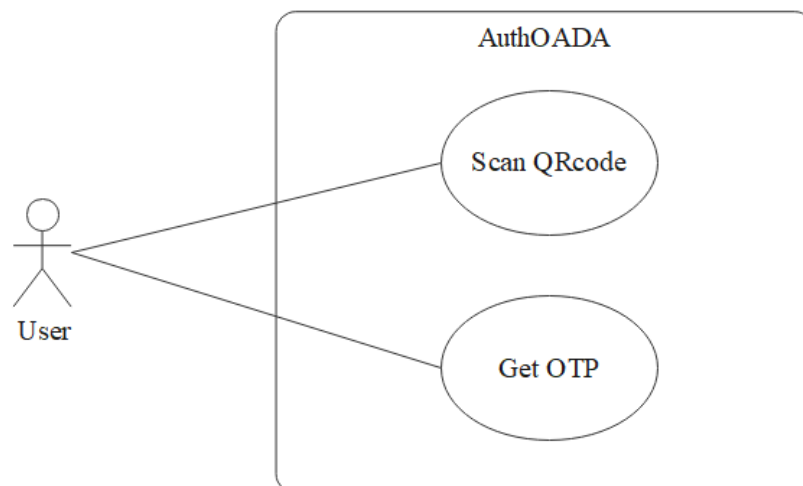


Fig. 3.4: Use case Diagram AuthOADA.

3.1.5.1 Description of Use Cases

Use case	Scan QR code
Actor	User
Objective	Add OTP generator account.
Abstract	The user scan the QRcode that holds the shared secret key.
Scenario	<ol style="list-style-type: none"> 1. The user opens the application and press the scan button. 2. The scan possess will launch, the user scan the QR code. 3. The application store the secret key and starts displaying a OTP every 30 second

Tab. 3.10: Scan Case

Use case	Get OTP
Actor	User
Objective	Generate OTP.
Abstract	The user will get the current OTP.
Scenario	<ol style="list-style-type: none"> 1. The user opens the application. 2. The application display the OTP based on time and secret key.

Tab. 3.11: Get OTP Case

3.2 Conception

3.2.1 Open Authentication Data Access Functions

3.2.1.1 Admin Login Sequence Diagram

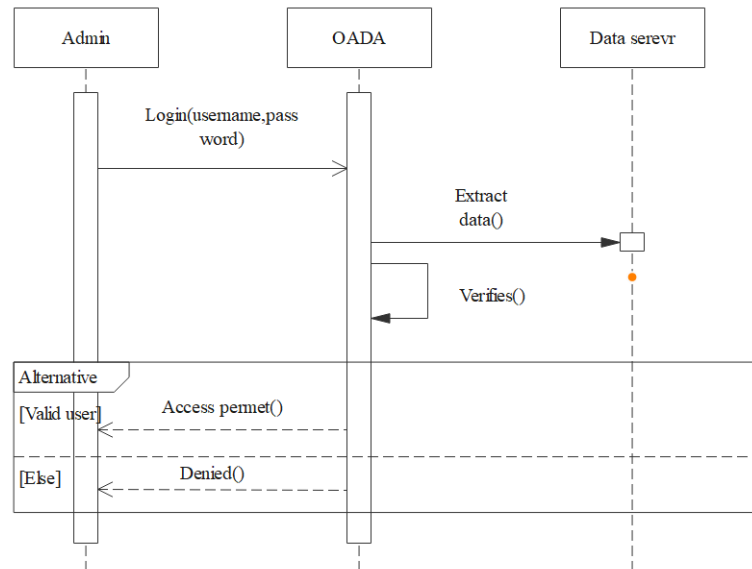


Fig. 3.5: Sequence Diagram-Admin Login.

3.2.1.2 Add User Sequence Diagram

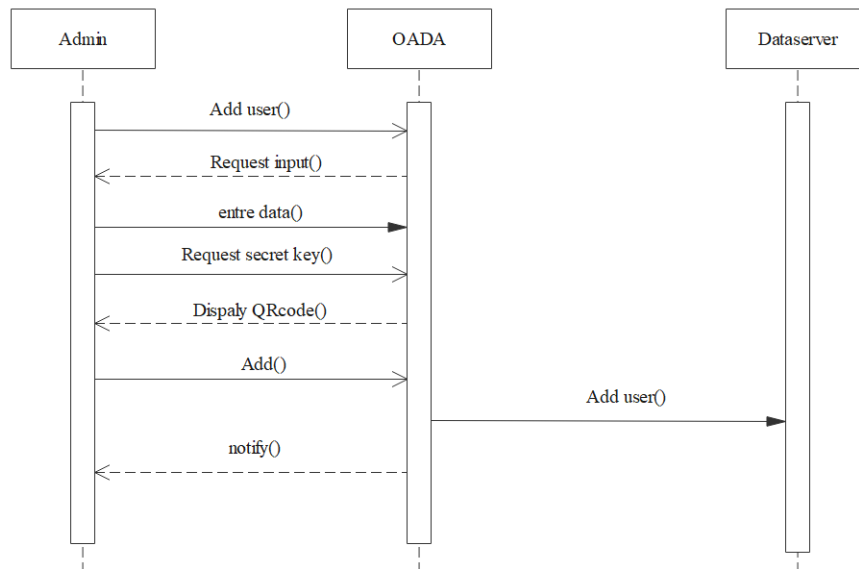


Fig. 3.6: Sequence Diagram-Add User.

3.2.1.3 Delete User Sequence Diagram

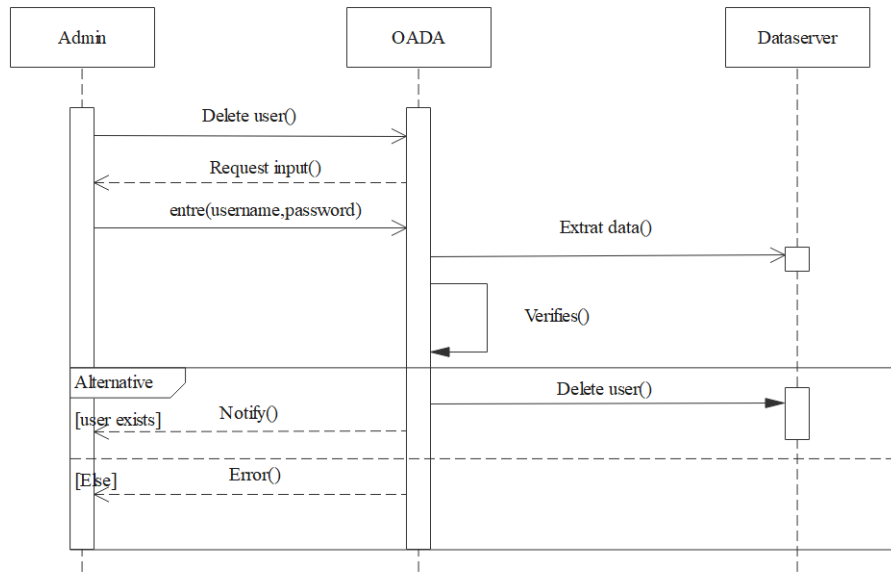


Fig. 3.7: Sequence Diagram-Delete User.

3.2.1.4 Search for User Sequence Diagram

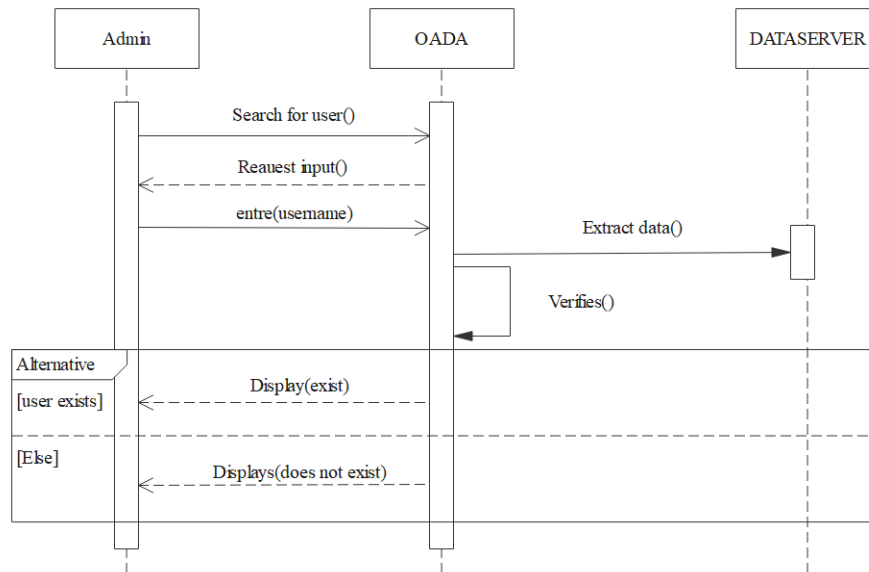


Fig. 3.8: Sequence Diagram-Search for User.

3.2.1.5 Display Users List Sequence Diagram

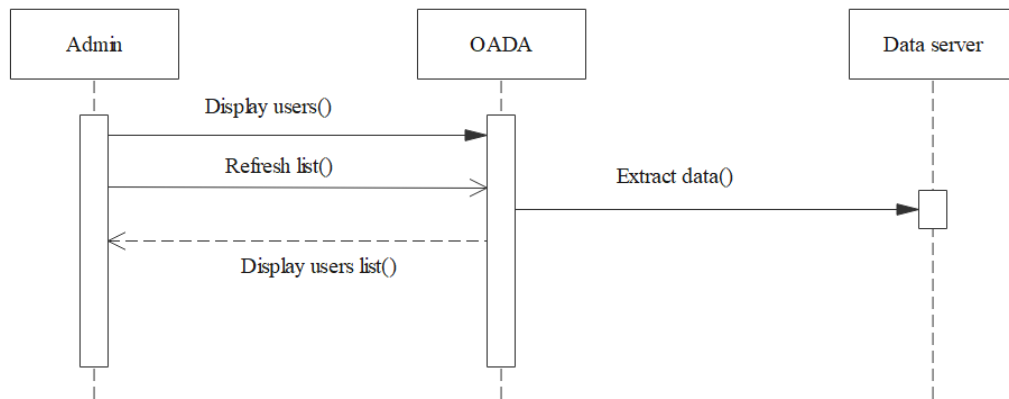


Fig. 3.9: Sequence Diagram-Display Users List.

3.2.1.6 Account Recovery Sequence Diagram

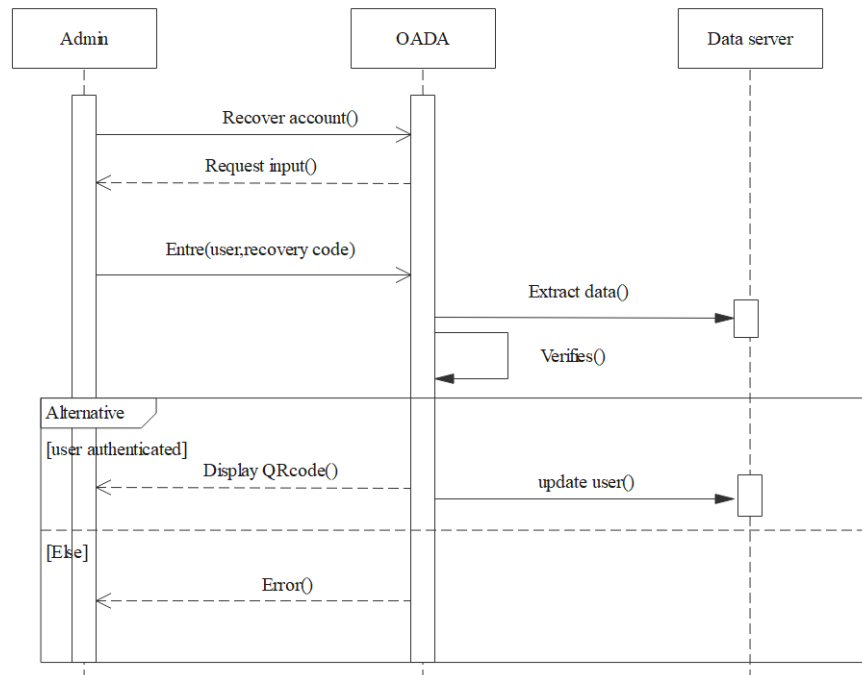


Fig. 3.10: Sequence Diagram-Account Recovery.

3.2.1.7 Update Password Sequence Diagram

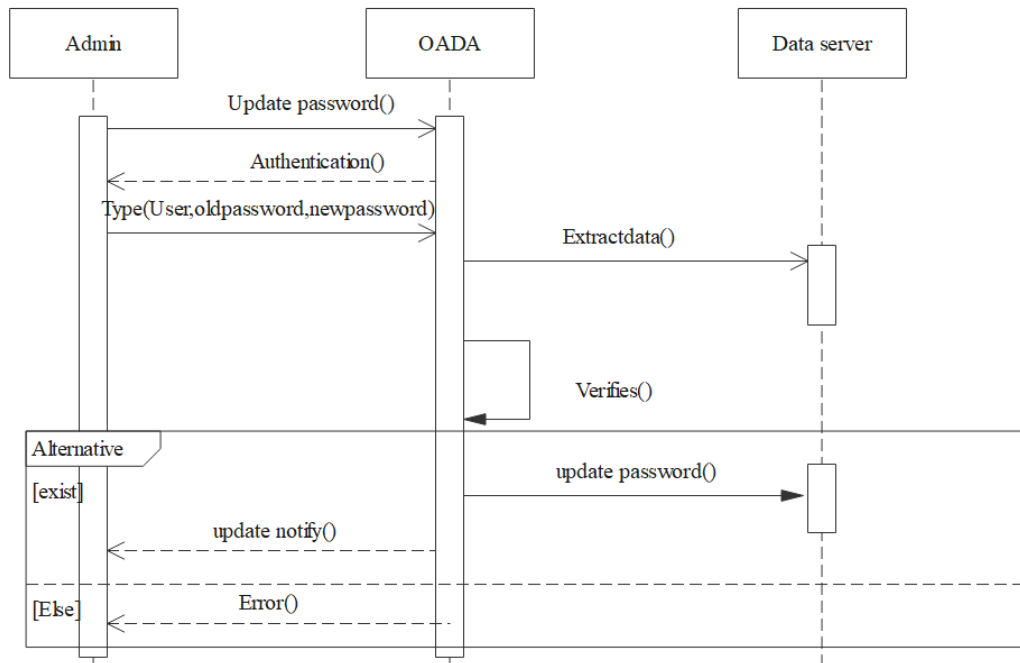


Fig. 3.11: Sequence Diagram- Update Password.

3.2.1.8 Config Server Connection Sequence Diagram

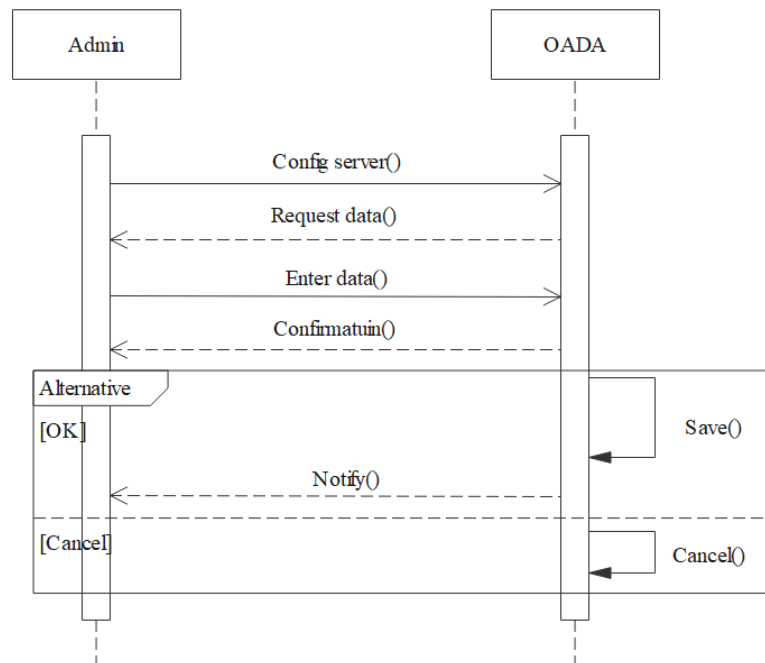


Fig. 3.12: Sequence Diagram-Server Configuration.

3.2.1.9 OADA State Diagram

In order to express the main functionality in each interface, we will use the UML state diagram.

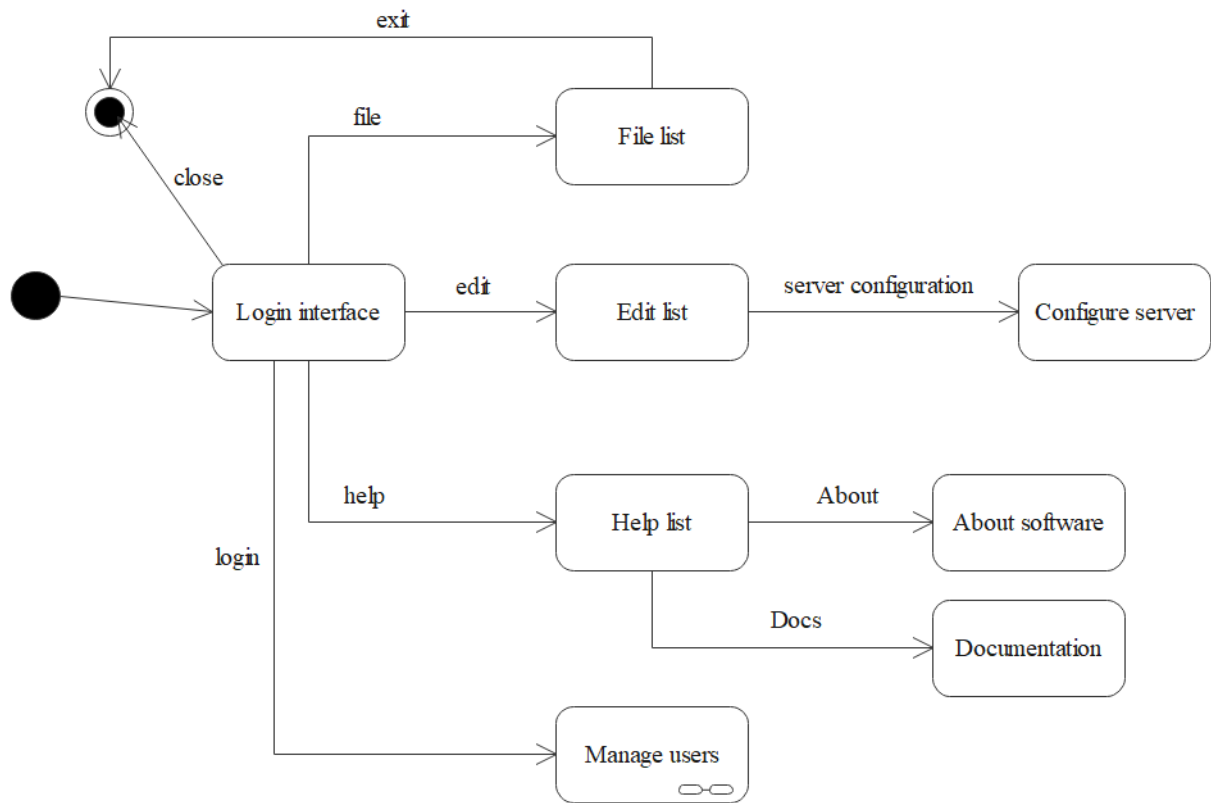


Fig. 3.13: OADA State Diagram- Main Interface.

3.2.1.10 OADA State Diagram-Manage Users

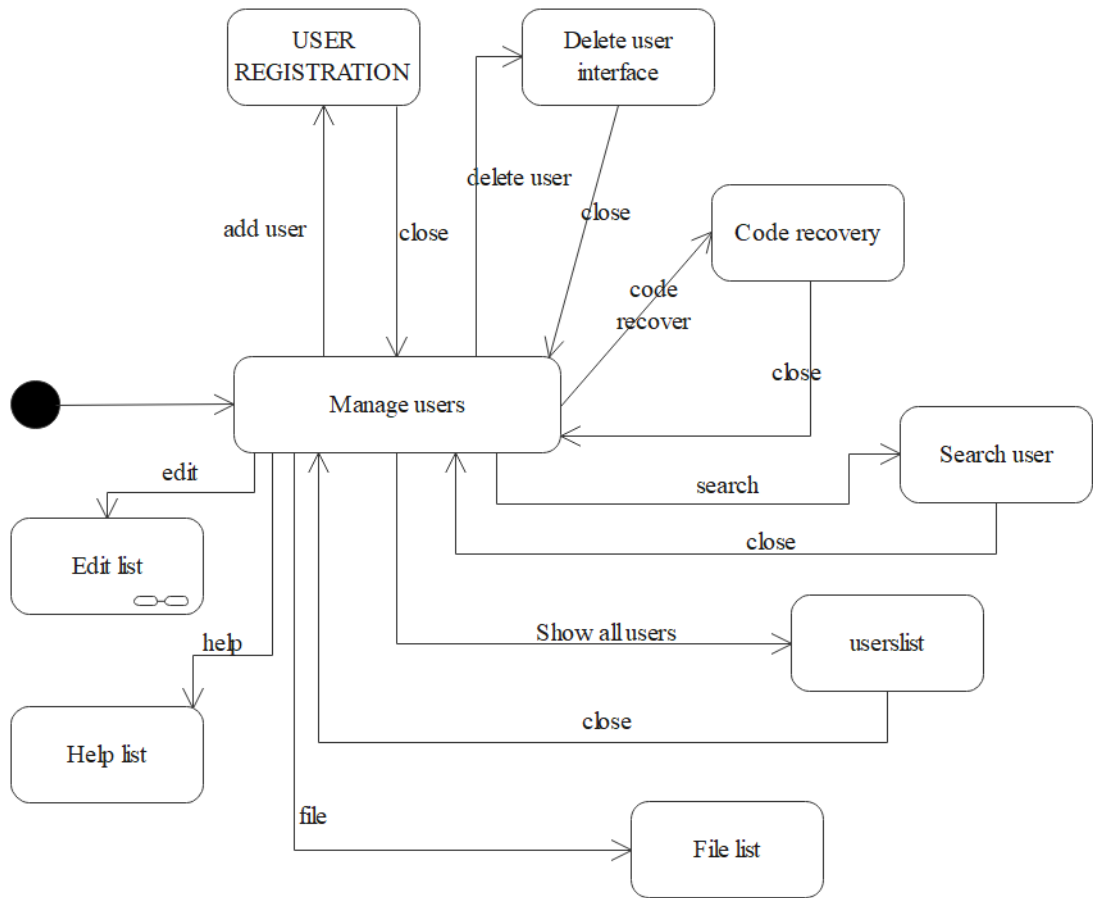


Fig. 3.14: OADA Manage Users State Diagram.

3.2.1.11 OADA State Diagram-Manage Users Edit List

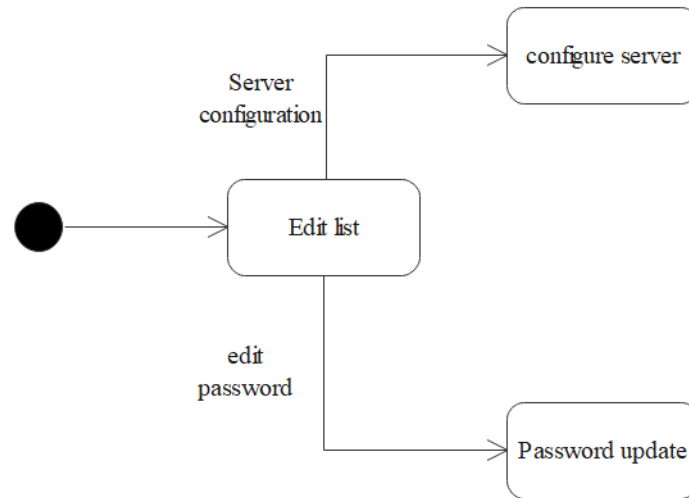


Fig. 3.15: OADA Manage Users State Diagram- Edit List.

3.2.2 Users Authentication

3.2.2.1 Web Interface Login Sequence Diagram

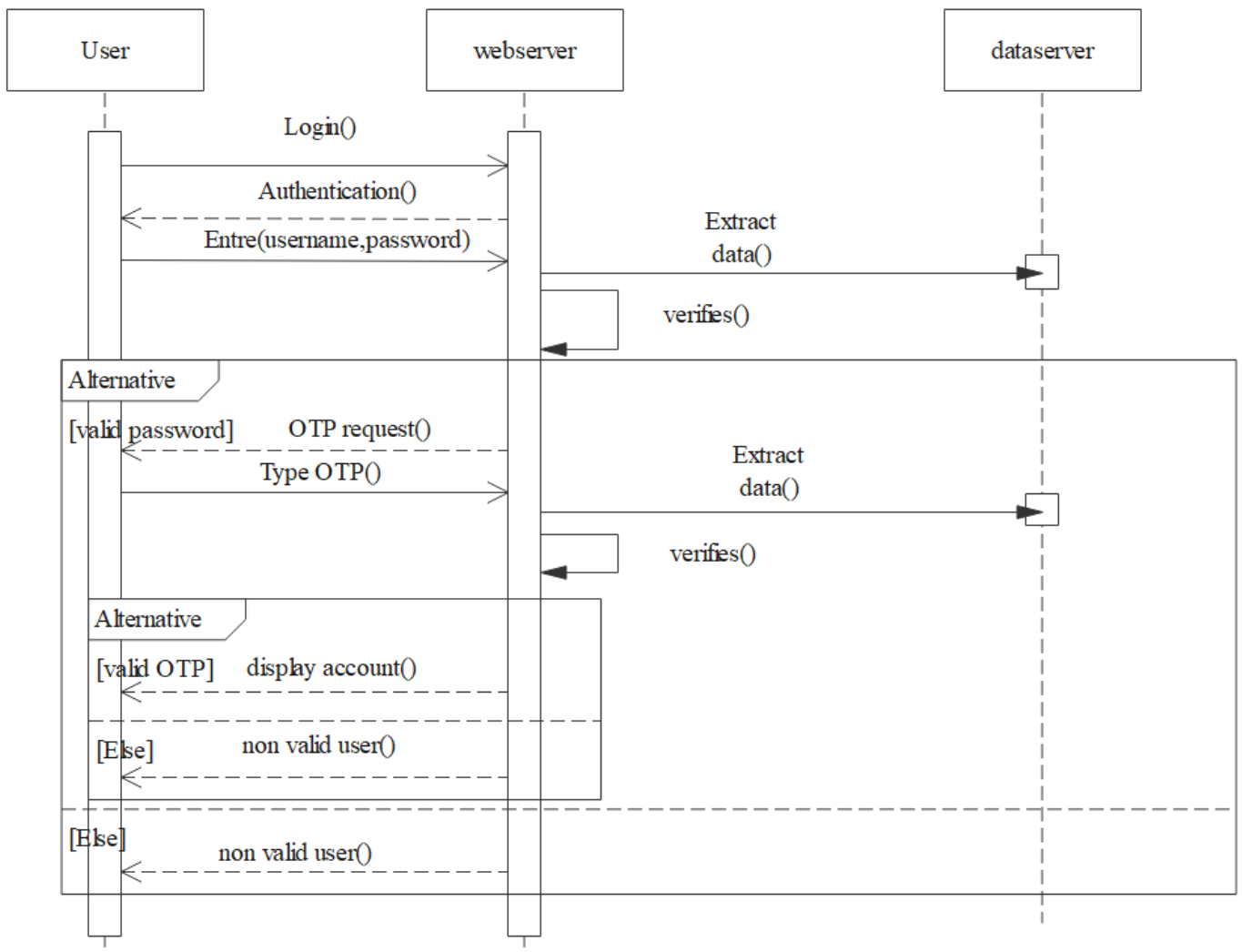


Fig. 3.16: Sequence Diagram- User Login.

3.2.2.2 OTP Generator

3.2.2.3 Sequence Diagram- Scan New Code

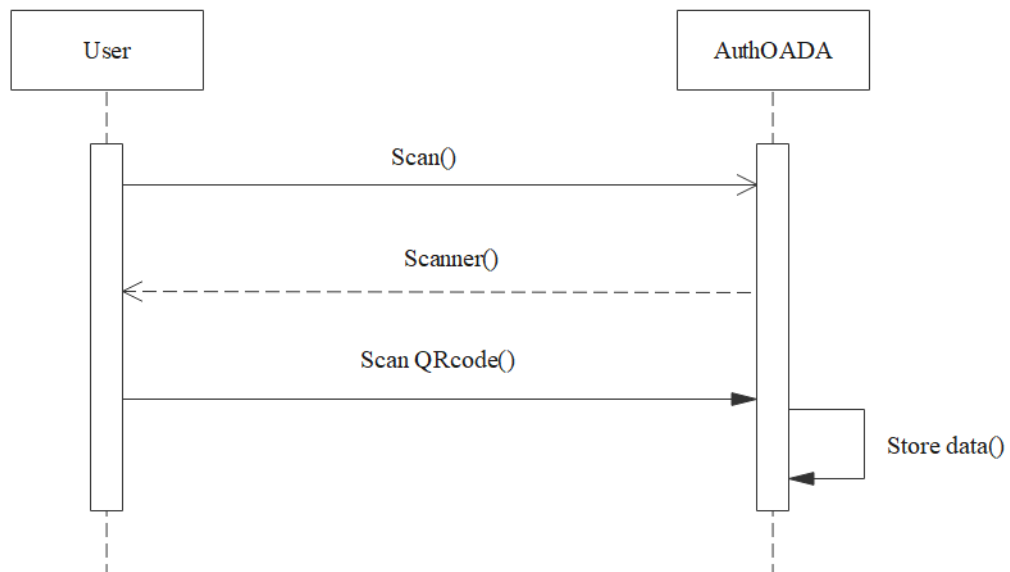


Fig. 3.17: Sequence Diagram-Scan New Code.

3.2.2.4 Sequence Diagram-Get OTP

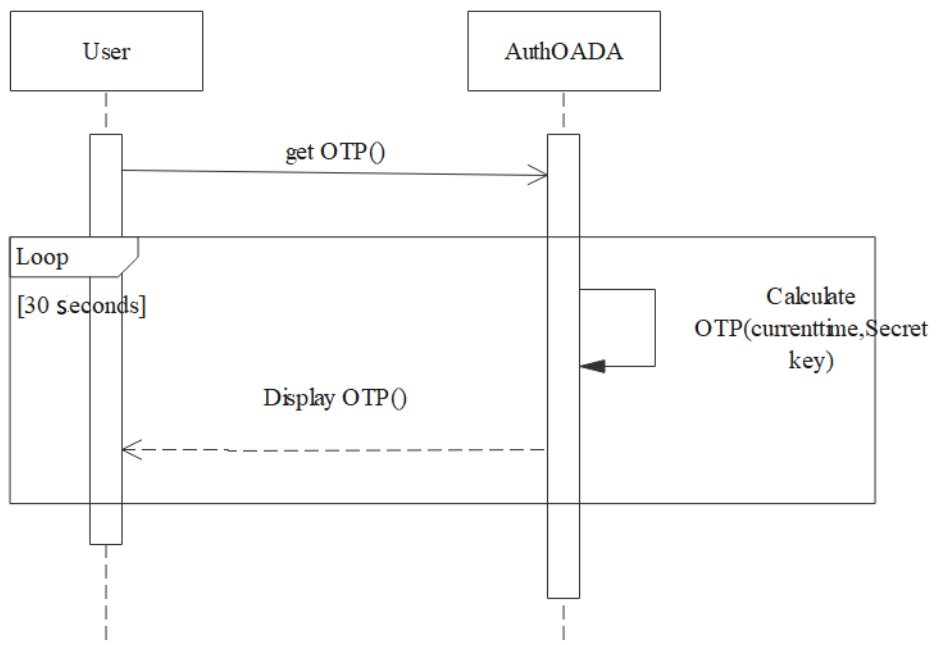


Fig. 3.18: Sequence Diagram-Get OTP.

3.3 Database Conception

3.3.1 Internal Database

In this part, we will realize the class diagram which represents the schema of the main tables of the database. In our internal database, we have two tables as shown follows:

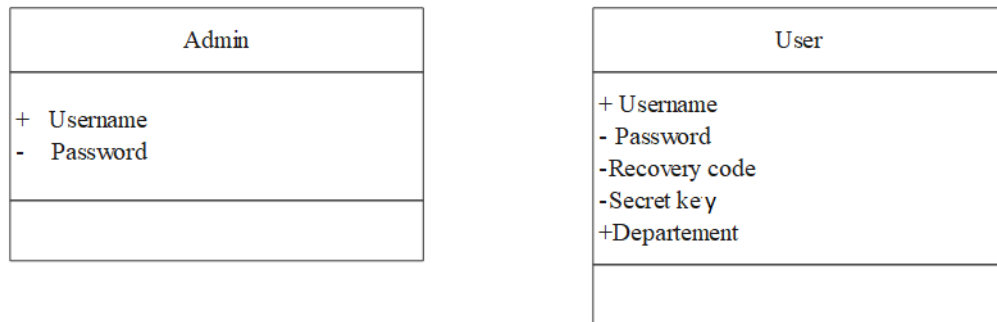


Fig. 3.19: Class Diagram-Internal Database.

3.3.2 Mobile Database

In this part, we will realize the class diagram which represents the schema mobile light database that contains only one table holding the shared secret key used to calculate the OTP.

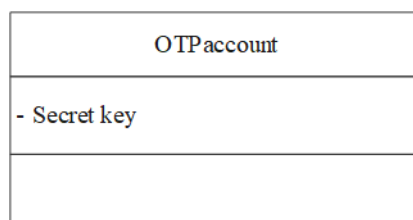


Fig. 3.20: Class Diagram-Mobile Database.

STRONG AUTHENTICATION ARCHITECTURE - OTP MECHANISM IMPLEMENTATION

4.1 Software Used

NetBeans

NetBeans IDE has a range of new tools for HTML5/JavaScript, in particular for Node.js, KnockoutJS, and AngularJS; enhancements that further improve its support for Maven and Java EE with PrimeFaces; and improvements to PHP and C/C++ support. More information in the *Netbeans official site*.

In this project, we use the NetBeans IDE to develop our java applications, besides the implementation of the basic authentication web interface. We are using the zxing and Qrgen library to manage the QR code.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, Android Studio offers features that enhance productivity when building Android apps. We use Android Studio to develop our android application that will generate the TOTP code. More details in the *Android Studio web page*.

Inno Setup

Inno Setup is free software for creating installers for Windows. We use it to create an installer for our java application.

4.2 OADA Administrator Tool Realization

Login interface

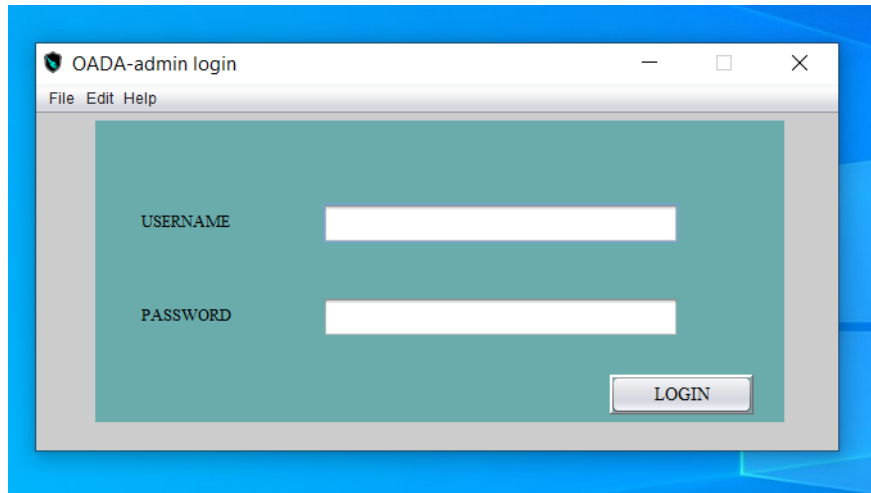


Fig. 4.1: Login Interface.

Mange Users

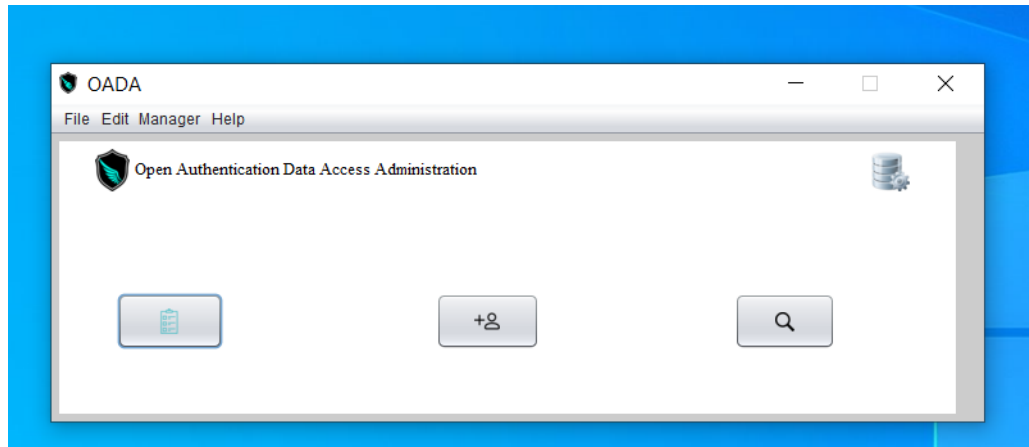
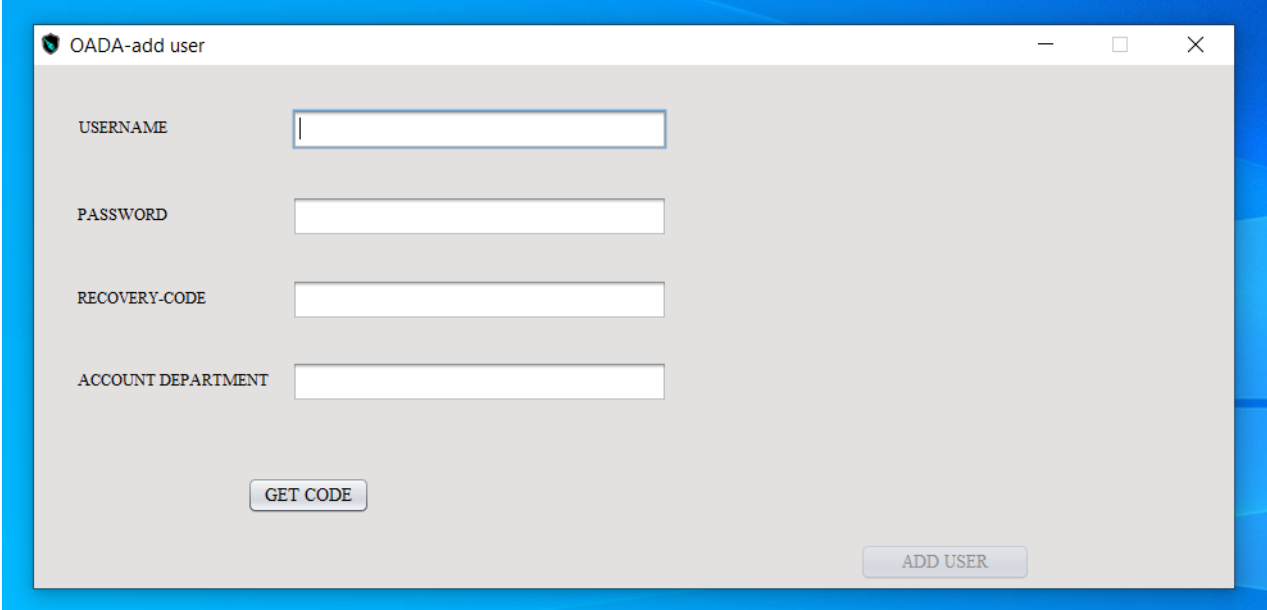


Fig. 4.2: Manage Users Interface.

Add Users

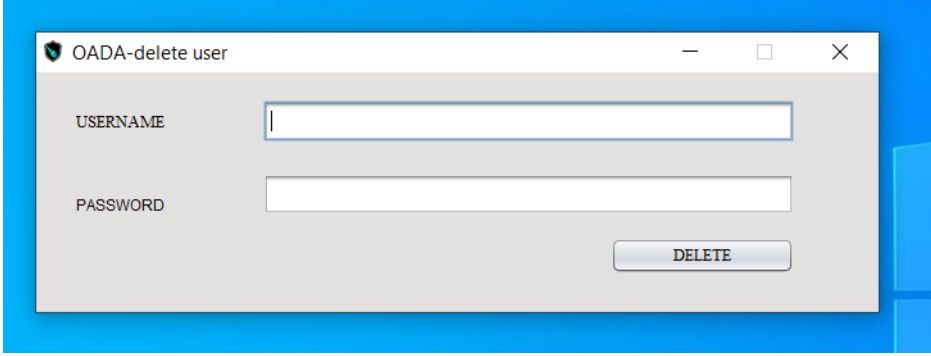


The screenshot shows a web browser window titled "OADA-add user". The interface is a form with the following elements:

- USERNAME: A text input field.
- PASSWORD: A text input field.
- RECOVERY-CODE: A text input field.
- ACCOUNT DEPARTMENT: A text input field.
- GET CODE: A button located below the RECOVERY-CODE field.
- ADD USER: A button located at the bottom right of the form.

Fig. 4.3: Add new User Interface.

Delete Users



The screenshot shows a web browser window titled "OADA-delete user". The interface is a form with the following elements:

- USERNAME: A text input field.
- PASSWORD: A text input field.
- DELETE: A button located at the bottom right of the form.

Fig. 4.4: Delete User Interface.

Search Users

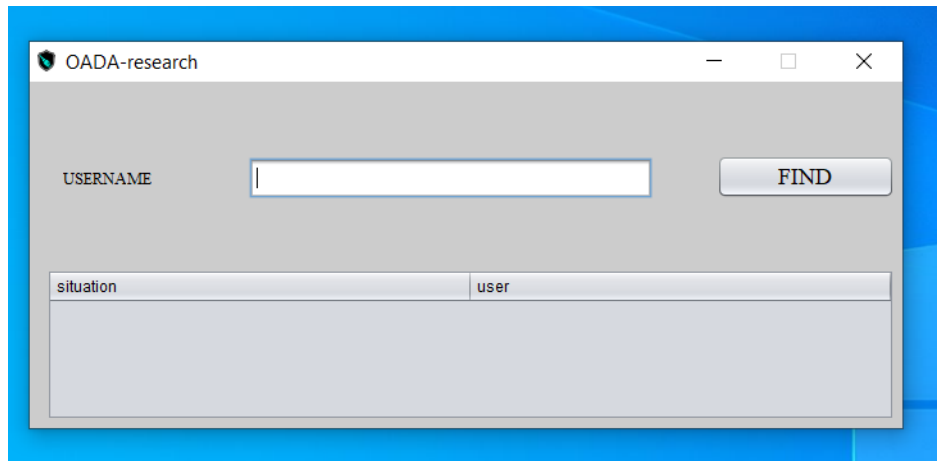


Fig. 4.5: Search for User Interface.

Recover User account

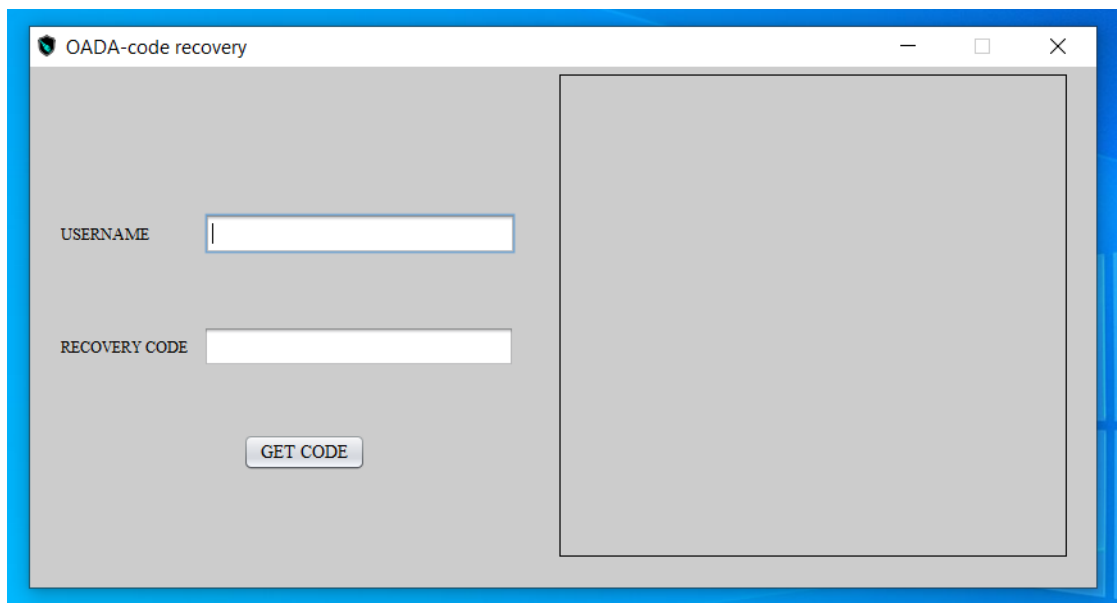


Fig. 4.6: Recover User account Interface.

Display Users list

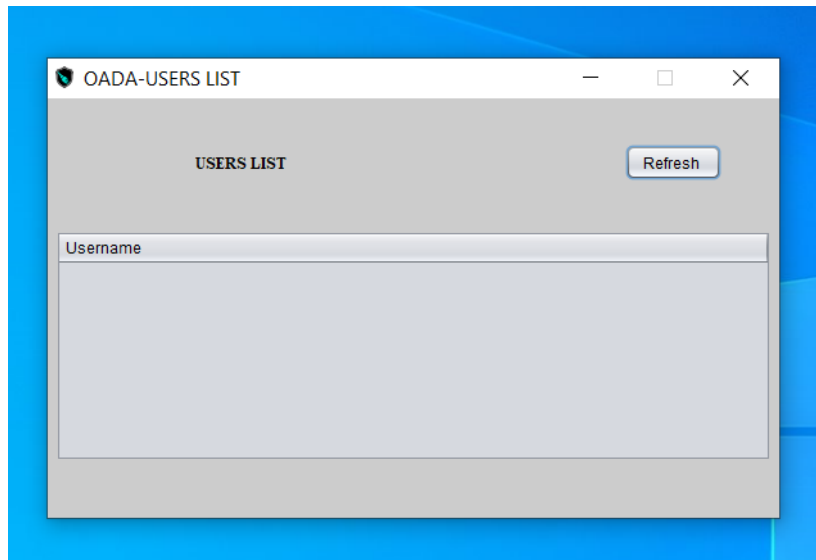


Fig. 4.7: Display users list Interface.

Server Configuration

The configuration to create a connection to the data server over the network:

- HOST: The hostname of the server, usually the IP address.
- DATABASE: The database name(oathdatabase).
- USER: The database user.
- PASSWORD: The password registered for the database user.

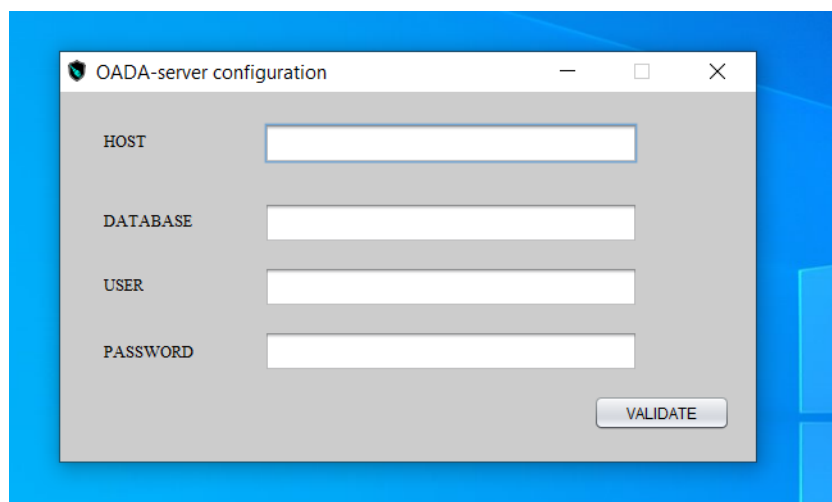


Fig. 4.8: Server Configuration Interface.

Documentation

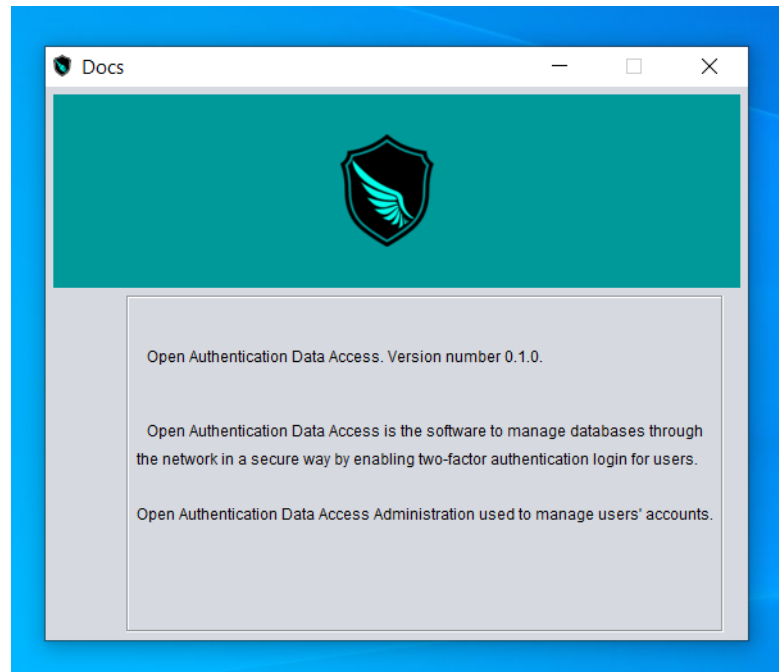


Fig. 4.9: Document Interface.

4.2.1 Data Server configuration

4.2.2 OADA Installer

We use **Inno setup** software to create the application installer, the Figure 4.10 represents the Inno setup installer file code source. After compiling the file it generates the installer shown in Figure 4.11.

```

set - Inno Setup Compiler 6.0.5 (u)
File Edit View Build Run Tools Help
#define MyAppName "OADA"
#define MyAppVersion "1.0.0"
#define MyAppPublisher "BFE, Inc."
#define MyAppExeName "OathAdmin.exe"

[Setup]
; NOTE: The value of AppId uniquely identifies this application. Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={DCA02382-FF14-4C81-A3C7-4C39F365403E}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
DefaultDirName={autopf}\{#MyAppName}
DisableProgramGroupPage=yes
LicenseFile=C:\Users\pcadmin\Documents\NetBeansProjects\OathAdmin\license.txt
InfoAfterFile=C:\Users\pcadmin\Documents\NetBeansProjects\OathAdmin\information.txt
; Uncomment the following line to run in non administrative install mode (install for current user only.)
;PrivilegesRequired=lowest
PrivilegesRequiredOverridesAllowed=dialog
OutputDir=C:\Users\pcadmin\Documents\inout
OutputBaseFilename=OADA
SetupIconFile=C:\Users\pcadmin\Documents\setupicon.ico
Compression=lzma
SolidCompression=yes
WizardStyle=modern

[Languages]
Name: "english"; MessagesFile: "compiler:Default.isl"
Name: "french"; MessagesFile: "compiler:Languages\French.isl"

```

Fig. 4.10: Inno-setup File .

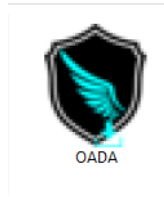


Fig. 4.11: OADA Launcher.

4.3 User Authentication

4.3.1 Web Site Realization

We develop our web interface using HTML, CSS and the java Servlet.

Authentication Interface

Contains:

- Textfield 'USERNAME'.
- Textfield 'PASSWORD'.
- Button 'LOGIN'.

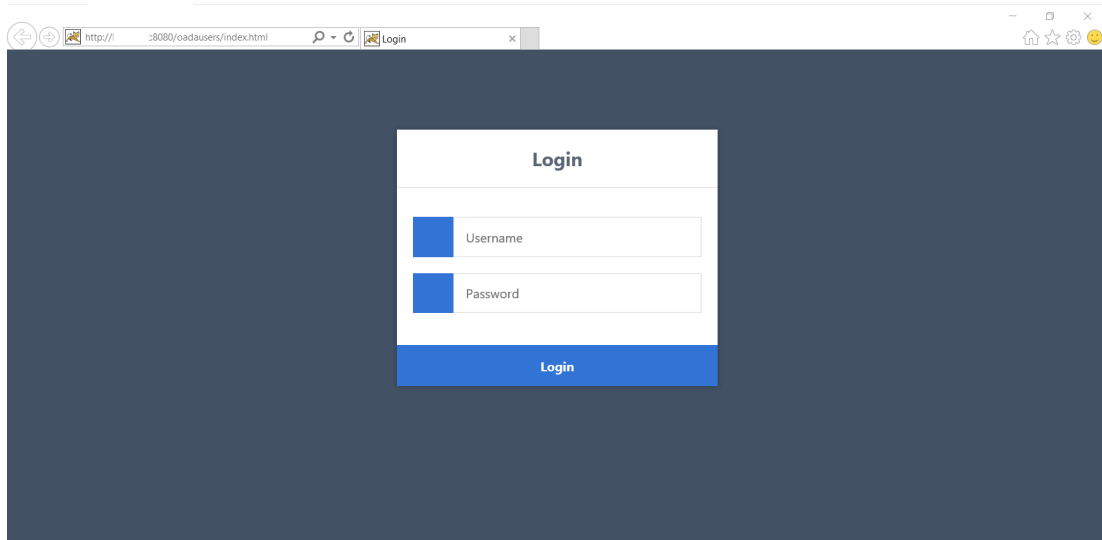


Fig. 4.12: Login Interface.

OTP Authentication Interface

We implement this class to authenticate the user based on the one time password.

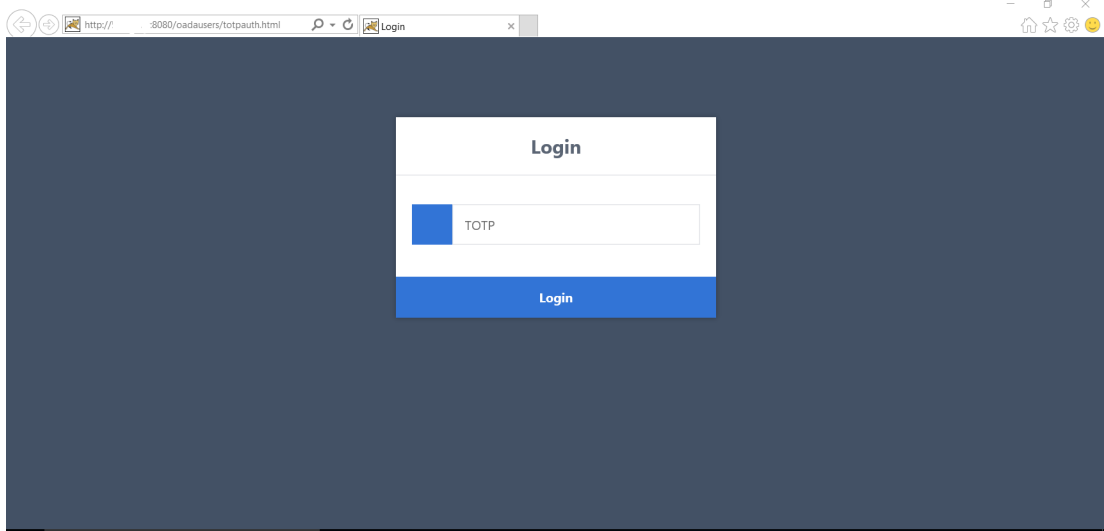


Fig. 4.13: OTP Authentication Interface.

4.3.2 OTP Generator Application Realization

Main interface

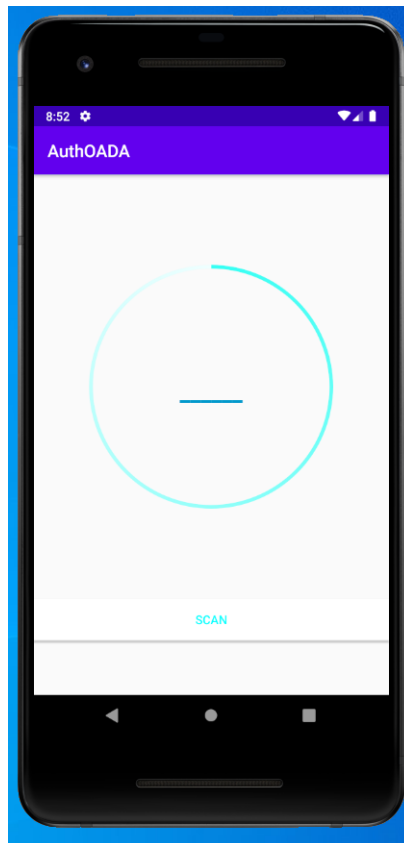


Fig. 4.14: OTP Generator Application.

OTP Generator Scanner

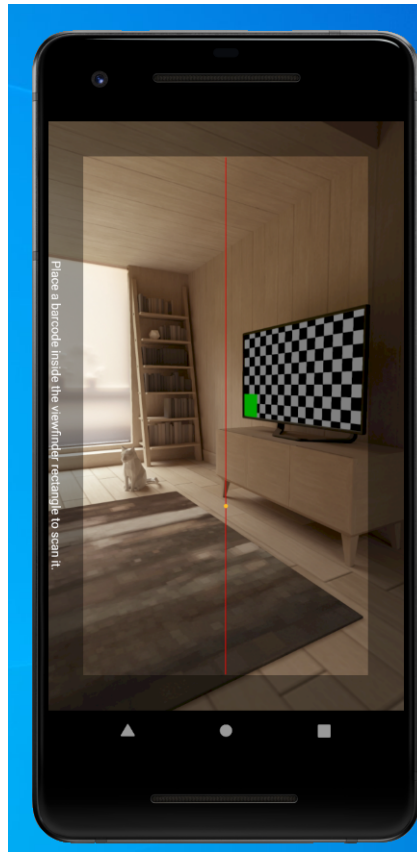


Fig. 4.15: OTP Generator Application Scanner.

5.1 Network Topology

In order to virtualize remote access using VPN SSL, we create the topology showed in Figure 5.1. This topology will be established using the GNS3 software. The sorts of equipment used are:

1. Cisco ASA Firewall: We will use the Anyconnect VPN tool provided by the ASA firewall.
2. Apache Tomcat: We create our web site by using the java servlet, therefore we use the Apache Tomcat server as servlet container.
3. MySQL server: We use My SQL server to store and manage data.
4. The administrator device contains the OADA administrator tools that manage the users' accounts and the ASDM to manage the Cisco ASA.
5. Cisco router.
6. Gns3 cloud to bridge our lab to the real world's internet connection.

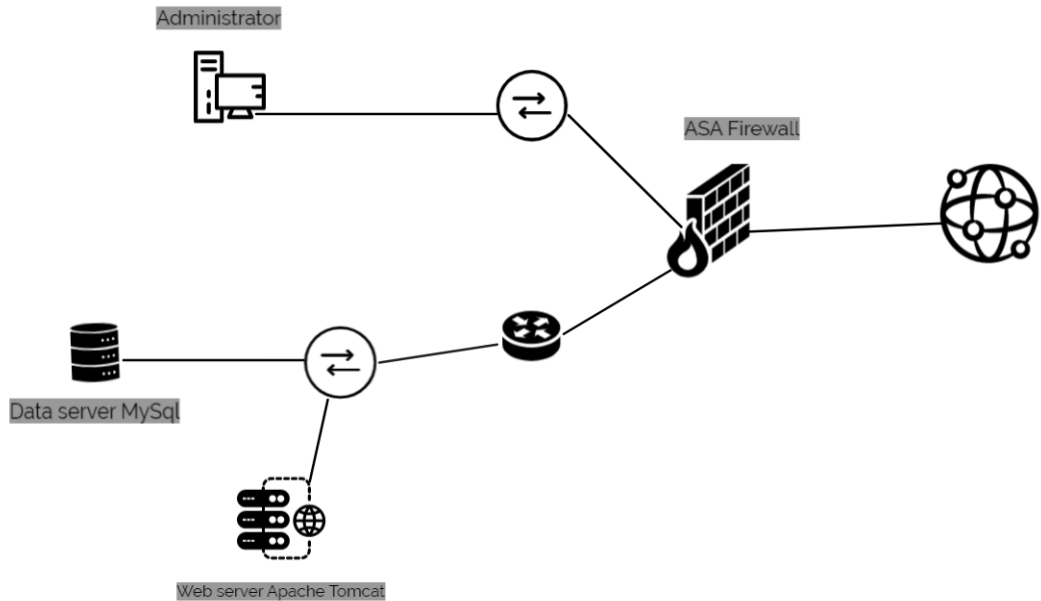


Fig. 5.1: The Topology of VPN SSL Tunnel.

The software used to enable and manage VPN SSL are:

- the ASA Version 9.8(1);
- the ASDM Version 7.8(1);
- the AnyConnect windows client Version 4.8.

The ASA firewall will be configured with three interfaces, the interface connected to the administrator device is configured as the inside interface which means that the security level for this interface is 100, the highest level. The interface connected to the internet is configured as an outside interface so it's an untrusted interface with the security level of 0, the third interface connected to our resources is configured as DMZ with the security level of 50. The ASA traffic policies are:

- the ASA allows the flow of traffic from a higher security level to a lower security level;
- the lower security level (outside or DMZ) traffic to the higher security level is denied by the ASA firewall.

The VPN SSL connection will set on the outside interface. We configure the Anyconnect using the ASDM tool installed on the administrator device.

So to gain access, the user will go through the process presented in Figure ?? as follows:

1. connecting the ASA firewall and download the AnyConnect client software;

2. connecting to the webserver through the VPN SSL tunnel;
3. pass the two-factor; the username/password authentication besides the TOTP authentication, to gain access.

5.2 Software Used

GNS3

GNS3 (Graphical Network Simulator) is a cross-platform graphical network simulator that simulates complex networks as close as possible to the way real networks perform. All of this without having dedicated network hardware. GNS3's graphical interface allows you to create virtualized network labs with a variety of routers, switches, and PCs, and Cisco IOS. GNS3 uses a backend hypervisor application to emulate the hardware that runs Cisco IOS. Because only the hardware is emulated, you run an actual IOS image file on your PC. In this project, we use GNS3 to simulate the whole process. More information in *GNS3 web site*.

Vmware Workstation Pro

VMware Workstation Pro is the industry standard for running multiple operating systems as virtual machines (VMs) on a single Linux or Windows PC. IT professionals, developers, and businesses who build, test, or demo software for any device, platform, or cloud rely on Workstation Pro. We use the VMware Workstation to create the users' computers and to run the GNS3 VM to virtualize the internal network devices. The *Vmware web site* provides more information.

5.3 The GNS3 Environment

5.3.1 The GNS3 VM Configuration

GNS3 VM requirements:

1. 1.5 GHz processor;
2. 4GB RAM;
3. 250MB free disk space.

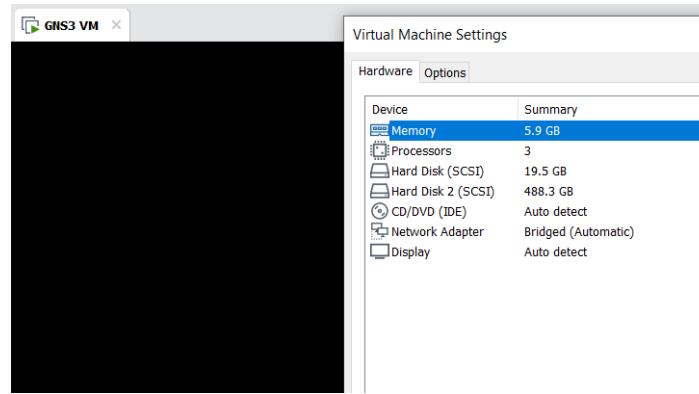


Fig. 5.2: GNS3 VM Configuration.

We are using the VM bridged network ¹ for GNS3 VM.

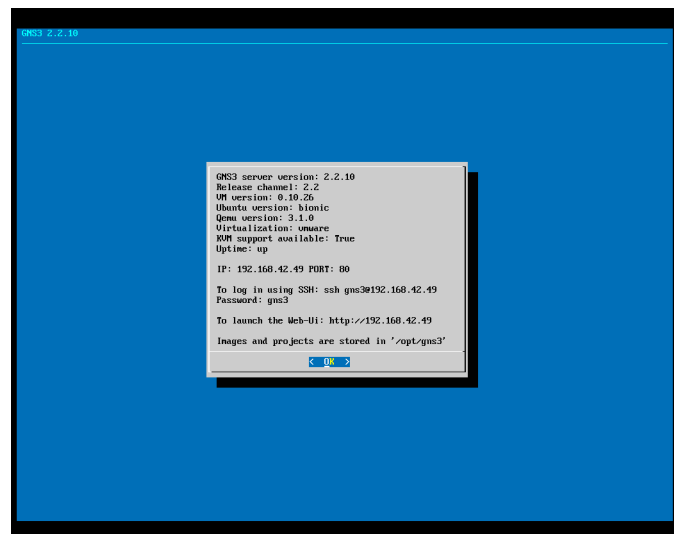


Fig. 5.3: GNS3 VM.

5.3.2 Establishing Network Topology

We create a network present in Figure 5.4. The Apache web server is hosted on the windows appliance machine. The MySQL server is installed on the Linux server appliance.

The Inside interface will be signed to the 10.11.11.0/24 network address. The DMZ interface is connected to the cisco router through the 192.168.4.10/24 network address, the cisco router connects this network to the internal resource network under the 10.3.3.0/24 network address.

The ASA firewall will be configured to use the Google public DNS server under

¹Bridged networking connects a virtual machine to a network using the host computer's Ethernet adapter.

the primary of 8.8.8.8 and to commit ICMP traffic. Also, we configure a user to enable ASA configuration through ASDM.

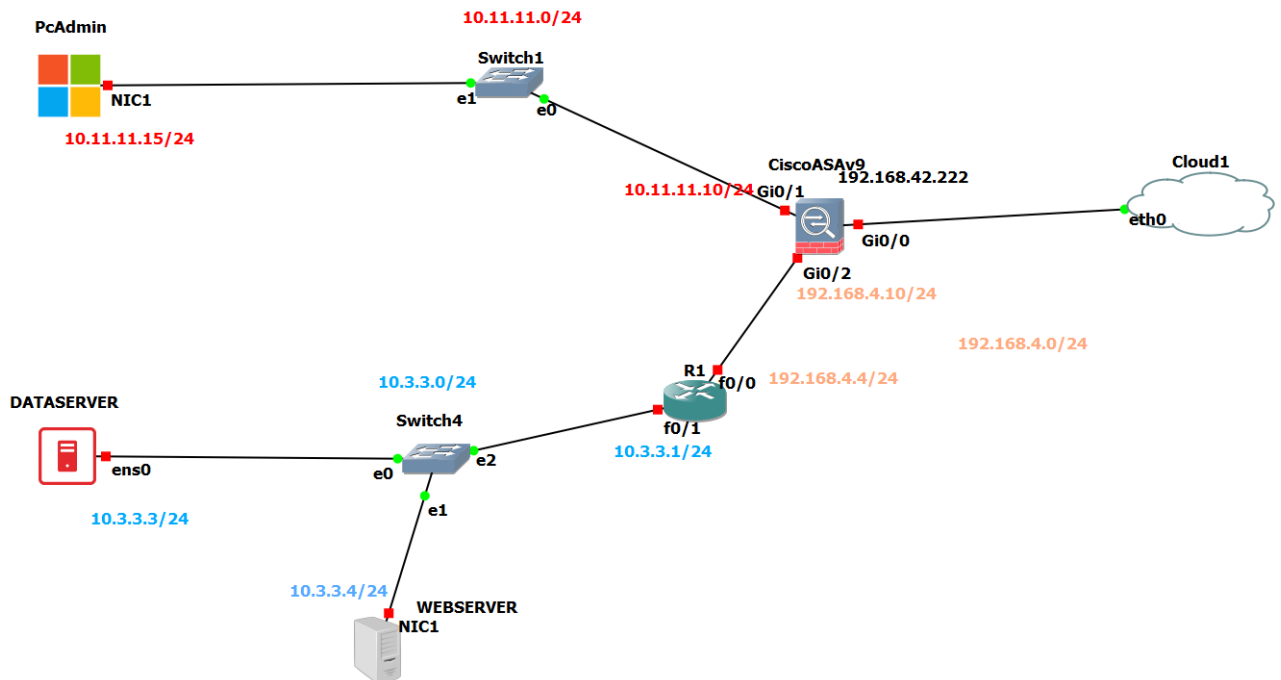


Fig. 5.4: GNS3 Network Topology.

5.3.2.1 Configure the ASA Firewall

- The configuration of the ASA interfaces.

```
ciscoasa# show interface ip brief
Interface      IP-Address      OK? Method Status      Prot
-----
GigabitEthernet0/0  192.168.42.222  YES DHCP    up          up
GigabitEthernet0/1  10.11.11.10    YES CONFIG  up          up
GigabitEthernet0/2  192.168.4.10   YES CONFIG  up          up
GigabitEthernet0/3  unassigned     YES unset   administratively down down
GigabitEthernet0/4  unassigned     YES unset   administratively down down
GigabitEthernet0/5  unassigned     YES unset   administratively down down
GigabitEthernet0/6  unassigned     YES unset   administratively down down
Management0/0      unassigned     YES unset   administratively down down
ciscoasa#
```

Fig. 5.5: GNS3 ASA Interfaces Configuration.

```
ciscoasa# sh run interface
:
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address dhcp setroute
:
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 10.11.11.10 255.255.255.0
:
interface GigabitEthernet0/2
 nameif dmzserver
 security-level 50
 ip address 192.168.4.10 255.255.255.0
:
```

Fig. 5.6: GNS3 ASA Interfaces Parameters Configuration.

- Configure the ASA route table.

```
ciscoasa(config)# sh rout

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, V - VPN
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route
Gateway of last resort is 192.168.122.1 to network 0.0.0.0

S*    0.0.0.0 0.0.0.0 [1/0] via 192.168.122.1, outside
      [1/0] via 192.168.42.129, outside
S     10.3.3.0 255.255.255.0 [1/0] via 192.168.4.4, dmzserver
C     10.11.11.0 255.255.255.0 is directly connected, inside
L     10.11.11.10 255.255.255.255 is directly connected, inside
C     192.168.4.0 255.255.255.0 is directly connected, dmzserver
L     192.168.4.10 255.255.255.255 is directly connected, dmzserver
C     192.168.42.0 255.255.255.0 is directly connected, outside
L     192.168.42.222 255.255.255.255 is directly connected, outside

ciscoasa(config)#
```

Fig. 5.7: GNS3 ASA Route Table Configuration.

- Configure the ASA DNS.

```
ciscoasa#
ciscoasa# conf t
ciscoasa(config)# dns domain-lookup outside
ciscoasa(config)# dns name-server 8.8.8.8
ciscoasa(config)#
```

Fig. 5.8: GNS3 ASA DNS Configuration.

5.3.2.2 Configure the Router

- The router configuration.

```
R1#show ip interface brief
Interface                IP-Address      OK? Method Status      Prot
FastEthernet0/0          192.168.4.4     YES NVRAM  up          up
FastEthernet0/1          10.3.3.1        YES NVRAM  up          up
Serial1/0                 unassigned      YES NVRAM  administratively down down
Serial1/1                 unassigned      YES NVRAM  administratively down down
Serial1/2                 unassigned      YES NVRAM  administratively down down
Serial1/3                 unassigned      YES NVRAM  administratively down down
NVI0                      192.168.4.4     YES unset  up          up
```

Fig. 5.9: GNS3 Router Configuration.

- The router route table.

```
+ - replicated route, % - next hop override
Gateway of last resort is 192.168.4.10 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 192.168.4.10
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   10.3.3.0/24 is directly connected, FastEthernet0/1
L   10.3.3.1/32 is directly connected, FastEthernet0/1
L   192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.4.0/24 is directly connected, FastEthernet0/0
L   192.168.4.4/32 is directly connected, FastEthernet0/0
R1#
```

Fig. 5.10: GNS3 Router Routing Configuration.

5.3.2.3 GNS3 Cloud Connection

We use the Ethernet adapter of the GNS3 VM to connect the cloud node to the network.

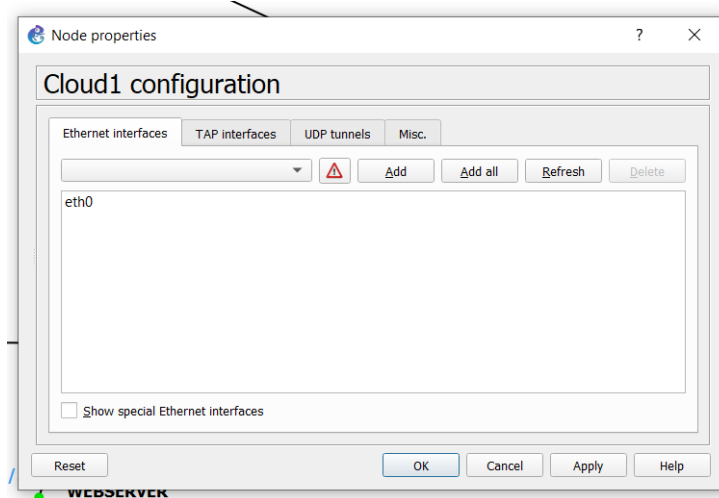


Fig. 5.11: GNS3 Cloud Configuration.

5.3.2.4 The VPCs Configuration

1. PcAdmin (Windows QEMU) configuration.

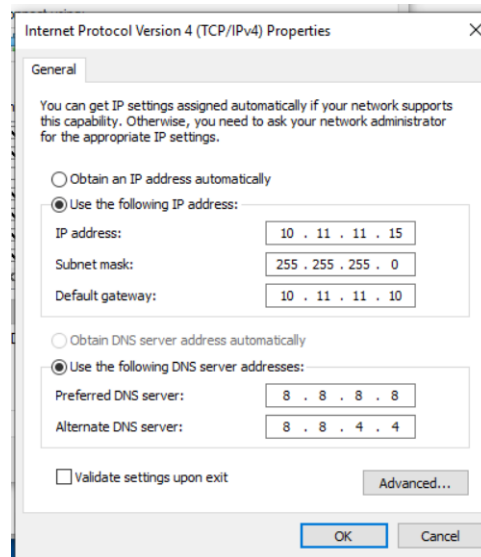


Fig. 5.12: GNS3 PcAdmin Configuration.

2. Data server (Linux server QEMU) configuration.



Fig. 5.13: GNS3 Linux QEMU Configuration.

3. Web server (Windows QEMU) configuration.

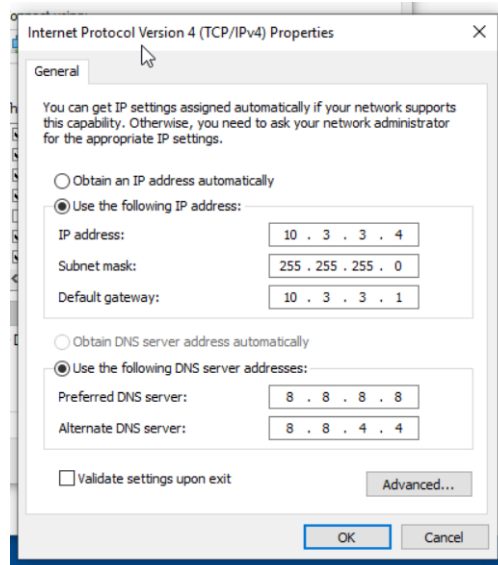


Fig. 5.14: GNS3 Windows QEMU Configuration.

5.3.2.5 ASDM Installation

1. Connect the ASA firewall.

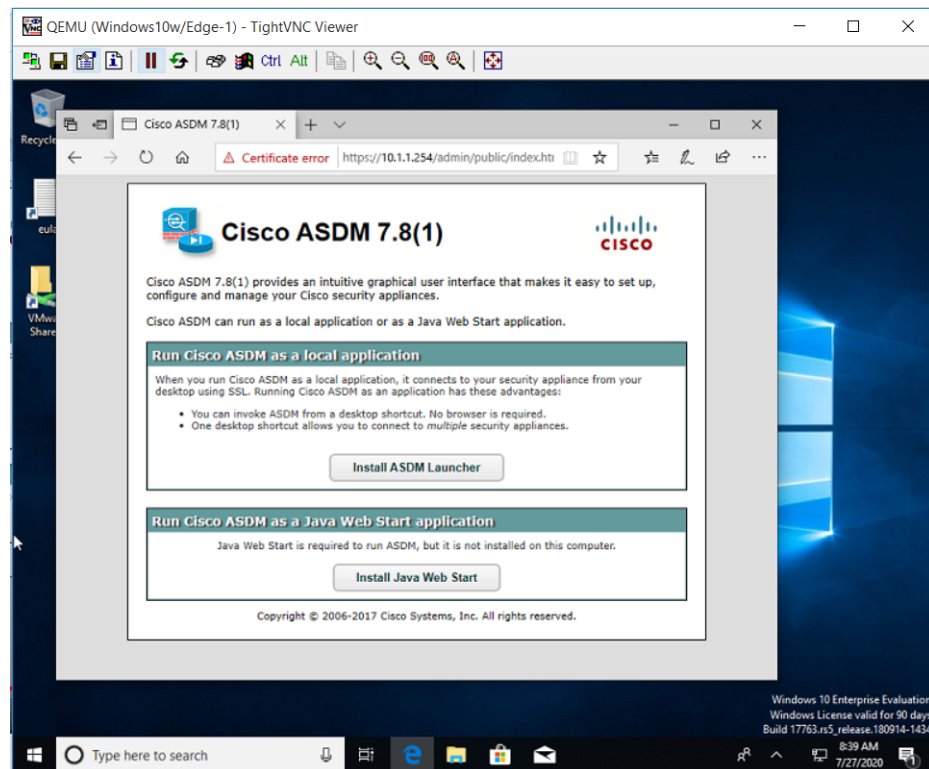


Fig. 5.15: GNS3 ASA ASDM.

2. Download java run-time.

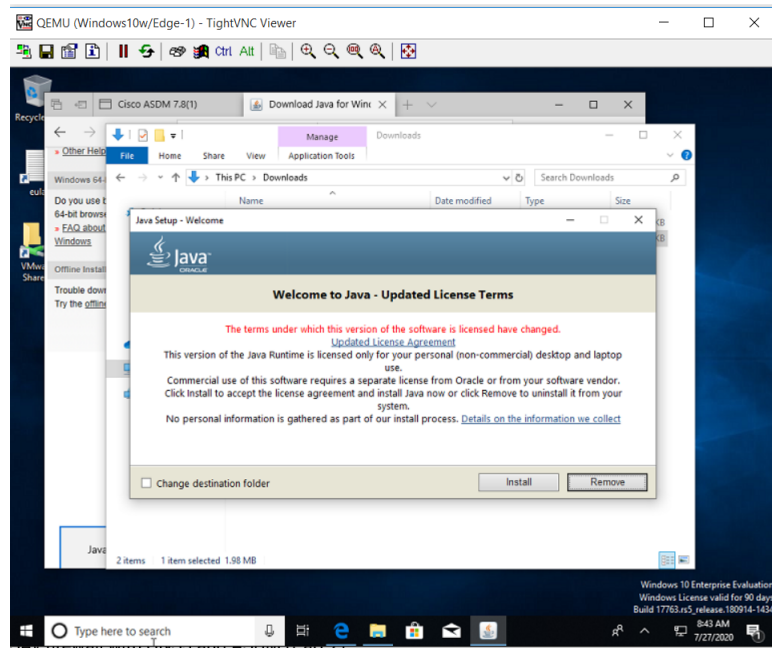


Fig. 5.16: GNS3 ASA ASDM JAVA Instalation.

3. Download ASDM.
4. Launch ASDM.

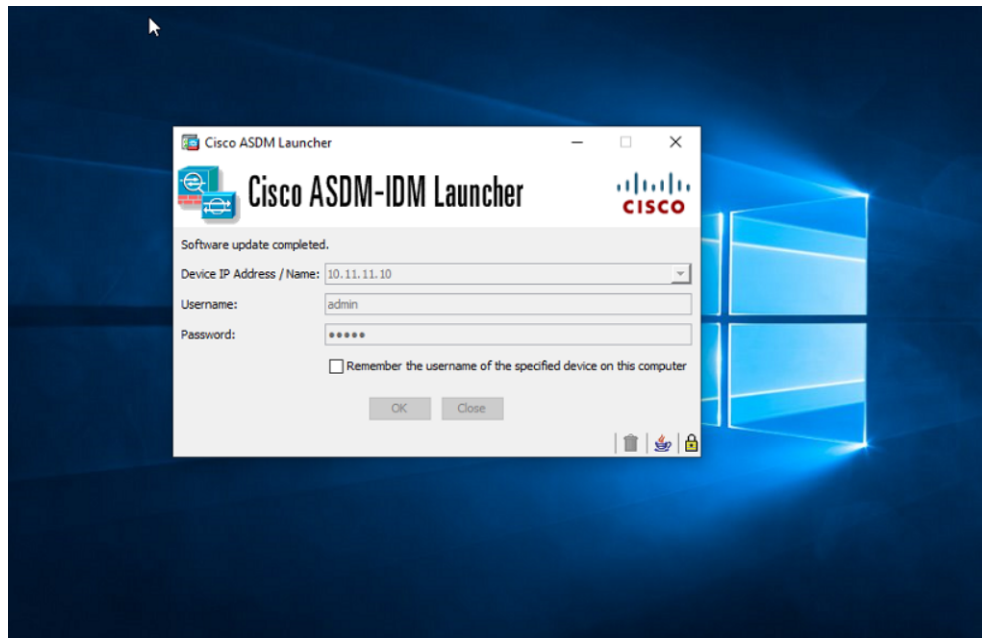


Fig. 5.17: GNS3 ASA ASDM Connect.

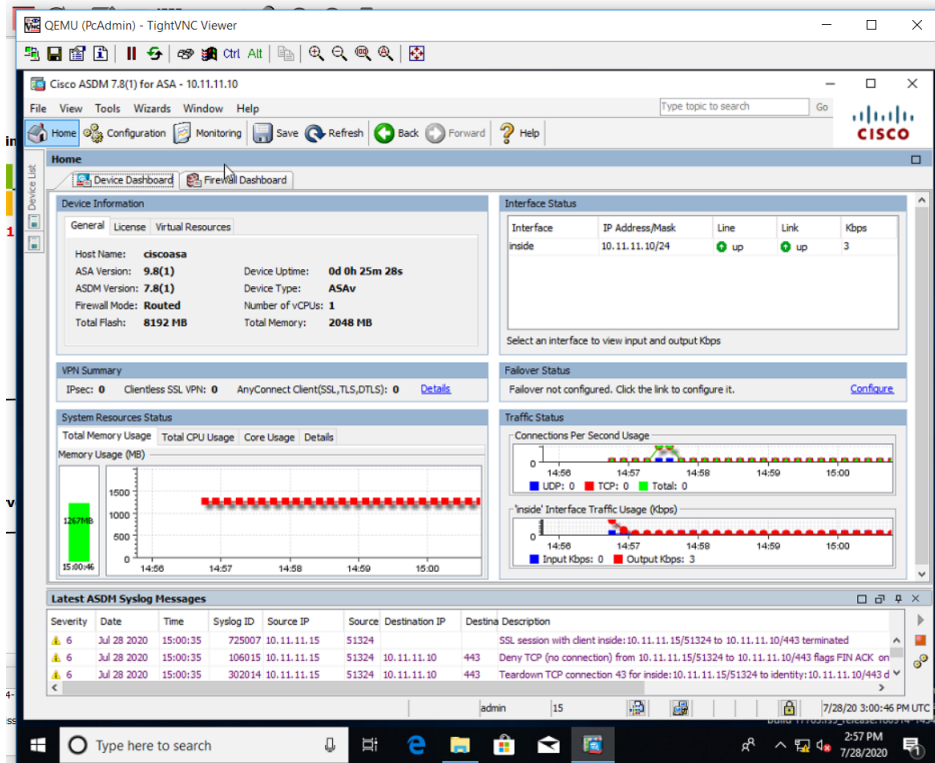


Fig. 5.18: GNS3 ASA ASDM Interface.

5. Enable the ICMP protocol.

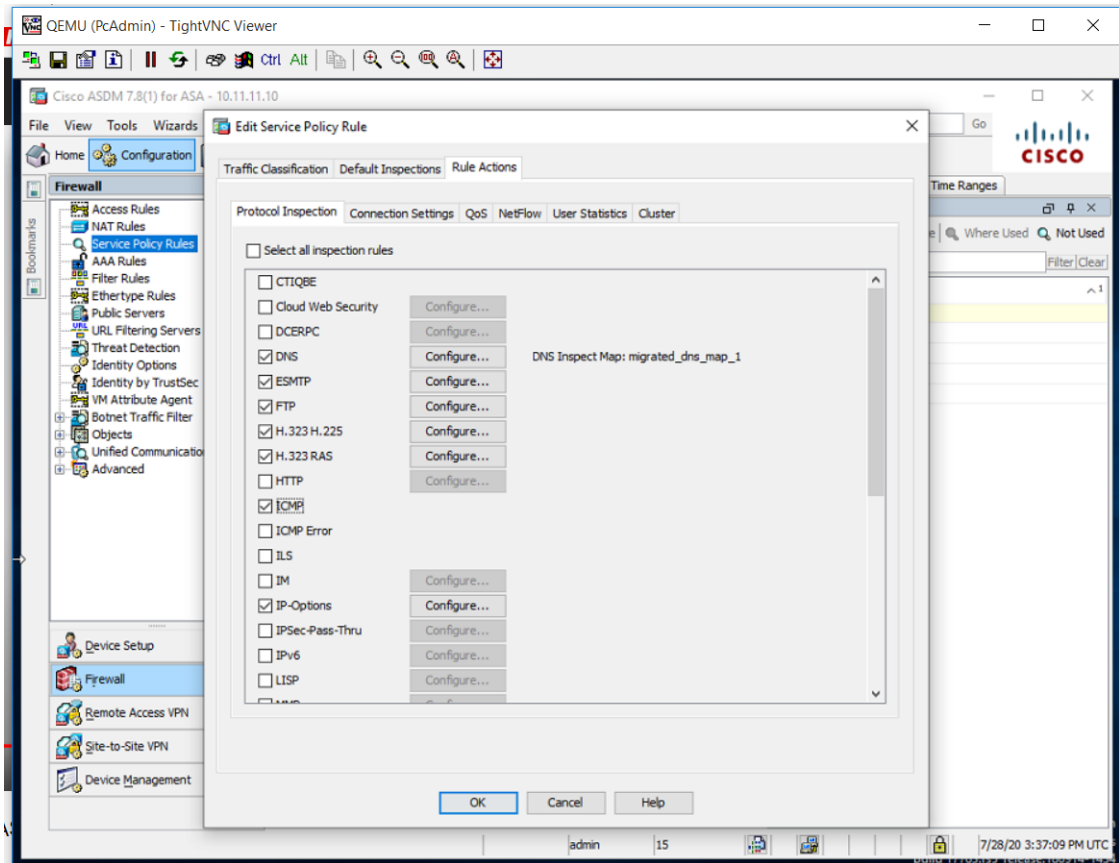


Fig. 5.19: GNS3 ASA ASDM ICMP Enabling.

5.3.2.6 Web Server Configuration

Deploy the web application in the Tomcat server in the GNS3 Windows QEMU.

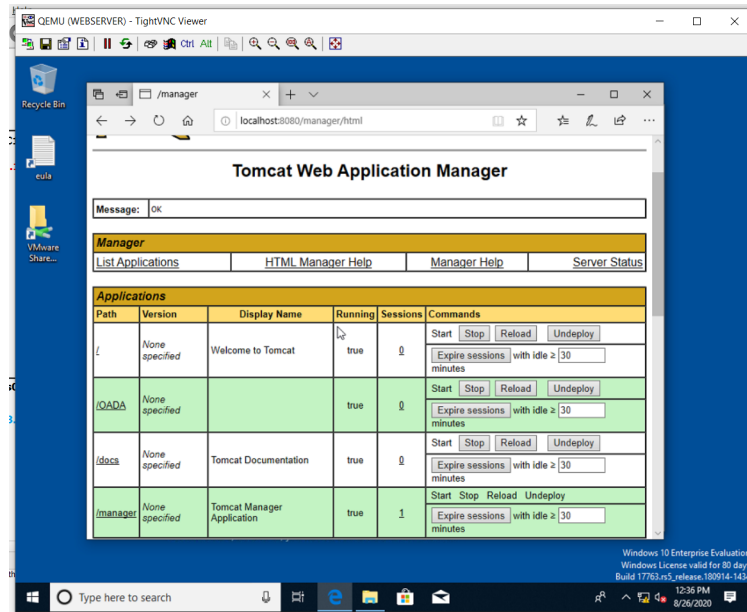


Fig. 5.20: Web Application Deployment.

5.3.2.7 Database Configuration

Described in section ??

5.3.2.8 AnyConnect Configuration

1. We log into the ASDM and launch the Configuration Wizard.

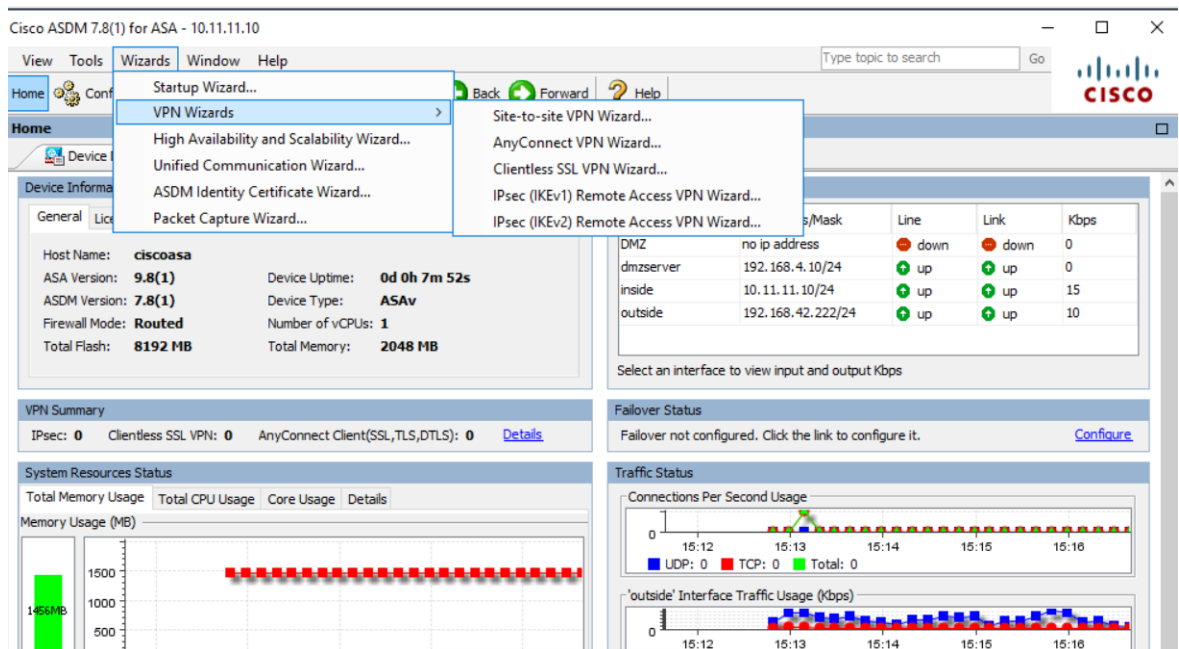


Fig. 5.21: ASDM Interface

2. We enter the Connection Profile Name and choose the interface on which the VPN will be achieved.

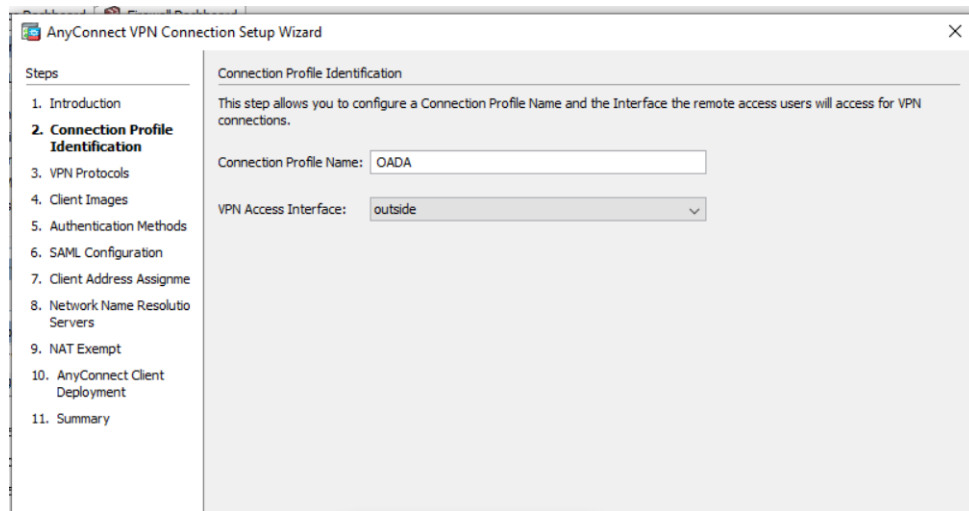


Fig. 5.22: AnyConnect Wizard Interface

3. Enable Secure Sockets Layer (SSL) and add the activate certificate.

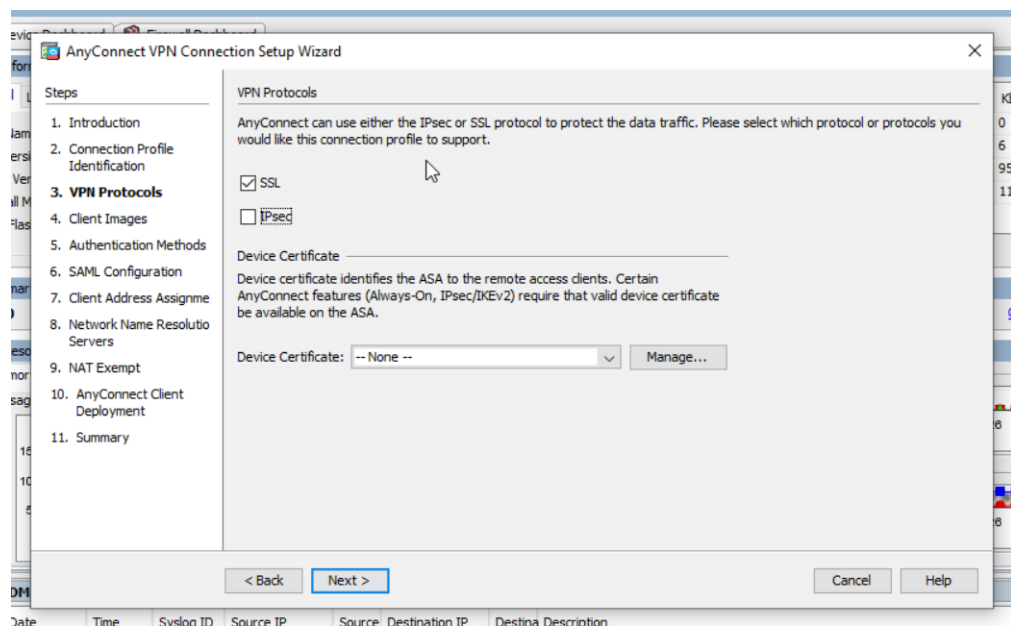


Fig. 5.23: AnyConnect Wizard Interface Enable SSL

4. We add the AnyConnect Client image (.pkg file).

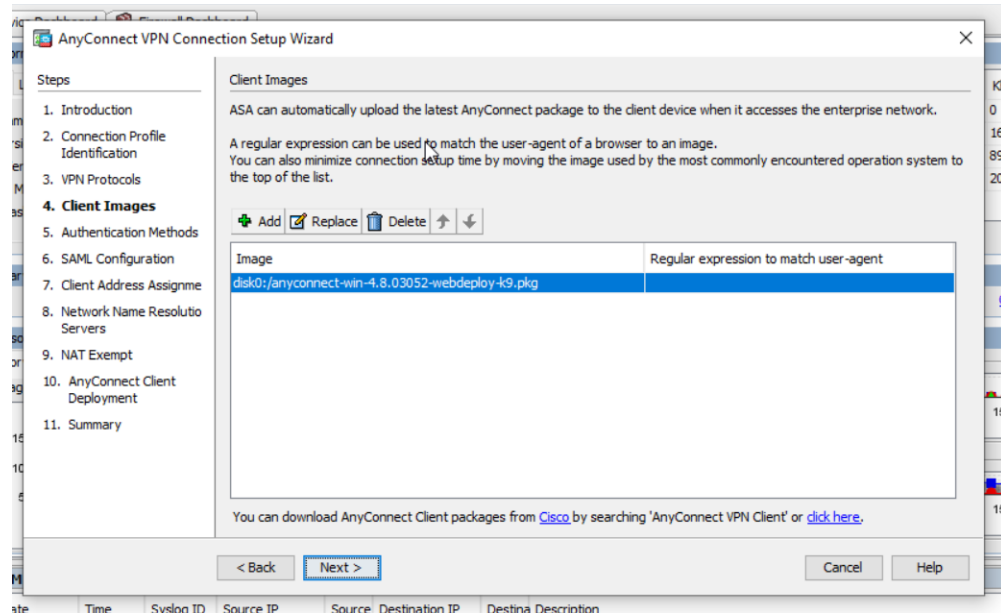


Fig. 5.24: AnyConnect Wizard Interface AnyConnect Client image

5. We chose the local user database for authentication.

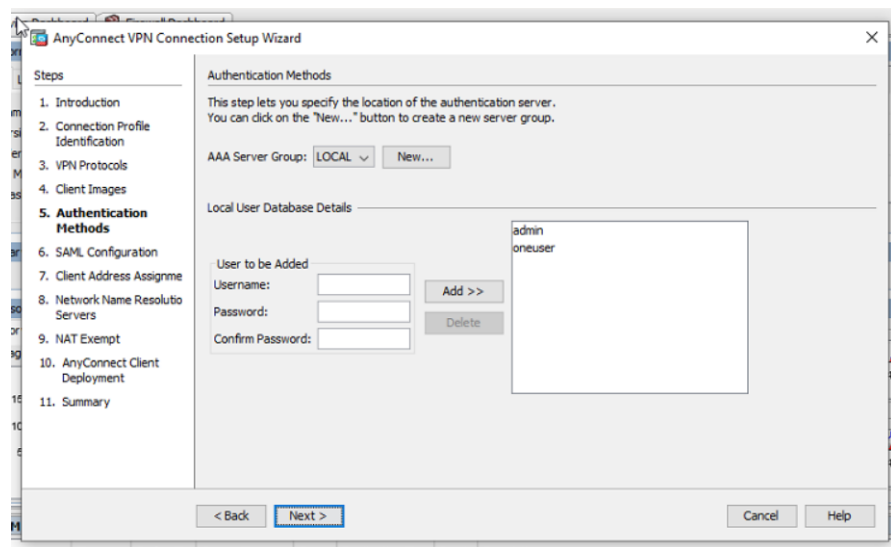


Fig. 5.25: AnyConnect Wizard Interface AAA Server

6. For the SAML Configuration, we leave the default settings for a SAML and the authentication method.

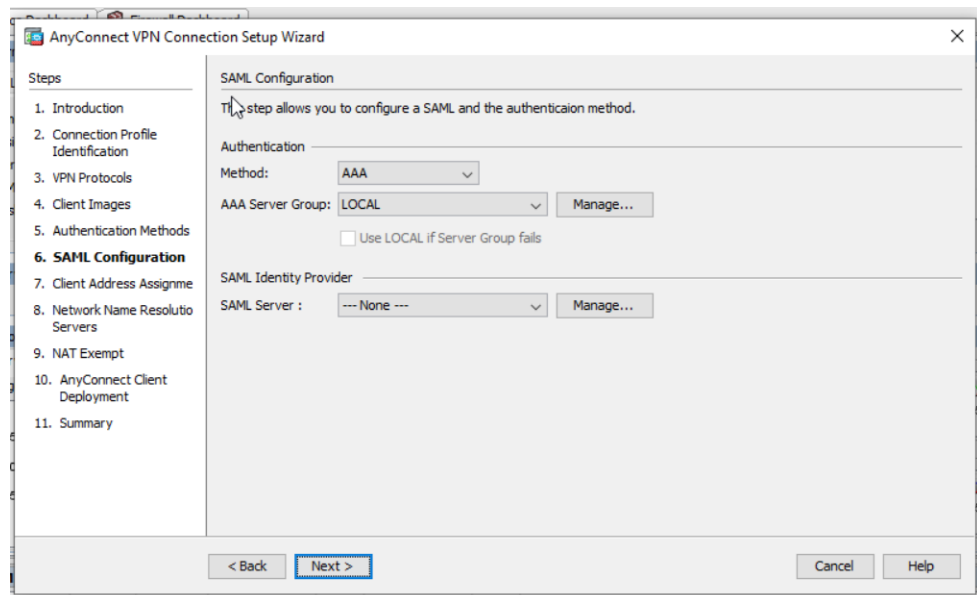


Fig. 5.26: AnyConnect Wizard Interface SAML Configuration

7. We configure the address pool for the VPN client.

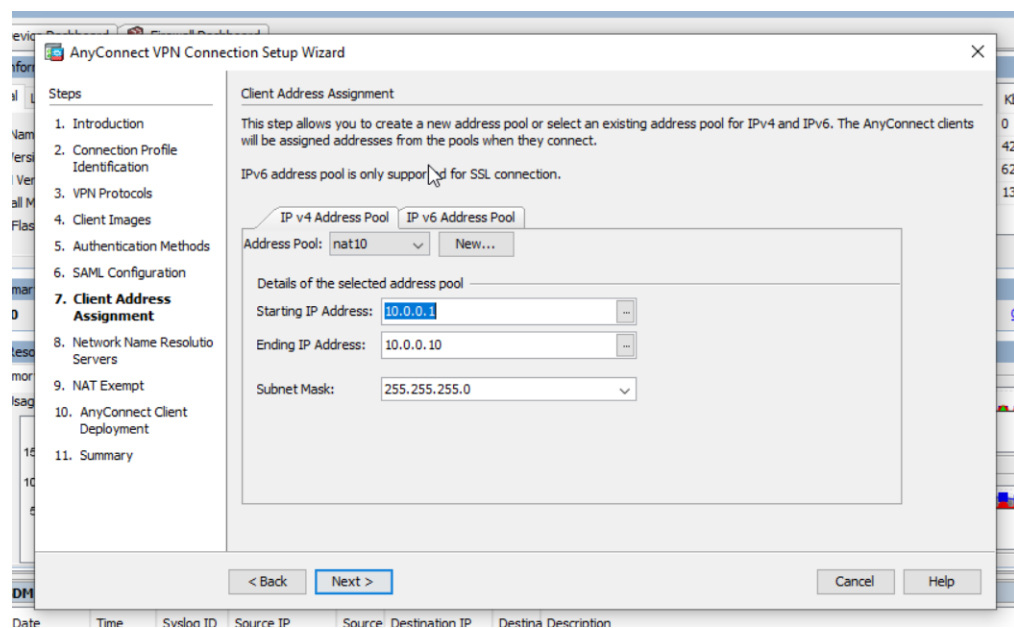


Fig. 5.27: AnyConnect Wizard Interface AnyConnect Address Pool

8. We configure the DNS server.

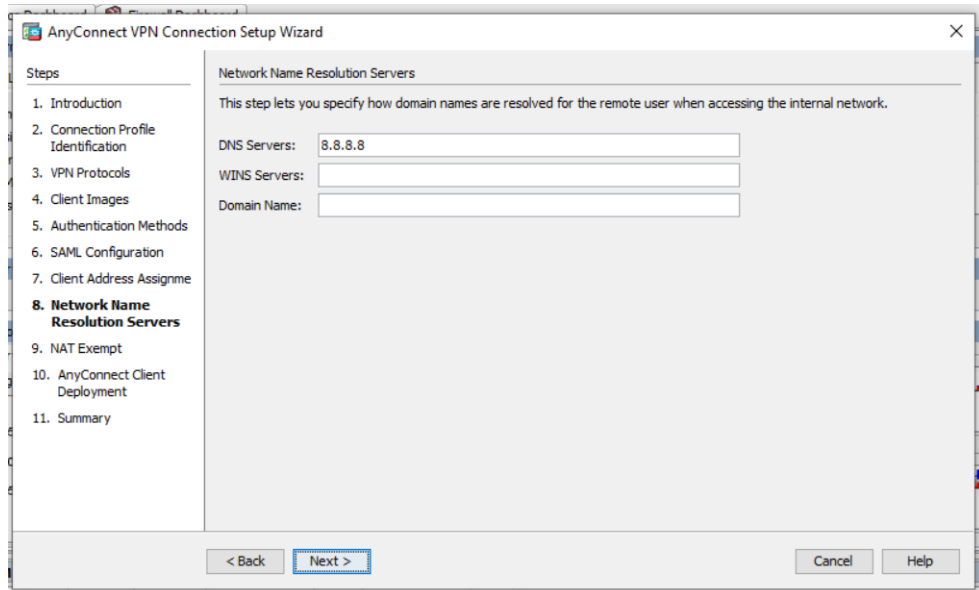


Fig. 5.28: AnyConnect Wizard Interface AnyConnect DNS

9. AnyConnect Profile created.

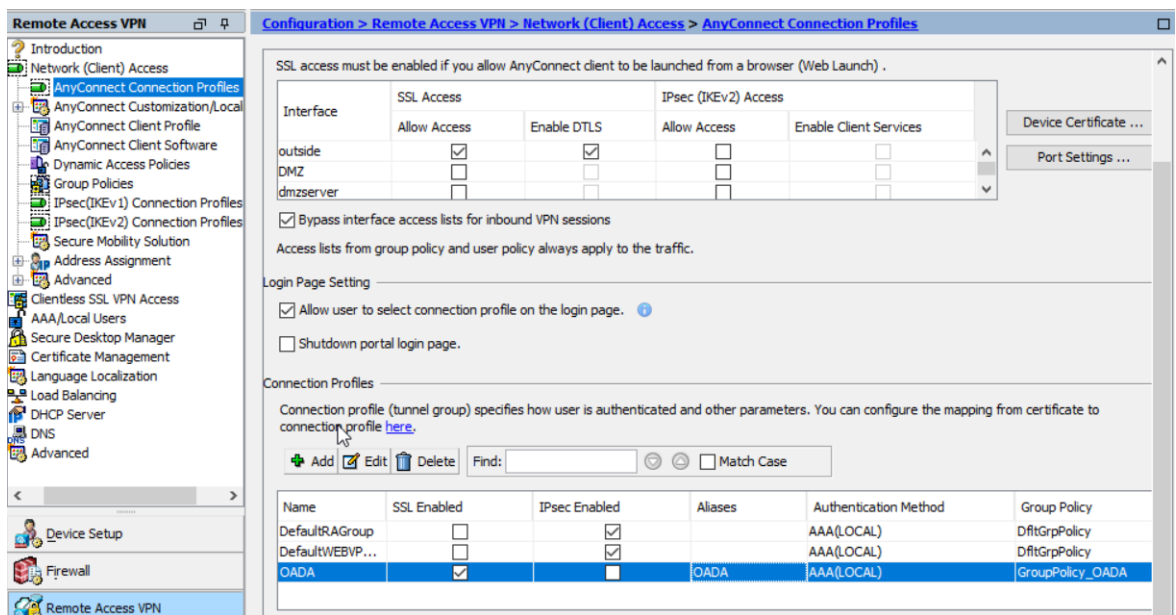


Fig. 5.29: AnyConnect Profile

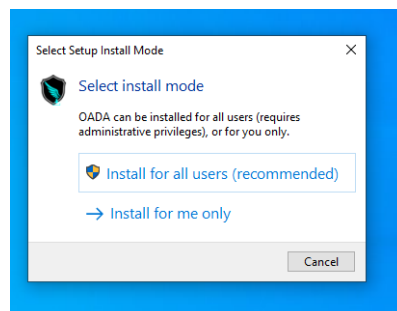
6

RESULT AND INTERPRETATION

6.1 OADA Application

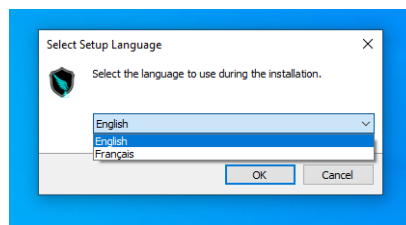
6.1.1 OADA Instalation Process

1. Launch the installer file.



Instalation Process.

2. Chose installation language.



Instalation Process.

3. Agree on license and chose installation folder.

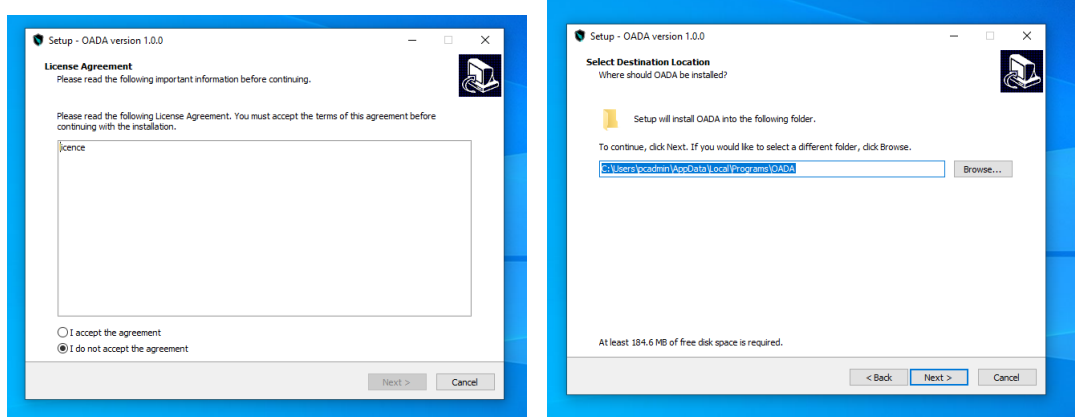
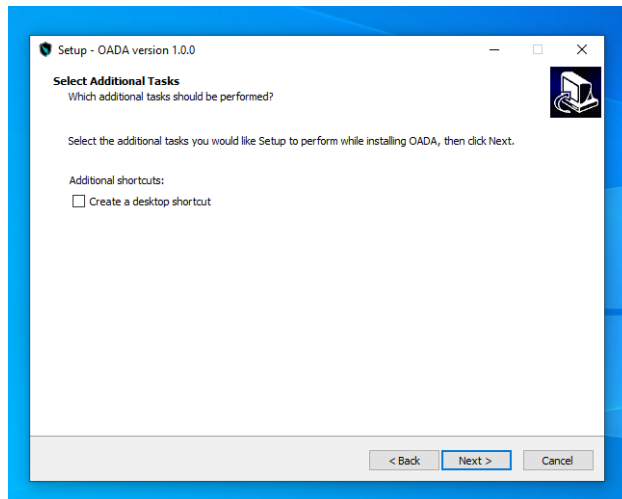


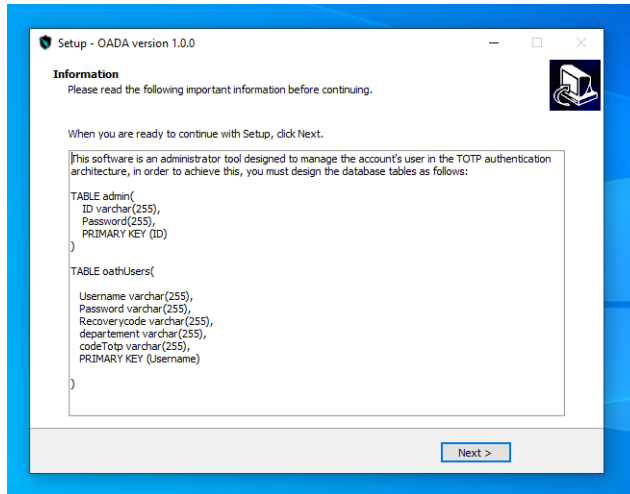
Fig. 6.1: Instalation Process.

4. Create a desktop shortcut and valid installation.



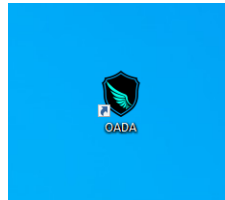
Instalation Process.

5. At the end of the installation, we display some information about configuring the database.



Instalation Process.

6. Run The application desktop shortcut.



Instalation Process.

6.1.2 OADA Application Examination

1. Main interface.

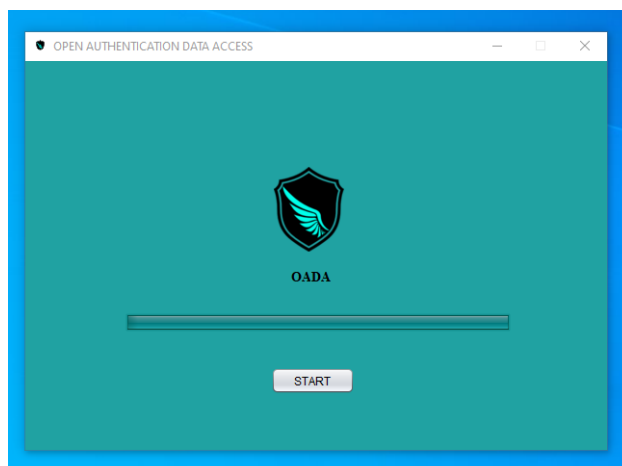


Fig. 6.2: OADA Application.

2. Provide the necessaire configuration to connect the database by selecting

server configuration from the edit list.

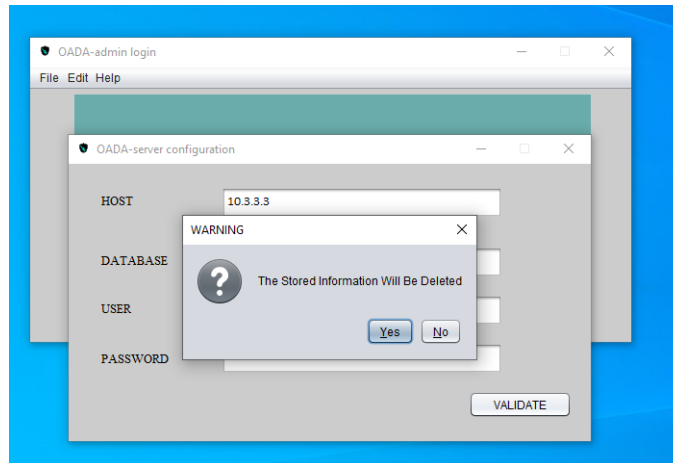


Fig. 6.3: OADA Server Configuration Interface.

3. Login.

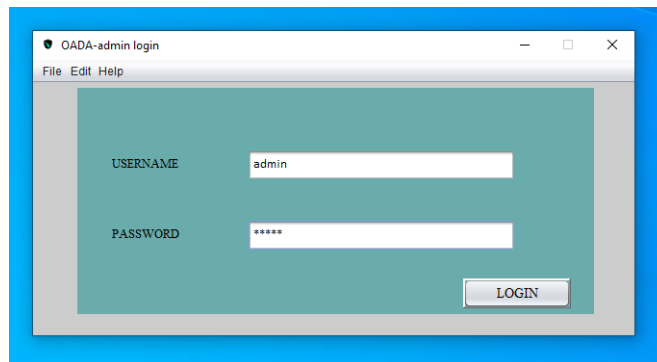


Fig. 6.4: OADA Application Login Interface.

4. Manage user interface.

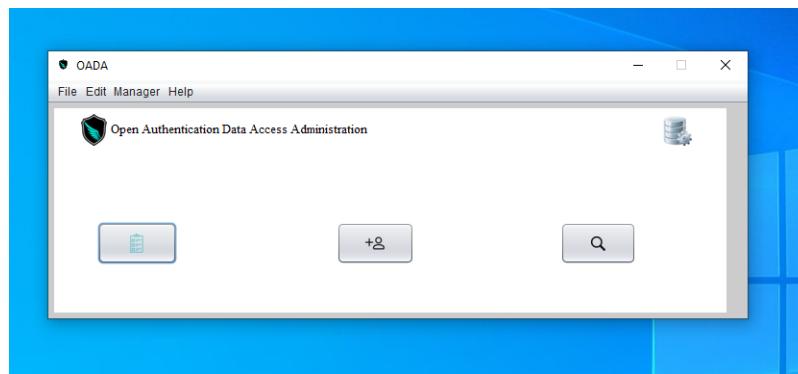


Fig. 6.5: OADA Application Manage Users Interface.

5. Add user.

(a) type user information;

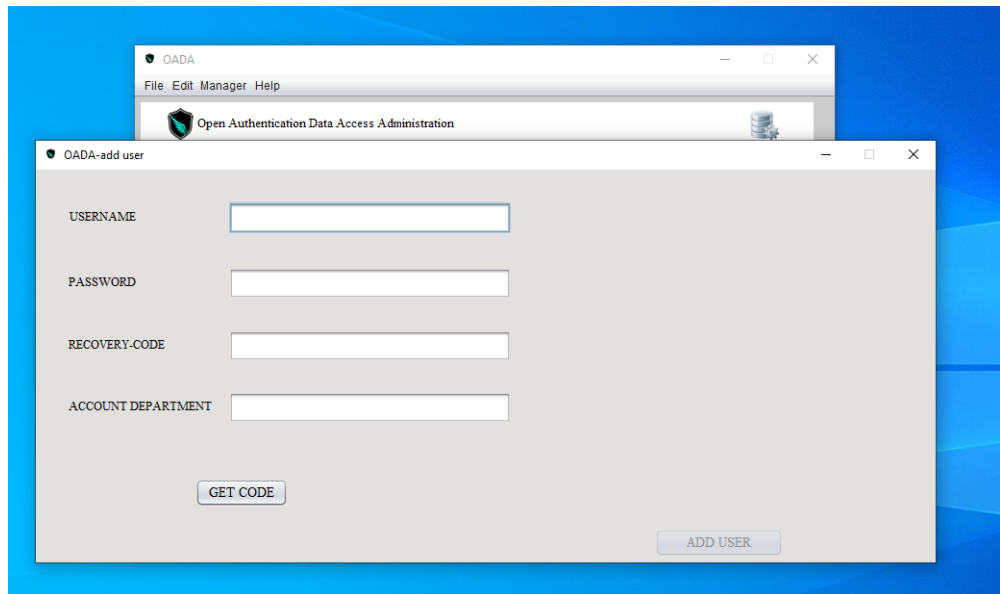


Fig. 6.6: OADA Application ADD Users Interface.

(b) get the QR code;

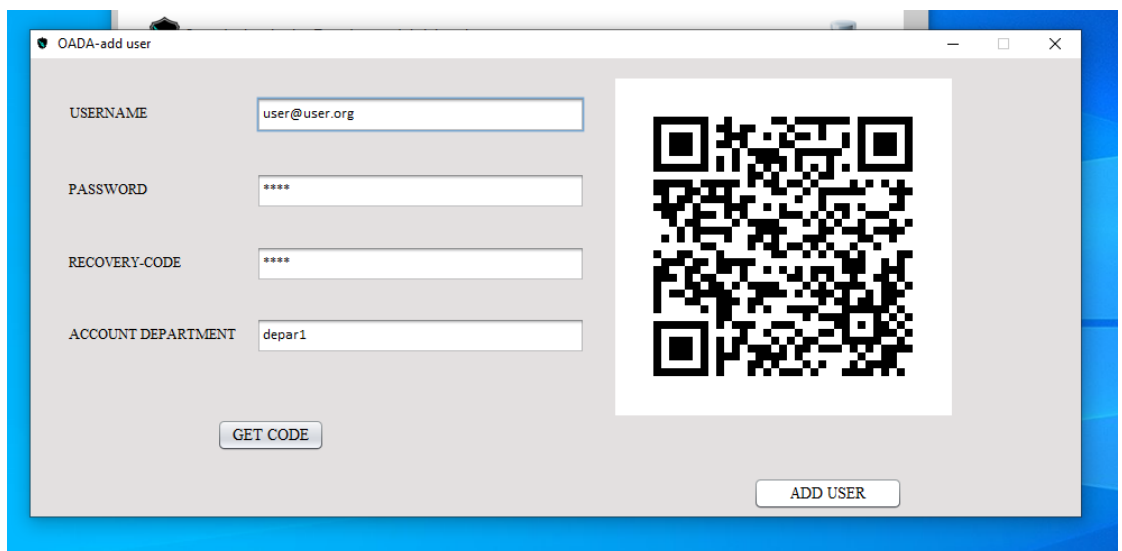


Fig. 6.7: OADA Application ADD Users Interface Generate QR-code.

6. Show users list by clicking the refresh button.

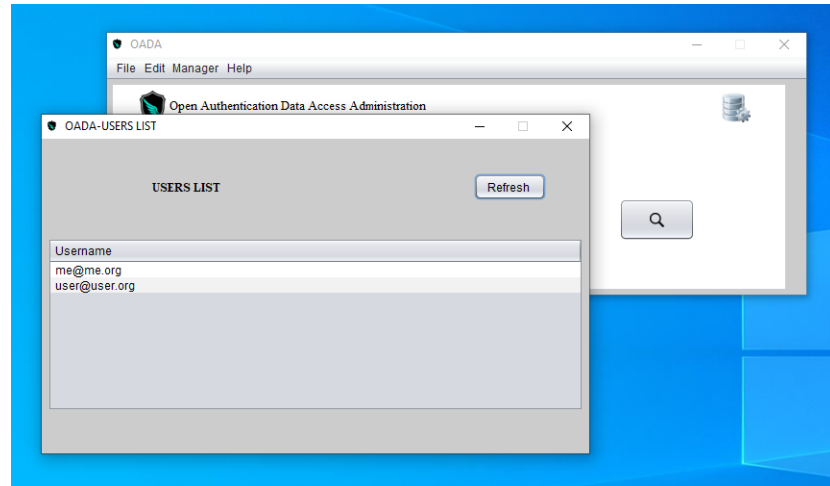


Fig. 6.8: OADA Application Users List Interface .

7. Search for a user by username.

(a) user exists;

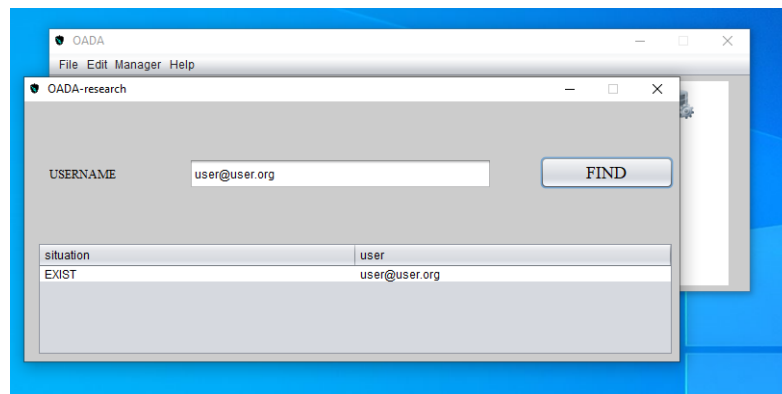


Fig. 6.9: OADA Application Find Users- Exist .

(b) user does not exist.

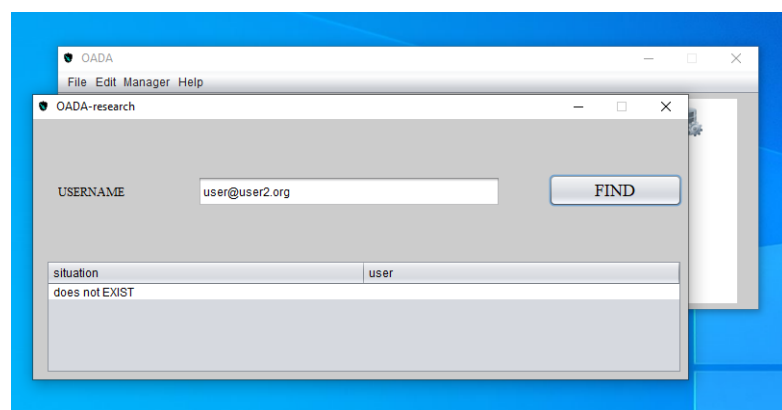


Fig. 6.10: OADA Application Find Users-Does not Exist .

8. Delete user.

(a) username/password invalid;

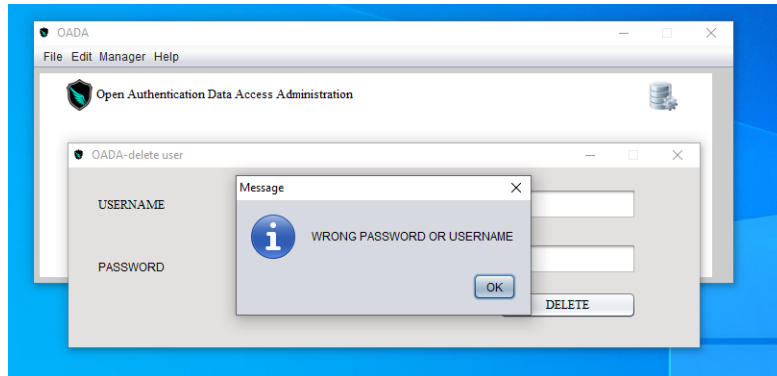


Fig. 6.11: OADA Application Delete Users-Invalid User/Password

(b) username/password valid.

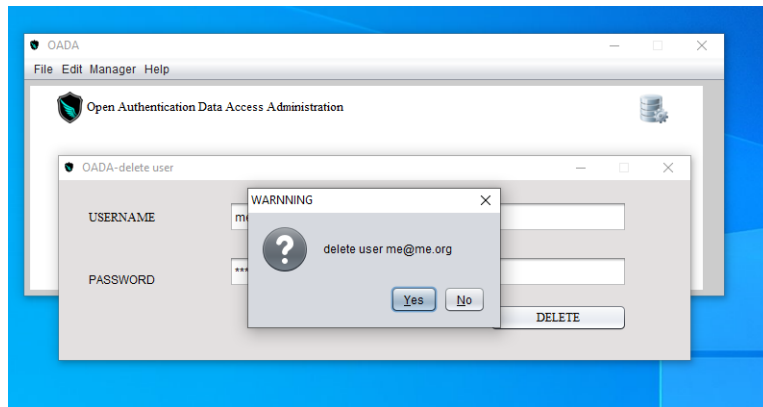


Fig. 6.12: OADA Application Delete Users-Valid User/Password

9. Help List.

(a) About;

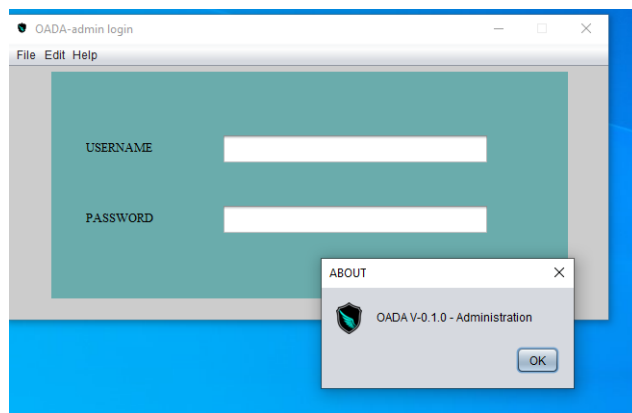


Fig. 6.13: OADA Application About .

(b) Docs.

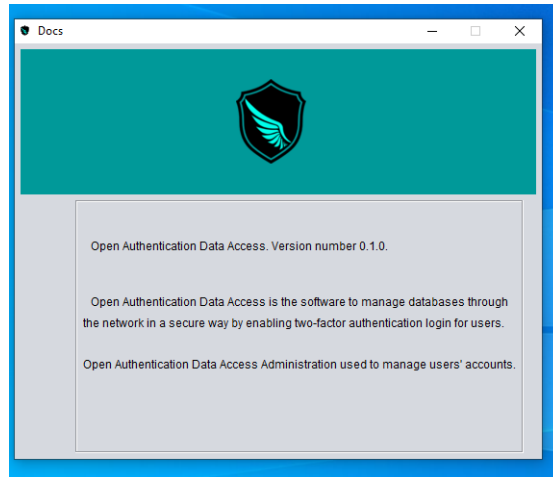


Fig. 6.14: OADA Application Docs .

6.2 VPN SSL Connection

1. Connect the ASA firewall, chose the group and type the username/password.

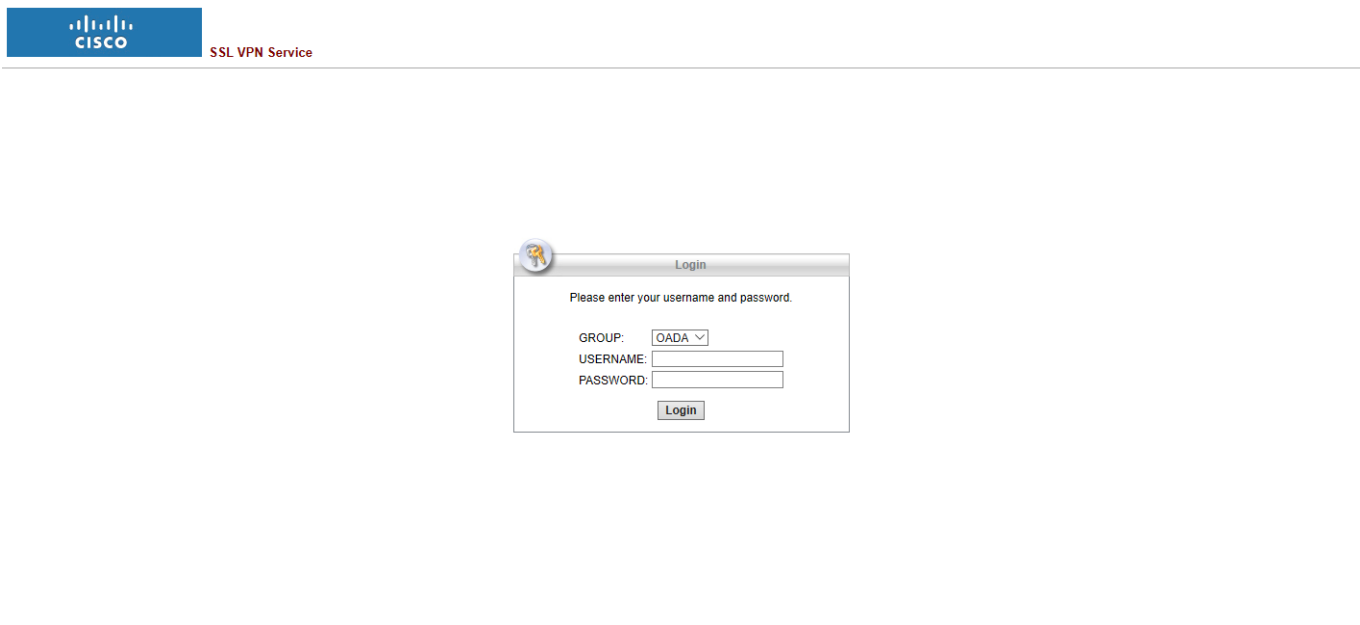


Fig. 6.15: ASA Connection.

2. Download AnyConnect software.

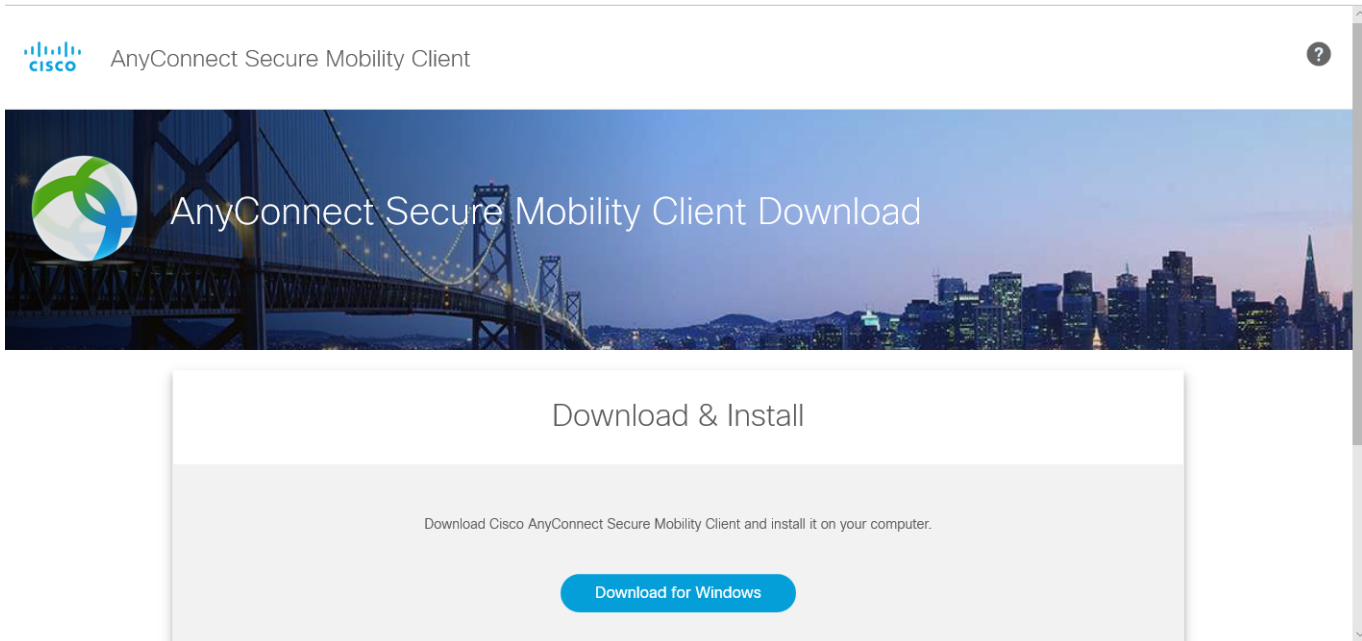


Fig. 6.16: Download AnyConnect.

3. Install AnyConnect software.

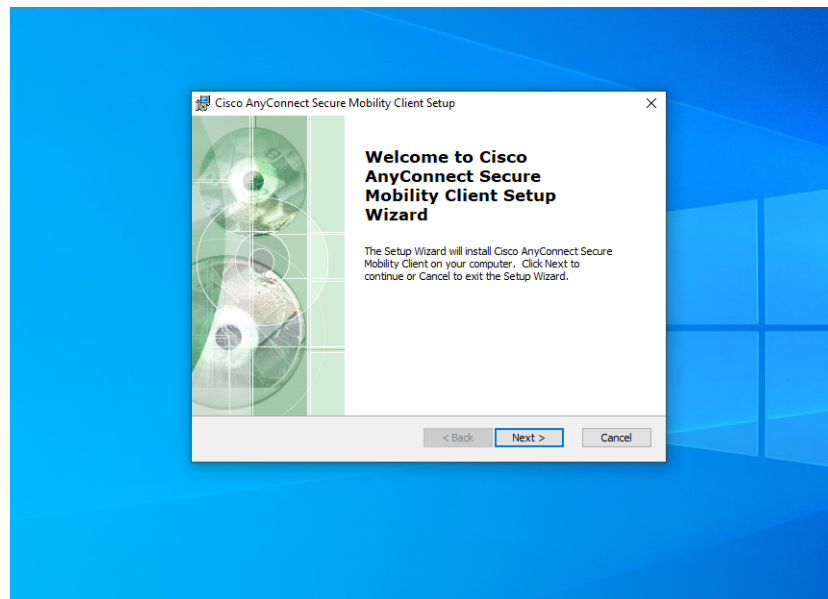


Fig. 6.17: Instalation of AnyConnect.

4. Enable VPN SSL connection.

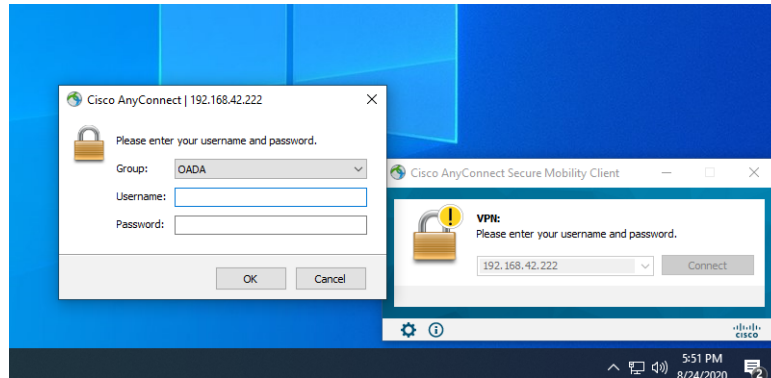


Fig. 6.18: VPN SSL Conection.

5. Obtain an IP address from the configured pool.

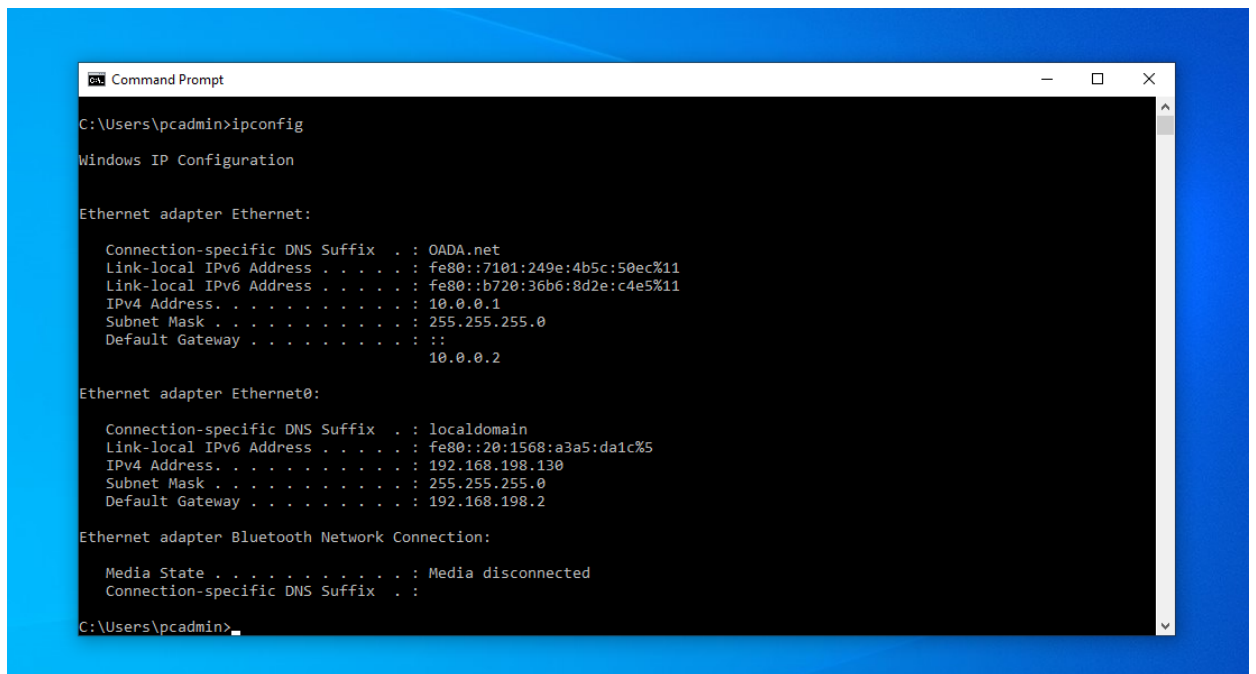


Fig. 6.19: IP Address.

6.3 User Authentication

- (a) Install the android application, and scan the QR code that has been provided while the creating account process.



Fig. 6.20: AuthOADA Android App.

(b) Connect to the webserver and provide valid Username/Password.

A screenshot of a web login interface. The interface is centered on a dark blue background. It features a white rectangular box with the title "Login" at the top. Below the title are two input fields: the first is for the username, and the second is for the password, with the word "Password" visible inside the field. At the bottom of the white box is a blue button with the text "Login" in white.

Fig. 6.21: Web Login Interface.

(c) Type the OTP generated from the AuthOADA android application.

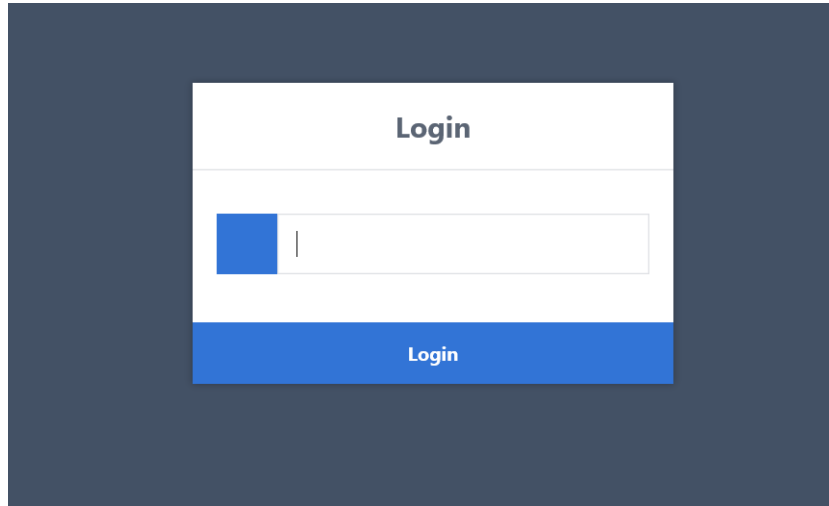


Fig. 6.22: Web TOTP Authentication Interface.

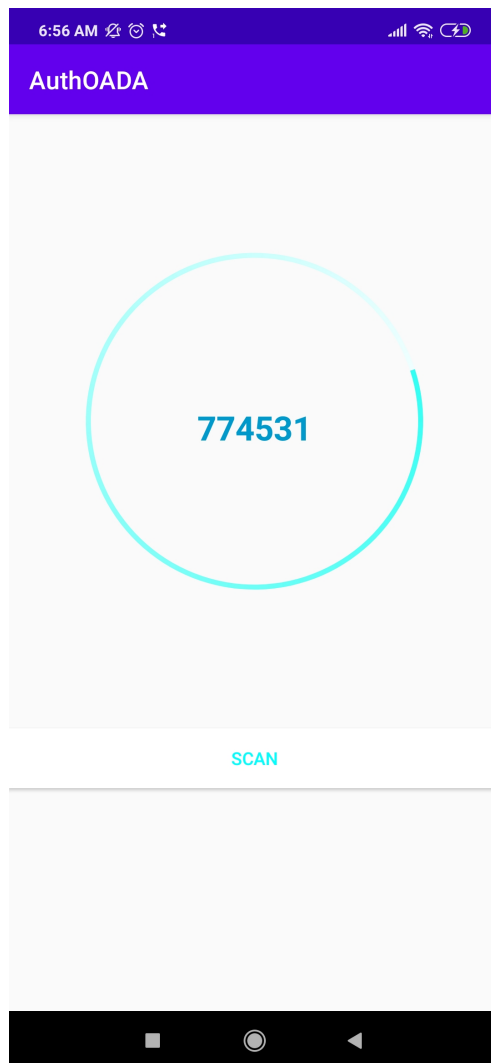


Fig. 6.23: OTP Code.

(d) Valid TOTP code.

Servlet profile at /oadausers

Fig. 6.24: Web Profile Servlet.

(e) Inalid TOTP code.

Servlet notexist at /oadausers

Fig. 6.25: Access Denied Servlet.

6.4 Results Discussion

In this project, we implement a strong authentication architecture based on the two approaches, the open authentication, and VPN SSL. First. We start by developing the oath methodologies, so we develop a basic function to perform this operation and then adopt them into the different environments; a basic web interface to authenticate the user, an android application to generate the TOTP code, and java application for administration purposes. Second, we create a VPN SSL tunnel over the ASA AnyConnect. Finally, We tested and validate the whole process therefore we confirmed that our theory and the demand correspond well to what we have achieved.

CONCLUSION

In this project, we implement a strong authentication project by adding extra layers of authentication to the classic username password mechanism. We bind the TOTP authentication with the VPN SSL technology that secures data exchange and adds an additional layer of the authentication process.

We create a basic web interface to authenticate the user through the two-factor authentication, the username/ password authentication then the TOTP authentication, the TOTP code is generated with the android application that we developed, besides a java application for administration features such as add new user, delete the user and search for a user. The VPN SSL tunnel was created using the ASA AnyConnect, and tho this nowadays conditions, we are not able to bring down this proposed architecture to a real network, tho it was planned for a traineeship in the Algeria Bank, so the network is simulated using the GNS3 tool, and no signed and activated certificate was provided and no group policies were presented.

It is possible to improve security by adding extra features to the architecture, such an alert that notifies the user in case the username/password authentication is validated, but the TOTP code authentication is not valid, also an alert based on the GPS could be added to the TOTP android generator, several of security features could be added to this project. Our purpose to create extra TOTP factor authentication inside a VPN SSL tunnel is achieved.

AES

In this program, we use AES-128 bit (16 bytes) as follows:

AES-key Derivation

```
try {  
  
    key = myKey.getBytes("UTF-8");//convert the usedkey string to bytearray  
    sha = MessageDigest.getInstance("SHA-1");//sha-1 to generate a hash then adjustthe result to 128bits=6 bytes  
    key = sha.digest(key);  
    key = Arrays.copyOf(key, 16);  
    secretKey = new SecretKeySpec(key, "AES");  
}
```

Fig. 6.26: AES Key Derivation .

AES Encrypt

```
Cipher thecipher = Cipher.getInstance("AES/ECB/PKCS5Padding");  
thecipher.init(Cipher.ENCRYPT_MODE, secretKey);  
return Base64.getEncoder().encodeToString(thecipher.doFinal(strToEncrypt.getBytes("UTF-8")));
```

Fig. 6.27: AES Encrypt Code.

AES Decrypt

To perform a decrypt operation, we change the opmode to DECRYPT_MODE.

```

Cipher thecipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
thecipher.init(Cipher.DECRYPT_MODE, secretKey);
return new String(thecipher.doFinal(Base64.getDecoder().decode(ToDecrypt)));

```

Fig. 6.28: AES Decrypt Code.

SHA-256

Java provides inbuilt MessageDigest ¹ class for SHA-256 hashing:

1. We use the digest() method calculate message digest of an input and return array of byte;
2. Convert the byte array to string of hex digits.

```

// Static getInstance method is called with hashing SHA
MessageDigest msdg = MessageDigest.getInstance("SHA-256");

// digest() method calculate message digest of an input and return array of byte
return msdg.digest(input.getBytes(StandardCharsets.UTF_8));
}

public static String toHexString(byte[] hash)
{
    // Convert byte array into signum representation
    BigInteger number = new BigInteger(1, hash);

    // Convert message digest into hex value
    BigInteger bi = new BigInteger(1, hash);
    return String.format("%0" + (hash.length << 1) + "x", bi);
}

```

Fig. 6.29: SHA-256 HASH Code.

HMac

We implement the Hmac function as follows:

```

SecretKeySpec signingKey = new SecretKeySpec(KEY.getBytes("UTF-8"), "HmacSHA256");
Mac mac = Mac.getInstance("HmacSHA256");
mac.init(signingKey);
byte[] rawHmac = mac.doFinal(VALUE.getBytes("UTF-8"));

```

Fig. 6.30: HMac Code.

¹This MessageDigest class provides applications the functionality of a message digest algorithm, such as SHA-1 or SHA-256.

Database Connection

We use the following MySQLAccess.java code to connect the database over the JDBC driver for MySQL.

```
public class MySQLAccess {
    public static Connection createconnection(String host,String database,String user,String password) {
        Connection con=null;

        try{
            Class.forName("com.mysql.jdbc.Driver");

            con=DriverManager.getConnection(
                "jdbc:mysql://" +host+":3306/" +database,user,password);

        }catch(Exception e){
            System.out.println(e);
        }
        return con;
    }
}
```

Fig. 6.31: Conect to Database.

TOTP Function Implementation

We implement the TOTP function Figure 6.32 based on this formula:

$$TOTP = HOTP(PrivateKey, CurrentTime)$$

- PrivateKey is the shared key between user and server;
- CurrentTime is calculated by the following formula:

$$CurrentTime = floor(Unixepoch/30)$$

```
static private String totp(String secret, double currentTimeMillis) {
    String time=String.valueOf(currentTimeMillis);
    String hmacV=hmacSha(secret,time);
    return hmacV;
}
```

Fig. 6.32: TOTP Function.

```
currentTimeMillis =Math.floor(Instant.now().getEpochSecond()/30);
```

Fig. 6.33: CurrentTime unixEpoch.


```
public static int getDecimal(String hex){
String digits = "0123456789ABCDEF";
    hex = hex.toUpperCase();
    int val = 0;
    for (int i = 0; i < hex.length(); i++)
    {
        char c = hex.charAt(i);
        int d = digits.indexOf(c);
        val = 16*val + d;
    }

    if(val<0){
        return val*-1; }
    else{
        return val;
    }
}
```

Fig. 6.34: Getdecimal Function.

BIBLIOGRAPHY

- [1] Serge Vaudenay. A Classical Introduction to Cryptography, Applications for Communications Security. Acid-free Paper. The United States of America. 2006.
- [2] Bart Preneel. Analysis and Design of Cryptographic Hash Functions. 2003.
- [3] Richard Klima, Neil Sigmon. Cryptology Classical and Modern. Acid-free Paper. Second Edition. The United States of America. 2019 .
- [4] KEITH M. MARTIN. Everyday Cryptography. CPI Litho (UK) Ltd, Croydon, CR0 4YY. Second Edition. United States of America. 2017.
- [5] Hans Delfs, Helmut Knebl. Introduction to Cryptography Principles and Applications. Acid-free paper. Second Edition. 2007.
- [6] Alexander W. Dent, Chris J. Mitchell. User's Guide to Cryptography and Standards. ARTECH HOUSE. 2005.
- [7] Johannes A. Buchmann, Evangelos Karatsiolis, Alexander Wiesmaier. Introduction to Public Key Infrastructures. Acid-free paper. 2013
- [8] Jazib Frahim, Qiang Huang. SSL Remote Access VPNs. Cisco Press. United States of America. 2008.
- [9] A.Y. Iskhakov, R.V. Meshcheryakov, I.A. Hodashinsky. Choosing a method for generating one-time passwords and an information transport technology in the authentication system for ACS. Network Security and Communication Engineering. July 2015. pp.15-17.

-
- [10] Jazib Frahim, Omar Santos, Andrew Ossipov. Cisco ASA All-in-one Next-Generation Firewall, IPS, and VPN SERVICES, Cisco Press. Third Edition, USA, 2014.
- [11] Klaus Schmeh. Cryptography and Public Key Infrastructure on the Internet Acid-free paper. Britain. 2003.
- [12] Jean-Philippe Aumasson. SERIOUS CRYPTOGRAPHY, A Practical Introduction to Modern Encryption. No Starch Press. United States. 2018.
- [13] Stephen A. Thomas. SSL AND TLS Essentials, Securing the web. Acid-free paper. USA. 2000.
- [14] Jazib Frahim, Qiang Huang. SSL Remote Access VPNs. Cisco press. USA. 2008.
- [15] Joan Daemen, Vincent Rijmen. The Design of Rijndael The Advanced Encryption Standard (AES). Second Edition. Germany. 2002.
- [16] D. M'Raihi, Salah Machani, Mingliang Pei, Johan Rydell. TOTP: Time-Based One-Time Password Algorithm. Request for Comments, 6238, May 2011.
- [17] David M'Raihi, Mihir Bellare, David Naccache, Frank Hoornaert, Ohad Ranen . HOTP: An HMAC-Based One-Time Password Algorithm. Request for Comments, 4226, 2005.
- [18] Rolf Oppliger. SSL and TLS, Theory and Practice. United States of America. 2009.
- [19] P. Leach, M. Mealling, R. Salz. A Universally Unique Identifier (UUID) URN Namespace. RFC, 4122. 2005.
- [20] Chris Hare, Karanjit Siyan. Internet Firewalls and Network Security. Second Edition. United States of America. 1996.
- [21] Behrouz A. Forouzan. INTRODUCTION TO CRYPTOGRAPHY AND NETWORK SECURITY. Acid-free paper. 2008.
- [22] Neal Krawetz. INTRODUCTION TO NETWORK SECURITY. Acid-free paper. First Edition. United States of America. 2007.
- [23] Jie Wang, Zachary A. Kissel. INTRODUCTION TO NETWORK SECURITY. First Edition, USA. 2015

-
- [24] Ryan Glabb, Laurent Imbert, Graham Jullien, Arnaud Tisserand, Nicolas Veyrat-Charvillon. Multi-mode operator for SHA-2 hash functions. *Journal of Systems Architecture*, 53, 2007. pp.127–138.
- [25] Asoke Nath, Tanushree Mondal. Issues and Challenges in Two Factor Authentication Algorithms. Vol 6. 31 January 2016. pp. 318-327.
- [26] William Stallings. *Cryptography and Network Security, Principles and Practice*. Courier Westford. Sixth Edition. USA. 2014.
- [27] Bhuvan Unhelkar. *Software Engineering with UML*. Acid-free Paper. First Edition. USA. 2018.