INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAT012

Thèse de doctorat

# Définition et conception d'un système optimal de la révocation de certificats

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique, Réseau, Sécurité

Thèse présentée et soutenue à Palaiseau, le 05 Juillet 2021, par

**ADJA ELLOH YVES CHRISTIAN**

Composition du Jury :

| | |
|---|---|
| Guy Pujolle<br>Professeur, Université Pierre-et-Marie-Curie | Président |
| Ken Chen<br>Professeur, Université Sorbonne Paris Nord | Rapporteur |
| Pascal Lorenz<br>Professeur, Université de Haute Alsace | Rapporteur |
| Ahmed Serhrouchni<br>Professeur (HDR), Télécom Paris | Directeur de thèse |
| Houda LABIOD<br>Professeure, Télécom Paris | Co-directrice de thèse |
| Badis Hammi<br>Professeur associé, EPITA | Invité |

# *REMERCIEMENTS*

Ces quatre dernières années m'ont permis d'atteindre mon rêve d'enfant, le doctorat. J'ai passé ces dernières années au sein d'une très belle équipe dont le soutient a été au-delà de mes attentes. Ma thèse en Cybersécurité m'a permis de découvrir le monde passionnant de la recherche de l'enseignement et à consolider mes perspectives futures. Le doctorat est un long et compliqué voyage qui ne peut aboutir sans le soutien des proches. C'est pourquoi, je tiens à remercier chaleureusement mes deux directeurs de thèse, Mr Ahmed Serhrouchni et Mme Houda Labiod d'avoir cru en moi en me permettant de faire cette thèse. Je n'aurai jamais pu proposer ce manuscrit sans leurs efforts.

Je tiens à remercier mon encadrant Badis, qui m'a, à la fois guidé et accompagné dans mes réflexions et travaux, mais qui a su me laisser libre de mes actions.

Je profite pour remercier mes collègues, Jean-Phillip, sara et Mohammed pour leurs précieux conseils et encouragements.

C'est avec amour que je remercie ma femme Gisele et mes deux enfants Liam et Naelle de remplir de bonheur ma vie, ainsi de me donner la force chaque matin d'aller de l'avant.

Grand merci à mes parents, Léocadie et Jacques pour leur soutien indéfectible morale et dans la prière pendant toutes ces années.

Je remercie également mes frères et amis : Naba, Issa, Annicet, Moussa d'avoir toujours été disponible pour moi dans les moments sombres de ma thèse.

Merci à Apôtre, maman Apôtre et la grande famille du CAPE pour vos prières. Puisse Dieu vous bénir au-delà de vos attentes pour tous ce que vous avez fait pour moi.

J'exprime ma gratitude à Mr Ken Chen, Mr Pascal Lorenz, Mr Hammi badis et Mr Guy Pujolle d'avoir accepté de faire partir de mon jury de thèse.

# Table des matières

# Table des figures

# Liste des tableaux

# Liste des abréviations

| | |
|---|---|
| **PKI** | **P**ublic **K**ey **I**nfrastructure |
| **VPKI** | **V**ehicular **P**ublic **K**ey **I**nfrastructure |
| **CRL** | **C**ertificate **R**evocation **L**ist |
| **TLS** | **T**ransport **L**ayer **P**rotocol |
| **CRLDP** | **C**ertificate **R**evocation **L**ist **D**istribution **P**oints |
| **ICRLDP** | **I**mproved **C**ertificate **R**evocation **L**ist **D**istribution **P**oints |
| **O-CRL** | **O**verissued **C**ertificate **R**evocation **L**ist |
| **d-CRL** | **d**elta **C**ertificate **R**evocation **L**ist |
| **CRT** | **C**ertificate **R**evocation **T**ree |
| **CRS** | **C**ertificate **R**evocation **S**ystem |
| **SCVP** | **S**imple **C**ertificate **V**alidation **P**rotocol |
| **OCSP** | **O**nline **C**ertificate **S**tatus **P**rotocol |
| **CA** | **C**ertification **A**uthority |
| **RCA** | **R**oot **C**ertification **A**uthority |
| **AA** | **A**uthorization **A**uthority |
| **EA** | **E**nrolement **A**uthority |
| **ECTL** | **E**xtended **C**ertificate **T**rust **L**ist |
| **CTL** | **C**ertificate **T**rust **L**ist |
| **TLM** | **T**rust **L**ist **M**anager |
| **CPOC** | **C**-ITS **P**oint **O**f **C**ontact |
| **MA** | **M**isbehavior **A**uthority |
| **DC** | **D**istribution **C**enter |
| **RSI** | **R**evocation **S**tatus **I**nformation |
| **LRSI** | **L**ight **R**evocation **S**tatus **I**nformation |
| **PGP** | **P**retty **G**ood **P**rivacy |
| **WoT** | **W**eb **O**f **T**rust |
| **Veins** | **V**ehicles **I**n **N**etwork **S**imulation |
| **CAM** | **C**ooperative **A**wareness **M**essage |
| **C-ITS** | **C**ooperative-**I**ntelligent **T**ransport **S**ystem |
| **DENM** | **D**ecentralized **E**nvironmental **N**otification **M**essage |
| **DoS** | **D**enial **o**f **S**ervice |
| **DSRC** | **D**edicated **S**hort **R**ange **C**ommunication |
| **ETSI** | **E**uropean **T**elecommunications **S**tandards **I**nstitute |
| **IEEE** | **I**nstitute **E**lectrical **E**lectronics **En**gineers |
| **SUMO** | **S**imulator **O**f **U**rban **M**obility |
| **ITS** | **I**ntelligent **T**ransport **S**ystem |
| **ITS-S** | **I**ntelligent **T**ransport **S**ystem **S**tation |
| **OBU** | **O**n-**B**oard **U**nit |
| **PC** | **P**seudonym **C**ertificate |
| **RSU** | **R**oad **S**ide **U**nit |
| **V2I** | **V**ehicle-**to**-**I**nfrastructure Communication |
| **V2V** | **V**ehicle-**to**-**V**ehicle Communication |
| **V2X** | **V**ehicle-**to**-**Everything** Communication |

| | |
|---|---|
| **DLT** | **D**istributed **L**edger **T**echnologie |
| **PoW** | **P**roof **O**f **W**ork |
| **PoS** | **P**roof **O**f **S**take |
| **DNS** | **D**omain **N**ame **S**ystem |

# Chapitre 1

# Introduction

Ce chapitre est le chapitre introductif, nous allons détailler le contexte, la problématique, les motivations et les objectifs de cette thése.

## 1.1 Contexte

La Public Key Infrastructure(PKI) est une infrastructure à clés publique dont l'objectif est de répondre à des besoins tel que : l'authentification de clés publiques, le contrôle d'accès et les fonctions d'autorisation, l'identification et l'authentification déterministe et automatisée. La prise en charge de ces services détermine les attributs contenus dans le certificat émis par l'autorité de certification (CA), ainsi que les informations de contrôle auxiliaires telles que la politique et les contraintes de chemin de certification. Le certificat à clés publiques à plusieurs attributs relatifs au propriétaire et à l'émetteur du certificat. La validation du certificat passe par la vérification de ces attributs (le sujet, la date, la signature, la révocation, la politique de certification, etc.). Un certificat délivré par une autorité de certification est censé être utilisé pendant toute sa période de validité. Cependant, diverses circonstances peuvent interrompre cette validité. Ces circonstances comprennent le changement de nom, le changement d'association entre le sujet et le CA, la compromission ou une suspicion de compromission de la clé privée correspondante. Sous telles circonstances, l'autorité de certification doit révoquer le certificat. Les services de validation et de vérification de la révocation des certificats sont attendus ou nécessaires dans plusieurs contextes, nous pouvons en citer certains parmi tant d'autres : les communications véhiculaires, le (WWW), l'authentification des utilisateurs et etc. Les communications véhiculaires sont au centre des véhicules de demain et d'une manière plus générale des smart-cities. La sécurisation de ces réseaux est un élément critique au vu des services en perspective. La sécurité des échanges inter-véhicules est basée notamment sur la signature numérique. Cette même signature nécessite d'une infrastructure d'échange de clés (PKI). Pour garantir l'anonymat des véhicules, des certificats à pseudonymes ont été standardisés. Afin de garantir la non-traçabilité des véhicules, ces dernières doivent changer fréquemment de certificats pseudonymes. Les services de validations et de vérifications de la révocation des certificats doivent être assurés par la PKI sur tous les certificats pseudonymes émis. Le RFC 5280 **[X.509_v2_CR]** défini deux catégories de méthodes de révocation, les méthodes qui permettent la vérification de la révocation des certificats en ligne et hors ligne. Dans cette dernière catégorie nous avons une méthode de révocation qui implique la publication périodique de la part de la CA d'une structure de données signée appelée CRL(Certificate révocation list). Ces deux approches sont matures mais couteuse en temps et en volume et les communications véhiculaires apportent de nouvelles contraintes. Ces travaux portent sur la vérification de la révocation des

certificats X509 et aussi bien des certificats à pseudonyme utilisé dans les communications véhiculaires. Notre objectif est de réduire en latence et en volume l'impact de la vérification de la révocation sur la validation des certificats X509 et pseudonymes. Nous avons dans ce contexte proposé la méthode de révocation combinant la Blockchain et les filtres de Bloom impliquant pour optimiser la méthode CRLDP. Nous avons ensuite proposé la méthode de révocation basée sur des choix cryptographiques pour optimiser la vérification de la révocation dans le contexte véhiculaire. Ces deux approches proposées permettent un gain en terme de volume, de latence et de résilience tout en maintenant un niveau de sécurité égal voire supérieur aux solutions classiques. Nos contributions ont été validé par une implémentation.

## 1.2 Problématique

Nos travaux visent à réduire l'impact de la latence due à la vérification des certificats à clés publiques sur la sémantique des applications clientes de la PKI. La publique Key Infrastructure (PKI) est une infrastructure de confiance qui fournit plusieurs services de sécurité tel que la validation et la vérification de la révocation des certificats a clés publiques. Les certificats sont émis par une autorité de certification à ses clients pour authentifier leurs paires de clés publiques/privées. Chaque certificat a différents attributs dont une durée de validité. Cette durée d'être peut-être interrompue avant son expiration naturelle suite à une révocation de la part de l'autorité de certification pour plusieurs motifs tels que : l'expiration naturelle, la perte ou la compromission de la clé privée associée au certificat et dans le cas extrême la perte ou la compromission de la clé privée de l'autorité de certification. Les RFCs 5280 [08] et 6960[San+13] définissent des méthodes de révocation des certificats. Parmi ces dernières, nous pouvons citer deux méthodes. Nous avons une méthode qui implique la publication périodique de la part de la CA d'une structure de données signées appelées CRL. Cette dernière est une liste horodatée référençant les certificats révoqués par une autorité de certification. Elle est signée par la CA ou par l'émetteur de la CRL (CRL issuer) et mise gratuitement à disposition dans un référentiel public. Cette approche présente différentes faiblesses, telles que la granularité temporelle de la révocation qui est limitée à la période d'émission de la CRL et la latence due au téléchargement et au traitement de longues listes de révocation. Nous avons la méthode qui implique l'utilisation du protocole OCSP (Online Certificate Status Protocol) pour déterminer l'état (de révocation) des certificats à clé publique. Le protocole OCSP peut être utilisé pour répondre à certaines des exigences opérationnelles de fournir des informations de révocation plus récentes que ce qui est possible avec les CRLs et peut également être utilisé pour obtenir des informations d'état supplémentaires. Cette méthode impose de nouvelles exigences de sécurité telles que : le vérificateur du certificat doit faire confiance au service de validation en ligne. La méthode de révocation impliquant la CRL est la plus utilisée dans le cadre des certificats x509 et pseudonyme. Le contexte véhiculaire ajoute une nouvelle dimension, car les véhicules doivent s'authentifier avec plusieurs certificats de type pseudonyme pour garder l'anonymat pendant leurs échanges et éviter la traçabilité. Révoquer un véhicule signifie donc révoquer plusieurs certificats pseudonymes, ce qui induit une croissance exponentielle des listes de révocation. Traiter de longues listes de révocation impact considérablement la vérification de la révocation des certificats par des délais non négligeables. De plus, aucun système de distribution des listes de révocation des certificats pseudonymes n'est proposé par le standard de référence [ETS]. L'existence de toutes ces méthodes de révocation

montre aussi bien la pertinence du sujet que la l'absence d'une méthode idéale. Effectivement, ces approches sont matures mais couteuses en volume et en temps ce qui impact la sémantique des applications.

## 1.3 Objectifs

Ces travaux ont pour objectif de réduire l'impact de la latence de la vérification de la révocation des certificats sur les applications clientes de la PKI. Ceci en optimisant en temps et en volume le service de vérification de la révocation des certificats de la PKI. Nous visons à optimiser en fonction du contexte, la méthode de révocation impliquant la publication d'une CRL. Nous voulons dans la première phase, concevoir une méthode de révocation impliquant la distribution de la CRL par l'autorité de confiance dans une Blockchain. Nous allons ainsi utiliser le système de distribution de la Blockchain pour distribuer les listes de révocation aux utilisateurs. Nous visons à créer une nouvelle structure de données pour compresser les listes de révocation de sorte à réduire les délais de téléchargement et de traitement. Nous savons bien que l'environnement véhiculaire introduit une nouvelle dimension en terme de volume de certificats à vérifier, nous allons donc concevoir une méthode de révocation capable de prendre en compte cette contrainte. Notre méthode de révocation va utiliser les filtres de bloom pour comprimer les listes de révocation. Nous allons aussi fragmenter les listes de révocation pour avoir plusieurs points de distribution. Nous comptons ainsi réduire la quantité de données à télécharger et la latence de téléchargement et de traitement des listes de révocation.

## 1.4 Contribution

Nous avons conçu, testé et évalué deux nouvelle approches de révocation pour la vérification de la révocation des certificats X509 et a pseudonymes. Nos contributions se divisent principalement en deux parties chacune relative à un contexte bien précis.

— Dans la première partie, nous avions pour objectif d'optimiser en volume, en temps et en résilience les méthodes de révocation utilisées dans la PKI X509. Nous avons ainsi combiné la résilience de la Blockchain avec les techniques de compression des filtres de bloom pour atteindre nos objectifs. Ceci dit, nous avons repris et optimisé la solution Certificate Revocation List Distribution (CRLDP), qui implique la distribution des listes de révocation sur plusieurs points de distribution afin d'avoir une architecture distribuée. Nous avons créé deux nouvelles structures de données appelées RSI (Revocation Status Information) et LRSI (Light Revocation Status Information). Nous avons utilisé le protocole de distribution de la Blockchain pour distribuer les listes de révocation. Les listes de révocation ont été distribuées sur plusieurs points de distribution tous situés dans la Blockchain. Les différents points de distribution ont été comprimés grâce aux filtres de bloom pour optimiser le stockage, la vérification et le téléchargement. Cette solution nous a permis d'atteindre nos objectifs. Nous avons ensuite testé et évalué notre solution à travers différentes métriques définies dans le chapitre 3. Ces travaux ont été publiés dans un journal, Annexe A, publication 3.

— Dans la deuxième partie des travaux, nous avons utilisé un procédé cryptographique pour optimiser la gestion de la révocation des certificats pseudonymes.

Nous sommes arrivés à la création de la méthode de révocation nommé Improved Certificate Revocation List Distribution Points (ICRLDP). Cette méthode reprend la notion de points de distribution (fragments) utilisé dans l'approche CRLDP. La structure de révocation, c'est-à-dire la CRL est restée inchangée, nous avons juste modifié les données à y mettre. Le ICRLDP a été conçu afin qu'il puisse gérer les listes volumineuses de révocation des certificats à pseudonymes sans de fortes répercussions en terme de latence sur les applications clientes. Nous avons fait usage de la cryptographie symétrique et asymétrique pour réduire à une simple clé symétrique le nombre d'informations à insérer dans les listes de révocation en cas de révocation d'un véhicule et non des centaines de certificats. Ceci a permis de compacter les différents fragments pour mieux les traiter et les distribuer. Notre solution nous permet de réduire l'impact de la vérification de la révocation sur la validation des certificats à pseudonyme. Ces travaux ont été publiés dans une conférence internationale, Annexe A, publication 4.

— Pour finir nous avons intégré une couche révocation dans le simulateur Veins. Effectivement, le simulateur ne contenait ni de couche certificat ni de couche révocation, ce que nous avons dû intégrer nous-même. Ces différentes intégrations nous ont permis d'avoir un cadre virtuel pour, simuler, tester et évaluer notre deuxième proposition.

## 1.5 Organisation du manuscrit

La suite du manuscrit est divisée en chapitres structurés selon l'ordre suivant. Le chapitre 2 est le chapitre de l'état de l'art qui est divisé en deux parties :

— Dans la première partie, nous avons fait une présentation et une analyse de l'architecture et des services de la PKI X509. Puis une analyse approfondie de la révocation, des méthodes de révocation et des comparaisons. Ensuite une analyse de l'application TLS et ses différents services.

— Dans la deuxième partie, nous avons fait une présentation des communications véhiculaires, notamment l'architecture, le fonctionnement, les acteurs et les composants. Ensuite, une analyse de la sécurité notamment, les menaces, les attaques, les architectures et les standards de sécurité.

Le Chapitre 3 est le chapitre dans lequel nous détaillons notre contribution. Le chapitre est divisé en deux parties.

— Notre première contribution est une methode de révocation pour une mise en œuvre dans la PKI x509.

— Notre deuxième contribution est une méthode de révocation pour une mise en œuvre dans la VPKI. Chaque partie est précédée par un état de l'art sur les contributions et autres travaux de recherche sue le même sujet.

Le chapitre 4 est celui de l'évaluation et des tests. Nous y détaillons la méthodologie utilisée pour tester nos deux solutions, suivi d'une analyse des résultats des tests.

Le chapitre 5 est le chapitre conclusif, il contient nos observations, nos conclusions et les perspectives futures.

# Chapitre 2

# Etat de l'art

## 2.1 Analyse : Infrastructure de confiance et Blockchain

Ce chapitre a pour but de décrire l'environnement théorique et technique de ces travaux de thèse et d'analyser les solutions et standards déjà existants pour faire face à la problématique abordée.

### 2.1.1 Les services et mécanismes de base de la sécurité

La Public Key Infrastructure (PKI) est une infrastructure qui assure un certain nombre de services et mécanismes de sécurité pour les utilisateurs. Ces services et mécanismes de sécurité sont les suivants :

— **La confidentialité** : La confidentialité [18] est un service essentiel pour assurer qu'une communication soit accessible seulement aux parties prenantes de la communication. C'est une protection contre les menaces qui peuvent causer la divulgation non autorisée d'informations alors qu'il faut veiller au caractère privé de l'information. Elle est principalement basée sur la cryptographie, en particulier, les algorithmes de chiffrement symétriques ou asymétriques.

— **L'intégrité** : Ce service de sécurité permet d'assurer que les données échangées ne soient pas soumises à une altération volontaire ou accidentelle. Il permet aux destinataires de détecter les manipulations de données effectuées par des entités non autorisées.

— **L'authentification** : Ce service de sécurité permet aux entités impliquées dans une communication de se prouver mutuellement leurs identités et de s'assurer mutuellement du droit à la communication. *Authentification = Identification + Vérification*

— **La non-répudiation** : C'est un service de protection contre l'éventualité qu'une partie prenante d'un échange puisse nier un de ses actes.

## 2.2 Public Key Infrastructure (PKI) X509 / PKIX

La PKI X509 est une infrastructure de confiance fournissant des services de sécurité aux utilisateurs à travers des certificats à clés publiques. L'objectif de la (PKIX) Internet est de répondre à des besoins : d'identification et d'authentification déterministes et automatisés, de contrôle d'accès et de fonctions d'autorisation. La prise en charge de ces services détermine les attributs contenus dans le certificat, ainsi

que les informations de contrôle auxiliaires telles que la politique et les contraintes du chemin de certification. Ces différents services offerts permettent aux utilisateurs d'authentifier leurs paires de clés et d'échanger de façon sécurisée des informations dans le réseau. Les applications clientes de la PKI sont nombreuses, nous pouvons citer des protocoles de sécurité standard anciens ou récents : 802.1x, IPSEC, TLS etc. Nous pouvons aussi citer le (WWW) et la messagerie électronique. Les services offerts parla PKI sont incontournables dans la sécurisation des échanges sur un médium non sécurisé.

### 2.2.1 L'architecture de la PKI

Les entités composantes de la PKI sont les suivantes :

— Une autorité de confiance appelée CA (CA). Il peut y avoir plusieurs autorités de confiance dans une PKI, dont une autorité racine,la plus haute autorité de la hiérarchie et des autorités subordonnés (SubCA). La RCA fournit les services nécessaires pour authentifier les clés publiques des utilisateurs et certifier les autorités inférieures (SubCA).

— L'autorité d'enregistrement (RA) est une SubCA. C'est l'autorité à laquelle le RootCA délègue la gestion des requêtes d'enregistrement des utilisateurs. Cette Autorité est optionnelle dans l'architecture de la PKI.

— Le CRL Issuer (l'émetteur des listes de révocation) est une entité optionnelle auquel le RootCA peut déléguer la tâche de la gestion des listes de révocation.

— Le répertoire (The repository) est un système distribué de stockage des certificats utilisateur et des listes de révocation. Il est aussi utilisé pour la distribution des certificats et des listes de révocation.

— L'entité finale : C'est l'utilisateur des services offerts par la PKI.

#### 2.2.1.1 Les protocoles de gestion

Les protocoles de gestion sont nécessaires pour prendre en charge les interactions en ligne entre l'utilisateur des services de la PKI et les entités de la PKI. Par exemple, un protocole de gestion peut être utilisé entre une autorité de certification et un système client auquel une paire de clés est associée, ou entre deux autorités de certification qui se certifient mutuellement. L'ensemble des fonctions qui doivent potentiellement être prises en charge par les protocoles de gestion sont les suivantes :

— **L'enregistrement :** il s'agit du processus par lequel un utilisateur se fait d'abord connaître à une autorité de certification (directement ou via une autorité d'enregistrement (RA), avant que cette autorité de certification n'émette un ou plusieurs certificats pour cet utilisateur.

— **L'initialisation :** avant qu'un système client puisse fonctionner en toute sécurité, il est nécessaire d'installer des éléments qui sont en relation avec des clés stockées ailleurs dans l'infrastructure. Par exemple, le client doit être initialisé en toute sécurité avec la clé publique et d'autres informations garanties de la ou des autorités de certification de confiance, à utiliser dans la validation des chemins de certification. En outre, un client doit généralement être initialisé avec sa ou ses propres paires de clés.

— **La certification :** il s'agit du processus dans lequel une autorité de certification émet un certificat pour la clé publique d'un utilisateur et renvoie ce certificat au système client de l'utilisateur et / ou publie ce certificat dans un référentiel.

— **La récupération de pair de clés :** ce service est optionnel, les clés du client de l'utilisateur (par exemple, la clé privée d'un utilisateur utilisée à des fins de chiffrement) peuvent être sauvegardées par une CA ou un système de sauvegarde de clé. Si un utilisateur a besoin de récupérer ces éléments de clé sauvegardés, un protocole d'échange en ligne peut être nécessaire pour prendre en charge cette récupération.

— **La mise à jour des clés :** Toutes les paires de clés des utilisateurs comme des entités de la PKI doivent être mises à jour régulièrement, c'est-à-dire remplacées par une nouvelle paire de clés et de nouveaux certificats.

— **La demande de révocation :** Une entité autorisée peut informer une autorité de certification d'une situation anormale nécessitant la révocation d'un certificat.

— **La certification croisée :** Deux autorités de certification échangent des informations utilisées pour établir une certification croisée. Un certificat croisé est un certificat émis par une autorité de certification à une autre autorité de certification qui contient une clé de signature de l'autorité de certification utilisée pour émettre les certificats.

Les protocoles en ligne ne sont pas le seul moyen de mettre en œuvre les fonctions de gestion. Pour toutes les fonctions, il existe des méthodes locales permettant d'obtenir le même résultat. Par exemple, lorsque des jetons matériels sont utilisés, de nombreuses fonctions peuvent être réalisées dans le cadre de la livraison de jetons physiques. En outre, certaines des fonctions de gestion peuvent être combinées en un seul échange. En particulier, deux ou plusieurs des fonctions d'enregistrement, d'initialisation et de certification peuvent être combiné en un seul protocole d'echange.

#### 2.2.1.2 Les protocoles operationnels

Les protocoles opérationnels sont nécessaires pour distribuer les certificats et les listes de révocation aux utilisateurs. Il existe des alternatives telles que ceux qui sont basés sur LDAP, HTTP, FTTP ou X.500.

#### 2.2.1.3 Les modèles de confiance

Toutes les entités d'une PKI forment un domaine de confiance distinct. Au sein de la PKI, la relation de confiance entre le RootCA, les CAs subordonnées et les utilisateurs varie, on parle plus couramment de modèle de confiance de la PKI. Il existe plusieurs modèles de confiance que nous citons maintenant :

— Le **Modéle de confiance Hiérarchique** : Le modèle de confiance hiérarchique est une structure arborescente inversée, à la racine se situe l'autorité de confiance suprême à qui tout le monde fait confiance, la RootCA. La RootCA certifie toutes les autorités de niveau inférieur dans la hiérarchie de l'arbre, elle émet le certificat des autorités subordonnées (subCA) et si nécessaire ceux des utilisateurs. Ensuite au niveau inférieur, c'est-à-dire au niveau des branches, on peut y trouver des autorités de certification subordonnées (subCA). Il peut y avoir sous certaines conditions plusieurs niveaux de certification subordonnée. Les subCAs émettent les certificats utilisateur et ceux des subCAs de niveau inférieur. La presence des subCAS n'est pas obligatoire. Au niveau des feuilles se trouve l'utilisateur qui est le consommateur des services de certification. Toutes les entités du modèle doivent faire confiance à l'autorité de certification racine et conserver son certificat. En général, les utilisateurs conservent

les certificats de toutes les CAs inclus dans le chemin de certification de leurs certificat.
Le modèle hiérarchique est schématisé dans la figure 2.1.

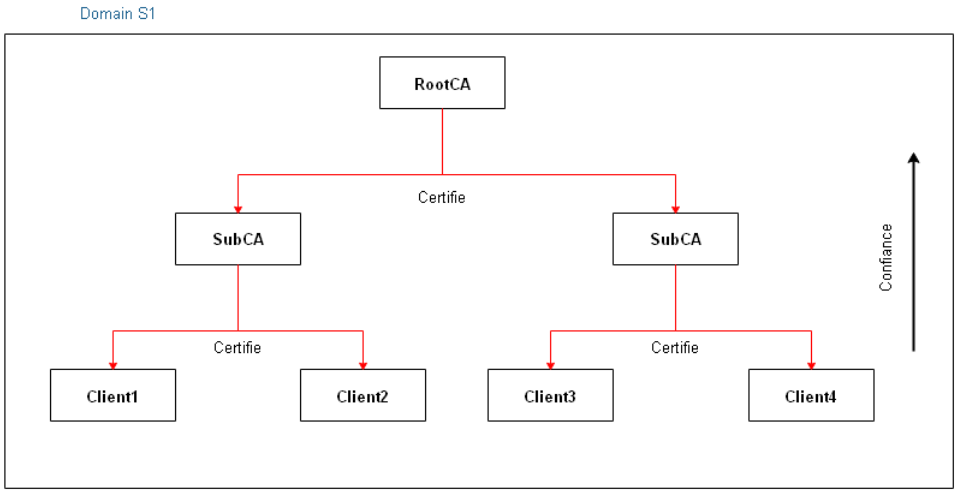



FIGURE 2.1 – Modéle de confiance hiérachique

— Le **Peer Trust Model (modèle de confiance entre paires)** est un modèle de confiance établi entre deux PKIs différentes. Précisément, il est mis en œuvre dans une communication entre deux utilisateurs appartenant au domaine d'influence de deux autorités racine différentes (deux PKIs différentes), ce qu'on appelle couramment la **cross-certification** (certification croisée). Dans ce cas, les deux RCAs peuvent établir une relation de confiance mutuelle ou unilatérale. C'est-à-dire que les deux RCAs s'entre-émettent des certificats dans le cas de relation mutuelle et une émission de certificat dans un seul sens dans le cas de relation de confiance unilatérale. Si les deux RCAs appartiennent au même domaine on parle de **domain cross-certification** et dans le cas où ils appartiennent à des domaine différents on parle de **inter domain cross-certification**. La figure 2.2 schématise le modèle de confiance dans les deux configurations **inter domain cross-verification** et **domain cross-certification**. La plus importante caractéristique de ces modèles de confiance est leur flexibilité, ce qui rend son extension très pratique. Cette flexibilité est aussi son point faible, avec le nombre croissant de CAs, car la construction du chemin de certification devient une tâche très difficile. De plus, il peut apparaître plusieurs chemins de certification, il peut y avoir une boucle infinie, ce qui rend difficile dans certains cas la vérification d'un certificat, augmentant ainsi la charge pour les utilisateurs. Par conséquent, un tel modèle est applicable à petite échelle.

— **Network Trust Model**, également connu sous le nom de modèle de confiance réseau, ce modèle de confiance est issu du concept d'Internet. Dans ce modèle il n'y a plus de hiérarchie parce que l'utilisateur devient lui-même sa propre CA, les certificats sont auto-signés. Ce modèle de confiance a l'avantage d'être simple, pratique et facile à utiliser. L'inconvénient de ce modèle est l'absence du service de révocation des clés. De plus, toute la gestion des clés est laissé à l'utilisateur. L'utilisateur doit lui-même assurer la sécurité de ses clés. La figure 2.3 schématise le modèle de confiance.

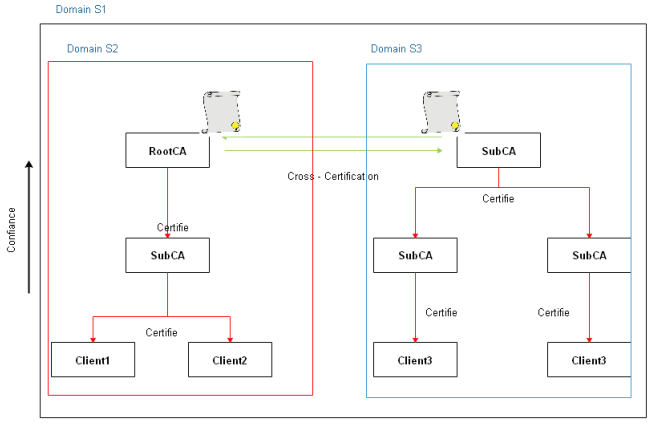



FIGURE 2.2 – Modéle de confiance entre paires

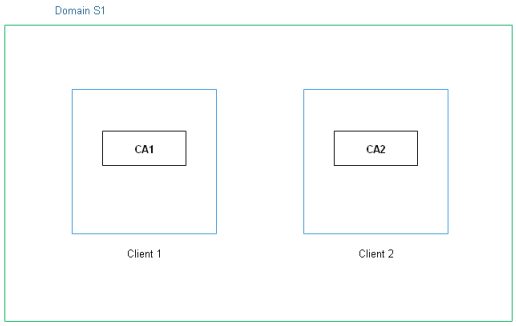



FIGURE 2.3 – Modéle de confiance réseau

— Le **Modèle de confiance hybride** est en pratique une combinaison de plusieurs des modèles ci-dessus. C'est-à-dire par exemple un modèle hiérarchique combiné avec un modèle Peer Trust Model. Le Peer Trust Model complète le modèle hiérarchique avec la certification croisée, mais le modèle hybride à la fin conserve les inconvénients du modèle hiérarchique.

### 2.2.2 Les standards PKI X509

La PKI X509 compte une famille de standards dont le but est de développer un profil pour faciliter l'utilisation de certificats X.509 dans les applications Internet. Le premier standard de la famille a été publié par ITU-T X.509 (anciennement CCITT X.509) ou ISO / CEI 9594-8s en 1988 dans le cadre des recommandations de l'annuaire X.500, définissant un format de certificat standard [X.509]. Parmis les differents standards de la PKI X509 nous pouvons citer le RFC 5280 [Coo+08] et le 6960. Le standard 5280 décrit la sémantique des certificats et des listes de révocation (CRLs) pour la PKI Internet. L'objectif de ce standard est de référencer un profil qui facilite l'utilisation des certificats X.509 dans le cadre des applications du réseau internet. Des applications telles que : le (WWW), la messagerie électronique, l'authentification des utilisateurs et IPSEC. Le standard 6960 spécifie un protocole pour vérifier la révocation d'un certificat numérique X509 sans nécessiter des listes de révocation des certificats (CRL). Le standard 6961, est une extension du standard 6960, il propose une nouvelle extension pour introduire le protocole OCSP Stanpling.

#### 2.2.2.1 X.509 v3

Le structure du certificat proposé par la norme de 1988 est la version 1 (v1). Lorsque X.500 a été révisé en 1993, deux autres champs ont été ajoutés, le Issuer unique identifier et Subject unique identifier, ce qui a donné le format de la version 2 (v2). La version actuelle est la version 3 (v3), cette dernière ajoute les extensions à la précédente version 2 (v2).

La structure ASN.1 du certificat X509 est la suivante :

```
Certificate  ::=  SEQUENCE  {
       tbsCertificate       TBSCertificate,
       signatureAlgorithm   AlgorithmIdentifier,
       signatureValue       BIT STRING  }

  TBSCertificate  ::=  SEQUENCE  {
       version         [0]  EXPLICIT Version DEFAULT v1,
       serialNumber         CertificateSerialNumber,
       signature            AlgorithmIdentifier,
       issuer               Name,
       validity             Validity,
       subject              Name,
       subjectPublicKeyInfo SubjectPublicKeyInfo,
       issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                            -- If present, version MUST be v2 or v3
                            subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                            -- If present, version MUST be v2 or v3
       extensions      [3]  EXPLICIT Extensions OPTIONAL
                            -- If present, version MUST be v3
       }

  Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }

  CertificateSerialNumber  ::=  INTEGER

  Validity ::= SEQUENCE {
       notBefore      Time,
       notAfter       Time }

  Time ::= CHOICE {
       utcTime        UTCTime,
       generalTime    GeneralizedTime }

  UniqueIdentifier  ::=  BIT STRING

  SubjectPublicKeyInfo  ::=  SEQUENCE  {
       algorithm            AlgorithmIdentifier,
       subjectPublicKey     BIT STRING  }
```

Les extensions sont la grande nouveauté qu'ajoute la version v3 du standard X509 décrit dans le tableau 2.1. L'extension rend flexible le standard car permet d'ajouter de nouveaux attributs au certificats sans apporter de modifications majeures au standard. Chaque extension se compose de trois champs :

| Fields | Champs | Definition |
|---|---|---|
| Version | Version | Version du certificat, nous utilisons actuellement la version 3 (v3) actuellement la version 3 (v3) |
| Serial Number | Numéro de série | C'est le numéro de série unique du certificat, c'est l'identifiant de ce dernier. Celui-ci sera posté dans la liste de révocation si le certificat est révoqué |
| signature AlgorithmIdentifier | L'identifiant de l'algorithme de signature | Ce champs permet de connaitre l'algorithme de signature qui sera utilisé par l'autorité de certification(CA) pour signer le certificat. |
| Issuer | Emetteur | Ce champs contient le nom de l'autorité de certification(CA) émettrice du certificat. |
| Validity | Période de validité | Ce champs contient les dates de début et de fin de validité du certificat. |
| Subject Name | Nom du propriétaire | Ce champs spécifie le nom en format DN dans la norme X.500 du propriétaire du certificat. C'est-à-dire celui à qui appartient la clé publique contenu dans le certificat. |
| subjectPublicKeyInfo | Clé publique | Ce champs contient la clé publique que le certificat lie à une identité et les algorithmes avec lesquels elle doit être utilisée. |
| issuerUniqueID | Identifiant unique de l'émetteur | Ce champ est optionnel, il permet d'ajouter un deuxième nom X.500 de (l'CA) dans le cas où celui-ci à un DN commun avec une autre CA. |
| subjectUniqueID | Identifiant unique du propriétaire | Ce champs est optionnel, il permet d'ajouter un deuxième nom X.500 du propriétaire dans le cas où celui-ci à un DN commun avec un autre utilisateur. |
| extensions | extensions | Ce champs est une liste d'extensions. |
| CA signature | La signature de CA | La signature es effectué avec la clé privée de l'CA sur tout le certificat sauf la signature elle-même. |

TABLE 2.1 – Format du certificat X509 v3

— Type : Ce champs décrit le type de données contenu dans le champs Value. C'est-à-dire une chaine de caractère, des données opaques (non interprétées), un nombre, pour la suite se référer au RFC [**cooper2008RFC**].

— Criticality : Le champs contient un flag d'un bit qui permet de marquer une extension comme critique ou pas.

— Value : Ce champs contient les données de l'extension. Le type de données contenu dans ces champs est décrit dans le champs Type.

Nous allons en citer quelques-unes relatives au cadre de nos travaux :

— cRLDistributionPoints : Cette extension décrit comment les listes de révocation sont obtenues. C'est une séquence de points de DistributionPoint. Le DistributionPoint est composé de trois champs, tous optionnels : distributionPoint, reasons et cRLIssuer. Ils sont tous optionnels mais l'un ne peut être présent sans l'autre. Le champ distributionPoint peut être omis si l'émetteur du certificat est le même que le CRLIssuer (DN). Si le champ DistributionPointName contient plusieurs valeurs, chaque nom décrit un mécanisme différent pour obtenir la même CRL. Par exemple, la même CRL pourrait être disponible via LDAP et HTTP. Si le champ distributionPoint contient un directoryName, l'entrée de ce directoryName contient la CRL actuelle émise par le cRLIssuer associé, pour les raisons associées. La CRL peut être obtenue par une l'application à partir de n'importe quel serveur d'annuaire configuré localement. Le protocole que l'application utilise pour accéder à l'annuaire (par exemple, DAP ou LDAP) est arbitraire. Si aucune raison n'est mentionnée, alors la CRL couvre toutes les raisons.

```
        id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::=  { id-ce 31 }

CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
```

```
        distributionPoint       [0]     DistributionPointName OPTIONAL,
        reasons                 [1]     ReasonFlags OPTIONAL,
        cRLIssuer               [2]     GeneralNames OPTIONAL }

    DistributionPointName ::= CHOICE {
        fullName                [0]     GeneralNames,
        nameRelativeToCRLIssuer [1]     RelativeDistinguishedName }
        ReasonFlags ::= BIT STRING {
        unused                  (0),
        keyCompromise           (1),
        cACompromise            (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold         (6),
        privilegeWithdrawn      (7),
        aACompromise            (8) }
```

— FreshestCRL : L'extension FreshestCRL identifie la manière dont la delta CRL obtenue. L'extension doit être marquée comme non critique par les autorités de certification conformes. L'extension est de type CRLDistributionPoints.

### 2.2.3 Transport Layer Security (TLS)

TLS est un protocole cryptographique de sécurisation des communications de bout en bout basé sur la PKI. Le but principal du protocole TLS est d'assurer la confidentialité et l'intégrité des données entre deux applications communicantes. Les propriétés de base des services de sécurité offerts par TLS sont les suivantes :

— Sécurité cryptographique : TLS doit être utilisé pour établir une connexion sécurisée entre deux parties.

— Interopérabilité : Les programmeurs indépendants devraient être en mesure de développer des applications hétérogènes capables d'échanger avec succès des paramètres cryptographiques à travers TLS sans connaître le code l'un de l'autre.

— Extensibilité : TLS a por objectif de fournir un cadre dans lequel il soit facile d'intégrer de nouvelles clés publiques et de nouvelles méthodes de chiffrement. Cela permettra également d'atteindre deux sous objectifs : éviter de créer un nouveau protocole (et risquer l'introduction de nouvelles faiblesses éventuelles) et éviter la mise en œuvre d'une toute nouvelle librairie de sécurité.

— Efficacité relative : les opérations cryptographiques ont tendance à être très gourmandes en ressources processeur, en particulier les opérations à clé publique. Pour cette raison, le protocole TLS a incorporé un schéma de mise en cache de session facultative pour réduire le nombre de connexions qui doivent être établies à partir de zéro. De plus, des précautions ont été prises pour réduire l'activité du réseau.

L'histoire de TLS sous le nom de SSL remonte à la fin des années 94, quand Netscape intègre la deuxième version de SSL (SSL 2.0) dans Netscape Navigator après avoir résolu de nombreux problèmes avec la première version. Microsoft a également introduit en 1995 un protocole de chiffrement nommé Private Communication

Technology (PCT) [Opp16] qui était très proche de SSL 2.0. La publication presque simultanée de ces deux protocoles de sécurité concurrents a créé beaucoup de confusion dans la communauté, car les applications devaient prendre en charge les deux protocoles pour permettre l'interopérabilité. Pour faire face à ce problème, l'Internet Engineering Task Force (IETF) a formé un groupe de travail en 1996 pour normaliser un protocole TLS unifié. Après une longue discussion avec les parties liées, la première version du protocole standard (TLS 1.0) est apparue en janvier 1999. En avril 2006, le protocole TLS version 1.1 (TLS 1.1) a été publié, suivi du TLS 1.2 en août 2008 qui intègre dans le protocole les extensions. Le protocole TLS 1.2 est spécifié dans le RFC5246 [**dierks2008RFC**]. La version la plus récente de TLS est le TLS 1.3. Cette dernière version suit les mêmes spécifications mais introduit des améliorations concernant les paramètres des algorithmes de chiffrement et de protocole TLS Handshake [RD18].

Le protocole TLS est composé de deux couches : le Handshake Protocol, qui est responsable de l'établissement, de la négociation des paramètres de connexion et de l'authentification des clés ; et le Record protocol, qui fournit un canal sécurisé pour la livraison des données. Précisément, le Handshake protocol établit les clés de session, qui sont utilisées pour crypter les données d'application dans le Record Protocol.

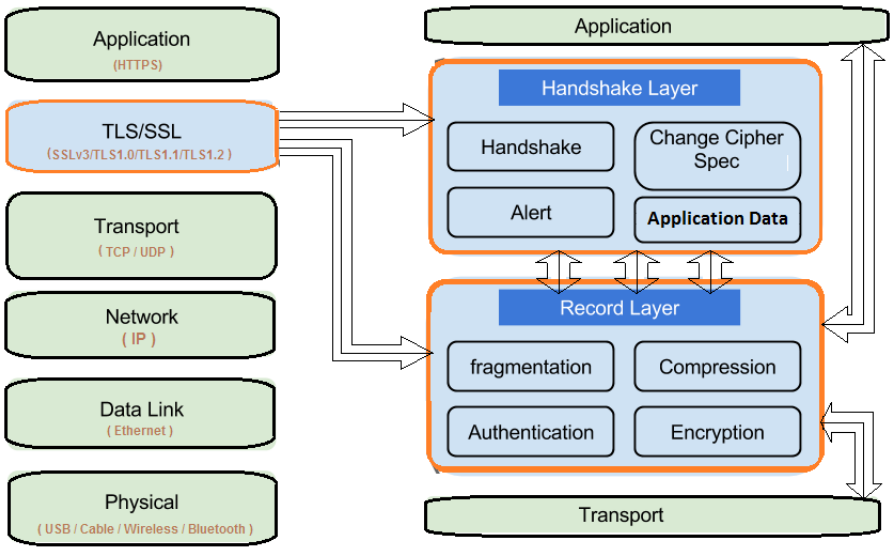



FIGURE 2.4 – Pile TLS

#### 2.2.3.1 TLS Handshake Protocol

Le protocole TLS Handshake, permet à un serveur et à un client de s'authentifier (mutuellement) et de négocier un algorithme de chiffrement et des clés cryptographiques avant que le protocole d'application ne transmette ou reçoive son premier octet de données. Le protocole TLS Handshake fournit une sécurité de connexion qui a trois propriétés de base :

— L'identité de l'interlocuteur peut être authentifiée en utilisant une cryptographie asymétrique à clé publique (par exemple, RSA [RSA], DSA [DSS], etc.). Cette authentification peut être rendue facultative, mais est généralement requise pour au moins un des pairs.

— La négociation d'un secret partagé est sécurisée, le secret négocié est indisponible pour les écoutes indiscrètes. Le secret partagé de toute connexion authentifiée ne peut être obtenu par une attaque man_in_the_middle.

— La négociation est fiable : aucun attaquant ne peut modifier les messages sans être détecté par les parties impliquées dans la communication.

Le Handshake Protocol est composé de quatre sous-protocoles : **handshake**, **Alert**, **ChangeCipherSpec** et **data**.

**2.2.3.1.1 TLS Handshake** Les paramètres cryptographiques de session sont produits par le protocole Handshake, qui fonctionne au-dessus de la couche Record TLS. Lorsqu'un client et un serveur TLS établissent la première connexion, ils conviennent d'une version du protocole, des algorithmes cryptographiques, s'authentifient si nécessaire mutuellement et utilisent des techniques de chiffrement à clé publique pour générer des secrets partagés. Le protocole **Handshake** est composé de plusieurs étapes :

— la première étape est celle de négociation dans lequelle le client et le serveur se mettent d'accord sur l'algorithme de chiffrement qui sera utilisée pour échanger les clés et les données, et sur les paramètres de connexion.

— puis l'étape d'authentification, dans laquelle le serveur et si requis le client se s'authentifient mutuellement par leurs certificats.

— Ensuite l'échange des clés de session, ou client et serveur s'échangent des nonces et un nombre spécial appelé pre-Master Secret. Ce nombre est utilisé pour générer la Master key, qui est ensuite utilisé pour le chiffrement des données

Toutes ces étapes se résument à une suite d'échanges de messages décrit dans la figure 2.5.

Les différents messages échangés sont les suivants :

— Le **ClientHello** est le premier message envoyé par celui qui demande une ouverture de connexion TLS, c'est-à-dire le client. Le message contient la liste des paramètres acceptés par le client, c'est-à-dire les ciphers suites, les extensions et un nonce.

— Le **ServerHello** est le message de réponse du serveur, il contient les paramètres acceptés parmi ceux qui ont été proposés par le client.

— Le **ServerCertificate** (Champs optionnel) message contient le certificat du serveur.

— Le **CertificateVerify** (Champs optionnel) est un message optionnel. Ce message est utilisé pour fournir une preuve explicite de la validité d'un certificat client. Ce message n'est envoyé qu'à la suite de l'envoie d'un certificat client qui a une capacité de signature.

— Le client répond à la requête de certificat du serveur en lui envoyant son certificat à travers le message **ClientCertificate**(Champs optionnel).

— Le Message **ServerHelloDone** indique la fin de la phasa **Hello**.

— Sur la base du nonce du serveur, le client génère un Pre-Master Secret aléatoire, puis le chiffre avec la clé publique donnée dans le certificat du serveur et l'envoie au serveur.

— Le client et le serveur génèrent un master secret à partir du Pre-Master Secret et s'échangent des valeurs aléatoires.

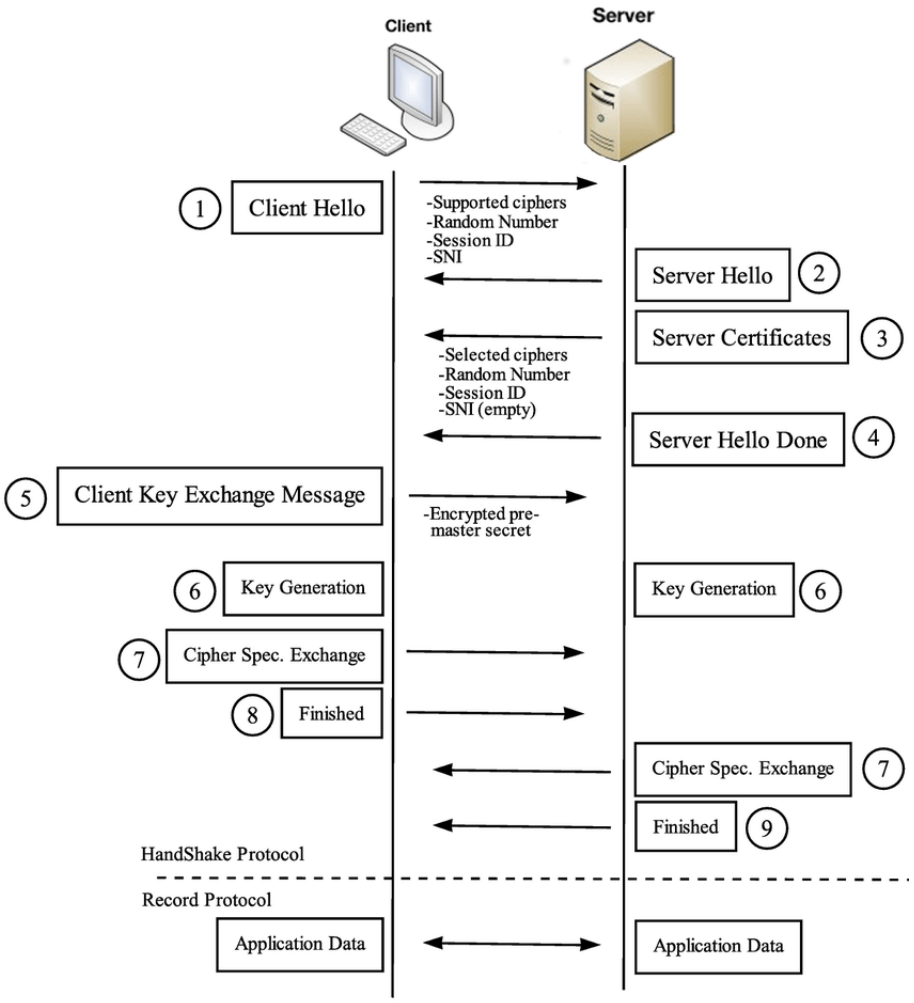



FIGURE 2.5 – TLS Handshake [Shb+16]

— Client et serveur s'échangent le message **ChangeCipherSpec** qui signale le début d'une session chiffrée avec le master secret.

— Le client envoie au serveur le message **Finished** pour s'assurer du bon déroulement du Handshake.

— Le serveur répond avec une message **Finished** pour confirmer le bon déroulement et la fin du de la phase Handshake.

**2.2.3.1.2 TLS ChangeCipherSpec** Le protocole de ChangeCipherSpec existe pour signaler les transitions dans les stratégies de chiffrement. Le protocole se compose d'un seul message, qui est chiffré et compressé dans la connexion courante, il ne peut être mis en attente. Le message consiste à un seul octet de valeur 1.

```
struct {
      enum { change_cipher_spec(1), (255) } type;
  } ChangeCipherSpec;
```

Le message ChangeCipherSpec est envoyé à la fois par le client et le serveur pour informer l'interlocuteur que les messages suivants seront protégés avec les clés nouvellement négociées. La réception de ce message déclenche une demande de la part

du récepteur vers couche Record, afin de copier l'état en attente de lecture dans l'état actuel de lecture. Immédiatement après l'envoi de ce message, l'expéditeur doit demander à la couche Record de faire de l'état d'écriture en attente l'état actif d'écriture. Le message ChangeCipherSpec est envoyé pendant le Handshake après que les paramètres de sécurité ont été convenus, mais avant que le message de vérification "Finished" soit envoyé.

#### 2.2.3.2 TLS Alert protocol

Le protocole alert est un des composants de la couche Handshake. Les messages d'alerte transmettent la gravité de l'alerte (avertissement ou fatal) et une description de l'alerte. Les messages d'alerte avec un niveau fatal entraînent l'arrêt immédiat de la connexion. Dans ce cas, d'autres connexions correspondant à la session peuvent se poursuivre, mais l'identifiant de session doit être invalidé, empêchant la session ayant échoué d'être utilisée pour établir de nouvelles connexions. Comme les autres messages, les messages d'alerte sont chiffrés et compressés, comme spécifié par l'état courante de la connexion.

#### 2.2.3.3 Record Protocol

Le protocole Record est utilisé pour l'encapsulation de divers protocoles de niveau supérieur. Il fournit une connexion sécurisée caractérisé par deux propriétés de base :

— Dans le cas d'une connexion confidentielle, la cryptographie symétrique est utilisée pour le chiffrement des données (par exemple, AES [AES], RC4 [SCH], etc.). Les clés symétriques sont générées pour chaque connexion et sont basées sur un secret négocié par un autre protocole (tel que le protocole TLS Handshake). Le protocole Record peut également être utilisé sans chiffrement.

— Dans le cas d'une connexion fiable, le transport de message comprend une vérification de l'intégrité du message à l'aide d'un HMAC. Les fonctions de hachage sécurisées (par exemple, SHA-1, etc.) sont utilisées pour les calculs MAC. Le protocole Record peut fonctionner sans MAC, mais n'est généralement utilisé que dans ce mode. Le protocole Handshake utilise le protocole Record comme moyen de transport pour la négociation des paramètres de sécurité.

A partir des couches supérieures, le Record protocol effectue ces différentes opérations sur les messages reçus :

— lecture des messages reçus,

— fragmentation des messages en petits blocks de données (chunks) pour faciliter les différents traitements suivants,

— calcul du MCA,

— chiffrement des blocks de données,

— Transmission des données aux couches inferieures après traitement.

Dans le sens contraire, partant des couches inferieures, plusieurs opérations sont effectuées dans la couche Record sur les messages reçus avant la transmission vers les couches supérieures.

— lecture des messages reçus,

— déchiffrement des blocks de données reçus,

— contrôle du MCA,

— décompression des messages si nécessaire,

— réassemblage des blocks de données,

— transmission des données aux couches supérieures.

L'authentification du serveur/client est une étape essentielle de la connexion TLS. Cette authentification est atteinte grâce à la PKI et a lieu pendant le TLS Handshake, lorsque client et/ou serveur échangent leurs certificats. Afin que la connexion TLS soit sécurisée, le client doit vérifier soigneusement que le certificat du serveur a été délivré par une autorité de certification valide, que le certificat n'est pas expiré (ou n'a pas été révoqué), le ou les noms énumérés dans le certificat soit cohérent avec le nom du domaine auquel le client se connecte et d'autres vérifications [Bla+06] [KD96].

## 2.3 Analyse : systèmes de tranports intelligent coopératifs (C-ITS)

La définition générale de communications véhiculaires fait référence à trois types de communications : les communications inter-véhiculaire (V-to-V), les communications intra-véhiculaire, les communications véhicule et infrastructure routier (V-to-I), les communications entre infrastructures routiers (I-to-I). Toutes ces communications sont généralisées sous le nom de V-2-X.


FIGURE 2.6 – Systémes de transports intelligents cooperatifs ( C-ITS) [ITS]

Ces communications utilisent des technologies sans fil ( IEEE 802.11p, 3G, 4G et 5G) et des mécanismes de sécurité. Certaines de ces technologies de communication supportent les communications à moyen et courte portée de manière répartie, par

exemple, DSRC (Dedicated Short Range Communications) et d'autres les communications à longue portée s'appuyant sur une infrastructure centralisée (par exemple, Cellulaire-V2X).

Dans ce sous-chapitre nous allons décrire l'architecture, les composants et les communications dans un système C-ITS.

### 2.3.1 Acteurs et composants

La technologie du système de positionnement mondial (GPS) a ouvert les portes à de nouvelles applications du système de transport intelligent coopératif (C-ITS) dans lequel les **véhicules connecté (CV)s** sont les avancées technologiques les plus prometteuses. En effet, les véhicules connectés (CV) introduisent des communications sans fil entre véhicules et infrastructures routières.

#### 2.3.1.1 Classification des stations ITS (ITS-S)

Nous avons quatre types de station ITS (ITS-S) communiquantes :

— Le premier type est appelé **véhicule connecté (CV)** et fournit des applications ITS aux véhicules, conducteurs et passagers. Chaque véhicule connecté intègre une unité embarquée appelé Onboard Unit (OBU) qui agit comme une passerelle V2X qui gère la communication radio, ainsi que le codage et décodage des messages V2X.

— Le deuxième type nommé **Central** comprend les serveurs des opérateurs de trafic, des Opérateurs routiers, des fournisseurs de services ou des fournisseurs de contenu.

— Le troisième type est appelé **Roadside**, qui regroupe les infrastructures routières connectées, telles que l'unité routière (RSU), les feux de circulation intelligents, le péage intelligent. Une station Roadside ITS fournit des applications ITS de manière indépendante ou coopérative aux ITS-Ss **Central** ou autres **Roadside** ITS-Ss.

— Le quatrième type est nommé **Personal** et regroupe les appareils personnels et nomades (par exemple, drones ou téléphones portables). Il fournit des applications ITS au propriétaire de l'appareil.

### 2.3.2 Structure et fonctionnement

Les systèmes de transport intelligents coopératifs (C-ITS) utilisent des technologies qui permettent aux véhicules de communiquer avec d'autres véhicules, avec des feux de circulation et des infrastructures routières, ainsi qu'avec d'autres usagers de la route [Mon20]. Le modèle de communication V2X définit plusieurs modes de communication :

— V-to-V : Cette appellation indique l'échange d'informations entre véhicules relatif aux alertes, aux obstacles sur la voie, informations sur le trafic, freinage brusque et etc.

— Intra-véhiculaire : Ce type de réseau indique les échanges d'informations à l'intérieur du véhicule.

— V-to-I : Ce type de réseau indique les échanges d'informations entre le véhicule et les infrastructures routiers.

— I-to-V : Ce type de réseau indique les échanges d'informations entre infrastructures routiers et le véhicule.

— I-to-I : Ce type de réseau indique les échanges d'informations entre les infrastructures routières.

— V-to-X : Ce réseau se réfère aux échanges entre véhicules et infrastructures routiers, nœuds cellulaires et toutes sortes de nœuds capable de communique avec le véhicule.

Nous définissons véhicule tout transporteur routier, c'est-à-dire une voiture, un bus, une bicyclette et etc. Chaque pays ou région suit son propre ensemble de normes définissant chacun des protocoles liés à tous les modes de communications. Nos travaux de thèse portent sur les normes définies par la European Telecommunications Standards Institute (ETSI).

### 2.3.3 Architecture de communication

La pile de communication est composée de 4 couches : Access Layer, Networking and Transport Layer, Facilities Layer et Applications layer. Ces couches sont bornées par deux couches verticales : Management et security comme le décrit le tableau (2.2) proposé par la contribution [Has18].

<table>
<tr><td rowspan="10">Management</td><td></td><td>ITS Station</td><td rowspan="10">Security</td></tr>
<tr><td></td><td>ITS Application</td></tr>
<tr><td>Application Layer</td><td rowspan="3">Facilities</td></tr>
<tr><td>Presentation Layer</td></tr>
<tr><td>Session Layer</td></tr>
<tr><td>Transport Layer</td><td rowspan="2">Networking and Transport</td></tr>
<tr><td>Network Layer</td></tr>
<tr><td>Data Link Layer</td><td rowspan="2">access</td></tr>
<tr><td>Physical Layer</td></tr>
</table>

TABLE 2.2 – Mapping entre les couches des architectures OSI et ETSI [Has18]

La **couche d'accès (access Layer)** utilise plusieurs technologies d'accès sans fil. Elle dépend des exigences de l'application (par exemple, d'importante bande passante pour une application en temps réel) et le contexte de l'environnement.

Les **couches réseau et transport (Network and Transport Layer)** prennent en charge les protocoles pour la diffusion des messages de la source à la destination. Cette couche est définie par un profil de communication qui contient au moins un protocole de transport et un protocole de mise en réseau.

La **couche d'installation (Facility Layer)** traite les messages V2X lors de leurs réception et de leurs émission. La couche prend en charge plusieurs types de messages. Chaque message à un rôle et une utilité bien précise. Par exemple, le message nommé CAM (Cooperative Awareness Message) a le rôle d'avertir les véhicules connectés environnant de la présence de l'émetteur du message. Le tableau 2.3 donne une breve description du message CAM.

La **couche application (Application Layer)** prend en charge les applications C-ITS. On distingue trois types d'applications : la sécurité routière (road safety), l'efficacité du trafic (traffic efficiency) et les services. Les applications de sécurité (safety applications) préviennent les accidents de la route et protègent la vie des usagers de

| Champs | Description |
|---|---|
| **Elements standards** | |
| DSRCmsgID | Identifiant du type de message |
| SecMark | Estampille temporelle |
| MsgCount | Numéro du message, dans le cas d'une séquence de messages |
| TemporaryID | Identifiant réseau |
| Latitude | Position : latitude |
| Longitude | Position : longitude |
| Elevation | Position par rapport au niveau de la mer |
| Speed | La vitesse de l'émetteur du message |
| Heading | Angle entre l'objet et le Nord |
| Yaw Rate | Cap par seconde |
| Lat. accel | accélération par rapport à la latitude |
| Long. accel | accélération par rapport à la longitude |
| Vet. accel | accélération par rapport à l'axe vertical |
| Positional accuracy | Précision à un écart type près |
| Brake System Status | Etat du système de freinage |
| Length | Longueur du Véhicule |
| Width | Largeur du Véhicule |
| **Métadonnées** | |
| SenderID | Identifiant de l'émetteur |
| Gentime | Date de la création du message |

TABLE 2.3 – Structure du message CAM [Mon20]

la route. Ces applications nécessitent d'une grande disponibilité des liens de communication (par exemple, la disponibilité des canaux radio) et d'une bande passante élevée (par exemple, une fréquence de réception élevée) pour atteindre une visualisation spatiale très précise et temporelle de tous les objets connectés environnants. Les applications d'efficacité du trafic (Traffic efficiency) optimisent l'itinéraire du conducteur. Par exemple, une application peut choisir un chemin optimal en termes de temps de conduite. Une application peut également adopter une conduite écologique en connaissant la durée de chaque feu de signalisation rencontré. Les services divertissent l'occupant du véhicule pendant son voyage. Ces applications nécessitent d'une large bande passante en raison de la grande quantité de données téléchargées. Par conséquent, la pile de communication utilisée regroupe les protocoles liés à l'utilisation d'Internet (p. lecture, diffusion multimédia).

## 2.4 La sécurité dans les communications V-2-X

Pour répondre aux exigences de sécurité dans un système il est nécessaire d'appréhender l'ensemble des contraintes qui pèsent sur ce système. Cette étape permet par la suite de quantifier les critères de sécurité. La mise en œuvre d'une politique de sécurité pour les communications V-2-X impose de bien comprendre et appréhender les caractéristiques et les contraintes.

### 2.4.1 Les architectures de sécurité

Il existe des organismes Européens et Américains qui ont créé leur propre architecture de sécurité basée sur la PKI pour les réseaux V-2-X. En Europe, nous avons l'ETSI [ETS12a] qui a défini son architecture de sécurité, le C-ITS Credential Management System (CCMS) pour les communications ITS (Cooperative Intelligent

Transport System (C-ITS)). Nous avons aussi le consortium (Car to Car communication consortium - C2CCC) qui à défini aussi une architecture de sécurité pour les systèmes de transports intelligents coopératifs (C-ITS) dans l'objectif d'harmoniser les différents standards. Aux Etats-Unis au sein du Vehicle Safety Communication (VSC) et du VSC-A (Vehicle Safety Communication - Applications), le NHTSA (National Highway Safety Administration a proposé le Security Credential Management System (SCMS) pour les communications véhiculaires.

### 2.4.2 Les services de sécurité

L'ETSI à définit à travers son standard de sécurité pour les communications ITS, les entitées fonctionnelles et leurs inter relations : EA(Enrollment Authority), AA(Authorization Authority) and ITS-S (Intelligent Transport System-Station).

La EA valide (après authentification) qu'un ITS-S soit fiable et donc peut accéder à une communication ITS. La AA donne à l'ITS-S l'autorisation sur des services spécifiques à travers des tickets d'autorisation (certificats). Le CI (Canonical Identifier) est un identifiant unique pour un ITS-S utilisé pour son l'inscription auprès d'une CA. Le tableau (2.4) proposé par la contribution [Has18] est un récapitulatif sur les services de sécurité offert par l'ETSI.

| **Services de sécurité** | **Architectures** |
|---|---|
| Authentification | Le service est atteint à travers la signature des messages |
| Confidentialité | Le service est atteint à travers le chiffrement symétrique et asymmétrique |
| Intégrité | Le service est atteint à travers les fonctions de hachage |
| Détection des responsabilités | Le service est atteint à travers la gestion à distance des comptes |
| Sécurité des messages | Le service est atteint à travers la PKI |
| Non-répudiation | Le service est atteint à travers la signature numérique |

TABLE 2.4 – Architecture de sécurité [Has18]

### 2.4.3 Standards

Il existe plusieurs standards de sécurité, parmi les plus importants nous citons, le IEEE 1609.2 et ceux de l'ETSI (TS et TR). Le standard de sécurité IEEE 1609.2 [16] présente des méthodes pour sécuriser les formats de message, les messages applicatifs et le traitement des messages utilisés par les équipements du WAVE(Wireless access in Vehicular Environments). Tous les problèmes de sécurité sont résolus par la PKI qui est un gestionnaire de clés et de certificats. Les exigences de sécurité dans ce standard telles que la confidentialité, l'authentification, la non-répudiation et l'intégrité sont assurées mais l'anonymat est limité et aucun mécanisme n'est défini pour la communication à sauts multiples en V-2-V.

Les standards ETSI [ETS][ETS12b][ETS12a] définissent les services de sécurité, l'architecture et la gestion de la sécurité des communications pour les stations ITS.

Nous souhaitons souligner que nous prenons en compte que les standards de sécurité ETSI.

Pour pouvoir sécuriser les communications véhiculaires, il est nécessaire de connaître les menaces possibles. Ainsi, nous détaillerons successivement les modèles d'attaquant et les attaques possibles, ainsi que les services et mécanismes de sécurité utilisés pour y répondre. Au-delà des problèmes de sécurité, les V-2-X soulèvent des contraintes temporelles. Par exemple, avec un environnement fortement dynamique, caractérisé par une topologie très changeante, et des connexions de courtes

durées, le déploiement d'une solution de sécurité doit faire face à des contraintes de temps et des configurations spécifiques.

### 2.4.4 ETSI PKI

Dans ce chapitre nous allons parler du C-ITS Credential Management System (CCMS) qui est une infrastructure de sécurité basée sur la PKI (Public Key Infrastructure) définit par ETSI pour le système C-ITS. La politique de certification (Certificate policy) Européen définit le modèle de confiance pour l'infrastructure de sécurité Européen ( EU CCMS) dans le cadre du C-ITS; ainsi que les exigences juridiques et techniques pour la gestion des certificats, pour les applications du système C-ITS par les entités émettrices et leur usage par les utilisateurs en Europe.

L'architecture de l' EU CCMS et les services de sécurité assurés sont présentés dans les sections suivantes.

#### 2.4.4.1 Architecture

L'architecture EU CCMS figure 2.7 est composée de deux couches, la couche de gouvernance (governance layer) qui est la couche supérieure, où nous trouvons les hautes autorités. Puis, nous avons la couche inférieure, dans laquelle nous trouvons les entités fonctionnelles de la PKI.

##### 2.4.4.1.1 La couche supérieure : la Governance layer

Dans cette couche nous trouvons :

— Gestionnaire des listes de confiance (Trust List Manager - TLM) : C'est l'autorité qui signe le European Certificate Trust List (ECTL), qui est tout simplement la liste des autorité racines connues et de confiance.

— Point de contact C-ITS (C-ITS Point Of Contact - CPOC) est l'autorité en charge de collecter les certificats des CAs racines de confiance et de les transmettre au TLM.

— L'autorité des politiques de certification (Certificate Policy Authority - CPA) est en charge de désigner et autoriser le TLM et le CPOC. Il a la charge d'approuver/révoquer les certificats des autorités racines dans le TLM. Le CTL est la liste de révocation des autorités Racines. Le CPA est constitué de représentants d'acteurs publics et privés.

##### 2.4.4.1.2 La couche inférieure

La couche inferieur se compose des entités fonctionnelles de la PKI ETSI.

— Une autorité de confiance appelée Root CA. Celle-ci est l'autorité racine, la plus haute autorité de la hiérarchie. Elle fournit les services nécessaires pour authentifier l'identité des utilisateurs et certifier les autorités inférieurs (EA, AA).

— L'autorité d'inscription (Enrolement Authority - EA) délivre un certificat d'inscription (EC) pour l'ITS-S qui en fait requête. Les ECs sont considérés comme une preuve d'identité, ils sont utilisés pour identifier et authentifier l'ITS-S au sein de la PKI. Ils ne sont pas utilisés dans les communications V2V, ils ne sont utilisés que pour demander un ticket d'autorisation (AT). Ils sont communément appelés certificat à long terme.

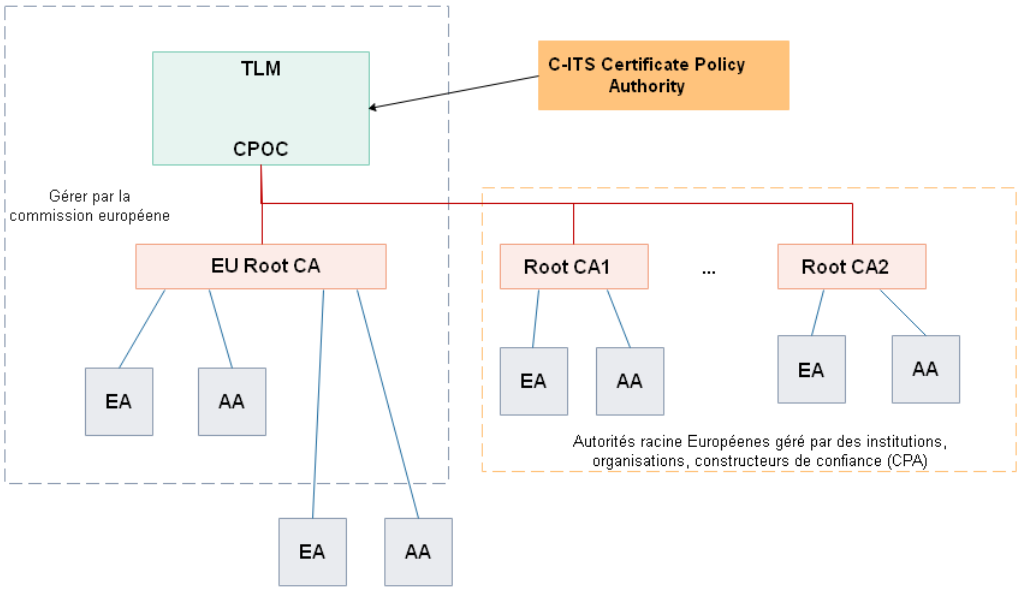



FIGURE 2.7 – Modèle de confiance EU CCMS

— L'autorité d'autorisation (Authorization Authority - AA) est certifiée par le Root CA. Elle émet et s'occupe de la gestion des certificats pour les stations ITS dans le cadre d'usages spécifiques. Le AA émet un ticket d'autorisation (AT) à l'ITS-S qui est utilisé pour signer les messages dans les communications V2V. Les ATs sont connus sous le nom de certificats pseudonymes. Les véhicules ont généralement un pool d'ATs valides qui sont modifiés pendant la communication pour empêcher le suivi du véhicule.

— Le Centre de Distribution (Distribution Center - DC) : fournit à la station ITS les listes de révocation des certificats pseudonymes (CRL) et pour les certificats long-terme(CTL) nécessaires pour valider les informations reçues provenant d'une autorité légitime et autorisée.

— Le Misbehavior Authority (MA) est l'autorité responsable de la détection des comportements illégaux et non autorisés vis-à-vis des protocoles de communications.

— Le Fabricant (Manufacturer) installe les informations nécessaires à la gestion de la sécurité dans l'ITS-S lors de sa production.

#### 2.4.4.2 Les Certificats

Dans cette sous-section nous allons décrire la structure des certificats utilisés dans le système C-ITS.

##### 2.4.4.2.1 Types de certificat

Les certificats pour un véhicule sont émis par une autorité de certification (RA, AA), la plupart du temps par une AA. Cette AA doit être de confiance pour les deux entités impliquées dans un échange pour que les parties puisse s'authentifier mutuellement. C'est la raison pour laquelle AA se réfère à une RootCA, qui est censée

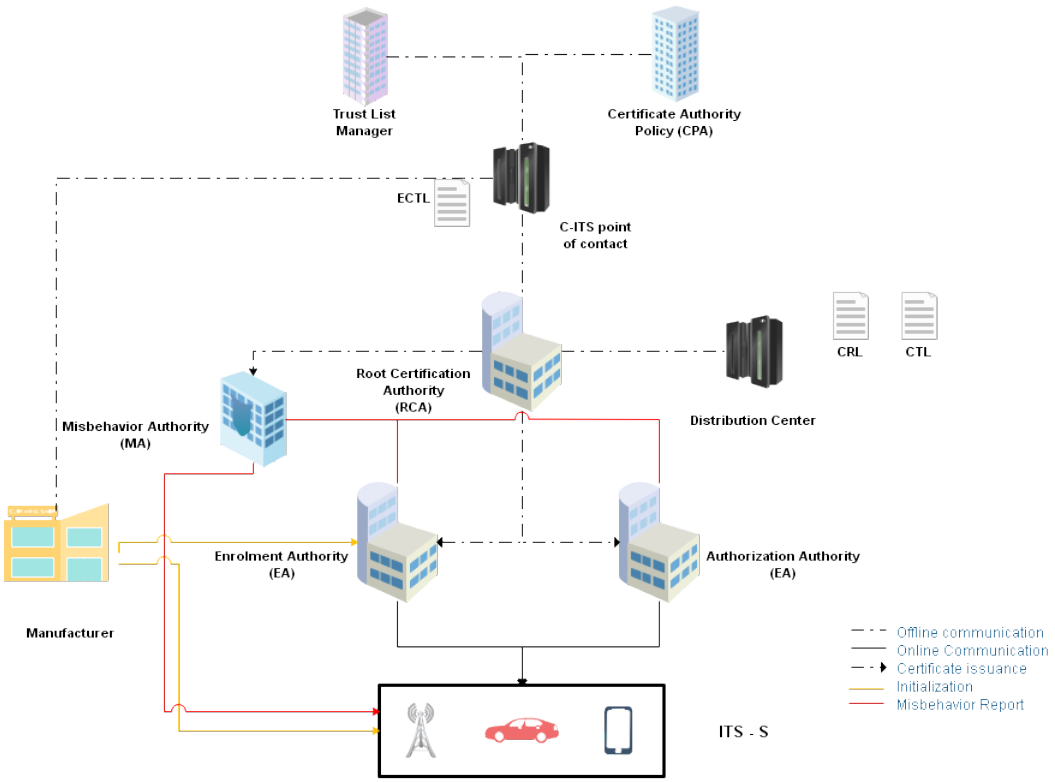



FIGURE 2.8 – Architecture PKI CCMS

être de confiance pour tous. Le certificat donne aussi des informations sur les droits et privilèges d'une station ITS. La PKI véhiculaire est caractérisé par deux types des certificats : les certificats long-terme et les certificats court-terme.

— *le certificat long-terme (EC)* est émis une seule fois, à la création du véhicule. Celui-ci est sensé le garder toute sa vie, il identifie directement le véhicule. Ce certificat n'est pas utilisé pour s'authentifier dans les communications. Aucune approche de distribution n'est proposée par les standards ETSI pour les listes de révocation des certificats long-terme.

— *le certificat court-terme (AT)* est lié à celui à long-terme et est pseudonyme il ne peut être utilisé que de façon complémentaire avec celui de long-terme. Il n'identifie pas directement le véhicule mais c'est un pointeur au certificat long-terme pour des raisons de privacy. Il est difficile de remonter au certificat long-terme depuis le court-terme. Le certificat court-terme comme le nom l'indique est à courte durée, il est renouvelé périodiquement. Les standards ETSI ne propose aucune approche de distribution pour les listes de révocation des certificats pseudonymes.

Chaque véhicule détient ces deux types de certificat.

#### 2.4.4.2.2 Structure des certificats

La structure ASN1 générale de tous les certificats utilisés dans la PKI ETSI selon la version 1.3.1 du standard TS 103 097 est la suivante.

```
CertificateBase ::= SEQUENCE  {
version                  Uint8(3),
type                     CertificateType,
issuer                   IssuerIdentifier,
toBeSigned               ToBeSignedCertificate,
signature                Signature OPTIONAL }

ToBeSignedCertificate ::= SEQUENCE  {
id                      CertificateId,
crCAaId                 HashedId3,
crlSeries               CrlSeries,
validityPeriod          ValidityPeriod,
region                  GeographicRegion OPTIONAL,
assuranceLevel          SubjectAssurance OPTIONAL,
appPermissions          SequenceOfPsidSsp OPTIONAL,
certIssuePermissions    SequenceOfPsidGroupPermissions OPTIONAL,
certRequestPermissions SequenceOfPsidGroupPermissions OPTIONAL,
canRequestRollover      NULL OPTIONAL,
encryptionKey           PublicEncryptionKey OPTIONAL,
verifyKeyIndicator      VerificationKeyIndicator,
... }
```

### 2.4.5 IEEE PKI

Le NHTSA a proposé une architecture de sécurité basée sur la PKI [Why+13]. Elle contient des entités fonctionnelles pour l'inscription des OBUs (fonctions d'amorçage) dans le cadre de l'obtention des certificats long-terme et court-terme (fonctions pseudonymes). L'objectif premier est la gestion de la confiance. Nous pourrons y observer que deux types de messages sont utilisés pour les communication V2V : BSM (Basic Safety Message) et les messages pour la transmission d'informations de sécurité. Les messages BSM sont signés et accompagnés par le certificat de l'envoyeur, qui sera utilisé pour vérifier l'authenticité du message. Dans le cas des échanges entre le véhicule et le système *SCMS (Sécurity Certificate Management System)*, le chiffrement asymétrique ECIES (Elliptic Curve Integrated Encryption Scheme) est utilisé pour garantir la confidentialité et la signature ECDSA (Elliptic Curve Digital Signature Algorithm) pour assurer la validité et l'authenticité des messages. Dans le cas des communications intra SCMS (entre entités), il est plutôt utilisé le chiffrement symétrique AES-CCM (Advanced Encryption Standard-Counter with CBC-MCA) pour garantir la confidentialité et le MCA (Message Authentication Code) pour l'intégrité. Leurs combinaisons garanti l'authenticité des messages. Cette architecture de sécurité garanti la confidentialité contre les entités internes et externes ; un seul composant SCMS ne peut lier deux certificats différents au même périphérique (pas de traçage possible) et aucune information stockée dans le SCMS ne peut associer des certificats à un véhicule ou à une personne physique. Le MA (Misbehavior Authority) assure le bon comportement des nœuds avec la production et la publication de la CRL et les Misbehavior reports (Rapport de mauvaise conduite). Le LOP (Location Obscurer Proxy) agit en tant que proxy anonymiseur, il mixte les rapports de mauvaise conduite envoyés par les OBUs au MAs.

Les composants de la PKI IEEE selon la contribution [Has18] sont les suivantes :

- **Security Certificate Management System(SCMS)** : Fournit des politiques et des normes techniques à l'ensemble de l'industrie véhiculaire connectés et aux audits.
- **Root CA** :C'est l'autorité racine, Elle émet les certificats des autorités subordonnées.
- **Intermediate Certificate Authority (ICA)** : C'est une autorité subordonnée, elle certifie l'ECA à partir de l'autorité racine.
- **Autorité de Liaison (Linkage Authority (LA) LA1, LA2)** : C'est l'entité qui génère les valeurs de liaison. La LA est couplée avec les (LA1 et LA2). Il communique avec le RA pour fournir des valeurs de liaison au AA.
- **Misbehavior Authority (MA)** : C'est l'entité qui produit / publie la CRL et le (Misbehavior report). Il se coordonne avec le RA, LA et AA pour générer la CRL.
- Le **Location Obscurer Proxy (LOP)** a pour rôle d'obscurcir l'emplacement des équipements Onboad (OBE) qui veulent communiquer avec les composants du SCMS et de réorganiser les Misbehavor report envoyé par l'OBE afin d'augmenter la confidentialité des utilisateurs.
- Authorization Certificate Authority (AA) : Le rôle de l'AA est de délivrer des certificats court-terme et de collaborer avec l'MA, l'RA et la LA afin d'identifier les valeurs de liaison à placer dans la CRL en cas de mauvaise conduite d'un utilisateur.
- **Registration Authority (RA)** : C'est l'autorité qui s'occupe de l'enregistrement des nouveaux utilisateurs. C'est elle qui reçoit les requêtes des certificats de l'OBE (On-Board Equipment) et les valeurs de liaison des LAs, et transfère les requêtes à la AA pour l'ajout de la clé. La RA crée et maintient une blacklist des certificats long-terme (EC).
- **Request Coordinator** : C'est une entité en relation avec les activités du RA. Son existence est nécessaire seulement s'il existe plusieurs RAs dans le SCMS.
- **Autorité d'inscription (Enrollement Certificate Authority - ECA)** : C'est l'autorité qui établit la connexion initiale entre OBE et SCMS. Elle vérifie que le type d'appareil soit conforme avec les informations contenues dans le laboratoire de certificat (Certificate Lab). Elle a le rôle d'émettre les certificats d'inscription / certificat à long-terme (EC) qu'elle envoie à l'OBE.
- **Device Configuration Manager (DCM)** : C'est l'entité qui s'occupe de la mise à jour du logiciel de l' OBE. C'est elle qui transmet les informations d'identification de l'OBE aux autres entités du SCMS. Elle Fournit à l'OBE les informations nécessaires pour qu'il puisse faire une demande de certificats à court terme à une RA.
- **Certfication services** : fournit des informations sur les types d'appareils certifiés pour recevoir des certificats numériques et spécifie le processus de certification.
- **CRL Store (CRLS)** : Sert de stockage et de distributeur pour la liste de révocation des certificats (CRL). Il s'agit d'une fonction de passage simple car les CRL sont signées par le générateur CRL (CRL Generator ).
- **CRL Broadcast (CRLB)** : C'est le module de diffusion de la CRL, il peut être effectué par le biais d'un équipement routier (Road Side Equipment - RSEs) ou d'un système de radio satellite, ou etc. Il s'agit d'une fonction d'intercommunication.

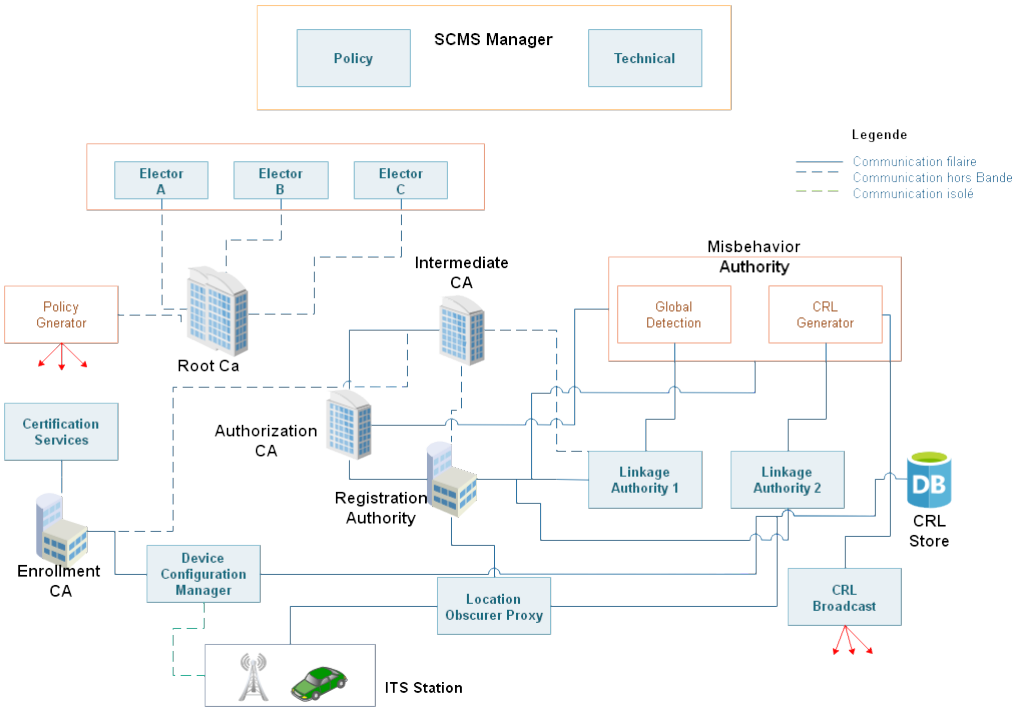



FIGURE 2.9 – Architecture PKI IEEE

#### 2.4.5.1 Les types de certificats

La PKI IEEE, tout comme la PKI ETSI utilisent deux types de certificats utilisateurs. Les Certificats pseudonymes qui sont à court terme et les certificats long-terme (EC).

#### 2.4.5.2 Structure du certificat

La structure des certificats IEEE est pareil à celle des certificats ETSI (AT et EC).

| Services de sécurité | Architectures |
|---|---|
| Authentification | Selon le NHTSA l'authentification est atteint à travers la signature digitale et le chiffrement. Selon l'ETSI via les messages signés. |
| Confidentialité | Le service est fournit selon le NHTSA et l'ETSI via le chiffrement symétrique. |
| Intégrité | Le NHTSA assure l'intégrité via le MCA. Le ETSI à travers les fonctions de Hachage. |
| Détection des responsabilités | Le service est fournit via la Misbehavior Authority selon le NHTSA. Selon le ETSI via l'accountability et la gestion à distance à travers la gestion à distance des comptes |
| Sécurité des messages | Le NHTSA et le ETSI utilisent la PKI |
| Non-repudiation | Le service est atteint grace au EDR utilisé pour tracer. |
| Vie privée | Le NHTSA utilise un proxy anonymiseur et un système de révocation protégeant la vie privée via le MA |

TABLE 2.5 – Les services de sécurité selon les architectures NHTSA et ETSI

### 2.4.6 Classes d'attaquants

Afin de mieux cerner les attaques possibles sur les communications véhiculaires il est nécessaire de définir les modèles d'attaquants possibles. La contribution [Has18] définit les critères de classification d'attaquant suivant :

— *Passif* : Un attaquant passif ne peut qu'écouter et analyser clandestinement le canal de transmission. Cette attaque peut être conduite par un voisinage curieux, mais aussi pour une entreprise qui cherche à créer des profils de conducteurs. Ce type d'attaques est plus facile à réaliser (il suffit de posséder le récepteur adéquat) et il est difficile à détecter puisque l'intrus n'apporte aucune modification sur les informations échangées.

— *actif* : Un attaquant actif peut générer, modifier, rejeter ou rejouer des messages afin de disséminer de fausses informations. Le but d'un attaquant actif est de s'octroyer des privilèges afin d'améliorer son environnement de conduite. Ainsi, il peut usurper l'identité d'un véhicule de secours pour faciliter son déplacement.

— *Interne* : Un attaquant interne est un membre authentifié du réseau qui peut communiquer avec les autres membres du réseau. Comme il fait partie du réseau, il possède déjà quelques avantages comme les clés publiques utilisées par les autres véhicules.

— *Externe* : Un attaquant interne peut causer plus de dommages au réseau que l'attaquant externe qui a un accès limité au système.

— *Malicieux* : Un attaquant malveillant n'a pas d'intérêts personnels à travers ses attaques et a pour but le dysfonctionnement du système. Par conséquent, il peut employer tous les moyens sans tenir compte des coûts correspondants et des conséquences. Les attaquants qui causent délibérément les accidents de la route sont considérés comme malicieux.

— *Rationnel* : Par opposition, un attaquant rationnel cherche un profit personnel, et ainsi, on peut prévoir les cibles d'attaques et les moyens employés. Les attaques rationnelles sont plus prévisibles que les attaques malicieuses.

— *Mal intentionné ou Involontaire* : Un attaquant est dit mal intentionné s'il vise délibérément à remettre en cause le bon fonctionnement du système. Ce type d'attaquant est à distinguer d'un attaquant involontaire qui peut par exemple lancer (sans le vouloir) une attaque à partir d'un capteur défectueux.

— *Indépendant ou collaboratif* : Les attaquants peuvent agir indépendamment les uns des autres ou bien collaborer. Lorsqu'ils collaborent, les attaquants s'échangent des messages et coopèrent afin de rendre l'attaque plus efficace. Par exemple, des véhicules attaquants collaboratifs annoncent un embouteillage fictif pour convaincre les véhicules honnêtes. Ces derniers vont alors changer de chemin, libérant ainsi la voie pour les attaquant.

— *Local ou Etendu* : Un attaquant peut avoir une portée d'action limitée, même s'il contrôle plusieurs entités. On dit qu'il est local parce qu'il a une portée limitée à un véhicule et des RSU, ce qui rend l'attaque limitée. Un attaquant étendu contrôle plusieurs entités ce qui lui confère une portée étendue.

### 2.4.7 Attaques de base

Le degré de sécurité des communications véhiculaires est l'un des thèmes critiques, parce que la transmission de l'information est propagée dans un environnement sans fil à accès ouvert. Selon [MBH14] il est possible classer les attaques en

quatre catégories principales : les attaques contre la disponibilité, les attaques contre l'authentification, les attaques contre l'authentification, les attaques contre la confidentialité, les attaques contre l'intégrité et les attaques contre la non-répudiation.

#### 2.4.7.1 Attaques visant la disponibilité

- **Déni de service (DOS) [HS+10] [RSS12a]** : L'entité malveillante empêche l'accès normal aux communications. Ce type d'attaque peut être généré en brouillant le canal radio, en surchargeant ou en épuisant les ressources du système par des requêtes abondantes, en exploitant la vulnérabilité des protocoles, ou en ayant une attitude non coopérative (refus de relayer des paquets par exemple).
- **Spamming [RSS12a]** : Ce type d'attaque à comme but de consommer la largeur de la bande du réseau et d'augmenter la latence de transmission. L'attaquant réalise cette attaque par envoi de messages de spam à tout le monde. Il est difficile de commander ce genre d'attaque en raison du manque d'infrastructure nécessaire et d'administration centralisée. Dans cette attaque, le malveillant dissimule les messages de spam à un groupe de stations. Ces messages sont justes comme des messages de publicité pour les différentes stations.
- **Attaque par trou noir [Bib+12]** : C'est une attaque selon laquelle un nœud refuse de participer aux communications ou quand une station établit une liaison avec cette station, cette dernière lâche la liaison pour former un trou noir. Toutes les communications vont se réorienter vers une station spécifique qui est en fait n'existe pas réellement et a pour but la perte de données.
- **Attaque grayhole [isaCA2010security]** : Cette attaque est similaire à l'attaque par trou noir, mais dans ce cas les paquets à détruire sont sélectifs.
- **Attaque au routage [SJP14]** : L'attaquant peut exploiter les vulnérabilités du systéme en déroutant ou détruisant les paquets. Ce type d'attaque inclut l'attaque par trou noir.
- **Attaque par Malware [RSS12a]** : Les attaques par Malware sont justes comme des virus qui entravent l'opération normale du système. Les communications véhiculaires sont normalement infectées par ces attaques quand il y a des actualisations ou des mises à jour de logiciel. De ce fait, les attaquants sont normalement initiés malveillants plutôt qu'étrangers.
- **Attaque Temporelle [RPH06]** : L'attaquant introduit des délais entre la réception et la retransmission des informations reçu. Ainsi les stations voisines ne reçoivent par les informations quand nécessaires ou même après un certain évènement.
- **Attaque Jamming [RSS12a] [RPH06]** : L'attaquant crée des interférences dans les fréquences utilisées par les stations du système pour communiquer.
- Greedy Behavior Attack[SLW19] [RHA04] : Cette attaque porte normalement sur la fonctionnalité du code d'authentification de message (MCA). C'est lorsque le véhicule malveillant utilise à mauvais escient le protocole MCA pour augmenter son usage de la bande passante, ce qui coûte cher aux autres utilisateurs. Cela peut entraîner la saturation du trafic, des collisions sur le canal de transmission et un retard dans les services légitimes de la station enregistrée et authentifiée.

#### 2.4.7.2 Attaques visant l'authentification

— **Usurpation d'identité ou de rôle** : Selon cette attaque, l'entité malveillante utilise une fausse identité ou de fausses lettres de créance pour se faire passer pour une entité légitime ou pour jouir des privilèges de cette dernière.

— **Attaque par Brute force** : Cette attaque vise à obtenir les informations (mot de passe, ID, etc) d'une station légitime ou à déchiffrer des données confidentielles en testant toutes les combinaisons possibles.

— **Attaque Sybill [XYG06]** : Un nœud malveillant peut revendiquer différentes identités dans le but de prendre de l'avantage sur les stations légitimes. Un attaquant transmet des messages multiples avec différents IDs (identificateurs) aux autres véhicules. De cette façon, d'autres véhicules estiment que ces messages viennent de différents véhicules. En d'autres termes, la tâche principale de l'attaquant est de fournir une illusion des véhicules multiples à d'autres véhicules et de le leurs imposer pour choisir l'itinéraire alternatif et pour laisser la tige pour les avantages de l'attaquant.

— **Attaque de personnification de station ou l'injection des messages erronés** : Chaque véhicule est sensé avoir un identifiant unique et avec l'aide de ces IDs chaque véhicule est identifié dans le réseau. Il devient l'élément le plus important quand un accident se produit pour révéler l'identité de l'utilisateur. Dans l'attaque de personnification de station, le but de l'attaquant est de changer son identité et d'agir comme un vrai créateur du message. Un attaquant reçoit le message original et change le contenu du message pour ses avantages, ex. rediriger le trafic routier de manière à permettre la libération de la route utilisée. Ensuite l'attaquant envoie ce message aux autres véhicules.

— GPS Spoofing[Fan+14] : L'usurpation GPS est le processus de génération d'une version falsifiée du signal GPS pour perturber le processus de navigation et de synchronisation temporelle du récepteur.

— **Attaque wormhole [Pat16]** : L'attaque Wormhole est un scénario dans lequel un attaquant enregistre les paquets en un point, et les rejoue à un autre.

— **L'attaque par tunnel (Tunneling attack)**[RSS12b] : L'attaquant connecte deux parties distantes du réseau Ad hoc en utilisant un canal de communication supplémentaire sous forme de tunnel. En conséquence, les deux stations distantes supposent qu'ils sont voisins et s'envoient des données en utilisant le tunnel. L'attaquant pourra effectuer une analyse du trafic ou une attaque de transfert sélectif.

— Attaque par réplication de clé/certificat[SLW19] : Cette attaque est du à l'utilisation par un attaquant d'une copie des clés et / ou de certificats d'un autre véhicule pour s'authentifier, ce qui peut compliqué à discerner par une autorités de la circulation qui veut identifient le véhicule. Plus précisément, le but de cette attaque est de semer la confusion chez les autorités de confiance (AT), notamment en cas de litige.

— Attaque par falsification de message (Message tampering) [SLW19] : Il s'agit d'une attaque très courante, dans laquelle l'attaquant modifie les messages échangés en communication V2V ou V2I.

— Attack par mascarade (Masquerading attack)[SLW19] : L'attaquant utilise de faux identifiants pour agir comme un autre véhicule. Cette attaque se produit lorsqu'un utilisateur ne prouve pas son identité et prétend être un utilisateur différent pour obtenir légalement un accès non autorisé.

— Attaque libre (Free-Riding Attack)[SLW19] **[lin2013CAhieving]** : Cette attaque est initiée par un utilisateur malveillant actif en faisant des faux efforts d'authentification tout en étant associée à une authentification de messagerie coopérative. Dans cette attaque, l'utilisateur malveillant peut tirer parti des contributions d'authentification d'autres utilisateurs sans avoir les siennes. Ce type de comportement s'appelle une attaque de free-riding. Cette attaque peut constituer une menace sérieuse pour l'authentification coopérative des messages.

— **Attaque par rejeu [RKS13]** : L'attaquant qui peut être un station malicieuse ou un utilisateur non autorisé, essai de prendre l'identité d'une station légitime ou (RSU) en rejouant les messages d'une connexion précédente dans une nouvelle connexion.

— **Session hijacking [RKS13]** : Cette attaque exploite le fait que l'authentification entre deux entités soit faite au début d'une session. En effet après une authentification réussi, l'attaquant prends le contrôle de la session.

— **Attaque par répudiation (perte des traces)** : L'attaquant arrive à nier toutes ou certaines actions effectuées.

#### 2.4.7.3 Attaques visant la confidentialité

— **Ecoute de canal** : Selon cette attaque, l'entité malveillante collecte les données transmises par les différentes stations afin d'extraire une information dont elle pourrait tirer profit. Cette attaque est facilitée par la nature du canal radio où les transmissions se font par diffusion.

— **Attaque Man in the Middle(MITM) [RKS13]** : Selon cette attaque, la station malicieuse se mets en écoute d'une communication entre deux véhicules, puis arrive à se faire passer pour une partie vers l'autre.

— **Attaque par accès non autorisé** : L'entité malicieuse essaie d'accéder à une communication sans avoir les droits ou privilèges nécessaire. Ceci peut causer des accidents, dommages ou permettre l'accès à des données sensibles.

— Attaque par analyse de trafic (Trafic analysis attack) [SLW19] : C'est une des attaques dangereuses qui menace la confidentialité. Dans cette attaque, après avoir écouté une transmission de message, l'attaquant analyse le message et essaie d'extraire et de rassembler le maximum d'informations utiles.

— L'attaque sociale (Social attack) [RH05] [SLW19] : L'attaque sociale est utilisée pour détourner l'attention du conducteur. L'attaquant envoie des messages immoraux et contraires à l'éthique aux conducteurs. Le but des attaquants est d'obtenir la réaction des conducteurs après avoir reçu ce type de messages immoraux. Par conséquent, cela affecte l'expérience de conduite et les performances du véhicule dans le système VANET.

#### 2.4.7.4 Attaques visant l'intégrité

— Attaques par **Suppression /fabrication/altération** de messages : Cette attaque a pour but d'effacer ou de créer ou de modifier les messages d'une communication sans que les stations impliquées ne s'en rendent compte.

— **Injection de messages d'erreur** : L'attaquant injecte intentionnellement de fausses informations dans les communications, affectant directement le comportement des différentes stations.

— **Attaque par Illusion [LT07] [isaCA2010security]** :L'attaquant va modifier les capteurs du véhicule de façon à imposer une fausse vision du monde au véhicule. Ainsi le véhicule sera contraint à partager de fausses informations.

— **Altération du Matériel** : Cette attaque peut être effectué pendant la maintenance annuelle, chez le constructeur par des employées malicieux. Ceux-ci peuvent obtenir ou insérer des données sensibles.

#### 2.4.7.5 Attaques visant la non-répudiation

— **Attaque de non répudiation (Non répudiation attack)** [AVD16] [SLW19] : Cette attaque se produit lorsqu'un attaquant refuse de s'engager dans l'activité d'envoi et de réception de messages en cas de litige.

## 2.5 La révocation

Le processus de vérification des certificats inclut une étape très importante, celle de la vérification de la révocation du certificat. Un certificat délivré par une autorité de certification est censé être utilisé pendant toute sa période de validité. Cependant, diverses circonstances peuvent interrompre cette validité avant son expiration. Ces circonstances comprennent le changement de nom, le changement d'association entre le sujet et le CA (par exemple, la fin d'un partenariat), la compromission ou une suspicion de compromission de la clé privée correspondante. Sous telles circonstances, l'autorité de certification doit révoquer le certificat. D'autres raisons peuvent s'ajouter à celle-ci, si on prend en considération le contexte véhiculaire, telles que : le vol du véhicule ou le changement de propriétaire. L'état de révocation n'est pas mentionné dans le certificat, il est distribué par la CA ou le CRL issuer à travers un autre canal aux applications clientes de la PKI. Cette information est nécessaire à la vérification et validation d'un certificat, raison pour laquelle elle doit être toujours disponible et accessible.

### 2.5.1 La Révocation dans la PKI X509

Les standard RFC5268 [08] et RFC6960 [San+13] décrivent plusieurs méthodes de révocation des certificats X509. Nous pouvons les classifier en deux groupes : les méthodes à notification en ligne et les méthodes à notification hors-ligne. Ceci signifie que la vérification de la révocation des certificats peut se faire en ligne ou en locale. Parmi les méthodes à notification en locale, nous avons une méthode qui implique la publication périodique de la part de la CA émettrice d'une structure de données signé appelé CRL. La méthode à notification en ligne est une alternative aux approches impliquant la CRL. Parmi les méthodes à notification en ligne, nous avons une méthode utilisant le protocole Online Certificate Status Protocol(OCSP). Le protocole OCSP est utilisé pour vérifier la révocation d'un certificat X509 sans passer par les listes de révocation des certificats.

#### 2.5.1.1 La révocation en chiffres

Le site [tri20] décrit une augmentation de 68% du nombre des certificats émis entre 2017 et 2018. Le site montre aussi qu'il existe plus de 200 autorités de certification, dont neuf émettrices de plus de 95% des certificats en cours d'utilisation. Nous observons à partir de ces données que la croissance asymétrique des listes de révocation suit la répartition non uniforme des parts de marché détenu par chaque

CA. Nous pouvons conclure après une analyse de différentes CRLs [Cen20], que le débit de révocation moyen est de 3000 certificats par jour. Nous résumons ces informations dans le tableau 2.6.

| Champs | Valeur |
|---|---|
| Quantité de certificats actifs | 145,980,146 |
| Autorités de certification | +200 |
| Autorités de certification Majeure | 9 |
| Certificats révoqués par jour | 3000 |
| Taux de révocation | 0.007% |

TABLE 2.6 – Resultats Statistiche révocation dans Internet

#### 2.5.1.2 Les méthodes de révocation

Il existe plusieurs méthodes de révocation, nous allons lister maintenant les méthodes les plus utilisées.

— **La CRL [08]** est une méthode qui implique la publication périodique de la part de la CA émettrice d'une structure de données signé appelé CRL. Une CRL est une liste horodatée référençant les certificats révoqués par une autorité de certification. La CRL est signée par la CA ou par l'émetteur de la CRL (CRL issuer) et mise à disposition dans un référentiel public. Chaque certificat révoqué est référencé dans la CRL par son numéro de série et sa date de révocation. Lorsqu'un système utilise un certificat pour, par exemple, vérifier la signature numérique d'un utilisateur distant, ce système vérifie non seulement la signature et la validité, mais aussi la révocation du certificat. Ceci en vérifiant que le numéro de série du certificat ne figure pas dans la CRL la plus récente disponible. La publication de la CRL par la CA ou le CRL issuer est périodique. La période dépend de la politique interne de la CA. Ainsi nous pouvons avoir des publications chaque heure, chaque jour ou chaque semaine. Il y a en général un décalage entre le moment de révocation d'un certificat et la publication de son numéro de série dans la CRL. Toute nouvelle entrée ajoutée à la CRL est faite dans le cadre de la mise à jour suivant la notification de révocation. Les CRLs peuvent être utilisées dans un champ varié d'applications et d'environnements couvrants différents objectifs d'interopérabilité et un espace encore plus large d'exigences opérationnelles et d'assurance. Ce profil établit une base commune pour les applications génériques nécessitant d'une interopérabilité et définit un ensemble d'informations attendues dans chaque CRL. En outre, le profil définit des emplacements communs pour les attributs fréquemment utilisés ainsi que des représentations communes pour ces attributs. La portée d'une CRL est l'ensemble des certificats qui peuvent apparaître sur une CRL donnée. Chaque CRL a une portée particulière. Par exemple, la portée peut être « tous les certificats émis par CA X», «tous les certificats CA émis par CA X», «tous les certificats émis par CA X qui ont été révoqués pour des raisons de compromission de clé et de CA», ou un ensemble de certificats basés sur des informations locales arbitraires, telles que «tous les certificats délivrés aux employés du NIST situés à Boulder». Si la portée de la CRL comprend un ou plusieurs certificats émis par une entité autre que l'émetteur de la CRL, il s'agit alors d'une CRL indirecte. La portée d'une CRL indirecte peut être limitée aux certificats émis par une seule autorité de certification ou peut inclure

des certificats émis par plusieurs autorités de certification. Si l'émetteur de la CRL indirecte est une autorité de certification, alors la portée de la CRL indirecte peut également inclure les certificats émis par l'émetteur de la CRL. L'émetteur de la CRL peut également générer des delta CRL. La version 2 de la CRL ajoute des extensions qui sont les suivantes :

— *Authority Key Identifier :* cette extension fournit un moyen d'identifier la clé publique correspondant à la clé privée utilisée pour signer la CRL. L'Authority Key Identifier peut être basée sur l'identifiant de clé de l'autorité(Le key identifier mentionnée dans le certificat du signataire de la CRL) ou sur le nom de l'émetteur et le numéro de série de son certificat. Cette extension est particulièrement utile lorsqu'un émetteur a plus d'une clé de signature, soit en raison de plusieurs paires de clés simultanées, soit en raison d'un changement.

— *Issuer Alternative Name :* cette extension permet d'associer des identités supplémentaires à l'émetteur de la CRL. Les options définies comprennent une adresse de messagerie électronique (rfc822Name), un nom DNS, une adresse IP et un URI. Plusieurs instances d'une forme de nom et plusieurs formes de nom peuvent être incluses.

— *CRL Number :* Cette extension n'est pas critique et permet aux utilisateurs de déterminer facilement quand une CRL particulière remplace une autre CRL. Cette extension décrit un numéro de séquence croissant de façon monotone pour une portée de CRL et un émetteur de CRL donnés. Le CRL Number prend également en charge l'identification des CRL complètes et delta CRL complémentaires. Les émetteurs de CRL conformes à ce profil doivent inclure cette extension dans toutes les CRL et doivent marquer cette extension comme non critique.

— *Delta CRL Indicator :* C'est une extension critique qui sert à identifier une delta CRL. Les Delta CRLs contiennent les mises à jour des listes de révocation précédemment distribuées, plutôt que toutes les informations qui apparaîtront dans une CRL complète.

— *Issuing Distribution Point :* C'est une extension critique qui identifie le point de distribution et la portée d'une CRL particulière. Il indique si la CRL couvre uniquement la révocation des certificats utilisateurs, des certificats CA, des certificats d'attribut ou un ensemble limité de raisons. Bien que l'extension soit critique, les implémentations conformes ne sont pas obligées de prendre en charge cette extension. Cependant, les mises en œuvre qui ne prennent pas en charge cette extension doivent soit traiter l'état de tout certificat non répertorié dans cette liste CRL comme inconnu, soit localiser une autre CRL qui ne contient aucune extension critique non reconnue.

— *Freshest CRL :* cette extension identifie la manière dont les informations de la delta CRL sont obtenues comme celle decrite dans le certificat x509.

— *Authority Information Access :* Cette extension n'est pas critique, elle définit l'accès aux informations de l'autorité de certification.

La structure de la CRL en format ASN.1 est la suivante :

```
CertificateList  ::=  SEQUENCE  {
        tbsCertList          TBSCertList,
        signatureAlgorithm   AlgorithmIdentifier,
```

| Champs |
|---|
| Version |
| Last Update |
| Next Update |
| Extensions |
| List of revoked Certificates |
| Signature |

TABLE 2.7 – Structure de la CRL Version 2

```
     signatureValue       BIT STRING  }
TBSCertList  ::=  SEQUENCE  {
     version                 Version OPTIONAL,
                                  -- if present, MUST be v2
     signature               AlgorithmIdentifier,
     issuer                  Name,
     thisUpdate              Time,
     nextUpdate              Time OPTIONAL,
     revokedCertificates     SEQUENCE OF SEQUENCE  {
          userCertificate         CertificateSerialNumber,
          revocationDate          Time,
          crlEntryExtensions      Extensions OPTIONAL
                       -- if present, version MUST be v2
                            }  OPTIONAL,
     crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
                        -- if present, version MUST be v2
                            }

-------------- CRL version 2 ASN.1 format -----------------------
```

— Le protocole Overissued CRL (O-CRL) [Coo99] est une optimisation de la CRL. Le protocole adresse un problème système assez important, celui du pic de requêtes. Effectivement, la CRL expire pour tous au même instant, ce qui peut induire une concentration de toutes les requêtes d'actualisation en certain moment. Le protocole propose d'avoir la même CRL avec de différentes dates d'expiration.

— L'approche Indirect CRL (I–CRL) se propose de regrouper plusieurs CRLs émises par différentes autorités de certification en une seule et unique CRL. Cette méthode l'objectif de faciliter la distribution et l'accès à la CRL.

— **la delta-CRL(d-CRL)** [08] [Coo00] [Cop00] est une optimisation de la méthode CRL. L'objectif de la delta-CRL est de réduire en volume la CRL à télécharger par chaque utilisateur. La delta-CRL est la liste des mises à jour de la CRL depuis la dernière publication de la CRL. Elle répertorie uniquement les certificats, dans son étendue, dont le statut de révocation a changé depuis l'émission d'une CRL complète référencée. La CRL complète référencée est appelée CRL de base. La portée d'une CRL delta doit être la même que la CRL de base qu'elle référence. L'utilisation de la delta CRL peut réduire considérablement la charge du réseau et le temps de traitement dans certains environnements. Les delta CRLs sont généralement plus petites que les CRL qu'elles mettent à jour, de sorte que les applications qui obtiennent des delta CRLs consomment

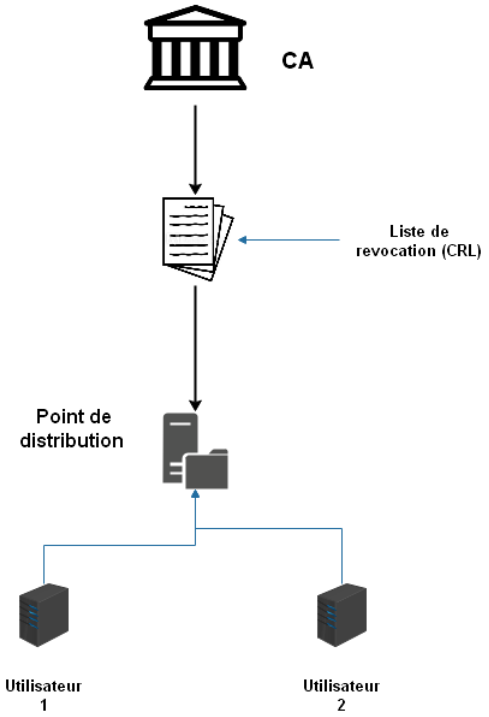



FIGURE 2.10 – La liste de revocation (CRL)

moins de bande passante réseau que les applications qui obtiennent les CRL complètes correspondantes. Les applications qui stockent les informations de révocation dans un format autre que la structure CRL peuvent ajouter de nouvelles informations de révocation à la base de données locale sans retraitement des informations. La présence de l'extension FreshestCRL dans le certificat X509 indique l'utilisation de la méthode de révocation impliquant la delta-CRL. L'extension FreshestCRL contient la référence d'un point de distribution de delta-CRL. La solution delta-CRL n'impact pas le nombre de données à stocker, ni le débit des requêtes, ni la périodicité, ni le délai de vérification de la révocation. Elle n'a d'impact que sur l'usage de la bande passante. La delta-CRL reste encore étroitement lié à la CRL.

— Le protocole **OCSP** (Online Certificate Status protocol) [San+13] est une alternatives aux CRLs, il fait partie des méthodes à notification en ligne. Le protocole OCSP offre une réponse temps réel différemment de l'approche périodique de la CRL. Le protocole OCSP permet aux applications de déterminer l'état (de révocation) des certificats X509. OCSP peut être utilisé pour répondre à certaines des exigences opérationnelles de fournir des informations de révocation plus récentes que ce qui est possible avec les CRLs et peut également être utilisé pour obtenir des informations d'état supplémentaires. Un client OCSP émet une demande d'état de révocation à un serveur OCSP et suspend l'acceptation des certificats en question jusqu'à ce que le répondeur lui fournisse une réponse. Le protocole est décrit dans la figure 2.11. La structure du message de requête est décrit dans le tableau 2.8.
La structure ASN.1 de la requête est la suivante :

```
The ASN.1 structure corresponding to the OCSPRequest is:
```

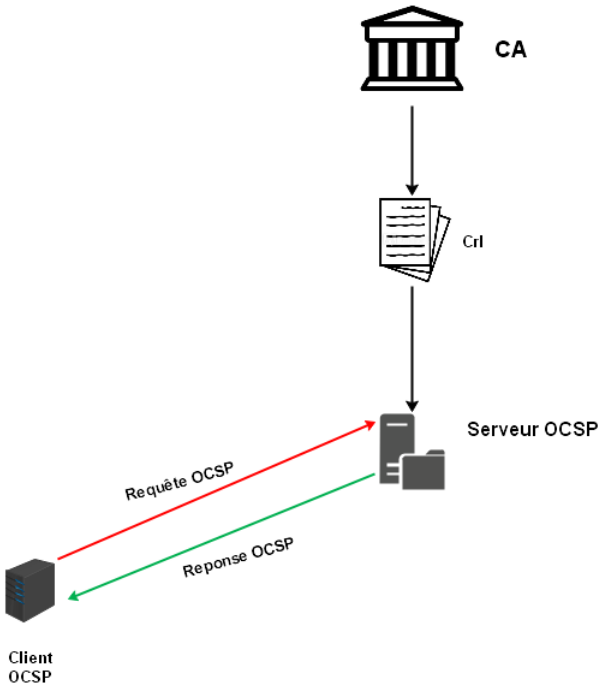



FIGURE 2.11 – Online certificate Status Protocol (OCSP)

| Champs |
|---|
| Version |
| Service (Service request) |
| Certificat cible (target certificate identifier) |
| Extensions |

TABLE 2.8 – Requête OCSP

```
   OCSPRequest     ::=     SEQUENCE {
       tbsRequest                  TBSRequest,
       optionalSignature   [0]     EXPLICIT Signature OPTIONAL }

   TBSRequest      ::=     SEQUENCE {
       version             [0]     EXPLICIT Version DEFAULT v1,
       requestorName       [1]     EXPLICIT GeneralName OPTIONAL,
       requestList                 SEQUENCE OF Request,
       requestExtensions   [2]     EXPLICIT Extensions OPTIONAL }

Signature       ::=     SEQUENCE {
       signatureAlgorithm      AlgorithmIdentifier,
       signature               BIT STRING,
       certs               [0] EXPLICIT SEQUENCE OF Certificate
   OPTIONAL}

   Version         ::=             INTEGER  {  v1(0) }
```

```
   Request          ::=     SEQUENCE {
       reqCert                      CertID,
       singleRequestExtensions      [0] EXPLICIT Extensions OPTIONAL }

   CertID           ::=     SEQUENCE {
       hashAlgorithm        AlgorithmIdentifier,
       issuerNameHash       OCTET STRING, -- Hash of issuer's DN
       issuerKeyHash        OCTET STRING, -- Hash of issuer's public key
       serialNumber         CertificateSerialNumber }

------- OCSP Protocol Request ASN.1 format ---------
```

La principale raison d'utiliser le hash de la clé publique de la CA en plus du hash du nom de la CA pour identifier l'émetteur vient du fait que deux CA puissent utiliser le même nom (l'unicité dans le nom est une recommandation qui ne peut pas être appliquée). Toutefois, deux autorités de certification n'auront jamais la même clé publique à moins que les autorités de certification n'aient explicitement décidé de partager leurs clés privées ou que la clé de l'une des autorités de certification n'ait été compromise. Le client peut choisir de signer la requête OCSP. Dans ce cas, la signature est calculée sur la structure tbsRequest. Si la requête est signée, le demandeur doit spécifier son nom dans le champ requestorName. De plus, pour les demandes signées, le client peut inclure des certificats qui pourraient aider le répondeur OCSP à vérifier la signature du client dans le champ certs de Signature.

Il n'y a pas de limite au nombre de requêtes qu'un utilisateur peut envoyer, et chaque requête ne doit pas être impérativement signé. La requête peut contenir une liste des numéros de série des certificats à vérifier. Après réception de la requête, le serveur se réserve le droit de répondre si la requête est bien formulée et s'il possède la réponse. Le serveur répond avec un message de réponse OCSP (OCSP response). Le message de réponse est signé par la CA émettrice du certificat ou le système auquel la CA à délégué la tâche. La structure du message est décrit dans le tableau 2.9.

| Champs |
|---|
| Version |
| Reponses |
| Extensions |
| Signature |

TABLE 2.9 – Reponse OCSP

La structure ASN.1 du message de réponse est la suivante :

```
OCSPResponse ::= SEQUENCE {
      responseStatus         OCSPResponseStatus,
      responseBytes          [0] EXPLICIT ResponseBytes OPTIONAL }

   OCSPResponseStatus ::= ENUMERATED {
       successful             (0),  -- Response has valid confirmations
       malformedRequest       (1),  -- Illegal confirmation request
       internalError          (2),  -- Internal error in issuer
```

```
        tryLater              (3),  -- Try again later
                                    -- (4) is not used
        sigRequired           (5),  -- Must sign the request
        unauthorized          (6)   -- Request unauthorized
    }
------- OCSP Protocol Response  ASN.1 format ---------
```

Le champ réponse peut contenir plusieurs réponses si la requête contenait plusieurs certificats à contrôler. Pour chaque certificat spécifié dans la requête il aura dans la réponse : Le numéro de série du certificat, l'état de révocation, durée de vie de la réponse et des informations optionnelles. Le champs état de révocation peut contenir trois valeurs : good, Revoked, unknown. Le protocole OCSP a l'inconvénient de ne marcher qu'en ligne. Les réponses OCSP contiennent 4 differentes dates :

- *thisUpdate* : l'heure la plus récente à laquelle l'état indiqué est connue du répondeur comme étant correcte.
- *nextUpdate* : l'heure à laquelle ou avant laquelle des informations plus récentes seront disponibles sur l'état du certificat.
- *producedAt* : l'heure à laquelle le répondeur OCSP a signé cette réponse.
- *revocationTime* : l'heure à laquelle le certificat a été révoqué ou mis en attente.

La vérification de la révocation en ligne peut réduire considérablement la latence entre la révocation d'un certificat et l'actualisation des listes de révocation. Une fois que la CA révoque un certificat, toute requête adressée au service en ligne pourra recevoir ces nouvelles informations en temps réel. Cependant, ces méthodes imposent de nouvelles exigences de sécurité telles que : le vérificateur du certificat doit faire confiance au service de validation en ligne alors que ce n'est pas nécessaire pour le référentiel public. L'avantage pour l'autorité de certification émettrice d'utiliser cette méthode de révocation n'est pas clair. Effectivement, fournir de la bande passante pour le répondeur OCSP peut être coûteux, en particulier pour les autorités de certification disposant de nombreux sites d'abonnés à fort trafic, et ce coût est une préoccupation pour de nombreuses autorités de certification. Le protocole est aussi sujet aux problèmes de vie privée parce que du côté serveur OCSP il possible tracer toutes les activités d'un utilisateur suivant toutes les requêtes faites.

— Le protocole **OCSP Stapling** est une approche proactive pour optimiser le protocole OCSP. Les répondeurs OCSP peuvent pré produire des réponses signées spécifiant le statut des certificats à un moment spécifique. L'heure à laquelle l'état était réputé correct doit être indiqué dans le champ thisUpdate de la réponse. L'heure à laquelle ou avant laquelle des informations plus récentes seront disponibles est indiqué dans le champ nextUpdate, tandis que l'heure à laquelle la réponse a été produite apparaîtra dans le champ generatedAt de la réponse. La méthode OCSP stampling exploite cette caratéristique. Il serait intéressant de souligner que le protocole déplace la charge de la vérification de la révocation du côté client vers le côté serveur. Le serveur ou celui qui veut prouver son identité (par exemple le serveur web) va interroger directement le répondeur OCSP sur la révocation de son propre certificat. La réponse du serveur est la preuve de non-révocation qui sera stockée et utilisée ensemble avec son certificat. Cette réponse mise en cache en toute sécurité est ensuite

envoyée au client avec le certificat pendant le TLS Handshake à travers l'extension **Certificate Status Request**. Cette extension à été integré par le RFC 6961 [Pet13] pour TLS. Ainsi le client pourra obtenir des informations de révocation sans avoir à ouvrir une nouvelle connexion ou contacter un serveur externe. Ceci signifie que le contrôle de la révocation n'a plus de répercussions sur la latence de la phase d'établissement de connexion. L'OCSP stapling résout le problème de protection de la vie privée d'OCSP car l'autorité de certification ne reçoit plus les demandes de révocation directement du client (navigateur). Le fait que les requêtes ne soient pas signées (quiconque peut envoyer des requêtes) facilite les attaques de déni de service. Le système entier reste couteux à mettre en place côté CA.

— **la CRL distribution point (CRLDP)** [08] est une methode de révocation basé sur la CRL. Cette approche vise à optimiser la scalabilité de la CRL en divisant les listes de révocation en plusieurs fragments. Ainsi chaque fragment est accessible par un point de distribution particulier. Les differents fragments sont mis dans un ordre logique pour permettre à chaque utilisateur de télécharger seulement le fragment nécessaire et non toute la CRL. Chaque certificat est lié à un point de distribution et contient les informations relatives à celui-ci. Les certificats des utilisateurs vont contenir dans le champ CRLDistributionPoints le lien vers le point de distribution auquel ils sont attachés, si cette methode est adoptée. Chaque point de distribution delivre un fragment de CRL, qui est une plus petite CRL avec toutes les carateristiques de cette derniére.

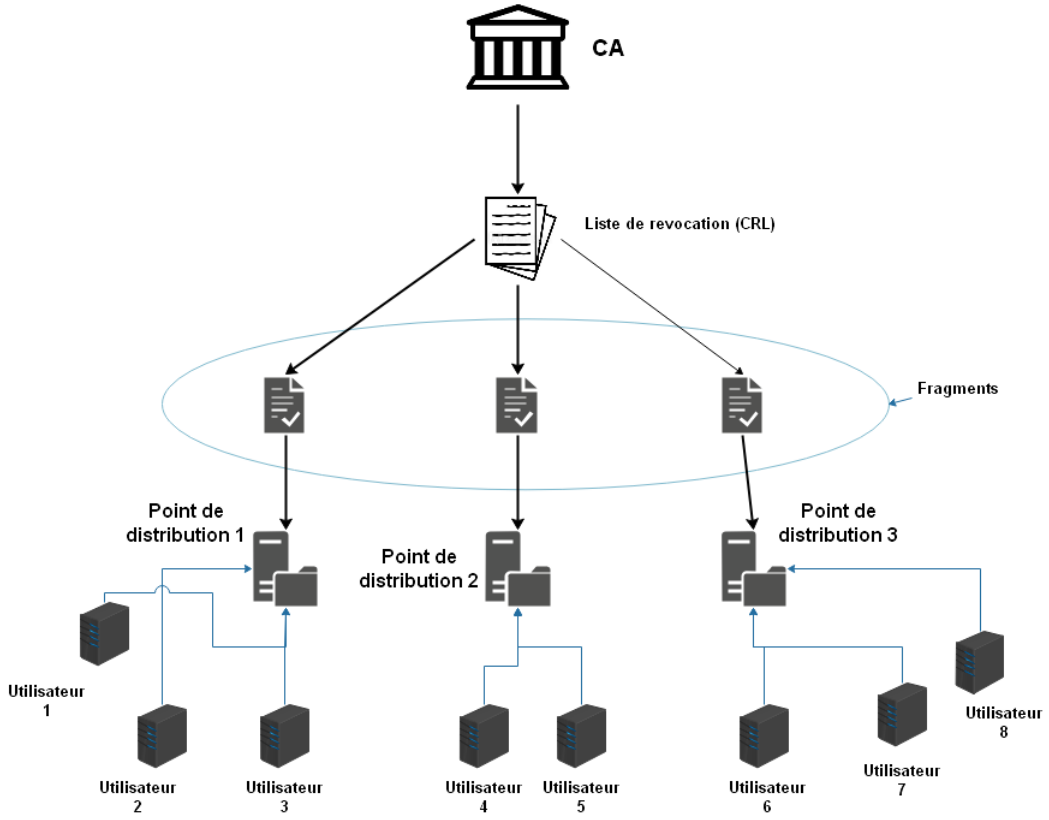



FIGURE 2.12 – Certificate Distribution Point

— Le Certificate Revocation Tree (CRT) [Muñ+04] est une methode de révocation qui implique la publication périodique de la part de la CA émettrice d'une structure de données signé appelé CRT. La structure CRT est basée sur l'arbre de merkle (MH [Mer90], elle s'obtient en réorganisant les listes de révocation en un arbre de merkle. Cette approche permet à l'utilisateur de reconstituer les informations de révocation de manière fiable partant d'une petite quantité de données. Le hash de chaque certificat révoqué représente une feuille de l'arbre de merkle, ainsi partant des feuilles les contenus sont hashé deux à deux progressivement jusqu'à atteindre la racine qui est le root hash. Le root hash est signé par la CA ou l'autorité émettrice de la CRL ce qui garantit l'intégrité et l'authenticité de l'arbre. Les feuilles de l'arbre ont un ordre très précis, ce qui permet de retrouver facilement la position d'un certificat si révoqué dans l'arbre. Un utilisateur peut vérifier la révocation d'un certificat ayant seulement quelques feuilles, quelques branches et le root Hash. Ceci est décrit dans la figure 2.13, on peut contrôler le statut du certificat $L_2$ ayant juste les hashs $N_{0,3}$, $N_{1,0}$, $N_{2,1}$ et le Root Hash. Cette approche à l'inconvénient d'être couteux en terme de calcul.

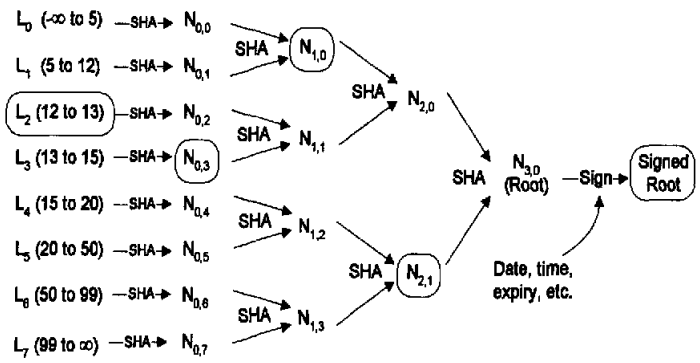



FIGURE 2.13 – Certificate Revocation Tree

— L'approche **Simple Certificate Validation Protocol (SCVP)** [07], n'est pas une methode de révocation comme la CRL et OCSP. C'est un procotole d'accés aux aux listes de révocation. selon cette approche, le client peut déléguer la vérification de la révocation à un serveur interne ou externe à son domaine, de confiance ou pas. Ce serveur pourrait redistribuer les informations acquises à d'autres utilisateurs comme un serveur OCSP ou un dépôt CRL. La vérification de la révocation des certificats est délégué par le client vers un serveur spécialisé.

#### 2.5.1.3 Comparaison des approches de révocation

Nous allons maintenant faire dans le tableau 3.5 une comparaison des différentes approches décrites ci-dessus selon quatre critères de comparaison que nous listons et décrivons :

— *Connectivité* : La Connectivité est un critère de comparaison très important. Elle permet de mettre en évidence le fait que la vérification de l a révocation des certificats se fasse en ligne en ligne, ou en local, ou les deux.

— *Périodicité* : Le critère de périodicité permet de mettre en avant la fréquence d'actualisation des listes de révocation. C'est-à-dire que l'utilisateur à accés à des informations periodiques ou temps réel.

— *Scalabilité* : Nous définissons une methode de révocation scalable si le volume de données necessaires à la verification de la révocation des certificats ne depends de la quantité des utilisateurs.

— *Charge de travail* : Ce critère permet de situer entre le client et le serveur, celui qui aura la charge de vérification du certificat.

— *Structure* : Ce critère permet de distinguer les différentes structures de données utilisées pour distribuer les listes de révocation.

| | Connectivité | Scalabilité | Periodicité | C. de travail | Structure |
|---|---|---|---|---|---|
| CRL | Offline | ✘ | Periodique | Client | CRL |
| delta-CRL | Offline | ✘ | Periodique | Coté Client | CRL |
| CRL.dist.point | Offline | ✓ | Periodique | Coté Client | CRL |
| CRT | Offline | ✘ | Periodique | Coté Client | CRT |
| OCSP | Online | ✓ | Temps Réel | Coté Serveur | OCSP |
| OCSP Stapling | Offline | ✓ | Temps Réel | Coté serveur | OCSP |
| SCVD | - | - | - | - | - |

TABLE 2.10 – Tableau comparatif des Systémes de distribution de la révocation.

### 2.5.2 La Révocation dans la VPKI

Le standard ETSI TS 102 941 propose deux methodes de révocation, qui sont des methodes préexistantes et déjà utilisées pour la vérification et la validation des certificats X509 : La CRL et la delta-CRL. Ces methodes de révocation sont utilisés dans le véhiculaire pour produire et distribuer les listes de révocation (CTL) des certificats des CAs. Aucune methode de révocation pour les certificats pseudonymes n'es défini par le standard. Nous allons maintenant présenter la CRL et la delta-CRL dans leurs application véhiculaire :

— Le principe de la **La CRL** décrit pour la PKIX est le même que pour la VPKI. La structure de données par-contre évolue pour s'adapter aux contraintes véhiculaires. La structure ASN1 est la suivante :

```
%%%%%%%%%%%%%%%%%%%%
%----CRL
%%%%%%%%%%%%%%%%%%%%

ToBeSignedCrl ::= SEQUENCE {
version Version,
thisUpdate Time32,
nextUpdate Time32,
entries SEQUENCE OF CrlEntry,
...
}
```

```
CrlEntry ::= HashedId8
```

— **la delta-CRL [ETS]** est utilisée aussi dans le véhiculaire comme une optimisation de la CRL en terme d'usage de la bande passante. Nous en avons déjà décrit les avantages et les limitations dans les sections suivantes.

La comparaison de ces deux approches a été déjà faite dans le sous-chapitre 2.4.1.

Révoquer une station ITS signifie, révoquer son certificat à long-terme et tous ses certificats pseudonymes. Nous savons que chaque station a plus d'un certificat pseudonyme, ainsi révoquer une seule station signifie ajouter plusieurs certificats à pseudonyme dans les listes de révocation. On prévoit à court terme, une croissance exponentielle des listes de révocation et donc des délais non négligeables de téléchargement et de traitement des listes de révocation. Les délais de vérification des certificats pourraient être impactés. Le standard pris en considération ne propose aucune méthode de révocation pour les certificats pseudonymes, raison pour laquelle nous nous sommes focalisés sur ce problème pour proposer des solutions.

## 2.6 La technologie BLOCKCHAIN

*La Blockchain est une technologie de stockage et de transmission d'informations, transparente, sécurisée, et fonctionnant sans organe central de contrôle. Par extension, une Blockchain constitue une base de données qui contient l'historique de tous les échanges effectués entre ses utilisateurs depuis sa création. Cette base de données est sécurisée et distribuée : elle est partagée par ses différents utilisateurs, sans intermédiaire, ce qui permet à chacun de vérifier la validité de la chaîne.* **Blockchain France** [Fra17]

*La Blockchain, ou chaîne de blocs, est une technologie de stockage et de transmission d'informations. Par extension, ce mot désigne une base de données numérique décentralisée. Souvent assimilée à un registre, cette base regroupe un historique de transactions électroniques.* **Banque de France**[Fan16].

*La technologie Blockchain est un réseau qui stocke des blocs d'information qui sont identiques à travers le réseau. L'information stockée sur une Blockchain est partagée, vérifiable, publique et accessible.* **Commission Européene[commissionEuropeene]**.

Plus de 90% des cryptos-monnaies actuelles se basent sur une technologie appelée Blockchain. Inventé en 2008 par Satoshi Nakamoto comme support pour le Bitcoin (première crypto-monnaie) [Nak08]. La grande innovation de la Blockchain est de permettre des transactions entre de parfaits inconnus sur un medium non sécurisé sans avoir besoin d'un tiers de confiance. Deux inconnus qui ne se font absolument pas confiance seront capables de s'accorder sur une réalité commune grâce à la Blockchain. Plus en détail, la Blockchain fait partie de la catégorie des registres distribués (DLT), elle contient toutes les transactions effectuées par les membres du réseau regroupées en structure de données appelées Bloc. Ce registre est distribué car chaque membre du réseau en possède une copie, le registre ne peut donc pas être modifié sans que personne ne s'en aperçoive. Rien ne se cache dans un réseau Blockchain, toutes les données et toutes les actions effectuées dans le réseau sont accessibles à tous. Toute écriture dans ce registre se fait strictement sous l'accord de tous les membres du réseau. Une des règles fondamentales de la Blockchain c'est

qu'on ne peut modifier les données déjà présentes dans la Blockchain, on peut qu'en ajouter de nouvelles. Les informations dans le registre sont reparties en groupe appelés blocs, les uns liés aux autres à travers des algorithmes cryptographiques ainsi à former une chaine (Blockchain).

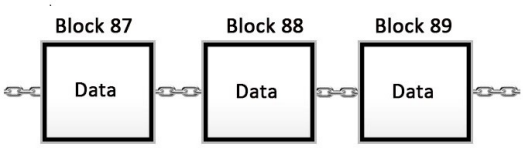



FIGURE 2.14 – La Blockchain

Selon [Nak08] la Blockchain est une chaine de blocs. Le bloc est une structure de données contenant des traces de tous ce qui a été fait par les membres du réseau. Ce que nous définissons communément transactions. Les transactions sont en effet les échanges entre les membres du réseau. Chaque bloc contient un hash calculé sur le bloc précédent, ce qui rend difficile la coupure de la chaine par un attaquant. En effet couper la Blockchain sous-entend casser un algorithme de hachage. La Blockchain peut être décomposée en trois parties : le réseau, le consensus et l'application.

- **Réseau** : La robustesse de la Blockchain dépend de la dimension de son réseau d'utilisateurs. La Blockchain est un système distribué, tous les nœuds du réseau détiennent un répliquât de la Blockchain. Chaque nœud représente une redondance du système, ce qui signifie qu'autant de nœuds il y a plus robuste sera le système. Selon [Nak08] la gouvernance de ce réseau est complètement démocratique. Néanmoins nous pouvons distribuer les nœuds en deux grands groupes :
  - *Validateur* : Les nœuds qui veillent au respect des règles du protocole et au bon fonctionnement du système. Dans certains cas et sous certains types de Blockchain ces validateurs sont incités à valider à travers des récompenses. Dans ce cas on parle de mineur.
  - *Utilisateur* : Les nœuds simples, sont ceux qui utilisent les différents service fournis par le système.
- **Le consensus** : La Blockchain se base sur un réseau distribué, qui demande une bonne synchronisation des nœuds. Chaque membre du réseau doit détenir une copie cohérente du registre. Nous parlons de consensus comme condition de synchronisation des différents nœuds.
- **L'application** : La Blockchain est un système complètement personnalisable en fonction de l'application qui s'y pose. On pourrait voir la Blockchain comme le hardware et l'application comme le software. En effet les Blockchains se différencient aussi en fonction de l'application.

Le système Blockchain offre cinq services, l'intégrité, la non-répudiation, horodatage, la disponibilité et l'immuabilité.

### 2.6.1 Le consensus de validation des blocs

Nous avons précédemment dit qu'on ne peut modifier les données qui sont déjà dans la Blockchain mais on peut y ajouter des données. Ceci évoque un problème

de synchronisation de toutes les copies détenues par des nœuds qui se méfient les uns des autres. Qui a le droit et à quel moment d'ajouter un bloc à la chaine. Ceci est un problème bien connu notamment dans les environnements distribués. Le problème des généraux byzantins [Lam83] [LSP19], resté irrésolu jusqu'à l'arrivée de la Blockchain. Je souligne que la Blockchain ne résout pas totalement le problème mais propose une solution applicable sous certaines conditions. La Blockchain propose plusieurs algorithmes de consensus efficaces en fonction du réseau d'utilisateurs.

#### 2.6.1.1 La preuve de travail (Proof of Work (PoW)

Cet algorithme de consensus permet d'octroyer en un temps spécifique le droit d'ajouter un bloc de données à la Blockchain. Il part sous l'hypothèse que les nœuds malveillants du système détiennent moins de 51% de la puissance de calcul total du réseau. Cette contrainte doit être vérifiée pour que le système soit fiable. Dans ce mécanisme, un mineur doit effectuer une quantité de travail, qui est le plus souvent un puzzle ou un défi mathématique, difficile à résoudre mais facile à vérifier. Ce défi est partagé à tous, le premier qui en vient à bout obtient le droit d'ajouter un bloc et tous les autres nœuds sont obligés de l'accepter après vérification de la solution du défi. La difficulté du défi peut être adaptée en fonction du débit de production de bloc voulu. C'est-à-dire qu'il est possible jouer sur la difficulté pour réguler le débit de production des blocs. Ceci est aussi une caractéristique fondamentale qui permet au PoW de passer à l'échelle. Pour résoudre le défi mathématique dans les temps, c'est-à-dire pour être concurrent il faut une capacité considérable de puissance de calcul. Ceci requiert de gros investissements financiers, raison pour laquelle le proof-of-work est à risque de centralisation, car le pouvoir de décision est concentré dans les mains des plus forts. Tout le monde veut miner et obtenir d'alléchantes récompenses, mais tout le monde n'a pas les moyens et la capacité de le faire. D'autre part, la PoW n'est pas du tout écologique, son usage massif à comme conséquence, la surproduction de matériel de calcul dédié et la consommation démesurée d'énergie. Le proof-of-work a une synchronisation lente ce qui s'adapte mieux aux grands réseaux d'utilisateurs.

#### 2.6.1.2 La preuve d'enjeux (Proof of Stake (PoS))

La PoS a été proposée afin de résoudre les faiblesses du Proof-of-Work. Cette approche se base toujours sur l'hypothèse des 51% mais par rapport aux enjeux. C'est-à-dire que le système reste fiable tant que les nœuds fiables détiennent plus de 51% du capital total du réseau. Différemment du Proof-of-Work, pour avoir le droit d'ajouter un bloc à Blockchain il faut avoir un considérable capital enjeu. D'autre part plus de capital le mineur détient dans le réseau plus de probabilité il a d'ajouter un block à la Blockchain. Le PoS est écologique car le minage ne met pas la nature en danger, il est le consensus le plus acclamé. L'approche se base sur le fait qu'il soit difficile voire impossible qu'une seule personne puisse détenir ou acquérir plus de 50 % du capital total. Chaque mineur est incité à miner les blocs par une récompense qui dépend de l'application. Le système reste injuste envers les plus faibles économiquement parlant, ce qui pourrait décourager beaucoup de nœuds à miner. Ce fait ouvre le risque de concentration du pouvoir de validation dans les mains d'une élite

| Critéres | PoW | PoS |
|---|---|---|
| Type de Blockchain | Orientée Blockchain publique | Orientée Blockchain Privée |
| Scalabilité | $>$ 100k | Efficacité inversement proportionnelle à la quantité des nœuds |
| Compromis l'attaque 51% | Déterminé selon la puissance de calcul des nœuds | Déterminé selon le capital enjeux |
| Difficulté du minage | Proportionnel à la puissance de calcul des mineurs | Proportionnel au capital détenu |
| Consomation d'energie | Energivore | Normal |

TABLE 2.11 – Comparaison des algorithmes de consensus

de nœuds et donc met le système à risque d'une attaque 51%. La PoS a beaucoup d'inconvénients raison pour laquelle elle n'est presque pas utilisée dans sa version de base, mais plutôt à travers des variantes telles que : le Delegated Proof-of-stake (DPOS)[Lar14], Ouroboros [Kia+17], Proof-of-activity [Ben+14] et etc. Il existe plusieurs systèmes Blockchain qui utilisent le PoS, et d'autres qui sont en cours de remplacement de la PoW par la PoS (ex. Ethereum [Woo+14], Dash [DD15], et Peercoin [KN12]).

### 2.6.2 Classification

Il est possible de classifier les Blockchains selon deux critères : l'accessibilité à la Blockchain et l'application de la Blockchain.

En fonction de l'accessibilité à la Blockchain, Il existe trois types de Blockchain : la Blockchain privée, la Blockchain à permission et la Blockchain publique. La limite entre ces trois types de Blockchain est encore controversée.

#### 2.6.2.1 La Blockchain Publique

La Blockchain publique se pose sur un réseau public auquel toute personne détenant un client peut s'y connecter. Tous les utilisateurs ont le droit d'écrire et de lire le registre. La publique vante sa grande distribution et la transparence des transactions. A contrario nous pouvons y observer des délais considérables de synchronisation, une absence totale de qualité de service et un débit de transaction faible. La plupart des crypto-monnaies connues utilisent la Blockchain du type publique.

#### 2.6.2.2 La Blockchain à permission

On parle couramment de Blockchain à permission quand il existe une ou plusieurs autorités délivrant des permissions aux utilisateurs. En d'autres termes, quand il existe un contrôle d'accès. Les Blockchains à permission sont moins distribuées à cause de la gestion d'accès qui devient très compliquée avec la croissance des nœuds utilisateurs. La Blockchain à permission se limite à des réseaux de taille moyenne. La taille réduite du réseau permet d'avoir un débit de transaction plus élevé et de fournir de la qualité de service.

#### 2.6.2.3 Blockchain Privée

Les Blockchains privées sont très souvent utilisées pour faire des tests. On parle de Blockchain privée quand tous les droits sont détenus par une seule entité et l'accès à ces droits est restreint. La Blockchain privée n'est pas forcément confidentiel.

En fonction de l'application de la Blockchain nous pouvons distinguer deux types de Blockchains : les Blockchains orientées crypto-monnaie et les Blockchains orientées smart-contract.

#### 2.6.2.4 Blockchain Orientée crypto-monnaie

*La crypto-monnaie est un moyen de paiement virtuel, utilisable essentiellement sur Internet, s'appuyant sur la cryptographie pour sécuriser les transactions et la création d'unités et échappant à tout contrôle des régulateurs et des banques centrales.* **Petit larousse**.

Les Blockchains orientées crypto-monnaie sont utilisées comme supports pour fournir un service de monnaie virtuelle (crypto-monnaie), et/ou un service de paiement électronique et/ou pour gérer des contrats élémentaux. La grande majorité des cas sont des Blockchains publiques. Nous pouvons prendre Bitcoin comme un exemple de ce type de Blockchain.

#### 2.6.2.5 Blockchain orientée smart-contrat

Les Blockchain orientées smart-contrat sont des Blockchains exploitées pour fournir des plateformes pour l'exécution des smart-contracts, pour la gestion des actifs et des smart-properties. Dans cette catégorie nous trouvons tous les trois types de Blockchain : privée, à permission et publique.

*A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs* **Nick Szabo**.

*Smart property is property whose ownership is controlled via the Bitcoin Blockchain, using contracts. Examples could include physical property such as cars, phones or houses. Smart property also includes non-physical property like shares in a company or access rights to a remote computer.* **Bitcoin Wiki**.

Les applications phares de cette catégorie sont Ethereum [Woo+14] et Hyperledger [Cac+16].

Le tableau comparatif 3.1 montre qu'il n'y a pas de Blockchain meilleure qu'une autre. Le choix de la Blockchain va fortement dépendre de l'application que nous voulons en faire et des contraintes que nous avons.

### 2.6.3 Blockchain Vs Base de données

La Blockchain n'est pas une base de données mais une DLT, il existe en effet des traits communs, mais la Blockchain reste différente de la base de données selon plusieurs aspects. Nous allons faire maintenant une comparaison entre les deux technologies selon différents critères listés ci-dessous.

#### 2.6.3.1 Origines

La Blockchain est beaucoup plus récente que la base de données. La technologie Blockchain naît en 2008 pendant que la base de données vient de bien loin. En effet il y a beaucoup de compétences relatives à la base de données, c'est une technologie bien maîtrisée qui a eu largement le temps d'évoluer. Ce qui n'est pas cas de la Blockchain, car la technologie commence à se faire de la place et doit encore mûrir. La base de données est bien connue donc facile et moins couteuse à mettre en place.

#### 2.6.3.2 CRUD

La base de données permet toutes les actions CRUD (Create, Read, Update et Delete). Il est possible de créer de nouvelles données ensuite les lire puis les effacer le nombre de fois que nous voulons dans la base de données. Ce qui n'est pas le cas avec la Blockchain, elle est append-only, on ne peut modifier et retirer les données qui y ont été insérées. On peut juste créer et lire le nombre de fois que nous voulons.

#### 2.6.3.3 Sécurité

La Blockchain est de base sécurisée, car l'intégrité, la non-répudiation et l'authentification sont des services de base. Un administrateur système ne peut manipuler à son vouloir les données présentes dans la Blockchain sans l'accord des propriétaires de ces données. Ce qui est très appréciée pour le domaine du partage des données. Dans le cas de la Base de données un administrateur peut modifier ou retirer certaines données et mêmes modifier ou effacer les logs pour cacher toutes traces sans que personne ne s'en aperçoive.

#### 2.6.3.4 Organisation

L'organisation interne des données est une grande faiblesse pour la Blockchain, car il n'y a pas d'organisation logique des données. La recherche des données à l'intérieur est un vrai casse-tête, il peut être nécessaire des heures pour retrouver certaines données, évidemment la dimension de la Blockchain aura un rôle majeur. A contrario dans la base de données, les données sont stockées en format clé et valeur, ce qui facilite la recherche des données stockées.

La base de données est plus connue et donc plus utilisée que la Blockchain. Nous allons maintenant définir une application que nous avons utilisée pour concevoir notre solution.

### 2.6.4 Namecoin

Namecoin [Kal+15] est une application de la Blockchain née en 2011 pour expérimenter d'autres applications outre à la crypto-monnaie.

*Namecoin est une technologie open source expérimentale qui améliore la décentralisation, la sécurité, la résistance à la censure, la confidentialité et la vitesse de certains composants de l'infrastructure Internet tels que le système de nommage (DNS) et le système de gestion identités.* Def. par **NAMECOIN**.

Nous pouvons définir Namecoin comme un système de nommage distribué, une sorte de DNS distribué. Elle permet d'associer une paire de clé asymétrique à un

nom de domaine. Sa Blockchain est un système de stockage distribué des associations d'informations en format clé/valeur et son réseau un système de distribution équilibré. Namecoin se pose sur une Blockchain publique dont l'accès en lecture est libre et gratuit pour tous. L'écriture est facturée sous forme de frais de transaction, réglé à travers une crypto-monnaie propre au système appelé Namecoin (NMC). Le système permet à l'utilisateur de stocker des clés jusqu'à 250 octets et des valeurs de 500 octets. Namecoin est à la base une Blockchain ce qui lui permet d'hériter certaines caractéristiques comme l'immuabilité des données, la résistance aux pannes, la non-répudiation et l'intégrité. Toutes ces caractéristiques font de Namecoin un système fiable. À la base le système a été conçu pour faire du DNS, mais au fil des années de nouvelle application exploitant Namecoin pour le stockage d'œuvres et de preuves numériques, pour la résolution légale de litiges notamment dans la revendication de la paternité de certaines œuvres, pour faire de la preuve de dépôt et etc.

## 2.7 Conclusions

À travers ce chapitre nous avons fait un état de l'art sur la PKI en général, puis nous nous sommes concentrés sur deux contextes d'utilisations, notamment la PKI X509 pour l'internet et la VPKI pour les systèmes C-ITS. Les différentes architectures, les services de sécurité assurés et les standards en vigueur ont été décrit. Ensuite, nous avons fait un état de l'art sur la problématique de la révocation, puis nous avons détaillé les différentes approches de distribution des listes de révocation utilisées dans la PKI X509 et dans VPKI. Enfin, nous avons proposé un tableau comparatif pour chaque approche en fonction de son domaine d'application. L'état de l'art sur la PKI et sur révocation nous permet de comprendre la pertinence du service de révocation dans la PKI. L'existence de plusieurs standards de révocation nous montre en premier lieu, la pertinence de notre sujet et ensuite l'absence d'une méthode de révocation idéale. Les chiffres nous montrent que la vérification de la révocation est couteuse en temps, en volume et en résiliense aussi bien dans la PKI véhiculaire que dans la PKI X509. L'état de l'art de la Blockchain nous permet de voir le résilence de cette technologie, ce qui justifie son intégration dans notre solution. L'ntégration des filtres de Bloom à notre solution est justifié par la nécessité de compresser les listes de révocation. Dans le prochain chapitre, nous allons décrire en détail nos contributions.

# Chapitre 3

# Contributions

## 3.1 Définition et conception de l'intégration de la PKI à la Blockchain

### 3.1.1 État de l'art sur les systèmes de révocation basées sur la Blockchain

Il existe de nombreuses contributions scientifiques basées sur la Blockchain qui traitent la révocation, nous allons les lister dans ce paragraphe.

La contribution [FVY14] propose une nouvelle PKI basée sur la Blockchain appelé Certcoin. La PKI Certcoin implémente un modèle de confiance distribuée proche au Web of Trust (WoT). Certcoin s'appuie sur une Blockchain de type publique appelée Namecoin. Le Certcoin est une PKI au modèle de confiance distribué, tous les utilisateurs peuvent certifier les certificats des autres utilisateurs. Chaque utilisateur devient une potentielle CA. Certcoin offre cinq types de services : l'enregistrement, l'actualisation, la lecture, la vérification et la révocation. L'enregistrement d'une nouvelle paire de clés peut être effectué par tous les utilisateurs. Le processus commence par la création d'une paire de clés, la clé publique sera signée par une autre paire de clés certifiée appartenant au même utilisateur ou à un autre utilisateur de confiance. La clé publique et la signature sont enregistrées dans la Blockchain par une transaction. Cette dernière sera inscrite dans la Blockchain qu'après validation de l'identité du signataire par le réseau. Le service d'actualisation est utilisé pour actualiser la clé publique d'un utilisateur. En effet, chaque paire de clés a une durée de vie, à laquelle expiration peut être nécessaire une actualisation. Chaque utilisateur peut modifier sa clé publique à travers une transaction contenant l'ancienne et la nouvelle clé publique. Cette transaction est signée par la clé privée de l'ancienne clé publique. Le service de lecture sert à vérifier la présence d'une clé publique dans la Blockchain et le service de vérification à assurer sa validité. La révocation d'une paire de clés se fait à travers le service révocation. Seul un propriétaire peut révoquer sa paire de clés. Chaque utilisateur possède deux pairs de clés de base, une online et une offline. La clé offline n'est presque jamais utilisée sauf cas exceptionnel tel que la révocation des clés. Cette paire de clés offline est utilisée pour signer la déclaration de révocation des clés online. Dans le cas où la paire de clés online est compromise, mais le propriétaire y a toujours accès, la transaction de révocation doit être signé par les deux clés privées (online et offline). Aucune solution n'a été proposée pour le cas où un utilisateur perd l'accès à ses deux paires de clés (online et offline).

L'auteur Mustafa Al-Bassam a proposé une infrastructure de gestion des clés (PKI) distribuées exploitant la Blockchain d'Ethereum [Al-17]. Le système SCPKI reprend le modèle de confiance de type WoT. Toute la PKI est décrite par un contrat déployé sur la Blockchain d'Ethereum. Le contrat peut être utilisé pour gérer ses propres clés ou les clés d'un utilisateur tiers. Ainsi, tout utilisateur peut stocker, lire,

actualiser et révoquer des clés à travers le contrat. Le modèle WoT permet que toute entité déjà enregistrée dans la Blockchain puisse certifier ses nouvelles clés ou les clés des nouveaux arrivants. La solution met à disposition aussi un service de révocation des clés de signature *RevokeSignature*. L'utilisateur peut révoquer que les clés qu'il a certifié. Il est à souligné que le protocole ne prend pas en compte le cas où un utilisateur n'a plus accès à sa clé privée. La méthode de révocation utilisée est la CRL, les listes de révocation sont stockées dans la Blockchain.

Les auteurs Lousie Axon et Michael Goldsmith proposent [AG16] une solution pour les problèmes de privacy dans les PKIs basées sur la Blockchain utilisant le modèle de confiance de type Web-of-trust (WoT). Le problème des infrastructures de gestion de clés exploitant la Blockchain comme support de distribution est l'exposition constante de l'association paire de clés - propriétaire. Ce fait remet en question la protection de la vie privée des utilisateurs. Toutes les clés publiques sont publiées dans la Blockchain et donc accessibles à tous. Les auteurs se proposent d'améliorer le protocole Certcoin [FVY14] en introduisant une couche d'anonymat, visant cacher le lien entre la paire de clés et l'identité du propriétaire. Toutes les étapes restent les mêmes que celle de Certcoin sauf l'actualisation. Pendant l'enregistrement, clés et identités de l'utilisateur sont publiées ensemble. À partir de ce moment toutes clés peuvent être actualisées sans dévoiler l'identité du propriétaire. Il devient ainsi difficile de faire le lien avec les clés utilisées pour l'enregistrement. Clés publiques et identité restent tout de même liées par un secret conservé par l'utilisateur, celui-ci peut à tout moment le dévoiler s'il le veut.

Il existe une autre proposition d'infrastructure de gestion de clés basée sur la Blockchain, appelé Authcoin [Lei+16]. Authcoin reprend l'idée générale de Certcoin en ajoutant une contrainte de contrôle d'identité. Le système lie un nom de domaine et un email à une paire de clés. Il est facile pour un attaquant de se faire passer pour quelqu'un d'autre parce qu'il n'y a aucun contrôle d'identité dans le modèle de confiance WoT. Authcoin fait face à ce problème en proposant un système d'authentification défi/réponse s'inspirant du système d'authentification de *Let's Encrypt*. Le système marche ainsi, un défi est envoyé à l'adresse mail du déclarant. Celui-ci doit le récupérer, le signer et le renvoyer, de manière à prouver la possession de l'adresse mail et de la clé revendiquée. Ces échanges sont stockés dans la Blockchain et accessible à tous. La Blockchain sera utilisée aussi comme preuve de propriété en cas de perte ou de vol de clés.

La contribution [Cor16] propose un système de gestion de clé au modèle de confiance hiérarchique utilisant la Blockchain comme base de données des certificats émis par différentes autorités de certification (CA). Différentes CAs partagent une Blockchain, cette dernière est utilisée pour stocker les certificats émis (appelés Blockchain Certificate). Les Blockchain Certificate sont des certificats non signés calqués à l'image des certificats x509. Ces certificats sont vérifiés par leurs présence dans la Blockchain. Leurs états de révocation sont aussi vérifiés par la présence de la transaction de révocation dans une autre Blockchain différent de celle des certificats. En résumé, la révocation d'un certificat se fait publiant ce dernier dans la Blockchain de révocation.

Le Cecoin [Qin+17] est une autre proposition d'infrastructure de gestion de clés basée sur la Blockchain. Le Cecoin vise un modèle de confiance intermédiaire au modèle hiérarchique et au modèle distribué (WoT). Le système proposé est composé

de trois acteurs fondamentaux. Le *Miner* qui représente la CA émettrice de certificats. La CA a la charge de vérifier, valider et inscrire dans la Blockchain les requêtes de certification, de renouvellement et de révocation des certificats. Le *Certificate owner* est un utilisateur de certificat, Il utilise le certificat pour authentifier sa paire de clés. Pour finir le *certificate user* est le consommateur des certificats, ce dernier vérifie la validité des certificats. Le mode de fonctionnement est le suivant : Chaque utilisateur émet une requête de certification/renouvellement/révocation qui sera validée et inscrite dans la Blockchain. Une fois inscrite dans la Blockchain chaque propriétaire pourra utiliser son certificat pour authentifier ses clés. Les consommateurs pourront vérifier la validité d'un certificat juste en vérifiant sa présence dans la Blockchain. Le processus est le même pour la révocation, un certificat est révoqué s'il existe dans la Blockchain une transaction de révocation contenant son identifiant. Le système met à disposition une seule Blockchain pour l'émission et la révocation des certificats. Le système peut être utilisé avec tous les types de Blockchain publique.

#### 3.1.1.1 Synthèse

Nous allons proposer dans ce sous-chapitre un tableau récapitulatif et comparatif des différentes contributions décrites précédemment en mettant en évidence :

— *Problématique :* Si la solution adresse d'autres problèmes de la PKI et indirectement la révocation ou directement la problématique de la révocation.

— *Type de Blockchain :* Le type de Blockchain utilisée : Privée, à permission ou publique.

— *Application :* L'application de la Blockchain exploitée.

— *Méthode :* Décrit l'approche utilisée par la méthode de révocation.

| **Approche** | **Problématique** | **T. Blockchain** | **Application** | **Méthode.** |
|---|---|---|---|---|
| [FVY14] | Indirectement | Publique | Blockchain propriétaire | CRL |
| [Ali+16] | Indirectement | Publique | Namecoin | CRL |
| [Al-17] | Indirectement | Publique | Ethereum | CRL |
| [AG16] | Indirectement | Publique | Blockchain propriétaire | CRL |
| [Lei+16] | Indirectement | Publique | Blockchain propriétaire | CRL |
| [Cor16] | Indirectement | Publique | Blockchain propriétaire | CRL |
| [Qin+17] | Indirectement | Publique | Blockchain propriétaire | CRL |

TABLE 3.1 – Comparaison des contributions

Toutes les propositions citées utilisent la Blockchain comme point de distribution des listes de révocation. Cette approche exploite l'architecture distribuée et la

résistance aux pannes de la Blockchain. L'utilisation de la Blockchain publique facilite l'accès aux données pour les utilisateurs. La CRL est definie ici comme une liste de transactions. Chaque transaction révoque un certificat. L'utilisation de la transaction comme structure de données pour chaque certificat / paire de clés révoquées à l'inconvénient d'être rigide, coûteux en terme de stockage et de calcul. Les données utiles dans la transaction peuvent arriver à représenter moins de 20 % de la dimension totale de la transaction. En outre, parcourir toute la Blockchain à la recherche d'une transaction pour vérifier un seul certificat peut introduire des délais non négligeables. Nous proposons par nos travaux une nouvelle méthode de révocation basée sur la Blockchain. Notre méthode introduit de nouvelles structures de données optimiser l'organisation des données et pour réduire la latence de vérification de la révocation des certificats. Nous proposons aussi trois nouveaux protocoles d'échanges de manière à optimiser les échanges entre les différentes entités de notre architecture.

### 3.1.2 Pourquoi la Blockchain ?

Nous avons choisi d'utiliser la Blockchain à cause des caratéristiques que notre méthode de révocation pourrait tirer de cette technologie. La Blockchain est un système distribué qui permet à de parfaits inconnu(es) de s'accorder sur une vérité ou sur une réalité. La Blockchain permet aussi d'échanger de l'information en toute sécurité dans le réseau Internet. La Blockchain va permettre aux clients de la PKI de télécharger les listes de révocation rapidement et en toute sécurité. Ceci reflète la relation existant entre la plupart des utilisateurs d'une PKI et ce que nous voulons atteindre comme service. La distribution de la Blockchain va permettre à notre méthode de révocation d'avoir un équilibrage des charges et une résistance aux pannes. Nous détaillons ci-dessous les caractéristiques de la Blockchain :

— *Intégrité* : Toutes les transactions sont hachées de même que tous les blocs. Le système assure l'intégrité sur tout le contenu de la Blockchain. L'intégrité par principe de la Blockchain permet de disséminer dans la Blockchain des fragments de la CRL sans devoir à assurer l'intégrité sur chaque fragment. Ceci permet de réduire la dimension des fragments et donc la quantité de données à télécharger par chaque utilisateur.

— *Non-Répudiation* : Toutes les transactions et les blocs sont signés. Chaque utilisateur s'authentifie dans le réseau à travers une paire de clés publique/privée. Il est difficile nié ses propres actes. Cette propriété permet de reconnaitre et d'authentifier facilement toutes les transactions émises dans la Blockchain par une autorité de certification.

— *Transparence* : Tous les échanges sont visibles et accessibles à tous. Cette propriété permet de lutter contre le misbehaving des autorités de certification. C'est-à-dire qu'il est possible de détecter rapidement et empêcher certaines CAs d'émettre des certificats dont elles n'ont pas droit.

— *Immuabilité* : Tout ce qui entre dans la Blockchain est censé y rester pour toujours. En effet, cette propriété nous assure que les fragments de CRL seront toujours présents et accessibles. Ceci est très utile pour certains cas comme les clés de signature, car même après expiration du certificat la clé peut continuer

à être utilisé pour vérifier des signatures.

— *Disponibilité* : L'architecture distribuée de la Blockchain lui permet d'être résistant aux pannes et aux attaques de déni de service. Cette propriété est cruciale, parce que les fragments doivent être accessibles à chaque moment par tous et le plus tôt possible. Un défaut d'accès aux listes de révocation ne produit pas une rupture de connexion (pour la plupart des navigateurs) mais une simple alerte qui peut être ignorée.

Toutes ces caractéristiques réunies font de la Blockchain une technologie candidate à notre solution.

### 3.1.3 Le systéme BL-CRLDP

Nous allons maintenant décrire l'architecture de notre approche de distribution.

#### 3.1.3.1 L'architecture du système

Notre système est composé de quatre entités (figure 3.1) : L'autorité de certification (CA) qui a le pouvoir de révoquer les certificats et émettre les listes de révocation dans la Blockchain qui fait office de stockage et de distribution. Le serveur est l'entité dont la révocation du certificat doit être vérifiée. Le client est un utilisateur, celui qui vérifie la révocation du certificat du serveur. Notre architecture est composée de trois protocoles d'échange : le protocole d'échange CA – Blockchain, le protocole d'échange Blockchain – serveur, le protocole d'échange serveur – client et le protocole d'échange client - Blockchain.

Ces quatre échanges sont complètement indépendants les uns des autres et non forcément successifs pour éviter que les défaillances d'un échange puissent impacter les autres.

##### 3.1.3.1.1 Description des échanges

Dans cette section nous allons décrire les différents échanges composant notre architecture.

**1 - L'échange CA $\Rightarrow$ Blockchain**

Cet échange (figure 3.2) s'effectue entre deux entités : la Blockchain et la CA. C'est une communication à un sens, de la CA vers la Blockchain. Ces deux entités communiquent à travers des transactions dont la structure est décrite dans le tableau 3.4. Le point de contact de la CA avec le réseau Blockchain est assuré par le serveur de révocation (repository). Celui-ci établit une connexion permanente avec un certain nombre de nœuds du réseau. Le serveur de révocation transmet toutes les RSIs par Broadcast à tous ses proches voisins.

La CA émet périodiquement (en fonction de sa politique interne) des transactions pour rafraîchir la Blockchain avec de nouvelles informations. Elle émet une transaction avec pour adresse source et pour adresse destination, sa propre adresse. Le champ data contient une RSI liée à un point de distribution. La RSI elle-même contient les informations du certificat à révoquer et un filtre de bloom calculé sur

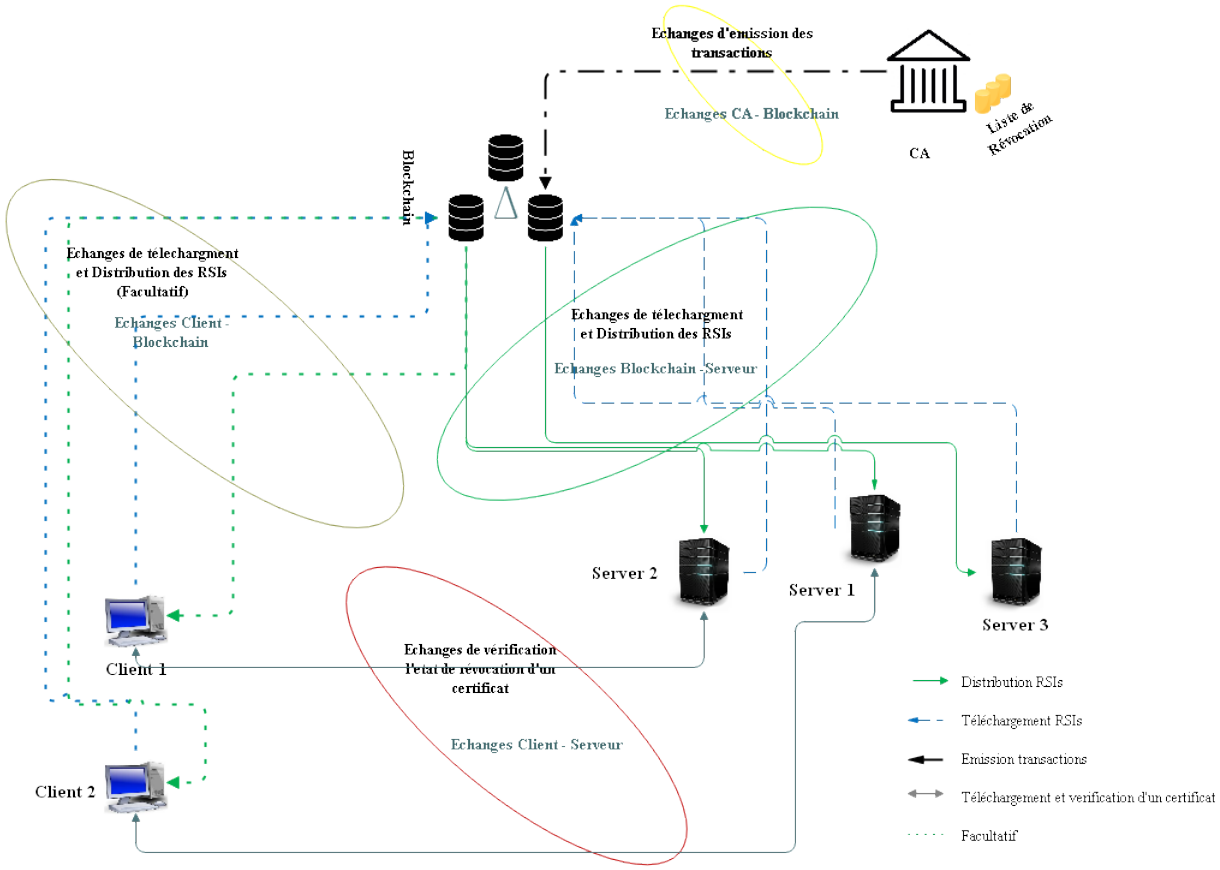



FIGURE 3.1 – Architecture du systéme

tous les certificats révoqués appartenant au point de distribution. L'objectif de cette transaction est juste d'enregistrer des données dans la Blockchain. Une seule transaction suffit pour révoquer un certificat, on ne peut révoquer plusieurs certificats par une seule transaction. C'est-à-dire que la CA va émettre autant de transactions que le nombre de certificats révoqués depuis la dernière actualisation des points de distribution. Chaque RSI contient les informations relatives au certificat à révoquer et la liste de tous les certificats déjà révoqués. Ainsi le point de distribution (fragment) est une chaine de RSI. Le chainage des RSIs est décrit dans la figure 3.3. Nous parlons de chainage de RSI car chaque RSI contient à l'instant $t$ la date d'émission de la précédente RSI et celle de la suivante. Ces dates reflètent la politique de révocation de la CA émettrice.

Nous observons deux cas à chaque fois qu'un point de distribution doit être actualisé :

— Cas1 : Il n'y a aucune nouvelle information, c'est-à-dire qu'aucun certificat n'a été révoqué depuis la dernière actualisation.

— Cas 2 : Il y a de nouvelles informations, un ou plusieurs certificats ont été révoqués depuis la dernière actualisation.

Dans le cas 1, une seule transaction sera effectuée et la RSI sera composée des informations suivantes :

— les champs *Serial Number* et *Revocation Date* sont laissés vides

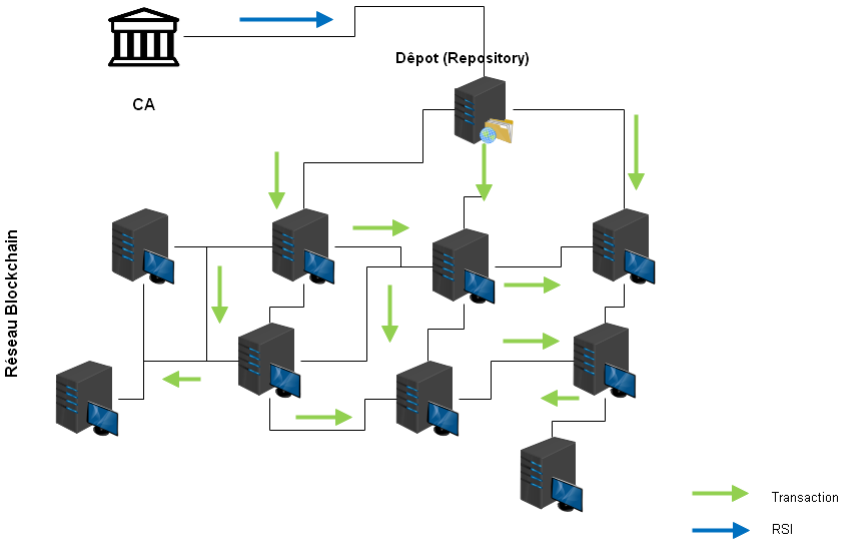



FIGURE 3.2 – Échange CA Blockchain

— le champ *Next Update* defini la date de la prochaine actualisation de la CRL

— le filtre reste le même que celui utilisé pour l'actualisation précédente

Dans le cas 2, les différents champs du RSI sont remplis correctement avec un filtre calculé sur la liste de tous les certificats révoqués jusqu'à ce moment-là. Toutes transactions émises sont signées de même que le RSI, car c'est la RSI qui sera utilisée par la suite par les utilisateurs, la transaction est juste le cocon.

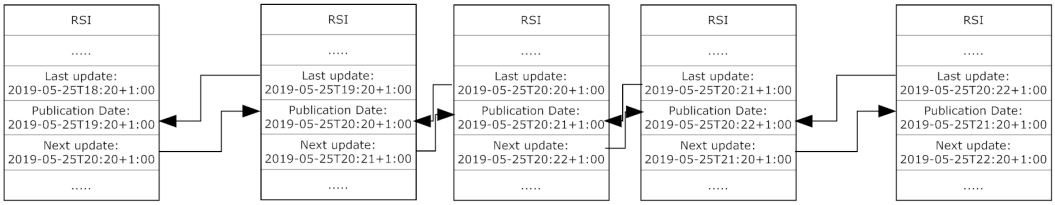



FIGURE 3.3 – Chainage des transations

### 2 - L' échange Blockchain ⇒ Server

Nous allons appeler *client* celui qui utilise les services de la PKI pour vérifier la révocation des certificats utilisateurs.

La Blockchain choisi est de type publique, donc accessible à tous. Un quelconque serveur/client peut s'y connecter à chaque moment pour récupérer toutes informations pertinentes. Toutes les transactions émises par les utilisateurs sont visibles et accessibles à tout moment par tous les nœuds du réseau. Les serveurs et les clients peuvent s'y connecter grâce à des APIs pour télécharger toutes les transactions émises par une CA spécifique. Le serveur peut récupérer sa RSI depuis la Blockchain de deux manières :

— En mode *non-sollicité* (figure 3.4) : Nous sommes dans le cas où le serveur peut maintenir une connexion permanente avec le réseau Blockchain. Ainsi le serveur pourra recevoir directement toutes les transactions émises dans le réseau et donc ceux de la CA. Il doit ignorer par défaut toutes transactions de la CA

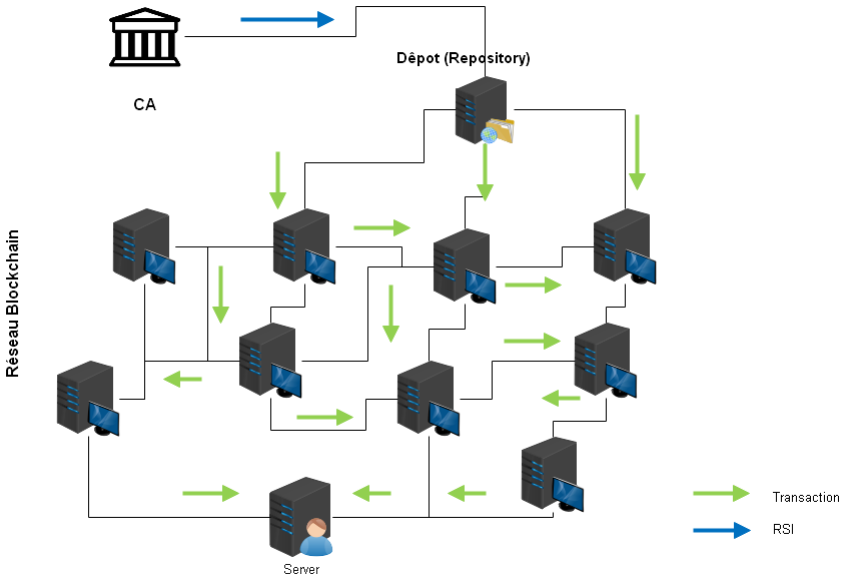



FIGURE 3.4 – Echange Blockchain - server mode non sollicité

non liée à son point de distribution. Ce mode est particulier car il n'est pas nécessaire télécharger et stocker toute la Blockchain

— En mode *sollicité* (figure 3.5) : Nous sommes dans le cas où le serveur ne peut maintenir une connexion permanente avec le réseau. Ce dernier pourra récupérer la RSI de la Blockchain chaque fois qu'il se connecte. Ce mode a l'inconvénient de devoir télécharger et stocker toute la Blockchain.

Le serveur va récupérer des transactions puis détacher de ces transactions les différentes RSIs qui seront ensuite stockées en local. Les transactions sont signées de même que les RSIs, ce qui permettra au serveur et au client de les authentifier. Si un serveur ne détient pas les dernières actualisations, aucune connexion avec un client ne pourra aboutir. Chaque RSI pointe à la précédente et à la suivante ainsi de suite formant une chaine, de façon à détecter facilement la dernière RSI publiée.

### 3 - L' échange Client ⇔ Server

Dans cette section nous décrivons comme notre approche de révocation s'intègre au protocole TLS comme décrit dans la figure 3.6. Les échanges commencent par le client qui peut être ou pas membre du réseau Blockchain, car il n'est pas censé interroger la Blockchain. Ce dernier demande à ouvrir avec le serveur une connexion TLS. Ensuite, les deux entités devront négocier dans le protocole TLS Handshake l'usage de la Blockchain comme point de distribution de la révocation. Ceci s'effectue à travers une nouvelle extension (b-revocation), qui va permettre aux deux de s'accorder sur l'usage de notre approche. Le TLS Handshake sera modifié ainsi :

Le client ouvre la connexion avec un *clientHello* contenant la liste des paramètres acceptés dans lequel va figurer la révocation par la Blockchain. Une fois reçut le clientHello si le serveur ne supporte pas la nouvelle extension, l'extension sera simplement ignorée. Dans le cas où l'extension est supportée et acceptée, le serveur va répondre avec un *ServerHello* et envoyer en même temps que son certificat le Revocation Status Infos (RSI) au client. Le client commence par vérifier la validité du

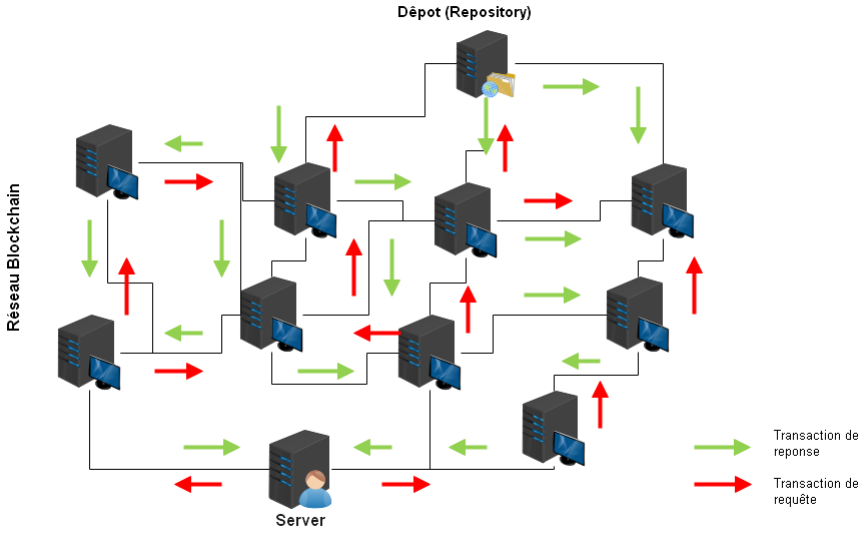



FIGURE 3.5 – Echange Blockchain - server mode sollicité

certificat serveur puis sa révocation. Le client vérifie que le bon RSI lui a été envoyé par le serveur, que le RSI n'est pas expiré puis l'appartenance du certificat du serveur au filtre de bloom. La vérification de l'appartenance du certificat du serveur au filtre récupéré depuis le RSI conduit à deux situations distinctes :

— Le filtre donne une réponse négative, c'est-à-dire que le certificat n'est pas inclus dans le filtre. Nous savons bien que le filtre de Bloom ne peut indiquer la présence d'un element quand elle n'est pas incluse (pas de faux négatifs). Donc dans le cas négatif, aucun doute, le certificat n'est pas révoqué ; l'échange peut donc continuer.

— Le filtre donne une réponse positive, il faudrait faire un ultérieur contrôle parce que le filtre peut produire des faux positifs. Ainsi, le client doit faire une requête au serveur pour télécharger tous les RSIs détenus par le serveur relatif à son point de distribution. Le client va ensuite lui-même récuperer chaque RSI pour contrôler directement tous les certificats révoqués sans passer par le filtre. Pour atténuer le nombre de données à télécharger, le RSI sera renvoyé par le serveur sans le filtre dans sa forme LRSI. La forme LRSI permet de gagner en temps et en bande passante, car le filtre occupe plus que la moitié de la dimension totale du RSI.

#### 4 - L' échange Client ⇔ Blockchain

L'échange entre la Blockchain et le client est facultatif, il est effectué seulement et seulement si le client possède un certificat. Dans ce cas de figure, le client va se comporter comme un serveur et l'échange sera pareil que celui du Blockhcain - Server.

### 3.1.3.2 Les structures

Le Blockchain-based CRL Distribution Points est un système composé : de deux nouvelles structures de révocation (Revocation Status Info (RSI) et Light Revocation

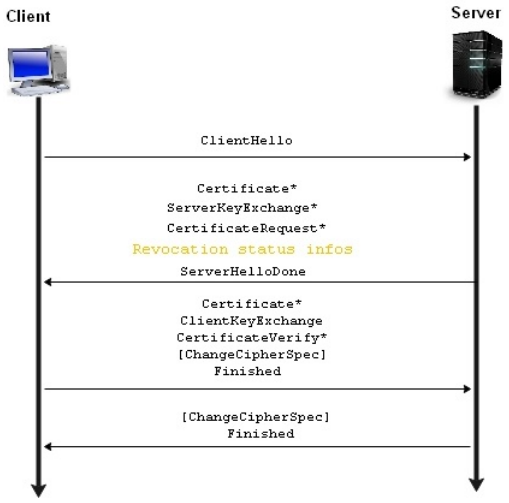



FIGURE 3.6 – Echange TLS modifié

Status Info (LRSI)). La revocation Status Info (RSI) ainsi que la Light Revocation Status Info (LRSI) sont des extensions de la structure de données utilisée par la méthode CRL Distribution Points (CRLDP). Le protocole de distribution utilisé est celui de la Blockchain. La CA publie périodiquement dans la Blockchain ses listes de révocation sous forme de plusieurs listes de RSIs. La publication dans Blockchain se fait par le biais d'une transaction. La Blockchain est le dépôt. Une fois la transaction émise, cette dernière est partagée tous les nœuds connectés à la Blockchain par le protocole de distribution de la Blockchain. La fréquence d'émission des transactions vers la Blockchain dépend de la politique interne de la CA. Une fois reçu la RSI, chaque utilisateur la garde en local pendant toute sa durée de validité. Cette dernière sera utilisée par les utilisateurs comme preuve de la validité de leurs certificats. Le LRSI est une structure de révocation compacte dérivée du RSI. Elle est échangée au niveau utilisateur.

#### 3.1.3.2.1 Revocation Status Info (RSI)

Nous avons conçu une nouvelle structure de données qui reprend et optimise la structure utilisée par la CRLDP, pour une distribution dans la Blockchain. Les différents champs du RSI sont décrits dans le tableau 3.2.

| Partie 1 | | Partie 2 | |
|---|---|---|---|
| Champs | Dimension (octets) | champs | Dimension (octets) |
| Version | 2 | RSI ID | 4 |
| Next Update | 4 | Publication Date | 2 |
| Bloom filter | 350 | Last Update | 4 |
| | | Serial Number | 15 |
| | | Revocation Date | 4 |
| Hash Algorithm | | 2 | |
| Merkle root | | 32 | |
| Signature Algorithm | | 2 | |
| Signature | | 64 | |

TABLE 3.2 – RSI Structure

Le RSI est composé de deux parties, qui sont les feuilles d'un arbre de Merkle dont la racine se trouve dans le champ (Merkle root). La partie 1 résume le passé à travers le filtre de bloom (voir annexe B). Cette répresente l'historique des certificats révoqués jusqu'à l'actualisation courante.

— le champ *Next Update* indique la date de la prochaine actualisation

— le champ *Version*, la version du protocole

— le champ *Serial Number* indique le numéro de série du certificat révoqué

— le champ *Publication date* indique la date à laquelle la transaction de révocation a été émise,

— le champ *Revocation Date* est présent s'il y a eu un seul certificat révoqué depuis la dernière publication. Dans le cas où il y a eu plusieurs révocations, plusieurs transactions seront effectuées par la CA, pour un débit d'une transaction par certificat révoqué

— Le champ *Next Update* s'associe au champ *Revocation Date*. Il permet d'organiser les certificats révoqués entre deux publications. Il est égal au champ *Publication Date*, sauf pour le dernier certificat révoqué. Ainsi faisant on obtient une chaine qui se termine par un RSI avec un *Next Update* relatif à la date de la prochaine actualisation de la CRL.

#### 3.1.3.2.2 Light Revocation Status Info (LRSI)

Nous proposons une structure de données allégée, sans filtre, appelée le Light RSI (LRSI). L'objectif de la LRSI est d'alléger les échanges entre utilisateurs, quand il faut transférer toute la CRL du serveur au client. La structure du LRSI est décrite dans le tableau 3.3.

| Partie 1 | | Partie 2 | |
|---|---|---|---|
| Champs | Dimension (octets) | champs | Dimension (octets) |
| Hash | 32 | RSI ID | 4 |
| | | Publication Date | 2 |
| | | Last Update | 4 |
| | | Serial Number | 15 |
| | | Revocation Date | 4 |
| Hash Algorithm | | 2 | |
| Merkle root | | 32 | |
| Signature Algorithm | | 2 | |
| Signature | | 64 | |

TABLE 3.3 – Structure allégée de la RSI (Light RSI)

La seule différence avec la structure RSI est l'absence du filtre de bloom. Le filtre de bloom est remplacé par un hash dans le LRSI. Cet hash assure l'intégrité de la partie 1 de la structure de données, il est calculé par la CA et non par le serveur. Le hash de la partie 1 combiné avec le hash de la partie 2 permettent de vérifier la signature et donc de certifier l'authenticité des données. Le hash sur la partie 1 ne sera pas recalculé par le client. Ceci ne pose pas de problème de sécurité vu que les données, car la partie 1 ne sont pas utilisées.

#### 3.1.3.2.3 Le Fragment (La liste des RSIs)

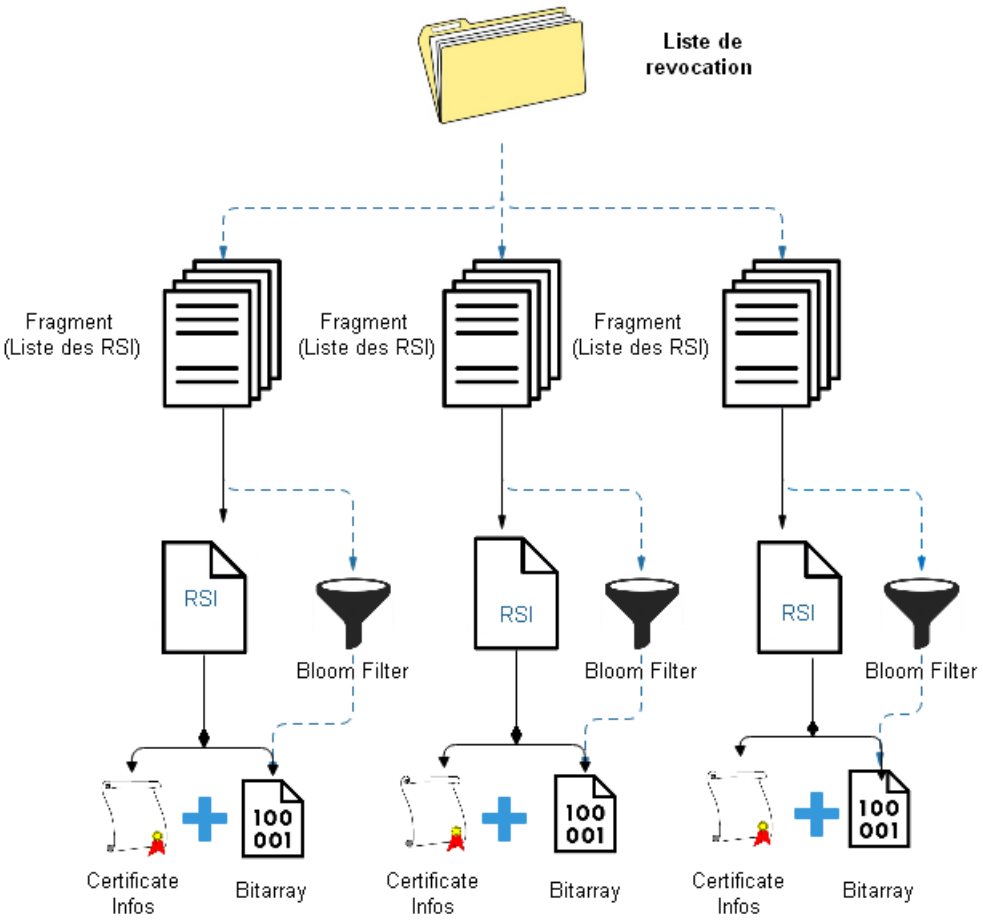



FIGURE 3.7 – Structure de la liste des RSIs

La liste de révocation est definie comme une liste de RSIs (figure 3.7). Dans l'objectif de réduire la quantité de données à télécharger par le client, la liste de révocation relative à une CA est découpée en plusieurs morceaux appelés "fragment". Ainsi, nous trouvons dans la Blockchain plusieurs listes d'RSIs (fragments ou points de distributions), chacun avec son propre identifiant. Chaque utilisateur, précisément chaque certificat est lié à un seul point de distribution explicitement indiqué dans son certificat. Chaque utilisateur aura à télécharger qu'un seul bout et non toutes les listes de révocation de la CA. La fréquence d'actualisation des points de distribution dépend de la politique interne de l'autorité émettrice.

#### 3.1.3.2.4 La structure d'une transaction

Nous proposons maintenant dans le tableau 3.4, la structure générique d'une transaction dans la Blockchain. C'est une transaction qui va héberger notre structure de révocation.

| Emetteur | Destinataire | Actif | Données / RSI |
|---|---|---|---|

TABLE 3.4 – Structure de la transaction

La transaction est composée des champs suivants :

— *Emetteur* : ce champ contient l'adresse de l'émetteur de la transaction

— *Destinataire* : ce champ contient l'adresse du destinataire de la transaction

— *Actif* : ce champ contient la somme de l'actif à transférer.

— *Données* : Ce champ peut contenir des données arbitraires de l'émetteur de la transaction. C'est ce dernier qui contient le RSI.

#### 3.1.3.3 Les étapes de la révocation des certificats

Nous avons principalement trois étapes :

— La première étape s'effectue entre l'autorité révocatrice des certificats et le dépôt ou point distribution. L'autorité révocatrice peut être la CA elle-même ou une autre autorité autorisée par la CA. Cette autorité émet une liste de révocation communément, qui est la liste de tous les certificats révoqués. La structure de la liste de révocation est communément appelée structure de révocation. C'est la structure qui est envoyée aux points de distributions. Notre première contribution est de proposer une nouvelle structure de révocation appelée Revocation Status Info (RSI). Le RSI prend la place de la structure de Révocation échangé entre la CA et le serveur de révocation.

— La deuxième étape s'effectue entre le point de distribution et les applications clientes. La manière selon laquelle l'utilisateur récupère les données depuis le point de distribution dépend de l'approche de distribution utilisée. Selon, l'approche CRL où CRLDP l'utilisateur va lui-même récupérer les listes de révocation depuis le point de distribution. Certaines approches proposent un protocole d'échange entre utilisateurs et la CA pour avoir des informations récentes, notamment le protocole OCSP. Notre contribution se propose d'utiliser la Blockchain comme point de distribution. La structure de données utilisée est la RSI.

— La troisième étape s'effectue entre utilisateurs, notamment entre client et serveur. Cette étape existe car le serveur doit se prémunir lui-même d'une preuve de la non-révocation de son certificat. C'est-à-dire que la charge de vérification de la revocation du certificat est déplacée du client vers le serveur. L'approche CRL et CRLDP prennent en charge que les deux premières étapes, les utilisateurs n'ont pas besoin de s'échanger des données supplémentaires relatives à la révocation parce que le client télécharge les listes de révocation directement du point de distribution. Nous avons aussi l'étape 3 dans le cas de l'approche OCSP stampling, car le serveur doit prouver lui-même la non-révocation de son certificat. Dans notre approche c'est le serveur qui doit contacter le serveur de révocation pour avoir la preuve de non révocation de son certificat. La structure de données échangée entre client et serveur et le RSI puis le LRSI (Light Revocation Status Info).

#### 3.1.3.4 Les caractéristiques du BL-CRLDP

Nous soulignons maintenant certaines caractéristiques du système BL-CRLDP :

— *Scalabilité* : Notre système repose sur un réseau de distribution distribuée avec plus de 5000 nœuds partout dans le monde. Il y aura toujours un nœud proche auquel se connecter pour récupérer la RSI. Il n'y a pas de single point of failure, car tous les nœuds du réseau possèdent l'information à laquelle nous voulons

accéder. Ceci permet une répartition des charges en cas de pic de requêtes. Notre approche passe à l'échelle car la croissance des utilisateurs du service n'impacte pas la quantité de données à télécharger pour vérifier la révocation d'un certificat. Notre système est composé de plusieurs points de distribution.

— *Rapide* : Notre système anticipe les requêtes en déplaçant la charge de vérification des certificats du client au serveur. Notre approche est différente des approches précédentes car le serveur anticipe en récupérant lui-même une preuve de la non-révocation de son certificat avant toute connexion. Ainsi faisant le client n'aura plus à ouvrir des connexions hors-bande vers des serveurs distants pour récupérer les listes de révocation.

— *Non-connecté* : Notre système peut-être défini non-connecté car aucune connexion supplémentaire n'est nécessaire pour vérifier révocation d'un certificat pendant une communication. Ainsi nous définissons notre système offline car tous le nécessaire pour vérifier le certificat serveur est détenu en locale par le serveur lui-même.

#### 3.1.3.5 Comparaison du système BL-CRLDP avec les approches concurrentes

Nous allons maintenant faire une comparaison (décrite dans le tableau 3.5) entre notre solution et les différentes approches présentés précédemment à travers quatre critères de comparaison que nous listons maintenant :

— *Consultation* : La consultation est un critère de comparaison très important. Il permet de mettre en évidence le fait que la solution permette une consultation des listes de révocation pour chaque certificat vérifié en ligne ou hors ligne. Un accès hors ligne aux listes de révocations est privilégié car permet un accès rapide et facile non dépendant d'une connexion à internet. Un accès en ligne est moins plébiscité à cause de sa forte dépendance de la qualité du réseau.

— *Périodicité* : Le critère de périodicité permet de mettre en avant le fait que l'utilisateur accède aux nouvelles informations périodiquement ou immédiatement.

— *Charge de travail* : Ce critère permet de souligner la répartition des charges le long du processus de vérification de la révocation entre client et serveur. Ce critère nous montre qui possède les listes de révocation entre client et serveur.

— *Structure* : Ce critère permet de distinguer les différentes structures de données utilisées par les méthodes de révocation.

— *Approche d'obtention* :Ce critère permet de souligner le mode utilisé par l'utilisateur pour récupérer les listes de révocation stocker dans le dépôt. Il existe de deux manières de procéder :

  — *Sollicité* : l'utilisateur effectue une requête ou va lui-même les chercher

  — *Non Sollicité* : l'utilisateur reçoit les informations sans devoir aller les chercher

## 3.2 Révocation des certificats pseudonymes : définition et conception d'une solution optimale

La PKI (Public Key Infrastructure) est une l'infrastructure de sécurité utilisé pour authentifier les clés publiques des véhicules. Nous parlons de Vehicular PKI (VPKI). Un véhicule détient deux types de certificats, un certificat à long terme (EC) et un

| | Consultation | Periodicité | C. de travail | Structure | Obtention |
|---|---|---|---|---|---|
| CRL | Offline | Periodique | Coté Client | CRL | Sollicité |
| delta-CRL | Offline | Periodique | Coté Client | CRL | Sollicité |
| CRL.dist.points | Offline | Periodique | Coté Client | CRL | Sollicité |
| CRS | Online | Temps Réel | Coté client | CRL | Sollicité |
| CRT | Offline | Periodique | Coté Client | Hash | Sollicité |
| OCSP | Online | Temps Réel | Coté client | Message OCSP | Sollicité |
| OCSP Stapling | Offline | Temps Réel | Coté serveur | Message OCSP | Sollicité |
| SCVD | - | - | - | - | - |
| BL-CRLDP | Offline | Periodique | Coté Client | RSI | Sollicité et non sollicité |

TABLE 3.5 – Comparaison des approches de révocation

certificat à court terme (AT) émis et signé tous deux par une autorité de confiance. Le certificat long terme est utilisé pour prouver de manière unique son identité, mais il

n'est jamais dévoilé à l'extérieur de la station ITS. Il est utilisé ad vitam par la station. Le certificat, à court terme comme le nom l'indique est un certificat pseudonyme à durée limitée. Sa durée de validité dépend du standard adopté, c'est celui qui est utilisé pour prouver son identité vers l'extérieur. Les échanges V2X sont signés par les certificats à pseudonymes. Ceci conduit à un renouvellement permanent des certificats en fonction de leur durée de vie pour éviter la traçabilité des véhicules.

La quantité maximale de certificats pseudonymes qu'une station ITS peut détenir à la fois n'est pas défini par le standard de référence. Il existe deux façons d'équiper un véhicule de ses certificats pseudonymes. Le véhicule peut être équipé progressivement de certificat, c'est-à-dire qu'il possède un seul certificat à la fois qui est renouvelé à son expiration. Il peut être équipé directement d'un groupe de certificats renouvelé périodiquement. Mes recherches prennent en compte que le deuxième cas de figure. La PKI véhiculaire assure la révocation des deux types de certificat, la révocation des certificats a long-terme et la révocation des certificats pseudonymes à court-terme. Cependant, mes travaux ciblent exclusivement la révocation des certificats pseudonymes à court terme. Le processus de vérification des certificats inclut une étape très importante, celle de la vérification de la révocation. Les raisons qui peuvent contraindre une CA à révoquer un véhicule avant l'expiration naturelle de son certificat sont multiples et ont déjà été définies dans les chapitres précédents, à ces raisons nous pouvons ajouter le vol du véhicule. La CA procède révoquant le certificat à long-terme du véhicule, cette information est publiée dans la liste de révocation des certificats à long terme révoqués (CTL). Ensuite, la AA révoque à son tour tous les certificats à court-terme de la station, publiant cette information dans la liste des certificats à court-terme révoqués (CRL).

### 3.2.1 Etude Statistique

En général, la AA assigne un groupe de certificats aux véhicules, qui devront être utilisés pendant une période de temps précise. La quantité de certificat assigné par véhicule et la période de renouvèlement n'a pas été arrêter par le standard pris en compte. Nous avons fait une estimation pour obtenir des chiffres partant des informations récupérées des etudes statistiques ménées par L'Eurostat [Eur19b] et par l'association des constructeurs européens (ACEA) [ACE]. La réglementation Européenne impose une révision obligatoire à chaque nouveau véhicule aprés quatre ans et ensuite chaque deux ans. Nous avons donc choisi quatre ans comme période de renouvellement, ainsi le groupe de certificat sera renouvelé pendant la révision. Une enquête effectuée par l'association des constructeurs européens (ACEA) [ACE] montre que les Français font en moyenne deux trajets et demi par jour pour des raisons professionnelles ou privées. Le temps de conduite journalière se situe en moyenne entre 1 heure et 1h30. Ces données nous ont permis de choisir 1 heure comme durée de vie de référence du certificat pseudonyme du véhicule. Selon nous 1 heure de durée de vie d'un certificat pseudonyme permettrait de limiter la traçabilité des véhicules dans leurs quotidiens. Partant d'une durée de vie d'une heure par certificat, chaque véhicule aura à stocker *35040* certificats pseudonymes qui couvriront la période des quatre années. Nous avons choisi la France comme aire géographique d'étude, parce que les études statistiques étaient facilement accessibles. L'Eurostat publie chaque an des études statistiques sur l'industrie automobile en France [Eur19b].

Cette étude nous informe que le nombre de voitures en circulation en France était de *33,020,000* véhicules en 2018. La moyenne des véhicules volés sur ces dix dernières années est de *177300* selon une étude publiée par le gouvernement Français

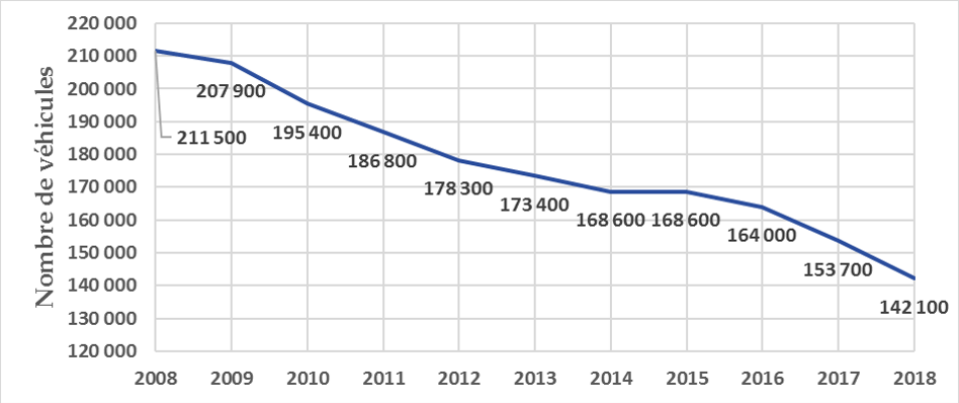



FIGURE 3.8 – Evolution du vol de véhicule en France [Pas+18]

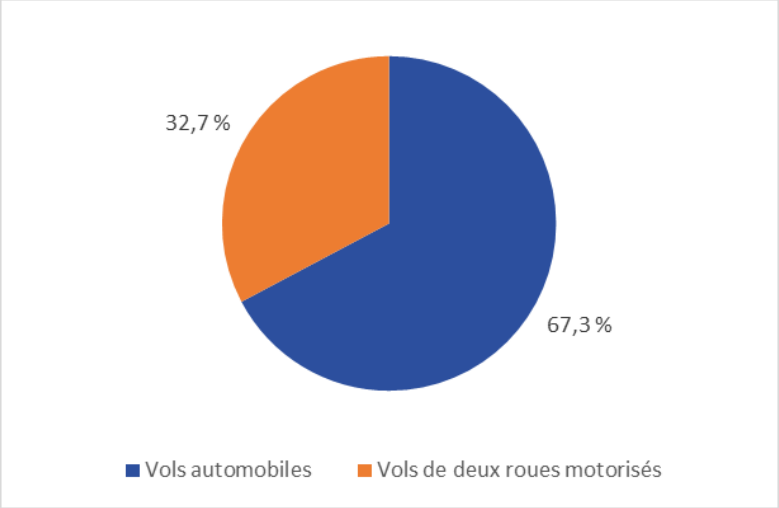



FIGURE 3.9 – Répartition [Pas+18]

[Eur19a] et l'union Européenne [Pas+18]. Cette moyenne est en continuelle décroissance 3.10. Nous résumons toutes ces informations dans le tableau 3.6. Ces données sont très importantes, car nous permettent d'estimer le taux de révocation des véhicules et donc la dimension des listes de révocation. Nous n'avons pu estimer les autres causes de révocation pour absence de données.

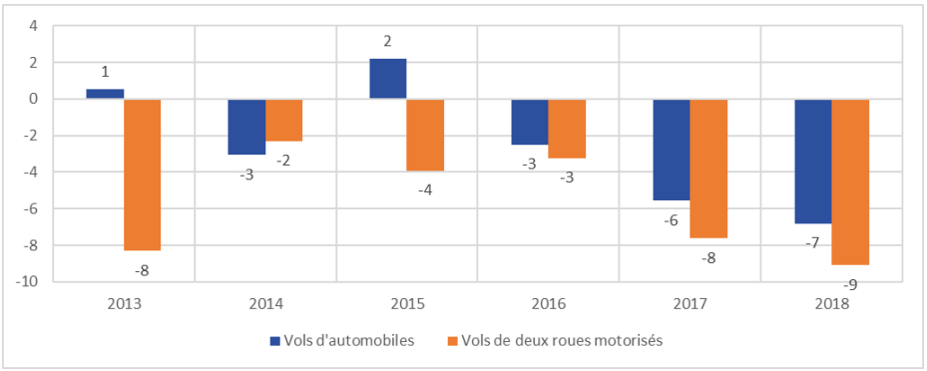



FIGURE 3.10 – Taux de décroissance [Pas+18]

La révocation d'un véhicule induit la révocation de plusieurs certificats pseudonymes de courte durée qui vont élargir en volume les listes de révocation. Les communications véhiculaires ajoutent une nouvelle dimension qui ne favorise pas la vérification de la révocation des certificats.

| **Champs** | **Valeur** |
|---|---|
| Quantité de certificat par véhicule | 35040 |
| Quantité de véhicule en circulation | 33,020,000 |
| Nouvelles Immatriculations | 2,692,748 |
| Estimation possible de nombre de certificat émis par année | $9.4 \cdot e^{10}$ |
| Nombre de véhicules volés | 177300 |
| Pourcentage révocation par rapport aux nouvelles immatriculations | 0.06% |
| Pourcentage révocation par rapport aux véhicules en circulation | 0.054% |

TABLE 3.6 – Résultats statistiques

### 3.2.2 Etat de l'art

La recherche bibliographique nous a permis de constater que plusieurs contributions ont été faites au fil des années en guise d'optimiser les solutions de distribution de la révocation de la VPKI. Je liste maintenant toutes les contributions trouvées relatives et utiles aux objectifs de cette thèse :

Jason j. Haas et Yih-chun Hu proposent dans la [HHL09] une méthode de révocation basée sur une version optimisée de la delta-CRL pour la dissémination des listes de révocation. Chaque véhicule est liée à des certificats pseudonymes, des nombres entiers et une clé symétrique $s_k$. En d'autres termes chaque véhicule partage avec le AA une clé symétrique. Un entier connu par tous $p$ est à affecter à chaque certificat de chaque véhicule.La clé symetrique est partagée avec la AA. Chaque certificat pseudonyme contient un champ spécial appelé Certificate-ID qui est le cryptogramme obtenu en chiffrant l'entier $p$ avec la clé $s_k$. Chaque fois qu'une AA révoque un véhicule, elle n'inscrit dans la CRL que la clé symmetrique partagée avec le véhicule et non la liste de tous ses certificats pseudonymes. Ainsi, nous observons une réduction considérable des données à inscrire dans CRL à chaque révocation. Pour vérifier la révocation d'un certificat, il suffit de récupérer toutes les clés présentes dans la CRL et essayer de déchiffrer avec chaque clé le champ certificate-ID du certificat à vérifier. Le certificat est révoqué s'il existe une clé dans la CRL capable de déchiffrer son certificate-ID. Cette solution est interessante que dans le cas de CRLs de petite dimension. En effet le chiffrement est couteux en terme de calcul et de latence, ce qui ne facilite pas l'utilisation de cette methode sur de grosses CRL. Pour faire face à ce problème les auteurs proposent l'usage des filtres de bloom en amont de l'interrogation de la CRL. Ainsi chaque véhicule récupère de la CRL toutes les clés y contenu puis calcule tous les cryptogrammes liés à chaque clé dans la CRL en chiffrant les différents entiers. Ces cryptogrammes sont ensuite enregistrés dans un filtre de bloom qui est stocké en local. Vérifier la révocation d'un certificat se réduit à contrôler l'appartenance du *Certificate-ID* au filtre. Pour conclure, chaque véhicule

peut anticiper les faux positifs du filtre en utilisant les certificats de Backup. C'est-à-dire que chaque véhicule stocke en local un certain nombre de certificats utilisés pour remplacer les certificats qui sont susceptibles de provoquer de faux positifs.Le filtre de bloom est une structure probabiliste, donc il est difficile de prévoir son comportement. Les certificats de backup peuvent se révéler aussi cause de faux positifs, raison pour laquelle ils ne peuvent pas être une solution acceptable aux faux positifs. La scalabilité du système est donc remise en cause.

La contribution [Ray+06] propose trois nouveaux protocoles pour la vérification de la révocation. Le protocole (1) Revocation using Compressed Certificate Revocation List (($RC^2RL$) est utilisé pour comprimer avec les filtres de bloom et pour disséminer des listes de révocation. La distribution des listes de révocation vesr les véhicules est assurée par les RSU. (2) Le deuxième protocole a pour objectif de contraindre les différentes stations au respect des normes de révocation. Le deuxième protocole propose une autre approche, Ccest-à-dire au lieu de révoquer un véhicule inscrivant tous ses certificats dans la CRL, on peut le révoquer en l'empêchant de faire usage de ses certificats. Ceci peut être fait en donnant le pouvoir aux AA d'obliger toute station révoquée à stopper toutes activités cryptographiques. Ce protocole est nommé Révocation of the TPD (RTPD). (3) Pour finir, le troisième protocole, le protocole Distributed Revocation Protocol (DRP) a pour objectif de permettre à chaque station d'être capable de détecter les stations malveillantes dans leurs rayon de communication. Ce protocole travaille sur le revers de la médaille, c'est-à-dire le cas figure où il y a une mauvaise répartition de l'infrastructure routière et le AA ne peut être contacté. La contribution ne propose pas de soulution faux positifs des filtres de bloom.

La contribution [PMH08] propose une méthode similaire à l'approche de révocation par la CRLDP pour optimiser scalabilité de la CRL. Les auteurs proposent des listes de révocation divisée en plusieurs fragments autonomes et vérifiables de sorte à ce qu'il ne soit pas obligatoire de posséder tous les fragments pour vérifier l'état de révocation d'un certificat, mais seulement un certain nombre. Une station n'aura plus à télécharger toute la CRL pour pouvoir vérifier la révocation d'un certificat, mais sera capable de le faire avec juste quelques fragments. Ceci permet une optimisation de l'usage de la bande passante et donc une atténuation de son l'impact sur les autres trafics. Le système est loin d'être optimale, il devient très complexe à gérer côté CA avec la croissance des listes de révocation.

La contribution[LHH08] vise la distribution des listes de révocation. Les auteurs proposent de réduire la latence en proposant une distribution de type épidémique.

#### 3.2.2.1 Synthèse

Nous proposons trois critères pour résumer et comparer les différentes contributions citées ci-dessus :

— **Proposition** : Ce critère définit le service proposé par la méthode de révocation : la distribution, la compression des listes de révocation ou les deux.

— **Approche de distribution** : Ce critère définit la méthode de révocation que la proposition vise à optimiser. En d'autres mots, la CRL, la delta-CRL ou la CRLDP.

— **Technique de compression** : Ce critère définit la technique de compression utilisée sur les listes de révocation : les filtres de bloom, la crypto ou autres.

| Contribution | Proposition | App. distr. | Tech. Comp. |
|---|---|---|---|
| [HHL09] | Compression | CRL | Crypto et Filtres de bloom |
| [Ray+06] | Compression | Non-Standard | Filtres de bloom |
| [Ray+07] | Compression | CRL | Filtres de bloom |
| [Rig+17] | Compression | Aucune | Filtres de bloom |
| [PMH08] | Distribution | CRLDP | – |
| [LHH08] | Distribution | CRL | – |

TABLE 3.7 – Comparaison des contributions

La majorité des propositions citées proposent une optimisation de la CRL, exception faite de la proposition [**papadimitratos2008 certificate**] qui utilise l'approche CRLDP sans aucune compression de données. L'état de l'art nous montre que la CRL est l'approche la plus utilisé par la pluspart des methodes de révocation.

### 3.2.3 L'adaptabilité des approches de distribution préexistante

Nous savon que les communications véhiculaires sont plus contraignante et dons les methodes de révocation traditionnelles doivent pouvoir répondre à des contraintes pour être applicables. Nous allons analyser la plupart des défis que nous observons dans le véhiculaire.

— *Mobilité* : Les véhicules sont en continuel mouvement avec une fréquence et une vitesse variable, ils ne suivent forcément pas les mêmes itinéraires. Cette mobilité inclut un changement continuel de la topologie du réseau et la difficulté à maintenir des sessions de communication.

— *Bande Passante* : Les communications véhiculaires sont contraignant en terme de bande passante due en partie aux balises et aux messages de sécurité qui transitent constamment. Nous disons qu'une approche peut répondre à cette contrainte, si la méthode peut faire mieux qu'une méthode de révocation basé sur la CRL.

— *Stockage* : Les stations ITS sont théoriquement à faible capacité de stockage. L'adjectif faible est relatif à une comparaison aux noeuds non mobiles que nous observons dans le réseau internet. Une méthode de révocation arrive à faire face à cette contrainte si le volume de données à stocker évolue linéairement en fonction du nombre de véhicules révoqués.

— *Puissance de calcul* : Les stations ITS sont théoriquement dotées de calculateurs à faible puissance, due au retard technologique dans ce domaine et à la structure du véhicule. L'adjectif faible est relatif à une comparaison a la puissance de calcul des noeuds traditionnelles. Nous évaluons la puissance de calcul en fonction du nombre de signatures *sig* à vérifier et du nombre de hash à calculer *hsh*.

— **La CRL [08] micali2008efficient** est la liste des certificats révoqués par une CA. La liste des certificats révoqués est publiée périodiquement (en fonction de la politique de la CA) dans un fichier appelé (CRL) stocké dans un dépôt qui est librement accessible en ligne.

  — *Mobilité* : Il n'est pas obligatoire d'ouvrir une connexion pour vérifier l'état de révocation d'un certificat. Le téléchargement de la CRL est effectué en amont et stocké en locale. La mobilité des véhicules à de l'impact sur la phase d'actualisation de la CRL et seulement, précisément sur le

téléchargement de la CRL. Nous pouvons conclure que la CRL peut faire face à la grande mobilité des véhicules.

- *Bande passante* : La CRL est télécharger periodiquement, ce qui optimal en terme d'usage de la Bande passante.
- *Stockage* : La CRL n'est pas optimale face à la contrainte du stockage, car sa croissance est exponentielle de la grande quantité de certificats pseudonymes détenus par chaque véhicule.
- *La puissance de calcul* : La quantité de calcul nécessaire pour vérifier une CRL est estimé à *1 sig* et *1 hsh*.

— **la delta-CRL [08] micali2008efficient** est une extension de la CRL avec l'objectif d'éviter de devoir télécharger toute la CRL à chaque fois. Le système est composé de deux parties, la CRL et les mises à jour de la CRL. La CRL est téléchargée seulement pendant la phase d'initialisation de la station ITS. Successivement ce sont les mises à jour qui sont téléchargées. Le client doit télécharger seulement une seule fois toute la CRL, puis successivement les mises à jour. Cette approche a l'inconvénient de rester encore étroitement lié à la CRL.

- *Mobilité* : Il n'est pas nécessaire d'ouvrir une connexion ou envoyer des messages supplémentaires pour vérifier l'état de révocation d'un certificat. Le téléchargement de la CRL est effectué en amont et stocké en locale. La mobilité des véhicules à de l'impact sur la phase d'actualisation de la CRL, précisément sur le téléchargement de la CRL et seulement. Nous pouvons conclure que la CRL peut faire face à la grande mobilité des véhicules.
- *Bande passante* : Cette approche est encore plus optimal que l'approche CRL.
- *Stockage* : La delta-crl n'est pas optimale face à la contrainte du stockage, car sa dimension totale évolue exponentiellement à cause de grand nombre de certificats pseudonymes détenus par chaque véhicule.
- *La puissance de calcul* : La quantité de calcul nécessaire pour vérifier une delta-CRL est pratiquement égale au cas de l'approche CRL. En général une signature (*1 sig* à vérifier et une fonction de hachage *1 hash* à calculer.

Nous avons fait une analyse comparative des approches de distribution des listes de révocation utilisées par la PKI x509 pour voir celui qui arrivait à avoir le bon compromis face aux contraintes de l'environnement véhiculaire. Nous les listons ci-dessous :

— **OCSP [08][pmicali2008efficient]** (Online Certificate Status protocol) d'utiliser le service de révocation sans devoir télécharger les listes de révocation. Le systéme est composé de dex entités, le client qui veut verifier la révocation d'un certificat cherche les informations de révocation (OCSP Requester) et le serveur dédié qui détient cette information (OCSP responder) reçu de la CA sous forme d'une CRL. Le principe est que le OCSP requester effectue une requête appelée "OCSP request" avec un ou plusieurs numéros de série et d'autres données à un serveur dédié appelé (serveur OCSP). Une fois reçu la requête, le serveur formule une réponse appelée (OCSP Response), horodaté et signé. Il n'est donc plus nécessaire télécharger toute la liste de révocation.

- *Mobilité* : Le protocole OCSP nécessite de l'ouverture d'une connexion à deux échanges vers un tiers serveur, ce qui est mis à dure épreuve par la

grande mobilité des nœuds et le changement continuel de la topologie du réseau. Le protocole OCSP est fortement impacté par la mobilité. Nous pouvons affirmer que le protocole OCSP ne peut faire face à la mobilité des véhicules.

- *Bande passante* : Le protocole OCSP répond aisément à la contrainte de la bande passante, car la vérification d'un certificat ne nécessite que le téléchargement d'une petite quantité de données. Quelle que soit la dimension des listes de révocation la quantité de données à envoyer et à recevoir sera toujours la même.
- *Stockage* : Le protocole OCSP peut faire face à la contrainte du stockage, car il n'y a pas de liste de révocation à télécharger mais que données attestant la validité du certificat. La quantité de données à stocker n'est pas influencée par la dimension de la liste de révocation.
- *La puissance de calcul* : La quantité de calcul nécessaire pour envoyer 1 requête et contrôler une réponse OCSP est estimé à *1 sig* à vérifier et *1 hsh*. Nous rappelons qu'il n'est pas obligatoire signer une requête.

- **OCSP Stapling [08] :** Le principe de cette approche est le même que OCSP à quelques différences prés. Les réponses du serveur OCSP sont stockées en locale par le propriétaire du certificat avec une durée de vie. Ainsi pendant un certain temps le client n'aura pas à ouvrir de nouvelles connexions vers le serveur de révocation pour redemander les mêmes informations. Les réponses OCSP sont signées et horodatées ce qui permet au client de vérifier la validité du certificat.

  - *Mobilité* : Le protocole OCSP Stapling nécessite de l'ouverture d'une connexion à deux échanges vers un tiers serveur pour récupérer l'état de révocation d'un certificat, ce qui est mis à dure épreuve par la grande mobilité des nœuds et le changement continuel de la topologie du réseau. Le protocole OCSP est fortement impacté par la mobilité. Il est aussi vrai qu'avec le protocole OCSP Stapling une fois reçu l'état de révocation d'un certificat il ne sera pas nécessaire redemander pendant un certain temps. Ceci n'a pas de grand impact car les véhicule sont en continuel mouvement, les échanges sont brefs et variés. Nous pouvons affirmer que le protocole OCSP ne peut faire face à la mobilité des véhicules.
  - *Bande passante* : Le protocole OCSP répond aisément à la contrainte de la bande passante, car la vérification d'un certificat ne necessite que le téléchargement d'une petite quantité de données. Quelle que soit la dimension des listes de révocation la quantité de données à envoyer et à recevoir sera toujours la même.
  - *Stockage* : Le protocole OCSP peut faire face à la contrainte de stockage. Car il n'y a pas de liste de révocation à télécharger mais que données attestant la validité du certificat. La quantité de données à stocker n'est pas influencée par la dimension de la liste de révocation.
  - *La puissance de calcul* : La quantité de calcul nécessaire pour envoyer une requête et contrôler une réponse OCSP est estimé à *1 sig* à vérifier et *1 hsh*. Nous rappelons qu'il n'est pas obligatoire signer une requête.

- **la CRL distribution point [08]** est une approche qui permet de reduire le volume de données à télécharger pour verifier la révocation d'un certificat. Les listes e révocation sont divisés en plusieurs fragments. Ainsi les fragments sont

mis dans un ordre logique pour permettre à chaque client de télécharger seulement les fragments nécessaires et non toute la CRL. C'est-à-dire que chaque certificat soit lié à un point de distribution.

- *Mobilité* : Il n'est pas nécessaire d'ouvrir une connexion pour vérifier l'état de révocation d'un certificat. Le téléchargement du fragment de CRL est effectué en amont et stocké en locale. La mobilité des véhicules a de l'impact sur la phase d'actualisation de la CRL qui est périodique, précisément sur le téléchargement des mises à jour. Nous pouvons conclure que la CRLDP peut faire face à la grande mobilité des véhicules.
- *Bande passante* :Les fragments sont téléchargés periodiquement à chaque actualisation, ce qui est optimal en terme d'usage de la Bande passante.
- *Stockage* : La CRLDP peut répondre à la contrainte du stockage, car la croissance des fragment est maitrisé. La croissance des listes de révocation est exponentielle, mais la division en fragment permet d'attenuer l'impact sur chaque véhicule.
- *La puissance de calcul* : Le quantité de calcul nécessaire pour vérifier l'état de révocation d'un certificat est estimé à *1 sig* à vérifier et *1 hsh*.

- **Simple Certificate Validation Protocol (SCVP)SCVP**, selon cette approche le client peut déléguer la vérification de la révocation à un serveur qui peut être de confiance ou pas. Ce serveur pourrait redistribuer les informations acquises de plusieurs manières tel que l'approche CRL, CRDP, OCSP et ainsi de suite.
  - *Mobilité* : Pour définir l'impact de la mobilité sur le système il faut d'abord définir la position du serveur auquel la révocation est déléguée. Si le serveur est interne au véhicule, tout dépendra alors de l'approche de révocation utilisée. Dans le cas échéant, c'est-à-dire, le serveur est externe à la station, ce qui serait compliqué à mettre en œuvre car le serveur va devoir être mobile autant que le véhicule.
  - *Bande passante* : La réponse du système à cette contrainte va dépendre de la méthode de révocation utilisée.
  - *Stockage* : La réponse du système à cette contrainte va dépendre de la méthode de révocation utilisée.
  - *La puissance de calcul* : La réponse du système à cette contrainte va dépendre de la méthode de révocation utilisée.

Le tableau 3.9 est une synthèse du comportement des différentes approches de distribution vis-à-vis des contraintes imposées par l'environnement des communications véhiculaires. Nous observons qu'aucune des approches n'arrive à satisfaire les différentes contraintes. Le seul mécanisme qui propose un bon compromis est l'approche CRLDP. C'est la raison pour laquelle nous avons choisi la CRLDP comme protocole de base de notre solution.

### 3.2.4 Modélisation du protocole de distribution

Nous décrivons dans ce chapitre notre solution, l'architecture utilisée et les principales entités impliquées dans notre approche.

Nous trouvons dans l'architecture de la VPKI (figure 3.11), une autorité de confiance racine appelée Root CA (RCA) qui assure certification des stations ITS et des autorités subordonnées (AA) dans une zone géographique prédéfinie, comme une ville,

| | Mobilité | Bande passante | Stockage | P. de calcul |
|---|---|---|---|---|
| CRL.dist.point | ✓ | ✓<br>la répartition<br>de la CRL sur les | $1sig + 1hsh$<br><br>✓ | |
| CRT | ✘ | Dépends du débit<br>de révocation | ✓ | $1sig + (H+1)hsh$ |
| OCSP | ✘ | ✓ | ✓ | $1sig + 1hsh$ |
| OCSP Stapling | ✘ | ✓ | ✓ | $1sig + 1hsh$ |
| SCVD | Dépend de la position<br>du serveur vis-à-vis<br>de la station | - | - | - |

TABLE 3.9 – Analyse des protocoles de distribution pour un système C-ITS

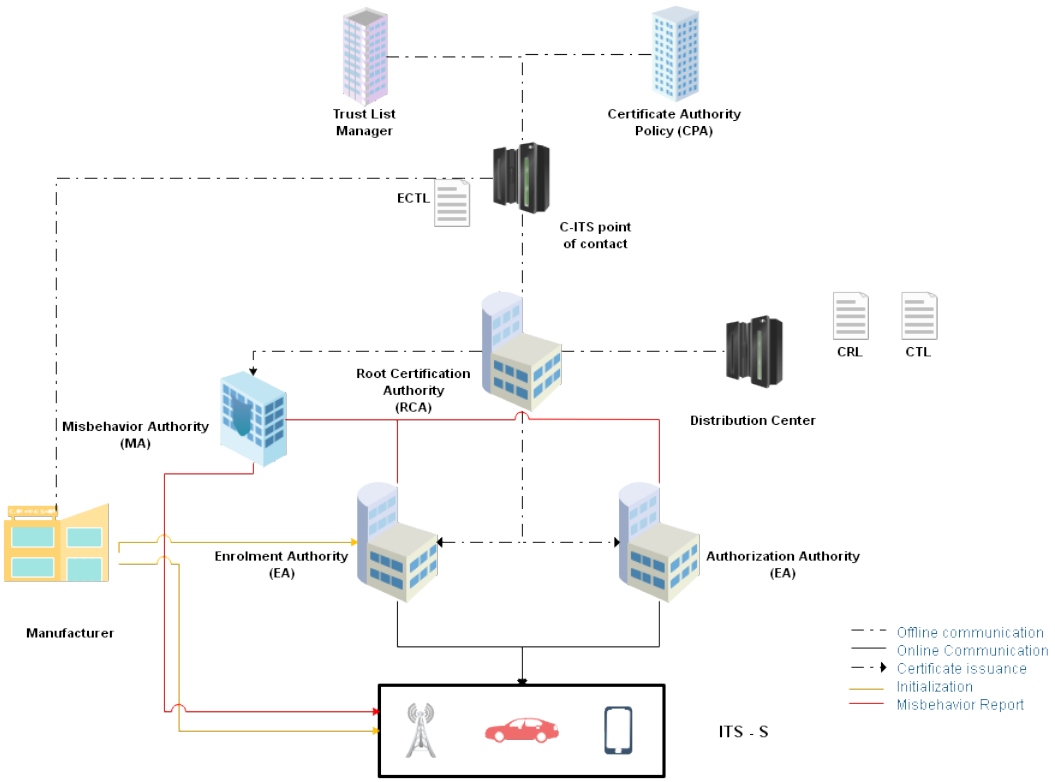



FIGURE 3.11 – Structure VPKI [Nev+17]

une région, un pays, etc. en enregistrant les différentes stations et en leur attribuant des certificats. Le RCA peut révoquer un véhicule et donc tous ces certificats, en inscrivant le certificat à long terme du véhicule dans la liste de révocation (CTL). Nous avons des autorités d'autorisation (AA) subordonnées au Root CA qui sont chargées d'enregistrer, de délivrer et de révoquer les certificats pseudonymes (certificats de courte durée) des stations ITS et les listes de révocation relatives aux certificats pseudonymes. Nous supposons que le RCA et les AAs sont fiables et ne peuvent être

compromis par de potentiels attaquants. Les RSUs sont des nœuds de l'infrastructure routière uniformément déployés le long des routes, chaque RSU est connecté aux AAs via un réseau câblé. Nous supposons que les RSUs soient bien déployés sur toutes les routes prises en considération. La connectivité véhicule-infrastructure routière est réalisée via des interfaces de communications dédiées à courte portée (DSRC) telles que ETSI ITS-G5. Les stations ITS signent et diffusent périodiquement des messages liés à la sécurité comme les messages de sensibilisation coopérative (CAM) et les messages de notification environnementale décentralisée (DENM), en y joignant leurs certificats. Nous supposons également que les différentes stations sont équipées d'un module de sécurité matérielle (HSM) inviolable stockant le matériel cryptographique.

#### 3.2.4.1 L'organisation des certificats

Afin de communiquer en toute sécurité, chaque véhicule doit se munir de deux types de certificats, un certificat à long terme et un certificat pseudonyme. Les certificats sont préinstallés dans les véhicules et mis à jour périodiquement à chaque contrôle technique ou révision. Le renouvellement des certificats pseudonymes peut se faire selon deux approches. Dans le premier cas, le véhicule se voit assigner un certificat à la fois. Chaque certificat ne sera rénouvelé qu'après expiration. Cette approche nécessite que le véhicule soit très souvent dans le rayon de communication du point de distribution des certificats situé à l'intérieur du RSU. Selon la deuxième approche le véhicule se voit assigner un groupe de certificats à la fois qui seront actualisés périodiquement groupe par groupe. Notre solution a été conçu pour prendre en compte seulement le deuxième cas. Les messages reçus par chaque véhicule, quelle que soit leurs provenance, doivent être authentifiés par le certificat de l'émetteur qu'ils contiennent. Dans le système C-ITS, l'anonymat des véhicules est de mise, c'est pourquoi il faut trouver le bon compromis entre la protection de la vie privée et la sécurité des véhicules. Les différentes stations doivent conserver autant possible leur anonymat pour éviter d'être tracées. L'utilisation d'un seul certificat pseudonyme peut protéger l'identité de la station mais ne peut protéger contre la traçabilité. Effectivement, chaque message envoyé, contient le certificat de l'expéditeur, ce qu'un attaquant peut facilement exploiter juste en écoutant les différentes communications pour tracer tous les messages envoyés par une victime et ainsi découvrir ses lieux d'intérêt, par exemple son domicile et son lieu de travail.

#### 3.2.4.2 Le groupe de Certificats

Dans le but de protéger chaque véhicule contre la traçabilité de ses actions, le AA délivre un groupe de certificats pseudonymes au véhicule dans lequel les différents certificats ne sont pas liés les uns aux autres. Cependant, l'ensemble des certificats pseudonymes est lié au certificat de longue durée qui identifie le véhicule. Il doit être difficile, voire impossible pour un attaquant de retrouver le certificat de longue durée à partir d'un certificat pseudonyme. Les véhicules doivent changer fréquemment le certificat pseudonyme afin d'être difficiles à tracer voir impossible. Chaque station est libre de choisir sa propre politique sur la fréquence de changement des certificats, par exemple de changer de certificat chaque jour, chaque heure ou à chaque message envoyé. Le nombre de certificats de courte durée à émettre pour chaque véhicule et la durée de vie du certificat dépend de la politique interne du AA ou du standard choisi. Néanmoins, certains paramètres doivent être pris en compte, comme le stockage interne du véhicule, l'utilisation de la bande passante, la

topologie du réseau, etc. Une enquête effectuée par l'association des constructeurs européens (ACEA) **??** montre que les Français font en moyenne deux trajets et demi par jour pour des raisons professionnelles ou privées. Le temps de conduite journalière se situe en moyenne entre 1 heure et 1h30. À partir des données fournies par cette enquête nous avons choisi de prendre 1 heure comme durée de vie de référence du certificat pseudonyme. Cette durée de vie selon nous va rendre difficile la traçabilité des véhicules. La normative Européenne impose un contrôle technique sur les véhicules neufs tous les 4 ans, et tous les 2 ans pour les véhicules de plus de quatre ans.

#### 3.2.4.3 Agrégation des certificats pseudonymes

Nous avons précédemment dit qu'un véhicule devait obligatoirement avoir et utiliser plusieurs certificats pour éviter d'être tracé par des utilisateurs malveillants. Ceci peut se résumer à un groupe de dizaines de milliers de certificats par véhicule, si nous partons sur une période de quatre ans. Révoquer un véhicule veut dire révoquer tous ces certificats, c'est-à-dire inscrire une dizaine de milliers de certificats par véhicule dans la liste de révocation. Selon nos estimations, ce fait pourrait provoquer une croissance exponentielle des listes de révocation. Nous avons l'objectif de proposer une méthode pour réduire la quantité de données à insérer dans les listes de révocation par véhicule révoqué. Nous croyons que cela va nous permettre de maitriser la croissance des listes de révocation. Nous comptons atteindre notre objectif grâce aux fonctions cryptographiques que nous décrivons dans ce chapitre.

Supposons qu'un véhicule V possède $c_{tot}$ certificats pseudonymes, supposons également que les certificats ne sont pas liés entre eux et qu'il est très difficile pour un attaquant de découvrir l'identité d'un véhicule à partir d'un certificat pseudonyme. Tous les certificats sont stockés localement en toute sécurité par les véhicules. Nous introduisons la clé symétrique de chiffrement/déchiffrement $S_k$ propre à chaque véhicule et partagée uniquement avec le AA. Chaque véhicule partage une $S_k$ avec le AA. La $S_k$ est utilisée pour produire un nouveau champ dans le certificat pseudonyme, le champ *verification-text* $(v_t)$. Le *verification-text* dévient le certificat ID (certID) afin de maintenir la compatibilité avec le standard comme décrit par l'équation 3.1.

$$size(v_t) + size(certID) < MaxSize(CertID) \tag{3.1}$$

La $S_k$ est une clé symétrique, elle est générée par le AA pour chaque véhicule. C'est une clé de chiffrement/déchiffrement qui accepte des données de longueur fixe en entrée et les permute en une sortie statique, c'est-à-dire qui ne change que si l'entrée change, la même entrée doit toujours produire la même sortie. Le AA génère aussi une suite d'entiers *(v, .., n)*, la plage de valeurs *(v, .., n)* peut être composée de valeurs aléatoires ou d'une suite de valeurs contiguë. $v$ et $n$ ne doivent pas nécessairement être de grands entiers, car la taille de l'entier n'influence pas la sécurité de notre système. En effet la sécurité du système dépend de l'algorithme de chiffrement et non des données à chiffrer. $v$ et $n$ sont connus de tous et sont les mêmes pour tous les certificats quelque soitt le véhicule. La valeur de $k$ $\{v - n = k\}$ doit être égale au nombre de pseudonymes délivrés au véhicule. Chaque entier $i$ dans la plage $(v...n)$ est lié à un seul certificat et n'est pas mentionné dans le certificat. Le *verification-text* est le cryptogramme obtenu chiffrant $i$ avec la clé $sk$. Pour chaque $i$, nous pouvons résumer le certificat comme décrit par l'équation 3.2 :

$$Cert_i = K_{pub}, E_{sk}, sig_{AA} \tag{3.2}$$

Où $K_{pub}$ est la clé publique du véhicule, le $E_{sk}$ est le verification-text et la $sig_{AA}$ est la signature du AA calculé sur l'ensemble du certificat. Toutes ces informations sont générées par le AA $k$ fois pour un véhicule. En d'autres termes la clé $S_k$ résume tous les certificats du véhicule. En effet, nous exploitons ce fait afin d'optimiser notre proposition et de réduire la taille des listes de révocation. La vérification de la révocation d'un certificat sous-entend selon notre approche le déchiffrement de plusieurs cryptogrammes. Nous définissons $v_{tot}$ le nombre de véhicules liés à une section et $rev_{tot}$ la quantité de véhicules révoqués présents dans une section. Révoquer un véhicule signifie écrire le $S_k$ du véhicule dans une section du fragment CRL. Le nombre de certificats liés à une section $cert_{section}$ peut être exprimé comme suit :

$$cert_{section} = v_{tot} * c_{tot} \tag{3.3}$$

Le AA doit effectuer plusieurs études afin d'attribuer à chaque section d'un fragment de CRL la quantité de véhicules qui minimise la valeur de $rev_{tot}$. Le pire scénario se produit lorsque $rev_{tot} = v_{tot}$ dans une section, car la vérification de la révocation des certificats liés à cette section nécessitera $rev_{tot}$ déchiffrements. Même si le certificat n'est pas révoqué. Nous avons le meilleur scénario quand $rev_{tot} = 0$, ce qui signifie qu'aucun déchiffrement n'est necessaire pour vérifier l'état de révocation du certificat. En d'autres mots, la section est vide, il n'y a aucun véhicule révoqué. L'approche idéale est d'avoir une répartition optimale des véhicules révoquées entre les différentes sections. Comme nous pouvons facilement en déduire, les deux scénarios décrits sont statistiquement improbables car il n'est pas possible de prédire la révocation d'un véhicule. La meilleure solution est d'affecter progressivement les véhicules aux sections, en fonction de l'évolution de la dimension des fragments.

Pour vérifier la révocation d'un certificat, le véhicule doit récuperer la section du fragment auquel le véhicule est attaché. Ensuite, le véhicule va utiliser toutes les clés symmetriques présente dans la section pour essayer de déchiffrer le cryptogramme du cetificat (voir la figure 3.12). Le certificat est révoqué, s'il existe une clé dans la section capable de déchiffrer son cryptogramme.

### 3.2.5 L'approche ICRLDP

Notre proposition est une méthode de révocation basée sur une nouvelle approche que nous appellons ICRDLP. Notre implique l'emission periodique de listes de révocation decomposés en plusieurs fragments. Chaque fragment est lié à un référentiel publique. La méthode ICRLDP fragmente l'entière les listes de révocation en plusieurs morceaux de telle sorte que chaque fragment devienne une plus petite liste de révocation. Chaque fragment est défini comme un point de distribution. Chaque station ITS est rattachée à un point de distribution, l'identifiant du point de distribution est mentionné dans les certificats pseudonymes. La taille de chaque fragment et le nombre de stations rattachées à un point de distribution sont définis par la politique interne du AA. Tous les certificats pseudonymes de chaque véhicule sont liés à un point de distribution unique. Le ICRLDP est l'adaptation pour la VPKI de l'approche CRLDP utilisé dans la PKIX. La structure de la CRL selon le système ICRLDP est décrite dans la figure 3.12.

Nous avons fait une estimation statistique de ce qui pourrait être la dimension des listes de révocation d'une CA afin d'éclaircir la problématique de montrer non seulement la nécessité et l'importance d'une méthode de révocation approprié. Nos estimations sont basées sur des données statistiques publiées par l'Union européenne [Eur19a] et le gouvernement français [Pas+18]. Le système ICRLDP est

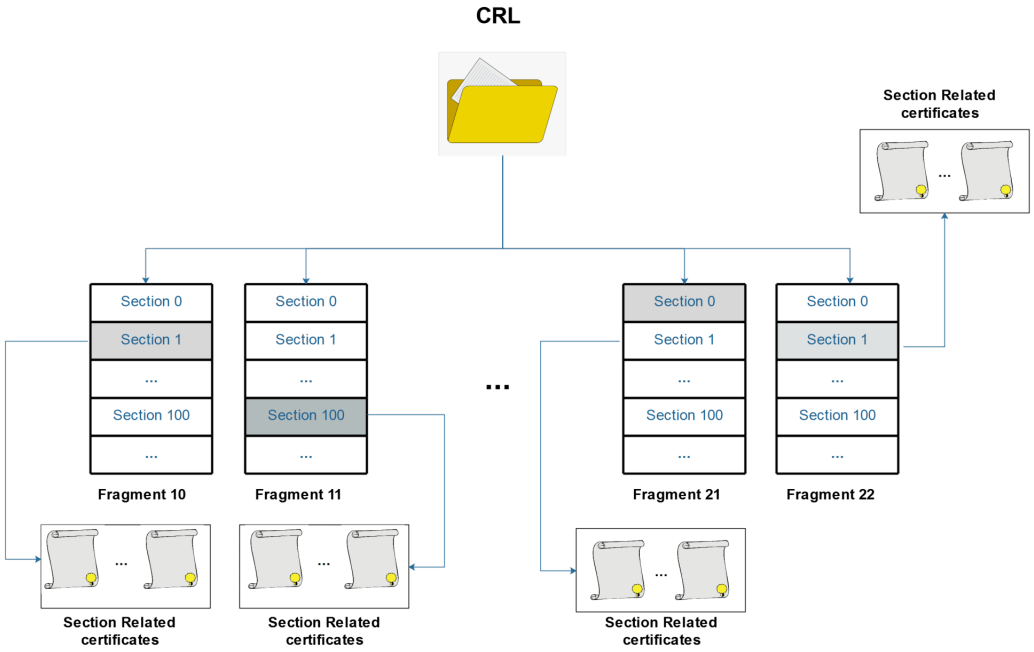



FIGURE 3.12 – Structure de la liste de révocation selon le protocole ICRLDP

caractérisé par certains paramètres : (1) le paramètre $F_d$ est la dimension d'un fragment exprimée en quantité de $S_k$ contenue dans le fragment, (2) le paramètre $F_n$ est la quantité de fragments composant la CRL et (3) Le paramètre $C_f$ est la fréquence de renouvellement des certificats. Le bon cocktail de ces paramètres va nous permettre de à trouver le bon compromis pour rendre notre système résilient. Comme précédemment décrit, le vol de voiture est l'une des possibles raisons qui peuvent provoquer la révocation d'une station ITS. Étant donné que des données réelles sur toutes les causes de révocation ne sont pas encore disponibles, pour cette étude, nous réduisons les causes de révocation aux seuls vols de véhicules. Nous avons choisi la France comme aire géographique d'études, parce que les données statistiques utilisées proviennent d'études menées par le gouvernement français. Les statistiques sur les vols de véhicules sont publiées chaque année pour chaque pays européen par la Commission européenne [Eur19a][Pas+18]. En 2018, la police a enregistré une moyenne de *177300* vols de véhicules en France. Nous le modélisons comme $v$. Le nombre de véhicules en circulation est estimé à *33 020 000* [Eur19b] que nous modélisons comme $V_c$. Le pourcentage de révocation des véhicules en circulation est donc d'environ *0,43%* pour l'année 2018. Nous définissons comme $\alpha$ la quantité des véhicules révoqués dont les causes sont outre que le vol. Ainsi, la taille de la liste de révocation estimée sur une année est :

$$size(CRL) \sim [(v * n_v) + \alpha] * size(CertID) \tag{3.4}$$

Nous avons fait une estimation numérique de la dimension de la CRL prenant une heure comme durée de vie pour chaque certificat pseudonyme. Nous affectons 4 ans à la variable $C_f$, et une dimension max pour le CertID de 16 octets. La taille de la CRL est estimée à $(177300 * 35040 * 16)/1024 = 97.071.750 KBytes$, en d'autres termes, 97 GBytes. Cette valeur est trop élevée, donc inacceptable pour éviter de forte latence de téléchargement des listes de révocation. Surtout pour des stations ITS a forte mobile et à contrainte de stockage. Cette dimension pourrait doubler chaque année, car elle dépend également du paramètre $C_f$. Nous croyons que la fragmentation de la liste de révocation en petits morceaux indépendants pourrait résoudre la problématique de la taille de la liste de révocation, la latence de téléchargement et l'utilisation de la bande passante. Dans cet exemple, la liste de révocation pourrait être divisée en 500 fragments. Cela produirait des fragments d'environ 19 Mo. Cette quantité de fragments est réaliste, c'est la quantité réelle de points de distribution dans le réseau internet selon l'article de conférence [Liu+15a] relatif aux informations récoltées en 2015. Nous résumons ces données dans le tableau 4.2.

| CA | Unique CRLs | Certificates | | Avg. CRL size (KB) | Rate % |
|---|---|---|---|---|---|
| | | Total | Revoked | | |
| GoDaddy | 322 | 1,050,014 | 277,500 | 1,184 | 26 |
| RapidSSL | 5 | 626,774 | 2,153 | 34,5 | 0.3 |
| Comodo | 30 | 447,506 | 7,169 | 517.6 | 1.6 |
| PositiveSSL | 3 | 415,075 | 8,177 | 441.3 | 1.97 |
| GeoTrust | 27 | 335,380 | 3,081 | 12,9 | 0.91 |
| VeriSign | 37 | 311,788 | 15,438 | 205.2 | 4.95 |
| Thawte | 32 | 278,563 | 4,446 | 25.4 | 1.59 |
| GlobalSign | 26 | 247,819 | 24,242 | 2,050.0 | 9.78 |
| StartCom | 17 | 236,776 | 1,752 | 240.5 | 0.73 |
| Total rate | | | | | 5.3 |

TABLE 3.10 – Statistiques des listes de révocation dans l'internet [Liu+15a]

L'approche ICRLDP n'est pas normalisée, nous devons donc apporter une légère modification à la structure standardisée du certificat pour permettre son éventuelle utilisation. La structure CRL ASN.1 décrite par la norme ETSI TS 102 941 V1.2.1 [ETS] est visible dans la figure 3.11.

| **Field** | **type** |
|---|---|
| Version | Integer |
| This update | Time32 |
| Next update | Time32 |
| Sequence of CRL Entry | HashedID8 [1] |

TABLE 3.11 – Structure ASN.1 de la CRL

Le type HashedID8 représente les 8 premiers octets de l'identifiant du certificat révoqué. Cette structure est légèrement modifiée permettant ainsi une rétro compatibilité. La nouvelle structure que nous proposons est visible dans la figure 3.12.

| Field | type |
|---|---|
| Version | INTEGER |
| CRLDP ID | OCTET STRING (SIZE(16)) |
| This update | Time32 |
| Next update | Time32 |
| Sequence of CRL Entry | OCTET STRING |

TABLE 3.12 – Structure ASN.1 de la CRLDP

Le type du champ Sequence of CRL Entry a été modifié en OCTET STRING, car ce champ contient maintenant des clés symétriques et non des identifiants de certificat. Comme il est possible d'observer sur la figure, nous avons ajouté un nouveau champ appelé ICRLDP ID, qui est l'identifiant unique du fragment. Il a une taille de $2^{16} - 1$ . Les bouts de CRL seront stockés que dans les RSUs. Nos points de distribution se trouvent maintenant dans le RSU. Nous supposons que les RSUs sont fiables et ont suffisamment de capacité de stockage pour stocker tous les points de distribution.

#### 3.2.5.1 Le nouveau champ

Certaines modifications doivent être apportées à la structure du certificat pour activer l'utilisation de notre mécanisme de révocation. Ces modifications sont mineures pour faciliter l'intégration et permettre la rétro compatibilité. Nous avons introduit une nouvelle valeur pour le champ du certificat [ETS] *crlseries*, la valeur *u* pour indiquer que les listes de révocation sont distribuées via le mécanisme ICRLDP.

$$CrlSeries ::= Uint16$$

Aucune autre étape de vérification n'a été introduite, chaque station va devoir vérifier le champ *crlseries*, comme d'habitude, pour connaître le méthode de révocation à utiliser pour vérifier le certificat.

#### 3.2.5.2 Caractéristiques du fragment

L'organisation interne des fragments à été conçu dans le but de faciliter la recherche de clés. Chaque fragment est divisé en sections et à chaque section est rattaché un groupe de véhicules $S_veh$. Chaque fragment a une dimension $F_d$. Le AA à la tâche de définir la valeur de $C_f$ et $F_d$. Une fois définit ces valeurs il est possible de faire une estimation de $F_n$ comme suit :

$$F_n = \frac{V_q * C_f}{F_d} \tag{3.5}$$

Nous modélisons la dimension d'une section avec ($Sec_k$) comme le nombre maximal de ($S_k$) que la section peut contenir. Nous définissons également $Tot_{veh}$ comme la quantité totale de véhicules en circulation. Il est ainsi possible de déduire la quantité maximale de véhicules couverte par chaque fragment ($Ff_{cov}$) ainsi :

$$Ff_{cov} = \frac{Tot_{veh}}{F_n} \tag{3.6}$$

La quantité maximale de véhicules couverte par chaque section $Fs_{cov}$ :

$$Fs_{cov} = \frac{Ff_{cov} * Sec_k}{F_d} \tag{3.7}$$

Nous définissons $v_{tot}$ comme le nombre de véhicules liés à une section et $c_{tot}$ le nombre de certificats pseudonymes appartenant à chaque véhicule. Nous définissons également $rev_{tot}$ comme le nombre de véhicules révoqués par section. Le nombre de certificats liés à une section $cert_{section}$ est exprimé ainsi :

$$cert_{section} = v_{tot} * c_{tot} \tag{3.8}$$

Le certificate ID est maintenant l'hexadécimal du *verification-text*. Celui-ci est composé de deux parties : *ID du point de distribution* et *Numéro de section*.

$$CertificateID = \{ID_{distrib}, Sec_{Number}\}$$

. Chaque certificat est rattaché à une section d'un point de distribution.

Nous proposons une subdivision interne du fragment afin de réduire le nombre de clés $S_k$ à tester pour la vérification de la révocation. Grâce à cette optimisation, la vérification de la révocation du certificat implique de tester uniquement les clés de la section à laquelle est rattaché le certificat et non toutes les clés contenues dans le fragment. Cette subdivision en sections produit deux scénarios extrêmes. Le meilleur scénario est d'avoir un seul véhicule révoqué par section ($rev_{tot} = 0$), ainsi la vérification de la révocation n'impliquera aucun déchiffrement. Le deuxième scénario est le pire, qui se produit lorsque tous les véhicules d'une section sont révoqués ($rev_{tot} = v_{tot}$). C'est le scénario à absolument éviter. Les deux scénarios sont extrêmes et statistiquement improbables car il n'est pas possible de prédire la révocation des véhicules. Il est convenable pour le AA d'affecter progressivement chaque véhicule a la section en fonction de l'évolution de la taille des différentes sections.

Le graphe 3.13 nous décrit l'évolution dans le temps d'un de la dimension d'un fragment de CRL en fonction de la durée de vie du groupe de certificats. Nous pouvons observer que le fragment croît dans le pire des scénarios jusqu'à la date de renouvellement des certificats. À partir de ce moment tous les certificats expirent et peuvent donc être retirés de la CRL pour ceux qui sont révoqués. Ainsi le fragment recommence à croître en dimension jusqu'au futur renouvellement. Nous pouvons observer que la taille maximale du fragment est maitrisée, par contre la croissance ne l'est pas. Le graphe 3.13 décrit l'évolution dans le temps de la dimension du fragment. Nous pouvons observer sur l'axe des abscisses reflète la fréquence de renouvellement du groupe de certificats en années. L'axe des ordonnées décrit la taille de la CRL en nombre de clés révoquées (lignes). Les différentes courbes décrivent la fréquence de renouvellement des groupes de certificats fixée à 1, 2 et 4 ans.

Le graphe 3.14 met en relation la quantité des fragments $F_n$ sur l'axe des abscisses avec l'évolution de la dimension des fragments $F_d$ sur l'axe des ordonnées.

#### 3.2.5.3 Les contraintes

Une AA doit effectuer une étude approfondie des paramètres tels que la distribution géographique des véhicules, la zone d'autorisation, $S_veh$, $C_f$ et $F_d$ pour éviter la traçabilité des véhicules et optimiser notre approche de distribution. Neanmoins, la selection de ces derniers par la AA doit être fait au vu des contraintes suivantes :

— Un fragment ne peut avoir une dimension égale ou supérieure à celle des listes de révocation

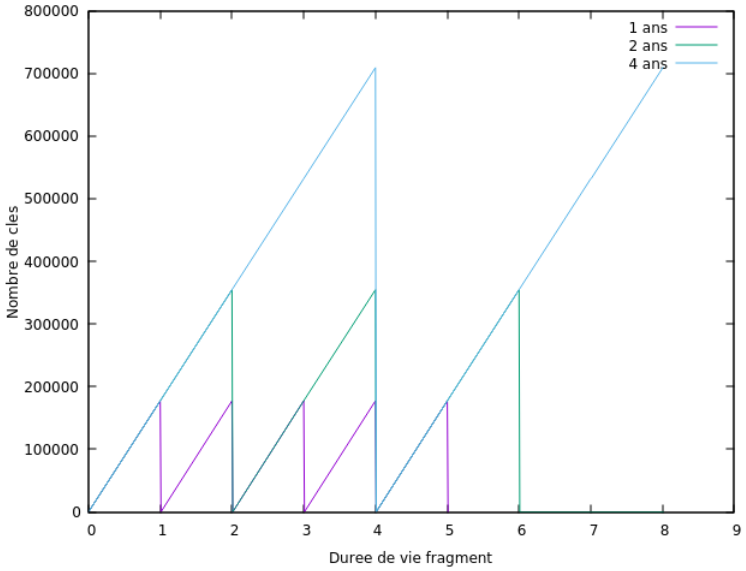



FIGURE 3.13 – Évolution dans le temps de la dimension du fragment

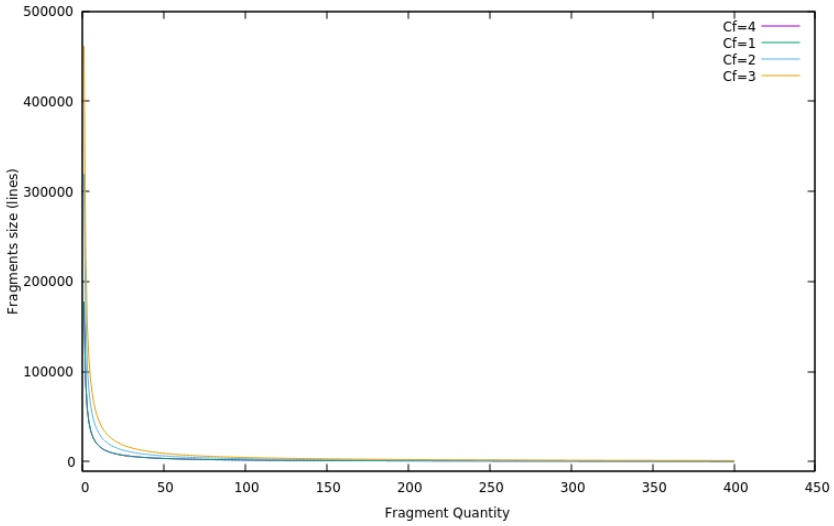



FIGURE 3.14 – Évolution de la dimension des fragments selon la quantité des fragments

— la fragmentation des listes de revocation doit être conforme à la distribution des véhicules sur la zone géographique d'intérêt
— chaque fragment doit être signé par le AA
— les fragments doivent être indépendants les uns des autres
— les fragments sont distribués par les RSUs
— les véhicules peuvent partager leurs propres fragments

#### 3.2.5.4 Dissémination des fragments

Tous les fragments, indifféremment de la AA émettrice sont stockés par des nœuds d'infrastructure tels que les RSUs. Tous les messages envoyés doivent contenir le certificat de l'émetteur tel que décrit dans le standard ETSI et maintenant un

fragment des listes de révocation aussi.

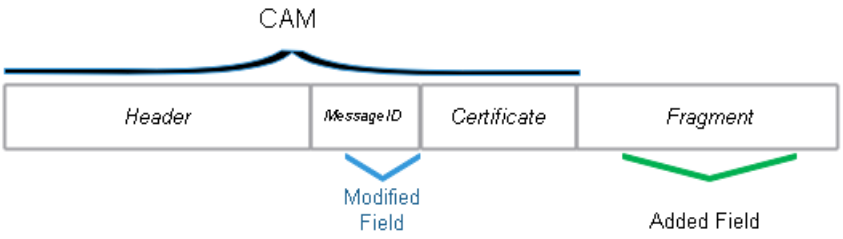



FIGURE 3.15 – Nouvelle structure du message CAM

Les véhicules doivent télécharger les mises à jour du fragment auquel ils sont rattachés lorsqu'ils se trouvent dans le rayon de communication des RSUs. Les différents véhicules peuvent également s'échanger des fragments à jour de manière sécurisée, sous la condition qu'ils soient rattachés au même point de distribution. Une station qui souhaite mettre à jour son fragment doit envoyer une requête ICRLDP. Ce n'est pas un nouveau type de message, mais un message CAM qui a un *messageID* particulier (Fig 3.15). Nous proposons donc une nouvelle valeur *p* pour le champ *messageID*. IL servira à identifier le message CAM qui contient une requête ICRLDP (ICRLDP Request). À la requête sera attaché un fragment valide mais expiré, car le véhicule tente de mettre à jour son fragment expiré. Ce message sera ignoré par toutes les stations qui ne sont pas rattachées au même point distribution avec l'expéditeur. La requête ne sera acceptée que par les RSUs ou les stations appartenant au même point de distribution qui possède l'information demandée. À ce point on distingue deux cas : (1) si le récepteur n'est pas un RSU, il répondra par un message CAM ordinaire. Ensuite, l'expéditeur récupérera le fragment joint au message. (2) si le récepteur est un RSU, ce dernier répondra avec un nouveau message CAM, avec le fragment demandé rattaché. Tous les véhicules qui appartiennent au même point de distribution et dans le rayon de communication du RSU recevront ce message et pourront automatiquement mettre à jour leurs propres fragments si nécessaires en évitant d'envoyer de nouvelles requêtes. Les autres véhicules ignoreront le message car il ne contient pas le fragment auquel ils sont rattachés.

L'algorithme 1 décrit le processus de réponse qui commence par la réception d'un ICRLDP request. La station réceptrice, récupère l'entête du fragment attaché au message dans le but de contrôler si elle possède le fragment et la bonne version du fragment demandé. Le contrôle s'effectue avec la fonction *meme_fragment(N->crldpid)*. Ensuite, si la station ne possède pas le bon fragment, le message de requête sera simplement ignoré. Dans le cas contraire, Il faudrait contrôler si la station réceptrice est un RSU. Ce contrôle est effectué avec la fonction *je_suis_RSU()*. Si la station est un RSU, celle-ci va créer un message CAM avec creer_CAM_Message(N->crldpid->fragment). Cette dernière va attacher le fragment demandé au message CAM puis l'envoyer, elle peut le faire car elle possède tous les fragments. Dans le cas où la station réceptrice n'est pas un RSU, un nouveau message CAM sera tout de même créé et envoyé avec send_CAM_Message(). Ce message CAM sera ordinaire, car il va contenir le fragment de l'expéditeur qui est en fait l'objet de la requête.

**Algorithme 1 :** Algorithme de réponse à une requête ICRLDP

**Input :** Ct= ICRLDP Request
**Output :** S=Bool ( True : J'ai le fragment , False : Je n'ai pas le fragment)

```
1 algorithme-reponse (Ct,S)
    1: N ← Ct− > fragment;
    2: if N ∉ ∅ then
           ;
           if meme_fragment(N− > crldpid) then
               ;
               if je_suis_RSU() then
                   ;
                   Vt = creer_CAM_Message(N− > crldpid− > fragment);
                   send_CAM_Message(Vt);
               end
               else
                   Vt = creer_CAM_Message(monFragment);
                 send_CAM_Message(Vt);
               end
           end
    3: else
           return False;
       end
        end
    4: else
           return Error;
        end
        return True;
```

Il est facile de constater que notre approche fonctionne également dans une zone où les RSUs ne sont pas bien répartis ou absents, parce que les stations peuvent s'échanger les fragments.

#### 3.2.5.5 Vérification de l'état de révocation

Le RCA procède d'abord par la révocation du certificat à long terme de la station, puis tous les certificats à court terme détenus sont invalidés par le AA. Le RCA peut révoquer un certificat pour de nombreuses raisons : mauvais comportement ou compromission du véhicule, perte de clés privées, vol du véhicule, etc. Afin de révoquer un véhicule, le AA publie dans les listes de révocation la clé symétrique $S_k$ qui peut décrypter tous les différents *verification-text* des certificats pseudonymes du véhicule révoqué. Il apparaîtra donc pour chaque certificat révoqué la ligne,

$$line_i = \{S_k\}$$

Où $S_k$ est la clé de chiffrement utilisée pour produire le champ *verification-text*. Lorsqu'un message est reçu, la station vérifie d'abord la validité du certificat joint. Ensuite, vérifie que le fragment attaché est le bon et ensuite l'état de révocation du certificat. Le véhicule retire la bonne section du fragment, ensuite essaie clé par clé a.k.a $S_k$ de déchiffrer le champs *verification-text* du certificat. Le certificat est révoqué s'il existe un $S_k$ dans la section du fragment capable déchiffrer son champ *verification-text*.

Les fragments sont signés par le AA, il sera donc toujours possible de les authentifiés. Chaque fragment est mis à jour périodiquement, il contient donc une date d'expiration. Il n'y a aucun moyen pour l'attaquant d'utiliser un fragment obsolète. Nous allons dans le prochain chapitre implémenter, tester et analyser nos deux solutions que nous avons décrit dans ce chapitre.

# Chapitre 4

# Validation et évaluation

Dans ce chapitre, nous allons décrire la mise en oeuvre de nos deux solution BL-CRLDP et ICRLDP.

## 4.1 BL-CRLDP

Nous allons détailler l'implémentation et évaluer notre proposition BL-CRLDP dans ce chapitre.

### 4.1.1 Implémentation

Un nœud CA a été conçu pour se connecter à un réseau Blockchain et émettre des transactions. Nous avons développé un client et un serveur capable de se connecter aussi au même réseau Blockchain et de récupérer des transactions de celle-ci. Le client et le serveur sont capables d'utiliser une connexion TLS pour communiquer entre eux. Pour finir, nous avons proposé une extension TLS pour permettre au client et au serveur de négocier dans le Handshake protocol l'utilisation de notre méthode de révocation. Tous les tests ont été effectués en local avec des nœuds interconnectés à partir de réseau différents. L'architecture de l'implémentation est décrite dans la figure 4.1.

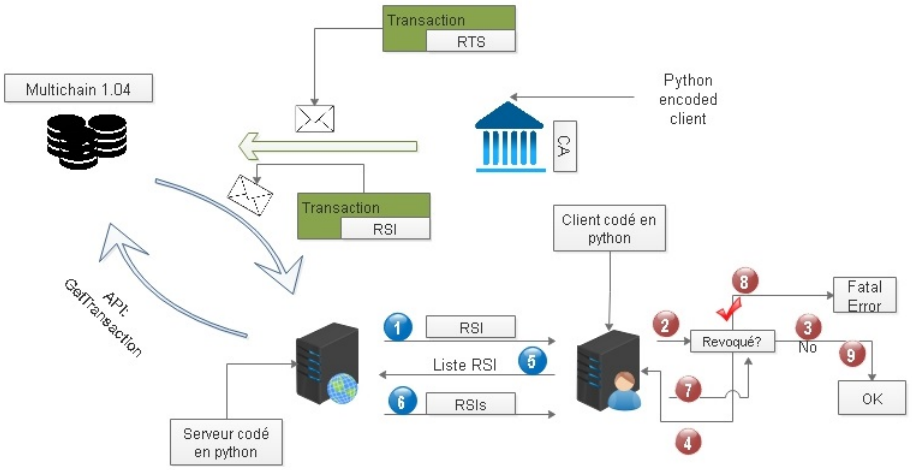



FIGURE 4.1 – Architecture de l'implémentation

### 4.1.2 Les caractéristiques techniques

Pour l'implémentation du protocol OCSP en local nous avons utilisé la librairie de python "OCSPresponder". Pour les tests nous avons reproduit en local la Blockchain de NAMECOIN [Nam]. Notre choix s'y est porté parce qu'elle est autant robuste que celle de BITCOIN et que les données y sont stockées dans la Blockchain en mode clé et valeur. Chaque utilisateur a la possibilité d'y stocker des clés allant jusqu'à 255 octets de dimension et des valeurs allant jusqu'à 520 octets de dimension. Notre choix c'est aussi porté sur Namecoin à cause de la valeur faible de la monnaie, qui était de 1,52 $ (07/09/2018), ce qui permet des transactions à faible cout. Le volume journalier de transactions est relativement faible ce qui facilite la recherche des données. Nous avons utilisé le l'outil Multichain [Mul] version 1.04 pour simuler la Blockchain de Namecoin en local. Le client, le serveur et la CA ont été codés en python version 2.7 [Pyt] avec la librairie mmh3. Pour finir, la partie crypto a été effectué avec OpenSSL 1.0.2g [Ope]. Les caractéristiques des machines utilisées sont décrites dans le tableau 4.1.

| Type de noeud | CPU | architecture | Puissance de calcul | Memoire Ram | OS |
|---|---|---|---|---|---|
| Server | core i7-3770, X86_64 | 64 bits | 3.4 GHz | 8 GB | Ubuntu 16.04 |
| CA | Core i5-5300u, x86_64 | 64 bits | 2.3GHz | 8 GB | Kali Linux 4.19 |
| Client | Core i5-5300u, x86_64 | 64 bits | 2.3GHz | 8 GB | Kali Linux 4.19 |

TABLE 4.1 – Les caractéristiques techniques des machines

### 4.1.3 Les caractéristiques techniques du filtre de bloom

Nous avons utilisé le filtre de bloom pour réduire et maitriser la dimension des fragments. Le filtre de bloom est caractérisé par 4 paramètres (voir annexe) : le nombre d'éléments dans le filtre ($n$), le nombre de fonctions d'hachage utilisé ($k$), la dimension du filtre ($m$), la probabilité de faux positifs $p$. Pour optimiser l'utilisation du filtre, il faut trouver le bon compromis entre ces quatre paramètres. Le paramètre $n$ représente pour nous le nombre de certificats révoqués, nous avons effectué plusieurs tests en le faisant varier pour étudier le comportement de notre système par rapport à la dimension des fragments. Nous avons choisi un filtre de $m$=350 octets pour pouvoir contenir un fragment entier dans les 520 octets de données autorisées dans une transaction. Nous avons ensuite choisi $k$=2, car c'est la valeur qui optimise la probabilité de faux positif comme nous pouvons observer dans la figure 4.2.

### 4.1.4 Le taux de révocation

L'analyse statistique de la PKIX publié par l'article [Liu+15b] montre que les certificats révoqués représentent une petite portion des certificats émis par une CA.

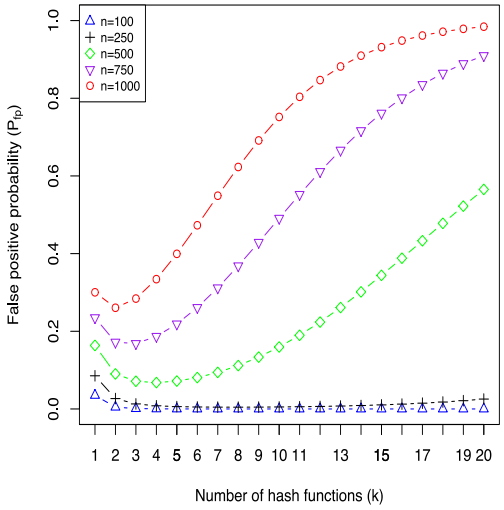



FIGURE 4.2 – Estimation de la valeur optimale de $k$

Cette portion estimé à 5.3% des certificats émis (voir tableau 4.2). Cette information est très importante car nous permet d'avoir des données réalistes sur lesquels fonder notre solution.

| CA | Unique CRLs | Certificates | | Moyenne CRL dimension (KB) | Taux % |
|---|---|---|---|---|---|
| | | Totale | Revoqué | | |
| GoDaddy | 322 | 1,050,014 | 277,500 | 1,184 | 26 |
| RapidSSL | 5 | 626,774 | 2,153 | 34,5 | 0.3 |
| Comodo | 30 | 447,506 | 7,169 | 517.6 | 1.6 |
| PositiveSSL | 3 | 415,075 | 8,177 | 441.3 | 1.97 |
| GeoTrust | 27 | 335,380 | 3,081 | 12,9 | 0.91 |
| VeriSign | 37 | 311,788 | 15,438 | 205.2 | 4.95 |
| Thawte | 32 | 278,563 | 4,446 | 25.4 | 1.59 |
| GlobalSign | 26 | 247,819 | 24,242 | 2,050.0 | 9.78 |
| StartCom | 17 | 236,776 | 1,752 | 240.5 | 0.73 |
| Taux | | | | | 5.3 |

TABLE 4.2 – Données statistiques de la révocation dans la PKIX [Liu+15b]

#### 4.1.4.1 Définition des paramètres

Les délais sont calculés du coté client, à partir de l'instant où la requête est envoyée par le client, jusqu'à la réception et à la fin du traitement de la réponse. $RTs$ = Délai de téléchargement d'une liste d'RSIs du serveur
$RTr$ = Délai d'envoie d'une requête de (RSI ou liste RSI) au serveur.
$RTl$ = Délao de téléchargement d'une liste d'RSIs
$Rt1$ = Durée de traitement d'une RSI par l'application cliente

$Rt2 =$ Durée de traitement d'une liste de RSIs par l'application cliente

$$Delai = RTr + RTs + Rt1 \tag{4.1}$$

$$Delai = RTr + RTl + Rt2 \tag{4.2}$$

#### 4.1.4.2 Tests Client⇒serveur

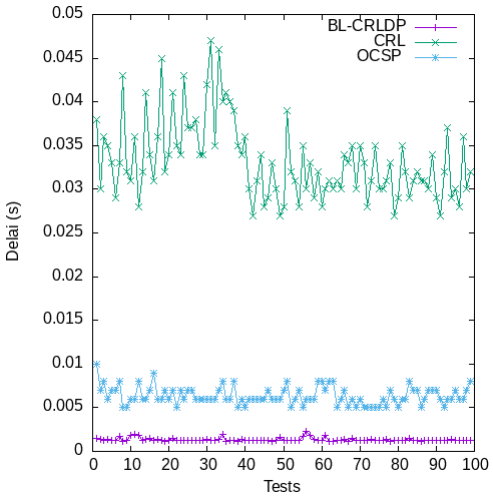
Dans cette section, nous décrivons et testons une mise en oeuvre de notre méthode de révocation. Nous avons effectué plusieurs tests divisés en deux parties. Dans la première partie nous avons implémenté et testé notre méthode prenant en compte que des certificats non révoqués. Dans la deuxième partie nous avons plutôt utilisé des certificats révoqués. Dans chaque partie l'utilisation des filtres de bloom nous confronte à deux situations qui doivent être prises en compte pour estimer les délais. Nous avons étudié chaque cas avec une série de tests et analysé les résultats.

**4.1.4.2.1 Première partie : Le certificat à contrôler n'est pas révoqué** La première partie et celle des certificats non révoqués, ce cas représente environ 94% des cas. Effectivement moins de 5% des certificats émis sont révoqués. Nous avons commencé par tester notre méthode dans le cas ou le certificat du serveur n'est pas révoqué. L'utilisation du filtre de bloom pour vérifier la révocation d'un certificat nous confronte à deux situations : le cas où le certificat n'est pas révoqué et le filtre nous retourne une réponse négative et le cas où le certificat n'est pas révoqué et le filtre nous retourne un réponse positive. Le filtre de bloom n'admet pas de faux négatifs, donc une réponse négative peut-être accepté sans ultérieures contrôle. Ainsi le délai est calculé du coté client de l'instant où la requête est envoyée par le client jusqu'à la réception et au traitement de la réponse. Nous avons fait plusieurs tests faisant varier la valeur de $n$=20,100, 250,500,750,1000. Ce qui équivaut à un fragment contenant $n$ certificats révoqués. Toutes les séries de tests ont été effectués en parallèle avec les méthodes CRL et OCSP afin de les comparer. Nous pouvons l'observer dans le graphe 4.4.

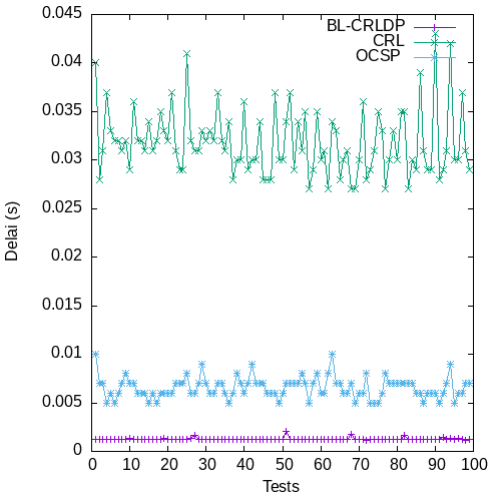
Le graphe 4.5 résume les différents tests effectués en les comparant aux résultats obtenus avec les approches CRL et OCSP. Il est possible d'observer que dans le cas où le certificat à contrôler n'est pas révoqué notre approche est la meilleure en terme de délais. Ce cas représente 95 % des cas réels dans l'internet.

Nous avons ensuite testé le cas où le certificat n'est pas révoqué et le filtre nous retourne une réponse positive, le filtre nous dit que le certificat est révoqué. Nous sommes dans le cas d'un faux positif, il faudrait télécharger tous les LRSIs du serveur et les contrôler. Mais ce cas s'inscrit dans les cas étudiés dans la deuxième partie.

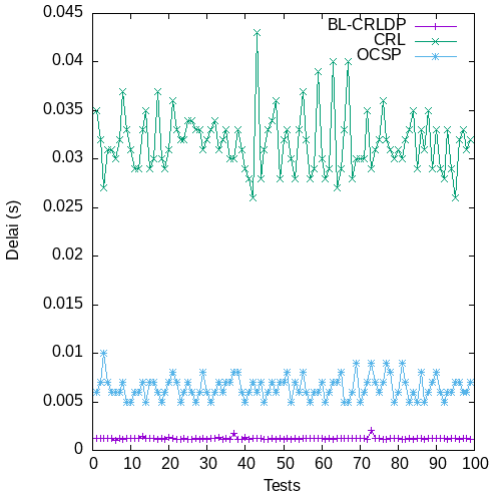
**4.1.4.2.2 Deuxième partie : Le certificat à contrôler est révoqué** Nous avons effectué des tests avec un certificat révoqué. Ce cas représente que 5% des cas réels. Nous avons testé notre méthode dans ce cas précis. Nous sommes ainsi confrontés à deux cas : le cas où le certificat est révoqué et le filtre nous retourne une réponse positive et le cas où le certificat est révoqué et le filtre nous retourne une réponse négative. Dans le premier cas, le délai est calculé du coté client de l'instant où la requête est envoyée par le client jusqu'à la réception et au traitement de la réponse. Nous avons effectué plusieurs tests faisant varier la valeur de $n$=20,100, 250,500,750,



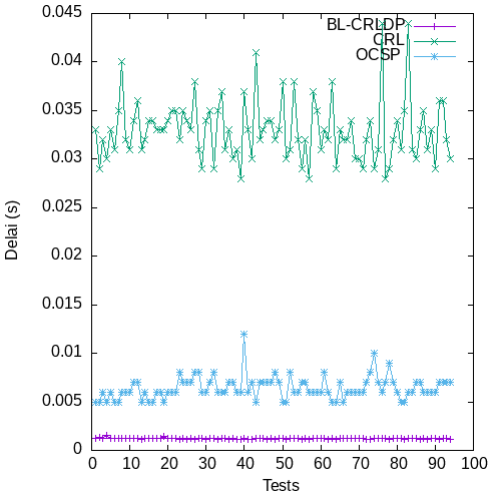

(A) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 20 certificats révoqués



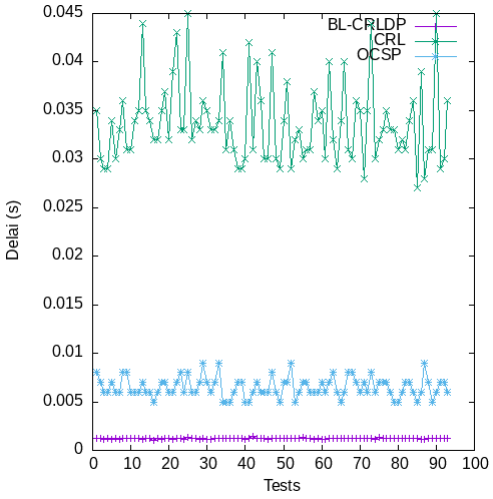

(B) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 100 certificats révoqués



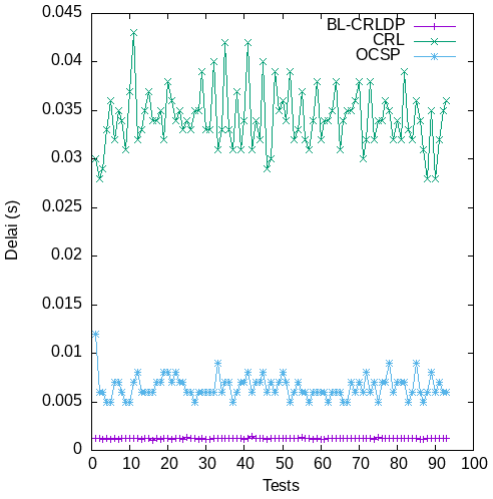

(C) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 250 certificats révoqués




(D) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 500 certificats révoqués




(E) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 750 certificats révoqués




(F) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 1000 certificats révoqués

FIGURE 4.4 – Résultats et comparaison dans le cas des certificats non révoqués

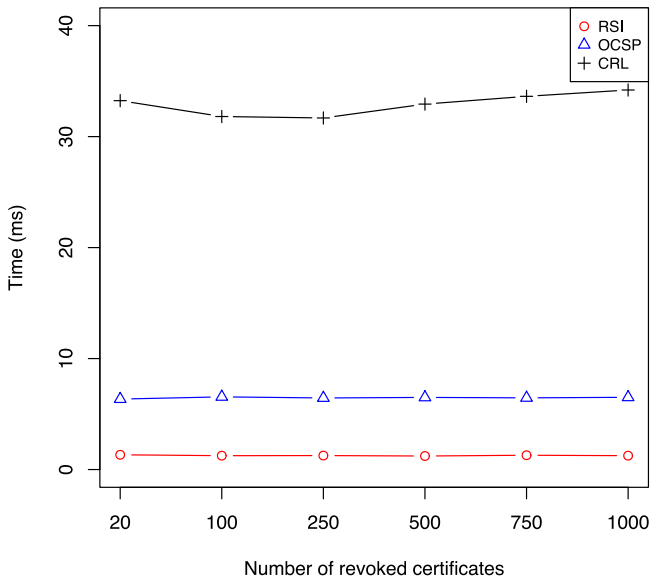



FIGURE 4.5 – Cas des certificats non révoqués

1000 du filtre. Toutes les séries de tests ont été effectués en parallèle avec le protocole de distribution CRL et OCSP de sorte à les comparer. Nous pouvons l'observer dans les graphes 4.7.

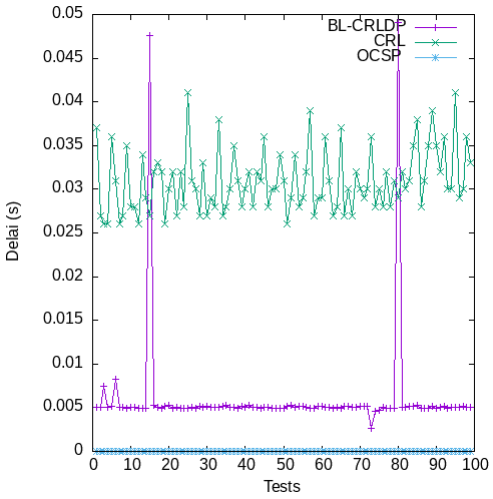
Le graphe 4.8 résume les différents tests effectués. Nous avons les avons comparé aux résultats obtenus avec les approches CRL et OCSP. Il est possible d'observer que dans le cas où le certificat à contrôler n'est pas révoqué notre approche est optimale en terme de délais. Ce cas représente 95 % des cas réels dans l'internet.

Ensuite, nous avons testé le cas où le certificat est révoqué et le filtre nous retourne une réponse négative. Ce cas ne se présente jamais car le filtre de bloom n'admet pas de faux négatifs.

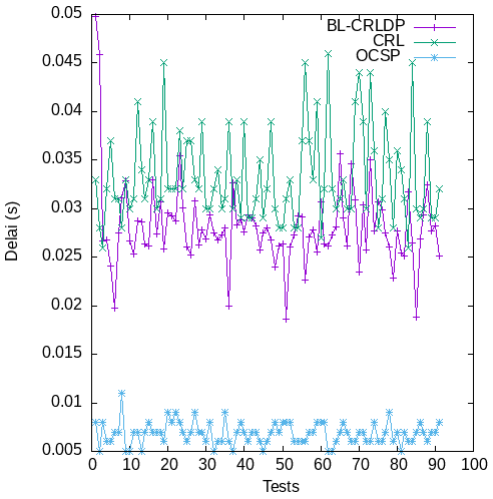
Comme nous pouvons observer par les résultats, le nombre de certificats révoqués est directement proportionnel au temps de vérification et la probabilité de faux positifs. Les résultats nous montrent aussi que pour 1000 certificats révoqués par filtre, nous avons un taux de faux positifs de plus de 28% ce qui n'est pas admissible. Un seul faux positif peut entrainer un retard de 0.1 s, trop élevé pour un protocole comme TLS.

| RSI 100 certificats | | | | | | |
|---|---|---|---|---|---|---|
| Nombre de transactions necessaires | Nombre d'échanges TLS sup. | Volume données échangés (octets) | Filtre | | | Temps moyen(s) |
| | | | Dimension | Nombre Hash | Prob.faux | |
| 1 | 0 | 100*173+499 | 350 | 2 | 0.00006 | 0.028 |
| RSI 250 certificats | | | | | | |
| 1 | 2 | 250*173+499 | 350 | 2 | 0.07 | 0.063 |
| RSI 500 certificats | | | | | | |
| 1 | 2 | 500*173+499 | 350 | 2 | 0.07 | 0.063 |
| RSI 750 certificats | | | | | | |
| 1 | 2 | 750*173+499 | 350 | 2 | 0.19 | 0.081 |
| RSI 1000 certificats | | | | | | |
| 1 | 2 | 1000*173+499 | 350 | 2 | 0.28 | 0.1 |

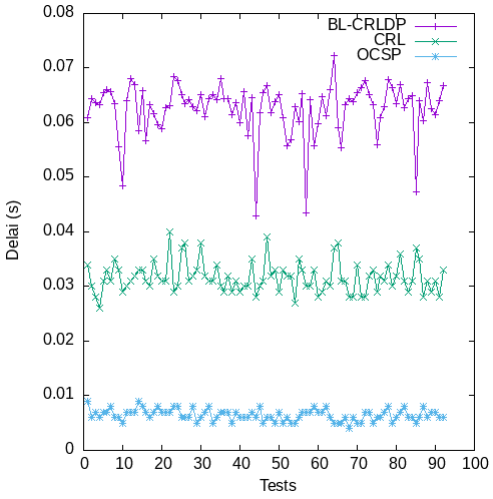
TABLE 4.3 – Données des tests dans le cas des certificats révoqués



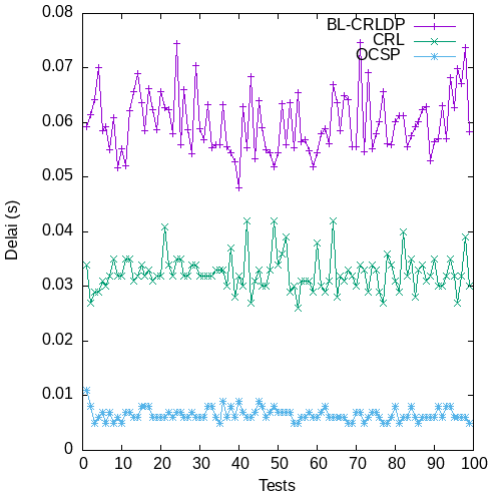

(A) Comparaison des délais de réponses dans le cas d'un certificat révoqué et un fragment contenant 20 certificats révoqués



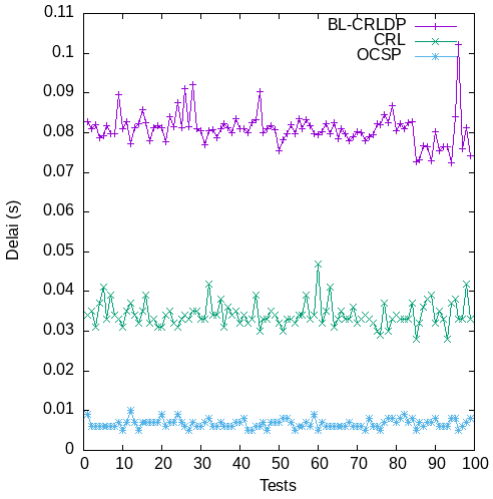

(B) Comparaison des délais de réponses dans le cas d'un certificat révoqué et un fragment contenant 100 certificats révoqués



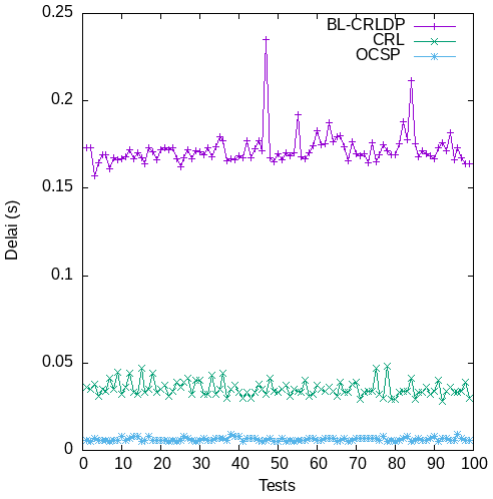

(C) Comparaison des délais de réponses dans le cas d'un certificat révoqué et un fragment contenant 250 certificats révoqués




(D) Comparaison des délais de réponses dans le cas d'un certificat révoqué et un fragment contenant 500 certificats révoqués




(E) Comparaison des délais de réponses dans le cas d'un certificat révoqué et un fragment contenant 750 certificats révoqués




(F) Comparaison des délais de réponses dans le cas d'un certificat non révoqué et un fragment contenant 1000 certificats révoqués

FIGURE 4.7 – Résultats et comparaison dans le cas des certificats non révoqués

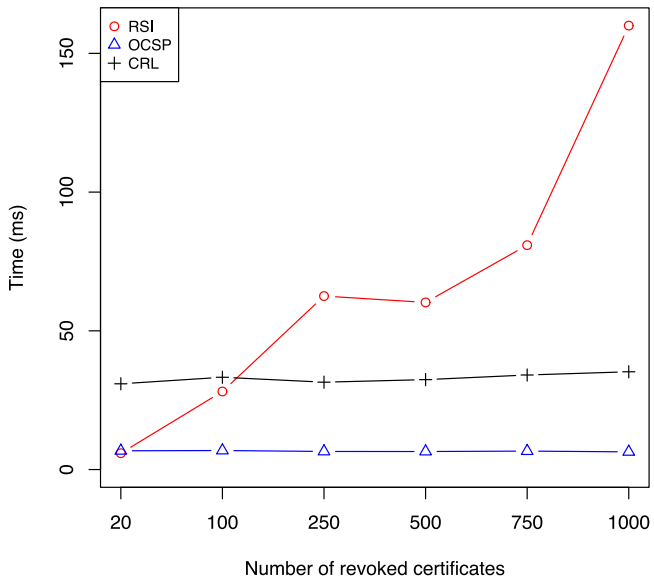



FIGURE 4.8 – Cas des certificats révoqués

Nous avons ainsi plafonné la taille du fragment à 750 certificats révoqués, car au-delà, la probabilité de faux positifs n'est plus acceptable. Nous pouvons le constater dans le tableau 4.3. Nous croyons, que la quantité de 750 ertificats est réaliste, car notre filtre pourrait couvrir un maximum de 15.000 certificats. Selon les chiffres publiés par la contribution [Liu+15a], Verisign possède 39 CRLs pour environ 311,788 certificats, qui veut dire environ 8000 certificats par CRL.

#### 4.1.4.3 Tests Blockchain⇒serveur

Nous allons maintenant décrire le protocole d'échange entre la Blockchain et le serveur. Nous avons déployé une Blockchain en local avec Multichain et développé en python un serveur capable de s'y connecter. Notre implémentation visait à estimer le temps nécessaire à un serveur pour télécharger son fragment de la Blockchain. Pour se faire et pour avoir des résultats réalistes, nous avons produit une Blockchain de environ 500.000 blocks. Nous avons éparpillé un peu partout dans la Blockchain des fragments à retrouver. La Blockchain de Namecoin fait environ 400000 blocs en date 02/2019. Nous avons testé seulement le mode sollicité du protocole, car le faire pour le mode non sollicité n'aurait pas été nécessaire. Les délais incluent le temps nécessaire pour retrouver une transaction et la traiter.

Le graphe 4.9 décrit le temps nécessaire pour récupérer une transaction à différentes hauteurs de la Blockchain, partant d'une Blockchain de 500.000 blocs. Nos tests nous ont permis d'estimer le temps moyen nécessaire pour traiter un bloc. Ce temps est estimé à 0,001s avec un écart type de 65.567. La recherche dans la Blockchain commence toujours par le bloc le plus récent, c'est pourquoi à la hauteur 500.000 les délais sont relativement faibles puis croissent progressivement en remontant la Blockchain. Le graphe nous montre que les délais peuvent devenir considérables, c'est tout à fait normal parce qu'il n'y a pas d'organisation logique de données comme dans la base de données. Il faut procéder bloc par bloc puis transaction par

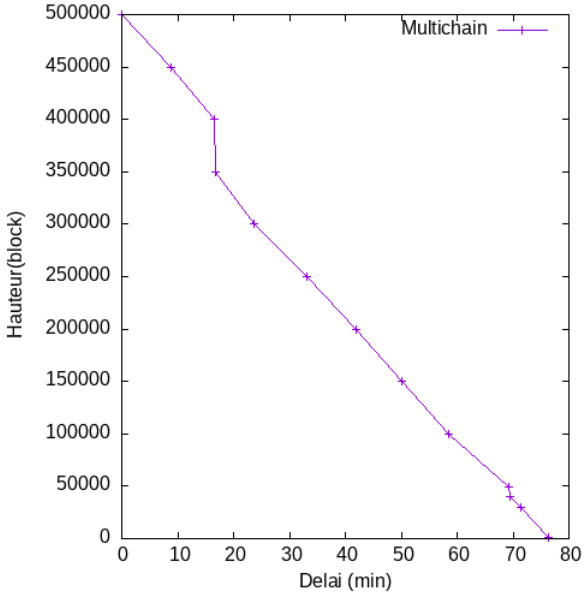



FIGURE 4.9 – Temps nécessaire pour récupérer et traiter une transaction selon la position dans la Blockchain

transaction jusqu'à retrouver la donnée recherchée. Cet échange est effectué périodiquement et en amont de toutes communications client / serveur. Notre protocole d'échange n'est donc pas impacté par la latence de recherche des informations.

### 4.1.5 Implémentation de notre nouvelle extension TLS

L'outil OpenSSL est une implémentation open source du protocole TLS. Originairement basé sur les librairies de Eric Young e Tim Hudson, OpenSSL est disponible pour la plus grande part des systèmes d'exploitation Linux-like, inclus GNU/Linux e macOS, e aussi pour Microsoft Windows. Nous avons utilisé la version 1.0.2g du 01 mars 2016. Nous avons modifié le Handshake protocol de TLS, précisément les deux messages HELLO pour y insérer une extension qui va permettre de négocier l'utilisation de notre solution.

#### 4.1.5.1 Les répertoires et fichiers

Dans le but d'intégrer notre extension nous avons modifié les fichiers suivants :

— **APPS :** le dossier contient les applications d'OpenSSL. Les applications client (s_client.c) et server(s_server) ont fait objet de modifications pour que le client et server OpenSSl soient capables de reconnaitre l'extension *blockchain_rev*.

— **SSL :** contient presque tous les fichiers *.c* qui devront être modifiés.

Nous avons ajouté l'extension *blockchain_rev : 9* dans le fichier (Tls1.h). Nous avons fait la déclaration des variables de l'extension dans le fichier (ssl.h), qui contient aussi les variables de session, et les variables globales de TLS. Ensuite, nous avons déclaré les variables de l'extension pour l'initialisation à chaque session dans le fichier (ssl_sess.c). Pour finir, la déclaration et le code des différentes fonctions pour manipuler l'extension dans le fichier (ssl_lib.c).

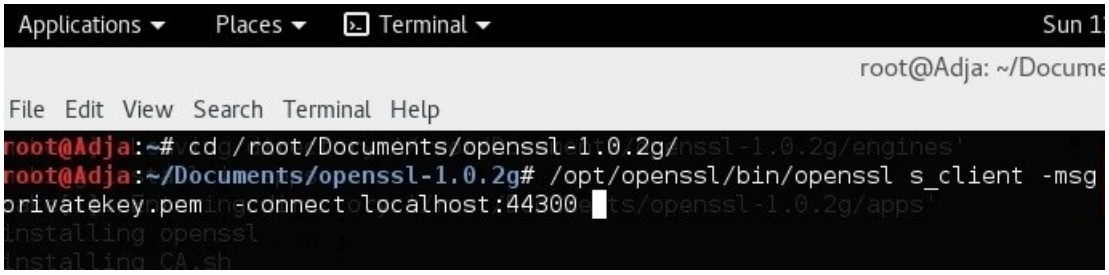



FIGURE 4.10 – Modification du client

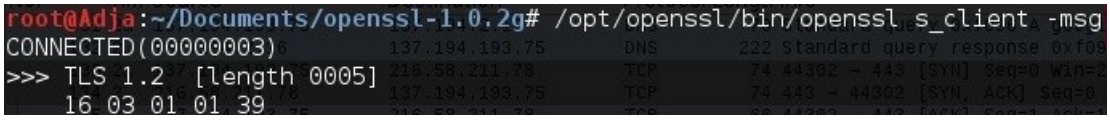



FIGURE 4.11 – Modification du server

**4.1.5.1.1 Test 1 : Modification du clientHello** Le premier type de test que nous avons effectué pour voir le bon fonctionnement de mes modifications a été avec un serveur Google. Nous avons ouvert une connexion TLS avec le serveur de Google même s'il ne supportait pas notre extension. Le schéma est résumé dans la figure 4.12.

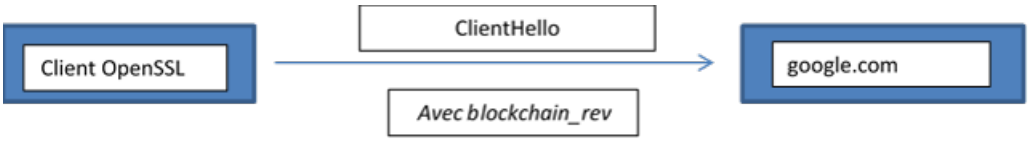



FIGURE 4.12 – Schéma de la requête

Nous pouvons observer le paquet ClientHELLO envoyé dans la figure 4.13 Les

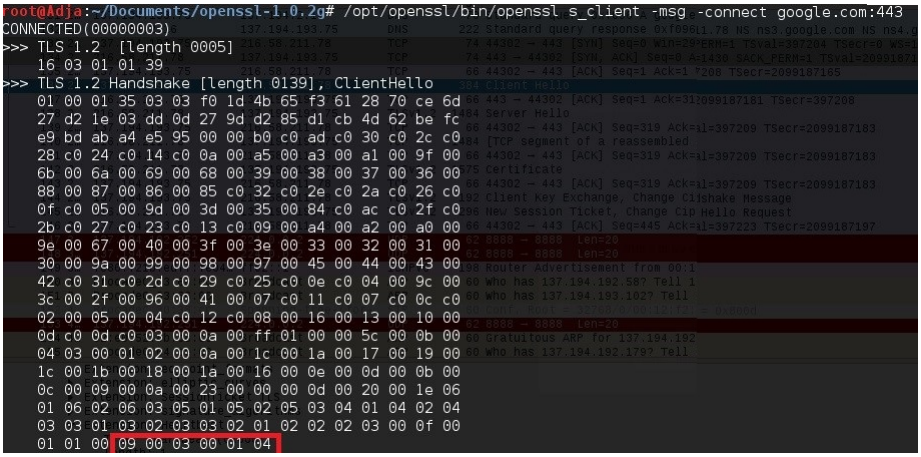



FIGURE 4.13 – Structure paquet ClientHELLO envoyé à google.com

différents codes indiquent :

— 00 09 : code extension (sur deux octets)

— 00 03 : Dimension totale de toute l'extension

— 00 01 : Dimension du champ suivant

Nous avons la confirmation que l'extension a été prise en charge dans le clientHELLO de notre version d'OpenSSL.

**4.1.5.1.2 Test 2 : Négociations** Cette fois-ci le test sera effectué entre un client et un serveur OpenSSL, tous deux modifiés. Dans ce cas nous pourrons aussi voir la réponse du serveur par le message ServerHELLO et tester donc le bon fonctionnement.


FIGURE 4.14 – Schéma requête avec client et serveur modifiés

Nous pouvons observer la requête dans la figure 4.14 et la réponse dans la figure 4.15. Nous pouvons observer maintenant dans la figure 4.16 le contenu de la requête


FIGURE 4.15 – Schéma réponse avec client et serveur modifiés

client et dans la figure 4.17 le contenu de la réponse serveur.

### 4.1.6 Conclusions

Notre objectif était de proposer une méthode de révocation capable de concurrencer les méthodes de révocation classiques, ce que nous pensons avoir atteint. Les résultats décrits dans les graphes 4.8 montrent que notre solution est plus intéressante que les méthodes standard dans 95 % des cas observés. Dans les cas restants, notre approche reste concurrente jusqu'à des fragment de 750 éléments. Au délà il faudrait des études plus approfondies. Comme travaux futurs, il serait pertinent d'optimiser notre solution, notamment dans le cas des certificats effectivement révoqués. Comme nous avons pu l'observer, ce cas est pénalisant pour notre approche. La réussite de la modification du protocole Handshake nous permet d'assurer l'applicabilité de notre solution dans TLS qui était un de nos objectifs.

## 4.2 ICRLDP

Dans ce chapitre nous allons implémenter notre solution (ICRLDP) et effectuer des tests de performance. Nous allons conclure avec les résultats des tests.

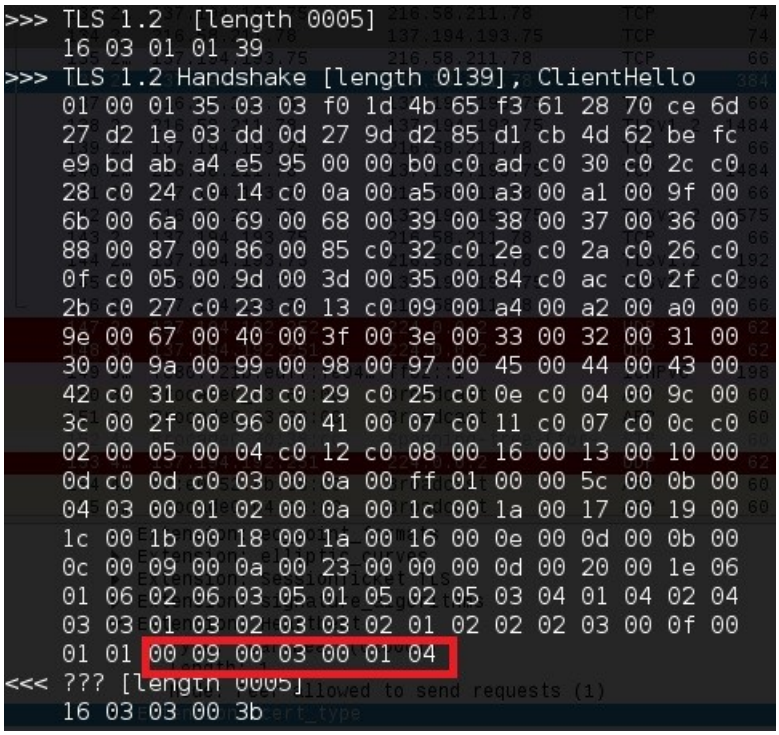



FIGURE 4.16 – Contenu requête avec client et serveur modifiés

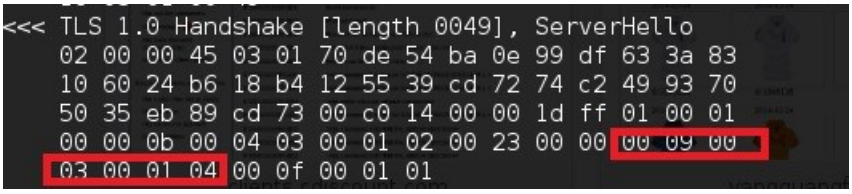



FIGURE 4.17 – Contenu réponse avec client et serveur modifiés

### 4.2.1 Implémentation et Analyse des performances

Nous commençons par décrire notre scénario de mise en œuvre (figure 4.18), puis une analyse des résultats et enfin une comparaison des résultats obtenus avec la méthode impliquant la CRL.

Nous avons utilisé le simulateur réseau appelé Veins [Vei] pour effectuer nos tests. Veins est un framework open source pour la simulation des réseaux véhiculaires. Il est basé sur deux simulateurs bien établis : OMNeT ++ [OMN] qui est un simulateur de réseau basé sur les événements, et SUMO [SUM] qui est un simulateur de trafic routier. Notre application est codée en C++ et exécutée dans le simulateur. Le simulateur a été installé sur un ordinateur avec les caractéristiques suivantes : core i7-3770, X86 _64 64 bits, 3,4 GHz, 8 Go de mémoire RAM, Ubuntu 16.04. Nous avons utilisé Veins pour reproduire les conditions réelles du trafic routier, en utilisant une partie de la carte du 15 *ième* arrondissement comma aire geographique. Nous nous sommes concentrés sur une aire de 600 m * 600 m autour de notre laboratoire à Télécom Paris. Nous avons collecté des données depuis environ *400* véhicules différents. Nous avons utilisé que des messages CAM pour demander et recevoir un fragment du point de distribution situé dans le RSU. Aucun nouveau message n'est introduit. Les fragments sont directement créés à partir des RSUs. Nous n'avons pas

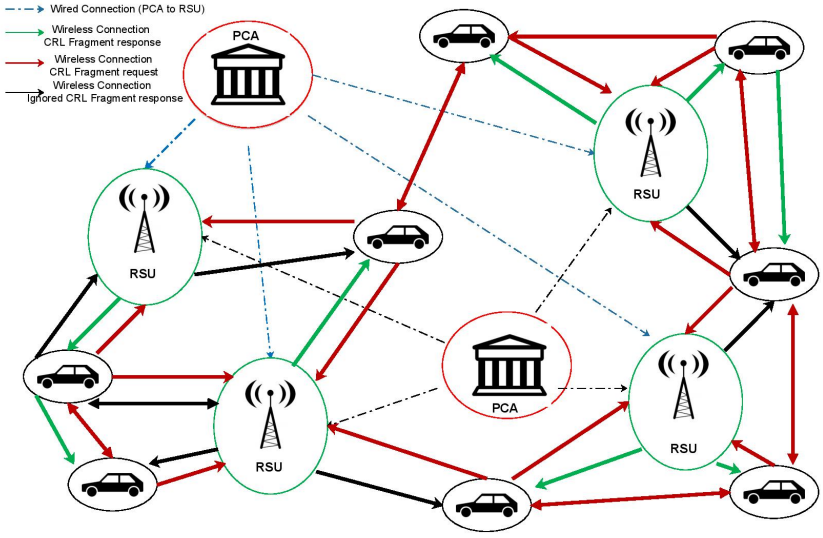



FIGURE 4.18 – Scénario de tests

pris en compte les échanges entre le AA et RSU, parce que nous partons de l'hypothèse que la communication entre ces deux entités est bien établie et fiable. Les entités de notre simulation sont les véhicules et les RSUs. Nous avons évalué selon trois métriques : le nombre de messages reçus, le nombre de requêtes envoyées et le temps nécessaire pour acquérir ou mettre à jour un fragment. Nous avons analysé le nombre de messages reçus par véhicule dans l'objectif d'estimer les couts et les effets sur les communications de notre solution. Nous avons également analysé le nombre de requêtes envoyées par véhicule pour estimer le nombre de requêtes qui doivent être envoyés pour obtenir ou mettre à jour un fragment. La dernière métrique exprime le temps minimal requis pour obtenir ou mettre à jour un fragment. Nous avons effectué dans chacun des cas de nombreux tests d'une durée d'une heure chacun. Les paramètres de simulation sont décrits dans le tableau 4.4.

| **Info** | **Value** |
|---|---|
| Quantité de véhicules | 80 |
| Nombre d'accidents simulés | 4 |
| Durée de prise en charge de l'accident | 50s |
| Dimension des fragments | 500 lines |
| Durée des tests | 1 h |
| Distribution des RSUs | Uniforme |
| Vitesse maximale des véhicules | 50 km/h |

TABLE 4.4 – Paramètres de simulation

$t_{req}$ = *Moment d'émission de la première requête .*
$t_{res}$= *Moment de réception du fragment demandé.*
$\sigma$ = *écart type.*

$\tau_{updt}$= *Latence d'actualisation d'un fragment .*
$\bar{m}$= *Moyenne des messages totale reçus. Tous messages compris.*
$\bar{r_m}$= *Quantité moyenne de requêtes envoyés, avant de recevoir une réponse favorable.*
$B_r$=*Beacon interval*
$\tau_{updt} = t_{res} - t_{req}$

| Fields | $F_n$ | $B_r$ (s) | $\tau_{updt}$ | | $\bar{m}$ | | $\bar{r_m}$ | |
|---|---|---|---|---|---|---|---|---|
| | | | $\bar{A}$ | $\sigma$ | $\bar{A}$ | $\sigma$ | $\bar{A}$ | $\sigma$ |
| CRL | 1 | 0.25 | 7.1 | 11.7 | 3062 | 2596.1 | 54.5 | 78.1 |
| | | 0.5 | 3.6 | 5.4 | 1640 | 1342.5 | 31.3 | 42.4 |
| | | 0.75 | 2.7 | 5.4 | 1123 | 973 | 18.1 | 23.6 |
| | | 1 | 2.3 | 3.1 | 842 | 686.8 | 16.5 | 20.2 |
| | | 5 | 0.8 | 1.8 | 198 | 181.2 | 5.08 | 10.3 |
| ICRL DP | 2 | 0.25 | 0.8 | 3 | 2805 | 2088 | 10.8 | 32.1 |
| | 5 | | 0.5 | 2 | 2962 | 2351 | 6.6 | 21.1 |
| | 10 | | 0.5 | 2.8 | 3757 | 2478.5 | 5.1 | 20.9 |
| | 15 | | 0.03 | 0.2 | 4771 | 2972.2 | 3.5 | 11.8 |
| | 20 | | 0.05 | 0.2 | 3135 | 2328.2 | 5.3 | 18.4 |
| | 2 | 0.5 | 1.1 | 4 | 2040 | 1409 | 1.7 | 10.5 |
| | 5 | | 0.6 | 2.9 | 2018 | 1302.1 | 3.8 | 12.3 |
| | 10 | | 0.2 | 0.7 | 1576 | 1172.4 | 2.6 | 7.4 |
| | 15 | | 0.2 | 1.7 | 1878 | 1273 | 2.3 | 8.6 |
| | 20 | | 0.2 | 0.9 | 1465 | 1038.8 | 3 | 8.3 |
| | 2 | 0.75 | 0.07 | 0.4 | 1095 | 959 | 5.9 | 25.4 |
| | 5 | | 0.03 | 0.2 | 958 | 762.4 | 2.9 | 6.9 |
| | 10 | | 0.03 | 0.2 | 928 | 777.2 | 0.1 | 0.9 |
| | 15 | | 0.06 | 0.3 | 948 | 807.6 | 5 | 17.8 |
| | 20 | | 0.03 | 0.2 | 908 | 747.3 | 0.1 | 0.8 |
| | 2 | 1 | 0.2 | 0.6 | 613 | 697 | 11.2 | 28.6 |
| | 5 | | 0.06 | 0.2 | 893 | 772.9 | 2.7 | 6.2 |
| | 10 | | 0.04 | 0.2 | 795 | 585.5 | 2.1 | 4 |
| | 15 | | 0.06 | 0.2 | 719 | 646.4 | 2.8 | 5.6 |
| | 20 | | 0.08 | 0.5 | 641 | 620.3 | 4.5 | 16.5 |
| | 2 | 5 | 0.3 | 0.8 | 171 | 142.8 | 2.5 | 4.6 |
| | 5 | | 0.16 | 0.4 | 168 | 131.3 | 1.7 | 2 |
| | 10 | | 0.16 | 0.4 | 229 | 143.4 | 1.4 | 1 |
| | 15 | | 0.19 | 0.6 | 163 | 132 | 1.7 | 2.8 |
| | 20 | | 0.2 | 0.6 | 210 | 138 | 1.6 | 1.6 |

TABLE 4.5 – Résultats simulation

Nous avons effectué de nombreux tests divisés en deux parties. Nous avons testé tout d'abord la méthode CRL puis notre méthode ICRLDP. Dans la première partie des tests nous avons fait varier la fréquence de transmission des balises CAM afin de contrôler si ce facteur n'a pas d'impact sur les résultats. Nous avons fait évoluer le délai de transmission dans la plage des valeurs recommandées par le standard, c'est à dire $100ms < x < 10000ms$. $x$ représente la fréquence de transmission. Dans la deuxième partie, nous avons effectué de nombreux tests en utilisant cette fois notre méthode ICRLDP comme méthode de révocation. Nous avons pour chaque

fréquence de balise dans la première série de tests, utilisés un différent nombre de fragments. Les résultats obtenus et décrits dans le tableau 4.5 nous donnent trois informations fondamentales :

— Le nombre de messages reçus $\bar{m}$ par véhicule est proportionnel à la fréquence de transmission des messages CAM. Cela s'explique car si nous augmentons la fréquence d'envoi de messages, il y a plus de messages échangés dans le réseau et donc plus de messages reçus par véhicule.

— Nous constatons que plus élevée est la fréquence d'envoi des messages, mauvaise est la réception des messages. Nous pouvons l'interpréter comme une conséquence des interférences et des collisions entre les messages.

— Le nombre de fragments a un effet décisif sur le délai nécessaire à obtenir ou actualiser un fragment côté station ITS. C'est-à-dire que le délai d'actualisation des fragments de CRL est meilleur quand nous avons beaucoup de fragments. Cette dernière observation prouve que notre solution est scalable.

Les données obtenues de chaque véhicule pendant la première et la deuxième série de tests peuvent être consultées par dans les annexes : A et B. La comparaison des deux systèmes mis en examen est effectué à travers les graphes 4.25 4.32 4.39.

Nous nous sommes concentrés sur ces trois métriques car nous pensons qu'elles résument clairementles avantages de notre protocole de distribution.

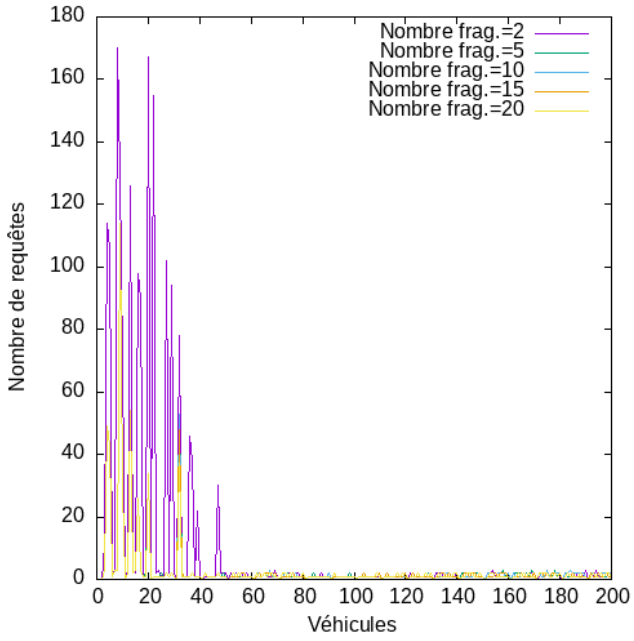



FIGURE 4.19 – Nombre de requêtes envoyées par véhicule selon une fréquence de balises CAM de 0.25s

Le graphe 4.19 montre le nombre de requêtes envoyées par véhicule sur une période d'une heure, dans le cas d'une fréquence des balises CAM de 0.25s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation de la liste de révocation.

Le graphe 4.20 montre le nombre de requêtes envoyées par véhicule sur une période d'une heure, dans le cas d'une fréquence des balises CAM de 0.5s et selon la

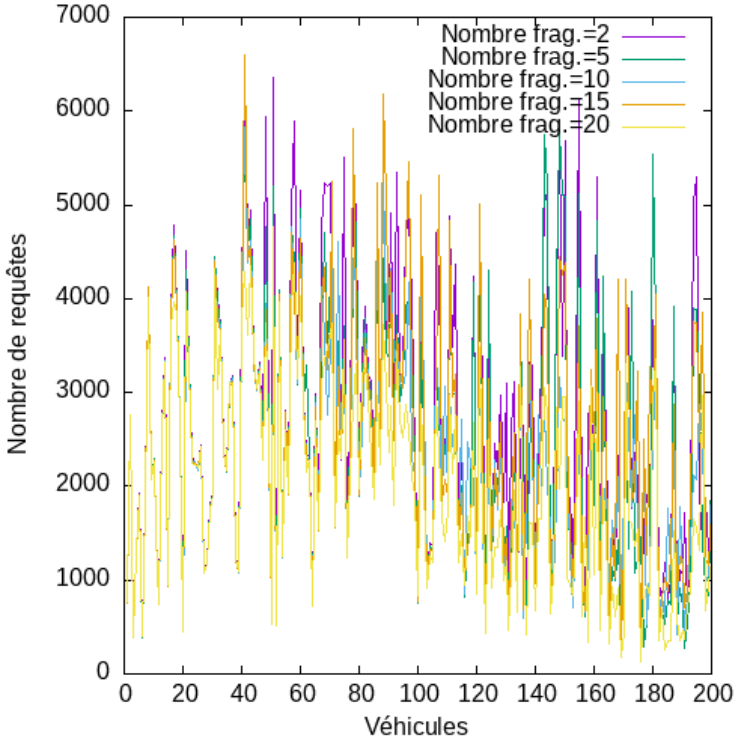



FIGURE 4.20 – Nombre de requêtes envoyées par véhicule selon une fréquence de balises CAM de 0.5s

méthode ICRLDP. Chaque courbe indique une différente fragmentation de la liste de révocation.

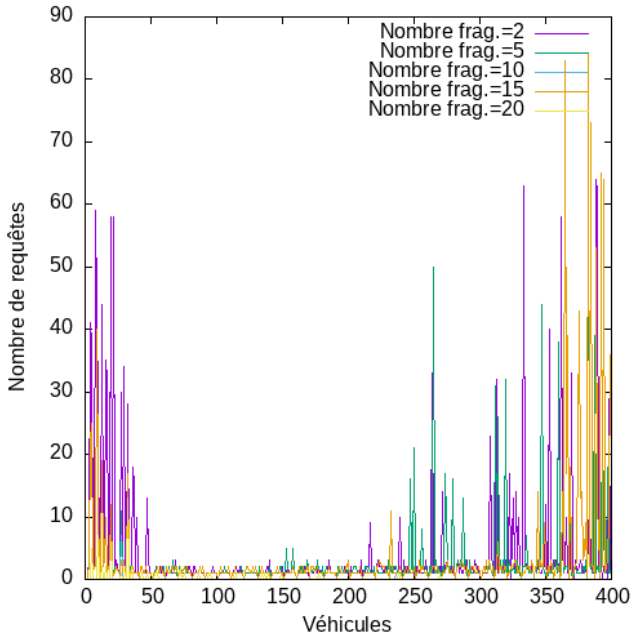



FIGURE 4.21 – Nombre de requêtes envoyées par véhicule selon une fréquence de balises CAM de 0.75s

Le graphe 4.21 montre le nombre de requêtes envoyées par véhicule sur une période d'une heure, dans le cas d'une fréquence des balises CAM de 0.75s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation de la liste de révocation.

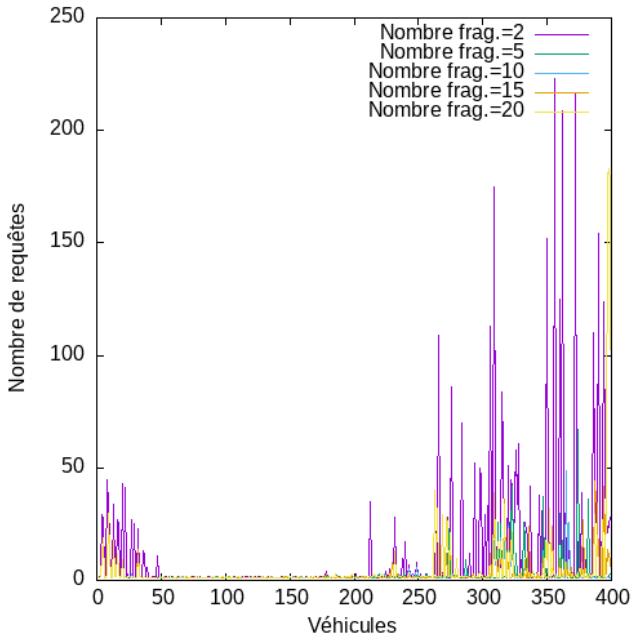



FIGURE 4.22 – Nombre de requêtes envoyées par véhicule selon une fréquence de balises CAM de 1s

Le graphe 4.22 montre le nombre de requêtes envoyées par véhicule sur une période d'une heure, dans le cas d'une fréquence des balises CAM de 1s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation de la liste de révocation.

Le graphe 4.23 montre le nombre de requêtes envoyées par véhicule sur une période d'une heure, dans le cas d'une fréquence des balises CAM de 5s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation de la liste de révocation.

Le graphe 4.38 montre le nombre de messages reçus par véhicule sur une période d'une heure selon la méthode CRL standard. Chaque courbe indique une différente fréquence des balises CAM.

Le graphe 4.25 compare la moyenne des requêtes envoyées par véhicule selon les deux approches de révocation CRL et ICRLDP. Nous observons que la moyenne des requêtes à envoyer par véhicule pour récupérer un fragment selon la méthode ICRLDP est nettement inférieure à toutes les approches. Ce fait a comme conséquence directe de réduire la latence d'actualisation des fragments car l'information est obtenue rapidement. Le fait de permettre aux stations de répondre à un impact positif, car permet de réduire les délais d'attentes et donc le nombre de requêtes nécessaires pour avoir un fragment. Il est très probable qu'un véhicule lié à notre point de distribution soit dans les environs.

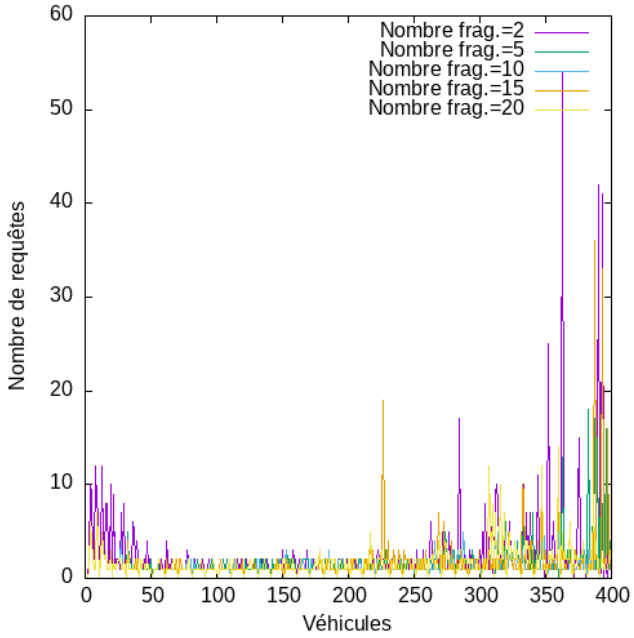



FIGURE 4.23 – Nombre de requêtes envoyées par véhicule selon une fréquence de balises CAM de 5s

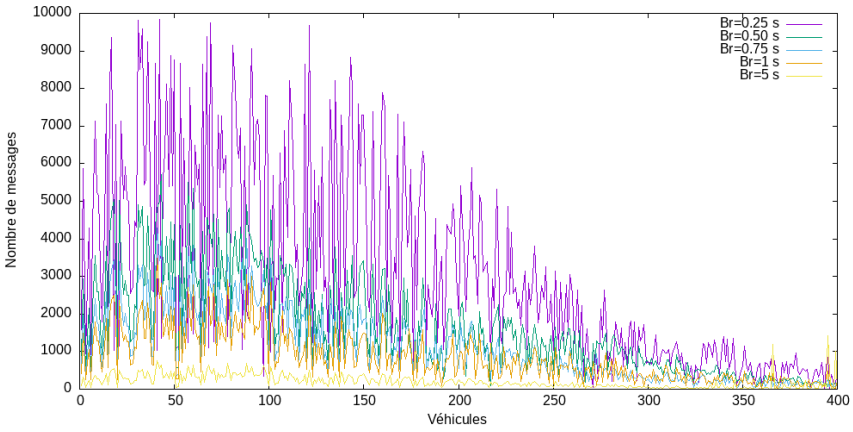



FIGURE 4.24 – Nombre de messages reçus par véhicule selon la méthode CRL standard

Le graphe 4.26 montre le temps nécessaire à chaque véhicule pour obtenir un fragment, dans le cas d'une fréquence des balises CAM de 0.25s, selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de révocation.

Le graphe 4.27 montre le temps nécessaire à chaque véhicule pour obtenir un fragment, dans le cas d'une fréquence des balises CAM de 0.50s, selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de révocation.

Le graphe 4.28 montre le temps nécessaire à chaque véhicule pour obtenir un fragment, dans le cas d'une fréquence des balises CAM de 0.75s, selon la méthode ICRLDP. Chaque courbe indique in différente fragmentation des listes de révocation.

Le graphe 4.29 montre le temps nécessaire à chaque véhicule pour obtenir un fragment, dans le cas d'une fréquence des balises CAM de 1s, selon la méthode

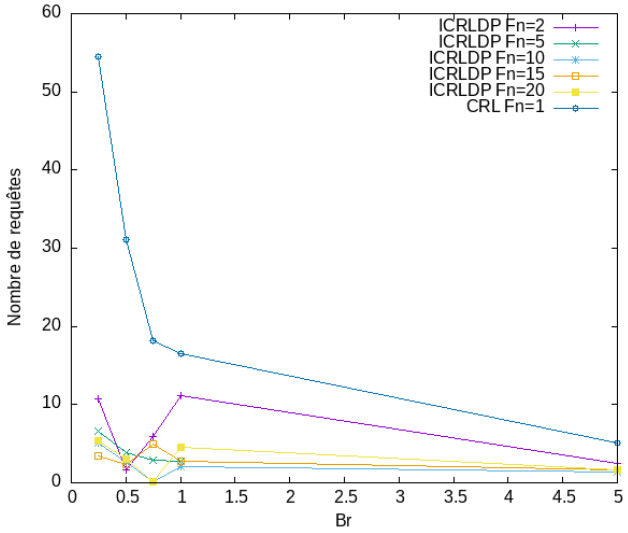



FIGURE 4.25 – Nombre moyen de requêtes envoyées par véhicule avant de recevoir un fragment

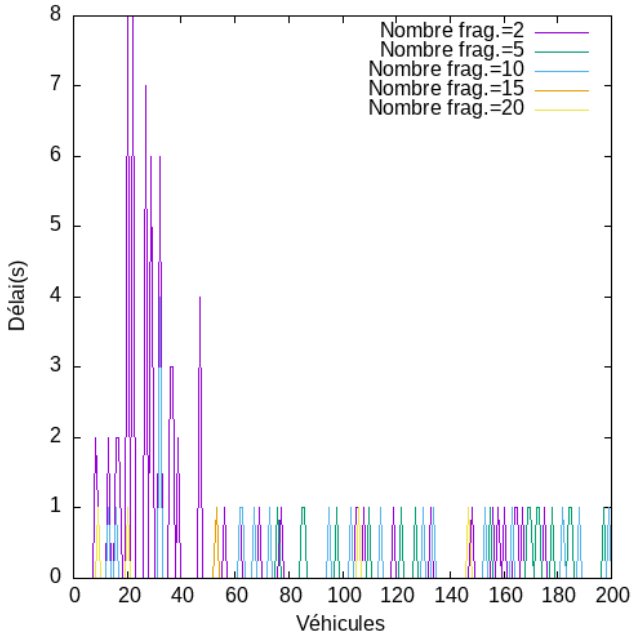



FIGURE 4.26 – Temps nécessaire pour obtenir un fragment selon une fréquence des balises CAM de 0.25s

ICRLDP. Chaque courbe indique une différente fragmentation des listes de révocation.

Le graphe montre le temps nécessaire à chaque véhicule pour obtenir un fragment, dans le cas d'une fréquence des balises CAM de 5s, selon la méthode ICRLDP. Chaque courbe indique une différente des listes de révocation.

Le graphe 4.31 montre le temps nécessaire à chaque véhicule pour obtenir la CRL. Chaque courbe indique une différente fréquence des balises CAM.

Le graphe 4.32 met en relation la moyenne des délais calculée sur 400 véhicules avec la fréquence d'envoi des balises CAM en fonction de l'approche utilisée et de la fragmentation des listes de révocation.

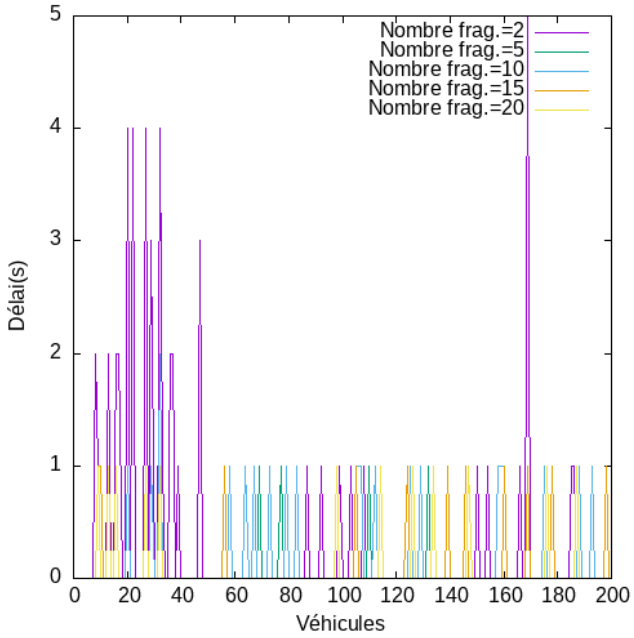



FIGURE 4.27 – Temps nécessaire pour obtenir un fragment selon une fréquence des balises CAM de 0.50s

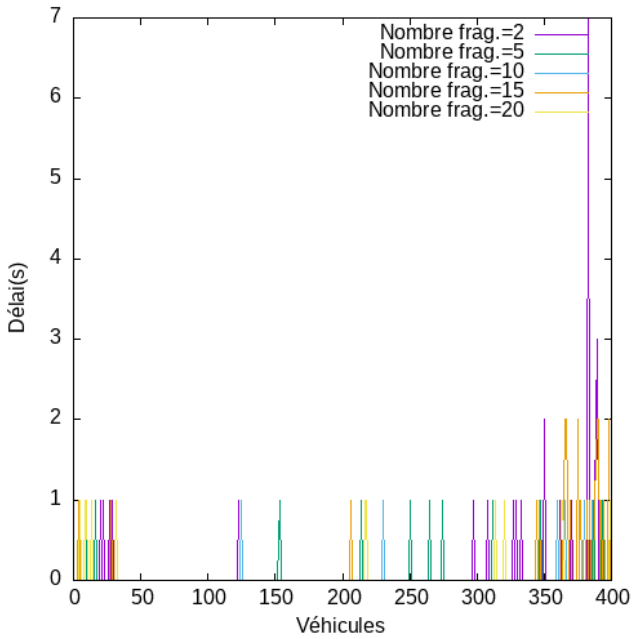



FIGURE 4.28 – Temps nécessaire pour obtenir un fragment selon une fréquence des balises CAM de 0.75s

Le graphe 4.32 compare les moyennes des délais de réception par véhicule des deux approches CRL et ICRLDP. Nous constatons que les délais observés avec l'approche CRL sont largement supérieurs, parce qu'il il faut toujours attendre d'être dans le rayon de communication du RSU pour actualiser son fragment. L'approche ICRLDP montre des délais moins importants. Ces résultats sont dus au fait que dans la grande majorité des cas, il y a toujours des véhicules aux alentours qui possèdent le bon fragment et donc qui peuvent disséminer les fragments. Il n'est pas obligatoire

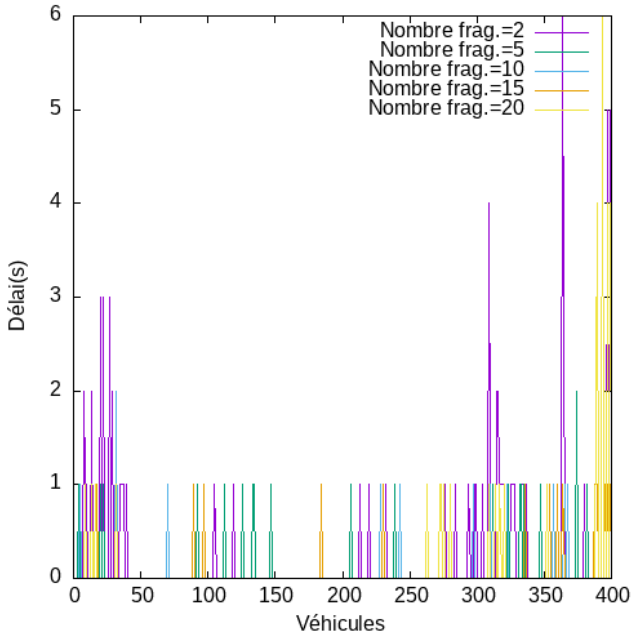



FIGURE 4.29 – Temps nécessaire pour obtenir un fragment selon une fréquence des balises CAM de 1s.

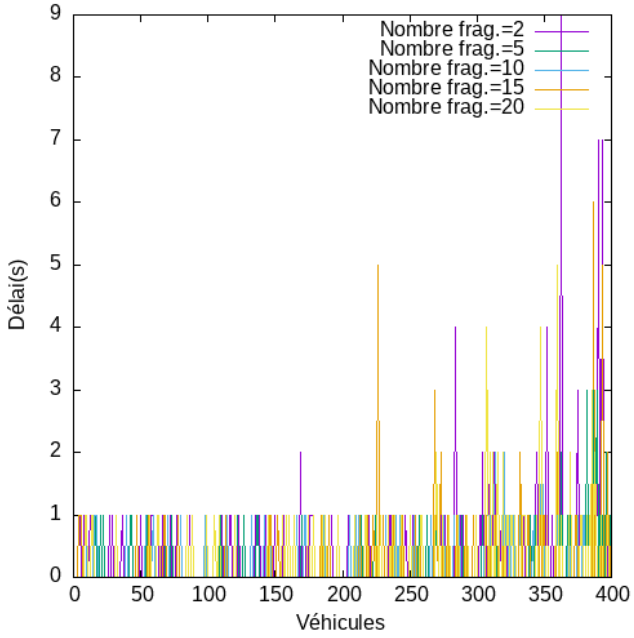



FIGURE 4.30 – Temps nécessaire pour obtenir un fragment selon une fréquence des balises CAM de 5s

de se rapproche d'un RSU, ce qui réduit les délais d'attente.

Le graphe 4.33 montre le nombre de messages reçus par véhicule sur une période d'une heure pour une fréquence des balises CAM de 0.25s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe 4.34 montre le nombre de messages reçus par véhicule sur une période d'une heure pour une fréquence des balises CAM de 0.50s et selon la méthode

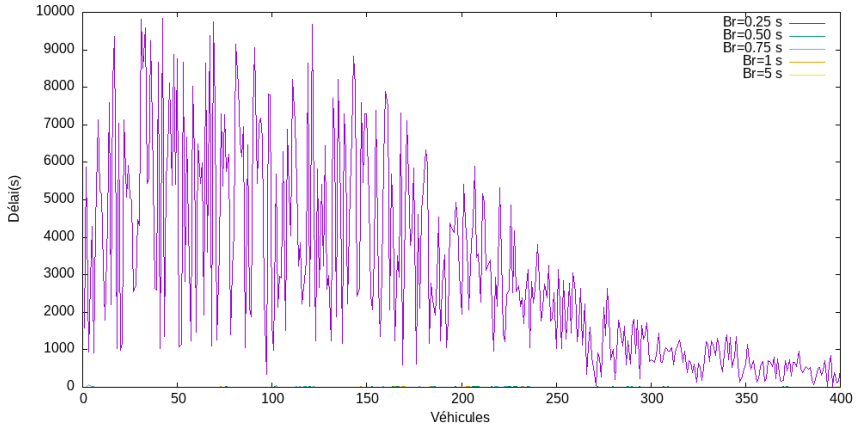



FIGURE 4.31 – Temps nécessaire à chaque véhicule pour obtenir la CRL

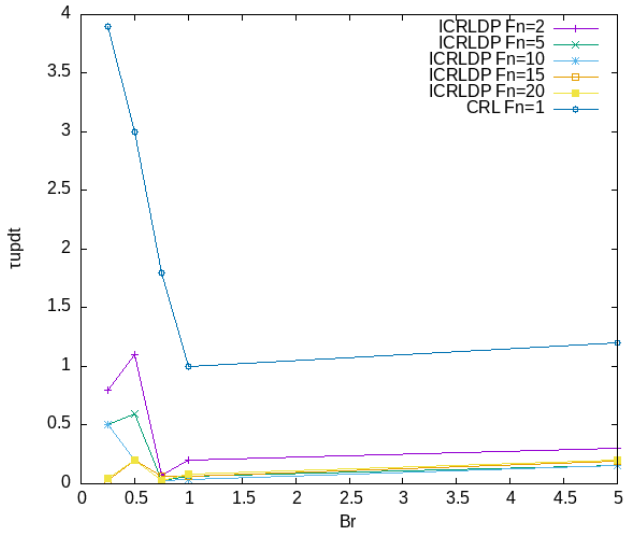



FIGURE 4.32 – Résumé des délais des approches CRL et CRLDP

ICRLDP. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe 4.35 montre le nombre de messages reçus par véhicule sur une période d'une heure pour une fréquence des balises CAM de 0.75s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de révocation.

Le graphe 4.36 montre le nombre de messages reçus par véhicule sur une période d'une heure pour une fréquence des balises CAM de 1s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe 4.37 montre le nombre de messages reçus par véhicule sur une période d'une heure pour une fréquence de balises CAM de 5s et selon la méthode ICRLDP. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe 4.38 montre le nombre de messages reçus par véhicule sur une période d'une heure selon la méthode CRL standard. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe met 4.39 en relation la moyenne (calculée sur 400 véhicules) du nombre

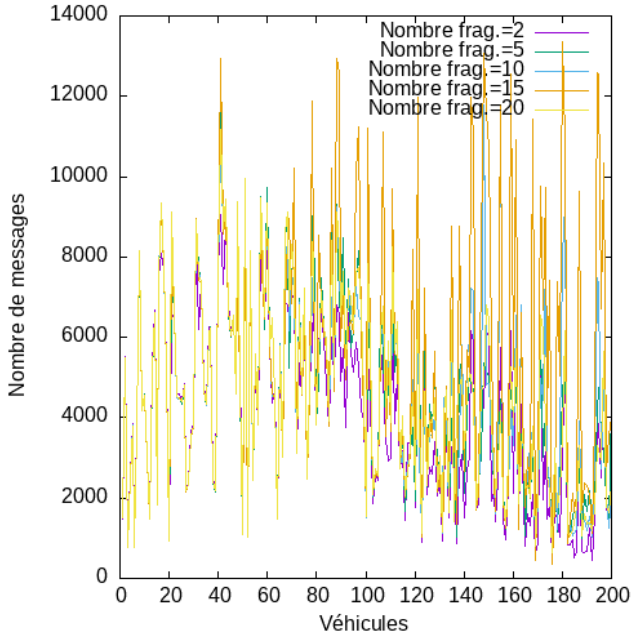



FIGURE 4.33 – Nombre de messages reçus selon une fréquence des balises CAM de 0.25s

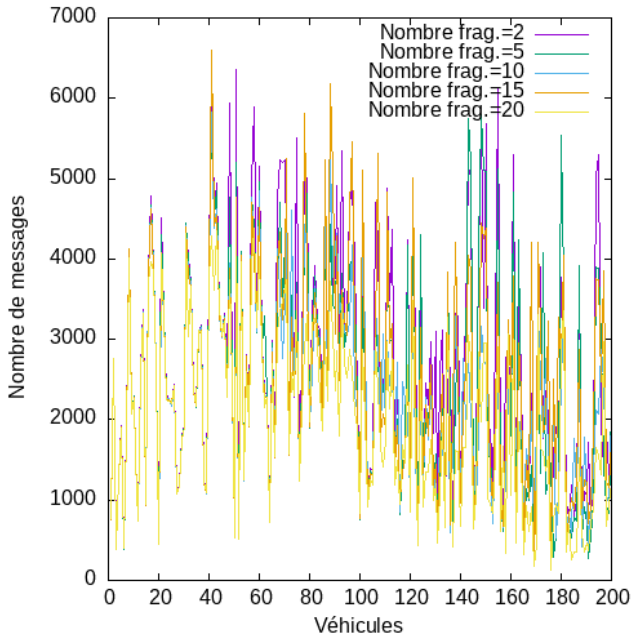



FIGURE 4.34 – Nombre de messages reçus selon une fréquence des balises CAM de 0.5s

de messages reçus sur une période d'une heure avec la fréquence d'envoi des messages CAM dans les cas ICRLDP et CRL. Chaque courbe indique une différente fragmentation des listes de revocation.

Le graphe 4.39 compare les moyennes des messages reçus par véhicule des deux approches CRL et ICRLDP. Le graphe montre clairement que la méthode produit légèrement moins ou la même quantité de message que l'approche ICRLDP. Ceci est très intéressant car l'utilisation de la bande passante devrait être le point critique de

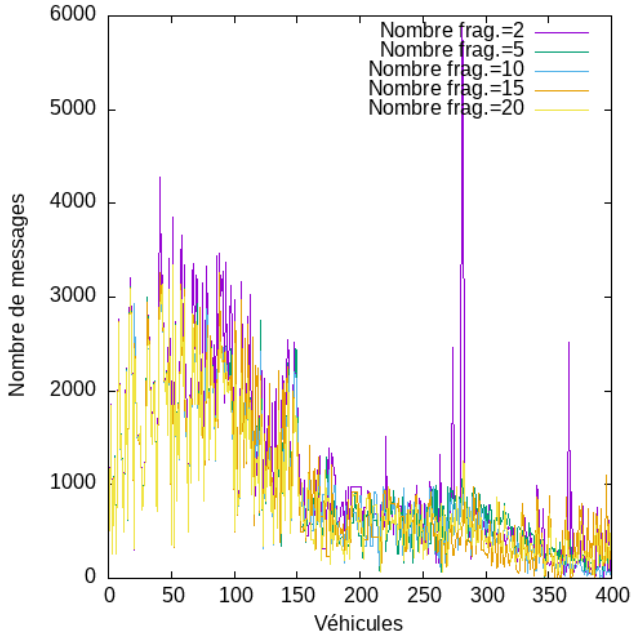



FIGURE 4.35 – Nombre de messages reçus selon une fréquence des balises CAM de 0.75s

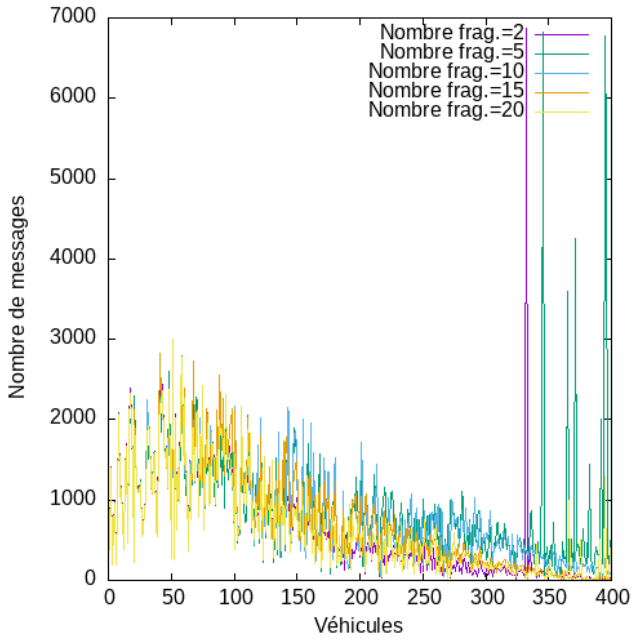



FIGURE 4.36 – Nombre de messages reçus selon une fréquence des balises CAM de 1s

notre approche à cause du droit de réponse attribué aux véhicules. Nous observons par contre que les résultats sont pareils à la méthode standard. Notre approche permet donc de réduire la latence et la quantité de données téléchargée par véhicule sans accroitre l'utilisation de la bande passante. Nous croyons que le fait d'autoriser les véhicules à répondre permet à certains véhicules d'avoir le fragment sans faire de requête. Le graphe permet aussi d'observer qu'il est possible de réduire l'impact de notre approche sur la bande passante en utilisant une optimale fragmentation des

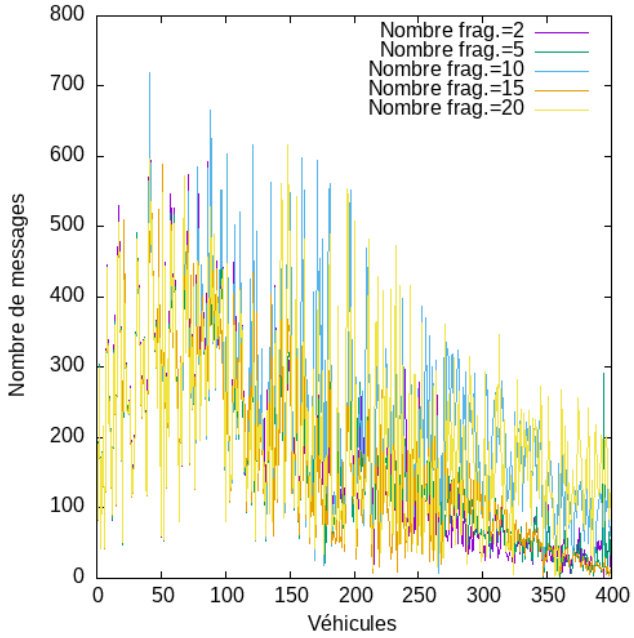



FIGURE 4.37 – Nombre de messages reçus selon une fréquence des balises CAM de 5s

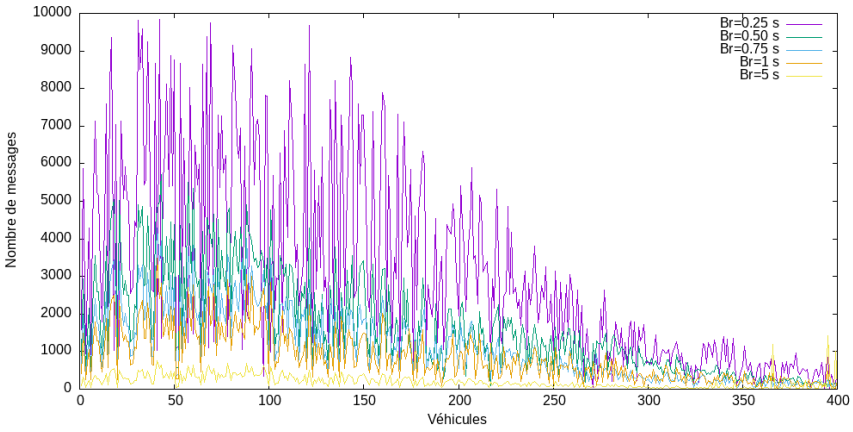



FIGURE 4.38 – Nombre de messages reçus par véhicule selon l'approche CRL

listes de révocation.

#### 4.2.1.1 Conclusions

Ce chapitre nous a permis d'implémenter et d'évaluer notre méthode de révocation nommée ICRLDP. Notre solution a été obtenu en optimisant la méthode CRLDP à travers un procédé cryptographique. Notre approche n'introduit pas de nouveau types de message et n'impacte pas la bande passante. L'analyse des résultats obtenus montre que notre approche est réaliste et peut faire face aux exigences de scalabilité mieux que les approches prisent en comparaison. Je peux affirmer que l'objectif fixé a été atteint. Nous avons proposé une solution qui fait face à une partie des problèmes liée à la révocation. Mais des questions restent encore ouvertes, telles que : comment trouver la quantité optimale de fragments, comment répartir les sections

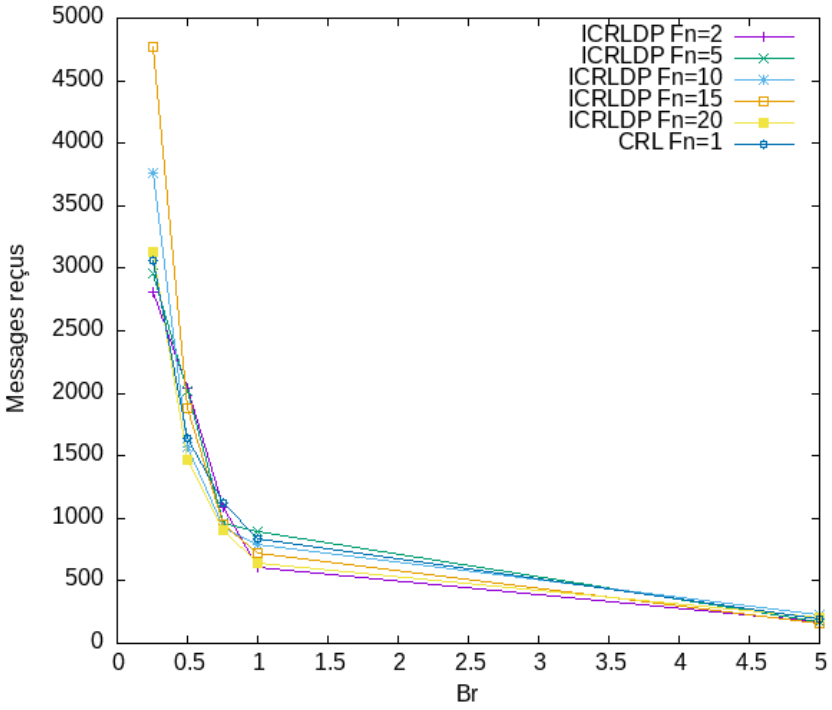



FIGURE 4.39 – Moyenne des messages reçus selon les approches CRL et ICRLDP

dans un fragment tout en évitant la traçabilité. Le problème de la traçabilité post-révocation n'a pas été adressé. En effet les listes publiques de révocation permettent de dévoiler tous les certificats détenus et donc de tracer une fois publié toutes les actions dans le réseau d'un véhicule avant sa révocation. Nous laissons ouvertes ces pistes de recherche pour les prochaines contributions sur le sujet.

# Chapitre 5

# Conclusions

La PKI est l'infrastructure de confiance que nous pouvons définir mature pour sa fiabilité éprouvée. Certains services de la PKI, comme par exemple la gestion de la révocation peut être optimisée pour répondre aux besoins des applications à venir, dont les enjeux sont toujours plus importants.

## 5.1 Conclusions

Nos travaux de thèses ont débuté par une longue analyse et un état de l'art sur la PKI x509, sur ses mécanismes de sécurité et sur la révocation des certificats. Ensuite nous avons fait un état de l'art et une analyse approfondie de la technologie Blockchain, de ses applications et de sa capacité à répondre à notre problématique. Cette analyse est synthétisée dans le sous-chapitre état de l'art du chapitre III. L'analyse de la Blockchain nous a montré sa résilience, ce qui justifie son importance pour optimisation des méthodes de révocation classiques. Ainsi nous avons conçu le Blockchain-based CRLDP qui est une version optimisée de la méthode CRLDP. Nous avons aussi analysé la compatibilité des différentes approches de révocation existante avec la Blockchain. La CRLDP, la Blockchain et les filtres de bloom nous ont permis de concevoir la méthode BL-CRLDP. Nous avons implémenté et confirmé la validité de notre approche à travers plusieurs tests de performances. Les résultats et les critères d'évaluation sont résumés dans le chapitre V. Nous avons ensuite fait un état de l'art sur les systèmes C-ITS et la VPKI. Les informations obtenues suite à cette analyse sont détaillées dans le chapitre II. Nous avons conçu la méthode ICRLDP pour le contexte véhiculaire en combinant la méthode CRLDP avec des choix cryptographiques. Nous avons effectué en environnement virtuel plusieurs tests de performance. Les résultats sont décrits dans le chapitre IV.

## 5.2 Perspectives futures

Des pistes restent ouvertes pour de nouvelles contributions sur le sujet. Il pourrait être intéressant de combiner nos deux solutions pour avoir une interopérabilité entre les deux contextes étudiés. L'approche BL-CRLDP peut être revue pour optimiser son comportement face aux faux positifs des filtres de bloom. Nous pouvons réfléchir sur comment utiliser les approches de révocation classique pour répondre aux faux positifs des filtres de bloom. Il pourrait intéressant voir comment intégrer le Blockchain-based CRLDP à TLS 1.3 et proposer un RFC dans ce sens. Dans le cas de la solution ICRLDP, il pourrait interessant étudier la fragmentation idéale des les listes de révocation. Il pourrait aussi pertinant d'étudier la répartition des véhicules sur les différentes sections des fragments. Notre approche ne prend pas en compte la

problématique de la traçabilité post révocation. Pour finir on pourrait penser à budgétiser ces solutions en comparaison avec les solutions classiques. Car il est établi que la PKI coûte trop cher et la révocation tient une part importante dans l'exploitation.

# Bibliographie


[Lam83] Leslie LAMPORT. « The weak Byzantine generals problem ». In : *Journal of the ACM (JACM)* 30.3 (1983), p. 668-676.

[Mer90] Ralph C MERKLE. *A certified digital signature. In" Advances in Cryptology—CRYPTO'89 Proceedings*. 1990.

[KD96] Paul C KOCHER et Tim DIERKS. « The TLS Protocol Version 1.0 ». In : (1996).

[Coo99] David A COOPER. « A model of certificate revocation ». In : *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*. IEEE. 1999, p. 256-264.

[Mye+99] Michael MYERS et al. *X. 509 Internet public key infrastructure online certificate status protocol-OCSP*. Rapp. tech. RFC 2560, 1999.

[Res99] Eric RESCORLA. *Diffie-hellman key agreement method*. Rapp. tech. 1999.

[Arn+00] Andre ARNES et al. « Selecting revocation solutions for PKI ». In : *Fifth Nordic Workshop on Secure IT Systems (NORDSEC 2000)*. Citeseer. 2000.

[Coo00] D. A. COOPER. « A more efficient use of delta-CRLs ». In : *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000*. 2000, p. 190-202.

[Cop00] Cooper A COPER. « A More Efficient Use of Delta CRL ». In : *The Proceedings of the 2000 IEEE Symposium on Security and Privacy*. 2000, p. 190-202.

[NN00] Moni NAOR et Kobbi NISSIM. « Certificate revocation and certificate update ». In : *IEEE Journal on selected areas in communications* 18.4 (2000), p. 561-570.

[Woh00] Petra WOHLMACHER. « Digital certificates : a survey of revocation methods ». In : *Proceedings of the 2000 ACM workshops on Multimedia*. 2000, p. 111-114.

[El +02] Magda EL ZARKI et al. « Security issues in a future vehicular network ». In : *European Wireless*. T. 2. 2002.

[Hou+02] Russell HOUSLEY et al. « Internet x. 509 public key infrastructure Certificate and certificate revocation list (crl) profile (RFC 3280) ». In : *Internet Engineering Task Force, Network Working Group* (2002).

[Mit02] Michael MITZENMACHER. « Compressed bloom filters ». In : *IEEE/ACM Transactions on Networking (TON)* 10.5 (2002), p. 604-612.

[BM04] Andrei BRODER et Michael MITZENMACHER. « Network applications of bloom filters : A survey ». In : *Internet mathematics* 1.4 (2004), p. 485-509.

[Haj04] Ibrahim HAJJEH. *Securite des echanges. Conception et validation d'un nouveau protocole pour la securisation des echanges*. Rapp. tech. 2004.

[Muñ+04] Jose L MUÑOZ et al. « Certificate revocation system implementation based on the Merkle hash tree ». In : *International Journal of Information Security* 2.2 (2004), p. 110-124.

[RHA04] Maxim RAYA, Jean-Pierre HUBAUX et Imad AAD. « DOMINO : a system to detect greedy behavior in IEEE 802.11 hotspots ». In : *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. 2004, p. 84-97.

[ET05] Pasi ERONEN et Hannes TSCHOFENIG. *Pre-shared key ciphersuites for transport layer security (TLS)*. Rapp. tech. 2005.

[Jai05] Gaurav JAIN. *Certificate revocation : A survey*. 2005.

[Mic05] Silvio MICALI. *The internet of things*. report. 2005.

[PP05] Bryan PARNO et Adrian PERRIG. « Challenges in securing vehicular networks ». In : *Workshop on hot topics in networks (HotNets-IV)*. Maryland, USA. 2005, p. 1-6.

[RH05] Maxim RAYA et Jean-Pierre HUBAUX. « The security of vehicular ad hoc networks ». In : *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*. ACM. 2005, p. 11-21.

[Bla+06] Simon BLAKE-WILSON et al. *Transport layer security (TLS) extensions*. Rapp. tech. 2006.

[Hou+06] R HOUSLEY et al. *Internet x. 509 public key infrastructure : Certificate and crl profile (RFC 4630)*. Rapp. tech. RFC 4630, 2006.

[RPH06] Maxim RAYA, Panos PAPADIMITRATOS et Jean-Pierre HUBAUX. « Securing vehicular communications ». In : *IEEE wireless communications* 13.5 (2006).

[Ray+06] Maxim RAYA et al. « Certificate revocation in vehicular networks ». In : *Laboratory for computer Communications and Applications (LCA) School of Computer and Communication Sciences, EPFL, Switzerland* (2006).

[XYG06] Bin XIAO, Bo YU et Chuanshan GAO. « Detection and localization of sybil nodes in VANETs ». In : *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*. ACM. 2006, p. 1-8.

[Xu+06] Wenyuan XU et al. « Jamming sensor networks : attack and defense strategies ». In : *IEEE network* 20.3 (2006), p. 41-47.

[Cal+07] Giorgio CALANDRIELLO et al. « Efficient and robust pseudonymous authentication in VANET ». In : *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM. 2007, p. 19-28.

[LT07] Nai-Wei LO et Hsiao-Chien TSAI. « Illusion attack on vanet applications-a message plausibility problem ». In : *Globecom Workshops, 2007 IEEE*. IEEE. 2007, p. 1-8.

[Pap+07] Panagiotis PAPADIMITRATOS et al. « Architecture for secure and private vehicular communications ». In : *2007 7th International Conference on ITS Telecommunications*. IEEE. 2007, p. 1-6.

[RH07] Maxim RAYA et Jean-Pierre HUBAUX. « Securing vehicular ad hoc networks ». In : *Journal of computer security* 15.1 (2007), p. 39-68.

[Ray+07] Maxim RAYA et al. « Eviction of misbehaving and faulty nodes in vehicular networks ». In : *IEEE Journal on Selected Areas in Communications* 25.8 (2007).

[07] *Server-Based Certificate Validation Protocol (SCVP)*. IETF, 2007. URL : https://tools.ietf.org/html/rfc5055.

[Coo+08] David COOPER et al. « RFC 5280 : Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile ». In : *IETF, May* (2008).

[DR08] Tim DIERKS et Eric RESCORLA. « RFC 5246-the transport layer security (TLS) protocol version 1.2 ». In : *Internet Engineering Task Force* (2008).

[08] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF, 2008. URL : https://tools.ietf.org/html/rfc5280.

[LHH08] Kenneth P LABERTEAUX, Jason J HAAS et Yih-Chun HU. « Security certificate revocation list distribution for VANET ». In : *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*. ACM. 2008, p. 88-89.

[Mic08] Silvio MICALI. *Efficient certificate revocation*. US Patent 7,337,315. Fév. 2008.

[Nak08] Satoshi NAKAMOTO. « Bitcoin : A peer-to-peer electronic cash system ». In : (2008).

[PMH08] Panagiotis Panos PAPADIMITRATOS, Ghita MEZZOUR et Jean-Pierre HUBAUX. « Certificate revocation list distribution in vehicular communication systems ». In : *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*. ACM. 2008, p. 86-87.

[HHL09] Jason J HAAS, Yih-Chun HU et Kenneth P LABERTEAUX. « Design and analysis of a lightweight certificate revocation mechanism for VANET ». In : *Proceedings of the sixth ACM international workshop on VehiculAr Inter-NETworking*. ACM. 2009, p. 89-98.

[HS+10] Halabi HASBULLAH, Irshad Ahmed SOOMRO et al. « Denial of service (dos) attack and its possible solutions in VANET ». In : *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 4.5 (2010), p. 813-817.

[IZC10] Jesús Téllez ISAAC, Sherali ZEADALLY et José Sierra CAMARA. « Security attacks and solutions for vehicular ad hoc networks ». In : *IET communications* 4.7 (2010), p. 894-903.

[Bad11] Mohamad BADRA. « Securing communications between multiple entities using a single TLS session ». In : *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE. 2011, p. 1-4.

[Bad+12] Mohamad BADRA et al. « Extending TLS to secure multihost applications exchanges ». In : *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. IEEE. 2012, p. 2500-2505.

[Bib+12] Vimal BIBHU et al. « Performance analysis of black hole attack in VANET ». In : *International Journal Of Computer Network and Information Security* 4.11 (2012), p. 47.

[ETS12a] TS ETSI. « 102 940 : Intelligent Transport Systems (ITS) ; Security ; ITS communications security architecture and security management ». In : *Technical specification, European Telecommunications Standards Institute* (2012).

[ETS12b] TS ETSI. « ETSI TS 102 867 V1.1.1-Security-Mapping for IEEE 1609.2, 2012 ». In : *Technical specification, European Telecommunications Standards Institute* (2012).

[Fah+12] Sascha FAHL et al. « Why Eve and Mallory love Android : An analysis of Android SSL (in) security ». In : *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, p. 50-61.

[Geo+12] Martin GEORGIEV et al. « The most dangerous code in the world : validating SSL certificates in non-browser software ». In : *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, p. 38-49.

[KN12] S KING et S NADAL. « Peercoin–secure & sustainable cryptocoin ». In : *Aug-2012 [Online]. Available : https ://peercoin. net/whitepaper ()* (2012).

[RSS12a] Ajay RAWAT, Santosh SHARMA et Rama SUSHIL. « VANET : Security attacks and its possible solutions ». In : *Journal of Information and Operations Management* 3.1 (2012), p. 301.

[RSS12b] Ajay RAWAT, Santosh SHARMA et Rama SUSHIL. « VANET : Security attacks and its possible solutions ». In : *Journal of Information and Operations Management* 3 (jan. 2012).

[TRL12] Sasu TARKOMA, Christian Esteve ROTHENBERG et Eemil LAGERSPETZ. « Theory and practice of bloom filters for distributed systems ». In : *IEEE Communications Surveys & Tutorials* 14.1 (2012), p. 131-155.

[LL13] Xiaodong LIN et Xu LI. « Achieving efficient cooperative message authentication in vehicular ad hoc networks ». In : *IEEE Transactions on Vehicular Technology* 62.7 (2013), p. 3339-3348.

[Pet13] Y PETTERSEN. « The transport layer security (TLS) multiple certificate status request extension ». In : *RFC 6961*. 2013.

[RKS13] Ram Shringar RAW, Manish KUMAR et Nanhay SINGH. « Security challenges, issues and their solutions for VANET ». In : *International Journal of Network Security & Its Applications* 5.5 (2013), p. 95.

[San+13] Stefan SANTESSON et al. « X. 509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP. » In : *RFC* 6960 (2013), p. 1-41.

[Why+13] William WHYTE et al. « A security credential management system for V2V communications ». In : *2013 IEEE Vehicular Networking Conference*. IEEE. 2013, p. 1-8.

[Yee13] P. YEE. « Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile ». In : *IETF, January* (2013).

[Ben+14] Iddo BENTOV et al. « Proof of activity : Extending bitcoin's proof of work via proof of stake [extended abstract] y ». In : *ACM SIGMETRICS Performance Evaluation Review* 42.3 (2014), p. 34-37.

[Fan+14] Yawen FAN et al. « A cross-layer defense mechanism against GPS spoofing attacks on PMUs in smart grids ». In : *IEEE Transactions on Smart Grid* 6.6 (2014), p. 2659-2668.

[FVY14] Conner FROMKNECHT, Dragos VELICANU et Sophia YAKOUBOV. « Certcoin : A namecoin based decentralized authentication system 6.857 class project ». In : *Unpublished class project* (2014).

[Lar14] Daniel LARIMER. « Delegated proof-of-stake (dpos) ». In : *Bitshare whitepaper* (2014).

[MBH14] Mohamed Nidhal MEJRI, Jalel BEN-OTHMAN et Mohamed HAMDI. « Survey on VANET security challenges and possible cryptographic solutions ». In : *Vehicular Communications* 1.2 (2014), p. 53-66.

[SJP14] Priyanka SIROLA, Amit JOSHI et Kamlesh C PUROHIT. « An analytical study of routing attacks in vehicular ad-hoc networks (VANETs) ». In : *International journal of computer science engineering (IJCSE)* 3.4 (2014), p. 210-218.

[Woo+14] Gavin WOOD et al. « Ethereum : A secure decentralised generalised transaction ledger ». In : *Ethereum project yellow paper* 151.2014 (2014), p. 1-32.

[BAG15] Tamás BÉCSI, Szilárd ARADI et Péter GÁSPÁR. « Security issues and vulnerabilities in connected car systems ». In : *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*. IEEE. 2015, p. 477-482.

[DD15] Evan DUFFIELD et Daniel DIAZ. « Dash : A privacycentric cryptocurrency ». In : *GitHub URL : https ://github.com/dashpay/dash/wiki/Whitepaper* (2015).

[Kal+15] Harry A KALODNER et al. « An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design. » In : *WEIS*. Citeseer. 2015.

[Liu+15a] Yabing LIU et al. « An end-to-end measurement of certificate revocation in the web's PKI ». In : *Proceedings of the 2015 Internet Measurement Conference*. ACM. 2015, p. 183-196.

[Liu+15b] Yabing LIU et al. « An end-to-end measurement of certificate revocation in the web's PKI ». In : *Proceedings of the 2015 Internet Measurement Conference*. 2015, p. 183-196.

[PS15] Jonathan PETIT et Steven E SHLADOVER. « Potential cyberattacks on automated vehicles. » In : *IEEE Trans. Intelligent Transportation Systems* 16.2 (2015), p. 546-556.

[16] *1609.2-2016 - IEEE Standard for Wireless Access in Vehicular Environments*. ieee, 2016. URL : https://standards.ieee.org/standard/1609_2-2016.html.

[Ali+16] Muneeb ALI et al. « Blockstack : A global naming and storage system secured by blockchains ». In : *2016 {USENIX} Annual Technical Conference ({USENIX}{ATC} 16)*. 2016, p. 181-194.

[AG16] LM AXON et Michael GOLDSMITH. « PB-PKI : A privacy-aware blockchain-based PKI ». In : (2016).

[AVD16] Maria AZEES, Pandi VIJAYAKUMAR et Lazarus Jegatha DEBORAH. « Comprehensive survey on security services in vehicular ad-hoc networks ». In : *IET Intelligent Transport Systems* 10.6 (2016), p. 379-388.

[Cac+16] Christian CACHIN et al. « Architecture of the hyperledger blockchain fabric ». In : *Workshop on distributed cryptocurrencies and consensus ledgers*. T. 310. 2016, p. 4.

[Cor16] F CORELLA. « Implementing a PKI on a Blockchain ». In : *Pomcor. com* 25 (2016).

[Eur16] Commission EUROPEENE. *Blockchain Technologies*. Commission Europeene, 2016. URL : https://ec.europa.eu/digital-single-market/en/blockchain-technologies.

[Fan16] Banque de FANCE. *La blockchain ou « chaîne de blocs » est une technologie de stockage et de transmission d'informations.* Banque de Fance, 2016. URL : https://abc-economie.banque-france.fr/mot-de-lactu/blockchain.

[Lei+16] Benjamin LEIDING et al. « Authcoin : validation and authentication in decentralized networks ». In : *arXiv preprint arXiv :1609.04955* (2016).

[Opp16] Rolf OPPLIGER. *SSL and TLS : Theory and Practice*. Artech House, 2016.

[Pat16] Al-Sakib Khan PATHAN. *Security of self-organizing networks : MANET, WSN, WMN, VANET*. CRC press, 2016.

[Shb+16] Wazen SHBAIR et al. « A multi-level framework to identify HTTPS services ». In : avr. 2016. DOI : 10.1109/NOMS.2016.7502818.

[SKK16] Hyun Min SONG, Ha Rang KIM et Huy Kang KIM. « Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network ». In : *2016 international conference on information networking (ICOIN)*. IEEE. 2016, p. 63-68.

[Aug+17] Daniel AUGOT et al. « Transforming face-to-face identity proofing into anonymous digital identity using the bitcoin blockchain ». In : *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE. 2017, p. 25-2509.

[Al-17] Mustafa AL-BASSAM. « SCPKI : a smart contract-based PKI and identity system ». In : *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*. ACM. 2017, p. 35-40.

[Fra17] Blockchain FRANCE. *Qu'est-ce que la Blockchain ?* Blockchain France, 2017. URL : https://blockchainfrance.net/decouvrir-la-blockchain/c-est-quoi-la-blockchain/.

[Kia+17] Aggelos KIAYIAS et al. « Ouroboros : A provably secure proof-of-stake blockchain protocol ». In : *Annual International Cryptology Conference*. Springer. 2017, p. 357-388.

[Liu+17] Yuan LIU et al. « An identity management system based on blockchain ». In : *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE. 2017, p. 44-4409.

[MR17] Stephanos MATSUMOTO et Raphael M REISCHUK. « IKP : Turning a PKI around with decentralized automated incentives ». In : *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, p. 410-426.

[MDB17] Axel MOINET, Benoıt DARTIES et Jean-Luc BARIL. « Blockchain based trust & authentication for decentralized sensor networks ». In : *arXiv preprint arXiv :1706.01730* (2017).

[Nev+17] Gregory NEVEN et al. « Privacy-preserving attribute-based credentials in cooperative intelligent transport systems ». In : nov. 2017, p. 131-138. DOI : 10.1109/VNC.2017.8275631.

[Qin+17] Bo QIN et al. « Cecoin : A decentralized PKI mitigating MitM attacks ». In : *Future Generation Computer Systems* (2017).

[Rig+17] Giovanni RIGAZZI et al. « Optimized certificate revocation list distribution for secure V2X communications ». In : *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2017, p. 1-7.

[A M18] W. Chuang A. MELNIKOV Ed. *Internationalized Email Addresses in X.509 Certificates*. Rapp. tech. RFC 8398, 2018.

[Has18] Hamssa HASROUNY. « Trust management and security solutions for vehicular networks ». Theses. Institut National des Télécommunications, juil. 2018. URL : https://tel.archives-ouvertes.fr/tel-01892393/file/These-Hamssa-Hasrouni.pdf.

[Hou18] R. HOUSLEY. *Internationalization Updates to RFC 5280*. Rapp. tech. RFC 8399, 2018.

[Lou+18] Junjun LOU et al. « A blockchain-based key management scheme for named data networking ». In : *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE. 2018, p. 141-146.

[Pas+18] G PASAOGLU et al. « Insécurité et délinquance 2018 : premier bilan statistique ». In : (2018).

[RD18] Eric RESCORLA et Tim DIERKS. « The transport layer security (TLS) protocol version 1.3 ». In : (2018).

[Sal+18] Tara SALMAN et al. « Security services using blockchains : A state of the art survey ». In : *IEEE Communications Surveys & Tutorials* 21.1 (2018), p. 858-880.

[18] *The best web browsers*. Digital Trends, 2018. URL : https://www.digitaltrends.com/computing/best-browser-internet-explorer-vs-chrome-vs-firefox-vs-safari-vs-edge/.

[Yak+18] Alexander YAKUBOV et al. « A blockchain-based pki management framework ». In : *The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018*. 2018.

[Eur19a] EUROSTAT. « Criminal acts in Europe 2008-2017 ». In : (2019). URL : https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Crime_statistics&stable=0&redirect=no.

[Eur19b] EUROSTAT. « The french automotive industry, analysis and statistic ». In : *ANALYSIS STATISTICS 2019* (2019). URL : https://ccfa.fr/wp-content/uploads/2019/09/ccfa-2019-en-web-v2.pdf.

[HP19] Thomas HARDJONO et Alex PENTLAND. « Verifiable anonymous identities and access control in permissioned blockchains ». In : *arXiv preprint arXiv :1903.04584* (2019).

[LSP19] Leslie LAMPORT, Robert SHOSTAK et Marshall PEASE. « The Byzantine generals problem ». In : *Concurrency : the Works of Leslie Lamport*. 2019, p. 203-226.

[SLW19] Muhammad Sameer SHEIKH, Jun LIANG et Wensong WANG. « A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets) ». In : *Sensors* 19.16 (2019), p. 3589.

[Cen20] Internet Storm CENTER. *Certificate revoked per day*. 2020. URL : https://isc.sans.edu/crls.html.

[Mon20] Jean-Philippe MONTEUUIS. « Resilience by design & failures forecasting for a connected autonomous vehicle ». Theses. Institut Polytechnique de Paris, jan. 2020. URL : https://tel.archives-ouvertes.fr/tel-02513392.

[tri20] Hosting TRIBUNAL. *21+ SSL Statistics that Show Why Security Matters so Much*. 2020. URL : https://hostingtribunal.com/blog/ssl-stats/#content-3.

[ACE] ACEA. *ACEA ReportVehicles in use Europe 2019*.

[ETSa] TS ETSI. *103 097-v1.3.1 (2017-10)-Intelligent Transport Systems; Security header and certificate formats*.

[ETSb] TS ETSI. *ETSI TS 102 941 V1.2.1 (2018-05), Intelligent Transport Systems (ITS); Security; Trust and Privacy Management*.

[For] FORCEPOINT. « Digging into certificate revocation lists ». In : (). URL : https://www.forcepoint.com/blog/x-labs/digging-certificate-revocation-lists.

[ITS] ITS-CENCENELEC.EU. *CEN and ETSI deliver first set of standards for Co-operative Intelligent Transport Systems (C-ITS) : A key step towards connected cars in Europe*. URL : https://www.cencenelec.eu/news/press_releases/PublishingImages/ETSI_ITS_09_2012-670.jpg.

[Mul] MULTICHAIN. *Multichain*. URL : https://www.multichain.com/.

[Nam] NAMECOIN. *Namecoin*. URL : https://namecoin.org/.

[NOO] NOOSHU. « The impact of SSL certificate revocation on web performance ». In : (). URL : https://nooshu.github.io/blog/2020/01/26/the-impact-of-ssl-certificate-revocation-on-web-performance/.

[OMN] OMNET++. *OMNET++ : Discrete Event Simulator*. URL : https://omnetpp.org/.

[Ope] OPENSSL. *OpenSSL*. URL : https://www.openssl.org/.

[Pyt] PYTHON. *Python 2.7*. URL : https://www.python.org/download/releases/2.7/.

[Rus] Aaron RUSSELL. « How Do Browsers Handle Revoked SSL/TLS Certificates ? » In : (). URL : https://www.ssl.com/article/how-do-browsers-handle-revoked-ssl-tls-certificates/#firefox.

[SS] P SIVAKUMAR et Kunwar SINGH. « Privacy based decentralized Public Key Infrastructure (PKI) implementation using Smart contract in Blockchain ». In : *technical report* ().

[SUM] SUMO. *SUMO : Simulation of Urban Mobility*. URL : https://www.eclipse.org/sumo/.

[Vei] VEINS. *Veins : the open source vehicular network simulation framework*. URL : https://veins.car2x.org/.

[www] WWW.STATISTA.COM. « Share of new cars sold that are connected to the Internet worldwidefrom 2015 to 2025 ». In : URL : http://www.statista.com/statistics/275849/number-of-vehicles-connected-to-the-internet/.

[Fad08] Ahmad FADLALLAH. « IP traceback solutions for Internet attacks ». In : *Thèse Informatique et Réseaux, EDITE*. Télécom Paristech. 6 février 2008.

[Llo05] Cédric LLORENS. « Mesure de la sécurité logique d'un Réseau d'opérateur Multiservices ». In : *Thèse Informatique et Réseaux, EDITE*. Télécom Paristech. Novembre 2005.

## Annexe A

# Publications

1. Adja elloh yves christian, B. Hammi, A. Serhrouchni, H. Labiod. “Total eclipse : How To Completely Isolate a Bitcoin Peer”, 3rd International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC’2018), Shanghai, China,

2. B. Hammi, Adja elloh yves christian. "Fake Check Scams : A Blockchain Based Detection Solution", 9th International Conference on Computer Science and Information Technology (CCSIT 2019)
3. Yves Christian Elloh Adja, Badis Hammi, Ahmed Serhrouchni, Sherali Zeadally, A blockchain-based certificate revocation management and status verification system,Computers Security, Volume 104, 2021,102209,ISSN 0167-4048

4. Adja elloh yves christian, Ahmed Serhrouchni. "Improved CRL Distribution point (ICRLDP) for Cooperative Intelligent Transportation Systems (C-ITS)", IFIP/IEEE International Symposium on Integrated Network Management (IM 2021), Bordeaux, France, May 18-20, 2021)

# Annexe B

# Les filtres de Bloom

Le filtre de bloom est une structure de données probabiliste qui permet de tester l'appartenance d'un élément à un ensemble de données. Un groupe de données $S = (x_1, x_2, x_3, ..., x_n)$ de n éléments est décrit à travers un filtre de bloom par un tableau de m bits initialisé a 0 (B.1). Le filtre utilise k fonctions de hachage $h_1, ..., h_k$

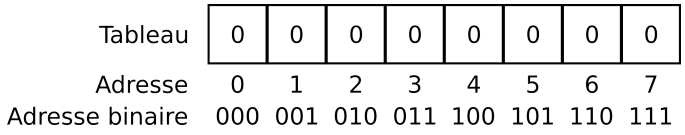



FIGURE B.1 – Bloom filter

pour déterminer la position des k bits à mettre à 1. Pour insérer un élément dans le filtre, il faut mettre à 1, un bit dans une position défini. Precisément, la donnée doit être haché avec k fonctions de hachage qui vont déterminer chacun un bit à mettre à 1. Nous pouvons prendre l'exemple avec la donnée "alice" et k=2, ce qui va produire (B.2). Donc chaque élément de l'ensemble sera représenté par $h_k$ positions dans un

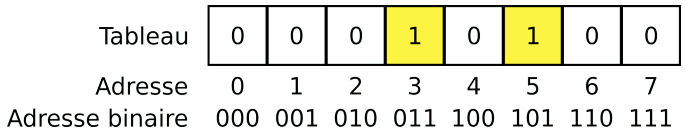



FIGURE B.2 – Bloom filter

tableau mis à 1. Pour vérifier l'appartenance d'une donnée à un ensemble il faudra retrouver la position du bit représentant la donnée dans le tableau et contrôler sa valeur. Le filtre de bloom n'admet pas de faux négatifs, mais des faux positifs (une réponse positive peut être donnée pendant que la donnée n'appartient pas à l'ensemble), parceque si le tableau n'est pas assez grand, il peut arriver que plusieurs données occupent la même position dans le filtre. Le filtre de bloom est caractérisé par, la dimension du filtre, le nombre de fonctions de hachage, le nombres d'éléments contenu dans l'ensemble et la probabilité de faux positifs.

ECOLE DOCTORALE

**Titre :** Définition et conception d'un système optimal de la révocation de certificats

**Mots clés :** PKI, Blockchain, CRL, V2X, Revocation, Certificate, x509, VPKI

**Résumé :** La Public Key Infrastructure(PKI) est une infrastructure à clés publique dont l'objectif est de répondre à des besoins tel que : l'authentification de clés publiques, le contrôle d'accès et les fonctions d'autorisation, l'identification et l'authentification déterministe et automatisée. La prise en charge de ces services détermine les attributs contenus dans le certificat, ainsi que les informations de contrôle auxiliaires telles que la politique et les contraintes de chemin de certification. La validation du certificat passe par la vérification de ces attributs. Un certificat délivré par une autorité de certification est censé être utilisé pendant toute sa période de validité. Cependant, diverses circonstances peuvent interrompre cette validité. Ces circonstances comprennent le changement de nom, le changement d'association et etc.. Sous telles circonstances, l'autorité de certification doit révoquer le certificat. Les services de validation et de vérification de la révocation des certificats sont attendus ou nécessaires dans plusieurs contextes, nous pouvons en citer certains parmi tant d'autres: les communications véhiculaires, le (WWW), l'authentification des utilisateurs et etc. Les communications véhiculaires sont au centre des véhicules de demain et d'une manière plus générale des smart-cities. La sécurisation de ces réseaux est un élément critique au vu des services en perspective. La sécurité des échanges inter-véhicules est basée notamment sur la signature numérique. Cette même signature nécessite d'une infrastructure d'échange de clés (PKI). Le RFC 5280 défini différents méthodes de révocation. Parmi celles-ci nous avons une méthode de révocation qui implique la publication périodique de la part de la CA d'une structure de données signée appelée CRL(Certificate révocation list). Cette approche, qui est la plus utilisée, est mature mais couteuse en temps et en volume et les communications véhiculaires apportent de nouvelles contraintes. Ces travaux portent sur une solution d'optimisation de la révocation des certificats X509 et à pseudonyme utilisés dans les communications véhiculaires. Notre objectif est de réduire en terme de latence et de volume de données l'impact de la révocation des certificats sur la validation des certificats X509 et pseudonymes. Nous avons dans ce contexte proposé une méthode de révocation intégrant la Blockchain et les filtres de bloom à la gestion de la révocation dans la PKI X509. Nous avons ensuite proposé une méthode de révocation basée sur des choix cryptographiques classiques pour réduire l'impact de la révocation sur la validation des certificats pseudonymes. Ces deux approches proposées permettent un gain en terme de volume, de latence et de résilience tout en maintenant un niveau de sécurité égal voire supérieur aux solutions classiques. Nos contributions ont été validé par une implémentation.

**Institut Polytechnique de Paris**
91120 Palaiseau, France

ECOLE DOCTORALE

**Title :** Definition and design of an optimal certificate revocation system

**Keywords :** PKI, Blockchain, CRL, V2X, Revocation, Certificate, x509, VPKI

**Abstract :** The Public Key Infrastructure (PKI) is a public key infrastructure whose objective is to meet needs such as public key authentication, access control and authorization functions. The support for these services determines the attributes contained in, as well as auxiliary control information such as policy and certification path constraints. The validation of the certificate goes through the verification of these attributes. A certificate is supposed to be used throughout its validity period. However, various circumstances may interrupt this validity. These circumstances include a name change, an association change, etc. Under such circumstances, the CA must revoke the certificate. Certificate revocation validation and verification services are expected or necessary in several contexts, we can cite some among many others: vehicular communications, (WWW), user authentication, etc. Vehicle communications are at the center of the vehicles of tomorrow and, more generally, of smart cities. Securing these communications is a critical element given the services in perspective. The security of inter-vehicle exchanges is based on the digital signature. This signature requires a key exchange infrastructure (PKI). The RFC 5280 defines different revocation methods. Among these, we have a revocation method, which requires the periodic publication by the CA of a signed data structure called CRL (Certificate revocation list). This work relates to a solution for optimizing the revocation of X509 and pseudonym certificates used in vehicular communications. Our goal is to reduce in terms of latency and data volume the impact of certificate revocation on X509 and vehicles pseudonyms certificates validation. In this context, we have proposed a revocation method integrating the Blockchain and bloom filters to the management of revocation in the PKI X509. We then proposed a revocation method based on classic cryptographic choices to reduce the impact of revocation on the validation of pseudonymous certificates. These two proposed approaches allow a gain in terms of volume, latency and resilience while maintaining a level of security equal or even greater than conventional solutions. Our contributions have been validated by an implementation.

**Institut Polytechnique de Paris**
91120 Palaiseau, France