# Cryptographie asymétrique

La **cryptographie asymétrique**, ou **cryptographie à clé publique**, est une méthode de chiffrement qui s'oppose à la cryptographie symétrique. Elle repose sur l'utilisation d'une clé publique (qui est diffusée) et d'une clé privée (gardée secrète), l'une permettant de coder le message et l'autre de le décoder. Ainsi, l'expéditeur peut utiliser la clé publique du destinataire pour coder un message que seul le destinataire (en possession de la clé privée) peut décoder, garantissant la confidentialité du contenu. Inversement, l'expéditeur peut utiliser sa propre clé privée pour coder un message que le destinataire peut décoder avec la clé publique ; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message.

## 1 Historique

### 1.1 Concept

Le concept de cryptographie à clé publique — autre nom de la cryptographie asymétrique — est généralement attribué à Whitfield Diffie et à Martin Hellman qui l'ont présenté au public à la *National Computer Conference* en 1976[1], puis publié quelques mois plus tard dans *New Directions in Cryptography*[2]. Le concept aurait cependant été découvert indépendamment par d'autres chercheurs à la même époque.

Ralph Merkle aurait fait la même découverte à la même époque[3], même si ses articles[4] ne furent publiés qu'en 1978.

### 1.2 Mise en œuvre

Dans leur article de 1976, W. Diffie et M. Hellman n'avaient pas pu donner l'exemple d'un système à clé publique, n'en ayant pas trouvé. Il fallut attendre 1978 pour avoir un exemple[5] donné par Ronald Rivest, Adi Shamir et Leonard Adleman, le RSA, abréviation tirée des trois noms de ses auteurs. Les trois hommes fondèrent par la suite la société RSA Security. Le système Merkle-Hellman[6] est généralement considéré comme la première réalisation pratique d'un système de chiffrement à clé publique[7], il a cependant été prouvé non-sûr par Shamir en 1982[8].

### 1.3 Recherches secrètes du GCHQ

Parallèlement aux recherches publiques, les services du chiffre britannique (GCHQ, *Government Communications Headquarters*) auraient mené des recherches secrètes aboutissant à des concepts et outils de chiffrement asymétrique dès la première moitié des années 1970[9] :

- James Ellis, du GCHQ aurait proposé le concept avant Hellman et Diffie.
- C.C. Cocks aurait décrit ce qu'on a appelé l'algorithme RSA dès 1973
- M. J. Williamson aurait inventé un protocole d'échange de clé très proche de celui de Diffie et de Hellman dès 1974.

Ces découvertes n'auraient été rendues publiques par le GCHQ qu'en 1997[9].

## 2 Fonctionnement

### 2.1 Principe général

La **cryptographie asymétrique**, ou *cryptographie à clé publique* est fondée sur l'existence des fonctions à sens unique et à brèche secrète.

Les fonctions à sens unique sont des fonctions mathématiques telles qu'une fois appliquées à un message, il est extrêmement difficile de retrouver le message original.

L'existence d'une brèche secrète permet cependant à la personne qui a conçu la fonction à sens unique de décoder facilement le message grâce à un élément d'information qu'elle possède, appelé clé privée.

Supposons qu'Alice souhaite recevoir un message secret de Bob sur un canal susceptible d'être écouté par un attaquant passif Eve.

- Alice transmet à Bob une fonction à sens unique pour laquelle elle seule connait la brèche secrète.
- Bob utilise la fonction transmise par Alice pour chiffrer son message secret
- Alice réceptionne le message chiffré puis le décode grâce à la brèche secrète

- Si Eve réceptionne également le message alors qu'il circule sur le canal public, elle ne peut le décoder, même si elle a également intercepté l'envoi de la fonction à sens unique, car elle n'a pas connaissance de la brèche secrète.

La terminologie classiquement retenue est :

- pour la fonction à sens unique et brèche secrète : « clé publique »
- pour la brèche secrète : « clé privée »

En pratique, sont utilisées des fonctions de chiffrement classiques, les termes « clé publique » et « clé privée » correspondant alors à des paramètres employés pour ces fonctions.

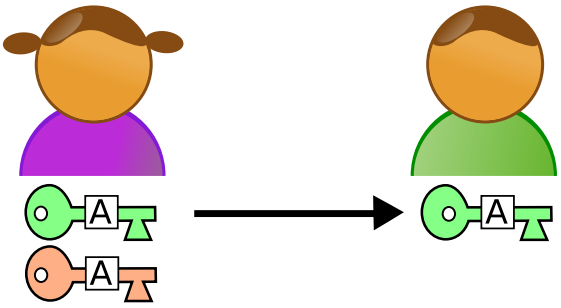
## 2.2 Fonctionnement pratique

Alice souhaite pouvoir recevoir des messages chiffrés de n'importe qui.

### 2.2.1 Diffusion des clés publiques

Elle génère alors une valeur à partir d'une fonction à sens unique et à brèche secrète à l'aide d'un algorithme de chiffrement asymétrique (liste ici), par exemple RSA[10].

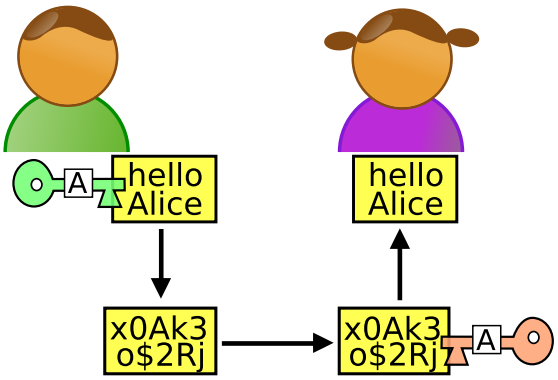
Alice diffuse à tout le monde la fonction pour coder les messages (notée clé publique) mais garde secrète la fonction de décodage (notée clé privée).




*1re étape : Alice génère deux clés. La clé publique (verte) qu'elle envoie à Bob et la clé privée (rouge) qu'elle conserve précieusement sans la divulguer à quiconque.*

### 2.2.2 Chiffrement

Un des rôles de la clé publique est de permettre le chiffrement ; c'est donc cette clé qu'utilisera Bob pour envoyer des messages chiffrés à Alice. L'autre clé — l'information secrète — sert à *dé*chiffrer. Ainsi, Alice, et elle seule, peut prendre connaissance des messages de Bob. La connaissance d'une clé ne permet pas de déduire l'autre.




*2e et 3e étapes : Bob chiffre le message avec la clé publique d'Alice et envoie le texte chiffré. Alice déchiffre le message grâce à sa clé privée.*

### 2.2.3 Authentification de l'origine

D'autre part, l'utilisation par Alice de sa clé privée sur le condensat d'un message, permettra à Bob de vérifier que le message provient bien d'Alice : il appliquera la clé publique d'Alice au condensat fourni (condensat chiffré avec la clé privée d'Alice) et retrouve donc le condensat original du message. Il lui suffira de comparer le condensat ainsi obtenu et le condensat réel du message pour savoir si Alice est bien l'expéditeur. C'est donc ainsi que Bob sera rassuré sur l'origine du message reçu : il appartient bien à Alice. C'est sur ce mécanisme notamment que fonctionne la signature numérique.

# 3 Analyse fonctionnelle

## 3.1 Analogies

### 3.1.1 Le coffre-fort

Le chiffrement : Alice a choisi un coffre-fort. Elle l'envoie ouvert à Bob, et en garde la clé. Lorsque Bob veut écrire à Alice, il y dépose son message, ferme le coffre, il n'a pas besoin de la clé pour cela, et le renvoie à Alice. À sa réception, seule Alice peut ouvrir le coffre, puisqu'elle seule en possède la clé, à supposer le coffre inviolable, et que personne ne puisse refaire la clé.

L'authentification ou la signature : Alice place un message dans le coffre-fort qu'elle ferme avec sa clé privée avant de l'envoyer à Bob. Si Bob parvient, à l'aide de la clé publique d'Alice (dont il dispose), à lire la lettre c'est que c'est bien celui d'Alice et donc que c'est bien elle qui y a placé le message.

### 3.1.2 La boîte à deux serrures

Une autre analogie envisageable serait d'imaginer une boîte avec deux serrures différentes. Lorsque l'on ferme

la boîte d'un côté, seule la clé correspondant à l'autre serrure permet l'ouverture de la boîte et vice-versa. Une des clés est privée et conservée secrète, l'autre est dite publique et un exemplaire peut-être obtenu par quiconque souhaite utiliser la boîte.

Pour chiffrer un message Bob prend la boîte, y place son message, et la ferme à l'aide de la clé publique. Seul le détenteur de la clé privée permettant d'accéder à l'autre serrure, Alice en l'occurrence, sera en mesure de rouvrir la boîte.

Pour signer un message, Alice le place dans la boîte et ferme celle-ci à l'aide de sa clé privée. Ainsi n'importe qui ayant récupéré la clé publique pourra ouvrir la boîte. Mais comme la boîte a été fermée par la clé privée, cette personne sera assurée que c'est bien Alice, seule détentrice de cette clé, qui aura placé le message dans la boîte et fermé ladite boîte.

## 3.2 Inconvénients/limites

En contrepartie de leurs propriétés spécifiques, les chiffrements asymétriques sont globalement moins performants que leurs équivalents symétriques : les temps de traitement sont plus longs et, pour un niveau de sécurité équivalent, les clés doivent être beaucoup plus longues.

Si le chiffrement asymétrique permet de se prémunir des écoutes passives (*eavesdrop*), la transmission initiale de la clé publique sur un canal non sécurisé expose à des attaques de l'homme du milieu. Pour se prémunir contre ce risque on fait généralement appel à une infrastructure à clés publiques.

## 3.3 Articulation avec le chiffrement symétrique

La cryptographie asymétrique répond à un besoin majeur de la cryptographie symétrique : le partage *sécurisé* d'une clé entre deux correspondants, afin de prévenir l'interception de cette clé par une personne tierce non autorisée, et donc la lecture des données chiffrées sans autorisation.

Les mécanismes de chiffrement symétrique étant moins coûteux en temps de calcul, ceux-ci sont préférés aux mécanismes de chiffrement asymétrique. Cependant toute utilisation de clé de chiffrement symétrique nécessite que les deux correspondants se *partagent* cette clé, c'est-à-dire la connaissent avant l'échange. Ceci peut être un problème si la communication de cette clé s'effectue par l'intermédiaire d'un medium non sécurisé, « en clair ». Afin de pallier cet inconvénient, on utilise un mécanisme de chiffrement asymétrique pour la seule phase d'échange de la clé symétrique, et l'on utilise cette dernière pour tout le reste de l'échange.

# 4 Applications

## 4.1 Mécanismes d'authentification [réf. souhaitée]

Article détaillé : Signature numérique.

Un inconvénient majeur de l'utilisation des mécanismes de chiffrement asymétriques est le fait que la clé publique est distribuée à toutes les personnes : *Bob*, *Carole*, … souhaitant échanger des données de façon confidentielle. De ce fait, lorsque la personne possédant la clé privée, *Alice*, déchiffre les données chiffrées, elle n'a aucun moyen de vérifier avec certitude la provenance de ces données (*Bob*, ou *Carole* …) : on parle de problèmes d'authentification. Afin de résoudre ce problème, on utilise des mécanismes d'authentification permettant de garantir la provenance des informations chiffrées. Ces mécanismes sont eux aussi fondés sur le chiffrement asymétrique.

Principe d'authentification par chiffrement asymétrique :

Objectif : Bob souhaite envoyer des données chiffrées à Alice en lui garantissant qu'il en est l'expéditeur.

1. Bob crée une paire de clés asymétriques : il conserve la clé privée et diffuse librement la clé publique (notamment à Alice)
2. Alice crée une paire de clés asymétriques : clé privée (qu'elle conserve), clé publique (qu'elle diffuse librement, notamment à Bob)
3. Bob effectue un condensat de son message « en clair » puis chiffre ce condensat avec *sa propre clé privée*
4. Bob chiffre son message avec *la clé publique d'Alice*.
5. Bob envoie le message chiffré accompagné du condensat chiffré.
6. Alice reçoit le message chiffré de Bob, accompagné du condensat.
7. Alice déchiffre le message avec *sa propre clé privée*. À ce stade le message est lisible mais elle ne peut pas être **sûre** que Bob en soit l'expéditeur.
8. Alice déchiffre le condensat avec *la clé publique de Bob*. Alice utilise la même fonction de hachage sur le texte *en clair* et compare avec le condensat déchiffré de Bob. Si les deux condensats correspondent, alors Alice peut avoir la certitude que Bob est l'expéditeur. Dans le cas contraire, on peut présumer qu'une personne malveillante a tenté d'envoyer un message à Alice en se faisant passer pour Bob !

Cette méthode d'authentification utilise la spécificité des paires de clés asymétriques : si l'on chiffre un message en

utilisant la clé publique, alors on peut déchiffrer le message en utilisant la clé privée ; l'inverse est aussi possible : si l'on chiffre en utilisant la clé privée alors on peut déchiffrer en utilisant la clé publique.

### 4.2 Certificats

La cryptographie asymétrique est également utilisée avec les certificats numériques, celui-ci contenant la clé publique de l'entité associée au certificat. La clé privée est quant à elle stockée au niveau de cette dernière entité. Une application des certificats est par exemple la mise en œuvre d'une infrastructure à clés publiques (PKI) pour gérer l'authentification et la signature numérique d'une entité, par exemple un serveur web (Apache avec le module SSL par exemple), ou simplement un client souhaitant signer et chiffrer des informations à l'aide de son certificat de la façon décrite dans les sections précédentes.

## 5 Une clé privée inviolable ?

Un chiffrement symétrique au moyen d'une clé de 128 bits propose $2^{128}$ ( ~ 3,4 $10^{38}$ ) façons de chiffrer un message. Un pirate qui essaierait de déchiffrer le message par la force brute devrait les essayer une par une.

Pour les systèmes à clé publique, il en va autrement. Tout d'abord les clés sont plus longues (par exemple 1 024 bits minimum pour RSA) ; en effet, elles possèdent une structure mathématique très particulière (on ne peut pas choisir une suite de bits aléatoire comme clé secrète, par exemple dans le cas du RSA, seuls les nombres premiers sont utilisés). Certains algorithmes exploitant cette structure sont plus efficaces qu'une recherche exhaustive sur, par exemple, 1 024 bits. Ainsi, dans le cas de RSA, le crible général de corps de nombres est une méthode plus efficace que la recherche exhaustive pour la factorisation.

Il faut noter le développement actuel de la cryptographie utilisant les courbes elliptiques, qui permettent (au prix d'une théorie et d'implémentations plus complexes) l'utilisation de clés nettement plus petites que celles des algorithmes classiques (une taille de 160 bits étant considérée comme très sûre actuellement), pour un niveau de sécurité équivalent.

Dans son édition du 6 septembre 2013, le journal le Guardian affirmait que la NSA était capable de déchiffrer la plupart des données chiffrées circulant sur Internet[11]. De nombreuses sources ont cependant indiqué que la NSA n'avait pas mathématiquement cassé les chiffrements mais s'appuierait sur des faiblesses d'implémentation des protocoles de sécurité[12],[13].

## 6 Références


- Pierre Barthélemy, Robert Rolland, Pascal Véron (préf. Jacques Stern), Cryptographie : principes et mises en œuvre, Hermes Science Publications : Lavoisier, coll. « Collection Informatique », 22 juillet 2005, Broché, 414 p. (ISBN 2-7462-1150-5, ISSN 1242-7691, OCLC 85891916).

[1] W. Diffie and M.E. Hellman, *Multiuser cryptographic technics*, Proceedings of AFIPS National Computer Conference, 109-112, 1976

[2] W. Diffie and M.E. Hellman, *New directions in cryptography*, IEEE transactions on information theory, 22(1976), 644-654

[3] A.J. Menezes, P.C Van Oorschot, S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997, p. 47

[4] R.C. Merkle, *Secure communications over insecure channels*, Communications of the ACM, 21(1978),294-299

[5] Ronald Rivest, Adi Shamir, Leonard Adleman, *A Method for Obtaining Digital Signatures and Public-key Cryptosystems*, Communications of the ACM, 21(1978), 120-126

[6] R.C. Merkle, M.E. Hellman,, *Hiding information and signatures in trapdoor functions*, IEEE transactions on information theory, 24(1978),525-530

[7] A.J. Menezes, P.C Van Oorschot, S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997, p. 300

[8] A. Shamir, *A polynomial time algorithm for breaking the basic Merkle-Hellman crypto-system*, Advances in cryptology, Proceedings of Crypto 82, 1983, 279-288

[9] S. Singh, Histoire des codes secrets, Paris, attès, 1999, 430 p. (ISBN 978-2-7096-2048-2) (Chapitre *Alice et Bernard s'affichent en public*, section *Un autre historique* …)

[10] (en) Cryptographic communications system and method

[11] James Ball, Julian Borger et Glenn Greenwald, Revealed : how US and UK spy agencies defeat internet privacy and security, The Guardian, 6 septembre 2013, consulté le 7 octobre 2013

[12] Bruce Schneier, The NSA Is Breaking Most Encryption on the Internet, Schneier on Security, 5 septembre 2013, consulté le 7 octobre 2013

[13] Tom Simonite, NSA Leak Leaves Crypto-Math Intact but Highlights Known Workarounds, MIT Technology Review, 9 septembre 2013, consulté le 7 octobre 2013


## 7 Voir aussi

- Cryptographie hybride
- Infrastructures à Clés Publiques (PKI)
- Authentification

- Authentification forte
- Alice et Bob
- Signature numérique
- Chiffrement
- Liste d'*algorithmes de cryptographie asymétrique*
  - RSA, le plus utilisé d'entre eux
  - Cryptosystème de ElGamal
  - Cryptosystème de Merkle-Hellman

# 8 Logiciels de cryptographie asymétrique

- *Pretty Good Privacy* ou PGP, logiciel existant en versions payante et gratuite (aux fonctionnalités plus réduites).
- *GNU Privacy Guard* ou GPG ou GnuPG, version Libre (open-source) et gratuite de PGP.
- OpenSSL (Open Secure Socket Layer), version libre et gratuite permettant, notamment, de développer des fonctionnalités à base de chiffrement asymétrique.
- Acid Cryptofiler, Logiciel développé par la Délégation Générale pour l'armement pour un usage gouvernemental.

-  Portail de la cryptologie

# 9 Sources, contributeurs et licences du texte et de l'image

## 9.1 Texte

- **Cryptographie asymétrique** *Source :* https://fr.wikipedia.org/wiki/Cryptographie_asym%C3%A9trique?oldid=114354619 *Contributeurs :* MichelLaglasse, Francis, Calo, Dtcube, Ryo, Alvaro, Looxix, Orthogaffe, Oz, Ske, Gloumouth1, HasharBot, Raph, NucleoS, Robbot, Saimonn, Guillaume~frwiki, Giudicelli, Anakin, JPaul, Archibald, MedBot, Sam Hocevar, Xmlizer, Theocrite, Ollamh, SpICE, Aither, Escaladix, ClementSeveillac, Mika, Dake, Bibirico, Criric, JujuTh, Sador, Antoinetav, Padawane, DocteurCosmos, Elg, Chobot, Vazkor, Probot, Gzen92, RobotQuistnix, Ash Crow, Jerome66, Toutoune25, MagnetiK-BoT, Grecha, Freewol, Esprit Fugace, Escounda, SashatoBot, MetalGearLiquid, Epsilon0, Akira Yuki, Callyope, Thijs !bot, Smertrios, JAnDbot, Ce`dric, Dfeldmann, PouX, Rei-bot, Salebot, Bot-Schafter, Speck-Made, Zorrobot, Vincent Lextrait, Smaret, TXiKiBoT, VolkovBot, BlueGinkgo, AmaraBot, Chicobot, Bdlbb, Ptbotgourou, AlleborgoBot, Rabatakeu, SieBot, Louperibot, Hypnocrate, OKBot, Alecs.bot, □□□□ robot, DumZiBoT, Ficbot, Ir4ubot, ZiziBot, AgatheD, HerculeBot, ZetudBot, Bub's wikibot, JackPotte, GrégoireG, LinkFA-Bot, Luckas-bot, Zandr4, Anne Bauval, Lomita, Nejimban, Catschlum, Jackou06, Iqspro, EmausBot, Gyrostat, Dalnord, Vthierry, Grelot-de-Bois, Arbiel, Skhaen, OrlodrimBot, Robert.Rolland, Gcob, Elliot James, Metamorforme42, Ramzan, Razibot, Unique Nitrogen, Housterdam, JurgenNL, Addbot et Anonyme : 110

## 9.2 Images

- **Fichier:Asymmetric_cryptography_-_step_1.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/0/01/Asymmetric_cryptography_-_step_1.svg *Licence :* CC-BY-SA-3.0 *Contributeurs :* own work, based on png version originally uploaded to the Commons by Dake. *Artiste d'origine :* odder
- **Fichier:Asymmetric_cryptography_-_step_2.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/1/11/Asymmetric_cryptography_-_step_2.svg *Licence :* CC-BY-SA-3.0 *Contributeurs :* own work, based on png version originally uploaded to the Commons by Dake. *Artiste d'origine :* odder
- **Fichier:Crypto_key.png** *Source :* https://upload.wikimedia.org/wikipedia/commons/5/5b/Crypto_key.png *Licence :* LGPL *Contributeurs :* Transféré de fr.wikipedia à Commons. *Artiste d'origine :* Original téléversé par Dake sur Wikipedia français Later versions were uploaded by Croquant at fr.wikipedia.
- **Fichier:Question_book-4.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/6/64/Question_book-4.svg *Licence :* CC-BY-SA-3.0 *Contributeurs :* Created from scratch in Adobe Illustrator. Originally based on Image:Question book.png created by User:Equazcion. *Artiste d'origine :* Tkgd2007

## 9.3 Licence du contenu

- Creative Commons Attribution-Share Alike 3.0