

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHEMATIQUE ET
INFORMATIQUE
DEPARTEMENT INFOFRMATIQUE
N° :



DOMAINE : Mathématique et Informatique
FILIERE : INFORMATIQUE
OPTION : RESEAUX

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par : Sarir Ibrahim Anis

Intitulé

**PROTOCOLE D'AUTHENTIFICATION BASEE
SUR CODE QR**

Soutenu devant le jury composé de :

Mlle Saoudi Lalia

Université de M'sila

Président

Dr Chikouche Noureddine

Université de M'sila

Rapporteur

M Mezrag Fares

Université de M'sila

Examineur

Année universitaire : 2016 /2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHEMATIQUE ET
INFORMATIQUE
DEPARTEMENT INFOFRMATIQUE
N° :



DOMAINE : Mathématique et Informatique
FILIERE : INFORMATIQUE
OPTION : RESEAUX

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par : Sarir Ibrahim Anis

Intitulé

**PROTOCOLE D'AUTHENTIFICATION BASEE
SUR CODE QR**

Soutenu devant le jury composé de :

Mlle Saoudi Lalia

Université de M'sila

Président

Dr Chikouche Noureddine

Université de M'sila

Rapporteur

M Mezrag Fares

Université de M'sila

Examineur

Année universitaire : 2016 /2017

Table des matières

INTRODUCTION GÉNÉRALE	1
CHAPITRE 1 Codes QR	3
1.1. Définition.....	3
1.2. Structure d'un Code QR	4
1.3. Utilisation Code QR	6
1.3.1. Téléphone.....	6
1.3.2. Application.....	6
1.4. Génération un Code QR	6
1.5. Lecture un Code QR.....	6
CHAPITRE 2 SECURITE INFORMATIQUE.....	8
2.1. Introduction	8
2.2. Les primitive de la sécurité informatique.....	8
2.3. Les Attaques	8
2.3.1. Définition une attaque.....	8
2.3.1. Attaque de l'homme du milieu	8
2.3.1. Attaque par rejeu	9
2.3.1. Écoute clandestine.....	9
2.4. La Cryptographie.....	9
2.5. Chiffrement symétrique.....	10
2.5.1. Les algorithmes chiffrement Symétrique.....	10
2.5.2. Avantages.....	11
2.5.3. Inconvénients	11
2.6. Chiffrement asymétrique	11
2.6.1. Cryptographie sur les courbes elliptiques(ECC).....	12
2.6.2. Avantages et inconvénients.....	14
2.7. Fonction de hachage	14
2.8. Signature numérique.....	15
2.8.1. La signature ECDSA.....	15

2.9.	Authentification.....	16
2.10.	Protocole d'authentification.....	17
2.10.1.	Authentification Forte	17
2.10.2.	Authentification Mutuelle	17
2.10.3.	Authentification Unidirectionnelle.....	18
Chapitre 3	Travaux existants	20
3.1.	Introduction	20
3.1.1.	Notation.....	20
3.2.	Travaux existant	21
3.2.1.	Protocole Huang et al. [21]	21
3.2.2.	Protocole Liu et Zhu [22].....	26
3.3.	Discussion	29
Chapitre 4	Conception et implémentation d'un protocole d'authentification basée sur Code QR.....	30
4.1.	Protocole Authentification Mutuelle basée Sur Code QR.....	30
4.1.1.	Modèle d'attaquant :	30
4.1.2.	Architecture de Protocole.....	31
4.1.3.	Description du protocole.....	32
4.2.	Implémentation.....	35
4.2.1.	Application.....	36
4.3.	Analyse de performance	39
4.4.	Analyse de sécurité.....	40
CONCLUSION GENERALE		42
BIBLIOGRAPHIE		43

Liste des figures

Figure 1.1 Exemple QR code simple contient le texte « Hello World ».....	3
Figure 1.2 Structure QR code.....	5
Figure 2.1 Chiffrement Symétrique	10
Figure 2.2 Chiffrement Asymétrique	12
Figure 2.3 Fonction de hachage	15
Figure 2.4 Authentification Mutuelle.....	18
Figure 2.5 Authentification Unidirectionnelle	18
Figure 3.1 Phase d'inscription.....	22
Figure 3.2 Phase vérification.....	23
Figure 3.3 Phase de login et authentification	25
Figure 3.4 Phase d'inscription.....	26
Figure 3.5 Phase Login	27
Figure 3.6 Phase Authentification.....	28
Figure 4.1 schéma de Protocole Authentification	31
Figure 4.2 schéma Phase d'inscription.....	32
Figure 4.3 schéma Login et Authentification.....	34
Figure 4.4 Entre de l'application	36
Figure 4.5 Application Phase Inscription.....	37
Figure 4.6 Application Phase Login et Authentification	38
Figure 4.7 Application Phase Authentification.....	39

INTRODUCTION GÉNÉRALE

Les facteurs traditionnels d'authentification des utilisateurs peuvent être divisés en trois catégories : ce que vous connaissez par exemple, le nom de l'utilisateur et le mot de passe, comme les cartes à puce et les informations sur l'empreinte digitale. L'authentification par mot de passe, appelée «authentification simple», est l'un des types d'authentification les plus utilisés et acceptés.

Il présente des avantages naturels et pratiques, mais ses inconvénients sont également évidents : les utilisateurs ont tendance à utiliser des mots de passe simples et faciles à deviner et réutilisent souvent les mots de passe sur différents sites. Par conséquent, certains attaquants ont des chances de voler le mot de passe de l'utilisateur en lançant des attaques de supposition hors ligne et des attaques de « phishing ». En attendant, le serveur stocke beaucoup d'informations sous forme de texte clair, ce qui peut être facilement volé par les attaquants.

En conséquence, les informations personnelles de l'utilisateur seront claires. Beaucoup de chercheurs ont tendance à se concentrer sur l'authentification à deux facteurs, qui hérite non seulement des avantages de l'authentification par mot de passe, mais aussi d'améliorer la sécurité de ce type de système d'authentification. Au cours des dernières années, l'authentification à deux facteurs, en particulier le système d'authentification basé sur la carte à puce, en raison de son faible calcul et de sa haute sécurité, a attiré de plus en plus d'attention des chercheurs [19] [20] [21], Mais elle présente également des problèmes inévitables, tels que la convivialité des utilisateurs. Même si les cartes à puce sont petites et faciles à transporter, de nombreux utilisateurs ne sont toujours pas habitués à transporter un périphérique supplémentaire lorsqu'ils doivent se connecter au système.

Avec le développement de téléphones intelligents, leurs fonctionnalités et leurs applications deviennent plus parfaites. Dans une certaine mesure, la performance de certains téléphones intelligents pourrait rivaliser avec de petits ordinateurs. Les téléphones intelligents sont devenus une nécessité de la vie quotidienne des gens. Donc, nous proposons une nouvelle forme d'authentification à deux facteurs, mot de passe et téléphone, conformément à cette tendance. C'est une solution parfaite pour répondre aux considérations pratiques de l'utilisateur, tandis que la

quantité de calcul et de stockage des téléphones intelligents est beaucoup plus grande que la carte à puce. Nous adoptons une variété de techniques générales actuelles et nous proposons un protocole d'authentification à trois. Il y a quatre acteurs impliqués dans le processus d'authentification : utilisateur, serveur distant, ordinateur actuel et téléphone intelligent. Le protocole amélioré utilise le type d'authentification à deux facteurs : mot de passe et téléphone. L'utilisateur doit d'abord s'inscrire au serveur distant par son nom, l'adresse mac téléphone et son adresse e-mail. Ensuite, le téléphone scanne QR-Code qui contient les informations de jeton générées par le serveur et stocke le jeton dans le téléphone.

Organisation du mémoire

Ce mémoire est organisé de la manière suivante :

Chapitre 1 :

Dans ce chapitre nous présentons Code QR sa structure et ces utilisations (Lecture et génération Code QR).

Chapitre 2 :

Dans ce chapitre nous présentons les primitives de la sécurité informatique quelque attaques et en définie des algorithmes cryptographie symétrique et asymétrique ainsi que signature numérique et authentification.

Chapitre 3 :

Dans ce chapitre nous présentons des protocoles d'authentification existants.

Chapitre 4 :

Dans le dernier chapitre nous réalisons un protocole amélioré ainsi que on discute sur l'analyse de performance et l'analyse de sécurité de protocole.

CHAPITRE 1

QR CODE

CHAPITRE 1 Codes QR

1.1. Définition

Quick Response est une matrice spéciale de code à barres ou code bidimensionnel qui a été inventé par Denso Wave au Japon en 1994 [5]. Lisible par des lecteurs de codes à barres QR appropriés ou par une caméra de téléphones, Le code incluait jusqu'à 7 089 caractères numériques 4296 caractères alphanumériques (ASCII) 2953 octets (binaires 8-bit) et 1817 kanji (caractères japonais). [4] la figure suivant montre un exemple de QR Code «Hello World »



Figure 1.1 Exemple QR code simple contient le texte « Hello World »

Les codes QR sont similaires aux codes à barres standard qui apparaissent actuellement sur tous les types de produits de consommation, sauf que les codes QR peuvent contenir beaucoup plus d'informations que les codes à barres traditionnels. Vous pouvez penser à un code QR comme un lien hypertexte sur papier qui relie le monde physique avec le monde en ligne. Pour les spécialistes du marketing, les codes QR permettent à des publicités, des brochures, des affiches - même des vêtements ou des panneaux d'affichage de diriger les utilisateurs vers des pages d'atterrissage mobiles qui contiennent beaucoup plus d'informations et d'interactivité que celles qui peuvent être fournies sur la page imprimée. Cette intégration entre impression et web via mobile ajoute une nouvelle dimension de communication à tout effort de marketing ou de diffusion. [4]

1.2. Structure d'un Code QR

Un module est un bloc de code QR noir ou blanc contenant un ou plusieurs pixels. La taille du code QR est déterminée par le nombre de pixels composant un module et la capacité de données du code QR est déterminée par le nombre de modules contenus dans le code QR. Le niveau de capacité de correction d'erreur (ECC) est une caractéristique du code QR qui indique combien de modules peuvent être sales ou endommagés tout en permettant la restauration des données. Il existe quatre niveaux ECC différents, L, M, Q et H. [5]

Niveau	Capacité de correction d'erreur
L	7%
M	15%
Q	25%
H	30%

Table 1 Pourcentage de capacité de correction d'erreur

La structure d'un code QR est composée par des fonctions spéciales :

- Les patterns de détection (Finder pattern) : Trois blocs de taille fixe de taille 7×7 modules sont toujours disposés en haut à gauche, en haut à droite et en bas à gauche dans n'importe quelle version. Cette disposition permet au scanner d'orienter correctement le code QR pour le décodage. [1]
- Les séparateurs : permettent de séparer les patterns de détection de position du reste du code. [1]
- Les patterns d'alignement (Alignment pattern) : sont composés de 3 carrés concentriques superposés et de taille 5×5 modules pour le premier noir, 3×3 modules pour le blanc et un seul module central noir. Le nombre de ces patterns d'alignement dépend de la version de QR Code généré. [1]
- Patterns de synchronisation (timing Patterns) : permettent de densifier le QR Code par l'alternance de modules blancs/noirs de déterminer la version du QR Code et de fournir des positions de référence pour le calcul des coordonnées de chaque module. [1]
- Module sombre (Dark module) : Chacune des versions QR a un seul module sombre près de la partie inférieure gauche du modèle de recherche. L'emplacement exact du

module sombre est $((4 \times \text{Version}) + 9) \bmod 8$, où le module de code QR en haut à gauche est (0, 0). [1]

- Réservé la zone d'information de format (Reserved the Format Information Area) : Des bandes de modules existent à côté des séparateurs, sauf sur le côté gauche du modèle du viseur supérieur droit et sur le côté supérieur du motif du viseur inférieur gauche. Ces bandes enregistrent des informations concernant la tolérance aux pannes, le format de données stockées et le masque de données. [1]
- Réservé le domaine d'information de version (Reserved the Version Information Area) : Les versions de code QR 7 et supérieures contiennent deux zones pour stocker les informations de version. Les zones sont de deux blocs, l'un de taille 6×3 qui est placé sur le côté supérieur de la bas à gauche Finder pattern et un autre de la taille 3×6 qui est placé sur le côté gauche du top Finder pattern. [1]
- Données et zone ECC (Data and ECC Area) : Les zones restantes pour tous les modèles de fonction stockent les bits de données de correction générale et de correction d'erreur. Toutes les données sont codées en tant que valeurs de bits et sont dessinées sous forme de modules blancs ou noirs dans un ordre défini. [1]

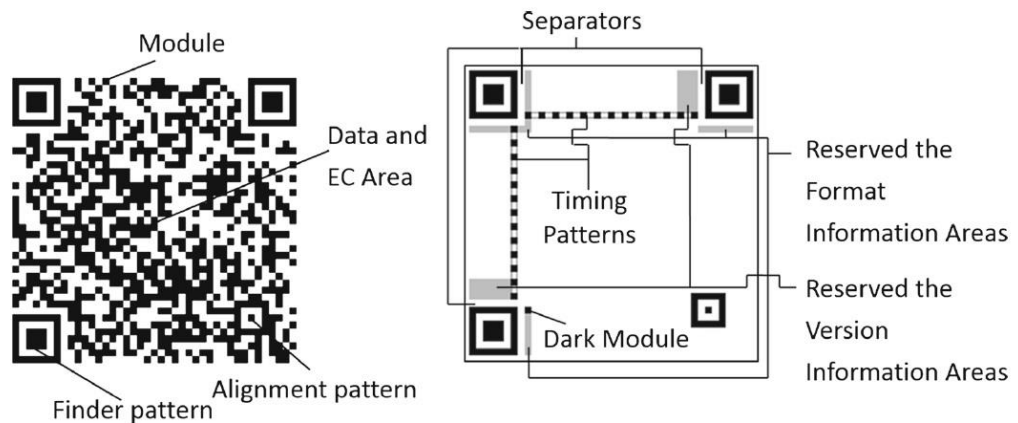


Figure 1.2 Structure QR code

1.3. Utilisation Code QR

Deux choses sont nécessaires pour réussir à décoder un code QR : un téléphone intelligent, une application de lecture de codes QR.

1.3.1. Téléphone

Afin d'utiliser les codes QR, vous devez posséder un téléphone cellulaire qui peut exploiter un logiciel de lecture. Ces téléphones peuvent télécharger et installer des applications, accéder à Internet et être assortis d'une caméra. De manière générale, on appelle ces types de téléphones « Smartphone ».

1.3.2. Application

Il y a un certain nombre d'applications que l'on peut utiliser en vue de décoder un code QR. Elles fonctionnent sensiblement toutes de la même façon. Par exemple « Scanner QR Android ».

1.4. Génération un Code QR

- Analyse des données à encoder et paramétrage du niveau de code correcteur.
- Convertir les caractères de données dans un flux de bits.
- Implémenter la correction des erreurs.
- Insérer les données avec le code correcteur dans la matrice.
- Générer la matrice et évaluer le résultat qu'elle retourne.
- Générer le QR Code au format image.

En peut l'utilisé quelque outils en ligne « <http://zxing.appspot.com/generator> » ou application dans les mobiles Android « QR Code Generator ». [7]

1.5. Lecture un Code QR

- Reconnaître le bit 1 ou 0
- Identifier le taux de code correcteur
- Identifier la version du QR Code
- Découvrir la région à décoder
- Lire les données et le code correcteur
- Détecter/Corriger les erreurs

Décoder les données [7]

CHAPITRE 2

SECURITE

INFORMATIQUE

CHAPITRE 2 SECURITE INFORMATIQUE

2.1. Introduction

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

2.2. Les primitive de la sécurité informatique

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

La sécurité informatique vise généralement cinq principaux objectifs :

- **La confidentialité** : consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées.
- **L'intégrité** : c'est-à-dire garantir que les données sont bien celles que l'on croit être.
- **L'authentification** : consistant à assurer que seules les personnes autorisées aient accès aux ressources.
- **La disponibilité** : permettant de maintenir le bon fonctionnement du système d'information.
- **Le non répudiation** : permettant de garantir qu'une transaction ne peut être niée.

2.3. Les Attaques

2.3.1. Définition une attaque

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables. [2]

2.3.1. Attaque de l'homme du milieu

L'attaque de l'homme du milieu est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. Le canal le plus courant est une connexion à Internet de l'internaute λ . L'attaquant doit d'abord être capable d'observer et d'intercepter les messages d'une victime à l'autre. L'attaque « homme du milieu » est particulièrement applicable dans la méthode d'échange de clés Diffie-Hellman, quand cet échange est utilisé sans authentification.

Avec authentification, Diffie-Hellman est en revanche invulnérable aux écoutes du canal, et est d'ailleurs conçu pour cela. [9]

2.3.1. Attaque par rejeu

Une attaque par rejeu est une forme d'attaque réseau dans laquelle une transmission est malicieusement répétée par un attaquant qui a intercepté la transmission. Il s'agit d'un type d'usurpation d'identité. [10]

2.3.1. Écoute clandestine

En général, la majorité des communications réseau se produisent dans un format non sécurisé ou "texte clair", ce qui permet à un attaquant qui a accédé aux chemins de données de votre réseau d'écouter ou d'interpréter le trafic. Lorsqu'un attaquant écoute vos communications, il est appelé reniflement. La capacité d'un escroc à surveiller le réseau est généralement le plus grand problème de sécurité auquel les administrateurs sont confrontés dans une entreprise. Sans les services de cryptage fort qui sont basés sur la cryptographie, vos données peuvent être lues par d'autres comme il traverse le réseau. [11]

2.4. La Cryptographie

Le mot Cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire les messages doit être plus sécurisé. Le verbe chiffrer est plus préférable que le verbe Crypter.

La cryptologie est essentiellement basée sur l'arithmétique. Il s'agit de transformer un texte de lettres en une succession de chiffres (dans le cas de l'informatique sous forme de bits car l'ordinateur est basé sur le binaire). Le message initial (**message clair**) après quelque modification il se rend incompréhensible et ce résultat s'appelle **le message chiffré**. Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement. La méthode inverse, consistant à retrouver le message original, est appelée déchiffrement. [2]

Le chiffrement se fait généralement à l'aide d'une clé de chiffrement, le déchiffrement nécessite quant à lui une clé de déchiffrement.

Les clés symétriques : il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.

Les clés asymétriques : il s'agit de clés utilisées dans le cas du chiffrement asymétrique (auss appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement. [2]

2.5. Chiffrement symétrique

Le chiffrement symétrique (aussi appelé chiffrement à clé privée ou chiffrement à clé secrète) consiste à utiliser la même clé pour le chiffrement et le déchiffrement. [2]



Figure 2.1 Chiffrement Symétrique

Le chiffrement consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles. Ainsi, le moindre algorithme (tel qu'un OU exclusif) peut rendre le système quasiment inviolable (la sécurité absolue n'existant pas).

2.5.1. Les algorithmes chiffrement Symétrique

Chiffrement Par bloc : Le chiffrement par bloc est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique, l'autre étant le chiffrement par flot. La principale différence vient du découpage des données en blocs de taille généralement fixe. La taille de bloc est comprise entre 32 et 512 bits, dans le milieu des années 1990 le standard était de 64 bits mais depuis 2000 et le concours AES le standard est de 128 bits. Les blocs sont ensuite chiffrés les uns après les autres. Il est possible de transformer un chiffrement de bloc en un chiffrement par flot en utilisant un mode d'opération comme ECB (chaque bloc chiffré indépendamment des autres) ou CFB (on chaîne le chiffrement en effectuant un XOR entre les résultats successifs). Les algorithmes les plus connus sont DES, AES. [5]

Chiffrement Par Flux : Le chiffrement de flux, chiffrement par flot ou chiffrement en continu est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique, l'autre étant le chiffrement par bloc. Un chiffrement par flot arrive à traiter les données

de longueur quelconque et n'a pas besoin de les découper. Le plus connu et utiliser RC4 utilisé notamment par le protocole WEP du WIFI. [5]

2.5.2. Avantages

- Le chiffrement/déchiffrement est très rapide, les algorithmes de chiffrement symétrique sont généralement beaucoup moins complexes que les algorithmes de chiffrement asymétrique
- Utilise peu de ressources systèmes, toujours dans le même principe d'algorithme moins complexe

2.5.3. Inconvénients

- Le chiffrement symétrique n'assure que la confidentialité des données, contrairement au chiffrement asymétrique que nous aborderons juste après et qui permet d'assurer des principes de sécurité supplémentaire.
- Une clé symétrique correspond à un échange entre 2 personnes, pour communiquer avec d'autres personnes il faudra une autre clé symétrique.
- Soit un grand nombre de clé selon le nombre de personnes avec qui on communique
- L'utilisation d'une clé unique présente un problème :
- Communiquer la clé de manière sûre à la personne avec laquelle on souhaite dialoguer.
Il est nécessaire de garantir la confidentialité de cette clé. Les échanges qui suivront reposent sur celle-ci. En d'autres termes, si une tierce personne accède à la clé, elle pourra lire modifier, altérer tous les échanges qui s'effectueront entre les 2 protagonistes de départ. [6]

2.6. Chiffrement asymétrique

Le principe de chiffrement asymétrique (appelé aussi chiffrement à clés publiques). Dans un cryptosystème asymétrique (ou cryptosystème à clés publiques) les clés existent par paires (le terme de bi-clés est généralement employé) : Une clé publique pour le chiffrement, Une clé secrète pour le déchiffrement.



Figure 2.2 Chiffrement Asymétrique

RSA : Le chiffrement RSA est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. RSA a été breveté par le « Massachusetts Institute of Technology » (MIT) en 1983 aux États-Unis. Le brevet a expiré le 21 septembre 2000. Il utilise une paire de clés (des nombres entiers) composée d'une clé publique pour chiffrer et d'une clé privée pour déchiffrer des données confidentielles. Les deux clés sont créées par une personne, souvent nommée par convention Alice, qui souhaite que lui soient envoyées des données confidentielles. Alice rend la clé publique accessible. Cette clé est utilisée par ses correspondants (Bob, etc.) pour chiffrer les données qui lui sont envoyées. La clé privée est quant à elle réservée à Alice, et lui permet de déchiffrer ces données. La clé privée peut aussi être utilisée par Alice pour signer une donnée qu'elle envoie, la clé publique permettant à n'importe lequel de ses correspondants de vérifier la signature.

2.6.1. Cryptographie sur les courbes elliptiques(ECC)

Dans la cryptographie à clé publique chaque utilisateur ou dispositif prendre part à la communication ont généralement une paire de clés, la clé publique et une clé privée, et un ensemble d'opérations associées. Avec les touches pour faire les opérations cryptographiques. Seul l'utilisateur particulier connaissant la clé privée, où en tant que la clé publique est distribué à tous les utilisateurs participant à la communication.

Chaque système cryptographique à clé publique nécessite un ensemble de prédéfini constantes à être connues de tous les dispositifs qui participent à la communication. Dans le cas de cryptographie à courbe elliptique "Paramètres de domaine" sont des constantes. Cryptographie à clé publique, contrairement à la cryptographie à clé privée, ne nécessite aucune partagé secret entre les parties communicantes, mais il est beaucoup plus lent que la cryptographie à clé privée.

Les paramètres de domaine de courbe elliptique sont : $T = (P, a, b, g, n, h)$.

Une courbe elliptique sur un corps K est une courbe définie par une équation de la forme $y^2 = x^3 + ax + b$ où $a, b \in K$ et $4a^3 + 27b^2 \neq 0$.

La sécurité de l'ECC dépend de la difficulté d'Elliptic Courbe logarithme discret problème. Soient P et Q deux points une courbe elliptique de telle sorte que $Q = kP$, où k est un scalaire. Compte tenu de P et Q , il est mathématiquement impossible d'obtenir k , si k est suffisamment large. K est le logarithme discret de Q à la base P .

ECC publique Crypto système : Dans les systèmes de chiffrement à clés publiques à courbe elliptique, supposons que entité A veut envoyer un message « m » à l'entité B en toute sécurité. Ordre d'un point de la courbe peut être définie comme une valeur n de telle sorte que, $nP = P + P + P + \dots n \text{ fois} = O$ (infini).

Génération de clés publique et privée : Les deux entités dans le système de chiffrement d'accord sur le domaine Les paramètres (a, b, P, G, N) . G est appelé point de générateur et n est l'ordre de G .

Maintenant A génère un nombre aléatoire $n_A < n$ que sa clé privée et calcule sa clé publique mis $PA = G.n_A$

B génère un nombre aléatoire $n_B < n$ que sa clé privée et calcule sa clé publique mis $PB = G.n_B$

Génération de clé commune : Après l'échange de la clé publique entre les deux parties Une entité calcule sa clé commune par Computing $K = n_A.PB$

Entité B calcule sa clé commune par Computing $K = n_B.PA$ Les deux touches ci-dessus ont la même valeur parce que: $n_A.PB = n_A.(n_B.G) = n_B(n_A.G) = n_B.PA$.

Chiffrement : Envisager un message « m » envoyé de A vers B . ' A ' choisit un aléatoire entier positif ' k ', une clé privée « n_A » et génère la clé publique $PA = n_A \times G$ et produit le texte chiffré ' cm ' constitué de pair de points $cm = \{kG, P_m + KPB\}$ où G est la point sélectionné sur la courbe elliptique de base, $PB = n_B.G$ est le public clé de B avec la clé privée n_B .

Déchiffrement : Pour déchiffrer le texte crypté, B multiplie le 1er point de la paire par B secrète et soustrait le résultat du 2ème point $H + KPB - n_B(kG) = P_m + k(n_B G) - n_B(kG) = P_m$. [15]

2.6.2. Avantages et inconvénients

- Résolvent le problème de la transmission des clés (clé de chiffrement publique → on peut envoyer un message chiffré à quelqu'un que l'on n'a jamais rencontré)
- Sont beaucoup plus lents que les algorithmes symétriques
- Sont vulnérables à une attaque « à texte clair connu » (la clé de chiffrement étant publique un pirate peut essayer de nombreux textes clairs et voir si le texte chiffré correspond)
- Sont vulnérables à une attaque « de l'intermédiaire » (un pirate peut substituer une clé publique par la sienne, intercepter et déchiffrer les messages, puis les chiffrer avec la correcte clé publique et les renvoyer). Attaque indécélable simplement → nécessité de la certification des clés publiques
- Il n'est pas prouvé que les problèmes mathématiques sous-jacents ne puissent être résolus par de meilleurs algorithmes
- Il n'est pas prouvé que la cryptanalyse est de même difficulté que les problèmes mathématiques sous-jacents (pour RSA il est juste prouvé que trouver d ou $(p-1)(q-1)$ est de même difficulté que la factorisation)
- Les progrès des algorithmes et de la technologie informatique imposent des clés de taille élevée (au moins 1024 bits)

Dans la pratique, les algorithmes à clé publique sont surtout utilisés pour chiffrer et transmettre une clé de session (le message est ensuite chiffré en utilisant un algorithme symétrique et cette clé de session) cryptosystèmes hybrides. [12]

2.7. Fonction de hachage

Une fonction de hachage est une fonction permet de hache un message (texte) c'est-à-dire une suite de caractères assez courte représentant le texte qu'il haché.

La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair D'autre part, il doit s'agir d'une fonction à sens unique (one-way function) afin qu'il soit impossible de retrouver le message original à partir du condensé. Le haché représente en quelque sorte l'empreinte digitale (finger print) du document.

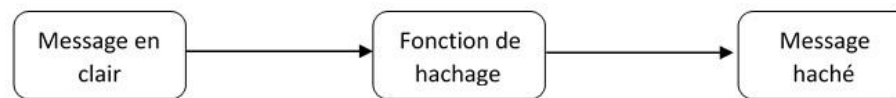


Figure 2.3 Fonction de hachage

- **MD5** : (MD signifiant Message Digest). Développé par Rivest en 1991, MD5 crée une empreinte digitale de 128 bits à partir d'un texte de taille arbitraire en le traitant par blocs de 512 bits. Il est courant de voir des documents en téléchargement sur Internet accompagnés d'un fichier MD5, il s'agit du condensé du document permettant de vérifier l'intégrité de ce dernier)
- **SHA** : (pour Secure Hash Algorithm, pouvant être traduit par Algorithme de hachage sécurisé) crée des empreintes d'une longueur de 160 bits
- **SHA-1** : est une version améliorée de SHA datant de 1994 et produisant une empreinte de 160 bits à partir d'un message d'une longueur maximale de 264 bits en le traitant par blocs de 512 bits. [2]

2.8. Signature numérique

Le paradigme de signature numérique (appelé aussi signature électronique) est une opération permettant de garantir l'authentification d'un utilisateur et vérifie l'intégrité de message reçu. La signature numérique assure la fonction de non répudiation c'est-à-dire d'assurer que l'émetteur a bien envoyé le message. [2] Les concepts de signature numérique sont principalement basés sur la cryptographie asymétrique. Cette technique permet de chiffrer avec un mot de passe et de déchiffrer avec un autre, les deux étant indépendants. [12]

2.8.1. La signature ECDSA

La signature numérique Elliptic Curve Digital Signature Algorithm (EC-DNA) est un algorithme de signature numérique à clé publique, variante de DSA. Il fait appel à la cryptographie sur les courbes elliptiques.

Signature algorithme

Pour signer message m de sender A , A utilise le private key d_A and public key $Q_A = d_A * G$

Calculé $e = \text{HASH}(m)$, en hache avec SHA-1

Choisir un nombre aléatoire k de $[1, n - 1]$

Calcule $r = x1 \pmod n$

Calcule $s = k - 1(e + dAr) \pmod n$.

La signature est le (r, s)

Signature Vérification :

Pour B Vérifier la Signature de A, B doit posséder le public key QA

Vérifier que r et s sont intégrés dans l'intervalle $[1, n - 1]$. Sinon La signature est invalide

Calcule $e = \text{HASH}(m)$

Calcule $w = s^{-1} \pmod n$

Calcule $u1 = ew \pmod n$ and $u2 = rw \pmod n$

Calcule $(x1, y1) = u1G + u2QA$

The signature est valide si $x1 = r \pmod n$, sinon invalide. [15]

2.9. Authentification

L'accès aux ressources d'un système d'information par une entité, se décompose en trois sous-processus, l'authentification, l'identification et le contrôle d'accès.

L'authentification pour un système informatique est un processus permettant au système de s'assurer de la légitimité de la demande d'accès faite par une entité (être humain ou un autre système...) afin d'autoriser l'accès de cette entité à des ressources du système conformément au paramétrage du contrôle d'accès. L'authentification permet donc, pour le système, de valider la légitimité de l'accès de l'entité, ensuite le système attribue à cette entité les données d'identité pour cette session (ces attributs sont détenus par le système ou peuvent être fournis par l'entité lors le processus d'authentification). C'est à partir des éléments issus de ces deux processus que l'accès aux ressources du système pourra être paramétré.

L'authentification désigne le processus visant à confirmer qu'un commettant est bien légitime pour accéder au système. Il existe quatre facteurs d'authentification classiques qui peuvent être utilisés dans le processus d'authentification d'un commettant :

Utiliser une information que seul le commettant connaît (ce que l'on connaît).

Utiliser une information unique que seul le commettant possède (ce que l'on possède).

Utiliser une information qui caractérise le commettant dans un contexte donné (ce que l'on est).

Utiliser une information que seul le commettant peut produire (ce que l'on sait faire). [11]

2.10. Protocole d'authentification

2.10.1. Authentification Forte

En sécurité des systèmes d'information, une procédure d'identification qui requiert la concaténation d'au moins deux facteurs d'authentification.

Les systèmes d'authentification courants utilisent un seul facteur (en général un mot de passe). Le principe de l'authentification forte est d'utiliser plusieurs facteurs de nature distincte afin de rendre la tâche plus compliquée à un éventuel attaquant. Les facteurs d'authentification sont classiquement présentés comme suit :

Ce que l'entité connaît (un mot de passe, un code NIP, une phrase secrète, etc.)

Ce que l'entité détient (une carte magnétique, RFID, une clé USB, un PDA, une carte à puce, un smartphone, etc.). Soit un élément physique appelé jeton d'authentification.

Ce que l'entité est, soit une personne physique (empreinte digitale, empreinte rétinienne, structure de la main, structure osseuse du visage ou tout autre élément biométrique)

Ce que l'entité sait faire ou fait, soit une personne physique (biométrie comportementale tel que signature manuscrite, reconnaissance de la voix, un type de calcul connu de lui seul, un comportement, etc.)

Où l'entité est, soit un endroit d'où, à la suite d'une identification et authentification réussie, elle est autorisée (accéder à un système logique d'un endroit prescrit). [11]

2.10.2. Authentification Mutuelle

Processus dans lesquelles deux parties, en général un client et un serveur, s'authentifient l'un l'autre. Cette authentification permet à chaque partie de connaître l'identité de l'autre. Dans le cadre d'une authentification mutuelle, le serveur demande également un certificat au client. Fonctionnalité également appelée authentification bidirectionnelle.

R1 : Clé Alice

R2 : Clé Bob

F() : Fonction

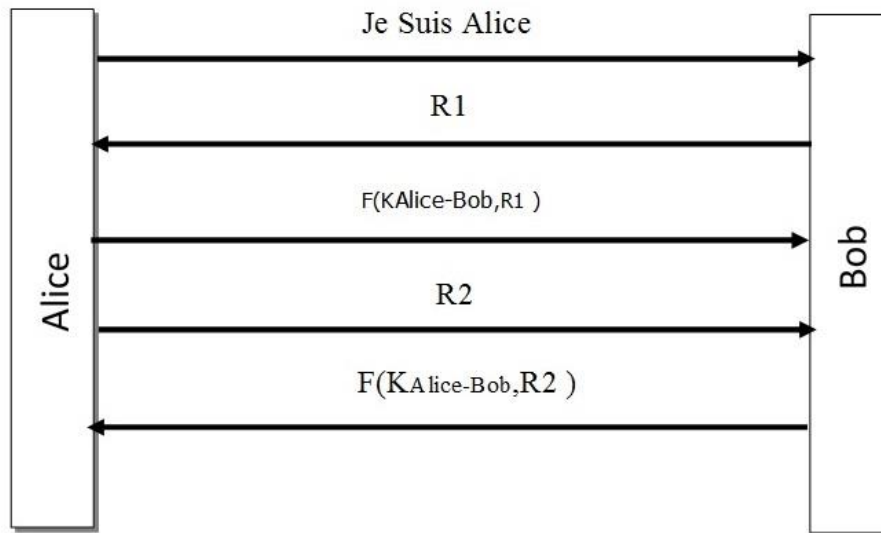


Figure 2.4 Authentication Mutuelle

2.10.3. Authentication Unidirectionnelle

La configuration des communications entre un serveur et un client SSL peut utiliser l'authentification SSL unidirectionnelle ou bidirectionnelle. L'authentification unidirectionnelle crée un fichier de clés certifiées sur le client et un fichier de clés sur le serveur. Dans cet exemple, le certificat CA "A" existe dans le fichier de clés certifiées sur le client SSL et dans le fichier de clés sur le serveur SSL.[16]

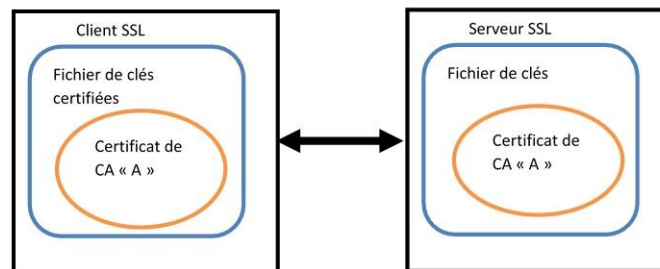


Figure 2.5 Authentication Unidirectionnelle

CHAPITRE 3

TRAVAUX EXISTANTS

Chapitre 3 Travaux existants

3.1. Introduction

En informatique, un protocole est un ensemble de règles qui déterminent comment les messages sont transmis dans les télécommunications et la mise en réseau informatique. Un protocole d'authentification vise à confirmer les identités de chaque côté en échangeant des données privées chiffrées. Un tel protocole est généralement appliqué lorsque les utilisateurs souhaitent accéder à des services Web...etc.

Un protocole d'authentification mutuelle est un type avancé de protocole d'authentification qui est appliqué quand un niveau de sécurité supplémentaire est requis. Alors on va développer et implémente un nouveau Protocole authentification qui se base sur Codes QR, Avant on va parler un petit peut sur quel que Protocole qui déjà développé.

3.1.1. Notation

U : utilisateur.

CT : Centre de Traitement.

CR : Centre d'enregistrement.

O : Ordinateur

S : Serveur.

ID : identification de l'utilisateur.

Blue_adds : adresse du périphérique Bluetooth.

Em_add : adresse e-mail.

K : serveur distant à clé à long terme.

Jeton : le serveur distant aux utilisateurs de jetons.

H(.) : fonction hash simple-sécurisée.

$E_{QR}(\cdot)$: algorithmes de cryptage QR Code.

$D_{QR}(\cdot)$: algorithmes de décryptage QR Code.

OPW : mot de passe généré par le serveur distant.

RAQE : Algorithme révisé du codage des codes.

TL : temps limité.

3.2. Travaux existant

3.2.1. Protocole Huang et al. [21]

Ils ont également utilisé la logique Gong-Needham-Yahalom avec plusieurs modèles d'attaque bien connus pour analyser la sécurité du protocole.

L'architecture du protocole il existe trois directeurs dans le protocole, l'utilisateur (U), Centre de traitement (CT) et le centre d'enregistrement (CR). L'ensemble du protocole se compose de trois phases. La première phase avant tout la phase de construction du système, servent à établir des relations entre CR, CT et U.

Après avoir terminé cette étape, toutes les CT et les entreprises concernées seront vérifiées par CR. La phase de connexion et d'authentification, la dernière phase du protocole, s'exécute dans une situation non vérifiée. Le but initial de cette phase était d'obtenir une authentification mutuelle entre CT et U par CR échangeant des informations d'authentification préenregistrées à CT et U, respectivement. Si CT et U ont acquis les informations d'authentification attendues, U estime que CT est vérifié et est autorisé à accéder au service de CT. [1]

- **Algorithme révisé du codage des codes QR (RAQE)**

Entrée : un message 1 M1, un message 2 M2

Sortie : un code QR révisé RQC

Etape 1 : Selon les longueurs de M1 et M2, déterminez la version du code QR et le niveau de corrélation des erreurs qui doivent être utilisés. Les capacités de corrélation normales et d'erreur des paramètres de code QR sélectionnés sont supérieures aux longueurs de M1 et M2, respectivement.

Etape 2 : Encodage M1 en tant que code QR QC en utilisant l'algorithme de code QR normal avec un paramètre présélectionné.

Etape 3 : Convertir M2 en une chaîne de bits sb.

Etape 4 : Générer un code QR révisé RQC en révisant les modules de code QR comme suit :

$$Modele = \{NOIR \text{ si } sb = 0, BLANC \text{ si } sb = 1\}$$

Les emplacements des modules révisés sont limités aux mots de code de correction des données et des erreurs et sont choisis pour ne pas être répétés par le générateur aléatoire R (s), où s est une graine aléatoire égale à H (M1).

Etape 5 : RQC de sortie. [21]

- **Phase d'inscription**

Etape 1 : U prépare le message public (MP) et le message secret (MS) comme information d'enregistrement, puis transmet $M = (MP, MS) + ksr$ à CR, désigne une fonction de cryptage asymétrique à l'aide de la clé publique CR + kpu et clé privée - Kpr. Les messages publics ont une faible demande de confidentialité, comme le nom d'utilisateur ou le numéro d'identification, Sinon, les messages secrets sont des données très sensibles telles que les numéros de carte de crédit ou la conviction ou les dossiers médicaux.

Etape 2 : CR acquiert (MP, MS) en décryptant M_i puis calcule $e_i = E_{Ski}(MS)$, où Ski est une touche aléatoire.

Etape 3 : RC choisit de manière aléatoire un bit string btKi pour coder en tant que code QR QRCXi, où la longueur de btKi est égale à la longueur de Ski.

Etape 4 : RC applique RAQE pour générer un QR code révisé en entrant MP et tki.

Étape 5 : RC calcule $rski = (Ski \oplus btKi)$ et applique RAQE à nouveau pour générer un QR code révisé RQCBi en saisissant ei et rski.

Etape 6 : passe RC (PIDi, QRCXi, RQCAi, q1, q2, ..., qn) + ksr à U où PIDi est une identité unique pour identifier un Ui et stocke QRCXi et RQCBi dans la base de données hôte.

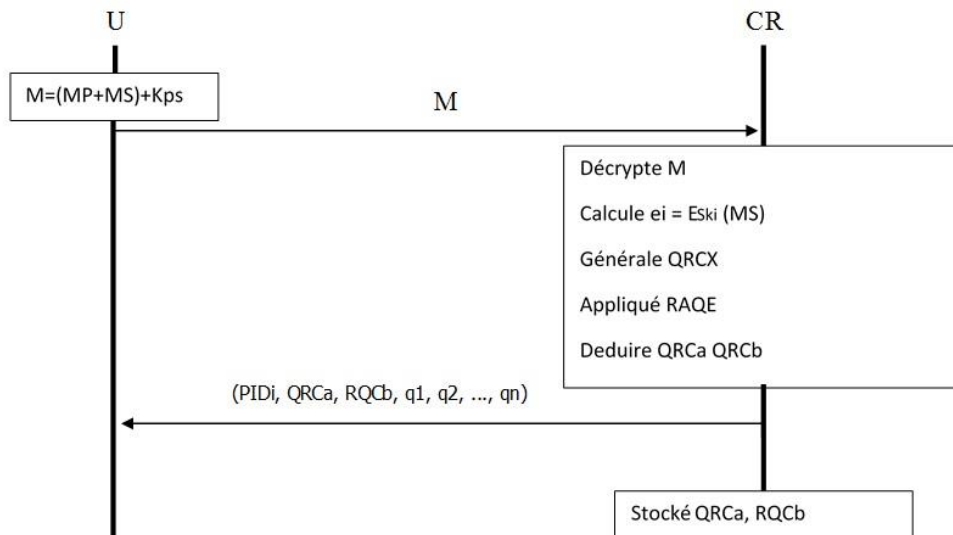


Figure 3.1 Phase d'inscription

- **Phase de vérification**

Etape 1 : CT obtient RQCai de Ui, lorsque Ui accède d'abord aux services CT.

Etape 2 : CT décode directement RQCai à l'aide d'un ordinateur ou d'un appareil mobile équipé d'appareils photo pour obtenir MPi.

Etape 3 : CT envoie un message (Request, qi) + ksr à RC pour obtenir RQCbi, où + kir est la clé publique, et la clé privée -kir est partagée entre CT et CR.

Etape 4 : CT applique RAQD pour extraire btKi en saisissant RQCai et pour extraire rski en entrant RQCbi. CT décrypte ensuite Ski en calculant $btKi \oplus rski$.

Etape 5 : CT obtient ei de RQCbi à l'aide d'un ordinateur ou d'un appareil mobile équipé de caméras, puis calcule $MSi = DSki(ei)$, où $DSki(.)$ Représente une fonction de décryptage symétrique à l'aide de la clé Ski.

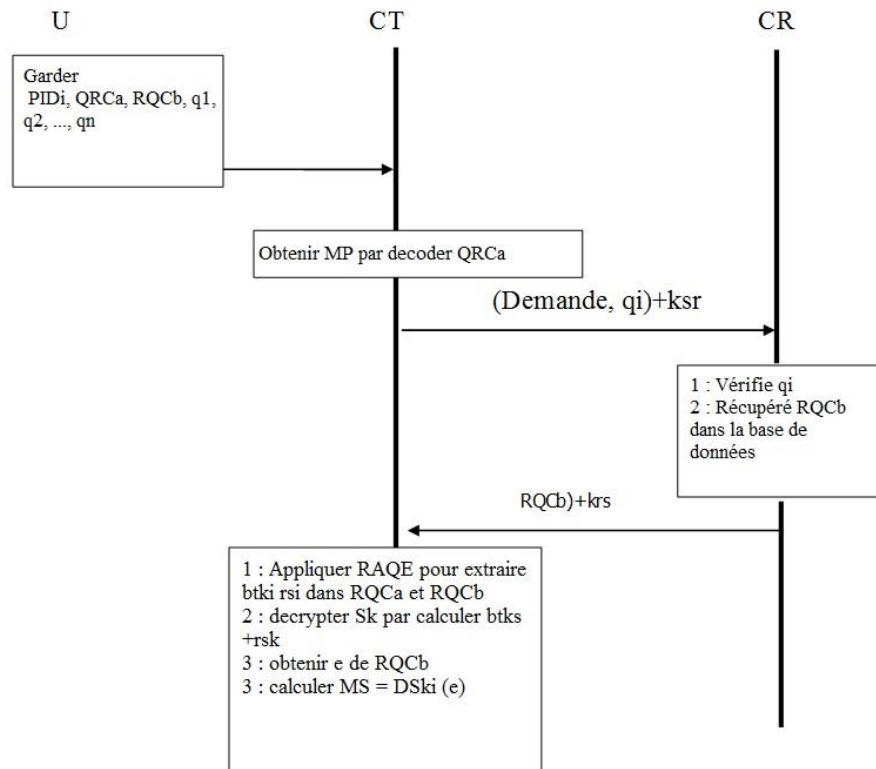


Figure 3.2 Phase vérification

- **Phase de Login et d'authentification**

Etape 1 : U_i demande à CT de se connecter en envoyant PID_i et $MU = (PID_i, Ts, L, q_i) + K_{sr}$.

Etape 2 CT reçoit PID_i et MSU , puis envoie $MCT = (MU, q_i, NCT) + K_{ir}$ à CR, après la vérification de PID_i , où NCT est un non pour CT.

Etape 3 : décrypte RC MCT et MU pour vérifier q_i et PID_i . Si détient, récupère QRC_{ai} et RQC_{bi} à partir de la base de données fournie par U_i dans la phase d'enregistrement.

Etape 4 : Le CR décode QRC_{ai} pour obtenir la valeur v_{xi} et génère un équivalent NCE non équivalent à v_{xi} en utilisant un générateur de mot de passe unique (OTP).

Etape 5 : CR calcule $p = (NCR + Ts + L)$ et modifie p pour être équivalent en longueur à v_{xi} en abandonnant ses bits les plus élevés. Ensuite, RC code pour $p_v = (p \oplus v_{xi})$ en tant que code QR $QRCY_i$.

Etape 6 : CR envoie $MCR = ((QRCY_i, H(p_v)) + K_{sr}, Ts, L, NRC, NIA) + K_{ir}$ à CT.

Etape 7 : CT reçoit le MCR de CR et envoie $MCT2 = (QRCY_i, H(p_v)) + K_{sr}$ à U_i .

Etape 8 : U_i reçoit $MIA2$ de CT. Ensuite, U_i décode QRC_{ai} pré-utilisé à l'aide d'un ordinateur ou d'un appareil mobile équipé d'appareils photo pour obtenir v_{xi} et récupère p en calculant $(p_v \oplus v_{xi})$. U_i envoie $MU2 = H(p + 1, q_i)$ à CT.

Etape 9 : CT vérifie U_i en calculant si $MU2$ est égal à $H(p + 1, q_i)$.

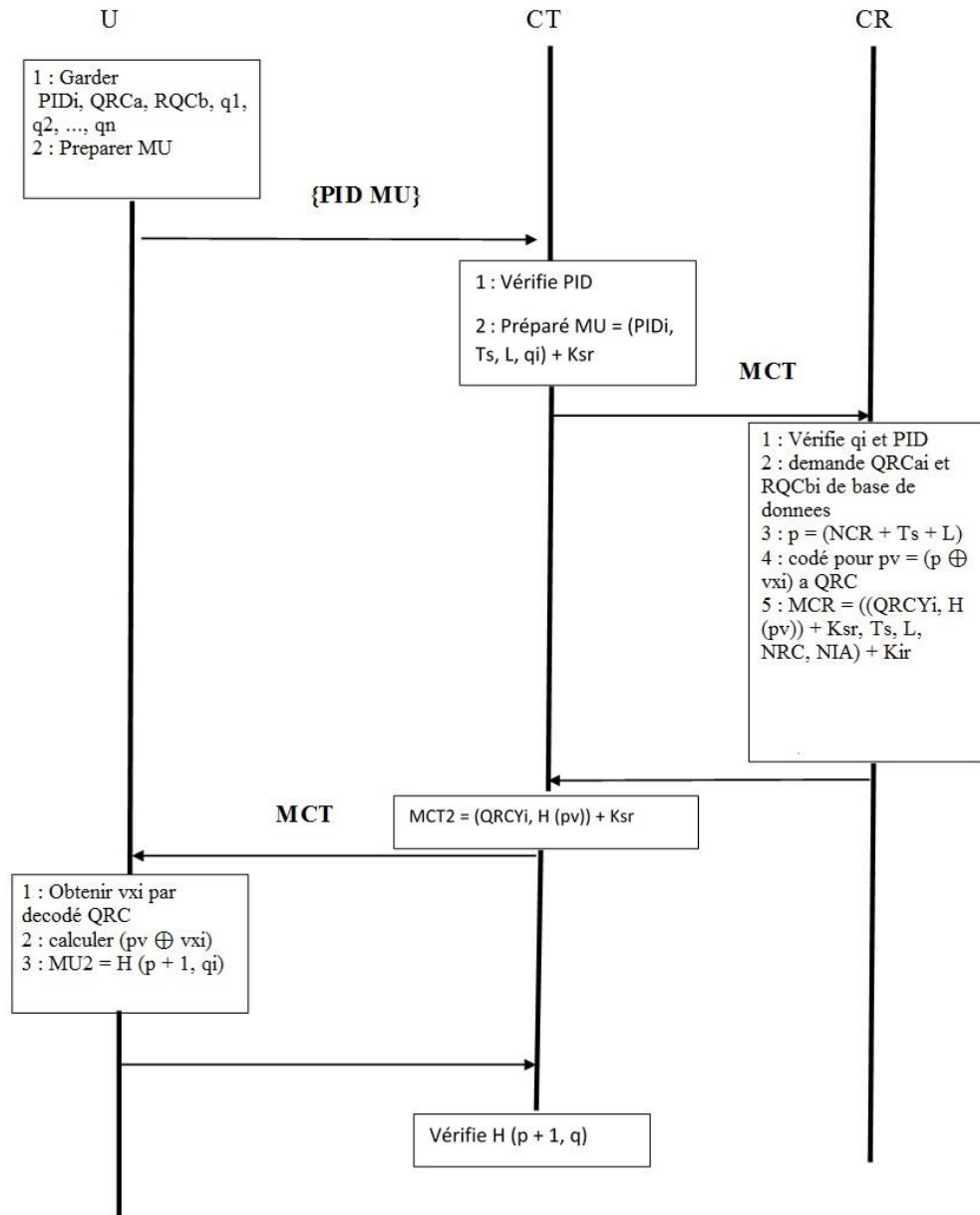


Figure 3.3 Phase de login et authentification

3.2.2. Protocole Liu et Zhu [22]

Le protocole authentification basée sur quatre facteurs l'ordinateur de l'utilisateur, mobile, Ordinateur et le Serveur. Les informations d'inscription dans le protocole, le nom de l'utilisateur et l'adresse et l'adresse du téléphone Bluetooth du téléphone. Cette adresse du Bluetooth elle a une grande importance dans le protocole.

Le protocole utilise également un code bidimensionnel très populaire et commercial, la technologie QR Code, en tant qu'OOB visuel, transmettant les jetons d'authentification distribués par le serveur d'enregistrement et certaines informations secrètes.

- **Phase d'inscription**

Etape 1 : U Soumet ID , Blue_adds et Em_add vers CR

Etape 2 : CR vérifie si ID Blue_adds et Em_add est existé. CR requis a U de saisie des nouveau information sinon CR $\text{Jeton}_u = H(\text{ID}_u || K || T) \quad E_{QR}(\text{Jeton}_u)$

Etape 3 : U calcule $D_{QR}(E_{QR}(\text{Jeton}_u)) = \text{Jeton}_u$ enregistrer avec sécurité.

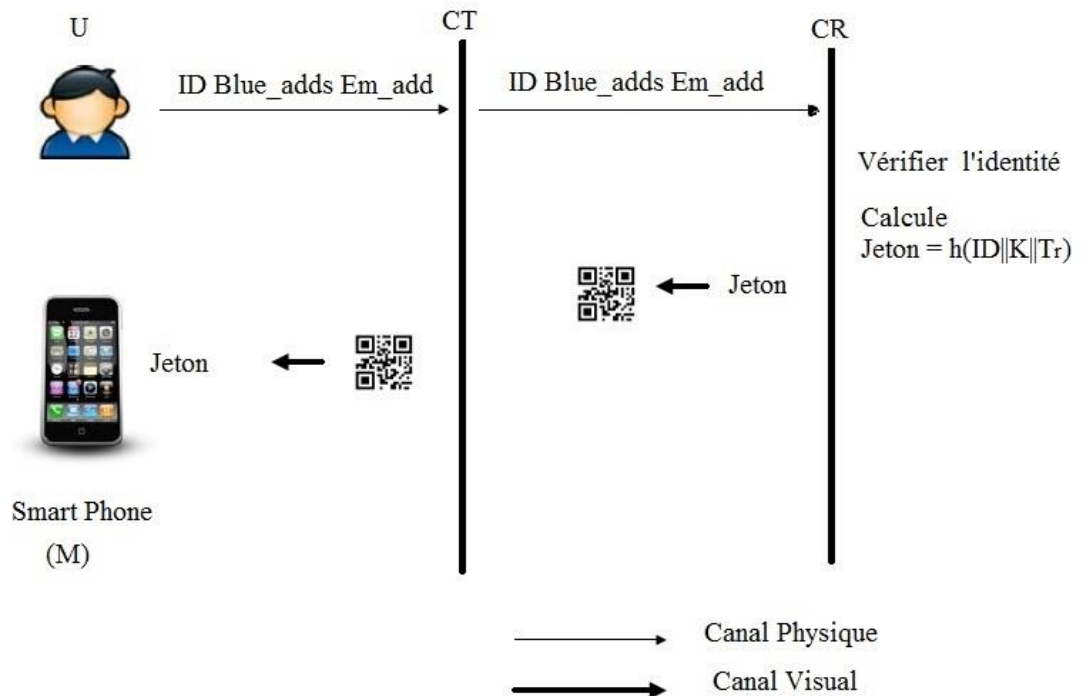


Figure 3.4 Phase d'inscription

- **Phase Login**

Etape 1 : U envoie une demande de connexion et des entrées ID_U , B_Adds et Em_add vers CT, Alors CT lance une connexion Bluetooth avec M d'après $Blue_adds$.

Etape2 : CR génère un mot de passe OPW et le transmet à CT, et CT le transmet au téléphone mobile de l'utilisateur M via Bluetooth, puis l'utilisateur obtient OPW.

Etape3 : U soumet $\{ID_U, H(OPW||T_1), T_1\}$ vers S

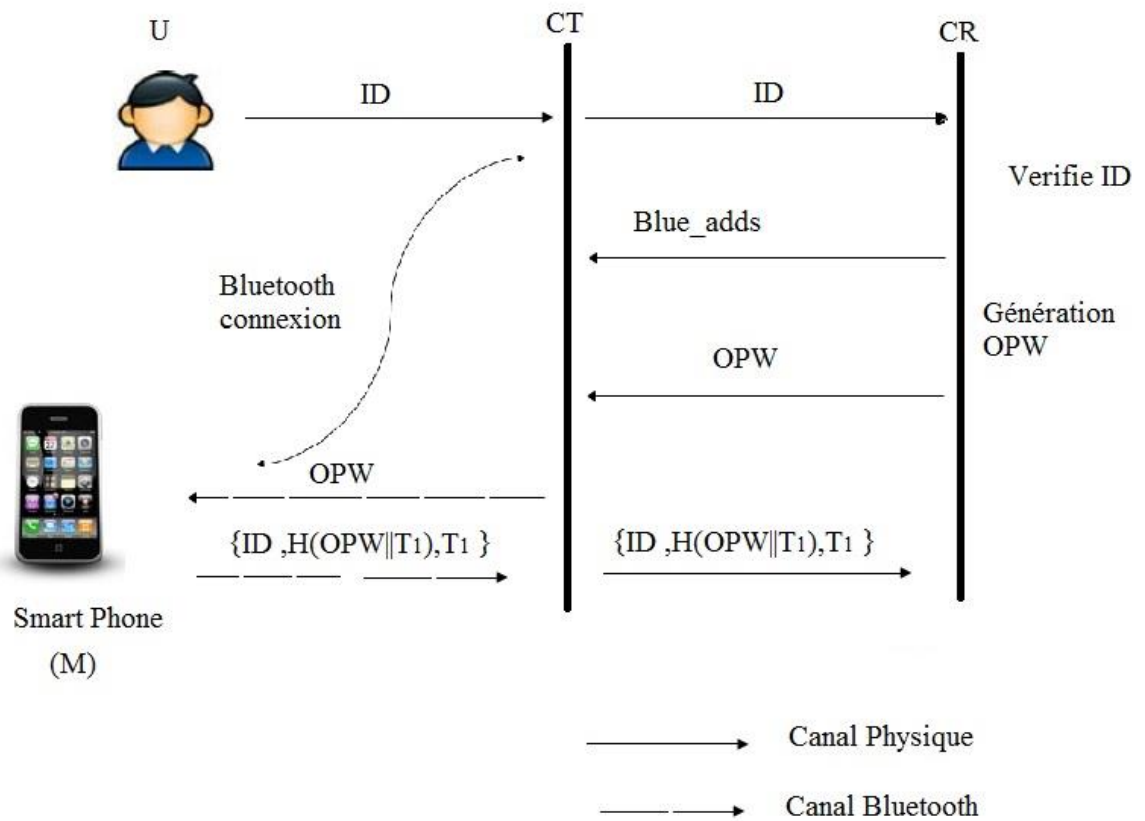


Figure 3.5 Phase Login

• Phase authentication

Etape1 : CR authentifie $\{ID_U, H(OPW||T_1), T_1\}$ Et génère un nombre aléatoire R, CR calcule $Jeton_u = H(ID_U || K || T)$ $A_u = E_{Jeton_u}(T_1 || T_2 || R)$, $B_u = H(T_1 || T_2 || R)$.

Etape 2 : CR transmet $E_{QR}(A_u)$ et B_u à U.

Etape 3 : Après avoir reçu les informations de CR, U calcule $D_{QR}(E_{QR}(A_u)) = A_u$ $D_{Jeton_u}(E_{Jeton_u}(T_1 || T_2 || R)) = T_1 || T_2 || R$, U vérifie $T_1 - T_2 < 2\Delta T$, T Est l'horodatage actuel de U.

Etape 4 : U calcule $C_u = E_{Jeton_u}(T_2 || T_3 || R)$ $D_u = h(T_2 || T_3 || R)$ Puis envoie le message $\{C_u, D_u\}$ via Bluetooth, finalement transférer vers CR.

Etape 5 : Lorsque vous recevez le message de U, CR calcule $D_{QR}(E_{QR}(C_u)) = C_u$, $D_{Jeton_u}(C_u) = D_{Jeton_u}(E_{Token_u}(T_2 || T_3 || R)) = T_2 || T_3 || R$.

Etape 6 : CR calcule $Jeton_{u1} = H(ID_U || K || T_r || T_{a1})$ $E_{QR}(Jeton_{u1})$

Etape 7 : U calcule $D_{QR}(E_{QR}(Jeton_{u1})) = Jeton_{u1}$ Et le stocke sur le téléphone au lieu de l'ancien.[]

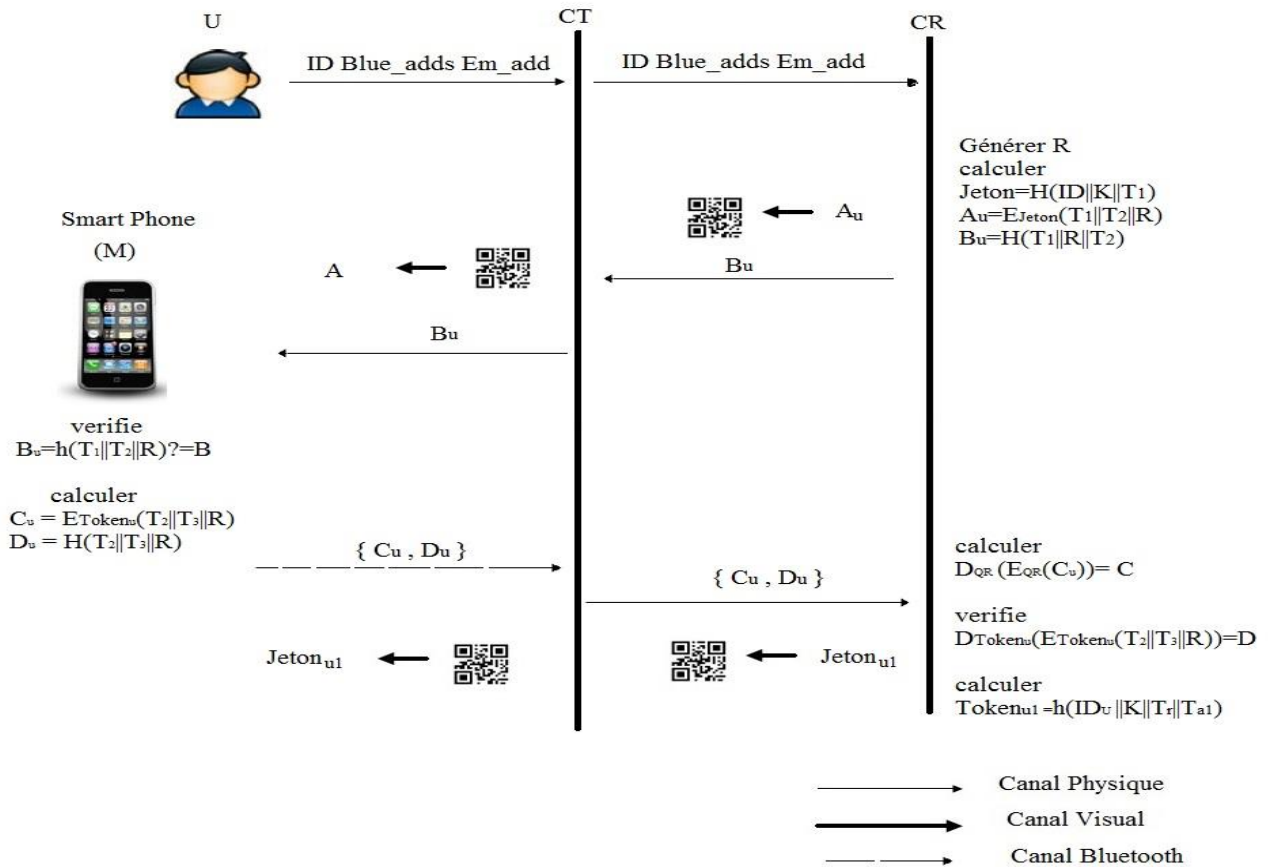


Figure 3.6 Phase Authentication

3.3. Discussion

Après l'étude de protocole Liu et Zhu on a trouvé quelque inconvénients dans la phase inscription et login. Premièrement après l'inscription et enregistrement les données dans le mobile le serveur n'utilisent jamais ces données dans le protocole et en plus utilise autre fonction comme la génération de mot de passe (OPW) pour authentifie l'utilisateur alors que facile d'utilise les données enregistrer mieux que le OPW. Ensuite la donnée transférer (ID) entre l'ordinateur et serveur ne sont pas protégé avec une fonction de chiffrement. Enfin l'utilisation fonction de chiffrement avec la signature numérique dans la phase d'authentification le protocole devient très lourd sans grand importance, il faut mieux qu'on utilise que la signature numérique pour réaliser l'authentification mutuelle parce que on a réalisé les deux premières phases inscription et login avec succès.

CHAPITRE 4

CONCEPTION ET IMPLEMENTATION D'UN PROTOCOLE D'AUTHENTIFICATION BASEE SUR CODE QR

Chapitre 4 Conception et implémentation d'un protocole d'authentification basée sur Code QR

4.1. Protocole Authentification Mutuelle basée Sur Code QR

Notre protocole est Protocole authentification Mutuelle. Nous devrions être prendre en compte l'ordinateur de l'utilisateur actuel parmi la phase d'authentification, Et considère l'ordinateur actuel comme un périphérique approuvé et un canal crédible dans notre protocole. Cela peut augmenter la sécurité. On a utilisé la signature numérique ECDSA et fonction de cryptage ECC (ECDH) qui on facile et rapide. Pour une plus souplesse de communication et de stockage on utilise les Codes QR car de plus en plus de gens commencent à se préoccuper d'un tel petit code bidimensionnel, car c'est vraiment pratique, les gens peuvent ajouter des amis, se connecter à un site Web et même compléter le paiement en le scannant. En ce qui concerne les réseaux sociaux et les objectifs commerciaux, il est déjà partout. On utilise système Android dans le système de l'utilisateur et le serveur car il est très fréquent simple et facile à l'utiliser dans les derniers années.

4.1.1. Modèle d'attaquant :

Les attaquants peuvent à première vue être classées en 2 grandes catégories :

- Les attaquants passifs : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.
- Les attaquants actifs : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

Les attaquants peuvent être classés non-seulement par leurs connaissances mais également suivant leurs capacités d'attaques dans une situation bien définie. Ainsi, on dénombrera les capacités suivantes :

Transmission de messages sans capacité d'écoute.

Écoute et transmission de messages.

Écoute et perturbation des communications.

Écoute perturbation et transmissions de messages.

Écoute et relai de messages.

4.1.2. Architecture de Protocole

Notre protocole est basé sur quatre parties l'ordinateur de l'utilisateur (mobile M), Utilisateur (U), Ordinateur (O), Serveur (S). Comme vous le voyez (Figure 4.1) des connexions entre les composants.

Entre Ordinateur et le serveur une connexion internet il faut que le serveur et ordinateur équipé avec processeur puissant pour exécuter des fonctions arithmétique, générer code QR et une mémoire de stockage car il y a une base de données dans le serveur.

Ordinateur et mobile équipé avec un écran d'affichage très bonne résolution pour afficher le code QR et une application de lecture Code QR sous le système Android.

Avec tout cette connexion il faut que nous sécurisé alors nous proposons un protocole qui va sécuriser la connexion entre eux.

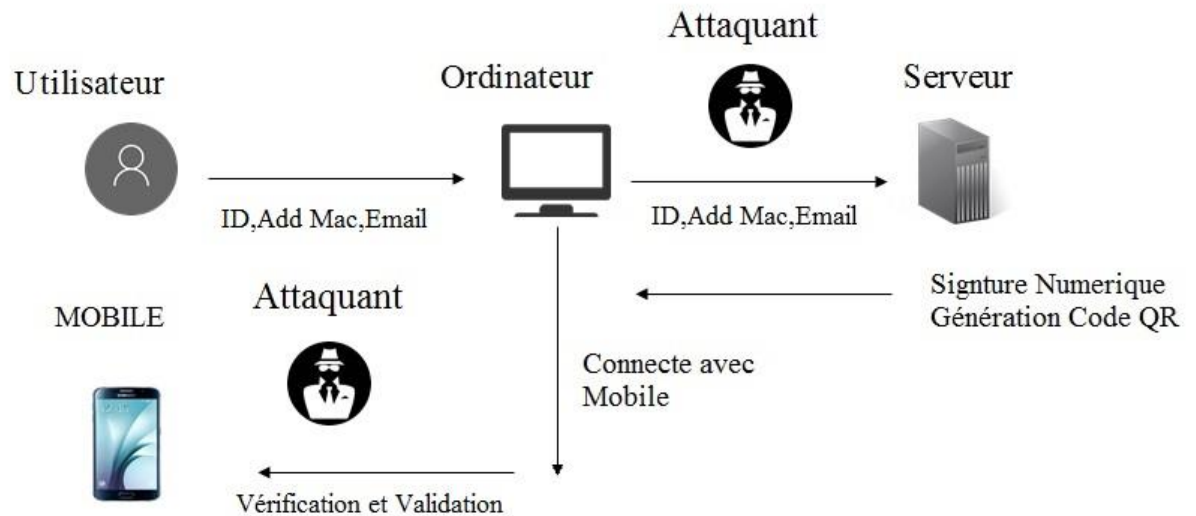


Figure 4.1 schéma de Protocole Authentification

4.1.3. Description du protocole

Notre protocole est basé sur trois phases, Phase Inscription cette phase utilisateur saisie et enregistre les données personnel dans la base de données du serveur, Phase Login cette phase relie l'utilisateur avec le serveur après une vérification et validation le code QR enregistré sur utilisateur, et la dernier phase, Phase Authentification cette phase authentifie l'utilisateur pour manipulé les différentes opérations avec le serveur.

- **Phase d'Inscription**

La description détaillée est comme le suivant (voir aussi la Figure 4.2) :

Etape 1 : U Soumet ID, Add_Mac et Em_add vers S

Etape 2 : S vérifie si ID Add_Mac et Em_add est existé. S requis a U de saisie des nouveau information sinon S $\text{Jeton}_u = \text{Sign}(\text{ID}_u \parallel \text{Add_Mac} \parallel \text{Em_add})$ $E_{QR}(\text{Jeton}_u)$

Etape 3 : U calcule $D_{QR}(E_{QR}(\text{Jeton}_u)) = \text{Jeton}_u$ enregistrer avec sécurité.

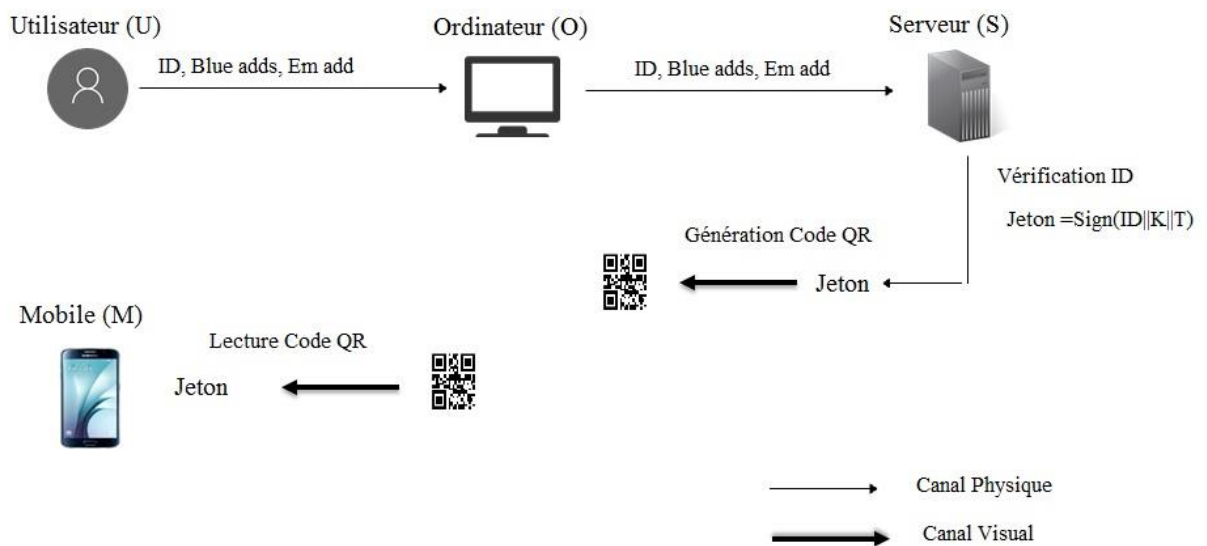


Figure 4.2 schéma Phase d'inscription

Résultat :

Serveur : Enregistrement les données de l'utilisateur ID, adresse mac et email dans la base de données, utilisent fonction hachage (SHA-512), signature numérique (ECDSA), génération code QR.

Mobile : Enregistre la signature numérique, par lecture code QR.

La description détaillée est comme le suivant (voir aussi la Figure 4.3) :

- **Phase De Login et Authentification**

Etape 1 : U envoie une demande de connexion et des entrées ID_u vers O, Alors O chiffre ID avec $ECIES(E_{ECIES}(ID))$ et envoi vers S, U génère code QR enregistré avant $E_{QR}(Jeton_u)$. S décrypte $ECIES(ID) \{D(E_{ECIES}(ID))\}$, S vérifie ID et lut $D_{QR}(E_{QR}(Jeton_u))$ pour valide si existe ou non .

Etape 2 : S soumet $S=Sign(ID||T_1)$, $Jeton\{S,T_1\}$ $E_{QR}(Jeton)$ vers U.

Etape 3 : U soumet $D(E_{QR}(Jeton))=Jeton\{S,T_1\}$, valide la signature $V(S)=S$.

Etape 4 : U $S'=Sign(ID||T_2)$, $Jeton \{S',T_2\}$, $E_{QR}(Jeton)$ vers S

Etape 5 : S soumet $D(E_{QR}(Jeton))=Jeton\{S',T_2\}$, valide la signature $V(S')=S$.

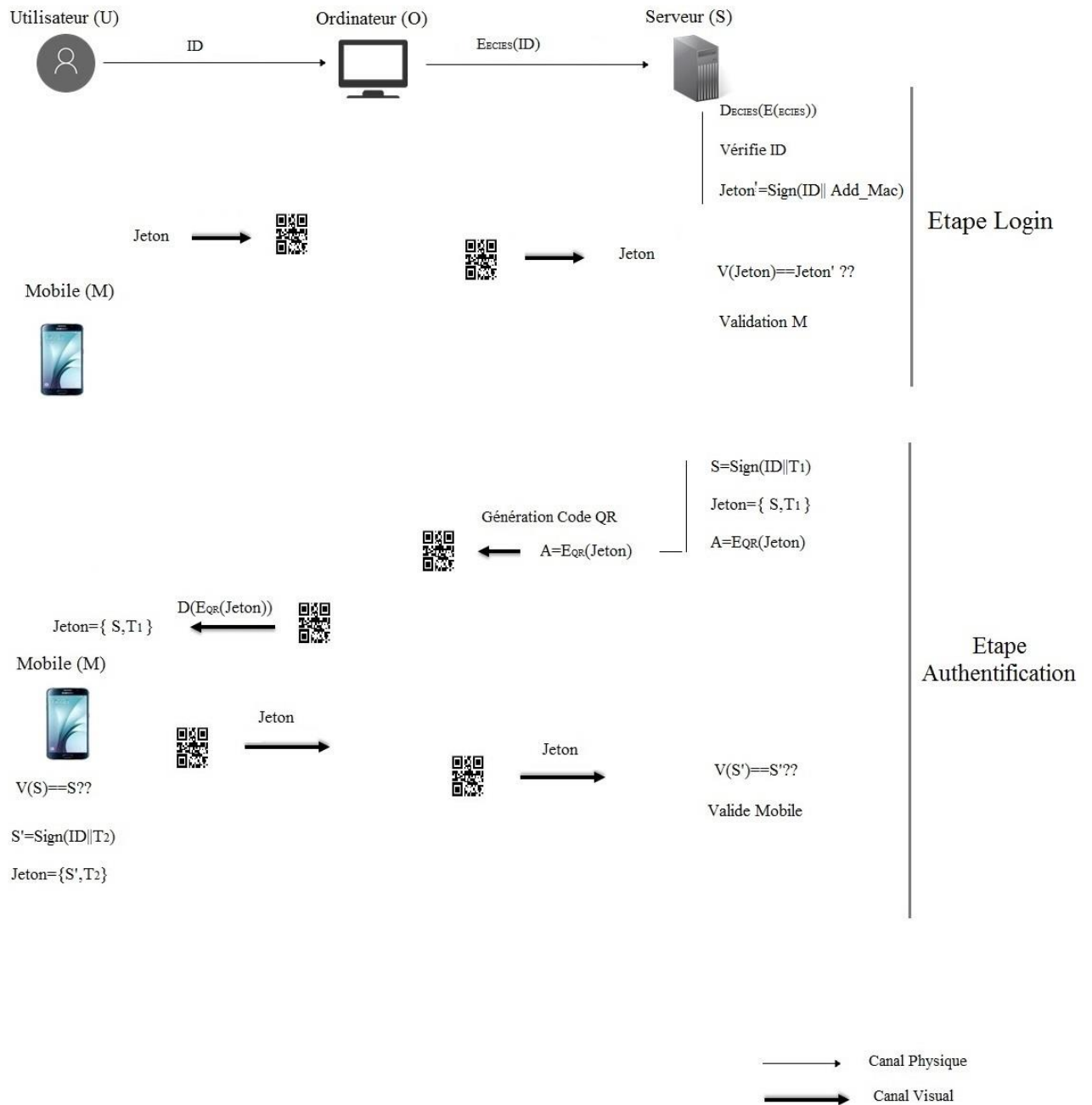


Figure 4.3 schéma Login et Authentification

Résultat :

Serveur : Validation le mobile après des connexions sécurisées avec chiffrement asymétrique (ECIES) et la signature numérique (ECDSA).

Serveur : Enregistre que le mobile est fiable pour change avec lui les données et les opérations

Ordinateur : transmission les Jeton.

Mobile : Prêt pour une connexion fiable et sécurisé avec le serveur.

4.2. Implémentation

Notre protocole d'authentification a était implémenté dans un système Android avec l'IDE Android Studio qui utilise langage Java on le présente Android Studio dans quelque ligne.

Android Studio : [17] Google à créer un IDE complet pour la création d'application mobile Android nommé Android Studio, annoncé lors du Google I/O le 15 mai 2013. Il est open source et disponible gratuitement, permettant de réaliser des projets sur différents types de support, tablette ou smartphone. Principalement utilisé pour éditer des fichiers Java étant le langage d'une application Android native ainsi que des fichiers de mise en page « XML » avec la possibilité de visualiser le rendu et les manipuler en utilisant une interface graphique. Installer JDK (Java Development Kit), désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé.

Propriété :

- Un système de construction souple à base de « Gradle ».
- Un émulateur rapide et riche en fonctionnalités.
- Un environnement où il est possible de développer pour tous les appareils Android.
- Instant « Run » pour pousser des modifications à votre application en cours d'exécution sans la construction d'une nouvelle APK.
- Modèles de code et l'intégration « GitHub » pour vous aider à construire des fonctionnalités d'applications communes et exemple de code d'importation.
- outils de tests approfondis et des cadres.
- outils de Lint pour attraper la performance, la facilité d'utilisation, la compatibilité des versions, et d'autres problèmes.
- « C++ » and « NDK » support.

- Support intégré pour Google Cloud Platform , le rendant facile à intégrer Google Cloud. [17]

Dans le protocole on a utilisé aussi la bibliothèque ZXING pour Lire et Générer les Codes QR.

ZXING : [18] Une bibliothèque prenant en charge le décodage et la génération de codes à barres (comme QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codabar) dans les images. Les codes à barres suivants sont pris en charge par le décodeur: UPC-A, UPC-E, EAN-8, EAN-13, Code 39 , Code 93, Code 128, ITF, Codabar, MSI, RSS-14 (toutes les variantes), QR Code, Data Matrix, Aztec et PDF-417.

Le codeur prend en charge les formats suivants : UPC-A, EAN-8, EAN-13, Code 39, Code 128, ITF, Codabar, Plessey, MSI, QR Code, PDF-417, Aztec.

Bibliothèque Cryptographique qui Nous Utilisons « KEYGENPARAMERSPEC » pour génération de clés et « BouncyCastle » [17] pour utiliser ECDSA pour la signature numérique et ECIES pour crypté et décrypté les données.

4.2.1. Application

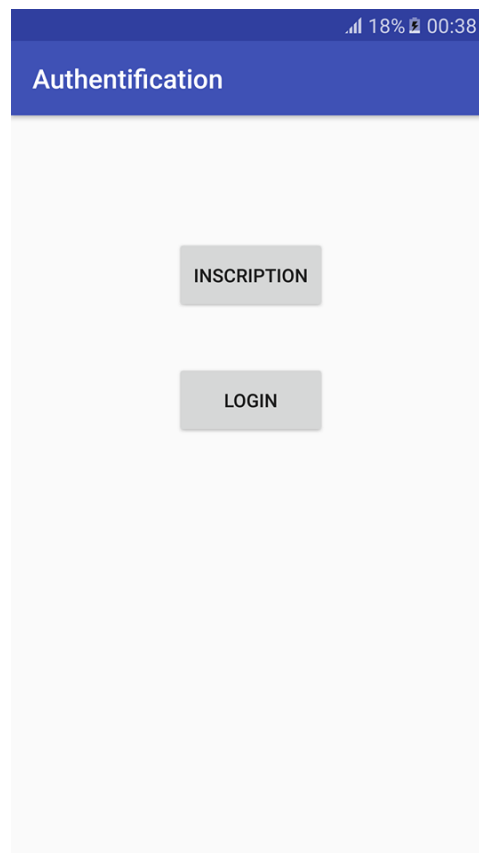


Figure 4.4 Entre de l'application

- **Phase Inscription**

Authentification

ID 1234

Adresse Mac 70-1A-04-5F-9B-3B

Email exemple@gmail.com

ENREGISTRER



Figure 4.5 Application Phase Inscription

- **Phase Login et Authentification**

Authentification

Entre le ID

1234

LOGIN

AUTHENTIFICATION

Figure 4.6 Application Phase Login et Authentification



Figure 4.7 Application Phase Authentification

4.3. Analyse de performance

Calcule : Tout au long du processus de Protocole, il ne s'agit que d'une arithmétique de fonction hash à sens unique et d'un code QR. Pour les opérations de cryptage et de décryptage, on utilise ECIES et ECDSA car est plus rapide et nécessite moins de puissance de calcul que les autres algorithmes de chiffrement à clé publique de première génération. Par exemple, une clé de

chiffrement ECIES 160 bits offre la même sécurité que la clé de chiffrement RSA 1024 bits et peut être jusqu'à 15 fois plus rapide.

Technologie : Notre schéma s'appuie sur la technologie de transmission des messages. Évite également la consommation de ressources d'énergie. Comparé à des schémas similaires, notre système est plus efficace et plus simple. Le schéma utilise également un mot de passe unique qui combiné avec la technologie de génération des nombres aléatoire, pour améliorer la sécurité d'authentification.

4.4. Analyse de sécurité

En supposant que l'environnement certifié soit relativement fermé, y compris les zones communes plus larges, comme les Banks les gares et les aéroports. Dans ces endroits, de nombreux facteurs interfèrent avec le processus d'authentification que nous ne pouvons pas contrôler.

Nous assure que les informations transmises sur le canal visuel, le programme ne peuvent pas être écoutés, bloqués et modifiés. Mais par contre le message transmis via Bluetooth peut être écouté par des attaquants, en d'autres termes, un attaquant pourrait répondre et modifier ces informations.

Authentification Mutuelle : Pour assurer l'authentification mutuelle dans la dernière phase on utilise la signature numérique (ECDSA) pour signé les messages changer entre le serveur et mobile.

Confidentialité : la phase login assure que le serveur et mobile sont les seuls personnes autorisées, après la validation de message signé par ECDSA qui est enregistré dans le mobile, le serveur et mobile ouvre une session de communication.

Intégrité des données : Pour assure que les données ne sont pas modifié dans la transmission on a utilisé fonction de chiffrement ECIES.

Déni de service : Dans notre protocole on utilise la technologie de « Horodatage », dans chaque fonction de signature en génère « $T_1, T_2, \dots$ » En vérifie que si $T_2 - T_1 > TL$ alors il y a une interruption un attaquant dans le réseau alors on coupe la connexion entre eux.

Attaque par replay : Supposons qu'un intrus essaie de reproduire une demande légale entre Mobile et Ordinateur ou entre Ordinateur et le Serveur. Le Horodatage T et OPW empêcheront un destinataire d'accepter cette demande.

Attaque Homme au milieu : Notre protocole peut réaliser une authentification mutuelle entre Serveur et Mobile, une attaque Homme au milieu n'est pas réalisable. L'Ordinateur joue un rôle très

important qui chiffre à l'aide de clés pré-partagées ECIES(ID) avant l'envoi vers Serveur. Le Serveur pour assure la communication véritable avec Mobile il génère Mot de passe.

Écoute clandestin : Le risque de sécurité d'écoute est très élevé pour tout protocole d'authentification. Pour éviter ce risque, notre protocole chiffre tous les messages échangés avec ECIES et Signe avec ECDSA.

Sécurité de l'information mobile : Dans notre schéma d'authentification, le téléphone mobile comme un jeton de supports de stockage à long terme joue un rôle crucial dans l'authentification, et l'utilisateur doit le stocker en toute sécurité. Cependant, une fois que le téléphone mobile est perdu ou volé par quelqu'un l'utilisateur peut changer le téléphone de liaison du serveur et l'utilisateur peut relayer un nouveau téléphone. Donc, même si les personnes qui reçoivent un téléphone ne peuvent pas se connecter via le téléphone mobile directement et ne peuvent pas accéder à la clé à long terme K du serveur depuis « Jeton », en raison de la sécurité de la fonction hash à sens unique, notre système pourrait donc garantir Sécurité de l'information.

CONCLUSION GENERALE

La technologie a pour vocation première de simplifier la vie de l'homme. A priori, elle est à son service. Pourtant, depuis la nuit des temps, l'homme entretient avec elle un rapport ambigu. Fasciné par les progrès et la modernité qu'elle symbolise, effrayé par ses possibles dérives.

Dans ce mémoire, nous avons présenté Code QR et c'est caractéristique Lecture Génération le Code bidimensionnelle. Ensuite nous avons présente la sécurité générale dans l'informatique comme les primitives de la sécurité, les attaques, les primitive de chiffrements, chiffrement symétrique et asymétrique ainsi que la signature numérique, fonction de hachage et la définition de l'authentification. Enfin on a parlé sur quelque protocole est déjà développé et on a implémenté un nouveau protocole authentification mutuelle qui est amélioré, nous utilisons la signature numérique EDCSA et chiffrement ECIES, le message est transféré par un canal Visual de camera de mobile qui joue un grand rôle dans notre protocole facile, rapide et sécurisé.

Le protocole peut réaliser une authentification mutuelle, entre Mobile et le Serveur avec toute sécurité car le Serveur résisté à l'attaque par rejeu et l'attaque homme au milieu, il utilise fonction de cryptage asymétrique, et les fonctions de hachages avant de Transfer les données.

Tant que le mobile est avec l'utilisateur l'attaquant ne peut pas obtenir, les informations de communication contiennent toutes des opérations de hachage et cryptographie, ce qui garantit que l'information ne peut pas être écoutée par un attaquant.

Après une analyse d'utilisabilité et une analyse de sécurité, nous pouvons constater que notre schéma est plus simple, sécurisé et amical par rapport à d'autres types de schémas similaires, et il convient très bien aux applications pratiques.

BIBLIOGRAPHIE

- [1] C-T Huang¹, Y-H Zhang, and L-C Lin, Mutual authentications to parties with QR-code applications in mobile systems , 2011.
- [2] J-F Pillou, J-Pe Bay, Tout sur la sécurité informatique , editeur Paris, 2016.
- [3] Secure Digital Documents Using Steganography and QR Code, Brunel universite London 2013.
- [4] K. M.G S Kariyawasam, Mobile phone micropayment protocol based on QR code, Master, University of Colombo School of Computing 2016.
- [5] D Wave, the Inventor of QR Code, <http://www.qrcode.com> ,consulte le 03/03/2017.
- [6] D Wave, the Inventor of QR Code, <http://www.denso-wave.com/qrcode>.
- [7] Mickaël De Almeida, QR Code le code barre version 2D, 2011.
- [8] QR Jacking, <https://www.owasp.org/index.php/Qrljacking>, consulté 10/03/2017.
- [9] E Barkan, E Biham et N Keller, Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication Technion , Computer Science department ,2006.
- [10] S. van de Beek et F. Leferink, , Vulnerability of Remote Keyless-Entry Systems Against Pulsed Electromagnetic Interference and Possible Improvements , IEEE Transactions on Electromagnetic Compatibility, p. 1259–1265.
- [11] Microsoft, <https://technet.microsoft.com/en-us/library/cc959354.aspx>, consulte le 10/03/2017.
- [12] The Internet Engineering, Task Force (IETF®).
- [13]Microsoft, [https://technet.microsoft.com/fr-fr/library/dd469792\(v=ws.11\).aspx](https://technet.microsoft.com/fr-fr/library/dd469792(v=ws.11).aspx), consulte le 11/03/2017.
- [14]Microsoft, [https://technet.microsoft.com/fr-FR/library/dd469759\(WS.10\).aspx](https://technet.microsoft.com/fr-FR/library/dd469759(WS.10).aspx), consulte le 11/03/2017.
- [15]Samuel Grau, Courbes elliptiques Implémentation de la signature électronique, Université de Rouen, 2004/2005.
- [16] IBM, www.ibm.com , Consulté le 04/05/2017.
- [17] Android, www.developer.android.com, Consulte le 05/05/2017.
- [18] ZXING, www.zxingnet.codeplex.com, Consulte le 05/05/2017.
- [19]L Lamport , Password authentication with insecure communication, Communications of the ACM, 1981, 24, (11), pp 770-772.

- [20] M S Hang, L H Li. A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics, 2000 , 46, (1) pp 28-30.
- [21] M L Das, Saxena A, Gulati V P. A dynamic ID-based remote user authentication scheme. IEEE Transactions on Consumer Electronics, 2004, 50, (2) pp 629–631.
- [21] C-a Huang · Y-H Zhang and al, Mutual authentications to parties with QR-code applications in mobile systems, 2016
- [22] S Liu and S Zhu, A Novel QR Code and mobile phone based Authentication protocol via Bluetooth, International Conference on Materials Engineering and Information Technology Applications, MEITA, 2015
- .

ملخص:

QR كود هو نوع من الباركود ببعدين يتألف من وحدات سوداء مرتبة في مربع أبيض ترتيب هذه النقاط تحدد المعلومات الواردة.

في هذه المذكرة ننجز بروتوكول المصادقة المتبادلة الذي يركز على QR كود، وهذا البروتوكول يسمح بنقل ووصول الى البيانات بكل أمان وحرية. هذا الأخير يستعمل التوقيعات الرقمية، وظيفة التجزئة ووظيفة التشفير لتوفير الحماية للمستخدم. انجز هذا البروتوكول على Android Studio.

كلمات المفتاحية: المصادقة المتبادلة، QR كود، التوقيعات الرقمية، وظيفة التجزئة، وظيفة التشفير، Android Studio.

Abstract :

The QR code is a two-dimensional bar code type made up of black modules arranged in a square with a white background. The arrangement of these points defines the information contained in the code.

We realize a mutual authentication protocol based on QR Code. This protocol guarantees transmission and access to data with all the security that uses digital signature, hash function and encryption function. We implemented on Android Studio.

Keyword : Authentification, QR Code, Digital Signature, Hash,

Résume :

Le code QR est un type de code-barres en deux dimensions constitué de modules noirs disposés dans un carré à fond blanc. L'agencement de ces points définit l'information que contient le code

Nous réalisons un protocole authentification mutuelle qui se base sur Code QR. Ce protocole garantie la transmission et accès a les données avec tout la sécurité qui utilise la signature numérique, fonction hachage et fonction de chiffrement. Nous avons implémenté sur Android Studio.

Mot clé : Authentification, Code QR, Signature numérique, Hachage, Chiffrement, Android Studio.