



La cryptographie basée sur l'identité

Au service de la confidentialité des données

Tuteur : Julien Francq

Telecom ParisTech – cycle ingénieur - troisième année

Année universitaire 2013/2014

Responsable pédagogique : Jean Leneutre



Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 1

Ce document et son contenu sont la propriété de CASSIDIAN et ne peuvent être reproduits et/ou communiqués sans autorisation. Toute utilisation autre que l'objet de sa communication est interdite. This document and its content are property of CASSIDIAN and must not be duplicated and/or disclosed without authorization. Any use other than that for which it was intended is prohibited.

Notice de Copyright

Ce document et son contenu sont la propriété de Cassidian Cybersecurity et ne doit pas être copié ni diffusé sans autorisation. Toute utilisation en dehors de l'objet expressément prévu est interdite. Les parties non publiques de ce document sont remplacées par la mention : « *Partie non publique* ».

Il est strictement interdit de reproduire, distribuer et utiliser le contenu de ce document sans l'autorisation préalable de l'auteur. Les contrefacteurs seront jugés responsables pour le paiement des dommages. Tous droits réservés y compris pour les brevets, modèles d'utilité, dessins et modèles enregistrés.

Copyright © [2013] – CASSIDIAN CYBERSECURITY - Tous droits réservés.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 2

Ce document et son contenu sont la propriété de CASSIDIAN et ne peuvent être reproduits et/ou communiqués sans autorisation. Toute utilisation autre que l'objet de sa communication est interdite. This document and its content are property of CASSIDIAN and must not be duplicated and/or disclosed without authorization. Any use other than that for which it was intended is prohibited.

Remerciements

Je remercie la société Cassidian CyberSecurity pour m'avoir accueilli durant ces six mois.

Je remercie plus particulièrement Julien Francq, mon maître de stage, pour m'avoir fait confiance dans la réalisation de ce projet et m'avoir accueilli chaleureusement comme stagiaire au sein de son service.

Je tiens aussi à adresser mes remerciements à Nicolas pour son aide déterminante durant mon stage et plus généralement toutes les personnes que j'ai eu à côtoyer pendant ces six mois pour leurs conseils, leur aide et leur bonne humeur.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 3

Table des matieres

Introduction générale.....	6
----------------------------	---

Partie 1 : Etat de l'Art

1. La cryptographie basée sur l'identité	9
1.1. Introduction	9
1.2. Principe de fonctionnement et comparaison.....	11
1.1.1. Chiffrement symétrique.....	11
1.1.2. Chiffrement asymétrique.....	12
1.1.3. Le chiffrement basé sur l'identité	15
1.1.4. Avantages et inconvénients de l'IBC.....	17
2. Fonctionnement technique de l'IBC.....	19
2.1. Première approche.....	19
2.3. Couplages et courbes elliptiques.....	23
2.4. Choisir une courbe elliptique « IBC-friendly »	25
2.5. Trouver une courbe « IBC-friendly ».....	27
3. Les applications de l'IBC	29
3.1. Signature numérique	29
3.1.1. Schémas de signature courte	29
3.1.2. Signature basée sur l'identité.....	30
3.2. Chiffrement basé sur l'identité (IBE)	32
3.3. Protocole d'accord sur une clé	33
3.4. Identity-Based Broadcast Encryption (IBBE)	35
3.5. IBC et sécurité du <i>Cloud-computing</i>	38

Partie 2 : Notre solution basée sur l'IBC

4. Introduction	41
5. Motivations et problématiques.....	42
6. Formalisation des besoins	44

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 4

6.1. Des besoins multiples	44
6.2. Quelques exemples.....	46
6.2.1. <i>Dans le secteur de la santé</i>	46
6.2.2. <i>Autour des relations privées/publiques</i>	47
7. Réponses aux besoins.....	50
7.1. Réponses apportées par les technologies existantes	50
7.2. Notre solution à base d'IBE	53
8. Implémentation de notre solution basée sur l'IBE	56
8.1. Vue d'ensemble	56
8.2. IBEncrypt	59
8.2.1. <i>Rôles et missions</i>	59
8.2.2. <i>Liens avec les autres modules</i>	59
8.2.3. <i>Diagramme fonctionnel</i>	60
8.2.4. <i>Implémentation</i>	60
8.3. Le PKG	62
8.3.1. <i>Rôles et missions</i>	62
8.3.2. <i>Liens avec les autres modules</i>	62
8.3.3. <i>Diagramme fonctionnel</i>	63
8.3.4. <i>Implémentation</i>	63
8.4. L'Add-In Outlook	65
8.4.1. <i>Rôles et missions</i>	65
8.4.2. <i>Liens avec les autres modules</i>	65
8.4.3. <i>Implémentation</i>	65
9. Interface graphique et résultats	68
9.1. Sur un scénario	68
9.2. Récapitulatif	72
9.3. Perspectives et futures améliorations	73
Conclusion générale	74
Bibliographie	74

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 5

Introduction générale

La cryptographie basée sur l'identité (ou IBC pour « *Identity-Based Cryptography* ») est un domaine récent de la cryptographie moderne. Imaginée en 1984 par Shamir, célèbre mathématicien et cryptographe à l'origine de l'algorithme RSA lors d'un exercice de pensée, l'IBC ne restera pour l'essentiel qu'une fiction jusqu'en 2001, lorsque deux chercheurs de l'université de Stanford proposèrent un algorithme de chiffrement basée sur l'identité (*Identity-Based Encryption from the Weil Pairing* Franklin, D. Boneh (1)). Depuis, ce domaine est en pleine effervescence dans le monde académique : de très nombreuses équipes de recherche à travers le monde ont proposé successivement depuis le début des années 2000 nombre d'algorithmes basés sur « l'identité » de plus en plus sophistiqués et sécurisés (chiffrement, signature, broadcast, etc.). Ainsi, la première partie de ce rapport aura pour objectif premier de faire un état de l'art du domaine pour donner aux lecteurs les éléments de compréhensions, les enjeux et les contraintes essentielles de ce domaine prometteur.

Aujourd'hui, les processus cryptographiques traditionnels (la cryptographie symétrique et la cryptographie asymétrique) ne nécessitent plus autant qu'auparavant d'avoir une connaissance précise des éléments techniques fondamentaux sur lesquels ils reposent pour être utilisés en pratique au stade industriel et applicatif. En effet, la couche applicative est suffisamment dense et les algorithmes suffisamment matures pour pouvoir faire quasi abstraction des connaissances plus fondamentales. A rebours, la cryptographie basée sur l'identité repose sur des concepts et des objets mathématiques beaucoup plus jeunes, tout du moins concernant leur utilisation en cryptographie. C'est le cas des courbes elliptiques, et c'est encore plus le cas des « couplages » que nous introduirons dans la première partie du rapport (cf. page 19). Ces objets, encore peu connus et utilisés en comparaison avec les « robustes » corps finis utilisés par l'algorithme RSA, posent encore de nombreuses interrogations vis-à-vis de la sécurité.

Si l'IBC semble être un domaine prometteur, en témoigne l'importante activité à son sujet encore aujourd'hui, elle reste pour le moment à un niveau essentiellement académique et de recherche appliquée. Malgré quelques expérimentations à travers le monde (Voltage Security, HP Labs, Microsoft Recherche) très peu d'applications utilisant l'IBC ont pour le moment vu le jour. **La véritable motivation du travail effectué pendant six mois sur ce sujet était donc de franchir le cap entre le stade de recherche et le stade industriel en développant une solution innovante reposant sur l'IBC.**

Ainsi, dans la seconde partie du rapport, qui constitue **la véritable valeur ajoutée du travail effectué** pendant six mois, nous présenterons notre solution basée sur l'IBC, laquelle permet, entre autres, de sécuriser les échanges d'information dans les entreprises et organisations de tout type de manière plus simple, efficace et économique en comparaison avec les solutions actuelles du marché. A titre d'illustration, nous montrerons que notre solution pourrait permettre de garantir la confidentialité des données personnelles de santé, tout en conservant un haut niveau de praticité, dans un contexte où celles-ci sont de plus en plus amenées à être numérisées, transférées et partagées entre les différents praticiens et professionnels de santé dans ce qu'il est commun d'appeler aujourd'hui le « *Cloud Santé* ».

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 6

Avant-propos

Ce rapport est structuré en deux parties. La première consiste en une synthèse du domaine de la cryptographie basée sur l'identité : principe général, fonctionnement technique, sécurité des algorithmes, etc. Cependant, cette première partie ne doit pas être vue uniquement comme un état de l'art. En effet, les enjeux de sécurité sont primordiaux pour un système cryptographique quel qu'il soit. Avant de nous lancer dans le développement d'une solution, il était indispensable de savoir quel niveau de sécurité nous pouvions atteindre et quelles seraient les contraintes sur notre chemin. Nous verrons que les conclusions de cette première partie permettent justement de faire les bons choix techniques pour développer un produit utilisable en pratique et à haut niveau de sécurité.

La deuxième partie consiste en une description de la solution que nous avons développée pendant six mois. Cette description n'est pas purement technique. Un produit répond avant tout à une problématique et à des besoins. Une nouvelle technologie n'est vraiment utile que si elle permet de lever des contraintes actuelles. C'est la raison pour laquelle la partie « technique » correspondant au développement de la solution ne commence qu'au milieu de cette seconde partie.

Deux remarques préliminaires par rapport à ce qui vient d'être dit :

- Les deux parties peuvent être lues de manière indépendante. Le lecteur davantage intéressé par l'aspect applicatif pourra tout à fait commencer par la seconde partie, et revenir à la première après coup pour en comprendre le fonctionnement technique « bas niveau ». Dans ce cas, nous conseillons néanmoins aux lecteurs de lire préalablement les sections introductives 1.1 et 1.2 (pages 9 à 15) qui posent les éléments de compréhensions minimaux sur le fonctionnement de la cryptographie basée sur l'identité.
- Ce rapport pourra paraître comme relativement abstrait à première vue. Le lecteur ne doit pas oublier que la principale contribution de notre travail pendant six mois consistait à développer d'une solution complète de chiffrement : implémentation des algorithmes, d'un service web de génération des clés, et d'une *Add-In* pour Outlook 2010/2013. Ainsi, autant pour des raisons pratiques que pour des raisons de confidentialité, la granularité et la précision vis-à-vis du code source n'ont pas été possibles.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 7

Partie 1 : Etat de l'art

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 8

1. La cryptographie basée sur l'identité

1.1. Introduction

Les différences entre les processus cryptographiques reposent essentiellement sur la gestion des clés de chiffrement et de déchiffrement : comment construire, se mettre en accord, authentifier, distribuer et utiliser les clés ? Nous rappellerons plus en détails dans la section suivante (cf. page 11) en quoi diffèrent les deux processus cryptographiques « historiques », la cryptographie symétrique et la cryptographie asymétrique (ou cryptographie à clé publique).

Contentons-nous pour le moment de rappeler l'un des défauts majeurs de la cryptographie asymétrique, largement utilisée aujourd'hui à travers le monde pour sécuriser les communications électroniques. La cryptographie asymétrique rend obligatoire l'utilisation d'infrastructures à clé publique (ou PKI pour « *Public Key infrastructure* ») permettant l'échange de clés entre les utilisateurs. En effet, quand deux individus ne partageant pas de secret veulent communiquer de manière sécurisée, ils doivent générer des paires de clés et soumettre leur clé publique à une autorité de certification (ou CA pour « *Certificate Authority* » en anglais) afin que cette dernière garantisse leur identité. Ce n'est qu'une fois ce certificat reçu que les deux protagonistes pourront échanger des messages chiffrés sans craindre d'être espionnés¹. Bien que ce processus continue à montrer son efficacité (la sécurité des paiements sur Internet repose sur la cryptographie asymétrique) nous verrons que celui-ci se révèle néanmoins couteux financièrement et difficile à administrer pour la plupart des entreprises : gestion des bases de données, génération des certificats, révocation, etc. Par ailleurs, de tels processus asymétriques à base de PKI peuvent être déroutants pour des non connaisseurs, du fait de leur relative complexité (nécessité de publier préalablement sa clé publique, d'obtenir un certificat, etc.) ce qui n'aide pas à leur adoption massive dans les PME/PMI et dans certaines administrations de l'Etat. Même pour des utilisateurs plus expérimentés, l'implémentation des processus de cryptographie asymétrique reste complexe dans leur mise en œuvre à grande échelle, en particulier dans les grandes entreprises (mauvaise résistance à la montée en charge, évolutivité limitée, gestion des certificats laborieuse, etc.). Alors que la sécurité des données numériques n'a jamais été aussi importante du fait des menaces d'espionnage à grande échelle et des attaques ciblées, il serait heureux de pouvoir disposer aujourd'hui d'une alternative.

En 1984, Adi Shamir, célèbre mathématicien et cryptographe à l'origine de l'algorithme RSA, introduisait le concept de cryptographie basée sur l'identité (ou IBC pour « *Identity-Based Cryptography* ») (2). Comme ne le verrons, la principale innovation de l'IBC est de pouvoir se servir de n'importe quelle information sur le destinataire comme clé publique. Il peut s'agir d'une adresse

¹ Sans certification, un individu malveillant pourrait se faire passer pour le destinataire auprès de l'expéditeur en lui donnant sa propre clé publique. C'est ce qu'on appelle l'attaque de « l'homme au milieu ».

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 9

IP, d'un numéro de téléphone ou encore d'une adresse email. Comme nous le verrons, cette particularité permet de réduire drastiquement la complexité du processus cryptographique en éliminant, en particulier, la nécessité de générer et distribuer les certificats des utilisateurs. Nous verrons dans la suite que cette caractéristique a de nombreuses conséquences positives permettant de rendre le processus cryptographique beaucoup plus fonctionnel.

Dans les années 1980, après la publication de Shamir, une première application utilisant l'algorithme de chiffrement RSA a pu être implémenté, la signature basée sur l'identité (cf. page 29). Si d'autres schémas cryptographiques comme le chiffrement basé sur l'identité (ou IBE pour « *Identity-Based Encryption* ») ont également été imaginés à l'époque, ils n'ont par contre pas pu être réalisés en pratique, faute de théorie mathématique permettant de les implémenter (cf. pages 15 et 32): « *At this stage we have concrete implementation proposals only for identity-based signature schemes, but we conjecture that identity-based cryptosystems exist as well and we encourage the reader to look for such systems.* » (2).

L'implémentation pratique de l'IBE et d'autres applications de la cryptographie basée sur l'identité sont restés des problèmes ouverts jusqu'en 2001, lorsque deux équipes de recherche distinctes ont trouvé simultanément deux solutions différentes (Boneh and Franklin (3), et Cocks (4)) dont la première, qui sera développée dans ce rapport, repose sur le concept mathématique de « *couplage* » (cf. page 19). Nous reviendrons plus largement sur les applications de l'IBC dans la section 3 du rapport (cf. page 29).

La cryptographie basée sur l'identité (IBC) est un type de processus cryptographique asymétrique dans lequel une chaîne de caractères arbitraire est utilisée comme clé publique. Historiquement cette chaîne de caractère représentait *l'identité* du destinataire (adresse email, numéro de téléphone). Toutefois, cette chaîne de caractère peut être choisie totalement librement. Nous en donnerons une illustration dans la seconde partie de ce rapport (cf. page 53).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 10

1.2. Principe de fonctionnement et comparaison

Dans cette section, nous allons étudier le principe de fonctionnement de la cryptographie basée sur l'identité. Pour cela, nous considérerons l'opération de chiffrement comme exemple pratique afin de présenter les différences avec les processus traditionnels, à savoir les systèmes cryptographiques symétriques et asymétriques. Bien évidemment, le chiffrement n'est qu'une primitive cryptographique parmi d'autres, et nous étudierons les applications de l'IBC de manière plus exhaustive dans la section 3 du rapport (cf. page 29). Par ailleurs, les considérations mathématiques sur l'IBC ne seront examinées que dans la section suivante. En particulier nous n'introduirons pas ici la notion de « *couplage* » et nous nous contenterons de présenter le fonctionnement de l'IBE à haut niveau.

1.1.1. Chiffrement symétrique

Le premier processus de chiffrement, datant des années 1970, est un processus symétrique. Dans ce cas, la même clé est utilisée pour chiffrer et déchiffrer l'information. Le gestionnaire des clés (ou KM pour « *Key Manager* » en anglais) génère une nouvelle clé pour chaque message que l'expéditeur veut envoyer. Un canal privé doit permettre à l'expéditeur et au destinataire de partager cette clé en toute sécurité avant d'envoyer le message chiffré sur le canal public. Pour cela, Alice (l'expéditrice) peut faire une requête auprès du gestionnaire de clés en spécifiant le destinataire du message qu'elle veut envoyer. Le KM, à partir d'une « *Master Key* » lui renvoie alors une clé et garde en mémoire que cette clé va être utilisée par Alice pour envoyer un message à Bob, le destinataire.

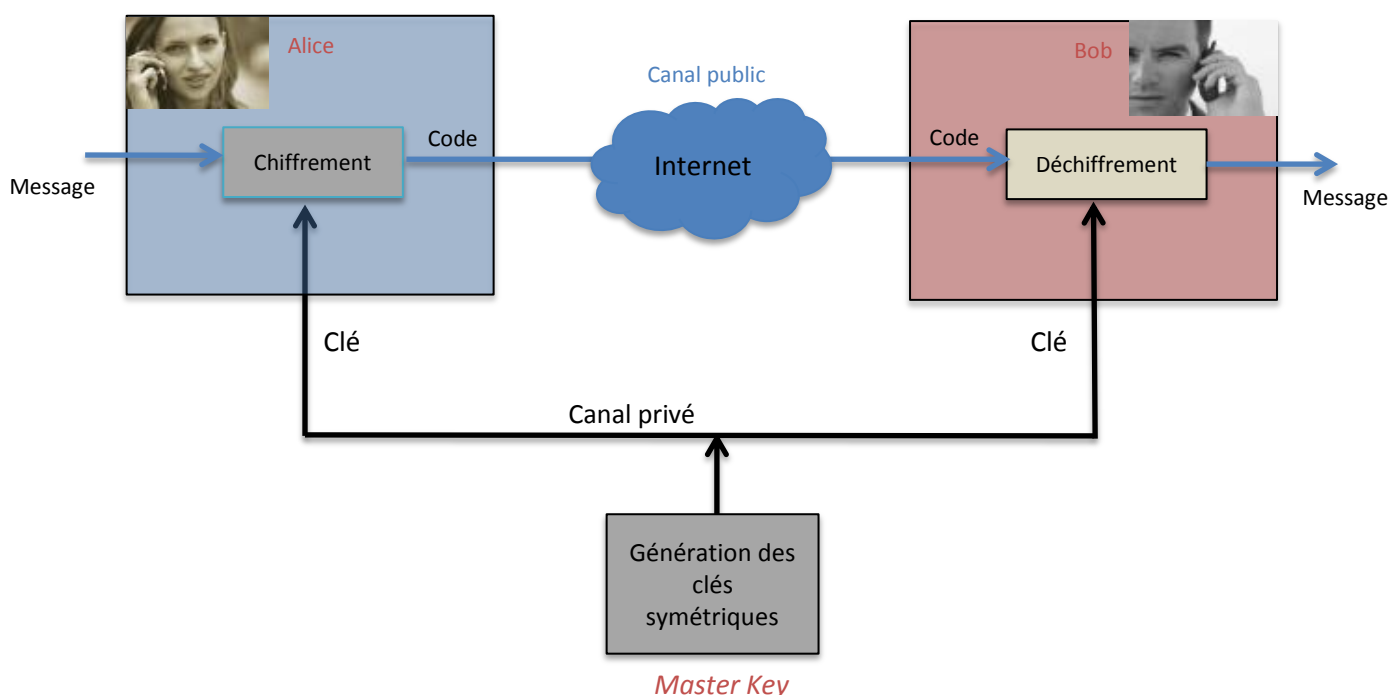


Figure 1 : Les systèmes cryptographiques à clés publiques utilisent la même clé pour chiffrer et déchiffrer

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 11

Concernant l'implémentation pratique de ce processus, il faut disposer d'une base de données des destinataires autorisés. Bob, pour obtenir cette même clé afin de déchiffrer le code secret qu'Alice va lui envoyer, doit s'authentifier auprès du KM. La clé est alors récupérée à partir de la base de données et le nom du destinataire est « recoupé » avec la liste des destinataires autorisés. Si tout est valide, la clé est envoyée à Bob.

L'avantage des systèmes cryptographiques symétriques est qu'ils sont extrêmement rapides. Les performances des implémentations réelles permettent d'atteindre des débits très élevés. Ainsi, la cryptographie symétrique est très efficace en termes de rapidité pour traiter d'importantes masses d'information. Mais les systèmes cryptographiques symétriques présentent aussi de nombreux défauts. L'un des plus dommageables est de nécessiter une clé pour chaque message qu'envoie Alice à Bob. Cela peut nécessiter rapidement d'importantes capacités de stockage et de gérer des bases de données complexes. La gestion des clés est donc difficile avec un tel système. Par ailleurs, le partage d'une clé commune entre les protagonistes n'est pas évident, et dans bien des cas il est difficile de pré-partager une telle clé. C'est pour cette raison que la cryptographie symétrique est si peu adaptée aux paiements sur Internet par exemple. De même lorsqu'on souhaite envoyer un email à quelqu'un que l'on ne connaît pas, on partage rarement une clé avec lui. Par ailleurs, il n'est pas évident de se mettre d'accord sur une clé commune, tout simplement parce qu'on ne dispose pas forcément d'un canal privé.

1.1.2. Chiffrement asymétrique

Au début des années 1980, une série d'innovations et de découvertes en mathématiques a permis la construction de nouveaux algorithmes cryptographiques. Ces algorithmes, qualifiés d'algorithmes asymétriques (ou algorithmes à clé publiques) permettent de construire des processus utilisant des clés différentes pour le chiffrement et le déchiffrement des données. L'échange de clés Diffie-Hellman et l'algorithme RSA sont à l'origine de bien des implémentations de la cryptographie asymétrique. La clé de chiffrement est dite publique tandis que la clé de déchiffrement est dite privée. Le destinataire (Bob) génère en premier lieu une paire de clés (publique, privée). Pour chiffrer les données, l'expéditeur utilise la clé publique du destinataire et envoie ensuite les données à travers un canal public. Seul le destinataire en question pourra déchiffrer le texte chiffré grâce à sa clé privée.

Un tel système repose sur une infrastructure à clé publique (ou PKI de l'anglais « *Public Key Infrastructure* ») qui a pour rôle de certifier les clés publiques des utilisateurs. L'idée est de pouvoir assurer à l'expéditeur que telle clé publique correspond bien à tel destinataire. L'enjeu d'une telle certification est de la plus haute importance : sans elle, Alice et Bob risquent de subir une attaque de l'homme au milieu (ou « *Man in the Middle Attack* » en anglais). Un attaquant pourrait en effet facilement se faire passer pour Bob en substituant sa clé publique à la sienne et pouvoir ainsi déchiffrer le message à l'aide de sa clé privée.

Autrement dit, la cryptographie asymétrique introduit le concept de tiers de confiance. Les deux destinataires d'une communication ne se font pas forcément confiance entre eux (et ne pré-partagent pas forcément de clé) et vont donc faire confiance à une même troisième personne (en anglais noté TA pour « *Trusted Authority* ») qui garantira à chacun l'identité de l'autre.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 12

Les processus de cryptographie asymétrique présentent ainsi plusieurs avantages sur leurs homologues symétriques :

- Ils ne nécessitent pas de canal privé.
- Il n'y a plus besoin de clé de chiffrement unique pour chaque message envoyé. On utilise toujours la même clé publique pour envoyer des messages chiffrés à un destinataire donné.
- Ainsi, il n'y a plus besoin de contacter le serveur à chaque nouveau message et par ailleurs, le nombre de clés à générer et (éventuellement) à stocker est plus faible.

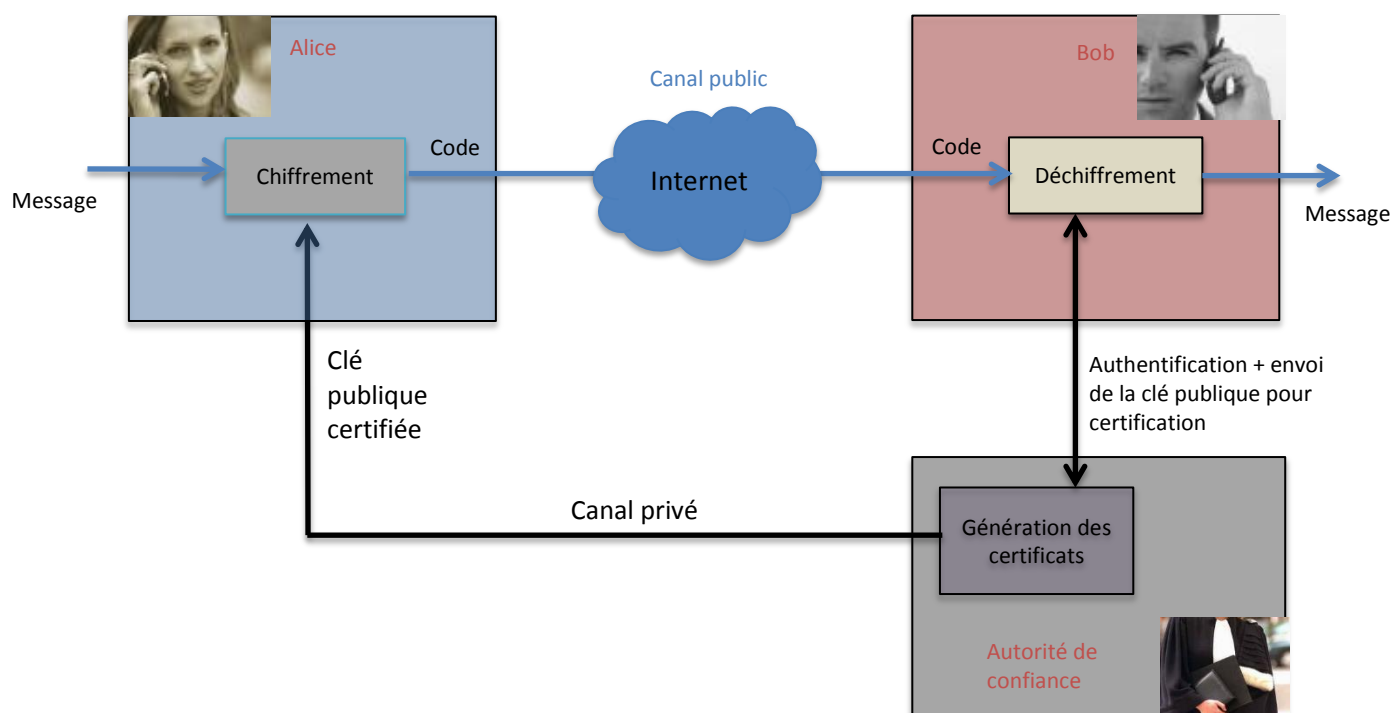


Figure 1 : Les systèmes cryptographiques à base de PKI utilisent des couples de clé (privée, publique).

Néanmoins, le chiffrement par système asymétrique présente d'autres désavantages de poids :

- Pour envoyer un message chiffré, il faut que le destinataire ait publié sa clé publique et que celle-ci ait été certifiée par la TA.
- Le système de PKI nécessite une gestion complexe des certificats : publication, mise à jour, révocation, etc. Cela entraîne une complexité supplémentaire non négligeable pour l'implémentation.
- Les clés sont beaucoup plus longues qu'en cryptographie symétrique (de 128 à 256 bits pour l'algorithme symétrique de référence AES contre 1024 bits minimum pour l'algorithme à clé publique RSA).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 13

On voit donc que la cryptographie symétrique et la cryptographie asymétrique présentent leurs propres avantages et inconvénients. Jusqu'à très récemment on ne disposait donc pas en pratique de processus permettant de cumuler tous les avantages (longueur de clé, souplesse et simplicité de la gestion des clés, facilité d'utilisation pour l'expéditeur et le destinataire, etc.). Il fallait donc faire un choix entre les deux solutions précédentes et en accepter les aspects négatifs.











	Cryptographie symétrique	Cryptographie asymétrique
Performances (débit)		
Longueur des clés		
Volume des clés (clés réutilisables)		
Ne nécessite pas de clé pré-partagée		
Ne nécessite pas de certificats (et de gestion des bases de données associées)		

Tableau 1 : pas de solution miracle, cryptographie symétrique et asymétrique ont chacune leurs points forts et leurs points faibles...

Par ailleurs, certains inconvénients sont communs aux deux procédés. Par exemple, dans les deux cas, l'expéditeur ne peut pas envoyer un message à un destinataire qu'il ne connaît pas « à la volée » c'est-à-dire sans aucune « opération » préalable. Dans un cas (symétrique) il doit en effet pré-partager un secret avec son destinataire (ou trouver un canal privé pour ce faire), dans l'autre (asymétrique) il doit d'abord récupérer auprès d'une autorité de certification (CA) la clé publique certifiée de ce dernier...

Maintenant que nous avons rappelé le fonctionnement des deux processus de chiffrement traditionnels, ainsi que les avantages et inconvénients qui leur sont associés, nous pouvons nous pencher sur le processus de chiffrement basé sur l'identité (ou IBE).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 14

1.1.3. Le chiffrement basé sur l'identité

Tout comme le chiffrement asymétrique traditionnel, le chiffrement basé sur l'identité repose également sur l'existence d'une tierce partie de confiance. Ici, cette troisième personne est nommée communément le *générateur de clés privées* (ou PKG de l'anglais « *Private Key Generator* »). En toute rigueur, la cryptographie basée sur l'identité n'est qu'un sous ensemble de la cryptographie asymétrique, elle en partage donc avec elle certaines caractéristiques. Toute la spécificité de l'IBC par rapport aux systèmes à PKI classiques vient de la façon dont sont gérées et utilisées les clés. Détaillons pas à pas le processus de chiffrement basé sur l'identité (ou IBE pour « *Identity-Based Encryption* ») :

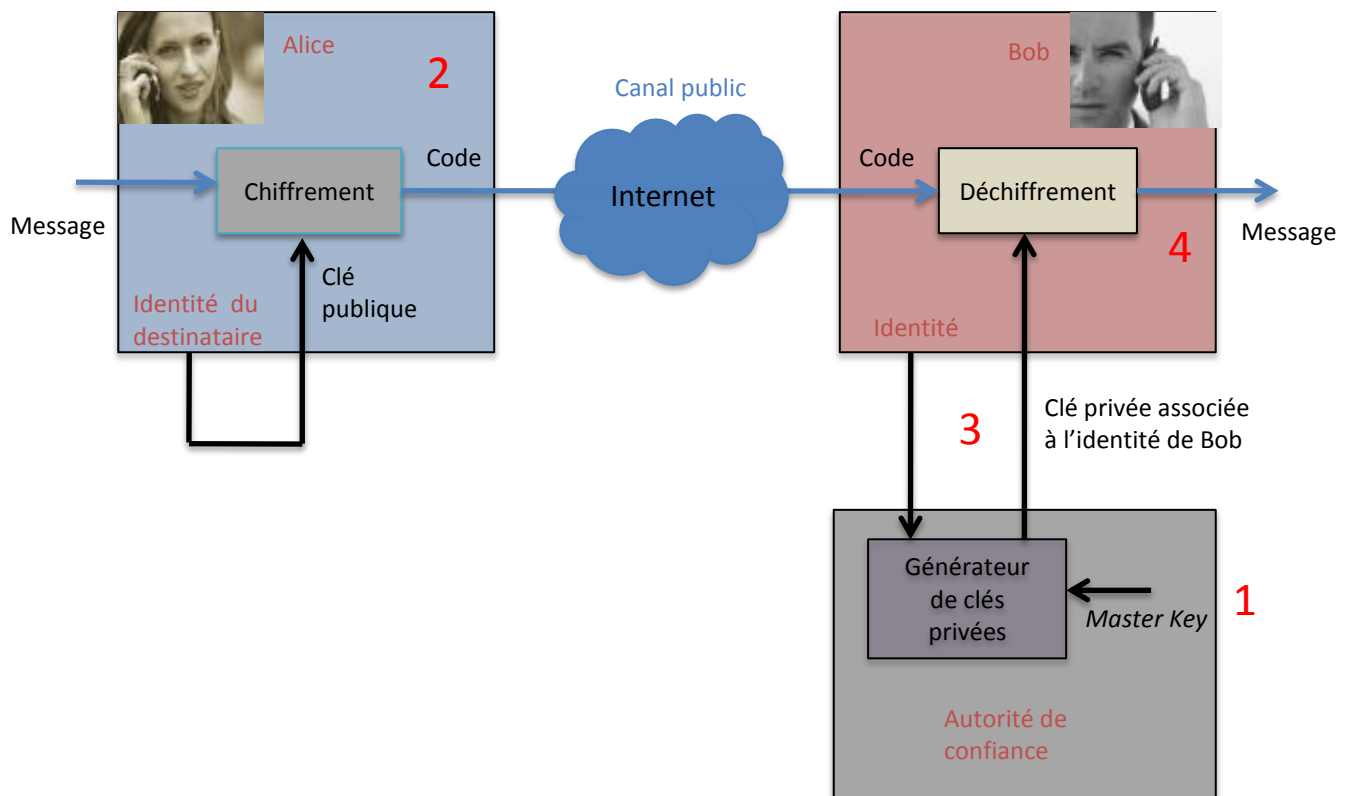


Figure 2 : Avec le chiffrement basé sur l'identité, Alice peut choisir une clé publique pour Bob. Ce dernier n'a pas besoin d'avoir une clé de déchiffrement avant qu'Alice ne lui envoie un message chiffré

1. Le PKG doit générer une paire de clé privée/publique (noté pk_{PKG} et sk_{PKG} dans la suite). La clé pk_{PKG} est rendue publique pour tous les utilisateurs du service. Ces clés sont communément appelées « *Master Public Key* » (ou paramètres du système) et « *Master Private Key* », respectivement.
2. L'expéditeur (Alice) prépare un message M pour Bob. Elle va alors utiliser l'identité de Bob, ID_{Bob} et la pk_{PKG} afin de chiffrer M et d'obtenir alors le texte chiffré C. Alice envoie ensuite le texte chiffré C à Bob à travers le canal public. Il faut en particulier noter ici que

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 15

ID_{Bob} et pk_{PKG} sont toutes les deux connues d'Alice. En particulier, pk_{PKG} peut soit être transmis par le PKG à l'utilisateur, soit être directement stocker en local lors de l'installation du logiciel de chiffrement sur les postes utilisateurs.

3. Bob reçoit le texte chiffré C. Bob va alors s'authentifier auprès du PKG², lequel lui renverra le cas échéant sa clé privée $sk_{ID_{Bob}}$ à travers un canal sécurisé. Si ID_{Bob} est une adresse email, le PKG peut envoyer par exemple un nonce sur son email afin de vérifier que la personne correspondant à ID_{Bob} correspond bien à la personne l'ayant contacté. A titre d'exemple ce nonce peut correspondre à un lien SSL hypertexte permettant à Bob de télécharger sa clé privée sur un serveur sécurisé.
4. Finalement, Bob déchiffre C en utilisant sa clé privée $sk_{ID_{Bob}}$ et obtient M.

Remarque : Une variation du protocole précédent est la suivante. Au lieu de déchiffrer lui-même le code secret, Bob peut laisser le PKG le faire à sa place. Cette solution peut permettre de renforcer l'aspect « *user-friendly* » du chiffrement basé sur l'identité mais pose par contre un problème de sécurité du fait que le message en clair devra alors transiter sur le réseau.

Nous avons ici donné une première idée du fonctionnement d'un protocole basé sur l'identité. Ce n'est évidemment qu'une approche préliminaire permettant de mieux appréhender les différences essentielles avec les systèmes cryptographiques traditionnels. En particulier, nous n'avons pas donné ici d'algorithme précis du déroulement du protocole, cela fera l'objet de développements ultérieurs (cf. page 32). Dans la section 3 du rapport (cf. page 29) nous reviendrons en profondeur sur le protocole précédent et sur d'autres, comme la signature basée sur l'identité par exemple. Avant cela, il nous faudra tout d'abord nous intéresser aux techniques sous-jacentes sur lesquelles reposent l'IBC, et en particulier aux objets mathématiques garantissant un haut niveau de sécurité, ce qui est justement l'objet de la section suivante.

Pour le moment, contentons-nous de conclure ce chapitre en nous intéressant de plus près aux avantages et inconvénients intrinsèques de la cryptographie basée sur l'identité, à la lumière du protocole précédent.

² L'authentification peut se faire de manière « classique » pas différents moyens : login et mot de passe, mot de passe à usage unique, carte à puce, etc.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 16

1.1.4. Avantages et inconvénients de l'IBC

Nous avons déjà explicité certains des **avantages de l'IBC** précédemment, voici néanmoins une liste plus complète :

- Peu de préparation est nécessaire pour qu'un destinataire puisse recevoir des messages chiffrés.
- L'expéditeur n'a pas besoin de rechercher en ligne la clé publique du destinataire, il lui suffit de connaître son « identité ».
- Il n'y a plus de certificat.

Nous allons maintenant voir que du fait du protocole lui-même, l'IBC présente un inconvénient important en termes de sécurité. Le désavantage le plus notable de l'IBC est lié à la sécurité du protocole. De manière intrinsèque, la gestion des clés dans l'IBC a pour conséquence l'existence d'une *autorité de séquestre* (ou « **Key Escrow** » en anglais). En effet, avec la cryptographie basée sur l'identité, le PKG peut déchiffrer ou signer les messages de n'importe quel utilisateur sans être détecté. Pire, le PKG peut rendre publiques des clés privées sans que cela soit vu ou remarqué. A rebours, pour attaquer un utilisateur avec les systèmes cryptographiques à PKI traditionnels, le CA doit générer activement un faux certificat contenant une fausse clé publique. Par ailleurs il est impossible pour le CA d'effectuer une attaque passive : il ne peut pas déchiffrer un message pour un utilisateur en utilisant sa clé publique... Ainsi, la confiance donnée au PKG est bien plus forte que celle donnée au CA, et un PKG malveillant sera extrêmement critique pour la sécurité du système. C'est donc évidemment un élément à prendre en compte.

Par ailleurs, il faut noter que la possibilité pour le PKG de pouvoir signer à la place d'un utilisateur supprime la propriété de non répudiation des systèmes à PKI traditionnels. Néanmoins, cette spécificité peut aussi devenir un point fort, en particulier dans des contextes où les utilisateurs peuvent avoir entièrement confiance dans le PKG : entreprises, *start-up*, administrations publiques. On peut également envisager l'utilisation de l'IBC entre deux entités dont l'une peut faire suffisamment confiance à l'autre pour lui permettre d'administrer le PKG : EADS et des organisations gouvernementales, les hôpitaux et le ministère de la Santé, etc.

La place et le pouvoir du PKG présentent également d'autres avantages dans certains contextes :

- L'autorité de séquestre peut permettre d'améliorer la facilité d'utilisation et la convivialité du protocole en opérant lui-même les opérations cryptographiques pour l'utilisateur, voir même en ne requérant aucune installation du côté du client. Cette fonctionnalité peut être particulièrement utile pour des entreprises souhaitant mettre en place un système automatique de chiffrement des emails à partir de l'analyse de leur contenu.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 17

- Si le destinataire n'a pas besoin de recevoir sa clé privée, celle-ci peut être conservée uniquement au niveau du PKG, lequel dispose d'un niveau de sécurité souvent bien plus élevé qu'une simple machine utilisateur.
- Grace à l'autorité de séquestre, une entreprise peut retrouver des données chiffrées d'un utilisateur ayant perdu sa clé privée, puis qu'il lui suffit de régénérer la clé privée à partir de son identité et de la « *Master Key* ».

Avant de refermer ce premier chapitre définitivement notons que tout n'est pas si simple avec l'IBC : en particulier, pour que le protocole de chiffrement basé sur l'identité (IBE) soit réellement sécurisé, remarquons que l'authentification entre le PKG et l'utilisateur est à prendre en considération. En effet, si la clé privée du destinataire du message transite en clair entre eux deux, une attaque « *Man In The Middle* » pourra permettre à un attaquant de l'obtenir. On peut aussi imaginer chiffrer cette information, mais alors on retrouve les inconvénients et contraintes des protocoles cryptographiques traditionnels dont nous avons longuement parlé précédemment. Nous soulèverons une nouvelle fois et en détails ces questions dans le chapitre 4 du présent rapport.
















	Cryptographie symétrique	Cryptographie asymétrique	Cryptographie basée sur l'identité
Performances (débit)			
Longueur des clés			 (cf. chapitre suivant)
Volume des clés (clés réutilisables)			
Ne nécessite pas de clé pré-partagée			
Ne nécessite pas de certificats (et de gestion des bases de données associées)			

Tableau 2 : Les avantages de l'IBC par rapport aux processus cryptographiques traditionnels

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 18

2. Fonctionnement technique de l'IBC

Dans ce chapitre, nous allons nous intéresser au fonctionnement technique et à la sécurité de la cryptographie basée sur l'identité (IBC). Dans un premier temps, nous « brosserons » de manière générale les grands principes mathématiques sous-jacents. Ce sera en particulier l'occasion d'introduire le concept de « **pairing** » ou **couplage** en français. Ensuite, nous nous concentrerons sur la théorie des courbes elliptiques, lesquelles sont des objets mathématiques au cœur des algorithmes utilisés par l'IBC. Nous verrons en particulier que toutes les courbes elliptiques ne font pas de bonnes candidates pour faire de l'IBC, et que toutes ne permettent pas de garantir une sécurité satisfaisante. Nous définirons en particulier une catégorie de courbes que nous qualifierons d'« *IBC-friendly* ». L'objectif final de ce chapitre, outre évidemment la compréhension générale des notions mathématiques et des algorithmes permettant de réaliser en pratique de l'IBC, est d'être capable de choisir avec discernement une « bonne » courbe elliptique pour réaliser de l'IBC de manière efficace et sécurisée.

Dans la suite, E désigne une courbe elliptique définie sur un corps fini K et \mathbb{F}_q désigne un corps fini où $q = p^m$ avec p premier. On dira alors que p est la caractéristique de \mathbb{F}_q .

2.1. Première approche

La vaste majorité des schémas cryptographiques basés sur l'identité, et plus particulièrement ceux établis depuis les articles de 2001 (Boneh and Franklin (3), et Cocks (4)), sont basés sur des objets mathématiques appelés **applications bilinéaires non dégénérées** (ou « *Bilinear nondegenerate maps* » en anglais). Une application bilinéaire non dégénérée est une application jumelant des éléments d'un groupe cyclique de départ (qu'on notera G dans la suite) vers un groupe cyclique d'arrivée de même ordre q premier, où le problème du logarithme discret est difficile dans le groupe de départ. On rappelle ci-dessous le problème du logarithme discret :

Le problème du logarithme discret (DLP) : Soit G un groupe cyclique d'ordre q que nous noterons multiplicativement. Soit α un générateur du groupe G . Ainsi, tout élément x du groupe s'écrit d'une unique façon sous la forme :

$$x = \alpha^k \text{ avec } 0 \leq k \leq q - 1.$$

L'exposant k est le logarithme discret de l'élément x de G . Le problème du logarithme discret (DLP) est de trouver k lorsqu'on se donne x . Pour certains groupes, le problème du logarithme discret est difficile. Cette propriété est largement utilisée en cryptographie.

La sécurité de l'IBC est basée sur le choix d'applications bilinéaires non dégénérées bien particulières où il est facile de calculer le résultat dans le groupe cyclique étant donné les deux opérandes de départ, mais où il est très difficile de calculer l'inverse. Autrement dit, on cherche une

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 19

application bilinéaire non dégénérée à sens unique. Cette propriété est équivalente à la « *Bilinear Diffie-Hellman Assumption* » sur laquelle nous allons revenir un peu plus loin.

Couplage (ou « couplage ») : Soit G_1 , G_2 et G_t trois groupes cycliques de même ordre q . Une application bilinéaire de $G_1 \times G_2$ vers G_t est une fonction $e : G_1 \times G_2 \rightarrow G_t$ telle que pour tous $u \in G_1, v \in G_2$ et $(a, b) \in \mathbb{Z}^2$:

$$e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$$

On peut également utiliser la notation additive pour G_1 et G_2 :

$$e(aU, bV) = e(bU, aV) = e(U, V)^{ab}$$

Les deux notations sont utilisées dans la littérature.

Dans notre cas, on aura toujours $G_1 = G_2 = G$ mais nous donnons ici la définition générale d'une application bilinéaire. Une application bilinéaire triviale est l'application jumelant n'importe quelle paire d'élément du groupe de départ G_1 vers l'identité du groupe d'arrivée G_t . On qualifie une telle application bilinéaire de *dégénérée*. Puisque ce cas ne nous intéresse évidemment pas, une application bilinéaire admissible devra être non dégénérée, comme nous le précisons plus haut. Par ailleurs, on considérera toujours des groupes G et G_t d'ordre q où q un nombre premier de « grande taille ». Enfin, pour pouvoir être utilisé en pratique, un *couplage* e doit être calculable. Avant d'aller plus loin, rappelons quelques problèmes cryptographiques classiques.

« **Decisional Diffie-Hellman assumption** » (DDH): Soit $u \in G$ et $(a, b, c) \in \mathbb{Z}^3$. Peut-on distinguer les deux distributions de probabilité (u^a, u^b, u^{ab}) et (u^a, u^b, u^c) calculatoirement parlant ?

Dans certains espaces, DDH est un problème difficile. Par contre, dans un groupe cyclique G , si on dispose d'une application bilinéaire non dégénérée à sens unique, DDH n'est plus un problème difficile. En effet, il suffit de vérifier que $e(u^a, u^b) = e(u, u^c)$ ce qui est faisable dans un temps court puisque nous avons supposé notre application bilinéaire non dégénérée.

« **Computational Diffie-Hellman assumption** » (CDH): Calculer u^{ab} à partir de (u^a, u^b) . CDH reste un problème difficile dans G .

« **Bilinear Decisional Diffie-Hellman assumption** » (BDDH): Distinguer $(u, u^a, u^b, u^c, e(u, v)^{abc})$ de $(u, u^a, u^b, u^c, e(u, v)^z)$. BDDH est un problème présumé difficile dans G .

Un groupe où CDH reste difficile mais où DDH est facile est qualifié de groupe **GDH** (pour « *Gap Diffie-Hellman* »).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 20

La question est maintenant de savoir 1) quels groupes utiliser en pratique pour la cryptographie (et en particulier pour l'IBC) et 2) comment définir concrètement un *couplage* dans ce groupe ?

- Le groupe cyclique G est typiquement un **sous-groupe de points du groupe abélien associé à une courbe elliptique E** . Sans rentrer dans les détails, ce que nous ferons au paragraphe suivant, une courbe elliptique est un cas particulier de courbe algébrique, munie entre autres de la propriété d'addition géométrique sur ses points. Soit E une courbe elliptique et P, Q deux points de cette courbe. On définit leur somme $P+Q$ comme le symétrique du point d'intersection entre la courbe de E et la droite passant par P et Q (cf. Figure 3). On peut démontrer que cette opération « $+$ » est bien une loi de composition. Ici on utilise la notation additive, alors que précédemment nous avons surtout utilisé la notation multiplicative pour G . Les deux notations existent dans la littérature, ce qui ne doit pas faire peur au lecteur. Ce qu'il faut simplement retenir c'est que nous disposons d'une loi de composition sur les courbes elliptiques laquelle nous permet de l'utiliser en tant que groupe cyclique. Comme nous l'avons expliqué précédemment, sur une courbe elliptique E , calculer aU (ou u^a en utilisant la notation multiplicative) à partir de a et u est facile alors que l'opération inverse (trouver a étant donné u et u^a) est difficile.
- Le groupe cyclique G_t est un **sous-groupe d'un groupe multiplicatif d'un corps fini**.

En pratique, E sera une courbe elliptique définie sur le corps fini $K = \mathbb{F}_q$ tandis que $G \subseteq E(\mathbb{F}_q)$ sera un groupe cyclique d'ordre r et $G_t = \mathbb{F}_{q^k}$.

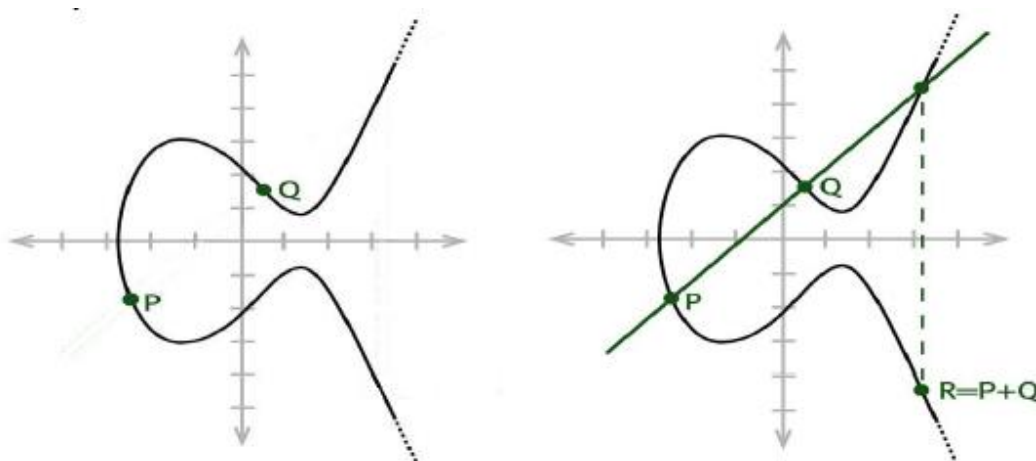


Figure 3 : opération sur les points d'une courbe elliptique

Concernant le *couplage* à choisir, les trois plus connus sont le couplage de Weil, le couplage de Tate et le couplage de Ate, qui sont des objets mathématiques complexes, qu'il n'est néanmoins pas indispensable de comprendre pour les utiliser. Nous ne rentrerons pas dans les détails pour le moment. Pour conclure, voyons comment les courbes elliptiques et le *couplage* peuvent permettre de construire un algorithme d'échange de clés entre deux individus. Cet algorithme pourra être très

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 21

facilement modifié pour chiffrer un message, l'objectif ici est de voir globalement comment l'IBC met à profit la théorie du *couplage* sur les courbes elliptiques.

Application du *couplage* à un protocole d'échange de clés basé sur l'identité :

Il nous faut tout d'abord une fonction de hachage qui permette de dériver un point P_{ID} de G (qui, nous le rappelons, est un sous-groupe de points du groupe abélien associé à une courbe elliptique E) à partir d'une identité ID (c'est-à-dire une chaîne de caractère). Le PKG (cf. chapitre 1) choisit aléatoirement un secret s et publie un triplé de points (P, Q, Q) comme paramètre du système (c'est la *Master Public Key*). On a par ailleurs la relation $Q = sP$. Quand un utilisateur dont l'identité correspond au point P_{ID} demande sa clé privée au PKG, il reçoit le point $P_{ID} = s \times P_{ID}$. Par rapport au chapitre précédent, P_{ID} correspond à la clé privée sk_{IDBob} du destinataire.

Après cette phase d'initialisation et d'extraction de clé privée, n'importe quel utilisateur va pouvoir échanger une clé avec son destinataire ID de la façon suivante :

1. Il choisit une valeur aléatoire r .
2. Il envoie au destinataire ID le point rQ .
3. Il calcule la valeur $e(P_{ID}, rQ)$.
4. De son côté, le destinataire ID calcule $e(P_{ID}, rQ)$.

Il est facile de vérifier que du fait de la propriété du *couplage* e , les deux valeurs calculées par chacun des deux protagonistes sont égales. En effet :

$$e(P_{ID}, rQ) = e(P_{ID}, rsP) = e(P_{ID}, P)^{rs} = e(sP_{ID}, rQ) = e(P_{ID}, rQ)$$

Notons que personne d'autre ne peut calculer la clé échangée car seul le destinataire connaît P_{ID} (qui correspond à sa clé privée).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 22

2.2. Couplages et courbes elliptiques

Dans ce paragraphe, nous allons revenir sur l'historique de l'application des *couplages* à la cryptographie. Nous verrons en particulier comment les *couplages* interviennent dans la sécurité de la cryptographie basée sur les courbes elliptiques et dans quels cas ils peuvent être utilisés pour faire de l'IBC.

Les *couplages* ont été découverts par les mathématiciens dans les années 1950, tandis que leur première application à la cryptographie, et plus particulièrement à l'ECC (pour « *Elliptic Curve Cryptography* ») date des années 1980-1990. Au départ, les *couplages* de Weil et Tate furent utilisés comme moyens d'attaques contre les systèmes cryptographiques basés sur les courbes elliptiques. L'idée était de réduire le problème du logarithme discret sur les courbes elliptiques au DLP sur des corps finis, où celui-ci se révèle plus « facile ». C'est ce qu'on appelle la **réduction de Menezes-Okamoto-Vanstone** (ou MOV) (5).

Etant donné $u \in G$ et $u^a \in G$ on peut calculer $e(u, u) \in G_t$ et $e(u, u^a) \in G_t$. Puis, on peut utiliser le logarithme discret sur G_t pour obtenir a . Un *couplage* étant une application d'une courbe elliptique vers un corps fini³, celui-ci va intuitivement permettre de transposer un problème de l'espace de départ au même problème dans l'espace d'arrivée. En fait, un théorème assure que pour n'importe quel *couplage* donné, le DLP sur l'espace d'arrivée ne peut pas être plus difficile que le DLP sur l'espace de départ.

Un paramètre critique des courbes elliptiques est l'**embedding-degree** (cf. II.2), nous le noterons k . Pour une courbe elliptique dont le groupe abélien associé est d'ordre n , le problème du logarithme discret requiert $2^{n/2}$ opérations. En effet, le meilleur algorithme connu pour résoudre le problème du logarithme discret sur les courbes elliptiques est le « *parallelized Pollard rho algorithm* » (6), lequel est basé sur le paradoxe des anniversaires et dont le temps d'exécution est en $O(\sqrt{n})$. Ainsi, une courbe de 256 bits est suffisante pour respecter le « standard » de sécurité de 128 bits.

On peut démontrer que sur une courbe donnée à n bits, un *couplage* permet de transposer le DLP sur la courbe vers un DLP sur un sous-groupe multiplicatif d'un corps fini à $k \times n$ bits. Cette propriété illustre le fait qu'**à sécurité constante, les courbes elliptiques permettent de travailler sur des espaces k fois plus petits que les algorithmes traditionnels type RSA**. Néanmoins, revers de la médaille, à nombre de bits constant les opérations sont beaucoup plus coûteuses en temps sur les courbes elliptiques. Par ailleurs, plus k sera grand plus on pourra se permettre de travailler avec des courbes avec n petit, ce qui peut être un avantage non négligeable pour certaines applications où la mémoire disponible est faible. On voit également qu'un k petit risque de présenter un risque pour la sécurité du système cryptographique.

Avec une courbe E avec $n = 256$ bits et $k = 2$, le DLP sur cette courbe n'est pas plus difficile que le DLP sur un sous-groupe d'un corps fini à 512 bits, ce qui est aujourd'hui facilement cassable.

³ En fait en toute rigueur un *couplage* ne fait pas correspondre E mais le sous-groupe G associé à E .

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 23

En effet, en 2007, l'équipe d'Antoine Joux a annoncé avoir réussi à calculer le logarithme discret d'un nombre à 530 bits (7). En 2012, le record était de 923 bits (8).

Fort heureusement, une courbe elliptique « normale » a un très grand *embedding-degree*. Typiquement, une courbe de 256 bits a un *embedding-degree* k autour de 2^{255} .

A ce stade, on peut se dire qu'il suffit de prendre un *embedding-degree* grand pour que le problème de sécurité ne soit plus d'actualité. Hélas, pour être utilisable pour de l'IBC, un *couplage* nécessite que le produit $k \times n$ ne soit pas trop grand. En effet, lorsque le couplage est utilisé pour faire de l'IBC, de nombreuses opérations relativement lourdes seront effectuées⁴ sur les résultats du *couplage*, dont la dimension est justement $k \times n$ bits. Ainsi un *couplage* utilisé pour faire de l'IBC, et non plus pour attaquer une courbe elliptique, requiert d'utiliser une courbe « faible » dans le sens où le paramètre de sécurité k ne doit pas être trop grand. Dans leur article de 2001, Boneh et Franklin utilisèrent une courbe de 512 bits qui avait la propriété d'être « supersingulière », ce qui se traduisait dans ce cas par un paramètre $k=2$. Ainsi, la sécurité était analogue à celle du problème DLP dans un corps fini de 1024 bits, ce qui est fortement similaire au RSA 1024 bits.

Les courbes elliptiques sont fortement sécurisées car personne n'est à même de trouver une structure interne permettant d'augmenter la vitesse de résolution du problème du logarithme discret (ou d'autres problèmes associés comme CDH par exemple) – sauf les couplages mais qui ne sont utilisables que pour attaquer certaines courbes dites « faibles ». Ainsi, concernant la cryptographie basée sur l'identité, **il faudra trouver un juste équilibre entre sécurité et performances** (calculabilité des couplages). Si on utilise une courbe « normale » *i.e.* avec un *embedding-degree* k trop grand, l'implémentation pratique des protocoles IBC demandera trop de temps de calcul. Au contraire, si l'on utilise une courbe trop « faible » (*i.e.* avec un k trop petit) alors on fait courir un risque important à la sécurité du système car on est alors capable de résoudre le problème du logarithme discret par réduction MOV.

Dans la partie suivante, nous allons reprendre en grande partie la conclusion précédente en théorisant plus rigoureusement les conditions nécessaires au choix d'une « bonne courbe » elliptique et en particulier d'une « bonne courbe » pour faire de la cryptographie basée sur l'identité.

⁴ Comme exponentiation modulaire par exemple.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 24

2.3. Choisir une courbe elliptique « IBC-friendly »

Les couplages sont des objets mathématiques à double tranchant pour la cryptographie : d'un côté, ils apportent des propriétés uniques permettant en particulier d'implémenter des protocoles novateurs comme l'IBE par exemple (cf. pages 15 et 32), d'un autre côté, ils peuvent être utilisés pour casser le DLP sur les courbes elliptiques par réduction MOV. Nous allons voir maintenant comment choisir une courbe elliptique permettant de mettre à profit les avantages des couplages tout en conservant un haut niveau de sécurité. Pour cela commençons par définir plus rigoureusement le paramètre k , ou « *embedding-degree* », que nous avons introduit dans le paragraphe précédent.

Embedding-degree : Soit E une courbe elliptique définie sur le corps fini $K = \mathbb{F}_q$. Soit $G \subseteq E(\mathbb{F}_q)$ un groupe cyclique d'ordre r (où r est le plus grand possible). Soit k le plus petit entier positif tel que r divise $q^k - 1$. Alors, on dit que k est l'*embedding-degree* de G . Intuitivement l'« *embedding-degree* » correspond au **ratio des tailles q^k de l'extension du corps fini G_t et de r , l'ordre de G** . Pour des raisons de sécurité, en particulier du fait de la possibilité d'attaque par réduction MOV (5), on souhaite que ce rapport soit le plus élevé possible. On retrouve donc logiquement l'assertion de la partie précédente.

$$k = \frac{\#E(\mathbb{F}_{q^k})}{\#E(\mathbb{F}_q)} \sim \frac{q^k}{r}$$

Souvent on parlera d'*embedding-degree* d'une courbe $E(\mathbb{F}_q)$ au lieu d'*embedding-degree* du sous-groupe de $E(\mathbb{F}_q)$ d'ordre r où r est le plus grand diviseur premier de $n = \#E(\mathbb{F}_q)$ (i.e. le nombre de points de $E(\mathbb{F}_q)$). Cet abus de langage vient du fait qu'on cherchera toujours des courbes où r et $\#E(\mathbb{F}_q)$ sont proches.

Nous avons vu que construire un couplage sur une courbe elliptique relève d'un équilibre difficile. Comme déjà dit dès le début de ce chapitre, \mathbb{F}_q doit être le plus grand possible de manière que $E(\mathbb{F}_q)$ puisse contrecarrer toute tentative d'attaque sur le logarithme discret. Par ailleurs dans la partie précédente nous avons vu que $G_t = \mathbb{F}_{q^k}$ devait être également le plus gros possible de manière à résister aux attaques par réduction MOV. Mais, dans le même temps, \mathbb{F}_q et \mathbb{F}_{q^k} doivent être le plus petit possible pour minimiser le temps de calcul et l'espace de stockage. Plus précisément :

Condition 1 : r doit être le plus grand entier premier possible de manière à ce que les attaques sur le logarithme discret d'ordre r soient inefficaces. Puisque $q \sim \#E(\mathbb{F}_q)$ il en va de même pour q .

Condition 2 : q doit être le plus petit possible de manière à ce que les calculs dans \mathbb{F}_q soient les plus rapides possibles.

Condition 3 : q^k doit être le plus grand possible de manière à ce que les attaques par réduction MOV sur le corps fini \mathbb{F}_{q^k} soient inefficaces. On peut également montrer que

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 25

pour des raisons de sécurité, q ne doit pas avoir un faible poids de Hamming, ni être une puissance première trop petite.

Condition 4 : q^k doit être aussi petit que possible pour que les opérations et calculs dans le corps fini \mathbb{F}_{q^k} soient efficaces. Toutes choses égales par ailleurs, q^k doit être aussi petit que possible pour que les opérations soient le plus rapides possibles.

Les trois premières conditions doivent être vérifiées pour toutes primitives cryptographiques sur les courbes elliptiques, pas seulement pour les applications de l'IBC utilisant les couplages. Seule la quatrième contrainte est due à l'utilisation spécifique des applications bilinéaires. Cette dernière condition, requérant que \mathbb{F}_{q^k} soit suffisamment petit pour y faire des calculs, est responsable à elle seule de la difficulté de trouver une courbe adaptée à l'IBC (*i.e.* avec un k petit). Dans la suite, nous appellerons les courbes elliptiques vérifiant les 4 conditions précédentes des courbes « **IBC-friendly** » (ou « *pairing-friendly* »).

Il est difficile aujourd'hui de savoir *a priori* quel est l'*embedding-degree* k minimum à choisir pour garantir un bon niveau de sécurité. De nombreuses attaques sont réalisées sur des courbes elliptiques dont le paramètre k est de plus en plus grand. Typiquement, on pense aujourd'hui qu'avoir un ordre r de 160 bits est acceptable. Concernant q^k , 1024 bits semble un bon compromis pour nombre d'applications, et les calculs sur de tels corps finis sont tout à fait faisables. Idéalement on a $r \sim \#E(\mathbb{F}_q) \sim q$ et donc q est également de l'ordre de 160 bits. Cela nous donne un *embedding-degree* :

$$k = \frac{\#E(\mathbb{F}_{q^k})}{\#E(\mathbb{F}_q)} \sim \frac{q^k}{r} = \frac{1024}{160} = 6.4$$

Un paramètre de sécurité égal à 7 devrait donc être satisfaisant pour le moment du point de vue de la sécurité. Néanmoins, l'équipe du professeur Antoine Joux est aujourd'hui spécialisé dans les attaques contre les systèmes cryptographiques sur les courbes elliptiques et ses attaques avancent vite. Après discussions avec un spécialiste du domaine, un paramètre de sécurité supérieur à 10 semble donc plus raisonnable afin de se mettre à l'abri, un certain temps au moins.

Avant de clore ce paragraphe, il nous faut définir un autre paramètre important. Il s'agit du paramètre ρ défini comme la taille du corps fini K relativement à l'ordre du sous-groupe G :

$$\rho = \frac{\log(q)}{\log(r)}$$

Précédemment nous avons vu que dans l'idéal nous voulions choisir un sous-groupe G pour le couplage le plus grand possible c'est-à-dire en prenant $r \sim \#E(\mathbb{F}_q) \sim q$. Dans ce cas, on a alors $\rho = 1$. En effet, des courbes elliptiques avec une petite valeur de ρ sont avantageuses si l'on souhaite accélérer les opérations arithmétiques sur celles-ci. Par exemple, considérons une courbe elliptique avec un sous-groupe G de 200 bits et $\rho = 1$ et une autre avec un sous-groupe de même taille mais avec $\rho = 2$. La première courbe est donc définie sur 200 bits alors que la deuxième est définie sur 400 bits. Ainsi, les opérations pourront être effectuées beaucoup plus rapidement sur la première.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 26

2.4. Trouver une courbe « IBC-friendly »

Dans la littérature on trouve de nombreuses taxonomies de courbes « IBC-friendly ». Le lecteur pourra en particulier se référer aux références (9) et (10). Comme nous l'avons vu dans les paragraphes précédents, le paramètre de sécurité k est étroitement lié au niveau de sécurité souhaité pour le système cryptographique. Nous avons repris un certain nombre de résultats présents dans la littérature que nous avons résumés dans le tableau ci-dessous :

Niveau de sécurité	Ordre du sous-groupe r	Taille du corps fini d'arrivée q^k	<i>Embedding-degree</i> k avec $\rho = 1$	<i>Embedding-degree</i> k avec $\rho = 2$
80	160	960-1280	6-8	3-4
112	224	2200-3600	10-16	5-8
128	256	3000-5000	12-20	6-10
192	384	8000-10000	20-26	10-13
256	512	14000-18000	28-36	14-18

Tableau 3 : Lien entre niveau de sécurité et *embedding-degree* (Source : (9))

Ainsi, comme on peut le voir sur la Figure 6, il va nous falloir construire des courbes avec des paramètres de sécurité très divers pour garantir tel ou tel niveau de sécurité. Par ailleurs, nous avons distingué les cas selon les valeurs du paramètre ρ . Pour rappel, plus ρ est proche de 1, plus les opérations arithmétiques seront effectuées rapidement sur la courbe ainsi construite. On regardera donc le plus souvent la colonne correspondante à $\rho = 1$. Quoi qu'il en soit, il va nous falloir construire des courbes elliptiques dont le paramètre de sécurité correspond au niveau de sécurité souhaité. Pour un niveau de sécurité analogue à l'AES 128 bits, une courbe avec $k=12$ est au moins nécessaire.

Pour construire de telles courbes, une première solution pourrait être, naïvement, de générer aléatoirement des courbes elliptiques jusqu'à en trouver une qui respecte les quatre conditions précédentes tout en ayant le paramètre k souhaité. Hélas, une courbe elliptique choisie au hasard aura un très grand paramètre de sécurité de manière presque certaine (avec une probabilité très proche de 1). Plus précisément, une courbe prise au hasard aura un paramètre moyen $k \sim 2^{255}$. Un paramètre de sécurité aussi grand la rendra évidemment insensible aux attaques de type MOV (5) et Frey-Ruck (11) mais *a contrario* cela la rendra inutilisable en pratique pour faire de l'IBC car un tel paramètre k provoque des temps de calcul pour les couplages considérable. Il va donc falloir définir des approches constructives pour trouver des courbes « IBC-friendly ».

Une seconde approche serait de rechercher notre courbe « IBC-friendly » parmi les courbes dites « **supersingulières** » (ou « *Supersingular elliptic curve* » en anglais) : Soit $E[n]$ le sous groupe n -torsion de E c'est à dire l'ensemble des points divisant n dans E . Alors E est dite supersingulière si et seulement si $E[n] = \{O\}$ où O est le point de E à l'infini. La propriété intéressante de ces courbes est d'avoir un petit paramètre de sécurité et l'avantage est qu'elles sont très faciles à concevoir. D'ailleurs la première implémentation d'un protocole IBE utilisait une courbe supersingulière (3). Hélas, le paramètre de sécurité de ces courbes ne dépasse pas 6. Or nous avons vu que cela

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 27

risquait très vite de provoquer un problème de sécurité avec l'augmentation des puissances de calcul. D'ailleurs, le Tableau 3 illustre bien le lien entre sécurité et paramètre k : un paramètre de sécurité d'au plus 6 ne pourra satisfaire que des niveaux de sécurité en deçà de 80 bits.

Les courbes supersingulières sont donc trop « faibles » pour garantir un niveau de sécurité satisfaisant

Voici résumé dans le tableau ci-dessous la liste de courbes supersingulières :

Embedding-degree k	Ordre du groupe q	$E(\mathbb{F}_q)$
1	p^{2m}	$q \pm 2\sqrt{q} + 1$
2	p^{2m+1} ou p^{2m} et $p \not\equiv 1[4]$	$q + 1$
3	p^{2m} et $p \not\equiv 1[3]$	$q \pm \sqrt{q} + 1$
4	2^{2m+1}	$q \pm \sqrt{2q} + 1$
6	3^{2m+1}	$q \pm \sqrt{3q} + 1$

Tableau 4 : Liste des courbes supersingulières

Si l'on souhaite pouvoir garantir des niveaux de sécurité supérieurs à 80 bits (cf. Tableau 3), il va donc falloir être capable de construire des courbes avec des paramètres de sécurité ni trop grands ni trop petits : supérieurs à ceux des courbes elliptiques supersingulières, tout en restant raisonnables.

Des méthodes de construction de courbes ayant un paramètre de sécurité supérieur à 6 existent. C'est par exemple le cas de la méthode de multiplication complexe (ou « CM ») qui permet de construire une courbe ayant un paramètre k donné avec pour seules données d'entrées un entier n et un nombre premier q . Nous ne présenterons pas ces méthodes dans ce rapport. Le lecteur intéressé pourra se référer aux références bibliographiques (10) et (12).

Une collaboration active avec des chercheurs travaillant sur la sécurité des courbes elliptiques nous a permis d'arriver à la conclusion, qu'à l'heure actuelle, une courbe avec $k = 10$, comparable au niveau de sécurité de l'AES 192 bits, serait un bon compromis.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 28

3. Les applications de l'IBC

3.1. Signature numérique

La signature numérique est un mécanisme permettant d'authentifier l'auteur ou l'expéditeur d'un document électronique. Dans un premier temps, nous allons étudier un schéma de signature courte rendu possible grâce à la cryptographie basée sur les couplages sur les courbes elliptiques. Dans un deuxième temps nous décrirons un protocole de signature électronique basé sur l'identité.

3.1.1. Schémas de signature courte

En 2001, Dan Boneh, Ben Lynn, et Hovav Shacham (1) ont conçu un schéma de signature numérique courte (souvent contracté en « BLS ») permettant à un utilisateur de vérifier que le signataire d'un message est authentique. Ce schéma repose sur l'utilisation des couplages qui servent plus précisément à la phase de vérification que nous allons expliciter plus loin. Par ailleurs, dans ce schéma, les signatures sont des sous-groupes de points de courbes elliptiques. Nous avons vu précédemment que le niveau de sécurité d'une courbe elliptique E à n bits est équivalent au niveau de sécurité d'un corps fini de $k \times n$ bits où $k > 1$ est le paramètre de sécurité de la courbe elliptique utilisée. Ainsi, en utilisant des courbes elliptiques avec un k « suffisamment » grand (cf. page 23) on peut se permettre de signer les messages avec des clés plus courtes qu'avec les méthodes traditionnelles de signature numérique, comme par exemple FDH (pour « Full Domain Hash ») ou DSA qui reposent sur RSA, tout en conservant un niveau de sécurité équivalent. Ce protocole permet par exemple de signer avec une clé de 170 bits pour une sécurité équivalente à du DSA-320 bits en utilisant une courbe à $k=2$.

Pour pouvoir implémenter le BLS il faut utiliser des groupes où CDH reste difficile mais où DDH est facile. C'est ce que nous avons appelé des groupes GDH (pour « Gap Diffie-Hellman », cf. page 19). Nous détaillons ci-dessous le schéma de ce protocole :

Schéma du protocole de signature courte BLS

Soit une courbe elliptique E définie sur le corps fini K . Soit G un sous-groupe GFH de cette courbe elliptique et g un de ses générateurs. G et g sont tous deux des paramètres du système. Soit $\sigma \in G$ une signature. On note par ailleurs G^* l'ensemble composé des éléments de G sauf l'identité. Le schéma de signature numérique GDH permet de créer une signature pour n'importe quel message $M \in \{0,1\}^*$. Soit h une fonction de hachage $h : \{0,1\}^* \rightarrow G^*$.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 29

Le protocole est composé de trois étapes *KeyGen*, *Sign* et *Verify* :

- **Production de la clé** (ou « *Key Generation* ») : Choisir aléatoirement un nombre s dans \mathbb{Z}_p^* et calculer $y = g^s$. La clé publique est y . La clé privée est s .
- **Signature** : Etant donné un message $M \in \{0,1\}^*$ et la clé privée s , calculer $h = h(M)$ puis $\sigma = h^s$. La signature est $\sigma \in G^*$.
- **Vérification** : Etant donné la clé publique y , un message M et la signature σ , calculer $h = h(M)$ puis vérifier que $e(g, \sigma) = e(y, h)$. Si ce n'est pas le cas la signature est rejetée.

La preuve de sécurité théorique du protocole BLS fait l'hypothèse que la fonction de hachage h est un *random oracle*. Plus récemment en 2008, Dan Boneh et Xavier Boyen ont publié un article (13) décrivant un protocole de signature courte prouvé infalsifiable dans le modèle « *adaptive chosen message attack* » sans oracle aléatoire et avec les mêmes avantages que les protocoles précédents. Dans ce nouveau protocole, la courbe elliptique doit être définie sur un groupe non plus GDH mais SDH (pour « *Strong Diffie-Hellman* »). L'hypothèse SDH définie pour l'occasion est fortement analogue à la « *Strong RSA assumption* » (14) avec laquelle elle partage de nombreuses caractéristiques.

Concernant les performances, plus le paramètre de sécurité k est grand plus on pourra se permettre de prendre une taille de clé l petite. Par exemple dans l'article de Boneh de 2001, la courbe elliptique utilisée est une courbe supersingulière de paramètre $k = 6$ et définie sur le corps $\mathbb{F}_{3^{97}}$. Avec cette courbe, la signature est de 154 bits (ce qui équivaut dans ce cas à un niveau de sécurité de 923 bits sur les corps finis). Par ailleurs, avec un processeur Pentium III de 1GHz, le temps de calcul total pour la signature et la vérification est au total de 2.9s (1). Dans un autre article plus récent, avec un Core 2 Duo de 1,6 GHz, ce temps était réduit à 0.17s (15).

3.1.2. Signature basée sur l'identité

De manière analogue au protocole de chiffrement basé sur l'identité que nous avons introduit page 15, il pourrait être extrêmement pratique et utile de disposer d'un schéma de signature numérique où la vérification de la signature ne dépend que de l'identité du signataire. La construction d'un tel protocole est en réalité étroitement liée à celui de l'IBE. Boneh et Franklin ont décrit dans (3) une méthode générique pour convertir un protocole d'IBE en protocole de signature numérique basé sur l'identité. En particulier, la signature d'un message M se fait par l'intermédiaire de la clé privée définie dans l'IBE et les calculs de couplages sont encore une fois utilisés pour vérifier l'identité de l'expéditeur du message.

En 1984, Shamir présentait déjà un schéma de ce type mais en utilisant les algorithmes classiques sur des corps finis reposant sur le problème de la factorisation (2). Ce n'est que beaucoup plus récemment, au début des années 2000, suite à l'article de Franklin et Boneh de 2001 (3), que des protocoles de signature basés sur l'identité utilisant les couplages sur les courbes elliptiques ont été proposés. C'est par exemple le cas de (16). Dans ces articles, l'étape de

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 30

vérification de la signature requerrait le calcul d'au moins 2 ou 3 couplages. Or, le temps de calcul des couplages est l'étape la plus longue de ces algorithmes. Nous allons présenter ici un schéma de signature plus récent et performant ne nécessitant qu'un seul calcul de couplage. Ce schéma est tiré de (17) et (18).

Signature basée sur l'identité

Soit une courbe elliptique E définie sur le corps fini K . Soit G un sous-groupe de points sur une courbe elliptique. Soit g un générateur de G . Le protocole est composé de quatre étapes : *Setup*, *Extract*, *Sign* et *Verify*.

- **Setup** : Le PKG choisit au hasard un entier relatif s . Il choisit également 3 fonctions de hachage cryptographiques $h_1 : \{0,1\}^* \rightarrow G$, $h_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ et $h_3 : G \rightarrow \mathbb{Z}_q$. Enfin, il calcule et rend public $g_{pub} = g^s$. Les paramètres du système sont $(g, g_{pub}, h_1, h_2, h_3)$ et s est la « Master Key » privée du système connu uniquement du PKG.
- **Extract** : Dans ce schéma, la signature est réalisée avec $S_{ID} = Q_{ID}^s$ où $Q_{ID} = h_1(ID)$ est la clé publique associée à l'identité ID servant à vérifier la signature de l'expéditeur.
- **Sign** : Pour signer un message $M \in \{0,1\}^*$ l'expéditeur doit calculer la paire (R, S) avec : $R = g^s$ et $S = (g^{h_2(M)} + S_{ID}^{h_3(R)})^{1/s}$.
- **Verify** : Pour vérifier une signature (U, V) d'un message M il faut vérifier que :

$$e(U, V) = e(g, g)^{h_2(M)} \cdot e(g_{pub}, Q_{ID})^{h_3(R)}$$

La fonction de hachage h_3 peut ici être omise sans risque de sécurité particulier. Le coût en termes de calculs correspond principalement aux trois calculs de couplage lors de la phase de vérification (les autres calculs sont négligeables par rapport à ceux-là). Néanmoins, notons que $e(g, g)$ est une valeur constante dans G_t et peut donc être calculée puis stockée pour les vérifications ultérieures. Par ailleurs, $e(g_{pub}, Q_{ID})$ ne dépend pas du message M mais uniquement de l'identité ID de l'expéditeur. Ainsi, le coût de ce schéma sera amorti par de multiples utilisations. Dans cette situation, le coût de ce schéma ne nécessite qu'un seul calcul de couplage $e(U, V)$, sauf à la première utilisation.

La preuve de sécurité théorique de ce protocole fait l'hypothèse que les fonctions de hachage h_1 , h_2 et h_3 sont des oracles aléatoires. Plus récemment, Brent Waters a présenté dans un article de 2005 (19) un protocole de signature basé sur l'identité, sécurisé dans l'hypothèse CDH et sans oracle aléatoire.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 31

3.2. Chiffrement basé sur l'identité (IBE)

Nous avons déjà parlé du protocole de chiffrement basé sur l'identité au premier chapitre. Nous exposons ici le schéma proposé en 2001 par Boneh et Franklin (3).

Le protocole IBE se compose de quatre étapes : *Setup*, *Extract*, *Chiffrement*, *Déchiffrement*. Il existe différentes manières de chiffrer le message dans l'étape de chiffrement du protocole. Nous en donnons ici une parmi d'autres. Il ne s'agit pas de la version la plus sécurisée mais le principe et la structure étant identiques pour toutes les variantes, il sera aisé d'en implémenter une autre ensuite.

Protocole IBE (Boneh et Franklin 2001 (3))

Soit une courbe elliptique E définie sur le corps fini K . Soit G un sous-groupe de points sur courbe elliptique d'ordre q . Un protocole IBE est composé de cinq phases :

- **Setup** : Le PKG définit un couplage $e : G \times G \rightarrow G_t$ et sélectionne un point $P \in G$. Il choisit au hasard un entier relatif s . Il choisit également deux fonctions de hachage cryptographiques $h_1 : \{0,1\}^* \rightarrow G$ et $h_2 : G_t \rightarrow \{0,1\}^*$. Enfin, il calcule et rend public $g_{pub} = g^s$.
- **Extract** : Le PKG génère la clé privée correspondant à l'identité ID en calculant $S_{ID} = Q_{ID}^s$ où $Q_{ID} = h_1(ID)$ est la clé publique associée à l'identité ID servant à vérifier la signature de l'expéditeur. Notons que h_1 a pour rôle de faire correspondre à l' ID d'un utilisateur un point sur la courbe elliptique.
- **Chiffrement** : Pour chiffrer un message $M \in \{0,1\}^*$, l'expéditeur doit choisir aléatoirement $r \in \mathbb{Z}_q$ puis calculer $U = g^r$. L'expéditeur calcule également $g_{ID} = e(Q_{ID}, g_{pub})^r$. Pour chiffrer le message M , l'expéditeur fait $C = M \oplus h_2(g_{ID})$ et envoie finalement la sortie (U, C) au destinataire.
- **Déchiffrement** : Le destinataire, une fois le message codé C reçu, doit faire une requête auprès du PKG pour obtenir sa clé privée. Bien sûr cette phase est critique du point de vue de la sécurité et en particulier une authentification mutuelle entre le destinataire et le PKG est indispensable pour empêcher une attaque « *man in the middle* ». Nous ne rentrerons pas ici dans des considérations pratiques et laissons cela pour le chapitre suivant. Une fois sa clé privée S_{ID} reçu, le destinataire calcule :

$$x = e(S_{ID}, U) = e(Q_{ID}^s, g^r) = e(Q_{ID}^r, g^s) = e(Q_{ID}^r, g_{pub}) = e(Q_{ID}, g_{pub})^r = g_{ID}$$

Une fois cette valeur obtenue l'utilisateur n'a plus qu'à déchiffrer le message en calculant $M = C \oplus h_2(g_{ID})$.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 32

3.3. Protocole d'accord sur une clé

Nous avons vu au chapitre précédent que l'IBC repose en particulier sur l'utilisation des couplages sur les courbes elliptiques. Outre les avantages de l'IBC présentés dans le premier chapitre, l'utilisation des couplages et des courbes elliptiques permet d'accroître le niveau de sécurité des protocoles à nombre de bits d'encodage constant (cf. réduction MOV). Néanmoins, ces avantages se font au détriment des performances, en particulier le calcul des couplages peut prendre beaucoup de temps. Ainsi la cryptographie basée sur les couplages (et l'IBC) est beaucoup moins rapide que les schémas cryptographiques « traditionnels » (cryptographie symétrique surtout mais également cryptographie avec PKI). D'où l'idée de se servir uniquement de la cryptographie basée sur l'identité pour échanger une clé secrète entre les deux protagonistes laquelle sera ensuite utilisée pour faire du chiffrement symétrique.

Protocole d'accord sur une clé basée sur l'identité

Soit une courbe elliptique E définie sur le corps fini K . Soit G un sous-groupe de points sur courbe elliptique. Soit g un générateur de G . Un protocole de mise en accord de clé entre deux individus est composé de deux étapes :

- ✓ L'authentification mutuelle afin de se protéger contre les attaques « *man in the middle* ».
- ✓ La distribution de clé entre eux deux.

Or, la phase d'authentification peut être effectuée très simplement en utilisant le schéma de signature numérique basée sur l'identité que nous avons détaillé précédemment. Ainsi, nous nous contenterons ici de détailler la deuxième phase du protocole, celle de l'échange de clé entre Alice et Bob. Ce protocole, tout comme celui de l'IBE sur lequel nous reviendrons dans quelque temps, est composé de 4 étapes.

- **Setup** : Le PKG définit un couplage $e : G \times G \rightarrow G_t$. Il choisit au hasard un entier relatif s . Il choisit également deux fonctions de hachage cryptographiques publiques $h_1 : \{0,1\}^* \rightarrow G$ et $h_2 : G_t \rightarrow \{0,1\}^*$. Enfin, il calcule et rend public $g_{pub} = g^s$.
- **Extract** : Le PKG génère la clé privée correspondant à l'identité ID en calculant $S_{ID} = Q_{ID}^s$ où $Q_{ID} = h_1(ID)$ est la clé publique associée à l'identité ID servant à vérifier la signature de l'expéditeur. Notons que h_1 a pour rôle de faire correspondre à l' ID d'un utilisateur un point sur la courbe elliptique.
- **Echange de clé étape 1** : L'expéditeur choisit un nombre aléatoire $r \in \mathbb{F}_q$ et calcule $U = Q^r$ et $k = h_2\{e(Q_{ID}, g_{pub}^r)\}$. Puis il envoie U au destinataire.
- **Echange de clé étape 2** : Une fois que le destinataire a bien reçu U de la part de l'expéditeur Alice (la vérification se fait grâce aux signatures numériques) il calcule $k = h_2\{e(S_{ID}, g^r)\}$. A cette étape, Alice est certaine que seul Bob peut retrouver k car lui seul connaît sa clé privée S_{ID} .

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 33

Par ailleurs, il est aisé de vérifier que du fait de la propriété du couplage e , les deux valeurs calculées par chacun des deux protagonistes sont égales. En effet :

$$e(Q_{ID}, g_{pub}^r) = e(Q_{ID}, g^{rs}) = e(Q_{ID}, g)^{rs} = e(sQ_{ID}, g) = e(S_{ID}, g^r)$$

Et ainsi, Alice et Bob partagent une clé k dont ils vont pouvoir se servir comme clé de chiffrement symétrique pour des communications futures.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 34

3.4. *Identity-Based Broadcast Encryption (IBBE)*

Nous avons vu au Chapitre 1 que l'un des avantages principaux de l'IBC est de ne pas nécessiter de gestion des clés publiques et des certificats. Par ailleurs les clés publiques utilisées pour le chiffrement ne sont autres que les « identités » (adresses emails, numéros de téléphone, etc.) des utilisateurs. Grâce à ces deux avantages, nous allons voir que nous pouvons facilement concevoir un protocole de diffusion de contenu chiffré (ou *broadcast*) vers de multiples destinataires (et même groupes de destinataires) par exemple en fonction de leur rôle dans une organisation. Les résultats exposés ci-dessous sont tirés des références (20) et (21).

Dans un tel protocole un (ou plusieurs) *broadcaster(s)* souhaite(nt) pouvoir transmettre un message de manière confidentielle uniquement aux *utilisateurs* qui en ont le « droit », du fait de leur appartenance à un *groupe*. Ainsi, l'on souhaite construire un protocole avec les propriétés suivantes :

- Un message chiffré avec une certaine clé de chiffrement ne pourra être déchiffré que par les utilisateurs appartenant à un groupe donné \mathbb{U}^{ID_i} .
- Pouvoir créer facilement des groupes d'utilisateurs sans devoir impliquer la participation de ces derniers. Autrement dit un utilisateur pourra être catégorisé dans un ou plusieurs groupes sans qu'il n'en ait eu le choix ni même l'information.
- Les groupes pourront être très simplement mis à jour par le broadcaster (suppression ou ajout d'un utilisateur) sans devoir impliquer la participation des utilisateurs.
- Le protocole étant basé sur l'identité, le broadcaster pourra diffuser un message à un groupe d'utilisateurs uniquement grâce à l'identifiant ID de ce groupe.

Un tel protocole peut être extrêmement utile dans une entreprise. En effet dans toutes les organisations, les employés ont des rôles et des activités bien définies. Ces rôles peuvent être amenés à évoluer rapidement du fait de nouveaux projets, de changements de départements, de restructurations ou encore de promotions des employés. Par ailleurs, les structures de ces entreprises sont de plus en plus souvent matricielles et les employés sont ainsi regroupés en catégorie : « ingénieur R&D », « RH », « administrateur », etc. Evidemment, toutes ces catégories d'employés n'ont pas les mêmes droits d'accès à l'information qui transite dans l'entreprise. Par exemple, le dernier brevet déposé par l'équipe de R&D n'aura rien à faire sur la boîte email d'une secrétaire...

Ainsi, si le PDG d'une entreprise souhaite communiquer une information importante uniquement aux employés habilités « Confidentiel-Défense » (CD), alors, en tant que broadcaster, il pourra diffuser son message uniquement aux membres du groupe « CD ». Par ailleurs, si un nouvel employé obtient cette habilitation, il pourra être très facilement rajouté au groupe. De même, si un employé, du fait d'une faute professionnelle perd cette habilitation, la suppression de sa présence dans le groupe sera également aisée.

Pour ce protocole, on définit :

- Un broadcaster \mathcal{B} comme la (ou les) personne(s) pouvant diffuser de l'information.
- Un ensemble de m utilisateurs $\mathcal{U} = \{U_1, U_2, \dots, U_m\}$.
- Un ensemble de k groupes d'utilisateurs $\{\mathbb{U}^{ID_1}, \mathbb{U}^{ID_2}, \dots, \mathbb{U}^{ID_k}\}$ pouvant contenir un sous ensemble $\subseteq \mathcal{U}$ sachant que chaque utilisateur U_j peut appartenir à plusieurs groupes.
- \mathcal{B} possède une clé de chiffrement privée ε_i (correspondant au groupe \mathbb{U}^{ID_i}) et au sein d'un groupe \mathbb{U}^{ID_i} chaque utilisateur a une clé de déchiffrement privée D_i .

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 35

Par ailleurs il est bon de préciser que :

- Chaque groupe \mathbb{U}^{ID_i} a une unique identité ID.
- Chaque utilisateur possède une clé de déchiffrement qui lui est propre.
- Un message *broadcasté* à un groupe pourra être déchiffré par tous les membres de ce groupe.
- Pour des raisons de clarté, dans le protocole que nous allons ici présenter, un message ne peut être envoyé qu'à un seul groupe en même temps. Néanmoins, le protocole « complet » est disponible dans (20).

Protocole d'IBBE :

Soit une courbe elliptique E définie sur le corps fini K . Soit G un sous-groupe de points sur une courbe elliptique d'ordre q . Un protocole IBBE est composé de cinq phases :

- **Setup** : Le PKG définit un couplage $e : G \times G \rightarrow G_t$ et sélectionne un point $P \in G$. Il choisit au hasard un entier relatif s . Il choisit également deux fonctions de hachage cryptographiques $h_1 : \{0,1\}^* \rightarrow G$ et $h_2 : G_t \rightarrow \{0,1\}^*$.
- **Extract** : La construction de la clé de chiffrement doit prendre en compte l'identité du groupe ID_i ainsi que l'appartenance de tel ou tel utilisateur à ce groupe. En effet puisque l'on souhaite pouvoir mettre à jour facilement (et même construire) un groupe, il faut qu'un nouvel utilisateur du groupe puisse déchiffrer le message, et qu'un ancien utilisateur du groupe (ou un utilisateur non adhérent) ne le puisse plus (pas). Ainsi, il doit exister une application ϕ_1 entre la clé de chiffrement ε_i et un groupe \mathbb{U}^{ID_i} et une application ϕ_2 entre la clé de chiffrement ε_i et les clés de déchiffrements D_i . Nous laissons le lecteur intéressé par les détails de cette implémentation se référer à la référence bibliographique (20) et nous ne donnerons ici que les grandes lignes : on construit $E_1 = uP$ et $E_2 = uvP$ où u et v sont des entiers construits de manière à prendre en compte l'appartenance de chaque membre au groupe \mathbb{U}^{ID_i} . Par ailleurs, on construit les clés privées de déchiffrement comme $D_i = (xu_i + 1)v_i[q] \times h_1(ID)$ où (u_i, v_i) sont des paires d'entiers dites « *qualified* » (cf. (20) p. 6-7) et où x , la « *master private key* », est un nombre premier pris aléatoirement dans \mathbb{Z}_q . Le triplet (E_1, E_2, x) constitue la clé de chiffrement pour l'étape suivante.
- **Chiffrement** : Pour chiffrer un message $M \in \{0,1\}^*$, le broadcasteur \mathcal{B} doit choisir aléatoirement $r \in \mathbb{Z}_q$ puis calculer $R = E_2^r$ et $b = e(E_1, Q_{ID}^{x+1})$. Le message codé est obtenu en faisant $C = M \oplus h_2(b^r)$. La sortie (C, R) est ensuite broadcastée vers les utilisateurs du groupe \mathbb{U}^{ID_i} .
- **Déchiffrement** : Pour déchiffrer un message codé C , les utilisateurs du groupe ID_i cible doivent utiliser leur clé personnelle D_i pour calculer :

$$e(R, D_i) = e(E_2^r, Q_{ID}^{d_i}) = e(P^{rvu}, Q_{ID}^{(xu_i+1)v_i}) = e(P^{ru}, Q_{ID}^{(xu_i+1)u_i}) = e(E_1^r, Q_{ID}^{x+1}) = b^r$$
Une fois cette valeur obtenue l'utilisateur n'a plus qu'à déchiffrer le message en calculant $M = C \oplus h_2(b^r)$.
- **Mise à jour dynamique** : Pour supprimer ou ajouter un utilisateur à un groupe \mathbb{U}^{ID_i} , il suffit au broadcasteur \mathcal{B} de recalculer sa clé de chiffrement privée :
 - Quand un nouvel utilisateur U_z doit être rajouté au groupe \mathbb{U}^{ID_1} , \mathcal{B} fait $E_1 \leftarrow u_z \cdot E_1$ et $E_2 \leftarrow u_z \cdot E_2$
 - Quand un nouvel utilisateur U_z , doit être retiré du groupe \mathbb{U}^{ID_1} , \mathcal{B} fait $E_1 \leftarrow \frac{E_1}{u_z}$

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 36

$$\text{et } E_2 \leftarrow \frac{E_2}{u_{z'}}$$

On utilise alors la nouvelle paire (E_1, E_2) comme clé de chiffrement.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 37

3.5. IBC et sécurité du *Cloud-computing*

Aujourd'hui l'utilisation du « *cloud* » est de plus en plus récurrente au sein des entreprises. D'une part, les employés, dans un cadre privé stockent souvent leurs données personnelles sur des espaces en ligne. D'autre part, la collaboration professionnelle utilisant des services de *cloud computing* comme moyen de partage et d'échange de données tant à devenir omniprésent dans bon nombre d'organisations. Les services de stockage et d'échange basés sur le *cloud* comportent en effet de nombreux avantages pour les entreprises : en déportant leurs données sur le *cloud* les entreprises n'ont plus besoin de construire, administrer et maintenir les infrastructures de stockage locales. Par ailleurs, l'utilisation des services de *cloud computing* par les équipes et les employés d'une entreprise permet de rendre la coopération et le travail plus efficace et de gagner en productivité.

Hélas, à rebours de ces avantages indéniables, le stockage d'informations sur le *cloud* implique également d'importants problèmes de sécurité. En effet, en utilisant un service de stockage en ligne, l'entreprise perd sa souveraineté et sa volonté propre en matière de gestion et de sécurité de ses données : elle doit faire confiance au CLP (pour « *Cloud Service Provider* ») auquel elle a fait appel, lequel gère comme bon lui semble la sécurité de ses serveurs. Or, si le CLP ne fait pas d'effort particulier pour construire une politique de sécurité efficace, des attaquants dans une optique d'espionnage économique s'attaqueront aux serveurs du CLP pour voler les informations stratégiques des entreprises l'utilisant comme espace de stockage. Dans un contexte où les APTs (pour « *Advanced Persistent Threats* ») sont de plus en plus fréquentes, stocker des données critiques sur des espaces de stockage en ligne administrés et sécurisés par un tiers, sans mesure de protection, est un véritable suicide économique pour une entreprise. Une idée simple pourrait donc être de chiffrer ses données avant de les envoyer sur le *cloud*. Pour une fonction de stockage, cette solution est satisfaisante (il suffit à l'entreprise de faire du chiffrement symétrique par exemple). Mais pour partager des informations entre plusieurs employés, groupes ou départements, voir même avec certains partenaires externes, l'utilisation de la cryptographie est plus difficile. Comme nous l'avons expliqué au début de ce rapport, avec les protocoles de cryptographiques traditionnels le management des clés de déchiffrement peut rapidement devenir compliqué et coûteux (mise à jour des certificats, gestion des bases de données de clé, etc.). Plus précisément, du fait de cette complexité il est impensable d'avoir une paire de clé unique pour chaque fichier mis en ligne. Or, à moins de pouvoir physiquement transmettre à ses interlocuteurs la clé de déchiffrement, il existe un risque qu'un attaquant récupère cette clé et puisse ainsi accéder à toutes les informations stockées par l'entreprise sur le *cloud*.

Heureusement grâce aux chapitres précédents, nous savons que nous disposons maintenant de protocoles de chiffrement basés sur l'identité qui permettent de chiffrer efficacement l'information tout en s'abstrayant des contraintes précédentes. L'idée est donc d'utiliser l'IBE pour ne permettre le déchiffrement des informations stockées uniquement à un groupe d'utilisateur prédéfini. Beaucoup d'articles ont été écrits à ce sujet autant concernant des protocoles d'authentification que de chiffrement. Le lecteur souhaitant approfondir ce sujet pourra se référer aux références (22), (23) et (24). Nous ne ferons ici qu'aborder de manière informelle une possibilité parmi d'autres d'utilisation de l'IBC comme moyen de sécurisation des données stockées sur le *cloud*.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 38

Imaginons qu'une entreprise souhaite partager un document avec certains de ses employés et collaborateurs afin que ceux-ci puissent le lire et le modifier. Une bonne solution serait effectivement d'utiliser un espace de stockage sur le *cloud*, mais comme dit précédemment, cela pose d'importants problèmes de sécurité (confiance envers le CSP, stockage en clair, etc.). L'ensemble des employés et collaborateurs forme le groupe des personnes autorisées à lire et modifier le document. L'entreprise souhaite que seuls les membres du groupe puissent déchiffrer le document. L'entreprise prend alors le rôle de PKG (cf. Chapitre 1 et 2). Grâce aux avantages de l'IBC, on peut imaginer un protocole générant une paire de clé pour chaque fichier mis en ligne. Ainsi, même si un attaquant réussit à intercepter la clé de déchiffrement envoyée par le PKG aux membres du groupe, il ne pourra déchiffrer que le fichier correspondant, et encore, il reste aux attaquants à trouver le bon fichier (celui qui correspond à la clé de déchiffrement) sur le *cloud*, ce qui n'est pas si simple. En effet, le PKG chiffre le document qu'elle souhaite partager au groupe avec la clé publique :

$$Pub_D = Hash_{pub}(ID_G || ID_D)$$

Où $Hash_{pub}$ est une fonction de hachage publique, ID_G est l'identifiant du groupe et ID_D l'identifiant propre au document. L'identifiant ID_D est généré localement par l'entreprise (le PKG) et est dérivé à partir des métadonnées (MD) en utilisant une fonction de hachage telle que $ID_D = h(MD)$ ($h(MD)$ étant supposé unique). Ainsi, à partir de la clé de déchiffrement, un attaquant potentiel ne pourra *a priori* pas remonter jusqu'aux métadonnées du fichier et donc ne pourra pas identifier le fichier correspondant à la clé de déchiffrement qu'il aurait potentiellement récupéré.

Evidemment, une fois que l'entreprise a mis le document en ligne, reste le problème de la distribution des clés de déchiffrement aux membres du groupe. Plusieurs possibilités sont envisageable, mais comme dit précédemment, l'avantage de notre protocole est qu'une attaque « *man in the middle* » ne permettra pas à l'attaquant de faire beaucoup de dégâts.

Par ailleurs, à l'heure du *Big-Data* et de l'espionnage à grande échelle (cf. actualité toute récente sur le projet américain Prism par exemple) l'autre avantage de ce protocole est de garantir la confidentialité des données stockées par rapport à CSP lui-même. En effet, même dans le cas d'un CSP « malhonnête » qui souhaiterait utiliser les données stockées par ses clients à des fins marketing ou autres, celui-ci ne serait pas capable de « lire » l'information car ne disposant pas de la clé de déchiffrement (laquelle ne transite jamais par ses serveurs). De plus, puisque chaque fichier dispose d'une clé de déchiffrement différente, même dans le cas où le CSP aurait réussi à déterminer une clé, il ne pourrait déchiffrer qu'un seul fichier.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 39

Partie 2 : Notre solution basée sur l'IBC

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 40

4. Introduction

La cryptographie basée sur l'identité a connu un important retentissement dans le monde académique entre 2001 et aujourd'hui. Sur cette période, des centaines d'articles scientifiques ont été publiés décrivant des protocoles cryptographiques basés sur l'identité toujours plus sophistiqués et sécurisés. A rebours, très peu d'applications pratiques de ces nouveaux algorithmes ont vu le jour, malgré des avantages théoriques indéniables.

Ce domaine de la cryptographie, encore très peu mature, reste donc jusqu'ici cantonné au seul monde de la recherche, même si quelques acteurs comme *Voltage Security* ont commencé à l'utiliser à des fins commerciales, sans grand succès toutefois. Cette forte différence entre monde académique et monde industriel s'explique entre autres, par la difficulté pour les développeurs du niveau applicatif de s'approprier les concepts mathématiques sous-jacents (couplages, courbes elliptiques, etc.) afin d'implémenter les algorithmes de l'IBC. En particulier, le choix d'une courbe « *IBC-friendly* » relève d'un équilibre subtil nécessitant une expertise poussée pour les courbes elliptiques.

Par ailleurs, contrairement aux processus cryptographiques traditionnels, l'IBC est encore mal documenté, et les bibliothèques implémentant les « couplages » sont difficiles à prendre en main. Ainsi, le « *gap* » entre niveau fondamental et niveau applicatif, encore très important, empêche nombre d'entreprises et *start-up* ne disposant pas de connaissances mathématiques et cryptographiques bas niveau d'innover dans ce domaine.

Ainsi, notre principale motivation, après la synthèse théorique du chapitre précédent a été justement de passer le cap théorique, en montrant que l'IBC, et plus particulièrement l'IBE, pouvaient être utilisés avec grand profit dans des applications pratiques au niveau industriel. L'objet de ce second chapitre est donc de présenter les différents travaux applicatifs effectués pendant six mois.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 41

5. Motivations et problématiques

L'échange de données a toujours fait partie du monde connecté dans lequel nous vivons. Par exemple, depuis les années 1990, les emails sont devenus omniprésents dans les grandes organisations et entreprises. En moyenne, 72 courriels sont reçus et 33 sont envoyés par jour et par employé. Ces échanges d'informations, outre le fait de croître d'année en année, prennent aujourd'hui des formes différentes à travers des espaces collaboratifs et de partage tels ceux liés au « *cloud computing* » par exemple. Ainsi, dans bien des cas, on ne souhaite plus envoyer des données ou transmettre une information qu'à une ou plusieurs personne(s) bien identifiées. On souhaite dorénavant échanger et partager des informations, collaborer à distance sur un travail commun, etc. Par exemple, dans les entreprises, les espaces de travail collaboratifs comme Microsoft SharePoint permettent aux employés de travailler sur un sujet commun plus efficacement qu'ils ne pourraient le faire en s'envoyant de multiples emails entre eux. Dans le domaine de la Santé, les praticiens et professionnels du secteur souhaitent pouvoir partager facilement des données sur leurs patients afin d'améliorer la prise en charge et le diagnostic. Par exemple, un médecin généraliste peut souhaiter partager les données relatives aux antécédents médicaux d'un patient avec son confrère cardiologue afin de personnaliser la prescription médicamenteuse. Même l'usage des emails est amené à évoluer. Dans certaines situations, on peut vouloir envoyer un email sans connaître d'interlocuteur précis, mais seulement une fonction, un rôle ou un attribut générique identifiant un groupe d'individus dans une organisation. Dans un hôpital, un radiologue peut vouloir communiquer des données avec un chirurgien, sans pour autant connaître à l'avance avec quel chirurgien en particulier il souhaite communiquer. Dans une entreprise, on peut vouloir dialoguer avec un technicien du service informatique, sans savoir à qui d'adresser exactement, etc.

Ces nouveaux moyens de communications (service de type « *cloud* », etc.) sont très pratiques et permettent globalement d'augmenter la productivité de ceux qui les utilisent. Leur adoption massive est un mouvement mondial qui n'est pas prêt de s'arrêter. En particulier, les services liés au « *cloud computing* » sont très à la mode depuis quelques années. Ils permettent de délocaliser des données vers des espaces de stockage accessibles en ligne. Toute personne y étant autorisée pourra stocker et télécharger des données, vers, et à partir de ces serveurs situés dans les « nuages ». Grâce aux technologies du « *cloud computing* » tous les praticiens et professionnels d'un CHU⁵ pourront partager des données sur les patients passés et présents afin d'améliorer la communication entre eux, gagner du temps et faire moins d'erreurs. Ce « *cloud santé* » est de plus en plus pressenti pour devenir le nouveau moyen de communication entre, et à l'intérieur des hôpitaux, cliniques, cabinets et centres d'exams médicaux.

Bien que très pratiques, ces nouveaux vecteurs de transmission de l'information posent néanmoins d'importantes problématiques. Dans les entreprises de toutes tailles, le travail collaboratif et l'échange d'information en général, que ce soit par le biais des emails ou grâce à un service de type « *cloud* », présente en effet des risques majeurs d'un point de vue de la sécurité. Du fait de l'interconnexion de la plupart des organisations et entreprises mondiales au réseau Internet, le

⁵ Centre Hospitalier Universitaire

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 42

cyberespace est de plus en plus utilisé à des fins d'espionnages de tous types. Les attaques informatiques, hier opportunistes et diffuses, aujourd'hui ciblées et persistantes sont désormais motivées le plus souvent par l'obtention d'informations confidentielles. Une entreprise peut espionner son concurrent pour gagner des parts de marché et asseoir sa position. Un Etat peut souhaiter obtenir des informations stratégiques sur les décisions d'un autre pour mieux orienter sa politique étrangère et industrielle. Une compagnie d'assurance ou une banque peut vouloir obtenir les données médicales de ses clients à leur insu afin d'estimer leur probabilité à tomber malade ou leur espérance de vie, avant d'accepter, ou non, de leur vendre un service ou de leur faire un prêt...

Une solution, pour pallier ce problème de sécurité serait de chiffrer l'ensemble des données transitant par emails et sur les espaces collaboratifs. Aujourd'hui, la cryptographie asymétrique à base de PKI dont nous avons rappelé les grandes lignes de fonctionnement au chapitre premier, permet aux entreprises qui en ont les moyens financiers et humains de pouvoir chiffrer les emails envoyés entre ses employés. Nous montrerons dans la suite que, d'une part, les solutions à base de PKI sont déjà très coûteuses et complexes pour sécuriser les moyens traditionnels de partages de l'information (comme les emails par exemple) mais qu'elles se révèlent encore moins adapté quand il s'agit des nouveaux usages dont nous venons de parler.

Le véritable problème est donc de **concevoir une solution à la problématique de confidentialité des données dans un contexte de partage et de mise en commun de l'information**, qui permette de conserver la praticité des services de type « cloud » (tout particulièrement pour l'utilisateur final) tout en garantissant une sécurité optimale. Une solution prometteuse serait de pouvoir définir des contrôles d'accès basés sur le rôle ou la fonction des individus. Ces systèmes de contrôle d'accès, souvent appelée **RBAC** (pour « *Role-based Access Control* ») sont des systèmes considérés comme « idéaux » pour les entreprises et organisations. Un rapport de 2010 du NIST (*National Institute of Standards and Technology*) indiquait que la mise en place du RBAC dans les organisations pouvait avoir des conséquences économiques très positives. Les systèmes RBAC permettent notamment de :

- Réduire les couts d'administration système et réseau ainsi que de la maintenance des politiques de contrôle d'accès.
- Réduire les temps morts des employés grâce à une meilleure souplesse dans la définition des politiques de droit d'accès.

Pour le NIST, les avantages apportés par le RBAC en termes de facilité d'utilisation et d'administration pouvait amener une organisation de 10 000 employés à faire plus de 1 400 000\$ d'économie.

Nous montrerons dans la suite que cet équilibre praticité/sécurité ne peut pas être atteint avec les technologies existantes à base de PKI. En particulier, le fait de rendre possible le partage d'informations non pas à une ou plusieurs personne(s) donnée(s) mais à un « type » d'individu (identifié par leur fonction au sein d'une organisation par exemple) se révèle particulièrement périlleux avec les solutions actuelles du marché. A rebours, nous montrerons que l'implémentation des algorithmes d'IBE permet de lever la contrainte « sécurité *versus* praticité ».

Nous pensons que la solution que nous avons imaginée et développée est réellement innovante et présente de nombreux avantages par rapport aux solutions existantes.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 43

6. Formalisation des besoins

6.1. Des besoins multiples

Bien consciente des enjeux stratégiques et économiques liés à la sécurité de leur système d'information, les entreprises et organisations souhaitent que les services de partage et de travail collectifs en ligne soient le plus sûrs possibles. D'où un **besoin majeur de confidentialité et de sécurité**. Ce besoin est encore renforcé dans le contexte particulièrement sensible des données personnelles de santé. Ici plus que l'enjeu économique, c'est la vie privée et l'éthique médicale qui est en jeu. Les données de santé concernent des hommes et des femmes, elles sont de l'ordre de l'intime et doivent donc être manipulées avec une extrême prudence par les praticiens. Si l'évolution des usages tend vers le partage des données, et ce au bénéfice certain du patient (meilleure prise en charge, diagnostic individualisé, prescription personnalisée, meilleure continuité des soins, etc.), cela doit se faire dans un cadre très strict. Ainsi, toutes les données stockées, et partagées doivent être systématiquement chiffrées avant leur transmission sur le réseau. Concernant le déchiffrement, dans les entreprises et encore plus dans le monde de la santé, seul un cercle fermé et bien déterminé d'individu doit pouvoir accéder aux données en clair.

Justement, **flexibilité et sureté** de la politique d'accès aux données chiffrées sont primordiales. Sureté, car le futur système devra s'assurer que le déchiffrement des données confidentielles n'est possible que si les politiques associées sont bien satisfaites. Flexibilité, car nous pensons qu'il faut donner la capacité à l'utilisateur de décider lui-même des politiques qui contraignent le déchiffrement des données lorsqu'il les envoie « dans les nuages ». Plus précisément, nous pensons qu'aujourd'hui une manière efficace de répondre au besoin de confidentialité et de vie privée est de **contraindre le déchiffrement en fonction du rôle de l'individu** dans son organisation. Par exemple, dans un ministère, un Directeur Général peut vouloir chiffrer un email qui puisse être lu uniquement par les chefs de bureau. Dans une entreprise, un fichier Excel partagé sur SharePoint ne doit pouvoir être lu ou modifié que par les membres du département marketing par exemple. Au sein d'un hôpital, un médecin urgentiste peut souhaiter stocker et partager un document en ligne qui ne puisse être lu que par un psychiatre, etc.

Si les entreprises sont plus enclins qu'auparavant à faire des efforts d'un point de vue de la sécurité, elles ne sont par contre pas prêtes à sacrifier la praticité et la facilité d'utilisation pour cela. Elles refuseront par exemple des solutions de sécurité qui entraveraient les gains de productivité permis par les solutions de stockage et de partage en ligne. La conception d'un « *cloud* santé » sécurisé est vaine si ce dernier devient si difficile à utiliser en pratique qu'il en perd ses avantages comparatifs vis-à-vis des solutions traditionnelles. Ainsi, il est indispensable avant toutes autres choses de prendre en considération la **facilité d'utilisation pour l'utilisateur final**.

Cette facilité d'utilisation est d'ailleurs très importante vis-à-vis du besoin de sécurité lui-même. En effet, dans le cas où les protections de sécurité sont trop contraignantes, il n'est pas rare de voir l'utilisateur lui-même les contourner pour en éviter les effets négatifs. Parce qu'en sécurité encore plus qu'ailleurs « le meilleur est l'ennemi du bien » il faudra prendre garde à la **transparence** afin

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 44

d'éviter les effets contre-productifs de la « sur-sécurisation » des services de type « cloud » ou des boîtes emails.

Si le système devra être simple pour l'utilisateur, il devra l'être également du point de vue des administrateurs. Une administration complexe est couteuse en temps et donc en argent. De plus, plus la configuration est difficile, plus les risques d'erreurs, de pannes et de bugs sont importants. Ainsi, une administration facilitée n'est pas un luxe dans des contextes particulièrement critiques comme celui de la santé.

Le système devra également être **robuste vis-à-vis de la montée en charge**. A l'avenir le nombre de données et messages échangés et partagés va continuer à croître de manière exponentielle. Le système devra pouvoir gérer cette augmentation. Par ailleurs, la demande peut varier fortement sur une faible échelle de temps dans les grandes organisations. Dans un hôpital par exemple, en cas de crise sanitaire ou d'accident de grande ampleur, le nombre de chiffrements/déchiffrements de données va monter en flèche en quelques dizaines de minutes. Ainsi, il est nécessaire de disposer d'un système capable de maintenir ses fonctionnalités et ses performances en cas de forte demande.

Enfin, **l'évolutivité** est aussi un besoin important dans le monde professionnel en général. Si demain, un grand groupe industriel décide d'étendre son service de chiffrement des données sensibles à une autre de ces filiales, cela doit pouvoir se faire facilement. De même, si le « cloud santé », d'abord cantonné à quelques hôpitaux, doit s'étendre à tout un département voir à tout le territoire national, sa mise à l'échelle ne doit pas être trop complexe à mettre en œuvre.

Synthèse des besoins :

- Sécurité et confidentialité du système de bout en bout.
- Flexibilité et praticité des contraintes de déchiffrement (utilisation des rôles).
- Facilité d'administration.
- Transparence et simplicité d'utilisation pour l'utilisateur final.
- Robustesse à la montée en charge.
- Evolutivité.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 45

6.2. Quelques exemples

Nous présentons ici trois exemples dans des contextes différents : un premier dans le secteur de la Santé, un autre dans une entreprise et un troisième impliquant des relations privées/publiques entre un ministère et un grand groupe industriel. Ces exemples permettront aux lecteurs de mieux cerner les besoins énoncés plus haut.

6.2.1. Dans le secteur de la santé

Aujourd'hui dans le secteur médical, la plupart des échanges et interactions entre les différents praticiens se fait par format papier. Nous décrivons ci-dessous un scénario en y intégrant une future solution de sécurisation des données numériques de santé.

Scénario

Un patient P est suivi par un médecin généraliste et doit subir des examens approfondis en vue d'une opération de la main. Son médecin le dirige vers l'hôpital Necker pour prendre un premier rendez-vous avec l'un des membres de l'équipe du Dr. Ikierzek, grand spécialiste reconnu de la chirurgie de la main. Le médecin souhaite transmettre les antécédents médicaux du patient P sans connaître *a priori* la personne qui prendra en charge son patient. Lors du premier rendez-vous avec un des membres de l'équipe, on demande au patient P d'effectuer une radio de la main au centre d'examen avant de pouvoir se faire opérer. Les résultats seront ensuite transmis par le centre d'examen...

Etapes du cas d'usage

1. Le médecin généraliste veut transmettre de manière sécurisée les antécédents médicaux du patient P. Il souhaite que seuls les praticiens ayant le rôle de « membre de l'équipe du Dr. Ikierzek » puissent lire ces informations car il connaît très bien le Dr. Ikierzek à qui il fait totalement confiance.
2. Le médecin généraliste place donc le document sur le « *cloud* santé », regroupant tous les établissements de santé de la région, en le protégeant avec le rôle souhaité⁶.
3. Pour préparer la future opération avec le patient P l'un des chirurgiens spécialistes de la main demande à son interne de consulter les antécédents médicaux du patient. L'interne se rend sur le *cloud* et examine les antécédents médicaux. Il peut effectivement y accéder puisque qu'il possède le rôle de « membre de l'équipe du Dr. Ikierzek ». Il demande

⁶ Tout l'enjeu sera justement de permettre à l'expéditeur de protéger des données en spécifiant une politique de contrôle d'accès basée sur les rôles. Nous verrons dans la section suivante à quel point les solutions de chiffrements actuelles permettent, ou non, de répondre à ce problème.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 46

également au patient P d'aller faire une radio au service de radiologie se trouvant dans le même centre hospitalier.

4. Suite à la radio de la main du patient P, l'opérateur du centre d'examen place les documents sur le *cloud* régional en les protégeant avec le rôle « chirurgien de l'hôpital Necker ».
5. Pour préparer l'opération, le chirurgien prenant en charge le patient P ce jour-là consulte les résultats de la radio. Il ne peut demander à son interne de le faire pour lui, car celui-ci n'est pas encore chirurgien. Par contre, lui y a bien accès car il dispose du rôle « chirurgien de l'hôpital Necker » en plus de celui de « membre de l'équipe du Dr. Ikierzek ».

6.2.2. Autour des relations privées/publiques

Dans les administrations centrales de l'Etat, beaucoup d'informations concernant des entreprises publiques, mi privées et privées transitent. Certaines de ces informations sont confidentielles. C'est par exemple le cas à la DGCIS⁷ et à la DGCCRF⁸ au Ministère de l'Economie et des Finances. Ces directions générales sont par exemple amenées à lire des informations, concernant les opérateurs télécom. Ces informations sont souvent sensibles du point de vue de la concurrence entre les opérateurs. Il peut s'agir par exemple de choix et d'arbitrages effectués par ces derniers concernant une question technique donnée. Les opérateurs qui révèlent ces données aux administrations, ne souhaitent pas les voir tomber entre de mauvaises mains, au premier rang desquelles celles de leurs concurrents... Nous avons imaginé ici un scénario autour de cette situation.

Scénario

En tant que garant du respect de la concurrence et la bonne information du consommateur la DGCCRF s'est intéressée aux différences de qualité et d'accessibilité des services de télévision par DSL et câble. Afin d'arbitrer de manière éventuelle en faveur d'une meilleure information du consommateur, la DGCCRF demande aux opérateurs télécom de lui remettre des informations concernant leur politique d'encodage et de correction des erreurs des flux TV afin de les étudier. Les départements des relations publiques respectifs des opérateurs vont donc transmettre ces informations aux hauts-fonctionnaires du bureau des médias, des télécommunications, des biens et services culturels. Après un travail interne, la DGCCRF communiquera avec les opérateurs pour faire avancer le dossier.

Etapes du cas d'usage

1. Suite à la demande de la DGCCRF, un responsable des relations publiques de chez Orange souhaite envoyer de manière sécurisée les informations requises à l'un des hauts-fonctionnaires du bureau 6B.

⁷ Direction Générale de la Compétitivité, de l'Industrie et des services

⁸ Direction Générale de la Concurrence, de la Consommation, et de la Répression des Fraudes.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 47

2. Pour cela, il envoie un email à l'adresse « *bureau6B@dgccrf.gouv.fr* » qui est l'adresse de l'assistante du chef du bureau. Comme ces informations sont critiques, il souhaite que seul quelqu'un de haut placé puisse y avoir accès. Il « protège »⁹ donc son message avec le rôle « haut-fonctionnaire du bureau 6B ».
3. L'assistante ne peut pas déchiffrer le message, car elle ne dispose pas du rôle adéquat. Elle peut néanmoins lire l'entête de l'email, indiquant de quoi il s'agit. Elle demande alors au chef de bureau quel est le haut-fonctionnaire en charge de ce dossier et lui transmet ensuite le message.
4. Matthieu D. possède le rôle « haut-fonctionnaire du Bureau 6B », il ouvre donc avec succès le message et sa pièce jointe. Un accusé de réception est envoyé au responsable de chez Orange qui connaît désormais son interlocuteur pour la suite des événements...

Remarque : nous avons ici décrit ce scénario avec l'utilisation classique des emails. Nous aurions pu, tout comme pour le scénario précédent, avoir recours à un service de type *cloud*. L'intérêt est justement de faire comprendre aux lecteurs que la solution développée, devra, et pourra s'intégrer à différentes technologies et applications. En effet, on peut ici tout à fait imaginer un *cloud* servant aux relations entre l'Etat et les opérateurs, de même que l'on pourrait imaginer un *cloud* commun entre les agences gouvernementales et EADS par exemple. Nous reparlerons de tout cela lorsque nous aborderons le concept de PKG (« *Private Key Generator* ») dans l'implémentation des algorithmes d'IBE pour notre solution.

6.2.3. Pour le travail collaboratif et le partage

Dans les entreprises, les employés sont amenés de plus en plus à travailler de sur des espaces collaboratifs en ligne. Il peut s'agir d'échanges de fichiers grâce à un espace de stockage sur le *cloud* comme *Dropbox* ou *Skydrive*. Il peut également s'agir d'une interface de travail commune permettant de modifier en temps réel et de manière simultanée divers documents comme Microsoft SharePoint ou Google Documents. Le scénario suivant se base sur ce contexte.

Scénario

Dans une PME, un espace de stockage sur le *cloud* est mis en place pour faciliter les échanges de fichiers entre les 95 employés. L'entreprise cherche à nouer un nouveau contrat avec un grand groupe français qui souhaite sous-traiter certains éléments de sa production. L'équipe dirigeante de la PME souhaite travailler efficacement sur sa stratégie de communication pour gagner l'appel d'offre. La plupart des membres de l'équipe sont souvent en déplacement, ils doivent donc pouvoir travailler à distance et de manière parfaitement sécurisée. Les informations qui sont échangées par les membres de l'équipe dirigeante sont en effet particulièrement critiques.

⁹ Nous verrons ce que cela signifie dans la section suivante.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 48

Étapes du cas d'usage

1. Le PDG souhaite partager un premier document préparatoire aux autres membres de l'équipe de direction uniquement, afin de réduire au maximum les risques de diffusion.
2. Il va donc placer ce document sur le *cloud* de son entreprise en le protégeant préalablement de manière à ce que seul les « membre de l'équipe de direction » puissent y avoir accès.
3. Un stagiaire un peu trop curieux fouille l'espace de stockage de l'entreprise. Même s'il réussit à accéder au document en question, il ne pourra pas le lire, car il n'est pas « membre de l'équipe de direction ».
4. Le directeur de la technologie souhaite rajouter certains détails techniques sur le document partagé par le PDG. Puisqu'il est bien « membre de l'équipe de direction » il va pouvoir télécharger, modifier, pour uploader le fichier.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 49

7. Réponses aux besoins

7.1. Réponses apportées par les technologies existantes

Voyons comment les solutions actuelles, à base cryptographie asymétrique par PKI, arrivent, ou non, à répondre aux besoins définis précédemment. Pour cela, reprenons la liste alors établie point par point :

- **Sécurité et confidentialité du système de bout en bout**

La cryptographie asymétrique est une solution adaptée ayant fait ses preuves depuis longtemps concernant la sécurisation des données sensibles. Nous avons rappelé dans le premier chapitre son mode de fonctionnement, basée jusqu'à aujourd'hui sur une infrastructure à clé publiques (ou PKI) laquelle a pour rôle de prendre en charge la gestion des clés et des certificats. Chaque clé publique, traditionnellement liée à une personne qui l'a générée, doit en effet être certifiée avant d'être utilisée. Quoi qu'il en soit, cette solution répond parfaitement au besoin de sécurité exprimé dans la section précédente, elle est d'ailleurs toujours utilisée aujourd'hui pour sécuriser les transactions en ligne. Nous ne reviendrons pas ici sur les algorithmes utilisés classiquement pour faire de la cryptographie asymétrique, comme l'algorithme RSA.

- **Flexibilité et praticité des contraintes de déchiffrement (utilisation des rôles)**

Même si, comme nous l'avons rappelé au chapitre 1, la cryptographie asymétrique à base de PKI nécessite une administration complexe du fait de la présence de certificats et de lourdes bases de données pour les stocker, celle-ci reste toutefois relativement adaptée lorsque les critères de protection des données sont liés à la seule identité des individus. En effet, une clé publique est générée pour être utilisée par un et un seul individu. Ainsi, quand l'on souhaite envoyer des données confidentielles à quelqu'un, on chiffre ses données avec sa clé publique certifiée. Ainsi, on est sûr que seule la personne à qui appartient cette clé pourra effectivement les déchiffrer.

A contrario, nous avons vu dans la section précédente, et en particulier lors des *scenarii*, que bien souvent celui qui souhaite transmettre des données ne connaît pas l'identité de son destinataire. Il peut ne connaître que le rôle (la fonction) de celui-ci à l'intérieur d'un cadre

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 50

professionnel donné, voir même seulement un « attribut » générique sur la (ou les) personne(s) destinatrice(s). Or, la cryptographie asymétrique reposant sur une PKI, associe un certificat à une *identité*. Si l'on souhaite contraindre l'accès à des données non plus en fonction d'une ou plusieurs identité(s), mais en fonction d'autres aspects comme l'adéquation à des politiques de droits d'accès prédéfinies par l'expéditeur (comme les *rôles* par exemple) alors l'utilisation de certificats n'est plus possible telle qu'elle.

Pour pouvoir répondre à ce besoin, on a d'autres choix que de complexifier davantage le système en prenant une voie détournée. Un composant supplémentaire doit être implémenté en plus de la PKI et de son autorité de certification pour pouvoir interpréter la politique de droit d'accès spécifiée par l'expéditeur de manière à autoriser, ou non, les utilisateurs à déchiffrer les données. Ce nouveau composant devra être un élément de confiance du système.

Pour chiffrer un message, l'expéditeur devra envoyer un « *package* » composé de deux sous éléments :

- Les données confidentielles chiffrées grâce à une clé symétrique.
- La clé symétrique plus les politiques de divulgation chiffrées avec la clé publique du composant supplémentaire de confiance.

Pour déchiffrer les données, le destinataire devra d'abord communiquer avec le composant de confiance pour s'y authentifier. Une fois cette authentification effectuée, le composant de confiance devra déchiffrer les politiques de divulgation et vérifier qu'elles sont effectivement satisfaites. Le cas échéant, il enverra à l'utilisateur la clé symétrique de déchiffrement. Or, nous avons vu au chapitre premier que les clés symétriques devaient absolument être à usage unique pour des raisons de sécurité. Ainsi, les utilisateurs d'un tel service devront communiquer avec ce composant de confiance à chaque nouveau jeu de données auquel ils veulent accéder.

Par ailleurs, il est important de remarquer que dans cette approche les politiques de droits d'accès sont des entités passives, dans le sens justement où elles ne servent pas à protéger *activement* les informations en en garantissant la confidentialité. Ici c'est la clé publique du composant de confiance additionnel qui sert véritablement à protéger les données.

En outre, il faudra nécessairement implémenter des mécanismes supplémentaires pour faire le lien entre les certificats de la PKI et l'interprétation des politiques de divulgations par le composant supplémentaire de confiance.

Les systèmes à PKI, déjà suffisamment complexes en temps normal, se révèlent donc être encore moins flexibles quand il s'agit de mettre en place une politique de droit d'accès basée sur les rôles des individus. Ainsi, sauf à vouloir faire un effort de développement très important à partir de l'existant, et ce, pour déboucher sur un système encore plus complexe, il semble évident qu'une telle solution ne sera pas des plus adaptées...

▪ **Simplicité d'administration du système**

Les systèmes cryptographiques reposant sur des infrastructures à clé publique sont complexes à administrer. Nous en avons déjà beaucoup parlé au chapitre précédent. Il faut stocker les clés publiques en ligne pour pouvoir les rendre disponibles aux utilisateurs, ce qui requiert d'importantes bases de données. Il faut également signer les demandes de certificat et établir des listes de révocation pour ces derniers.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 51

Par ailleurs, le besoin précédent de flexibilité à travers l'utilisation du RBAC, complexifie encore d'avantage le système qui en devient encore plus difficilement administrable.

▪ **Facilité d'utilisation pour l'utilisateur final**

Aujourd'hui, pour chiffrer des données avec l'une des solutions existantes à bases de PKI, il faut d'abord systématiquement vérifier en ligne que le ou les destinataire(s) possèdent bien une clé publique certifiée. Ce n'est qu'une fois ces clés obtenues que l'expéditeur pourra chiffrer ses données. Outre le fait de nécessiter des requêtes réseaux récurrentes, cela signifie également qu'un utilisateur ne peut chiffrer un message à n'importe qui.

Justement, pour pouvoir être le destinataire de données chiffrées, un utilisateur devra d'abord générer un couple de clés privées/publiques, faire certifier sa clé publique et la transmettre par lui-même ou par l'intermédiaire de la PKI aux futurs expéditeurs.

Pour toutes ces raisons, les systèmes cryptographiques actuels peuvent apparaître à l'utilisateur comme difficiles à utiliser et obscurs dans leur fonctionnement.

▪ **Robustesse à la montée en charge**

La gestion du RBAC avec solutions à PKI nécessite l'utilisation d'un composant supplémentaire de confiance. Les utilisateurs devront communiquer avec ce composant de confiance à chaque nouveau jeu de données auquel ils veulent accéder. Ainsi, en cas de montée en charge du service, c'est-à-dire quand le nombre de demandes de déchiffrement augmentera subitement, le nombre de connexions réseaux au composant supplémentaire en fera de même. Si la demande augmente au-delà d'un seuil critique, le système risque d'être submergé, de perdre en performance, voire de s'interrompre. Dans des contextes critiques comme celui d'un service d'urgences hospitalières, de tels risques ne sont pas supportables.

▪ **Evolutivité**

Avec un système à PKI, une entreprise souhaitant étendre son service de chiffrement généralisé devra pour chacun des nouveaux employés concernés certifier sa clé publique (en l'ayant éventuellement généré) et la stocker sur une base de données en ligne. Si l'espace de stockage n'est plus suffisant, il faudra d'ailleurs acheter de nouvelles bases de données.

Par ailleurs, du fait d'une demande moyenne plus importante et de risques de pics de demandes plus aigües (cf. paragraphe précédent) il sera parfois nécessaire de se doter de nouveaux serveurs pour résister à la montée en charge.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 52

7.2. Réponses apportés par l'IBE

Notre solution, que nous décrirons en détails dans la partie suivante, repose sur l'algorithme de chiffrement basée sur l'identité de Franklin et Boneh (3) que nous décrivions dans le chapitre 1. Sans revenir sur les détails de cet algorithme, rappelons que la propriété fondamentale de celui-ci est de permettre à un utilisateur de chiffrer des données avec n'importe quelle chaîne de caractères. Dans le chapitre 1, nous avons proposé par exemple d'utiliser l'adresse email de l'utilisateur pour lui envoyer des emails chiffrés. Ici, nous allons utiliser le rôle du (ou des) destinataire(s) comme chaîne de caractère pour le chiffrement. Ainsi, nous pourrions tout autant parler de RBE (« *Role-Based Encryption* ») que d'IBE (« *Identity-Based Encryption* »).

Voyons maintenant comment cela se traduit en termes de réponse aux besoins exprimés plus haut.

▪ Sécurité et confidentialité du système de bout en bout

La sécurité de l'IBC repose sur les courbes elliptiques et les couplages. Nous avons vu au chapitre précédent qu'à nombre de bits constant, les courbes elliptiques utilisées par les algorithmes de l'IBC étaient plus sûrs que les corps finis des algorithmes traditionnels type RSA à condition que leur paramètre de sécurité k (ou *embedding-degree*) soit suffisamment grand. Vu autrement sur un exemple, le niveau de sécurité d'une courbe elliptique de seulement 256 bits (avec un paramètre de sécurité $k = 10$, cf. page 23), est équivalent à celui d'un corps fini de 2560 bits.

Ainsi, dans l'état actuel des connaissances, la sécurité des algorithmes d'IBC, et en particulier celui de l'IBE, n'est théoriquement pas moins bonne que celui des algorithmes type RSA, reposant sur l'utilisation d'une PKI. On peut même dire qu'à niveau de sécurité égal, on peut se permettre avec l'IBC d'avoir des espaces et donc des clés aléatoires plus petites.

▪ Flexibilité et praticité des contraintes de déchiffrement (utilisation des rôles)

L'IBE est par définition parfaitement flexible puisqu'elle permet de chiffrer des données avec n'importe quelle politique de divulgation et ce de manière *active*. Replaçons-nous un instant dans un contexte médical. Si un urgentiste souhaite envoyer un patient vers le service de médecine interne en transmettant de manière sécurisée les conclusions des examens préliminaires, il n'aura qu'à chiffrer les données avec la chaîne de caractère « Interniste ». On peut évidemment aller plus loin dans la flexibilité et la précision avec par exemple la chaîne « Interniste + Hôpital Cochin ».

Ainsi, la faculté de pouvoir chiffrer avec n'importe quelle politique de divulgation, notamment avec des rôles, est déjà présente dans l'IBE. Le chiffrement par les rôles, intrinsèquement possible avec l'IBE, est par ailleurs *actif* : contrairement aux solutions à base de PKI, où la politique de divulgation n'est qu'une métadonnée supplémentaire, elle est ici utilisée *activement* comme clé publique pour protéger les données.

Par ailleurs, nous verrons plus loin dans le rapport, que du fait de la structure de notre solution basée sur l'IBE, il n'est plus nécessaire de communiquer avec le PKI (ou PKG) à chaque fois que l'on souhaite déchiffrer des données. Un praticien ayant déjà une clé de déchiffrement correspondant à son rôle « Interniste + Hôpital Cochin » pourra déchiffrer tous les données et

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 53

messages futurs chiffrés avec cette même chaîne de caractère, et ce, sans avoir besoin de communiquer avec qui que ce soit.

▪ **Simplicité d'administration du système**

Si la chaîne de caractères servant à protéger les données peut être vue par analogie avec la cryptographie à PKI comme une clé publique, elle n'en partage pas les inconvénients systémiques. En particulier avec une solution de chiffrement à base d'IBE (comme celle que nous présenterons en détails dans la partie suivante du rapport) il n'est plus nécessaire de stocker les clés publiques pour permettre aux utilisateurs de chiffrer les données. De même, nous avons vu que la notion de certificat n'existe plus. Ainsi, d'une part les expéditeurs de données n'ont plus besoin de vérifier en ligne la validité d'une clé publique, mais en plus cela permet de supprimer les serveurs servant à stocker et administrer les certificats.

Par ailleurs, nous avons vu qu'il était possible avec l'IBE de chiffrer des données avec n'importe quelle chaîne de caractères. Or, un élément très important vis-à-vis de la protection des données en environnement complexe et dynamique est de pouvoir modifier les droits d'accès des utilisateurs. Par exemple, un pédiatre de l'hôpital Necker peut être muté à la Pitié Salpêtrière. Ce pédiatre aura toujours sur sa machine personnelle la clé de déchiffrement « pédiatre + hôpital Necker ». Or, on peut souhaiter légitimement que celui-ci ne puisse plus lire à l'avenir les dossiers des futures patients. L'idée est donc de concaténer automatiquement (c'est-à-dire sans l'aval de l'utilisateur) la date au moment du chiffrement. Par exemple, si un chirurgien souhaite chiffrer des données avec la chaîne « pédiatre + hôpital Necker », la solution concatènera automatique la date du jour et finalement les données seront chiffrées avec « pédiatre + hôpital Necker + 31/10/2013 ». Du côté du déchiffrement, si un praticien ne dispose pas de la clé de déchiffrement correspondante, il devra en faire la demande au PKG. Ce dernier concatènera la date du jour en plus du rôle de l'utilisateur si celui-ci répond effectivement aux conditions de divulgation (c'est-à-dire si celui-ci est bien en l'occurrence pédiatre à l'hôpital Necker, etc.). Dans le cas inverse, l'individu ne pourra pas obtenir sa nouvelle clé. Ainsi, si notre médecin pédiatre est muté à la Pitié Salpêtrière le 31 octobre 2013, il ne pourra obtenir la clé de déchiffrement « pédiatre + hôpital Necker + 01/11/2013 » ni celles à des dates ultérieures... Cette fonctionnalité, équivalente à la révocation des certificats des PKI, a l'avantage de ne plus nécessiter de serveur pour administrer la révocation des certificats.

En outre, nous n'avons pas encore abordé le problème de la récupération de données chiffrées en cas de perte de clé par les utilisateurs. Avec les solutions actuelles à base de PKI, afin de pouvoir déchiffrer des données même en cas de besoin (perte de la clé déchiffrement, sinistre, etc.), on stocke les clés privées des utilisateurs dans d'importantes bases de données sécurisées (ou serveurs de récupération). Avec les solutions à base d'IBE, cela n'est plus nécessaire puisque le PKG (« *Private Key Generator* » - cf. chapitre 1) peut générer les clés de déchiffrement sur demande et « à la volée ».

▪ **Facilité d'utilisation pour l'utilisateur final**

Contrairement aux systèmes utilisant une PKI, l'utilisateur n'a plus besoin de vérifier en ligne que le (ou les) destinataire(s) possède(nt) bien une clé publique pour chiffrer des données. Il n'y a plus

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 54

aucune connexion réseau nécessaire. L'utilisateur pourra chiffrer ses données quoi qu'il arrive, même si son destinataire ne dispose pas encore du moindre rôle dans l'entreprise. Cela permet en outre de pouvoir envoyer des données chiffrées « dans le futur ». Un individu peut par exemple chiffrer un fichier avec la chaîne de caractère « infirmière + 05/12/2016 ».

De même, avec les solutions actuelles, pour pouvoir être le destinataire de données chiffrées, un utilisateur devra d'abord générer un couple de clés privée/publique, faire certifier sa clé publique et la transmettre par lui-même ou par l'intermédiaire de la PKI aux futurs expéditeurs. Avec un système basé sur l'IBE, l'utilisateur n'a aucune action à entreprendre.

Enfin, l'utilisateur n'a plus besoin de s'authentifier auprès de l'autorité de certification à chaque nouvelle demande de déchiffrement. Il en a éventuellement besoin une fois par jour si une date de péremption est incluse dans les clés ou s'il change de fonction ou de poste au sein de l'entreprise.

▪ Robustesse à la montée en charge

Comme nous venons de le voir, les requêtes des utilisateurs auprès du PKG sont beaucoup plus rares qu'avec une PKI. Le système sera donc bien mieux à même de résister à une montée en charge. En outre, le temps de calcul nécessaire à la génération des clés privées est d'autant limité que celles-ci sont générées uniquement sur demande.

Par ailleurs, l'absence de serveur de révocation des certificats allège considérablement la charge qui pèse sur le système. Le PKG n'a pas besoin de vérifier, supprimer ou révoquer quoi que ce soit, il ne fait que générer et envoyer les clés de déchiffrement correspondant aux rôles liés à l'identité de l'utilisateur.

▪ Evolutivité

Le service de chiffrement à base d'IBE peut être généralisé extrêmement facilement. Mis à part le fait d'enregistrer les nouveaux utilisateurs dans la liste d'authentification (et de spécifier leur rôle respectif si besoin) aucune action n'est nécessaire. Il n'y a rien à générer, rien à stocker, rien à certifier.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 55

8. Implémentation de notre solution basée sur l'IBE

Dans cette partie, nous allons décrire plus en détails la solution introduite précédemment reposant sur la propriété de chiffrement par une chaîne de caractère au choix de l'utilisateur faisant office de politique de divulgation. Nous monterons d'abord comment notre solution, construite sous forme de différents modules indépendants, peut être utilisée dans des applications et des contextes d'usages très variés. Puis, à partir de cette architecture générale, nous décrirons plus en détails chacun des modules fonctionnels en explicitant notamment leur rôle, leurs connexions avec les autres modules, et évidemment la manière dont nous les avons implémentés concrètement.

8.1. Vue d'ensemble

L'objectif de notre implémentation était de pouvoir s'adapter à n'importe quel contexte applicatif (chiffrement des emails, sécurisation d'un *cloud*, etc.). Notre solution se devait donc d'être particulièrement flexible. Pour cette raison, nous avons choisi de « découper » notre solution en modules (ou « briques ») indépendants et interconnectables en fonction des besoins spécifiques et du contexte applicatif. Par ailleurs, cette modularité avait également pour objectif de rendre la modification et l'amélioration de notre solution très facile pour des évolutions futures. Par exemple, le changement de courbe elliptique pour les couplages ou la modification de la méthode d'authentification des utilisateurs n'oblige à intervenir que localement dans un seul des modules.

Nous avons tout d'abord développé deux briques que l'on pourrait qualifier de « bas niveau » car indispensables à toutes les applications. Ces deux briques, sont les suivantes :

- **IBEncrypt** : Implémentation des phases de chiffrement et déchiffrement de l'algorithme d'IBE étudié au chapitre 1 (cf. page 32).
- **PKG** : Implémentation d'un PKG (« *Private Key Generator* ») capable, en particulier, de générer une clé privée à partir d'une chaîne de caractère.

Un deuxième cercle de modules correspond aux fonctionnalités additionnelles, non indispensables à toutes les applications. Dans le cadre des 6 mois de travail auxquels fait suite ce rapport, ce cercle n'est composé pour le moment que d'un seul module, mais s'étoffera au cours du temps :

- **Un module de gestion des rôles et des identités des utilisateurs** : ce module, déjà existant en interne chez Cassidian Cybersecurity, a été intégré à notre solution après un travail de collaboration avec l'équipe de développement.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 56

Enfin, le troisième cercle correspond aux modules d'interfaçages. Il peut s'agir d'un module d'intégration à un produit déjà existant ou encore d'un module de dialogue avec une autre solution ou un autre composant. Ces modules concernent tout particulièrement l'utilisateur final car c'est le plus souvent ce module que celui-ci est amené à utiliser. Au jour d'aujourd'hui, nous avons développé un module d'interfaçage :

- **IBE Outlook Add-In** : Surcouche compatible avec Outlook 2010 et 2013 permettant d'envoyer des emails chiffrés en utilisant le rôle ou l'identité du destinataire. Cette *Add-In* a été personnalisée pour être utilisée dans un contexte médical semblable à celui du scénario décrit plus haut (cf. page 46).

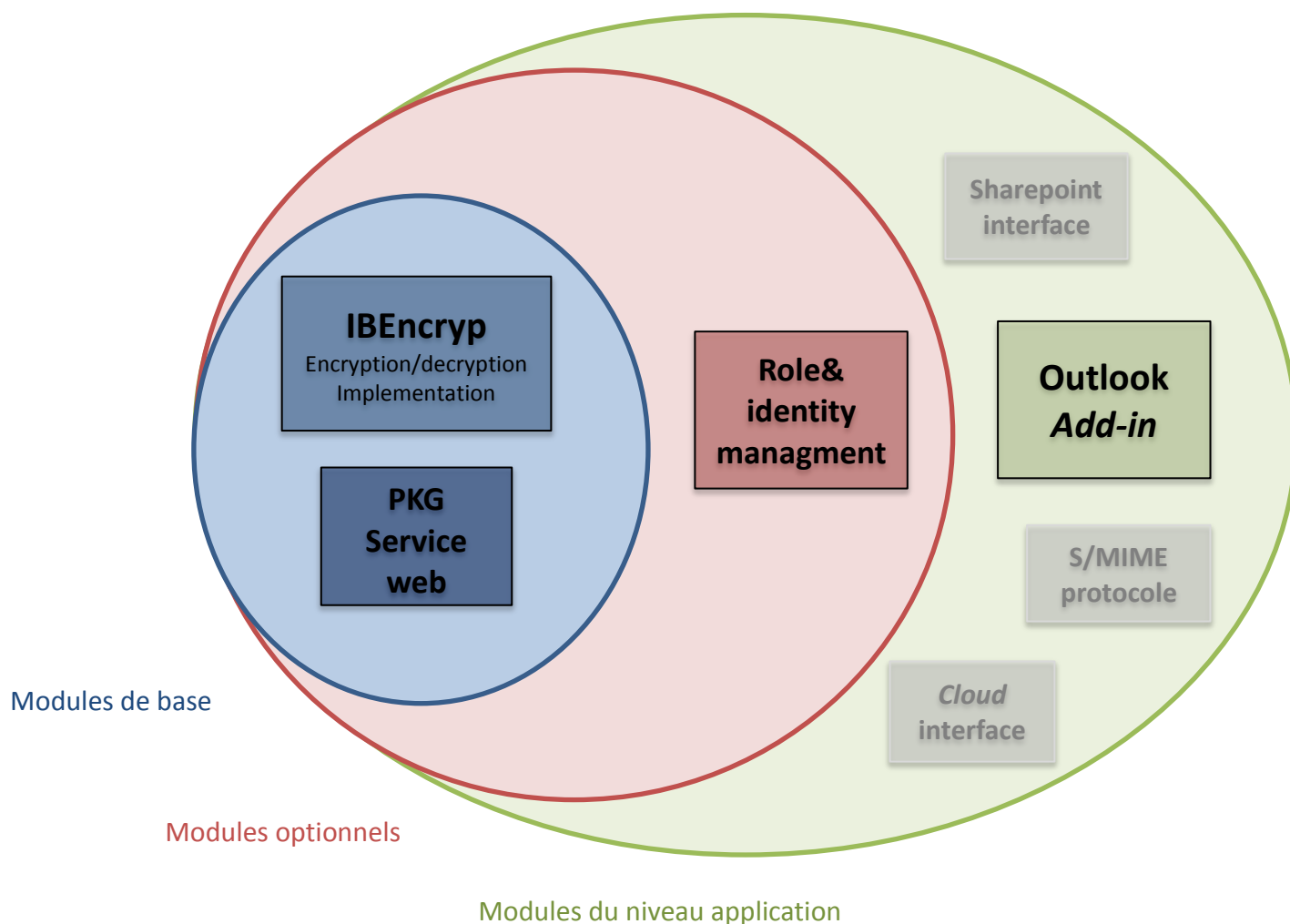


Figure 4 : La structure modulaire de notre solution

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 57

Sur la figure ci-dessus, les modules en gris sont des modules en cours de développement ou à l'étude. Par exemple, le module d'interfaçage avec un espace de type « *cloud* » permettra de pouvoir chiffrer « à la volée » les documents en demande de transfert vers le nuage. La brique « S/MIME » consiste à utiliser le protocole du même nom pour rendre le chiffrement et déchiffrement totalement transparent au contexte utilisateur, permettant ainsi de pouvoir étendre notre solution à l'ensemble des clients emails, sans travail spécifique supplémentaire. Nous reviendrons plus en détails sur les perspectives futures à la fin de ce rapport.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 58

8.2. IBEncrypt

8.2.1. Rôles et missions

Ce module correspond à l'implémentation des algorithmes de chiffrement et déchiffrement de l'IBE. Il a donc en particulier pour missions de :

- Chiffrer des données de tous types en utilisant n'importe quelle chaîne de caractère.
- Déchiffrer des données grâce à la clé de déchiffrement correspondant à la chaîne de caractère utilisée pour le chiffrement.

8.2.2. Liens avec les autres modules

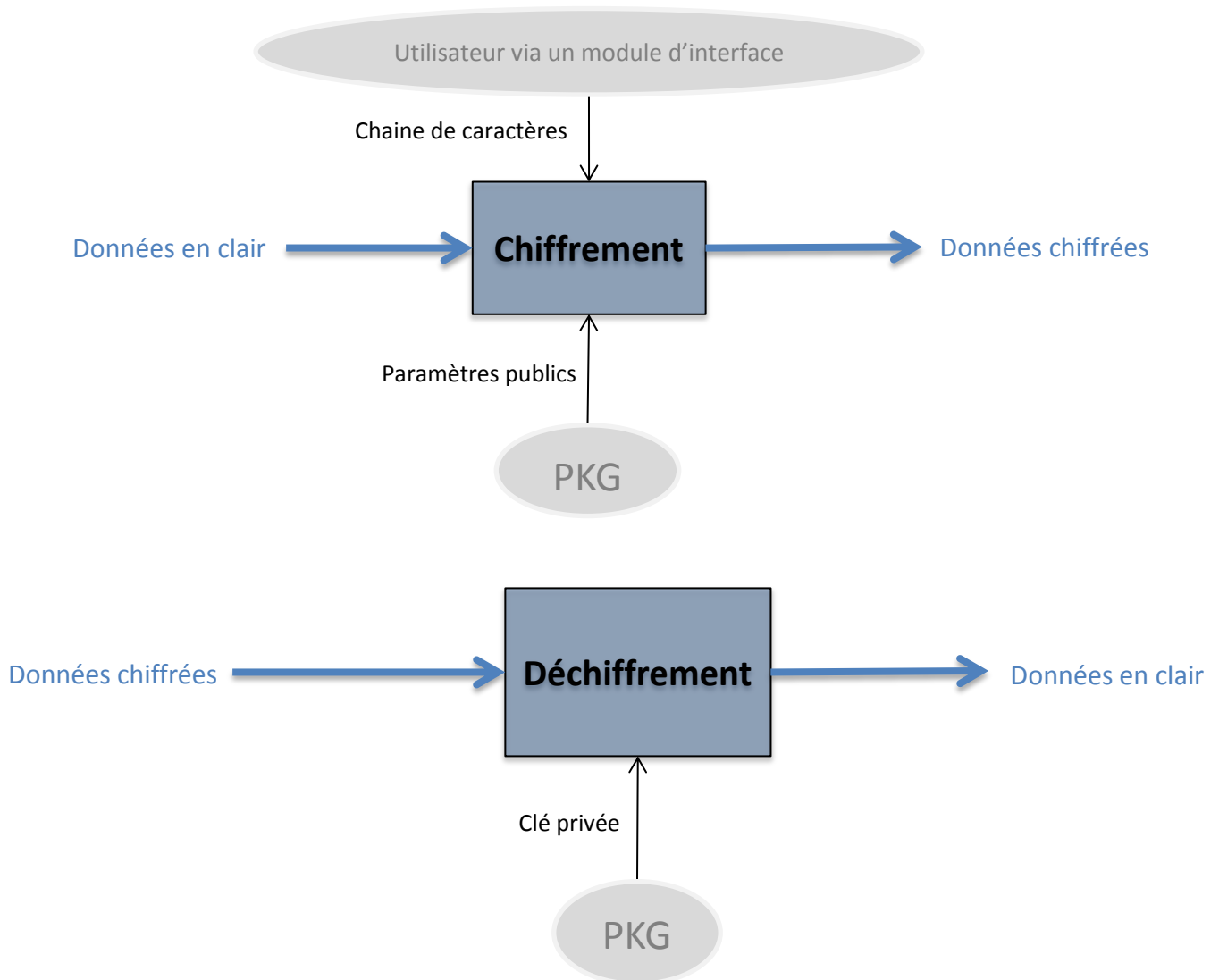
Comme nous l'avons vu au chapitre précédent, la phase de chiffrement de l'algorithme d'IBE requiert deux valeurs d'entrées que sont premièrement une chaîne de caractères arbitraire (une identité ou un rôle par exemple) et deuxièmement les paramètres publics générés en amont par le PKG. Ces paramètres peuvent être soit téléchargés du PKG vers le terminal de l'utilisateur à la première utilisation du service, soit être directement inclus dans les modules de la solution qui doivent être installés sur les terminaux utilisateurs. Rappelons que ces paramètres publics ne sont en réalité qu'un moyen d'identifier le PKG avec lequel on veut communiquer, ce qui devient indispensable lorsqu'il en existe plusieurs. Concernant le choix par l'utilisateur de chaînes de caractère servant au chiffrement, celui-ci est souvent fait au niveau des modules de la couche application (module en vert sur la Figure 4).

Le déchiffrement quant à lui requiert uniquement une clé de déchiffrement (ou clé privée). Cette clé est générée puis distribuée par le PKG. Nous y reviendrons dans la section suivante lorsque nous aborderons l'implémentation du PKG.

Ce module peut lui-même être utilisé par d'autres. Une fonction du niveau « application » pourra faire appel aux fonctions de chiffrement et de déchiffrement ici implémentées. Par exemple, une application de chiffrement des emails n'aura qu'à charger ce module sous forme de librairie et importer les fonctions souhaitées.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 59

8.2.3. Diagramme fonctionnel



8.2.4. Implémentation

Par rapport à l'algorithme d'IBE de base, où les données sensibles sont directement chiffrées avec la chaîne de caractères choisie par l'utilisateur, nous chiffrons d'abord les données avec une clé symétrique traditionnelle, puis nous chiffrons cette clé avec la chaîne de caractères. L'objectif est d'augmenter les performances du système. Nous avons vu en effet au chapitre précédent que les couplages sont gourmands en temps de calcul. Ainsi, si les données à chiffrer sont volumineuses, un chiffrement direct prendrait trop de temps. On préfère donc chiffrer par IBE une clé de session courte puis profiter de la rapidité du chiffrement symétrique (cf. page 11) pour chiffrer les données avec cette clé de session. Notons d'ailleurs qu'il n'y a rien de novateur à cela, les implémentations de chiffrement traditionnel type RSA ont toujours fait de même.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 60

Les algorithmes de chiffrement et de déchiffrement ont été codés en langage C afin d'obtenir des performances optimales. Nous avons également utilisé la librairie cryptographique MIRACL pour les calculs de couplage sur les courbes elliptiques. Cette bibliothèque *open-source* est aujourd'hui mondialement utilisée par la communauté des cryptographes et des chercheurs. Elle est par ailleurs très régulièrement mise à jour, ce qui permet de profiter des dernières nouveautés concernant la sécurité des courbes elliptiques. Par ailleurs, nous avons également utilisé la bibliothèque Microsoft « Security.Cryptography » pour la génération des clés symétriques aléatoires.

A ce sujet, nous avons pour le moment fait le choix d'utiliser une courbe dont le paramètre de sécurité k est égal à 10, ce qui correspond à un niveau de sécurité analogue à celui de l'AES 192 bits. Mais, ce choix de la courbe peut être modifié extrêmement facilement en changeant les fichiers de ressources appelé par le module. Aucune modification du code n'est nécessaire mise à part le changement de valeur d'une macro spécifiant la courbe à utiliser dans la librairie MIRACL. Cela permettra de pouvoir élever le niveau de sécurité à l'avenir si besoin (par exemple dans le cas où une attaque sur la courbe était découverte). On pourrait même donner la liberté du choix du niveau de sécurité à l'utilisateur final sans beaucoup de travail supplémentaire.

Les deux fonctions implémentées (chiffrement et déchiffrement) sont compilées sous forme de DLL afin de pouvoir être utilisées par d'autres modules de la solution, en particulier les modules d'interface avec d'autres composants (client email, *cloud*, outil de travail collaboratif, *etc.*).

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 61

8.3. Le PKG

8.3.1. Rôles et missions

Le PKG a le rôle de « gendarme » du déchiffrement. C'est lui qui vérifie que les politiques de divulgation choisies par l'expéditeur des données sont bien vérifiées par les utilisateurs qui souhaitent y accéder. Le PKG a quatre missions principales :

- L'authentification des utilisateurs. L'authentification consiste à vérifier que la personne qui fait la requête d'une clé de déchiffrement est bien la personne qu'elle prétend être.
- Dans certaines applications, il doit aussi s'occuper de la communication avec des modules optionnels. Dans notre solution actuelle, le PKG communique avec le module de gestion des rôles et des identités pour récupérer le (ou les) rôle(s) des utilisateurs.
- La génération des clés privées à partir d'une chaîne de caractères (dans notre cas l'identité ou le rôle de l'utilisateur authentifié).
- La transmission des clés de déchiffrement : une fois l'utilisateur authentifié, le PKG lui donne accès aux clés de déchiffrement auxquelles il a le droit pour qu'il puisse les stocker en local sur son terminal.

8.3.2. Liens avec les autres modules

Comme nous l'avons déjà expliqué, le PKG communique avec les utilisateurs souhaitant déchiffrer des données dont il n'a pas encore la clé de déchiffrement. Cette communication se fait en pratique à travers un (ou plusieurs) module(s) du niveau applicatif. Les entrées et sorties du PKG sont transmises par et vers ces modules. Par exemple, l'authentification de l'utilisateur se fera directement au niveau du module d'interface utilisateur, puis les signaux seront envoyés en entrée du PKG pour vérification. Il en va de même dans l'autre sens pour les clés privées.

Par ailleurs, selon les cas de figure, le PKG devra également transmettre en amont à ces mêmes modules son identifiant (ou paramètres publics) pour permettre le chiffrement des données.

Le PKG est amené également à être en lien avec d'autres modules optionnels (en rouge sur la Figure 4) afin d'obtenir certaines informations sur l'utilisateur (comme son rôle par exemple) ou de faire d'autres vérifications (par exemple, dans le cas où une date de validité est utilisée pour chiffrer les données, le PKG devra vérifier l'heure et le date lorsqu'un utilisateur fait une requête de clé privée). Aujourd'hui le PKG communique avec le module de gestion des rôles et des identités.

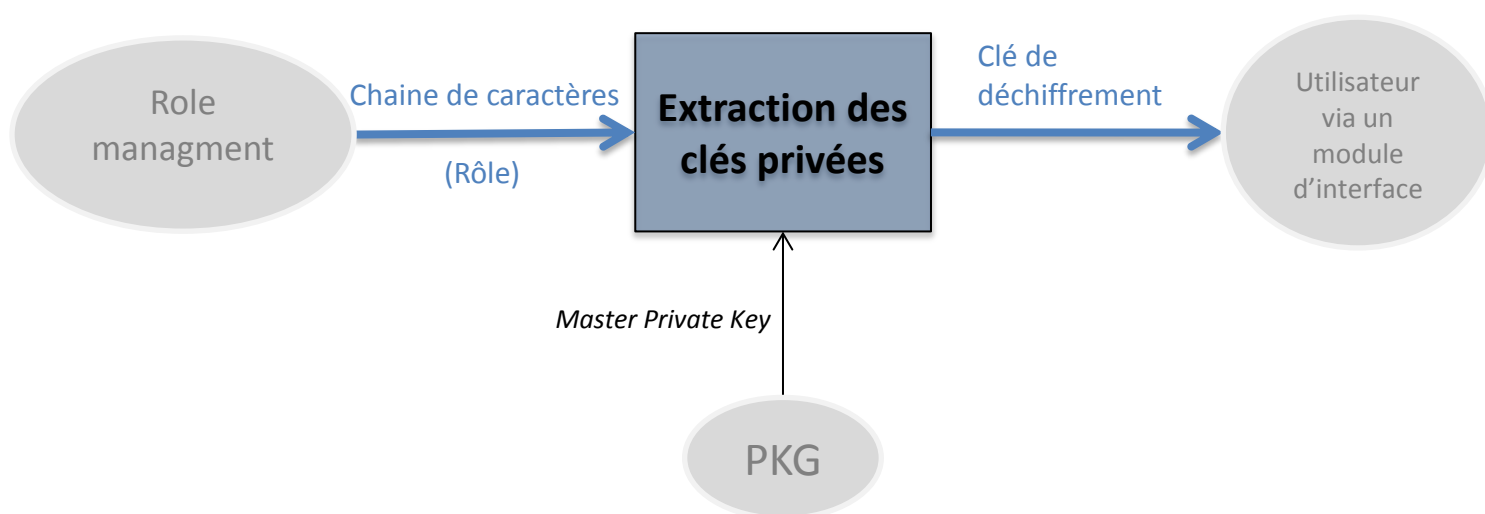
Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 62

8.3.3. Diagramme fonctionnel

Fonction de génération des paramètres publics et de la *Master Private Key* (cf. phase *Init* de l'algorithme d'IBE p. 32) :



Fonction d'extraction des clés privées dans le cas où la politique de divulgation utilisée est le rôle de l'utilisateur :



8.3.4. Implémentation

Le PKG est un serveur hébergé sur une machine physique. Sur ce serveur sont stockés les exécutables des fonctions *initialisation* et *extraction* ainsi que les paramètres publics et la *Master Private Key*.

Les algorithmes d'initialisation et d'extraction ont été implémentés en langage C de la même manière que les algorithmes de chiffrement et de déchiffrement décrits dans la section précédente. Lors de l'installation de notre solution dans une entreprise ou dans un hôpital par exemple, il suffira

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 63

au PKG d'exécuter le programme d'initialisation et générer ainsi la *Master Private Key* et les paramètres publics une bonne fois pour toutes.

Concernant l'authentification des utilisateurs au PKG, celle-ci constitue un sous module indépendant du reste. Ainsi, on peut changer de méthode d'authentification sans pour autant avoir à modifier le reste du code. Pour le moment, nous avons mis en place un système très simple d'identifiant et mot de passe qui sont stockés dans un annuaire utilisant le protocole LDAP¹⁰. A l'avenir, rien n'empêche de changer cette méthode pour un système d'authentification forte (carte à puce, SMS, etc.).

La communication entre le *PKG* et les utilisateurs se fait par l'intermédiaire d'un service web hébergé du côté serveur et développé en ASP.NET. Lorsqu'un utilisateur souhaite s'authentifier avec ses identifiants du côté de l'interface utilisateur, ceux-ci sont transmis directement au service web coté serveur, qui va alors envoyer une requête de vérification vers l'annuaire LDAP.

Si l'authentification se passe bien, l'utilisateur pourra récupérer la (ou les) clé(s) privée(s) dont il est propriétaire légitime. Le service web communiquera alors avec le module de gestion des rôles grâce au protocole LDAP. Si l'utilisateur existe bien dans l'annuaire alors ce dernier renverra au service web du PKG la liste de rôle correspondante. Pour chacune des chaînes de caractères de cette liste, le PKG génère une clé privée « à la volée » en exécutant le programme d'extraction (cf. p. 32 et p. 62). Le service web s'occupera finalement de télécharger les clés privées sur le terminal utilisateur.

¹⁰ *Lightweight Directory Access Protocol*

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 64

8.4. L'Add-In Outlook

8.4.1. Rôles et missions

Nous avons développé un module de niveau applicatif pour Outlook 2010 et 2013 intégrant notre solution basée sur l'IBE pour offrir un service innovant de chiffrement des emails, très simple à utiliser. Si cette *Add-In* peut être utilisée dans n'importe quel contexte (entreprise, administration, etc.) pour sécuriser les communications par emails, nous l'avons personnalisée spécifiquement pour répondre au besoin de confidentialité des données de santé dans les hôpitaux et centres médicaux.

Avertissement : Cette *Add-In* ne doit pas être vue comme un produit fini utilisable tel quel dans le milieu médical, mais comme un cas d'usage fictif ne prenant pas en compte le code de la Santé publique et certaines contraintes réelles du milieu médical (Carte Professionnel de Santé, intégrité des données, non-répudiation, etc.). L'objectif de ce module est seulement d'illustrer, sur un exemple pratique, la valeur ajoutée » de notre solution basée sur l'IBE.

Son rôle est de faciliter les partages d'informations entre les praticiens du domaine de la santé. Nous verrons dans la partie suivante de ce rapport, que ce module, couplé avec ceux déjà décrits plus haut (IBEncrypt, PKG et le module de gestion des rôles et des identités) permet de garantir un haut niveau de sécurité tout en répondant parfaitement aux besoins exprimés dans la partie précédente, et en particulier la facilité d'utilisation et d'administration. Les différentes missions de ce module sont les suivantes :

- Permettre à l'utilisateur de chiffrer facilement ses emails avec la politique de divulgation voulue (identité, rôles, etc.).
- Déchiffrer de manière transparente les emails chiffrés que reçoit le destinataire.
- Communiquer avec le service web du PKG lorsque l'utilisateur a besoin d'obtenir une nouvelle clé privée.

8.4.2. Liens avec les autres modules

En tant que module applicatif, l'*Add-In* Outlook est connecté aux modules fondamentaux de notre solution (PKG et IBEncrypt). Comme nous l'avons déjà expliqué plus haut (cf. p 59 et 62) IBEncrypt servira à chiffrer et déchiffrer les emails dans Outlook. L'*Add-In* Outlook devra en outre communiquer avec le module PKG lorsque une clé de chiffrement sera nécessaire et que l'utilisateur ne la possède pas encore.

8.4.3. Implémentation

Cette *Add-In* Outlook a été développée en C# dans l'environnement .NET. Une *Add-In* Microsoft est une solution qui s'intègre à un produit Microsoft pour en personnaliser l'interface et le

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 65

fonctionnement. Ainsi, si un tel programme permet de modifier l'interface graphique d'Outlook, il permet aussi et surtout d'en modifier le comportement lorsque certains événements surviennent (nouvel email reçu, email envoyé, clic utilisateur sur un bouton de l'interface, etc.) et d'y ajouter de nouvelles fonctionnalités (chiffrement et déchiffrement par exemple). Nous présenterons en détails l'interface graphique et le comportement de notre *Add-In* pour l'utilisateur final sur un scénario complet dans la partie suivante. Pour le moment, contentons-nous de décrire le fonctionnement technique de nos fonctions de chiffrement et de déchiffrement dans Outlook.

Le chiffrement d'un email

Lorsqu'Outlook génère, par action de l'utilisateur ou de manière automatique, l'évènement « *Email Sent* » notre *Add-In* le détecte (cette évènement survient par exemple quand l'utilisateur clique sur le bouton « envoyer »). Dans le cas où l'email doit être chiffré, l'*Add-In* va bloquer l'envoi du message en clair. Il va ensuite donner la main au module IBEncrypt qui chiffrera l'email. Enfin, l'*Add-In* enverra automatiquement le message chiffré au(x) destinataire(s).

Plus précisément, IBEncrypt chiffre l'email à envoyer et l'*Add-In* encapsule l'email chiffré dans un autre email en clair en tant que pièce jointe. C'est cet email qui sera ensuite envoyé au destinataire. En plus des données chiffrées, cet email contient également en pièces jointes l'ensemble des données nécessaires au déchiffrement et en particulier la chaîne de caractère avec laquelle a été chiffré le message¹¹.

A noter que nous chiffons entièrement les emails et non uniquement le corps du message. L'ensemble des champs du message (To, CC, From, corps, pièces jointes, etc.) sont chiffrés comme un tout (un fichier au format « .msg ») par IBEncrypt.

Nous verrons dans la partie suivante (cf. p 68) comment l'utilisateur peut concrètement chiffrer un email à l'aide de l'interface utilisateur que nous avons rajouté à Outlook.

Le déchiffrement d'un email

Pour le déchiffrement, si l'utilisateur a déjà la clé privée adéquate, IBEncrypt déchiffrera l'email et Outlook affichera le message en clair automatiquement. Si l'utilisateur ne possède pas la clé adéquate, mais qu'il pense avoir le droit d'obtenir la clé, du fait de son identité ou de son rôle, l'*Add-In* contactera le Service web hébergé sur le serveur du PKG et lui permettra de s'authentifier directement avec ses identifiants dans Outlook.

Selon les besoins, plusieurs solutions sont possibles pour le déchiffrement. Le déchiffrement peut se faire à la volée quand l'utilisateur reçoit un nouvel email, c'est-à-dire quand Outlook génère l'évènement « *New Email* ». Dans ce cas, et à condition que l'utilisateur possède la clé adéquate, l'*Add-In* détectera l'évènement et déchiffrera directement l'email en faisant appel à IBEncrypt. Puis, il enregistrera l'email en clair dans la boîte de réception. Dans le cas contraire, l'*Add-In* laissera l'email tel qu'il a été envoyé, c'est-à-dire chiffré.

Le déchiffrement peut se faire également quand l'utilisateur double-clique sur un email pour le lire c'est-à-dire quand l'évènement « *Email Open* » survient. La procédure est alors identique : IBEncrypt déchiffre l'email puis l'*Add-In* affiche l'email en clair dans une nouvelle fenêtre.

¹¹ Rappelons qu'en pratique, c'est une clé symétrique qui sert à chiffrer les données confidentielles, elles-mêmes chiffrées par IBE.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 66

L'authentification avec le PKG

L'authentification se fait actuellement par identifiant et mot de passe. Lorsque l'évènement « Email Open » est détecté, si l'utilisateur ne possède pas la clé de déchiffrement, l'*Add-In* demandera à l'utilisateur de lui donner ses identifiants et les transmettra au service web du PKG pour vérification.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 67

9. Interface graphique et résultats

Dans cette ultime section, nous allons présenter, à travers un scénario typique d'usage, l'interface utilisateur et administrateur de notre solution basée sur l'IBE. Nous ferons ensuite le bilan de ses points forts et défauts actuels. Finalement, nous discuterons des améliorations futures et des perspectives à court et moyen terme pour celle-ci.

9.1. Sur un scénario

Le scénario suivant a pour objectif de faire comprendre au lecteur la manière d'utiliser en pratique notre solution. Nous en profiterons à chaque étape importante pour nous arrêter sur les éléments importants de l'interface.

Un jeune pneumologue, le Dr. Bossart, est titularisé au CHU de Rennes. Le système d'information du centre hospitalier est équipé avec notre solution de chiffrement des données...

Etape 1 : ajouter un nouvel utilisateur dans la base de données LDAP

Afin de permettre au Dr. Bossart d'échanger des données chiffrées avec les autres membres du personnel de l'hôpital, l'administrateur système doit tout d'abord l'enregistrer dans la base de données des utilisateurs du système. Il utilise pour cela le module de gestion des rôles.

On commence par créer son compte utilisateur dans le SI du CHU en spécifiant ses informations personnelles (nom, prénom, numéro de téléphone, etc.) :

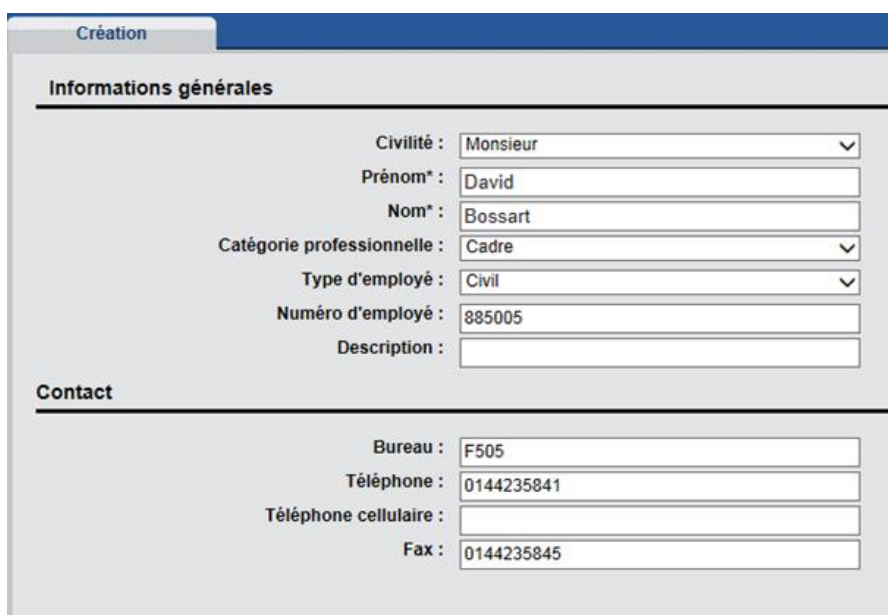


Figure 5 : Enregistrement d'un nouveau profil d'utilisateur dans la base de données LDAP

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 68

On spécifie ensuite les rôles du Dr. Bossart. Sur la figure ci-dessous, on peut voir que le Dr. Bossart possède deux rôles, celui de Médecin et celui de Pneumologue. Cela signifie que le Dr. Bossart pourra obtenir du PKG les deux clés de déchiffrements correspondants à ces deux rôles. Par ailleurs, un utilisateur pourra toujours obtenir la clé privée correspondant à son identité c'est-à-dire à son email dans le système d'information de l'hôpital. Ainsi, le Dr. Bossart aura aussi accès aux emails chiffrés avec la chaîne de caractères « david.bossart@churennnes.fr ».

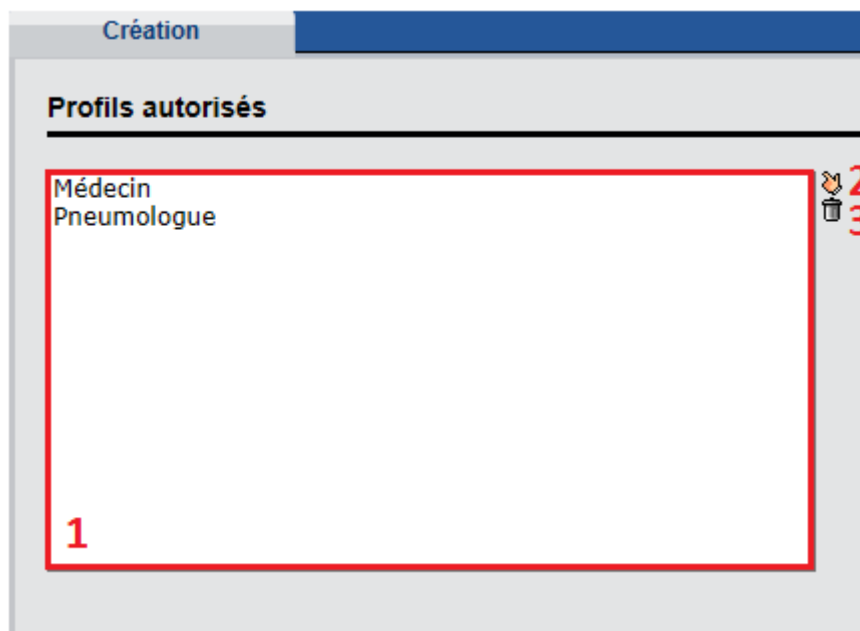


Figure 6 : Ajouter ou supprimer des rôles à un utilisateur.

L'encadré rouge numéroté « 1 » correspond aux rôles déjà enregistré du Dr. Bossart. L'administrateur peut en ajouter d'autres en cliquant sur le bouton numéroté « 2 ». Il peut aussi en supprimer (bouton numéroté « 3 »). Une fois le compte créé, l'administrateur pourra toujours supprimer ou ajouter des rôles de la même manière.

L'identifiant du Dr. Bossart sera « david.bossart » et son mot de passe, initialement « azerty123 » devra être modifié à la première connexion.

Etape 2 : Envoi d'emails chiffrés

Un patient P vient voir le Dr. Bossart dans le département de pneumologie du CHU. Pour confirmer son diagnostic, le Dr. Bossart demande au patient P d'aller faire une radio des poumons. Pour accélérer le processus, il souhaite envoyer le dossier du patient P directement par email au service de radiologie. Il ouvre alors Outlook sur sa machine professionnelle.

La figure ci-après illustre les nouvelles fonctionnalités intégrées à Outlook grâce à notre *Add-In* :

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 69

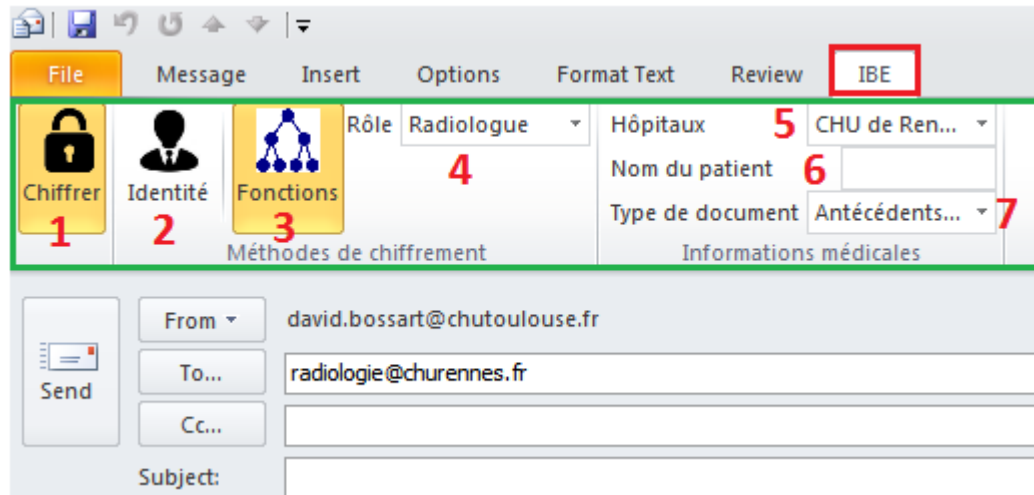


Figure 7 : Interface graphique de chiffrement des emails

Arrêtons-nous un instant sur l'interface de l'*Add-In* Outlook. Comme on peut le voir, nous avons créé un nouvel onglet « IBE » (encadré rouge) en plus de ceux déjà existants par défaut (« Message », « Insertion », *etc.*) dans Outlook 2010. En cliquant sur cet onglet, un ruban apparaît avec différents éléments (encadré vert) :

1. Bouton « Chiffrer » : l'utilisateur choisit ou non d'envoyer son email de manière chiffrée en cliquant sur ce bouton. Il va alors avoir plusieurs possibilités de chiffrement selon s'il clique sur le bouton 2 ou sur le bouton 3.
2. Bouton « Identité » : l'email sera chiffré avec l'identité de l'utilisateur, c'est-à-dire son adresse email. Cette option sera utile si l'utilisateur connaît exactement la personne à qui il veut envoyer des données.
3. Bouton « Fonctions » : l'email sera chiffré avec un rôle.
4. Liste déroulantes des rôles disponibles localement : dans notre exemple il s'agira des différentes fonctions des praticiens et personnels administratifs du CHU (chirurgiens, ophtalmologiste, urgentiste, secrétaire, *etc.*). L'utilisateur n'a pas la possibilité de rajouter lui-même de nouveaux rôles pour le chiffrement.
5. Notre système peut être mutualisé sur plusieurs établissements de Santé. L'utilisateur peut vouloir spécifier la localité du destinataire. Il peut vouloir communiquer uniquement avec des chirurgiens du CHU de Brest par exemple.
6. L'utilisateur peut spécifier l'identité de son patient.
7. L'utilisateur peut préciser de quel type de document médical il s'agit (radio, scanner, analyses, *etc.*) parmi une liste prédéfinie par l'administrateur. L'utilisateur n'a pas la possibilité de rajouter lui-même de nouveaux types de documents médicaux.

Ici, le Dr. Bossart a souhaité chiffrer son message avec le rôle « Radiologue ». Il a précisé qu'il s'agissait d'antécédents médicaux et que l'hôpital était le CHU de Rennes. Il ne lui reste plus qu'à choisir un destinataire (ici la liste de diffusion radiologie@churennnes.fr) et à appuyer, comme il en a l'habitude, sur le bouton « Envoyer ». Son email sera automatiquement chiffré et envoyé avec les options sélectionnées.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 70

Etape 3 : Réception et lecture d'emails chiffrés

Après avoir effectué la radio des poumons, le Dr. Bernstein (le radiologue ayant pris en charge le patient P) veut la transmettre au Dr. Bossart. Comme il connaît son identité, il va chiffrer son message avec l'adresse email du Dr. Bossart. De cette façon il est sûr que seul le Dr. Bossart pourra déchiffrer son message (Il cliquera donc sur le bouton n°2 de la Figure 7).

Le Dr. Bossart reçoit un email chiffré de la part de `claudio.bernstein@churennes.fr` sur sa boîte de messagerie. Comme il ne possède pas encore ses clés privées, lorsque le Dr. Bossart tente d'ouvrir l'email, une boîte de dialogue d'authentification s'ouvre. Le Dr. Bossart rentre alors ses identifiants. Comme nous l'avons expliqué à la section précédente, en cliquant sur OK, l'Add-In Outlook transmet ces informations vers le Service web du PKG qui procédera à la vérification et lui renverra le cas échéant ses clés privées.

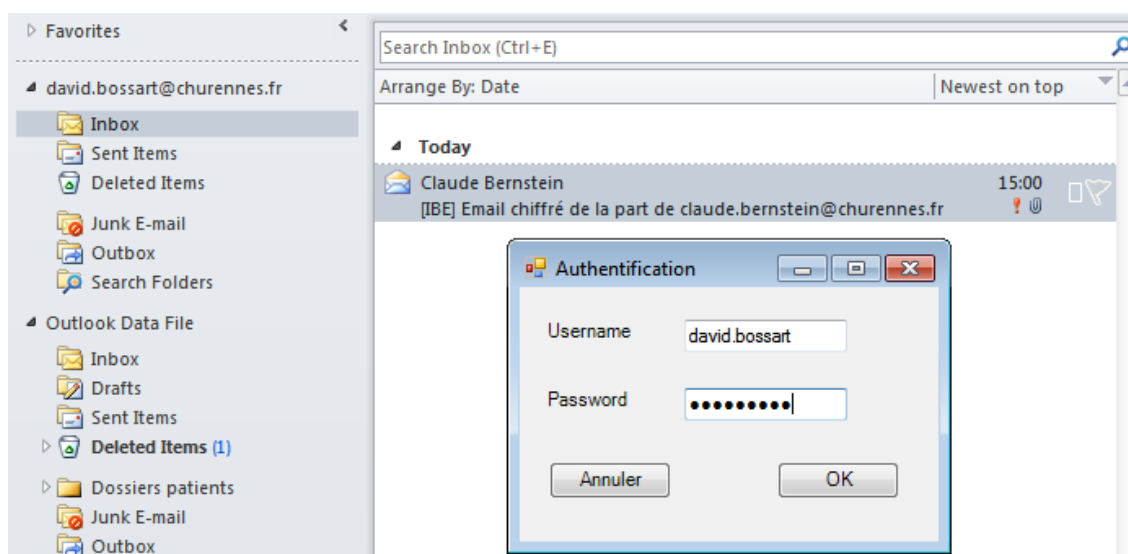


Figure 8 : Authentification auprès du PKG intégrée à Outlook

Si la politique de divulgation autorise le Dr. Bossart à ouvrir le message (du fait de son identité ou de ses fonctions au sein du CHU de Rennes) alors l'email s'ouvre en clair dans une nouvelle fenêtre.

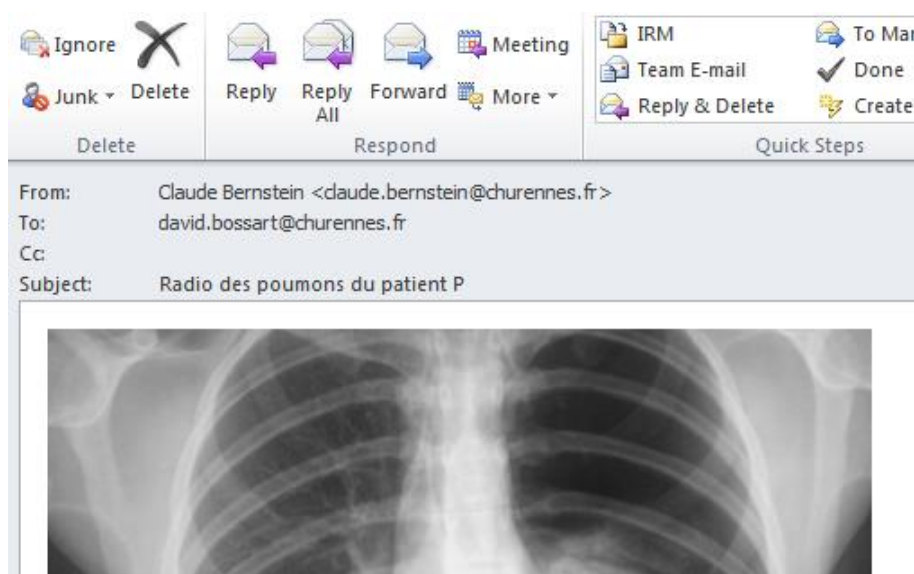


Figure 9 : Déchiffrement et affichage d'un email

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGE	NON PROTREGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 71

9.2. Récapitulatif

Voici une liste des avantages de notre solution par rapport aux systèmes de chiffrement du marché :

Solution de chiffrement « *user friendly* »

- Plus besoin d'obtenir la clé publique du destinataire pour le chiffrement.
- La clé de chiffrement est une chaîne de caractères arbitraire au choix de l'utilisateur.
- Ne modifie pas les habitudes des utilisateurs.

Solution flexible et légère

- La *Master Private Key* et les paramètres publics = 350 octets
- Les clés étant générées à la demande, il n'est pas nécessaire de les stocker en ligne.
- Chiffrement possible avec une identité, un rôle, ou un attribut.
- La politique de divulgation sert à chiffrer les données de manière active. Il n'y a pas besoin de module supplémentaire.
- Il est possible d'inclure directement une date de validité des clés lors du chiffrement.

Solution facile à installer et à administrer

- Installation (ou réinitialisation) du système rapide et *offline* : ne nécessite aucune connexion réseau, aucune génération ni stockage de clé.
- Plus de certificat, plus d'autorité de certification, plus de liste de révocation.
- Plus de base de données de clé publique.
- Plus de stockage des clés privées.
- Robuste à la montée en charge.
- Evolutif

Voici ce qui nous semble également être les inconvénients et points faibles :

- Impossible pour le moment d'empêcher un utilisateur de lire les données chiffrées dans le passé, même s'il n'en a plus le droit. Par exemple, si un utilisateur était auparavant médecin au CHU de Toulon, il pourra toujours déchiffrer les anciens messages chiffrés avec le rôle « Médecin » + « CHU Toulon » même si ce n'est plus le cas, puisqu'il dispose toujours, *a priori*, des anciennes clés de déchiffrement.
- Le PKG est une autorité de séquestre. Il peut générer les clés privées de tous les utilisateurs. Il faut donc faire une confiance absolue en cette entité. Tel quel, notre système ne peut donc pas être utilisé dans des contextes de concurrences entre plusieurs entreprises par exemple.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTEGE	NON PROTEGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 72

9.3. Perspectives et futures améliorations

La révocation des clés de déchiffrement est un élément essentiel. Par manque de temps, ce sujet n'a pas été suffisamment traité pour le moment, ce dont nous avons entièrement conscience. Nous avons vu plus haut que cela posait des problèmes de sécurité. Il s'agit donc d'une priorité de premier rang pour le travail à venir.

En ce qui concerne notre *Add-In* Outlook, il nous faudra encore retravailler l'intégration et améliorer l'expérience utilisateur, tout particulièrement en ce qui concerne prévisualisation des messages. Concernant l'utilisation de notre solution pour la sécurité des services email en général, nous souhaitons à moyen terme pouvoir nous intégrer à n'importe quel client email. Pour cela, nous réfléchissons à adapter notre solution à la norme cryptographique S/MIME.

De nouveaux modules applicatifs sont également en cours de développement. En particulier, nous travaillons à l'intégration de notre solution à Microsoft SharePoint pour faciliter le travail collaboratif en milieu professionnel. Par ailleurs, un travail plus spécifique vis-à-vis d'une application au « *Cloud Santé* » est également en cours. Si ce travail lié au milieu médical se précise, il nous faudra nous pencher sur le respect des contraintes du code de la Santé Publique (Décret confidentialité, authentification à l'aide des cartes professionnelles de santé).

Vis-à-vis des modules de base, nous pensons dans le futur réécrire les algorithmes de chiffrement et déchiffrement pour permettre aux utilisateurs de choisir des politiques de divulgations plus fines à l'aide de porte logique « OR » et « AND ». Avec ces nouveaux algorithmes, il sera possible de chiffrer des données avec une politique du type : ((« membre de la brigade de répression des fraudes » AND (« 75 » OR « 93 »)).

Nous réfléchissons également à implémenter de nouveaux algorithmes. Dans un futur proche, nous intégrerons la possibilité de signer les données par l'identité, ce qui ne représente en réalité que très peu de travail supplémentaire. A moyen terme, nous pourrions aussi nous pencher sur l'algorithme d'IBBE (cf. p 35) pour diffuser des données en *multicast* en utilisant les rôles ou les attributs des utilisateurs.

Enfin, il nous faudra garder un œil attentif aux attaques sur les courbes elliptiques. Rappelons qu'un élément majeur pour l'avenir sera de toujours disposer d'une « courbe elliptique d'avance » pour anticiper de nouvelles attaques et garantir la sécurité de notre système.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 73

Conclusion générale

A l'heure de l'espionnage numérique de masse, il devient de plus en plus urgent pour les entreprises d'agir. Le chiffrement des données est un moyen efficace pour réduire l'impact et les conséquences éventuelles du cyber espionnage. Hélas, nous avons vu que les systèmes actuels, reposant sur l'utilisation d'une infrastructure à clé publique, étaient actuellement trop coûteux et trop complexes à utiliser et à administrer pour la plupart des entreprises, à commencer par les PME/PMI et certaines administrations de l'Etat.

A la lumière de ce rapport, nous pouvons affirmer que la cryptographie basée sur l'identité, bien plus qu'un simple domaine de recherche académique foisonnant, représente dès aujourd'hui un terrain extrêmement fertile pour des innovations majeures. La cryptographie basée sur l'identité, et plus particulièrement l'algorithme d'IBE proposé en 2001 par Franklin et Boneh peut aujourd'hui être implémenté avec succès afin d'aboutir à de réelles applications industrielles. Ainsi, la solution que nous avons développée, permettant le chiffrement des données dans un contexte d'usages nouveaux, tel le *cloud computing* et les espaces de travail collaboratifs, n'est qu'une illustration du potentiel d'innovation de l'IBC.

Notre solution permet nous semble-t-il de réduire la « barrière à l'entrée » que doivent aujourd'hui franchir les organisations pour installer, utiliser et administrer un système complet de chiffrement des données (emails, documents partagés, etc.). Il nous a semblé également que la gestion des droits d'accès aux données numériques était aujourd'hui un enjeu majeur pour des organisations toujours plus dynamiques et complexes. C'est la raison pour laquelle, nous avons voulu mettre à profit le fort potentiel de l'IBE pour concevoir et développer une solution de chiffrement des données basée sur les rôles et les attributs des individus. Dans notre solution, la « clé » servant à chiffrer et la politique de divulgation des données ne font qu'un. C'est cette propriété de l'IBC qui permet à notre solution de réduire drastiquement les contraintes actuelles. En effet, les technologies du marché à base de PKI sont mal adaptées à la gestion de droits d'accès du type RBAC, car elles souffrent d'un problème de flexibilité et sont difficiles à administrer.

Un enjeu majeur nous semble également être celui de la confidentialité de données personnelles de Santé. Alors que le gouvernement souhaite relancer ce chantier important, à travers la généralisation du Dossier Médical Personnel (DMP), nous pensons que l'IBC ferait un excellent candidat pour en garantir la sécurité tout en permettant une gestion facilitée tant pour les praticiens que pour les patients.

Deux mouvements aujourd'hui antagonistes se font parallèlement : d'un côté la bataille pour la sécurisation des données dans un contexte d'espionnage mondial, et de l'autre la volonté des individus de pouvoir partager et s'échanger toujours plus facilement leurs données. Nous pensons que la cryptographie basée sur l'identité est une solution pour réconcilier ces deux mouvements apparemment contradictoires.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 74

Bibliography

1. *Short signatures from the Weil pairing*. **Dan Boneh, Ben Lynn, et Hovav Shacham**. s.l. : Computer Science Department, Stanford University, 2001.
2. *Identity-based Cryptosystems and Signature Schemes*. **Shamir, A.** LNCS 196, s.l. : Springer-Verlag, 1984, Vol. Proceedings of CRYPTO 84'.
3. *Identity-Based Encryption from the Weil Pairing*. **Franklin, D. Boneh et M.** LNCS 2139,, s.l. : Springer-Verlag, 2001, Vol. Proceedings of CRYPTO 2001.
4. *An Identity Based Encryption Scheme Based on Quadratic Residues*. **Cocks, C.** LNCS 2260, s.l. : Springer-Verlag, 2001, Vol. Proceedings of IMA 2001.
5. *Reducing elliptic curve logarithms to logarithms in a finite field*. **A. Menezes, T. Okamoto, and S. Vanstone**. New York : ACM Press, 1991, Vols. STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing.
6. *Parallel collision search with cryptanalytic applications*. **Wiener, P.C. van Oorschot and M.J.** s.l. : Journal of Cryptology, 1999, Vols. 12:1-18.
7. **Kleijung, Thorsten**. Discrete logarithms in $GF(p)$ – 160 digits. *listserv.nodak*. [Online] 5 February 2007. <http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0702&L=NMBRTHRY&P=R45&D=0&I=-3&T=0>.
8. **Kyushu University, NICT and Fujitsu Laboratories**. Achieve World Record Cryptanalysis of Next-Generation Cryptography, 2012. [Online] 2012. <http://www.nict.go.jp/en/press/2012/06/PDF-att/20120618en.pdf>.
9. *Pairings for cryptographers*. **S. Galbraith, K. Paterson, and N. Smart**. s.l. : Discrete Applied Mathematics, 2008, Vols. 15:3113-3121.
10. *A taxonomy of pairing-friendly elliptic curves*. **David freeman, Michael scott et Edlyn teske**.
11. *A remark concerning m -divisibility and the discrete logarithm in the divisor class group of curves*. **Ruck., G. Frey and H.** s.l. : Math. of Computaions, 1994, Vols. 62:865–874.
12. *Constructive and Destructive Facets of Weil Descent on Elliptic Curves*. **P. Gaudry, F. Hess, and N. P. Smart**. 2000 : Department of Computer Science, University of Bristol, Vols. Technical Report CSTR-00-016.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 75

13. *Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups.* **oracles, Short signatures without random.** 2, s.l. : Journal of Cryptology, 2008, Vol. 21.
14. *Signature schemes based on the strong RSA assumption.* **Shoup, Ronald Cramer and Victor.** 3, s.l. : ACM TISSEC, 2000, Vol. 3.
15. *Short Signature from the bilinear pairings.* **S.Akleylek, BB.Kirlar, O.sever et Z.Yuce.** Ankara, Turkey : Institute of applied Mathematics, 2004.
16. *Cryptosystems based on pairings.* **R.Sakai, K.Ohgishi et M.Ksahara.** Okinawa : SCIS2000, 2000.
17. *ID-based Signatures from Pairings on Elliptic Curves.* **Paterson, Kenneth G.** University of London : Mathematics Departement, 2002.
18. *Efficient Identity based Signature Schemes based on Pairings.* **Hess, F.** St. Johns : In K. Nyberg and H. Heys, 2003, Vol. LNCS 2595.
19. *Efficient Identity-Based Encryption Without Random Oracles.* **Waters, Brent.** s.l. : Cryptology ePrint Archive, 2004, Vol. Report 2004/180.
20. *Identity-Based Broadcasting.* **Yi Mu, Willy Susilo et Yan-Xia Lin.** pp 177-190, s.l. : Dept. of Electrical and Information Technology, Lund University , 2003, Vols. Progress in Cryptology – INDOCRYPT 2003.
21. *A Flexible Role-based Secure Messaging Service : Exploiting IBE Technology in a Health Care Trial.* **Marco Casassa Mont, Pete Bramhall, Chris R. Dalton et Keith Harrison.** s.l. : HP Laboratories Bristol, 2003, Vols. HPL-2003-21.
22. *ID-Based Cryptography for Secure Cloud Data Storage.* **Nesrine Kaaniche, Aymen Boudguiga, Maryline Laurent.** s.l. : Institut Mines-Telecom, 2012.
23. *Identity-Based Authentication for Cloud Computing.* **Hongwei Li, Yuanshun Dai, Ling Tian et Haomiao Yang.** s.l. : University of Electronic Science and Technology of China, 2009, Vol. CloudCom 2009.
24. *Secure cloud storage based on cryptographic techniques.* **PENG Yong, ZHAO Wei, XIE Feng, DAI Zhong-hua, GAO Yang, CHEN Dong-qing.** s.l. : Elsevier, 2012.
25. *Separating Decision $Di\pm e$ -Hellman from $Di\pm e$ -Hellman.* **Nguyen., A. Joux and K.** s.l. : Cryptology ePrint Archive, 2001, Vol. Report 2001/003.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 76

26. **Wood P., et al.** *Internet Security Threat Report*. Cyber Security Intelligence, Symantec. 2012. p. 52. URL : http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_2011_21239364.en-us.pdf. 21239364.
27. **Kaspersky.** *Operation "Red October" : victims of advanced cyber-espionage network*. 2013. URL : http://www.securelist.com/en/images/vlill/kaspersky_lab_infographic_red_october_victims_by_country.png.
28. **MITRE.** <http://www.mitre.org/>. [Online]
29. **Mandiant.** *M-trends : The advanced persistent threat*. 2010. p. 30. URL : <https://www.mandiant.com/resources/m-trends/>.
30. **Falliere.N, et al.** *W32. Stuxnet Dossier*. s.l. : Symantec, 2011. p. 69. URL : http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.
31. **IPA.** *Design and Operational Guide to Protect against "Advanced Persistent Threats"*. IPA's working group for threats and measures. 2011. p. 64. URL : http://www.ipa.go.jp/security/vuln/documents/eg_newattack.pdf.
32. **J., Oltsik.** *Understanding and Addressing APTs*. s.l. : The Enterprise Strategy Group, 2012. p. 10, Livre blanc. URL : <http://www.trendmicro.co.uk/media/wp/esg-apt-deep-discovery-whitepaper-en.pdf>.
33. **JM., Bockel.** *La cyberdéfense : un enjeu mondial, une priorité nationale*. Paris : Sénat, 2012. p. 158. URL : <http://www.senat.fr/rap/r11-681/r11-6811.pdf>. 681.
34. **R., Miller.** *advanced persistent threats: defending from inside out*. Security Management, CA technologies. s.l. : CA technologies, 2012. p. 16, White Paper. URL : <https://www.ca.com/us/register/forms/collateral/Advanced-Persistent-Threats-Defending-From-The-Inside-Out.aspx>.
35. **Mandiant.** *M-trends : When Prevention Fails*. s.l. : Mandiant, 2011. p. 32. URL : <https://www.mandiant.com/resources/m-trends/>.
36. **hackmageddon.com.** 2012 Cyber Attacks Timeline Master Index. *hackmageddon.com*. [Online] 2012. <http://hackmageddon.com/2012-cyber-attacks-timeline-master-index/>.

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTÈGE	NON PROTÈGE	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 77

37. **Kaspersky Lab's Global Research & Analysis Team.** "Red October" Diplomatic Cyber Attacks Investigation. *securelist*. [Online] 2013.

http://www.securelist.com/en/analysis/204792262/Red_October_Diplomatic_Cyber_Attacks_Investigation.

38. **TrendLabsSM APT Research Team.** *Spear-Phishing Email: Most Favored APT Attack Bait*.

s.l. : Trend Micro, 2012. p. 6. URL : <http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf>.

39. www.sourcefire.com. <http://www.sourcefire.com/security-technologies/advanced-malware-protection/fireamp>. [Online]

40. www.net-security.org. http://www.net-security.org/malware_news.php?id=1968. [Online]

41. www.fireeye.com. <http://www.fireeye.com/products-and-solutions/malware-analysis.html>. [Online]

Classification société <i>Clearance level</i>	Statut <i>Status</i>	Référence interne <i>Internal reference</i>	Version	Edition	Date de l'édition <i>Issue date</i>	Langue <i>Language</i>	Page
NON PROTREGÉ	NON PROTREGÉ	XXX/XXX/XXX/XXX	1.0	XX	20 novembre 2013	FR	Page 78

**FOR MORE INFORMATION PLEASE CONTACT**

Cassidian CyberSecurity / Postal code / Town / Country / T: +12 (0) 3456.7-8910 / F: 12 (0) 3456.7-8911