Du secret à la confiance... quelques éléments de cryptographie

Colin, Jean-Noël

Published in:
L'identification électronique et les services de confiance depuis le règlement eIDAS

Publication date:
2016

Document Version
le PDF de l'éditeur

Du secret à la confiance... quelques éléments de cryptographie

Jean-Noël Colin*

Introduction

Le règlement 910/2014 du Parlement européen et du Conseil sur *l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur* vise à instaurer un cadre législatif destiné à développer la confiance dans les transactions électroniques auprès des citoyens et autres acteurs, économiques ou administratifs, privés ou publics, et par là à promouvoir un recours de plus en plus large à des services numériques.

Entre autres aspects, le règlement suggère divers instruments technologiques, tels que la mise en place de services d'identification électronique au sein des États-membres de l'Union européenne et leur reconnaissance à un niveau transnational, ainsi que le développement de mécanismes d'authentification et de sécurisation des documents et transactions électroniques, dont la validité pourrait être attestée par une infrastructure de confiance. La mise en œuvre de tels instruments ne peut s'envisager que dans une perspective d'utilisation de standards largement adoptés, prouvés et supportés par une large communauté, et d'interopérabilité.

La cryptographie offre un certain nombre de primitives permettant de rencontrer différents critères de sécurité, tels que la confidentialité, l'intégrité ou l'authenticité d'une information, et par extension, d'une personne ou d'un système. En tant que tel, elle offre donc les outils de base nécessaires à la mise en œuvre des mécanismes préconisés.

Cette contribution s'attache à décrire les fondements de la cryptographie : chiffrement et déchiffrement, empreinte numérique et signature numérique ; il assemble ensuite ces éléments de base pour élaborer des solutions plus complexes, tels qu'un service de confiance ou un schéma d'identification.

* Professeur à l'Université de Namur.

CHAPITRE I. Éléments de base de cryptographie

De tout temps, l'homme a cherché à préserver la confidentialité de certaines informations ; on trouve des traces de méthodes visant à garantir le secret dans les civilisations égyptienne (usage de hiéroglyphes non reconnus), romaine (chiffrement de César) ou grecque (usage de la scytale) [3] ; en Inde, le kama-sutra recommande aux femmes de maîtriser entre autres arts, celui de l'écriture secrète (*Mlecchita-Vikalpa*) [3]. Au fil des siècles, de nouvelles méthodes de chiffrement ont été tour à tour élaborées, analysées et mises à mal pour finalement être abandonnées en raison du faible niveau de sécurité offert. C'est avec l'essor des communications modernes (télégraphe, téléphone, réseaux informatiques) que le besoin de secret est véritablement devenu vital et que la cryptographie s'est considérablement développée. 1970 est généralement reconnu comme le début de la cryptographie moderne, avec l'avènement de méthodes de chiffrement telles que Lucifer ou DES, et les bases de la cryptographie asymétrique [3]. Avant de poursuivre dans ce chapitre, il convient de préciser certains éléments :

- les méthodes décrites dans la suite de ce document sont pour la plupart destinées à être exécutées sur un ordinateur ou autre dispositif informatique. À cet égard, il est utile de rappeler que toute information traitée, stockée ou échangée par un ordinateur l'est sous forme binaire, soit une séquence d'unités d'information élémentaires, appelés bits (pour binary digit), ne pouvant prendre qu'une des deux valeurs : 0 ou 1. Un byte (ou octet) est un ensemble de 8 bits. C'est le choix d'une méthode de codage qui permet de donner du sens à ces séquences binaires, qui peuvent représenter indifféremment des nombres, des caractères d'alphabets variés, ou encore des images, sons ou toute autre forme de données. Citons l'exemple du code ASCII qui établit une correspondance entre les caractères alpha-numériques[1] et leur représentation sous forme de nombres ; dans ce code, la lettre 'A' est représentée par le nombre 65, qui peut être converti en notation binaire 01000001. Au sein de l'ordinateur, tout est donc soit 1, soit 0, et traité comme tel.
- la cryptographie moderne s'appuie sur des méthodes décrites par des algorithmes (*cipher* en anglais) et font pour la plupart usage de clés ; les algorithmes sont le plus souvent publics, ce qui permet leur validation par une large communauté de cryptanalystes et le renforcement de la confiance des utilisateurs ; par contre, il est impératif que les clés utilisées restent connues des seules personnes concernées, car c'est de ce secret que dépendra la sécurité de l'ensemble du système. Ceci découle du principe de Kerckhoffs, qui énonçait au 19ème siècle qu'un procédé cryptographique doit rester sûr même si tout est connu à son sujet, sauf la clé.
- les termes *numérique* et *digital* sont souvent utilisés de manière interchangeable pour désigner ce qui est représenté par des nombres et traité comme tel. On parlera ainsi de signature digitale ou numérique. Le sujet fait débat[2], le terme *digital* étant considéré comme un anglicisme ; dans la suite de cet article, nous utiliserons le terme numérique, tout en soulignant qu'il est souvent remplacé par son anglicisation.

[1] Pour être exact et complet, le code ASCII étendu définit un code numérique sur 8 bits permettant de représenter 256 caractères différents : lettres majuscules et minuscules, chiffres, signes de ponctuation, caractères accentués, caractères graphiques ainsi que certains caractères de contrôle utilisés à des fins techniques de traitement et de communication.

SECTION 1. – Chiffrement et déchiffrement

Le chiffrement d'une donnée consiste à transformer cette donnée, lisible, en une donnée inintelligible, à l'aide d'une méthode et d'une clé de chiffrement. L'opération inverse, le déchiffrement, consiste à reconstituer la donnée initiale à partir de la donnée chiffrée, en utilisant une méthode et une clé de déchiffrement.

Une clé est une séquence de bits. Sa longueur est définie par l'algorithme choisi. Par exemple, l'algorithme DES utilise des clés de 56 bits, tandis que l'algorithme AES utilise des clés de 128 bits.

Plus formellement, si l'on désigne par m, le message à chiffrer, c, le message chiffré, E, l'algorithme de chiffrement, D, l'algorithme de déchiffrement, K_e et K_d, les clés de chiffrement et déchiffrement, respectivement, on écrira :

- $c = E(K_e, m)$, le texte chiffré c est le résultat du chiffrement du texte clair m à l'aide de l'algorithme E et de la clé K_e
- $m = D(K_d, c)$, le texte clair m peut être retrouvé à partir du texte chiffré c à l'aide de l'algorithme D et de la clé K_d

Un algorithme de chiffrement sûr doit produire un résultat impossible à distinguer d'une production purement aléatoire, ceci afin de ne livrer aucune information quant au texte clair à la simple lecture du texte chiffré.

[2] http://www.academie-francaise.fr/digital ou http://www.lemonde.fr/economie/article/2014/01/14/dilemme-numerique_4347680_3234.html

Ces définitions montrent que le déchiffrement nécessite la connaissance de la clé de déchiffrement. Le secret de l'information à protéger dépend donc du secret de cette clé, qui ne peut être connue que des personnes autorisées. L'on pourrait être tenté d'essayer toutes les clés possibles pour déchiffrer le texte clair. Cependant, considérant qu'une taille clé de n bits permet de définir 2^n clés possibles[3], l'utilisation de clés suffisamment longues permet de rendre impossible cette recherche exhaustive, en tout cas dans un temps raisonnable.

Les sections qui suivent présentent les deux catégories de méthodes de chiffrement : la cryptographie symétrique et la cryptographie asymétrique.

§ 1. Cryptographie symétrique

La cryptographie symétrique, aussi appelée cryptographie à clé secrète, définit des méthodes de chiffrement et déchiffrement comme décrit ci-dessus, avec la particularité que la clé de déchiffrement est identique à la clé de déchiffrement. Autrement dit :

- $c = E(K, m)$, le texte chiffré c est le résultat du chiffrement du texte clair m à l'aide de l'algorithme E et de la clé K
- $m = D(K, c)$, le texte clair m peut être retrouvé à partir du texte chiffré c à l'aide de l'algorithme D et de la même clé K

DES (Data Encryption Standard) ou AES (Advanced Encryption Standard) sont des exemples d'algorithmes de chiffrement symétrique, qui ont été adoptés comme standards par le gouvernement américain, et plus largement par la communauté internationale ; l'usage du DES (et ses variantes) est vivement déconseillé aujourd'hui, en raison de la taille limitée de la clé (56 bits) et de la relative lenteur de l'algorithme. Il est recommandé d'utiliser AES, qui utilise des clés de 128, 192 ou 256 bits.

Dans la cryptographie symétrique, une même clé K est utilisée pour chiffrer le texte clair et déchiffrer le texte chiffré. Tout échange de message chiffré présuppose donc le partage d'une clé commune entre l'émetteur et le destinataire du message. Cette gestion des clés pose différents problèmes :

- une clé doit être produite par un procédé aléatoire, afin d'empêcher un adversaire de la retrouver plus efficacement que par une recherche exhaustive. Cela suppose donc que soit les correspondants sont capables de mettre en œuvre le même procédé et donc disposer de la même clé, soit que l'un des deux est chargé de générer la clé et de la communiquer à son correspondant[4]. Cela requiert donc des mécanismes sûrs de génération, de transmission et de stockage de clés ; ceci est d'autant plus important que c'est de la clé que dépend la sécurité du crypto-système.
- l'émetteur doit disposer d'autant de clés que de destinataires, afin d'envoyer des textes chiffrés lisibles uniquement par le destinataire visé ; dans un réseau de n correspondants, chacun doit donc disposer au minimum de $n-1$ clés différentes, et ce uniquement pour les échanges bilatéraux.

La cryptographie symétrique offre donc des moyens puissants pour garantir la confidentialité des données, mais leur mise en œuvre s'accompagne de difficultés importantes liées au nombre de clés à gérer, et au fait que les clés doivent être partagées, ce qui augmente considérablement leur surface d'attaque.

§ 2. Cryptographie asymétrique

La cryptographie asymétrique s'appuie sur une approche différente de la gestion des clés. Elle utilise deux clés différentes pour le chiffrement et le déchiffrement ; comme décrit plus haut :

- $c = E(K_e, m)$, le texte chiffré c est le résultat du chiffrement du texte clair m à l'aide de l'algorithme E et de la clé K_e
- $m = D(K_d, c)$, le texte clair m peut être retrouvé à partir du texte chiffré c à l'aide de l'algorithme D et de la clé K_d

Ici, K_e est différent de K_d ; ce qui est chiffré avec K_e ne peut être déchiffré qu'avec K_d, et réciproquement. Bien que K_e et K_d soient liées par une relation mathématique, il est pratiquement impossible[5] de déduire l'une à partir de l'autre.

Le principe de base est donc de doter chaque utilisateur d'une paire de clés, K_e et K_d ; l'une est sa clé *privée* ou clé *secrète*, et reste donc connue de lui seul, tandis que l'autre est sa clé *publique*, pouvant être distribuée. C'est de cet usage des clés que la cryptographie asymétrique est aussi appelée cryptographie à clé publique.

Alice dispose donc de KS_A et KP_A, et Bob de KS_B et KP_B, leurs clés secrètes et publiques respectivement. Pour envoyer un message à Bob de manière

[3] À titre d'exemple, l'algorithme AES utilise des clés de 128 bits, soit 2^{128} clés possibles, ou approximativement $3{,}4 \times 10^{38}$.

[4] D'autres approches existent, telles que les protocoles de négociation de clé (Diffie-Hellman) ou le recours à un tiers de confiance (Needham-Schroeder par exemple).

[5] Impossible au sens de la calculabilité, c'est-à-dire dans un temps raisonnable (*'computationally unfeasible'*).

à ce que seul celui-ci soit capable de le lire, il faut donc qu'Alice chiffre ce message avec la clé publique de Bob KP_B, qu'elle connaît, par définition. Le message ne pouvant être déchiffré qu'à l'aide de la clé secrète de Bob, KS_B, connue de lui seul, la confidentialité du message est donc garantie.

Là où la cryptographie symétrique utilise des procédés basés sur la manipulation des bits composant le message, la cryptographie asymétrique utilise la théorie des nombres comme base de ses algorithmes et s'appuie sur des problèmes complexes à résoudre qui nécessiteraient un temps de calcul tel qu'il est considéré comme impraticable aujourd'hui, et garantit ainsi la sécurité des procédés de chiffrement et déchiffrement. Les problèmes communément utilisés sont la factorisation des nombres, les logarithmes discrets et plus récemment, les courbes elliptiques, qui permettent d'utiliser des clés moins longues tout en garantissant le même niveau de sécurité. On parle aussi de fonctions à sens unique avec trappe, c'est-à-dire de fonctions mathématiques qu'il est aisé de calculer, mais qu'il est difficile d'inverser, sauf à connaître un élément particulier (trappe), ici, la clé secrète.

L'algorithme RSA, du nom de ses inventeurs[6], est l'un des principaux algorithmes de chiffrement asymétrique, qui s'appuie sur le problème de la factorisation de grands nombres. El Gamal est un autre algorithme de chiffrement asymétrique qui se base sur le problème des logarithmes discrets. Il en existe de nombreux autres, d'usage plus ou moins répandu.

Il est intéressant de noter que lorsqu'Alice envoie un message à Bob, elle peut le chiffrer avec la clé publique de celui-ci (KP_B) pour assurer sa confidentialité ; mais elle peut aussi chiffrer le message avec sa propre clé secrète (KS_A) ; ce message chiffré ne peut en aucun cas être considéré comme confidentiel car tout correspondant en possession de la clé publique d'Alice (KP_A) est en mesure de le déchiffrer. Cependant, le message peut être considéré comme authentique, car seule Alice a pu le chiffrer avec sa clé secrète (KS_A) qu'elle est seule à connaître. Cette propriété sera exploitée plus loin pour réaliser une signature numérique.

SECTION 2. – Empreinte numérique

L'empreinte numérique d'une donnée est une valeur calculée à partir de cette donnée au moyen d'une fonction de hachage[7] à sens unique. Une telle fonction calcule à partir d'une donnée de taille arbitraire son empreinte[8] sous la forme d'un résultat unique, de longueur fixe, d'apparence aléatoire, et à partir duquel il est impossible[9] de reconstituer la donnée originale. Une modification même infime de celle-ci a un impact sur l'ensemble de l'empreinte.

[6] Ron Rivest, Adi Shamir et Leonard Adleman.

[7] De l'anglais, *hash*. On appelle aussi une telle fonction, fonction de compression.

[8] Condensat, digest en anglais.

Il est important de noter qu'une fonction de hachage n'apporte aucun secret. Son fonctionnement est publiquement décrit. Son intérêt réside dans son caractère à sens unique, qui permet de définir un équivalent compressé à la donnée originale.

À titre d'exemple, voici l'empreinte calculée par la méthode SHA-1 du texte *'Quousque tandem abutere, Catilina, patientia nostra ?'* :

- a081ccb6070a9a4cf5137c8bff6ecad80a6a50d9

Par comparaison, l'empreinte du texte *'Quousque tandem abutere, Catilina, patientia nostra !'*, identique au précédent à un signe de ponctuation près, est :

- 6be608c1cc7a1f63a04508967d4d98cd0882adcd

L'empreinte ayant une taille fixe, le nombre d'empreintes différentes est aussi fixe ; par ailleurs, le nombre de données dont on peut calculer l'empreinte est infini, ce qui induit l'existence de collisions, à savoir la possibilité que deux données aient la même empreinte.

Plus formellement, une fonction de hachage H, calcule à partir d'une donnée m son empreinte h, telle que :

- $h = H(m)$
- m est de taille quelconque
- h est de taille fixe, définie par la fonction de hachage
- connaissant h, il est impossible[10] de retrouver m tel que $h = H(m)$
- de même, il est impossible[10] de trouver une collision, à savoir deux messages m_1 et m_2 tels que $H(m_1)=H(m_2)$

Il existe de nombreuses fonctions de hachage, notamment MD5 (empreinte de 128 bits, soit 2^{128} empreintes possibles), SHA-1 (Secure Hash, empreinte de 160 bits, soit 2^{160} empreintes possibles), SHA-2 (empreinte de 224, 256, 384 ou 512 bits) ou SHA-3 (empreinte de 224, 256, 384 ou 512 bits). Aujourd'hui, l'usage des algorithmes MD5 et SHA-1 est déconseillé en raison de faiblesses détectées ; une évolution vers SHA-2 ou SHA-3 est recommandée, ce dernier étant le fruit de la dernière sélection en 2012 par le National Institute of Science and Technology américain.

Un usage de l'empreinte numérique est de vérifier l'intégrité d'une donnée. Supposons qu'Alice souhaite vérifier qu'une donnée en sa possession

[9] Au sens de la calculabilité.

[10] Au sens de la calculabilité.

est identique à celle détenue par Bob, sans pour autant échanger cette donnée. Il lui suffit de demander à Bob de lui transmettre l'empreinte de la donnée qu'il détient et de la comparer à celle qu'elle possède. Si les empreintes correspondent, c'est que les données elles-mêmes sont identiques.

SECTION 3. – Signature numérique

Le chiffrement d'un message permet de garantir la confidentialité de celui-ci, en limitant l'accès aux seuls détenteurs de la clé de déchiffrement. L'empreinte numérique permet d'en définir une forme compressée, unique, pratiquement impossible à forger, et qui peut être utilisée pour garantir l'intégrité du message.

Dans certaines situations, il est non seulement nécessaire d'assurer la confidentialité et l'intégrité d'un message, mais aussi son authenticité, autrement dit son origine. Lorsque cette origine peut être établie de manière irréfutable, on parle alors de non-répudiation.

La signature numérique vise à atteindre cet objectif. Elle se base sur les mécanismes présentés précédemment : cryptographie asymétrique et empreinte.

Supposons qu'Alice envoie un message à Bob, et souhaite que ce dernier soit sûr de l'origine du message. Comme indiqué à la section 1, § 2, Alice peut chiffrer le message avec sa clé secrète ; Bob déchiffrera alors le message avec la clé publique d'Alice et si le déchiffrement réussit, il obtiendra le message original avec la garantie qu'il a bien été chiffré par Alice, elle seule connaissant sa clé secrète. Cette approche souffre cependant d'un défaut majeur : la cryptographie asymétrique, s'appuyant sur des opérations mathématiques complexes, est peu appropriée pour le chiffrement de volumes de données importants.

Dès lors, plutôt que chiffrer son message avec sa clé secrète, Alice va d'abord calculer l'empreinte numérique de son message, et chiffrer cette empreinte à l'aide de sa clé secrète ; l'empreinte étant de taille très réduite (quelques bytes), elle peut être chiffrée rapidement, sans être exagérément pénalisée par les faibles performances de la cryptographie asymétrique. Alice calcule donc $sig_A = E(KS_A,H(m))$, et transmet son message accompagné de sa signature (m, sig_A) à Bob.

Pour vérifier la signature apposée au message, Bob commence par calculer l'empreinte du message reçu : $h'=H(m)$. Il déchiffre ensuite la signature d'Alice à l'aide de la clé publique de celle-ci et obtient $h''=D(KP_A,sig_A)$.

Il compare enfin h' et h'', et en cas d'égalité, il obtient la garantie que le message est bien intègre et authentique.

En effet, si le message avait été modifié durant la transmission, l'empreinte calculée par Bob serait inévitablement différente de celle produite par Alice. Et la clé secrète d'Alice étant connue d'elle seule, personne ne peut avoir chiffré l'empreinte à sa place. La signature est donc bien authentique.

Au-delà de l'authenticité, ce mécanisme de signature numérique présente d'autres propriétés intéressantes :

- la signature est infalsifiable : seule Alice peut chiffrer avec sa clé secrète ;
- la signature ne peut être copiée pour authentifier un autre message : tout message différent du message original a nécessairement une empreinte différente, qui donnera lieu à une signature différente ;
- la signature rend le document inaltérable : toute modification au document modifierait son empreinte et invaliderait sa signature ;
- la signature est irrévocable : Alice étant seule en possession et responsable de sa clé secrète, nul autre ne pourrait avoir produit cette signature.

L'algorithme DSA (Digital Signature Algorithm) est un mécanisme de signature défini dans le Digital Signature Standard du FIPS américain en 1994, et réactualisé en 1998, 2000, 2009 et 2013 [4] ; il utilise la fonction de hachage SHA-1 pour le calcul de l'empreinte, l'authenticité est établie par une méthode basée sur les logarithmes discrets. D'autres méthodes utilisent SHA-1 (ou l'un de ses successeurs) avec un chiffrement RSA. Dans les deux cas, le principe reste identique.

Il est à noter qu'il existe certaines méthodes de signature numérique basées sur la cryptographie symétrique, et donc l'usage de clés partagées. Un exemple est HMAC (Hash Message Authentication Key) ; dans ce cas, plutôt que calculer l'empreinte sur base du message uniquement, Alice la calcule sur une fonction du message et de la clé partagée avec Bob : $h=H(f(K,m))$. Elle transmet à Bob le message et l'empreinte (m,h) ; Bob calcule lui-même l'empreinte du texte reçu et de la clé partagée $(h'=H(f(K,m))$, et valide l'authenticité du message en cas de correspondance avec l'empreinte reçue $(h=h')$.

CHAPITRE II. Service de confiance

Les mécanismes présentés au chapitre 2 permettent d'assurer la sécurité d'une information vis-à-vis de critères de sécurité tels que la confidentialité, l'intégrité, l'authenticité et la non-répudiation. Ils constituent les

fondations sur lesquelles s'appuient des fonctionnalités plus avancées, tels que la certification de clés, base de l'authentification de sites internet, l'horodatage ou le cachet électronique.

SECTION 1. – Certificat numérique et infrastructure à clés publiques

Comme nous l'avons indiqué, une clé n'est jamais qu'une suite de bits dans le cas de la cryptographie symétrique, ou d'un ou plusieurs nombres dans le cas de la cryptographie asymétrique, ces nombres étant eux-mêmes représentés dans un ordinateur comme une séquence binaire. Par conséquent, il est impossible en observant une clé de pouvoir déterminer à quelle entité, personne ou système, elle appartient. Ce défaut ouvre la porte à différentes attaques telles que l'usurpation d'identité ou une attaque Man-in-the-Middle, où un adversaire malicieux intercepte les échanges entre deux parties et les modifie à son avantage.

Dans un contexte de cryptographie asymétrique, il est donc indispensable pour instaurer une confiance dans les échanges, et donc entre les parties communicantes, de disposer d'un mécanisme de certification de clés, par lequel le lien entre une clé et l'identité de son propriétaire est établi de manière incontestable. En effet, selon le mécanisme de signature numérique décrit plus haut, un document est authentifié par le chiffrement de son empreinte à l'aide de la clé privée de son émetteur ; si la signature peut être validée (déchiffrée) à l'aide d'une clé publique dont le lien avec l'émetteur a été certifié, la confiance dans l'acte de signature est totale. La certification de la propriété d'une clé est au cœur des services offerts par une Infrastructure à Clés Publiques (PKI – Public Key Infrastructure) [8].

Une infrastructure à clés publiques est organisée autour d'une autorité de certification (AC), ensemble d'éléments humains, techniques et organisationnels, qui délivre des *certificats numériques* attestant du lien irréfutable entre une clé[11] et son propriétaire. Un tel certificat peut ensuite être utilisé par le propriétaire de la clé ou toute autre personne pour communiquer la clé ; le certificat étant émis par une autorité de confiance, nul ne contestera le lien entre la personne et sa clé. La confiance qu'un utilisateur peut avoir dans un certificat dérive directement de celle qu'il a dans l'autorité de certification émettrice.

[11] Il s'agit bien évidemment ici d'une clé publique, une clé secrète n'ayant pas vocation à être dévoilée.

§ 1. Nature du certificat

Les certificats les plus utilisés sont les certificats conformes au standard X.509 [6]. Pour avoir du sens et pouvoir être validé, un certificat doit au minimum être porteur des informations suivantes :

- le nom de l'émetteur ;
- le nom du sujet, propriétaire de la clé ;
- la clé publique du sujet ;
- la période de validité de certification ;
- la signature numérique du certificat par l'émetteur.

La norme X.509 définit trois versions du format de certificat, chacune venant compléter les versions précédentes en ajoutant des informations supplémentaires pouvant y être intégrées.

Le tableau 1 présente les différentes informations définies par chacune des versions du standard. La version du certificat définit l'information qu'il contient, le numéro de série permet de l'identifier parmi les certificats émis par l'AC, l'algorithme de signature spécifie la méthode utilisée pour produire la signature numérique du certificat (*cf.* Chapitre 2, Section 3). Les deux identifiants uniques ont été introduits dans la version 2 du standard afin de rendre l'identification du sujet et de l'émetteur non-ambigüe ; ils sont cependant fort peu utilisés. Le mécanisme d'extension introduit dans la version 3 permet de compléter le contenu du certificat avec des informations telles que les usages autorisés de la clé, des dénominations alternatives pour l'émetteur et le sujet, un lien vers une politique d'usage de la clé ou encore un pointeur vers une liste de certificats révoqués (*cf. infra*).

	v1	v2	v3
Version	✓	✓	✓
Numéro de série	✓	✓	✓
Algorithme de signature	✓	✓	✓
Émetteur	✓	✓	✓
Période de validité	✓	✓	✓
Sujet	✓	✓	✓
Clé publique du sujet	✓	✓	✓
Identifiant unique du sujet		✓	✓
Identifiant unique de l'émetteur		✓	✓
Extensions			✓
Signature	✓	✓	✓

Table 1. Attributs d'un certificat X.509

La question du nommage dans un certificat, tant pour le sujet que pour l'émetteur, est cruciale. Comment en effet attribuer un certificat à Jean Dupond, sans disposer d'information complémentaire permettant d'identifier, sans ambiguïté possible, la personne dont il s'agit ? Pour ce faire, on a recours généralement à une notation inspirée du standard X.500 qui adopte une approche hiérarchique, basée sur une découpe géographique et organisationnelle (pays, état, localité, organisation, département). Cette désignation peut être complétée par toute autre information pertinente. Ainsi, les certificats délivrés par l'autorité officielle belge, présents sur la carte d'identité nationale, intègrent le numéro de registre national comme élément d'identification, rendant ainsi le nommage unique.

Il est important de noter que l'information incluse dans un certificat est réputée validée par l'autorité émettrice. Il y a donc une réelle relation de confiance qui s'instaure entre l'utilisateur du certificat et l'AC qui l'a signé, le premier accordant crédit à l'information dont le certificat est porteur parce qu'il fait confiance à l'AC. Il est donc vital pour le bon fonctionnement de cet écosystème qu'une AC valide les informations incluses dans les certificats qu'elle émet. À cet égard, on distingue différentes classes de certificats qui se différencient par le degré de contrôle exercé par l'AC sur les informations fournies par le sujet du certificat (contrôle limité de propriété d'adresse email ou de nom de domaine internet, validation de l'identité à distance par transmission de documents ou validation de l'identité par présentation en personne du demandeur).

Les certificats X.509 sont à la base du protocole SSL/TLS sur lequel s'appuie la sécurité du Web, en particulier le protocole https. Lorsqu'un navigateur se connecte sur un site Web selon ce protocole, il obtient le certificat du serveur, ce qui lui permet de vérifier son identité[12]. Par exemple, lorsque l'on se connecte sur https://www.google.be et que l'on observe le certificat renvoyé par ce site, on obtient l'information présentée à la figure 1.

On y retrouve l'identité du sujet (google.com)[13], de l'émetteur (Google Internet Authority G2), sa période de validité, et son empreinte, calculée selon deux méthodes différentes[14]. En consultant l'onglet 'Détails', on observe que l'algorithme de signature est 'SHA-256 With RSA Encryption'

[12] Chaque internaute peut vérifier les informations décrites ici par lui-même ; tous les navigateurs récents permettent d'accéder aux paramètres de sécurité de la page en cours de visite, le plus souvent en cliquant sur un petit cadenas dans la barre de navigation.

[13] Bien que l'on ait accédé au site www.google.be, on constate que le certificat est émis au nom de www.google.com, sans que cela ne pose de problème au navigateur. Cela est dû au fait que www.google.be est repris dans les noms alternatifs du sujet (DNS Name : *.google.be).

[14] Il est recommandé d'abandonner l'algorithme SHA-1 au profit de l'un de ses successeurs. Le certificat n'inclut cette empreinte que pour des raisons de compatibilité.

En analysant le certificat plus en détail, on trouve les dénominations suivantes pour le sujet et l'émetteur, où l'on retrouve la notation hiérarchique décrite plus haut :

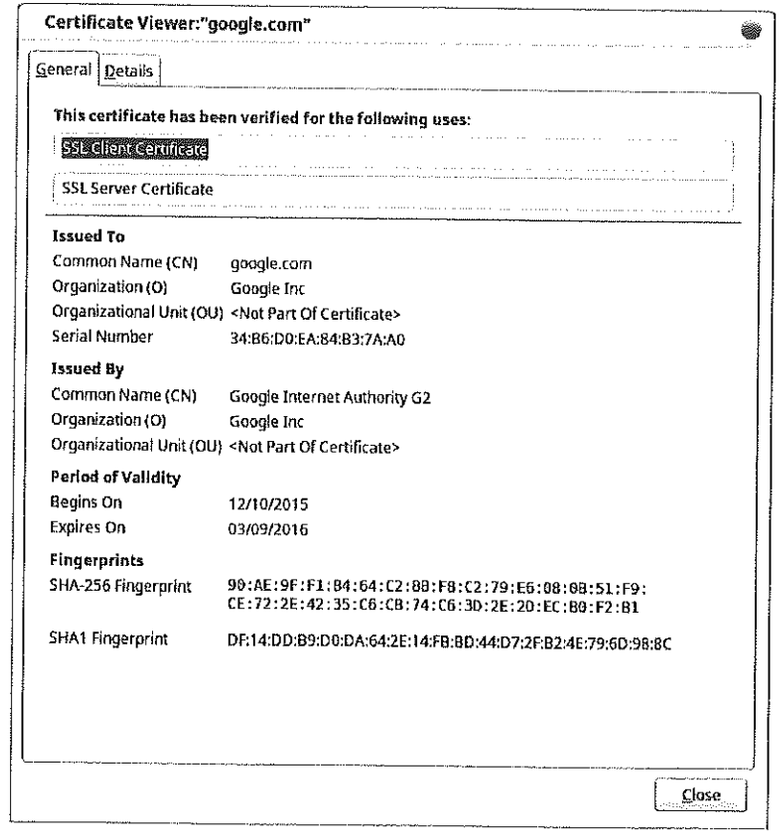

Figure 1 – Certificat X.509 du site www.google.be

- Issuer : C=US, O=Google Inc, CN=Google Internet Authority G2
- Subject : C=US, ST=California, L=Mountain View, O=Google Inc, CN=*.google.com

§ 2. Validation d'un certificat

La validation d'un certificat procède en deux étapes principales : la validation des informations contenues dans le certificat et la validation cryptographique de sa signature. Pour ce qui est des informations dont le

certificat est porteur, il convient de valider le sujet (le nom correspond-il au nom attendu ?), la période de validité (l'usage est-il permis à la date en cours ?), l'utilisation (le certificat peut-il être utilisé aux fins visées ?).

La signature du certificat est produite selon le schéma décrit à la Section 3 du Chapitre 2. Il s'agit donc du chiffrement avec la clé privée de l'émetteur (AC) de l'empreinte numérique des informations contenues dans le certificat. Pour la valider, il est nécessaire de disposer de la clé publique du signataire ; cette clé est généralement mise à disposition par l'AC sous la forme d'un certificat numérique, lui-même porteur d'une signature. Ainsi, le certificat de la figure 1 est signé par l'AC *'Google Internet Authority G2'* ; la clé publique de cette AC est distribuée dans un certificat dont le sujet est *'Google Internet Authority G2'* et l'émetteur est *'GeoTrust Global CA'*. Ce certificat est signé avec la clé privée de l'émetteur, et pour valider cette signature, il est nécessaire de disposer de la clé publique de ce dernier, disponible dans un autre certificat, dont le sujet est *'GeoTrust Global CA'* et l'émetteur est *'GeoTrust Global CA'*. La figure 2 illustre cette hiérarchie de certificats, aussi appelée « chaîne de confiance ».

Il peut sembler interpellant de constater qu'un certificat est émis par l'entité qu'il concerne (GeoTrust Global CA). Il s'agit d'un certificat auto-signé[15], c'est-à-dire signé par son propre sujet, qui peut se traduire par l'auto-affirmation par cette entité de la propriété de sa clé. Sans cette auto-signature, la validation d'un certificat nécessiterait une chaîne infinie de certificats pour valider les différentes signatures.

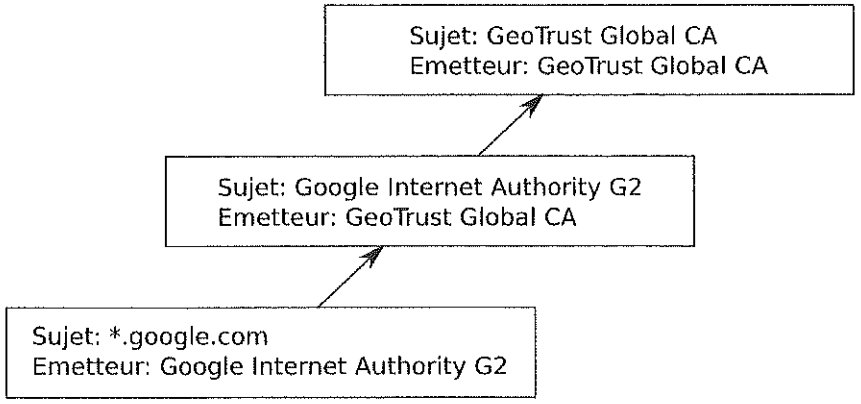

Figure 2 – Hiérarchie de certificats et chaîne de confiance

[15] Aussi appelé certificat-racine, car il ne nécessite pas d'autre certificat pour être validé.

L'utilisateur d'un certificat ne pourra avoir confiance dans les informations contenues dans ce certificat, essentiellement le lien certifié entre l'identité du sujet et la clé publique, que dans deux cas de figure :

- il fait le choix délibéré de faire confiance à ce certificat ; le degré de sécurité de cette situation est faible, dans la mesure où aucun contrôle cryptographique de la signature n'est réalisé ;
- il valide la signature du certificat en obtenant le certificat de l'émetteur, dont il validera la signature en obtenant le certificat de l'émetteur de ce second certificat, et de proche en proche il remonte la chaîne de confiance jusqu'à un certificat qu'il a explicitement accepté comme étant de confiance.

La validation du certificat présenté par un site Web consulté en https suit cette logique ; en reprenant l'exemple de la figure 1, le navigateur, après avoir vérifié que le certificat du site portait un nom identique (ou compatible) à celui du site qu'il consulte, qu'il est valide pour ce qui est des dates et des usages autorisés, va obtenir le certificat de l'AC signataire pour valider la signature du premier certificat. Pour vérifier la signature de ce second certificat, le navigateur devra remonter de proche en proche la chaîne de confiance pour arriver au certificat racine, auto-signé. Si ce certificat est accepté comme étant de confiance par le navigateur, ou que l'utilisateur, sollicité pour le valider, décide de l'accepter comme tel, cela provoque en cascade la validation implicite de tous les certificats situés en aval de la chaîne de confiance, donc celui de l'AC intermédiaire et celui du site lui-même. La validation d'un certificat consiste donc à remonter la chaîne jusqu'à un certificat reconnu comme étant de confiance.

Il nous paraît utile d'insister une nouvelle fois sur la sémantique d'un certificat numérique. Comme décrit plus haut, il atteste de la propriété du lien entre une clé publique et une identité (d'une personne ou d'un système), ce lien ayant été certifié par une autorité de confiance. L'utilisateur d'un certificat pourra donc, une fois le certificat validé avec succès, utiliser la clé incluse pour chiffrer un message à destination de son propriétaire (*cf.* Chapitre 2, Section 1, § 2) ou pour valider une signature émise par celui-ci (*cf.* Chapitre 2, Section 3), en ayant la certitude de s'adresser à l'identité confirmée par l'AC.

Bien qu'un certificat fasse mention des dates avant et après lesquelles il n'est pas valide, il peut être nécessaire d'abréger cette période de validité et mettre fin à l'usage du certificat, par exemple parce qu'il contient des données erronées ou obsolètes, parce que sa raison d'être a disparu, parce que la clé secrète correspondant à la clé publique qu'il contient a été compromise et ne peut donc plus être utilisée. Afin de gérer cette situation,

toute autorité de certification maintient une liste authentifiée de certificats révoqués[16], qu'elle tient à tout moment à la disposition de ses utilisateurs pour leur permettre de vérifier qu'un certificat qui aurait été validé avec succès au regard de la procédure décrite plus haut n'a pas entretemps été révoqué. Ce contrôle nécessite donc une vérification active de la part de l'utilisateur du certificat.

L'usage des certificats et la mise en place de l'infrastructure à clés publiques, composée d'un ensemble d'autorités de certifications permet de garantir l'authentification de sites Web. Chaque navigateur est préconfiguré par son fournisseur avec une liste d'AC reconnues comme de confiance, ce qui permet de ne pas solliciter sans cesse une validation par l'utilisateur lorsqu'il consulte un site sécurité (https). Ce n'est que si la chaîne de confiance du certificat du site ne permet pas de remonter à un certificat connu du navigateur que l'utilisateur est alerté et qu'une validation lui est demandée.

Même si de nombreux acteurs sont présents sur le marché des autorités de certification (à titre d'exemple, la version 43.0 du navigateur Firefox embarque plus de 90 organisations offrant des services d'autorité de certification), il est néanmoins fortement dominé par quelques acteurs principaux : selon le Web Technology Survey[17], le 1er décembre 2015, les certificats émis par les AC Comodo, Symantec, GoDaddy et GlobalSign représentaient respectivement 40.8 %, 29.4 %, 13 % et 7.8 % du certificats utilisés sur le Web. Ce morcellement pose un certain nombre de questions quant à la sécurité de cet écosystème, qui dépend de la sécurité en vigueur au sein de chacune des organisations, des nécessaires mécanismes d'audit à mettre en place, et du degré de responsabilité assumé par une CA en cas d'incident chez l'un de ses clients, imputable à un manque de sécurité de sa part [7].

Enfin, il nous semble important de souligner qu'un certificat numérique n'est fondamentalement rien d'autre qu'un ensemble de données informatiques auxquelles on a appliqué le processus cryptographique de signature numérique décrit plus haut. Ceci sous-entend qu'il est à la portée technique de tout un chacun de créer ses propres certificats. Cependant, ceux-ci ne seront jamais acceptés dans un scénario d'usage public, en raison de l'impossibilité de remonter une chaîne de confiance vers une AC reconnue.

[16] CRL – Certificate Revocation List.

[17] http://w3techs.com/technologies/overview/ssl_certificate/all.

SECTION 2. – Horodatage sécurisé

Comme nous l'avons discuté dans les sections précédentes, la signature numérique permet de garantir l'authenticité d'une information, par le chiffrement asymétrique à l'aide de la clé secrète connue du seul signataire d'une empreinte du document à authentifier, ce qui permet de garantir tant son intégrité que son origine [2].

Dans certaines situations, il peut être nécessaire d'établir de manière irréfutable l'existence d'un document ou d'une donnée à un moment précis. Bien que de nombreux systèmes informatiques intègrent des fonctions d'horodatage (dates de création/modification de fichiers par exemple), ces fonctions ne sont pas sécurisées, et sont donc sujettes à manipulation. Une fonction d'horodatage sécurisé vise à garantir non seulement l'authenticité et l'intégrité d'un document, mais aussi le moment précis où ce document a été soumis pour authentification [5]. Une telle fonction est typiquement mise en œuvre par une autorité de confiance, un service d'horodatage sécurisé (SHS).

Supposons qu'Alice souhaite faire authentifier et horodater un message. Une approche naïve consisterait pour Alice à transmettre ce message au SHS, qui enregistrerait la date et l'heure de réception, ainsi que le document. Pour valider la date, Bob devrait donc interroger SHS avec le document dont il veut vérifier la date, et SHS consulterait sa base de données pour répondre. Cette approche est impraticable dans la mesure où elle nécessite de communiquer la donnée originale à SHS, ce qui n'est pas toujours faisable ou souhaitable, où elle exige de SHS de stocker tous les documents horodatés, donc une taille de base de données très importante, et où la sécurité de l'ensemble dépend uniquement de SHS, dont il serait audacieux d'affirmer qu'il est exempt de toute vulnérabilité ou suspicion de collusion.

Une autre approche est de s'appuyer sur le mécanisme de signature numérique : soit m, le message à horodater. Alice calcule $h^o = H(m)$, où H est une fonction de hachage sûre (*cf.* Section 2.2). Elle envoie h^0 à SHS, le service d'horodatage sécurisé.

SHS ajoute à h^o la date et l'heure de réception, dh, calcule l'empreinte de ces deux informations juxtaposées $h^1 = H(h^o+dh)$, et chiffre cette empreinte à l'aide de sa clé secrète $sig_{SHS}=E(K,S_{SHS},h^1)$; SHS transmet ensuite à Alice l'horodatage et la signature (dh, sig_{SHS}).

Lorsque Bob veut valider l'horodatage de m, il s'adresse à Alice, qui lui produit la date dh, reçue de SHS, ainsi que sig_{SHS}. Pour vérifier que cette information est authentique, Bob calcule les empreintes $h^2=H(m)$ et $h^3=H(h^2+dh)$, et compare le résultat (h^3) au déchiffrement avec la clé publique de SHS de sig_{SHS}. Si les deux valeurs correspondent, c'est donc que la signature est

authentique, autrement dit, que SHS a bien attesté que le document dont l'empreinte est h' a été présenté à l'horodatage au moment dh.

Cette solution évite à SHS d'avoir à connaître le contenu du document, ou de stocker quelque information que ce soit.

Ce scénario est donc une extension du mécanisme de signature numérique qui authentifie non seulement les données mais aussi certaines métadonnées (ici, la date et l'heure). Notons qu'il existe des approches plus élaborées qui permettent d'éviter le risque de collusion entre Alice et SHS ; de même, des protocoles avancés de signature numérique supportent la signature de groupe, où l'un des membres signe au nom du groupe, ou la signature par délégation, où Bob peut signer pour le compte d'Alice, mais sans qu'il lui soit nécessaire de connaître la clé secrète de celle-ci. Le lecteur intéressé pourra consulter [1,5].

CHAPITRE III. Service d'identification électronique

Les mécanismes présentés au chapitre précédent constituent des primitives de base permettant de garantir non seulement la confidentialité d'une information ou d'un échange, mais aussi son intégrité et surtout son authenticité, c'est-à-dire son origine, avec un niveau de confiance élevé. L'infrastructure à clés publiques a introduit la notion de tiers de confiance, chargé de confirmer la propriété d'une clé cryptographique. Se basant sur une approche identique, il est possible de certifier d'autres types d'informations, telles que des données d'identification ou d'autorisation d'accès à un service ou une information.

Pour permettre à une autorité de certifier une quelconque information à propos d'un sujet, il est d'une importance vitale d'authentifier celui-ci. Au-delà de l'identification, l'authentification est un processus par lequel une entité (personne ou système) prouve son identité ; on envisage classiquement trois types d'éléments sur lesquels ce processus peut se baser :

- un élément connu de l'entité, tel un mot de passe ou un code PIN
- un élément matériel dont l'entité dispose, par exemple une carte à puce
- un élément lié à une caractéristique physique ou physiologique de l'entité, comme une empreinte digitale

Pour pallier aux faiblesses de certains de ces éléments, par exemple le choix d'un mot de passe facile à deviner, il est possible de les combiner dans une authentification multifactorielle, nécessitant par exemple une carte à puce embarquant des clés et certificats cryptographiques (élément

matériel) et un code PIN permettant de libérer l'accès à la clé secrète contenue sur la carte pour réaliser l'authentification.

Avec l'évolution d'une informatique centralisée vers un modèle hautement réparti, de nombreuses approches ont été proposées pour assurer une gestion distribuée des processus d'authentification et d'autorisation ; parmi celles-ci, citons Kerberos [10], un système dans lequel une autorité est chargée d'authentifier les utilisateurs, et leur délivre des jetons en fonction de leur profil leur permettant d'accéder à différents services (impression, accès à des fichiers...). Ces jetons sont sécurisés et sont validés par ces services. D'autres systèmes et protocoles ont été développés, adaptés à des contextes d'utilisation divers (internes à une organisation ou ouverts, protocoles web ou spécialisés...) et offrant des fonctionnalités plus ou moins avancées (délégation de droits, gestion de mandats...) ; on peut mentionner le Central Authentication Service[18], OpenID[19], oAuth[20], ou encore le Security Assertion Markup Language (SAML)[12,13] ou l'eXtensible Access Control Model Markup Language (XACML)[11] qui sont largement utilisés et sous-tendent les mécanismes d'authentification et d'autorisation de nombreux systèmes.

Une telle gestion décentralisée s'appuie sur 3 acteurs principaux [13] :

- le sujet, personne ou système, qui souhaite prouver son identité, ses attributs ou privilèges ;
- la partie requérante (*relying party*), aussi appelée fournisseur de service, qui demande à la partie certifiante de confirmer l'identité, les attributs ou les privilèges d'un sujet ;
- la partie certifiante (*asserting party*), aussi appelée fournisseur d'identité, qui certifie l'identité, les attributs ou les privilèges d'un sujet.

Ces acteurs s'inscrivent dans un espace ou cercle de confiance (*realm*) au sein duquel plusieurs acteurs de même type peuvent co-exister.

Les mécanismes de gestion distribués s'appuient sur un flux de traitement standard :

1. le sujet souhaite accéder à un service offert par la partie requérante ;
2. celle-ci demande à la partie certifiante de procéder à l'authentification du sujet et le cas échéant, de lui communiquer certaines informations le concernant ;

[18] http://jasig.github.io/cas/4.1.x/index.html
[19] http://openid.net/
[20] http://oauth.net/

3. la partie certifiante authentifie le sujet et en cas de succès, transmet à la partie requérante les informations demandées ;
4. finalement, se basant sur les informations reçues, la partie requérante décide de donner ou non accès au service demandé.

Ce scénario peut être complexifié en mettant en jeu par exemple, une partie certifiante qui garantit l'identité d'un sujet, une autre communiquant ses informations (attributs) et une troisième ses privilèges. Ces trois natures d'information (identité, attributs et privilèges) sont au cœur du langage SAML qui sous-tend de nombreux systèmes de gestion d'identité. Ce langage définit des formats de messages, appelés assertions, qui contiennent des informations certifiées par la partie certifiante. Il définit aussi les moyens techniques de véhiculer ces messages (protocoles et bindings), ainsi que des cas d'utilisations avancés, qui vont au-delà des considérations de ce document.

Dans le scénario décrit plus haut, les échanges entre les différentes parties doivent être sécurisés pour permettre à la partie requérante de pouvoir déterminer le degré de confiance qu'elle peut accorder aux informations reçues. Cette sécurisation s'appuie sur deux éléments majeurs :

- la confidentialité, l'intégrité et l'authenticité des messages échangés entre les différentes parties ; à cette fin, les mécanismes présentés au chapitre 2, sections 1 et 3 et au chapitre 3, section 1, sont pleinement utilisés ;
- la définition d'un langage commun, permettant à la partie requérante d'exprimer ses exigences et à la partie certifiante d'expliciter son mode de fonctionnement, par exemple quant aux données demandées, quant à la méthode d'authentification utilisée, quant au niveau d'assurance dans le contrôle d'identité effectué et donc dans les informations communiquées ; la définition d'un vocabulaire commun est essentiel pour permettre à toutes les parties d'échanger.

Le langage SAML, de par sa standardisation et sa large adoption, a été utilisé comme fondation des projets STORK et STORK 2[21], deux projets européens, visant à développer un cadre d'interopérabilité entre plateformes de gestion des identités basées sur les systèmes nationaux des États membres, permettant ainsi à un citoyen d'un État membre d'accéder sur base de son mécanisme d'identification électronique national à des services offerts par un autre état membre. Cette approche a été validée dans différents contextes tels que la santé (eHealth), les services financiers (eBanking), l'apprentissage à distance (eLearning) ou encore l'accès aux services publics au travers d'expériences pilotes.

[21] https://www.eid-stork.eu/ et https://www.eid-stork2.eu/

Les résultats de ce projet ont été intégrés dans l'initiative de la Commission européenne 'Connecting Europe Facility' en tant que service de base ('building block') pour le volet identification électronique (eId)[22]. Ce service vise à permettre aux citoyens de l'Union d'accéder aux services électroniques offerts par les États membres grâce à leur système d'identification national, en offrant des services d'authentification transfrontaliers.

Conclusion

Dans ce document, nous avons présenté des mécanismes cryptographiques permettant de garantir la sécurité d'une information, qu'il s'agisse de sa confidentialité, de son intégrité ou son authenticité. Cette garantie découle de la complexité des problèmes mathématiques sous-jacents à ces méthodes, qui, dans l'état actuel des connaissances et de la technologie, ne peuvent être résolus dans un temps suffisamment court pour pouvoir être compromis (Chapitre 2).

Bien qu'intrinsèquement sûrs, ces mécanismes sont mis en œuvre dans des systèmes dont la sécurité globale doit aussi être assurée. C'est entre autres le rôle de l'infrastructure à clés publiques de certifier la propriété d'une clé publique, et dès lors de permettre une authentification des échanges menés sous sa protection (Chapitre 3).

Nous basant sur une telle approche, nous avons montré comment ces différents composants pouvaient être assemblés pour créer un espace de confiance au sein duquel des parties peuvent échanger des informations à propos d'un sujet de manière sûre, et avec un degré de confiance défini (Chapitre 4).

De tels espaces existent aujourd'hui. Ils sont le fruit d'une évolution des modèles et des technologies, et seront certainement amenés à évoluer et à être intégrés dans de nouveaux systèmes.

Il est dès lors vital d'intégrer qu'au centre de la sécurité de ces systèmes se trouvent les services de confiance, autorités de certification, services de signature ou d'horodatage par exemple, et que la confiance dans un tel édifice est intimement liée au degré de confiance qui peut leur être accordé. Il est donc vital pour garantir un fonctionnement sûr à un écosystème basé sur ces mécanismes, prérequis à son adoption large et consentie, d'assurer un contrôle de qualité strict et efficace des prestataires de services de confiance et d'accompagner sa mise en place d'un cadre réglementaire et organisationnel approprié. C'est là l'un des enjeux majeurs du règlement 910/2014.

[22] https://joinup.ec.europa.eu/software/cefeid/home

Bibliographie

[1] Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 2nd edition, 1996.

[2] Jonathan Katz & Yehuda Lindell, *Introduction to Modern Cryptography*, Chapman & Hall/CRC, 2015.

[3] Simon Singh, *Histoire des codes secrets*, Jean-Claude Lattès, 1999.

[4] *Digital Signature Standard (DSS)*, National Institute of Standards and Technology, FIPS PUB 186-4, 2013.

[5] Steve Burnett & Stephen Paine, *RSA Security's Official Guide to Cryptography*, RSA Press, 2001.

[6] *Information technology – Open Systems Interconnection – The Directory : Public-key and attribute certificate frameworks – ITU X.509 Recommendation*, International Telecommunication Union, 2012.

[7] Axel Arnbak, Asghari Hadi, Michel Van Eeten & Nico Van Eijk. *Security Collapse in the HTTPS Market*. ACM Queue, Août 2014.

[8] Niels Ferguson & Bruce Schneier, *Practical Cryptography*, Wiley Publishing, Inc. 2003.

[9] Christof Paar & Jan Pelzl, *Understanding Cryptography*, Springer-Verlag, 2010.

[10] B. Clifford Neuman et Theodore Ts'o. *Kerberos : An Authentication Service for Computer Networks*, IEEE Communications, 32(9) :33-38. September 1994.

[11] *eXtensible Access Control Markup Language (XACML) Version 3.0*, OASIS Standard, 22 January 2013.

[12] *Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, 15 March 2005.

[13] David Birch, *Digital Identity Management : Perspectives On The Technological, Business and Social Implications*, Gower Publishing Ltd, 2007.