République Algérienne Démocratique et Populaire
C.D.T.A
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab de Blida
Faculté des Sciences
Département d'Informatique

Mémoire de Master 2

en Informatique

Spécialité : Sécurité des systèmes d'informations

Thème

Sécurisation des canaux de communication pour les Webinaires inter-universités

Réalisé par

Nouma Imene

Tamssaouete sonia

Soutenu le : 10/07/2018

Devant le jury

Mme. N. Boustia	*Présidente*
Mr. Kameche Abdellah Hicham	*Examinateur*
Mr. Cherif Zahar Amine	*Promoteur*
Mr. TIBERKAK Allal	*Encadreur*
Mr. HENTOUT Abdelfetah	*Encadreur*

2017/2018

Dédicaces1

Que ce travail témoigne de mes respects :
A mes parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi .

A mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

A toute la famille Tamssaouete, à ma grande famille, mes oncles, et en particulier ma grand-mère

A ma belle-famille Ouchene, en particulier mon fiancé, ma belle-mère et mon beau-père

A mes amies Fatima , Aldjia , Ouardia et Imene :

Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des sœurs et des amies sur qui je peux compter.

En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

Tamssaouete Sonia

Dédicaces2

To my beloved parents whose affection, love, and prayers gave me patient and power.

To our dear friends.

To our dear family.

To the ones we love.

And to the ones that love us.

And to everyone that helped us.

We dedicate this humble work.

Nouma Imene

Résumé

Dans le cadre de rendre l'enseignement à distance et le travail collaboratif plus accessible, les universités algérienne ont recours aux webinaires (web séminaire). Les webinaires référent aux réunions interactives et aux séminaires via Internet. Autrement dit, c'est une conférence en ligne entre un conférencier et plusieurs participants séparés géographiquement. Cependant, ils présentent plusieurs problèmes en terme de sécurité.

Le but de ce projet est de développer un système de sécurisation des canaux de communications entre le conférencier et les participants afin d'assurer la sécurité des informations transmises par ces canaux.

Pour la réalisation de notre projet, nous avons inspiré des problèmes de sécurité de la technologie web temps réel (WebRTC) ainsi que ces mécanismes de sécurité, qui s'entoure sur la sécurisation de la signalisation par le protocole TLS et la sécurisation de flux multimédia par le protocole DTLS.

Mots clés :WebRTc, Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS) .

Abstract

In the context of making e-learning and collaborative work more accessible, Algerian universities are adopting webinars (web seminar). Webinars refer to interactive meetings. In other words, it is an online conference between a speaker and several geographically separated participants. However, it introduces several problems in terms of security.

The purpose of this project is to develop a system for securing the communication channels between the speaker and the participants in order to ensure the security of the data transmitted by these channels.

For the implementation of our project, we have inspired the security problems of real-time web technology (WebRTC) as well as its security mechanisms, which evolves around the security of signalling by TLS protocol and the security of multimedia flows by the DTLS protocol.

Key words : WebRTc, Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS).

Table des matières

Table des figures

Liste des abréviations

AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
DES	Data Encryption Standard
DNS	Domain Name System
DTLS	Datagram Transport Layer Security
PKI	Public Key Infrastructure
RTCP	Real-time Transport Control Protocol)
RTP	Real-time Transport Protocol
SRTP	Secure Real-time Transport Protocol
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol

Introduction générale

Internet a fait ces prouve et a envahi presque tous les domaines, y compris le domaine de l'éducation et de l'enseignement, et cela, de fait que internet propose des solutions faciles permettant de rendre l'enseignement à distance plus accessible, d'encourager le travail collaboratif qui permet d'évoluer et d'échanger des connaissances, de casser toute les barrières qui empêchent la science à faire des pas en avant. Les universités algériennes, comme les autres universités du monde, s'intéressent à la science et veulent améliorer la qualité d'enseignement. De ce fait, elles ont recours aux webinaires (web séminaire) pour faciliter la collaboration entre les universités à l'échelle nationale.

Cependant, les solutions webinaire, comme les systèmes de vidéoconférence, sont affectées par des vulnérabilités liées à Internet, et qui menaces la sécurité et la perturbation de bon fonctionnement du système suite à des attaques comme les attaques de déni de service, et les vols d'identité,

Pour cela, la sécurité dans les vidéoconférences, n'est pas seulement une nécessité mais plutôt une obligation, avec laquelle nous pouvons réduire, au maximum, le risque d'attaques qui viole la disponibilité, l'intégrité et la confidentialité des données échangés entre les participants d'une vidéoconférence.

Le travail que nous avons effectué a pour objectif de faire une étude des problèmes de sécurité inhérents dans les systèmes de vidéoconférence, tout en s'inspirant de la technologie WebRTC, qui est une des solutions de vidéoconférence, pour pouvoir par la suite proposer une solution pour sécuriser une plateforme de vidéoconférence qui sera déployé pour les universités algériennes.Ainsi, est organisé ce document comportant quatre chapitres :

Le premier chapitre intitulé *Les systèmes de vidéoconférence*, est un état de l'art

sur les système de vidéoconférence. Il traite l'aspect général du fonctionnement d'une vidéoconférence. il dresse un panorama de protocoles utilisés dans ce type de système à savoir les *de l'initialisation de communication* et *les protocole de transport de flux multimédia* ainsi qu'un ensemble de notions serviables dans ce contexte. Une solution de vidéoconférence, qui est le WebRTC, sera incluse dans ce chapitre.

Quant au deuxième chapitre nommé *Sécurité des systèmes de vidéoconférence*, comme son nom l'indique, traite l'aspect sécurité dans les systèmes de vidéoconférence. Des problème en terme de sécurité seront exposés au fil de ce chapitre. les solutions existantes pour palier à ces problème seront met en évidence au long de ce chapitre .

Le troisième chapitre intitulé *Conception*, est consacré pour concevoir un système de sécurité pour sécurité la plateforme de vidéoconférence en question, de ce fait, des attaques sur cette plateforme seront détailler au cours de ce chapitre. ce dernier, inclura l'ensemble des solutions proposées pour remédier aux problème de sécurités cités et d'autres qui sont similaires à ce qui été exposés ; mais qui ne sont pas cités.

le dernier chapitre, qui porte le titre*Implémentation*, se concentre sur l'aspect développement de la solution proposée. Les étapes et les outils de développement de la solution seront met en évidence. Des tests de validation de validation de la solution proposée ainsi que leur résultat seront discutés au cours du ce chapitre

Enfin, ce mémoire s'achèvera par une petite conclusion est quelque recommandations pour améliorer le niveau du sécurité du système de vidéoconférence en question.

Chapitre 1

Les systèmes de vidéoconférence

1.1 Introduction

La vidéoconférence est un outil permettant à deux personnes ou plus d'avoir une communication en temps réel à distance. La communication en temps réel peut être audio, vidéo, chat, partage de fichiers et partage d'écran. L'idée de la vidéoconférence ne date pas d'hier. En effet, la mise en œuvre de cette idée a débuté dans les années 60 purement à but de démonstration dans des expositions mais n'a jamais réellement vu le jour dans ces années-là. A cette époque, l'installation d'une telle technologie était très coûteuse et donnait des résultats médiocres. A cause de la lenteur des lignes téléphoniques utilisées pour transporter le signal, il n'a pas été possible de rendre abordable la vidéoconférence à grande échelle. Il aura fallu attendre jusque dans les années 80 pour que les premiers appels visiophoniques deviennent possibles après une amélioration de la technologie, notamment de ses méthodes de codage et la baisse de coût des équipements.

Cependant, déjà à cette époque, une question restait sans réponse à savoir l'utilité réelle de la visiophonie. Quel est l'avantage de voir son interlocuteur quand un simple coup de téléphone devrait suffire ? En vérité, le besoin de cette technologie se fait de plus en plus ressentir sur le marché. L'internationalisation des entreprises nécessite un contact fréquent avec des personnes positionnées en des lieux géographiquement éloignés et les coûts de déplacement d'un employé à la participation d'une réunion peuvent vite devenir onéreux. C'est entre autres pour ces raisons que la vidéoconférence peut s'avérer utile si elle est utilisée à bon escient.

Cependant, durant de nombreuses années, la vidéoconférence est restée une technologie attendue mais non-exploitable. A l'heure actuelle, grâce aux avancées technologiques dans le domaine de télécommunication, les entreprises peuvent désormais l'utiliser et découvrir ses bénéfices et surtout son retour sur l'investissement.

1.2 Concepts de la vidéoconférence :

Il existe deux types de réunions utilisant la vidéo : *point-à-point* appelé la visiophonie et *vidéoconférence multipoint.* La vidéoconférence point-à-point, dans sa forme la plus basique, relie un individu ou un groupe d'individus à un autre[1].

Contrairement à la visiophonie qui est la communication avec vidéo entre deux interlocuteurs seulement, la vidéoconférence permet de faire dialoguer plus que deux personnes en même temps et donc de simuler une conférence à plusieurs individus éloignés géographiquement[2].

Pour cela, il existe plusieurs façons de s'organiser mais tout d'abord il faut définir ce qu'on appelle un *point* .

1.2.1 Notion de point :

On appelle point chaque équipement permettant de faire de la vidéoconférence. Par exemple : une salle équipée, un ordinateur ou un téléphone portable sont des points qui vont servir à se connecter avec plusieurs interlocuteurs[1].

1.2.2 Mode de diffusion :

La mise en place d'une réunion par vidéoconférence selon les outils disposé. Cette mise en place est appelée « mode de diffusion ». Elle représente l'organisation d'une vidéoconférence. Elle peut être constituée de plusieurs personnes disposant chacune de son propre système pour interagir dans la conférence, ou alors, d'un seul système dédié à plusieurs personnes en même temps[1].

Mode point à point :

Une simple visiophonie est un mode point à point. C'est à dire que seul deux interlocuteurs sont en relation avec chacun leur dispositif nécessaire[1].

FIGURE 1.1 – Mode point à point

Mode broadcast :

Dans ce mode plusieurs points écoutent un seul point. Ce système est utilisé lorsqu'un message important doit être diffusé à plusieurs endroits et que les autres points n'ont aucune raison d'interagir entre eux[1].

FIGURE 1.2 – Mode Broadcast

Mode multipoint :

Ce dernier mode de diffusion appelé multipoints va faire interagir tous les points entre eux. Dans ce mode, le partage de la conférence est complet et égal pour tous. Chaque interlocuteur pourra ainsi s'exprimer et se faire entendre par tous les autres[1].

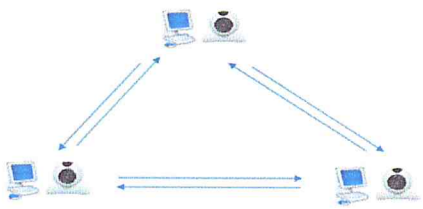

FIGURE 1.3 – Mode multipoint

1.3 Protocole de signalisation :

La connexion de deux pairs nécessite une initialisation de la communication ou *signaling*[3, 4]. Cette initialisation requiert impérativement un serveur distant qui va servir d'intermédiaire aux différents participants. L'initialisation se résume à l'authentification des participants et à l'échange d'informations nécessaires à la localisation des multiples participants (adresse IP, port)[3, 4].

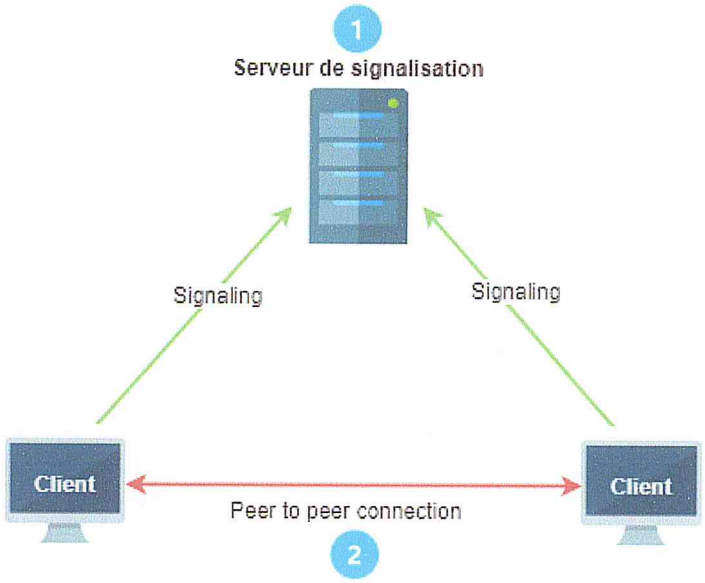

FIGURE 1.4 – Initialisation de la communication

1. Chaque participant s'identifie au niveau de serveur de signalisation et échange l'information nécessaire à l'établissement d'une connexion avec les autres participants.
2. Chaque participant peut maintenant partager directement son contenu multimédia avec les autres participants et la communication peut alors commencer.

Plusieurs protocoles concurrents d'initialisation sont utilisés dans le contexte de signalisation tel que *SIP*, *XMPP*, *Socket.IO* ou encore différents protocoles personnalisés et/ou propriétaires [5, 3].

1.4 Technologie WebRTC :

La technologie *WebRTC (Web Real-Time Communication)* est une technologie qui prend en charge la communication navigateur à navigateur *browser-to-browser* pour les

appels vocaux, chat vidéo et le partage de fichiers point à point sans avoir besoin de *plugins* internes ou externes [5, 6, 3, 4, 7]. La technologie est développée par *Google*et a été publiée en open source en mai 2011. Actuellement, Microsoft Edge12, Google Chrome 28, Mozilla Firefox 22, Safari 11, Opera 18, Vivaldi 1.9 etc. supportent *WebRTC* sans *plugins* externes[5].

WebRTC diffère des méthodes existantes pour communiquer via un navigateur dans la mesure où d'autres utilisent un modèle client-serveur où tout le trafic passe par le serveur. Cela signifie que pour que deux clients communiquent entre eux, leur trafic doit passer par le serveur. Avec *WebRTC*, le serveur est seulement utilisé pour établir la connexion entre les points et leur trafic va directement à l'autre point[5, 3, 4, 7].

WebRTC se distingue d'abord par le choix du protocole de transport qui le soutient. En effet, *WebRTC* a déterminé *UDP* comme couche de transport (du modèle OSI) qui se prête particulièrement bien au contexte temps-réel où la latence devient critique. Si UDP n'offre aucune garantie par rapport à TCP (qui garantit l'arrivée des paquets, mais aussi l'ordre de ceux-ci), il se prête beaucoup mieux aux flux multimédias en temps réel qui peuvent généralement supporter la perte de certaines données (codecs audio et vidéo)[2]. Comme le diagramme précédant en témoigne(figure 1.5), d'autres protocoles sont mis en

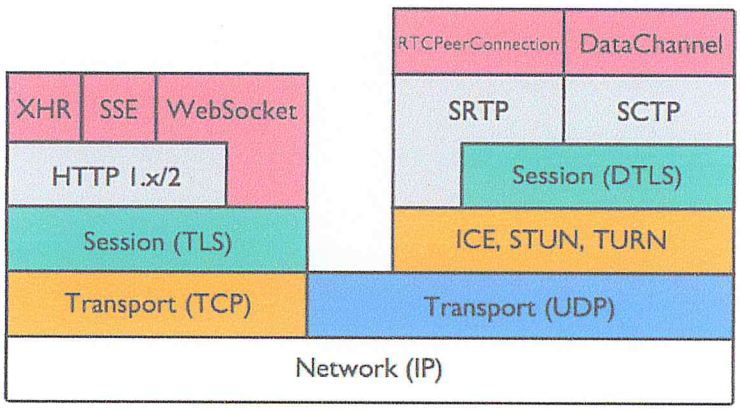

FIGURE 1.5 – Protocoles de Web à gauche et WebRTC à droite

œuvre pour permettre notamment le cryptage du contenu, mais aussi le contrôle du débit, le contrôle de la congestion, la gestion des erreurs, etc., tous nécessaires pour éviter la submersion du réseau et la dégradation de la communication [7, 5, 3] :

DTLS est utilisé comme couche obligatoire de sécurité et de cryptage [8, 9, 10].
SRTP est le protocole utilisé pour le transport des flux audio/vidéo [11, 10].
SCTP est utilisé comme transport des autres données applicatives [5, 3, 4].

1.4.1 Protocoles de transports :

1. **Protocole UDP :**

   Le protocole de datagramme utilisateur (UDP) est le protocole de transport sans confirmation. UDP est un protocole simple qui permet aux applications d'échanger des datagrammes sans accusé de réception ni remise garantie . Le traitement des erreurs et la retransmission doivent être effectués par d'autres protocoles. UDP n'utilise ni fenêtrage, ni accusés de réception, et ne met en place aucun contrôle de flux. Par conséquent, la fiabilité doit être assurée par les protocoles de couche application.

   Les messages UDP peuvent être perdus, dupliqués, remis hors séquence ou arriver trop tôt pour être traité lors de leurs réceptions. UDP est un protocole particulièrement simple conçu pour des applications qui n'ont pas à assembler des séquences de segments. Son avantage est un temps d'exécution court qui permet de tenir compte des contraintes de temps réel ou de limitation d'espace mémoire sur un processeur, contraintes qui ne permettent pas l'implémentation de protocoles beaucoup plus lourds comme TCP.

   Dans des applications temps-réel, UDP est le plus approprié, cependant il présente des faiblesses dues au manque de fiabilité. Des protocoles de transport et de contrôle temps-réel sont utilisés au-dessus du protocole UDP pour remédier à ses faiblesses et assurer sa fiabilité . Ces protocoles sont RTP et RTCP et sont détaillés dans le paragraphe suivant.

2. **Protocole TCP :**

   Le protocole TCP est un protocole de contrôle de transmission, il fait partie de la couche transport du modèle OSI. Il est orienté connexion, c'est-à-dire, il assure un circuit virtuel entre les applications utilisateurs. Le protocole TCP établit un mécanisme d'acquittement et de retransmission de paquets manquants. Ainsi, lorsqu'un paquet se perd et ne parvient pas au destinataire, TCP permet de prévenir l'expéditeur et lui réclame de renvoyer les informations non parvenues.

Il assure d'autre part un contrôle de flux en gérant la congestion qui module le débit d'émission des paquets. Il permet donc de garantir une certaine fiabilité des transmissions. TCP assure un service fiable et est orienté connexion , cependant il ne convient pas à des applications temps réel à cause des longs délais engendrés par le mécanisme d'acquittement et de retransmission.

1.4.2 Protocoles de transfert de flux multimédia :

Pour transporter la voix ou la vidéo sur IP, le protocole*IP (Internet Protocol)* de la couche réseaux et le protocole UDP de la couche transport sont utilisés. En effet, si TCP présente l'avantage de gérer un transfert fiable (renvoi des paquets IP en cas d'erreur), il est malheureusement incompatible avec un flux temps-réel. Mais ces deux protocoles UDP et IP ne suffisent pas à assurer le transport de la voix. De fait, UDP est un protocole sans correction d'erreur, et à aucun moment l'arrivée des paquets dans leur ordre d'émission est assurée.

1. **Protocole RTP :**

   Pour le transport de données temps réel telles que la voix ou la vidéo, il est nécessaire d'utiliser deux protocoles supplémentaires : RTP (Real-Time transport Protocol) et RTCP (RTP Control Protocol)[12, 13, 4]. RTP et RTCP sont des protocoles qui se situent au niveau de la couche application et s'appuient sur le protocole de transport UDP. RTP et RTCP peuvent utiliser aussi bien le mode *Unicast* (point à point) que le mode *Multicast (multipoint)*. Le but de RTP et de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo).

2. **RTCP (Real-time Transfert Control Protocole) :** RTCP *(Real Time Control Protocol)* est un protocole de contrôle utilisé conjointement avec RTP pour contrôler les flux de données et la gestion de la bande passante. RTCP permet de contrôler le flux RTP, et de véhiculer périodiquement des informations de bout en bout pour renseigner sur la qualité de service de la session de chaque participant à la session [12].

   Des quantités telles que le délai, les paquets reçus et perdus sont très importants pour évaluer la qualité de service de toute transmission et réception temps réelle [12, 4].

1.5 Protocoles utilisés dans la résolution de problème de NAT :

Le réseau Internet est en fait constitué de plusieurs millions de réseaux publics et privés reliés par de nombreux câbles, routeurs et une panoplie d'autres dispositifs comme des pare-feu ou encore des routeurs NAT *(Network Address Translation)*(voire Annexe A) ainsi que différentes restrictions déterminées par les fournisseurs de service Internet (ISP) [6].

1.5.1 Network Adress traversal (NAT) :

La majorité des appareils disposent d'une adresse IP privée et nécessite un dispositif NAT qui s'occupe de transformer l'adresse privée en adresse publique (ainsi que les ports) pour pouvoir accéder à l'Internet[2, 14].

Types de NAT :

la figure 1.6 représente les différents type de NAT qui existent.

Full cone NAT	Association indépendante de la destination.
	Filtrage indépendant de la destination.
Restricted cone NAT	Association indépendante de la destination.
	Filtrage par adresse uniquement.
Port-restricted cone NAT	Association indépendante de la destination.
	Filtrage par adresse et port.
Symmetric NAT	Association dépendante (adresse ou adresse/port).
	Filtrage par adresse et port.

FIGURE 1.6 – Type de NAT

NAT symétrique(Symemetric NAT) :

* Le NAT symétrique considère la destination lorsqu'il choisit quelle association utiliser (Association dépendante). De ce fait, la moindre modification dans l'adresse et le port de destination imposera au NAT de créer une nouvelle association. Bien évidemment, tout changement dans la source impliquera aussi une modification de l'association[2, 14, 1].

* Aussi, le NAT symétrique applique un mécanisme de filtrage simple, n'autorisant le trafic entrant que pour les couples adresse et port ayant au préalable été contactées par le client(Filtrage par adresse et port) [2, 14, 1].

1.5.2 Traversée de NAT :

Les NAT ne se préoccupent en effet que des informations réseaux de routage et transport (TCP/UDP/IP), et ignorent totalement les données applicatives utilisées par les protocoles de vidéoconférences afin d'établir leurs communications. Le problème est que ces données contiennent parfois des adresses IP privées qui ne sont alors pas exploitables par les correspondants situés en dehors du réseau privé. Cela conduit alors inévitablement à des problèmes dans l'établissement des appels (échec global, appel établi mais sans canal audio, perte de messages de signalisation).Pour pallier à ce problèmes, des solutions ont été proposées, entre autre, on trouve des serveurs *Session Traversal Utilities for NAT (STUN)* et des serveurs *Traversal Using Relays around NAT (TURN)*

1.5.3 Serveur STUN :

Un serveur STUN permet au client STUN de connaître les informations réseaux qu'il expose à l'externe, soit son adresse IP publique, le port, mais aussi le type de routeur NAT devant lui [15].

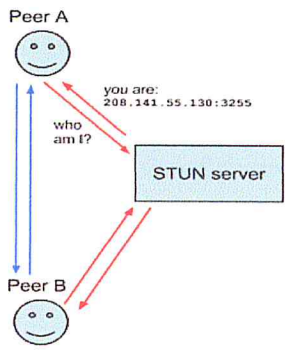

FIGURE 1.7 – Stun

1.5.4 Serveur TURN :

Un serveur TURN est une extension à STUN lorsque celui-ci ne parvient pas à établir la connexion. Il va permettre de relayer le média d'un point communicant à l'autre et devra être présent durant toute la communication contrairement à STUN. TURN est une opération définitivement plus exigeante et la communication directe devrait en tout temps être privilégiée [16] .

La figure 1.10 montre à quoi ressemble une simple architecture d'une communication multimédia après avoir ajouté les serveurs STUN et TURN :

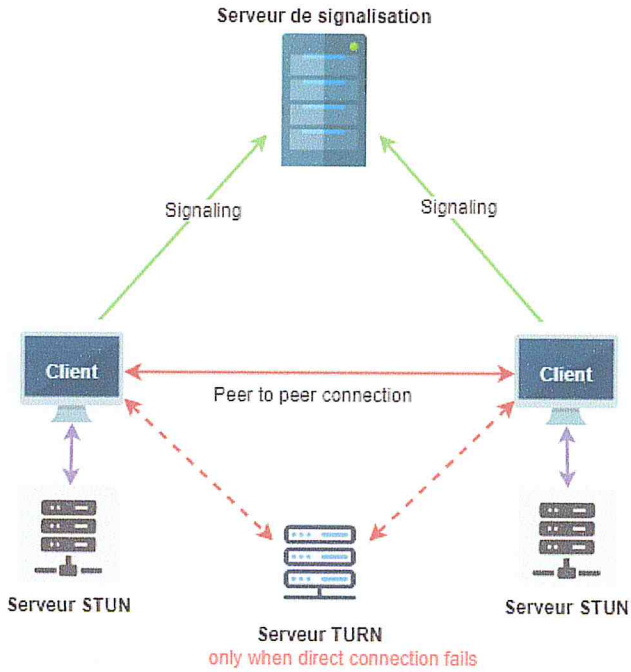

FIGURE 1.8 – Architecture d'une communication avec serveurs TURN et STUN

TURN Doit contourner la restriction de NAT Symétrique en ouvrant une connexion avec un serveur TURN et retransmettre toutes les informations par le biais de ce serveur. Une connexion avec un serveur TURN doit être créer et dire à tous les points d'envoyer des paquets au serveur qui seront alors expédiés. Cela vient évidemment avec une certaine surcharge sur le serveur TURN ce qui fait que ce dernier ne sera utilser que s'il n'y a pas d'autres alternatives [16] .

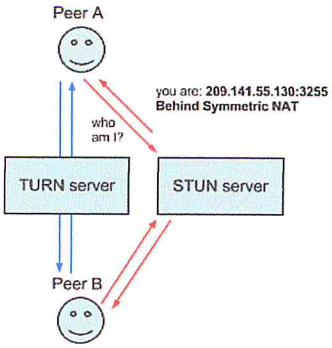

FIGURE 1.9 – Turn

1.6 MCU dans la vidéoconférences :

WebRTC tel qu'il est implémenté actuellement ne prend en charge que la communication point à point mais peut être utilisé dans des scénarios de réseau plus complexes par exemple avec plusieurs points chacun communiquant directement, point à point ou via une unité de contrôle multipoint (MCU).

Le MCU *(Multipoint Control Unit)* est utilisée pour mettre en place des conférences multimédias entre plusieurs utilisateurs, au moins deux. Comme l'illustre la figure 1.10, tous les utilisateurs qui veulent participer à une conférence doivent se connecter au MCU afin d'y définir et de négocier les paramètres de communication à utiliser.

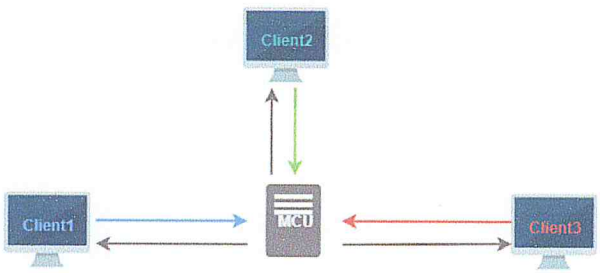

FIGURE 1.10 – Modèle Multipoint Conferencing Unit(MCU)

Contrairement aux autres types d'équipement, la MCU ne s'intéresse pas uniquement à la signalisation, mais aussi au transfert des données multimédias. Nous allons décrire les entités constituantes d'un MCU, avant d'examiner comment s'effectue une conférence multimédia.

La MCU désigne un équipement composé de deux entités, qui jouent un rôle complémentaire, le MC *(Multipoint Controller)* et le MP *(Multipoint Processor)[1].*

MC (Multipoint Controller) :

Le Multipoint Controller est chargé de la négociation des paramètres. En préalable à la conférence, tous les utilisateurs négocient l'ensemble des paramètres de communication qu'ils désirent utiliser, selon les capacités de leur terminal et leur souhait. Ils conviennent notamment du mode d'adressage (unicast ou multicast), du type de flux souhaité (audio, vidéo ou les deux), du codec à utiliser et de la bande passante nécessaire[1, 17].

Le MC n'intervient que pour les signaux de contrôle, à l'exclusion donc des données multimédias proprement dites, auxquelles il ne s'intéresse pas. Contrairement au MP, le MC n'est pas un intermédiaire de la communication. Il est toujours soit émetteur soit récepteur des messages et ne fait pas transiter les messages qu'il reçoit d'un poste vers un autre[1, 17].

MP (Multipoint Processor)

Le *Multipoint Processor* est un centre de traitement des flux multimédias.

Dans une conférence, chaque utilisateur peut disposer de paramètres spécifiques. L'un peut réclamer un codec audio de très bonne qualité, un autre un codec de moins bonne qualité, un troisième ajouter la vidéo en plus de l'audio. Pour satisfaire ces demandes, tous les utilisateurs se connectent auprès de l'entité MP, laquelle leur délivre à chacun, dans la limite de ses possibilités, les flux qu'ils sollicitent[1, 17].

1.7 Conclusion :

Dans ce chapitre, nous avons parlé sur les systèmes de vidéoconférence, nous avons vu leurs principes du fonctionnement, les protocoles déployés et une des solutions qui existe, qui est utilisée pour des solutions basées web, qui est la technologie WebRTC.

Dans le chapitre qui va suivre, nous allons voir les problèmes de sécurité dans ces systèmes et les mécanismes pour contrer à ces problèmes.

Chapitre 2

Sécurité des systèmes de vidéoconférence

2.1 Introduction :

Comme tout autres systèmes, les systèmes de vidéoconférence peuvent être soumis aux différents types d'attaques, soit pour en perturber le fonctionnement, soit pour intercepter les informations transmises, et même pire, le rendre indisponible .

Cependant, pour éviter toute divulgation d'information, le trafic réseau doit être chiffré, de manière que quiconque n'ayant pas le droit à avoir l'information ne puisse y parvenir.

Dans ce chapitre, nous commençons par présenter les propriétés qui sont à la base de l'expression des besoins de sécurité. Nous détaillons par la suite les principales attaques de sécurité contre les systèmes de vidéoconférence. On terminera par présenter les mécanismes et les moyens à mettre en place pour contrer ces problèmes.

2.2 Propriétés de sécurité :

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire les vulnérabilités d'un système contre les menaces accidentelles ou intentionnelles [18].

Selon[19], on peut définir le mot " *sécurité* " comme étant la capacité d'un système informatique à résister à des agressions externes physiques (incendie , inondation, bombe, etc) ou logique (erreurs de saisie, intrusions , etc).

Dans cette section , nous présentons les propriétés de sécurité usuelles et celles qui peuvent être dérivées .

L'ensemble des propriétés de sécurités est couramment vu comme l'ensemble dérivé de trois propriétés principales, à savoir, *La confidentialité, l'intégrité* et *la disponibilité*, ou encore, le triangle *CIA(Confidentiality, Integrity, Availability)*[20]. Plusieurs définitions existent dans la littérature, nous en présentons ici une synthèse.

2.2.1 Confidentialité :

La propriété de confidentialité peut être exprimée comme suit :

Une information est dite confidentielle si elle n'est accessible que par les entités autorisées[21, 22].

Cette propriétés implique que l'information ne doit être accessible que par ceux qui ont le droit. En d'autres termes, une personne non-autorisée à accéder à une information quelconque ne doit en aucun cas y accéder [23, 24] .

2.2.2 Intégrité :

La propriété d'intégrité désigne le fait que l'information ne puisse être altérée partiellement (modifiée) ou altérée totalement (détruite ou supprimée) à l'insu de son propriétaire tant de manière intentionnelle qu'accidentelle [24]. Autrement dit *Une information est dite intègre si elle ne peut être altérée ou supprimée que par les entités autorisées* [21, 22].

Comme dans le cas de la confidentialité , la propriété d'intégrité spécifie un ensemble des entités ayant l'autorisation pour modifier ou supprimer une information [23]. Tout autre entité n'appartenant pas à cet ensemble et désireuse de modifier ou encore de supprimer cette information n'en pourra pas aboutir[23] .

2.2.3 Disponibilité :

C'est la propriété par laquelle est mesurée la fiabilité d'un système [25, 23, 26]. Un système qui n'est pas disponible est dit défaillant. Elle exprime la possibilité d'accéder à

une information ou une ressource. Une définition pour cette propriété peut être comme suit :

Une information est dite disponible si elle est accessible par les entités autorisés à tout instant où il leur est permis d'y accéder[20, 22]

La notion temporelle d'accès est relative au domaine d'application , l'accès à une information ou un service doit être plus rapide dans un système critique(exemple :système de gestion des patient dans un hôpital) que dans un système non-critique.

Comme cité plus haut, ces trois propriétés(*CIA*) sont les piliers de la sécurité , deux autres propriétés s'ajoutent, l'authentification et la non-répudiation [21, 22].

2.2.4 Authentification :

L'authentification est le processus d'établissement de confiance dans l'identité des utilisateurs. C'est le processus qui consiste à vérifier si l'entité est celle qu'elle prétend être [21, 27]. L'authentification est la clé pour assurer la sécurité.

L'authentification se compose de deux volets, qui sont l'identification et la vérification [23, 22].

L'identification consiste à présenter l'identité de l'entité, cette dernière étant une information non-secrète, connue au moins par l'entité qui la proclame ainsi que par le système informatique [23, 20, 28] .

Quant à la vérification, elle consiste à prouver l'identité de l'entité proclamée [23, 20, 24]. La procédure de vérification se fait en introduisant une information de plus pour le système, c'est-à-dire, l'entité doit présenter une information ou une combinaison des informations suivantes :

* **Vérification à base de quelque chose que l'utilisateur connait :** C'est une méthode qui permet de vérifier l'identité des utilisateurs en se basant sur des informations mémorisées par le systèmes lors de la phase d'inscription, entre autre, un mot de passe , une question secrète [23, 20, 28].

* **Vérification à base de quelque chose que l'utilisateur possède :** C'est une méthode selon laquelle les utilisateurs sont authentifiés par des informations contenues dans un physique en leur possession comme un certificat numérique, une carte à puce, un passeport [28, 20, 23] .
* **Vérification à base de ce qui est propre à l'utilisateur :** Ensemble de méthodes permettant la vérification automatique de l'identité des personnes sur la base des caractéristiques personnelles, physiologiques et/ou comportementales [28, 20, 23], par exemple, les empreintes digitales, reconnaissance faciale et reconnaissance vocale.

L'authentification peut être de deux types, à savoir l'authentification faible et l'authentification forte [23]. le premier type englobe l'authentification à base des mots de passes fixes ou à usage unique [20], quant au deuxième type, qui est l'authentification forte, consiste en un *protocole d'authentification*[20]. Ce dernier est un type de protocole cryptographique, dont le but est l'authentification des parties qui souhaitent communiquer en toute sécurité. Il comporte une séquence de messages échangés entre deux parties permettant l'utilisation/la possession d'un secret, dans le but d'être confirmé [23, 28] .

Le but d'établir un tel protocole entre des parties communicantes, est de pouvoir prouver un secret qui sera créer et/ou transférer ou utiliser pour le reste de la session, et ce, afin d'assurer la confidentialité de toutes les données communiquées [23]. Les primitives cryptographiques les plus déployées dans les protocoles d'authentification comprennent : les signatures électroniques, le hash, les mécanismes de chiffrement/déchiffrement [20, 29, 27].

Il existe plusieurs protocoles d'authentification et méthodes disponibles. Chaque protocole d'authentification emploie certaines méthodes pour réaliser l'authentification, bien que la mise en œuvre puisse différer en termes de robustesse et des processus impliqués.

La majorité des protocoles d'authentification utilisent un secret [28, 20], ce dernier étant soit pré-partagé ou dérivé pour mener le processus d'authentification d'identité [28, 23]. Ils utilisent des nombres aléatoires, des fonctions de hachage, des défis et des estampilles temporelles pour améliorer la robustesse ou ajouter des fonctionnalités au

protocole [28]. Comme exemple de protocoles d'authentification, nous avons le protocole *Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), Extensible Authentication Protocol (EAP), SSL/TLS, IPSec, RADIUS, kerberos* [23].

2.2.5 Non-répudiation :

C'est le fait que l'émetteur d'un message ne puisse nier l'avoir envoyé et le récepteur l'avoir reçu. Le reçu que l'on signe au livreur, la lettre recommandée avec accusé de réception sont des mécanismes de non répudiation [28, 23].

Dans cette section, nous avons fait un petit tour d'horizon sur les besoins cruciaux en terme de sécurité. Dans la section qui va suivre, nous allons voir les problèmes de sécurité qu'on peut avoir dans les systèmes de vidéoconférence et qui violent les propriétés de sécurité citées avant.

2.3 Problèmes de sécurité liées à la vidéoconférence :

La sécurité des systèmes de vidéoconférence n'est pas vraiment différente de celle des autres applications du monde IP. Nous allons dans un premier temps voir les sources potentielles et les objectifs d'une attaque, et on examinera par la suite les attaques possibles dans les systèmes de vidéoconférence.

2.3.1 Sources et objectifs d'une attaque :

Les vulnérabilités dont les attaques peuvent tirer parti peuvent avoir cinq origines :[1]

* Les protocoles.
* Les logiciels.
* Les systèmes d'exploitation.
* L'infrastructure physique.
* Les erreurs humains.

Chacune d'elles est une source potentielle de faille, qu'il convient d'étudier avec précaution dans la mise en place d'une solution de vidéoconférence.

Une attaque peut avoir trois objectifs [1] :

1. Acquisition de service : L'objectif d'une telle attaque est de s'approprier des droits et fonctionnalités qui n'ont pas véritablement été attribués à l'attaquant [1].
2. Interception de service : Cette attaque compromet la confidentialité du service et vise à en analyser ou modifier le contenu [1].
3. Interruption de service : L'objectif est purement de nuire au bon déroulement du service en cherchant à le mettre hors d'usage [1].

2.3.2 Attaques contre la vidéoconférence :

Une vidéoconférence est constitué de deux parties : la signalisation qui instaure la conférence, et les flux de media qui transporte la voix, l'audio. Les types d'attaques les plus fréquentes contre un système vidéoconférence sont :

Usurpation ARP (ARP spoofing) :

Lorsqu'une machine essaye de communiquer avec une autre qui se trouve dans le même réseau qu'elle, elle a recours au protocole *ARP*. Ce dernier récupère l'adresse *MAC* de la machine destinataire et l'enregistre sur un tableau qu'on appelle le cache *ARP* [14, 30].

L'ARP spoofing consiste à envoyer des réponses *ARP* à une machine cible pour usurper une autre machine. Ainsi, à chaque fois que la cible voudrait communiquer avec la deuxième machine, elle le fera avec la machine usurpatrice [14, 30].

Usurpation IP(IP spoofng) :

L'usurpation IP consiste à forger un paquet avec une adresse IP source falsifiée [28, 14, 31]. Cela peut avoir comme objectif de surpasser un système d'authentification qui se base sur les adresses IP, comme il peut avoir comme objectif le lancement d'une attaque en modifiant son identité [32, 33].

Homme du milieu (Man in the middele) :

L'attaque de l'homme du milieu consiste à faire passer le trafic transitant entre deux machines par le biais d'une troisième qui est sous la prise d'un pirate sans que les intervenants ne le suspectent [14, 6, 33]. Cela est possible par exemple en appliquant de l'usurpation *ARP* au niveau des deux machines [28].

Déni de service :

Un *DOS (Denial of Service)* est une attaque qui consiste à utiliser toutes les ressources d'un service pour qu'il ne soit plus accessible [14]. Ainsi, personne ne peut plus utiliser un serveur de signalisation ou tout autre service disponible. Le plus gros problème de cette attaque est qu'elle ne peut pas être contrée, sauf en fournissant plus de ressources [32, 13, 33].

Il existe plusieurs façons pour faire l'attaque de *DOS*, entre autre, on trouve *SYN Flooding* et *ICMP Flooding* [28, 14].

1. **Inondation SYN (SYN Flooding) :**

   Le protocole TCP/IP s'appuie sur un modèle appelé le *three way handshake* pour établir des connexions *TCP* [1](voire figure 2.1). Ce modèle se dirige normalement de la façon décrite dans la figure 2.1 [1]. Le client commence par envoyer une demande de connexion *(SYN)*. A la réception de cette demande le serveur réserve des ressources et envoie son acceptation sous forme d'acquittement *(SYN-ACK)*. Le client à son tour doit accuser la réception de cette acceptation et s'attache à ces ressources réservées. Et puis, la communication peut s'établir suivant cette connexion.

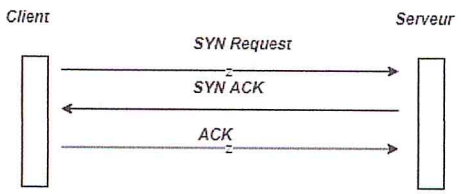

FIGURE 2.1 – établissement d'une connexion TCP

   Une personne malveillante n'acquittera jamais la réponse du serveur et tentera de demander plus de connexions *TCP* plus vite que le *timeout* des semi-connexions. Cela occupera une partie de ressource du serveur jusqu'à l'épuisement.

2. **ICMP(ICMP flooding) :** L'inondation *ICMP* consiste en l'envoie de plusieurs *ping* à la cible. Cela nécessite que la bande passante de l'attaquant soit nettement importante par rapport à celle de la cible [14].

Une variante intéressante de cette attaque s'appelle le *smurf*. Au lieu d'envoyer des *ping* vers la cible, avec l'attaque *smurf* [14] on envoie un *ping* vers une adresse

de *broadcast* (voire Annexe A) en usurpant l'adresse IP. Cette adresse IP recevra les réponses de *ping* de la part de toutes les machines qui sont actives dans le réseau.

DoS distribue (Distributed DoS) :

Le *DDoS* est une dérivée distribuée du déni de service [14]. Il consiste en l'envoi d'attaques *DoS* depuis un ensemble de machines [14]. Ces dernières peuvent être en collaboration pour une cause quelconque, comme elles peuvent être sous la prise d'une personne malveillante. Chose qui amplifie considérablement l'impact de l'attaque sur la cible.

Écoute clandestine (Eavesdropping) :

L'eavesdropping [32] est l'acte d'écouter secrètement une conversation de vidéoconférence sans avoir l'accord des participants. Ceci pourrait être réalisé en capturant les paquets. En effet, un attaquant peut intercepter et enregistrer le trafic en utilisant des analyseurs de réseau. Il peut également décoder la conversation [32][13].

Attaques sur le protocole RTP

Après l'établissement d'une session entre deux entités communicantes, le protocole RTP est utilisé pour transporter les données multimédia (trafic en temps réel) entre les parties communicantes [11, 1, 31].

Dans ce contexte, ce protocole est cible de plusieurs attaques parmis les on trouve : l'attaque de charge utile *(Payload attack)* et l'attaque de falsification *(Tampering attack)* [32].

1. ***Attaque de charge utile :***

   Un utilisateur malveillant peut utiliser l'attaque d'homme-du-milieu *(Man-In-The-Middle)* [20] pour intercepter et modifier la charge utile d'un message de flux RTP. Comme il est indiqué dans [34]et [31], le flux multimédia transitant entre deux entités est transporté par le protocole RTP. Ce protocole est une extension au protocole UDP (User Datagram Protocol) [31] tout en ajoutant des informations de séquencement [11, 1]. Cela signifie qu'un attaquant pouvant inspecter la

charge utile du message est capable d'intercepter la communication. En outre, si un attaquant peut modifier le message, alors il serait en mesure de modifier le sens de la conversation en injectant son propre message ou dans un autre cas, d'introduire un bruit afin de dégrader la qualité du message.

2. ***Attaque de falsification :***

   Le paquet RTP comporte deux champs dans son en-tête qui peuvent être manipulés par un attaquant. Ces champs sont le numéro de séquence et les champs d'horodatage *(timestamp fields)*. L'attaquant peut modifier avec succès la séquence du paquet et rend ainsi la conversation dénuée de sens [32, 1, 11].

DNS Spoofing :

L'objectif de cette attaque est de rediriger, à l'insu des point communicants, les participants d'une vidéoconférence vers des serveurs des pirates (Serveur de signalisation). Pour la mener à bien, le pirate utilise des faiblesses du protocole *DNS (Domain Name System)* et/ou de son implémentation au travers des serveurs de nom de domaine. A titre de rappel, le protocole *DNS* met en œuvre les mécanismes permettant de faire la correspondance entre une adresse *IP* et un nom de machine [28, 33, 35]. Il existe deux principales attaques de type *DNS Spoofing* [28]. Nous allons voir seulement un seul type qui est *DNS ID Spoofing*. Concrètement, le but du pirate est de faire correspondre l'adresse IP d'une machine qu'il contrôle à un nom réel et valide d'une machine publique.

Description de l'attaque DNS ID Spoofing :

Si une machine **A** veut communiquer avec une machine **B**, la machine **A** a obligatoirement besoin de l'adresse IP de la machine **B**. Cependant, il se peut que A possède uniquement le nom de B. Dans ce cas, A va utiliser le protocole *DNS* pour obtenir l'adresse IP de B à partir de son nom. Une requête *DNS* est alors envoyée à un serveur *DNS*, déclaré au niveau de *A*, demandant la résolution du nom de *B* en son adresse IP. Pour identifier cette requête un numéro d'identification lui est assigné. Ainsi, le serveur *DNS* enverra la réponse à cette requête avec le même numéro d'identification. L'attaque va donc consister à récupérer ce numéro d'identification (en sniffant, quand l'attaque est effectuée sur le même réseau physique, ou en utilisant une faille des systèmes d'exploitation ou des serveurs *DNS* qui rendent prédictibles ces numéros) pour pouvoir envoyer une réponse

falsifiée avant le serveur DNS. Ainsi, la machine A utilisera, sans le savoir, l'adresse IP du pirate et non celle de la machine B initialement destinataire [28].

2.4 Mécanismes de sécurité dans la vidéoconférence :

La plupart des mécanismes de sécurité dans les systèmes de vidéoconférence et plus précisément à l'entour de la signalisation s'intéressent à l'authentification, le chiffrement et l'intégrité des messages de signalisation. Parmi ces mécanismes, nous évoquons *TLS Transport Layer Protocol .*

Cependant, des mécanismes telle que DTLS et SRTP servent pour assurer la sécurité dans le transport des flux média.

Avant de détailler ces mécanismes, on doit présenter quelque concepts fondamentaux dans les mécanises de sécurité. 0 Nous allons présenter dans la suite de ce chapitre quelques mécanismes permettant de faire face à ces attaques dans un système de vidéoconférence .

Concepts fondamentaux :

1. ***Cryptographie :***

   la cryptographie est la science qui utilise les mathématiques pour chiffrer et déchiffrer des données, elle permet de stocker des informations sensibles ou de les faire véhiculer à travers des réseaux peu sûr, comme Internet, de telle sorte qu'elles ne soient intelligibles qu'au destinataire convenu [36, 37]. Elle offre des outils permettant de minimiser les risques et susceptible de rendre l'échange de l'information confidentielle, non falsifiable, authentique et non-altérable .

   la cryptographie se divise en deux grandes familles. La première est la cryptographie à clé secrète, aussi appelée cryptographie symétrique. Cette dernière est basée sur l'échange préalable d'une clé secrète commune d'une taille variable (elle peut être sous la forme d'un chiffre, d'une lettre, d'un livre de code), permettant aux communicants de crypter ou décrypter un message. La seconde famille est connue sous le nom de la cryptographie à clé publique, ou encore cryptographie asymétrique. Elle se différencie de la cryptographie à clé secrète par le fait qu'elle ne nécessite aucun échange secret préalable.

Chiffrement symétrique :Le chiffrement symétrique utilise la même clé pour chiffrer et déchiffrer un message. La connaissance de cette clé est cruciale pour la confidentialité des informations échangées[38, 17, 39, 40, 2].

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions [37]. Cela offre un moyen rapide et efficace pour chiffrer un message. Plusieurs algorithmes ont été développé , entre autre DES [41, 42, 37],TDES, AES , RC4 .

Chiffrement asymétrique : La cryptographie asymétrique se base sur des problèmes mathématiques complexes. Elle se base sur le principe de deux clés : clé publique, clé privée [43, 44, 45]. La première clé, qui est publique, est mise à la disposition de tout le monde, tous ce qui est chiffré avec cette clé ne peut être déchiffrer qu'avec la clé privée, qui doit être confidentielle et connue seulement par son propriétaire.

Un chiffrement asymétrique est défini par trois algorithmes, à savoir l'algorithme de génération des clés, l'algorithme de chiffrement et l'algorithme de déchiffrement.

Parmi les crypto-systèmes asymétrique, on trouve *RSA* [36], du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adlemanest. C'est un système de chiffrement asymétrique basé sur l'arithmétique modulaire et notamment sur le problème de factorisation [46, 47, 48].

2. ***Fonction de hachage :***

Une fonction de hachage est une fonction à sens unique permettant d'obtenir un résumé d'un texte [29, 26, 49, 50], c'est-à-dire. une suite de caractères assez courte représentant le texte qu'il résume. La fonction de hachage doit [51, 52] :

* Être telle qu'elle associe un et un seul résumé à un texte en clair (cela signifie que la moindre modification du document entraine la modification de son résumé)[52].
* Être une fonction à sens unique afin qu'il soit impossible de retrouver le message original à partir du résumé[52].

3. ***Public key infrastructure (PKI) :***

On désigne sous l'appellation PKI *(Public Key Infrastructure)* l'ensemble des moyens matériels,logiciels, composants cryptographiques, mis en œuvre par des personnes, combinés par des politiques,des pratiques et des procédures requises, qui permettent de créer, gérer, conserver, distribuer et révoquer des certificats basés sur la cryptographie asymétrique [53, 54].

4. ***Certificat numérique :***

   Un certificat contient, tel un passeport ou une carte d'identité, un ensemble d'informations administratives sur son propriétaire (nom, prénom, adresse électronique, etc.), sur sa validité, et sur l'organisme d'émission. Il contient également la clé publique de l'utilisateur et un sceau (la signature électronique de l'autorité de certification) nécessaire à la vérification de son authenticité. Il permet de garantir l'identité du possesseur de la clé privée associée.[20, 54]

   Plusieurs normes de certificats existent, les plus employées sont *X509.*

5. ***Politique de sécurité et modèle de contrôle d'accès :***

   Les premières mesures en sécurité informatique consistent à contrôler les différents accès possibles à un système et à autoriser ou non certains nombre d'actions en fonction des utilisateurs .

   Le contrôle d'accès est définit comme n'importe quel mécanisme par lequel un système autorise ou interdit le droit à des entités actives d'accèder à des entités passives ou d'effectuer des opérations. En d'autres terme, le contrôle d'accès est le mécanisme par lequel les propriétés de sécurité, tel que la confédentalité et l'ingrité, sont préservées [25] .

   Une politique de contrôle d'accès est vu comme étant l'ensemble des lois, règles, pratiques qui régissent la façon par laquelle l'information sensible et les autres ressources sont gérées et protégées [25].

2.4.1 TLS

TLS est un mécanisme qui est basé sur le Secure Socket Layer (SSL) et fonctionne sur TCP. Il permet la confidentialité, l'intégrité et l'authentification du serveur. L'authentification mutuelle peut être assurée par l'échange de certificats durant la procédure de prise de contact TLS. Bien que les certificats auto-signés ne peuvent pas être toujours crédible, le maintien d'une infrastructure à clé publique exige beaucoup d'efforts de gestion[10, 55].

2.4.2 DTLS

Le protocole DTLS (Datagram Transport Layer Security) est conçu pour construire un trafic TLS sur datagramme qui ne nécessite pas de transmission de données fiable ou en ordre. Parce que le TLS nécessite un canal de transport fiable tel que TCP, il ne peut pas être utilisé pour sécuriser le trafic de datagramme non fiable. DTLS est une variante de TLS compatible avec les datagrammes [11, 9].

Avant que les données d'application puissent être envoyées ou reçues, le protocole DTLS nécessite une prise de contact pour établir les paramètres cryptographiques. Cette prise de contact nécessite une série de messages de va-et-vient entre le client et le serveur. Le i*Handshake DTLS* exige que tous les messages soient correctement reçus. Ainsi, dans un trafic de datagramme non fiable, les paquets manquants ou retardés doivent être retransmis [9].

2.4.3 SRTP

SRTP est conçu pour sécuriser la multiplication à venir des échanges multimédias sur les réseaux. Il couvre les lacunes de protocoles de sécurité existants, dont le mécanisme d'échanges de clés est trop lourd. Il aussi est bâti sur le protocole RTP. Les principaux services offerts par SRTP sont :

* Rendre confidentielles les données RTP, que ce soit l'en-tête et la charge utile ou seulement la charge utile.
* Authentifier et vérifier l'intégrité des paquets RTP. L'émetteur calcule une empreinte du message à envoyer, puis l'envoie avec le message même.
* La protection contre le rejoue des paquets. Chaque récepteur tient à jour une liste de tous les indices des paquets reçus et bien authentifiés.

2.5 Conclusion

Dans ce deuxième chapitre nous avons présenté les propriétés de sécurité. Nous avons aussi présenté des problèmes de sécurités dans les systèmes de vidéoconférence et les mécanisme à mettre en œuvre pour palier à toute violation de sécurité.

Chapitre 3

Conception

3.1 Introduction :

Dans ce chapitre, nous décrivons l'architecture fonctionnelle du système de vidéoconférence à sécuriser. Nous présentons les différents éléments qui le composent et on donne le rôle de chaque composant ainsi que les interactions entre ces différents composants. Nous traitons ensuite les attaques qui peuvent survenir sur cette architecture. Pour chaque attaque, on donne les composantes concernées. Enfin, nous proposons des solutions pour contrer à ces attaques et assurer la sécurité du système.

3.2 Description de l'architecture du système :

La figure 3.1 représente l'architecture du système de vidéoconférence qu'on veut sécuriser.

Comme le montre la figure 3.1, ce système se compose principalement de :

Conférencier : C'est l'entité qui veut démarrer une conférence.

Participants : C'est les entités qui se joindraient à une conférence démarrée.

MCUs (Multi point control unit) : C'est des entités logicielles qui servent pour la diffusion des flux audio et vidéo pour les autres entités qui sont relies à eux. C'est des entités intermédiaires qui relient le conférencier et les participants.

Serveur de signalisation : Il constitue de l'élément de base de l'architecture. C'est une entité logicielle qui a pour rôle de mettre en contact le conférencier avec

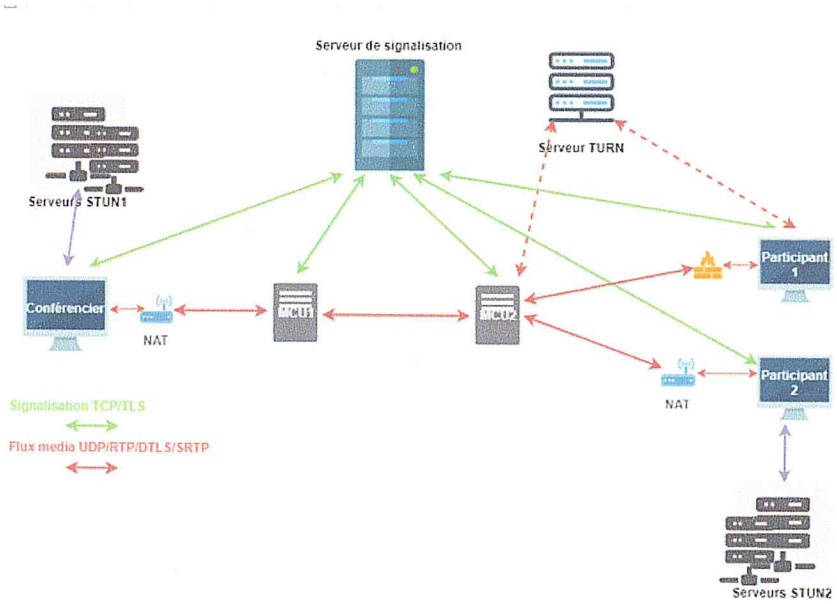

FIGURE 3.1 – Architecture fonctionnelle du système de vidéoconférence

les autres participants vu que ces derniers ne savent pas comment se joindre les uns aux autres. Mais ils savent comment joindre le serveur de signalisation et qui a les informations des autres entités de système.

Serveur STUN : C'est une entité permettant aux participants et aux conférenciers de connaitre les informations réseaux qu'ils exposent à l'externe (adresse IP publique, numéro de port).

Serveur TURN : C'est une entité permettant à un client *TURN* (Participant) d'allouer et utilisé une adresse relayé pour relayer le média d'une entité a une autre entité quand le serveur *STUN* ne parvient pas à établir la connexion.

Pare-feu(Firewall) : Une entité logicielle qui filtre les communications dans le but de contrer les attaques subtiles. Un pare-feu possède au minimum deux interfaces, l'une connectée au réseau privée et l'autre connectée au lien d'accès à Internet. Il filtre les paquets entrant au réseau privé ou sortant du réseau privé vers Internet selon plusieurs critères. Parmi ces critère en trouve l'adresse IP source, l'adresse IP destination, Le numéro de port source et le numéro de port destination.

NAT : C'est une entité qui se trouve entre le réseau local et Internet, il fait le routage de paquet IP de réseau privé vers Internet et inversement. Le logiciel *Nat* ou le périphérique *NAT* se charge de translater les adresses de machines locales, en utilisant son adresse, dans le cas ou il a une seule, ou ses adresses, dans le

cas où il possède plusieurs adresses. Lorsqu'il reçoit un paquet d'une machine du réseau local, il change l'adresse pour mettre sa propre adresse IP, change le port pour mettre une valeur à lui, puis stocke cette information dans une table. Lorsqu'il reçoit la réponse, il vérifie dans ses tables qu'il possède bien l'entrée correspondante, puis réécrit le paquet avec l'adresse IP de la machine émettrice.

3.2.1 Principe de fonctionnement :

Dans cette partie, on va présenter le principe du fonctionnement du système de vidéoconférence. Pour cela on propose un scénario du fonctionnement où on a un conférencier qui organise une conférence dans laquelle on aura un participant. Le conférencier et le participant étant géographiquement séparés.

On commence d'abord par illustrer ce fonctionnement sous forme d'un diagramme de séquence qu'on détaillera sous forme d'un scénario de fonctionnement nominal.

Diagramme de séquence :

La figure 3.2 représente un diagramme de séquence pour le scénario cité avant.

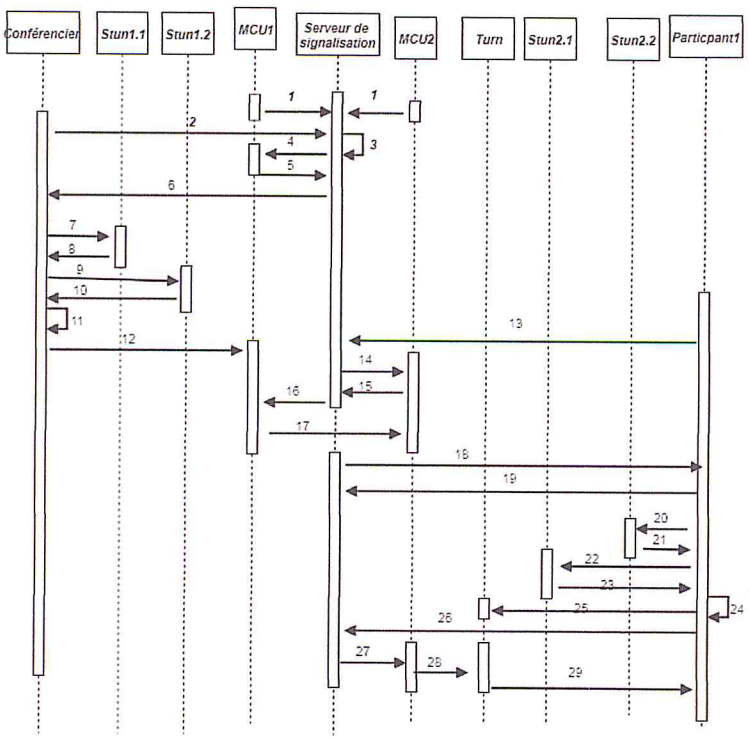

FIGURE 3.2 – Diagramme de séquence de scénario nominal

Scénario de Fonctionnement nominal :

Pour bien comprendre le diagramme de séquence qui se trouve dans la figure 3.2, on propose ce scénario nominal par lequel on met en évidence les différents messages échangés par les différentes entités, et ce dans le but de voir les interactions entres elles.

1. MCU1/MCU2 rejoignent le serveur de signalisation Ils annoncent leurs présence.
2. Le conférencier sollicite le serveur de signalisation et demande de démarrer une conférence.
3. Le serveur de signalisation demande l'adresse IP et le port d'écoute de MCU adéquat.
4. MCU crée un canal UDP ensuite il envoi l'adresse IP et port au serveur de signalisation.
5. Le serveur de signalisation envoi ces informations au conférencier.
6. Conférencier envois les frames vidéo de la conférence au MCU adéquat.
7. Le client demande de rejoindre la conférence.
8. Le serveur de signalisation calcule le chemin (Conférencier, MCU1, MCU2, Client)
9. Le serveur de signalisation demande l'adresse IP et le numéro de port du MCU2.
10. Le MCU2 ouvre un canal UDP, ensuite il envoie son adresse IP et port au Serveur de signalisation.
11. Le serveur de signalisation envoi l'adresse IP et port de MCU2 au MCU1.
12. Le MCU1 envoi les frames vidéo de la conférence au MCU2.
13. Le serveur de signalisation demande l'adresse IP et port de participant.
14. Le participant ouvre le canal UDP, ensuite il sollicite les serveurs Stun pour avoir son adresse IP publique et le numéro du port publique du canal créé.
15. Si le NAT est symétrique le participant demande une adresse relayée du serveur TURN.
16. Le participant envoi l'adresse IP et port relayé de serveur TURN au Serveur de signalisation.
17. Le serveur de signalisation envoi l'adresse relayé de client au MCU2.
18. MCU2 envoi les frames vidéo de la conférence au participant à partir de serveur TURN.

3.3 Éléments à sécuriser :

3.3.1 Découverte du système :

Avant de lancer des attaques, l'attaquant a besoin d'avoir une idée sur l'ensemble du système et d'identifier les applications qui y tournent. Pour chaque application découverte, le pirate cherche ses vulnérabilités et tente de les exploiter en lançant des attaques adéquates.

L'attaque qui permet d'identifier les systèmes s'appelle un balayage de systèmes. Il en existe différents types :

Balayage ICMP :

Le balayage *ICMP* consiste à envoyer un *ping (icmp request)*à l'ensemble des machines susceptibles d'être présentes sur le réseau. A chaque fois qu'on reçoit une réponse au *ping (icmp response)* cela fera une machine active dans le réseau. On peut tenter le *ping* sur chaque adresse *IP* comme on peut faire un *ping* directement sur une adresse de *broadcast.*

Balayage TCP :

Le balayage *TCP* consite à envoyer un paquet *SYN* sur chaque port d'une machine distante afin de découvrir s'il y a une application qui y fait l'écoute.

Balayage semi-ouvert :

Le balayage semi-ouvert ressemble au balayage *TCP*, seulement il ne respecte pas le *handshake* pour ne pas donner à la cible la main pour journaliser l'événement.

3.3.2 Les attaques possibles :

Une fois que l'attaquant à fait le balayage du système, il peut réussir une ou plusieurs attaques des attaques qu'on a cité dans le chapitre 2.

Usurpation ARP (ARP spoofing) :

L'attaque ARP Spoofing peut toucher aux :

Serveur de signalisation.

Conférencier.

Participant.

MCUs.

Serveur Stun.

la figure 3.3 montre le cas où le pirate vise le MCU1. Il procède de la manière suivante :

1. Déterminer les adresses IP de MCU1 et conférencier.
2. Envoyer une requête *ARP* non sollicitée au MCU1, pour l'informer du changement de l'adresse MAC de conférencier à son adresse MAC.
3. Désactive la vérification des adresses MAC sur la machine d'attaque.

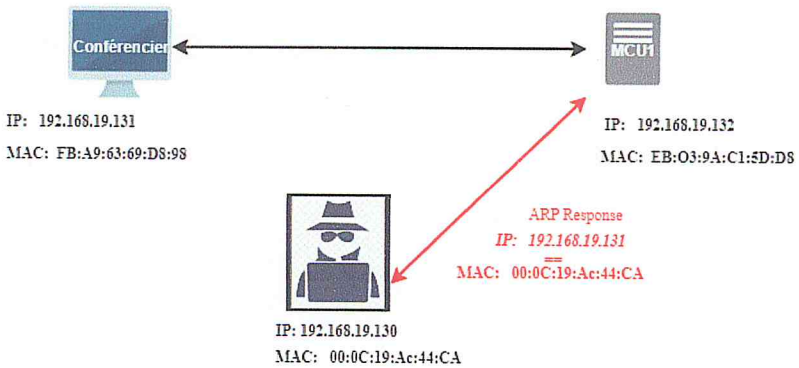

FIGURE 3.3 – Schéma de l'attaque arp spoofing

Homme du milieu (Man in the middle) :

Une personne malhonnête réussit l'attaque de l'homme au milieu à plusieurs niveau dans le système de vidéoconférence représenté par l'architecture qui se trouve dans la figure 3.1, et ce, dans le but d'avoir des informations, soit sur la signalisation, ou sur les flux multimédia.

Le pirate peut s'interposer entre :

* Le serveur de signalisation et le MCU.
* Le serveur de signalisation et le conférencier(participant).
* Le MCU1 et le MCU2.
* Le serveur Stun et le participant.

Une façon pour mener à bien une attaque de l'homme au milieu est d'effectuer l'attaque d'ARP spoofing sur les deux entités communicantes. Pour bien comprendre, nous allons détailler le cas ou le pirate s'interpose entre le conférencier et le MCU1.

Comme le montre La figure 3.3. Le pirate a l'adresse MAC 00 :0C :19 :AC :44 :CA, le MCU1 l'adresse MAC 00 :0C :29 :4A :A4 :41 et le conférencier a l'adresse MAC FB :A9 :63 :69 :D8 :98.

Si le pirate envoie des paquets au MCU1 avec l'adresse IP du source du conférencier mais en laissant son adresse MAC (ce qui est faisable si on construit nos paquets nous même plutôt que si on laisse notre carte réseau le faire), la table ARP de MCU1 va donc enregistrer le couple suivant :(192.168.1.31 , 00 :0C :19 :AC :44 :CA).

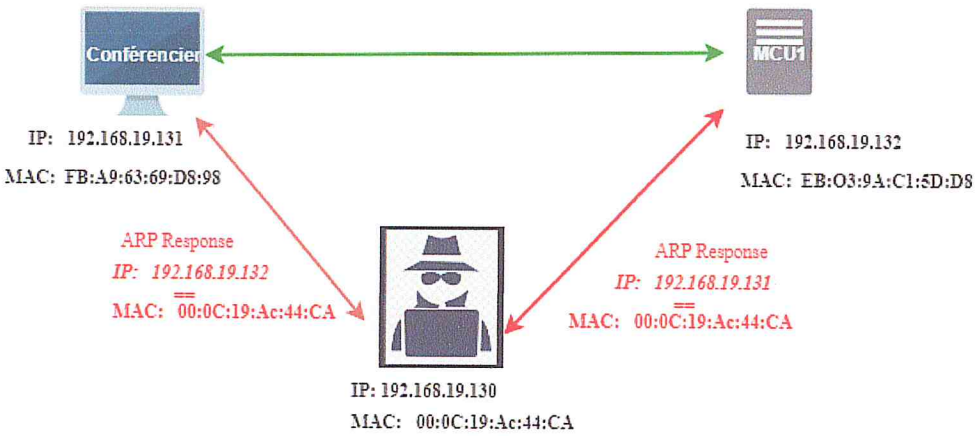

FIGURE 3.4 – Schéma final de l'attaque Man In The Middele

Cela vient du fait que l'enregistrement est marqué comme dynamique, donc volatile, et qu'il peut être mis à jour à chaque réception de paquet présentant des couples IP-MAC différents.

Dans l'autre partie, il envoie des paquets au conférencier avec l'adresse IP du source du MCU1 mais en laissant son adresse MAC, la table ARP de conférencier va donc enregistrer le couple suivant :(192.168.1.32 , 00 :0C :19 :AC :44 :CA)(voire figure 3.4).

On voit donc que si les tables ARP de MCU1 et du conférencier sont falsifiées, il va former ces trames avec l'adresse IP du MCU1 mais va en fin de compte les envoyer au

pirate car il formera ses requêtes avec comme adresse MAC de destination celle du pirate, et les trames formées avec l'adresse IP du conférencier vont être envoyer au pirate.

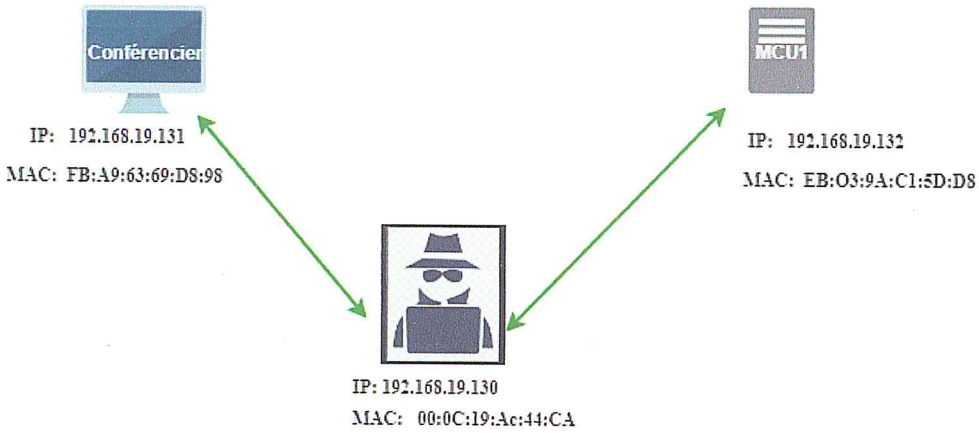

FIGURE 3.5 – Schéma final de l'attaque Man In The Middele

Déni de service :

L'attaque de déni de service peut toucher au serveur de signalisation, les serveurs *Stun*, les MCUs.

Nous allons voir le cas ou l'attaque déni de service touche au serveur de signalisation '

Une personne malhonnête réussit cette attaque afin de paralyser le serveur de signalisation.

Deux paradigmes peuvent être employés. Le premier paradigme consiste à utiliser une adresse *IP* d'une entité légitime (MCU, Conférencier, Participant). Quant au deuxième paradigme, il consiste en l'utilisation d'une adresse *IP* non valide. Ces deux cas seront détaillés dans ce qui va suivre.

Paradigme 1 :

Dans ce scénario, le pirate se sert de l'adresse *IP* de MCU1 pour inonder le serveur de signalisation avec une grande quantité de paquet *SYN*. Il envoie des paquets *TPC* avec comme adresse source l'adresse *IP* de MCU1 et comme adresse *IP* destination celle de serveur de signalisation. Le serveur de signalisation quand

il reçoit cette demande *SYN* qui parait parvenir de MCU1, alloue les ressources nécessaires pour l'établissement d'une connexion *TCP* avec le MCU1 et il lui envoie une réponse *SYN-ACK*. A ce niveau nous avons deux cas de figures

1. ***Cas1*** :

   Le MCU1 est toujours actif dans le réseau. Dans ce cas, il reçoit un *SYN-ACK* inattendu de la part de serveur de signalisation. Il répond par un *RST* ce qui mettrait fin à la connexion en attente et libèrerait les ressources vu qu'un *RST* est plus rapide qu'un message de négociation de fin de connexion.

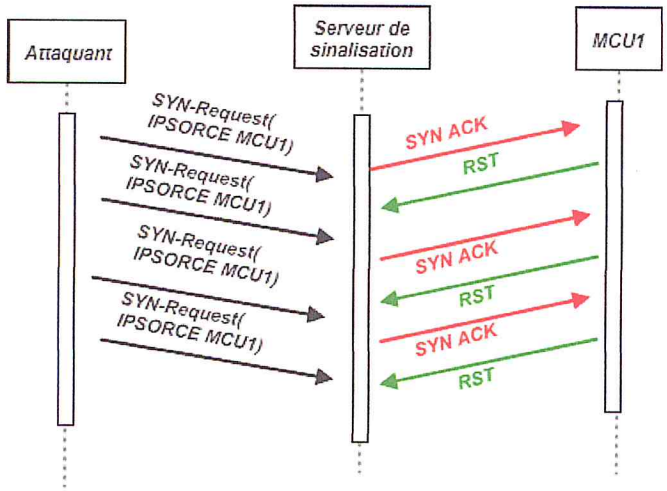

FIGURE 3.6 – Utilisation d'adresse IP de MCU1

Comme l'illustre la figure 3.6, quelque soit le nombre de *SYN*envoyés, le fait que le MCU1 soit toujours active et réponds par des messages *RST* qui libère les ressources alloués par le serveur de signalisation empêchera l'attaque de se réaliser correctement.

2. ***Cas1*** :

   Nous avons vu lors du premier cas que l'attaque *SYN Flooding* ne peut pas se réaliser correctement, et cela dû au fait que le MCU1 est toujours actif et répond par des messages *RST* qui libère les ressources allouées par le serveur de signalisation.

Pour mener à bien cette attaque, on doit empêcher le MCU1 à envoyer ces messages, c'est-à-dire déconnecté le MCU1 du réseau, ce qui nécessite un accès physique.

Ce qui est évident, c'est qu'un pirate n'a pas un accès physique au MCU1. Une autre alternative sera la mise hors service de MCU1 en le saturant, et ceci, peut se réaliser en lui envoyant des messages *ICMP* massif.

Même dans ce cas, MCU1 peut répondre avec un message *RST* au message *SYN-ACK* inattendu parvenu de serveur de signalisation ce qui ne permettra pas de réussir l'attaque.

Paradigme 2 :

Dans ce cas, le pirate forge des paquets *TCP* avec une adresse source non valide, et les envoie vers le serveur de signalisation. A la réception de ces paquets, le serveur de signalisation réserve des ressource pour établir une nouvelle connexion et ajout cette demande à sa liste d'écoute , comme illustré dans la figure Figure 5, tout en envoyant un message *SYN-ACK* à cette adresse qui n'est pas valide. Il est claire que dans ce cas, le serveur de signalisation ne va pas recevoir un message *SYN-ACK* de pirate , qui tente de l'inonder avec des demande *SYN* en les envoyant pendant une durée qui soit inférieur au *timeout*, et ce, dans le but d'avoir plusieurs semi-connexion ouverte et saturer la liste d'écoute de serveur de signalisation et le rendre incapable à traiter les demandes parvenant des utilisateurs légitimes.

Pour contrer à l'attaque DOS par inondation *SYN*, on propose de configurer le serveur de signalisation ainsi que les serveurs *Stun* d'une façon à limiter le nombre de connexions pour la même source et de libérer les connexions semi-ouvertes après un délai raisonnable.

DoS distribue (Distributed DoS) :

Tout comme dans le cas d'une attaque *DoS*, Le *DDoS* peut toucher aux :

* Serveur de signalisation.
* Conférencier.
* Participant.
* MCUs.
* Serveur Stun.

Nous allons prendre comme exemple de victime le serveur de signalisation , et comme moyen d'attaque, le serveur Stun.

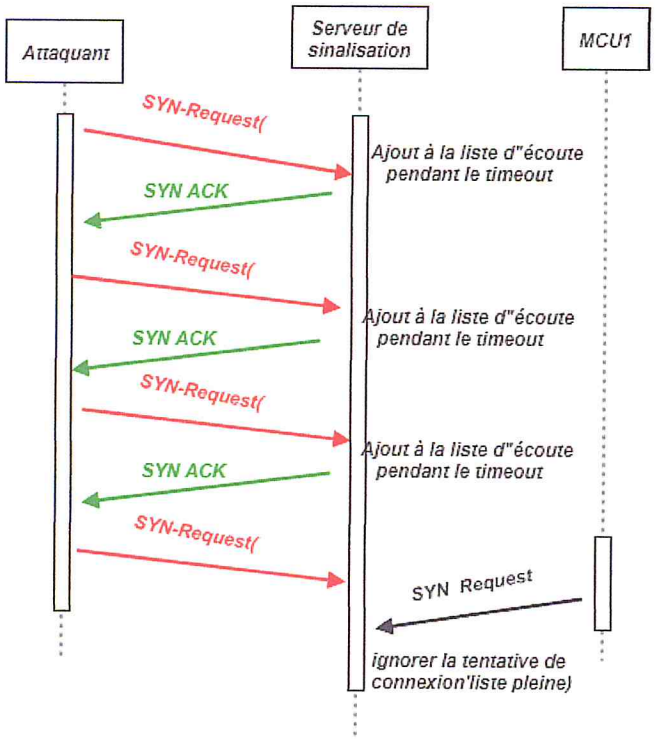

FIGURE 3.7 – Inondation par message SYN avec une adresse source non valide

Nous avons vu lors du premier chapitre que le serveur Stun permet aux participants et aux conférencier de savoir quelle adresse IP publique qu'ils exposent. Si jamais un pirate réussit à prendre le contrôle d'un serveur Stun.A chaque fois qu'un participant ou un conférencier demande son dresse IP publique, le pirate lui répond en lui envoyant l'adresse publique de serveur de signalisation.

Remarquez bien que quand les participant et les conférencier communiquent l'adresse IP publique fournit par le serveur Stun et ui est celle de serveur de signalisation, donc tous les paquets se dirigent vers cette adresse. Au bout d'une certaine durée le serveur de signalisation devient hors service, et ce dû u fait qu'il est submergé d'un trafic provenant de plusieurs source.

DNS Spoofing :

L'attaque DNS Spoofing peut toucher aux :

* Serveur de signalisation.
* MCUs.
* Serveur Stun.

Une personne malhonnête se sert du nom du domaine de serveur de signalisation pour rediriger tout le trafic vers un autre serveur de signalisation pour lequel il prend le contrôle.

Dans ce qui va suivre, nous allons parler de quelques attaques spécifique au serveur Stun et au serveur Turn.

3.3.3 Attaque liées aux serveurs STUN :

Muet un client :

La figure 3.8 montre le schéma d'une attaque Muet un client. Dans cette attaque, l'attaquant fournit à un client STUN (participant ou conférencier) une adresse réflexive (ou publique) falsifié en interceptant la communication entre le client STUN et les serveurs STUN (1,2). Il remplace l'adresse privée de client avec une adresse falsifiée. Le serveur STUN répond à la demande de client en envoyant une adresse publique falsifiée(3). Le client fournit son adresse réflexive falsifiée au serveur de signalisation(4), ce dernier a son tour envoie cette adresse au MCU (5). Cette adresse réflexive qu'il lui a été fournie est une adresse qui ne mène nulle part(6). En conséquence, le client ne recevra aucun des paquets qu'il s'attend à recevoir lorsqu'il distribue l'adresse réflexive au serveur de signalisation. Car, le serveur de signalisation envoi cette adresse au MCU et le MCU envoi les paquets l'adresse qui n'existe pas.

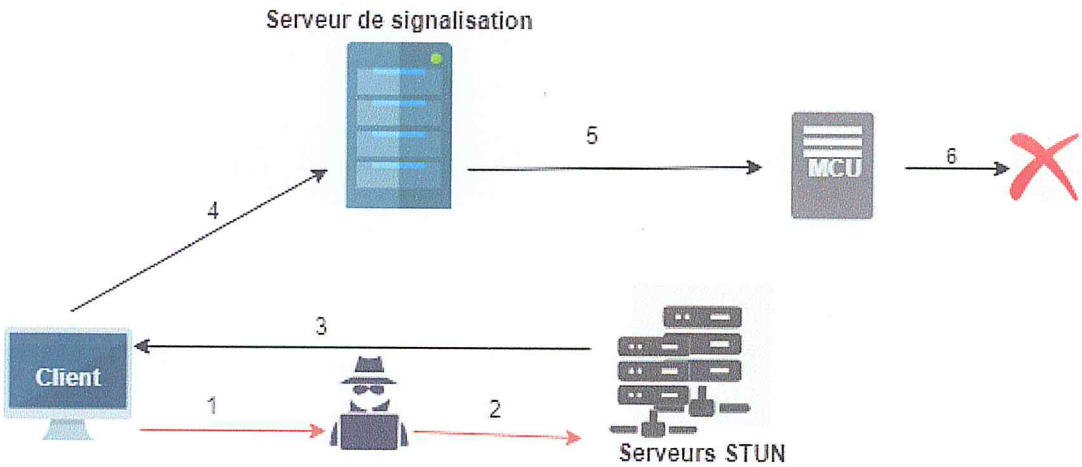

FIGURE 3.8 – Muet un client

Usurpation de l'identité d'un client :

L'attaque de l'usurpation de l'identité d'un client illustrée par la figure 3.9 est similaire à l'attaque Muet un client. Cependant, l'adresse réflexive falsifiée pointe vers l'attaquant lui-même. Cela permet à l'attaquant de recevoir le trafic destiné au client(6).

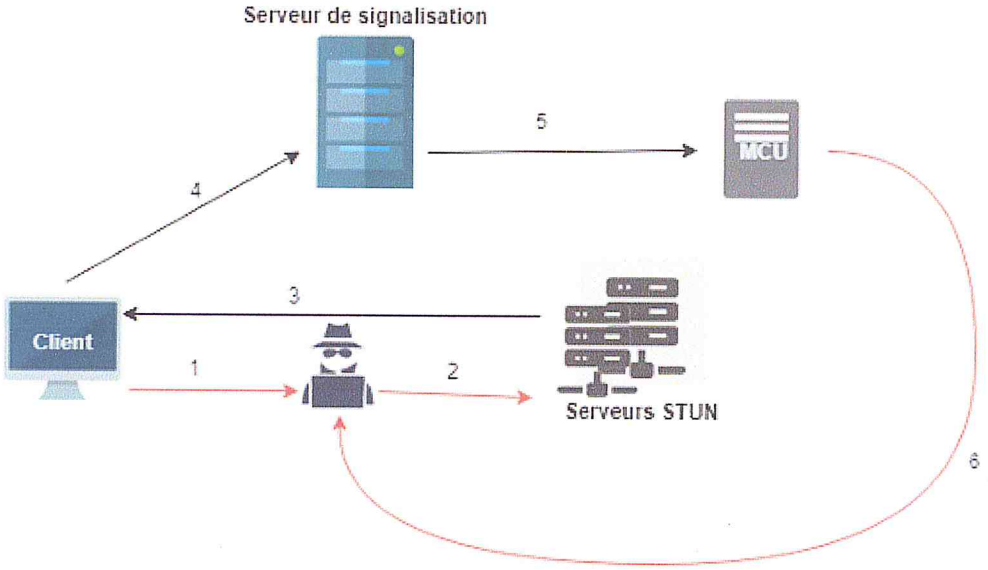

FIGURE 3.9 – Usurpation de l'identité d'un participant

Eavesdropping (L'écoute) :

L'attaquant dans une attaque *Eavesdropping* oblige le client à utiliser une adresse réflexive qui se dirige vers lui-même (voire figure 3.10). Il transmet ensuite tous les paquets qu'il reçoit au client (7). Cette attaque permettrait à l'attaquant d'observer tous les paquets envoyés au client. Cependant, pour lancer l'attaque, l'attaquant doit avoir déjà pu observer les paquets du client vers le serveur STUN.

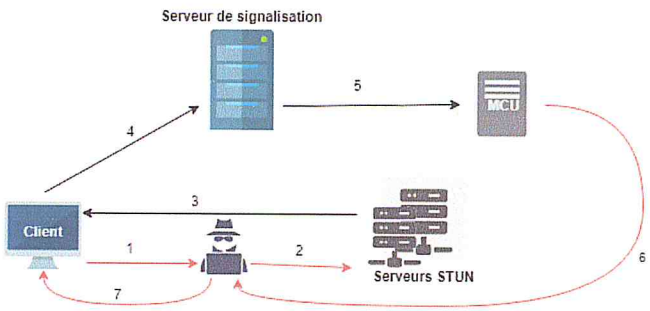

FIGURE 3.10 – l'écoute clandestine

L'utilisation de TLS et DTLS pour sécuriser les communication entre le client et les serveur STUN peut nous protéger contre ces attaques.

3.3.4 Attaque liées aux serveurs Turn :

Écoute de trafic (Eavesdropping Traffic) :

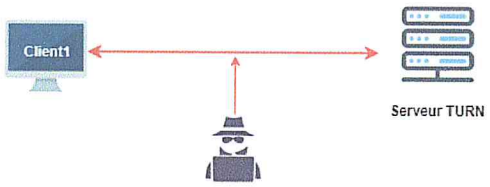

FIGURE 3.11 – l'écoute du trafic

Le protocole TURN se préoccupe principalement de l'authentification et de l'intégrité des messages. La confidentialité n'est qu'une préoccupation secondaire, car les messages de contrôle TURN n'incluent pas d'informations particulièrement sensibles.

pour empêcher l'écoute sur une connexion TURN Le contenu des messages de contrôle est l'adresse IP de client TUR(voire figure 3.11 , TLS doit être employé.

Concernant la confidentialité des données relayées par TURN elles sont assurées par le protocole d'application lui-même (par exemple RTP) car l'exécution de TURN sur TLS ne protège pas les données multimédia entre TURN et le client. Pour assurer la confidentialité des données multimédia nous devons chiffrer les données afin de les protéger, pour les données multimédias en temps réel, la confidentialité peut être assurée en utilisant SRTP.

Attaques de dictionnaire hors ligne :

Le mécanisme d'identification utilisé par TURN est soumis à des attaques de dictionnaire hors ligne. Un attaquant capable d'espionner un échange de messages entre un participant et un serveur STUN peut déterminer le mot de passe en essayant un certain nombre de mots de passe candidats et en vérifiant si l'un d'entre eux est correct. Cette attaque fonctionne lorsque les mots de passe sont d'une entropie faible, comme un mot du dictionnaire. Cette attaque peut être atténuée en utilisant des mots de passe forts avec

une grande entropie. Dans les situations où une atténuation encore plus importante est nécessaire, le transport TLS entre le client et le serveur peut être utilisé.

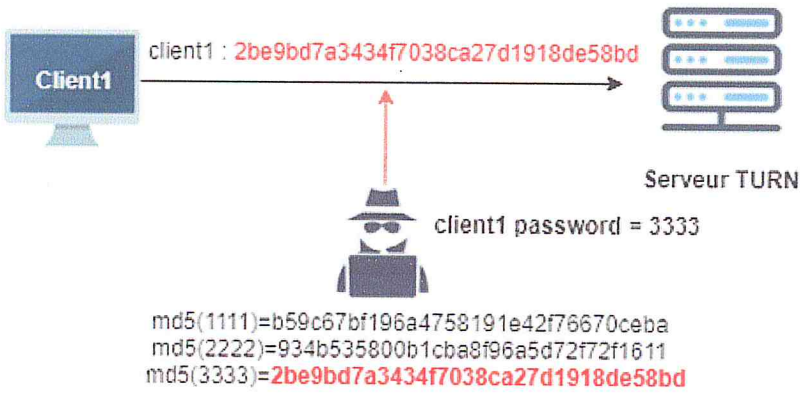

FIGURE 3.12 – Attaque par dictionnaire hors ligne

3.3.5 Loop attack :

Un attaquant pourrait tenter de faire boucler indéfiniment des paquets de données entre deux serveurs TURN. L'attaque procède comme suit. Tout d'abord, l'attaquant envoie une requête *Allocate* au serveur A, en utilisant l'adresse source du serveur B. Le serveur A enverra sa réponse au serveur B, et pour que l'attaque réussisse, l'attaquant doit avoir la possibilité de voir ou deviner le contenu de cette réponse, afin que l'attaquant puisse apprendre l'adresse de transport relayée allouée. L'attaquant envoie alors une requête *Allocate* au serveur B, en utilisant l'adresse source du serveur A. Encore une fois, l'attaquant doit pouvoir voir ou deviner le contenu de la réponse, de sorte qu'il puisse envoyer l'adresse de transport relayée allouée.

Utilisant la même technique d'adresse de source usurpée, l'attaquant lie alors un numéro de canal sur le serveur A à l'adresse de transport relayée sur le serveur B, et lie de même le même numéro de canal sur le serveur B à l'adresse de transport relayée sur le serveur A. envoie un message *ChannelData* au serveur A.Le résultat est un paquet de données qui passe de l'adresse de transport relayée sur le serveur A à l'adresse de transport relayée sur le serveur B, puis de l'adresse de transport du serveur B à l'adresse de transport du serveur A, puis à nouveau autour de la boucle.

Cette attaque est atténuée comme suit en imposant une limite par nom d'utilisateur sur la bande passante utilisée pour relayer les données par des allocations appartenant

à ce nom d'utilisateur, afin de limiter l'impact de cette attaque sur d'autres allocations.

3.3.6 DoS contre le serveur TURN :

Un participant souhaitant perturber le service à d'autres participant, il peut obtenir une allocation, puis l'inonder de trafic. Le but est de submerger le serveur et de l'empêcher de traiter d'autres utilisateurs légitimes

Ceci est atténué par la recommandation que le serveur limite la quantité de la bande passante qu'il relaie pour un nom d'utilisateur donné. Cela n'empêche pas un client d'envoyer une grande quantité de trafic, mais il permet au serveur d'ignorer immédiatement le trafic en excès.

Comme chaque allocation utilise un numéro de port sur l'adresse IP du serveur TURN, le nombre d'allocations sur un serveur est fini. Donc, un attaquant pourrait tenter de tous les consommer en demandant un grand nombre d'allocations. Ceci est empêché par la recommandation que le serveur impose une limite le nombre d'allocations actives à la fois pour un utilisateur donné.

3.4 Solution générale

Les attaques cités avant ne peuvent tirées partit si les communications entre les différents composants du système sont chiffrées. De ce fait, on exige l'utilisation du protocole TLS qui utilise les certificats pour sécuriser la signalisation, Et le protocole DTLS, qui n'est rien d'autre qu'une adaptation du protocole TLS pour sécuriser les paquets UDP, et ceci, pour sécuriser les flux multimédia qui transitent sur le système de vidéoconférence.

Une mise en évidence de *TLS/SSL handshaking* est indispensable pour compréhension de la solution proposée.

Le *TLS/SSL handshaking* peut être fait de deux façons, qui sont, *One-way TLS/SSL*et le *Two-way TLS.*

1. **One-way TLS/SSL handshaking :**

   La figure 3.13 montre le TLS/SSL handshaking pour l'authentification unidirectionnelle entre un client TLS et un serveur TLS.

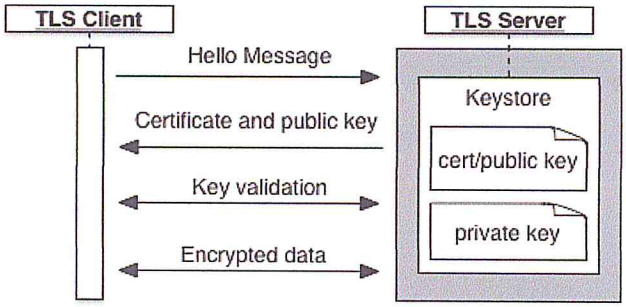

FIGURE 3.13 – One-way handshake

Dans une configuration TLS unidirectionnelle, les messages échangés entre un client et un serveur sont :

* **Hello message :** Le client envoie une requête de session au serveur.
* **Certificate and public key :**Le serveur répond avec un certificat, qui contient sa clé publique. Ce certificat provient du keystore du serveur, qui contient également la clé privée du serveur. La clé privée n'est jamais envoyée au client, pour un certificat signé, le client demande à l'autorité de certification (CA) d'authentifier le certificat.
* **Key validation :** Le client et le serveur échangent plusieurs autres messages pour valider les clés.
* **Encrypted data :** Le client commence le transfert de données TLS avec le serveur.

* **Two-way TLS handshake :**

La figure 3.14 montre le TLS/SSL handshaking pour l'authentification bidirectionnelle entre un client et un serveur :

Dans ce scénario :

Si le serveur TLS utilise un certificat auto-signé ou un certificat qui n'est pas signé par une autorité de certification approuvée, on doit crée un fichier de clés certifiées sur le client appelé truststore. Le client a une copie du certificat du serveur dans son fichier de clés certifiées. Lors de handshaking, le client compare le certificat dans son truststore avec le certificat envoyé par le serveur pour vérifier l'identité du serveur.

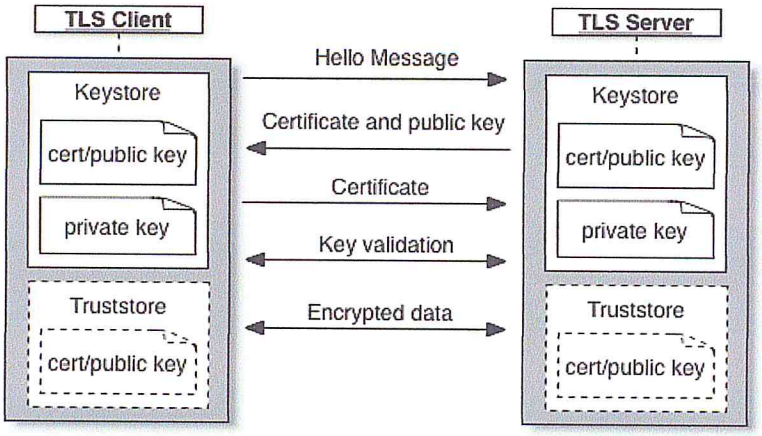

FIGURE 3.14 – two-way handshake

Si le client TLS utilise un certificat auto-signé ou un certificat qui n'est pas signé par une autorité de certification approuvée, un fichier truststore doit être crée sur le serveur. Le serveur possède une copie du certificat du client dans son truststore. Pendant le handhsaking, le serveur compare le certificat dans son truststore avec le certificat envoyé par le client pour vérifier l'identité du client.

Server certificate et client certificate :

L'utilisation de certificat de serveur et certificat de client se diffère en ce qui suit :

Les certificats clients, comme leur nom l'indique, sont utilisés pour identifier un client. Ils sont destinés à authentifier le client sur le serveur, la clé de certificat client est utilisé juste pour l'authentification, tanque la clé de certificat de serveur est utilisé dans l'authentification et le chiffrement.

3.5 Conclusion :

Dans ce chapitre, nous avons présenté le système de vidéoconférence à sécuriser. Nous avons vu les attaques qui peuvent y touché. Nous avons proposé l'utilisation de protocole TLS et DTLS pour réduire ou éliminer complètement ces problèmes.

Dans le prochain chapitre, nous allons implémenter et tester cette solution.

Chapitre 4

Réalisation du système

4.1 Introduction

Pour satisfaire nos objectifs de sécurité, on a choisi d'implémenter le protocole *TLS* sur *TCP* et le protocole *DTLS* sur *UDP* pour assurer l'authenticité du serveur de signalisation en utilisant les certificats, et la confidentialité des données échangées en utilisant le chiffrement, aussi l'intégrité des données échangé et l'authentification du client en ajoutant l'option de certificat de client.

4.2 Architecture générale du système

Notre système de sécurisation des canaux de communication pour les webinaires se déploie sur des canaux sécurisés par les protocoles TLS et DTLS. Ces canaux sont choisis pour leurs avantages cités dans le chapitre précédent (chapitre trois) .

Les échanges entre les différents clients et le serveur de signalisation s'effectuent par *socketchannel* (un canal connecté à un socket TCP) avec le protocole *TLS 1.2* ce qui est basé sur TCP pour garantir le transfert des données de façons fiable et en toute sécurité en garantissant la confidentialité en chiffrant les données circulant dans ses canaux.

Cependant, les échanges entre les participants ou les conférenciers avec les MCUs s'effectuent par *datagramchannel* (un canal qui peut envoyer et recevoir des paquets UDP) avec le billet de protocole *DTLS 1.2*, ce choix a été pris pour la diffusion des conférences

car le DTLS est basé sur UDP ce qui offre une performance en termes de délais car en peut tolérer des pertes de paquet dans une communication en vidéo temps réel.

Notre système est composé de quatre applications, l'application du serveur de signalisation, l'application de MCU et celle du conférencier et participant relié par des canaux sécurisés.

* L'application de serveur de signalisation représente le cœur du système ou toutes les applications communiquent avec lui via des *socketchannel TLS*.
* L'application MCU représente le lien entre deux applications conférencier et MCU ou MCU et participants.
* L'application participant et l'application conférencier permettant de voir et de lancer une conférence.
* Fichier Keystore Contient le certificat et la clé privée de chaque entité de notre système, utilisée pour identifier l'entité lors d'une communication TLS entre le serveur de signalisation et les autres applications, et DTLS pour la communication entre les participants ou les conférenciers avec les MCUs.
* Fichier Truststore qui contient tous les certificats des entités qui font part de notre système. Ils sont des certificats auto-signés dans notre plateforme de teste.
* Des certificats auto-signé pour chaque entité de notre système, créé par l'utilité de keytool.

4.3 Environnement et outils du développement

Nous avons utilisé des outils différents qui sont :

4.3.1 Partie matérielle :

Un ordinateur portable ayant comme processeur i5-7200 CPU et RAM 8 GO avec un système d'exploitation de 64 bits et *Windows* 10 professionnel.

4.3.2 Partie logicielle :

Langage de programmation java version 9 :

Java est un langage de programmation orienté objet développé par *Sun Microsystems* (aujourd'hui racheté par Oracle), Une de ses plus grandes forces est son excellente portabilité, la version 9 de java est sorti en juillet 2017.

Environnement de développement :

NetBeans nightly build :

Netbeans nightly ou NetBeans IDE Build 201804200002 est un environnement de développement intégré (IDE) qui permet aux utilisateurs d'écrire et d'exécuter plus facilement des programmes Java. Contrairement à netbeans 8.1, neatbeans nightly build supporte la version 9 de java ce qui est utilisé dans notre implémentation de DTLS.

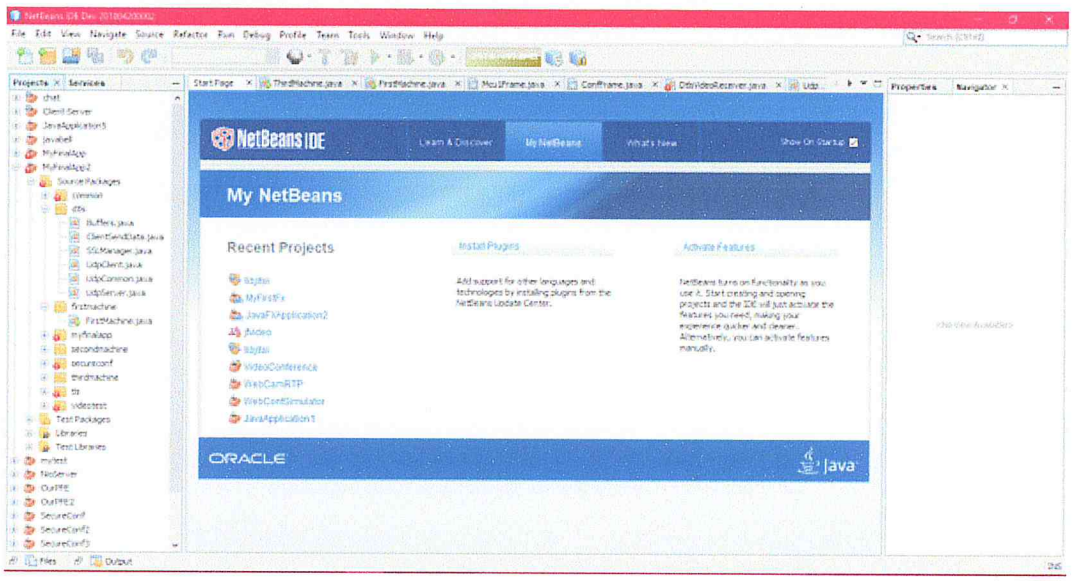

FIGURE 4.1 – L'interface netbeans

JSSE :

L'extension Java Secure Socket (JSSE) permet la création des communications Internet sécurisées. Elle fournit une structure et une implémentation pour une version Java des protocoles SSL et TLS. Également, elle inclut des fonctionnalités

pour le cryptage des données, l'authentification du serveur, l'intégrité des messages et l'authentification de client. Grâce à JSSE, on peut assurer le passage sécurisé des données entre un client et un serveur indépendamment des protocoles applicatif, tel que HTTP (Hypertext Transfer Protocol), Telnet ou FTP, via TCP / IP. Tout cela en faisant l'abstraction des algorithmes de sécurité sous-jacents complexes et les mécanismes de « handshaking ». Le JSSE minimise le risque de créer des vulnérabilités de sécurité subtiles et dangereuses. De plus, elle simplifie le développement d'applications en servant de bloc de construction que les développeurs peuvent intégrer directement dans leurs applications.

L'API JSSE est capable de prendre en charge la version SSL 3.0 et 1.0, 1.1 et 1.2. Ces protocoles de sécurité encapsulent un flux bidirectionnel normal, ainsi que l'API JSSE ajoute un support transparent pour l'authentification, le cryptage et la protection de l'intégrité. L'implémentation JSSE fournie avec le JDK prend en charge SSL 3.0, TLS (1.0, 1.1 et 1.2) et DTLS (versions 1.0 et 1.2). Cependant elle ne prend pas en charge SSL 2.0.

Keytool : La génération et la manipulation des clés sont effectuées à l'aide de l'utilitaire keytool, qui se trouve dans le kit JDK standard. keytool est un utilitaire de gestion de clés et de certificats qui permet aux utilisateurs d'administrer leurs propres paires de clés publiques/privées et certificats.

Ici, nous allons l'utiliser pour créer des paires de clés publiques/privées, et pour extraire les certificats de clé publique de ces paires et les placer dans leurs propres fichiers.

4.4 Présentation des interfaces :

Dans ce qui va suivre, nous allons présenter les interfaces de notre système de vidéoconférence.

4.4.1 Interface de serveur de signalisation :

L'interface de serveur de signalisation représentée dans la figure 4.2, offre un aperçue sur les différents clients qui ce connecte au serveur de signalisation, elle contient un *buton*

start qui permet la mise en ligne de serveur de signalisation et un *buton deconnect* qui permet de le déconnecter.

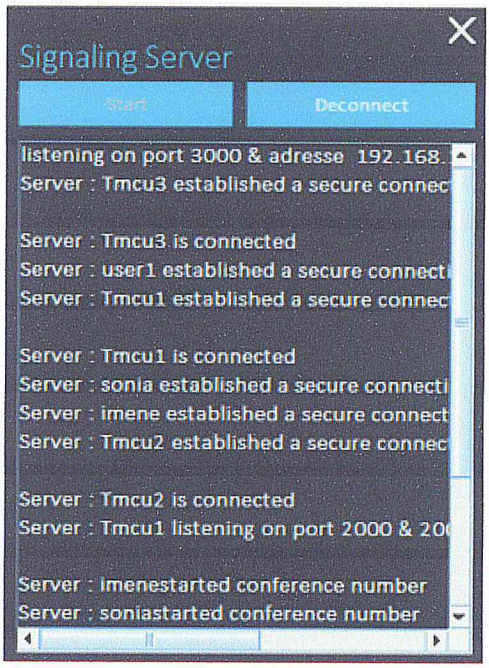

FIGURE 4.2 – L'interface de serveur de signalisation

4.4.2 Interface login :

L'interface logine qui se trouve dans la figure 4.3, permet l'authentification d'un conférencier ou d'un participant au niveau de serveur de signalisation.

4.4.3 Interface de MCU :

Cette interface (figure 4.4) montre un aperçu sur les conférences qui passe par lui. Le button listen permet de recevoir des connexions, le button connect permet de connecter au prochain entité MCU ou participant pour lui transférer les données.

Interface de conférencier :

L'interface du conférencier(figure 4.5) permet au conférencier de lancer un conférence en appuyant sur le bouton *start conference.*

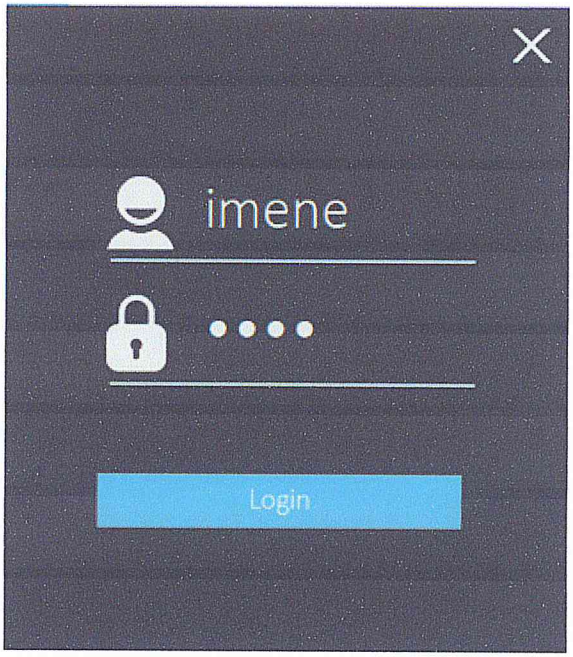

FIGURE 4.3 – L'interface Login

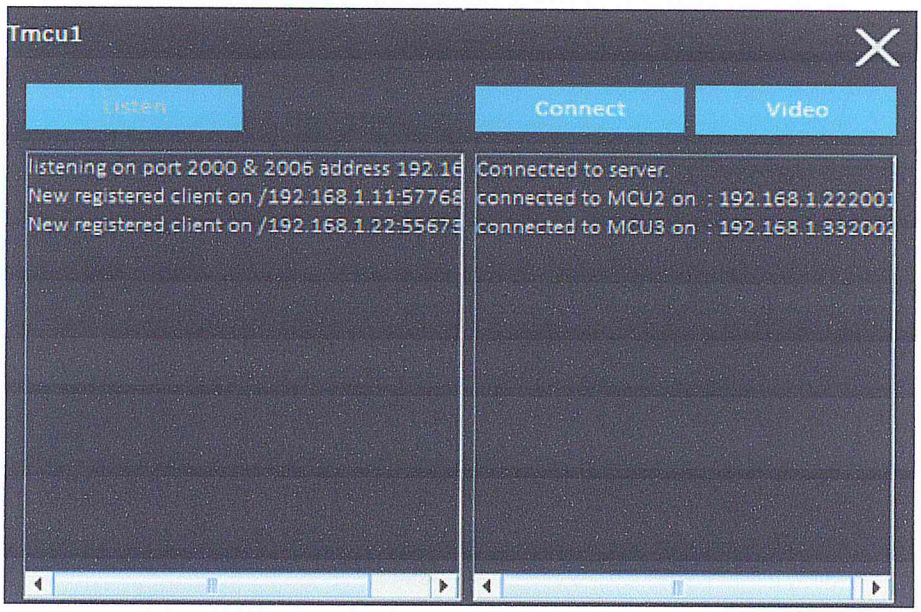

FIGURE 4.4 – L'interface de MCU

4.4.4 Interface de participant :

L'interface participant qui se trouve dans la figure 4.6 permet au participant de voir une conférence en appuyant sur le bouton *button watch conference.*

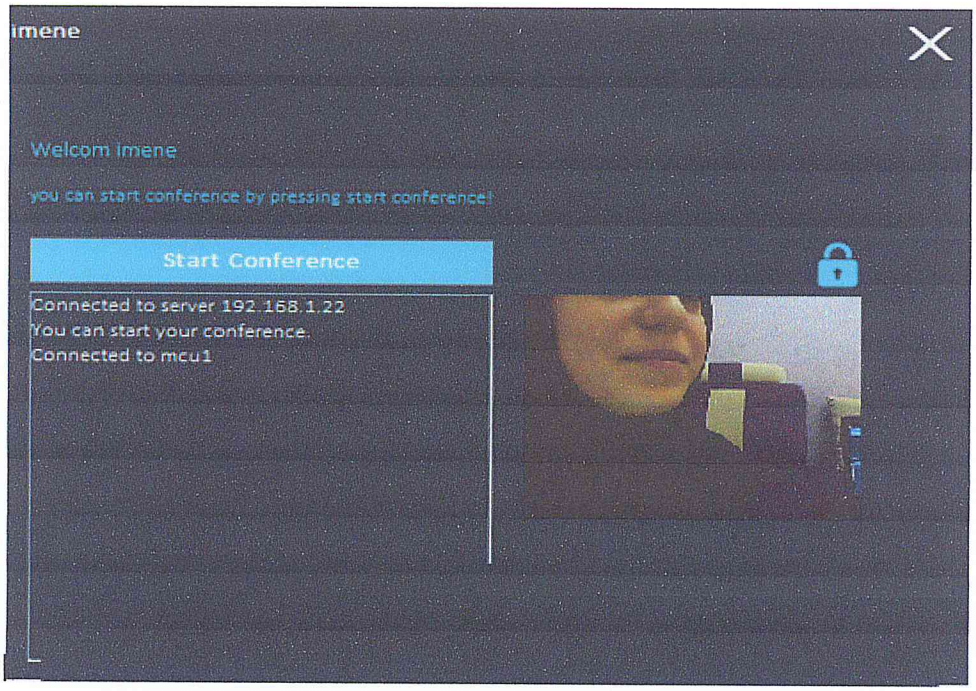

FIGURE 4.5 – L'interface de conférencier

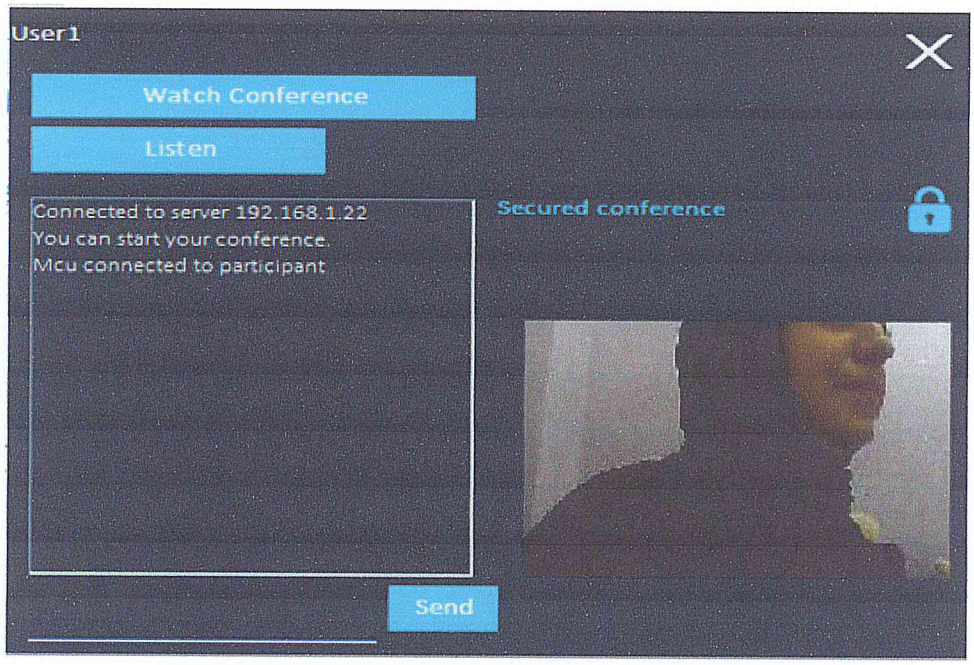

FIGURE 4.6 – L'interface de participant

4.5 Scénario et tests du système :

Dans cette partie nous illustrons les étapes d'un scénario de test qui explique le fonctionnement de notre système.

4.5.1 Outils de tests :

wireshark :

Wireshark (dont son interface est représentée par la figure 4.7) est un logiciel d'analyse réseau (sniffer) qui permettant de visualiser l'ensemble des données transitant sur la machine qui l'exécute, et d'obtenir des informations sur les protocoles applicatifs utilisés. Les octets sont capturés en utilisant la librairie réseau *PCAP*[1] , puis regroupés en blocs d'informations et analysés par le logiciel.

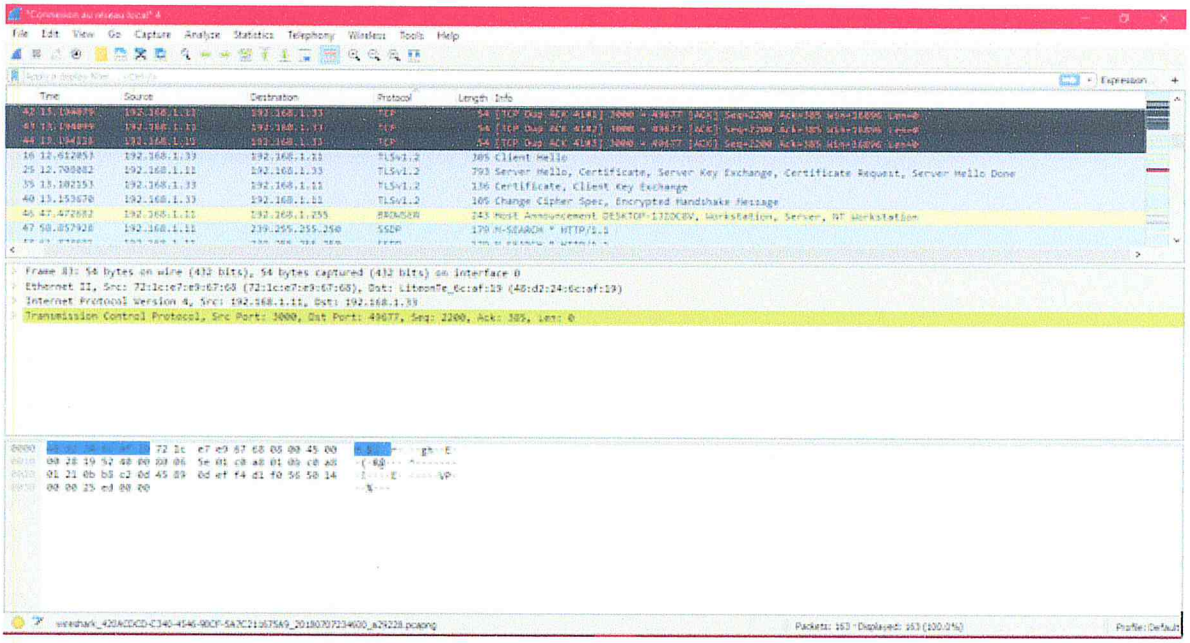

FIGURE 4.7 – L'interface de wireshark

Architecture Globale de test :

La figure 4.18 illustre un exemple de scénario de tests du système de sécurisation des canaux dans ce qui suit nous détaillerons les étapes de ce scénario.

1. Le serveur de signalisation est actif et attend l'arrivée des clients sur l'adresse 192.168.1.11 et le port 3000.
2. Les trois MCU arrivent et se connecte via TLS au serveur de signalisation à son adresse et port.
3. Les deux conférenciers connectent au serveur de signalisation via TLS.

1. https ://fr.wikipedia.org/wiki/Wireshark

4. Les trois MCU se mettre en écoute sur les adresses 192.168.1.11 pour le MCU 1 et 192.168.1.22 pour le MCU 2 et 192.168.1.33 pour le MCU 3.
5. Le conférencier1 connecte au MCU 1 à l'adresse 192.168.1.11 et port 2000.
6. Un canal DTLS est ouvert entre le conférencier1 et le MCU1 et le conférencier commence la diffusion de vidéo.
7. Un canal DTLS est ouvert entre le conférencier 1 et le MCU1 et le conférencier commence la diffusion de vidéo.
8. Les participants se connectent au serveur de signalisation par TLS et choisissent de voir la conférence.
9. Les participants reçoivent les frames vidéo sécurisés par le protocole DTLS.

4.5.2 La gestion des clés :

La première étape de la préparation de notre test consiste à générer une paire de clés publique/privée et des certificats auto signées pour chaque entité de notre système (Serveur de signalisation, conférencier, MCU, Participant) et les stockés dans un fichier keystore en utilisant la commande suivante :

keytool -genkey -keyalg RSA -keysize 2048 -validity 360 -alias mykey -keystore myKeyStore.jks -genkey/-genkeypair : Génère une paire de clés (une clé publique et une clé privée associée).

keyalg :

Le nom de l'algorithme utilisé pour générer la clé. La valeur RSA signifie qu'on a utilisé l'algorithme RSA.

keysize : La taille en bits de la clé à générer. Normalement, les tailles de clé sont des multiples de 8 qui s'alignent avec un nombre d'octets. **validity :**Le nombre de jours pendant lesquels le certificat attaché à la paire de clés doit être valide. **alias :**Le nom dans le KeyStore Java la clé générée doit être identifié par. Un alias ne peut pointer que vers une seule clé. **keystore :**Nom du fichier KeyStore dans lequel on va stocker la paire de clés générée. Si le fichier n'existe pas, il sera créé.

```
C:\Users\User\Desktop\certificates>keytool -genkey -keyalg RSA -keysize 2048 -validity 360 -alias SSkey -keystore SSkeystore.jks
Entrez le mot de passe du fichier de clés :
Ressaisissez le nouveau mot de passe :
Quels sont vos nom et prénom ?
  [Unknown]:
```

FIGURE 4.8 – Création d'un keystore pour le serveur de signalisation

La figure 4.8 représente l'exemple de serveur de signalisation :

Lorsque nous exécuterons cette commande, on nous posera une série de questions (votre nom, organisation, et autres). Les informations que nous fournissons seront utilisées pour créer un certificat auto-signé qui associe l'information avec une clé, il nous sera également demandé d'entrer des mots de passe pour le fichier de clés et, éventuellement, des mots de passe pour la paire de clés que nous avons créés.

```
C:\Users\User\Desktop\certificates>keytool -genkey -keyalg RSA -keysize 2048 -validity 360 -alias SSkey -keystore SSkeystore.jks
Entrez le mot de passe du fichier de clés :
Ressaisissez le nouveau mot de passe :
Quels sont vos nom et prénom ?
  [Unknown]:  SS
Quel est le nom de votre unité organisationnelle ?
  [Unknown]:  PFE
Quel est le nom de votre entreprise ?
  [Unknown]:  Saad dahleb
Quel est le nom de votre ville de résidence ?
  [Unknown]:  blida
Quel est le nom de votre état ou province ?
  [Unknown]:  us
Quel est le code pays à deux lettres pour cette unité ?
  [Unknown]:  us
Est-ce CN=SS, OU=PFE, O=Saad dahleb, L=blida, ST=us, C=us ?
  [non]:  oui
```

FIGURE 4.9 – Les informations fournies pour l'obtention d'un certificat

deuxième étape consiste à exporter les certificats contenant la clé publique à partir de keystore dans un fichier.cert avec la commande suivante :

keytool -export -alias mykey -keystore myKeyStore.jks -file mykey.cert

Lorsque nous exécuterons cette commande un fichier *.cert*sera crée et dans ce fichier on aura le certificat de notre entité dans l'exemple de la figure 4.9 c'est le serveur de signalisation.

Cette opération nécessite d'entrer le mot de passe de fichier créé précédemment pour pouvoir y accéder.

La troisième et dernière étape est de créer un fichier truststore pour stocker nos cer-

```
C:\Users\User\Desktop\certificates>keytool -export -alias SSkey -keystore SSKeyStore.jks -file SSkey.cert
Entrez le mot de passe du fichier de clés :
Certificat stocké dans le fichier <SSkey.cert>
```

FIGURE 4.10 – La création d'un certificat

tificats dedans avec la commande suivante :

keytool -import -file mykey.cert -alias mykey -keystore myTrustStore.jks

```
C:\Users\User\Desktop\certificates>keytool -import -file SSkey.cert -alias SSkey -keystore myTrustStore.jks
Entrez le mot de passe du fichier de clés :
Ressaisissez le nouveau mot de passe :
Propriétaire : CN=ss cert, OU=pfe, O=saad dahleb, L=blida, ST=us, C=us
Emetteur : CN=ss cert, OU=pfe, O=saad dahleb, L=blida, ST=us, C=us
Numéro de série : 51235605
Valide du Thu Jun 14 17:48:27 CEST 2018 au Sun Jun 09 17:48:27 CEST 2019
Empreintes du certificat :
        SHA 1: F8:7A:05:7C:F2:26:58:4E:49:58:13:91:83:D1:22:0D:91:7E:C8:0D
        SHA 256: DF:58:E8:A5:F9:B1:1D:31:C5:69:07:16:6A:A0:20:F0:52:7F:75:E1:06:36:56:3A:A5:52:E7:A3:35:57:F1:79
Nom de l'algorithme de signature : SHA256withRSA
Algorithme de clé publique du sujet : Clé RSA 2048 bits
Version : 3

Extensions :

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B3 45 AD C7 2E 28 D3 16   BF 7F ED 97 54 5F B5 92  .E...(......T_..
0010: 6F C5 22 26                                        o."&
]
]

Faire confiance à ce certificat ? [non] :  oui
Certificat ajouté au fichier de clés
```

FIGURE 4.11 – L'ajout de certificat de serveur de signalisation au truststore

4.5.3 Test de confidentialité :

En utilisant wireshark, on a capturé les paquets de données qui passent en temps réel sur notre réseau dans les deux cas suivants :

1. **CAS1 : UDP sans protocole DTLS :**

   Dans le cas d'utilisation de protocole UDP sans DTLS l'attaquant peut facilement voir les données échangées comme le montre la figure 4.12 suivante :

2. **Cas 2 : UDP avec le protocole DTLS implémenté :**

   Les paquets circulant dans notre système sont cryptés par le protocole DTLS version 1.2, donc un attaquant ne peut rien voire des données échanger entre les différentes entités.

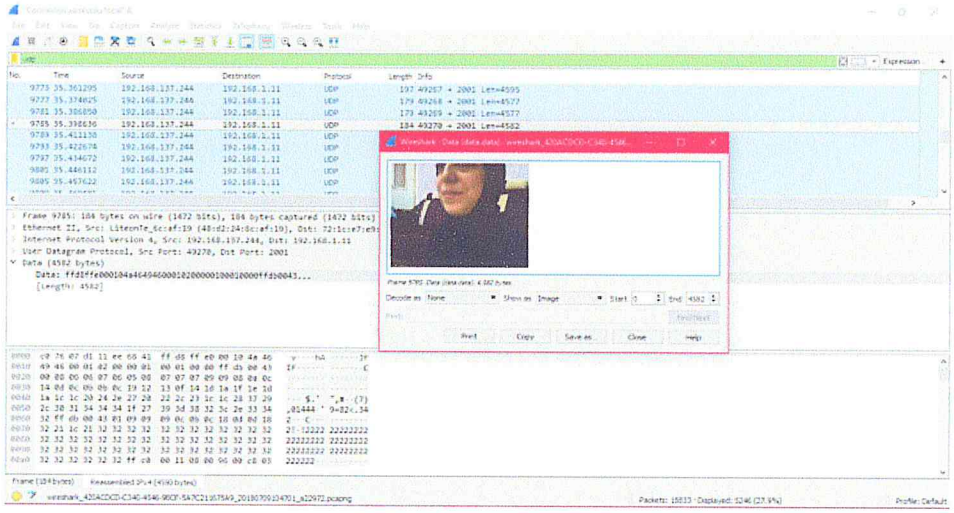

FIGURE 4.12 – capture des paquets UDP sans le protocole DTLS

la figure 4.13 montre le résultat d'interecption des paquets UDP Sécurisés pas DTLS .

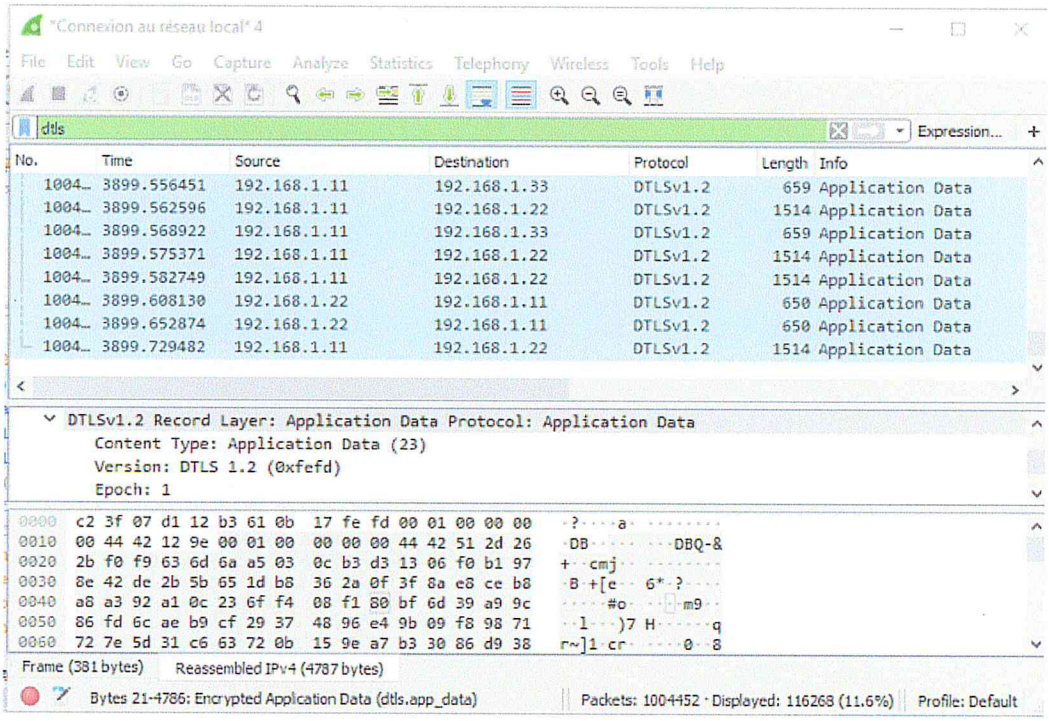

FIGURE 4.13 – capture des paquets UDP sécurisés avec DTLS

4.5.4 Test de mécanisme de protection d'identité :

1. **Cas 1 : entité connue au niveau de serveur de signalisation :**

Pour ce test on affect un certificat légitime qu'il est inclus dans le truststore à une entité, ici le MCU3, la figure 4.14 montre le code qu'il nous a permet de donner le keystore légitime.

```
String keystorepathmcu3 = "C:\\Users\\s-com\\Desktop\\certificates\\mcu3keystore.jks";
//String keystorepathmcu3 = "C:\\Users\\s-com\\Desktop\\certificates\\TestCert.jks";
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        //video case
        Mcu1Frame mcu3 = new Mcu1Frame("Tmcu3", keystorepathmcu3);
        mcu3.setVisible(true);

    }
});
```

FIGURE 4.14 – Affectation de keystore légitime

variable keystorepathmcu3 sera affecté au keystore avec la fonction keyStore.load(). Maintenant on essaye de connecter au serveur et on observe le comportement de

```
//load the client's keystore (public and private keys + certificate )
KeyStore keyStore = KeyStore.getInstance("JKS");
//keyStore.load(new FileInputStream("C:\\Users\\User\\Desktop\\certificates\\C1keystore.jks"), "12
keyStore.load(new FileInputStream(keystorefile), "123456".toCharArray());
KeyManagerFactory keyFact = KeyManagerFactory.getInstance("SunX509");
keyFact.init(keyStore, "123456".toCharArray());

//load the trustStore (the file containing all the trusted certificates
KeyStore trustStore = KeyStore.getInstance("JKS");
trustStore.load(new FileInputStream("C:\\Users\\s-com\\Desktop\\certificates\\myTrustStore.jks"),
TrustManagerFactory trustFact = TrustManagerFactory.getInstance("SunX509");
trustFact.init(trustStore);

//Initialize the context
context = SSLContext.getInstance("TLSv1.2");
```

FIGURE 4.15 – Affectation de keystore légitime2

serveur et de l'entité, et les paquets échangés entre l'entité et le serveur avec wireshark (voire figure 4.16) : Puisque on a implémenté le two-way TLS, on peut observer dans les étapes de handshake capturé par wireshark, que le serveur envoi son certificat avec un server hello, après le client envoi son certificat après la vérification de certificat de serveur. La communication réussie car l'identité de serveur est vérifiée au niveau de client et l'identité de client est vérifier au niveau de serveur de signalisation.

2. **Cas 2 : entité non connue au niveau de serveur de signalisation :**

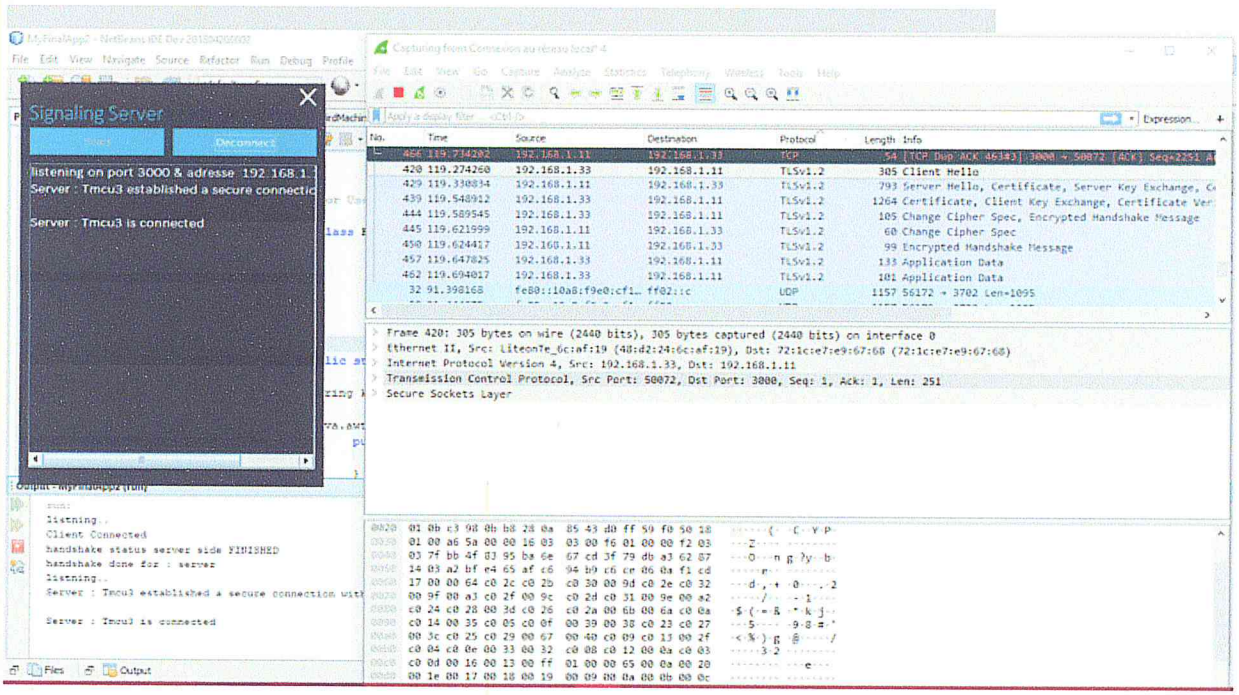

FIGURE 4.16 – capture des paquets entre le serveur et l'entité

Pour ce test on affect un certificat qu'il n'est pas inclus dans le truststore a une entité est on essaye de connecter avec cette entité au serveur de signalisation.

Le code qu'il nous a permet de donner le keystore illégitime est illustré dans la figure 4.17.

```
//String keystorepathmcu2 = "C:\\Users\\User\\Desktop\\certificates\\mcu2keystore.jks";
// String keystorepathmcu3 = "C:\\Users\\s-com\\Desktop\\certificates\\mcu3keystore.jks";
 String keystorepathmcu3 = "C:\\Users\\s-com\\Desktop\\certificates\\TestCert.jks";
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        //video case
        Mcu1Frame mcu3 = new Mcu1Frame("Tmcu3", keystorepathmcu3);
        mcu3.setVisible(true);

    }
});
```

FIGURE 4.17 – Affectation de keytore illégitime

Maintenant on essaye de connecter au serveur et on observe le comportement de serveur et de l'entité, et les paquets échangés entre l'entité et le serveur avec wireshark : Dans ce cas on peut observer dans les étapes de handshake capturé par wireshark, que le handshake entre le serveur de signalisation et l'entité illégitime

est interrompu (voire figure **??**, et la communication échoue car le certificat de l'entité n'est pas reconnu par le serveur et cela nous garantit l'authentification des clients au niveau de serveur de signalisation.

4.6 Conclusion :

Au fil de ce dernier chapitre, nous avons présenté la mise en œuvre de notre solution des canaux de sécurité en choisissant deux protocoles, le protocole TLS pour les données échangées entre les clients et le serveur de signalisation, et DTLS pour les frames vidéo échangées entre les MCU et les conférenciers et les participants.

Ainsi, nous avons montré l'efficacité de l'approche en capturant les paquets échangés entre les entités pour assurer la confidentialité, et en essayant un certificat illégitime pour tester l'authentification. Ce qui assure la confidentialité des données et l'authentification des clients.

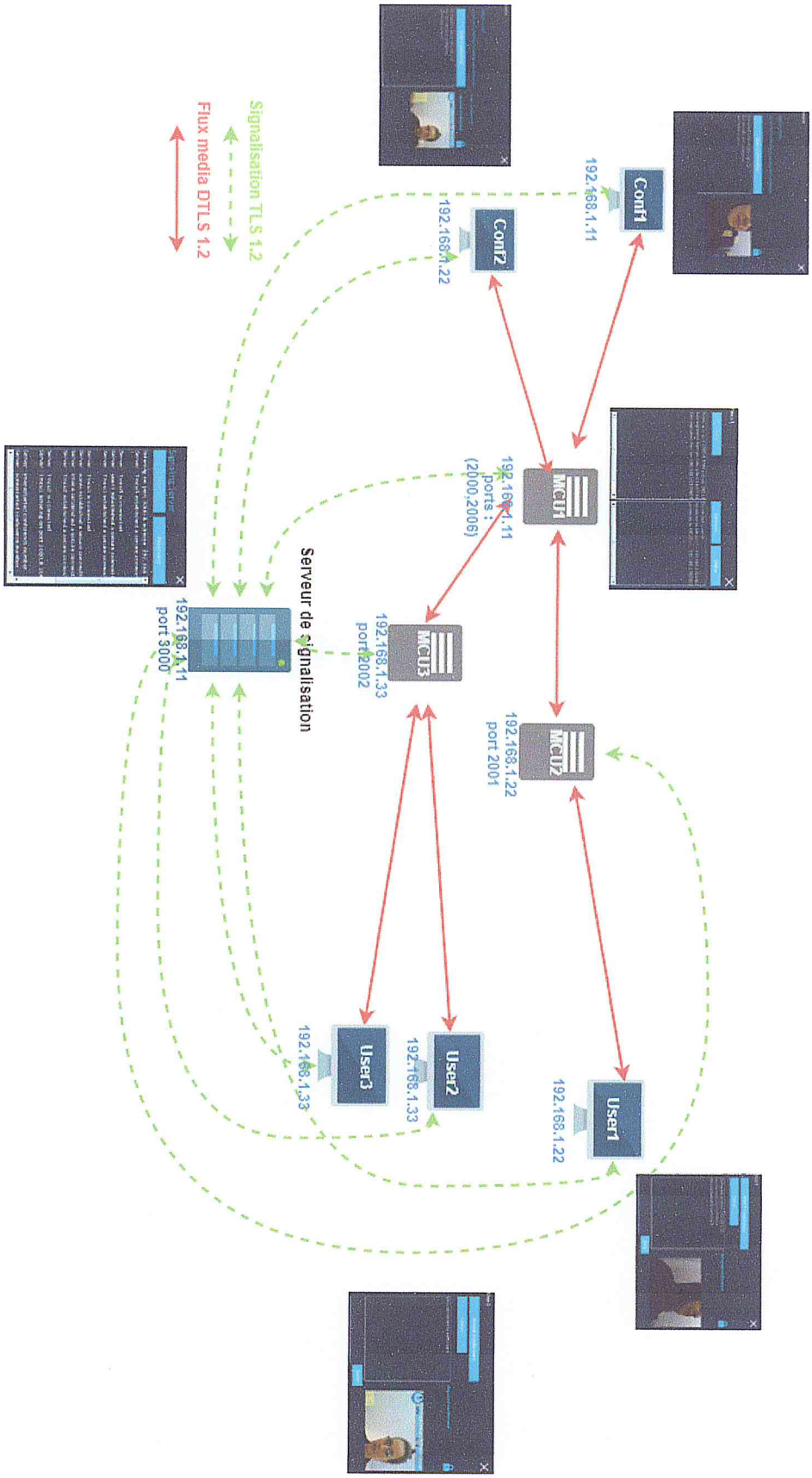

FIGURE 4.18 – L'architecture globale de test

Conclusion générale et perspectives

Ce travail a été réalisé dans le cadre de la sécurisation d'un système de vidéoconférence déployé par les universités algériennes.

Plusieurs phases ont été accompli avant d'arriver à ces lignes.

La première phase, se trouve dans le premier chapitre , était d'étudier l'existant par rapport aux solutions de vidéoconférence qui existe dans la littérature, ce qui nous a permet de comprendre le fonctionnement de ces systèmes. Nous avons vu que le fonctionnement de ces derniers s'entoure généralement sur deux choses. La première étant l'initialisation de la communication, la deuxième étant le transfert de flux multimédia. Nous avons présenté le protocole TCP utilisé pour initier une communication entre un conférencier et un participant qui se fait par l'intermédiaire d'un serveur nommé *serveur de signalisation* Nous avons aussi vu, qu'une fois la communication est initié, des données multimédias seront échangé entre le participant et le conférencier par le bais des entités nommées *MCU* dont le rôle est de diffuser les flux multimédias aux participants (UDP,RTP ,RTCP).Nous avons parlé d'une des solution de vidéoconférence qui est le WebRTC.

Le deuxième chapitre reflète la deuxième phase de ce travail.Nous avons basé, dasn ce chapite, sur l'aspect sécurité dans les systèmes de vidéoconférence. Nous avons vu que les propriétés de sécurité (la confidentialité, l'intégrité, la disponibilité la non-répudiation et l'authentification) sont à la base de l'expression des besoins de sécurité. Nous avons aussi parlé des violations de ces propriétés par des attaques comme le dénni de service, l'usurpation de l'identité E t nous avons conclus ce chapitre par quelque mécanismes de sécurité (comme le chiffrement des données, l'utilisation des certificats) qui sont employé dans des protocoles tel quel TLS et DTLS qui peuvent servir pour sécuriser la signalisation

et le transfert des flux multimédias dans une vidéoconférence.

Dans le chapitre trois, nous avons exposé l'architecture de la plateforme de vidéoconférence sur laquelle nous avons agit. Nous avons expliqué son fonctionnement par un schéma de focntionnement qui met en évidence les interactions entre les entités qui la composent. Nous avons décelé les problèmes de sécurité qui peuvent touché à ces composantes (à titre d'exemple l'attaque déni de service qui paralyse le serveur de signalisation). Et vers la fin de ce chapitre, nous avons proposé de faire appel aux prtocol TLS pour chiffrer l'étape de signalisation,et aux protocole DTL pour chiffrer le flux multimédia, et ceci, dû au fait que la plupart des attaques citées sont dû au fait que les donnée échangées sont en claire.

Dans le chapitre quatre, nous avons présenté les différentes phases de mise en œuvre de la solution proposée dans le chapitre trois, allant des outils de développement arrivant jusqu'à la phase de test par laquelle nous avons pu prouvé que la solution proposée répond au travail demandé et aux objectifs fixés au début.

Cependant, des améliorations peuvent être apporter, et d'autre solutions peuvent être employé pour augmenter la sécurité de ce système. On recommande l'utilisation de protocole SRTP pour sécuriser les flux multimédias transportés par le protocole RTP.

On recommande l'élaboration d'une politique de sécurisé pour assurer la sécurité la totalité de ce système (la sécurité physique des serveurs, la sécurité logicielle, le contrôle d'accès).

Bibliographie

[1] Laurent Ouakil and Guy Pujolle. *Téléphonie sur IP : SIP, H. 323, MGCP, QoS et sécurité, Asterisk, VoWiFi, offre multiplay des FAI, Skype et autres softphones, architecture IMS...* Editions Eyrolles, 2011.

[2] Xavier Carcelle. *Réseaux CPL par la pratique : Avec trois études de cas : réseau domestique, réseau d'entreprise et réseau de desserte de collectivité locale.* Editions Eyrolles, 2011.

[3] R. Vápeník, M. Michalko, J. Janitor, and F. Jakab. Secured web oriented videoconferencing system for educational purposes using webrtc technology. In *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 495–500, Dec 2014.

[4] F. Lehner, W. Mazurczyk, J. Keller, and S. Wendzel. Inter-protocol steganography for real-time services and its detection using traffic coloring approach. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pages 78–85, Oct 2017.

[5] H. Bostani and J. C. Grégoire. Usable authentication systems for real time web-based audio/video communications. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pages 117–121, March 2017.

[6] Christer Jakobsson. Peer-to-peer communication in web browsers using webrtc a detailed overview of webrtc and what security and network concerns exists, 2015.

[7] Michel Buffa, Alain Giboin, and Thierry Bergeron. Etat de l'art sur les techniques de transfert data/audio/vidéo basées Web. working paper or preprint, March 2015.

[8] Eric Rescorla and Nagendra Modadugu. Rfc 6347, datagram transport layer security version 1.2. *Internet Engineering Task Force*, 2012.

[9] R. Madhu and B. Neelima. Performance analysis of dtls protocol. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pages 331–334, July 2017.

[10] A. Hussein, I. H. Elhajj, A. Chehab, and A. Kayssi. Securing diameter : Comparing tls, dtls, and ipsec. In *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pages 1–8, Nov 2016.

[11] Z. Linlin, L. Weimin, Z. Wei, and L. Shaowei. The implementation of a secure rtp transmission method based on dtls. In *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pages 379–383, Sept 2013.

[12] Nasser Adnane, Nabyl Mersel, Wahiba Larbi, et al. *Etude et Mise en Place D'une Solution VoIP Sécurisée Cas d'étude.* PhD thesis, Universite de bejaia, 2017.

[13] A. Dakur and S. Dakur. Eavesdropping and interception security hole and its solution over voip service. In *2014 IEEE Global Conference on Wireless Computing Networking (GCWCN)*, pages 6–10, Dec 2014.

[14] Cédric Llorens, Laurent Levier, Denis Valois, and Benjamin Morin. *Tableaux de bord de la sécurité réseau.* Editions Eyrolles, 2011.

[15] J Rosenberg, R Mahy, P Matthews, and D Wing. Rfc 5389 : Session traversal utilities for nat (stun). *Internet Engineering Task Force (IETF), Tech. Rep*, 2008.

[16] R Mahy, P Matthews, and J Rosenberg. Traversal using relays around nat (turn) : Relay extensions to session traversal utilities for nat (stun) rfc 5766. *Internet Society Request for Comments*, page 6, 2010.

[17] G Bisiaux and B Rappachi. Déploiement de la visioconférence ip dans un établissement. *Etat de l'art et évolution des protocols.*

[18] Sourour MEHAROUECH. Optimisation de la fiabilité et la pertinence des systèmes de détection et prévention d'intrusions. 2010.

[19] Joelle Roué. *Analyse de la résistance des chiffrements par blocs aux attaques linéaires et différentielles.* PhD thesis, Université Pierre Et Marie Curie, 2015.

[20] Bernard Boutherin and Benoît Delaunay. *Sécuriser un réseau Linux.* Editions Eyrolles, 2006.

[21] Raghav Bhaskar. *Protocoles cryptographique pour les réseaux mobile Ad-Hoc.* PhD thesis, Ecole Polytechnique, 2005.

[22] Olivier Sanders. *Conception et optimisation des mécanismes cryptographiques anonymes.* PhD thesis, Ecole Doctorale de Science Mathématique de Paris Centre, 2015.

[23] Laurent Bloch, Christoph Wolfhugel, Christian Queinnec, Hervé Schauer, Florence Henry, and Nat Makarévitch. Sécurité informatique. *Principes et méthodes. Eyrolles, 276p*, 2007.

[24] Jérémy Jean. *Cryptanalyse de primitives symétriques basées sur le choffrement AES.* PhD thesis, Université Paris Diderot(Pris 7), 2013.

[25] Mickaek Salaun. *Intégration de l'utilisateur au contrôle d'accès : du processus cloisonné à l'interface homme-machine de confiance.* PhD thesis, Ecole Doctorale Informatique Télécommunication et Eectronique (Paris), 2008.

[26] Christina Bourra. *Analyse de fonction de hachage cryptographiques.* PhD thesis, Université de Pierre et Marie Curie, 2012.

[27] Hsiao-Ying Lin, Wen-Guey Tzeng, et al. Anonymous password based authenticated key exchange with sub-linear communication. *Journal of Information Science and Engineering*, 25(3) :907–920, 2009.

[28] Jöelle MUSSET. Sécurité informatique : Ethical hacking : Apprendre l'attaque pour mieux se défendre, 2009.

[29] Amandine Jambert. *Outils cryptographiques pour la protection des contenus de la vie privée des utilisateurs.* PhD thesis, Université Bordeaux 1, 2011.

[30] Aroua Biri. *Proposition de nouveaux mécanismes de protection contre l'usurpation d'identité pour les fournisseurs de service Internet.* PhD thesis, Université Pierre et Marie Curie, 2011.

[31] P. Segeč, M. Moravčík, J. Hrabovský, J. Papán, and J. Uramová. Securing sip infrastructures with pki; the analysis. In *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 1–8, Oct 2017.

[32] Ahmed Aouadi. *Mise en place d'une solution open Source VoIP et Visioconférence multi-sites sécurisée.* PhD thesis, Université Virtuelle de Tunis, 2015.

[33] Reza Neshat. Implementation of security for a video-conferencing system management module, 2015.

[34] A. L. Alexander, A. L. Wijesinha, and R. Karne. An evaluation of secure real-time transport protocol (srtp) performance for voip. In *2009 Third International Conference on Network and System Security*, pages 95–101, Oct 2009.

[35] Jeetendra Pande. Introduction to cyber security. 2017.

[36] Bozzini David. *Cryptographie et surveillance digitale*, chapter 12. Musée d'ethnographie, 2016.

[37] Zakaria Kaddouri. Mise en oeuvre de nouvelles techniques pour la sécurité informatique basées sur les algorithmes évolutionnistes et les fonctions de hachage. 2014.

[38] Nazim Benaissa. *la composition des protocoles de sécurité avec méthode B événementielle.* PhD thesis, Université de Nancy, 2010.

[39] S Demba. *Courbes el liptiques, Cryptographie à clés publiques et Protocoles déchange de clés.* PhD thesis, PhD thesis, Université Cheikh Anta DIOP de Dakar, 2013.

[40] Marion Videau. *Critères de sécurité des algorithmes de chiffrement à clé secrète.* PhD thesis, Université Pierre et Marie Curie-Paris VI, 2005.

[41] Pascal Lafourcade. *Vérification de protocoles cryptographiques en présence de théories équationnelles.* PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2006.

[42] Omary Fouzia. *Application des algorithmes évolutionnistes à la cryptographie.* PhD thesis, Université Mohammed Rabat, 2006.

[43] Nadia El Mrahbet. *Arithmétique des couplages performance et résistance aux attaques par canaux cachés.* PhD thesis, Université Montpellier 2, 2009.

[44] Davide Alessio. *About some topics in cryptology : Revocable Anonymity and a Generalization of Goldwasser-Micali cryptoscheme.* Theses, Université Rennes 1, December 2011.

[45] Thomas Plantard. *Arithmétique modulaire pour la cryptographie.* Theses, Université Montpellier II - Sciences et Techniques du Languedoc, December 2005.

[46] Jérémy Métairie. *Contributions aux opérateurs arithmétiques GF* (2^m) *et leurs applications à la cryptographie sur courbe elliptiques.* PhD thesis, Université de Rennes 1, 2016.

[47] Thomas Plantard. *Arithmétique modulaire pour la cryptographie.* PhD thesis, Université Montpeller 2, 2006.

[48] Julien Eynard. *Approche arithmétique RNS de la cryptographie asymétrique.* PhD thesis, Université de Pierre et Marie Curie Paris 5, 2015.

[49] Damien Vergnaud. *Primitives et construction en cryptographie asymétrique.* PhD thesis, Ecole Normale Supérieure Paris Centre, 2014.

[50] Gael Thomas. *Design et Analyse de sécurité pour les constructions en cryptographie symétrique.* PhD thesis, Limoges, 2015.

[51] Aude Plateaux. *Solutions opérationnelles d'une transaction électronique sécurisée et respecteuse de la vie privée.* PhD thesis, Université de Caen Basse-Normadie, 2014.

[52] Julien Devigne. *Protocoles de re-chiffrement pour le stockage de données.* PhD thesis, Université de Caen, 2013.

[53] Hua Jiang, Xianru Du, Yongxing Jia, and Weizhi Wang. An identity-based security mechanism for p2p voip. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pages 481–485, June 2010.

[54] A. Singh and C. Singh. Analysis of security threats and protocols for lte networks. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 1660–1663, Aug 2017.

[55] Tim Dierks and Eric Rescorla. Rfc 5246 : The transport layer security (tls) protocol. *The Internet Engineering Task Force*, 2008.

[56] Didier Capozzi. *Visioconférence, une nouvelle façon de communiquer.* Haute école de gestion de Genève (HEG-GE), 2011.

[57] Mark Baugher. The secure real-time transport protocol (srtp). *IETF RFC 3711*, 2004.

[58] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi. Detecting internet abuse by analyzing passive dns traffic : A survey of implemented systems. *IEEE Communications Surveys Tutorials*, pages 1–1, 2018.

[59] Julen Lollve. *Entrelacement des mécanismes d'identification et de respect de la vie privée pour la protection des contenus externalisés.* PhD thesis, Ecole Doctorale Matisse, 2016.

[60] Jean-Gabriel Kammerer. *Analyse de nouvelles primitives cryptographiques pour les schémas Diffie-Hellman.* PhD thesis, Université de Rennes 1, 2013.

[61] Marion Daubignard, David Lubicz, and Graham Steel. A Secure Key Management Interface with Asymmetric Cryptography. Research Report RR-8274, INRIA, 2013.

[62] Evmorfia-Iro Bartzia. *A formalization of elliptic curves for cryptography.* Theses, Université Paris-Saclay, February 2017.

[63] J. Flohr and E. P. Rathgeb. Rosiee : Reduction of self inflicted queuing delay in webrtc. In *2017 29th International Teletraffic Congress (ITC 29)*, volume 3, pages 7–12, Sept 2017.

Annexe A

Glossaire du protocole TCP/IP

IP Internet Protocol :Le support de travail des protocoles de la couche de transport TCP et UDP.

TCP Transport Control Protocol : de transport des datagrammes IP en mode connecté.

UDP User Datagram Protocol : Protocol de transport des datagrammes IP en mode non connecté ou mode datagramme.

Adresse IP : nombre de 32 bits (4 octets) permettant l'identification d'un hôte et le réseau auquel il appartient. Chaque adresse IP contient deux informations adresse du réseau et adresse d'hôte.

Classe A : L'adresse IP est formée d'un octet pour identifier le réseau, trois octets pour les hôtes. La plage d'adresses varie de 1.x.y.z à 127.x.y.z.

Classe B : L'adresse IP est formée de deux octets pour identifier le réseau, deux octets pour les hôtes. La plage d'adresses varie de 128.0.x.y à 191.255.x.y.

Classe C : L'adresse IP est formée de trois octets pour identifier le réseau, un octet pour les hôtes. La plage d'adresses varie de 192.0.0.z à 223.255.255.z.

Classe D : classe d'adressage destinée pour faire du *multicast*, ou *multipoint*. Contrairement aux trois premières classes qui sont dédiées à l'unicast ou point à point. La plage d'adresses varie de 224.0.0.0 à 239.255.255.255.

Classe E :Classe d'adressage expérimentale. La plage d'adresses varie de 240.0.0.0 à 247.255.255.255.

Broadcast : Adresse qui désigne toutes les machines d'un réseau, tous les bits de la partie hôte sont à 1.

Multicast : adressage pour s'adresser en une seule fois à un groupe de machines.

Adresse source Adresse IP de l'émetteur, à l'origine de l'information.

Adresse destination : Adresse IP du récepteur de l'information.

ARP Address Resolution Protocol : Protocol permettant d'avoir l'adresse MAC à partir de l'adresse IP.

ICMP Internet Control Message Protocol : Mécanisme permettant de contrôler et de détecter des échecs de transmission au niveau IP.

Port : entier de 16 bits non signé appartement à l'intervalle [1, 65535], utiliser pour identifier un service ou une application.

DNS Domain Name Server : Serveur de noms gérant la correspondance entre les noms et les adresses des machines.

NAT Network Adresse Translater : Protocole permettant de convertir des adresses IP non routables vers des adresses publiques et donc routées sur Internet.