Chapitre 3 : Cryptographie et sécurité de la carte à puce

1.1. Introduction

La sécurité et la protection de la vie privée sont des éléments clés de l'industrie. Elle a donné naissance à une importante activité économique qui touche la fabrication de composants électroniques, le développement de logiciels, les processus de personnalisation et de distribution, les machines-outils, les périphériques, le développement d'applications, les tests de conformité, les tests de sécurité, etc.

Dans ce chapitre on va donner quelques éléments essentiels à la compréhension de la cryptologie, et les méthodes principales de cryptographie utilisées pour assurer la sécurité des cartes à puces, avec un aperçu sur les attaques qui cherchent de nuire la sécurité afin d'accéder au contenu.

1.2. Principes de la cryptographie

La cryptographie est l'art de chiffrer et déchiffrer les messages échangés entre un émetteur et un récepteur.

Le chiffrement des messages consiste à transformer une information à l'aide d'une convention secrète. La fonction de transformation constitue l'algorithme cryptographique, dont le secret réside dans des paramètres appelés clés. Lorsque l'on déchiffre le message, on réalise l'opération inverse en connaissant ces clés. Dans les cartes à puces, la cryptographie met en œuvre divers mécanismes qui ont pour but d'assurer soit la confidentialité des informations, soit l'authentification des cartes ou des utilisateurs, soit encore la signature des messages. L'ensemble des moyens mettant en œuvre la cryptographie forme un cryptosystème. Seules les cartes dotées d'un crypto processeurs permettent de gérer le chiffrement asymétrique. II en existe trois catégories selon qu'ils sont symétriques, asymétriques ou « à apport nul de connaissance ».

1) **Symétrique** qui utilise la même clé pour le chiffrement et le déchiffrement (DES, AES)

2) **Asymétrique** avec une clé publique et une clé privée générées par une procédure mathématique comme dans l'algorithme RSA

3) **A apport nul de connaissance**

1.3. La cryptographie classique

Dans cette section on va aborder quelques types de la cryptographie classique :

1) Le chiffrement par substitution

2) Le chiffrement par transposition

1.3.1. Chiffrement par substitution

Dans la méthode de chiffrement par substitution, à chaque lettre ou groupe de lettres on substitue une autre lettre ou un autre groupe de lettres. La substitution simple (substitution mono alphabétique). Nous identifions quatre types de chiffrement par substitution.

1.3.1.1. Chiffrement de César

Le chiffrement de César consiste à décaler les lettres de 3 positions.

Exemple

Texte en claire : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Texte chiffré : D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Le message " hello wold " devient « khoorzrog »

- Il n'existe que 26 décalages possibles
- Le chiffrement de César est très vulnérable à l'analyse des fréquences

1.3.1.2. Chiffrement par substitution

À chaque lettre de l'alphabet en clair, on associe une lettre au hasard.

Exemple

Texte en claire : a b c d e f g h i j k l m n o p q r s t u v w x y z

Texte chiffré : U X O M P S N B E A Z Q W D G V K H C R T Y I F J L

a	b	c	d	e	f	g	h	i	g	k	l	m	n	o	p	k	r	s	t	u	v	w	x	y	z
U	X	O	M	P	S	N	B	E	A	Z	Q	W	D	G	V	K	H	C	R	T	Y	I	F	J	L

Le message " hello wold " devient « BPQQG IGHQM »

1.3.1.3. Substitution polyalphabétique

Le principe consiste à remplacer chaque lettre du message en clair par une nouvelle lettre prise dans ou plusieurs alphabets aléatoires associés. Par exemple, on pourra utiliser n substitutions monoalphabétiques ; celle qui est utilisée dépend de la position du caractère à chiffrer dans le texte en clair. On choisit une clé qui sert d'entrée dans la grille polyalphabétique incluant autant de symboles qu'il y a de lettres différentes à chiffrer. Chaque caractère de la clé désigne une lettre particulière dans la grille de codage. Pour coder un caractère, on doit lire le caractère correspondant du texte en clair en utilisant la grille polyalphabétique et le mot clé associé dans l'ordre séquentiel (on répète la clé si le chiffrement se fait par la méthode de VIGENERE et de BEAUFORT. L'illustration la plus simple qui corresponde à ce principe est l'utilisation d'une fonction à base de ou exclusif (XOR).

1.3.1.4. Substitution mono-alphabétique

Nous avons vu que le chiffrement de César présente une sécurité très faible, la principale raison est que l'espace des clés est trop petit : il y a seulement 26 clés possibles, et on peut attaquer un message chiffré en testant toutes les clés à la main.

Au lieu de faire correspondre circulairement les lettres, on associe maintenant à chaque lettre une autre lettre (sans ordre fixe ou règle générale).

Par exemple :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	Q	B	M	X	I	T	E	P	A	L	W	H	S	D	O	Z	K	V	G	R	C	N	Y	J	U

Pour crypter le message

ETRE OU NE PAS ETRE TELLE EST LA QUESTION

On regarde la correspondance et on remplace la lettre **E** par la lettre **X**, puis la lettre **T** par la lettre **G**, puis la lettre **R** par la lettre **K**...

Le message crypté est alors :

XGKX DR SX OFV XGKX GXWWX XVG WF ZRXVGPDS

Pour le décrypter, en connaissant les substitutions, on fait l'opération inverse.

Avantage : nous allons voir que l'espace des clés est gigantesque et qu'il n'est plus question d'énumérer toutes les possibilités.

Inconvénients : la clé à retenir est beaucoup plus longue, puisqu'il faut partager la clé constituée des 26 lettres.

1.3.1.5. Espace des clés

Mathématiquement, le choix d'une clé revient au choix d'une bijection de l'ensemble $\{A, B, \ldots, Z\}$ vers le même ensemble $\{A, B, \ldots, Z\}$. Il y a 26! choix possibles. En effet pour la lettre A de l'ensemble de départ, il y a 26 choix possibles (nous avions choisi F), pour B il reste 25 choix possibles (tout sauf F qui est déjà choisi), pour C il reste 24 choix... enfin pour Z il ne reste qu'une seule possibilité, la seule lettre non encore choisie. Au final il y a $26 \times 25 \times 24 \times \ldots \times 2 \times 1$ soit 26! choix de clés. Ce qui fait environ 4×10^{26} clés. Il y a plus de clés différentes que de grains de sable sur Terre ! Si un ordinateur pouvait tester 1 milliard de clés par seconde, il lui faudrait alors 12 milliards d'années pour tout énumérer.

1.3.1.6. Attaque statistique

La principale faiblesse du chiffrement mono-alphabétique est qu'une même lettre est toujours chiffrée de la même façon. Par exemple, ici E devient X. Dans les textes longs, les lettres n'apparaissent pas avec la même fréquence. Ces fréquences varient suivant la langue utilisée. En français, les lettres les plus rencontrées sont dans l'ordre :

E S A I N T R U L O D C P M V Q G F H B X J Y Z K W

avec les fréquences (souvent proches et dépendant de l'échantillon utilisé) :

La répartition des lettres en français est donnée est dans le tableau ci-dessous :

Tableau 3. 1 : Fréquence des lettres en français

E	S	A	I	N	T	R	U	L	O	D
14.69%	8.01%	7.54%	7.18%	6.89%	6.88%	6.49%	6.12%	5.63%	5.29%	3.66%

Méthode d'attaque : dans le texte crypté, on cherche la lettre qui apparaît avec un grand pourcentage, et si le texte est assez long cela devrait être le chiffrement du E, la lettre qui apparaît ensuite dans l'étude des fréquences devrait être le chiffrement du S, puis le chiffrement du A... On obtient des morceaux de texte clair sous la forme d'un texte à trous et il faut ensuite deviner les lettres manquantes.

Exemple

Soit à déchiffrer le message suivant par l'attaque statistique :

LHLZ HFQ BC HFFPZ WH YOUPFH MUPZH

On compte les apparitions des lettres :

H : 6 **F** : 4 **P** : 3 **Z** : 3

On suppose donc que le **H** crypte la lettre **E**, le **F** la lettre **S**, ce qui donne

***E** ES* ** ESS** *E ***SE ****E**

D'après les statistiques **P** et **Z** devraient se décrypter en **A** et **I** (ou **I** et **A**). Le quatrième mot "**HFFPZ**", pour l'instant décrypté en "**ESS****", se complète donc en "**ESSAI**" ou "**ESSIA**". La première solution semble correcte ! Ainsi **P** crypte

A, et **Z** crypte **I**. La phrase est maintenant :

***E*I ES* ** ESSAI *E ***ASE **AIE**

En réfléchissant un petit peu, on décrypte le message :

CECI EST UN ESSAI DE PHRASE VRAIE

1.3.2. Chiffrement par transposition

Avec le principe de la transposition toutes les lettres du message sont présentes, mais dans un ordre différent. Il utilise le principe mathématique des permutations. Plusieurs types de transpositions existent dans la littérature, nous citons :

1.3.2.1. Transposition simple par colonnes

On écrit le message horizontalement dans une matrice prédéfinie, et on trouve le texte à chiffrer en lisant la grille verticalement. Le destinataire légal pour décrypter le message réalise le procédé inverse.

Exemple

Soit le texte à chiffrer ‘’I LOVE MY ENGLISH TEACHER’’ utilise une matrice [6× 4]

I	L	O	V
E	M	Y	E
N	G	L	I
S	H	T	E
A	C	H	E
R			

Figure 3. 1 : Transposition simple par colonnes

Le texte chiffré est : IENSA RLMGH COYLT HVEIE E

1.3.2.2. Transposition complexe par colonnes

Dans ce type de chiffrement, on modifie l’ordre des lettres d’un texte en clair.

Exemple

Soit à chiffrer le message « hello wold » avec la clé 213.

avec la clé 213 le message « hello wold » devient « eolhlolwd ».

2	1	3
h	e	L
l	o	w
o	l	d

Figure 3. 2 : Texte chiffré avec la clé 213

1.4. Le chiffrement symétrique

Les systèmes symétriques sont synonymes de systèmes à clés secrètes. Une même clé est utilisée pour le chiffrement et le déchiffrement, d’où l’obligation que celle-ci reste confidentielle.

Sur la Figure (3.3.) l’émetteur (Alice) et le destinataire (Bob) doivent se mettre d’accord préalablement sur la clé (k) à utiliser, pour ceci ils ne doivent pas utiliser le réseau de communication standard qui est susceptible d’être espionné (par Oscar). Chaque fois qu’Alice

veut transmettre un message (m) à Bob, elle utilise sa clé secrète (K) et une fonction de chiffrement (E) pour chiffrer ($C = E(m)$), et elle envoie le résultat de ce chiffrement par l'intermédiaire du même canal. Bob utilise à son tour la même clé secrète et le même algorithme public pour déchiffrer le message codé qu'il a reçu.

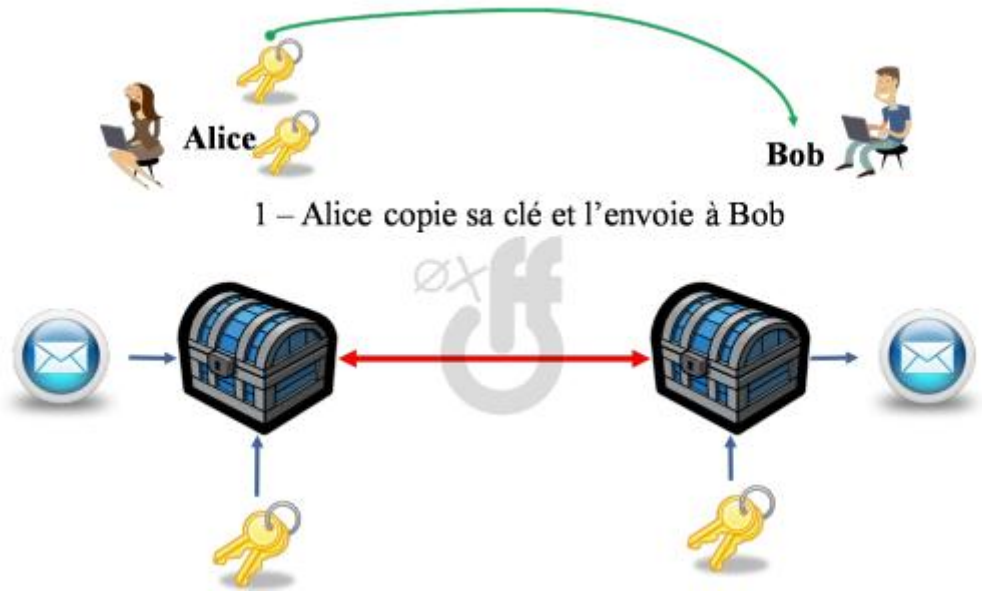

Figure 3. 3 : Schéma illustrant Alice et Bob échangent leur message

Les problèmes de cette technologie sont les suivants :

- Si la clé secrète est compromise (volée, extorquée, piratée, ...) par un opposant, alors ce dernier pourra déchiffrer tous les messages encodés avec celle-ci. Oscar peut même se faire passer pour Alice ou Bob.
- Les clés doivent être distribuées secrètement : c'est très difficile à l'échelle planétaire (se rencontrer, utiliser un messager sûr, etc...).
- Si une clé différente est utilisée pour chaque paire différentes d'utilisateurs du réseau, le nombre total des clés augmente très rapidement en fonction du nombre total d'utilisateurs.

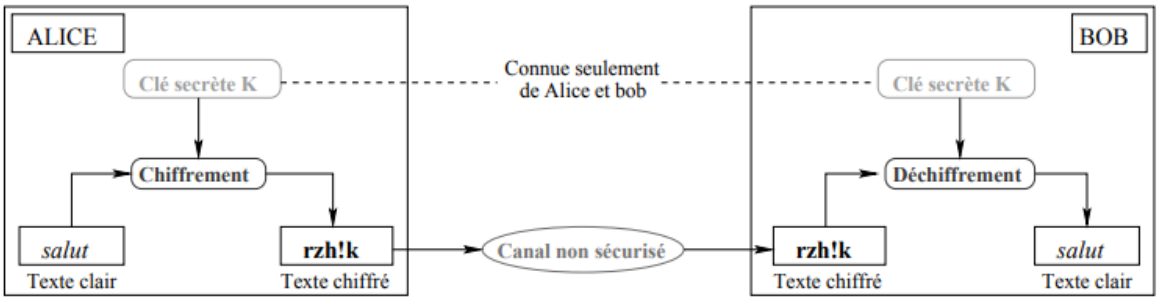

Figure 3. 4 : Schéma illustrant le fonctionnement du chiffrement symétrique

Cependant, en pratique, il se pose un problème majeur : comment échanger de manière sûre les clés ? C'est le principal défaut de ce système.

1.4.1. La cryptographie pour cartes à puces

Les algorithmes de la cryptographie symétrique les plus utilisés dans les cartes à puce sont DES (Data Encryption Standard), le triple DES, 3DES et AES (Advanced Encryption Standard).

Le DES naît en 1975 suite à une requête d'IBM en 1960 pour son programme de recherche sur le chiffrement informatique. Au début, les spécialistes de la NSA (National Security Agency, le service de sécurité intérieure américain) se cassent les dents dessus donc IBM est contraint de l'utiliser sous une forme plus simple que prévu. L'utilisation du D.E.S. se généralise alors peu à peu dans les administrations américaines. Depuis, le D.E.S. est remis à niveau tous les 5 ans environ pour faire face à la puissance croissante des ordinateurs qui le mettent en péril.

1.4.2. Classes de chiffrements symétriques

On distingue deux catégories de chiffrement symétrique :

1.4.2.1. Les chiffrements symétriques par blocs

Le chiffrement par bloc (en anglais block cipher) est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique, l'autre étant le chiffrement par flot. La principale différence vient du découpage des données en blocs de taille généralement fixe. La taille de bloc est comprise entre 32 et 512 bits, dans le milieu des années 1990 le standard était de 64 bits mais depuis 2000 et le concours AES le standard est de 128 bits.

En cryptographie, un mode d'opération est la manière de traiter les blocs de texte clairs et chiffrés au sein d'un algorithme de chiffrement par bloc. Historiquement, les modes d'opération ont été abondamment étudiés pour leurs propriétés de propagation d'erreurs lors de divers scénarios de modification de données durant le chiffrement. Les développements suivants ont considéré que la protection de l'intégrité était un objectif à atteindre par des moyens complètement différents. Mais aujourd'hui il existe des modes d'opérations qui associent chiffrement et authentification de manière efficace.

Plusieurs modes existent, certains sont plus vulnérables que d'autres :

- Dictionnaire de codes (Electronic Code Book, ECB)
- Enchaînement des blocs (Cipher Block Chaining, CBC)
- Chiffrement à rétroaction (Cipher Feedback, CFB)
- Chiffrement à rétroaction de sortie (Output Feedback, OFB)
- Chiffrement basé sur un compteur (CounTeR, CTR)
- Chiffrement avec vol de texte (Cipher Text Stealing, CTS)

Exemple d'algorithmes : DES, AES, IDEA, RC6, BLOWFISH, …

1.4.2.2. Aspect technique de l'algorithme DES

- DES utilise une clé K de 56 bits utiles, qui est codée sur 64 bits et dont les bits 8, 16, 24, 32, 40, 48, 56 et 64 sont utilisés comme code de détection d'erreur et correcteur de la clé. DES est un crypto-système par blocs qui opère sur des blocs de 64 bits.

- La clé du système DES est trop courte pour les puissances de calcul actuelles. La taille de la clé secrète est de 56 bits ce qui la rend aujourd'hui vulnérable aux attaques par force brute.

Le message, au préalable converti en binaire, est découpé en blocs Bi de 64 bits. La clé K, comporte 56 bits. Pour chaque bloc Bi, on applique l'algorithme suivant :

1) On effectue une permutation initiale des bits du bloc Bi. On appelle alors G0 et D0 les parties de 32 bits droite et gauche du bloc obtenu.

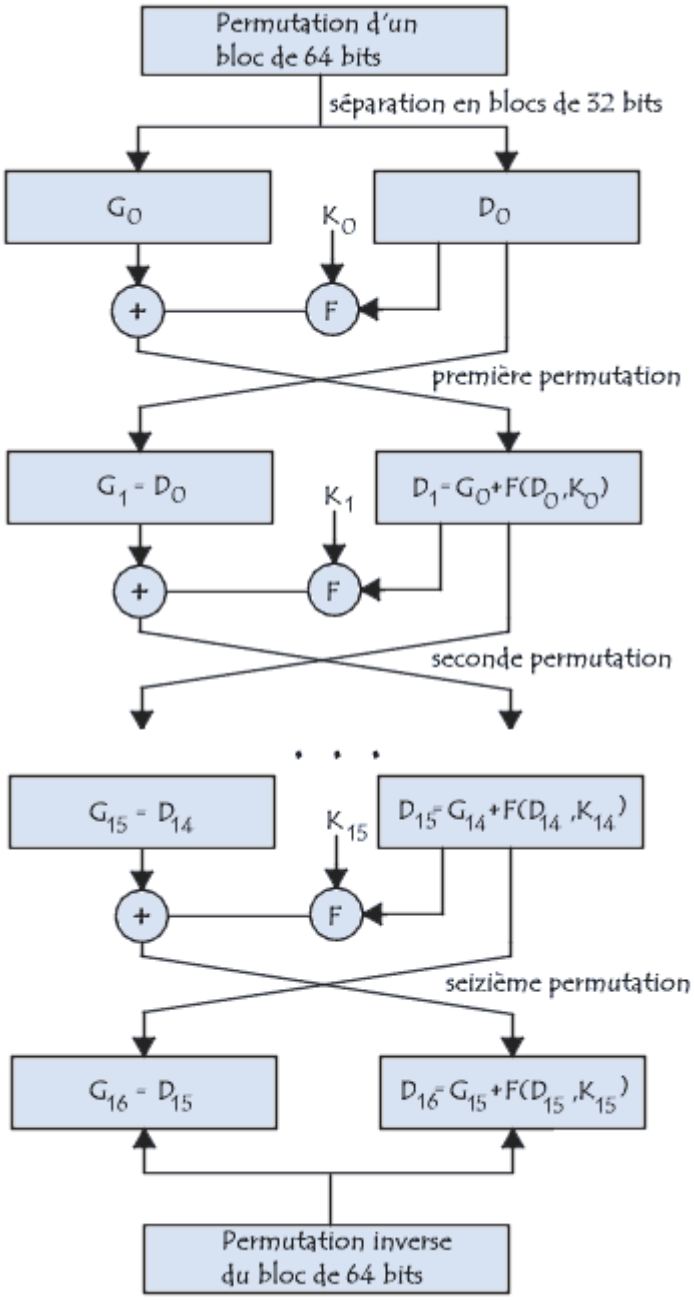

Figure 3. 5 : Mécanisme de l'algorithme DES

2) On répète 16 fois la procédure suivante :

$Gi = Di - 1$

Di = Gi-1 XOR F(Di-1 , Ki-1) (XOR est représenté par $\oplus$ sur le schéma de la Figure (3.5))

où Ki est un bloc de 48 bits de la clé K, et f une fonction composée successivement d'une expansion de bits, d'un XOR, d'une réduction de bits, et d'une permutation de bits.

1) On recompose un bloc B'16 en "recollant" D16 et G16 dans cet ordre.

2) On effectue la permutation inverse de la permutation initiale 1).

Le schéma résumant les différentes parties de l'algorithme est donné sur la Figure (3.5) :

Le décodage se fait en utilisant la même clé K mais en déroulant l'algorithme dans le sens inverse.

- Inconvénients

Les algorithmes de chiffrement de type DES sont fortement menacés par les puissances de calcul des ordinateurs. Il n'est en effet pas impossible de balayer la plupart des clés pour casser le code. Un nouveau système, le AES (Advanced Encryption Standard) est prévu pour le remplacer.

1.4.2.3. Aspect technique de l'algorithme triple DES

Le triple DES noté aussi 3DES ou TDES est un algorithme de chiffrement symétrique par bloc utilisant trois fonctions successives de l'algorithme DES sur le même bloc de données de 64 bits. L'algorithme utilise trois clé DES (K1, K2, K3), chacune de 56 bits. Deux configurations d'utilisation sont possibles qui sont : La première est 3DES avec une clé à 168 bits. Dans ce cas, les trois clés sont toutes différentes. La deuxième utilisation est 3DES avec une clé de 112 bits où les clés K1 et K3 sont identiques. Dans le chiffrement avec 3DES l'utilisation de l'algorithme DES se fait dans l'ordre suivant : DES (avec la clé K1), DES^{1}(avec la clé K2) et DES (avec la clé K3) comme le montre la Figure (3.6) :

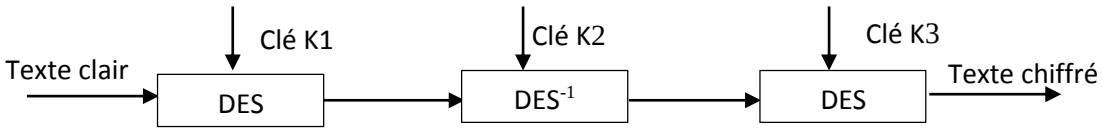

Figure 3. 6 : Schéma global du chiffrement 3DES

1.4.2.4. Décryptage DES

Le déchiffrement avec 3DES a la même structure que le chiffrement sauf que l'utilisation de l'algorithme DES se fait dans l'ordre suivant : DES^{-1}(avec la clé K_3), DES (avec la clé K_2) et DES^{-1}(avec la clé K_1).

1.4.2.5. Aspect technique de l'algorithme AES

L'Advanced Encryption Standard a fait l'objet d'un appel d'offre datant de 1997. Il s'agissait de remplacer le DES dont la taille des clés (56 bits) était devenue trop petite pour les performances des ordinateurs modernes. Les spécifications étaient une longueur de blocs de 128 bits (ou de 256 bits) et une longueur de clé paramétrable : 128 ou 192 ou 256 bits. Parmi les 15 candidats, le candidat retenu (en 2000) se nomme RIJNDAEL (mais on l'appelle simplement l'AES). Il est dù à deux chercheurs Belges, Rijmen et Daemen.

AES est un algorithme de chiffrement par blocs, les données sont traitées par blocs de 128 bits pour le texte clair et le chiffré. La clef secrète a une longueur de 128 bits, d'où le nom de version : AES 128 (il existe deux autres variantes dont la clef fait respectivement 192 et 256 bits). Le schéma illustrant le chiffrement et le déchiffrement AES est donnée sur la figure ci-dessous :

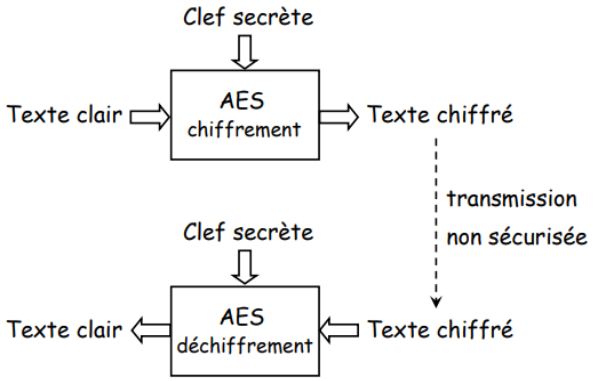

Figure 3. 7 : Schéma illustrant le chiffrement et le déchiffrement

L'algorithme prend en entrée un bloc de 128 bits (16 octets), la clé fait 128, 192 ou 256 bits. Les 16 octets en entrée sont permutés selon une table définie au préalable. Ces octets sont ensuite placés dans une matrice de 4x4 éléments et ses lignes subissent une rotation vers la droite. L'incrément pour la rotation varie selon le numéro de la ligne. Une transformation linéaire est ensuite appliquée sur la matrice, elle consiste en la multiplication binaire de chaque élément de la matrice avec des polynômes issus d'une matrice auxiliaire, cette multiplication est soumise à des règles spéciales selon GF(28) (Groupe de Galois ou corps fini). La transformation linéaire garantit une meilleure diffusion (propagation des bits dans la structure) sur plusieurs tours. Finalement, un XOR entre la matrice et une autre matrice permet d'obtenir

une matrice intermédiaire. Ces différentes opérations sont répétées plusieurs fois et définissent un « tour ». Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours.

1.4.2.6. Le chiffrement symétrique par flot

Le chiffrement de flux ou chiffrement par flot (en anglais stream cipher). Un chiffrement par flot arrive à traiter les données de longueur quelconque et n'a pas besoin de les découper. Un chiffrement par flot se présente souvent sous la forme d'un générateur de nombres pseudo-aléatoires avec lequel on opère un XOR entre un bit à la sortie du générateur et un bit provenant des données.

Exemple d'algorithmes : RC4, Bluetooth E0/1, GSM A5/1

1.5. Chiffrement asymétrique

Ce type de cryptographie est basé sur deux clés séparées et distinctes : l'une publique et l'autre maintenue secrète ou privée. L'algorithme asymétrique est utilisé pour assurer :

- La confidentialité : seul le propriétaire de la clé privée pourra lire le message chiffré avec la clé publique correspondante ;
- La non-altération et la non-répudiation : seul le propriétaire de la clé privée peut signer un message pour produire une signature. Cette dernière une fois déchiffrée avec la clé publique prouvera l'authenticité du message. Cette propriété est très utilisée dans les cartes à puce.

L'algorithme de la cryptographie asymétrique le plus utilisé dans les cartes à puce est le RSA. Ce dernier est nommé par les initiales de ses trois inventeurs : Rivest- Shamir - Adelman. Dans ce qui suit, la méthode d'utilisation de l'algorithme est détaillée en commençant par la méthode de la génération des clés RSA et ensuite la méthode d'utilisation de l'algorithme de cryptage et de décryptage.

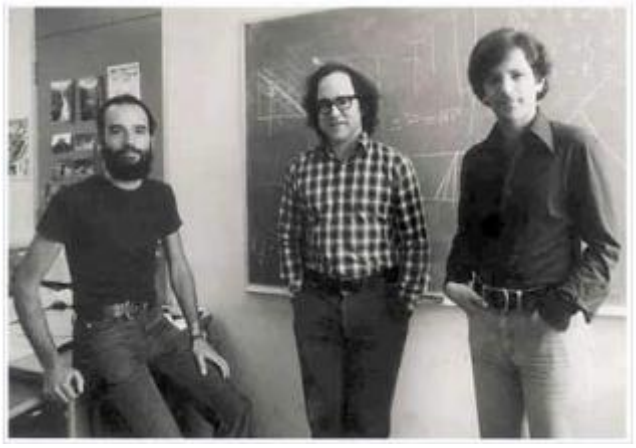

Figure 3. 8 : L'équipe RSA en 1977

Le RSA est basé sur la théorie des nombres premiers, et sa robustesse tient du fait qu'il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers. Alors qu'il est facile de multiplier deux nombres premiers, il est très difficile de retrouver ces deux entiers si l'on en connaît le produit.

Dans RSA, les blocs de message sont représentés par des entiers compris entre 0 et n-1. Pour envoyer un message m à Bob, Alice va donc chercher la clé publique de Bob et elle calcule le message chiffré c correspondant par l'utilisation de la formule : $c = m^e \bmod n$.

Lorsqu'il reçoit le message chiffré c, Bob retrouve le texte clair en calculant $c^d \bmod n = m$.

Le schéma illustrant le fonctionnement de ce type de chiffrement est donné sur la figure ci-dessous :

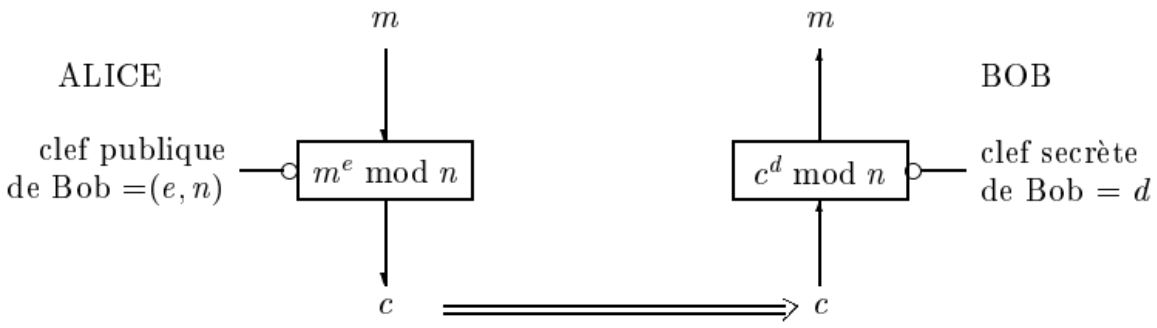

Figure 3. 9 : Chiffrement à clé publique

1.5.1. Comment fonctionne un algorithme RSA ?

Actuellement le chiffrement RSA est automatisé. Cependant le chiffrement peut être effectué avec les fonctions ci-après :

Petit rappel : Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs (qui sont alors 1 et lui-même, par exemple 5,7,13...)

1.5.1.1. Génération des clés RSA

A partir de deux grands nombres premiers aléatoirement choisis : p et q. Quatre autres valeurs sont calculées de la façon suivante :

1. Le produit $n = p \times q$, est appelé le modulo de chiffrement ;

2. La valeur de l'indicatrice d'Euler en n est définie par le produit :

$\varphi(n) = (p - 1) \times (q - 1)$;

3. L'entier naturel **e** est choisi premier avec $\varphi(n)$ appelé exposant de chiffrement ;

4. L'entier naturel **d** est l'inverse de **e** modulo $\varphi(n)$ appelé exposant de déchiffrement ; d peut se calculer efficacement par l'algorithme d'Euclide étendu.

Le couple (n, e) définit la clé publique et le couple (n, d) définit la clé privée associée.

1.5.1.2. Exemple de génération de clés

Par exemple, Alice peut diffuser sa clé publique (e, n) sur Internet.

Quand Bob voudra envoyer un message secret à Alice, il cherche la clé publique d'Alice (e, n) sur Internet et crypte son message avec la clé publique d'Alice : $c = m \times e \pmod{n}$.

p, q et d constituent la clé privée d'Alice. Seule Alice connait p, q, et d.

Un tiers, Carole, ne peut pas comprendre ce que Bob a écrit à Alice car Carole n'a pas la clé privée. Quand Alice reçoit le message de Bob, elle le décrypte en utilisant sa clé privée d, n en effectuant $cd \pmod{n} = m$.

Si l'on ne connait pas d, on ne peut pas décrypter le message crypté c et connaitre le message m.

Pour connaitre d, on doit connaitre $(p-1)\times(q-1)$ pour trouver d à partir de l'équation :

$$e \times d \equiv 1 \pmod{(p-1)\times(q-1)}.$$

Or, pour avoir $(p-1)\times(q-1)$, on doit en premier lieu être capable de factoriser le grand nombre n en p et q, tous deux premiers.

A partir du moment où tous les nombres sont de très grands nombres (100 chiffres au moins), nous pouvons dire qu'il est impossible en pratique qu'un tiers obtienne d, et puisse donc décrypter le message crypté.

Exemple d'application

Les variables étant données : p = 29, q = 31, e = 13, m = 123 ;

==>Nous calculons : $n = p \times q = 899$

(p-1)×(q-1) = 840

d = 517 car e×d = 13×517 = 8×(p-1)×(q-1) + 1

Pour crypter,

$c = 123^{13} \pmod{899} = 402$

Et pour décrypter, $m = 402^{517} \pmod{899} = 123$

Le chiffrement RSA est purement mathématique, tout message doit d'abord être codé par des nombres entiers (n'importe quel encodage : ASCII, Unicode, voire A1Z26).

1.5.2. Comment déchiffrer le RSA sans connaitre la clé privée ?

Le RSA est basé sur la théorie des nombres premiers, et sa robustesse tient du fait qu'il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers.

Pour retrouver la clé privée, un pirate doit être en mesure de réaliser la décomposition en facteurs premiers du nombre n pour retrouver ses 2 facteurs p et q.

Alors qu'il est facile de multiplier deux nombres premiers, mais il est très difficile de retrouver ces deux entiers si l'on en connaît le produit.

Remarque

La donnée de n est la clé publique : elle suffit pour chiffrer. Pour déchiffrer, il faut connaitre p et q, qui constituent la clé privée. Le problème de factorisation de grands entiers étant très

difficile, la connaissance de la clé publique n ne permet pas de retrouver les entiers p et q, qui constituent la clé secrète.

1.5.3. Les défauts du RSA

Le RSA présente quelques défauts qui sont :

Premièrement, sa sécurité n'a jamais été démontrée mathématiquement.

L'idée que le noyau mathématique du RSA est inviolable est une conclusion expérimentale tirée de l'échec des tentatives connues de décryptage sans la clef secrète depuis sa création en 1977.

Mais il est possible que l'on puisse casser le RSA à l'aide d'algorithmes spécifiques de factorisation, voire d'ordinateurs quantiques, et il n'est même pas sûr que casser le RSA revienne à factoriser n = p×q, peut-être est-il possible de déchiffrer un message, c'est-à-dire d'obtenir d, sans passer par p et q.

Deuxièmement, si une organisation avait réussi à casser des clés de plus de 512 bits RSA, elle le garderait certainement pour son pays et l'information ne circulerait pas forcément dans le monde. Donc la confiance qu'on lui accorde ne doit pas être une confiance aveugle.

Troisièmement, les conseils pour ne laisser transparaître aucune faille de sécurité dans le RSA se font de plus en plus nombreux avec les années (principalement pour le choix de p, q et e).

N'oublions pas que le 22 Août 1999, une clé RSA de 512 bits a été cassée : un entier n de 155 chiffres décimaux a été factorisé.

Citons enfin le cas du "RSA for paranoids" (paranoïaques) proposé par A. Shamir (un des inventeurs du RSA) en 1995, qui était censé rassurer les utilisateurs du RSA classique qui craignaient des failles de sécurité. "RSA for paranoids" permettait d'allonger la clé sans trop augmenter les coûts de codage.

Résultat : trois ans plus tard, ce prétendu renforcement du RSA est cassé.

Ainsi, un des inventeurs du RSA, croyant renforcer son système pour rassurer les paranoïaques, a, au contraire, augmenté leurs craintes.

L'algorithme de la cryptographie asymétrique le plus utilisé dans les cartes à puce est le RSA.

Ce système est beaucoup plus sûr qu'un chiffrement à simple clé de par sa conception, mais il présente quelques inconvénients. : Les temps de traitement sont plus longs, et pour un niveau de sécurité équivalent, nécessite des clés plus longues.

1.6. Crypto systèmes à apport nul de connaissance

Crypto système à apport nul de connaissance : Zero knowldege repose sur une idée introduite par Goldwasser, Micali et Rackoff en 1985, et optimisé par Guillou et Quisquater en 1987 pour les cartes à microcontrôleur.

Dans le système précédent, il est nécessaire qu'un secret de base soit utilisé par le vérifieur (cas de système symétrique) ou par le prouveur (cas de système asymétriques). Dans le système à apport nul de connaissance, le vérifieur n'a pas besoin de secret et le prouveur possède un secret. Il s'agit d'une méthode très puissante pour authentifier et signer des messages, sans donner la moindre information sur le secret utilisé.

1.7. Sécurité des cartes à puces

La sécurité est la principale qualité de la carte à puce. C'est d'ailleurs pour cette raison qu'elle a été choisie pour les transactions bancaires. Cette sécurité accrue est rendue possible grâce à un ensemble de techniques variées.

L'ISO 78 a défini 6 services de sécurité :

1) Authentification (de la source et/ou du destinataire) ;
2) Contrôle d'accès (qui nécessite une authentification préliminaire) ;
3) Confidentialité des données (les données illicitement récupérées doivent être inutilisables);
4) Intégrité des données (empêcher les modifications des données, les doublons) ;
5) Non-répudiation (un message, son envoi et sa réception ne peuvent être contestés) ;
6) Protection de l'analyse du trafic (la relation entre deux personnes doit rester secrète).

1.8. Les attaques

Le but d'un attaquant est de pouvoir accéder aux informations et aux secrets contenus dans la carte (code PIN, clé(s) secrète(s) cryptographique(s), etc...) ou tout simplement de nuire aux applications embarquées afin de tester le niveau sécuritaire de la carte.

Les attaques sont de deux types : physiques et logicielles. Dans cette section on va étudier les deux types d'attaques qu'il s'agit matériels et logiciels.

1.8.1. Attaques matérielles ou physiques

Une attaque physique est une attaque menée sur la partie électronique de la carte. En effet, les algorithmes cryptographiques sont souvent implémentés sur des modules physiques que l'attaquant peut observer, voir perturber. Ces attaques peuvent être de deux types : invasives et non invasives.

1.8.1.1. Attaques invasives

Elles permettent de récupérer un ensemble d'information de la carte en se basant sur une cartographie des circuits. Ces attaques sont dites de ''reverse engineering'' car l'attaquant tente de déduire les algorithmes utilisés, leurs implémentations, les systèmes de sécurité mis en place et les informations contenus dans la puce à partir de l'analyse ou la modification des circuits intégrés dans la carte. Pour pouvoir arriver à ce résultat, l'attaquant cherche à isoler les circuits de manière physique ou chimique. Le plus souvent, la puce n'est plus réutilisable après ces attaques, d'où l'aspect invasif. On peut citer comme exemple :

– La modification de circuit à l'aide du FIB (Focused Ion-Beam) permettant d'ajouter ou de retirer des nouvelles pistes conductrices sur la puce.

- Equipement **très coûteux mais possibilités énormes pour l'attaquant**

Figure 3. 10 : Equipement FIB

– La rétro-conception des blocs fonctionnels du microprocesseur dans le but de déterminer toutes les informations secrètes de la carte.

– Le sondage (probing en anglais) physique, c'est-à-dire la pose de micro-sondes sur les bus de la puce dans le but de déterminer ou falsifier l'information qui y circule.

1.8.1.2. Attaques non invasives

Dans le cas des attaques non invasives, on trouve plusieurs types d'attaques qui sont :

a) Attaques par conditions anormales

L'attaquant fait fonctionner la carte avec des valeurs en dehors des normes acceptables de fonctionnement de la carte. Ces caractéristiques peuvent par exemple être la fréquence d'alimentation ou la tension aux bornes de la carte. Aujourd'hui, une majorité des cartes possèdent des détecteurs d'activités anormales permettant de désactiver la carte pour lutter contre ce type d'attaque.

b) Attaques par canaux cachés

Ce sont des attaques par observation du signal puis d'analyse statistique. Les paramètres d'observation pouvant être le temps d'exécution, la consommation du courant ou même l'émission électromagnétique. L'attaque par analyse du temps d'exécution est basée sur le temps d'exécution des instructions. L'attaquant essaie de déduire des informations sur le type d'opération, les opérandes pour avoir des informations sur un programme ou un algorithme donné.

c) Attaques par injection de fautes

Elles sont encore appelées attaques par perturbation. En effet, l'attaquant tente d'injecter des modifications physiques dans l'environnement de la carte (lumineuses, impulsions électriques, magnétiques, etc...) pour introduire des modifications dans le contenu des mémoires de la carte. Le but étant d'introduire des fautes lors de l'exécution d'un programme afin de provoquer des sorties erronées exploitables, d'éviter un test, d'appeler une autre sous fonction, de sauter l'appel de vérification d'un code PIN, etc... Cependant l'attaquant doit pouvoir localiser les cellules mémoires à attaquer et synchroniser l'attaque pour qu'elle coïncide avec la fenêtre d'opportunité offerte par le programme. Pour cela, l'attaquant peut observer l'activité de la carte à l'aide d'attaques par canaux cachés.

1.8.2. Attaques logicielles

Avec l'apparition des cartes « ouvertes » permettant de charger des applications dans la carte après émission de celle-ci, les attaques logicielles sont de plus en plus répandues. Elles utilisent des failles pour contourner les protections mises en place. Généralement, il s'agit d'une mise à défaut des mécanismes d'isolation et d'intégrité du code et des données des applications embarquées.

1.9. Les niveaux d'attaques

L'intrus peut effectuer quatre niveaux d'attaques, l'attaque est une tentative de cryptanalyse.

- L'attaque par cryptogramme (par message chiffré seulement) : ou le cryptanalyste ne connait qu'un ensemble de message chiffrés, il peut soit retrouver seulement les messages en clair, soit retrouver la clef. En pratique, il est très souvent possible de deviner certaines propriétés du message en clair (format ASCII, présence d'un mot particulier, …), ce qui permet de valider ou non le décryptement.

- L'attaque à message en clair connu : ou le cryptanalyste connait non seulement les messages chiffrés mais aussi les messages en clair correspondants, son but est alors de retrouver la clef. Du fait de la présence, dans la plupart des messages chiffrés, de parties connue (en-têtes de paquets, champs communs à tous les fichiers d'un type donné, …).

- L'attaque à message en clair choisi : ou le cryptanalyste peut, de plus choisir des messages en clair à chiffrer et donc utiliser des messages apportant plus d'informations sur la clef. Si le cryptanalyste peut de plus adapter ses choix en fonction des messages chiffrés précédents, on parle d'attaque adaptative.

- L'attaque à message chiffré choisi : Dans ce cas, le cryptanalyste peut choisir des messages chiffrés pour lesquels il connaitra le message en clair correspondant ; sa tâche est alors de retrouver la clef. Ce type d'attaques est principalement utilisé contre les systèmes à clef publique, pour retrouver la clef privée.

1.10. Comment sécuriser une carte à puce ?

Il existe de nombreuses contre-mesures pour résister à des attaques malicieuses. Ces contre-mesures peuvent être logicielles et/ou matérielles selon le type d'attaques. Pour atteindre cet objectif, tous les chemins de test doivent être irréversiblement détruites, voire inexistants. Des détecteurs d'intrusion, des alarmes et des mécanismes de protection doivent être implémentés et exploités par le système d'exploitation. Le composant doit aussi être capable d'exécuter des tâches de contrôle, non seulement dans les conditions normales de fonctionnement, mais aussi, et surtout, dans les conditions anormales.

1) Interdire le mode test

Tous les chemins de test doivent être invisiblement détruits. Des détecteurs d'intrusion, des alarmes et des mécanismes de protection doivent être implémentés et exploités par le système d'exploitation.

2) Résister aux attaques

Attaques actives ou invasives, les composants doivent résister aux tentatives de lecture des mémoires par dépassivation des couches de protection physique, lecture des bus par l'utilisation de faisceau d'ions …

Concernant les attaques passives (ne perturbant pas le fonctionnement du composant), ces types d'attaques peuvent être résumés comme suit :

- Attaque de type SPA (Single power Analysis) : fondés sur un espionnage des signaux électriques ou électromagnétiques émis par une machine.

- Attaque du temps (Timing Attack) : dans ce type d'attaque il est également possible de connaitre certaines données sensibles, comme des clés cryptographiques, en exploitant simplement le temps de traitement des différentes instructions du processeur.

- Attaque de type DPA (Differential Power Analysis) : ce type d'attaque est fondé sur une analyse statistique des variations de puissance consommée pendant l'exécution d'un algorithme cryptographique connu.

3) Des mémoires intactes

On doit contrôler l'intégrité des données et des programmes, ainsi que le déroulement et la reprise des diverses opérations en cas d'interruption brutale d'un processus pour une raison quelconque, comme la coupure de l'alimentation de la carte en cours de transaction (cas d'un arrachage de la carte).

4) Conserver l'union microprocesseur-logiciel

En termes de sécurité, le logiciel et le matériel sont inséparables. Dans les composants actuels, des dispositifs sont employés au niveau du matériel, de type ACL (Access Control List) ou MMU (Memory Management Unit), complété par des mécanismes de cloisonnement des données et des applications supervisées par logiciel. Toute tentative d'accès frauduleux à une donnée ou un programme doit être détectée et enregistrée.

1.11. Sécurité des communications

Les différents modes de communication (contact, sans contact) peuvent être utilisés pour tenter de compromettre les biens de la carte. Si un risque de ce type existe, il est indispensable de sécuriser les communications, en les chiffrant si on veut préserver leur confidentialité et/ou en les signant si l'on veut compromettre l'intégrité. Dans certains cas, il peut également être nécessaire d'assurer la disponibilité du canal de communication.

Pendant longtemps, le risque d'attaques sur les communications via les contacts est faible. Ainsi, les PIN des cartes étaient présentés en clair entre le lecteur et la carte. C'est encore souvent le cas mais de plus en plus, certaines applications considèrent que ce risque doit être pris en compte et qu'il faut mettre en place des contre-mesures. C'est le cas, par exemple, pour les cartes bancaires où il est prévu que le PIN puisse être présenté chiffré ou encore, que

l'intégrité des informations retournées par la carte vers le lecteur doit être assuré (mode CDA (Combined Data Authentification) en EMV par exemple).

Toutefois, en mode contact, la plupart des attaques nécessitent une intervention sur les lecteurs ou des aménagements sur les cartes elles-mêmes ce qui limite souvent leurs rentabilités ou leurs généralisations.

Les attaques applicables au mode contact sont généralement applicables au mode sans contact mais avec parfois un mode opératoire simplifié ou une plus grande discrétion. De plus, il existe des attaques que l'on peut considérer comme spécifiques au mode sans contact et qu'il est bon de rappeler, ainsi que leurs limites.