Introduction à la cryptographie

Master Ingénierie des logiciels

1.0

Table des matières

Introduction à la cryptographie

I

Cette partie du cours introduira les mécanismes de base de la cryptographie moderne qui permettent de réaliser quatre services de sécurité fondamentaux :

1. La confidentialité
2. L'intégrité des données
3. L'authentification de l'origine de données
4. La non-répudiation de l'origine

Pour chacun de ces services nous rappellerons la définition puis nous introduirons le

mécanisme cryptographique permettant de le réaliser.

1. La cryptographie

Définition : La cryptographie

Le mot « Cryptographie » est composé des mots grecques :

- CRYPTO = caché
- GRAPHY = écrire

C'est donc l'art de l'écriture secrète.

C'est une science permettant de préserver la confidentialité des échanges.

Définition : Cryptanalyse

La cryptanalyse est l'art de décrypter des messages chiffrés.

Objectifs

Parmi les objectifs de la cryptographie :

- Garantir la confidentialité
- Vérifier l'intégrité des données
- Gérer l'authentification
- Assurer la non-répudiation

2. La confidentialité

2.1. Confidentialité et chiffrement

Définition : Confidentialité

La confidentialité est la propriété qui assure que l'information est rendu inintelligible aux individus, entités, et processus non autorisés.

Définition : Chiffrement / déchiffrement

Le chiffrement est une transformation cryptographique qui transforme un message clair en un message inintelligible (dit message chiffré), afin de cacher la signification du message original aux tierces entités non autorisées à l'utiliser ou le lire. Le déchiffrement est l'opération qui permet de restaurer le message original à partir du message chiffré.

Clé de chiffrement

Dans la cryptographie moderne, l'habilité de maintenir un message chiffré secret, repose non pas sur l'algorithme de chiffrement (qui est largement connu), mais sur une information secrète dite CLE qui doit être utilisée avec l'algorithme pour produire le message chiffré.

Selon que la clé utilisée pour le chiffrement et le déchiffrement est la même ou pas, on parle de système cryptographique symétrique ou asymétrique.

Fondamental : Chiffrement symétrique

récepteur. Cette clé dite symétrique est utilisée par l'émetteur pour chiffrer le message te par le récepteur pour le déchiffrer en utilisant un algorithme de chiffrement symétrique.

Image 7 : Chiffrement Symétrique

Exemple : Algorithmes de chiffrement symétriques

Il existe deux types d'algorithmes de chiffrement symétrique :

1. Chiffrement par bloc : division du texte clair en blocs fixe, puis chiffrement bloc par bloc
    - DES: IBM, Standard NIST 19767
    - 3DES: W. Diffie, M. Hellman, W. Tuchmann 1978.
    - IDEA: Xuejia Lai et James Massey en 1992
    - Blowfish: Bruce Schneier en 1993
    - AES (Rijndael): Joan Daemen et Vincent Rijmen 2000

2. Chiffrement par flux : le bloc a une dimension unitaire (1 bit, 1 octet, ...), ou une taille relativement petite

    - RC4: Ron Rivest 1987
    - SEAL: Don Coppersmith et Phillip Rogaway pour IBM 1993.

Fondamental : Chiffrement asymétrique

Dans un système asymétrique, le récepteur génère une paire de clés asymétrique : une clé publique qui est diffusée à tout le monde et une clé privée maintenue secrète chez le récepteur. La particularité de cette paire de clé est que tout message chiffrée avec la clé publique ne peut être déchiffré qu'avec la clé privée correspondante. D'où la confidentialité des messages chiffré avec la clé publique d'un récepteur. Bien évidemment la clé privée correspondante ne peut être calculée à partir de la clé publique correspondante.

Image 8 : Chiffrement Asymétrique

Exemple : Algorithmes de chiffrement asymétrique

RSA: Rivest, Shamir et Adleman 1978

Diffie et Hellman 1976

2.2. Introduction à DES

Aperçus générale

DES (Data Encryption Standard) est l'un des algorithmes pionnier de chiffrement symétrique. Il est basé sur un ensemble de permutations et substitutions comme présenté ci-dessous. C'est un algorithme qui opère sur des blocs de 64 bits, et utilise une clé de 56 bits qui était suffisante à l'époque. Il a été définis officiellement dans FIPS46-3. Il est constitué d'une permutation initiale, un calcul médian en fonction de la clé et une permutation finale.

La permutation initiale

La figure ci-dessous illustre la permutation initiale sur un bloc de 64 bits sur lequel opère DES

Permutation initiale

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Image 9 : Permutation initiale de DES

Algorithme DES

Cette figure résume les 16 itérations de l'algorithme DES. La fonction f intervenant à chaque phase sera présentée dans le paragraphe suivant.

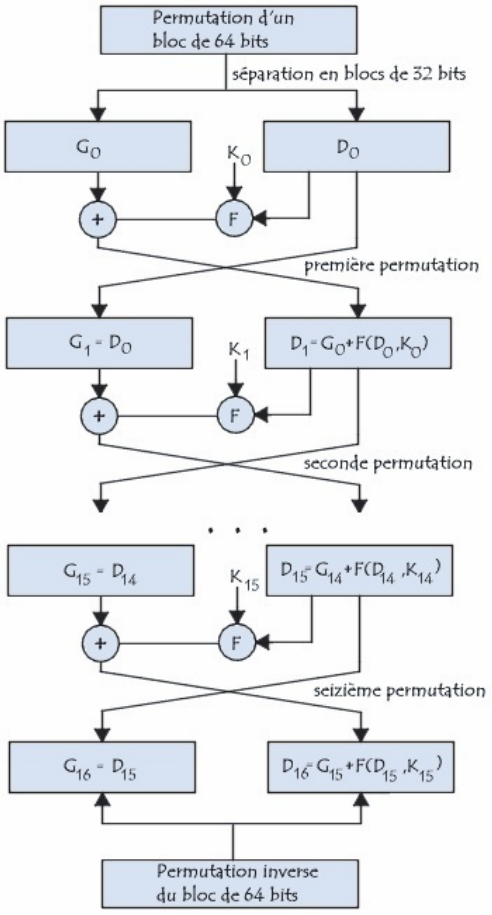
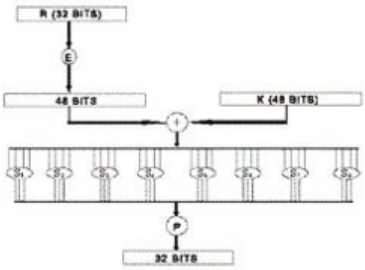


Image 10 : Algorithme DES

La fonction f de DES

La figure suivante illustre la fonction f et ses différentes sous fonctions, ainsi qu'un exemple d'une S-box

f



E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

S_1

Ligne No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Image 11 : La fonction f

⚠ *Attention : Sécurité de DES*

DES fut raisonnablement sûr à l'époque de son invention. RSA Security a lancé le DES Challenge qui a permis de mettre fin à la robustesse de DES à la cryptanalyse :

- DES Challenge I 1997: DESCHALL a cassé la clé DES en 96 j
- DES Challenge II-1 1998: Distributed.net a résussi à casser la clé DES en 41j
- DES Challenge II-2 1998: EFF Deep Crack a cassé la clé DES en 56h
- DES Challenge III 1999: Deep Crack et Distributed.net ont cassé la clé DES en 22h15

En 2000 AES deviens le standard à la place de DES

2.3. Les modes d'opération du chiffrement symétrique

Dans le chiffrement symétrique l'algorithme opère sur un bloc. Pour chiffrer un ensemble de blocs constituants le message à chiffrer il est nécessaire de définir une stratégie d'opération sur la succession des blocs à chiffrer.

Il existe quatre modes définis dans FIPS 81 (1980)

- Electronic Code Book (ECB),
- Cipher Block Chaining (CBC),
- Cipher FeedBack (CFB) et
- Output FeedBack (OFB).

Syntaxe : Notation

Nous adoptons la notation suivante dans la description des quatre modes d'opération :

- T[n]: n-ième bloc du texte clair
- C[n]: n-ième bloc du texte chiffré
- E(m): fonction de chiffrement
- D(m): fonction de déchiffrement
- IV : Initialization Vector
- ^ : XOR

Electronic Code Book (ECB)

Chiffrement :C[n] = E(T[n])

Déchiffrement :T[n] = D(C[n])

Le même texte clair et clé de chiffrement donnent le même texte chiffré.

CBC : Cipher Block Chaining

Chiffrement :

- C[0] = E(T[0] ^ IV)
- C[n] = E(T[n] ^ C[n-1]) , si (n > 0)

Déchiffrement :

- T[0] = D(C[n]) ^ IV
- T[n] = D(C[n]) ^ C[n-1] , si (n > 0)

IV est envoyé en clair avec le message chiffré

CFB : Cipher Feedback

I[n]: bloc temporaire

Chiffrement :

- I[0] = VI
- I[n] = C[n-1] , si (n > 0)
- C[n] = T[n] ^ E(I[n]

Déchiffrement :

- I[0] = VI
- I[n] = C[n-1], si (n > 0)
- T[n] = C[n] ^ E(I[n])

Offre une sécurité plus élevée

OFB : Output Feedback

I[n]=nième bloc temporaire

R[n]=nième bloc temporaire second

Chiffrement :

- I[0] = VI
- I[n] = R[n-1] , si (n > 0)
- R[n] = E(I[n])
- C[n] = T[n] ^ R[n]

Déchiffrement :

- I[0] = VI
- I[n] = R[n-1] , si (n > 0)

- R[n] = E(I[n])
- T[n] = C[n] ^ R[n]

2.4. RSA : Rivest Shamir et Adleman 1978

Fondamental : Principe de RSA

RSA esr fondé sur la difficulté de factoriser des grands nombres qui sont le produit de deux grands nombres premiers.

Cryptographiquement parlant, on peut dire que multiplier deux grands nombres premiers est une fonction à sens unique: Il est facile de multiplier deux nombres pour obtenir un produit, mais difficile de factoriser ce produit et de retrouver les deux grands nombres premiers.

Algorithme RSA

Initialisation

- Choisir deux nombres premiers, p et q, les deux étant plus grands que 10100.
- Calculer n = p · q (n est le modulus)
- Choisir e aléatoire tel que e et ((p - 1) · (q - 1)) n'aient aucun facteur commun excepté 1
- Trouver d tel que : ed = 1 mod((p - 1)(q - 1)).
- Clé publique : (n,e).
- Clé privée : (n,d) ou (p,q,d) si on désire garder p et q.

Chiffrement/Déchiffrement

- L'expéditeur crée le texte chiffré c à partir du message m : c = me mod(n), où (n,e) est la clé publique du destinataire
- Le destinataire reçoit c et effectue le déchiffrement : m = cd mod(n), où (n,d) est la clé privée du destinataire.

Attention : Sécurité de RSA

Ce qui est connus est la clé publique (e,n)

Pour déchiffrer un message m, il faut connaître d tel que ed=1 mod(p-1)(q-1)

Pour calculer d il faut donc connaître p et q

Or on sait que n=pq et on connaît n

Il faut donc factoriser n en ses facteurs premiers p et q

Or personne n'a pus le faire en un temps raisonnable.

2.5. Échange de clé Diffie-Hellman

Objectif de l'algorithme

Deux entités voudraient se mettre d'accord sur un secret (en échangenat des messages publiques) afin de s'échanger des messages confidentiels

Fondamental

L'algorithme de Diffie Hellman a été fondé sur la difficulté du calcul du logarithme discret

Algorithme Diffie-Hellman

La figure suivante illustre les différentes étapes à suivre pour se mettre d'accord sur un secret commun en échangeant des messages publiques :

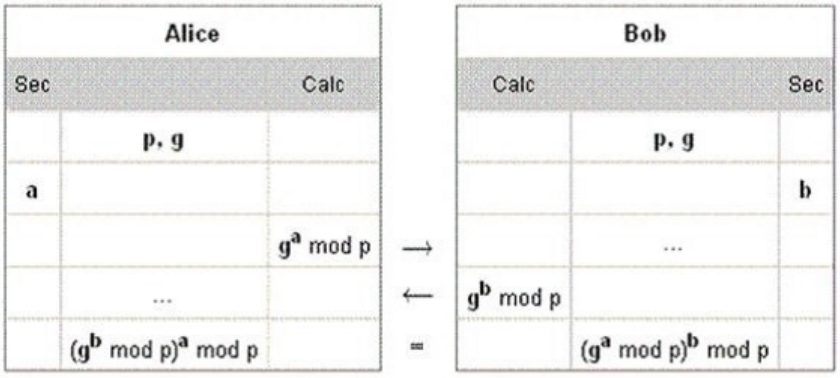


Image 12 : Algorithme Diffie-Hellman

Attention : Sécurité de Diffie-Hellman

Au final, Alice et Bob partagent le secret gab mod p, mais une tierce partie ayant intercepté les messages échanger entre Alice et Bob ne pourra calculer ce secret. En effet, cette tierce partie connaît ga mod p, et gb mod p, mais pour calculer gab mod p, il faut calculer a à partir de ga mod p, ou b à partir de gb mod p. Or personne ne sait comment calculer le logarithme discret.

3. Intégrité de données

Définition : Intégrité de donnée

C'est la propriété qui permet de vérifier qu'une données n'a pas été modifiée par une entité tierce (accidentellement ou intentionnellement).

Une fonction de hachage est typiquement utilisée pour vérifier l'intégrité de données.

Définition : Fonction de hashage cryptographique

Une fonction de hashage associe à une chaîne binaire (de longueur variable) une chaîne de longueur fixe. Une fonction de hashage cryptographique a les propriétés suivantes :

- Étant donné m, il est facile de calculer h(m)
- Étant donné h, il est difficile de calculer m tel que h(m)=h
- Étant donné m, il est difficile de trouver un autre message, m', tel que h(m)=h(m').

Méthode : Comment utiliser une fonction de hashage pour contrôler l'intégrité de données.

La figure ci-contre illustre comment utiliser une fonction de hashage pour vérifier l'intégrité d'un document numérique.

Initialement le code de hashage du document numérique est calculé et stocké dans un endroit sûre. Ultérieurement ce code est recalculé et comparé à celui qui a été stocké.

Si les deux valeurs sont égales alors le document n'a pas été modifié. Sinon, le document a subit une modification.

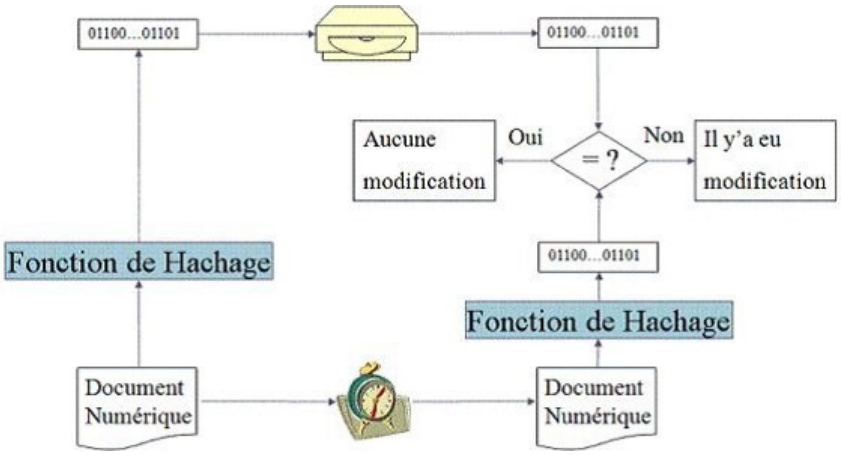


Image 13 : Intégrité de données

Exemple

Il existe plusieurs fonctions de hashages ; En voici quelques unes :

- MD2 (Message Digest 2) : Opère sur des bloques de 16 octets, manipule des mots de 8 bits Output 128 bits.
- MD4 (Message Digest 4) : Manipule des mots de 32 bits, plus performant sur des processeurs 32 bits.
- MD5 (Message Digest 5) : Une passe de plus / MD4, plus sûre
- SHA-1 (Secure Hash Algorithm) : Proposé par le NIST Input message 264 octets (au max), output 160 bits.

4. Authentification de l'origine de données

Définition : Authentification de l'origine

C'est la propriété qui permet de vérifier que la source de données est bien l'identité prétendue.

Définition : Message Authentication Code (MAC)

C'est est un mécanisme cryptographique qui permet de vérifier l'authenticité de l'origine des données et leur intégrité en même temps.

Un MAC est une famille de fonctions hk paramétrée par une clé secrète k avec les propriétés suivantes :

- Étant donné une clé k et un message m, hk(m) est facile à calculer,
- Étant donné zéro ou plusieurs paires (mj,hk(mj)), il est rès difficile de calculer n'importe quelle paire (m, hk(m)) pour n'importe quel message m.

Méthode : Comment utiliser un MAC pour garantir l'authentification de l'origine

Pour garantir d'authenticité de l'origine, l'émetteur et le récepteur doivent partager une clé symétrique.

Cette clé sera utilisé par l'émetteur pour calculer un MAC sur le message à envoyer. Ce MAC (code de hashage) est la preuve d'authenticité qui accompagnera le message.

Le récepteur utilisera la même clé secrète pour calculer le MAC de nouveau sur le message reçus. Le MAC nouvellement calculé sera comparé au MAC accompagnant le message. Si les deux valeurs sont égales alors le message et l'origine sont authentiques. Sinon, soit le message ou l'origine n'est pas authentique.

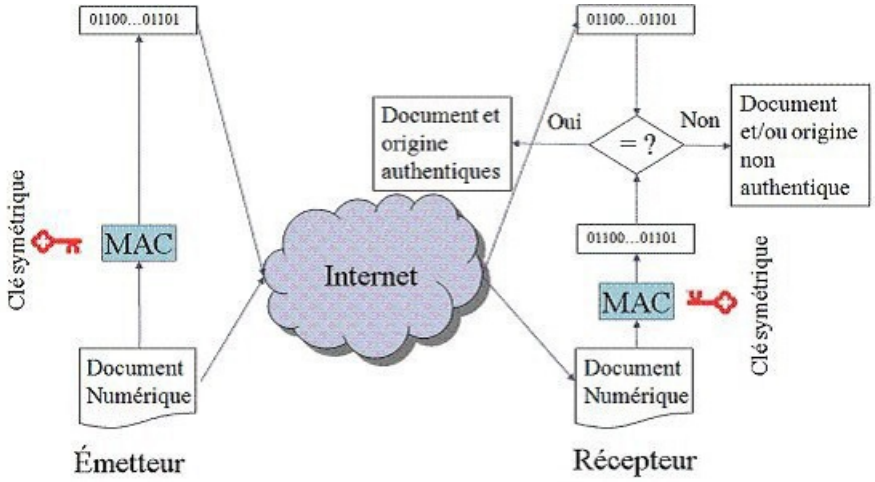


Image 14 : Authentification de l'origine

Exemple : Exemple de MAC

HMAC : Mihir Bellare, Ran Canetti, et Hugo Krawczyk 1996 FIPS PUB 198, RFC 2104

- HMAC-MD5
- HMAC-SHA-1

$$HMAC_k(m) = h((K + opad) \| h((K + ipad) \| m))$$

Opad = 0x5c5c5c5c...5c5c

Ipad = 0x363636...3636

5. Non-répudiation de l'origine

Définition : Non-répudiation de l'origine

La non répudiation de l'origine assure que l'émetteur du message ne pourra pas nier avoir émis le message dans le futur.

La signature digitale

La signature digitale est un mécanisme cryptographique qui permet d'assurer la non répudiation de l'origine.

Ce mécanisme repose sur un système cryptographique asymétrique

La signature est calculée en utilisant la clé privé de l'émetteur

La signature est vérifiée en utilisant la clé publique de l'émetteur

Méthode : Comment utiliser la signature digitale pour assurer la non-répudiation de l'origine ?

L'émetteur du message génère sa paire de clés (publique, privée). Il diffuse sa clé publique te maintient sa clé privée secrète. Pour signer un document l'émetteur commence par calculer le code hashage du document puis signe ce code de hashage avec sa clé privée. Le résultat de cette dernière opération (chiffrement avec clé privée dans le cas de RSA) est la signature digitale qui accompagnera le document. Quand le récepteur reçoit le message et la signature digitale, il recalcule le code de hashage, déchiffre la signature avec la clé publique de l'émetteur et compare les deux codes de hshages. Si les deux codes sont similaires alors la signature est valide.

L'émetteur ne pourra pas nier dans le futur avoir émis le message puisque y a que lui qui peut générer la signature digitale avec sa clé privée secrète.

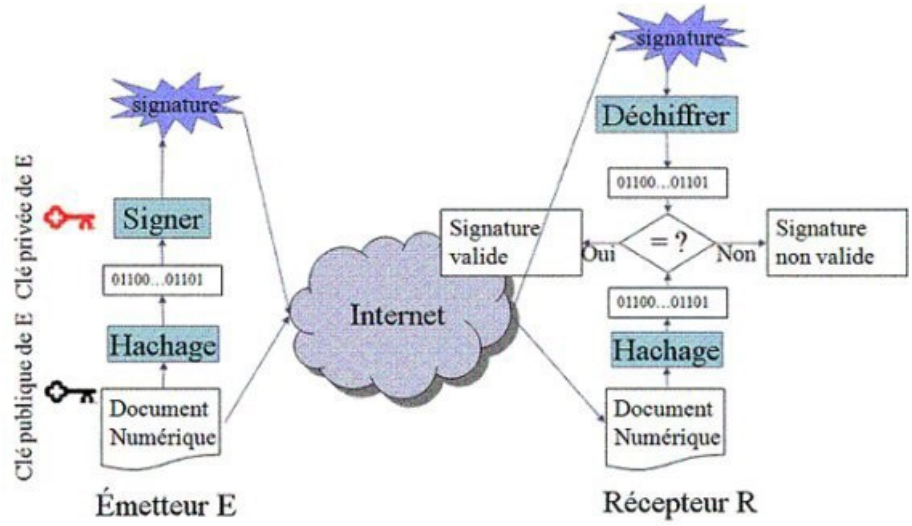


Image 15 : Signature digitale et non-répudiation

Exemple : Signature digitale avec RSA

Initialisation

- Choisir deux nombres premiers, p et q, les deux étant plus grands que 10100
- Calculer n = p · q (n est le modulus)
- Choisir e aléatoire tel que e et ((p - 1) · (q - 1)) n'aient aucun facteur commun excepté 1
- Trouver d tel que : ed = 1 mod((p - 1)(q - 1)).
- Clé publique : (n,e).
- Clé privée : (n,d) ou (p,q,d) si on désire garder p et q.

Signature digitale

- L'expéditeur crée la signature s à partir du message m : s = md mod(n), où (n,d) est la clé privée de l'expéditeur.
- Le destinataire reçoit s et m et effectue la vérification de m : m = se mod(n), où (n,e) est la clé publique de l'expéditeur.

RSA est connu pour être très lent, ayant une clé très longue.

6. Historiques

Quelques fais marquants de l'histoire de la cryptographie

- 50 av. JC. : Julius Cesar utilise une simple substitution de l'alphabet pour les communications gouvernementales
- 1918 : Gilbert Vernam, mathématicien américain, inventa le one-time pad, l'algorithme de chiffrement le plus sûr jusqu'à aujoud'hui, mais impratiquable
- 1923 : Dr. Albert Scherbius, hollondais résidant en Allemagne, met au point la machine Enigma qui sert à encoder des messages. Le prix très cher en fait un échec.
- 1925 : La marine de guerre allemande reprend le projet Enigma en le confiant au Chiffrierstelle, le service de chiffrement
- 1937 : Enigma M3 est adoptée par le Wehrmacht, l'armée allemande
- 1939 : début de la seconde guerre mondiale, où des milliers de scientifiques britanniques, polonais et français travaillaient pour solutionner Enigma, et les milliers de messages chiffrés. L'équipe de Alan Turing trouva la solution
- 1976 : IBM publie un algorithme de chiffrement basé sur Lucifer. Il devient le DES (Data Encryption Standard)
- 1976 : Whitfield Diffie et Martin Hellman introduisent l'idée d'un système à clé publique
- 1978 : l'algorithme de chiffrement à clé publique RSA est publié par Rivest, Shamir et Adleman
- 1978 : Le RC4 est développé par Ronald Rivest pour la RSA Security et sera gardé secret jusqu'en
- 1994, où l'algorithme est rendu public anonymement dans une liste de distribution de Cypherpunks
- 1991 : Phil Zimmermann rend disponible sa première versio de PGP
- 1992 : IDEA est inventé en Suisse par Xuejia Lai et James Massey
- 1992 : MD5 est développé par Ronald L. Rivest
- 1994 : Ron Rivest, déjà auteur de RC2 et RC4, publie RC5
- 2000 : Rijndael devient l'AES, le standard du chiffrement avancé

7. La libraire OpenSSL

Le projet OpenSSL

OpenSSL (http://www.openssl.org) est une libraire qui compte 60.000 lignes de code (langage C). Elle est utilisées par de nombreuses applications; openssh, apache+mod_ssl,... Elle est fondé sur la bibliothèque cryptographique SSLeay d'Eric Toung et Tim Hudson. L'objectif initial de cette librairie était la mise en oeuvre des protocoles SSL et servir comme bibliothèque cryptographique.

La bibliothèque OpenSSL

La librairie OpenSSL en décline en deux formes :

1. Interface de programmation en C
    - Bibliothèque SSL/TLS (libssl.a)
        - Mise en oeuvre des protocoles SSLv2, SSLv3, TLSv1
    - Bibliothèque cryptographique

- Cryptographie clé publique et certificats X509: RSA, DSA, DH
- Chiffrement: DES, 3DES, Blowfish, RC2, IDEA, RC4, + modes ECB, CBC,CFB,OFb pour les algorithmes par blocs
- Hachage: MD2, MD4, MD5, SHA1, MDC2, RIPEMD160

2. Suite d'applications en ligne de commande openssl(1)

    - Un ensemble de commandes permettant de réaliser les différentes opérations cryptographiques