Cursus Master Ingénierie
Informatique
systèmes
et réseaux

Université de Strasbourg

Projet de recherche et de documentation scientifique

Printemps 2022

Sécurité des communications

Quelles sont les menaces visant les communications électroniques
et comment s'en protéger ?

Léon Gall, Florent Hardy
Licence 2 CMI Informatique

Encadré par
Stéphane Cateloin
Directeur du département d'informatique
Université de Strasbourg

Table des matières

1 Introduction

Dans le cadre de notre projet de recherche et de documentation scientifique de deuxième année du Cursus Master Ingénierie (CMI) en informatique, nous avons eu l'occasion d'effectuer des recherches bibliographiques sur la sécurité des communications. En effet, lorsque les protocoles fondamentaux d'internet ont été développés, ils n'incluaient pas de mécanismes de sécurité. Dès lors que l'utilisation d'internet s'est étendue au grand public, il a été nécessaire de mettre en œuvre des protocoles permettant de garantir une sécurité au transport des données.

Cette unité d'enseignement a été pour nous une première introduction à la recherche, ayant pour objectif d'acquérir des compétences telles que de trouver des sources d'information, et reconnaître leur niveau de fiabilité ou bien encore d'extraire et synthétiser les informations utiles d'une ressource.

Nous tenons à remercier Stéphane Cateloin, notre encadrant, qui nous a accompagné et guidé durant toute la durée de nos recherches documentaires.

Ce document est une synthèse des différentes recherches que nous avons menées. Nous avons choisi d'étudier les menaces visant les communications électroniques, ainsi que les mécanismes qui ont été proposés pour s'en protéger. La sécurité des communications étant un thème très vaste, nous avons choisi de ne pas traiter de la sécurité des moyens physiques mis en place pour communiquer, ou de la sécurité du contenu des messages, mais de nous focaliser sur les communications électroniques et leur sécurité en tant que moyen d'envoi et de réception d'informations peu importe leur type et l'infrastructure utilisée.

Pour ceci, dans ce rapport, nous allons commencer par expliquer ce que sont les communications électroniques, en poursuivant par la présentation de la sécurisation de ces communications, suivi d'exemples de mises en œuvres, pour finir avec les différentes perspectives et problématiques de recherche actuelles.

2 Les communications électroniques

Dans ce document, nous entendons par communication électronique l'échange d'informations par le biais d'un réseau de machines et de son infrastructure. Afin d'étudier la sécurité des communications électroniques, il est d'abord nécessaire de comprendre le fonctionnement des communications électroniques.

2.1 Les modèles et les protocoles

Pour les communications réseau, il est nécessaire de créer des conventions sur lesquelles les personnes qui souhaitent communiquer peuvent se reposer. Le code ASCII, par exemple, associe à une suite binaire des valeurs décimales ou les lettres de l'alphabet latin. Ce code est une convention permettant à deux personnes de communiquer en faisant passer leur message par une machine qui ne possède pas le même alphabet. Les modèles et protocoles présentés ci-dessous sont également des conventions, la différence étant que ces derniers permettent des communications à distance et non de coder une donnée en binaire.

2.1.1 Le modèle OSI

Le modèle théorique OSI (*Open Systems Interconnection*) proposé par l'ISO [1] est une hiérarchie en 7 couches permettant de catégoriser les moyens mis en œuvre pour communiquer[1]. Lors d'une émission, le message va subir des traitements qui ont pour but le bon acheminement des informations. Ces traitements sont communément appelés *encapsulations*. Les couches sont :

7 - **Application** : la couche la plus proche de l'humain. Elle sert de point d'accès à un réseau.

6 - **Présentation** : sert à formater l'information reçue afin qu'elle soit lisible par une autre machine.

5 - **Session** : sert à la gestion des connexions. Elle assure l'ouverture et la fermeture de connexions.

4 - **Transport** : est responsable du contrôle de flux ainsi que du contrôle d'erreur. C'est à partir de cette couche qu'une connexion peut être qualifiée de bout-en-bout. L'ensemble des traitements effectués de la couche application à celle-ci seront lues uniquement par le destinataire et jamais par le réseau par lequel passe le message.

3 - **Réseau** : a pour objectif le routage des données notamment en accordant des adresses logiques aux réseaux et aux machines. Lors d'une émission, les messages sont découpés en paquets.

2 - **Liaison** : doit permettre le routage au sein d'un même réseau, notamment en prenant en compte les adresses physiques des machines au sein d'un réseau. Lors d'une émission, les messages sont découpés en trames.

1 - **Physique** : est responsable de l'acheminement des données d'une entité physique à une autre, par exemple via des câbles faisant passer un courant électrique.

Chaque couche utilise les données de la couche directement supérieure et donnent leurs données à la couche directement inférieure.

Ce modèle théorique sert de calque sur lequel peuvent se poser des modèles pratiques. Il n'est pas nécessaire de suivre les indications de ce modèle à la lettre afin de créer un modèle pratique fonctionnel. Il suffit de suivre les principaux aspects qui sont l'encapsulation des données afin qu'elles respectent les conventions de l'émetteur et du récepteur ainsi qu'assurer le déroulement sans erreurs de l'acheminement. Un exemple de modèle pratique fonctionnel ne respectant pas entièrement cette architecture est le modèle de communication TCP/IP.

1. L'ISO (*International Organization for Standardization*) est un organisme de normalisation international.

2.1.2 Le modèle TCP/IP

Le modèle TCP/IP porte le nom de deux protocoles qu'il utilise, le protocole TCP (*Transmission Control Protocol*) et le protocole IP (*Internet Protocol*). TCP est un protocole appartenant à la couche de transport du modèle OSI, IP est un protocole appartenant à la couche réseau du modèle OSI. Les autres couches ne sont pas spécifiées et le modèle TCP/IP ne pose aucune contrainte à part l'utilisation des protocoles TCP et IP[2]. Les couches non spécifiées sont regroupées dans des catégories : *application* pour les couches de haut niveau, *interface réseau* pour les couches bas niveau.

En plus de la fragmentation des données en segments, et de leur remise en ordre lors de la réception, le protocole TCP est chargé d'assurer la fiabilité de la transmission. Par exemple, un message ayant subi une encapsulation afin de correspondre au protocole TCP possédera des sommes de contrôles qui décrivent le message. Si, au moment de la réception, la description du message ne correspond pas au message, cela veut dire qu'il y a eu des pertes ou des modifications au moment de l'acheminement. Dans ce cas TCP n'autorisera pas la transmission du message à la couche application du récepteur.
Le protocole TCP permet une connexion bout en bout, c'est-à-dire qu'il offre aux couches applications de l'émetteur et du récepteur une proximité virtuelle. Du point de vue des applications, les messages ne passent pas au travers d'un réseau. La mise en place d'une connexion TCP se fait par des messages de synchronisation (SYN) et de reconnaissance (ACK), qui assurent que les partis communiquent selon des paramètres partagés[3].

Le protocole IP est un protocole d'adressage et de routage permettant la transmission de messages entre différents réseaux locaux en leur attribuant chacun une adresse unique : une adresse IP. À l'aide de ces adresses IP les routeurs peuvent acheminer, en direction du récepteur, le flux de données. Lors de la réception d'un message l'adresse IP de destination est lue (flèches vertes représentant la lecture des adresses IP cf. Figure 2.1). La lecture est suivie d'une recherche du prochain routeur dans la table de routage, le message est ensuite envoyé vers le routeur associé à l'adresse IP.

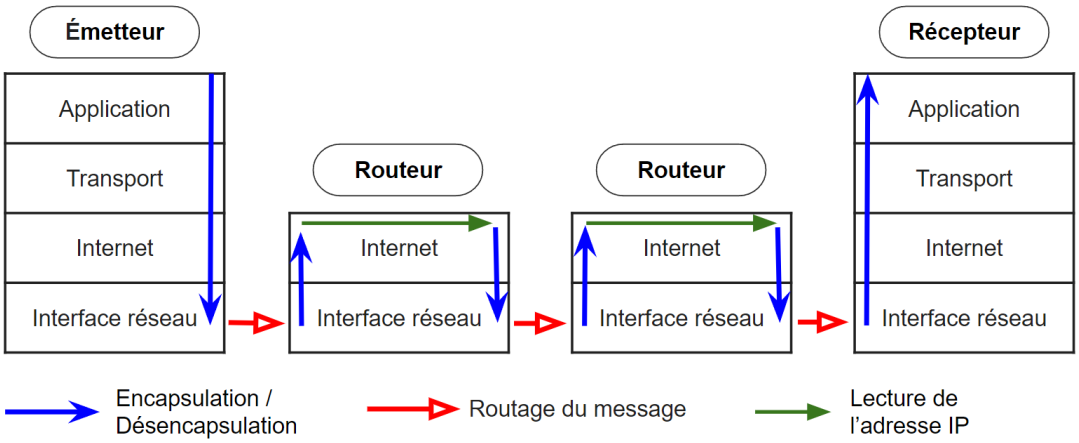

FIGURE 2.1 – Illustration d'un envoi de message dans le cadre du modèle TCP/IP

Dans le modèle TCP/IP, les messages subissent donc des encapsulations (flèches bleues qui passent entre les couches sur la Figure 2.1) afin d'être routés correctement (flèches rouges liant émetteur, routeur et récepteur cf. Figure 2.1), mais également afin d'assurer le contrôle de flux et le contrôle d'erreur. Le protocole TCP permet une communication fiable et le protocole IP permet les communications à distance.

2.2 Les applications

Le modèle TCP/IP ne pose aucune contrainte quant aux protocoles d'applications qui peuvent être utilisés. Deux exemples de protocole de la couche application sont HTTP (*Hypertext Transfer Protocol*) et BGP (*Border Gateway Protocol*). DNS est un service facilitant la navigation web en permettant la

traduction entre un nom de domaine textuel et l'adresse IP associée.

2.2.1 HTTP

HTTP (*HyperText Transfer Protocol*) est un protocole utilisé, entre autres, pour la communication entre client et serveur[4]. Lors d'une navigation web par exemple, un client peut demander le contenu d'une page web à un serveur via des requêtes HTTP.
Une requête HTTP est constituée d'un mot clé définissant l'intention de la requête, de la ressource ciblée, de la version de HTTP utilisée et d'en-têtes.

2.2.2 DNS

Le service DNS (*Domain Name System*) fonctionne, par exemple, avec HTTP. Il est plus pratique de retenir du texte plutôt que des adresses IP et DNS est le service qui permet aux noms de domaines d'exister. Les noms de domaines sont des identifiants sous forme de texte représentant des adresses IP. Les traductions d'une forme vers l'autre se font via des requêtes envoyées aux serveurs DNS. Lors d'une navigation web par exemple, si on souhaite accéder à un nom de domaine, il faut pouvoir traduire le texte en une adresse IP accessible par le protocole IP. Cette traduction est faite en demandant à un serveur DNS s'il détient dans sa table l'association entre le texte, qui est un nom de domaine, et son adresse IP. S'il ne possède pas cette information, le serveur DNS fera une demande aux serveurs DNS situés plus haut dans la hiérarchie. Ces serveurs auront le même comportement jusqu'à trouver le serveur DNS qui détient l'association entre nom de domaine et adresse IP[5].

2.2.3 BGP

BGP (*Border Gateway Protocol*) permet la communication entre systèmes autonomes, ou AS (*Autonomous System*). Un AS est un système cohérent de machines régies par une unique entité. Dans le contexte de BGP, un AS est formé de routeurs qui dictent la direction que doit prendre un message entrant, sortant, ou passant par l'AS. Les informations contenues au sein des tables de routage des routeurs BGP, prennent la forme d'un préfixe d'adresse IP couplé au chemin menant à l'AS que le préfixe représente. Ce chemin est exprimé sous forme de liste d'AS à parcourir afin d'arriver à destination. BGP est donc un protocole de routage à vecteurs de chemin. La table de routage est mise à jour à chaque réception d'un message de changement de routage provenant d'un autre AS[6] (cf. l'AS 3 de la Figure 2.2).

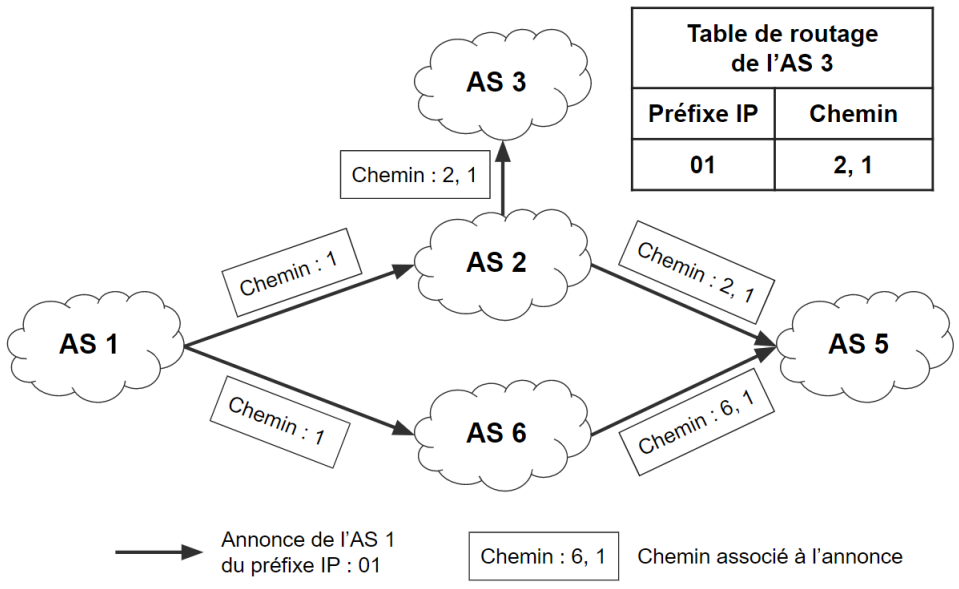

FIGURE 2.2 – Illustration d'une annonce BGP

Lors de l'envoi et propagation d'une annonce BGP, les AS, par lesquels passent l'annonce, mettent à jour le chemin précisé dans l'annonce (cf. Figure 2.2). À la réception d'un message, les AS actualisent leur table de routage (voir la table de routage de l'AS 3 de la Figure 2.2) afin d'inclure l'association entre préfixe et chemin. Si un AS reçoit plusieurs annonces pour un même préfixe, il fera son choix selon des critères politiques, sociaux ou économiques plutôt que de choisir un chemin optimal. Dans le cas de l'AS 5 de la figure, l'AS fera son choix entre le chemin contenant l'AS 6 et l'AS 2 selon ses préférences qui se reflètent dans la configuration des routeurs et des annonces.

Ces protocoles ont été développés à la fin du 20e siècle, une période où les communications électroniques n'étaient pas encore popularisées, où seules les personnes renseignées utilisaient ces protocoles et où les informations communiquées n'étaient pas sensibles. Aujourd'hui, ces moyens de communication sont à la portée de tous et les types de données acheminées ont été multipliés. On peut, de nos jours, effectuer des transactions bancaires à l'aide de ces protocoles, ce qui soulève des problématiques de sécurité quant à l'authentification des parties ainsi qu'à l'intégrité et la confidentialité des transmissions.

2.3 Problématiques de sécurité

L'utilisation croissante des protocoles de communication et le caractère sensible des données acheminées nécessite de s'intéresser aux vulnérabilités, en termes de sécurité, des protocoles utilisés. Pour ce faire, il faut s'interroger sur ce que pourrait faire une entité malveillante, sur la façon dont elle pourrait s'y prendre pour lire ou altérer des informations et sur l'utilisation potentielle de ces informations.

2.3.1 Données lisibles et intelligibles

Les données transmises par la couche application du modèle TCP/IP peuvent contenir des informations sensibles. Lorsque ces informations transitent, l'adresse IP du destinataire est désencapsulée et lue afin de router le message. Rien n'empêche à une machine se faisant passer pour un routeur de lire l'intégralité du message. Ce message étant intelligible, cette lecture porte atteinte à la sensibilité des données. Dans le cas d'une transaction bancaire, la lecture permettrait le vol des données bancaires.
Les outils permettant la lecture du trafic s'appellent les *sniffers*, ou *renifleurs* en français. Il n'existe que très peu de moyens d'empêcher les lectures passives[7], qui consistent à lire, sans modifier, le trafic, à cause de leur discrétion.

Ce problème est omniprésent dans le protocole HTTP qui n'intègre aucun moyen d'assurer l'inintelligibilité des messages hormis ceux liés à l'authentification mais même ceux-ci possèdent des faiblesses, notamment l'attaque de l'Homme du milieu.

2.3.2 Usurpation d'identité

Une problématique de l'authentification apparaît lors de la première communication entre l'émetteur A et le récepteur B, qui sert à authentifier les parties. Supposons que A et B possèdent un moyen sûr de communiquer après s'être échangés leurs certificats (le principe derrière les algorithmes de chiffrement). Les certificats sont des messages particuliers qui permettent l'échange de clés. L'usage de ces clés est détaillé dans la partie 3.1 : confidentialité de ce document. Le problème apparaît si une entité C se trouve entre A et B. Au moment de l'échange des certificats de A et B, C peut conserver les certificats de A et de B et envoyer son propre certificat en se faisant passer pour A aux yeux de B et en se faisant passer pour B aux yeux de A. Dans cette situation, A et B pensent communiquer entre eux de manière sécurisée et pourtant C est non seulement capable de lire les données qui transitent mais peut également forger des messages en se faisant passer pour A ou B. C'est l'attaque de l'Homme du milieu. Dans cet exemple, l'attaque exploite la faiblesse suivante : afin de créer un canal de communication sécurisé, il faut un canal de communication sécurisé, mais ce type d'intrusion apparaît à chaque fois qu'une entité se trouve entre un émetteur et un récepteur, en étant capable de lire, modifier ou forger des données.

Dans le protocole HTTP, il existe deux méthodes d'authentification nommées Digest et Basic[8] qui fonctionnent via un défi et une réponse. Le mode de fonctionnement de ces défis est spécifié dans la partie 4.1 de ce document sur les exemples de mise en œuvre de la sécurité. Le serveur envoie un défi

au client et si ce dernier est capable de répondre correctement, alors l'authentification est effectuée. Digest rend inintelligible le message du client au serveur, Basic non. Si une authentification Digest, supposée sécurisée, se fait au sein d'un système qui subit une attaque de l'Homme du milieu, l'intrus peut proposer un défi Basic au client, en sachant qu'il pourra récupérer les informations du client et contourner la sécurité de l'authentification Digest (cf. Figure 2.3).

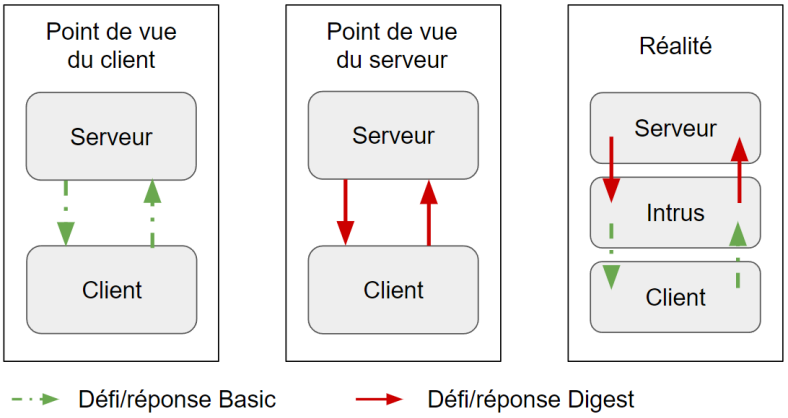

FIGURE 2.3 – Illustration d'une attaque de l'Homme du milieu lors d'une authentification HTTP Digest

Dans le cas de DNS, qui ne possède pas de méthode d'authentification sécurisée, n'importe quel serveur DNS peut prétendre posséder l'information recherchée par le client et renvoyer l'information qu'il souhaite. L'empoisonnement DNS ou *DNS Poisoning* consiste à introduire dans la mémoire cache d'un serveur DNS des informations erronées. Pour ce faire, une requête DNS est envoyée à un serveur DNS. Suite à quoi, le serveur attend une réponse des différents serveurs DNS auxquels il a transmis la requête. À partir de là, l'entité malveillante envoie des requêtes réponses similaires à ce qu'attend le serveur DNS cible de l'attaque. Si l'entité malveillante envoie une requête avec un identifiant de transaction correct, la requête est acceptée et le serveur DNS cible garde en mémoire cache l'association entre nom de domaine et nouvelle adresse IP. Les requêtes DNS pour le nom de domaine cible de l'attaque, si maniées par le serveur DNS empoisonné, renverront la nouvelle adresse IP choisie par l'entité malveillante.

Dans le protocole BGP, il n'existe pas d'authentification intrinsèque[5][9], n'importe quel AS peut envoyer n'importe quel type de message. Un AS peut prétendre en être un autre ou prétendre qu'un préfixe IP d'un autre AS lui appartient. Ces attaques sont communément appelées *hijack* provenant de la traduction anglaise de *détournement*, ou *blackholing* qui est la traduction de *création de trou noir*. Dans la première appellation, le détournement des données est mis en avant, elle se base donc sur le manque de confidentialité, l'AS malveillant a accès aux données de l'AS qu'il usurpe. La seconde fait référence au fait qu'un AS qui en usurpe un autre peut choisir de ne pas servir de passerelle, de ne pas faire transiter les données. Les données qui arrivent dans l'AS malveillant y restent donc et l'ensemble des AS dont le transit dépend de l'AS usurpé ne recevront jamais les données qui leur sont dédiées.

Les protocoles et services cités plus haut possèdent des vulnérabilités en termes de lecture des données et en termes d'authentification des utilisateurs. Ces vulnérabilités ont déjà été exploitées comme par exemple le vol des préfixes par hijack BGP du domaine : panix.com, empêchant l'accès au site pendant plusieurs heures[9] ; ou encore les attaques de l'Homme du milieu rendues possibles sur les services proposés par Google des suites d'une attaque sur l'organisme DigiNotar[10][11]. Avec ces attaques, la personne responsable a récupéré plusieurs identifiants et mots de passe d'utilisateurs des services de Google.
Il est donc nécessaire de modifier ces protocoles ou d'y ajouter des niveaux de sécurité si l'on souhaite les utiliser pour véhiculer des données sensibles.

3 Sécurité des communications

Dès l'Antiquité, la sécurité des communications fût un enjeu majeur. Par exemple, lorsque des dirigeants voulaient s'échanger des informations secrètes, en laissant leurs subordonnés transmettre les messages, ils ne pouvaient pas se permettre que le message soit compris ou modifié par des tiers, ou encore qu'une autre personne mal intentionnée envoie un message se faisant passer pour un dirigeant. La recherche de sécurité dans les échanges est née, et s'est naturellement divisée en trois parties : la confidentialité, l'intégrité et l'authentification.

3.1 Confidentialité

La confidentialité consiste à rendre inintelligibles des données avant de les transmettre, afin qu'une personne non autorisée ayant accès à ces données ne puisse les comprendre. Puisque ces données doivent être comprises par le destinataire du message, le chiffrement doit se faire suivant une fonction inversible, permettant un déchiffrement des données.

Une des premières fonctions de chiffrage fût le chiffrement de César, historiquement utilisé pendant les campagnes de guerres romaines, pour la transmission des stratégies de façon confidentielle. Le fonctionnement est assez simple : chaque lettre est substituée par une autre, qui est la n-ième lettre suivante dans l'alphabet. Par exemple, avec $n = 3$, A sera remplacée par D, B par E, etc. On appelle ce type de chiffrement un chiffrement par substitution.

Avec les techniques de cryptanalyse [1] actuelles, cette technique de chiffrement est vulnérable : si l'on connaît un mot, il est possible de déchiffrer le message en entier. Pour contrer les capacités de cryptanalyse actuelles, les algorithmes de cryptographie doivent respecter certaines contraintes, telles que le fait que deux données proches, comme par exemple deux mots différents de seulement une lettre, doivent fournir un cryptogramme différent, non proche. Aussi, l'attaque par force brute, i.e. testant toutes les possibilités doit être théoriquement impossible, en nécessitant un temps de calcul extrêmement long. Le chiffrement des données est aujourd'hui réalisé suivant deux méthodes : le chiffrement symétrique et le chiffrement asymétrique, respectant ces conditions.

Le chiffrement symétrique[12] consiste à chiffrer et déchiffrer un message, suivant un algorithme (tel qu'AES par exemple), avec la même clé secrète, qui est donc commune à l'émetteur et au destinataire. Le chiffrement de César est symétrique. En effet, au chiffrage et déchiffrage, le même décalage dans l'alphabet doit être effectué. Si nous reprenons notre exemple précédent, $n = 3$ est la clé secrète de ce système. Le principal avantage de la cryptographie symétrique est son chiffrement assez rapide. En effet, elle utilise des opérations telles que des XOR ou des décalages de bits, entre le message et la clé secrète. De plus, elle ne nécessite qu'une clé secrète de taille relativement réduite (proche de 128 bits) par rapport aux clés utilisées dans la cryptographie asymétrique. Afin de permettre les opérations, le message doit être de même taille que la clé secrète. Il est par conséquent soit complété par un padding (octets tous égaux permettant d'atteindre la bonne longueur), soit divisé en différents blocs, suivant si il est trop court ou trop long. Mais cette cryptographie présente également des défauts : le principal est qu'il faut que les deux correspondants se mettent d'accord sur leur clé secrète en amont, de façon sécurisée. En outre, le nombre de clés secrètes devient très grand quand un grand nombre de personnes communiquent entre-elles : pour que n personnes communiquent deux-à-deux, il faut $n(n+1)/2$ clés différentes.

Le chiffrement asymétrique[12], quant à lui, utilise deux clés reliées mathématiquement. Une clé secrète,

1. La cryptanalyse est l'ensemble des moyens permettant le décryptage d'une donnée, à partir de l'analyse de la donnée chiffrée.[7]

appelée clé privée, et une clé publique, non secrète. La clé publique est obtenue à partir de la clé privée, mais il serait extrêmement difficile d'obtenir la clé privée à partir de la clé publique. Ainsi, une personne possédant la clé publique ne pourrait trouver la clé privée. Ceci assure la sécurité de ce système cryptographique.
Chaque correspondant possède un couple (clé publique, clé privée), et peut avoir accès aux clés publiques des autres correspondants. Si une personne A veut envoyer un message à une autre personne B, elle va chiffrer son message avec la clé publique de B, suivant un algorithme (comme RSA par exemple) puis va envoyer le message chiffré à B qui pourra de cette façon déchiffrer le message à l'aide de sa clé privée. Ce type de cryptographie résout le problème du nombre de clés en cryptographie symétrique, ou de l'échange des clés, mais possède un principal inconvénient : le temps de calcul est bien plus long que pour un système symétrique. En effet, elle utilise des exponentiations, longues à calculer, suivant la taille de la clé. Or, la taille de la clé publique doit être très grande (proche de 1024 bits), car elle est par définition le produit de deux nombres premiers de grande taille, pour garantir une sécurité aux attaques par force brute.

Pour tirer profit de la rapidité de la cryptographie symétrique tout en conservant les avantages du chiffrement asymétrique, un compromis a été trouvé : la clé de session. Une clé symétrique est envoyée (ou calculée grâce à un secret partagé Diffie-Hellmann[13] par exemple), via un système asymétrique. Une fois la clé réceptionnée, les deux personnes peuvent communiquer de façon confidentielle, en utilisant la cryptographie symétrique. C'est notamment ce qu'utilise le protocole TLS pour chiffrer les communications.

3.2 Intégrité

L'intégrité consiste à s'assurer que les données ne soient pas modifiées par un tiers durant le transfert.

Un moyen de s'assurer de l'intégrité d'une communication est de joindre au message transmis un hachage de celui-ci. Un hachage est le résultat d'une fonction qui, à partir d'opérations telles que des ou exclusifs (XOR), de décalages de bits, etc. appliqués sur un message, retourne un autre message. Cette fonction a été conçue pour être à sens unique : à partir d'un hachage, il ne sera pas possible de retrouver le message initial. De plus, puisque généralement, la fonction prend en entrée un message de longueur plus grande que la sortie, des collisions (des valeurs différentes en entrée produisant la même sortie) pourront forcément exister. Plus la probabilité de collision est faible, meilleure est la fonction de hachage. SHA-512 est une fonction de hachage utilisée actuellement. Pour toute entrée, elle donne une valeur codée sur 512 bits. La longueur du hachage est importante afin de limiter les collisions, mais surtout d'augmenter le temps nécessaire à obtenir le résultat par force brute. En passant de 256 bits à 512 bits, la taille est multipliée par deux mais le nombre de combinaisons possibles est quant à elle multipliée par 2^{256}.

Un autre moyen assurant l'intégrité est le code d'authentification du message (MAC). Son principe est similaire aux fonctions de hachage, mais à la place de hacher simplement le message (auquel cas deux messages identiques de deux personnes différentes auraient des hachages identiques), elle va prendre en compte une clé secrète qui doit être connue des deux parties. Ceci permet en même temps l'authentification des parties.

Pour vérifier l'intégrité, imaginons qu'une personne A veuille envoyer un message à une personne B. Elle va par conséquent envoyer des données contenant le message ainsi que le hachage de ce message. Ainsi, lorsque B va recevoir le message, il va ré-appliquer la même fonction de hachage puis comparer les deux hachages. S'ils sont identiques, alors le message est intègre. Cependant, si un attaquant modifiait le hachage en même temps que le message, le destinataire pourrait penser à tort que le message est intègre. Pour éviter ceci, il est nécessaire d'authentifier l'émetteur du message.

3.3 Authentification

L'authentification consiste à s'assurer que la personne avec laquelle nous communiquons est bien celle qu'elle prétend être.

Pour ceci, il faut apposer au message ce que l'on nomme une signature. Celle-ci utilise la cryptographie asymétrique. Supposons qu'une personne A souhaite envoyer un message à une personne B. Pour authentifier son message, A va chiffrer son message avec sa clé privée, de cette façon, B pourra s'assurer que c'est effectivement A qui lui a envoyé le message (puisque seul A possède sa clé privée), en déchiffrant le message avec la clé publique de A. Ceci forme la signature du message, de la même façon qu'une signature sur papier.
Malheureusement, cette méthode n'est pas suffisante. En effet, elle ne résiste pas à l'attaque de l'homme du milieu. Si une personne C se place entre A et B, et qu'elle indique à A que la clé publique de B est la sienne, et indique à B que sa clé publique est celle de A, lorsque les messages seront chiffrés, et/ou authentifiés, ils passeront par C qui pourra les déchiffrer, les lire, les modifier puis les renvoyer sans même que A et B ne s'en rendent compte. Il y a donc une problématique concernant la gestion des clés publiques.

Pour éviter les désagréments de ce type, des autorités de certifications ont vu le jour, certifiant le lien entre l'identité d'une personne et sa clé publique. Ces tiers de confiances distribuent des certificats composés d'au moins une clé publique, d'informations d'identifications, telles que le nom, la localisation ou encore l'adresse électronique, ainsi que d'une signature construite à partir de la clé privée de l'autorité. Il existe plusieurs tiers de confiance : pour le protocole TLS, ce sont des entreprises privées qui en ont la charge, tandis que pour BGP, c'est l'IANA (*Internet Assigned Numbers Authority*) qui distribue les certificats. Pour obtenir un certificat, il faut donc faire une demande à une autorité de certification, en donnant des informations personnelles, une clé publique, et en payant le montant demandé. Ce tiers de confiance vérifiera les informations puis accordera un certificat signé avec sa clé privée. Lors des communications authentifiées, le destinataire vérifiera qu'il fait confiance au tiers ayant signé le message. Généralement, plus une autorité est ancienne, et utilisée, plus elle est de confiance.

Les données transmises contiendront par conséquent le message signé, ainsi que le certificat délivré par une autorité de certification. Le fait qu'un message soit authentifié implique également une non-répudiation des données. En effet, une personne ne pourra nier être la source de transmission de données, si elles ont été signées et certifiées.

4 Exemples de mises en œuvre de la sécurité des communications

Afin de mettre en œuvre la sécurité des communications, il faut combiner les différents moyens donnant la confidentialité, l'intégrité et l'authentification.

4.1 HTTP

Le protocole applicatif HTTP (*HyperText Transfer Protocol*) possède de nombreuses failles de sécurité. En effet, les données n'étant pas chiffrées, les communications sur le réseau sont lisibles, non confidentielles. Cela peut être problématique si quelqu'un écoute les communications lors d'une tentative de connexion à un service : la personne verrait le nom d'utilisateur ainsi que le mot de passe en clair. Elle pourrait donc s'en resservir pour se connecter au service de façon illégitime.

Un autre problème d'HTTP est l'authentification. Deux méthodes sont proposées : l'authentification Basic, et Digest[8]. L'authentification Basic transmet les informations de l'utilisateur en base 64 (formatées de la façon suivante : *Pseudo:MotDePasse*). Ces données ne sont pas chiffrées et il est aisé de revenir à une base 2 depuis la base 64. Cette méthode n'est donc absolument pas sécurisée.
L'authentification Digest est déjà plus sécurisée. Elle repose sur des défis : lors d'une demande d'accès à une ressource protégée, le serveur renvoie au client un nombre aléatoire, appelé nonce. Pour résoudre le défi, le serveur doit renvoyer un hachage des informations suivantes : identifiant, mot de passe, nonce, méthode HTTP et URI [1]. De cette façon, la confidentialité des données est assurée, ainsi que le blocage du rejeu de la requête, la valeur du nonce étant unique. Cependant, le Digest n'offre pas de confidentialité au-delà de la protection du mot de passe de l'utilisateur, le reste des informations circulant de façon lisible.

Pour sécuriser ce protocole applicatif, un nouveau protocole sécurisé a vu le jour : HTTPS. Il se base sur HTTP et TLS, pour permettre des communications confidentielles, intègres et authentifiées.

4.2 TLS

TLS (*Transport Layer Security*) est un protocole situé entre la couche application et la couche transport du modèle TCP/IP. Il permet l'authentification d'un serveur, ainsi que la confidentialité et l'intégrité des données. Il est le successeur du protocole SSL (*Secure Sockets Layer*), qui est déprécié suite à la découverte de failles dans ce protocole.

TLS est mis en place avec une poignée de main avant l'échange de données, se déroulant en trois étapes principales décrites en détail dans la RFC 8446[14].

Premièrement, le client envoie un message *ClientHello* au serveur. Il est composé, entre autres, de sa version la plus récente du protocole TLS, d'un nonce aléatoire, et d'une liste d'algorithmes de cryptographie (ciphers) supportés par le client, classés par ordre décroissant de préférence. Un cipher est de la forme : TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 [2]. En particulier, ECDHE représente l'algorithme utilisé pour l'échange des clés (ici Diffie-Hellmann), RSA représente l'algorithme utilisé pour

1. HTTP propose des méthodes, qui consistent à décider de la façon d'envoyer les données. Par exemple, GET transmettra les données via l'URI. L'URI (de l'anglais *Uniform Resource Identifier*) permet l'identification de la ressource. Le plus souvent, il s'agit d'une URL.
2. Exemple issu de Wikipédia.[15]

l'authentification, AES celui pour le chiffrement des données avec la clé de session, et SHA256 la fonction de hachage utilisée pour signer les différents messages.

Le serveur répond avec le message *ServerHello*. Celui-ci contient le nonce donné par le client, sa version la plus récente de TLS, ainsi que le premier cipher du client qu'il possède dans sa propre liste de ciphers. En outre, il transmet également son certificat délivré par un tiers de confiance, permettant d'authentifier le serveur. Suivant l'algorithme d'échange de clés choisies, il joint à cela le nonce chiffré avec sa clé privée, afin de permettre le calcul du secret partagé suivant l'échange de clé Diffie-Hellmann[13], ou encore sa clé publique.

Ensuite, suivant l'algorithme d'échange de clés, le client répond avec le nonce chiffré avec sa clé privée, afin de permettre le calcul du secret partagé au serveur, ou alors il répond avec la clé de session choisie, chiffrée avec la clé publique du serveur. Il peut également ajouter son certificat délivré par un tiers de confiance, si demandé par le serveur. Il sera principalement demandé lors d'échanges entre deux serveurs.

Après avoir annoncé la fin du serrage de mains, les messages sont envoyés de façon chiffrée suivant l'algorithme, et la clé de session définie durant l'accord. De plus, l'intégrité de chaque message est vérifiée grâce au hachage des données, à l'aide d'une fonction également définie durant l'accord.

4.3 DNS

DNS (*Domain Name System*), comme précisé dans la partie 2.3.2, est le système permettant la traduction d'une adresse textuelle, un nom de domaine, en une adresse IP. Ce système ne possède pas de méthode d'authentification des serveurs DNS. Cela veut dire qu'un serveur DNS peut répondre à n'importe quelle requête reçue. Un serveur DNS malveillant peut donc renvoyer des informations délibérément fausses et rediriger ses victimes vers des pages malveillantes ou empêcher le trafic vers le nom de domaine subtilisé. C'est le principe derrière ce qui est communément appelé *DNS Poisoning* ou *empoisonnement DNS*.

La solution à cette vulnérabilité est l'introduction de DNSsec qui est un service ayant les mêmes devoirs que DNS, mais qui offre, entre autres, une méthode d'authentification. Cette authentification se fait par des signatures créées à l'aide d'algorithmes de chiffrement asymétriques et où l'autorité de confiance est l'IETF (*Internet Engineering Task Force*). L'IETF assure que le serveur DNS racine, supérieur hiérarchique de tous les autres serveurs DNS, soit authentifié et possède donc un couple de clés, une privée et une publique, qui permettent la signature de certificats. Ce serveur DNS racine offre aux serveurs DNS inférieurs hiérarchiquement un certificat de confiance qui inclut un couple de clés. Ces serveurs, ayant obtenu la confiance du serveur DNS racine, peuvent à leur tour distribuer des clés aux serveurs auxquels ils font confiance. Cette distribution de clés permet la création d'une chaîne de confiance qu'il suffit de remonter afin de s'assurer de l'identité d'un serveur DNS.

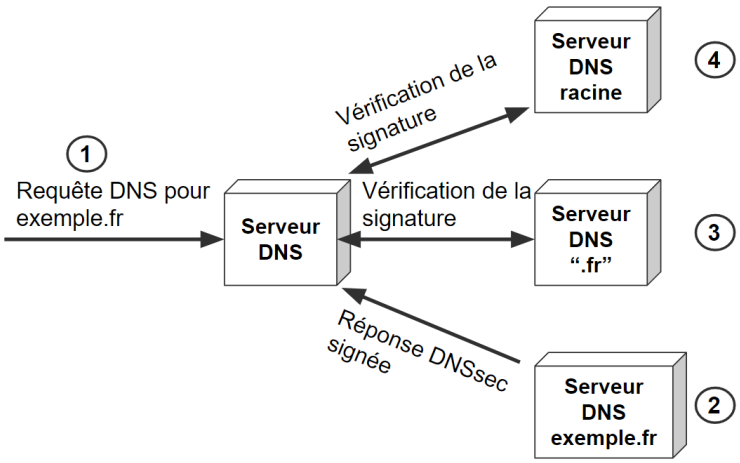

FIGURE 4.1 – Illustration d'authentification DNSsec

Si l'origine d'une réponse DNSsec ayant pour nom de domaine *exemple.fr* est incertaine (cf. Figure 4.1), il suffit de récupérer la signature qui a accompagné le message réponse et de faire une requête au serveur DNS responsable des noms de domaines en *.fr* . Ce dernier nous assurera que le serveur DNS qui a répondu à la requête fait partie de ceux à qui il a distribué un couple de clés, et donc un serveur DNS auquel il fait confiance. Si l'on ne fait pas confiance à ce serveur DNS responsable des noms de domaines en : *.fr*, il est possible de remonter jusqu'au serveur DNS racine, en vérifiant à chaque fois que la signature ait été créée à l'aide d'une clé distribuée par le supérieur hiérarchique en termes de serveurs DNS. Le serveur DNS racine est supposé fiable grâce à l'intervention de l'IETF.

4.4 BGP

Le protocole BGP (*Border Gateway Protocol*) est, comme précisé dans la partie 2.3.2, vulnérable à des attaques telles que le *hijacking* et le *blackholing* qui sont rendues possibles dû au manque d'authentification dans les messages transmis. Un AS (*Autonomous System*) peut annoncer n'importe quel préfixe IP comme le sien et faire en sorte que le trafic en direction de ce préfixe IP aille dans sa direction. Une motivation à faire ceci, peut être que le passage par différents AS est monnayé, suivant des accords politiques. Faire passer du trafic supplémentaire par son AS peut donc être rentable.

Pour résoudre ce problème, l'IANA (*Internet Assigned Numbers Authority*) peut, au moment de l'attribution des préfixes IP, également procurer une ROA (*Route Origin Authorization*) signée avec un algorithme de chiffrement asymétrique, qui dicte quels AS ont le droit d'annoncer un préfixe IP[16][17] (voir l'exemple sur la Figure 4.2). L'IANA est l'autorité responsable de la distribution des préfixes d'adresses IP aux RIR (*Regional Internet Registry*, comme la *RIPE NCC* qui est l'autorité de distribution d'adresses IP en Europe). Ces RIR peuvent à leur tour distribuer à des fournisseurs d'accès internet, ou LIR (*Local Internet Registry*), des plages d'adresses IP et créer leur version d'une ROA. Lors d'une annonce d'un préfixe, il suffit donc de vérifier si l'AS qui a fait l'annonce est autorisé à faire cette annonce, c'est à dire que dans la ROA se trouvent le préfixe IP et l'AS (cf. Figure 4.2).

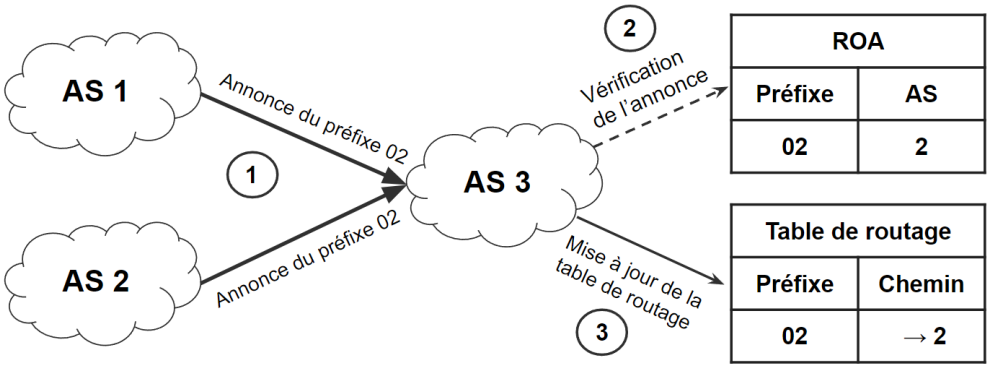

FIGURE 4.2 – Illustration de l'usage d'une ROA dans le cadre de BGP

L'annonce de l'AS 1 de la Figure 4.2 ne correspond pas à ce qui est présent sur la ROA, l'annonce est donc rejetée par l'AS 3 au moment de la vérification. Concernant l'annonce de l'AS 2, elle est vérifiée par l'AS 3 qui met à jour sa table de routage en conséquence.

4.5 VPN

Un VPN (*Virtual Private Network*) est un réseau virtuel privé[7][18]. Il a été conçu afin de permettre aux employés travaillant à distance de se connecter au réseau interne de l'entreprise, pour qu'ils aient accès aux ressources privées. Il permet également de protéger ses données personnelles, telles que sa localisation par exemple.

Il fonctionne de la façon suivante : un client souhaite faire une requête sur internet en passant par un

VPN. Il va donc tout d'abord se connecter au serveur VPN, pour mettre en place une connexion chiffrée entre eux, que l'on peut voir comme un tunnel. Ensuite, il effectue sa requête, passant par le tunnel chiffré, et le serveur VPN se charge d'envoyer la requête sur internet si besoin. Par exemple, dans le cas d'un travailleur distant, il peut se connecter au serveur VPN étant dans le réseau interne de l'entreprise, puis effectuer des requêtes comme s'il était dans le réseau interne de l'entreprise lui-même, puisque c'est le VPN qui envoie les requêtes (cf. Figure 4.3).

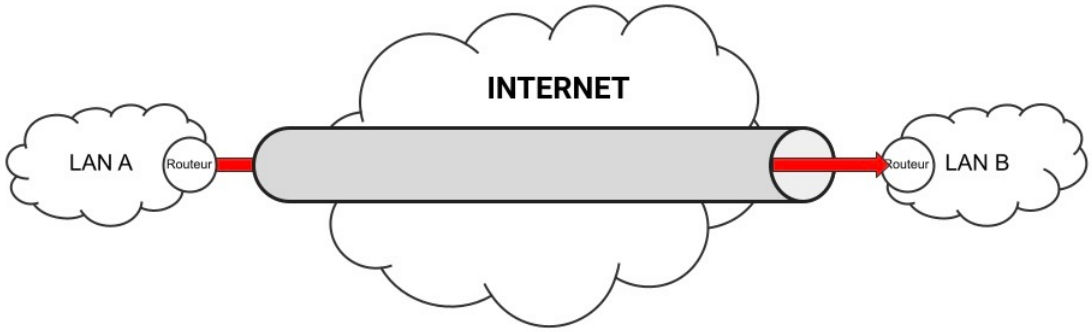

FIGURE 4.3 – Illustration d'un tunnel VPN

Pour un VPN IPsec, afin de créer le tunnel chiffré, des clés sont échangées à la suite d'une poignée de main similaire à TLS. Ensuite, les paquets IP ayant pour destination la ressource demandée sont chiffrés avec le protocole IPsec[19], avant d'être encapsulés dans un nouveau paquet IP, ayant pour destination le serveur VPN. Une fois arrivé au serveur VPN, lorsque le paquet IP est désencapsulé, la donnée est déchiffrée donnant un nouveau paquet IP qui peut être à présent ré-encapsulée puis routé. Il existe également d'autres protocoles de VPN, tel que L2TP[20] qui est un protocole routé (de couche liaison du modèle OSI), utilisant IPsec pour le chiffrement des données, ou bien encore openVPN[21] utilisant TLS.
Par conséquent, un VPN assure la confidentialité, l'intégrité et l'authentification avec des protocoles tels qu'IPsec.

4.6 ARP

Lors d'une première communication à un réseau, l'émetteur ne peut immédiatement connaître l'adresse physique de la machine avec laquelle il souhaite communiquer. ARP (*Address Resolution Protocol*) est le protocole, spécifique à IPv4, qui sert à faire une demande au sein du réseau local pour savoir à quelle machine appartient une certaine adresse IP, afin de récupérer son adresse MAC (*Media Access Control*) qui est l'adresse physique spécifique à une machine. Le problème étant que n'importe quelle machine peut répondre à la requête ARP et prétendre être la machine destinataire.
Il se trouve que dans un réseau qui utilise et nécessite ARP, il n'est pas possible de rendre le protocole sécurisé. Il faut soit réserver les ports du switch, servant au routage local, pour chaque machine auquel cas le switch peut vérifier que l'adresse MAC de la machine corresponde au numéro de port, soit faire la transition vers IPv6. IPv6 n'utilise pas ARP mais ND (*Neighbor Discovery*) et même SEND (*Secure ND*) qui permet la résolution d'adresses physiques avec confidentialité et authentification des parties.

5 Perspectives et problématiques de recherche

Actuellement, il y a diverses perspectives de recherche. La première est la sécurité menacée par l'informatique quantique. En effet, les algorithmes de cryptographie utilisent la difficulté à factoriser un nombre en produit de nombres premiers. En utilisant des nombres très grands, il est alors presque impossible de réussir à casser les systèmes cryptographiques actuels, en un temps raisonnable. L'informatique quantique viendrait bouleverser ceci, en utilisant des principes tels que la superposition quantique [1], qui permet de tester un grand nombre de combinaisons en une seule opération. L'algorithme de Shor[22] est un algorithme permettant de factoriser un nombre en produit de nombres premiers en un temps polynomial à l'aide d'un ordinateur quantique. De nombreuses fonctions de cryptographie seraient par conséquent vulnérables, notamment RSA. La recherche de nouveaux algorithmes résistants aux informatiques quantique est par conséquent primordiale. Une idée est de tirer profit de l'informatique quantique et d'utiliser des algorithmes quantiques pour sécuriser les communications.

Une autre problématique de recherche est la démocratisation des moyens de sécurisation. Comment peut-on rendre des protocoles ou des services plus populaires et surtout plus accessibles ? Une couche de sécurité n'est efficace que si elle est utilisée. Deux exemples sont la transition vers IPv6 [2], qui empêche toute attaque sur la résolution d'adresses physiques à l'aide de *SEND*, et le déploiement des moyens d'authentification pour les AS qui utilisent BGP. En 2019, seuls 16% des préfixes IP avaient été distribués avec un certificat qui permettrait aux AS de s'identifier[17] et pourtant l'absence d'identification a déjà créé des problèmes tels que le vol des préfixes du domaine : panix.com, ce qui a empêché l'accès au site pendant plusieurs heures[9]. Si le système d'authentification BGP avait été totalement déployé, l'incident n'aurait jamais eu lieu, il est donc légitime de se demander comment déployer un protocole sécuritaire.

Dernièrement, nous avons étudié la nécessité d'un tiers de confiance lors de la vérification de l'association entre une entité et un couple de clés asymétriques. Comme leur nom l'indique, la sécurité créée par ces tiers de confiance dépend de la confiance qui leur est accordée. Cette confiance est centralisée sur ces entités, ce qui permet des abus. En 2015, l'autorité de confiance Symantec a créé des certificats au nom de Google, sans que Google ne les ait demandés[24]. Symantec, et les autorités de certification de manière générale, peuvent donc créer des certificats et permettre à quiconque de se faire passer pour Google ou tout autre organisme. Il suffit qu'une personne malintentionnée s'empare de la faculté de créer et distribuer des certificats pour rendre inefficaces les authentifications basées sur ces certificats. En 2011 l'autorité de certification DigiNotar subit une intrusion et l'intrus crée des certificats de confiance qu'il utilise pour se faire passer pour Google[10][11]. Une solution à la centralisation de la confiance serait de la décentraliser, via la blockchain par exemple[7][24]. La blockhain est une technologie de stockage d'informations décentralisée, sur laquelle pourraient être stockées les clés publiques d'organismes sans devoir passer par un tiers de confiance centralisé. À la réception d'un message signé avec une clé privée, il suffirait de récupérer la clé publique de l'organisme sur la blockchain et de vérifier la signature.

1. La superposition quantique est un concept de l'informatique quantique dans lequel un qubit (bit quantique) peut avoir pour valeur 0, 1 ou les deux en même temps.
2. Moins de 40% des utilisateurs de Google utilisent une adresse IPv6[23].

6 Conclusion

Pour conclure, afin de répondre à des problématiques de sécurité, la sécurité des communications a dû mettre en œuvre des mécanismes assurant la confidentialité, l'intégrité et l'authentification. Ainsi, il a fallu sécuriser des protocoles tels que HTTP, devenant HTTPS[25] (HTTP sécurisé par le protocole TLS) pour des communications entre clients et serveurs, ou encore le DNS en utilisant DNSsec.

Bien que ces protocoles soient aujourd'hui sécurisés, et permettent des communications sûres, de nouvelles failles voient le jour régulièrement, ce qui entraîne des mises à jours et améliorations dans les protocoles, comme avec SSL[26] (*Secure Sockets Layer*) devenant TLS suite à diverses failles découvertes au sein de ce protocole. Malgré tout, en informatique, l'humain reste la faille principale, commettant des erreurs, ou faisant confiance à des personnes malveillantes. Ceci implique que des attaques pourront toujours avoir lieu, malgré toutes les avancées en terme de sécurité.
Nous pouvons à présent nous demander comment évoluera la sécurité des communications lorsque des ordinateurs quantiques verront le jour, étant capables d'outrepasser la cryptographie actuelle...

À travers cette unité d'enseignement, nous avons pu découvrir la recherche documentaire dans le domaine de la sécurité des communications informatiques. Ce fût un travail intéressant qui nous a enseigné diverses compétences telles que la méthodologie pour effectuer des recherches pointues sur un thème scientifique précis, en utilisant des sources telles que des sites internet de documentation, ou à travers des livres sur les réseaux informatiques et leur sécurité. Ces compétences nous seront utiles pour la suite de nos études, et pour nos futurs métiers respectifs.

Afin de construire notre bibliographie, à chaque fois que nous consultions une ressource intéressante, nous notions dans un document au format *bibtex* les informations permettant de citer la ressource, puis nous résumions le document dans notre fichier collaboratif de synthèse. En outre, pour s'assurer que nos ressources étaient fiables, nous avons à chaque fois essayé de croiser les différentes informations avec d'autres sources, et nous avons cherché qui était auteur des informations ainsi que sa légitimité à les donner. Ainsi, nous avons pu séparer les ressources fiables des non-fiables.
Pour garder une vision d'ensemble sur le sujet, même lorsque nous approfondissions une thématique particulière, nous avons réalisé une carte mentale, que nous avons alimenté au fur et à mesure de nos recherches. Cette vision globale nous a permis d'être sûrs de bien couvrir l'intégralité du sujet, en ne laissant rien de côté.

En outre, les compétences que nous avons acquises sur les réseaux informatiques et leur sécurisation, nous donneront de l'avance pour certaines matières de notre cursus à venir.

Bibliographie

[1] Oracle, "Couches de protocoles et modèle OSI." https://docs.oracle.com/cd/E19957-01/820-2982/ipov-7/index.html. Consulté le 23-02-2022.

[2] IBM, "TCP/IP protocol." https://www.ibm.com/docs/en/aix/7.1?topic=protocol-tcpip-protocols. Consulté le 09-02-2022.

[3] Oracle, "Modèle d'architecture de protocoles TCP/IP." https://docs.oracle.com/cd/E19957-01/820-2982/ipov-10/index.html. Consulté le 22-02-2022.

[4] Mozilla, "Un aperçu de HTTP." https://developer.mozilla.org/fr/docs/Web/HTTP/Overview. Consulté le 23-02-2022.

[5] G. Pujolle, *Les réseaux, L'ère des réseaux cloud et de la 5G*. Eyrolles, 2018.

[6] Cloudflare, "What is BGP." https://www.cloudflare.com/learning/security/glossary/what-is-bgp/. Consulté le 09-03-2022.

[7] G. Solange, *Cybersécurité : Sécurité informatique et réseaux*. DUNOD, 5. ed., 2016.

[8] P. J. Franks, P. Hallam-Baker, L. C. Stewart, J. L. Hostetler, S. Lawrence, P. J. Leach, and A. Luotonen, "HTTP Authentication : Basic and Digest Access Authentication." RFC 2617, June 1999.

[9] G. A. Bahaa Al-Musawi, Philip Branch, "BGP Anomaly Detection Techniques : A Survey." https://cs.gmu.edu/~eoster/2019-795/2019-795-papers/BGP%20Anomaly%20Detection%20Techniques%20Survey.pdf, 2017.

[10] Veracode, "Man in the Middle (MITM) Attack." https://www.veracode.com/security/man-middle-attack. Consulté le 23-04-2022.

[11] Wikipedia, "DigiNotar." https://fr.wikipedia.org/wiki/DigiNotar. Consulté le 23-04-2022.

[12] J. Dordoigne, *Réseaux informatiques : Notions fondamentales*, ch. 9. ENI, 8. ed., 11 2019.

[13] E. Rescorla, "Diffie-Hellman Key Agreement Method." RFC 2631, June 1999.

[14] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3." RFC 8446, Aug. 2018.

[15] Wikipedia, "Cipher suite." https://en.wikipedia.org/wiki/Cipher_suite. Consulté le 11-04-2022.

[16] Cloudflare, "RPKI - The required cryptographic upgrade to BGP routing." https://blog.cloudflare.com/rpki/. Consulté le 16-03-2022.

[17] C. P. Loïc Miller, "A Taxonomy of Attacks Using BGP Blackholing." https://loicmiller.com/talk/a-taxonomy-of-attacks-using-bgp-blackholing/a-taxonomy-of-attacks-using-bgp-blackholing-slides.pdf. Consulté le 07-04-2022.

[18] Cloudflare, "What is a VPN ?." https://www.cloudflare.com/learning/access-management/what-is-a-vpn/. Consulté le 01-04-2022.

[19] Cloudflare, "What is IPsec ?." https://www.cloudflare.com/learning/network-layer/what-is-ipsec/. Consulté le 01-04-2022.

[20] C. Pignataro and N. McGill, "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values." RFC 5641, Aug. 2009.

[21] Wikipedia, "OpenVPN." https://en.wikipedia.org/wiki/OpenVPN. Consulté le 23-04-2022.

[22] Wikipedia, "Shor's algorithm." https://en.wikipedia.org/wiki/Shor%27s_algorithm. Consulté le 25-03-2022.

[23] Google, "IPv6 Adoption." https://www.google.com/intl/en/ipv6/statistics.html. Consulté le 13-04-2022.

[24] Remme, "How Blockchain addresses Public Key Infrastructure shortcomings." https://remme.io/blog/how-blockchain-addresses-public-key-infrastructure-shortcomings. Consulté le 14-04-2022.

[25] E. Rescorla, "HTTP Over TLS." RFC 2818, May 2000.

[26] A. O. Freier, P. Karlton, and P. C. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0." RFC 6101, Aug. 2011.