UCLouvain

École polytechnique de Louvain

Signature électronique et archivage

Les enjeux de la préservation à long terme de la signature électronique

Auteurs: **Séverine Mersch-Mersch, Noémie Nicaise**
Promoteurs: **Yves Deville, Aurore François**
Lecteurs: **Florian Delabie, Olivier Pereira**
Année académique 2020–2021
Master [120]: ingénieur civil en informatique, Master [120]: ingénieur civil en science des données

Résumé

Ce mémoire est centré sur la thématique de la préservation à long terme de signatures électroniques. L'objectif premier est de dresser un inventaire des méthodes de validation et de préservation de celles-ci ainsi que de les comparer.

Dans un premier temps, ce travail expose le cadre juridique et la problématique liés à l'archivage et à la signature électronique.

Ensuite, ce travail décrit de manière didactique les différents concepts utilisés pour créer une signature électronique : les méthodes de hachage, le chiffrement symétrique et asymétrique, le chiffrement RSA, les formes et formats d'une signature électronique. Il illustre son application directe au moyen d'une demande de ressource utilisant des certificats et des signatures.

De plus, il explique en détail différents types de signatures (basique, horodatée, LT, LTA) ainsi que leur processus de validation. Il analyse différentes méthodes de préservation, à savoir l'augmentation de signature, l'Evidence Record Syntax, la TrustChain, et les compare.

Enfin, une interface est créée dans un but pédagogique : découvrir les différents intervenants et les différentes actions réalisables dans le cadre d'archivage électronique de documents signés.

Cette œuvre est placée sous licence Creative Commons Attribution 4.0 International License. Pour consulter une copie de cette licence, visitez le site `http://creativecommons.org/licenses/by/4.0/` ou envoyez un courrier à Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Abstract

This thesis focuses on the long-term preservation of electronic signatures. The primary objective is to make an inventory of methods for validating and preserving electronic signatures as well as comparing them.

First of all, this work exposes the legal framework and the problematic regarding the archiving and the electronic signature.

Then, this work describes in a didactic way the different concepts used to create an electronic signature : hashing methods, symmetric and asymmetric encryption, RSA encryption, forms and formats of an electronic signature. It illustrates its direct application by means of a resource request using certificates and signatures.

In addition, it explains in detail different types of signature (basic, with time, LT, LTA) as well as their validation process. It analyzes different preservation methods : signature augmentation, Evidence Record Syntax, TrustChain and compare them.

Finally, an interface is created for an educational purpose : to discover the various stakeholders and the various actions that can be carried out in the context of electronic archiving of signed documents.

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit `http://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Remerciements

Nous tenions à remercier l'ensemble des personnes qui nous ont accompagnées lors de ce mémoire.

Premièrement, nous remercions nos promoteurs, Yves Deville et Aurore François pour leur aide : que ce soit pour nous aiguiller, pour soulever l'importance de notre travail lors de moments plus difficiles ou pour la relecture du mémoire écrit.

Deuxièmement, nous remercions Thomas Kopp et Stéphane Ries de l'entreprise LuxTrust, prestataire de services de confiance qualifié, pour leur aide et leur patience face à nos nombreuses questions.

Troisièmement, nous remercions de tout coeur nos familles et amis qui nous ont soutenues et encouragées tout au long de l'année. En particulier, à Sylvie et Jacques Mersch pour leur relecture critique et leurs remarques pertinentes, à Jean-Luc Nicaise pour la relecture générale et l'orthographe, et à Anne et Marc Levie ainsi que Françoise et Charles Postal pour l'accueil dans des lieux calmes de concentration.

Table des matières

Introduction **6**

1 Contexte général **9**
1.1 Cadre juridique : le Digital Act – le règlement eIDAS belge 10
1.1.1 Les prestataires de services de confiance 12
1.1.2 La signature électronique 13
1.1.3 Le certificat électronique 14
1.1.4 Le cachet électronique 15
1.1.5 L'horodatage électronique 15
1.1.6 Le service d'envoi recommandé électronique et l'authentification de site internet 16
1.1.7 L'archivage électronique 16
1.2 La problématique de l'archivage et de la signature électronique . . . 17
1.3 Objectifs du mémoire 21

2 La signature électronique **23**
2.1 Les fonctions de hachage 24
2.2 Les mécanismes de chiffrement 25
2.2.1 Le chiffrement RSA 26
2.3 La création d'une signature électronique 28
2.4 Les formes et formats de signature 29
2.5 Application : demande de ressource et demande de certificat 31
2.6 Autre méthode et sa limite : le tatouage 37
2.6.1 Les concepts généraux 37
2.6.2 Le tatouage numérique 37

3 La problématique de validation et de préservation des signatures électroniques **42**
3.1 La validation des signatures 42
3.1.1 La signature basique 43
3.1.2 La signature horodatée 47

3.1.3 La signature avec éléments de validation à long terme ou signature LT (Long Term) . . . 49
3.1.4 La signature assurant la disponibilité et l'intégrité à long terme des éléments de validation ou signature LTA (Long Term Availability) . . . 51
3.2 La préservation des signatures . . . 54
3.2.1 Evidence Record Syntax (ERS) . . . 55
3.2.2 TrustChain 1.0 . . . 61
3.2.3 TrustChain 2.0 . . . 66
3.3 Bilan des méthodes de préservation . . . 70

4 Simulation des signatures électroniques dans le contexte de l'archivage numérique 79
4.1 Interface : les différentes actions possibles . . . 80
4.2 Classes : attributs et fonctions . . . 85
4.3 Scénarios . . . 87
4.3.1 Scénario 1 : Étapes vers la création d'une signature LTA . . . 87
4.3.2 Scénario 2 : Révocation de certificats . . . 89
4.3.3 Scénario 3 : Expiration de certificats . . . 90
4.3.4 Scénario 4 : Utilisation de l'archivage automatique . . . 91
4.4 Différences entre notre simulation et un système de signatures électroniques dans un logiciel d'archivage . . . 91

Conclusion 94

A Lexique 97

Introduction

Dans notre société, le numérique est de plus en plus présent : que ce soit par l'envoi de mails au lieu de lettres, l'achat ou l'échange de biens et services en ligne, le stockage de photos, de vidéos, de documents scannés en plus des documents physiques, ou encore directement la création de documents numériques importants, tels que des contrats, plutôt que leur version papier. Les quantités de données augmentent très rapidement. En outre, dans le contexte actuel lié au coronavirus, les documents sont de plus en plus amenés à être échangés numériquement ou signés à distance de manière électronique [13].

Au vu de cette dématérialisation, plusieurs points importants sont à relever. D'abord, lorsqu'on reçoit un document numérique, on aimerait pouvoir vérifier l'identité de l'émetteur et vérifier que le document est bien ce qu'il prétend être, autrement dit authentifier le document. Mais, on aimerait également savoir si le document n'a pas été modifié, autrement dit s'il est intègre. C'est là que la **signature électronique** entre en jeu.

La signature électronique est présente dans de multiples domaines et pour de nombreuses applications. Nous pouvons citer plusieurs exemples concrets :

- un ingénieur en construction qui signe pour prouver que le travail qui est réalisé est sous sa supervision ;
- une pharmacienne qui signe pour prouver la réception d'un colis de médicaments ;
- un locataire et un bailleur qui signent un contrat de bail ou un état des lieux ;
- un employé et un employeur qui signent un contrat de travail ;
- un individu qui signe pour le paiement d'une transaction ;
- un citoyen qui signe afin de gérer des documents administratifs en relation avec le fisc ;
- une personnalité politique qui signe pour prouver qu'elle est bien à l'origine d'un texte.

De même, la signature peut aussi s'appliquer à l'UCLouvain : en tant qu'étudiantes-

monitrices, nous avons dû signer un contrat en version papier, l'EPL ne nous proposant pas encore de solution numérique. De plus, l'utilisation de la signature électronique pourrait s'étendre lors de la soumission d'un travail numérique ou dans le cadre d'examens, dont certains sont actuellement en distanciel, afin de s'assurer de l'identité de l'étudiant.

Il existe de nombreux types de signatures électroniques : une signature manuscrite scannée, une signature biométrique, une signature reposant sur des méthodes cryptographiques, etc.

Ces signatures reposent sur des méthodes telles que le hachage et le chiffrement, que nous découvrirons dans la suite. Ces méthodes apportent une sécurité pour identifier de manière univoque le signataire et prouver que le document signé n'a pas été modifié.

Cette identification et cette preuve d'intégrité peuvent se montrer nécessaires après un laps de temps non négligeable. En effet, certains documents signés doivent être conservés sur de longues périodes : tant dans une entreprise pour laquelle il faut préserver des contrats pendant plusieurs années, que dans une université telle que l'UCLouvain pour laquelle il faut préserver des diplômes durant de nombreuses années, ou dans le cadre d'archives où le but est de garder des documents pour une durée indéterminée.

Malheureusement, la signature électronique basique, à conserver elle aussi, n'est pas intemporelle. D'abord, elle repose sur des méthodes qui la rendent non sécurisée après un certain laps de temps, temps généralement plus court que la durée d'un contrat par exemple. Ensuite, le document sur lequel elle se base peut être amené à changer de format, ce qui, comme nous le verrons par la suite, invalide aussi sa signature.

Par conséquent, il est intéressant de dresser un inventaire des méthodes de préservation existantes. Étant donné que la signature est utilisée par de nombreux intervenants, pas toujours instruits sur la signature électronique, il est nécessaire de clarifier au maximum les subtilités de la signature, et de découvrir les différents moyens de la préserver.

De fait, cela permet de comprendre les protections apportées par une signature et d'améliorer la confiance donnée au numérique, mais aussi de percevoir la valeur juridique qui en découle.

Ce dernier point est marquant, car un document signé électroniquement peut être nécessaire comme preuve devant un juge, en cas de litige par exemple. Il est donc essentiel qu'un document signé numériquement puisse posséder la même valeur juridique que son équivalent papier.

Il y a quelques années, un juge pouvait refuser un document comme preuve juste au motif qu'il était électronique. Récemment, en 2016, une loi appelée Digital Act – dont nous élargirons la connaissance plus tard dans ce travail – a réglé ce problème et a permis de définir différents niveaux de qualification – simple, avancé et qualifié – donnant des niveaux de confiance différents aux documents signés.

Le niveau qualifié permet d'accepter le document de prime abord (tel un document physique). Ce cas implique qu'il revient à la personne qui conteste le document de prouver qu'il n'est pas acceptable, contrairement aux autres cas pour lesquels c'est à celui qui défend le document de prouver qu'il est véridique. Notons que c'est une instance particulière, passant par des vérifications précises, qui détermine si un service – telle une signature – est défini comme qualifié ou non.

À travers ce mémoire, subdivisé en quatre chapitres, nous allons découvrir étape par étape les notions en lien avec la signature électronique. Nous énumérerons les enjeux principaux de l'archivage numérique, et plus précisément ceux liés à la préservation à long terme des signatures électroniques. Ayant remarqué l'absence de comparaison entre les différentes méthodes de préservation existantes dans la littérature, nous dresserons l'inventaire de celles-ci. Ceci nécessitera des explications plus détaillées sur le fonctionnement des signatures électroniques, les cas de leurs utilisations, les points d'extensions pour les faire perdurer dans le temps et les méthodes de conservation. À la fin de ce document, vous pouvez également trouver un lexique reprenant les principaux concepts utilisés.

Le premier chapitre permettra de cerner le contexte d'apparition de la signature électronique et le cadre juridique mis en place par le règlement européen eIDAS. Il nous amènera aux principaux enjeux de l'archivage numérique et clarifiera les objectifs de ce mémoire.

Le deuxième chapitre sera dédié à la signature électronique : les bases sur lesquelles elle repose, son fonctionnement, les différentes formes et formats qu'elle peut prendre et une illustration de son utilisation.

Le troisième chapitre aura comme but de découvrir les différents processus de validation et de préservation de la signature électronique et de les comparer : un bilan reprendra les avantages et inconvénients des différentes méthodes.

Le quatrième et dernier chapitre sera consacré à une simulation d'utilisation à long terme de signatures électroniques, pour comprendre en détail et de manière pédagogique quels sont les principaux intervenants et les différentes actions réalisables dans le contexte d'archivage de documents signés. Cette simulation se trouve sur GitHub dans un répertoire open source à l'adresse suivante : `https://forge.uclouvain.be/nonicaise/simulation-de-signatures-electroniques`.

Chapitre 1

Contexte général

Avant d'entrer dans le vif du sujet, il est nécessaire de comprendre le contexte d'apparition de la signature électronique, les problèmes qu'elle devait résoudre et les questions qu'elle a soulevées.

Plusieurs entités, nationales comme européennes, y ont apporté un cadre formel. Au niveau européen, une directive datant de 1999 reprenait les dispositions à suivre par les États membres. En 2014, l'Union européenne a repris ces dispositions et les a complétées avec le règlement eIDAS (qui abroge cette directive). Celui-ci fut ensuite transposé par la Belgique sous le nom de Digital Act en 2016.

Après avoir indiqué les différents apports de ces règlements, ce chapitre mettra en avant les problèmes liés à l'archivage et à la signature électronique ainsi que la spécificité de la Belgique à ces propos. Enfin, il permettra de définir avec précision les objectifs de ce mémoire.

De la directive de 1999 au règlement eIDAS

C'est dans la directive 1999/93/CE [32] que la signature électronique fait son apparition dans un cadre juridique européen. Elle y était définie comme "une donnée sous forme électronique, qui est jointe ou liée logiquement à d'autres données électroniques et qui sert de méthode d'authentification" (art. 2 de la directive 1999/93/CE).

Un problème lié à cette directive est qu'elle ne fournissait pas de cadre transnational unique [45]. C'était plutôt un schéma à suivre par chaque pays. Chacun était donc libre de définir sa propre loi, rendant difficiles les transactions électroniques entre États membres. Comme chaque pays avait transposé la directive de manière différente, l'application législative de la signature électronique n'était pas harmonisée au sein de l'Union européenne. Bien que cette directive ait déjà précisé que l'efficacité juridique et la recevabilité comme preuve en justice ne devaient pas être refusées au seul motif que la signature se présentait sous forme électronique

(art. 5 de la directive 1999/93/CE), les règles pouvaient différer d'un pays à l'autre. Autrement dit, une signature électronique n'était pas recevable devant tous les tribunaux européens. La sécurité juridique était donc faible pour les utilisateurs et prestataires de ces services.

Un second problème est le cadre juridique trop restreint dans lequel s'appliquait cette directive. En effet, elle ne concernait que la signature électronique et ne prenait pas en compte le besoin d'identification électronique, d'authentification de site internet, d'envoi de recommandé électronique, etc.

Les utilisateurs étaient donc réticents au passage au numérique et à l'échelle européenne, la nécessité d'un cadre juridique beaucoup plus général s'est imposée. Le règlement européen eIDAS, objet de la section suivante, apporte ce cadre juridique pour différents services électroniques.

1.1 Cadre juridique : le Digital Act – le règlement eIDAS belge

Le règlement européen eIDAS [33], adopté le 23 juillet 2014 par le Conseil de l'Union européenne, porte sur l'identification électronique ainsi que l'utilisation et les conséquences juridiques des services de confiance. Ceux-ci sont des services électroniques normalement fournis contre rémunération. Ils consistent en (art. 3 §16 du règlement eIDAS) :

- la création, la vérification et la validation de signatures électroniques, de cachets électroniques ou d'horodatages électroniques, de services d'envoi recommandé électronique et de certificats relatifs à ces services ;
- la création, la vérification et la validation de certificats pour l'authentification de site internet ;
- la conservation de signatures électroniques, de cachets électroniques ou des certificats relatifs à ces services.

Dans la suite de ce chapitre, nous développerons les cinq mécanismes électroniques concernés :

- la signature électronique ;
- le cachet électronique ;
- l'horodatage électronique ;
- le service d'envoi recommandé électronique ;
- l'authentification de site internet.

La loi belge nommée **Digital Act** (ou encore *loi eIDAS et archivage électronique*) [40], adoptée le 21 juillet 2016, met en œuvre le règlement eIDAS et le complète en ajoutant des dispositions relatives à l'archivage électronique. Cela le rend plus complet et plus apte à faire face aux problèmes liés à l'archivage électronique.

Le but du règlement eIDAS et du Digital Act est de poser des normes non seulement pour améliorer la confiance dans les services de confiance, mais également pour renforcer la sécurité juridique des utilisateurs et des prestataires de ces services. Afin d'améliorer cette confiance, deux éléments importants ont été mis en avant [38] :

1. Un principe de **non-discrimination**. Il assure, dans tous les cas, qu'un juge ne peut pas refuser un document ou un service électronique comme élément de preuve sur le seul motif qu'il est électronique. Autrement dit, la valeur probante d'un document ou service électronique ne peut pas être contestée sous prétexte que ceux-ci sont électroniques ;
2. Une **présomption légale de conformité** pour les services de confiance électroniques qualifiés : l'intégrité et l'authenticité des services ne peuvent pas être remises en question. Étant donné que les services font l'objet de contrôles approfondis (cf. section 1.1.1 : les prestataires de services de confiance), ils offrent une confiance élevée.

Les services de confiance qualifiés sont en lien direct avec cette présomption de conformité. Ceux-ci permettent, en cas de litige, que la partie en désaccord endosse la charge de la preuve [1]. Supposons qu'une personne envoie un document signé de manière qualifiée à une autre personne. En cas de désaccord, c'est au destinataire de prouver que la signature n'était pas valide au moment de sa réception. Dans le cas contraire, c'est-à-dire en utilisant un service non qualifié, c'est à l'expéditeur d'endosser la charge de la preuve. Autrement dit, la présomption de conformité évoquée plus haut est équivalente à celle qui s'applique dans le domaine papier et physique uniquement lors de l'utilisation d'un service de confiance qualifié.

Actuellement, l'utilisation de services qualifiés n'est pas obligatoire : il faut attendre qu'il y ait plus de compétitivité et de propositions de différents prestataires. Mais les services qualifiés offrent une sécurité plus élevée, ce qui les rend attrayants.

1. Notons que si quelqu'un désire contester, il ne peut remettre en question l'authenticité et l'intégrité des services qualifiés (grâce à la présomption légale de conformité), mais peut prouver que la signature n'était pas valide, ou que l'envoi, dans le cas d'un recommandé, n'a pas été effectué, etc.

1.1.1 Les prestataires de services de confiance

Comme énoncé ci-dessus, il existe des prestataires qui offrent différents services de confiance au sein de l'Union européenne. Les listes de ces prestataires et des services de confiance qu'ils offrent, appelées **listes de confiance**, sont disponibles sur internet [11]. Les prestataires ont le choix parmi les types de services de confiance qu'ils veulent fournir, qu'ils soient qualifiés ou non.

Par contre, pour qu'un prestataire obtienne le statut qualifié pour un des services normés par le règlement eIDAS, il aura une certaine marche à suivre [16].

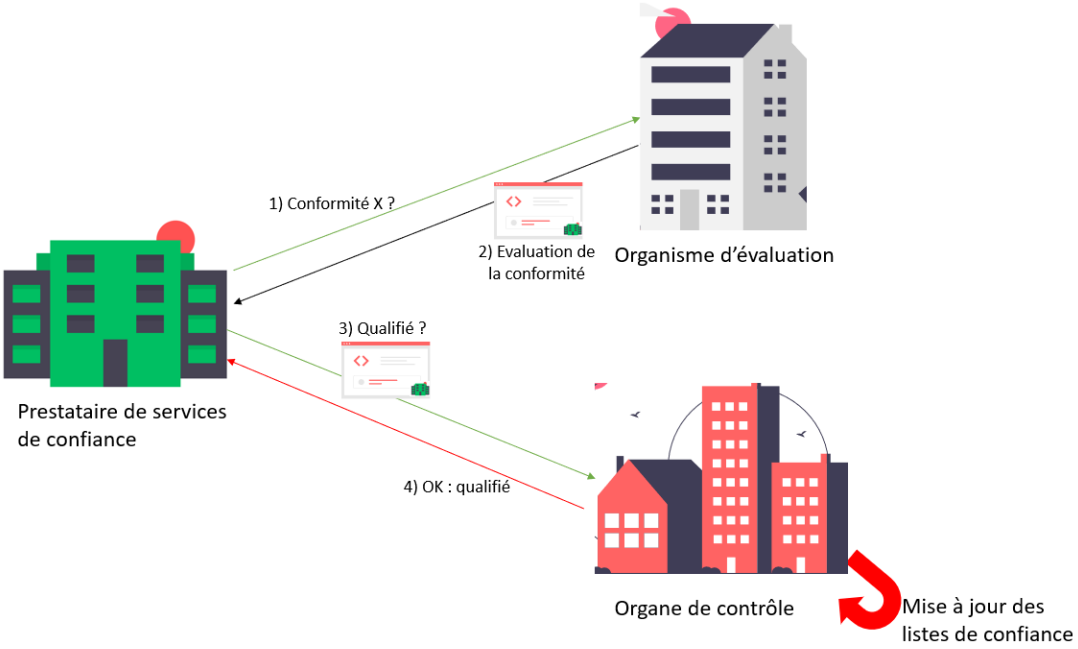


FIGURE 1.1: Obtention du statut qualifié

Voici la démarche (voir Figure 1.1 : suivre les flèches numérotées) : tout d'abord, le prestataire de services de confiance demande à un organisme accrédité d'évaluation de la conformité d'évaluer sa conformité et celle du service de confiance qualifié qu'il voudrait fournir par rapport aux exigences eIDAS. Ensuite, le prestataire notifie à l'organe de contrôle son intention de devenir qualifié et lui soumet également le rapport de conformité complété par l'organisme d'évaluation. Il revient enfin à l'organe de contrôle de vérifier si le prestataire de service de confiance respecte les conditions de la réglementation pour obtenir le statut de qualifié.

Toutes les exigences liées aux prestataires de services de confiance qualifiés sont reprises dans l'article 24 du règlement eIDAS et c'est à l'organe de contrôle de

vérifier si ces conditions sont bien respectées.

En Belgique, l'organe de contrôle est composé des agents de la Direction générale de la Qualité et de la Sécurité au sein du Service public fédéral Économie, P.M.E., Classes moyennes et Énergie [39].

1.1.2 La signature électronique

D'après le règlement eIDAS [33], la signature électronique représente des données sous forme électronique, qui sont jointes ou associées logiquement à d'autres données sous forme électronique et que le signataire utilise pour signer (art. 3 §10 du règlement eIDAS). Les signatures manuscrites scannées, les signatures biométriques (reconnaissance des empreintes digitales, reconnaissance vocale, etc.), les signatures au moyen d'une carte d'identité ou l'acceptation d'une transaction à l'aide d'une carte bancaire sont tous des exemples de signature électronique.

Une signature électronique peut être **avancée** si elle respecte les exigences suivantes (art. 26 du règlement eIDAS) :

1. Être liée au signataire de manière univoque ;
2. Permettre d'identifier le signataire ;
3. Avoir été créée à l'aide de données de création de signature électronique que le signataire peut, avec un niveau de confiance élevé, utiliser sous son contrôle exclusif ;
4. Être liée aux données associées à cette signature de telle sorte que toute modification ultérieure des données soit détectable.

Par exemple, signer un document numérique avec sa carte d'identité peut être assimilé à une signature avancée. En effet :

1. La carte d'identité appartient à une personne unique ;
2. Elle comprend toutes les informations pour identifier son propriétaire : numéro d'identification du Registre national, nom et prénom, lieu et date de naissance, etc. ;
3. Le code PIN de la carte est connu uniquement par son possesseur ;
4. La signature avec sa carte d'identité est directement liée au document numérique. Toute modification ultérieure sera remarquée, comme nous le constaterons dans le chapitre 2, section 2.5.

Une signature électronique sera **qualifiée** si, en plus d'être avancée, elle a été créée à l'aide d'un dispositif de création de signature électronique qualifié, et repose sur un certificat qualifié de signature électronique (art. 3 §12 du règlement eIDAS).

L'utilisation d'une signature électronique qualifiée a pour avantage d'être juridiquement équivalente à une signature manuscrite. Le choix est libre quant à l'utilisation d'une signature électronique simple, avancée ou qualifiée, mais chaque niveau est lié à des usages différents, dont nous pouvons citer quelques exemples [37].

La première, signature électronique simple, est utilisée pour donner ou recueillir un accord suite à un devis, signer un contrat de bail, de travail ou d'apprentissage, accepter des conditions générales ou encore signer un état des lieux. La deuxième, signature électronique avancée, sera plutôt utilisée dans l'optique de valider un contrat de vente, d'effectuer des contrats de prêts, de réaliser des contrats d'épargne ou de souscrire à des contrats d'assurance-vie. La troisième, signature électronique qualifiée, sert quant à elle à gérer des documents administratifs en relation avec le fisc, à traiter des actes d'avocats ou encore à soumettre des actes juridiques communautaires.

1.1.3 Le certificat électronique

Comme énoncé ci-dessus, une signature électronique qualifiée repose sur un certificat électronique qualifié. Ce dernier est une sorte de carte d'identité électronique [2]. Il contient les informations suivantes :

- le nom et prénom du propriétaire ;
- son adresse électronique ;
- sa clé publique, utilisée pour créer une signature (cf. chapitre 2, section 2.2) ;
- le nom de l'autorité de certification à l'origine du certificat ;
- les dates de validité liées au certificat ;
- la signature électronique avancée ou le cachet électronique avancé du prestataire de services de confiance qualifié délivrant le certificat.

Son but est de lier une personne à une clé publique, utilisée pour vérifier une signature électronique avancée ou qualifiée, et de donner aux autres utilisateurs une confiance en cette clé.

Un certificat peut être lié à une signature électronique, un cachet électronique ou encore à l'authentification de site internet. En fonction de son utilisation, il doit satisfaire à des exigences différentes, qui sont reprises dans les annexes du règlement eIDAS.

Les mécanismes liés aux certificats et leurs signatures sont cryptographiques, c'est-à-dire qu'ils ont pour but de protéger un message, soit pour qu'il soit illisible, soit pour déceler une modification. Cette protection repose sur un secret qui a de plus en plus de risque d'être découvert avec le temps. C'est pourquoi la durée de

validité d'un certificat est généralement limitée entre 2 et 3 ans [1, 15], pouvant aller jusqu'à 5 ans pour un particulier [1, 8].

1.1.4 Le cachet électronique

Tandis que la signature électronique est liée à une personne physique, le cachet électronique est quant à lui lié à une personne morale telle qu'une entreprise, un organisme ou encore un établissement public.

Comme l'explique LuxTrust, le cachet est "l'équivalent électronique du tampon en caoutchouc classique des entreprises, qui certifie et protège l'origine et la fiabilité des données" [29].

Dans le règlement eIDAS, il est défini comme des données sous forme électronique, qui sont jointes ou associées logiquement à d'autres données sous forme électronique pour garantir l'origine et l'intégrité de ces dernières (art. 3 §25 du règlement eIDAS). Tout comme la signature électronique, le cachet peut-être :

- **simple** ;
- **avancé** : celui-ci doit être (art. 36 du règlement eIDAS) :
  1. Lié au créateur du cachet de manière univoque ;
  2. Permettre d'identifier le créateur du cachet ;
  3. Avoir été créé à l'aide de données de création de cachet électronique que le créateur du cachet peut, avec un niveau de confiance élevé, utiliser sous son contrôle pour créer un cachet électronique ;
  4. Être lié aux données auxquelles il est associé de telle sorte que toute modification ultérieure des données soit détectable ;
- **qualifié** : il s'agit d'un cachet électronique avancé qui est créé à l'aide d'un dispositif de création de cachet électronique qualifié et qui repose sur un certificat qualifié de cachet électronique (art. 3 §27 du règlement eIDAS).

Il existe deux différences notables entre un cachet et une signature électronique. Premièrement, d'un point de vue légal, le cachet ne donne pas de preuve de l'engagement, il est uniquement une preuve de l'authenticité et de l'intégrité. Si on désire un engagement, la personne (physique) responsable de l'entreprise doit signer [29, 27]. Ensuite, l'apposition d'un cachet peut être automatisée, ce qui a un grand avantage pour sceller des quantités massives de données [29].

1.1.5 L'horodatage électronique

L'horodatage est, par définition, des données sous forme électronique qui associent d'autres données sous forme électronique à un instant particulier et établissent

la preuve que ces dernières données existaient à cet instant (art. 3 §33 du règlement eIDAS). Il permet donc de lier des données (un fichier, une signature, un cachet, etc.) à un moment (date et heure) universel, mais également d'attester qu'à un moment donné, une opération a été réalisée électroniquement ou une donnée existait.

En respectant les exigences suivantes (art. 42 du règlement eIDAS), l'horodatage sera **qualifié** :

1. Il lie la date et l'heure aux données de manière à raisonnablement exclure la possibilité de modification indétectable des données ;
2. Il est fondé sur une horloge exacte liée au temps universel coordonné ;
3. Il est signé au moyen d'une signature électronique avancée ou cacheté au moyen d'un cachet électronique avancé du prestataire de services de confiance qualifié, ou par une méthode équivalente.

Un horodatage électronique qualifié bénéficie d'une présomption d'exactitude de la date et de l'heure qu'il indique et d'intégrité des données auxquelles se rapportent cette date et cette heure (art. 41 du règlement eIDAS).

1.1.6 Le service d'envoi recommandé électronique et l'authentification de site internet

Le règlement eIDAS définit aussi des normes pour les mécanismes électroniques suivants :

— un service d'envoi recommandé électronique est un service qui permet de transmettre des données entre des tiers par voie électronique. Il fournit des preuves concernant le traitement des données transmises, y compris la preuve de leur envoi et de leur réception, et protège les données transmises contre les risques de perte, de vol, d'altération ou de toute modification non autorisée (art. 3 §36 du règlement eIDAS) ;

— l'authentification de site internet permet d'assurer à un visiteur d'un site internet que le site en question est tenu par une entité véritable et légitime. Cela afin d'instaurer un climat de confiance pour la réalisation de transactions commerciales en ligne (§67 des considérations du règlement eIDAS).

Comme ces services de confiance ne concernent pas directement la suite de notre mémoire, nous ne nous étendrons pas sur ceux-ci.

1.1.7 L'archivage électronique

L'archivage électronique est le seul service électronique à ne pas être défini dans le règlement européen eIDAS. Comme énoncé précédemment (dans l'introduction

de la section 1.1), il est présent dans le Digital Act. Cette loi insère de nouveaux éléments dans le Code de droit économique (CDE) [41], qui regroupe une grande partie des législations économiques [44].

Un **service d'archivage électronique** (SAE) est défini comme : "un service de confiance supplémentaire à ceux visés par le règlement eIDAS, qui consiste en la conservation de données électroniques ou la numérisation de documents papiers, et qui est fourni par un prestataire de services de confiance au sens du règlement eIDAS ou qui est exploité pour son propre compte par un organisme du secteur public ou une personne physique ou morale" dans le CDE (art. I.18, 17° du CDE).

Son but est donc de préserver sur le long terme les documents numériques (ou numérisés) archivés en évitant toute modification de ceux-ci afin qu'ils soient recevables comme preuve en justice [28]. Pour ce faire, ce service doit répondre à plusieurs caractéristiques [3, 28], dont celles qu'une archive physique doit posséder (données par le Conseil International des Archives [12]). Ces caractéristiques sont les suivantes :

- **l'accessibilité** : le document est facilement accessible et utilisable par les personnes habilitées ;
- **l'authenticité** d'un document : le document est bien ce qu'il prétend être, et qu'il a bien été créé au moment qui est indiqué par la personne qui est censée l'avoir produit ;
- **l'intégrité** : on doit savoir prouver que le document n'a pas été modifié depuis son versement sur le SAE ;
- **la confidentialité** : seules les personnes habilitées auront accès au document en question ;
- **la pérennité** : le document et ses métadonnées doivent être préservés durant une durée définie en garantissant sa lisibilité durant cette même période ;
- **la réversibilité** : le SAE doit permettre de récupérer chaque document et leurs éléments de preuve ;
- **la traçabilité** : toutes les opérations réalisées sur le document doivent être accessibles et recensées.

1.2 La problématique de l'archivage et de la signature électronique

Après avoir vu les différents concepts amenés par le règlement eIDAS et le Digital Act, nous pouvons mettre en avant les problèmes principaux, ainsi que la spécificité de la Belgique.

L'apparition de la signature électronique, comme énoncé par ces règlements, permet de répondre à un besoin d'identification de l'origine et d'intégrité des données numériques. Cependant, son apparition a aussi amené de nombreuses questions, notamment de la part des archivistes.

Tout d'abord, nous avons déjà énoncé le fait que la signature et le certificat lié à celle-ci utilisaient des méthodes cryptographiques, mais quelles sont ces méthodes ? Y a-t-il différents types de signature ? Quelles techniques pouvons-nous utiliser pour vérifier l'authenticité du document, c'est-à-dire contrôler qu'il n'a pas été modifié et qu'il provient de la personne spécifiée ? Autrement dit, comment pouvons-nous vérifier que la signature du document est valide ? Et pourquoi pouvons-nous avoir confiance en cette vérification ?

Ensuite, lorsqu'on reprend la ligne de vie d'une signature comme à la Figure 1.2, nous pouvons voir qu'en cas de litige, le document doit être vérifié à nouveau, des années après la vérification initiale, effectuée avant le versement. Dans les faits, ces vérifications sont identiques, mais séparées par un laps de temps conséquent, caractérisé par l'obsolescence rapide des technologies de sécurisation [4] : les méthodes cryptographiques utilisées justement pour signer et assurer l'intégrité.

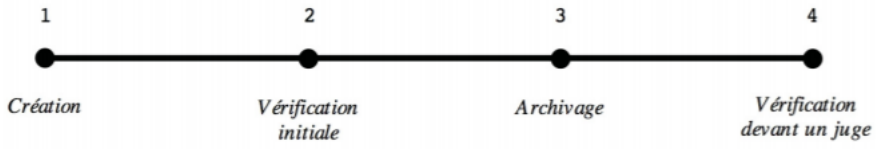


FIGURE 1.2: La ligne de vie d'une signature électronique [4]

La sécurité de ces méthodes cryptographiques diminue avec le temps [31]. Une question essentielle se pose alors, faisant encore débat lors du Conseil international des archives à Reykjavik en 2015 [30]. Cette question porte sur la pérennisation de la signature : comment préserver la validité de la signature électronique alors que les certificats associés ont une durée de vie très courte et que la sécurité des méthodes cryptographiques diminue ? La complexité de cette question peut être appuyée par le fait qu'en Europe peu de prestataires offrent des services de validation et de conservation à long terme de la signature. Plus particulièrement, en **Belgique**, il n'en existe même aucun pour la conservation de la signature électronique (et du cachet électronique) [11].

Enfin, la pérennisation de la signature électronique à elle seule n'est pas suffisante. Pour vérifier que le document n'a pas été modifié, il est logique qu'il faille avoir

accès à ce document en question, et le préserver avec la signature électronique [42]. Vient alors la question de l'archivage électronique : comment conserver un document numérique sur le long terme ? Cette conservation est confrontée à différents enjeux, enjeux repris par Arnaud Hulstaert [26].

L'obsolescence technologique

Le premier enjeu est l'obsolescence technologique et différents éléments sont sujets à cette obsolescence.

Premièrement, les composants matériels, les supports de stockage (hardware) ainsi que leurs périphériques de lecture se dégradent avec le temps. La dégradation peut survenir tant au niveau physique que technologique.

Utilisons l'exemple d'un CD-ROM. Celui-ci a fort été utilisé à partir des années 80 jusqu'aux années 2000. Nombreuses sont les dégradations physiques possibles : l'exposition prolongée du CD à la lumière, l'usure suite à de nombreuses lectures ou à de mauvaises manipulations. D'un point de vue technologique, nombreux sont les ordinateurs portables qui ne possèdent plus, à l'heure actuelle, le support nécessaire à la lecture du CD-ROM. Et même si certains ordinateurs possèdent encore un lecteur, il faut pouvoir en assurer l'exploitation, autrement dit, disposer d'un logiciel capable de lire les informations stockées dans le CD.

Or, les composants logiciels représentent le deuxième type d'éléments sujet à l'obsolescence pour deux raisons principales.

Tout d'abord, il n'y a pas forcément de compatibilité ascendante entre deux versions d'un logiciel. C'est-à-dire que des informations créées à partir d'une version X pourrait ne pas être compatible, pas exploitable avec une version X+1.

Le second problème est l'évolution du système d'exploitation sur lequel le logiciel tourne, ce qui peut rendre inutilisables certaines fonctionnalités du logiciel, voire le rendre totalement incompatible avec certains systèmes d'exploitation. Par exemple, un logiciel téléchargé avec une ancienne version de Linux ne sera peut-être pas utilisable avec une version plus récente.

Un dernier point amenant à l'obsolescence découle de l'évolution des composants logiciels. Il s'agit de l'évolution des formats et du type d'encodage. Ces derniers peuvent être modifiés au cours du temps. Il se peut qu'il en apparaisse de nouvelles sortes, ce qui engendre l'incompréhensibilité ou l'illisibilité de certains formats ou encodages plus vieux par de nouveaux logiciels. Afin de pallier ce problème, il est conseillé de migrer de format vers un nouveau plus adapté ou utilisé. Par exemple, un document sous format PDF 1.4 (qui était le format le plus adapté à la conservation il y a quelques années) va être converti en PDF/A qui est le format

standard actuel d'archivage à long terme. Cette nécessité de migrer de format mènera à la mise en évidence d'un autre problème, discuté ci-dessous.

La préservation des documents

Un deuxième enjeu est la préservation des documents. On voudrait pouvoir garantir l'intégrité d'un document, mais cela est contradictoire avec les stratégies de préservation, qui demandent une modification des formats, et donc du document à long terme. En effet, comme nous l'avons vu au paragraphe précédent, la migration de format est conseillée pour pallier l'obsolescence technologique. Ce changement de format modifie donc le document, ce qui implique que l'intégrité n'est pas conservée.

L'intégrité d'un document peut être altérée bien que l'information réelle qu'il contient, son contenu, n'ait pas été modifiée. Le contenant aura par contre subi une modification, ce qui empêchera la préservation de l'intégrité. Pour contrer cela, il est utile d'utiliser des formats qui sont pérennes dans le temps, comme un fichier XML ou un fichier PDF/A, c'est-à-dire des formats pour lesquels on peut s'attendre à ce qu'ils restent lisibles et utilisés sur le long terme.

L'organisation et le financement

Deux derniers enjeux plus logistiques sont également à prendre en compte : l'organisation et le financement. Stocker des documents est coûteux et demande notamment une bonne organisation au sein de l'organisme afin d'éviter de mauvaises surprises, comme une perte de données par exemple.

Les points susmentionnés concernent les enjeux de l'archivage à long terme d'un point de vue général. Ce service n'est cependant pas repris par le règlement eIDAS qui dissocie la problématique liée au document de celle liée à la signature. En effet, l'Union européenne spécifie bien [16] que "l'archivage électronique de documents et la préservation de signature électronique sont basés sur des solutions techniques différentes (attachées au document ou à la signature électronique) et diffèrent dans leur finalité (conservation du document ou préservation de la signature électronique)".

Toutefois, les archivistes mettent en avant les inconvénients de séparer la conservation du document de celle de la signature [30]. C'est là que la Belgique a pris les devants en ajoutant un volet relatif à l'archivage électronique par rapport à eIDAS, formant ainsi le Digital Act. Elle y spécifie qu'un SAE doit prendre en compte les dispositions relatives à la conservation de la signature électronique.

Cependant, il n'existe toujours pas de prestataire offrant ces services en Belgique. Cela est notamment dû à l'incompatibilité des deux préservations [4]. En effet, préserver le document sur le long terme implique de modifier le format de celui-ci, comme susmentionné. Par opposition, pour qu'une signature reste valide sur le long terme, le document lié ne doit pas changer de format pour permettre la validation à l'aide de méthodes cryptographiques. Nous faisons donc face à deux missions incompatibles.

1.3 Objectifs du mémoire

Le cadre juridique apposé à la signature électronique et aux autres mécanismes électroniques a permis d'améliorer la confiance et la sécurité juridique quant à leur utilisation : tout service électronique ne peut pas être refusé en justice sous le seul motif de se présenter sous forme électronique. Cependant, de nombreuses questions restent encore sans réponse, comme la section précédente le démontre.

Il est important de considérer chacun des problèmes liés à cette thématique, mais l'ensemble étant très large, ce mémoire va se focaliser plus particulièrement sur la problématique de la préservation de la signature électronique à long terme. Étant liées, les questions en rapport avec la signature et sa validation seront également traitées.

Si nous avons déjà donné en introduction le plan de notre mémoire, avec l'éclairage du premier chapitre, nous pouvons préciser ici son objectif. Il sera triple.

D'abord, au chapitre 2, le but sera de comprendre et d'exposer les mécanismes liés à la signature électronique : les fonctions de hachages et l'utilisation de clés de chiffrement. Une application directe de signatures reposant sur des certificats sera explicitée, pour en approfondir la compréhension.

Ensuite, dans le chapitre 3, le but sera de découvrir, de rassembler et de comparer des méthodes existantes répondant à notre problématique : la préservation à long terme de la signature électronique.

Enfin, au chapitre 4, la création d'une application pédagogique aura pour but de visualiser les différentes actions réalisables dans le contexte d'archivage numérique (fondation de prestataires de services de confiance, obtention de certificats et de clés de chiffrements correspondantes, création de signatures et d'horodatages de document, etc.) et de découvrir lesquelles sont exécutées par un système d'archivage autonome permettant la préservation à long terme automatique de documents numériques.

Le prochain chapitre sera donc dédié au fonctionnement de la signature électronique, se basant sur les fonctions de hachages et les mécanismes de chiffrement. Il identifiera les différentes formes et formats que peut prendre la signature électronique. Il donnera un exemple concret de son application. Enfin, il présentera une autre méthode (et sa limite) permettant de garantir l'authenticité et l'intégrité de documents : le tatouage numérique.

Chapitre 2

La signature électronique

La signature électronique est au document numérique ce que la signature manuscrite est au document physique : elle permet de garantir l'intégrité et l'authenticité lorsque son niveau de qualification est suffisant.

Dans le règlement eIDAS, la signature électronique est définie comme des données sous forme électronique, qui sont jointes ou associées logiquement à d'autres données sous forme électronique et que le signataire utilise pour signer (art. 3 §10 du règlement eIDAS [33]). De cette définition découlent plusieurs possibilités quant au type de signature : une signature manuscrite scannée, une signature biométrique, une signature-tatouage, une signature reposant sur des méthodes cryptographiques, etc.

Le règlement eIDAS a en plus défini différents niveaux de qualification d'une signature, déterminant différents niveaux de confiance : simple, avancé ou qualifié. Le premier peut être une signature manuscrite scannée, comparé au deuxième et au dernier niveau, qui au vu de leur définition, ne peuvent s'apparenter qu'à une signature cryptographique.

Dans ce qui suit, nous allons analyser plus en détail les principes technologiques utilisés pour la signature cryptographique : les mécanismes de chiffrement et les fonctions de hachage.

Nous commencerons par expliquer ces méthodes et par identifier des formes et formats qu'une signature peut prendre. Ensuite, nous donnerons une application concrète de son utilisation. Nous terminerons par citer une autre méthode permettant de garantir l'authenticité ou l'intégrité, le tatouage, et expliquerons sa limite.

2.1 Les fonctions de hachage

Le but des fonctions de hachage est de former une **empreinte numérique**, c'est-à-dire d'appliquer une transformation à un document pour avoir un résultat de taille fixe souvent plus petit que le document initial.

Pour illustrer le concept, voici un exemple d'une fonction de hachage : la fonction modulo 1000. En imaginant que les documents peuvent être exprimés comme de grands nombres utilisant leurs bits,

371983	===>	983
362930077853775380804808097	===>	97
467	===>	467

nous pouvons obtenir des valeurs de hachage de taille fixe (entre 0 et 999), généralement plus petites que le nombre représentant le document initial.

Dans notre cas d'application qu'est la signature électronique, nous avons besoin d'un type de hachage particulier : les fonctions de hachage cryptographiques, et plus précisément celles résistantes aux collisions. Leurs propriétés sont les suivantes [34] :

- la fonction de hachage doit être publique, et ne peut contenir de secret se basant sur son fonctionnement [1] ;
- la fonction de hachage peut prendre comme entrée un document de n'importe quelle taille et donnera en sortie une empreinte de taille fixe ;
- le calcul de la valeur d'un hash d'un document doit être relativement *facile* ;
- trouver ou construire un document sur base de la valeur d'un hash est pratiquement impossible ;
- il est pratiquement impossible de trouver deux documents différents ayant le même hash (appelé collision).

Ainsi, la modification d'un document, aussi légère soit-elle, implique la modification de la valeur de son hash.

Comme exemple de fonctions de hachage cryptographiques, nous pouvons citer les fonctions SHA (Secure Hash Algorithm), très connues.

1. Ceci est une extension du principe de Kerckhoffs, discuté dans la section 2.6.

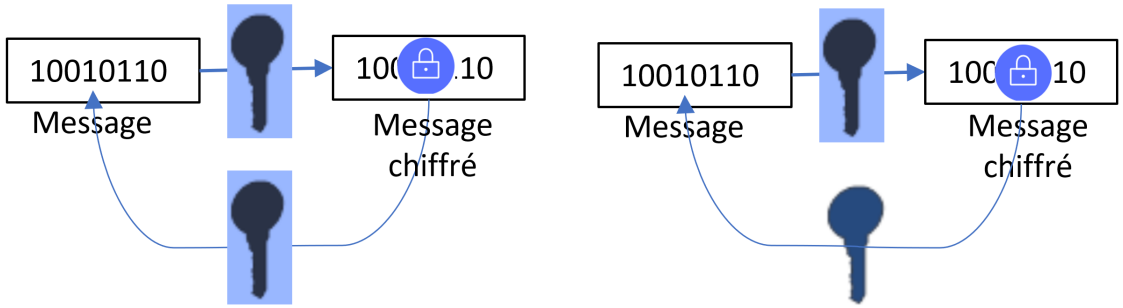
2.2 Les mécanismes de chiffrement

Le chiffrement a pour but de rendre un message non lisible à toute personne non concernée. Il transforme donc un message initial en un message codé illisible. Les personnes habilitées à le lire pourront le déchiffrer à l'aide d'un secret. Ce dernier est constitué de données partagées, publiques ou privées, appelées clés de chiffrement.

Deux mécanismes différents sont envisageables [2, 48] :

1. Symétrique : emploi d'une même clé secrète pour le chiffrement et le déchiffrement ;
2. Asymétrique : emploi d'une clé publique et d'une clé privée.

Dans le premier cas, l'expéditeur utilise la clé secrète partagée pour chiffrer son message et le destinataire utilise cette même clé pour le décoder.



(a) Symétrique (b) Asymétrique

FIGURE 2.1: Mécanismes de chiffrements

Dans le second cas, deux applications sont possibles [4]. Elles permettent soit la confidentialité soit l'authentification.

Pour la confidentialité, l'expéditeur utilise la clé publique du destinataire pour chiffrer son message. Celui-ci est ensuite décodé par le destinataire à l'aide de sa clé privée. Le message ainsi envoyé est alors uniquement lisible par le destinataire, puisqu'il est le seul à posséder la clé privée.

Si le but n'est pas de rendre un message illisible, mais de prouver que l'expéditeur est bien celui qu'il prétend être et donc de l'authentifier, l'ordre d'utilisation des clés sera inversé. L'expéditeur utilisera sa clé privée pour coder un message, et le destinataire la clé publique de l'expéditeur pour le décoder. Ainsi le destinataire sera certain que le message vient de l'expéditeur, mais la clé étant publique, n'importe qui pourrait lire le message.

C'est pour cette capacité d'authentification de l'expéditeur du message que le chiffrement asymétrique est utilisé pour signer électroniquement.

Les deux mécanismes de chiffrement garantissent la confidentialité du contenu en chiffrant le message à envoyer [2]. Une tierce personne, n'ayant pas accès à la clé secrète de décodage, ne pourra pas comprendre le message codé.

Le premier mécanisme a l'avantage d'être rapide [48], mais il nécessite d'avoir une clé secrète par groupe de correspondants, il a besoin de trouver un moyen sûr pour partager la clé et il n'assure pas la non-répudiation (toute personne possédant la clé peut écrire un message, on ne sait donc prouver quel correspondant a écrit tel message) [2]. Comme exemples nous pouvons citer DES, RC2, AES.
Le second a par opposition le désavantage d'être lent, mais n'a pas besoin de moyen sûr de partager la clé publique, car chacun garde sa clé privée. De plus, il assure la non-répudiation [2].

Comme exemple, nous expliquerons le chiffrement RSA qui est un mécanisme asymétrique, soit le type de mécanisme utilisé pour signer. Ce choix repose sur le fait qu'il est très répandu, par exemple utilisé par Adobe Acrobat Reader DC pour signer un PDF. Il sera également utilisé dans notre simulation expliquée au chapitre 4.

2.2.1 Le chiffrement RSA

La première étape consiste à fabriquer les clés privées et publiques, pour ce faire, il faut suivre les étapes suivantes [24] :

1. Choisir deux nombres premiers p et q distincts ;
2. Calculer le module de chiffrement : $n = p * q$;
3. Calculer l'indicatrice d'Euler en n : $\phi(n) = (p-1) * (q-1)$;
4. Choisir l'exposant de chiffrement e qui a un inverse dans $Z_{\phi(n)}$, c'est-à-dire que $(e * inv(e)) \bmod \phi(n) = 1$ et dont $PGCD(e, \phi(n)) = 1$;
5. Calculer l'exposant de déchiffrement d qui est l'inverse de e dans $Z_{\phi(n)}$;
   Cela se calcule grâce à l'algorithme d'Euclide étendu ou par $d = \frac{(k*\phi(n)+1)}{e}$, avec un k entier permettant à d d'être entier [21].

Au final, nous avons donc :
— une clé publique (de codage) : (e,n), à publier ;

2. Dans le cas où le chiffrement asymétrique est utilisé pour chiffrer et non signer un fichier.

— une clé secrète (de décodage) : (d,p,q), p, q et d doivent rester secret, mais uniquement d sera utilisé tel quel, p et q seront uniquement utilisés pour calculer n ($= p * q$).

Pour utiliser ces clés, le fonctionnement est le suivant.

— Pour coder un message m, on utilise la clé publique du destinataire : $M = (m^e)\ mod\ n$ avec $0 \leq$ m $\leq$ n-1. Notons que le message m est toujours plus petit que n.

— Pour décoder un message M, on utilise la clé secrète du destinataire : $m = (M^d)\ mod\ n$ avec $0 \leq$ M $\leq$ n-1.

Exemple

Création de clés

1. Soit p=13 et q=7 ;
2. Module de chiffrement : $n = p * q = 13 * 7 = 91$;
3. Indicatrice d'Euler : $\phi(n) = (p-1) * (q-1) = 6 * 12 = 72$;
4. Exposant de chiffrement e = 11 (PGCD(11,72) = 1) ;
5. Exposant de déchiffrement d = 59 (11*59 mod 72 = 1 ou encore on trouve k = 9 ; donc $d = \frac{(9*72+1)}{11} = 59$.

On obtient donc la **clé publique** (11, 91) et la **clé secrète** (59,13,7).

Codage & Décodage

Soit le mot MEMOIRE, on commence par transformer chaque lettre en nombre (correspondant à l'alphabet).

$$MEMOIRE \rightarrow 13.5.13.15.9.18.5 \tag{2.1}$$

Ensuite, on code chaque nombre par $M = (m^{11})\ mod\ 91$.

$$13.5.13.15.9.18.5 \rightarrow 13.73.13.85.81.86.73 \tag{2.2}$$

Cette suite de nombres sera donc envoyée sur un lien non sécurisé.

Le destinataire va donc la recevoir et décoder chaque chiffre au moyen de $m = (M^{59})\ mod\ 13 * 7$.

$$13.73.13.85.81.86.73 \rightarrow 13.5.13.15.9.18.5 \tag{2.3}$$

Enfin, en transformant chaque nombre en lettre, on retombe sur le message initial envoyé.

$$13.5.13.15.9.18.5 \rightarrow MEMOIRE \tag{2.4}$$

2.3 La création d'une signature électronique

Afin de créer la valeur de signature d'un document, on procède en plusieurs étapes [18].

1. La première étape consiste à construire les données à signer. Celles-ci comprennent le document, ou une représentation de celui-ci (le hachage du document), et des attributs liés à la signature, appelés *attributs signés* : l'identifiant du certificat à utiliser [3], le type de chiffrement utilisé pour signer, le type du document, le type d'engagement qu'apporte cette signature [4], etc.
2. La deuxième étape consiste à hacher cet ensemble de données pour former son empreinte.
3. La troisième étape consiste à chiffrer l'empreinte pour obtenir la valeur de la signature. Celle-ci utilise une clé privée, correspondant à la clé publique contenue dans le certificat référencé dans les attributs.

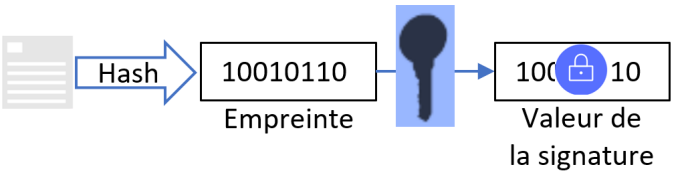


FIGURE 2.2: Calcul de la valeur de signature

4. Une dernière étape peut être exécutée, il s'agit d'ajouter d'autres attributs appelés *attributs non signés* puisqu'ils ne sont pas pris en compte dans le calcul de la valeur de la signature. Ces attributs sont optionnels, et permettent en général d'améliorer la longévité de la signature. Nous verrons ceci plus en profondeur dans le chapitre 3, section 3.1.

Ce qui s'appelle signature électronique est l'ensemble formé par la valeur de la signature et les attributs signés. Elle peut également contenir d'autres attributs non signés ainsi que le document signé, en fonction de sa forme et de son format, deux éléments que nous allons couvrir dans la prochaine section.

Pour être sûr que le document n'a pas été modifié, il sera nécessaire de vérifier que la valeur de la signature corresponde bien à l'ensemble des informations signées.

3. Cet identifiant est composé du hash du certificat et d'un identifiant unique précisant l'algorithme utilisé pour calculer ce hash.
4. Une signature peut avoir comme engagement de dire qui en est l'auteur, qui approuve le contenu, qui a reçu le document, qui a envoyé le document, etc.

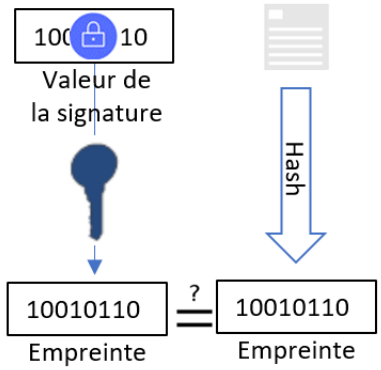


FIGURE 2.3: Vérification de la valeur de signature

Pour ce faire, d'un côté il faut retrouver l'empreinte liée à la signature au moyen de la clé publique correspondante. De l'autre côté, il faut récupérer les données qui ont été signées et les hacher. Si les deux étapes mènent à la même empreinte, les informations n'ont pas été modifiées.

2.4 Les formes et formats de signature

Tout d'abord, il existe plusieurs formes de signatures différentes [10] :

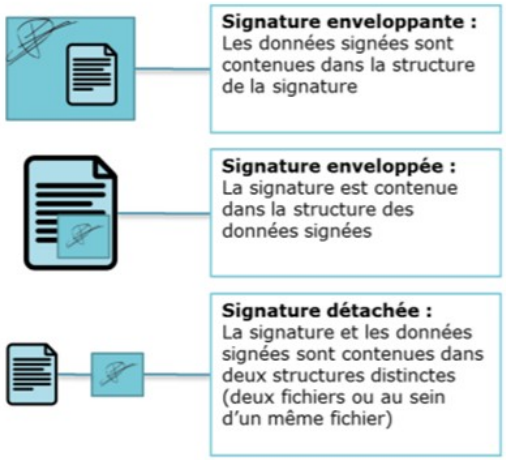


FIGURE 2.4: Formes de signature [10]

— **enveloppante** : la signature contient les données signées en elle-même ;
— **enveloppée** : la signature est intégrée au document et une seule structure contient le document et la signature ;

— **détachée** : la signature et le document sont deux structures distinctes, et donc potentiellement mis dans deux fichiers différents. Dans ce cas, la signature électronique est un fichier informatique autonome, distinct du fichier d'origine. Ce fichier autonome est appelé "jeton de signature".

Il existe aussi plusieurs formats de signature standardisés au niveau européen. Comme nous pouvons le voir sur la Figure 2.5, il existe trois grandes familles de signature, chacune avec une version simple et une version avancée.

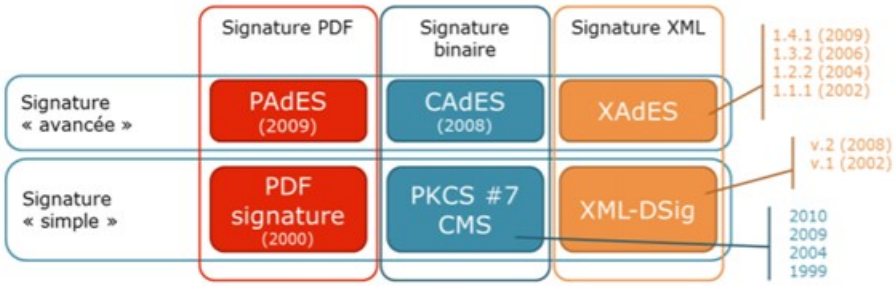


FIGURE 2.5: Formats de signature [10]

Le standard sur le marché est d'utiliser les signatures avancées AdES (Advanced Electronic Signature), standardisées au niveau européen par l'ETSI (European Telecommunications Standards Institute). Elles répondent à trois problèmes de sécurité clés : l'authentification, la non-répudiation et la préservation de l'intégrité du document.

Chaque format et forme aura son utilité en fonction de l'usage voulu [10]. Cela dépend de l'environnement métier (signature d'une personne physique ou morale, signature automatisée par un logiciel), du type de données à signer (données structurées ou non) mais aussi de l'équipement du destinataire. Par exemple, si le but est que le destinataire puisse recevoir un seul fichier facilement lisible et contenant la signature, on aura plus tendance à utiliser la signature enveloppée PAdES. Par contre, pour la signature électronique de fichiers structurés, comme un fichier XML, on utilisera la forme détachée si le document est volumineux, sinon, on utilisera la forme enveloppée ou enveloppante XAdES.

Ces différents formats et formes de signature seront importants dans le chapitre 3, section 3.1, où nous développerons la validation et la préservation de ces signatures.

2.5 Application : demande de ressource et demande de certificat

Supposons qu'Alice, une utilisatrice, ait envie d'une information particulière que Bob, un diffuseur, possède, mais que leur lien d'échange n'est pas 100% sécurisé [5]. Pour avoir la certitude que les informations n'ont pas été modifiées et qu'elles viennent réellement de Bob, la signature électronique va être utilisée.

D'abord, Bob va générer une paire de clés. Ensuite, il va utiliser sa clé privée sur les informations pour calculer la valeur de la signature [6]. Enfin, il va envoyer à Alice les informations, la clé publique et la valeur de la signature. Alice de son côté, va, au moyen de la clé publique reçue, vérifier si la valeur de la signature correspond bien aux informations et en déduire qu'elles n'ont pas été modifiées. Ainsi les informations sont arrivées à bon port et sont intègres.

Il reste cependant un doute : supposons qu'une personne malhonnête intercepte la demande d'Alice, se fasse passer pour Bob et envoie de fausses informations (avec des clés correctement générées) à Alice. Alice pourra vérifier que les données qu'elle reçoit ne sont pas modifiées, mais ne se rendra pas compte que ce n'est pas Bob qui les a envoyées.

C'est ainsi qu'un troisième intervenant va entrer en jeu : une autorité de certification (AC). Alice et Bob ont confiance en cette dernière. C'est pour cela que Bob va d'abord vouloir s'annoncer à l'AC pour que celle-ci lui fournisse un certificat, certificat qui contiendra sa clé publique ainsi que des informations permettant de l'identifier. Alice, de son côté, demandera à l'AC si elle peut avoir confiance en Bob (Figure 2.6) [2].

5. À ce stade, par sécurité, nous entendons l'éventualité d'une modification des informations. Il ne s'agit pas d'informations confidentielles transmises entre Alice et Bob.

6. Pour cette application de demande de ressource, une simplification a été opérée telle que la valeur de la signature est calculée uniquement sur le document, et non sur base de tous les attributs signés, tel que vu dans la section 2.3.

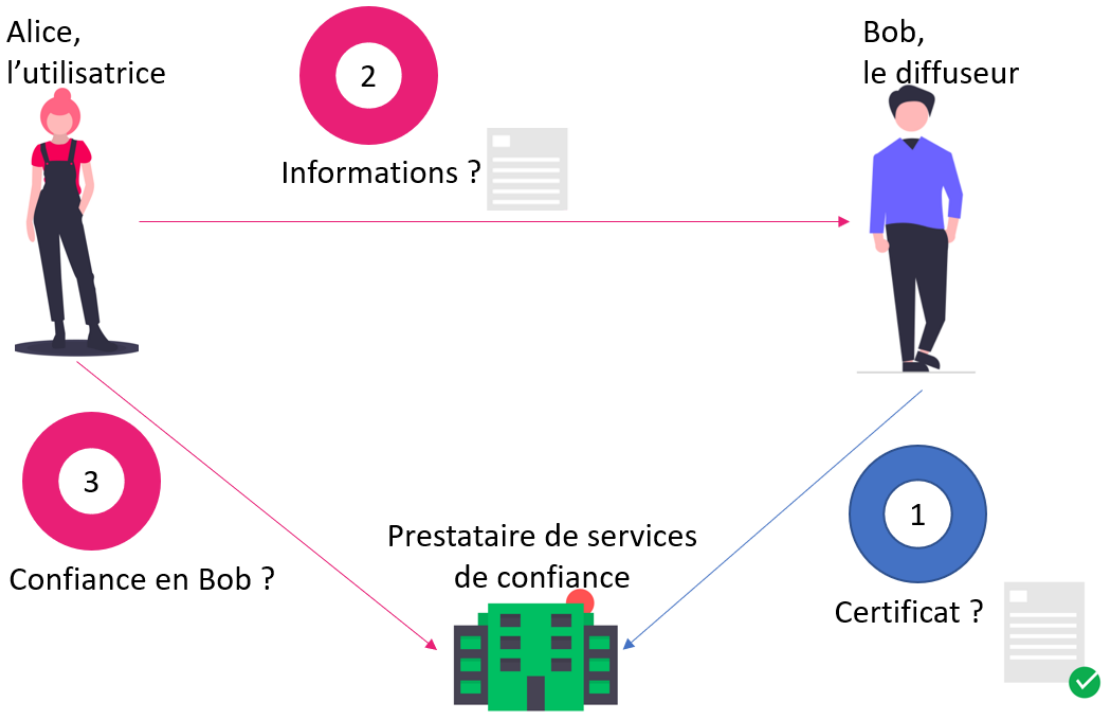


FIGURE 2.6: Demande de ressource

La première étape consiste en la création du certificat de Bob (Figure 2.7). Ce dernier va donner son identité au prestataire de services de confiance, une autorité de certification, qui va commencer par vérifier l'identité de Bob, puis va générer une clé privée et une publique.

Ensuite, l'AC va former un certificat comportant l'identité de Bob, des informations sur sa clé publique et sur les dates de validité ainsi que des informations précisant qu'elle est l'AC responsable de la création de ce certificat. Elle va rassembler ces informations, et les signer au moyen de sa propre clé privée : la signature ainsi que l'ensemble des informations forment le certificat.

Finalement, Bob aura accès à son certificat ainsi qu'à ses clés. Ainsi, la véracité des informations présentes dans le certificat est confirmée par deux processus : le certificat est lui-même signé par l'AC (on peut donc être sûr qu'il est intègre et que les informations dedans sont vraies, selon l'AC) et l'identité de Bob est vérifiée par l'AC (elle nous indique donc qu'on peut lui faire confiance) [4].

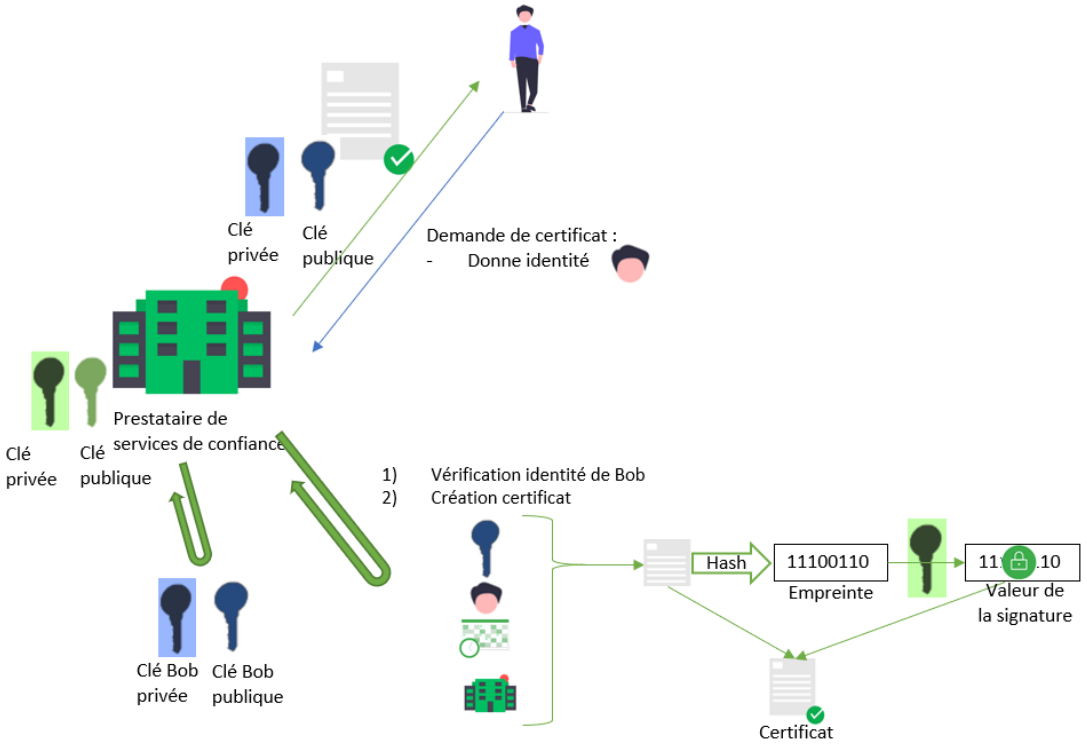


FIGURE 2.7: Demande de ressource étape 1 : demande de certificat

Notons que deux cas particuliers sont possibles pour disposer de clés [1, 27] :

— une approche classique utilise une carte à puce ou une clé USB avec clé privée confiée au client. Un désavantage important est que le coût est élevé, car du hardware est présent, ce qui est donc plus difficilement remplaçable ;

— une approche plus récente, apparue il y a environ 10 ans, utilise un serveur HSM (Hardware Security Modules), soit un serveur hautement sécurisé qui stocke les clés. Le client peut s'authentifier et créer des signatures directement en ligne.

Une seconde remarque concerne la signature effectuée par l'AC pour le certificat de Bob. Cela permet à Alice de faire confiance au certificat, et donc à la clé publique qui s'y trouve. Cependant, pour que l'AC effectue sa signature et ainsi permette à Alice de vérifier l'intégrité du certificat de Bob (expliqué à l'étape 3 ci-dessous), l'AC aura besoin de se baser sur une paire de clés. Afin d'assurer sa protection, la clé publique de l'AC devra, comme pour Bob, se trouver dans un certificat. Celui-ci devra donc être signé par une autre AC. La situation se répétera alors : cette AC devra elle aussi faire signer son certificat par une autre AC, et ainsi de suite.

Afin d'éviter de créer une chaîne infinie de certificats, le certificat auto-signé est apparu. Celui-ci est un certificat signé à l'aide de la clé privée liée à la clé publique présente dans ce même certificat. Une AC en possédant un, appelée AC racine, pourra dès lors signer des certificats à d'autres AC, dites intermédiaires. Ensuite, ces dernières peuvent fournir des certificats soit à d'autres AC, soit à des clients comme Bob.

Pour permettre la vérification du certificat de Bob jusqu'à son expiration, il faut que le certificat de l'AC l'ayant signé soit valide jusqu'à cette même date au minimum. La période de validité du certificat de l'AC doit donc être supérieure à celle du certificat de Bob. Une bonne pratique est de doubler cette période. Pour donner un exemple, si le certificat d'une AC racine a une période de validité de 12 ans, alors celle du certificat d'une AC intermédiaire sera de 6 ans, et enfin, celle d'une personne physique recevant son certificat de l'AC intermédiaire sera de 3 ans [1].

La deuxième étape consiste en l'envoi des données à Alice. Bob va rechercher les données (non sensibles) demandées par Alice et va les signer au moyen de sa clé privée. Il va ensuite envoyer la signature formée ainsi que son certificat à Alice.

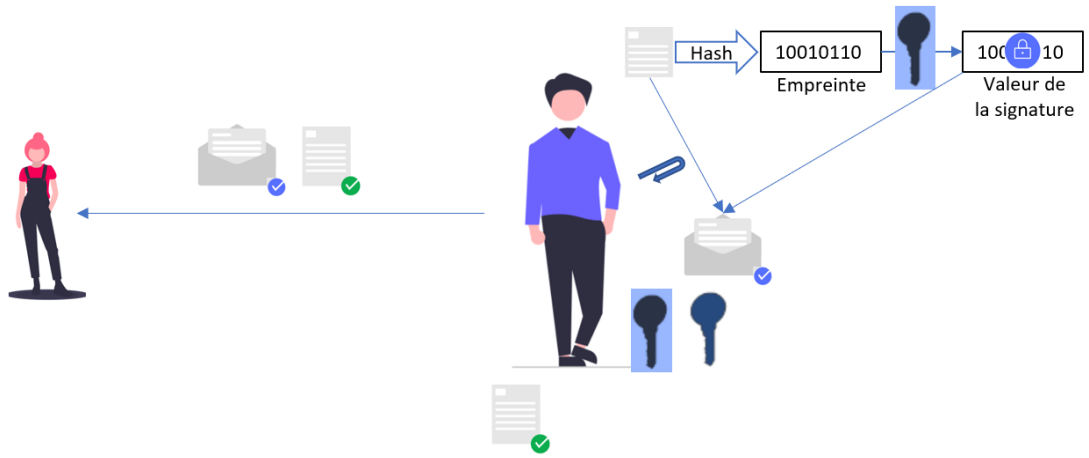


FIGURE 2.8: Demande de ressource étape 2 : envoi des données signées

La troisième et dernière étape consiste en l'acquisition des documents par Alice et en la vérification des données et du certificat de Bob [7] (Figure 2.9).

Alice va commencer par vérifier le certificat envoyé par Bob. Pour ce faire, elle va récupérer séparément les données du certificat et sa signature venant de

7. Une simplification a été effectuée ici afin de rendre le texte plus compréhensible. Logiquement, Alice devrait également vérifier la chaîne de certificats, afin d'obtenir la clé publique de l'AC et d'avoir confiance en celle-ci.

l'AC. D'un côté, elle va hacher les données du certificat. De l'autre côté, elle va retrouver l'empreinte liée à la signature au moyen de la clé publique de l'AC. Si les deux étapes mènent à la même empreinte, le certificat de Bob est intègre. Elle doit ensuite vérifier que la date de fin de validité du certificat n'est pas dépassée.

Notons à nouveau que deux méthodes sont possibles pour l'acquisition du certificat de Bob :

— soit Bob envoie son certificat à Alice ;
— soit Alice demande directement le certificat de Bob au prestataire de services de confiance qui stocke tous les certificats.

La seconde méthode est privilégiée, car elle permet de retirer tout certificat compromis ou révoqué de la circulation.

Enfin, elle peut alors récupérer la clé publique de Bob dans les données du certificat. Elle va utiliser cette dernière pour vérifier les données demandées. À nouveau, si le hachage des données demandées et l'empreinte de la signature (trouvée à l'aide de la clé publique de Bob) correspondent, elle peut être persuadée que les documents n'ont pas été modifiés et qu'ils sont bien ce qu'ils prétendent être.

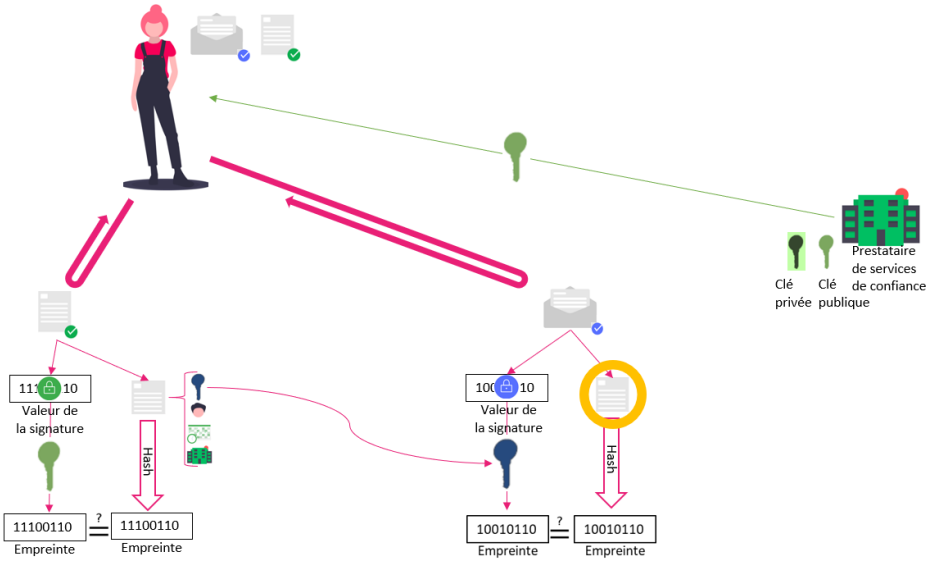


FIGURE 2.9: Demande de ressource étape 3 : validation des signatures et récupération des données

Dans la suite, si Alice ou d'autres personnes demandent de nouvelles ressources à Bob, ce dernier va pouvoir utiliser le même certificat pour signer les informations à envoyer.

La validité du certificat n'est cependant pas éternelle. En effet, les mécanismes liés aux certificats et leurs signatures étant cryptographiques, la durée de validité d'un certificat est limitée entre deux et cinq ans. Cette limitation est nécessaire du point de vue de la sécurité. Elle permet de limiter les fraudes liées à la compromission potentielle du certificat, mais également d'en améliorer la sécurité avec les mises à jour les plus récentes [1].

Lorsque son certificat expire, Bob doit alors en demander un nouveau à l'autorité de certification par la même procédure.

La vue d'ensemble de la demande de ressource (Figure 2.10) peut être plus explicite :

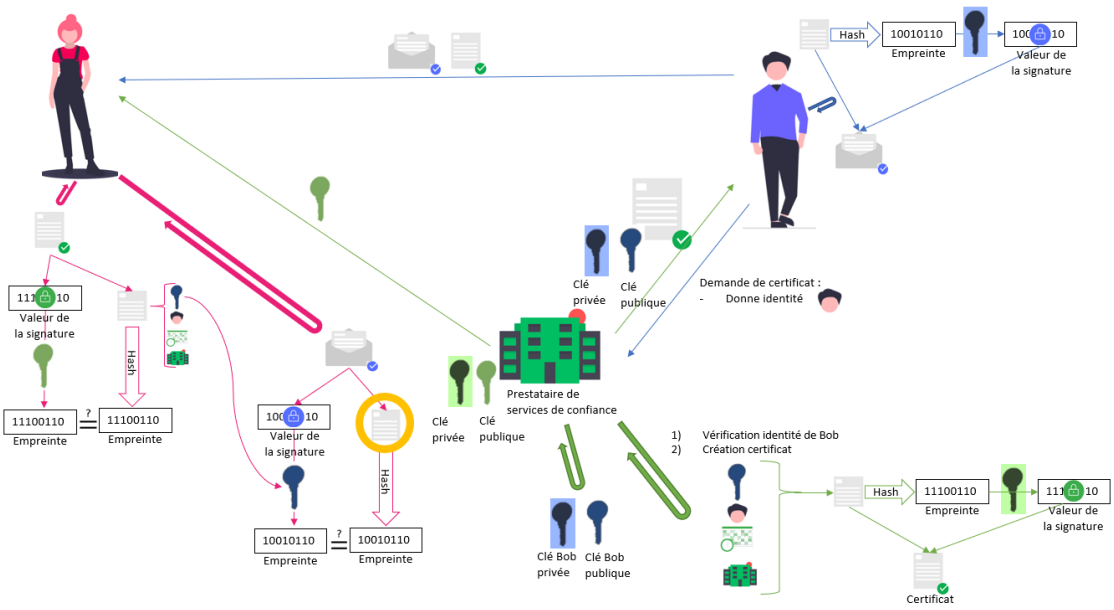


FIGURE 2.10: Demande de ressource schéma complet

Au final, l'utilisation de la signature permet d'atteindre 3 objectifs : l'intégrité des données (on sait que les documents n'ont pas été modifiés), l'authenticité (Bob a été identifié grâce à son certificat) et la non-répudiation (Bob ne pourra nier avoir signé les données).

2.6 Autre méthode et sa limite : le tatouage

La signature électronique n'est pas l'unique méthode pour garantir l'authenticité ou l'intégrité d'un document, un tatouage peut également être appliqué à cet effet. Venons-en d'abord aux concepts associés.

2.6.1 Les concepts généraux

La **cryptographie** est l'art de rendre un message (visible) indéchiffrable. Comme vu précédemment (cf. section 2.2 : les mécanismes de chiffrement), on utilise des techniques de chiffrement pour assurer l'authentification, la confidentialité et l'intégrité.

La **stéganographie** est l'art de camoufler un message, une information au sein d'un support anodin.

Le **tatouage** est l'art d'insérer un message dans un support ayant souvent un lien avec le support (parfois visible, parfois invisible).

Le **principe de Kerckhoffs** stipule que la sécurité d'un système doit reposer uniquement sur une clé secrète et non sur un algorithme de chiffrement. Ce dernier est donc supposé connu et public.

2.6.2 Le tatouage numérique

Un **tatouage numérique** est l'ensemble formé par un message ajouté à un document ou à un support. Citons comme exemples d'informations transmises des informations de copyright, sur le créateur, sur le propriétaire ou autre.

Il existe différents types de tatouages comparables sur plusieurs critères différents [14].

Le premier est la **visibilité**. Le tatouage visible a pour but d'informer que le document a des droits d'auteurs et d'en dissuader l'utilisation. Il peut par exemple s'agir de l'ajout "©" signifiant copyright, ou de la mise d'une image en fond. C'est un tatouage qui a pour désavantage d'être facile à enlever. Le tatouage invisible, quant à lui, implique la modification de bits de poids faible pour être invisible à l'œil nu. Celui-ci est parfois ignoré par le lecteur et est plus difficile à enlever : le retirer implique une dégradation du contenu. Il a comme but premier la traçabilité.

Le deuxième est la **robustesse**. Un tatouage fragile disparaît avec n'importe quelle modification du support. Ce genre de tatouage a pour but de repérer si un document a un tant soit peu été modifié. Il prouve donc l'intégrité. Un tatouage semi-fragile a pour but de disparaître en cas de modification malicieuse. Enfin,

un tatouage robuste a pour but de ne jamais disparaître dans le but de pouvoir récupérer dans tous les cas les informations cachées. Dans le cas où le message caché indique l'auteur, il peut prouver l'authenticité. Ainsi, on peut donc comprendre une propriété importante pour un tatouage : il doit subir les mêmes modifications que son support.

Le troisième est la **réversibilité**. Cela permet de retirer le message et nécessite donc l'utilisation de clés secrètes.

Enfin le quatrième est la **capacité**. Un tatouage de forte capacité permet d'ajouter beaucoup d'informations à un document. Prenons l'exemple d'une photo où nous pouvons ajouter de façon invisible quand elle a été prise et où elle a été prise.

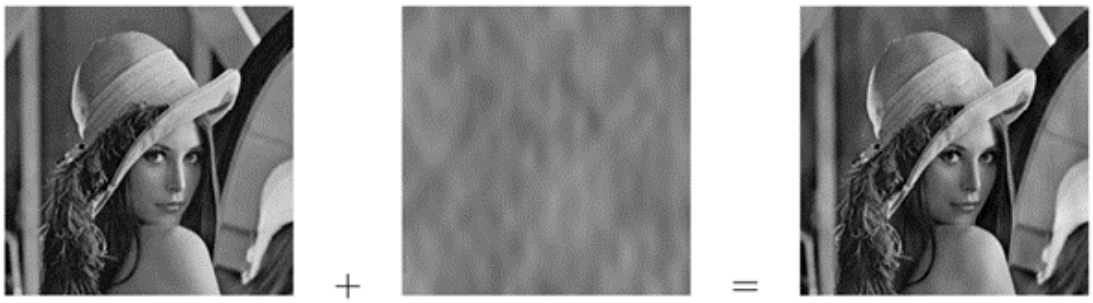


FIGURE 2.11: Tatouage d'une image [9]

Fonctionnement [8]

Le tatouage peut être considéré comme une sorte de communication (Figure 2.14). À une source, autrement dit à un support, on peut ajouter un tatouage invisible. Le message et son support sont codés, le message devient donc illisible, et est envoyé sur un canal non sûr. Il se peut donc qu'un bruit s'ajoute, que ce soit juste la modification à cause de la diffusion, ou d'une attaque malicieuse. Le message reçu est ensuite décodé et l'approximation du message original est calculée. Celle-ci est donc lisible, et on désire qu'elle se rapproche le plus possible du message initial.

8. Les images et explications suivantes viennent du cours "Le tatouage de documents numériques", de Marc Chaumont, professeur au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) [9].

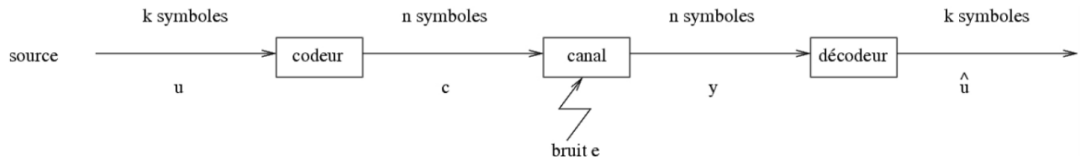
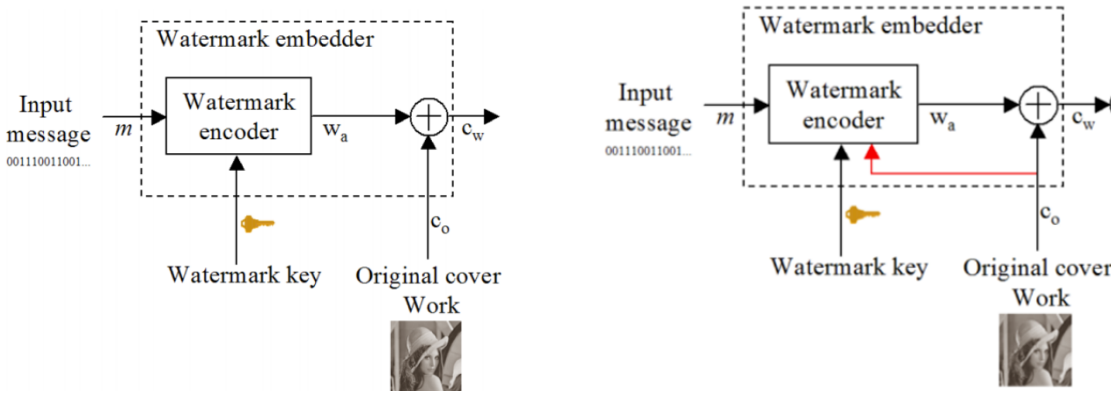


FIGURE 2.12: Communication via un canal

Il y a donc deux parties à cette communication, l'insertion du message et sa détection. Toutes deux peuvent être de deux types : aveugle ou informé. Aveugle signifiant le fait de ne pas connaître le support utilisé, et informé implique la connaissance du support initial non modifié.

Insertion aveugle vs Insertion informée

Dans l'insertion aveugle, le message à cacher est codé sans connaître l'image de base. Ensuite on ajoute ce message codé au support qu'on envoie sur le canal. Dans l'insertion informée, pour créer le code ajouté au support, on utilise non seulement le message à cacher, mais aussi l'image du support.

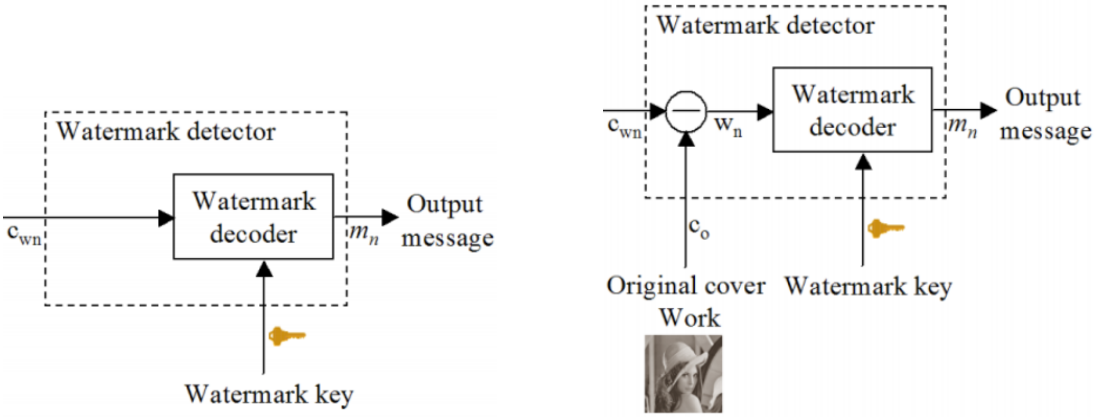


(a) Insertion aveugle (b) Insertion informée

FIGURE 2.13: Types d'insertion

Détection aveugle vs Détection informée

Dans la détection aveugle, le but va être de retrouver le message caché sans avoir accès au support initial utilisé. Dans la détection informée, le fait de connaître le support initial, permet de directement le soustraire à l'ensemble reçu et donc de décoder directement le message.



(a) Détection aveugle (b) Détection informée

FIGURE 2.14: Types de détection

L'ensemble insertion aveugle et détection informée se présente donc sous cette forme :

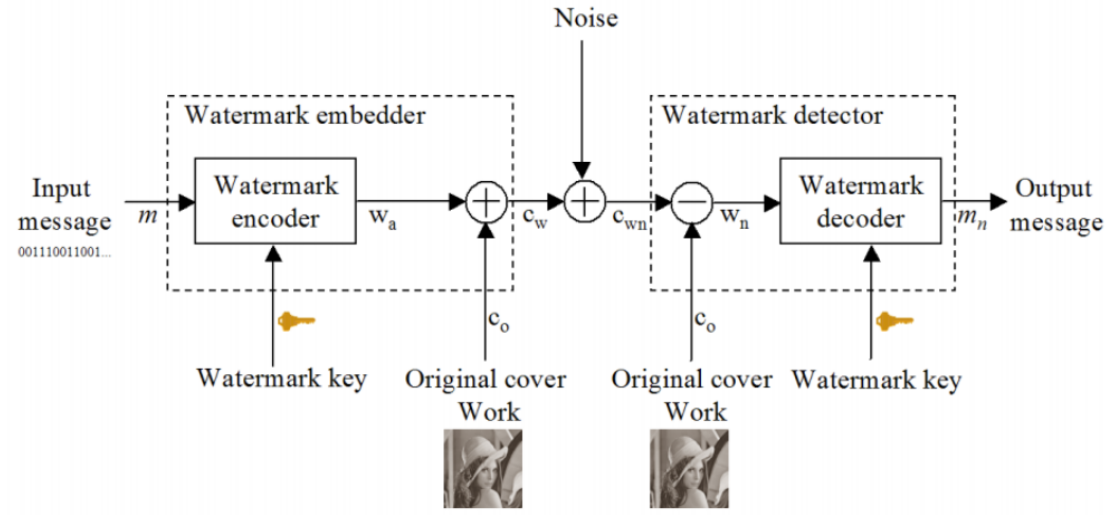


FIGURE 2.15: Insertion aveugle et détection informée

Exemples d'application

Le message peut permettre d'identifier le support diffusé. Supposons que les films passés dans une salle de cinéma comportent un tatouage invisible indiquant le lieu et la date de diffusion. Si on retrouve une séquence diffusée d'un film, on pourrait, au moyen du tatouage, retrouver le cinéma et la date de diffusion et ainsi retrouver l'éventuel coupable de cette diffusion non autorisée.

Le message peut permettre d'identifier le propriétaire ou l'acheteur du support. Ainsi en cas de doute ou de vol d'image, cela peut prouver qui en est le propriétaire ou l'acheteur et savoir si l'image montrée est dans ses droits.

Le tatouage peut aussi permettre de savoir si le support a été modifié, il permet donc de prouver l'intégrité du document.

Exemples de support

Les supports les plus utilisés sont les images, les vidéos et les fichiers audio. En effet, il est facile d'ajouter un changement imperceptible pour l'homme, et donc d'y cacher une information.

Par contre, ajouter un tatouage dans un texte ou un programme informatique est plus compliqué. En effet, il s'agirait d'ajouter des lettres ou de créer une redondance dans le texte permettant d'ajouter un message [9]. Ceci est donc beaucoup plus complexe et moins pratique lorsqu'il s'agit d'un document officiel qui doit être conservé tel quel.

C'est pour cette dernière raison que nous n'avons pas continué nos recherches sur le tatouage, notre but étant de préserver l'intégrité et l'authenticité de n'importe quel type de documents électroniques, principalement des documents écrits. Ce que le tatouage ne permet donc pas.

En effet, soit ce tatouage serait visible et perturberait le document de base : par exemple ajouter des lettres à certains endroits, des fautes à d'autres, avec une sorte de redondance permettant d'y cacher un message. Le document aurait donc des fautes d'orthographe ou des lettres en plus qui n'ont aucun sens avec le document de base qu'on désire conserver (par exemple un contrat). Soit le tatouage serait plus difficilement visible, et consisterait en la mise en page, l'ajout d'espaces ou de passages à la ligne. Mais si une personne réécrit mot à mot le texte sans la même mise en forme, le tatouage disparaîtrait.

Il est donc difficile d'avoir un tatouage robuste, et impossible de pouvoir authentifier l'auteur de manière certaine et univoque.

Dans le chapitre suivant, nous allons donc nous focaliser sur la signature cryptographique et les méthodes liées à sa préservation et à sa validation.

9. Le tatouage de texte a été utilisé par la Résistance dans ses messages aux Anglais : telle faute d'orthographe à telle place signifiait par exemple "Les Allemands m'ont pris" [35]. Ceci n'est malheureusement pas applicable pour des archives dans lesquelles l'ajout de fautes d'orthographe n'aurait pas de sens.

Chapitre 3

La problématique de validation et de préservation des signatures électroniques

Jusqu'ici, nous avons trouvé un moyen pour garantir l'authenticité et l'intégrité des documents à court terme : la signature électronique.

Malheureusement, certains documents signés doivent être conservés pour une durée de plus de 5 ans, soit au-delà de la validité du certificat correspondant à la signature (cf. chapitre 1, section 1.1.3). Or, si on veut pouvoir effectuer une vérification de cette signature des années plus tard, il faut trouver un moyen de préserver la validité des signatures. Ainsi, plusieurs problèmes se présentent à nous. Comment conserver à long terme des documents numériques signés tout en préservant leur intégrité ? Comment préserver une signature ? Comment la valider directement après sa création ou après un long laps de temps ?

Dans ce chapitre, nous découvrirons les types de signatures qui permettent la validation des documents à long terme et les différentes phases de cette validation. Ensuite, nous nous concentrerons sur la préservation de la signature, notamment au travers de multiples horodatages, par l'utilisation d'ERS, ou encore par la TrustChain. Enfin, nous ferons un bilan de ces méthodes, en spécifiant leurs avantages et inconvénients.

3.1 La validation des signatures

Afin d'avoir un modèle de signature validable sur le long terme, quatre classes de signatures ont été définies dans la norme ETSI EN 319 102-1 [18]. Les signatures AdES (CAdES, XAdES, PAdES), présentées dans la section 2.4, mettent en oeuvre

ces quatre classes. La première classe n'est validable que lorsque le certificat lié à la signature est encore valide. Elle correspond à la signature telle que nous l'avons vue dans le chapitre 2 à la section 2.3. La dernière permet quant à elle d'être validée sur le long terme, indépendamment de la disponibilité des informations nécessaires à la validation. Chaque classe est construite par-dessus la précédente et permet d'améliorer la capacité à valider une signature sur le long terme.

Cette section définit ces quatre classes et les processus pour valider chacune d'elle en se basant sur la norme ETSI EN 319 102-1 [18]. Notons qu'avant de procéder à une augmentation d'une signature, c'est-à-dire de passer d'une classe de signature à une autre, une validation de la signature de la classe inférieure doit être opérée.

Rappelons également que, comme nous l'avons vu dans la section 2.4, il existe différents formats de signature : enveloppante, enveloppée et détachée. Pour la suite de cette section, nous prenons l'hypothèse que la signature est enveloppante et donc que le document est contenu dans sa structure. Ce choix a été opéré pour se conformer à la norme ETSI et il permet d'homogénéiser les explications liées à la vérification de chaque classe. Dans les autres cas, l'unique différence réside dans la manière de récupérer le document. Pour une signature enveloppée, le document est la structure principale, avec la signature comme sous-structure. Pour une signature détachée, les données signées fournissent une référence vers le document.

3.1.1 La signature basique

Cette signature doit contenir les mêmes éléments que ceux énoncés dans la section 2.3 :

- le document (ou une référence vers celui-ci) ;
- la valeur de la signature ;
- une référence ou une copie du certificat de signature comme attribut signé ;
- optionnellement d'autres attributs signés ou non signés, tels que le type du document, le type d'engagement qu'apporte cette signature, la politique de signature, le lieu de création de la signature, etc. [1].

Ce type de signature peut être validée tant que le certificat auquel elle est liée n'est ni révoqué ni expiré, ce qui implique qu'elle n'est pas utilisable sur le long terme.

1. L'ensemble des attributs possibles est repris dans la norme ETSI EN 319 102-1 [18].

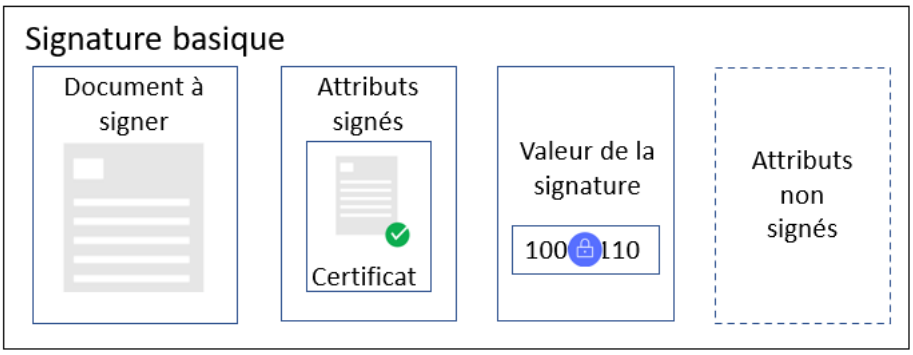


FIGURE 3.1: Signature basique

La validation de la signature basique

Plusieurs étapes sont nécessaires pour valider la signature basique [2].

1. **Vérification du format**, pour qu'il soit conforme au format de base afin que son contenu puisse au moins passer les vérifications cryptographiques.
2. **Récupération du certificat signant**, dont au moins l'identifiant se trouve dans la structure de la signature ; soit l'entièreté du certificat s'y trouve directement, soit il faut le retrouver au moyen de son identifiant et à l'aide de sources extérieures. L'identifiant du certificat comprend le hash de ses données [3]. À cette étape s'ajoute donc la vérification de la correspondance entre le hash du certificat complet à calculer et le hash contenu dans son identifiant.
3. **Initialisation du contexte de validation**. Il s'agit ici de récupérer l'ensemble des contraintes appliquées à la validation des certificats, à la cryptographie et aux éléments de signatures. Par exemple, de préciser si les clés de chiffrement utilisées pour signer doivent avoir une taille minimale ou encore si la signature doit avoir un format précis, contenant des éléments spécifiques. Toutes les contraintes peuvent être retrouvées dans la spécification ETSI TS 119 172-1 reprenant les politiques de signature [17]. Ces contraintes peuvent être implicites dans l'implémentation de la validation ou explicites en étant données initialement dans une politique de validation.
4. **Validation du certificat**. Cette étape vérifie si le certificat signant (récupéré à l'étape 2) est valide au moment de validation.

2. Ce processus est exécuté pour un *moment de validation*. Celui-ci peut être spécifié en argument de la validation, permettant ainsi de "valider" une signature à un moment dans le passé. S'il n'est pas précisé, on prend le démarrage du processus comme moment de validation.
3. Voir section 2.3 sur la création d'une signature électronique.

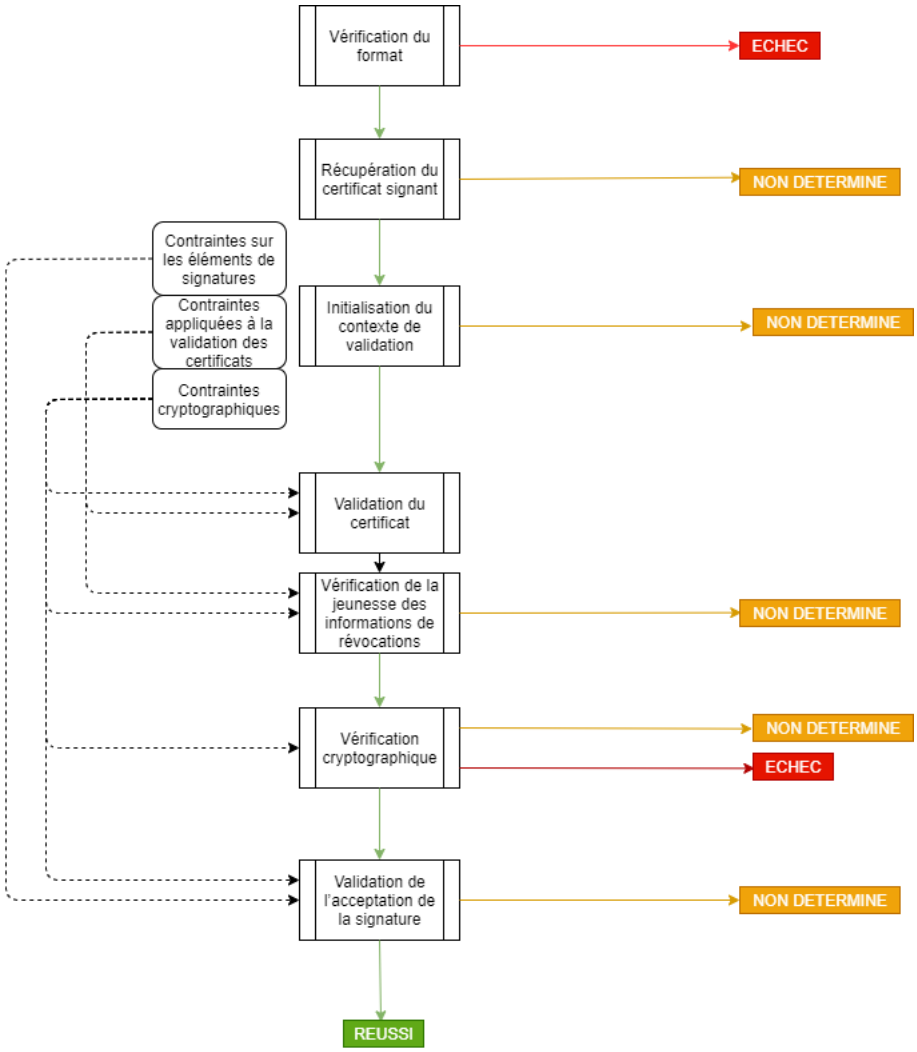


FIGURE 3.2: Processus de validation de la signature basique

Pour ce faire, il faut d'abord créer la chaîne de certification, c'est-à-dire remonter à partir du certificat signant, prenant en compte le certificat l'ayant signé, et le certificat ayant signé ce dernier certificat, etc., jusqu'au certificat racine.

Ensuite, il faut vérifier cette chaîne créée : que les dates de validité des certificats se chevauchent, qu'ils sont intègres et qu'ils sont non révoqués au moment de signer le certificat suivant [6]. Deux modèles de validation existent et le modèle à utiliser est spécifié par une contrainte obtenue à l'étape 3 :

— soit tous les certificats sont valides au moment de validation ;
— soit chaque certificat était valide au moment de signer le certificat suivant de la chaîne.

Enfin, il reste à vérifier que le certificat signant est valide au moment de validation et qu'il respecte les contraintes initialisées à l'étape 3 : contraintes de validation des certificats et cryptographiques.

5. **Vérification de la fraîcheur des informations de révocation**. Cette étape consiste à vérifier que le moment de publication des informations de révocation est suffisamment proche du moment de validation de la signature. Une contrainte définie à l'étape 3 contient la valeur maximale de fraîcheur. Cette valeur est la différence maximale acceptée entre le moment de validation et le moment de publication des informations de révocation. Cette étape permet donc de définir si oui ou non, les informations de révocation sont suffisamment récentes pour avoir confiance en celles-ci.
6. **Vérification cryptographique**. Cette étape vérifie l'intégrité des données signées, respectant les contraintes cryptographiques de l'étape 3 : elle vérifie que la valeur de la signature corresponde à l'empreinte calculée sur les données signées (au moyen des mécanismes utilisant des clés de chiffrement vus à la section 2.2) [4].
7. **Validation de l'acceptation de la signature**. Ceci implique d'autres vérifications, relatives à la signature elle-même ou aux attributs signés et non signés, présents dans la signature et prenant en compte cette fois les contraintes d'éléments de signature en plus des contraintes cryptographiques de l'étape 3.
8. Optionnellement la **présentation de la validation de signature**, qui présente les éléments utilisés (les données, l'identifiant du signataire, le moment de validation, les attributs signés et non signés, la politique de

4. Dans le cas de signatures détachées ou enveloppées, il y a seulement une représentation des données et il faut donc commencer par rechercher les données entières au moyen de ressources externes. Ainsi, une première vérification à faire est la correspondance des empreintes entre la représentation des données (déjà sous forme de hash) et les données entières (à hacher).

validation suivie), le résultat final de sortie (réussite, échec ou indéterminé), les raisons qui ont mené vers ce résultat, ainsi qu'un rapport de validation.

En sortie de ce processus de validation, on retourne un statut indiquant la réussite, l'échec ou l'indétermination de la validation, les raisons en cas d'échec ou d'indétermination ainsi que la chaîne de certificats créée à l'étape 4. Il peut aussi y avoir d'autres informations additionnelles rapportant des résultats d'étapes intermédiaires, ou précisant si certains attributs n'ont pas été utilisés, ou encore d'autres informations utiles.

3.1.2 La signature horodatée

Avant de passer à la signature en elle-même, il est important de définir plus précisément ce qu'est un horodatage. Nous avons déjà vu dans le chapitre 1, section 1.1.5, qu'un horodatage permet de lier une donnée à un moment précis et ainsi de prouver que cette donnée existait déjà à cet instant. Nous allons maintenant voir ce qu'est réellement un horodatage et les informations qu'il contient.

Un horodatage, ou jeton d'horodatage, est en fait une signature créée par une autorité d'horodatage sur des données bien particulières. Ces données, définies par le protocole d'horodatage [49], sont principalement :

— le hachage des données à horodater ;
— le moment auquel on ajoute l'horodatage, exprimé comme un temps universel coordonné (TUC) ;
— un numéro de série unique lié à l'horodatage.

Toutes ces données sont stockées dans une structure appelée TSTInfo et sont ensuite signées. Cette structure, la valeur de la signature et d'autres informations liées à la signature [5] sont ensuite stockées dans une même structure appelée SignedData, reprise par CMS (Cryptographic Message Syntax), une syntaxe de messages cryptographiques [25]. C'est cette dernière structure qui forme un horodatage.

Pour en revenir à la signature horodatée, elle joint un horodatage à une signature basique parmi les attributs non signés. Cet ajout permet de prouver que la signature existait déjà à un moment précis dans le temps. L'horodatage doit être ajouté par une autorité d'horodatage avant que le certificat signant ne soit révoqué ou n'expire. Il est donc important d'ajouter cet horodatage aussi rapidement que possible après la création de la signature, notamment pour réduire le risque de répudiation lié à cette création.

5. Comme pour une signature basique, il faut récolter les informations liées à la signature, notamment l'identifiant du certificat utilisé, l'algorithme de chiffrement et de hachage utilisés ainsi que d'autres attributs signés et non signés.

FIGURE 3.3: Signature horodatée

Au final, elle peut être validée lorsque le certificat signant a été révoqué après l'ajout de l'horodatage.

La validation de la signature horodatée

Plusieurs étapes sont nécessaires pour valider la signature horodatée.

1. **Initialisation de l'ensemble des horodatages** [6] **et du meilleur temps de signature**. Ces deux éléments seront tous deux utilisés au cours du processus. Le meilleur temps de signature est le temps le plus ancien pour lequel on est certain que la signature existait. Son initialisation se base sur une indication de temps définie en entrée ou sur le moment actuel si elle n'est pas définie. Sa valeur changera durant le processus pour être la plus exacte possible.
2. **Validation de la signature basique** comme expliqué dans la section précédente. Ce processus renvoie le statut et la chaîne de certificats utilisée.
3. **Validation des horodatages**. Dans un premier temps, il convient de vérifier que chaque empreinte d'horodatage générée respecte un format spécifié. Si ce n'est pas le cas, on ne considérera pas l'horodatage correspondant par la suite.

   Ensuite, il faut valider chaque horodatage. Cela consiste à effectuer le processus de validation d'une signature basique avec, en entrée, le jeton d'horodatage. En sortie, en plus d'envoyer les éléments transmis par ce processus, on y ajoutera le moment de génération de l'horodatage, les

6. Notons qu'il est possible d'avoir plusieurs horodatages séquentiels pour une signature horodatée. En effet, il se peut qu'il y ait une nécessité d'ajouter un horodatage si le précédent venait à expirer, ou si le but est d'avoir deux horodatages de différentes autorités.

données hachées présentes dans TSTInfo et éventuellement d'autres données présentes dans cette même structure.

Si cette validation est réussie, et que le moment de génération de l'horodatage est antérieur à la valeur du meilleur temps de signature, cette dernière est mise à jour pour correspondre au moment de génération qui indique donc un point plus ancien auquel la signature était déjà valide.

4. **Comparaison des temps**. D'une part, s'il y a plusieurs horodatages consécutifs, cette étape vérifie la concordance des temps entre ceux-ci.

   D'autre part, si la validation de la signature basique (étape 2) ne permettait pas de prouver que le certificat était valide, cette étape vérifie – au moyen du meilleur temps de signature – que la signature a été faite avant la révocation du certificat ou que les algorithmes utilisés étaient encore fiables au moment de la signature.

5. **Vérification du délai d'horodatage**. Dans le cas où une contrainte de validation spécifie un délai d'horodatage, il faut vérifier que l'horodatage n'a pas été effectué trop longtemps après la signature ; soit que le meilleur moment de signature (défini par l'horodatage le plus ancien) se situe avant la somme du délai d'horodatage et de l'attribut optionnel de signature définissant le véritable moment de signature.

6. En cas d'indétermination lors de l'étape 5 de la validation de la signature basique, c'est-à-dire lors de la vérification de la fraîcheur des informations de révocation, une seconde tentative est lancée, utilisant cette fois-ci la valeur mise à jour du meilleur temps de signature.

7. **Validation de l'acceptation de la signature**. Bien que cette validation ait déjà été effectuée lors de l'étape 7 de la validation de la signature basique, elle est effectuée à nouveau, avec cette fois, comme temps le meilleur temps de validation. Ceci permet de vérifier si les algorithmes utilisés étaient fiables à ce moment-là.

En sortie de ce processus de validation, un statut est retourné, indiquant la réussite, l'échec ou l'indétermination de la validation, accompagné du meilleur temps de signature et de la chaîne de certificats créée durant le processus (à l'étape 2). Comme pour la signature basique, il peut aussi y avoir d'autres informations additionnelles.

3.1.3 La signature avec éléments de validation à long terme ou signature LT (Long Term)

Cette signature incorpore tous les éléments requis pour la validation en tant qu'attributs non signés. Cela permet de s'assurer de la disponibilité de ces infor-

mations. En effet, la validité d'une signature horodatée peut s'établir tant que les éléments nécessaires à la validation sont disponibles en ligne. Une fois que ces informations ne sont plus disponibles ou si certains tiers n'y ont pas accès, il est indispensable de les ajouter à la signature pour que celle-ci puisse être validée.

Ces éléments de validation à long terme (Long Term Validation Data, LTVD) sont l'ensemble des certificats des chaînes de certification liées à la signature et à l'horodatage et les listes de révocation (Certificate Revocation Lists, CRL) ou les réponses OCSP (Online Certificate Status Protocol). Une CRL reprend toutes les informations concernant les certificats d'une autorité de certification, tandis qu'une réponse OCSP fournit les informations de révocation concernant un unique certificat.

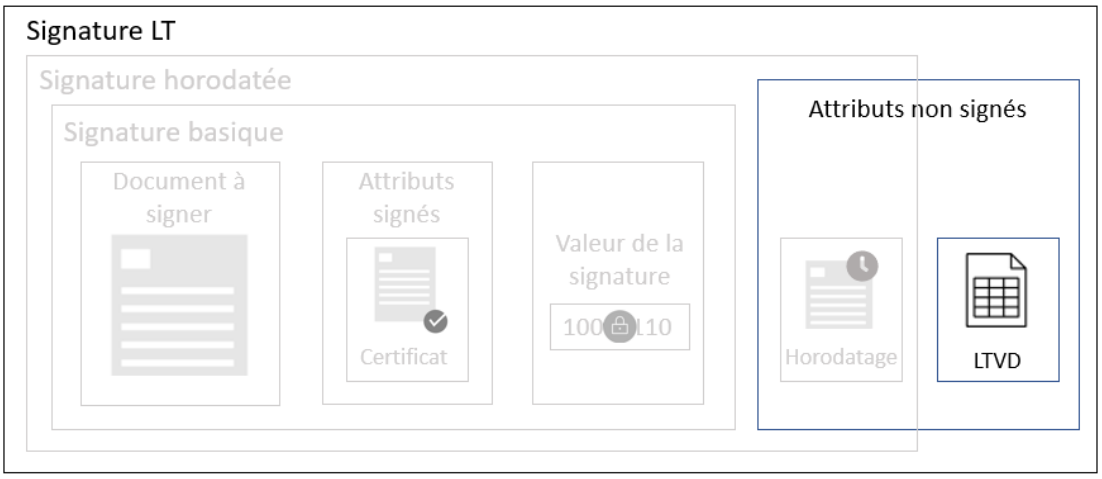


FIGURE 3.4: Signature LT

Cette signature peut être validée lorsque le certificat signant est révoqué, mais également lorsque les algorithmes utilisés ou que les tailles des clés ne sont plus à la pointe de la technologie. Cependant, notons bien que comme le précise Jean-Marc Rietsch [36], cela ne peut se faire que de deux manières :

— soit il faut parvenir à conserver la signature dans un environnement de confiance, par exemple un service d'archivage électronique fiable qui préservera la signature et son intégrité ;
— soit il faudra renforcer régulièrement la signature à l'aide de multiples horodatages, ce qui forme la classe 4 de signature ci-dessous.

La validation de la signature LT

Le processus de validation de la signature LT est exactement le même que celui de la signature horodatée. En effet, la deuxième ne diffère de la première

que par l'ajout de données de validation pouvant être utilisée lors du processus, ce qui permet d'éviter certaines réponses indéterminées. La différence se trouve donc à l'étape 2 lors de la validation de la signature basique, par l'ajout en entrée et l'utilisation de ces nouveaux éléments.

3.1.4 La signature assurant la disponibilité et l'intégrité à long terme des éléments de validation ou signature LTA (Long Term Availability)

Cette signature assure la disponibilité et l'intégrité des éléments de validation à long terme et peut aider à valider la signature malgré de nombreux éléments qui limitent sa validité. Hormis le dernier horodatage, dont le certificat doit être vérifié en ligne, le reste de la vérification peut se faire hors ligne, sans avoir accès à d'autres ressources.

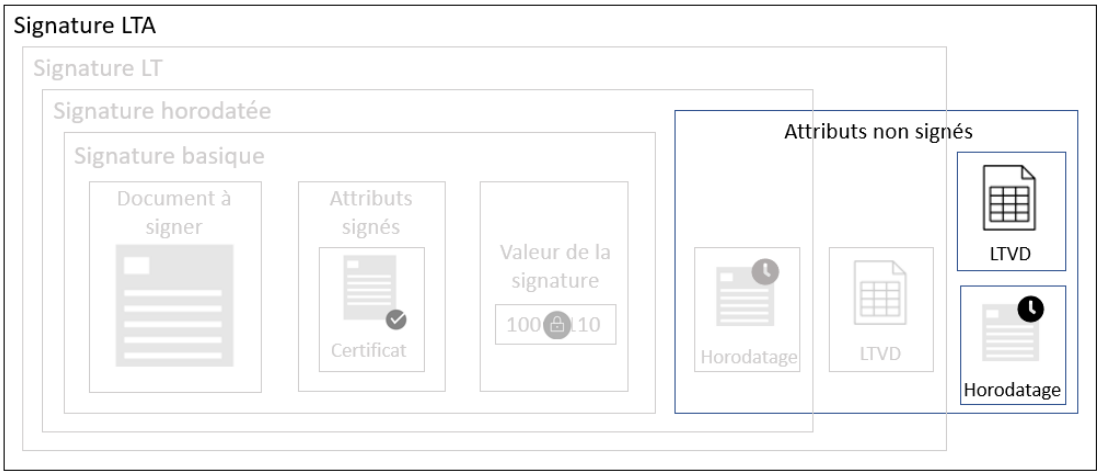


FIGURE 3.5: Signature LTA

Pour assurer cette intégrité, un horodatage doit être ajouté avant que les algorithmes utilisés deviennent vulnérables, les tailles de clés trop petites, ou que les certificats des précédents horodatages expirent ou soient révoqués. Cet horodatage doit couvrir le document ainsi que tous les autres éléments de la signature. L'ensemble est un processus itératif : avant que la protection fournie par le précédent horodatage devienne faible, un nouvel horodatage doit être ajouté, couvrant ces nouvelles informations et utilisant de nouvelles clés, des clés de longueur plus grandes que précédemment ou des algorithmes plus sécurisés.

Comme nous pouvons le voir sur la Figure 3.6, à chaque ajout d'horodatage, s'adjoindra de nouveaux éléments nécessaires à la validation à long terme, à nouveau ces éléments comprennent des certificats et des CRL ou des réponses OCSP.

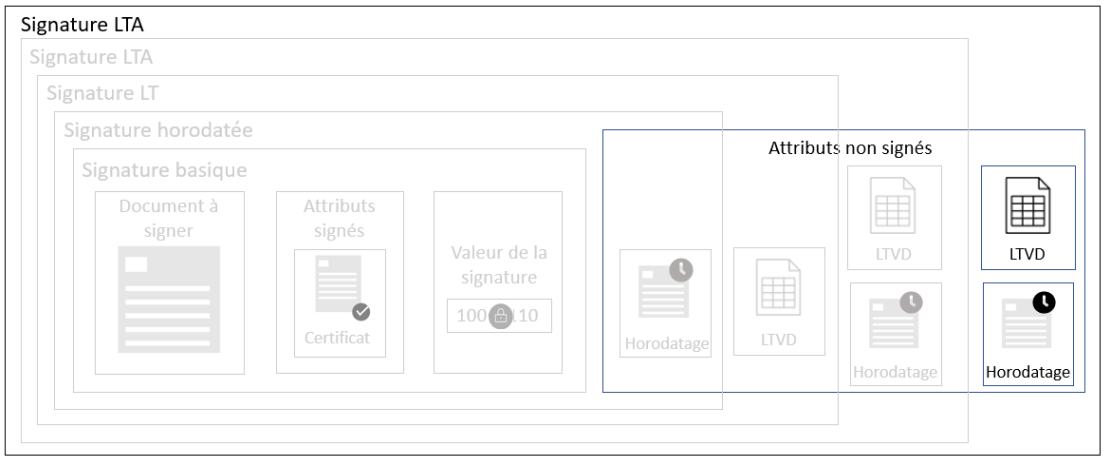


FIGURE 3.6: Signature LTA - nouvelle couche

La validation de la signature LTA

Le processus de validation est le suivant.

1. Dans le cas particulier où des Evidence Records (ER) sont donnés en plus en entrée : **vérification des ER**. Nous verrons cette structure dans la section 3.2.1 concernant ERS.
2. **Constitution de preuves de l'existence** (POE, Proof Of Existence) pour chaque objet dans la signature, c'est-à-dire préciser le moment spécifique auquel l'objet existait avec certitude. À ce stade, il s'agit du moment présent. Chaque POE ainsi créée est ajoutée à l'ensemble des POE, déjà initialisé en entrée.
3. **Validation de la signature horodatée avec éléments de validation à long terme** comme expliqué dans la section 3.1.2 concernant la signature horodatée. Si ce sous-point termine sur un statut réussi, et que les contraintes ne précisent pas qu'il faut valider des attributs de disponibilité et d'intégrité des données de validation à long terme, l'ensemble du processus de validation de la signature LTA s'arrête ici avec le statut réussi.
4. **Mise à jour du meilleur temps de signature comme POE**, avec la valeur du moment de validation retournée à l'étape précédente.
5. **Validation des horodatages**. De la même manière qu'à l'étape 3 de la validation de la signature horodatée, il faut vérifier la validité de chaque

horodatage. De plus, la fonction de hachage utilisée pour protéger les données à horodater doit être considérée comme fiable au moment de la génération de l'horodatage.

Dans le cas particulier où la validation d'un horodatage a échoué avec une indétermination, il faudra le valider dans le passé. Ce processus est expliqué à l'étape 6 ci-après. Il faut donc l'appliquer avec l'horodatage à vérifier dans le passé.

Si la validation est réussie, chaque donnée protégée par l'horodatage fournira une POE avec comme temps le moment de génération de l'horodatage.

6. **Validation de la signature dans le passé**. Ce processus est utilisé lorsque la validation d'une signature ou d'un horodatage échoue au moment présent avec un statut indéterminé. Il peut être décomposé en deux sous-étapes.

   (a) Validation du certificat dans le passé.

   Le but est de prouver la validité du certificat signant à une date dans le passé.

   Ce point se base sur le fait que si dans le passé une chaîne de certificats a su être utilisée pour valider un certificat, alors cette même chaîne devrait pouvoir être à nouveau utilisée au temps présent pour obtenir le même statut. Pour ce faire, une de ces conditions doit être satisfaite :

   — soit le statut de révocation du certificat qu'on désire valider peut être déterminé au temps présent, dû au fait qu'il n'ait pas encore expiré et que les informations de révocation sont à jour. On sait donc quand le certificat a été révoqué et donc jusque quand il était valide ;

   — soit le statut de révocation du certificat peut être vérifié à l'aide d'anciennes informations de révocation, pour lesquelles leur émetteur était encore considéré comme fiable et sous le contrôle de sa clé privée au moment du partage de ces informations. Ceci permet de prouver que le certificat ainsi que les informations de révocation existaient à cette date dans le passé.

   Au final, cette étape renvoie la chaîne utilisée et le moment auquel le certificat était valide et sa chaîne pouvait être validée. Au fur et à mesure de l'avancement du processus, ce moment est mis à jour par un moment antérieur chaque fois qu'il fait face à un certificat révoqué, un échec des contraintes cryptographiques ou un échec lié à la fraîcheur des informations de révocation.

   (b) Cette sous-étape permet de calculer la correspondance des temps pour affirmer que la signature a été générée lorsque le certificat signant était valide (existait, n'était ni révoqué ni expiré) et que les certificats de la chaîne n'étaient pas révoqués.

Ceci est effectué à l'aide des POE, et de la valeur du dernier moment auquel le certificat était valide, renvoyé par la sous-étape précédente.

En conséquence, ceci permet de sortir d'un statut indéterminé.

7. **Détermination du temps d'existence certain** : à l'aide des POE, cette étape détermine le premier moment à partir duquel l'existence de la signature peut être prouvée.
8. **Validation de l'acceptation de la signature**. Bien que cette validation ait déjà été effectuée lors de l'étape 7 de la validation de la signature basique, elle est effectuée à nouveau avec cette fois comme temps le temps d'existence certain. Ceci permet de vérifier si les algorithmes utilisés étaient fiables à ce moment-là.

En sortie des huit étapes de ce processus de validation, un statut est retourné, indiquant la réussite, l'échec ou l'indétermination de la validation, accompagné d'autres informations liées à la signature, en particulier les résultats intermédiaires, comme les résultats de validation des jetons d'horodatages.

3.2 La préservation des signatures

La préservation des signatures est étroitement liée à la validation de celles-ci. En effet, pour permettre la validation sur le long terme d'une signature, il est indispensable de la préserver. La signature LTA, dernière classe de signature vue au point précédent, est un exemple de méthode qui permet cette préservation.

Cependant, ce n'est pas la seule possibilité qui existe. Les sous-sections suivantes présentent deux autres techniques pour préserver les signatures. La première méthode, appelée Evidence Record Syntax (ERS), est un standard international d'archivage à long terme de documents électroniques et de données signées [22]. La seconde méthode, TrustChain, se base sur une décentralisation de la confiance en impliquant plusieurs institutions.

3.2.1 Evidence Record Syntax (ERS)

Comme expliqué par Tobias Gondrom [22, 23], ERS est un standard international d'archivage à long terme de documents électroniques et de données signées. Son but est de préserver l'intégrité de documents, signés ou non, en mutualisant les données nécessaires à cette préservation. Ceci permet d'avoir moins de données à stocker qu'en conservant séparément chaque document. Pour cela, une structure comportant plusieurs documents, des valeurs de hachage et un horodatage va être formée.

Au cours du temps, cette structure va se complexifier pour ajouter des métadonnées permettant la préservation à long terme : de nouvelles valeurs de hachage, de multiples couches d'horodatages ainsi que d'autres données nécessaires à la validation. Lorsqu'il sera nécessaire de vérifier qu'un document est intègre, une autre structure, appelée Evidence Record, contenant uniquement les données en lien avec le document, va être construite.

Voyons l'ensemble du processus itératif en détail.

3.2.1.1 Établissement de la structure de préservation de multiples documents

Création de la structure initiale

Tout d'abord, il faut créer la structure qui va contenir des valeurs de hachage et un unique horodatage initial pour protéger un ensemble de documents. Pour ce faire, on va procéder en suivant ces étapes :

1. Prendre plusieurs documents dont on a envie de préserver l'intégrité ;
2. Choisir une fonction de hachage et hacher séparément chaque document ;
3. Former l'arbre de hachage : remonter et hacher la concaténation des hashs (disposé par ordre croissant) jusqu'à obtenir la racine de l'arbre ;

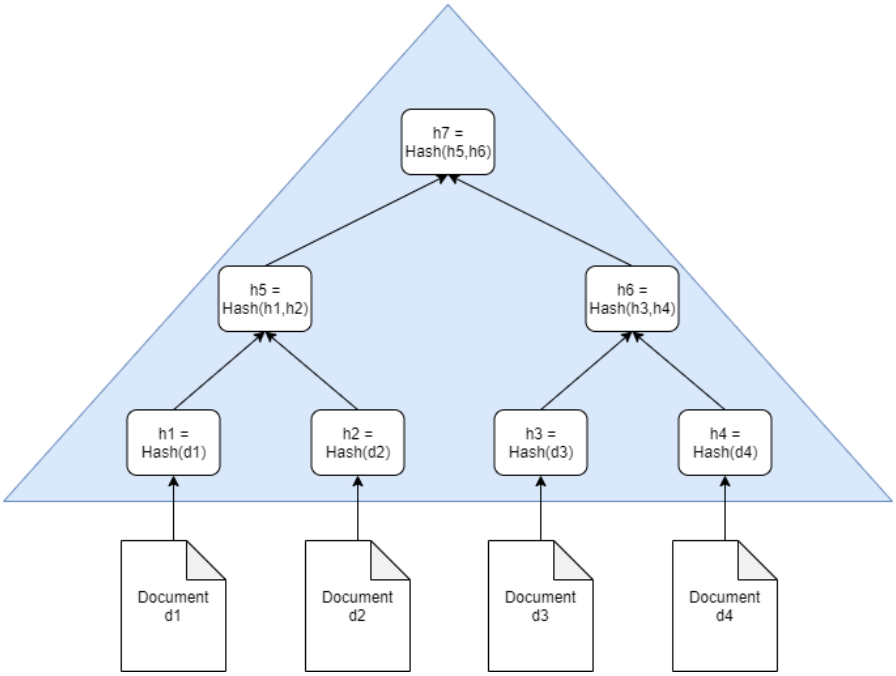


FIGURE 3.7: Arbre de hachage

4. Obtenir un horodatage uniquement sur le hash racine, utilisant la même fonction de hachage que celle de l'arbre ou une autre.

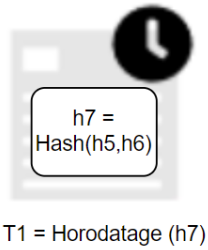


FIGURE 3.8: Horodatage sur le hash racine

À ce stade-ci, tous les documents sont protégés par l'unique horodatage et il est facile de prouver que les documents sont intègres. En hachant à nouveau chaque document et en reformant l'arbre de hachage, il faut uniquement vérifier que le hash racine corresponde à la valeur de hachage obtenue en déchiffrant l'horodatage.

Cette *protection* n'est cependant pas éternelle. Elle peut être corrompue par deux cas principaux [22] :

— soit l'horodatage devient peu fiable. Cela peut survenir si l'algorithme de chiffrement utilisé devient faible, si la longueur des clés devient trop petite, ou si le certificat de l'horodatage expire ou est compromis (et donc révoqué) ;

— soit la fonction de hachage utilisée pour construire l'arbre devient faible, ce qui se présente plus rarement. Par exemple, la fonction de hachage MD5 est devenue obsolète après plus de 20 ans [47]. SHA-1, quant à elle, était considérée comme fiable pendant plus de 10 ans, avant de trouver des collisions avec une complexité inférieure à 2^{69} opérations de hachage en 2005 [46]. En 2011, elle a officiellement été définie comme obsolète par le NIST (National Institute of Standards and Technology) [43].

Ces deux cas entraînent des ajouts de données différentes à la structure globale : le premier implique un *renouvellement de l'horodatage* et le second *un renouvellement de l'arbre de hachage.*

Renouvellement de l'horodatage

Avant que l'horodatage actuel ne devienne peu fiable, il est nécessaire de renouveler la protection de l'arbre à l'aide d'un nouvel horodatage. Un nouvel horodatage est donc créé sur base de l'ancien horodatage en utilisant la même fonction de hachage que celui-ci, mais avec d'autres clés et potentiellement une autre longueur de clés ou un autre algorithme de chiffrement (en fonction de la cause de la faiblesse de l'horodatage).

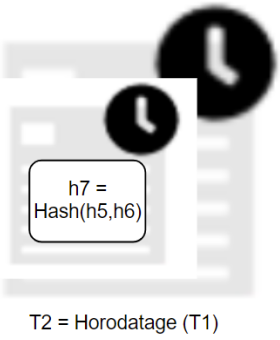


FIGURE 3.9: Renouvellement de l'horodatage

Renouvellement de l'arbre de hachage

Un autre cas de figure se présente lorsque la fonction de hachage utilisée dans la structure devient faible. Dans ce cas-ci, il faut renouveler entièrement l'arbre de hachage avec une nouvelle fonction de hachage et ajouter un nouvel horodatage à la racine de celui-ci [7].

On recrée un arbre de hachage complet avec une nouvelle fonction de hachage. Pour ce faire, nous devons :

1. Choisir une nouvelle fonction de hachage ;
2. Hacher tous les documents avec cette fonction ;
3. Hacher l'ensemble des arbres de hachage et des horodatages précédents ;
4. Pour chaque document, hacher la concaténation de son hash de l'étape 2 avec celui de l'étape 3 ;
5. Utiliser les valeurs de hachage de l'étape 4 comme feuilles, et reconstruire un arbre de hachage comme au point 3 de la création de la structure initiale.

Ainsi cette structure totale va avoir au fur et à mesure du temps des métadonnées en plus.

3.2.1.2 Vérification d'un document

Lors d'un litige, il est nécessaire de prouver l'intégrité et l'authenticité d'un document. C'est à ce moment qu'intervient un Evidence Record qui est l'élément clé pour la vérification : il contient tous les éléments qui y sont nécessaires.

Création d'un Evidence Record

Tout d'abord, pour avoir une preuve d'existence d'un seul document, on peut créer son **arbre réduit** formé à partir de l'arbre de hachage se rapportant au document. Il contient une liste des valeurs de hash nécessaires pour prouver le document concerné :

1. Le hash du document (h1, cf. Figure 3.7) ;
2. Le hash *frère* (h2), soit celui qui a le même *père* que le hash du document. Ainsi, en ayant le hash du document et celui de son frère on peut calculer celui de leur *père* (h5 = H(h1,h2)) et donc ne pas le stocker ;

7. Notons que l'horodatage devra également utiliser un nouvel algorithme de hachage s'il utilisait le même que l'arbre.

3. Le hash *frère* (h6) du *père* précédemment calculé (h5), et ainsi de suite, jusqu'à arriver à la racine de l'arbre (h7) qu'il ne sera pas nécessaire de stocker, vu qu'il est calculable à partir de tous ses fils.

Ces valeurs de hachage forment l'arbre réduit. En y ajoutant l'horodatage de la racine de l'arbre ainsi que d'autres attributs optionnels, tels que la fonction de hachage utilisée, on forme un **tampon d'archivage**, en abrégé **ATS** (Archive Time-Stamp).

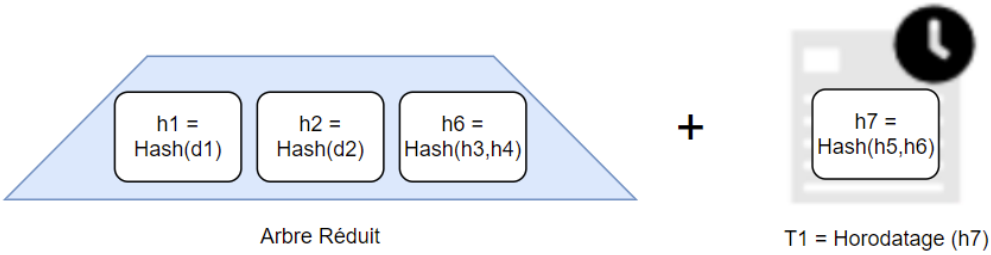


FIGURE 3.10: Tampon d'archivage

Ensuite, pour chaque renouvellement d'horodatage effectué, on ajoute un ATS à la structure. Chaque ATS contient ce nouvel horodatage et, comme arbre réduit, le hash de l'horodatage précédent. Plusieurs ATS forment une **chaîne de tampons d'archivage**. De plus, pour chaque renouvellement d'arbre de hachage, ce nouvel arbre – réduit – sera à la base d'une nouvelle chaîne d'ATS. Plusieurs chaînes forment ensemble une **séquence de chaînes de tampons d'archivage**.

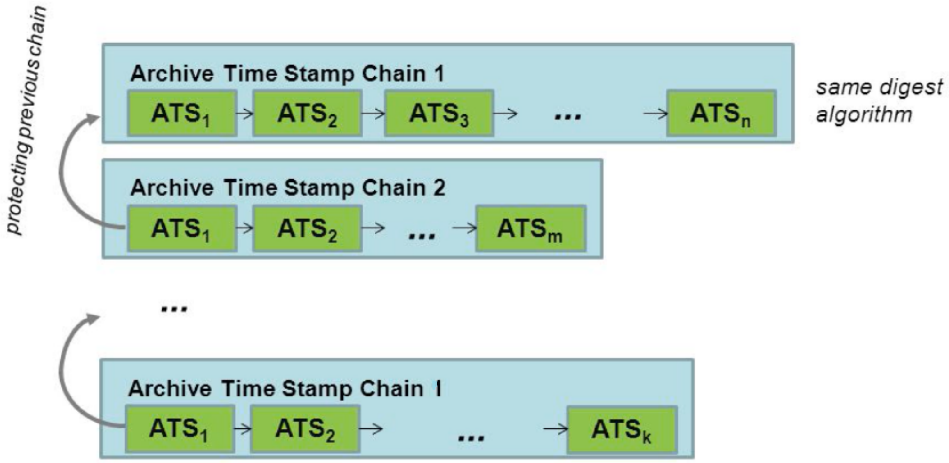


FIGURE 3.11: Séquence de chaînes d'ATS [5]

Au final, l'Evidence Record contient cette séquence de chaînes de tampons d'archivage et d'autres données de validation, comme des certificats et des informations de révocation.

Vérification d'un Evidence Record

Pour vérifier un Evidence Record se rapportant à un document signé ou non, il faut vérifier sa séquence de chaînes de tampons d'archivage à l'aide des étapes suivantes.

1. Vérifier que le **premier ATS** de la première chaîne contient le hash du document (à calculer à partir du document et voir s'il y a une correspondance).
2. Vérifier **chaque chaîne de tampons d'archivage**.
   — Chaque ATS doit être valide au moment de l'ajout de l'ATS suivant dans la chaîne.
   — Dans chaque chaîne, la même fonction de hachage doit avoir été utilisée et cette fonction doit être sécurisée au moment du premier ATS de la chaîne.
3. Vérifier que le premier ATS de toutes les chaînes, sauf la première, contient un hash de la concaténation du hash du document et du hash de l'ensemble des arbres de hachage et des horodatages précédents.

Afin de valider les ATS à l'étape 2 ci-dessus, il faut suivre les étapes suivantes :

1. Calculer le hash des données [8] à l'aide de la fonction de hachage spécifiée dans l'ATS ;
2. Chercher cette valeur de hachage dans l'arbre réduit. Si elle n'est pas présente, les données ont donc été modifiées et on s'arrête ;
3. Récréer l'arbre de hachage : prendre les hashs *frères* et calculer le *père* en concaténant les hashs et en utilisant la fonction de hachage. Le *père* sera utilisé avec ses propres hash *frères* du niveau supérieur : concaténés et hachés à nouveau pour obtenir le hash *père* encore plus haut. Continuer jusqu'à calculer le hash racine ;
4. Vérifier l'horodatage. Bien que ce ne soit pas précisé dans le standard RFC, il faut valider l'horodatage à l'aide de la validation de la signature dans le passé (comme pour la validation de la signature LTA, section 3.1.4 étape 6) avec toutes les informations nécessaires.

8. Celles-ci sont donc : pour le premier ATS de la première chaîne, le document ; pour chaque ATS à l'intérieur d'une chaîne, l'horodatage du dernier ATS ; et pour chaque ATS en début de chaîne, la concaténation du document et de tous les arbres et horodatages précédents.

Cette vérification de l'ER prouve que les données ont existé à un moment donné, spécifié par l'horodatage de l'ATS initial. Ces preuves d'existence sont utilisées comme POE lorsque l'ER se rapporte à une signature.

Après avoir eu le besoin de créer un Evidence Record pour prouver l'intégrité d'un document, deux possibilités s'offrent à nous :

— soit on peut conserver indépendamment cet ER, en ajoutant d'autres horodatages. Dans ce cas-là, cela ne protège que le document dont l'ER fait la preuve, et nécessite à nouveau beaucoup de métadonnées pour l'unique document : il s'agit donc de reprendre les mécanismes de conservation d'un document unique ;

— soit on ne conserve pas cet ER. Dans le cas où nous avons à nouveau besoin d'une preuve, on créera un nouvel ER sur base de la structure de préservation des multiples documents en permanence mise à jour.

3.2.2 TrustChain 1.0

Une autre manière de préserver des signatures est la méthode appelée TrustChain [7]. Celle-ci décentralise la confiance en un unique prestataire en impliquant plusieurs institutions (Figure 3.12). Celles-ci fonctionnent ensemble, pour confirmer la validité de signatures électroniques et les insérer dans une *blockchain* immuable.

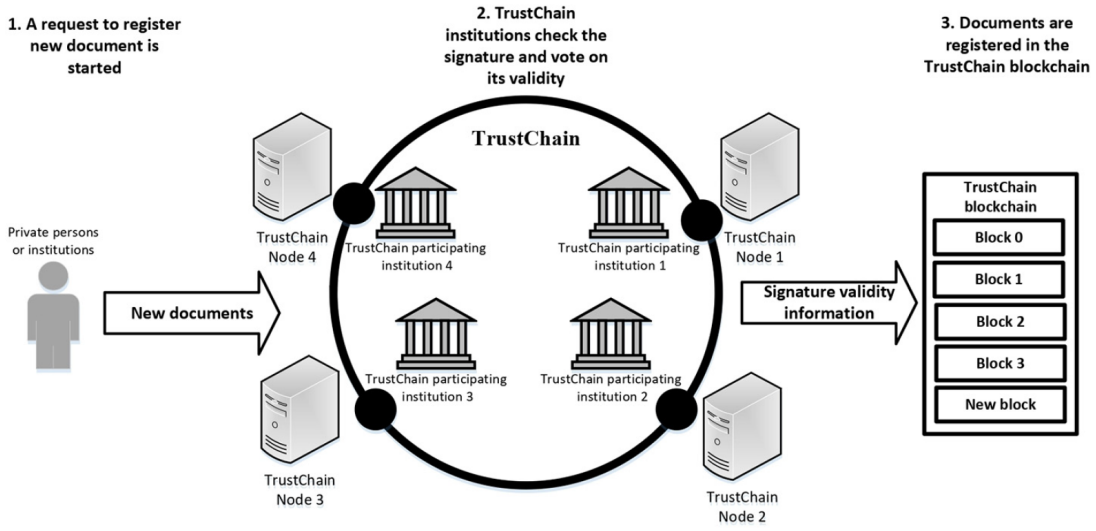


FIGURE 3.12: TrustChain 1.0 [8]

Une *blockchain* est une chaîne formée par une suite de plusieurs blocs contenant de l'information. Chacun de ces blocs dépend du bloc le précédant, ce qui donne

au final une *chaîne*. Cette dernière étant disponible publiquement, toute personne peut avoir accès aux informations qu'elle contient.

Dans le cas de la TrustChain, les caractéristiques de la chaîne sont les suivantes :
— toute personne peut voir le contenu de la chaîne ;
— toute personne peut demander d'ajouter du contenu à la chaîne ;
— seules certaines institutions peuvent écrire dans la chaîne ;
— le stockage contient uniquement des valeurs de hachages des documents pour éviter d'avoir une chaîne très lourde. De plus, ceci empêche la compréhension des informations stockées, et donc assure un certain degré de confidentialité.

Création d'un nouvel enregistrement

Toute personne ou institution – pas nécessairement les institutions participantes dans la TrustChain – peut désirer ajouter des informations liées aux documents signés qu'ils veulent préserver dans la TrustChain. Ces informations doivent être soumises sous forme d'enregistrement, appelé *record* en anglais.

Il y a donc un processus à suivre pour créer un nouvel enregistrement (Figure 3.13).

1. Au départ, il faut lire le document signé et vérifier sa validité en contactant l'autorité de certification.
2. Si la signature est invalide, le processus s'arrête.
3. Dans le cas où la signature est valide, il faut d'abord calculer une valeur de hachage du document et de la signature. Ensuite, il faut former un enregistrement sur base de ce hash calculé, du lien dirigeant vers le document, d'un horodatage et de métadonnées. Enfin, cet enregistrement est envoyé à une institution participante dans l'espoir qu'il soit ajouté à la TrustChain.

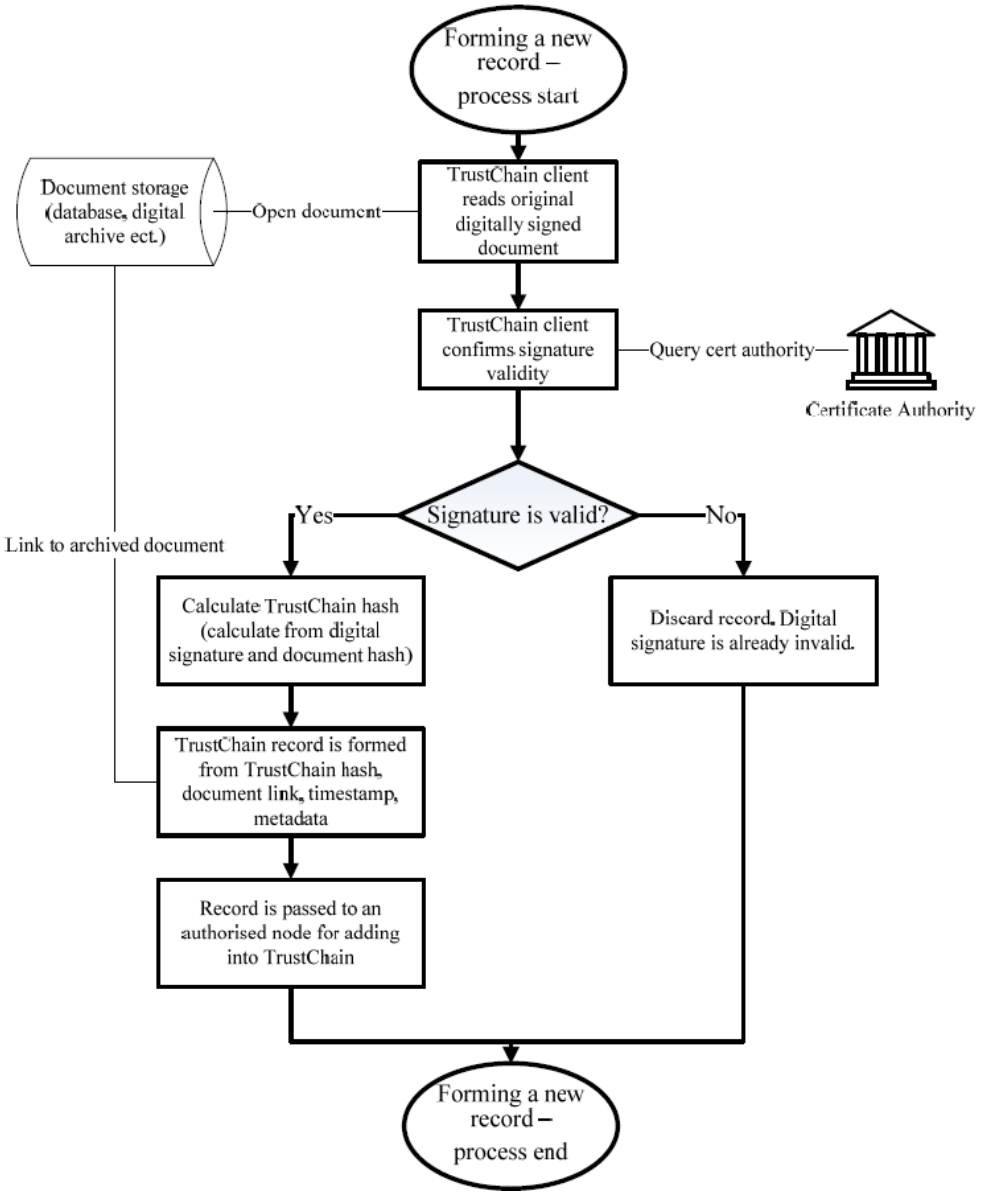


FIGURE 3.13: TrustChain 1.0 : création d'un nouvel enregistrement [7]

Ajout d'un nouveau bloc

L'ajout d'un bloc à la chaîne ne peut se faire que par un noeud – ou institution – participant. Durant ce processus, le noeud vérifie des enregistrements jusqu'à en avoir un certain nombre validés. Il les assemble et forme un nouveau bloc. Ce bloc, avant d'être ajouté à la chaîne, va devoir être accepté au moyen d'un vote à la majorité par les autres noeuds participants. Si ce n'est pas le cas, le processus

doit recommencer à collecter des enregistrements. Si le bloc proposé est accepté, le noeud calcule un hash à partir des enregistrements, des données du vote ainsi que des informations du dernier bloc de la chaîne. Ce hash est repris dans le bloc. Enfin, ce nouveau bloc est ajouté à la suite de la chaîne (Figure 3.14).

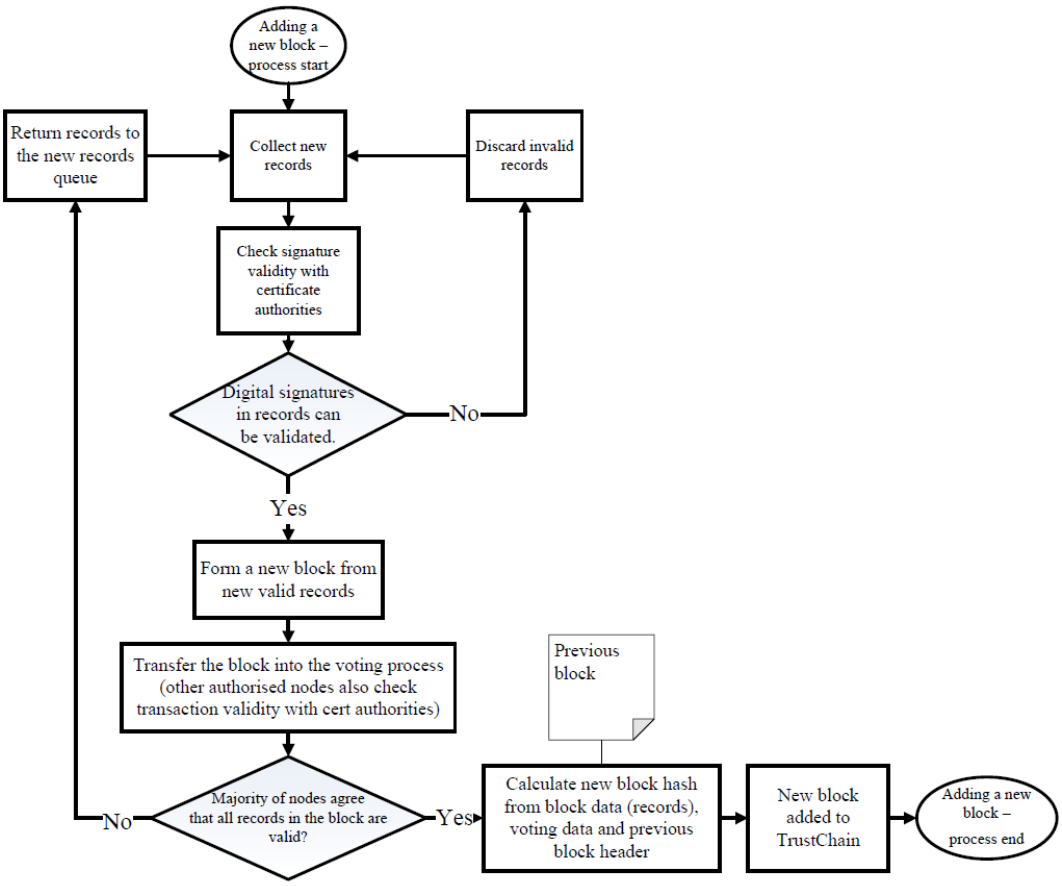


FIGURE 3.14: TrustChain 1.0 : ajout d'un nouveau bloc [7]

Contenu d'un bloc

Nous pouvons donc plus clairement citer toutes les informations contenues dans un bloc. Comme la Figure 3.15 le montre, il contient les différents enregistrements contenus dans le bloc, les informations concernant le vote ayant validé l'ajout de ce bloc à la chaîne, le hash du contenu du bloc et le hash du bloc précédent.

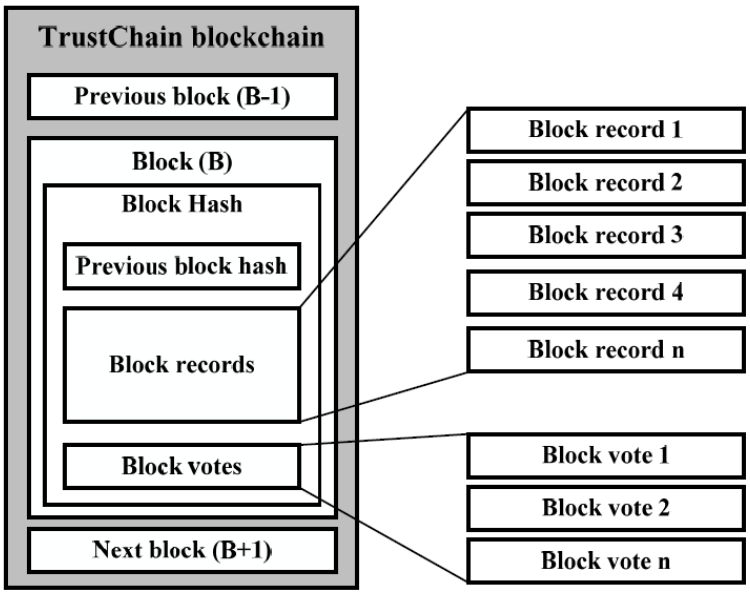


FIGURE 3.15: Bloc de la TrustChain [7]

Vérification d'un enregistrement, pour preuve de signature

Afin de vérifier un enregistrement, pour prouver la validité d'une signature, plusieurs étapes doivent être suivies (Figure 3.16). Ce processus de vérification peut être suivi par toute personne ou institution désireuse de vérifier l'intégrité d'un document.

D'abord, il s'agit de trouver le bloc contenant l'enregistrement recherché. Ce n'est pas forcément évident, puisque la chaîne peut devenir longue et l'ordre entre les blocs ne dépend que du moment d'ajout à la chaîne. Cela peut être géré par des ressources extérieures retenant les informations nécessaires.

Lorsque l'enregistrement a été lu dans le bloc correspondant, on peut trouver le lien référant vers le document original. On calcule ensuite le hash de ce document, et on regarde s'il correspond au hash se trouvant dans l'enregistrement. Si c'est le cas, on peut dire que le document est intègre.

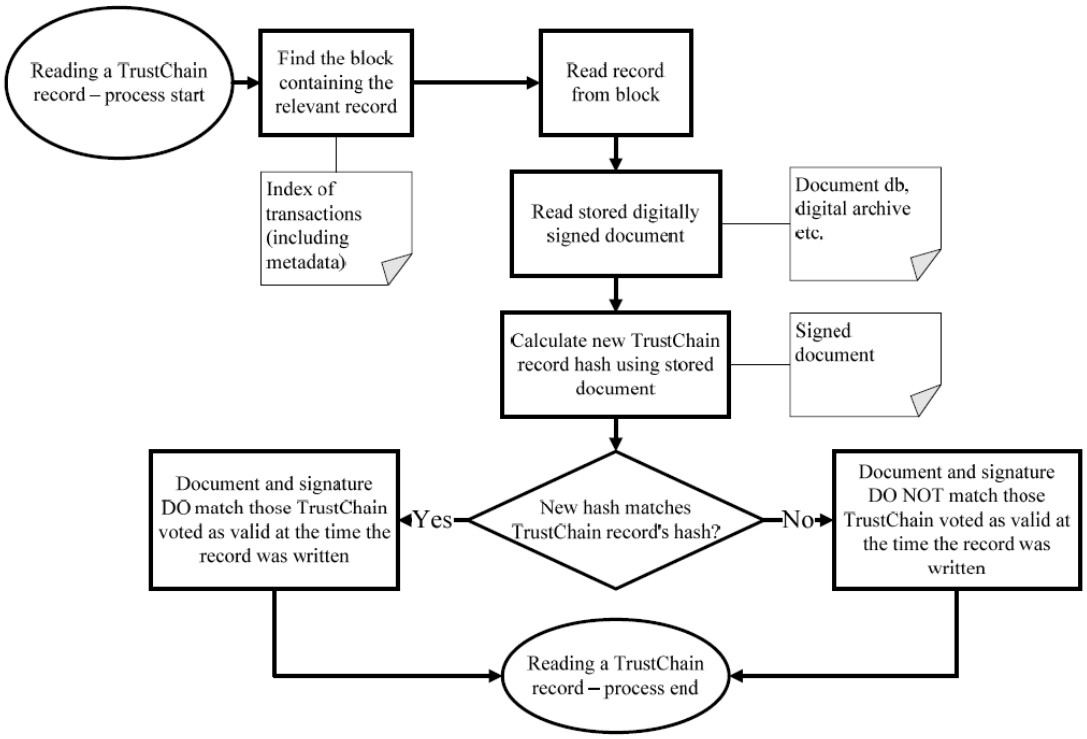


FIGURE 3.16: TrustChain 1.0 : vérification d'un bloc [7]

3.2.3 TrustChain 2.0

Les auteurs de la première version de TrustChain ont mis en avant un problème majeur lié à celle-ci : la confidentialité des différents documents signés n'est pas assurée. En effet, bien que tous les documents ne sont pas directement lisibles depuis la chaîne rendue publique (ils sont uniquement représentés sous forme de hash), lors du vote à la majorité effectué par les différents noeuds participants, les documents vont être entièrement visibles pour tous ces votants.

Ils ont donc pensé à une autre méthode, basée sur le même principe de *blockchain*, la TrustChain 2.0 [8]. Celle-ci, au lieu de stocker des enregistrements contenant des informations sur des documents signés, va contenir des enregistrements contenant des informations sur leurs certificats.

La structure est donc exactement la même que TrustChain 1.0 : une suite de blocs, contenant chacun le hash du bloc précédent, un ensemble d'enregistrements (cette fois-ci contenant des informations sur des certificats) et un ensemble de votes venant des noeuds participants indiquant s'ils valident le bloc.

Création d'un nouvel enregistrement

Pour préserver une signature, il faut demander l'ajout des certificats qui y sont liés (le certificat signant ainsi que ceux de sa chaîne de certification) à la TrustChain.

Pour ce faire, un processus permettant la création d'un enregistrement lié à ce certificat doit être suivi (Figure 3.17).

1. Lire le certificat et vérifier qu'il n'est pas déjà présent dans la TrustChain.
2. Vérifier que le certificat est encore valide.
3. S'il est valide ou s'il se trouve dans la TrustChain et n'est pas expiré, regarder son état de révocation et créer un nouvel enregistrement, contenant le hash du certificat, le lien vers le certificat complet et les dates de validité ou de révocation.
4. Dans les autres cas, la blockchain n'est pas modifiée.

Ajout des informations de révocation des certificats

Avant l'ajout d'un bloc à la chaîne, il est nécessaire de parcourir tous les certificats déjà présents dans la chaîne afin d'ajouter, pour chacun d'eux, des informations en cas de révocation.

Le processus recherche les informations de révocation de chaque certificat, à l'aide des CRL ou de réponses OCSP. Si un certificat n'était pas indiqué comme révoqué dans la TrustChain, alors que de nouvelles informations sont disponibles, un nouvel enregistrement de révocation est créé et ajouté au prochain bloc de la chaîne (Figure 3.18).

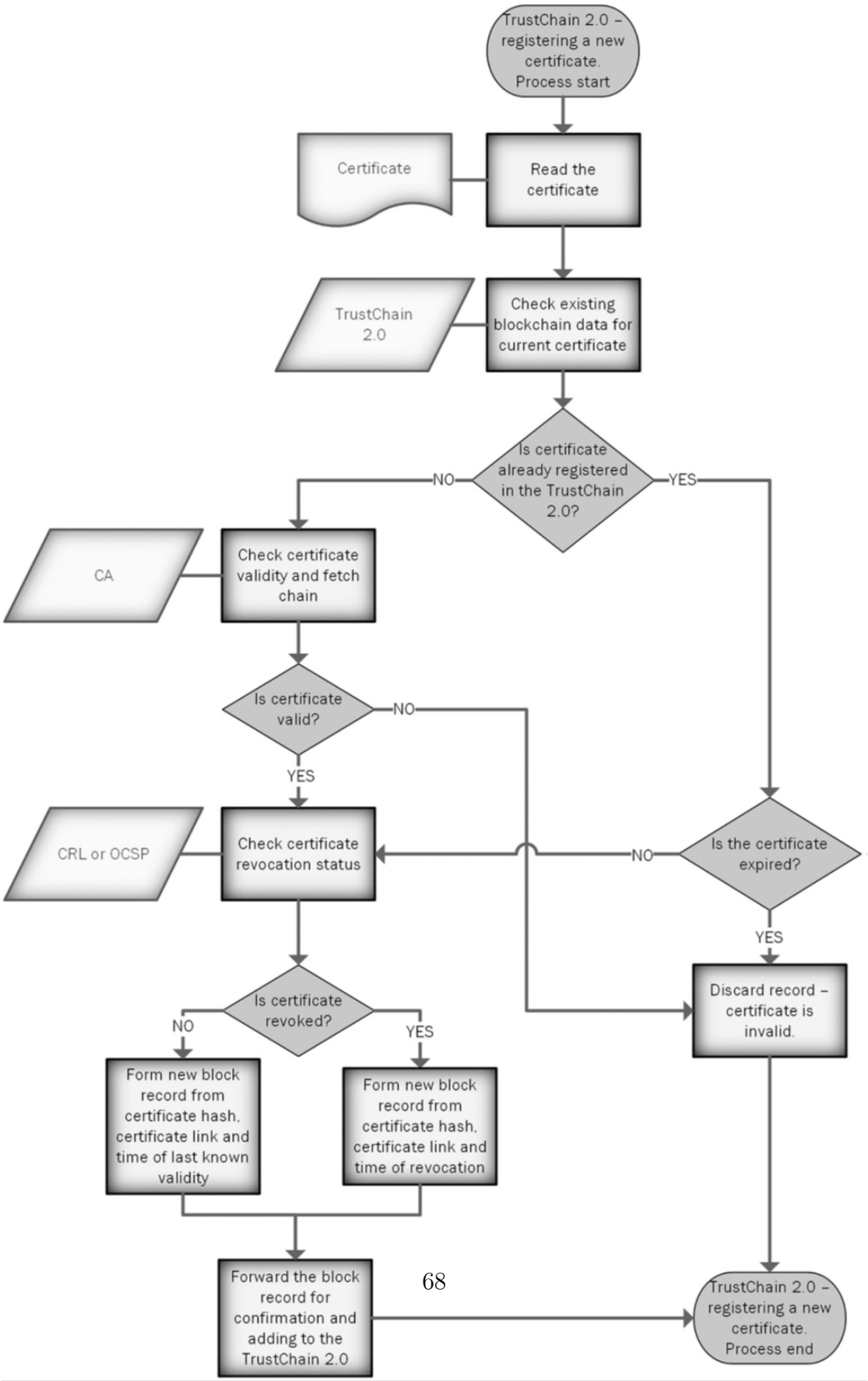


FIGURE 3.17: TrustChain 2.0 : création d'un enregistrement [8]

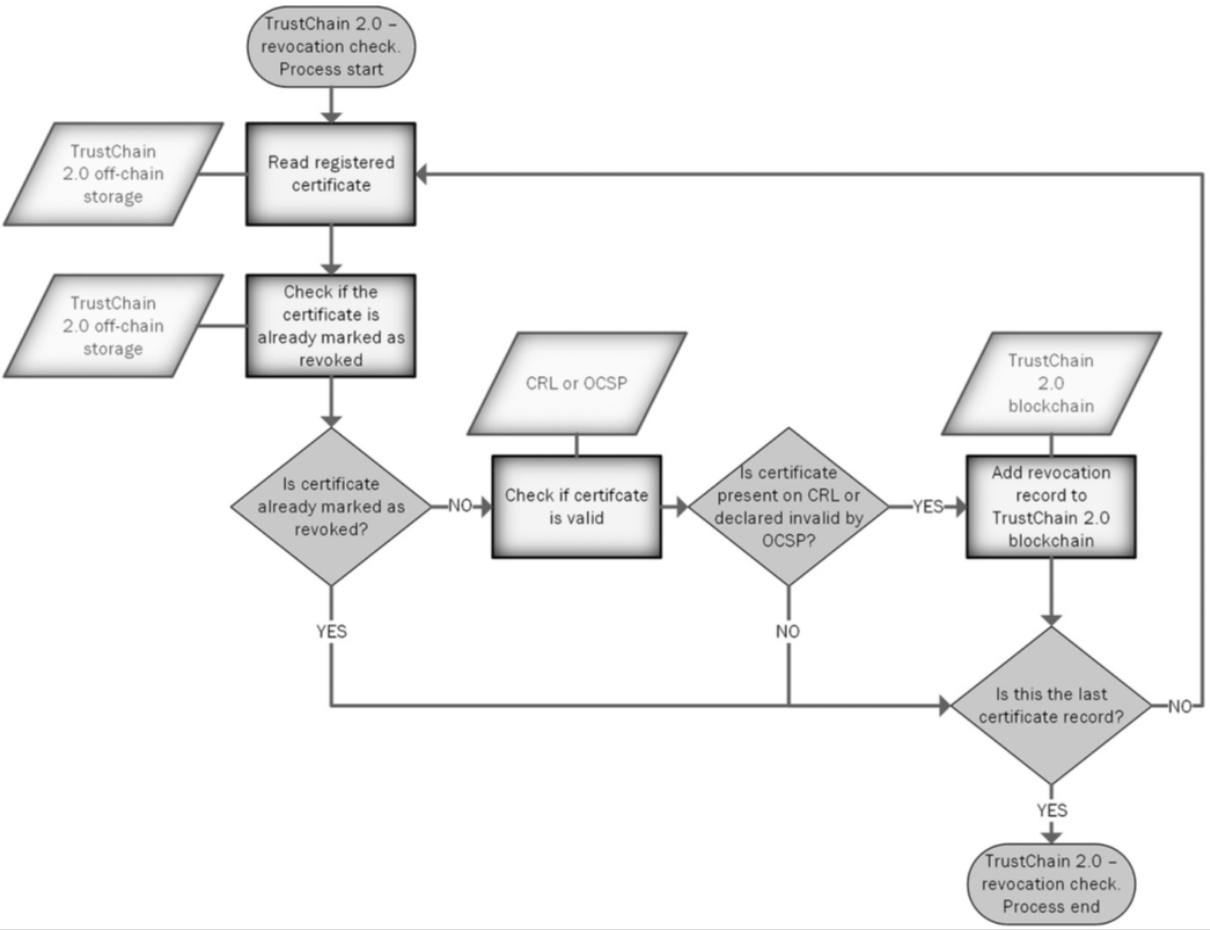


FIGURE 3.18: TrustChain 2.0 : ajout des informations de révocation [8]

Validation d'un certificat

Si un certificat est présent dans la chaîne de certification, il est possible de **valider une signature** se basant sur ce certificat.

Pour ce faire, il suffit de vérifier que la date de la signature tombe dans les dates de validités prouvées par la chaîne et qu'il n'y a pas d'informations de révocation concernant le certificat (Figure 3.19). Ces informations sont trouvées dans la *blockchain* et donc prouvées comme intègres.

Pour optimiser la recherche des informations concernant le certificat et ne pas devoir parcourir toute la chaîne depuis le premier bloc, une structure *hors chaîne* peut être fabriquée pour trouver plus rapidement à quel(s) bloc(s) se référer.

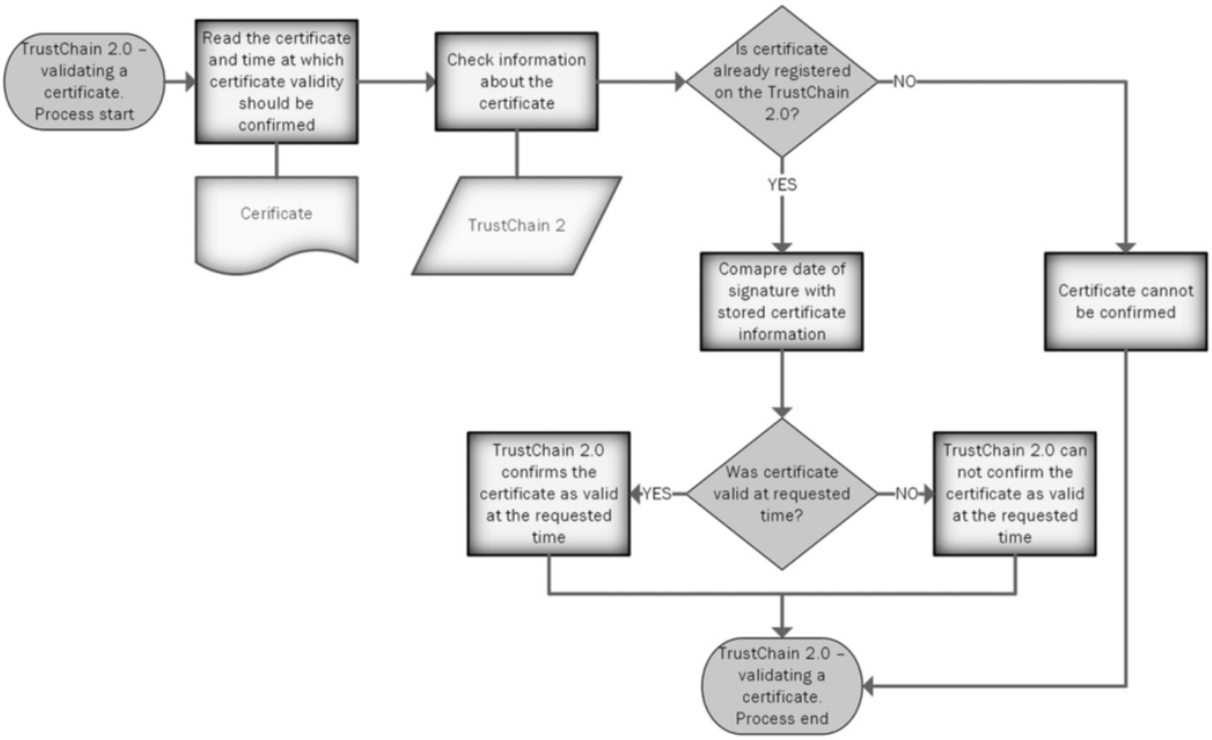


FIGURE 3.19: TrustChain 2.0 : validation d'un certificat [8]

Ainsi on peut valider une signature n'ayant même pas été annoncée à la TrustChain. Dans le cas où un signataire utilise plusieurs fois le même certificat pour signer, il annonce son certificat pour sa première signature. Toutes les signatures suivantes (se basant sur ce même certificat et donc ayant la même chaîne de certification) bénéficient de la même confiance.

3.3 Bilan des méthodes de préservation

Dans les sections précédentes, nous avons expliqué les principales méthodes pour préserver une signature électronique sur le long terme. Cette section va, quant à elle, nous permettre de cerner les principaux avantages et inconvénients de chacune de ces méthodes ainsi que les bonnes pratiques liées à leur utilisation.

Augmentation de signature

Tout d'abord, nous avons vu la signature basique. D'un point de vue juridique, cette signature est avancée, voire qualifiée, et repose donc sur un certificat (qualifié ou non). Elle permet de détecter une modification du document et d'identifier

le signataire. Cependant, cette mesure d'intégrité et d'authenticité ne dure pas indéfiniment, le certificat signant expirant entre 2 et 5 ans après sa création. Passée cette période, ou si le certificat est révoqué plus tôt, on ne peut plus avoir confiance ni en l'intégrité ni en l'authenticité du document signé.

Pour prolonger cette sécurité, l'augmentation de signature a été introduite. Au plus la classe est élevée, au plus longtemps la signature est protégée. En effet, la signature horodatée et par le même biais, la signature LT, permettent uniquement de valider la signature si le certificat signant est encore valide ou s'il a été révoqué, mais pas s'il a expiré. Pour ce faire, il est important d'apposer l'horodatage directement après la création de la signature [9], mais également d'augmenter la signature jusqu'à une signature LTA rapidement. De fait, si le certificat signant expire avant cette augmentation, la signature sera invalide.

Nous nous sommes demandé pourquoi la signature horodatée ne peut pas être considérée comme valide si le certificat signant a expiré après l'apposition de l'horodatage, contrairement au cas d'une révocation. En effet, après la date de révocation ou d'expiration, le constat est le même, le certificat ne peut plus être considéré comme fiable et n'est donc plus utilisable.

Nous avons confronté la norme européenne ETSI EN 319 102-1 [18], à la spécification ETSI TS 119 102-1 [19] plus récente (2018 comparé à 2016) qui corrige cette particularité en rendant l'expiration et la révocation équivalentes. Cette dernière spécification n'est pas encore une norme européenne, mais nous pouvons nous attendre à ce changement.

Cette dernière spécification permet alors à une signature horodatée d'être validée si le certificat signant a été révoqué, mais également s'il a expiré après la date de création de l'horodatage. La signature horodatée et LT pourront donc être validées jusqu'à ce que le certificat de l'horodatage expire ou soit révoqué.

Un point à souligner se rapporte à l'utilisation des horodatages. Lorsqu'un seul horodatage protège l'ensemble des données (signature, horodatages précédents, données de validation, etc.), il est important pour le prestataire qui s'occupe de préserver la signature de rester attentif à la révocation de son certificat et aux évolutions cryptographiques (liées à la taille des clés, aux fonctions de hachages et aux méthodes de chiffrement utilisées).

Pour être plus robuste, il peut demander plusieurs horodatages à différentes AH (ou plusieurs horodatages, reposant sur des certificats différents, à la même AH). Si l'un d'eux n'est plus valide, dû à un certificat révoqué, la signature pourra toujours être validée à l'aide d'un des autres horodatages. C'est d'ailleurs pour

9. Notamment si on impose un délai d'horodatage lors de la validation d'une signature horodatée (à l'étape 5).

cette raison que plusieurs horodatages peuvent être présents dans l'attribut non signé de la signature horodatée.

La signature LTA contient toutes les informations nécessaires à sa vérification sur le long terme, ce qui est à la fois un avantage et un inconvénient.

C'est un inconvénient, car la quantité de données à stocker au cours du temps ne cesse d'augmenter, avec de plus en plus d'horodatages et de certificats. Si la signature conserve les CRL, et non les réponses OCSP, cette quantité est encore plus importante, car la liste contient l'ensemble des certificats révoqués pour une même AC. Là où une réponse OCSP ne contient que la réponse liée à un unique certificat, celui concerné.

C'est aussi un avantage, car lors de la validation, il n'y a pas besoin d'aller demander des CRL ou des réponses OCSP à des ressources extérieures telles que des autorités de certification. Hormis pour le certificat lié au dernier horodatage, la validation entière peut se faire hors ligne. Il n'y aura pas d'informations manquantes amenant à un statut indéterminé, en prenant l'hypothèse que toutes les informations ont été ajoutées en temps requis, autrement dit quand elles étaient encore disponibles en ligne. La validation hors ligne peut également s'opérer pour la signature LT.

Un point valable pour toutes les classes de signature concerne les formats. En effet, si un utilisateur utilise des formats de signatures standardisés au niveau européen par l'ETSI (CAdES, XAdES et PAdES), il aura une sécurité supplémentaire quant à l'interopérabilité entre différents prestataires fournissant un service de préservation de signature électronique. Si son prestataire cesse ses activités, il n'aura pas de mal à trouver un nouveau prestataire avec la capacité de préserver sa signature.

Un dernier point important pour une signature, peu importe sa classe, est qu'il faut s'assurer de la disponibilité du document signé. En effet, pour toute validation, il faut recalculer le hash du document et donc y avoir accès. Il faut donc conserver le document dans un autre système qui permettra de le récupérer en cas de besoin.

ERS

Outre la signature LTA, ERS permet de préserver la validité d'une signature électronique sur le long terme, mais ces méthodes ont deux différences principales.

La première différence concerne la conservation des documents signés. En effet, ERS prend en compte les exigences d'un service d'archivage à long terme [23] et permet non seulement de préserver la signature, mais également n'importe quel

document ou donnée. C'est un grand avantage, car le document lié à une signature peut donc être protégé avec sa signature, alors que dans le cas d'une signature augmentée, le document est protégé uniquement si la signature est enveloppante. Tous les éléments nécessaires à la validation de la signature se trouvent donc dans une même structure et sont dès lors facilement accessibles.

La seconde différence réside dans la diminution du nombre d'horodatages à ajouter au cours du temps. Il est possible de s'apercevoir de cette diminution qualitativement et assez intuitivement. Pour simplifier, imaginons que la signature LTA ainsi que l'arbre de hachage d'ERS ne sont protégés que par un unique horodatage, même si pour ERS (comme pour la signature LTA), il est possible d'ajouter plusieurs horodatages du même ou de différentes autorités d'horodatage, dans le but d'améliorer la sécurité en cas de révocation.

Avant que l'horodatage ne devienne invalide, un nouvel horodatage est ajouté. Nous avons donc un nouvel horodatage pour chaque signature LTA. Pour ERS, il n'y a qu'un nouvel horodatage pour tout l'arbre de hachage, pouvant contenir un nombre indéfini de signatures et de documents. La quantité de données est donc bien inférieure qu'avec l'utilisation de multiples signatures LTA.

Néanmoins, n'oublions pas que si la fonction de hachage devient peu fiable, un nouvel arbre de hachage doit être construit sur base des documents et signatures préservés pour ERS, tandis que pour LTA, un nouvel horodatage doit être créé par document. La quantité de données stockées augmente dans les deux cas, mais la nécessité de changer de fonction de hachage se présente bien moins souvent que celle de renouveler un horodatage : 3 ans en général pour un horodatage (borné par la durée de validité de son certificat) comparé à 15 ans environ pour une fonction de hachage [46, 47]. Au cours du temps, la construction de nouveaux arbres de hachage est donc négligeable par rapport à l'ajout constant d'horodatages et de métadonnées.

Pour quantifier de manière précise la différence entre la quantité de données utilisées par ERS et l'augmentation de signature LTA, deux possibilités s'offraient à nous :

— soit calculer la quantité de données utilisées pour chacune des méthodes, à travers un programme Python.
  Comme nous le verrons dans le chapitre 4, nous avons simulé l'augmentation de signatures. En y ajoutant une fonction simulant ERS, nous aurions pu quantifier la différence de données utilisées.
  Cependant, tout ce calcul aurait été énormément dépendant de notre implémentation. De fait, utiliser un certain objet ou des structures de données particulières fait varier l'espace utilisé en mémoire et n'est donc ni réaliste ni objectif ;

— soit calculer en théorie ces quantités de données. Malheureusement, ce problème étant récent, nous n'avons pas trouvé dans la littérature de données propices à représenter la quantité de données utilisées par chaque structure (arbre de hachage, signature LTA, etc.).

Un dernier point à spécifier à propos d'ERS concerne l'Evidence Record. Comme déjà expliqué dans la section 3.2.1, l'ER une fois formé peut être conservé et augmenté séparément. Le désavantage est qu'on se retrouve à nouveau dans le cas d'un horodatage pour un unique document et on perd donc l'avantage principal d'ERS.

Ce choix peut être fait par le prestataire lié à la préservation de cette signature si, par exemple, le document a beaucoup de risques d'être à nouveau contesté, il serait préférable de conserver l'ER le concernant et de le préserver pour éviter de récréer toute la structure à nouveau. Ce choix est d'autant plus utile lorsque l'ER est préservé par un autre prestataire, qui n'a donc pas toute la structure ERS du premier. Cet ER est donc considéré comme un document à part entière à conserver.

TrustChain

Enfin, nous avons abordé la TrustChain, une dernière méthode pour préserver la signature électronique à long terme. Deux versions de la TrustChain ont été abordées.

La TrustChain 1.0 stocke comme enregistrement des hashs des documents signés, accompagnés d'autres informations complémentaires.

Cette méthode implique plusieurs institutions, s'opposant ainsi à ERS et à l'augmentation de signature, pour lesquelles un seul prestataire est responsable de la bonne conservation de la signature. Cette particularité est à double tranchant.

D'un côté, cela décentralise la confiance puisque plusieurs institutions sont responsables de valider les signatures. Un utilisateur ne doit donc pas avoir entièrement confiance en un unique prestataire. En outre, si une institution cesse ses activités, les autres seront toujours là pour s'occuper de la chaîne.

D'un autre côté, il faut trouver plusieurs participants prêts à s'occuper de cette blockchain. De plus, pour faire la validation initiale, il faut que les différentes institutions aient accès au document. Ce qui est un gros problème au niveau de la confidentialité de ceux-ci.

De ce problème de confidentialité découle la TrustChain 2.0, prototype tout récent datant de 2020, qui stocke les certificats signants dans des enregistrements. Celle-ci a les mêmes avantages et inconvénients que la version précédente, sauf

qu'elle n'a plus le problème de confidentialité, car elle ne vérifie pas la signature en elle-même, seulement le certificat lié à celle-ci.

Ces deux versions n'ont pas besoin d'ajouter en permanence des métadonnées pour prouver l'intégrité des documents et de leurs signatures. En effet, c'est le principe d'une blockchain, chaque bloc protège le précédent. Si une personne malhonnête désire modifier un bloc, il lui faudra attaquer tous les blocs suivants, ce qui semble impossible puisque la chaîne est publique et que plusieurs versions identiques circulent.

Par contre, la TrustChain n'est pas responsable de la conservation du document et de la signature. Il faut donc en conséquence une autre structure ou un autre système qui, en parallèle, va préserver le document et sa signature.

Dans le cas de la TrustChain 1.0, pour prouver l'intégrité d'un document et de sa signature, il suffira de calculer leur hash (venant du système les ayant conservés) et de le comparer avec celui présent dans l'enregistrement de la TrustChain 1.0. S'ils correspondent, on peut être certain que le document est resté intègre depuis l'ajout de l'enregistrement. En outre, la signature est considérée comme valide puisqu'elle a été vérifiée par l'ensemble des noeuds participants et qu'un vote à la majorité a autorisé son ajout à la blockchain.

Quant à la TrustChain 2.0, la seule vérification opérée consiste à vérifier que le certificat, dont on souhaite connaître le statut, était valide au moment demandé. Un autre processus indépendant est donc nécessaire pour vérifier une signature, à l'aide de ce certificat, considéré comme intègre s'il se trouve dans la TrustChain 2.0.

	Avantages	Inconvénients
Signature basique	- Intégrité - Authenticité	- Valable que quelques années - Besoin de conserver le document dans un autre système
Augmentation de signature	- Intégrité à long terme - Authenticité à long terme - Signature LTA assure la disponibilité des éléments de validation à long terme - Sécurité augmentée si plusieurs horodatages par couche	- Ajout de beaucoup de métadonnées à chaque étape - Augmentation vers la signature horodatée doit se faire rapidement, avant que le certificat signant n'ait expiré - Augmentation vers la signature LTA doit se faire à un moment précis, sinon invalidation - Danger d'invalidité en cas de révocation d'un certificat à un moment inattendu - Nécessité de rester attentif aux évolutions cryptographiques et aux révocations - Besoin de conserver le document dans un autre système - Confiance absolue en un prestataire
ERS	- Intégrité à long terme - Authenticité à long terme - Ajout de moins de métadonnées par rapport à la signature LTA - Sécurité augmentée si plusieurs horodatages par couche - Validation peut être faite hors ligne si l'ER a été extrait - l'ER peut être préservé séparément par la suite, par un autre prestataire - Possibilité de conserver le document et sa signature dans une même structure	- Ajout de métadonnées à chaque étape - Danger d'invalidité en cas de révocation d'un certificat à un moment inattendu - Nécessité de rester attentif aux évolutions cryptographiques et aux révocations - Confiance absolue en un prestataire
TrustChain 1.0	- Intégrité à long terme - Authenticité à long terme - Confiance décentralisée - Pas de répercussion si le certificat de la signature expire - Aucun problème si une institution cesse ses activités, l'ensemble des autres prennent la relève - Pas d'ajout de métadonnées au cours du temps, qui elles-mêmes expirent	- Besoin de plusieurs participants - En cas de données confidentielles, elles sont vues par plusieurs institutions - Besoin de conserver le document et sa signature dans un autre système - Toujours en développement
TrustChain 2.0	- Intégrité à long terme - Authenticité à long terme - Confiance décentralisée - Pas de répercussion si le certificat de la signature expire - Aucun problème si une institution cesse ses activités, l'ensemble des autres prennent la relève - Pas d'ajout de métadonnées au cours du temps, qui elles-mêmes expirent - Pas de problème de confidentialité	- Besoin de plusieurs participants - Besoin de conserver le document et sa signature dans un autre système - Toujours en développement

TABLE 3.1: Résumé de la comparaison des méthodes de préservation

En conclusion, on peut retirer plusieurs éléments de ce bilan.

Tout d'abord, il est toujours indispensable de conserver le document dans un environnement de confiance. Certaines méthodes de préservation de signature peuvent s'en occuper en même temps, tel qu'ERS, mais pour d'autres, il sera nécessaire d'utiliser une autre structure permettant de conserver les documents. Dans le cas de TrustChain 1.0, qui ne conserve que des hashs des informations et TrustChain 2.0, qui conserve des certificats, la signature doit également être conservée.

Ensuite, les propositions liées à la préservation sont multiples et variées, chaque solution a ses avantages et ses inconvénients. Les chercheurs du projet Interpares spécifient également d'autres méthodes utilisées [8], notamment la possibilité pour un utilisateur de resigner continuellement son document (ce qui prend du temps, nécessite une bonne organisation et ne concerne pas la préservation d'une signature, mais la création en chaîne de multiples signatures). Il n'y a donc pas une solution absolue à cette problématique.

De plus, de nouvelles méthodes peuvent toujours émerger. En effet, la méthode de TrustChain est toujours en cours de développement et peut donc encore évoluer. TrustChain 1.0 et 2.0 peuvent aussi fonctionner de pair et de nouvelles spécificités peuvent s'y ajouter [8].

Par contre, aucune des méthodes abordées n'apporte de solution lorsque le certificat a déjà expiré avant de protéger la signature d'une quelconque façon. Une possibilité est de resigner comme expliqué ci-dessus, mais ceci change le contexte de la signature (puisque la date est différente), et demande à nouveau l'intervention du signataire.

Ce sont les méthodes d'ERS et de TrustChain qui ressortent le plus. En effet, l'ajout de métadonnées liées à celles-ci est bien moins significatif qu'avec la signature augmentée. ERS repose sur le même principe que la signature LTA, mais diminue la quantité de données en associant dans une même structure de multiples signatures ; et TrustChain n'augmente pas la quantité de données au cours du temps.

Même si la signature basique ne permet pas, à elle seule, de prouver l'intégrité et l'authenticité d'un document sur le long terme, elle est tout de même utilisée dans ce contexte. De fait, un horodatage et un certificat ne sont autres que des signatures basiques.

Pour toutes les méthodes, la validité de la signature peut être contestée si elle

n'est pas liée à une source de temps fiable, d'où l'importance des horodatages. Prenons l'exemple de TrustChain 2.0. Si la signature n'est pas liée à un temps fiable, le processus de vérification peut penser que le certificat était valide au moment présumé de la création de la signature. Or, si le certificat est expiré, et que quelqu'un de malhonnête a pu voler ses clés, il peut créer une signature antidatée qui sera donc supposée valide, car la fausse date est dans la période de validité du certificat.

Enfin, nous n'avons pas abordé la mise en place fonctionnelle de ces méthodes. Cela peut faire ressortir d'autres avantages et inconvénients. Il est cependant important de notifier que le règlement eIDAS [33] prévoit, dans les exigences reliées à un prestataire de services de confiance qualifiés, que celui-ci, en cas d'arrêt de ses activités, maintienne disponible toutes les données délivrées et reçues qui peuvent être nécessaires comme preuve en justice (art. 24, §2, point h).

Chapitre 4

Simulation des signatures électroniques dans le contexte de l'archivage numérique

Dans les chapitres précédents, nous avons compris le contexte juridique dans lequel s'inscrit la signature électronique, nous avons découvert son fonctionnement se basant sur le chiffrement asymétrique, et plus spécifiquement le chiffrement RSA, et nous avons analysé les méthodes utilisées pour sa validation et sa préservation à long terme.

Ce chapitre a pour but d'approfondir la compréhension des liens entre les différentes structures et les différents acteurs créés pour la préservation à long terme de signatures électroniques. Pour ce faire, nous avons créé une simulation des différentes actions qu'il est possible de réaliser, afin de constater quelles en sont les conséquences.

Afin de laisser la possibilité à toute personne intéressée de découvrir les tenants et aboutissants de cette simulation, elle est disponible sur un répertoire GitHub libre d'accès à l'adresse suivante : `https://forge.uclouvain.be/nonicaise/simulation-de-signatures-electroniques`.

En utilisant une interface, nous pouvons avoir une vue globale et une maîtrise totale sur ce qui est réalisable : créer des prestataires de services de confiance (autorité de certification et d'horodatage), créer des certificats, révoquer des certificats, créer des signatures basiques, les augmenter grâce à l'ajout d'horodatages ou de métadonnées, et vérifier la validité de signatures. Une dernière action réalisable est l'archivage d'un document. Cette action donne toute la confiance au système d'archivage autonome qui va automatiquement préserver les documents signés, pour qu'à n'importe quel instant dans le futur, chaque document puisse être vérifié.

Nous commencerons par découvrir les actions réalisables par les différents intervenants (le système d'archivage, les prestataires de services de confiance ainsi que les clients). Ensuite, nous regarderons plus en détail l'implémentation même de notre simulation. Après cela, nous passerons en revue plusieurs scénarios afin d'illustrer l'utilité de l'interface. Enfin, nous énoncerons les simplifications effectuées dans notre simulation, afin de percevoir les différences majeures avec un système de préservation de signatures électroniques fonctionnant sur l'augmentation de signatures.

4.1 Interface : les différentes actions possibles

La simulation générale permet d'exécuter les différentes actions à son gré, dans l'ordre décidé, afin d'en observer les conséquences. En exécutant la fonction main du fichier `Simulation.py` une fenêtre telle que sur la Figure 4.1 s'ouvre.

Création de prestataires

Tout d'abord, nous pouvons créer deux types de prestataires de services de confiance.

Le premier est une **autorité de certification**. Celle-ci a pour but de créer des certificats, de les associer aux clients demandeurs, mais aussi de les révoquer (pour simuler une fraude par exemple). En encodant le nom de l'autorité qu'on veut former et en appuyant sur le bouton *valider*, nous créons une autorité de certification, contenant sa propre identité, des clés de chiffrement, un certificat racine, et une liste de révocation, au départ vide, qui contiendra les certificats révoqués qu'elle a fournis.

Le second prestataire de service de confiance que nous pouvons créer est une **autorité d'horodatage**. Celle-ci a pour but de créer des horodatages sur n'importe quelle information afin d'en garantir l'intégrité sur un certain laps de temps. En précisant un nom pour ce prestataire et pour l'autorité de certification, nécessaire à la formation de son certificat, une autorité d'horodatage est créée et permet la production d'horodatages qualifiés.

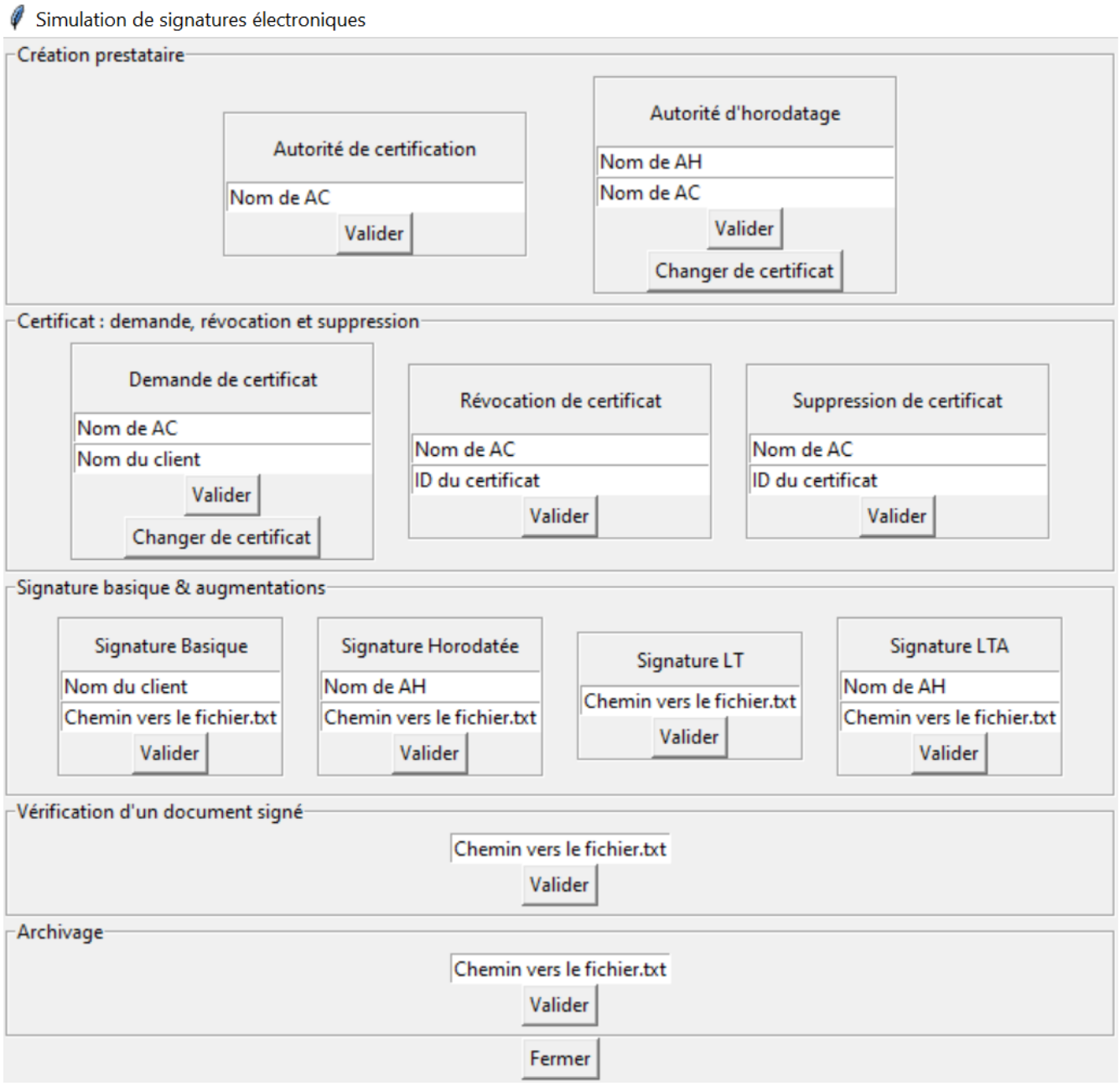


FIGURE 4.1: Interface de la simulation

Certificat : création, révocation et suppression

À partir d'une autorité de certification, il est possible de créer un **certificat** pour un client. En encodant l'identité du client et de l'autorité de certification auquel il désire se rapporter, l'autorité lance le processus de création d'un certificat : après avoir vérifié l'identité du client [1], il génère un certificat et des clés de chiffrement pour ce client. La clé privée est uniquement en la possession du client, tandis que la clé publique est présente dans son certificat.

Après avoir créé un certificat, l'interface répond en fournissant son identifiant unique. Si par la suite on désire **révoquer** ce certificat, il suffit d'encoder son ID et l'autorité de certification auprès de laquelle ce certificat a été créé. De cette manière, le certificat apparaît dans la liste de révocation de l'autorité de certification et ne peut plus être utilisé pour signer.

Il est aussi possible de **supprimer** un certificat, c'est-à-dire de ne plus conserver, au sein de l'autorité de certification, les informations relatives à celui-ci. Notons que ceci est réalisable à n'importe quel instant dans notre simulation, pour pouvoir en constater les conséquences sur les différents types de signatures, mais en réalité, tant que le certificat n'est pas expiré depuis un certain temps, une autorité de certification se doit de conserver ces données.

Signature d'un document

Tout client, dès lors qu'il possède un certificat, peut apposer sa signature sur un document, en spécifiant son nom et le fichier à signer. Ce processus vérifie d'abord que le client demandeur existe, que son certificat n'est ni révoqué ni expiré et que le document n'a pas été signé au préalable [2]. Ensuite, il crée une signature de type **basique** sur le document, grâce au certificat et à la clé privée du client. Cette signature comporte une date, mais qui n'est pas un attribut signé et donc, qui n'est pas protégée.

Augmentation d'une signature

Dès lors qu'une autorité d'horodatage existe, il est possible de créer des **horodatages**. Ceux-ci sont utilisés pour l'augmentation de signature.

1. Notre simulation ne fait qu'inscrire le nom du client et vérifier qu'il ne se trouve pas déjà dans la base de données des clients. En réalité, durant cette étape, il s'agit de vérifier toutes les informations d'identification univoque d'une personne (nom, prénom, adresse, etc.).

2. À nouveau, cela fut décidé pour notre simulation. En réalité, il existe des cas pour lesquels il est nécessaire d'ajouter plusieurs signatures à un document. Par exemple, pour un contrat qui doit être signé par plusieurs parties.

Ainsi, pour créer une **signature horodatée**, il faut spécifier le chemin vers le fichier comportant déjà une signature basique et l'autorité d'horodatage pour qu'elle ajoute un horodatage à la signature basique, si cette dernière est valide. L'ensemble forme une signature horodatée.

Quant à la **signature LT**, il ne faut que lui préciser le chemin vers le fichier. Le processus commence par vérifier que le document comporte une signature horodatée, et vérifie si celle-ci est valide. Si c'est le cas, le processus recherche, chez la ou les autorités de certification, les certificats complets liés à la signature basique et à l'horodatage, ainsi que leurs informations de révocation. Tous ces éléments sont ajoutés à la signature horodatée pour former une signature LT.

Finalement, pour devenir une **signature LTA**, il faut préciser une autorité d'horodatage en plus du chemin vers le fichier, pour la même raison que la signature horodatée. Le processus vérifie que la signature du document est de type LT ou LTA, puis vérifie si cette dernière est valide. Si c'est le cas, il ajoute non seulement le certificat lié au dernier horodatage ajouté, mais aussi les informations de révocation de ce certificat. Enfin, il ajoute un nouvel horodatage sur l'ensemble. Le tout forme une signature LTA.

Il est ensuite possible d'ajouter un nouvel horodatage sur une signature LTA pour former une nouvelle signature LTA. Le nombre d'ajouts, et donc de nouvelles signatures LTA, est indéfini, mais limité par la capacité de stockage.

Vérification d'un document signé

L'action qui nous intéresse le plus est la **vérification** d'un document signé. Celle-ci est la plus complexe et parcourt l'ensemble des données reprises dans la signature pour découvrir et vérifier sa validité. Le processus effectué reprend les validations des différentes classes de signature, comme vu dans le chapitre 3, de manière simplifiée. En conséquence, cette action permet de certifier que le document original et sa signature sont intègres.

Pour un document comportant une signature basique, il s'agit uniquement de vérifier la validité du certificat (qu'il soit intègre, ni révoqué, ni expiré) et puis, de vérifier que la valeur de la signature corresponde bien au document.

Pour un document comportant une signature horodatée, il faut d'abord vérifier l'horodatage : qu'il soit intègre et que son certificat soit valide. Ensuite, il faut vérifier la validité du certificat signant. Dans ce cas-ci, il ne peut être expiré, mais il peut être révoqué, si la date de révocation se situe après l'ajout de l'horodatage.

En effet, ce cas montre la plus-value d'une signature horodatée : la signature reste valide même si le certificat est révoqué.

Pour un document comportant une signature LT, la vérification est identique à la signature horodatée, sauf qu'au lieu d'aller chercher les informations des certificats et de leurs révocations aux autorités de certification correspondantes, ces informations sont directement dans la signature LT.

L'avantage est donc que si les autorités de certification n'ont pas gardé les informations des certificats (des certificats périmés depuis une longue durée par exemple, afin d'en stocker moins, ou pour d'autres raisons), les informations sont tout de même obtenues.

Enfin, pour un document comportant une signature LTA, il faut d'abord vérifier que le dernier horodatage apposé soit intègre, et que son certificat soit valide. Ensuite, il s'agit de traverser toutes les couches d'horodatage en vérifiant qu'ils étaient valides (intègre et certificat valide) au moment de l'ajout de l'horodatage suivant. Pour ceci, il faut utiliser les informations des certificats et de révocation directement présentes dans la signature. En dernier lieu, il faut vérifier que la signature basique et le certificat signant sont valides au moment de l'apposition du premier horodatage.

Archivage

La dernière action possible est d'ajouter un document, signé au préalable (à n'importe quel niveau), au **système d'archivage autonome**. Pour ce faire, il suffit d'entrer le chemin vers le fichier qu'on désire archiver, et d'appuyer sur la touche *valider*.

Le système d'archivage fait en sorte de préserver lui-même le document. Si ce n'est pas déjà fait, il commence par augmenter la signature jusqu'à obtenir une signature LTA. Au fur et à mesure du temps, avant que les certificats des horodatages expirent, il ajouter une nouvelle couche LTA. Les autorités de certification et d'horodatage fournissant le nécessaire à la création de ces nouvelles couches changent quant à elles de certificat régulièrement, avant que leur certificat expire. Dans notre simulation, l'autorité de certification utilise son certificat racine, possédant une durée de validité supérieure aux autres, pour signer d'autres certificats. De plus, une autorité de certification ne peut renouveler son certificat racine.

4.2 Classes : attributs et fonctions

Voyons désormais plus en détail les interactions entre les différentes composantes des actions, et les effets qu'elles impliquent. Sur la Figure 4.2, nous pouvons voir le diagramme de classes, représentant les différents constituants et les liens existants entre ceux-ci.

Tout d'abord, il existe la classe représentant la signature basique. C'est cette classe qui aura le plus d'instances créées. En effet, un objet `SignatureBasique` est créé lors de la signature d'un document par un client, lors de la signature d'un certificat par une autorité de certification et lors de la création d'un horodatage par une autorité d'horodatage. Il contient le type de chiffrement utilisé, les informations à signer, la valeur de la signature, l'identifiant du certificat signant ainsi qu'une information non signée : la date et l'heure de sa création.

Ensuite, toute instance de la classe `Certificat` contient un numéro de série, l'autorité de certification l'ayant créé, une clé publique, de quoi identifier le propriétaire du certificat, sa date et heure de création et de fin de validité, ainsi qu'une signature de son AC permettant de prouver son intégrité.

L'objet `AutoriteCertification` est capable de créer des certificats. Il a une identité et des clés (une privée et une publique) lui permettant de signer des certificats. Sa clé publique est présente dans son propre certificat, qui pour cette simulation, sera toujours un certificat racine, soit signé par sa propre clé privée. L'`AutoriteCertification` peut également révoquer un certificat et possède donc une CRL, qu'elle peut partager avec d'autres utilisateurs. Elle peut aussi *supprimer* un certificat, c'est-à-dire le retirer de ses certificats. C'est dans le cas où un certificat est supprimé que la signature LT devient utile [3].

La classe `Horodatage` permet de lier des informations à une source de temps fiable et de protéger ces deux éléments. Ceux-ci, ajoutés à un numéro de série, forment `TST_info`, qui est la structure de données protégée par une `SignatureBasique`. Tous ces éléments sont les attributs d'un objet `Horodatage`.

Tout `Horodatage` est créé par une `AutoriteHorodatage`. Cette dernière, de la même façon qu'une `AutoriteCertification`, possède une identité, une paire de clés et un certificat. Celui-ci n'est, par contre, pas un certificat racine, mais un certificat créé par une `AutoriteCertification`.

3. En réalité, un prestataire ne supprimera jamais un certificat non expiré, mais ceci est accepté dans notre simulation pour en observer les conséquences.

FIGURE 4.2: Diagramme des classes

Ensuite, il existe les classes représentant les différents types de signatures, contenant chacune une signature de la classe inférieure. La `SignatureHorodatee` contient une `SignatureBasique` en plus d'un horodatage. La `SignatureLT` contient non seulement une `SignatureHorodatee`, mais aussi des données de validations, soit les certificats complets et leurs informations de révocation. La `SignatureLTA` contient soit une `SignatureLT`, soit une `SignatureLTA`, y ajoute les données de validation de l'horodatage précédent (certificat complet et informations de révocation) et protège l'ensemble par un nouvel horodatage.

Enfin, un document est représenté par la classe `Document`, qui contient deux informations : le chemin vers le fichier et une signature, initialement non définie. Il est donc possible de signer *basiquement* un document, ainsi que de l'augmenter avec les différentes fonctions pour transformer sa signature en signature horodatée, LT ou LTA.

4.3 Scénarios

Pour donner des exemples, et pouvoir visualiser plus facilement l'utilité de l'interface, nous donnons ici plusieurs séquences d'actions à réaliser et les résultats qui en découlent.

Paramètres modifiables

Pour adapter la compréhension et la vitesse des répercussions, il est possible de changer la durée de validité des certificats, à la fois ceux des clients et des autorités d'horodatage et de certification. Ce paramètre est modifiable dans le fichier Python `GlobalVariables.py`.

Ces valeurs doivent être données par ordre croissant pour respecter la réalité : un certificat d'une personne physique a une durée de vie plus courte ou égale à celle d'un certificat d'une autorité d'horodatage, toutes deux plus courtes que la durée de validité du certificat racine utilisé pour signer ces certificats. Nous conseillons de mettre une durée de validité à ce dernier largement supérieure aux autres, vu qu'il n'est pas renouvelable dans notre simulation (cf. section 4.1).

4.3.1 Scénario 1 : Étapes vers la création d'une signature LTA

À travers ce scénario, vous pouvez suivre toutes les étapes pour avoir un document signé LTA, et vérifier, à chaque étape, la validité du document.

1. Créer une autorité de certification avec comme nom *MonAC*.

   En réponse, l'interface vous dit que l'autorité a été créée, et le moment auquel son certificat racine expirera.

2. Créer une autorité d'horodatage avec comme nom *MonAH* et comme autorité de certification *MonAC*, qui lui fournit des clés et un certificat.

   En réponse, l'interface vous dit que l'autorité a été créée, le numéro de série ID de son certificat, et le moment auquel ce dernier expirera.

3. Créer un client avec comme nom *Client1*, et comme autorité de certification *MonAC*, qui lui fournit des clés et un certificat.

   En réponse, l'interface vous dit que le client a été créé, le numéro de série ID de son certificat, et le moment auquel ce dernier expirera.

4. Signer un fichier en indiquant le client *Client1* et le chemin vers le fichier à signer. Supposons ici qu'il y ait un fichier *MonFichier.txt*.

   Plusieurs réponses de l'interface sont possibles :
   - en cas de fichier non existant, elle vous le fait savoir ;
   - si jamais le certificat du *Client1* a expiré, elle vous prévient que ce n'est pas possible de le signer. Pour résoudre ce problème, il est possible de changer le certificat du *Client1*, en en redemandant un à n'importe quelle autorité de certification, par exemple *MonAC* ;
   - si tout est en ordre, l'interface vous répond que le document *MonFichier.txt* a été signé.

   À ce stade, il est possible de vérifier le document signé *MonFichier.txt*. Si le certificat du client n'a pas expiré (et qu'il n'a pas été révoqué intentionnellement à l'aide de l'interface), l'interface répond que la signature basique du fichier *MonFichier.txt* est valide. Si au contraire, il a expiré, l'interface vous dit que la signature basique est invalide.

5. Augmenter la signature basique pour devenir une signature horodatée, en précisant *MonFichier.txt* et *MonAH* pour demander un horodatage.

   Si la signature basique a pu être validée et que le certificat de l'autorité d'horodatage est toujours valide, l'interface vous prévient que la signature a été augmentée. Dans le cas contraire, elle vous indique la raison de l'échec d'augmentation ; si l'autorité d'horodatage n'existe pas, si le document n'a pas de signature, si le fichier se trouve déjà dans le système d'archivage, si la signature basique est invalide, si le certificat de l'horodatage a expiré ou encore s'il a été révoqué.

   À ce stade, il est possible de vérifier le document signé *MonFichier.txt*. Si ni le certificat du client [4] ni celui de l'autorité d'horodatage n'ont expiré,

4. Dans ce cas, le certificat peut avoir été révoqué après l'ajout de l'horodatage.

l'interface répond que la signature horodatée du fichier *MonFichier.txt* est valide. Dans le cas contraire, l'interface vous dit que la signature horodatée du fichier est invalide.

6. Augmenter la signature horodatée pour devenir une signature LT, en précisant *MonFichier.txt*.

   Si la signature horodatée a pu être validée, l'interface vous prévient que la signature a été augmentée. À nouveau, dans le cas contraire, elle vous indique la raison de l'échec d'augmentation.

   À ce stade, la vérification du document donne les mêmes résultats qu'à l'étape précédente. En effet, le seul ajout revient aux informations de validation. La différence se remarque donc uniquement si les informations des certificats ne sont plus disponibles auprès de l'autorité de certification, car la signature LT les contient malgré tout dans sa structure.

7. Augmenter la signature LT pour devenir une signature LTA, en précisant *MonFichier.txt* et *MonAH* pour ajouter un horodatage.

   Si la signature LT a pu être validée, l'interface vous prévient que la signature a été augmentée. À nouveau, elle vous indique la raison en cas d'échec d'augmentation.

8. Augmenter à l'infini la signature LTA en précisant *MonFichier.txt* et *MonAH* pour ajouter des horodatages indéfiniment. Avant chaque ajout, la validité de la signature précédente est vérifiée.

9. Vérifier le document en indiquant *MonFichier.txt*. Cette vérification consiste à vérifier la dernière signature LTA et la concordance entre tous les horodatages. Étant donné que l'ajout d'une couche de LTA est refusé en cas de couche précédente invalide, la seule raison qui pourrait invalider cette signature est si le certificat du dernier horodatage a expiré ou a été révoqué.

4.3.2 Scénario 2 : Révocation de certificats

Voyons maintenant ce qu'il se passe lorsqu'on révoque des certificats en reprenant le scénario 1.

Prenons d'abord le certificat signant, soit le certificat pour la création de la signature basique et non ceux nécessaires pour les horodatages.

— Si, avant que le *Client1* signe *MonFichier.txt* (étape 4), le certificat est révoqué, la signature basique ne peut être créée.

— Si le certificat est révoqué après la formation de la signature basique, en essayant de valider le document, l'interface répond que la signature est invalide. Par la suite, si on essaye d'augmenter la signature vers une signature

horodatée, l'interface ne le fait pas, étant donné qu'elle vérifie d'abord si la signature basique est valide.
— Par contre, si le certificat est révoqué après l'ajout d'un horodatage (étape 5), la validation de cette signature est acceptée. En effet, une signature horodatée prouve que la signature a été faite avant la révocation et donc la validation est acceptée.
— Ainsi pour toutes les étapes suivantes, en cas de révocation ultérieure du certificat, il n'y a pas de conséquence sur la validité.

Ensuite, si on révoque les certificats liés aux horodatages, cela invalide la signature seulement si le certificat est révoqué avant l'ajout d'une nouvelle couche d'horodatage.
— Si on révoque le certificat de l'horodatage après avoir créé la signature horodatée, celle-ci ne peut plus être validée. Le processus d'augmentation en signature LT ne s'effectue donc pas.
— Si on révoque le certificat après avoir créé la signature de type LT, cette signature est invalide. En effet, bien que la validation de la signature se fasse grâce aux informations présentes dans la signature elle-même (qui ne contiennent donc pas l'information de révocation du certificat, car créée avant la révocation), une vérification particulière s'ajoute. Il s'agit de vérifier que les informations de révocation sont assez récentes, c'est-à-dire qu'elles n'ont pas changé [5]. Dans ce cas, la signature ne peut donc pas être augmentée.
— Si on révoque le certificat du premier horodatage après avoir créé la signature de type LTA, la validation de cette signature est acceptée si un certificat différent (et donc non révoqué) a été utilisé pour former l'horodatage de cette signature LTA.
— Si on révoque le certificat de l'horodatage de la dernière couche LTA, la signature devient instantanément invalide. Il faut donc bien avoir ajouté une couche LTA avant que l'horodatage précédent ne soit révoqué.

4.3.3 Scénario 3 : Expiration de certificats

Le cas de l'expiration des certificats est presque identique à leur révocation. La seule distinction réfère à la remarque déjà observée dans le chapitre 3 : la différence de validation de signature dans le cas de certificats révoqués ou expirés.

Au final, ceci implique que les signatures sont valides en cas d'expiration dans les mêmes cas que la révocation, à l'exception du cas où le certificat est expiré après

5. Ceci est un raccourci effectué par notre simulation, uniquement pour vérifier les signatures LT. En réalité, cette invalidité se remarque dans l'étape 5 de la validation de la signature basique pour l'horodatage : vérification de la fraîcheur des informations de révocation (cf. section 3.1.1).

l'ajout du premier horodatage (étape 5). La validation de cette signature n'est pas acceptée, car une signature horodatée ne peut protéger contre l'expiration, selon la norme européenne.

4.3.4 Scénario 4 : Utilisation de l'archivage automatique

Il est possible, à partir de l'étape 5 du scénario 1 (après que le document soit signé) et pour toutes les étapes suivantes (après de multiples augmentations de signature), de déposer le document signé sur le système d'archivage : tant que la signature liée au document est valide, c'est accepté.

De cette manière, le système d'archivage ajouter en permanence des couches LTA et met à jour le certificat de son autorité d'horodatage (avant qu'il expire), afin qu'à tout instant dans le futur, il soit possible de demander une validation du document et que celle-ci soit une réussite.

4.4 Différences entre notre simulation et un système de signatures électroniques dans un logiciel d'archivage

Notre simulation a comme but premier de permettre la compréhension de l'augmentation et des différentes validations de signature plus en détail. Par conséquent, il peut être utile de mettre en avant les simplifications effectuées par notre implémentation, pour pouvoir montrer les éléments qu'un véritable système d'archivage devrait posséder dans le cadre de la préservation de signatures électroniques au moyen d'augmentations de signatures.

- Tout d'abord, nous avons décidé de partir de zéro, et donc d'implémenter le chiffrement RSA. Pour ce faire, dans notre simulation, nous prenons comme base les nombres premiers p et q souvent petits, afin d'en déduire des clés. Une amélioration nécessaire serait d'utiliser des clés plus complexes, se basant sur des nombres premiers beaucoup plus grands. Et, par ailleurs, d'en optimiser le calcul.
- La plus grande simplification que nous avons opérée est que les validations ne suivent pas tous les processus comme vu au chapitre 3. Nous avons gardé uniquement les parties les plus importantes : la concordance des temps des horodatages et les vérifications cryptographiques. Cette simplification peut amener à des différences par rapport à la réalité, telle que le raccourci pour vérifier la fraîcheur des informations de révocation pour la signature LT : nous regardons uniquement s'il y a de nouvelles informations de révocation.

Cela permet de détecter que les informations présentes dans la signature LT ne sont pas à jour, et donc que celle-ci n'est pas valide (car le certificat de l'horodatage a été révoqué avant de l'augmenter en une signature LTA).

— Nous considérons toujours les méthodes cryptographiques utilisées comme fiables et nous n'avons donc pas implémenté la possibilité de changer de méthode de chiffrement, de fonction de hachage ou encore de taille des clés.
— Ensuite, il n'est pas inconcevable qu'un document doive être signé plusieurs fois, par plusieurs personnes différentes. Ceci serait donc une fonctionnalité à ajouter.
— De même, dans notre simulation, il n'est pas possible d'horodater plusieurs fois une information (par la même ou deux autorités différentes). Ceci pourrait permettre d'augmenter la sécurité en cas de faillite ou de révocation du certificat d'une autorité d'horodatage.
— De plus, il n'est pas possible d'avoir plusieurs certificats pour un même intervenant, alors qu'en réalité, cela arrive.
— Dans notre simulation, toute autorité de certification comporte un certificat racine. En réalité, il existe peu d'autorités de certification qui ont leur propre certificat racine : une nouvelle autorité de certification peut demander un certificat à une autre AC, elle-même ayant son certificat d'une autre AC, et ainsi de suite jusqu'à une autorité ayant son certificat racine [6]. À une échelle globale, n'avoir que peu d'AC possédant un certificat racine permet d'obtenir une meilleure sécurité. En effet, une confiance *aveugle* ne sera donnée qu'à peu de prestataires, qui pourront donc plus facilement être contrôlés.
— Notre simulation ne permet pas à une autorité de certification de changer son certificat racine : nous avons considéré que ce dernier avait un temps de validité très long par rapport aux autres certificats. Sur le long terme, un certificat est amené à expirer et donc même une autorité de certification racine doit pouvoir changer de certificat.
— Lorsque nous ajoutons les informations complètes par rapport à un certificat, il ne suffit pas d'ajouter les informations du certificat concerné, mais aussi celles de tous les certificats de sa chaîne de certification.
— Un système d'archivage autonome peut bien entendu archiver des documents non signés, et leur ajouter des horodatages pour prouver leur intégrité. Dans notre simulation, nous avons décidé de ne pas le supporter, car nous nous sommes focalisées, depuis le début de ce mémoire, sur la préservation de signatures.
— La vérification d'identité d'un client par une autorité de certification est en réalité beaucoup plus complexe que dans notre simulation : il s'agit de

6. C'est par cette suite de certificats qu'on retrouve une chaîne de certification.

vérifier les informations pour identifier le client de manière unique.
— Une autorité de certification devrait pouvoir retirer des certificats expirés de sa mémoire uniquement lorsqu'ils sont expirés. Cette simulation avait pour but de montrer les conséquences de certaines actions (entre autres l'utilité de la signature LT), c'est pourquoi cela ne correspond pas à la réalité.

Conclusion

À travers ce mémoire, nous avons découvert, étape par étape, les notions en lien avec la signature électronique.

Dans un premier temps, ce mémoire décrit le contexte d'apparition de la signature électronique. La directive européenne de 1999, le règlement européen eIDAS de 2014, transposé dans le Digital Act belge en 2016 lui ont donné un cadre formel. Celui-ci définit des normes pour les différents services électroniques en lien avec la signature, le cachet, l'horodatage, l'envoi de recommandé électronique, l'authentification de site internet, le certificat électronique ainsi que – uniquement pour la Belgique – l'archivage électronique.

Il précise également différents niveaux de qualification – simple, avancé et qualifié – donnant différents niveaux de confiance aux services. Tous les niveaux bénéficient du principe de non-discrimination et le niveau qualifié profite d'une présomption légale de conformité. Ce dernier niveau est attribué à des services électroniques fournis par des prestataires de services de confiance, devenus qualifiés après une démarche auprès de l'organe de contrôle de leur pays.

En outre, ce mémoire éclaircit la problématique de l'archivage et de la signature électronique, mettant en avant le besoin de préservation à long terme, difficile à combler.

Dans un deuxième temps, il explicite les principes technologiques utilisés pour la signature électronique : une fonction de hachage forme une empreinte numérique, un mécanisme de chiffrement rend un message illisible, le chiffrement RSA est un exemple de mécanisme asymétrique et la procédure pour former une signature électronique utilise le chiffrement asymétrique.

Une section parcourt également les différentes formes (enveloppante, enveloppée et détachée) et différents formats (PAdES, CAdES, XAdES et leur version simple) que peut prendre une signature électronique.

L'illustration d'une demande de ressource a permis d'exposer l'utilité des signatures électroniques, des certificats et des prestataires de services de confiance, notamment à travers la création d'un certificat, de signatures, et de la vérification de signatures pour la récupération des informations.

Une autre méthode garantissant l'intégrité et l'authenticité est présentée : le tatouage numérique. Elle a le désavantage d'être plus difficilement applicable aux documents écrits, raison pour laquelle elle n'a pas fait l'objet d'une analyse plus détaillée.

Dans un troisième temps, ce mémoire analyse et évalue différentes méthodes de validation et de préservation à long terme des signatures électroniques.

Il existe différentes classes de signatures (basique, horodatée LT et LTA) associées à différentes sécurités. Chacune d'elle a un processus de validation différent. La première classe n'est valide que lorsque le certificat auquel elle est liée n'est ni révoqué ni expiré, alors que, à titre de comparaison, la dernière permet d'effectuer la validation longtemps après la création de la signature, et donc après de nombreux événements qui limitent sa validité. Par conséquent, cette dernière assure l'intégrité et l'authenticité d'un document, mais elle a pour principal désavantage d'ajouter beaucoup de métadonnées au cours du temps.

ERS, de son côté, permet l'archivage à long terme d'un ensemble de plusieurs documents électroniques et de données signées en mutualisant les données nécessaires à leur préservation. Comparée à la signature LTA, la quantité de métadonnées ajoutées est moins importante. De plus, cette méthode permet non seulement la conservation de la signature, mais aussi celle du document en lien avec cette signature.

Tandis que les deux méthodes précédentes nécessitent une confiance absolue en un prestataire, les deux suivantes décentralisent cette confiance en impliquant plusieurs institutions. La TrustChain 1.0 constitue une blockchain de signatures et la TrustChain 2.0 constitue une blockchain de certificats, ce qui permet de répondre au problème de confidentialité posé par la première version.

Dans un quatrième et dernier temps, une simulation reprend les intervenants et les actions réalisables en rapport avec la préservation de documents signés : les prestataires de services de confiance, les clients, la création, la révocation et la suppression de certificats, la création et l'augmentation de signature, la vérification d'un document signé et l'archivage, c'est-à-dire la préservation automatique de signatures. Plusieurs exemples de scénarios illustrent les résultats qui découlent des différentes actions. Le code de cette simulation est mis à disposition dans un répertoire GitHub via le lien suivant `https://forge.uclouvain.be/nonicaise/simulation-de-signatures-electroniques`.

Une section énumère les différences majeures entre notre simulation et les fonctionnalités réelles qu'un système de préservation de signature électronique doit posséder.

En somme, la Belgique a bien perçu l'importance de conserver aussi bien une signature électronique que le document qui s'y rapporte. C'est pourquoi elle a ajouté à eIDAS un service de confiance concernant l'archivage électronique.

S'il est vrai que l'ERS peut conserver ensemble les signatures et les documents, il n'en reste pas moins que les possibilités de développement, notamment en termes de services de confiance, sont légion. En effet, le nombre de prestataires de services de confiance est faible en Europe, voire inexistant pour certains services en Belgique.

La problématique de l'archivage et les enjeux de la préservation à long terme

de la signature électronique sont des sujets très vastes que nous n'avons donc pas pu couvrir dans leur intégralité. Ainsi, plusieurs pistes sont encore à explorer.

Premièrement, une analyse détaillée concernant la préservation de documents non signés peut s'avérer importante. En effet, à travers ce travail, nous nous sommes centrés sur la préservation des signatures.

Deuxièmement, il peut être utile de s'attarder sur les solutions possibles aux problèmes de l'obsolescence technologique, de l'organisation et du financement au sein d'un système d'archivage.

Troisièmement, il est souhaitable d'exposer des solutions liées au changement de format inévitable pour que les documents restent lisibles. En effet, ce problème indique une incompatibilité entre deux objectifs :

— pour préserver la validité d'une signature, il faut que le document reste inchangé ;
— pour préserver le document au cours du temps, il faut adapter son format pour le garder intelligible, ce qui modifie les bits qui le constituent.

Les archivistes pointent le fait que la modification des bits n'implique pas nécessairement une corruption de l'authenticité [4], mais il faudrait une preuve que le document est resté authentique malgré un changement de format.

Quatrièmement, il peut être intéressant d'étudier les conséquences d'antidater ou postdater des signatures électroniques pour chacune des méthodes de préservation abordées. En effet, la date de signature étant un attribut non signé, il n'est donc pas protégé. Ainsi, il faut, d'une manière ou d'une autre, une preuve d'existence fiable à une certaine date et heure, par exemple à l'aide d'un horodatage.

D'autres problèmes sont à explorer dans un futur pas si lointain : au vu de l'évolution exponentielle de la puissance des ordinateurs et de la potentielle apparition des ordinateurs quantiques, la sécurité des algorithmes cryptographiques risque d'être réduite à néant [20].

Finalement, nous espérons que ce document vous aura été utile, qu'il vous aura permis d'apprendre les subtilités de la signature, qu'il aura permis d'en comprendre l'importance et qu'il incitera à la création de prestataires de services de confiance en Belgique.

Annexe A

Lexique

— **ATS, tampon d'archivage** (Archive Time Stamp) : ensemble formé par un arbre réduit et un horodatage permettant la vérification de l'existence de certains documents à un certain moment.
— **Arbre réduit** : suite de valeurs de hachage permettant de recalculer la racine d'un arbre de hachage à partir d'un document.
— **Archive** : document généré par une personne physique ou morale, et conservé pour garder des traces de certaines activités.
— **Authenticité** d'un document : fait que le document est bien ce qu'il prétend être, il a bien été créé au moment qui est indiqué par la personne qui est censée l'avoir produit.
— **CMS** (Cryptographic Message Syntax) : standard pour des messages protégés cryptographiquement.
— **CRL** (Certificate Revocation Lists) : listes reprenant les informations de révocation de tous les certificats d'une autorité de certification.
— **Certificat électronique** : document numérique identifiant et authentifiant une personne physique ou morale, et permettant de chiffrer des échanges.
— **Certificat racine** : un certificat auto-signé, c'est-à-dire un certificat signé à l'aide de la clé privée liée à la clé publique présente dans ce même certificat.
— **Chaîne de certification** : suite de certificats démarrant du certificat signant, prenant en compte le certificat l'ayant signé et ainsi de suite jusqu'au certificat racine.
— **Chaîne de tampons d'archivage** : suite d'ATS ordonnés chronologiquement.
— **Cryptographie** : art de rendre un message (visible) indéchiffrable.
— **Digital Act** : transposition du règlement eIDAS à la Belgique, ajoutant des dispositions relatives à l'archivage électronique.
— **ERS (Evidence Record Syntax)** : Méthode de préservation à long terme pour laquelle plusieurs de documents signés sont protégés ensemble par des

métadonnées, permettant d'en contenir moins.

— **ETSI** (European Telecommunications Standards Institute) : organisme qui produit des normes pour l'industrie des technologies de l'information et de la communication.
— **Enregistrement** : un ensemble d'informations liées à la préservation d'un document, tel que des signatures, des certificats, des valeurs de hachage, etc.
— **Evidence record (ER)** : ensemble de preuves pour un document au cours du temps. Il contient tous les ATS repris dans la séquence de tampons d'archivage, et d'autres données de vérification.
— **Hash, valeur de hachage** : empreinte numérique, résultat de taille fixe souvent plus petit que le document initial qu'il représente.
— **Horodatage** : signature *basique* qui permet de lier une donnée à un moment précis et ainsi de prouver que cette donnée existait déjà à cet instant.
— **Intégrité** d'un document : absence de modification du document.
— **Listes de confiance** : listes des prestataires et des services de confiance (qualifiés ou non) qu'ils offrent, disponibles en ligne.
— **Loi eIDAS** : Electronic Identification And trust Services, règlement européen qui apporte un cadre juridique pour différents services électroniques.
— **Long Term Validation Data (LTVD)** : données nécessaires pour la validation à long terme d'une signature. Cela comprend les certificats et les informations de révocation (réponses OCSP ou CRL).
— **Niveau qualifié** : niveau de qualité d'un service de confiance lui conférant la présomption légale de conformité.
— **Non-répudiation** : impossibilité, pour l'émetteur, de nier l'émission d'un message.
— **OCSP** (Online Certificate Status Protocol) : protocole donnant l'information de révocation d'un unique certificat. Il permet donc d'éviter de devoir demander une CRL qui contient beaucoup d'informations pour d'autres certificats.
— **Organe de contrôle** : organisme responsable de surveiller les prestataires de service de confiance. Il est responsable d'accorder ou de retirer le statut qualifié aux prestataires.
— **POE** (Proof Of Existence) : preuve d'existence, preuve qu'un objet existait à un moment spécifique.
— **Préservation, Conservation** : garantie de la validité des signatures ou cachets électroniques sur de longues périodes de temps.
— **Présomption légale de conformité** : impossibilité de remettre en question l'intégrité et l'authenticité de services de confiance qualifiés.
— **Principe de non-discrimination** : un document ne peut être refusé au seul motif qu'il est numérique.

— **Rapport de validation** : rapport résumant les différentes étapes et résultats obtenus durant la validation.
— **Séquence de tampons d'archivage** : suite de chaînes d'ATS.
— **SAE, Service d'archivage électronique** : service de confiance qui consiste en la conservation de données électroniques ou la numérisation de documents papiers.
— **Service de confiance électronique** : service électronique qui consiste en la création, la vérification et la validation de signatures électroniques, de cachets électroniques, d'horodatages électroniques, de services d'envoi recommandé électronique et de certificats relatifs à ces services ou à l'authentification de site internet et en leur conservation.
— **Signature électronique** : données sous forme électronique, qui sont jointes ou associées logiquement à d'autres données sous forme électronique et que le signataire utilise pour signer (art. 3 §10 du règlement eIDAS [33]).
— **Signature LT** : signature qui peut être validée lorsque le certificat signant est révoqué ou lorsque les algorithmes utilisés ou que les tailles des clés ne sont plus à la pointe de la technologie.
— **Signature LTA** : signature qui assure la disponibilité et l'intégrité des éléments de validation à long terme et peut aider à valider la signature malgré de nombreux éléments qui limitent sa validité.
— **Signature basique** : signature qui peut être validée tant que le certificat auquel elle est liée n'est ni révoqué ni expiré.
— **Signature horodatée** : signature qui peut être validée lorsque le certificat signant a été révoqué après l'ajout de l'horodatage.
— **TUC, Temps Universel Coordonné** : moment auquel on ajoute l'horodatage.
— **TrustChain** : Méthode de préservation de signatures à long terme basée sur le concept de blockchain.
— **Valeur probante** : caractère convaincant d'une preuve aux yeux d'un juge.
— **Validation** : processus de vérification et de confirmation de la validité d'une signature ou d'un cachet électronique (art. 3 §41 du règlement eIDAS [33]).
— **Versement** : transmission par un client d'un document numérique sur un système d'archivage électronique.

Bibliographie

[1] Agence nationale de la sécurité des systèmes d'information (ANSSI). Référentiel Général de Sécurité : Annexe A2 - Politique de Certification Type : "certificats électroniques de personne". https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_A2.pdf, 2014. (visité le 26/05/2021).

[2] Jean-Luc Archimbaud. Les principes techniques des certificats électroniques. *Les Cahiers du numérique*, 4(3) :101–110, 2003. mise en ligne sur Cairn.info le 01/09/2010 https://www.cairn.info/revue-les-cahiers-du-numerique-2003-3-page-101.htm# (visité le 24/02/2021).

[3] Archivage-numérique.fr. Qu'est-ce que le SAE ? https://www.archivage-numerique.fr/sae. (visité le 25/02/2021).

[4] Jean-François Blanchette. La conservation de la signature électronique : perspectives archivistiques. *Paris : Direction des Archives de France*, 2004. http://www.interpares.org/display_file.cfm?doc=ip2_dissemination_rep_blanchette_2004.pdf (visité le 12/05/2021).

[5] Aljoša Jerman Blažič and Svetlana Saljic. XML Evidence Record Syntax. https://slideplayer.com/slide/8611857/, 2010. (visité le 10/03/2021).

[6] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and Dave Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008. https://doi.org/10.17487/RFC5280 (visité le 29/04/20201).

[7] Vladimir Bralić, Magdalena Kuleš, and Hrvoje Stančić. A model for long-term preservation of digital signature validity : Trustchain. *INFuture2017 Proceedings : The Future of Information Sciences/Atanassova, I., Zaghouani, W., Kragić, B., Aas, K., Stančić, H., Seljan, S.(eds.). Zagreb : Department of Information and Communication Sciences, Faculty of Humanities and Social Sciences, University of Zagreb, Croatia*, pages 89–103, 2017. https://www.researchgate.net/profile/Hrvoje-Stancic/publication/321171227_A_Model_for_Long-term_Preservation_of_Digital_Signature_Validity_TrustChain/links/5a133437aca27217b5a11031/A-Model-for-Long-term

-Preservation-of-Digital-Signature-Validity-TrustChain.pdf (visité le 10/06/2021).

[8] Vladimir Bralić, Hrvoje Stančić, and Mats Stengård. A blockchain approach to digital archiving : digital signature certification chain preservation. *Records Management Journal*, 30(3) :345–362, 2020. https://doi.org/10.1108/RMJ-08-2019-0043 (visité le 10/06/2021).

[9] Marc Chaumont. Le tatouage de documents numériques. http://www.lirmm.fr/~chaumont/Tatouage.html. (visité le 15/11/2021).

[10] E. Combet and J. Bordier. Document pédagogique à l'attention des utilisateurs et porteurs d'applications de signature électronique. Technical report, ClubPSCo, 2012. https://clubpsco.fr/wp-content/uploads/2012/07/GT-1-GT-Interop-Document-pedagogique-signature-electronique-20120627.pdf (visité le 11/03/2021).

[11] Commission européenne. Trusted List Browser. https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home, 2021. (visité le 31/03/2021).

[12] Conseil International des Archives. Que sont les archives ? https://www.ica.org/fr/que-sont-les-archives, 2016. (visité le 25/02/2021).

[13] Rédaction de Futura. Pourquoi la signature électronique va remplacer la signature physique. https://www.futura-sciences.com/tech/questions-reponses/securite-signature-electronique-va-remplacer-signature-physique-14386/, 2021. (visité le 17/03/2021).

[14] DeftPDF. Different kinds of watermarks, 2019. https://fr.slideshare.net/JamesLin216/different-kinds-of-watermarks (visité le 14/02/2021).

[15] Philippe Delahaye. L'archivage à valeur probante au service de la signature électronique. https://www.docusign.fr/blog/archivage-valeur-probante-signature-electronique#:~:text=Validit%C3%A9%20des%20certificats%20des%20signature,%2D%C3%A9chu%20et%20non%20r%C3%A9voqu%C3%A9), 2018. (visité le 26/05/2021).

[16] DG CONNECT. Questions & Answers on Trust Services under eIDAS. https://ec.europa.eu/digital-single-market/en/news/questions-answers-trust-services-under-eidas, 2017. (visité le 01/03/2021).

[17] European Telecommunications Standards Institute. ETSI TS 119 172-1 v1.1.1 (: Electronic signatures and infrastructures (esi) ; signature policies ; part 1 : Building blocks and table of contents for human readable signature policy documents. https://www.etsi.org/deliver/etsi_ts/119100_119199/11917201/01.01.01_60/ts_11917201v010101p.pdf, 2015. (visité le 09/04/2021).

[18] European Telecommunications Standards Institute. ETSI EN 319 102-1 V1.1.1 : Electronic Signatures and Infrastructures (ESI) ; Procedures for Creation and Validation of AdES Digital Signatures ; Part 1 : Creation and Validation . https://www.etsi.org/deliver/etsi_en/319100_319199/31910201/01.01.01_60/en_31910201v010101p.pdf, 2016. (visité le 05/04/2021).

[19] European Telecommunications Standards Institute. ETSI TS 119 102-1 V1.2.1 : Electronic Signatures and Infrastructures (ESI) ; Procedures for Creation and Validation of AdES Digital Signatures ; Part 1 : Creation and Validation . https://www.etsi.org/deliver/etsi_ts/119100_119199/11910201/01.02.01_60/ts_11910201v010201p.pdf, 2018. (visité le 20/05/2021).

[20] Louis Fortier-Dubois. De la sécurité calculatoire des protocoles cryptographiques devant la menace quantique [Mémoire de maîtrise, Université Laval, Québec, Canada]. 2019. http://hdl.handle.net/20.500.11794/34907 (visité le 11/06/2021).

[21] GeeksforGeeks. RSA Algorithm in Cryptography. https://www.geeksforgeeks.org/rsa-algorithm-cryptography/. (visité le 28/02/2021).

[22] Tobias Gondrom. Evidence Record Syntax—a new International Standard for Long-Term Archiving of Electronic Documents and Signed Data. In *ISSE/SECURE 2007 Securing Electronic Business Processes*, pages 367–375. Springer, 2007. https://doi.org/10.1007/978-3-8348-9418-2_39 (visité le 13/04/2021).

[23] Tobias Gondrom, Ralf Brandner, and Ulrich Pordesch. Evidence Record Syntax (ERS). RFC 4998, August 2007. https://doi.org/10.17487/RFC4998 (visité le 10/03/2021).

[24] S. C. Gupta and Manju Sanghi. Matrix modification of RSA digital signature scheme. *Journal of Applied Security Research*, 16(1) :63–70, 2021. https://doi.org/10.1080/19361610.2019.1690350 (visité le 27/05/2021).

[25] Russ Housley. Cryptographic Message Syntax (CMS). RFC 5652, September 2009. https:/doi.org/10.17487/RFC5652 (visité le 24/04/2021).

[26] Arnaud Hulstaert. Préservation à long terme de l'information numérique. *smalsresearch.be*, 2010. http://www.smalsresearch.be/download/research_reports/deliverable/digital_preservation.pdf (visité le 03/04/2021).

[27] Thomas Kopp. Discussion au sujet de la signature électronique. (le 26/11/2020).

[28] Locarchives. Qu'est-ce qu'un système d'archivage électronique ? https://locarchives.fr/faq/quest-ce-quun-systeme-darchivage-electronique/. (visité le 25/02/2021).

[29] LuxTrust. Comprendre les aspects juridiques : signature électronique vs cachet électronique, 2019. https://www.luxtrust.com/fr/comprendre-les-aspects-juridiques-signature-electronique-vs-cachet-electronique/ (visité le 15/03/2021).

[30] Antoine Meissonnier and Hervé Streiff. Le règlement eIDAS : un cadre européen de l'archivage électronique ? *Gazette des archives*, 245(1) :47–53, 2017. https://doi.org/10.3406/gazar.2017.5512 (visité le 27/05/2021).

[31] Sang Bae Park. Security requirements for multimedia archives. *Advances in Multimedia*, 2015, 2015. https://doi.org/10.1155/2015/956416 (visité le 27/05/2021).

[32] Parlement européen et Conseil de l'Union européenne. Directive 1999/93/CE du Parlement européen et du Conseil, du 13 décembre 1999, sur un cadre communautaire pour les signatures électroniques. *Journal officiel n° L 013 du 19/01/2000 p. 0012 - 0020*, 1999. https://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:31999L0093 (visité le 22/02/2021).

[33] Parlement européen et Conseil de l'Union européenne. Règlement (UE) n°910/2014 du Parlement européen et du Conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE. *Journal officiel L 257 du 28.8.2014, p. 73-114*, 2014. https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=celex%3A32014R0910 (visité le 23/02/2021).

[34] Bart Preneel. Analysis and design of cryptographic hash functions. https://www.esat.kuleuven.be/cosic/publications/thesis-2.pdf, 2003. (visité le 07/05/2021).

[35] Professeur émérite Jacques Mersch à l'UNamur. Souvenirs de la guerre 40-45.

[36] Jean-Marc Rietsch. La validation/conservation des signatures électroniques. *livre blanc*, 2021. https://www.archimag.com/livres-blancs/validation-conservation-signatures-electroniques (visité le 23/04/2021).

[37] Guillaume Robez. Quelle sont les différents types de signature électronique ? https://independant.io/signature-electronique-simple-avancee-qualifiee/#cas, 2021. (visité le 01/03/2021).

[38] SPF Economie, P.M.E., Classes moyennes et Energie. Digital Act - Resumé. https://economie.fgov.be/sites/default/files/Files/Online/One-pager-Digital-Act-FR.pdf, 2018. (visité le 26/03/2021).

[39] SPF Economie, P.M.E., Classes moyennes et Energie. FAQ sur les services de confiance. https://economie.fgov.be/sites/default/files/Files/Online/FAQ-services-de-confiance.pdf, 2018. (visité le 15/03/2021).

[40] SPF Economie, P.M.E., Classes moyennes et Energie. Loi mettant en oeuvre et complétant le règlement (UE) n°910/2014 du parlement européen et du conseil du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE. http://www.ejustice.just.fgov.be/cgi_loi/change_lg.pl?language=fr&la=F&cn=2016072140&table_name=loi, 2020. (visité le 23/02/2021).

[41] SPF Economie, P.M.E., Classes moyennes et Energie. Signature électronique et autres services de confiance. https://economie.fgov.be/fr/themes/line/commerce-electronique/signature-electronique-et, 2020. (visité le 25/02/2021).

[42] Hrvoje Stančić. Long-term preservation of digital signatures. *Technical and field related problems of traditional and electronic archiving/Gostenčnik, Nina (ur.). Maribor : Pokrajinski arhiv*, pages 481–491, 2016. https://www.researchgate.net/publication/301364818_Long-term_Preservation_of_Digital_Signatures (visité le 20/05/2021).

[43] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full sha-1. In *Annual International Cryptology Conference*, pages 570–596. Springer, 2017. https://doi.org/10.1007/978-3-319-63688-7_19 (visité le 09/06/2021).

[44] Nicolas Thirion. Le Code de droit économique : principales innovations. https://www.larcier.com/fr/le-code-de-droit-economique-principales-innovations-2015-9782804480592.html, 2015. (visité le 01/03/2021).

[45] Universign. 13 points pour décrypter le règlement européen eIDAS sur la signature électronique - livre blanc règlement européen eIDAS. 2020. https://www.universign.com/fr/livres-blancs/le-reglement-eidas/ (visité le 20/04/2021).

[46] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In *Annual international cryptology conference*, pages 17–36. Springer, 2005. https://doi.org/10.1007/11535218_2 (visité le 09/06/2021).

[47] Tao Xie, Fanbao Liu, and Dengguo Feng. Fast collision attack on md5. *IACR Cryptol. ePrint Arch.*, 2013 :170, 2013. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.301.4421 (ivisté le 09/06/2021).

[48] Muneer Bani Yassein, Shadi Aljawarneh, Ethar Qawasmeh, Wail Mardini, and Yaser Khamayseh. Comprehensive study of symmetric key and asymmetric key encryption algorithms. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–7, 2017. https://doi.org/10.1109/ICEngTechnol.2017.8308215 (visité le 30/03/2021).

[49] Robert Zuccherato, Patrick Cain, Dr. Carlisle Adams, and Denis Pinkas. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, August 2001. https:/doi.org/10.17487/RFC3161 (visité le 24/04/2021).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**