INTRODUCTION À LA CRYPTOGRAPHIE

Rida Khatoun
rida.khatoun@gmail.com

Sommaire

- Introduction
- Services de sécurité basés sur la cryptographie
- Algorithmes de chiffrement symétrique
- Echange de clés : Diffie Hellman
- Fonctions de hachage
- Algorithmes de cryptographie asymétrique
- Cryptographie à courbes elliptiques (ECC)
- PKCS (Standards de la cryptographie asymétrique)
- PKI basée sur la cryptographie asymétrique
- Protocole SSL/TLS

Introduction - Terminologie

- (dé)chiffrement: (angl. encryption / decryption)
  - Transformation d'un message « lisible » ou texte clair (angl. ***plaintext***) en un message incompréhensible ou texte chiffré (angl. ***ciphertext***)
- Cryptogramme
  - Message chiffré ou texte chiffré
- Décrypter
  - Retrouver le message en clair à partir d'un cryptogramme sans être en possession de l'ensemble des éléments qui ont permis sa conception

Introduction - Terminologie

- Cryptographie
  - du grec « ***kruptos*** » (caché) et « ***graphien*** » (écrire)
  - La science relative à la création des cryptogrammes
- Cryptanalyse
  - La science relative au décryptage
- Cryptologie
  - La science regroupant la cryptographie et la cryptanalyse

Introduction - Terminologie

- Système cryptographique ou cryptosystème
  - {P, C, K, $<E_k,D_k>$}
  - C'est un procédé pour transformer un texte clair en texte chiffré et inversement
- Clé
  - Un secret associé au cryptosystème pour réaliser une transformation donnée
  - La clé a une taille fixe indépendante de la taille du message (à une exception près)
- Cléptographie
  - L'art de dérober les clés
- Stéganographie
  - L'art de la dissimulation

Introduction - Pourquoi la cryptographie?

- Confidentialité
  - Une des premières motivations de la cryptographie
  - Protéger l'information échangée entre deux ou plusieurs parties contre l'indiscrétion ou l'espionnage
- Intégrité
  - Lutter contre la falsification, voir également rumeur.
- Authentification
  - S'assurer de l'origine de l'information, pour appliquer un ordre émanant de la bonne source
- Non-répudiation

Services de sécurité

- **C**onfidentialité
- **I**ntégrité
- **A**uthentification
- **Identification**
- **Non répudiation**
- **Horodatage**

Services de sécurité - Confidentialité

Service de confidentialité

- Caractère réservé d'une information dont l'accès est limité aux personnes admises à la connaître
- ISO 7498-2 :
  - la propriété qu'une information n'est ni disponible ni divulguée aux personnes, entités ou processus non autorisés.
- Une information échangée entre deux ou plusieurs entités n’est accessible que par celles-ci.

Services de sécurité - Authentification

Service d'authentification

- Confirmation de la véracité de l'identité ou d'un élément spécifique à une entité déclarée
- ISO/IEC 2382/8:
  - Assure que l'identité de l'origine des données est bien l'identité revendiquée
- Dans la pratique l'authentification
  - consiste à relier des informations entre elles avec généralement un élément permettant de spécifier une entité

Services de sécurité - Intégrité

Service d'intégrité

- Propriété garantissant qu'une information n'a pas été modifié sans autorisation
- ISO 7498-2 :
  - la propriété assurant que des données n'ont pas été modifiées ou détruites de façon non autorisée
- Une information échangée entre deux ou plusieurs entités est reçue par tous telle qu'elle a été émise.
  - Dans un contexte d'échange l'authentification de l'origine accompagne le service d'intégrité.

Services de sécurité – Non répudiation

Service de non répudiation

- La répudiation consiste:
  - au fait que dans un échange où sont impliqués deux ou plusieurs entités , l'une de celles renie d'avoir participé à tout ou partie de l'échange
- La non répudiation consiste:
  - Au fait qu'aucune entité ne puisse répudier d'avoir participé à l'échange
- La non répudiation dans le contexte d'un émetteur et d'un récepteur:
  - Consiste donc à ce que ni l'émetteur et/ou le destinateur ne puisse répudier l'émission et/ou la réception d'un message
- La non répudiation relève de la notion de preuve au sens juridique du terme

Introduction - Pourquoi la cryptographie?

- Pour répondre aux besoins:
  - de stratégie militaire
    - Surprise, diversion, zizanie
    - Exemple:
  - de la diplomatie
    - Politique, économique, stratégique
    - Exemple: espionnage
  - de société
    - Lien amoureux en dehors des normes sociales
    - Exemple les célèbres lettres suivantes:

Introduction - Pourquoi la cryptographie?

- Pour répondre aux besoins:
  - de financiers
    - Banque, instrument de payements
    - Exemple: protection du patrimoine, carte bancaire
  - de l'informatique
    - Protection des moyens et ressources immatérielles
    - Télécommunications, réseaux, Internet.
  - des sociétés secrètes
    - Sectes, mafia, confréries, etc.
    - Usagesdu grand public
    - très variés

Introduction - les dates importantes

- -500 La skytale
- -400 Le code de Cesar
- 1585 Blaise Vigenère
- 1861 Friedrich W. Kasiski
- 1883 Kerckhoffs
- 1926 Gilbert S. Vernam
- 1939 Enigma
- 1940 Shanon

Introduction - les dates importantes

- 1970 Horst Fiestel – Lucifer
- 1976 DES
- 1976 Diffie Helmann
- 1978 RSA (Rivest-Shamir-Adleman)
- 1984 ROT13
- 1990 IDEA
- 1990 Crypto Quantique: Bennett, Brassard
- 1991 PGP: Phil Zimmermann
- 2000 AES

Introduction - Principes

- Fonctions de chiffrement
  - M : le message à transmettre
  - E : un procédé de chiffrement
  - k : un secret représentant la clé
  - On peut calculer le cryptogramme C avec différentes fonctions E: (E pour Encryption)
    - C = E (M)
    - ou C = E (k,M) qu'on note aussi C = E_k (M) ou C = $\{M\}_k$
  - Et pour retrouver le message clair à partir du cryptogramme on utilise la fonction inverse (ou des fois la même fonction):
    - M = E^{-1} (C)
    - ***ou*** M = E^{-1} (k, C) qu'on note également M = E_k^{-1} (C)
      ou on note D *au lieu de* E^{-1} (D pour Decryption)
  - On ainsi M = D_k (E_k (M))

Introduction - Principes

- Auguste Kerckhoffs (1835-1903)
  - Né en Hollande et enseigna notamment en France
  - Énonce les bases de la cryptographie moderne dans un journal militaire en 1883.
- Les principes de Auguste Kerckhoffs
  - La sécurité d'un système de cryptographie dépend que du secret de la clé.
  - Une information chiffrée ne peut être déchiffrée qu'avec la clé
  - La divulgation du système de codage n'a aucune conséquence sur les informations échangées
  - La clé doit être simple et modifiable.
  - Les cryptogrammes doivent être transportables
  - Le support de codage et les documents doivent être transportables.
  - Le système de codage doit être simple et certifié par des experts

Introduction - Principes

Substitution monoalphabétique

- Chiffrement par substitution mono alphabétique
  - Pour un alphabet donné: chaque symbole (ou groupe) est substitué par un autre symbole (ou groupe) (bijection)
  - Technique de chiffrement la plus utilisée durant le premier millénaire
- Codage utilisé par Jules César
  - Décalage de trois caractères sur l'ordre alphabétique
- Unix propose ROT13 (ROTation de 13 ou k = 13)
  - Pour éviter une lecture involontaire

Introduction - Principes

Substitution monoalphabétique

Le code de César

Soit la table suivante: alphabet latine sur 26 lettres

Et soit la table de substitution des caractères une à une

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

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

Message claire:

INTRODUCTIONALACRYPTOGRAPHIQUE

Message chiffré:

LQWURGXFWLRQDODODFUBSWRJUDSKLTXH

Introduction - Principes

Substitution monoalphabétique

- Propriétés de ce type de substitution:
  - Longueur du message chiffré est identique à celle du message claire
  - On substitue un caractère du message en claire par un autre de manière bijective
  - Valeurs des fréquences des caractères sont semblables claire/chiffré
- E_k (s) = s + k mod 26
  - avec s symbole clair
- D_k (s)= s – k mod 26
  - avec s symbole chiffré
- Dans le code de César k = 3

Introduction - Principes

Substitution monoalphabétique

- Retrouver la valeur de la clé *k* pour un code de substitution mono alphabétique:

  GV XGZ V KJPM QVGZPM XDIL

- Soit le code de substitution suivant:

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

Proposez une méthode de cryptanalyse?

Il y a 27! codes différents.

- Al Kandi au 9ième siècle réussit à briser le code par substitution et invente la cryptanalyse.
  - Analyse fréquentielle

Introduction - Principes

Substitution monoalphabétique

- Analyse fréquentielle
  - Calculer les fréquences d'apparition des caractères chiffrés et comparer celle-ci avec les fréquences de la langue en question
  - On peut dresser des tables de fréquences: des caractères, des bigrammes et des trigrammes
- Ensuite on applique une analyse au texte chiffré selon ces tables

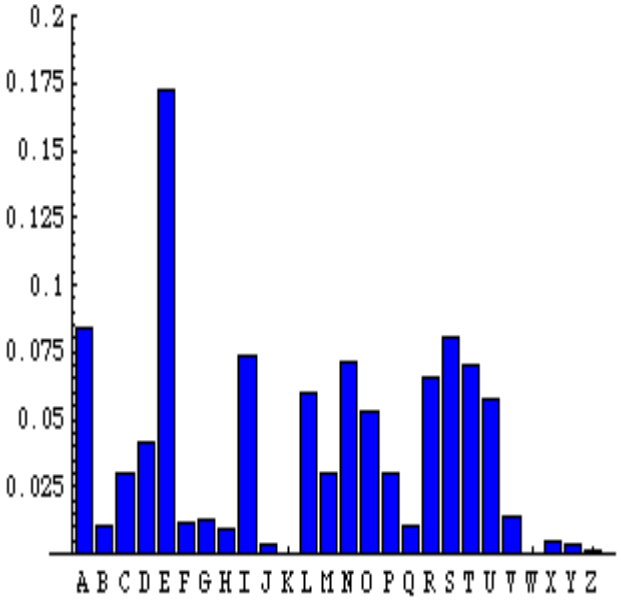
Introduction - Principes

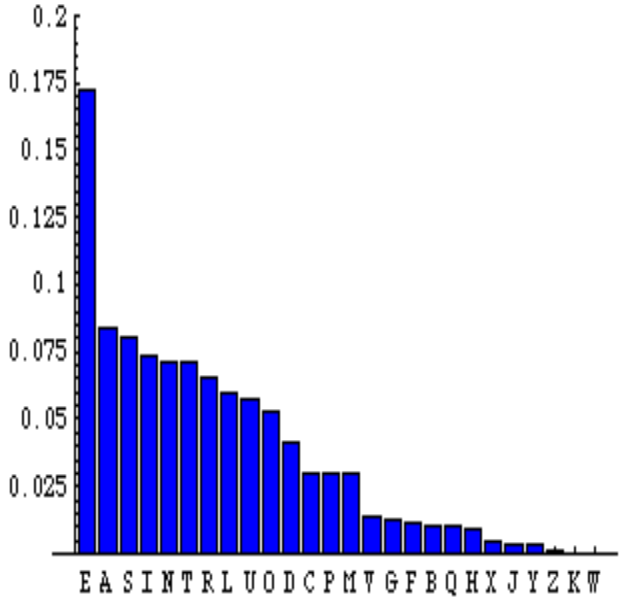
Substitution monoalphabétique

Fréquences d'apparition des lettres			
Lettre	Fréquence	Lettre	Fréquence
A	8.40 %	N	7.13 %
B	1.06 %	O	5.26 %
C	3.03 %	P	3.01 %
D	4.18 %	Q	0.99 %
E	17.26 %	R	6.55 %
F	1.12 %	S	8.08 %
G	1.27 %	T	7.07 %
H	0.92 %	U	5.74 %
I	7.34 %	V	1.32 %
J	0.31 %	W	0.04 %
K	0.05 %	X	0.45 %
L	6.01 %	Y	0.30 %
M	2.96 %	Z	0.12 %

Les 20 bigrammes les plus fréquents																				
Bigrammes	ES	DE	LE	EN	RE	NT	ON	ER	TE	EL	AN	SE	ET	LA	AI	IT	ME	OU	EM	IE
Nombres	3318	2409	2366	2121	1885	1694	1646	1514	1484	1382	1378	1377	1307	1270	1255	1243	1099	1086	1056	1030

Les 20 trigrammes les plus fréquents																				
Trigrammes	ENT	LES	EDE	DES	QUE	AIT	LLE	SDE	ION	EME	ELA	RES	MEN	ESE	DEL	ANT	TIO	PAR	ESD	TDE
Nombres	900	801	630	609	607	542	509	508	477	472	437	432	425	416	404	397	383	360	351	350





Introduction - Principes

Substitution monoalphabétique

- Le carré de Polybe -150 av JC. Le premier a avoir introduit le chiffrement par substitution.
- Carré 5x5, on substitue chaque lettre par ses coordonnées dans le tableau(L.C). Possibilité d'introduire d'autres symboles et d'agrandir le tableau.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	J
3	K	L	M	N	O
4	P	Q	R	S	T
5	U	V	X	Y	Z

Carré de Polybe avec une clé

	1	2	3	4	5
1	O	R	S	Y	A
2	B	C	D	E	F
3	G	H	I	J	K
4	L	M	N	P	Q
5	T	U	V	X	Z

- Nombre de symboles chiffrés complique l'analyse
- Message clair : ELLE EST ARRASSEE
- Message Chiffrée: 153232 151544 451143 431144 441515
- Possibilité de joindre une clé:
- Message chiffré: 244141 242413 511512 121513 132424

Introduction - Principes

Substitution monoalphabétique

- Les codes par substitution
  - Le code des templiers (symboles)
  - Le morse
  - Le code de Delastelle
  - Variantes du code de César
- Substitution de mots
  - Chaque mot est remplacé par un symbole
  - Nécessité d'un dictionnaire (ou cahier de code)
  - Problème: interception du cahier et changement de code
  - Combinaison mots et symboles: plus de symboles dans le chiffrement que de symboles de l'alphabet initial
  - Marie Stuart (1586 reine d'écosse) utilisé ce code qui fut cassé et elle a été exécuté après découverte de son complot contre la reine Elizabeth

Introduction - Principes

Substitution polyalphabétique

- Substitution polyalphabétique
- Code de Vigenère est une amélioration du code de César
  - Substitution variable fonction de la position du caractère et d'une clé
  - Etapes
    - Un message claire est découpé en bloc ayant la taille de la clé
    - On applique à chaque bloc le traitement suivant: la première lettre est décalée selon la première de la clé, idem pour la deuxième selon la deuxième de la clé, etc.

Introduction - Principes

Substitution polyalphabétique

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

Introduction - Principes

Substitution polyalphabétique

- Message = **I N T R O D U C T I ON**
- CLE = **O R S Y S O R S Y S O R Y S**
- Chiffré = W E LPG R L U RA C E

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

Introduction - Principes

Substitution polyalphabétique

- Chiffrement de Vigenère : Test de Kasiski
  - Cryptogramme

*XAUNMEESYIEDTLLFGSNBWQUFXPQTYORUTYIIN**UMQI**EULSMFAFXGUTYBXXAGBHMIFIIM**UMQI**DEKRIFRIRZQUHIENOO**OI**
GRMLYETYOVQRYSIXEOKIYPY**OIGR**FBWPIYRBQURJIYEM**JIGRY**KXYACPPQSPBVESIRZQRUFREDY**JIGRY**KXBLOPJARNPU
GEFBWMILXMZSMZYXPNBPUMYZMEEFBUGENLRDEPBJXONQEZTMBWOEFIIPAHPPQBFLGDEMFWFAHQ*

  - Test pour trouver la taille de la clef
    - UMQI se retrouve après 30 caractères
    - OIGR se retrouve après 25 caractères
    - JIGRY se retrouve après 30 caractères
  - La longueur de la clé doit être un diviseur de 30 et de 25 : il est possible qu'il s'agisse de 5

Séquence	Distance	Diviseurs de la distance					
UMQI	30	2	3	5	6	10	15
OIGR	25	-	-	5	-	-	-
JIGRY	30	2	3	5	6	10	15

Introduction - Principes

Substitution de polygrammes

- Remplacement d'un groupe de caractères au moyen
  - d'une table code de Playfair, remplacement:
    - Règle 1: deux lettres dans les coins d'un rectangle, par les deux autres lettres du coin du même rectangle: AH par KN
    - Règle 2: deux lettres sur la même ligne par les deux lettres à leur droite qui les suivent: AK par RX
    - Règle 3: deux lettres sur la même colonne par les deux lettres qui suivent la première: YS par AO et DI par FR
    - Règle 4: deux lettres identiques par un nul entre les deux (X
  - d'une fonction mathématique code de Hill $\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \pmod{26}$

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

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

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

Introduction - Principes

Approches pour le chiffrement

- Théorie de l'information de Shanon
  - Assure que le cryptanalyste ne dispose pas de suffisamment d'information pour décrypter le cryptogramme
    - Les méthodes de substitution et de transposition sont à la base de la cryptographie actuelle.
    - La confusion: supprime la relation entre clair et chiffré au moyen de la substitution.
    - La diffusion: répartie la redondance dans le texte chiffré au moyen de la transposition
- Théorie de la complexité de calcul
  - Assure que la cryptanalyse nécessite beaucoup de temps
    - Factorisation de nombres premiers
    - Logarithme discret

Algorithmes cryptographique

- Algorithmes de chiffrement symétrique
  - Par bloc
  - Par flot
- Fonction à sens unique
  - Algorithmes de chiffrement asymétrique
  - Fonctions de hachage avec et sans clé
- Protocole d'échange de clés : Diffie Helman

Algorithmes de chiffrement symétrique

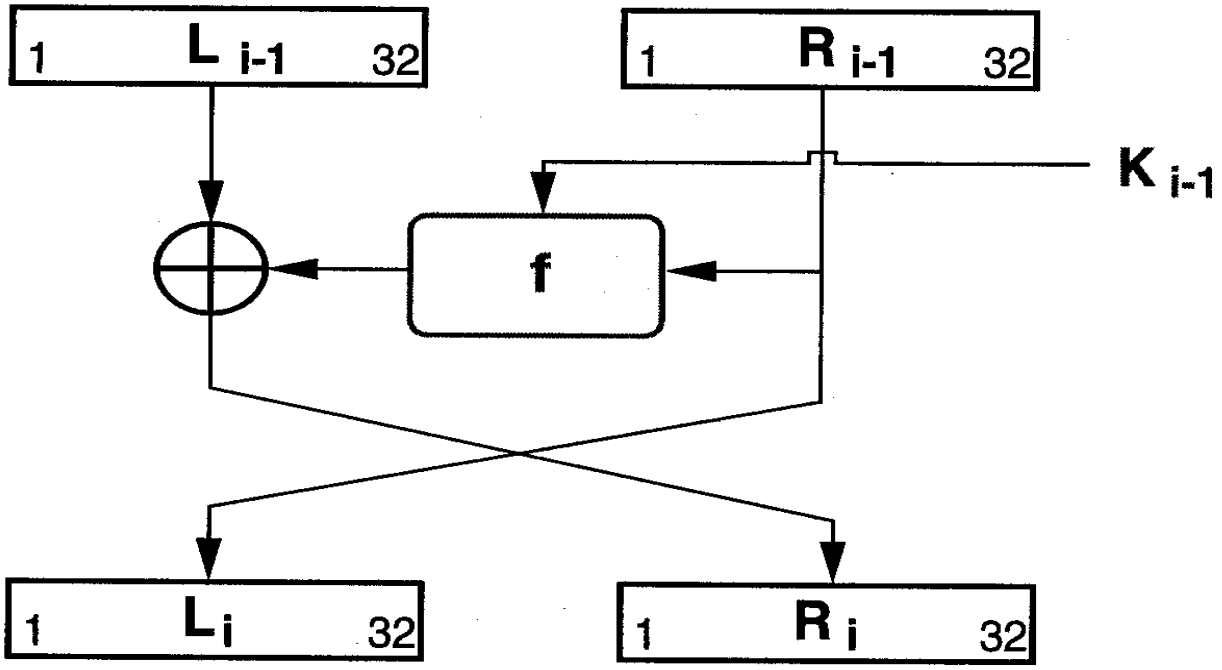
- Basé sur la substitution et la permutation
  - Principes de diffusion et de confusion à l'aide d'une clé
- Une fonction qui transforme un message en clair en message chiffré à l'aide d'une clé

K	:	la clé,
M	:	le message clair,
C	:	le message chiffré,
E	:	la fonction de chiffrement
E^{-1}	:	la fonction de déchiffrement

$$E(K, M) = C \quad \text{et} \quad E^{-1}(K, C) = M$$

Algorithmes de chiffrement symétrique

- Le schéma de FIESTEL à la base de la conception de plusieurs algorithmes de chiffrement symétrique



Algorithmes de chiffrement symétrique

- DES (Data Encryption Standard)
  - Algorithme de chiffrement symétrique
  - Par bloc avec feedback et sans feedback
- Initialement Lucifer algorithme d'IBM conçu par Fiestel
  - Devenu DES suite à un appel du NBS (National Bureau of Standardisation)
- Taille de clés de 56 bits (initialement sur 64 bits)
  - bridée par la NSA (8ème bit de parité)
- Standard 15 janvier 1977 du FIPS (PUB 46).
  - Version du 25 octobre 1999 FIPS PUB 46-3
- DES est des plus déployés parmi les algorithmes symétriques

Algorithmes de chiffrement symétrique

- DES fonctionne en trois étapes
- Le message est découpé en blocs de 64 bits
- Étape 1: permutation initiale et fixe d'un bloc
  - Les S Boxes (8) sont les tables qui définissent ces permutations.
- Étape 2: 16 itérations d'une transformation,
  - À chaque itération (schéma de FIESTEL)
    - calcul d'une clé de 48 bits à partir de la clé initiale (substitution et xor)
    - le bloc de 64 bits est découpé en deux blocs de 32 bits, ces blocs sont échangés selon un schéma de Feistel.
    - le bloc de 32 bits ayant le poids le plus fort subira une transformation.
- Étape 3: le résultat de la dernière ronde est transformé par la fonction inverse de la permutation initiale.

Algorithmes de chiffrement symétrique

- Diversification de la clef dans DES :
  - K de 64 bits est réordonné dans PC-1 qui supprime les bits de parités (en 8, 16,…,64)
  - LS_i : rotation circulaire vers la gauche d'une ou deux positions selon la valeur de *i*
  - PC2 : autre permutation de bits

56 bits

Table 3.1 Permuted choice 1 (PC-1)

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

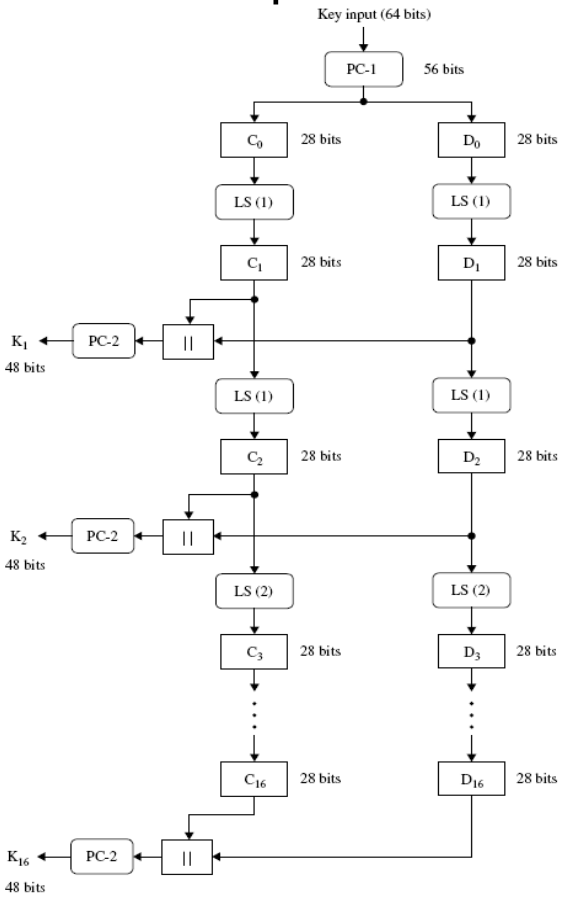
Table 3.2 Schedule for key shifts

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of left shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

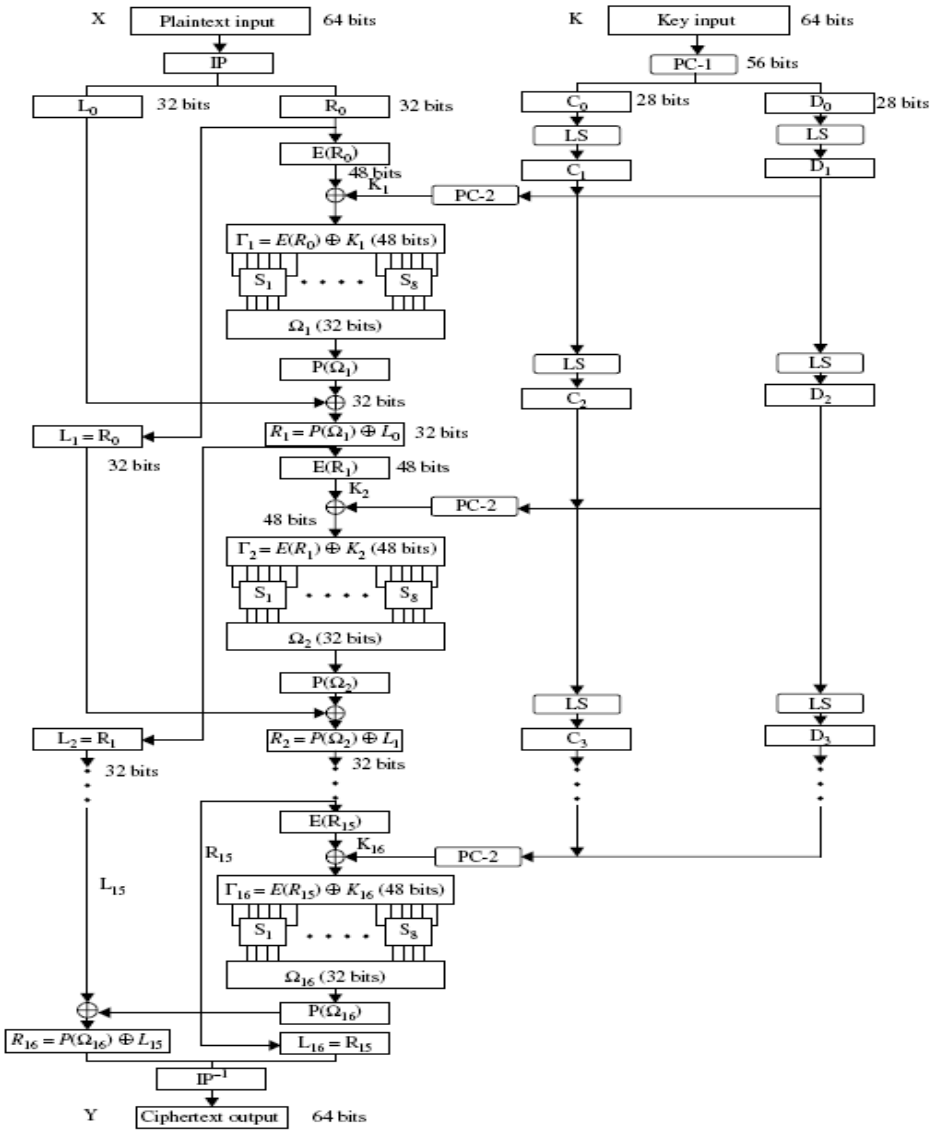
48 bits

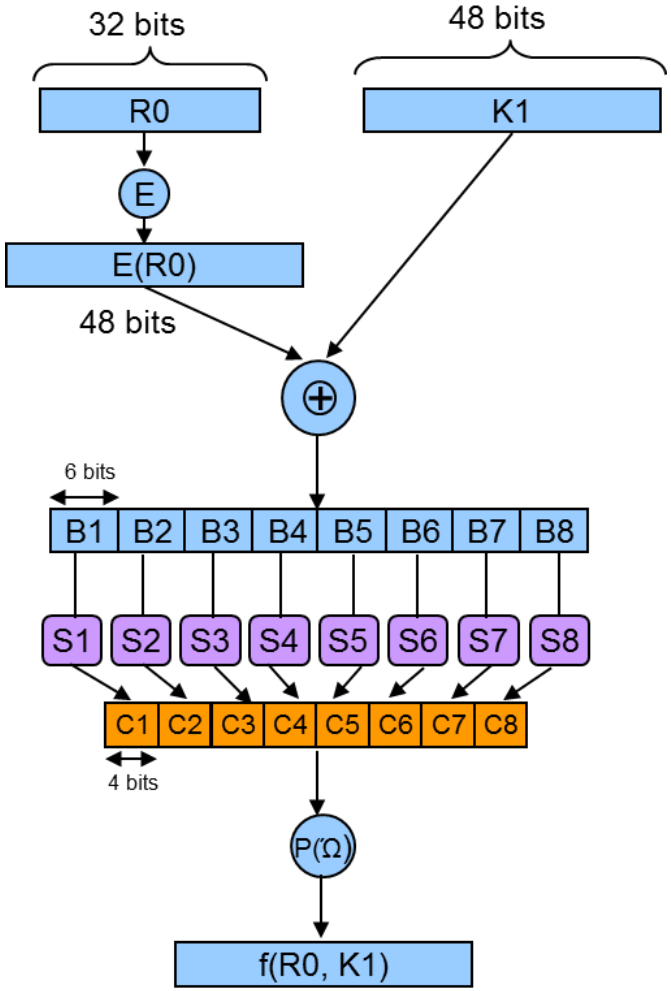
Table 3.3 Permuted choice 2 (PC-2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32



Algorithmes de chiffrement symétrique





Algorithmes de chiffrement symétrique

- 3DES est l'application de trois fois successives de DES
- Peut être usuel avec 3 ou deux clés différentes
- 3DES avec la même clé répétée trois fois est compatible DES
- La complexité de 3 X 56 bits est inférieure à celle d'une clé de 168 bits
- DES-ede3 (*Encryption, Decryption, Encryption*)

Algorithmes de chiffrement symétrique

- International Data Encryption Algorithm (IDEA)
  - Proposé par l'ASCOM en 1992 pour remplacer DES
- Taille de clé de 128 bits
- Mode de chiffrement par bloc de 64 bits
- Basé sur le schéma de FIESTEL et utilise 3 opérations :
  - Ou exclusif
  - Addition modulo 216
  - Multiplication modulo 216

Algorithmes de chiffrement symétrique

- En 1998 appel à projet du NIST pour remplacer DES
  - Pour tous secteurs : bancaire, militaire, Internet
  - Sécurité et performance supérieure à 3DES
  - Tailles de blocs et de clé supérieures à 128 bits
  - Flexible au niveau implémentation
- Plusieurs réponses:
  - MARS, RC6, Rijndael, Serpent, Twofish
- En 2000 : algorithme Rijndael retenu
  - devenu AES (Advanced Encryption Standard)

Algorithmes de chiffrement symétrique

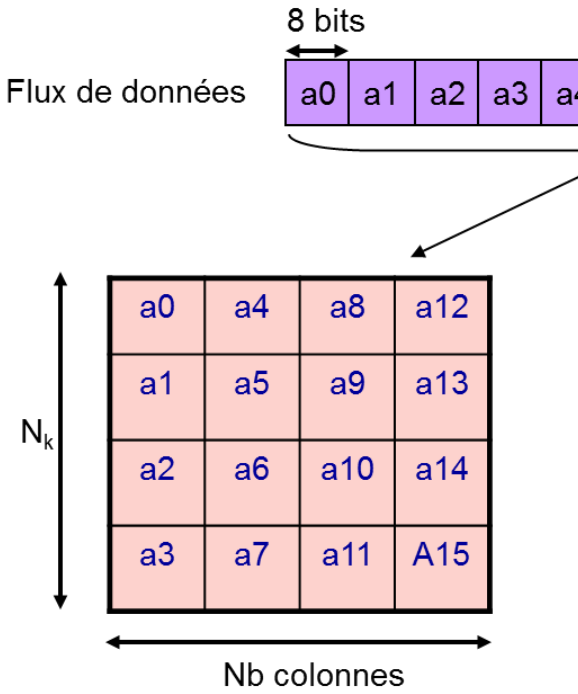
- AES n’est pas basé sur le schéma de FIESTEL
  - Consommation de mémoire moindre que DES
- La taille de clé: 128, 192 ou 256 bits
- Rotation basée sur des matrices 4x4
- Transformation linéaire pour garantir la diffusion
- Xor entre matrices
- Plusieurs tours sont appliqués avec ce même schéma au bloc

Algorithmes de chiffrement symétrique

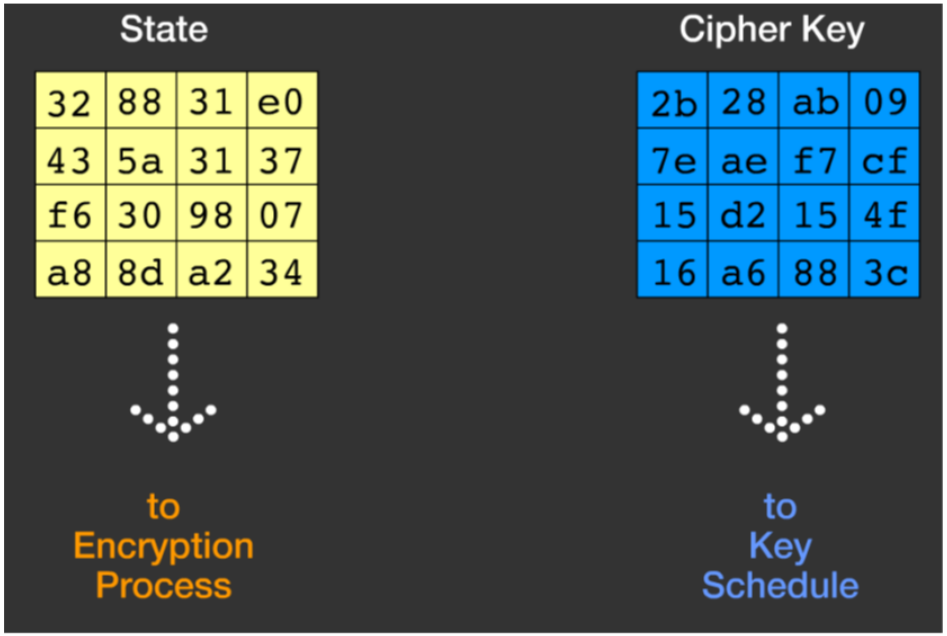
- Principe de l'algorithme AES
  - Chiffrement AES consiste en
    - Une addition initiale de clef (AddRoundKey)
    - Nr-1 tours (rondes) chacun divisé en 4 étapes
- Quatre étapes d'une ronde (tours)
  - SubBytes
    - Substitution non-linéiare où chaque octet est remplacé par un autre choisi d'une table SBox
  - ShiftRows
    - Transposition où chaque élément de la matrice est décalé cycliquement à gauche d'un certain nombre de colonnes
  - MixColumns
    - Produit matriciel
  - AddRoundKey
    - Addition de chaque octet avec l'octet correspondant dans une clé de tour obtenue par diversification

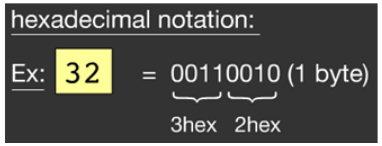
Algorithmes de chiffrement symétrique

- AES



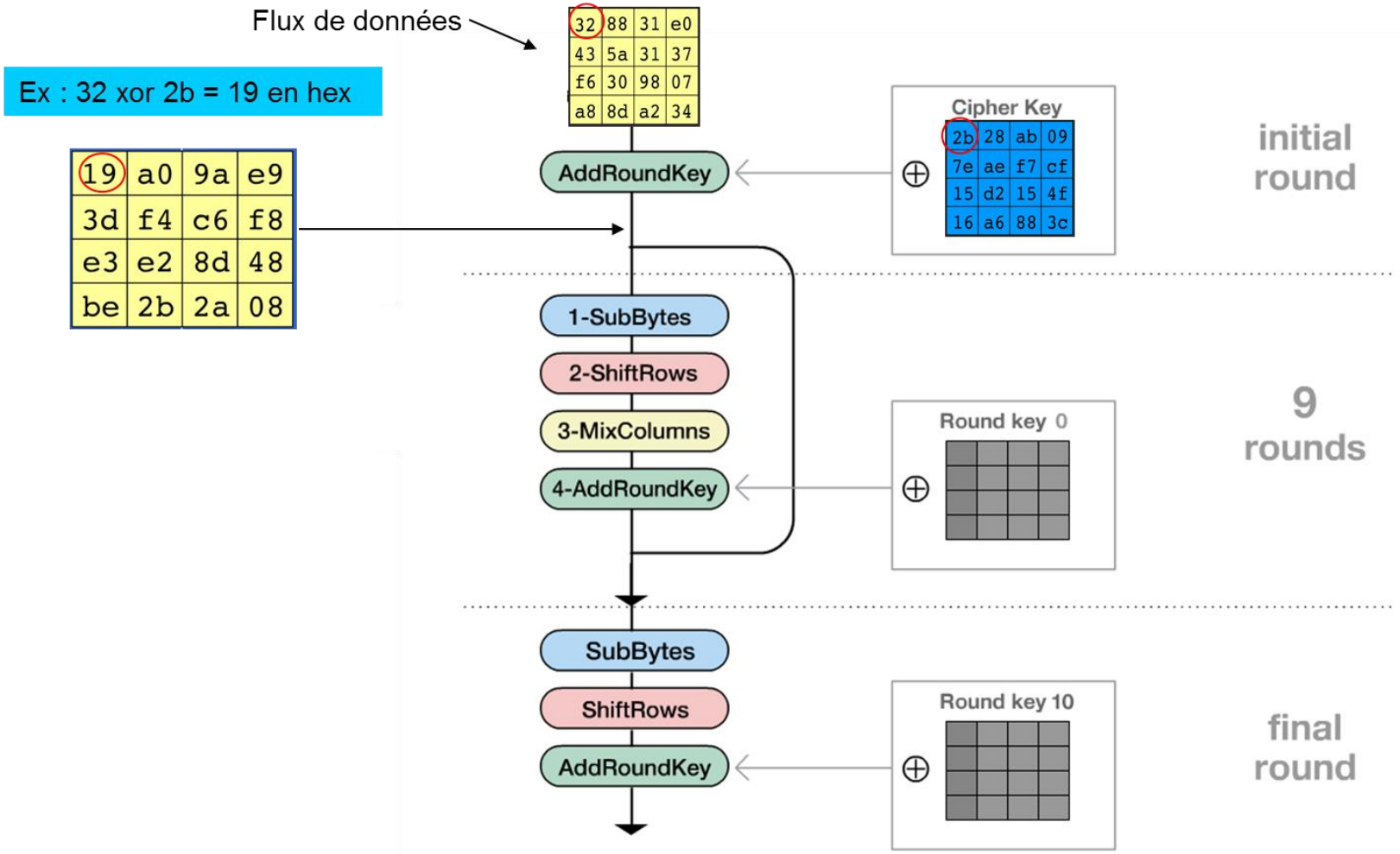
Représentation matricielle d'un bloc de 16 octets





Algorithmes de chiffrement symétrique

- AES



Algorithmes de chiffrement symétrique

- RC4
  - Algorithme de chiffrement à la volée ou stream
  - RC4 (Rivest Cipher 4 ou Ron's Code) conçu en 1987 par Ronald Rivest (Rivest Cipher 4 ou Ron's Code)
    - Algorithme de chiffrement symétrique de la même compagnie RS RC2, RC5 et RC6.
  - RC4 a été révélé septembre 1994 d'une manière anonyme sur la liste de diffusion Cypherpunks
    - RC4 est une marque déposée mais les implémentations non-officielles sont autorisées (pas breveté).
  - Il est intégré à la quasi-totalité des browsers (SSL/TLS)
  - Usuel dans le WIFI
  - Chiffrement en stream ou enfilé, taille des clés entre 40 et 128 bits

Algorithmes de chiffrement symétrique

- Blowfish
  - Conçu par Bruce Schneier 1993
  - Blocs de 64 bits
  - Nom du poisson-lune japonais (ou fugu)
  - Clés de 32 bits à 448 bits
  - Basé sur le schéma de Fiestel (16 tours)
  - S-Boxes taille fonction de la clé

Algorithmes de chiffrement symétrique

- Towfish (ancien candidat pour AES)
  - Conçu par B. Schneier, Niels Fergusson, …
  - Blocs de 128 bits
  - Clés de 128, 192, 256 bits
  - S-Boxes taille fonction de la clé
  - Basé sur le schéma de Fiestel (16 tours)
  - Très résistant à la cryptanalyse
  - Plus performant que AES pour une clé de 256 bits

Algorithmes de chiffrement symétrique

- DESX dérivé de DES
  - Conçu par Ron Rivest en 1984
  - Proposé pour contrer les attaques en force brute
  - Vulnérable comme DES aux attaques de cryptanalyse linéaire et différentielle

$$\mathbf{DESX\ (M) = K_2\ XOR\ DES_k\ (\ M\ xor\ K1)}$$

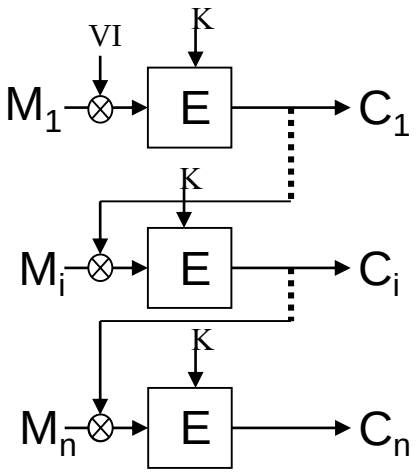
K_2 et K_1 sont calculés sur la base de la clé K

Algorithmes de chiffrement symétrique

Algorithme	Nom et commentaires	Type de chiffrement	Longueur de la clé en bits	Normalisé
DES	*Data Encryption Standard*	bloc de 64 bits	56	FIPS Pub 81,1981 ANSI X3.92, X3.105, X3.106 ISO 8372 ISO/IEC 10116
IDEA	*International Data Encryption Algorithm*,	bloc de 64 bits	128	
RC2	développé par Ronald Rivest	bloc de 64 bits	variable, 40.exp.	Non et propriétaire
RC4	développé par R. Rivest	enfilé	variable 40 - 128	Non, diffusé sur l'Internet en 1994
RC5	développé par R. Rivest	bloc de 32, 64 ou 128 bits	variable à 2048	Non et propriétaire
SKIPJACK	Confidentiel NSA .	bloc de 64 bits	80	Secret défense US
Triple DES		bloc de 64 bits	112	ANSI X9.52
AES	Advanced Encryption Standard	bloc de 128 bits	128,192, 256	FIPS197 2001

Algorithmes de chiffrement symétrique

- Chiffrement par bloc: on découpe un message par bloc et on chiffre bloc par bloc
  - DES, 3DES, AES, IDEA, …
- Deux principaux modes opératoires
  - Avec feedback: le chiffrement d'un bloc se base sur le chiffré du bloc précédent
  - Sans feedback: les blocs sont chiffrés de manière indépendante



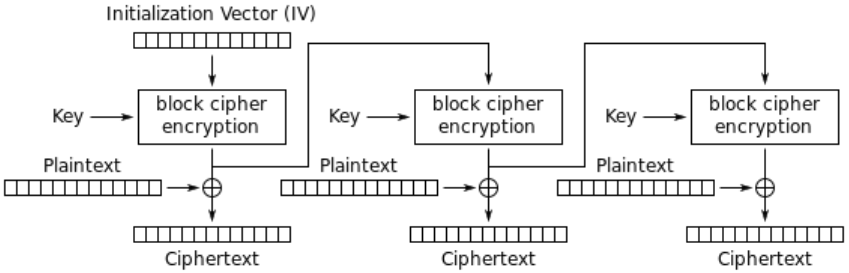
Avec feedback

Algorithmes de chiffrement symétrique

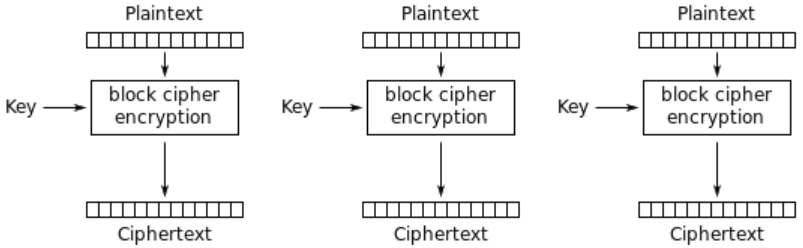
- Mode opératoire par blocs
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - CFB (Cipher Feedback Block)
  - OFB (Output Feedback Block)
  - CTR (CounTeR)
  - CTS (CipherText Stealing)
  - CBC-MAC

Algorithmes de chiffrement symétrique

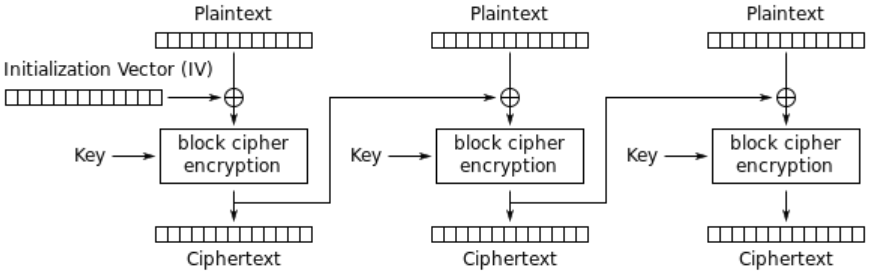
- Mode opératoire par blocs



Output Feedback (OFB) mode encryption



Electronic Codebook (ECB) mode encryption



Cipher Block Chaining (CBC) mode encryption

Algorithme de chiffrement symétriques - Exercice

- Soit t la taille d'une clé en bits
- La force brute consiste à tester la totalité de l'espace ou valeur de la clé
- Le teste d'une clé nécessite en moyenne l'exécution de 1000 instructions
- La puissance d'une machine est de 2000 MIPS

Algorithme de chiffrement symétriques - Exercice

- Pour t = 128 et disposant d'un texte clair/chiffré, calculer le temps nécessaire pour trouver la clé en force brute
- Sachant que la puissance des machines doublent tous les 18 mois
- Faites le même calcul pour les années 1980, 1990, 2000,2010, 2020, 2050, 2100, 2500

Algorithme de chiffrement symétriques - Exercice

Nombres de clés

$2^{128} = (2^{10})^{12} \approx (10^3)^{12} \approx 10^{36}$

Nombre d'instructions pour tester une clé

$1000 = 2^{10} = 10^3$

Nombre d'instructions pour tester 2^{128} clés

$10^{36} \text{ X } 10^3 = 10^{39}$

Nombre d'instructions en année pour une machine à 2000 MIPS

$2000 \text{ x } 10^6 \text{ X } 365 \text{ x } 3600 \text{ x } 24 = 10^{11} \text{x } 2 \text{ x} 365 \text{ x } 24$

$\approx 10^{16}$

Nombre d'années pour trouver tester 2^{128} clés

$10^{39} / 10^{16} = 10^{23}$

Algorithmes de chiffrement symétrique - Conclusion

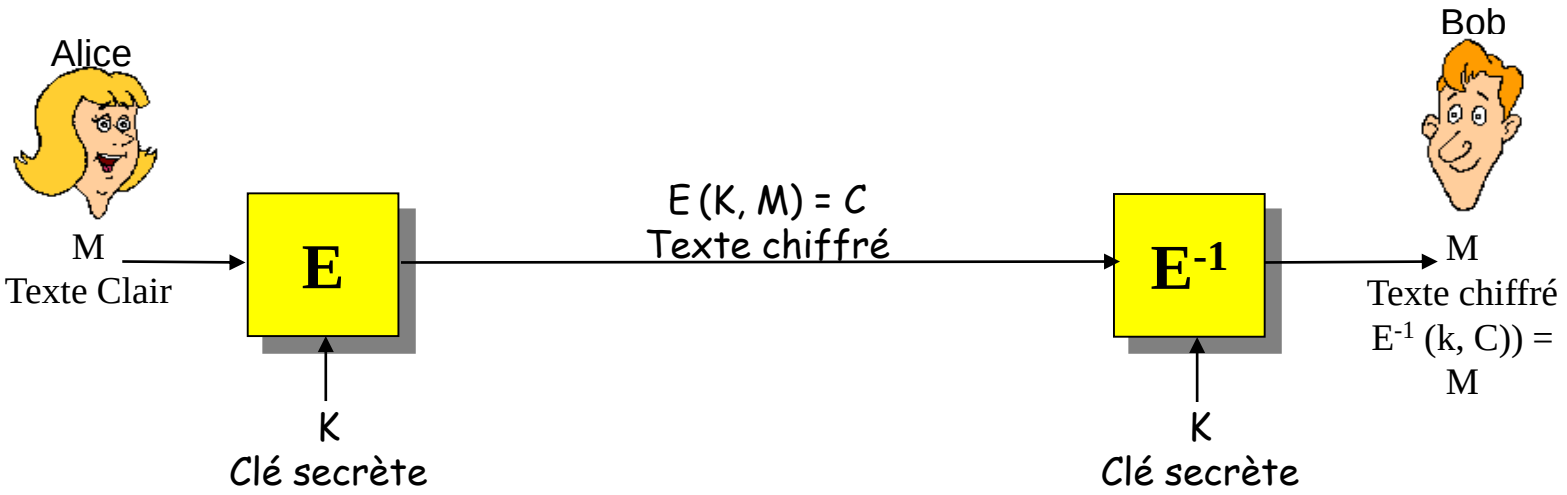
- La clé secrète ne doit pas être:
  - transmise sur le réseau
  - Stockée sur l'équipement en clair
- La taille de la clé est fixe
  - Généralement indépendante de la taille du message
  - Limitée en taille par la législation en cas de confidentialité
- Une clé différente pour chaque couple
  - Absence de gestion de clés
  - N x (N – 1) / 2

Algorithmes de chiffrement symétrique - Conclusion

- Problème de distribution des clés dans les grands réseaux
- Problème de gestion (attribution et révocation)
- Complexité => nombre de clés (maillage n X n)
- Problème législatif
- Problème de résilience (serveur central)

Algorithmes de chiffrement symétrique

Service de confidentialité



- Problème d'échange de clé
  - Le réseau n'est pas de confiance
- Une clé par couple
  - Gestion et renouvellement

Protocole d'échange de clé Diffie Hellman

Alice

Bob

Mise en accord publique sur 2 nombres premiers g et *p*

Génère une clef secrète *a*

Calcule son élément public

$A = g^a \mod p$

Envoie A à Bob

Calcule $K = (B)^a \mod P$

Génère une clef secrète *b*

Calcule son élément public

$B = g^b \mod p$

Envoie B à Alice

Calcule $K = (A)^b \mod P$

Attaquant : aucune information sur K à partir de A et B

Protocole d'échange de clé Diffie Hellman

Alice

Bob

g=11 et p=13

clef secrète *a=5*

A = g^a *mod p* = 11^5 *mod13*

A = 161051 mod 13

Envoie A = 7 à Bob

clef secrète b =19

B = g^b *mod p* =11^{19} *mod 13*

B = 61159090448414546291 mod 13

A=7

Envoie B=2 à Alice

B=2

Calcule K= $(B)^a$ mod p
alors K= $(2)^5$ mod 13

K = 32 mod 13

K=6

Calcule K= $(A)^b$ mod p =$(7)^{19}$ mod 13

K = 11398895185373143 mod 13

K=6

K = ?

(A=7, B=2, g=11, p=13)

Je ne connais pas a et b !!

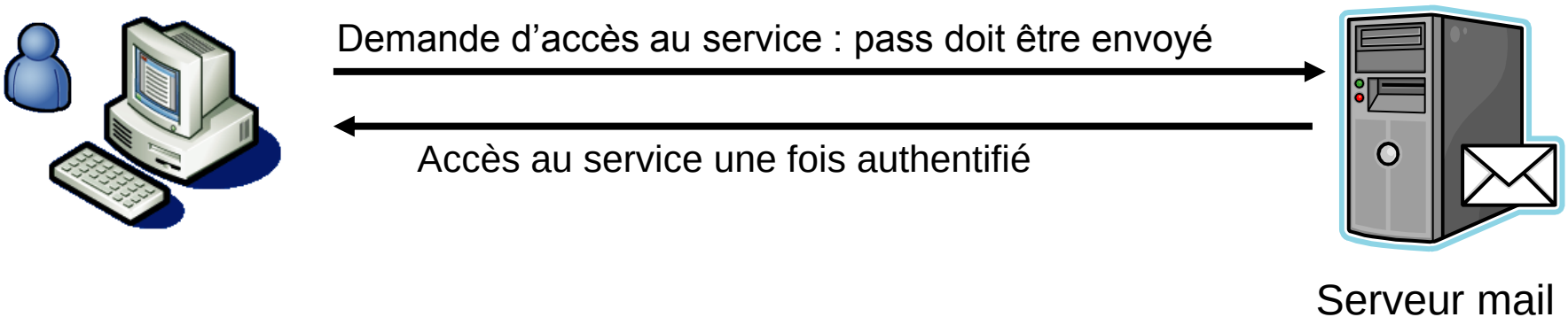
clef secrète pour un algorithme symétrique choisi (chiffrement et déchiffrement)

DH est sensible à l'attaque MITM !!!

Solutions de sécurité : Kerberos

- ISSU du projet Athena (1983) Massachusetts Institute of Technology
  - Sponsorisé par DEC et IBM
- Première version (v 4) (1989)
  - Difficulté à l'export car usage du chiffrement à 64 bits
- La version 4 intègre DES uniquement
  - La version 5 intègre DES, 3DES, AES, et RC4 et asymétrique
  - AFS utilise Kerberos v4.
  - DCE utilise Kerberos v5.
- La version actuelle (v 5)
  - RFC 1510 (protocole) et RFC1964 (mécanisme et le format d'insertion des jetons dans les messages Kerberos)
  - RFC1510 Obselete par RFC4120
  - Incluse dans Microsoft server 2003

- Authentification standard

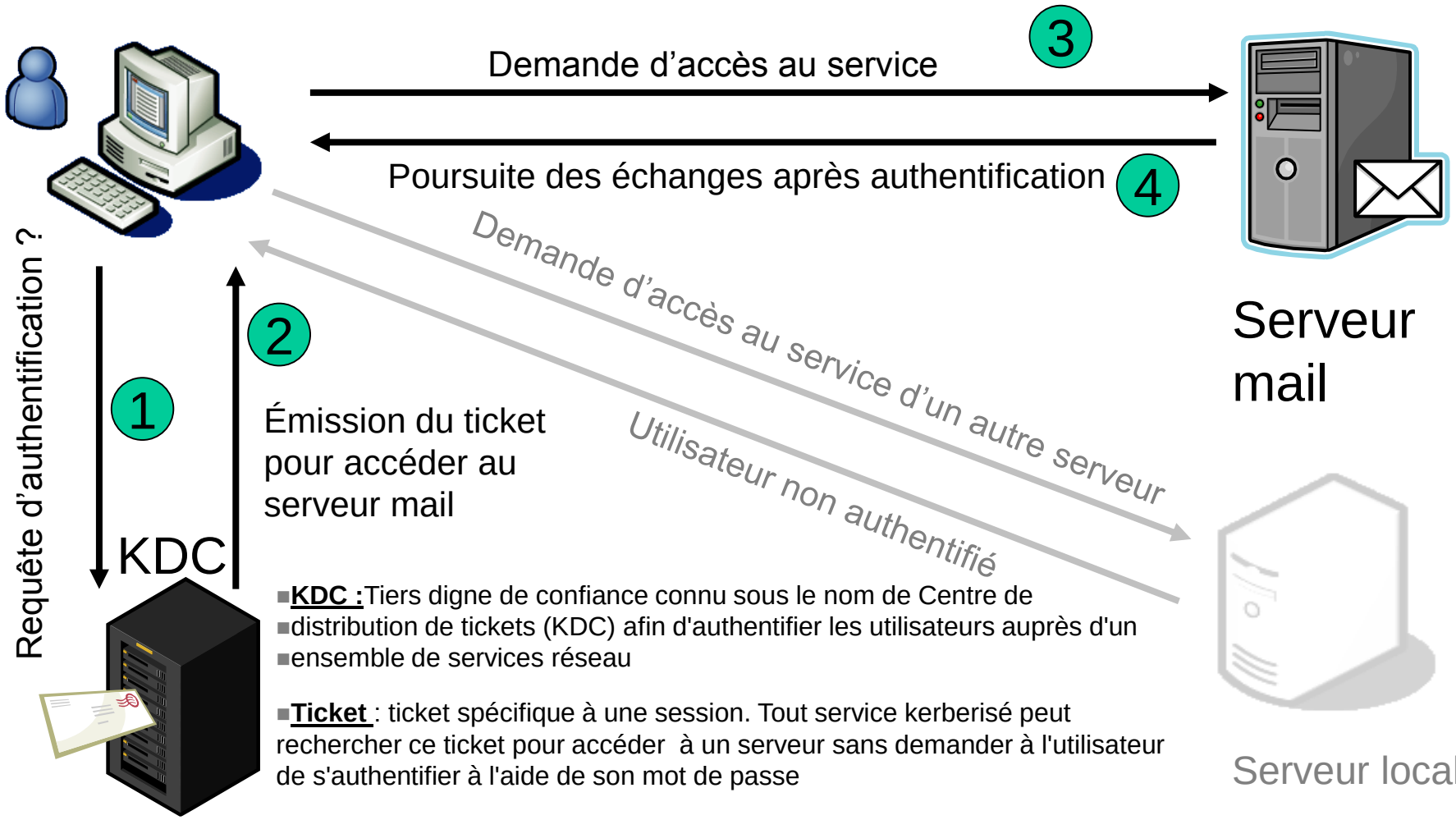


- Serveur : service demandé + service d'authentification
- Mot de passe envoyé sur le réseau
  - en clair ou chiffré
  - possibilité d'interception (WiFi, réseau locale, etc.)

Solutions de sécurité : Kerberos

- Kerberos est basé sur trois parties de confiance
  - Le client :utilisateur ou application sur une machine A
  - Le serveur : application ou ressource accessible par le client sur une machine B
  - Le tiers de confiance.
    - Le AS (Authentication Service)
    - Le KDC (Key Distribution Center)
    - Le KDC centralise les mots de passe de l'ensemble des entités.
- Port 88 (TCP/UDP) entre client et KDC

- Scénario Kerberos

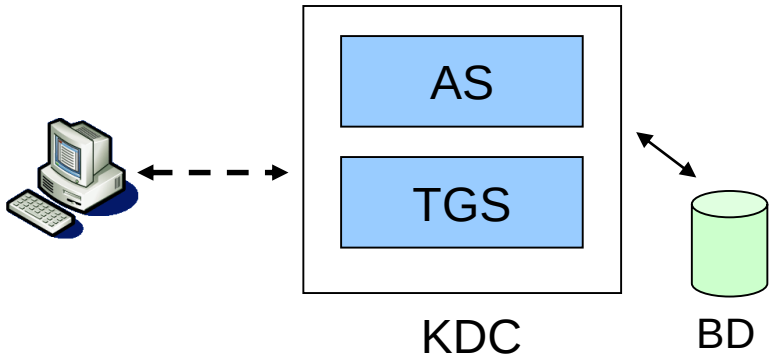


- **KDC :** Tiers digne de confiance connu sous le nom de Centre de distribution de tickets (KDC) afin d'authentifier les utilisateurs auprès d'un ensemble de services réseau

- **Ticket** : ticket spécifique à une session. Tout service kerberisé peut rechercher ce ticket pour accéder à un serveur sans demander à l'utilisateur de s'authentifier à l'aide de son mot de passe

Solutions de sécurité : Kerberos

- Terminologie Kerberos
  - Client : utilisateur ou programme pouvant recevoir un ticket
  - Principal (ou nom principal)
    - Nom unique d'un utilisateur ou d'un service autorisé à s'authentifier à l'aide de Kerberos.
    - Exemple : root[/instance]@REALM, joe/admin@exemple.com
  - Realm
    - Ensemble de machines protégées par Kerberos
    - Réseau composé d'un KDC et d'un ensemble de clients
  - Centre de distribution de clés (KDC)
    - Key Distribution Center
  - Service d'émission de tickets (TGS)
    - Ticket-granting Service
  - Ticket d'émission de tickets (TGT)
    - Ticket-granting Ticket
  - Service
    - Programme accessible sur le réseau
  - Password database : base de données de mots de passe UNIX standard, comme /etc/passwd ou /etc/shadow
  - Serveur d'authentification (AS)



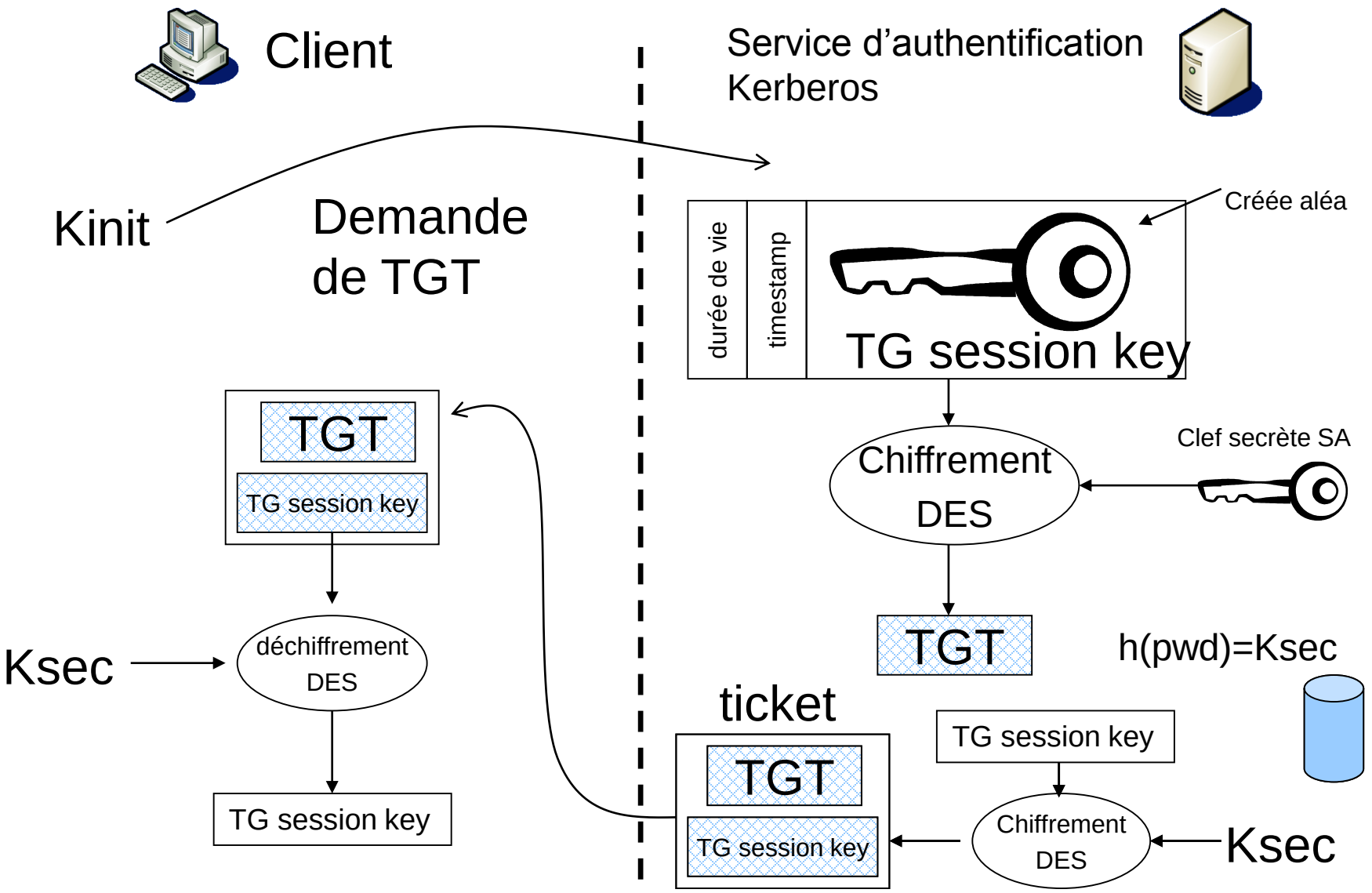
Solutions de sécurité : Kerberos

- AS –Authentication Service
- KDC – Key Distribution Server
- A – Le client
- B – Le service
- K_{KDC} - Clé secrète du KDC
- K_A - Clé du client dérivée du mot de passe
- K_B - Clé du Service dérivée du mot de passe
- K_{TG} - Clé de session de A
- K_{AB} - Clé de session de partagée par A et B

Solutions de sécurité : Kerberos

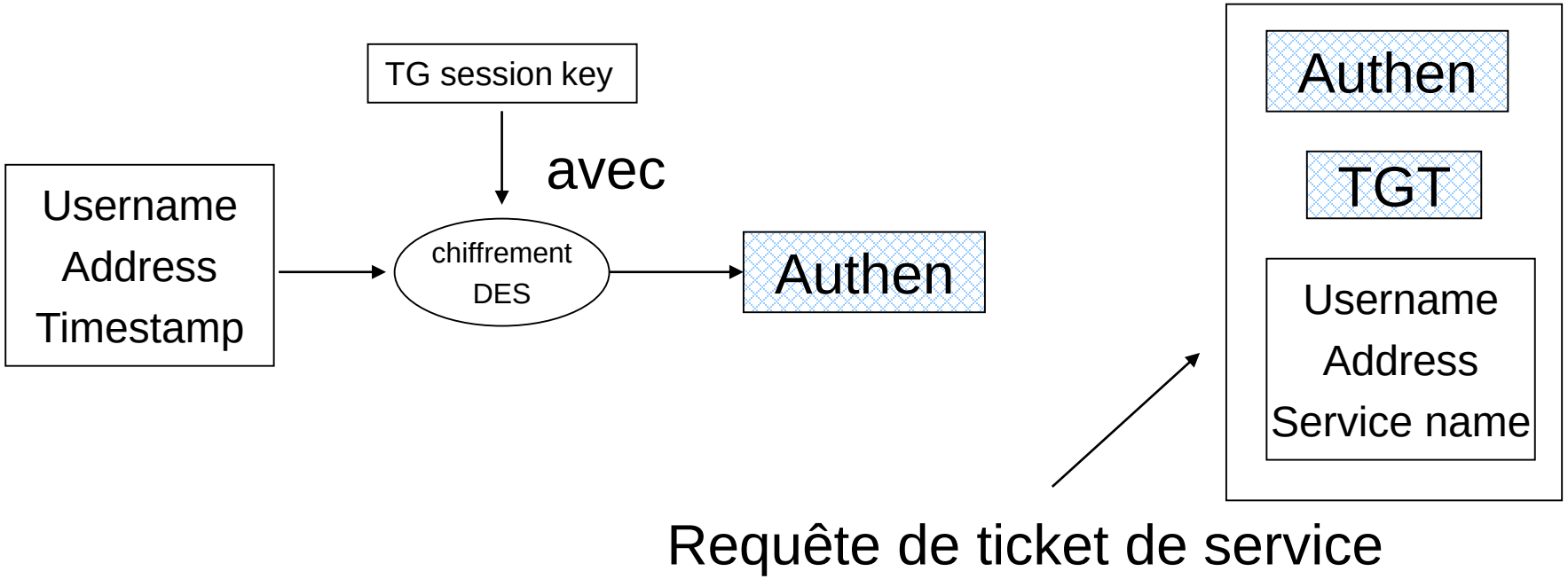
- Le Client s'authentifie auprès de AS et reçoit un TGT (Ticket-Granting Ticket)
  - $A \rightarrow AS: A$
  - $AS \rightarrow A : A, \{K_{TG}\}K_A, TGT$
  - $TGT = \{date, TTL_t, K_{TG}, A\}K_{kdc}$
- Le client présente son TGT au service de délivrance des tickets pour accès à un service et reçoit un TGS (Ticket-Granting Service)
  - $A \rightarrow KDC: \{A, adr., ts.\}K_{TG}$,TGT ,A ,adr.,B
  - $KDC \rightarrow A : \{K_{AB}\}K_{TG}, TGS$
  - $TGS = \{A, B, adr.,d, ts, K_{AB}\}K_B$
- Le client présente son TGS au service et partage ainsi une clé de session avec le service.

Solutions de sécurité : Kerberos



Solutions de sécurité : Kerberos

- TGT peut être intercepté
- Comment peut-on protéger le TGT ?
  - Le client crée un authentificateur
  - Paramètres *username, address, timestamp*



Solutions de sécurité : Kerberos

1 Requête de ticket de service

Authen | TGT | U, A, S

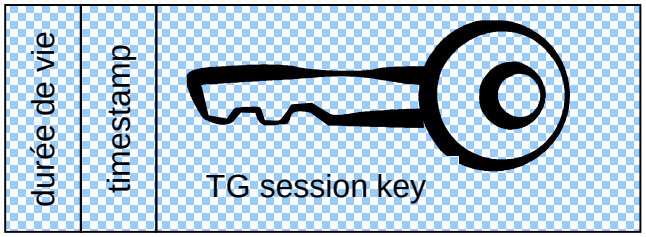
Chiffré par clé secrète session

Chiffré par clé sécrète SA ou KDC

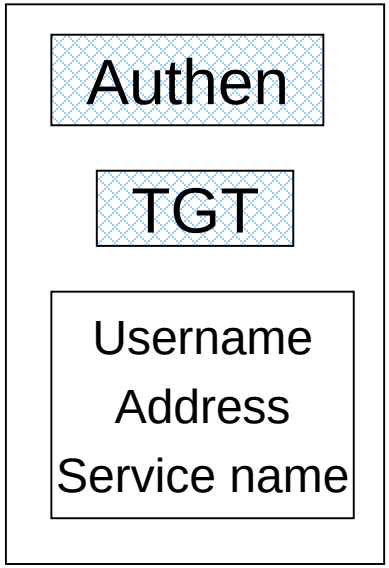
Service TGS du KDC

Après la réception de la requête :

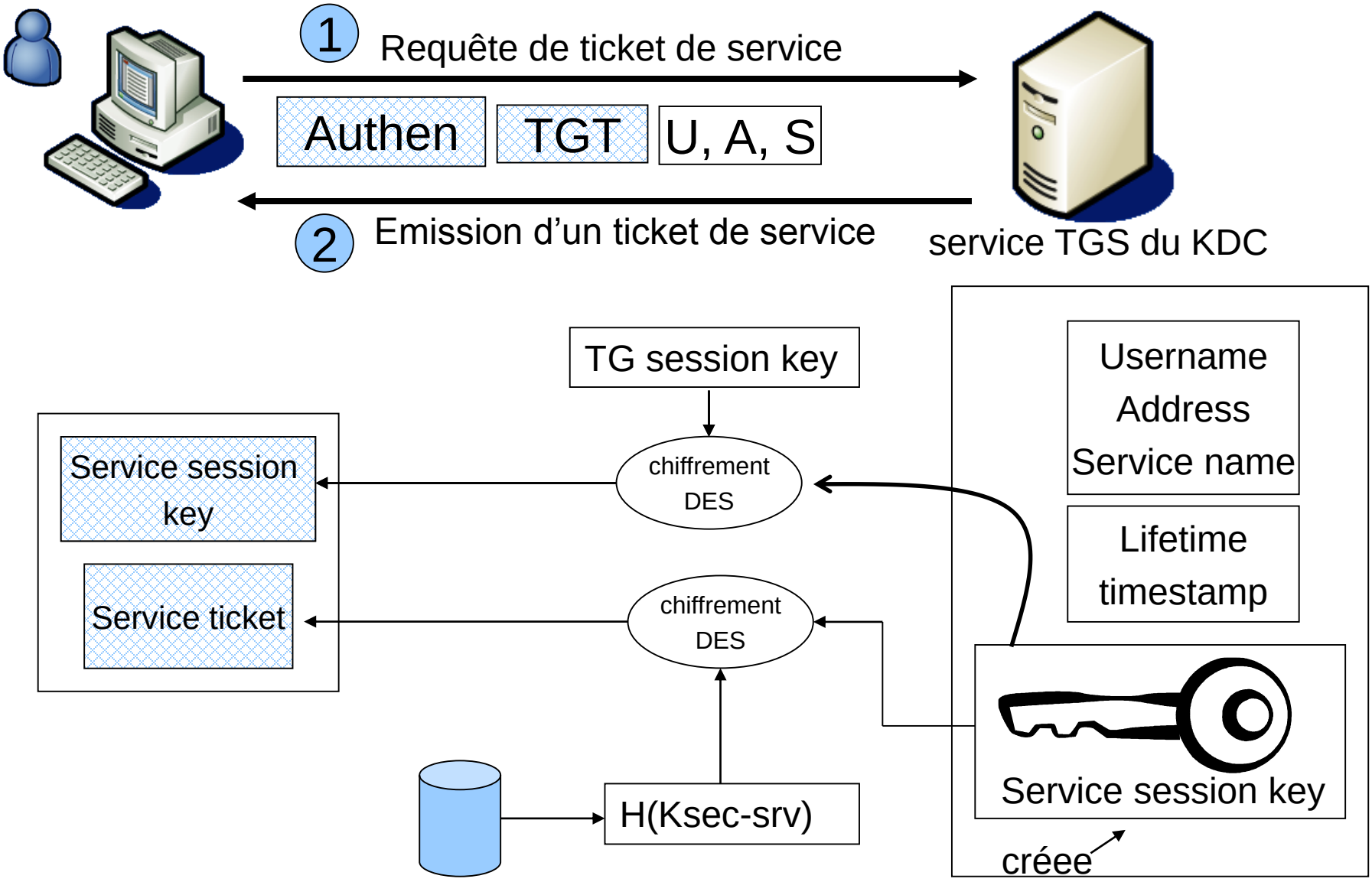
- Le service TGS du KDC décrypte le TGT qui avait été crypté par sa clef personnelle
- si le décryptage réussit, cela prouve son authenticité
- Dans le TGT figure la clef de session avec laquelle le client a crypté l'authentificateur : il peut alors le décrypter



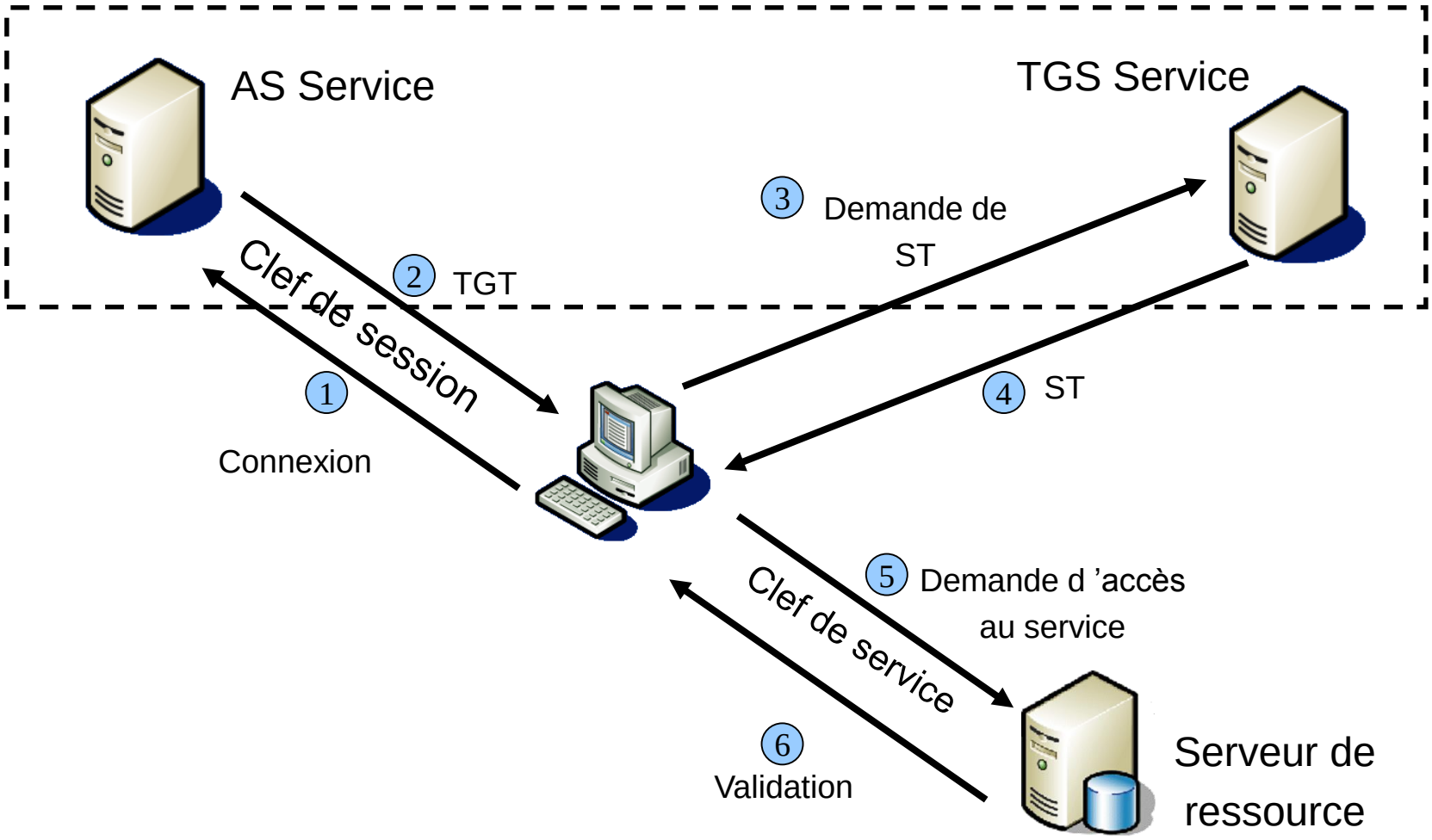
TGT chiffré



Solutions de sécurité : Kerberos



Solutions de sécurité : Kerberos



Solutions de sécurité : Kerberos

- Microsoft
  - Transparent à l'utilisateur
- Heimdal
  - Version libre sous Unix
  - http://www.pdc.kth.se/heimdal
- SEAM
  - Sun Entreprise Authentication Mechanism
- MIT
  - http://web.mit.edu/kerberos

Fonctions à sens unique

- Les fonctions à sens unique sont à la base de toutes les techniques cryptographiques modernes.
- Les fonctions à sens unique sont :
  - Chiffrement asymétrique
  - Fonction de hachage (avec ou sans clé secrète)
  - Protocole de Diffie-Hellman (échanges de clés)
- Une fonction à sens unique f de E vers F est une fonction telle que :
  - pour $x \in E$; f(x) est facilement calculable
  - pour $y \in F$; il est calculatoirement difficile de trouver $x \in E$ tel que $f(x) = y$

Fonctions de hachage

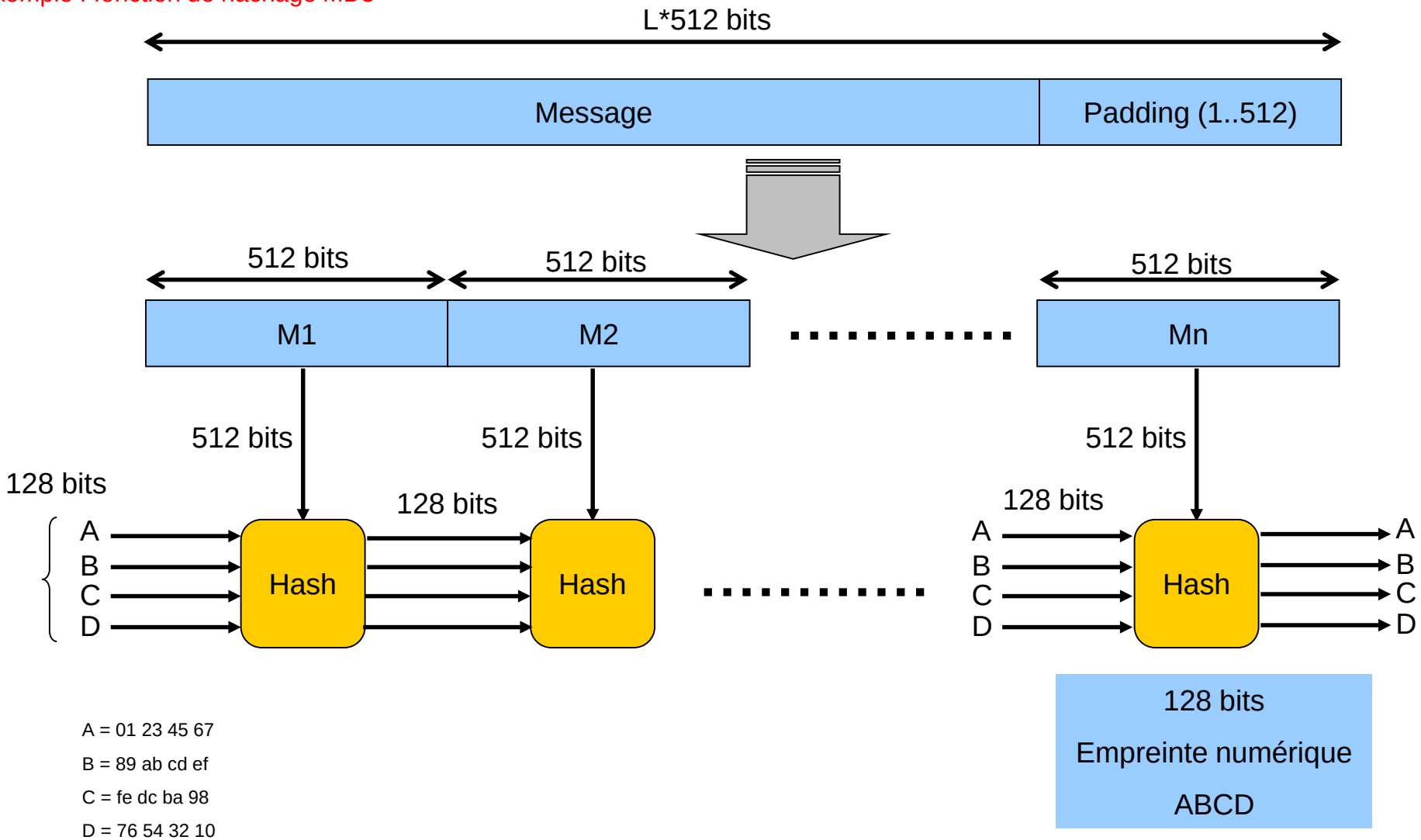
- Hachage : propriétés
  - Pour X_1 avec $H(X_1) = C_1$,
    - il est difficile de trouver un X_2 != X_1 telle que $H(X_2) = C_1$
  - Entrée de taille arbitraire et le résultat a une taille fixe 16, 20 oct.
    - On l'appelle le condensât, le message digest, l'empreinte, le fingerprint
- Exemple:
  - Fichier en entrée openssl.exe (taille 225 280 octets)
    - Sortie MD5(openssl.exe)= 1c:bd:cb:76:92:40:d0:53:99:e7:dd:7c:59:cf:04:44
  - Fichier en entrée copenssl.exe avec un carac. espace en plus (taille 225 281 octets)
    - Sortie MD5(copenssl.exe)= 73:61:18:df:2d:28:6e:f9:67:21:ad:1a:6a:ad:e7:f7

Fonctions de hachage sans clé

- Fonction de Hachage sans clé
  - A la base des fonctions de hachage à clés
  - Signatures numériques (avec le chiffrement asymétrique)
  - mot de passe: stockage des hachés
- Message Digest : MD2, MD4 et MD5.
  - Développé par Ron Rivest pour la RSA Security
  - **http://www.ietf.org/rfc/rfc1319.txt ;rfc1320.txt et rfc1321.txt.**
- RACE Integrity Primitives Evaluation Messages Digest
  - RIPEMD-128 et RIPEMD-160.
  - Développé par H. Dobbertin, A. Bosselaers et B. Preneel
  - http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html
- Secure Hash Algorithm SHA1 (standard SHS).
  - Développé par le NIST en 1995; ANSI X9.30.
  - http://www.itl.nist.gov/fipspubs/fips180-1.htm

Fonctions de hachage sans clé

Exemple : fonction de hachage MD5



A = 01 23 45 67

B = 89 ab cd ef

C = fe dc ba 98

D = 76 54 32 10

Fonction de hachage avec clé

- Fonction standardisée à l'IETF RFC2104 et au NIST FIPS PUB 198
- Hmac : Keyed-hash Message Authentication Code
- Fonction à trois paramètres : hmac (h, Ks, M) = mac
  - h : Fonction de hachage
  - Ks : Une clé secrète
  - M : Un message ou data
- Le résultat d'une fonction hmac se nomme un mac:
  - mac (message authentication code)
  - La taille du mac est identique à la taille du condensât de h
    - 20 octets pour SHA-1 et 16 octets pour MD5
- Fonction usuelle dans les protocoles SSL, TLS, IPsec pour les services d'authentification de l'origine des données et d'intégrité

Fonction de hachage avec clé

- ***hmac*** **(*h, K_s, M) = h (K_s XOR *opad* ||*h*(K_s XOR *ipad* ||*M*))**
  - ***ipad = octet 0x36 et opad = octet 0x5C***
- Soit ***B*** la taille des blocs ***(B = 64 octets)***
  - La taille de B dépend de la fonction h
- La taille de K_s doit être égale à celle de ***B***
  - Si la taille de K_s est inférieure à celle de ***B,*** un pad de ***0***
- Si la taille de K_s est supérieure à la taille du condensât de ***h***, on lui applique la fonction ***h***

Applications basées sur les fonctions de hachage

- One Time Password
  - Objectives:
    - Lutter contre les écoutes et le rejeu du mot de passe
      - N'empêche pas l'écoute mais empêche le rejeu
  - Un mot de passe à usage unique
    - Fonctionne avec toutes solutions à base de mot de passe
  - Nécessite une coordination entre deux entités
    - Explicite (Asynchrone) = Modèle Challenge/réponse
      - Basée sur un ou plusieurs paramètres échangés dans un sens
    - Implicite (Synchrone)
      - basée sur un numéro de séquence ou une horloge

Applications basées sur les fonctions de hachage

- One Time Password : propriétés
  - Pour la mise en œuvre de l'authentification simple
  - Nécessite une personnalisation par une phrase secrète
  - Parade structurelle contre les attaques par rejeu
    - A différencier avec l'attaque de l'homme au milieu, qui elle dépend plus du protocole d'authentification
  - Renforce les mots de passe
    - Parade contre l'usurpation des mots de passe par écoute
  - Permet une gestion des mots de passe
    - limiter la durée de vie de la phrase secrète partager
  - Capacité d'intégration aux protocoles
    - Traitement indépendant des protocoles d'authentification
  - Capacité de choix de la fonction de hachage
    - déterminant pour le niveau de robustesse

Applications basées sur les fonctions de hachage

- One Time Password
  - N'est pas un protocole d'authentification
    - Deux instances nécessaires pour l'authentification mutuelle
  - Ne concerne pas la protection du secret partagé (phrase secrète)
    - La protection du secret est de l'ordre de l'implantation
  - N'est pas une solution fermée
    - Possibilité de la renforcer par l'ajout de nouveaux paramètres
  - Plusieurs réalisations sont possibles
    - Problématique de l'interopérabilité
  - Ne permet pas la non-répudation
    - Pas de preuve au sens juridique

Applications basées sur les fonctions de hachage

- One Time Password : standards IETF
  - Groupe de travail de l'IETF en 2001
    - Pour proposer une alternative à S/KEY marque déposé de Bellcore
    - Pour intégrer de nouvelles fonctions de hachage
  - RFC 2289 (Obsoletes 1938): " A One-Time Password System", statut « standard »
    - La référence actuelle au niveau des implantations.
  - RFC4226 "HOTP: An HMAC-Based One-Time Password Algorithm", , statut « informational »
    - L'OATH (initiative for Open AuTHentication) basés sur un compteur et une fonction hmac HOTP (HMAC One Time Password) (Verisign, Aladdin, SafeHaus, etc.)
    - http://www.openauthentication.org/

Applications basées sur les fonctions de hachage

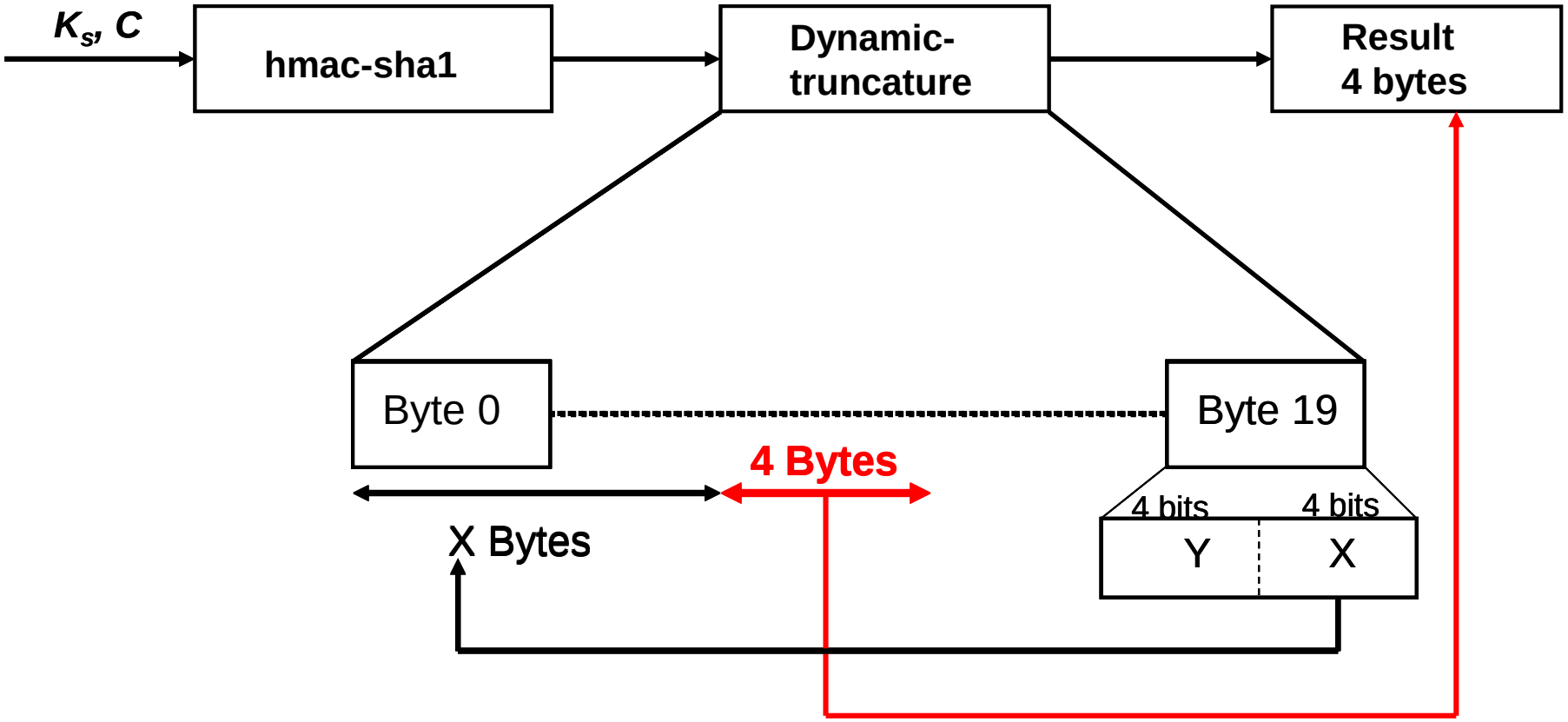
- RFC4226 “HOTP: An HMAC-Based One-Time Password Algorithm”, statut « informational »
- Trois phases:
  - Initialisation: Génération du secret partagé Ks
    - Une seule fois
    - Personnalisation du client
  - Génération: d’un mot de passe à usage unique (nombre sur 6 digits)
    - Basé sur Ks et un compteur (incrémenté à chaque génération)
    - Mode synchrone avec resynchronisation si nécessaire
    - Basé sur une fonction hmac
  - Authentification: sur la base de mot de passe à usage unique
    - Un nombre (résultat de la conversion de 4 octets)

Applications basées sur les fonctions de hachage

- HOTP est basé:
  - Sur une fonction hmac
  - Pour l'interopérabilité des différents systèmes d'authentification:
    - un choix figé de la fonction de hachage et de la sémantique des paramètres
  - La fonction hmac à la base de HOTP est composée de:
    - Sha-1 fonction de hachage
    - Ks clé secrète
    - C un compteur (en guise de data)

- HOTP

Calcul du HOTP sur la base du *hmac-sha1*



Applications basées sur les fonctions de hachage

- HOTP : Conversion des 4 bytes en un nombre ***DBC (Dynamic Binary Code)***

$$otp\text{-}passe = DBC \bmod 10^{d} \text{ avec } (d = 6)$$

Exemple:

Résultat du hmac

07 67 AE 34 67 B0 54 30 A1 56 07 67 AE 34 67 B0 54 30 A1 56

dynamic binary code DBC

DBC = 54 30 A1 56 en décimale = 1412473174

otp-passe = 473174

Applications basées sur les fonctions de hachage

- Le compteur HOTP
  - Le compteur du serveur est incrémenté après chaque authentification réussie
  - Le compteur du jeton est incrémenté après chaque demande de l'utilisateur
    - nécessité de synchronisation
- Le serveur HOTP
  - peut calculer la prochaine valeur HOTP et vérifier la valeur HOTP reçu par le client
- Le système peut exiger à l'utilisateur d'envoyer un e séquence de valeurs HOTP pour la resynchronisation

Applications basées sur les fonctions de hachage

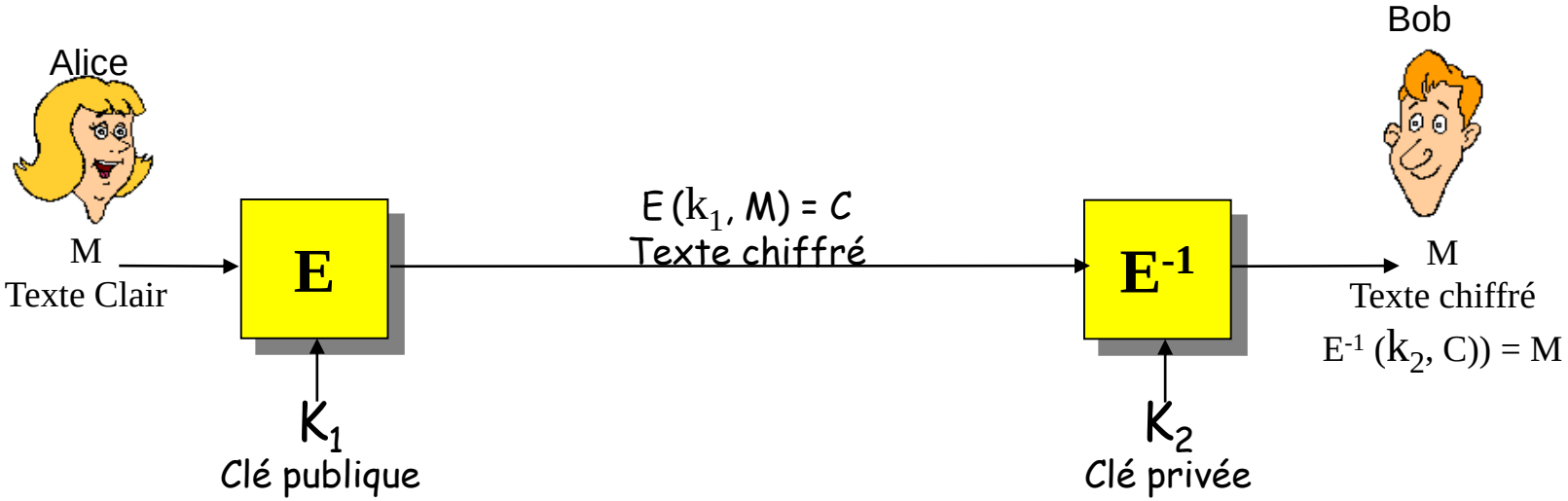
	S/KEY	OTP	HOTP
Standard	RFC1760	RFC2289	RFC4226
Date	Février 1995	février 1998	Décembre 2005
Taille de la passe phrase	Doit être supérieur à 8 caractères	Doit être supérieur à 9 caractères	Pas de phrase
Fonction de hachage	MD4, MD5	MD4, MD5, SHA1	SHA-1
Fonction de hachage avec clé	NON	NON	OUI - 128 bits min. 160 bits conseiller
Implantation	Disponible	Disponible	Disponible
Niveau interopérabilité	Moyen	Faible	Élevé
Marque déposé	OUI (Bellcore)	NON	NON

Cryptographie asymétrique

- 1976: Whilfried Diffie et Martin Helman
  - Définissent le concept de la cryptographie asymétrique
  - Deux clés: l'une chiffre et l'autre déchiffre
- 1977: Ron Rivest, Adi Shamir, Léonard Adleman
  - Définissent RSA: un algorithme asymétrique
    - basé sur la factorisation des nombres premiers
    - breveté par le MIT en 1983 (expiration en 2000)
    - commercialisé par la société RSA
    - Le plus déployé parmi les algorithmes asymétrique
    - Obligatoire dans plusieurs protocoles (SSL/TLS,PGP, SET,..)
    - Intégré à presque toutes les cartes de paiement
- Le NBS lance un appel : alternative de RSA
  - Adoption de ELGAMAL (DSA)
    - basé sur la complexité calculatoire des logarithmes discrets.
    - Complexité comparable à RSA

Cryptographie asymétrique

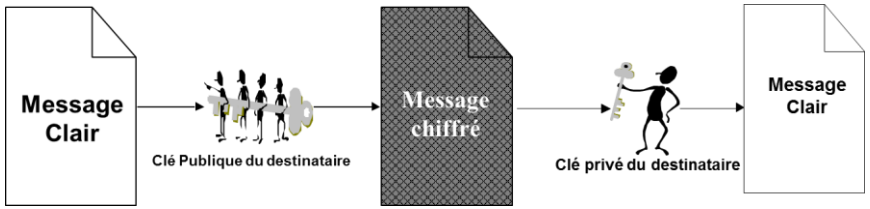
- Fonction basée sur la complexité de calcul
  - Relativement plus lente que le chiffrement symétrique
  - Chiffrement de 100 à 1000 fois plus lents à résistance égale
- Deux clés: si l'une chiffre l'autre déchiffre



- Il existe une relation unique entre les deux clés (K_1 et K_2)

Cryptographie asymétrique

- Alice et Bob vont disposer chacun d'une paire de clés uniquement
- Une clé va être rendue publique: Clé publique
  - Elle sera publiée notamment dans un annuaire
- Une clé restera privée: Clé privée
  - Cette clé doit être du domaine du privée et bien protégée
  - Elle ne doit jamais être



    - divulguée à un tiers
    - stockée en clair sur un support quelconque
    - échangée au travers du réseau en chiffré ou en clair

Cryptographie asymétrique - RSA

- Pour former les couples (e,n) et (d,n)
  - On choisit au hasard 2 grands nombres premiers p et q
  - On calcule n = p.q
  - On pose φ(n) = (p-1).(q-1)
  - On sélectionne e tel que : PGCD(e, φ(n))=1
  - On calcule d tel que :
    - e.d = 1 mod φ(n) (e et d sont inverses l'un de l'autre modulo φ(n))

Cryptographie asymétrique - RSA

- Clé Publique est :
  - le couple (e,n)
- Clé Privée est :
  - le couple (d,n)
- Soit M le message clair et C le message chiffré
  - Pour chiffrer M, on calcule :
    - $C = M^e$ modulo n
  - Pour déchiffrer on calcule :
    - $M = C^d$ modulo n

Cryptographie asymétrique - RSA

- Exercice

Alice

BoB

p= 13, q= 17
K_A^+ = ?
K_A^- = ?
e=5
C= ?, S= ?

p= 3, q= 11
K_B^+ = ?
K_B^- = ?
e=3
M= ?, V= ?

Message à envoyer M
= « a », 97 en ASCII

Calculer K_A^+ et K_A^- d'Alice.
Calculer K_B^+ et K_B^- de BoB.
Calculer le cryptogramme et la valeur de la signature ?
Comment Bob déchiffre C et vérifie S ?

Cryptographie asymétrique - RSA

- Exemple:
  - p = 3, q =11
  - n = p.q = 3 x 11 = 33
  - j = (p-1)(q-1) = 2 x 10 = 20
- Pour e = 3, d = 7
  - Car (e.d = 1 mod j): 3 x 7 = 1 mod j
- Pour un message: M = 29
  - Chiffrement:
    - $Y = M^e \bmod n = 29^3 \bmod 33 = 2$
  - Déchiffrement:
    - $Y^d \bmod n = 2^7 \bmod 33 = 29$

Cryptographie asymétrique - RSA

- Deux clés: si l'une chiffre, l'autre déchiffre
- Pour le même message: M = 29
  - Chiffrement avec la clé privée:
    - $Z = M^d \bmod n = 29^7 \bmod 33 = 17$
  - Déchiffrement:
    - $Z^e \bmod n = 17^3 \bmod 33 = 29$

Cryptographie asymétrique - RSA

- Choisir au hasard 2 nombres premiers
  - Ex : p = 13 et q = 17
  - Calculer n = p.q = 13*17=221
  - On pose j = (p-1).(q-1) = 12*16 = 192
  - Sélectionne e
    - e et j soient premiers entre eux avec 1 < e < j
    - « Deux entiers a et b sont premiers entre eux, s'ils n'ont aucun facteur en commun »
    - On choisit e = 5
  - Clé publique : (221, 5)
  - On calcule d tel que :
    - e.d = 1 mod j => 5.d = 1 mod 192
  - clé privée d = 77

Cryptographie asymétrique - RSA

- Clé publique (e,n) = (5,221)
- Clé privée d = 77
- M est le message à chiffrer : « bonjour »
- b= 98, o= 111, n= 110, j= 106, u= 117, r= 114
- Chiffrement
  - C = M^e modulo n
    - $C1=98^5 \bmod 221 = 115$
    - C2=76 , C3=145, C4=123, C5=76, C6=104, C7=173
    - C=**sLæ{Lhi**

Plaintext	b	o	n	j	o	u	r
Code	98	111	110	106	111	117	114
Ciphertext	s	L	æ	{	L	h	i
Code	115	76	145	123	76	104	173

Cryptographie asymétrique - RSA

- Déchiffrement
  - $M = C^d$ modulo n
    - $M1 = 115^{77} \bmod 221 = 98$ ……b
    - $M2 = 76^{77} \bmod 221 = 111$ ……o
    - $M3 = 145^{77} \bmod 221 = 110$ ……..n
    - $M4 = 123^{77} \bmod 221 = 106$ ……..j
    - $M5 = 76^{77} \bmod 221 = 111$…….o
    - $M6 = 104^{77} \bmod 221 = 117$……..u
    - $M7 = 173^{77} \bmod 221 = 114$……r

Plaintext	b	o	n	j	o	u	r
Code	98	111	110	106	111	117	114
Ciphertext	s	L	æ	{	L	h	i
Code	115	76	145	123	76	104	173

Cryptographie asymétrique - RSA

- Preuve que $C^d = M \bmod n$:
  - e.d = 1 mod j, donc il existe un entier k tel que e.d = 1 + kj.
  - Ainsi, $M^{e.d} = M^{1+kj} = M^{1+k(p-1)(q-1)}$.
  - Or, comme M et p sont premiers entre eux,
  - d'après le théorème de Fermat, $M^{p-1} = 1 \bmod p$.
  - Comme ici, p et q sont premiers, on a :

$$M^{1+k(p-1)(q-1)} = M \bmod p$$

$$\text{et } M^{1+k(p-1)(q-1)} = M \bmod q$$

Cryptographie asymétrique - RSA

- D'où, il existe deux entiers a et b tels que :
- $M^{e.d} = M + a.p = M + b.q$ par conséquent : a.p = b.q, p divise b.q, comme il est premier avec q, il divise b, il existe donc un entier c tel que a = c.p
- on a : $M^{e.d} = M + c.p.q$ avec p.q = n,
- d'où $M^{e.d} = M \bmod n$,
- comme $M^e = C \bmod n$,
- on a bien ce que l'on voulait démontrer : $C^d = M \bmod n$.

Cryptographie asymétrique - RSA

- Pour percer RSA, il faut pouvoir factoriser n.
  - Si on factorise n, on obtient p et q,
    - on calcule j = (p -1)(q - 1)
    - on calcule d, e.d = 1 mod j
- La factorisation de grands nombres est complexe
  Deux difficultés pour implémenter RSA:
  - La génération de grands nombres premiers (p et q)
  - L'exponentiation avec de grand facteur.
    - Un standard est défini: PKCS 1 (Public Key Cipher System).

Cryptographie asymétrique - RSA

- Pour briser RSA, il faut calculer l'exposant de déchiffrement
  - $d=e^{-1} \bmod (p-1)(q-1)$ où $pq=n$
  - Mais, il faut factoriser n !!!
  - Très difficile
  - Exemple !!!
    - N = 31074182404900437213507500358885679300373460228427275457201619488232064405180815045563468296717232867824379162728380334154710731085019195485290073377248227835257423864540146917366024776523466 09
    - P= 16347336458092538484431338838650908598417836700330 923121811108523893331001045081512121181675115 79
    - Q = 19008712816648221131268515739354139754718967899685154936666385390880271038021044989571912614655 71

Cryptographie asymétrique - RSA

- Recommandations
  - Ne jamais utiliser de valeur *n* trop petite
  - Ne pas utiliser de clé secrète trop courte (< racine n)
  - N'utiliser que des clés fortes ($p - 1$ et $q - 1$ ont un grand facteur premier)
  - Ne pas utiliser un *n* communs à plusieurs clés

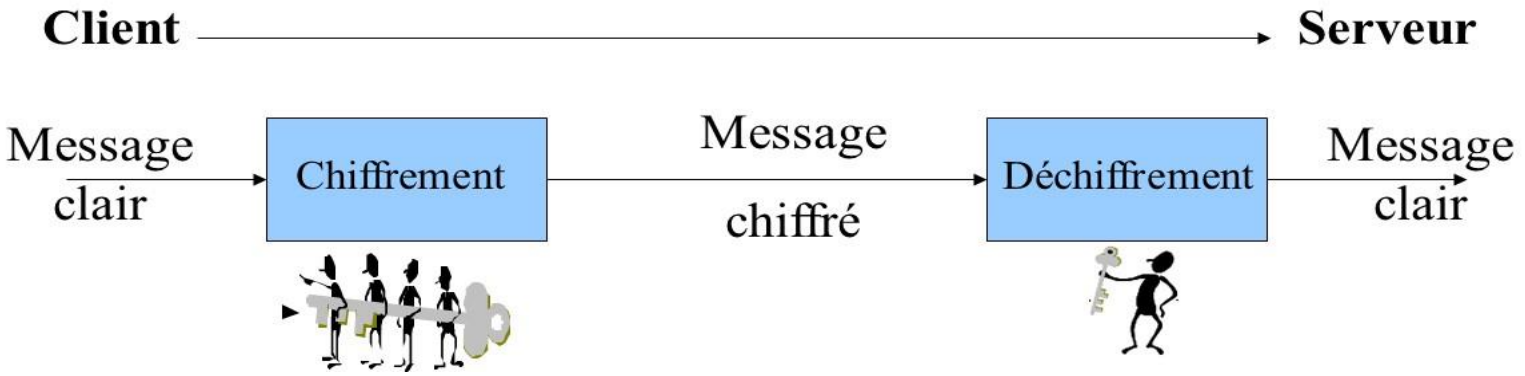
Cryptographie asymétrique - RSA

- Exercice
  - Bob et Bernard ont pour clé publique RSA respectivement (n, e1) et (n, e2)
  - e1 et e2 premiers entre eux
  - Alice envoie le même message *m* crypté par les clés publiques RSA de Bob et Bernard
  - Eve intercepte les deux messages cryptés et trouve m
  - Application numérique : m=2, n=21, e1=5, e2=13

- Solution
  - $C_1 = 2^5 \bmod 21 = 11$, $C_2 = 2^{13} \bmod 21 = 8$, $e1.u + e2.v = 1$, $u=-5$ et $v=2$
  - $C_1^u.C_2^v \bmod n = m^{e1.u} . m^{e2.v} \bmod n = m^{e1.u + e2.v} \bmod n = m$
  - $C_1^u.C_2^v \bmod n = m^{e1.u} . m^{e2.v} \bmod n = m^{e1.u + e2.v} \bmod n = m$
  - $C_1^u.C_2^v \bmod n = (2^5)^{-5} . (2^{13})^2 \bmod 21 = 2^1 \bmod 21 = 2$

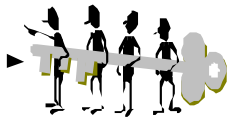
Cryptographie asymétrique – ElGamal

- Algorithme ElGamal
  - Publié par Tahar El Gamal en 1987
  - Sa sécurité dépend de la difficulté de calculer les logarithmes discrets ($3^k \equiv 5 \pmod 7$ => K=?)
  - Utilisé pour le chiffrement et la signature électronique.
  - Utilisé par le logiciel libre GNU Privacy Guard, et PGP (Pretty Good Privacy)



Cryptographie asymétrique – ElGamal

- Exemple
  - p=11, g=6, x=8
  - $y=6^8 \pmod{11}=4$
  - Public : 4, 6, 11
  - Private : x=8



Clef publique : p, g et y

Clef privée : x

Public key:

p (a prime number)
$g, x < p$ (two random numbers)
$y \equiv g^x \pmod{p}$
y, g and p: public key

Private key:

$x < p$

Enciphering:

k: a random number such that $\gcd(k, p-1) = 1$
$r \equiv g^k \pmod{p}$
$s \equiv (y^k \pmod{p})\ (m \pmod{p-1})$

Deciphering:

$m \equiv s/r^x \pmod{p}, 0 \leqslant m \leqslant p-1$

Cryptographie asymétrique – ElGamal

Alice

X=8

Mise en accord publique sur
p=11, g=6 et y=4

Message à chiffrer m =5

Générer k=7 random

$r = g^k \mod p$

$s = y^k \pmod p * m \pmod{p-1}$

Envoie r et s au serveur

$r = 6^7 \mod 11 = 8$

$s = 4^7 \mod 11 * 5 \mod 10 = 25$

(r,s) : (8, 25)

Recevoir r=8 et s=25

$m = s / r^x \mod(p)$

$r^x \mod 11 = 8^8 \mod 11$

16777216 mod11 = 5

m = 25/5 = 5

Pirate : aucune information sur x et k

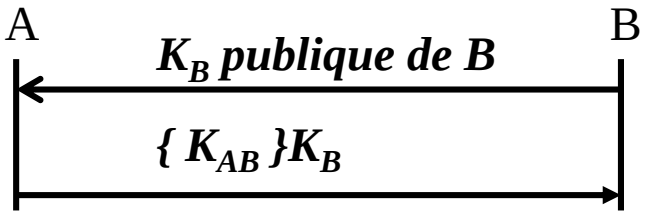
Cryptographie asymétrique – ElGamal

- Avantages
  - Difficile à casser
    - Basé sur les logarithmes discrets
  - Algorithme non déterministe
    - Deux chiffrements du même message M donneront deux messages chiffrés différents
- Inconvénients
  - La taille de ciphertext est 2 fois plus longue que le plaintext
  - El Gamal est 2 fois plus lent que le RSA

Cryptographie asymétrique

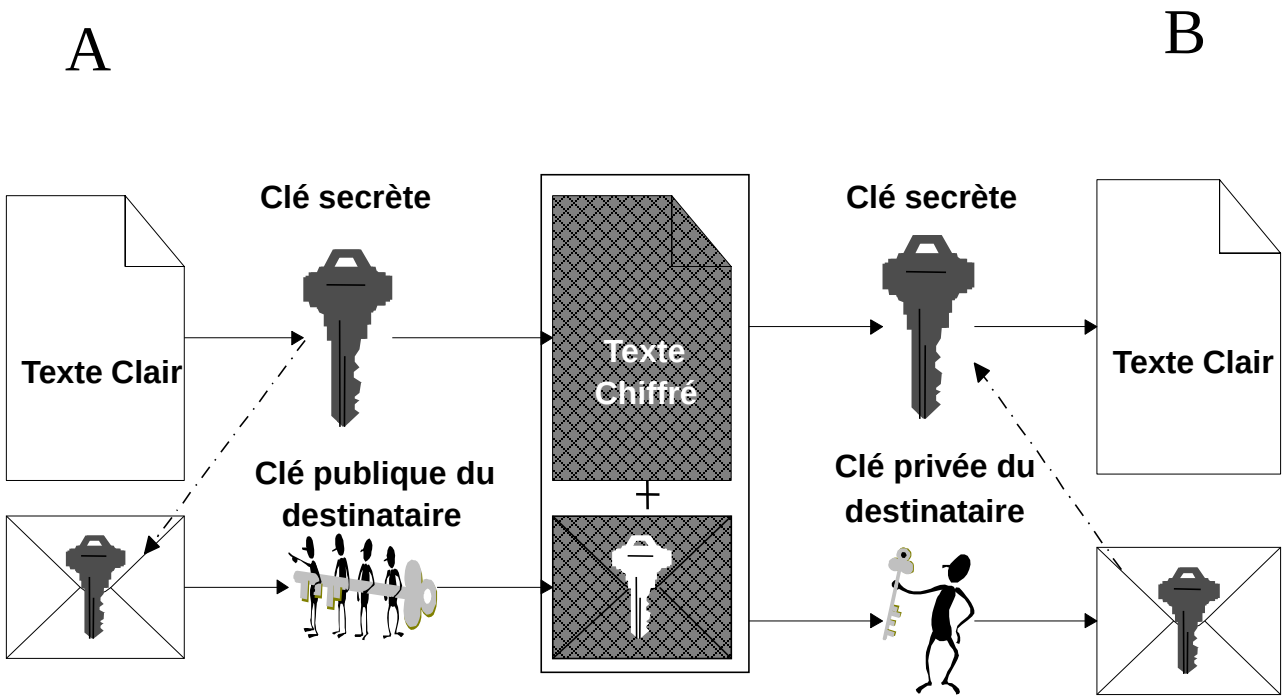
- Pas de service de confidentialité directement:
  - Pour raison de performance
- Service d'échange de clé (symétrique):
  - Pour la mise en œuvre du service de confidentialité
  - Établissement de la clé de session

K_{AB} généré par A



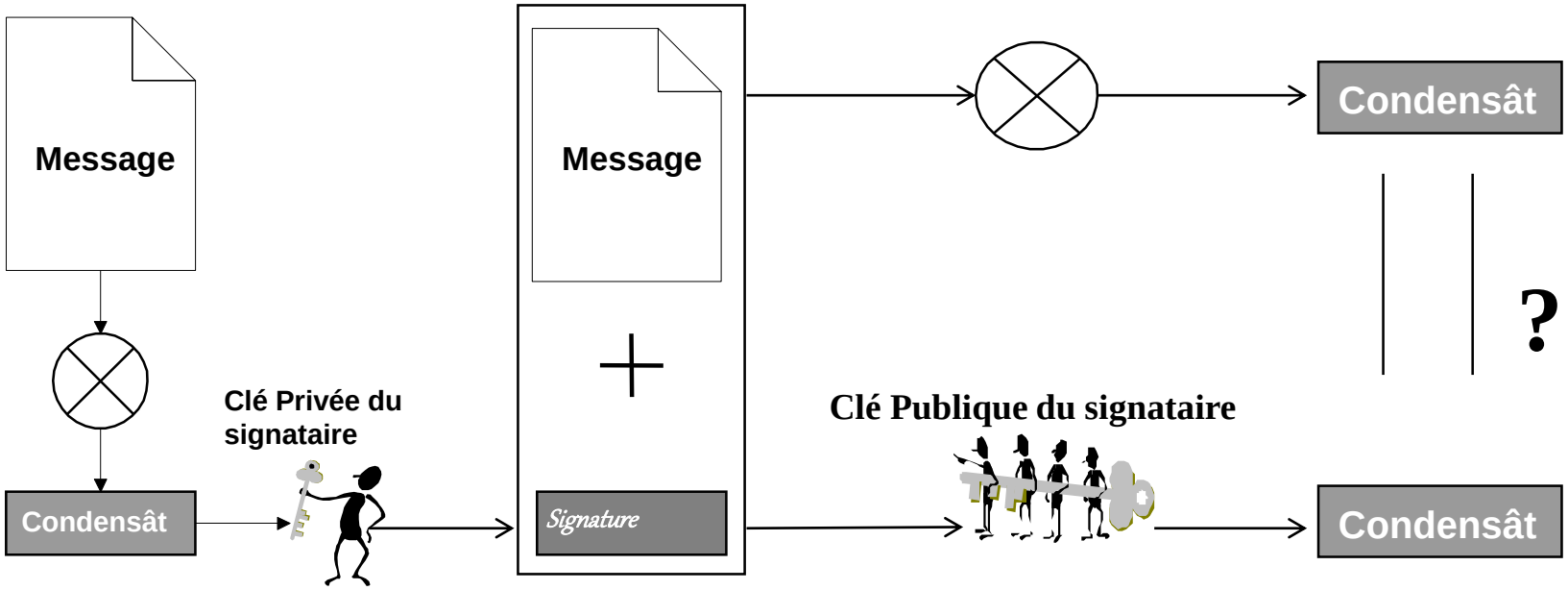
Cryptographie asymétrique

- Échange de clé et service de confidentialité



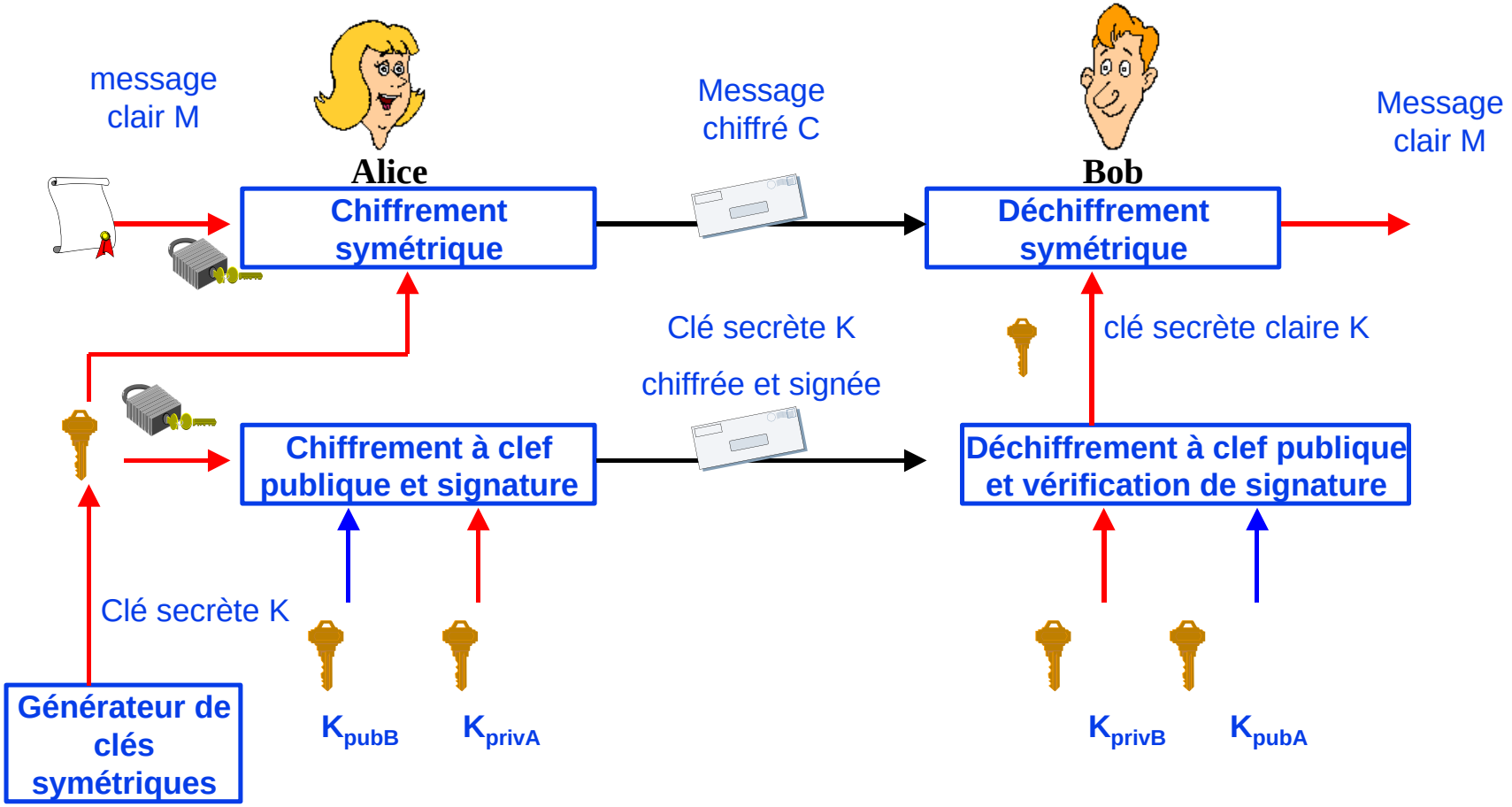
Cryptographie asymétrique

- Service d'intégrité et d'identification
  - La signature numérique



Cryptographies asymétrique/symétrique

- Services de sécurité



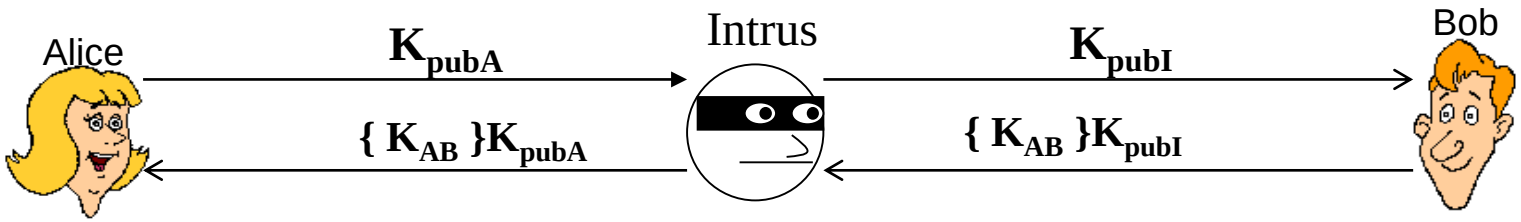
Cryptographies asymétrique/symétrique

- Attaques de l'homme au milieu: MITM

Alice dispose de

K_{pubA} clé publique

K_{privA} clé privée



Nécessité d'authentifier les clés publiques

Nécessité de gérer les clés publiques:

Durée de vie, usages, révocation, …

… déployer une infrastructure de confiance

Fonctions à sens unique : horodatage

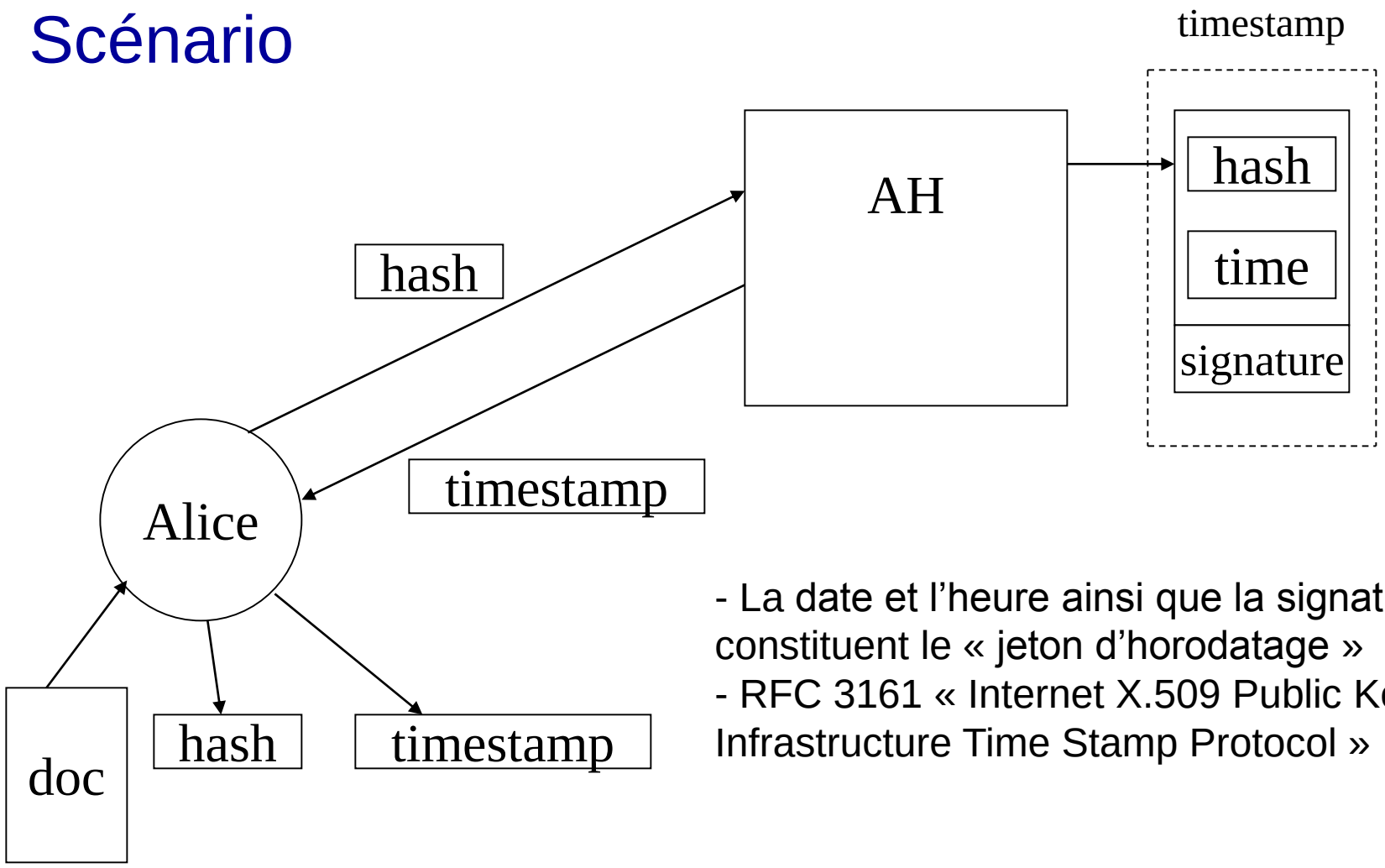
- Horodatage?
  - Certifie qu'un document existait à une certaine date
- Propriétés
  - Indépendant de l'environnement et de la localité physique
  - Pas de changement
  - Base de temps universel

Fonctions à sens unique : horodatage

- Autorité d'horodatage de confiance
  - Alice produit un hash de son document
  - Alice transmet ce hash à l'autorité d'horodatage (AH)
  - AH ajoute la date et le temps et signe le résultat
  - AH envoie le hash horodaté et signé à Alice
  - Alice enregistre le message horodaté

Fonctions à sens unique : horodatage

- Scénario



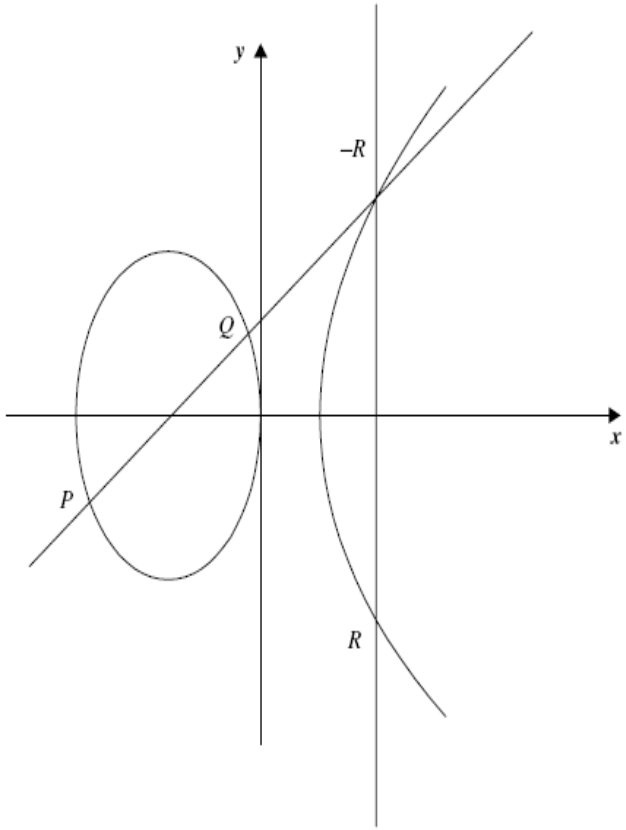
- La date et l'heure ainsi que la signature constituent le « jeton d'horodatage »
- RFC 3161 « Internet X.509 Public Key Infrastructure Time Stamp Protocol »

La cryptographie à courbes elliptiques

- Elliptic curve cryptography (ECC)
  - Concept proposé par deux chercheurs Miller et Klobitz en 1985
  - Alternative à la cryptographie classique à clef publique
- ECC permet de
  - Echanger de clefs sur un canal non-sécurisé
  - Chiffrer les données
  - Signer les données

La cryptographie à courbes elliptiques

- ECC est basée sur
  - Corps de Galois
    - Pour p premier et n entier supérieur ou égal à 1, on appelle $GF(p^n)$ le corps de Galois à p^n éléments de Z/pZ
    - Cet ensemble est muni d'une
      - addition (composantes à composantes)
      - Multiplication dans $GF(p^n)$
  - Equations de Weierstrass
    - Soit a et b dans $GF(2^n)$ avec b non nul. La courbe elliptique E associée est l'ensemble des points (x,y) dans $(GF(2^n))^2$ tels que
      - $y^2=x^3+ax+b$ auquel on adjoint un point spécial O appelé point à l'infini
      - Le discriminant $-(4a^3 + 27b^2) \neq 0$



La cryptographie à courbes elliptiques

- Exemple
  - Soient p = 23 et $E_{23}(1, 1) : y^2 = x^3 + x + 1$
  - Comme nous travaillons dans Zp :
    - $y^2 \bmod p = (x^3 + x + 1) \bmod p$
      - Equation satisfaite pour x=9, y=7
      - $(7)^2 \bmod 23 = (9^3 + 3 + 1) \bmod 23$
      - $49 \bmod 23 = 739 \bmod 23$
      - $3 = 3$
    - les couples (x, y) répondant à l'équation sont

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

La cryptographie à courbes elliptiques

- Addition de 2 points
  - $P(x_P,y_P)$ et $Q(x_Q,y_Q)$ deux points de $E_p(a, b)$
  - On détermine $R = P + Q = (x_R, y_R)$:
    - $x_R = (\lambda^2 - x_P - x_Q) \bmod p$
    - $y_R = (\lambda(x_P - x_R) - y_P) \bmod p$

$$\lambda = \begin{cases} \left(\frac{y_Q - y_P}{x_Q - x_P}\right) \bmod p \text{ si } P \neq Q \\ \left(\frac{3x_P^2 + a}{2y_P}\right) \bmod p \text{ si } P = Q \end{cases}$$

- Multiplication
  - une répétition d'additions (ex : 4P = P + P + P + P)

La cryptographie à courbes elliptiques

- Exemple
  - Soient P = (3,10) et Q = (9,7) dans $E_{23}(1, 1)$
  - P+Q = R(?,?)
    - $\lambda = [(y_P - y_R)/(x_P - x_R)] \bmod 23 = 11$
    - $x_R = (\lambda^2 - x_P - x_Q) \bmod p = (11^2 - 3 - 9) \bmod 23 = 17$
    - $y_R = (11(3 - 17) - 10) \bmod 23 = -164 \bmod 23 = 20$, car -164 = 23*(-8) + 20
    - Donc, P(3,10) + Q(9,7) = R(17,20)
  - 2*P=?, P=Q
    - $\lambda = [(3*(x_P)^2 + a)/(2*y_P)] \bmod 23 = (1/4) \bmod 23 = 6$
    - $x_R = (\lambda^2 - x_P - x_Q) \bmod p = 30 \bmod 23 = 17$
    - $y_R = (6(3 - 17) - 10) \bmod 23 = 21$
    - 2P= (17,21)
    - 3P = 2P+ P = (17,21) + (3,10)

La cryptographie à courbes elliptiques

- Diffie-Hellman en ECC
  - Il faut trouver un problème difficile
    - Ex: factorisation d'un produit en ses facteurs premiers
  - L'échange d'une clé par ECC entre Alice et Bob se déroule comme suit :
    - Alice et Bob se mettent d'accord sur $E_p(a, b)$ et un point de départ $G(x, y)$ dans $E_p(a, b)$ avec un ordre *n* élevé
    - Alice choisit un n_A inférieur à n qui sera sa clé privée et génère sa clé publique $P_A = n_A \times G$.
    - B choisit un n_B inférieur à n qui sera sa clé privée et génère sa clé publique $P_B = n_B \times G$.
    - A génère la clé secrète $K = n_A \times P_B$ et B génère la clé secrète $K = n_B \times P_A$.

résoudre une équation du troisième degré sur $GF(2^n) \Rightarrow$ Problème difficile

La cryptographie à courbes elliptiques

A

En clair

P=23, $E_p(a, b) : y^2 = x^3 + x + 1$, G(x,y)

B

Génère aléatoirement sa clé privée n_a, sa clé publique $P_A = n_A \times G$ et l'expédie à B

Génère aléatoirement sa clé privée n_b, sa clé publique $P_B = n_B \times G$ et l'expédie à A

Génère la clé secrète $K = n_A \times P_B$

Génère la clé secrète $K = n_B \times P_A$

Clé secrète partagée $K = n_A \times P_B = n_B \times P_A$

La cryptographie à courbes elliptiques

- Chiffrement de données par ECC
  - Rendre publique G(x,y) et $E_p(a, b)$: $y^2=x^3+ax+b$
  - Echanger les clefs ($n_A \times P_B$ et $n_B \times P_A$)
  - Encoder le texte clair m par des points Pm (x,y)
  - Exemple : ‘a’ = ‘0100100001’ = $P_a(9,1)$, ‘b’ = ‘0010110100’ = $P_b(5, 20)$, ….

A

En clair

$P=23, E_p(a, b) : y^2 = x^3 + x + 1, G(x,y)$

B

Echange des clefs $P_A = n_A \times G$, $P_B = n_B \times G$

$Cm = \{k.G, P_m + k.P_B\}$

$P_m + k.P_B - n_B.(k.G)$

$P_m + k(n_B.G) - n_B(k.G) =$ Pm

La cryptographie à courbes elliptiques

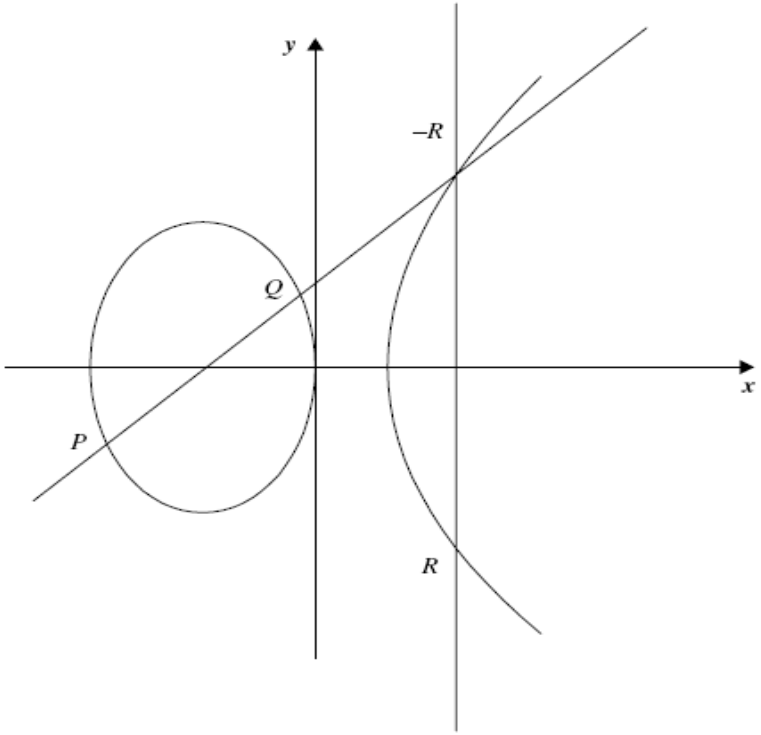
- Avantages
  - Calculs plus simples qu'avec les systèmes à base de modulo
  - Algorithmes sont moins gourmands en termes de temps de calcul et de matériel
  - Niveau de sécurité élevé avec une taille de clef assurant le même niveau de sécurité

Méthode	Taille des clefs					
DES	56	80	112	128	192	256
ECC	112	160	224	256	384	512
RSA	512	1024	2048	3072	7680	15360

  - ECC est une approche appelée à se répandre dans les applications pratiques
  - Utilisation pour systèmes embarqués

La cryptographie à courbes elliptiques

- Inconvénients
  - Complexité (*calculs*, *GF(2^n), +, **)
  - Grand nombre de brevets qui empêchent son développement



Standards PKCS

- Des standards pour la cryptographie à clé publique
- PKCS (Public Key Cipher System)
  - Définis initialement par la compagnie RSA pour activer l'usage de algorithme l'asymétrique RSA.
  - Principale objectif: interopérabilité entre les applications.
  - Pour la plupart ils sont standardisés à l'IETF. Certains sont obsolètes.
  - Ils couvrent la réalisation de tous les services de sécurité.
  - Utilité des PKCS
    - Ils définissent des formats pour tous les objets liés à la réalisation de ces services.
    - Ils définissent des interfaces entre certains composants (logiciel et matériel) de sécurité
- ASN1 et DER sont utilisés pour la représentation abstraite et interne de ces objets
- Une description de ces standards avec des exemples et donnée sur le site de RSA (http://www.rsa.com)

Standards PKCS

PKCS #1: RSA Cryptography Standard
PKCS #3: Diffie-Hellman Key Agreement Standard
PKCS #5: Password-Based Cryptography Standard
PKCS #7: Cryptographic Message Syntax Standard
PKCS #8: Private-Key Information Syntax Standard
PKCS #10: Certification Request Syntax Standard
PKCS #11: Cryptographic Token Interface Standard
PKCS #12: Personal Information Exchange Syntax Standard
PKCS #13: Elliptic Curve Cryptography Standard
PKCS #14: Pseudorandom Number Generation Standard
PKCS #15: Cryptographic Token Information Format Standard

Standards PKCS#1, RSA Cryptography Standard

- Spécifie les primitives RSA de
  - chiffrement, de déchiffrement de signature (selon un principe décrit dans PKCS #7) et de vérification
- Spécifie les schémas de chiffrement et de signature
- Spécifie les méthodes d'encodage de ces schémas
- Spécifie la syntaxe ASN.1 pour:
  - les clés publiques
  - les clés privées
  - les schémas mentionnés ci-dessus

- Description d'une méthode pour implémenter *l'algorithme Diffie-Helman*.
- **Une Autorité Centrale** (qui n'est pas précisée) **génère un nombre premier *p*, ainsi qu'un entier *g*, (telle que *g*<*p*).**
- Chaque partie génère une valeur privée x (resp. x') et une valeur publique y (resp. y'), qu'elle transmettra à l'autre partie.
- $y = g^x \bmod p$ et $y' = g^{x'} \bmod p$
- $Z = y^{x'} = y'^{x}$

Standards PKCS#5, Password-Based Cryptography Standard

- Description d'une méthode pour chiffrer une chaîne d'octets avec une clé secrète dérivée d'un mot de passe.
- Ce standard est destiné au chiffrement de clés privées (réfère PKCS #8).
- Définition de deux algorithmes de chiffrement de clé
  - MD2 avec DES-CBC
  - MD5 avec DES-CBC

Standards PKCS#7, Cryptographic Message Syntax

- Description d'une syntaxe pour une enveloppe et une signature numérique
- Cette syntaxe est récursive, une enveloppe peut être enveloppée à son tour, ou signée par une autre entité.
- La syntaxe inclut des attributs optionnels tels la date, les certificats, les CRLs.
- Ce standard permet une conversion vers du PEM (RFC1422).

- Description de syntaxe pour la clé privée :
  - la clé privée
  - des attributs
  - Identifiant de l'algorithme (ex: PKCS#1)

```
PrivateKeyInfo ::= SEQUENCE {
        version                 Version,
        privateKeyAlgorithm     PrivateKeyAlgorithmIdentifier,
        privateKey              PrivateKey,
        attributes [0] IMPLICIT         Attributes OPTIONAL }

PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
```

Standards PKCS#10, Certification Request Syntax Standard

- Description de la syntaxe pour les requêtes de certification (*certification requests*) vers l'autorité de certification.
- IETF a défini une autre structure (RFC 2511)(Internet X.509 Certificate Request Message Format)

Standards PKCS#11, Cryptographic Token Interface Standard

- Définit une API (application programming interface)
- Connu sous le nom de CRYPTOKI
- Permet de définir l'accès à des « tokens » cryptographiques comme les cartes à puce ou les « token USB ».
- Fournit les services suivants de:
  - Stockage des clés publiques/privée, des certificats, des valeurs d'authentification (PIN), et d'autres type de données.
  - chiffrement/déchiffrement
  - Signatures et de vérification
  - génération des clés
  - génération des nombres aléatoires
- La plupart des browsers supporte cette API

Standards PKCS#12, Personal Information Exchange Standard

- Ce standard est une généralisation et extension de PKCS#8
- Décription d'une syntaxe de transfert des informations d'identité personnelle
  - clé privée
  - certificats...
- Les informations peuvent être protégées de façon à être:
  - confidentielles
  - Intègres.
- C'est le format exigé pour intégrer la clé privé dans les différentes applications « clientes »

Standards PKCS#15, Cryptographic Token Information Format Standard

- Standardisation du format des fichiers et répertoires pour le stockage d'éléments cryptographiques.

- Ne standardise pas le calcul RSA

Standards PEM
Privacy Enhancement for internet electronic Mail

- But de PEM : fournir les services de confidentialité, authentification, intégrité, non-répudiation pour le courrier électronique
- Traitements PEM de bout en bout, et en aucun cas au niveau du Système de Transfert de Message.
- Les messages PEM adoptent les techniques d'encapsulation décrite dans la RFC 934

  ********** Begin xxxx **********

  ……

  ********** End xxxx **********
- Le contenu d'un messages PEM est codé en base64
- RFC 1421 - PEM : Message Encryption and Authentication Procedures
  - Trois types de messages:
    - ENCRYPTED : confidentialité, l'authentification, l'intégrité, et (dans le cas d'un Asymetric Key Management) la non répudiation de l'origine.
    - MIC-ONLY pour l'authentification, l'intégrité et (dans le cas d'un Asymetric Key Management) la non répudiation de l'origine.
    - MIC-CLEAR pour les mêmes services de sécurité que MIC-ONLY, mais pour un correspondant ne disposant pas de soft PEM.
- RFC 1422 - PEM : Key Management
- RFC 1423 - PEM : Algorithms, Modes, and Identifiers

Les certificats

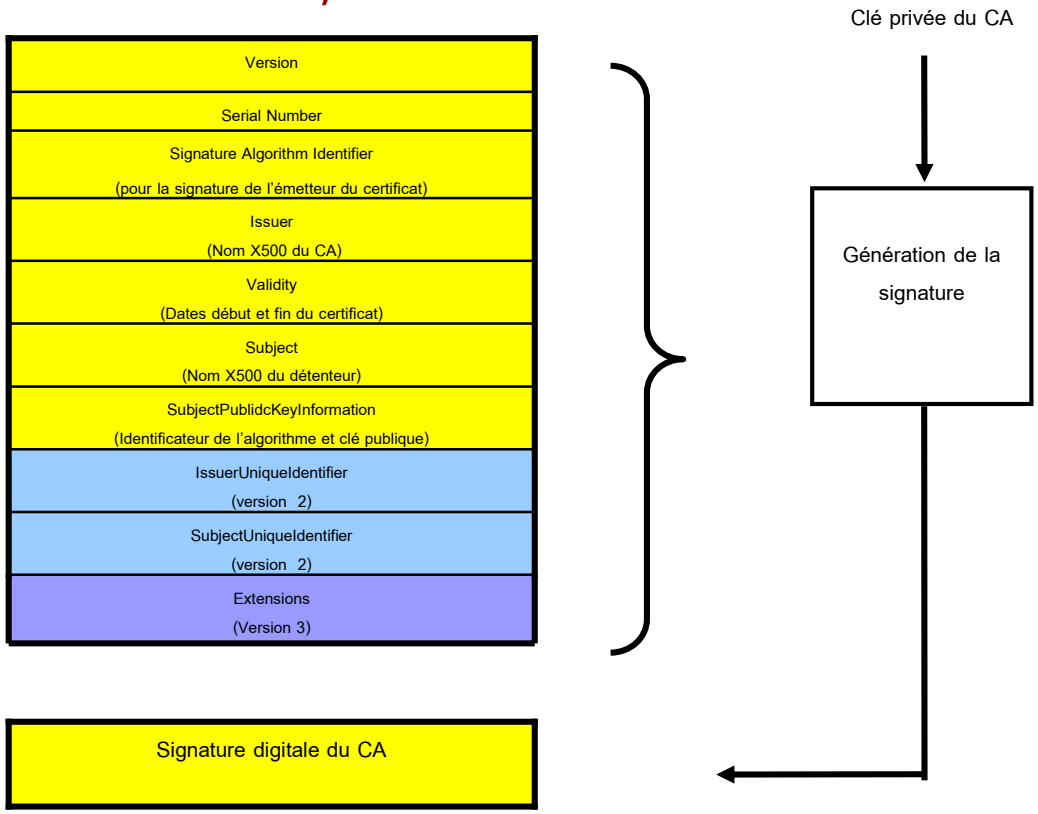
- « Certificat » = relève d'une autorité ou institution
- Le contenu = information « authentique »
- Mise en place d'un état de confiance en présence d'un certificat
- Dico: « Acte écrit qui rend témoignage de la vérité d 'un fait, d'un droit »
- Présence d'une autorité « reconnu » qui atteste de la véracité du contenu.
- Certificat = document « signé »

Certificats X.509

- Standard:
  - ITU-T X.509(03/2000), ou ISO/IEC 9594-8
    - Certificats de clé publique et d'attribut
  - RFC 3280: (définition de profil fonctionnel basé sur X509)
- Versions successives:
  - 1988 : v1
  - 1993 : v2 = v1 + 2 nouveaux champs
  - 1996 : v3 = v2 + extensions

Certificats X.509

- Structure de données permettant de lier différents éléments au moyen d'une signature
  - Le sujet ,la clef, l'émetteur du certificat, conditions de validité,…



Certificats X.509

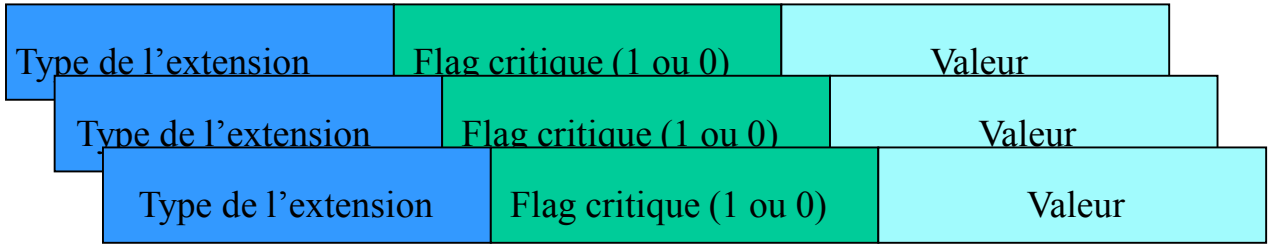
- Certificat personnel
  - permet d'authentifier un utilisateur.
- Certificat serveur
  - permet d'authentifier un serveur
- Certificat développeur
  - permet de signer et d'authentifier les programmes et macros développés.
- Certificat d'autorité de certification
  - permet de signer des certificats.

Certificats X.509

Certificate:
Data:
Version: 1 (0x0)
Serial Number: 1f:42:28:…:b3:ab:1f:1c
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, **O=**VeriSign, Inc., **OU=**Class 2 Public Primary Certification Authority - G2, **OU=** (c) 1998 VeriSign, Inc. - For authorized use only, **OU=**VeriSign Trust Network
Validity
Not Before: May 18 00:00:00 1998 GMT
Not After : May 18 23:59:59 2018 GMT
Subject: C=US, **O=**VeriSign, Inc., **OU=**Class 2 Public Primary Certification Authority - G2, **OU=** (c) 1998 VeriSign, Inc. - For authorized use only, **OU=**VeriSign Trust Network
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:a7:…:7f:77
Exponent: 65537 (0x10001)
Signature Algorithm: sha1WithRSAEncryption
11:45:….:ce:ef:

Certificats X.509/Extensions

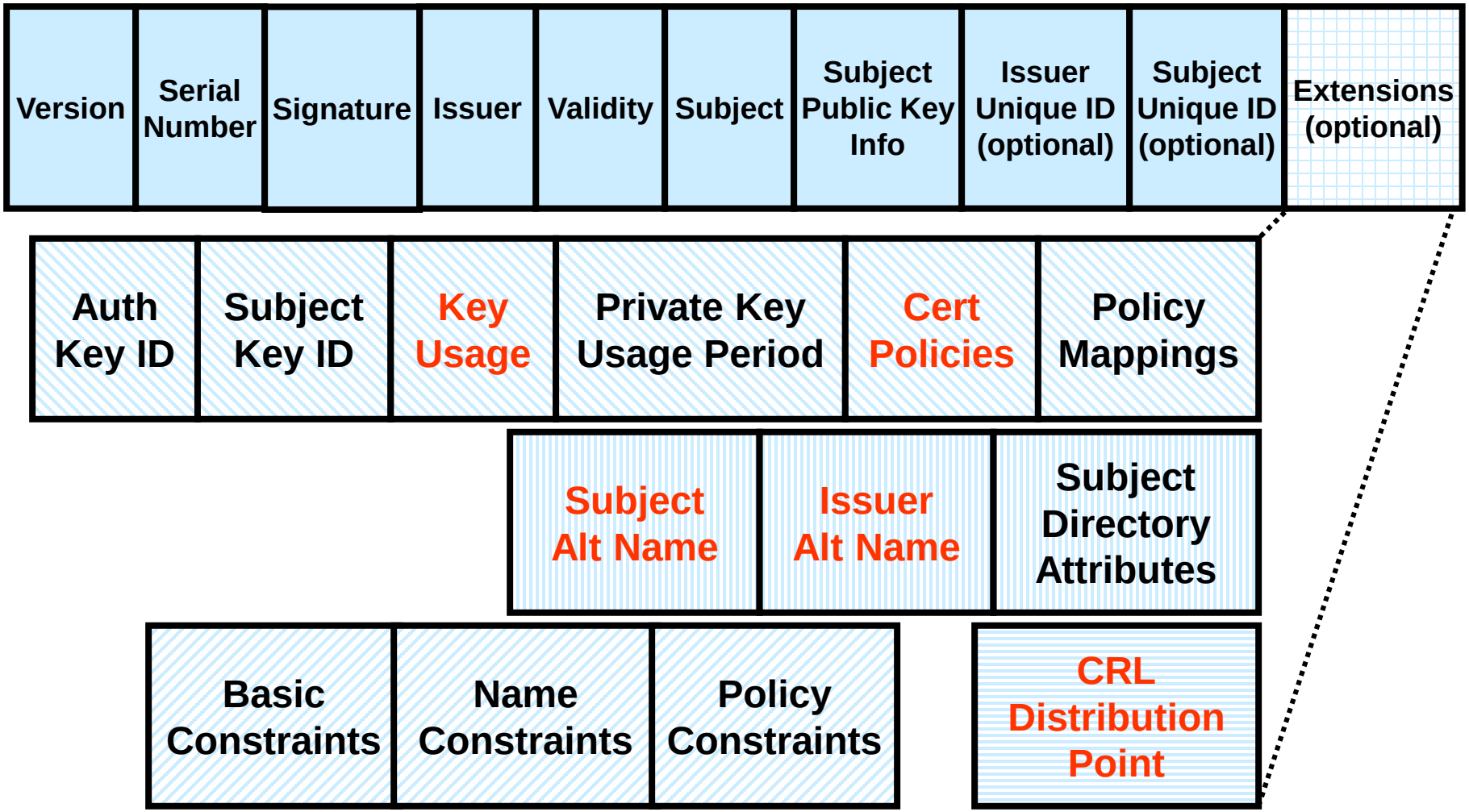
- L'extension répond au besoin de disposer d'autres informations sur la clé publique et son porteur que l'identité
- Concept d'origine: lien entre identité et clé publique
- L'extension attribut un rôle au certificat
- L'extension est défini dans ITU-T Rec. X.660 et ISO/IEC 9834-1
- Plusieurs extensions sont standardisées, possibilité de définir des extensions spécifiques (OID)
- Si l'application ne supporte pas une extension critique, elle abandonne le certificat



Certificats X.509/Extensions

- Extensions sur:
  - le nomage de l'objet et du signataire
  - les clés publiques/privés
  - la révocation
  - la politique de certification
  - le rôle
  - Autres … logo (RFC 3709: Internet X.509 Public Key Infrastructure:Logotypes in X.509 Certificates)

Certificats X.509/Extensions



Certificats X.509/Extensions

- **SubjectAlternativeName**
  - un nom X.500, une adresse X400, un nom rfc822 (adresse mail), un Directoryname (DNS), un nom EDI, une URL, une adresse IP, un OID… ou toute forme de nom,
- **IssuerAlternativeName**
  - Toute forme de nom;
- **SubjectDirectoryattributes**
  - transporte une séquence d'attributs de l'annuaire X.500 (numéro de tél., adresse, …)
- **AuthorityKeyIdentifier**
  - Permet de distinguer plusieurs clés utilisés par le même CA. Identifie la clé publique pour la vérification de la signature apposée au certificat.
- **SubjectKeyIdentifier**
  - Identificateur de clé unique par rapport à toutes les clés en possession du sujet.

Certificats X.509/Extensions

- **KeyUsage** : usage de la clé publique certifiée

  DigitalSignature

  NonRepudiation

  KeyEncipherment

  DataEncipherment

  keyAgreement

  keyCertSign

  CRLSign

  encipherOnly

  decipherOnly

Certificats X.509/Extensions

- **ExtendedKeyUsage :**
  - Indique un ou plusieurs buts pour l'usage de la clé publique: **serverAuth, clientAuth, codeSigning, emailProtection, timeStamping, OCSPSigning**

  ExtentededKeyUsage ::= SEQUENCE OF KeyPurposeId

  KeyPurposeId ::= OBJECT IDENTIFIER

- **PrivateKeyUsagePeriod:**
  - Indique la durée de vie de la clé privée (uniquement pour des clés de signature).

Certificats X.509/Extensions

- **CertificatePolicies** : liste des politiques de sécurité reconnues par le CA émetteur.
- **PolicyMappings** : relation entre politique de sécurité dans des domaines différents
- **IssuerNameconstraints:** utilisé dans les certificats de CAs, indique un espace de noms où tous les noms des sujets ultérieurs dans le chemin de certification doivent figurer
- **PolicyConstraints :** identification explicite d'une politique de sécurité, ou interdiction du mapping des politiques dans un chemin de certification

Certificats X.509/Extensions

- **BasicConstraints:** indique si le détenteur d'un certificat peut agir comme un CA, si oui, donne aussi la longueur de chemin de certification

Exemple:
X509v3 extensions:
X509v3 Basic Constraints: critical **CA:**TRUE

Certificats X.509/Révocation

- Un certificat est révoqué parce que:
  - La clé privée de l'autorité est compromise
  - La clé privée associée au certificat est compromise
  - Changement de statut du détenteur du certificat
  - Suspension du détenteur du certificat
  - …
- Plusieurs approches existent pour la révocation:
  - Différents modèles de publication de la liste de CRL
    - CRLs, Delta CRLs, CRL Distribution Points (CDPs)
  - Vérification en temps réel
    - OCSP (Online Certificat S Protocol)

Certificats X.509/Révocation

- Point de la liste CRL
  - Modèle traditionnel, supporté par toutes les plateformes. Liste noires des certificats révoqués.
  - Signée par la clé privée de l'autorité de certification et publiée dans l 'annuaire
  - Chaque entrée contient :
    - le numéro de série du certificat
    - la date de la révocation
    - d 'autres info comme la cause de la révocation
  - Adresse du serveur (ou des serveurs) CRL et nom du fichier contenant la liste des CRLs

Certificats X.509/Révocation

- La delta CRL :
  - Fournit les informations sur les certificats dont le statut a changé depuis la dernière CRL.
  - Réduit la quantité de données à échanger avec l'autorité de certification et améliore les temps de réponse et la sécurité de la vérification de la validité des certificats.
  - Les usagers maintiennent leurs propres bases de données de CRLs
  - Chaque delta CRL est associée à une CRL de référence
- CRL Distribution Point
  - Principe : diviser la CRL en des parties plus petites
  - Chaque certificat contient les informations permettant à l 'application de vérifier sa validité au bon endroit

Certificats X.509/Révocation

- OCSP: Online Certificate Status Protocol
- Permet la vérification temps réel de la validité du certificat
- repose sur un modèle client-serveur
- l ’application héberge un client qui interroge le serveur OCSP sur l ’état du certificat
- le serveur envoi l ’état du certificat dans un message signé

Certificats X.509

Certificate:
Data:
Version: 3 (0x2)
Serial Number:
27:30:02:b2:84:67:76:16:f3:a0:bc:1e:f6:27:ed:1e
Signature Algorithm: sha1WithRSAEncryption
Issuer:
C=US, O=RSA Data Security, Inc., OU=Secure Server Certification Authority
Validity
Not Before: Mar 18 00:00:00 2005 GMT
Not After : Mar 18 23:59:59 2006 GMT
Subject: C=FR, ST=Hauts de Seine, L=LA-DEFENSE, O=Credit Agricole SA, OU=SIB, CN=interactif.creditlyonnais.fr
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit): 00:cb:...:79:2b
Exponent: 65537 (0x10001)

Certificats X.509

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

X509v3 Key Usage:

Digital Signature, Key Encipherment

X509v3 CRL Distribution Points:

URI:http://crl.verisign.com/RSASecureServer.crl

X509v3 Certificate Policies:

Policy: 2.16.840.1.113733.1.7.23.3

CPS: https://www.verisign.com/rpa

X509v3 Extended Key Usage:

TLS Web Server Authentication, TLS Web Client Authentication

Authority Information Access:

OCSP - URI:http://ocsp.verisign.com

1.3.6.1.5.5.7.1.12:

0_.].[0Y0W0U..image/gif0!0.0..+.k...j.H.,{..0%.#http://logo.verisign.com/vslogo.gif

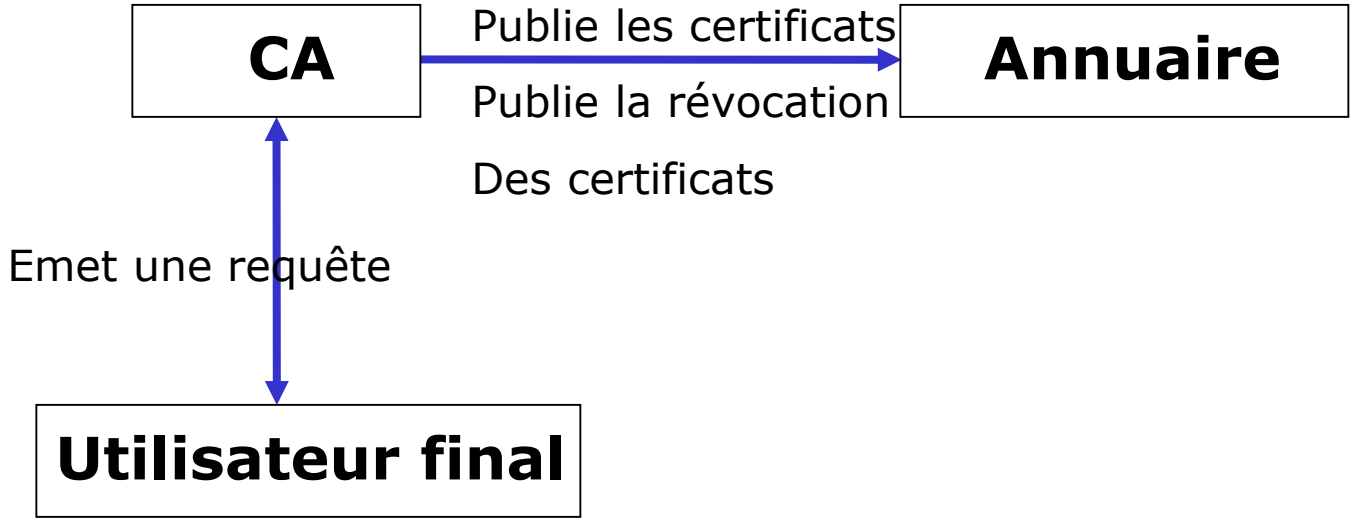
Signature Algorithm: sha1WithRSAEncryption 3c:45:...:28:4c

Certificats X.509/CRL

Certificate Revocation List (CRL):
Version 1 (0x0)
Signature Algorithm: md5WithRSAEncryption
Issuer: /C=US/O=RSA Data Security, Inc./OU=Secure Server
Certification Authority
Last Update: Mar 22 11:00:22 2005 GMT
Next Update: Apr 5 11:00:22 2005 GMT
Revoked Certificates:
Serial Number: 0103367C71DC0EDCDE861211763145D6
Revocation Date: Sep 30 16:09:14 2004 GMT
Serial Number: 010460ED39FE935092EE10167D681A38
Revocation Date: Jan 31 20:32:19 2005 GMT
…….
Serial Number: 7FFFCF8E6A350C6F3F6313C96F010E69
Revocation Date: Mar 31 11:29:23 2004 GMT
Signature Algorithm: md5WithRSAEncryption
33:65:...:17:cc

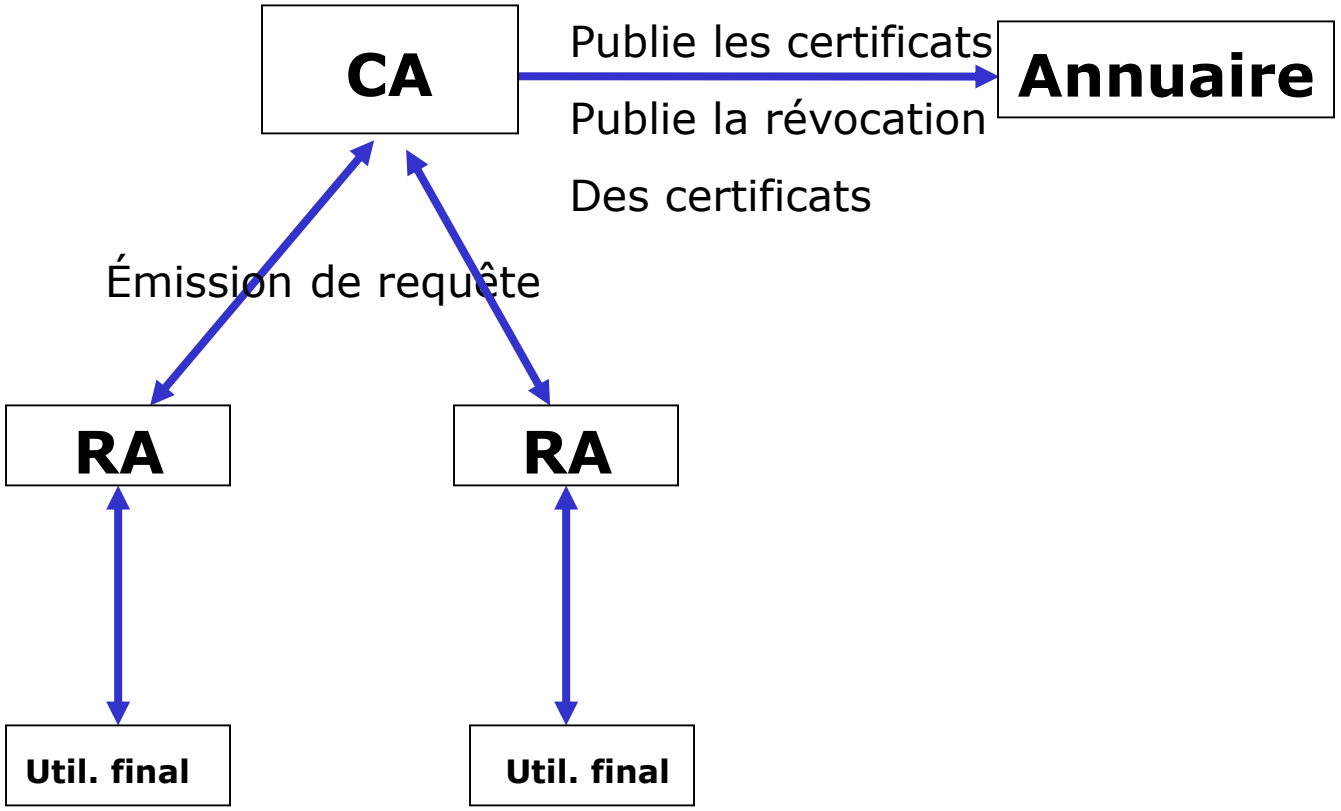
Gestion des certificats

- Comment être sûr qu'une clé publique est bien associé à un sujet?
- L'autorité de certification répond à cette question.
- Le CA vérifie l'authenticité de la requête, signe et publie le certificat.



Gestion des certificats

- L'autorité d'enregistrement (RA) vérifie les demandes de certificat de l'usager



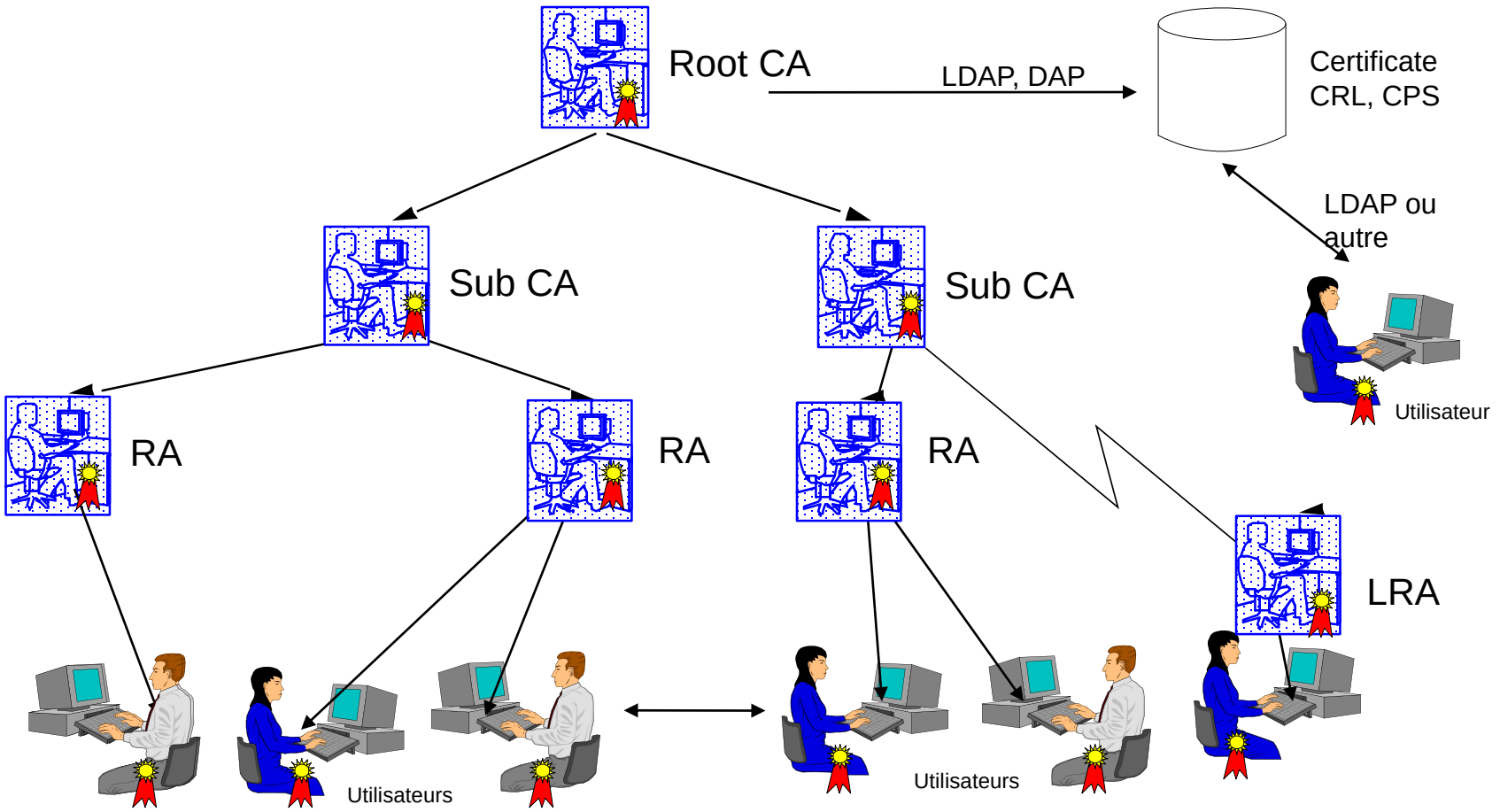
Gestion des certificats

Les composants d'une infrastructure de gestions des certificats

- Terminologie:
  - PKI: Public Key Infrastructure
  - IGC: Infrastructure de Gestion des clés
- Opérations de base de la PKI:
  - Fournit et gère les éléments de sécurité qui permettent la mise en œuvre de mécanisme d'authentification de chiffrement et de signature
  - Instaurer une tierce partie de confiance entre les acteurs.
  - **vérification, certification, révocation, publication**
- Composants de base de la PKI:
  - CA: autorité qui signe les certificats
  - RA: autorité qui vérifie les requêtes des usagers et les soumet au CA
  - Annuaire: contient les certificats et les certificats révoqués.
  - Les usagers.

Gestion des certificats

Les composants d'une infrastructure de gestions des certificats



Fonctions principales d'une infrastructure PKI

- Les fonctions de base :
  - Enregistrement des utilisateurs
  - Valider les modèles de confiance
  - Définir et gérer le Certification Practice Statement (CPS)
  - Gérer les clés et les certificats
  - Révocation et suspension des certificats
  - publication des certificats
  - Création et publication des CRL
  - Archivage et récupération des certificats
  - Maintenance et Responsabilité
- Les fonctions avancées
  - Services d'horodatage
  - Service de récupération des clés privées

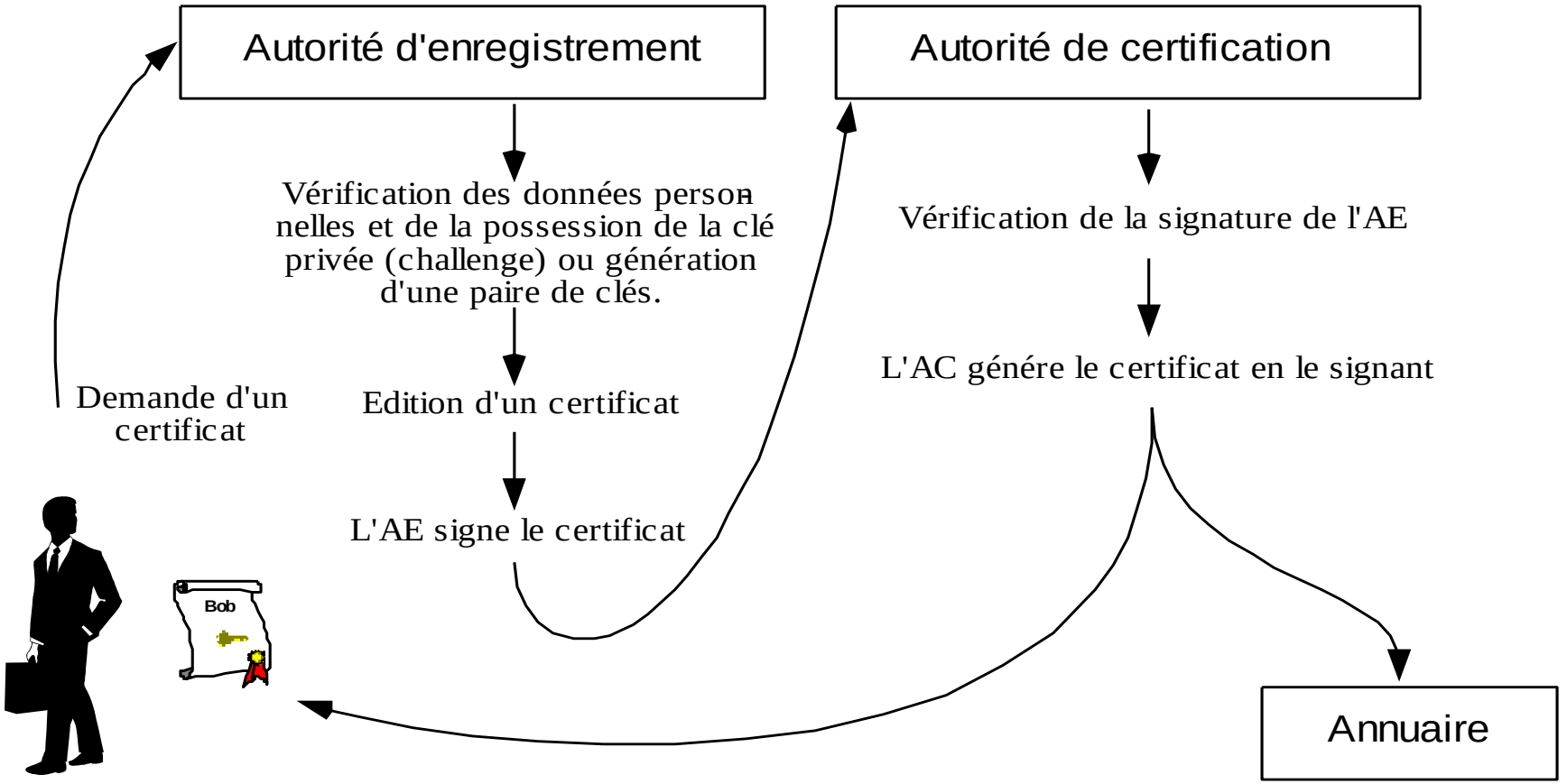
Modèle de confiance

- Modèle hiérarchique
  - Hiérarchie de CA : structure d'arbre
- Web of Trust
  - Liens de confiance sous forme transitive
- Cross-certification
  - Hiérarchie de CA avec des liens de confiance entre les racines
- Orienté utilisateur
  - La décision de confiance dépend de l'utilisateur

PKI/RA

- Autorité d ’enregistrement (RA)
- Vérification de l ’identité de l ’utilisateur (fonction de la politique de certification)
- Exécution de la politique de sécurité
- Traite uniquement les enregistrements, pas de révocation, etc.
- Fonctionne sous le CA
- L’enregistrement peut être local, ou externalisé

PKI/RA


Autorité d'enregistrement
Autorité de certification
Vérification des données personnelles et de la possession de la clé privée (challenge) ou génération d'une paire de clés.
Vérification de la signature de l'AE
L'AC génére le certificat en le signant
Demande d'un certificat
Edition d'un certificat
L'AE signe le certificat
Bob
Annuaire

Politique de certification (CP)

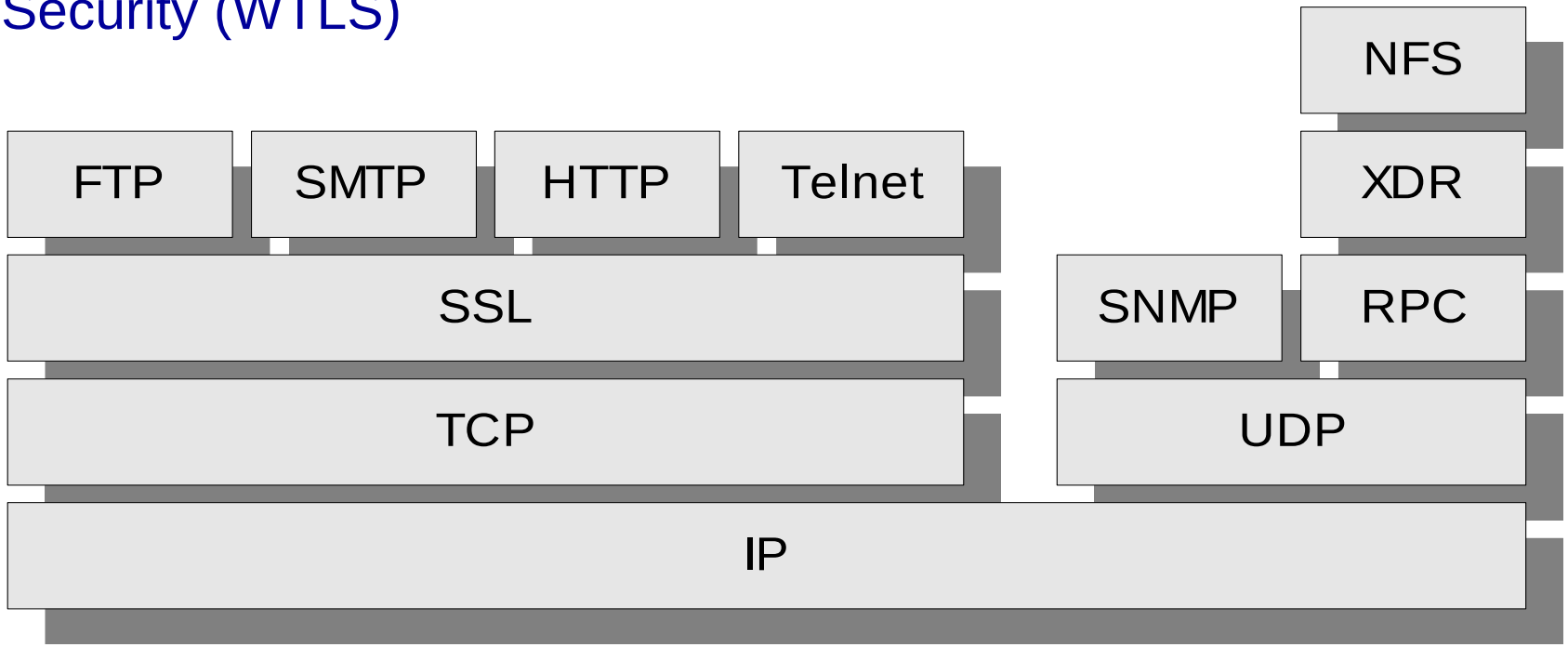
- Dérivée des politiques de sécurité en place
- Elle détermine :
  - le niveau d'assurance
  - le mode d 'Identification et d 'authentification
  - Durée de validité des certificats
  - période d 'émission de la CRL / révocation des certificats
  - publication des certificats
  - re-génération des certificats
  - les limites de responsabilité
  - le niveau des contrôles de sécurité
  - le niveau des audits
- Toutes ces info sont décrites dans le Certification Practice Statement (CPS)

Certification Practice Statement (CPS)

- Énoncé détaillé sur les procédures opérationnelles, les standards et les pratiques de l ’infrastructure pour remplir les fonctions identifiées dans la politique de certification, notamment :
  - l’émission d ’un certificat et l ’enregistrement des utilisateurs
  - les durées de vie et la révocation
  - le modèle de confiance et le processus de vérification
  - les modes de publication des certificats
- Conçu dans l’objectif de:
  - Définir les procédures pour le personnel
  - Limiter la responsabilité
- Nécessite le plus souvent la participation des conseillers juridiques.

Protocole SSL/TLS

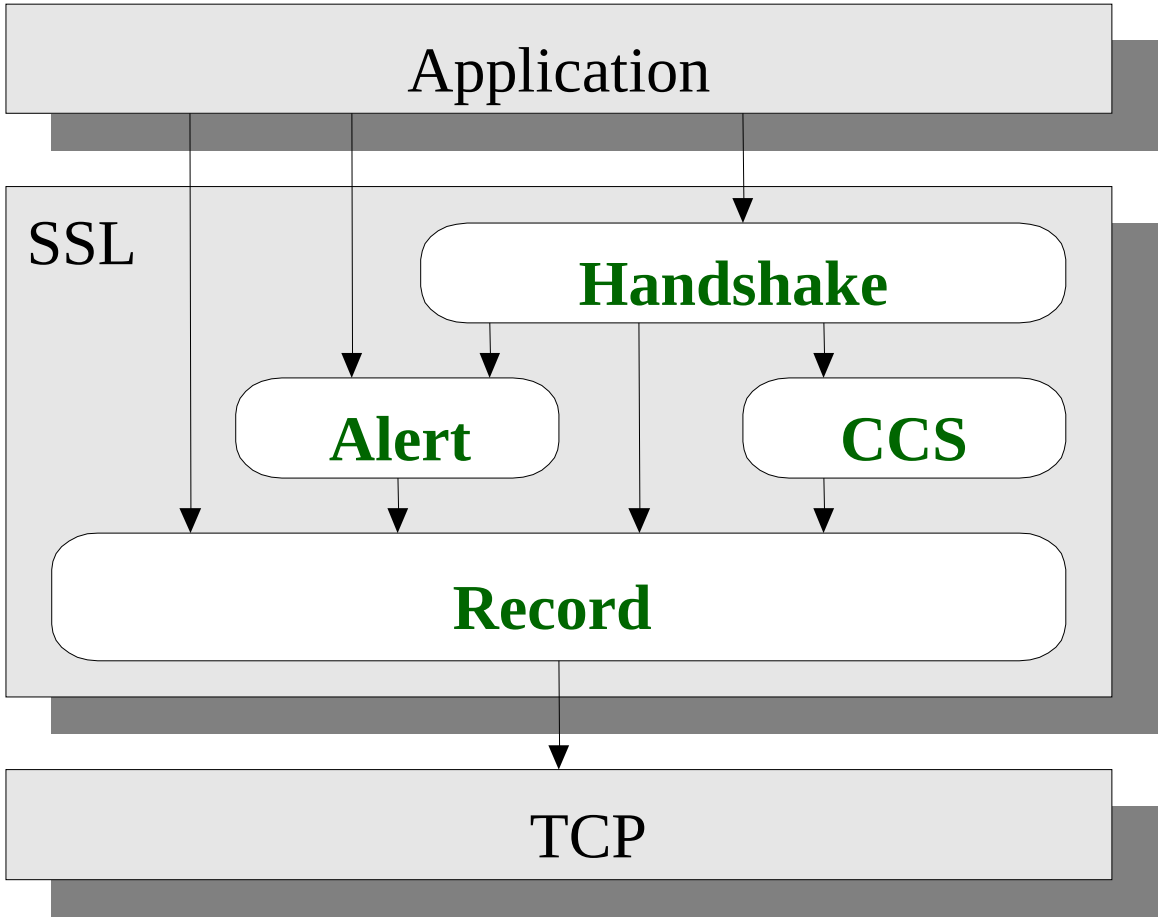
- SSL défini par *netsacpe* et intégré au browser
- Première version de SSL testé en interne Première version de SSL diffusé : V2 (1994)
- Version actuelle V3
- Standard à l ’IETF au sein du groupe Transport Layer Security (TLS)
- Standard au sein du WAP Forum Wireless Transport Layer Security (WTLS)



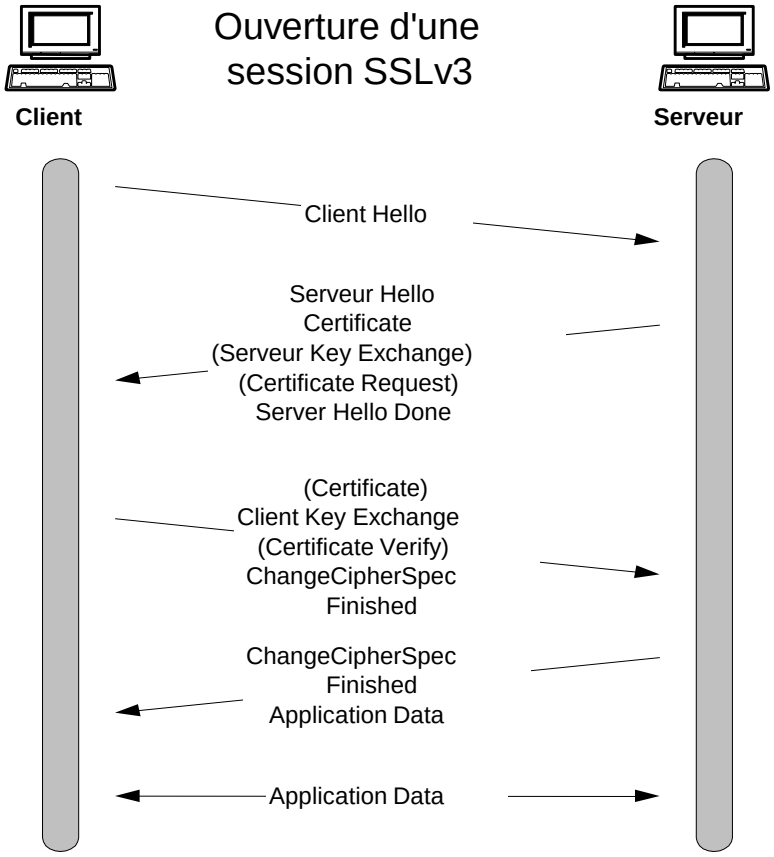
Protocole SSL/TLS

- Authentification
  - Serveur (obligatoire), client (optionnel)
  - Utilisation de certificat X509 V3
  - A l'établissement de la session.
- Confidentialité
  - Algorithme de chiffrement symétrique négocié, clé générée à l'établissement de la session.
- Intégrité
  - Fonction de hachage avec clé secrète : hmac(clé secrète, h, Message)
- Non Rejeu
  - Numéro de séquence

Protocole SSL/TLS



Protocole SSL/TLS

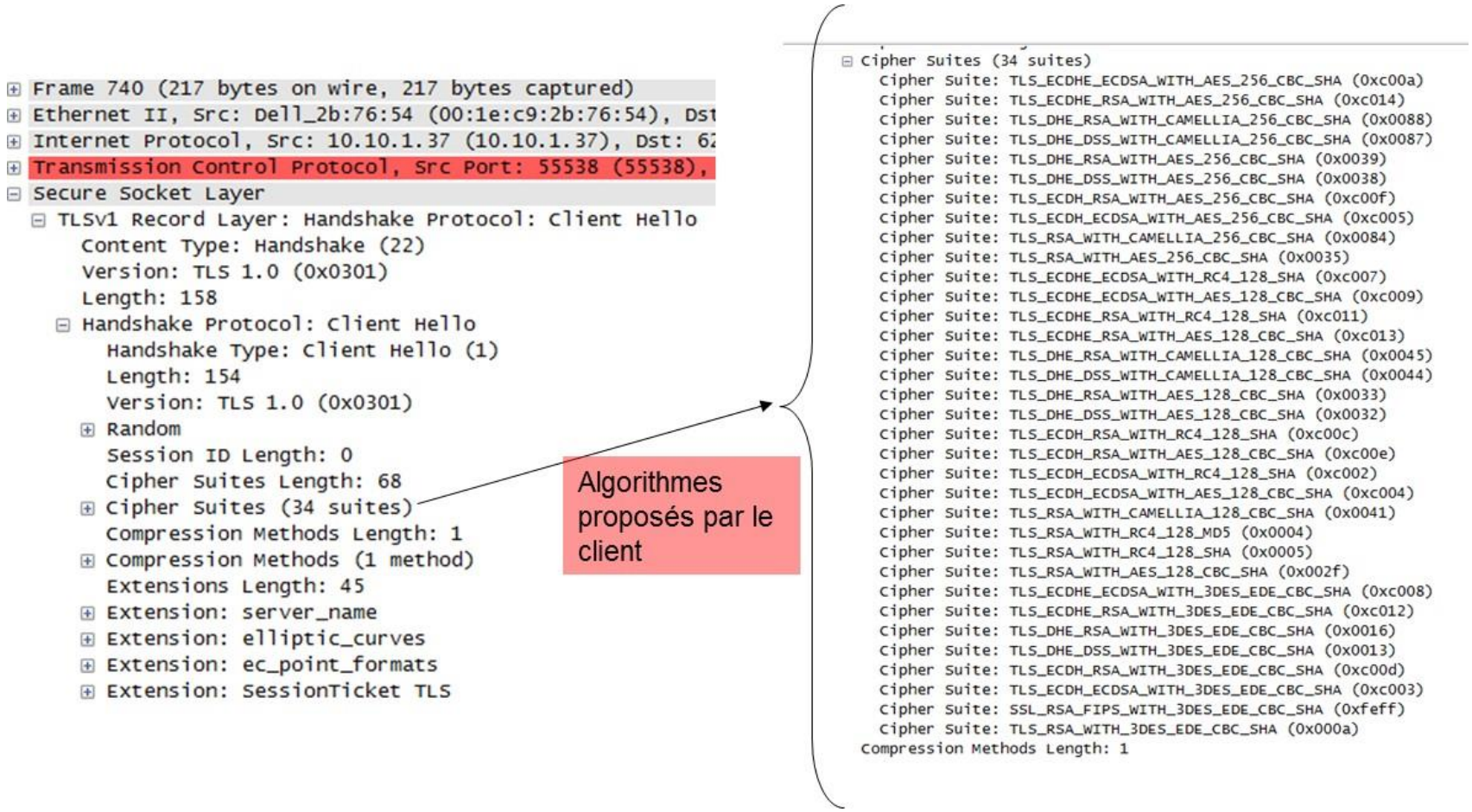


Protocole SSL/TLS - Handshake

- Authentification du serveur et éventuellement du client,
- Négociation des algorithmes de chiffrement et de hachage, échange d'un secret,
- Génération des clés.

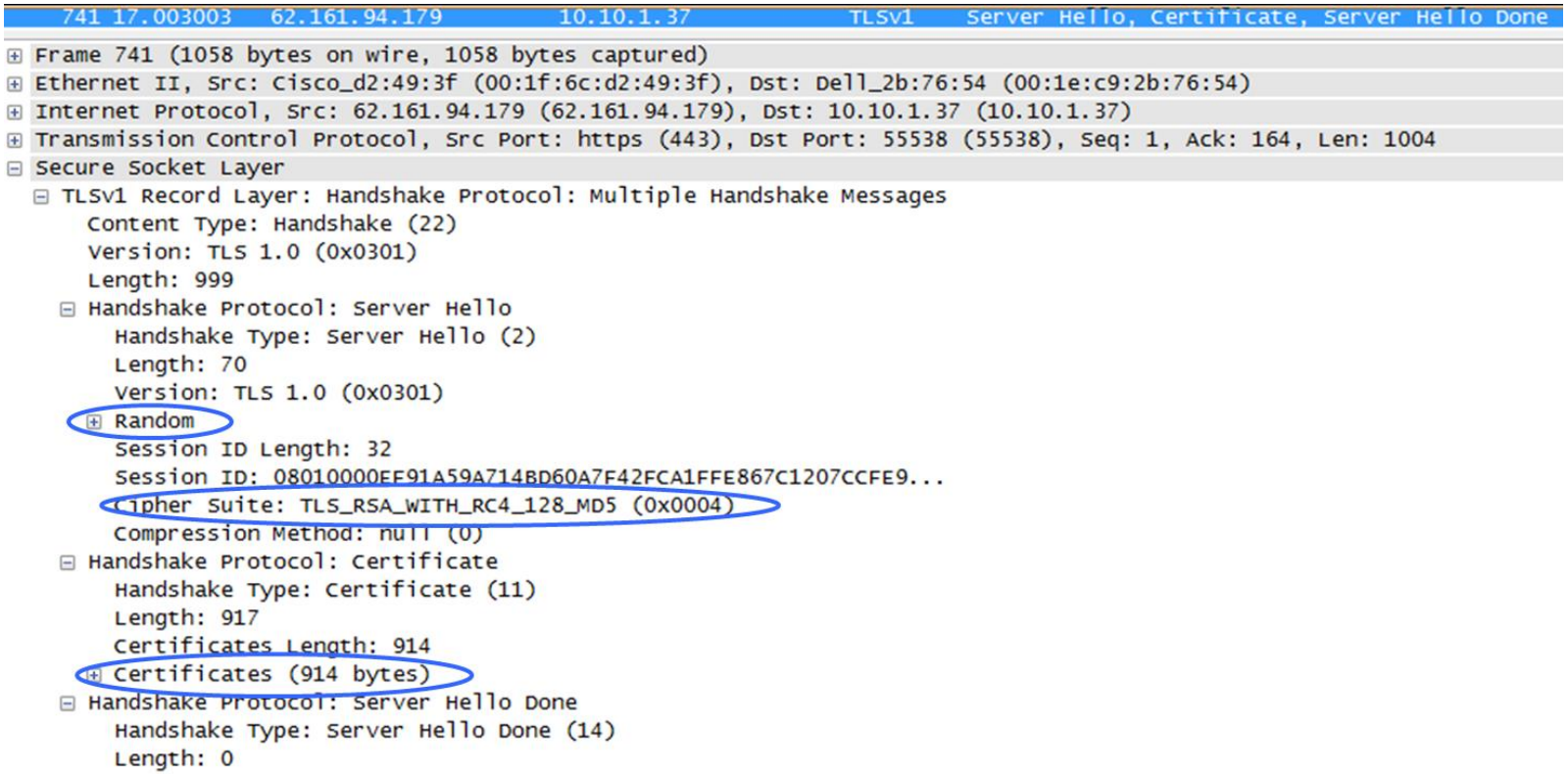
Protocole SSL/TLS - Handshake

Exemple : requête *ClientHello*



Protocole SSL/TLS - Handshake

Exemple : réponse *ServerHello*



Protocole SSL/TLS – Génération des clés

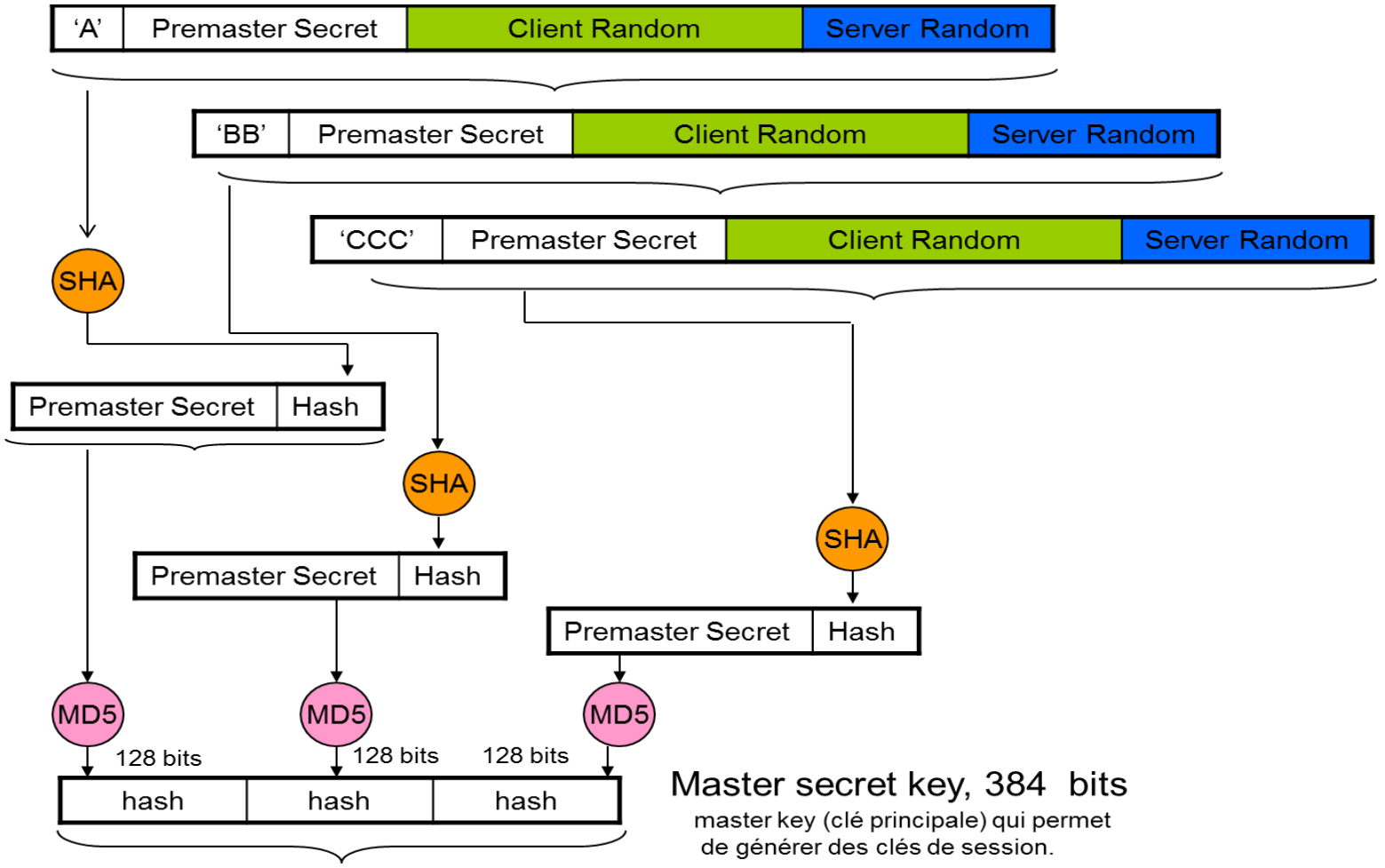
- Construction du *Master secret key* à l'ouverture d'une session
  - Calculé par le client et le serveur
  - master_secret =

    MD5(pre_master_secret || SHA('A' || pre_master_secret || ClientHello.random || ServerHello.random))||
    MD5(pre_master_secret || SHA('BB' || pre_master_secret || ClientHello.random ||ServerHello.random)) ||
    MD5(pre_master_secret || SHA('CCC' || pre_master_secret || ClientHello.random || ServerHello.random))

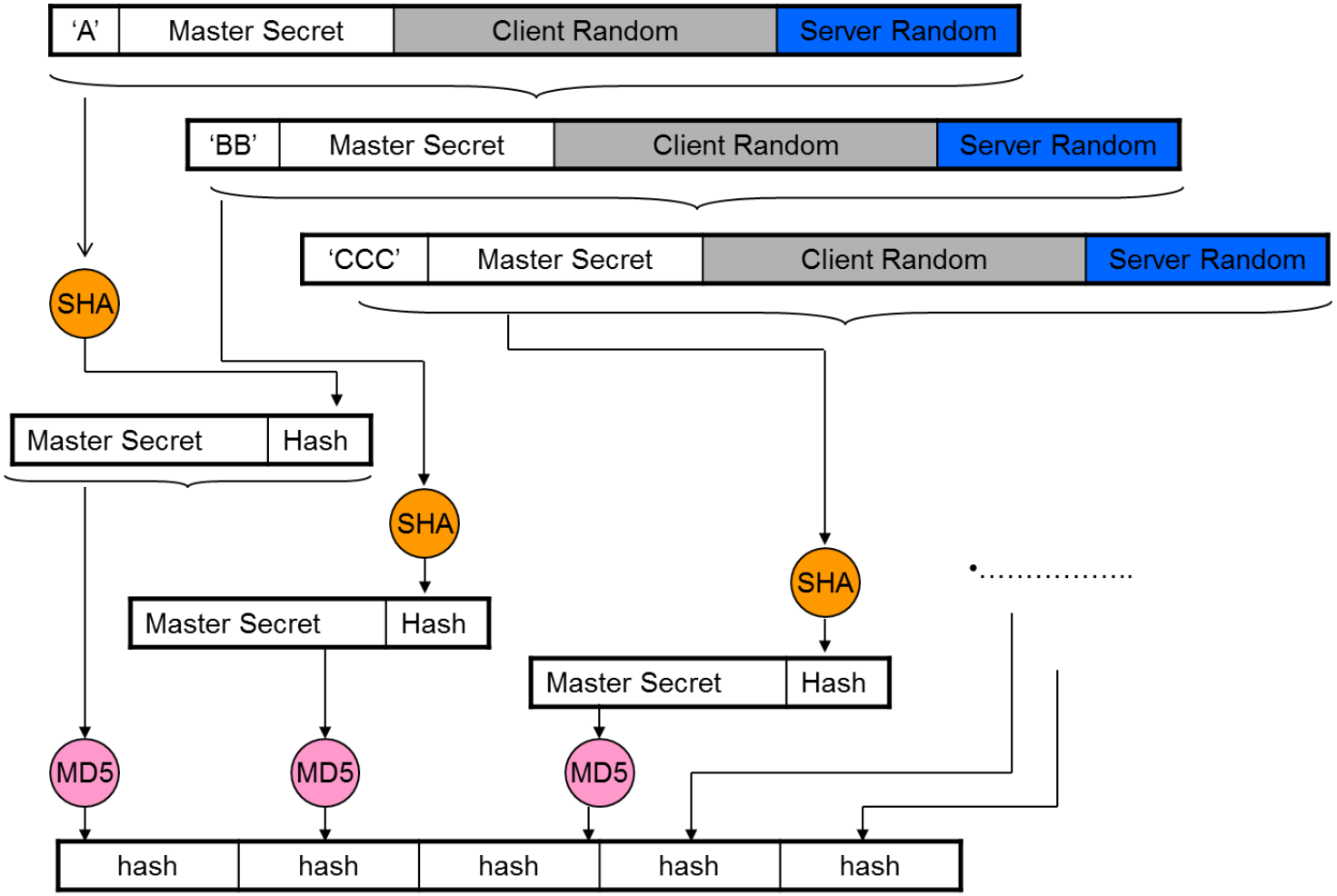
- Génération de secrets à l'ouverture d'une session ou connexion
  - key_block =

    MD5(master_secret || SHA('A' || master_secret || ServerHello.random || ClientHello.random))||
    MD5(master_secret || SHA('BB' || master_secret || ServerHello.random || ClientHello.random))||
    MD5(master_secret || SHA('CCC' || master_secret || ServerHello.random || ClientHello.random))||
  - Key_block= 2 clés MAC + 2 clés chiffrement

Protocole SSL/TLS – Génération des clés



Protocole SSL/TLS – Génération des clés



Protocole SSL/TLS – *ChangeCipherSpec*

- *ChangeCipherSpec* signale au *Record* toute modification des paramètres de sécurité,

- Constitué d'un message (1 octet)

Protocole SSL/TLS – Record

- Reçoit les données des couches supérieures : (*Handshake*, *Alert*, *CCS*, HTTP, FTP ...), et les transmet au protocole TCP.

- Après application de :
  - la fragmentation des données en blocs de taille maximum de 2^{14} octets
  - la compression des données, fonction prévue mais non supportée actuellement
  - la génération d'un condensât pour assurer le service d'intégrité
  - le chiffrement des données pour assurer le service de confidentialité

Protocole SSL/TLS – AlertProtocol

- Le protocole *Alert* peut être invoqué :
  - par l'application, par exemple pour signaler la fin d'une connexion
  - par le protocole *Handshake* suite à un problème survenu au cours de son déroulement
  - par la couche *Record* directement, par exemple si l'intégrité d'un message est mise en doute

Protocole SSL/TLS – AlertProtocol

Message	Contexte	Type
bad_certificate	échec de vérification d'un certificat	fatal
bad_record_mac	réception d'un MAC erroné	fatal
certificate_expired	certificat périmé	fatal
certificate_revoked	certificat mis en opposition (révoqué)	fatal
certificate_unknown	certificat invalide pour d'autres motifs que ceux précisés précédemment	fatal
close_notify	interruption volontaire de session	fatal
decompression_failure	les données appliquées à la fonction de décompression sont invalides (par exemple, trop longues)	fatal
handshake_ failure	impossibilité de négocier des paramètres satisfaisants	fatal
illegal_parameter	un paramètre échangé au cours du protocole Handshake dépasse les bornes admises ou ne concorde pas avec les autres paramètres	fatal
no_certificate	réponse négative à une requête de certificat	avertissement ou fatal
unexpected_message	arrivée inopportune d'un message	fatal
unsupported_certificate	le certificat reçu n'est pas reconnu par le destinataire	avertissement ou fatal

Annexe : standardisation et acteurs (de la cryptologie)

- Ces acteurs sont impliqués dans le domaine de la sécurité
  - Ne sont pas impliqués dans la conception directe des algorithmes de cryptographie à l'exception de RSA, du NIST, de l'ETSI et du 3GPP
  - ils influent directement sur leur usage par l'intégration ou pas dans les solutions de sécurité
- Organismes internationaux
  - ISO (International Organization for Standardization)
  - IEC (International Electrotechnical Commission)
  - ITU (International Telecommunication Union)
- Organismes Nationaux
  - AFNOR (Association Française de NORmalisation)
  - ANSI (American National Standards Institute)
  - BSI (British Standards Institute)
- Associations savantes et professionnels
  - IEEE (Institute of Electrical and Electronics Engineers)
  - IETF (Internet Engineering Task Force)
- Industriels
  - RSA Coorparetion
  - 3GPP (Third Generation Partnership Project Structure gouvernementales)
  - ETSI (European Telecommunications Standard Institute)
- Structures gouvernementale
  - NIST (National Institute of Standards and Technology) (anciennement NBS)
  - SSI (anciennement DCSSI)

Organismes internationaux

- ISO (International Organization for Standardization)
  http://www.iso.ch/
- IEC (International Electrotechnical Commission)
  http://www.iec.ch/
  - Organisation en: TC (Technical Committees), SC (Subcommittees) et WG (Working Groupes)
  - Travaux sur la cryptographie au:
    - TC68: concerne le monde de la finance
    - JTC1 (comité joint de ISO et IEC): JTC37 la biométrie et le JTC17 l'Identification
- ITU (International Telecommunication Union)
  http://www.itu.int/ITU-T/
  - Certificats X509 et l'annuaire: principale acteur

Organismes Nationaux

- AFNOR (Association Française de NORmalisation)
  - Coordination avec l'ISO TC68
  - http://www.afnor.fr
- ANSI (American National Standards Institute)
  - Groupe de travail ANSI.X9: adopte et profile des standards de sécuruité
  - Coordination avec l'ISO TC68
  - http://www.ansi.org/
  - http://www.x9.org/
- BSI (British Standards Institute)
  - Coordination avec l'ISO TC68
  - BS 7799 pour l'audit de sécurité a donné l'ISO 17799
  - Coordination avec l'ISO TC68
  - http://www.bsi-global.com/

Annexe : standardisation et acteurs (de la cryptologie)

Associations savantes et professionnels

- IEEE (Institute of Electrical and Electronics Engineers)
  - association technique et professionnel sans profit 365,000 membres 150 pays
  - Connu pour l'organisation de conférences scientifiques et ses publications
  - Conception des protocoles intégrant des mécanismes cryptographiques (WEP, TKIP, WPA)
  - Groupe de travail 802. Normalisation de l'accès (couche 1 et 2)
  - Groupe de travaille 802.11 sur l'accès radio
  - Groupe 1363 travaille sur les algorithmes de chiffrement asymétrique
  - http://www.ieee.org
- IETF (Internet Engineering Task Force)
  - Communauté ouverte d'architecte réseaux, d'opérateurs, d'équipementiers, de fournisseurs et de chercheurs concernés par l'évolution de l'architecture de l'Internet. L' « adhésion » est individuel.
  - La Mission de l'IETF est décrite dans le RFC3935
  - Conception des protocoles intégrant des mécanismes cryptographiques
  - Publication également des codes associés de fonctions cryptographiques (MD5)
  - IETF publie des RFCs
  - http://www.ietf.org

Annexe : standardisation et acteurs (de la cryptologie)

Industriels

- RSA Coorporation
  - La compagnie la plus active dans le domaine de la conception des algorithmes de chiffrement
  - Fondé par les concepteurs de l'algorithme asymétrique RSA
  - Principale contributeurs des standards sur la cryptographie asymétrique PKCS (Public Key Cipher System)
  - http://www.rsa.com
- 3GPP (Third Generation Partnership Project Structure gouvernementales)
  - Groupement des industriels des réseaux mobiles de troisième génération
  - Conception des algorithmes de cryptographie et des protocoles d'authentification (AKA)
  - http://www.3gpp.org
- ETSI (European Telecommunications Standard Institute)
  - Groupement des industriels des réseaux mobiles plutôt GSM (plus européen
  - Conception des algorithmes de cryptographie A3, A5, A8 et des protocoles d'authentification
  - http://www.3gpp.org

Annexe : standardisation et acteurs (de la cryptologie)

Structures gouvernementale

- NIST (National Institute of Standards and Technology) (anciennement NBS)
  - Agence fédérale du département américain du commerce
  - Promouvoir et propose des standards notamment en sécurité libre de droits
  - Accélere l'adoption de solutions par la conception de plateformes de test et de prototype
  - A l'origine de nombreux standards: SHA, ELGAMAL, A.E.S.
  - Les standards sont des FIPS (Federal Information Processing Standards)
  - http://www.nist.gov/
  - http://csrc.nist.gov/
- ANSSI - SSI (anciennement DCSSI)
  - Cellule dépendant du premier ministre Français
  - Organisme de régulation et de contrôle de la cryptographie à l'échelle nationale
  - Certificateur des produits de sécurité, délivre également les autorisations pour la fourniture, l'import, et l'export des produits basés sur la cryptographie

Challenge

- Extraire le modulo
  openssl rsa -pubin -in PublicKey.txt -text –modulus
- Convertir le modulo en décimal
  echo "ibase=16;valeurHexa" | bc
- script.rb

```
#!/usr/bin/env ruby
n=299992503
require 'prime'
Prime.each do |p|
 if (n % p)==0
   puts "prime1: #{p}"
 puts "prime2: #{n/p}"
 break
end
end
```

- Exécuter le script avec: ruby script.rb
  Voir le résultat

Challenge

- Structure ASN.1 d'une clef privée RSA

```
RSAPrivateKey ::= SEQUENCE {
version           Version,
modulus           INTEGER,  -- n
publicExponent    INTEGER,  -- e
privateExponent   INTEGER,  -- d
prime1            INTEGER,  -- p
prime2            INTEGER,  -- q
exponent1         INTEGER,  -- d mod (p-1)
exponent2         INTEGER,  -- d mod (q-1)
coefficient       INTEGER,  -- (inverse of q) mod p
otherPrimeInfos   OtherPrimeInfos OPTIONAL
}
```

Challenge

- Créer un fichier key.txt

```
asn1=SEQUENCE:rsa_key

[rsa_key]
version=INTEGER:0
modulus=INTEGER:187
pubExp=INTEGER:7
privExp=INTEGER:23
p=INTEGER:17
q=INTEGER:11
e1=INTEGER:7
e2=INTEGER:3
coeff=INTEGER:14
```

- Construire le fichier binaire DER

  openssl asn1parse -genconf key.txt -out newkey.der

- Format de la clé privée

  openssl rsa -in newkey.der -inform der -text -check

SMIME

- Générer votre paire de clés
- Créer une requête - certificat e-mail
- Demander la signature de la requête auprès de la CA locale
- Signer votre message avec le format S/MIME utilisé pour les e-mails
- Envoyer le message à votre collègue
- Vérifier la signature du message reçu

SMIME

- Signature

  openssl smime –sign –in data.txt –signer MyCertificat.crt –inkey MyPrivate.key – out signedmessage.sig

- Vérification

  Openssl smime –verify –in signedmessage.sig –CAfile CertCA.crt