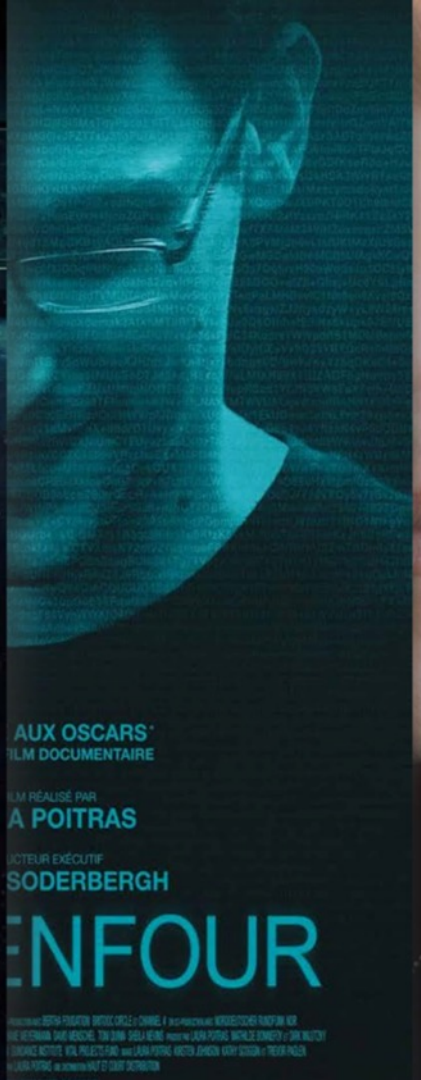

X GIBNEY OSCAR® WINNING DIRECTOR OF GOING CLEAR AND TAXI TO THE DARK SIDE
WORLD WAR 3.0
"A WHITE KNUCKLE THRILLER.
Clear, urgent, and positively terrifying at times."
-Peter Debruge, Variety
ZERO DAYS
COMING JULY 8

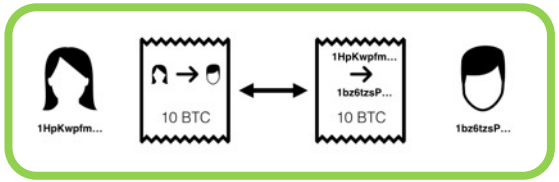

AUX OSCARS®
ILM DOCUMENTAIRE
LM RÉALISÉ PAR
A POITRAS
CTEUR EXÉCUTIF
SODERBERGH
ENFOUR

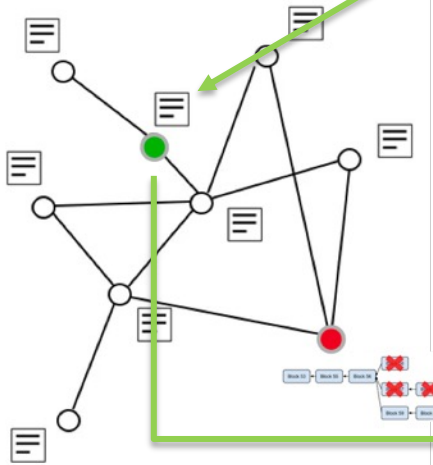

ONCE YOU'RE IN, THERE'S NO WAY
TEHRA

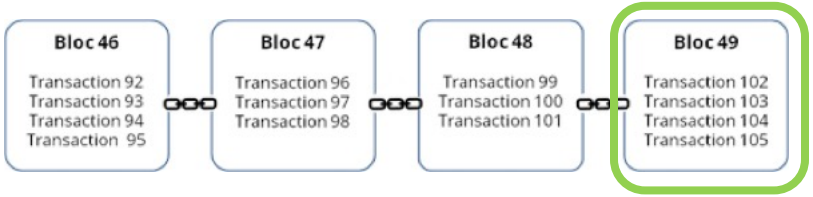
Bitcoin as Frankenstein creature ?

Bitcoin

Bitcoin

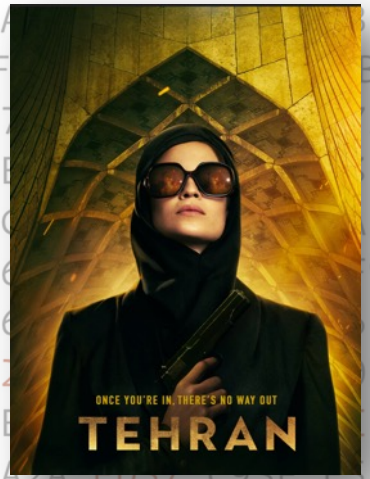






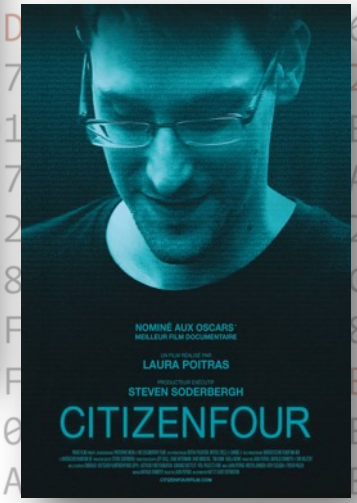
- preuve de possession - imputabilité
- anonymat - pseudonymat
- traçabilité - intégrité - non répudiation
- disponibilité - accessibilité
- évolutivité - adaptabilité - scalability
- autonomie et pérennité de fonctionnement

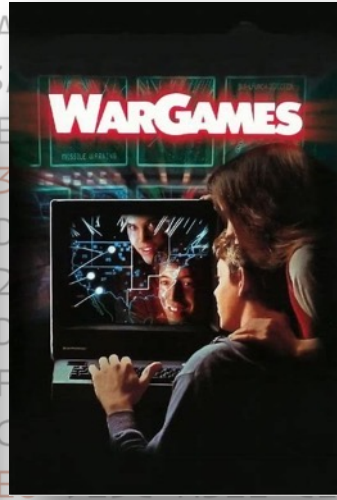
- consensus décentralisé et automatisé
- unicité de la dépense (anti double spending)


ONCE YOU'RE IN, THERE'S NO WAY OUT
TEHRAN


ZERO DAYS

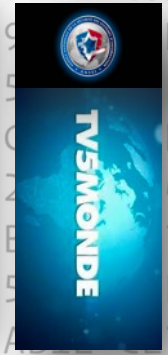

NOUS SOMMES
TOUS SUR
ÉCOUTE
JOSEPH GORDON-LEVITT
SHAILENE WOODLEY
SNOWDEN
OLIVER STONE


NOMINÉ AUX OSCARS
LAURA POITRAS
STEVEN SODERBERGH
CITIZENFOUR


WARGAMES


LE BUREAU DES LEGENDES

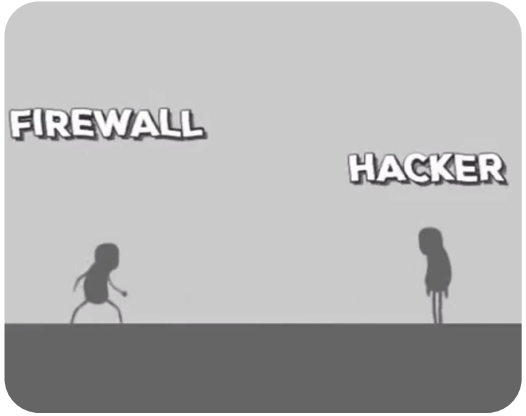

MR. ROBOT

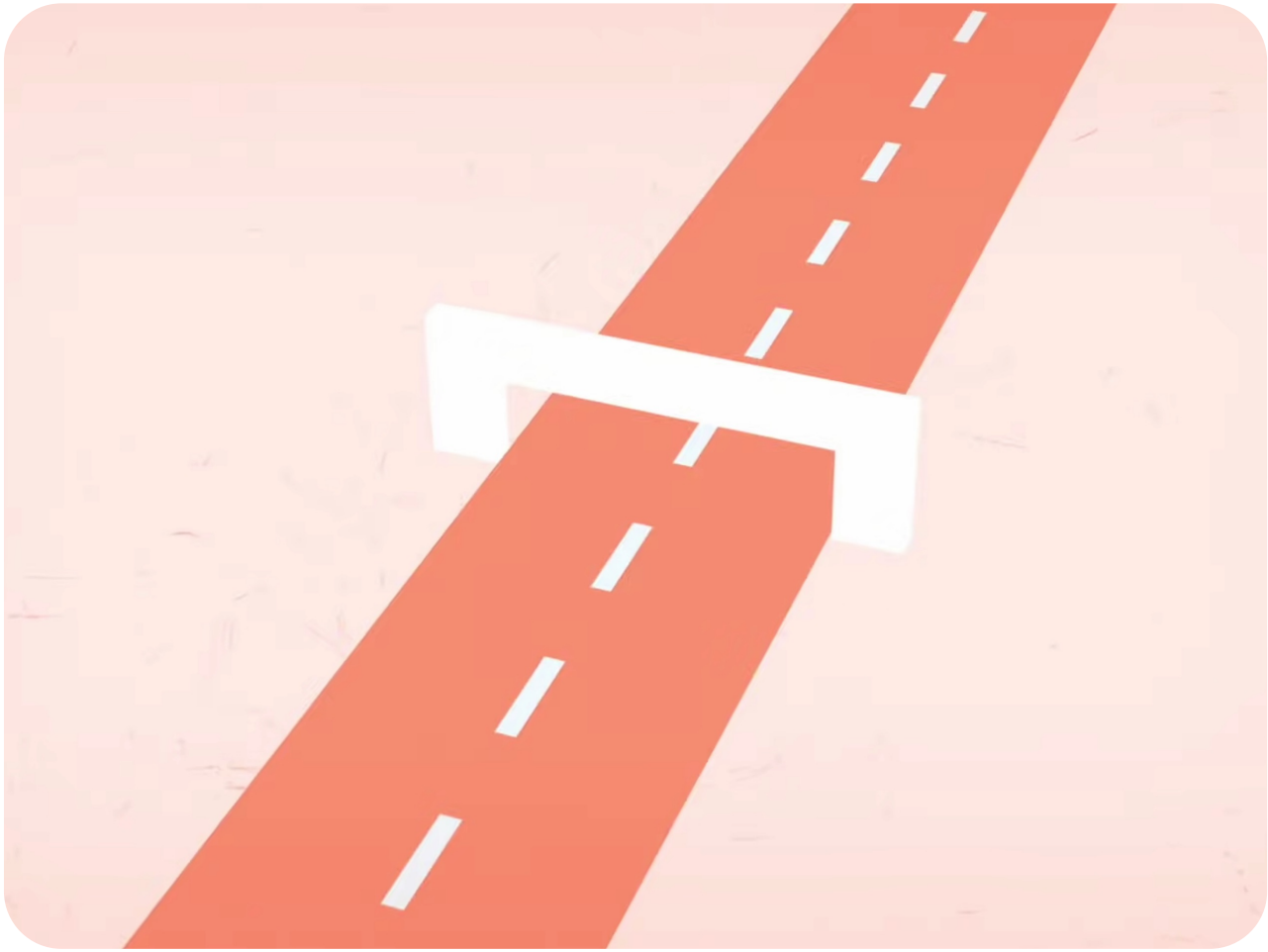

TV5MONDE

Cabinet d’Experts Cybersécurité

STELAU

le hacking
facile





Historique

rien n'a vraiment changé

il y a 27 ans,
le 11 août 1996 un certain **aleph_one**
publie
"*Smashing The Stack For Fun And Profit* »
dans une revue de hackers (*phrack n° 49*).

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

`smash the stack` [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

Introduction

Over the last few months there has been a large increase of buffer overflow vulnerabilities being both discovered and exploited. Examples of these are syslog, splitvt, sendmail 8.7.5, Linux/FreeBSD mount, Xt library, at, etc. This paper attempts to explain what buffer overflows are, and how their exploits work.

Basic knowledge of assembly is required. An understanding of memory concepts, and experience with gdb are very helpful. We also assume we are working with an Intel x86 CPU, system is Linux.

UaF + dF
RUST

```
testsc2.c
----------------------------------------------------------------------
char shellcode[] =
        "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0
        "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xc
        "\x80\xe8\xdc\xff\xff\xff/bin/sh";

void main() {
   int *ret;

   ret = (int *)&ret + 2;
   (*ret) = (int)shellcode;

}
----------------------------------------------------------------------
[aleph1]$ gcc -o testsc2 testsc2.c
[aleph1]$ ./testsc2
$ exit
[aleph1]$
----------------------------------------------------------------------

                    Writing an Exploit
                    ~~~~~~~~~~~~~~~~~~
                (or how to mung the stack)
                ~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Ce qu'il faut retenir :
N'importe quel appareil équipé d'un CPU qui fait tourner du code est potentiellement vulnérable.

US-984XN

http://goo.gl/SNEXRo

2:17 Former CIA Director: 'We Kill People Based On Metadata'

PRISM

STELAU

Stuxnet – Juin 2010

Un avant et un après Stuxnet ?

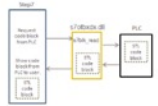
Objectif : ralentir le programme d'enrichissement nucléaire Iranien

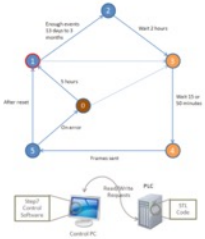
Moyen : saboter le fonctionnement des centrifugeuses d'enrichissement

Cible : atteindre la centrale de *Bushehr* et les centrifugeuses nucléaires de *Natanz*

Prise en compte de 33 types de convertisseurs de fréquences de 2 fabricants par le protocole Profibus

Step7 et PCL avec s7otbxdx.dll



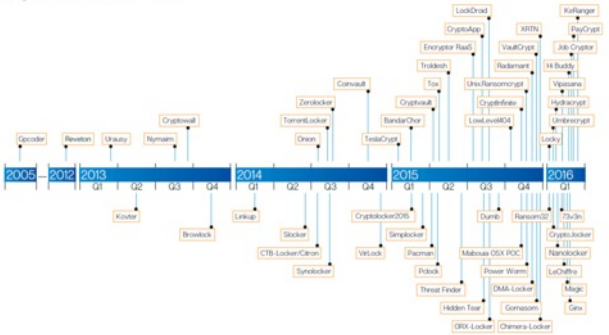


http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

STELAU

2016 année des ransomwares

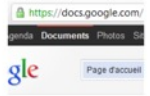
explosion des cas



STELAU

DigiNotar – Juillet-Août 2011

Affaire grave ou très grave



Objectif : politico-idéologico-religio…
hacker iranien ComodoHacker

Moyen : prise de contrôle totale du HSM de DigiNotar
(autorité de certification reconnue et importante
émettant certains certificats du gouvernement Hollandais)

Cible : le HSM de DigiNotar ? un simple PC hébergeant tous les systèmes de génération de certificats … très mal protégé

Conséquences: 500 certificats frauduleux fabriqués et un certificat *.google.com ayant permis l'espionnage d'utilisateurs Iraniens du 10 juillet au 29 aout 2011

Correctifs / Réactions : MAJ de tous les navigateurs et OS
+ audits de 54 autorités de certification

STELAU

Equation Group

and the Shadow Brockers

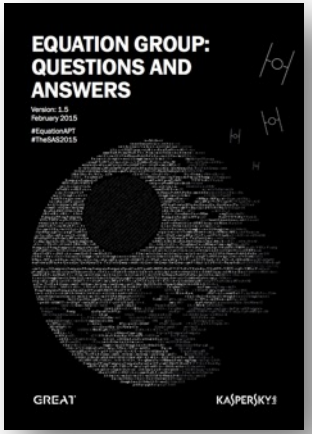
The first step is executing `bc-genpkt`, which generates an IKE packet of arbitrary size and fills some of it with arbitrary data.

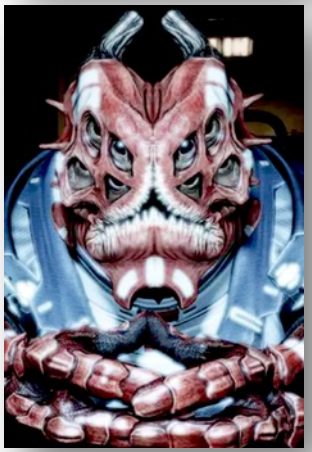
```
Usage: ./bc-genpkt [-h] [-o <file>] [-f <X>] [-r] [-s] [-v[vv]] size

    -h  help/usage
    -o file write data to named file
    -f X    fill remainder of large packets with character 'X'
    -r  randomize the initiator cookie
    -s  randomize the SPI
    -v[vv]  verbosity - show lengths, packet dumps, etc
    size    size of new packet, should be 96 <= size <= 65536 bytes

    Packets larger than 2528 bytes will be filled with random data
    unless the -f option is used.
```

This generates a packet file which can be used as input to the binary `bc-id`, which sends the packet to the victim host. Hector Martin notes that it sends a IKE packets with a large Group-Prime option, and speculates that if the victim host is replying using the request length but only filling in the requested 768 bit prime, then it returns a buffer of uninitialised data following it.

```
Usage:
   ./bc-id -t <dest IP> []
  Options:
      -t <dest IP>
      -l <local port>
      -p <remote port>
      -I <infile name>  [defaults to sendpacket.raw]
      -O <outfile name> [defaults to ".raw"]
      -f <packetfile name> Reads in packet from a file.
      -h              print this message
      -q              quiet mode.  Doesn't print hex of response pack
```

The strings in the `bc-id` binary shows that the program seems to patch some memory and look for a start string in the response. However Hector Martin





Outils NSA

quand la NSA perd ses outils

GitHub, Inc. [US] https://github.com/misterch0c/shadowbroker

- **EARLYSHOVEL** RedHat 7.0 - 7.1 Sendmail 8.11.x exploit
- **EBBISLAND (EBBSHAVE)** root RCE via RPC XDR overflow in Solaris 6, 7, 8, 9 & 10 (p and x86.
- **ECHOWRECKER** remote Samba 3.0.x Linux exploit.
- **EASYBEE** appears to be an MDaemon email server vulnerability
- **EASYFUN** EasyFun 2.2.0 Exploit for WDaemon / IIS MDaemon/WorldClient pre 9.5.6
- **EASYPI** is an IBM Lotus Notes exploit that gets detected as Stuxnet
- **EWOKFRENZY** is an exploit for IBM Lotus Domino 6.5.4 & 7.0.2
- **EXPLODINGCAN** is an IIS 6.0 exploit that creates a remote backdoor
- **ETERNALROMANCE** is a SMB1 exploit over TCP port 445 which targets XP, 2003, V 2008 R2, and gives SYSTEM privileges (MS17-010)
- **EDUCATEDSCHOLAR** is a SMB exploit (MS09-050)
- **EMERALDTHREAD** is a SMB exploit for Windows XP and Server 2003 (MS10-061)
- **EMPHASISMINE** is a remote IMAP exploit for IBM Lotus Domino 6.6.4 to 8.5.2
- **ENGLISHMANSDENTIST** sets Outlook Exchange WebAccess rules to trigger execut to send an email to other users
- **EPICHERO** 0-day exploit (RCE) for Avaya Call Server
- **ERRATICGOPHER** is a SMBv1 exploit targeting Windows XP and Server 2003
- **ETERNALSYNERGY** is a SMBv3 remote code execution flaw for Windows 8 and Serv
- **ETERNALBLUE is** a SMBv2 exploit for Windows 7 SP1 (MS17-010)
- **ETERNALCHAMPION** is a SMBv1 exploit
- **ESKIMOROLL** is a Kerberos exploit targeting 2000, 2003, 2008 and 2008 R2 domai
- **ESTEEMAUDIT** is an RDP exploit and backdoor for Windows Server 2003
- **ECLIPSEDWING** is an RCE exploit for the Server service in Windows Server 2008 an
- **ETRE** is an exploit for IMail 8.10 to 8.22
- **ETCETERABLUE** is an exploit for IMail 7.04 to 8.05

les Shadow Brokers fournissent la clé PGP

```
msf auxiliary(smb_ms17_010) > options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS     192.168.1.177    yes       The target address range or CIDR identifier
   RPORT      445              yes       The SMB service port (TCP)
   SMBDomain  .                no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

msf auxiliary(smb_ms17_010) > exploit

[*] 192.168.1.177:445     - Connected to \\192.168.1.177\IPC$ with TID = 2048
[*] 192.168.1.177:445     - Received STATUS_INSUFF_SERVER_RESOURCES with FID = 0
[!] 192.168.1.177:445     - Host is likely VULNERABLE to MS17-010!
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

WannaCry

Mai 2017

ETERNALBLUE

MS17-10

« We kill people based on metadata »

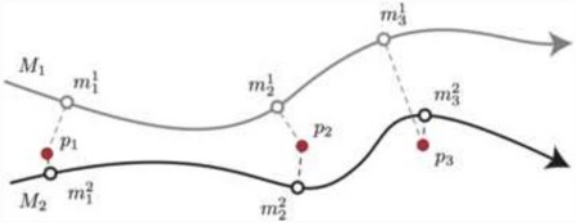
NSA Director Michael Hayden - "We Kill People Based on Metadata"

1,1 k vues • il y a 6 ans

The Silentist

Published on Apr 7, 2014 The Price of Privacy: Re-Evaluating the NSA, A Debate This year's Presidential

Figure 1



Trajectory-based identification. Two traces M_1 (grey) and M_2 (black) along with a set of three points (red) sampled from M_2. These points are classified as belonging to M_2 because the average distance to the corresponding nearest points in M_2 is lower than the average distance to the nearest points in M_1.

Les 3 objectifs de ce cours

- La cryptographie moderne c'est 5 primitives à connaître

- Essayer de comprendre l'aspect « *révolutionnaire* » de la cryptographie asymétrique

- En déduire le concept de signature cryptographique

Les vraies difficultés de la cryptographie moderne

1. THEORIE : Cryptologie
   - failles théoriques = mathématiques
   - cryptologue est un métier

2. CODE : Implémentations
   - erreurs/failles/vuln. = informatique
   - « *Do not implement cryptography yourself* ! »

3. UX : Usages
   - mauvais usages = ignorance/pusillanimité
   - bons usages = formation

Poncifs cybersécuritaires

- la faille c'est l'humain 🤓
- le filtre d'écran 🤢
- les cookies du RGPD 🤮
- « c'est sécurisé … » 😎
- « c'est crypté, c'est sécurisé » 😎😎

Sondage ?

Chiffrement symétrique

Chiffrement asymétrique

Vocabulaire cryptographique

- Cryptologie - science (**λόγος**) du secret (**κρυπτός**) :
  - cryptographie
  - cryptanalyse
- « *Do not implement crypto yourself* »
- Le bureau du chiffre du quai d'Orsay
- « *crypter* » est un néologisme

La cryptologie ne se limite plus aujourd'hui à assurer la confidentialité des secrets

- Cybersécurité = sécurité informatique = SSI
- SSI : sécurité des systèmes d'information
- C'est ~~crypté~~ chiffré donc c'est sécurisé … ne veut rien dire en SSI.

Algo. de chiffrement	Avec Clé	Sans Clé
message clair	chiffrer	~~*crypter*~~
message chiffré	déchiffrer	décrypter

La Cryptographie Moderne

F96D E8C2 27A2 59C8 7EE1
DA2A ED57 C93F E5DA 36ED
4EC8 7EF2 C63A AE5B 9A7E
FFD6 73BE 4ACF 7BE8 923C
AB1E CE7A F2DC F7AE 29A3
DA44 F235 A24C 963F F0DF
3CA3 599A 70E5 DA36 BF1E
CE77 F8DC 34BE 129A 6CF4
D126 BF5B 9A7C FEDF 3EB8
50D3 7CF0 C63A A250 9A76
FF92 27A5 5B9A 6FE3 D720

Modern Cryptography

Cryptographie

é	c3 a9
🔥	f0 9f 94 a5

```
F96D E8C2 27A2 59C8 7EE1
DA2A ED57 C93F E5DA 36ED
4EC8 7EF2 C63A AE5B 9A7E
FFD6 73BE 4ACF 7BE8 923C
AB1E CE7A F2DC F7AE 29A3
DA44 F235 A24C 963F F0DF
3CA3 599A 70E5 DA36 BF1E
CE77 F8DC 34BE 129A 6CF4
D126 BF5B 9A7C FEDF 3EB8
50D3 7CF0 C63A A250 9A76
FF92 27A5 5B9A 6FE3 D720
```

9 A 7 E

1001 1010 0111 1110

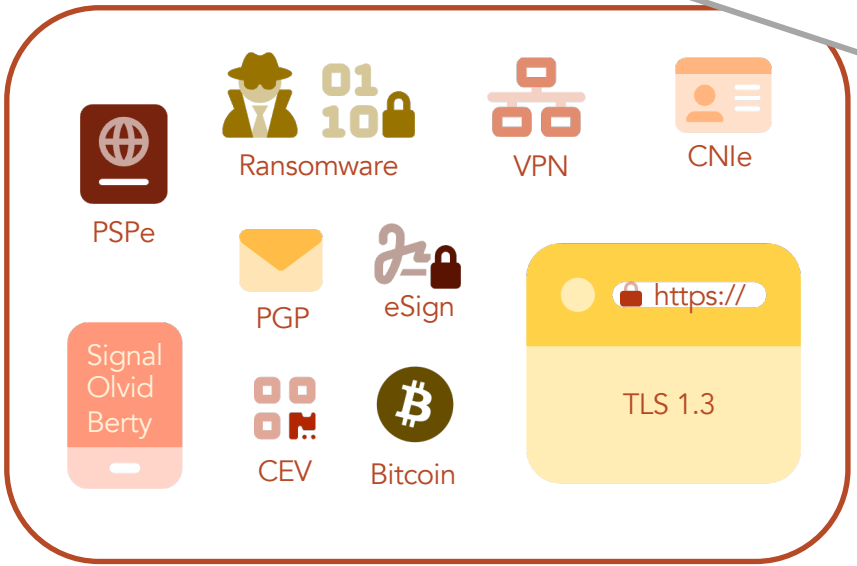
8 bits = 1 octet 8 bits = 1 octet

```
B  o  n  j  o  u  r
42 6f 6e 6a 6f 75 72
01000010 01101111 01101110 01101010 01101111 01110101 01110010
```

Cryptographie

F96D E8C2 27A2 59C8 7EE1
DA2A ED57 C93F E5DA 36ED
4EC8 7EF2 C63A AE5B 9A7E
FFD6 73BE 4ACF 7BE8 923C
AB1E CE7A F2DC F7AE 29A3
DA44 F235 A24C 963F F0DF
3CA3 599A 70E5 DA36 BF1E
CE77 F8DC 34BE 129A 6CF4
D126 BF5B 9A7C FEDF 3EB8
50D3 7CF0 C63A A250 9A76
FF92 27A5 5B9A 6FE3 D720

La cryptographie ne se limite plus aujourd'hui à assurer la confidentialité des secrets



Cryptographie

La cryptographie ne se limite plus aujourd'hui à assurer la **confidentialité** des secrets

1. Confidentialité
2. Intégrité
3. Authenticité

4. Disponibilité
5. Originalité
6. Non-répudiation

7. Traçabilité
8. Preuve à divulgation nulle (zero-knowledge)

	originalité	authenticité	intégrité	confidentialité	non répudiation	traçabilité
	✓	✓	✓			✓
		✓	✓	✓	✓	✓
		✓	✓		✓	✓
	✓	✓	✓	✓	✓	✓
		✓	✓		✓	
		✓	✓	✓		
		✓	✓	✓		
				✓		

Les 5 primitives cryptographiques

Chiffrement Symétrique

Chiffrement Asymétrique

Fonctions de hachage

Établissement de clé

Générateurs d'aléa

Cryptographie moderne

PSPe

Ransomware

VPN

CNIe

PGP

eSign

Signal
Olvid
Berty

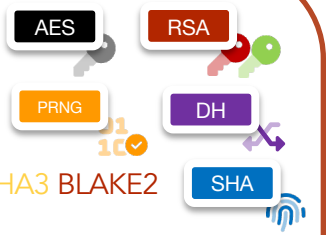
CEV

Bitcoin

https://

TLS 1.3

Crypto. Symétrique : AES 3DES
Crypto. Asymétrique : RSA ECC
Générateur d'aléa : PRNG
Etablissement de clé : DH ECDH
Hashage cryptographique : SHA1 SHA3 BLAKE2



Montages simples : HMAC TOTP S/MIME CEV
Montages robustes : Bitcoin PGP TLS PSPe CNIe
Montages complexes : TLS IKE Ransomware WannaCry

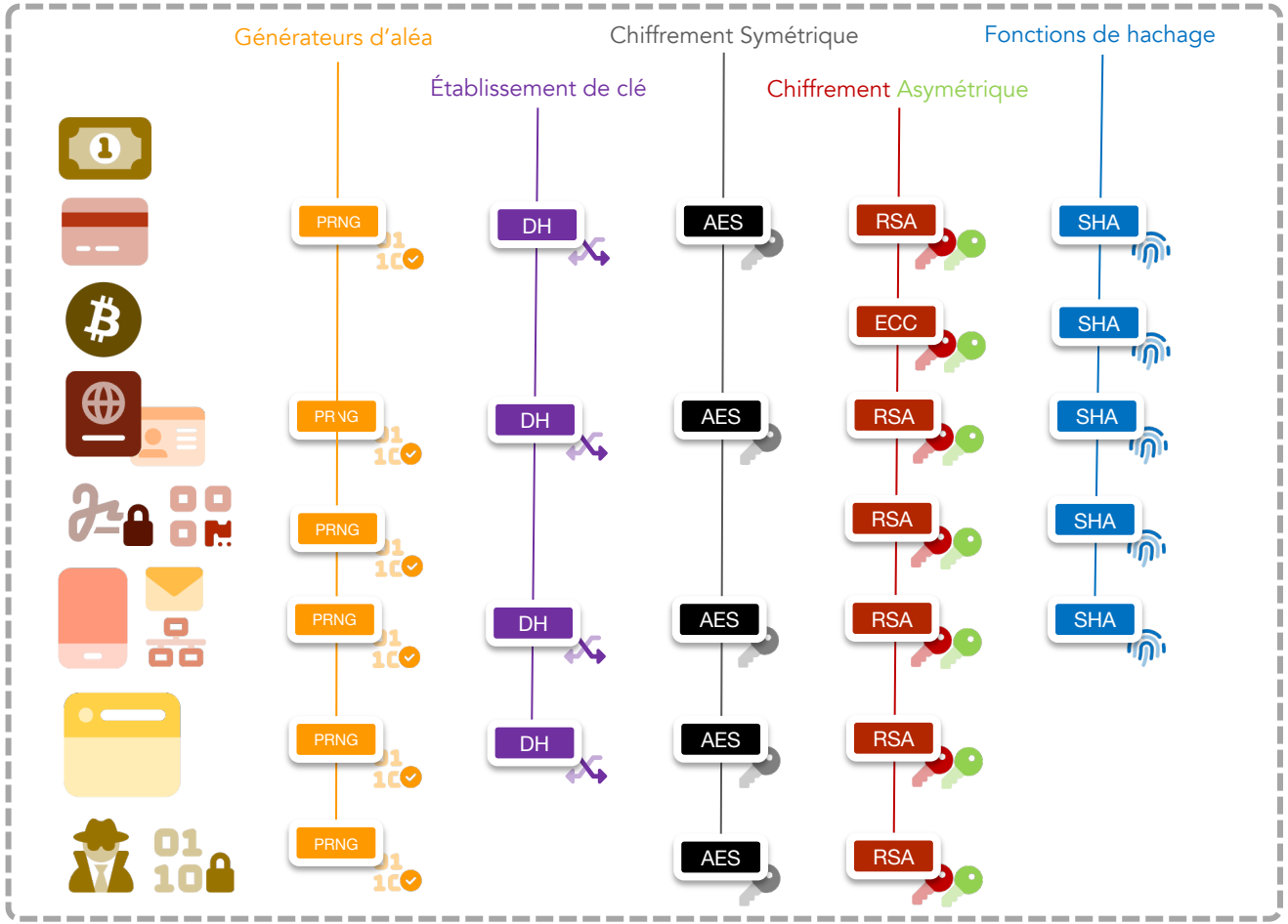
Cryptographie

La cryptographie ne se limite plus aujourd'hui à assurer la **confidentialité** des secrets

1. Confidentialité
2. Intégrité
3. Authenticité

4. Disponibilité
5. Originalité
6. Non-répudiation

7. Traçabilité
8. Preuve à divulgation nulle (zero-knowledge)



Les 5 primitives cryptographiques

Chiffrement Symétrique

Chiffrement Asymétrique

Fonctions de hachage

Établissement de clé

Générateurs d'aléa

Hello !

le secret

M

C

M

Hello !

26, 47, 10

91690410bec9

graine

798

aléa

Very Short Crypto Story

3000 ans de crypto. symétrique

recettes militaro-diplomatiques
de confusion et de diffusion

Confusion et **Diffusion**
« *tant bien que mal* »
de César à Enigma

100 ans de crypto. moderne

de Kerckhoffs ...
au crypto-système incassable

1. Principes de Kerckhoffs - 1883
2. One Time Pad - 1917

---

50 ans de crypto. asymétrique

LA véritable révolution

Résout la difficulté de
l'échange de clé
+
Permet l'usage du
principe de **Signature**

20 ans de crypto. quantique

révolution ? (ou pas)

<Crypto-Story>

Savoir réagir après avoir cassé le code de son adversaire

Feindre d'ignorer ce qu'on sait,
de savoir tout ce qu'on ignore, ...
avoir souvent pour grand secret
de cacher qu'il n'y en a point, ...

Beaumarchais - Le Mariage de Figaro (1778)

Confusion et Diffusion

Substitution et Permutation

Cryptographie Symétrique => Confusion et Diffusion

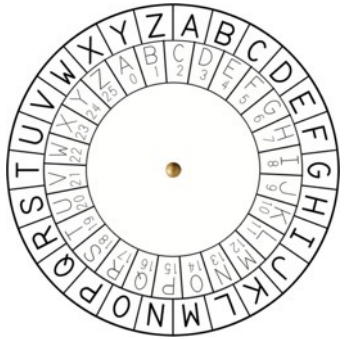
Confusion par Substitution

A → E L → T L → T O → Y ...

Diffusion par Permutation / Transposition

J E S U I S P A S L A => S I U S E J A L S A P

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



Principes de Kerckhoffs - 1883

Crypto symétrique moderne

Auguste Kerckhoffs - 1883

1. Il faut qu'il n'exige **pas le secret**, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi.

2. La **clef** doit pouvoir **en être communiquée** et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.

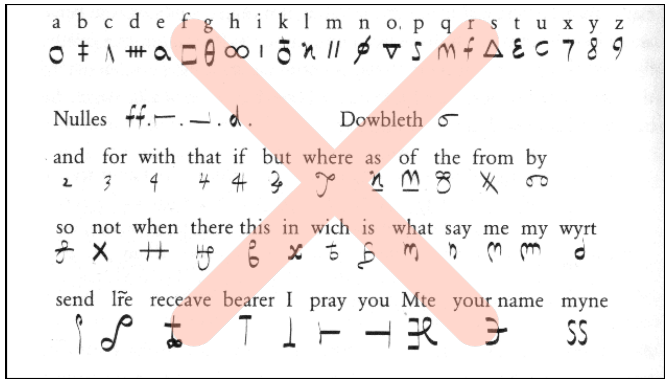
=> ce qui est gardé secret doit être ce qui est le moins coûteux à changer si le secret s'avérait divulgué

Notion de **clé secrète**

Algorithme connu de tous :

- Substitution
- Permutation

mais basées sur la clé secrète



Chiffrement Symétrique

cryptosystème parfait : « incassable »

Le Masque Jetable - One Time Pad

1917

- La clé doit être une suite de caractères au moins aussi longue que le message à chiffrer
- Les caractères composant la clé doivent être choisis de façon aléatoire
- Chaque clé, ou « masque », ne doit être utilisée qu'une seule fois
  - d'où le nom de masque jetable

One-Time Pad

• **Plain text:**	**H**	**O**	**W**	**A**	**R**	**E**	**Y**	**O**	**U**
	7	14	22	0	17	4	24	14	20
+									
OTP:	**13**	**2**	**1**	**19**	**25**	**16**	**0**	**17**	**23**
	N	C	B	T	Z	Q	A	R	X
Initial total:	**20**	**16**	**23**	**19**	**42**	**20**	**24**	**31**	**43**
Mod 26:	**20**	**16**	**23**	**19**	**16**	**20**	**24**	**5**	**17**
Ciphertext:	**U**	**Q**	**X**	**T**	**Q**	**U**	**Y**	**F**	**R**

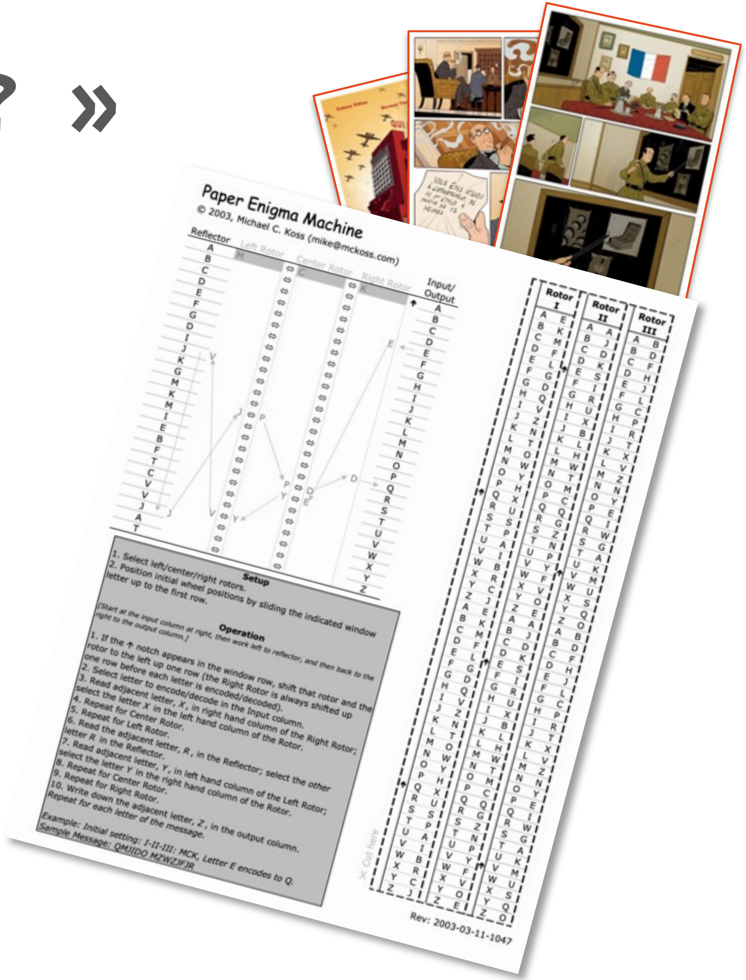
« Qui a cassé Enigma ? »

1932. Dans la salle de bains d'un hôtel bruxellois, un espion français photographie les premiers documents décrivant une nouvelle machine à coder *a priori* inviolable : Enigma. Une machine que s'apprêtent à adopter les services secrets allemands. Quelques mois plus tard, avec l'aide des Français, un groupe de mathématiciens polonais entreprend de percer à jour le fonctionnement complexe de la machine.

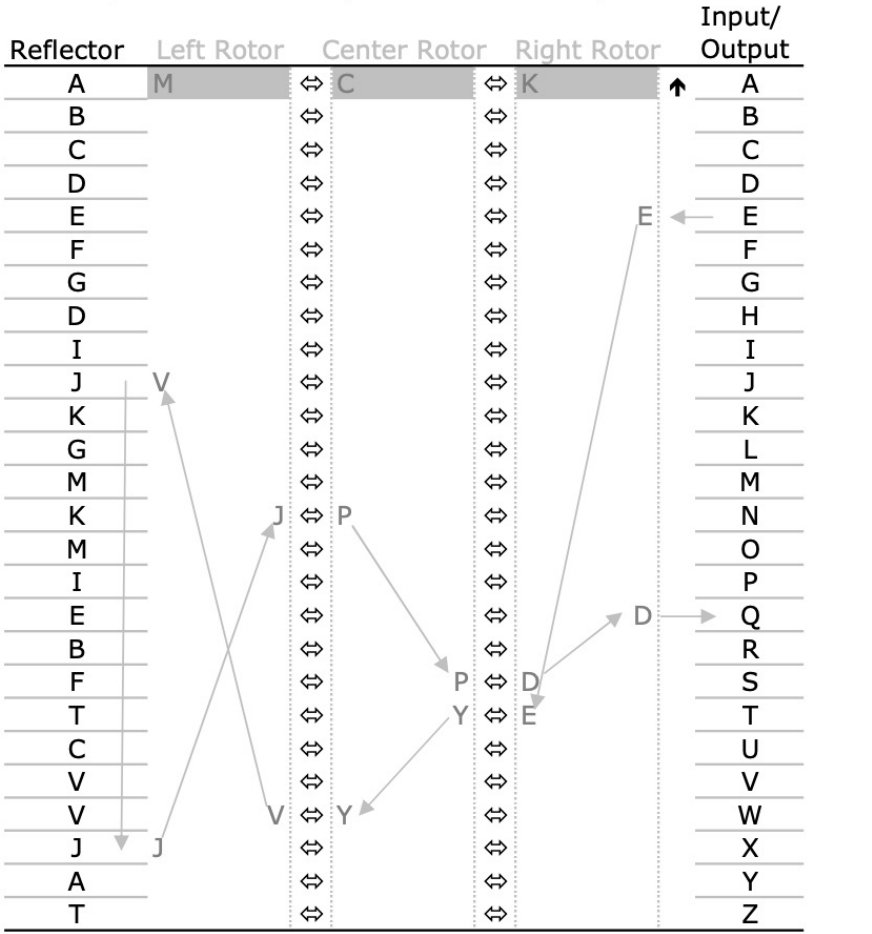
1940. Après la défaite française face aux nazis, les Français et les Polonais transmettent leurs trouvailles aux Britanniques. À Bletchley Park se déploie une gigantesque entreprise de décodage dont va dépendre l'issue de la guerre.

1942. Sous le nez des Allemands, à Vichy même, Français et Polonais continuent leurs efforts de décodage. La Gestapo est à leurs trousses et le MI-6 a pour priorité absolue de les exfiltrer. Pendant ce temps dans l'Atlantique, les U-Boote allemands mènent une traque dévastatrice contre les navires américains qui ravitaillent la Grande-Bretagne en armes, vivres et marchandises. Si on ne réussit pas très vite à décoder les messages de la Marine allemande, le Royaume-Uni ne tiendra pas.

À Bletchley Park, un des cerveaux les plus brillants de l'histoire scientifique, **Alan Turing**, va apporter une contribution décisive...



Paper Enigma Machine



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

Very Short Crypto Story

3000 ans de crypto. symétrique — *recettes militaro-diplomatiques de confusion et de diffusion*

100 ans de crypto. moderne — *de Kerckhoffs ... au crypto-système incassable*

50 ans de crypto. asymétrique — *LA véritable révolution*

DH 1976

RSA 1977

20 ans de crypto. quantique — *révolution ? (ou pas)*

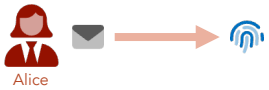
Les dates clefs de la cryptographie

https://fr.wikipedia.org/wiki/Histoire_de_la_cryptologie

Hachage

Hachage cryptographique

SHA — Secure Hash Algorithm



SHA

empreinte cryptographique

Une fonction de hachage cryptographique est une fonction sans clé qui permet de transformer une donnée de taille arbitraire en une chaîne de bits de taille h fixe, appelée haché.

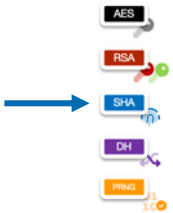
1. ~~Confidentialité~~
2. Intégrité
   - Très rapide
   - Gros volume
3. ~~Authenticité~~
4. ~~Disponibilité~~
5. ~~Originalité~~
6. ~~Non-répudiation~~
7. Traçabilité

R3

Fonctions de hachage

Les fonctions de hachage de la famille SHA-2 [FIPS180] et de la famille SHA-3 [FIPS202] dont la taille de sortie est supérieure ou égale à 256 bits sont recommandées.

Primitive	Taille de paramètre	R/O	Notes
SHA-2 [FIPS180, ISO10118-3]	h = 256 bits (SHA-256)	R	
	h = 384 bits (SHA-384)	R	
	h = 512 bits (SHA-512)	R	
	h = 256 bits (SHA-512/256)	R	
SHA-3 [FIPS202]	h = 256 bits	R	
	h = 384 bits	R	
	h = 512 bits	R	



1. BLAKE
2. RIPEMD
3. WHIRLPOOL

Source : Guide de sélection d'algorithmes cryptographiques - ANSSI

Hachage cryptographique

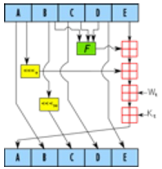
SHA

Secure
Hash
Algorithm

SHA1

b1e9feb2d6015f3fa4bfac79788cb21f03560984

```
SHA1-compress(H, M) {
    (a0, b0, c0, d0, e0) = H   // parsing H as five 32-bit big endian words
    (a, b, c, d, e) = SHA1-blockcipher(a0, b0, c0, d0, e0, M)
    return (a + a0, b + b0, c + c0, d + d0, e + e0)
}
```




```
SHA1-blockcipher(a, b, c, d, e, M) {
    W = expand(M)
    for i = 0 to 79 {
        new = (a <<< 5) + f(i, b, c, d) + e + K[i] + W[i]
        (a, b, c, d, e) = (new, a, b >>> 2, c, d)
    }
    return (a, b, c, d, e)
}
```

```
expand(M) {
    // the 512-bit M is seen as an array of sixteen 32-bit words
    W = empty array of eighty 32-bit words
    for i = 0 to 79 {
        if i < 16 then W[i] = M[i]
        else
            W[i] = (W[i - 3] ⊕ W[i - 8] ⊕ W[i - 14] ⊕ W[i - 16]) <<< 1
    }
    return W
}
```

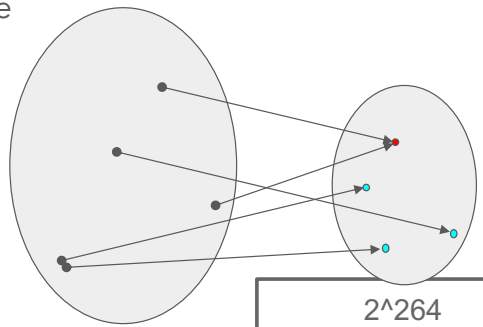
```
f(i, b, c, d) {
    if i < 20 then return ((b & c) ⊕ (~b & d))
    if i < 40 then return (b ⊕ c ⊕ d)
    if i < 60 then return ((b & c) ⊕ (b & d) ⊕ (c & d))
    if i < 80 then return (b ⊕ c ⊕ d)
}
```

- Fonctions à sens unique

  d'un espace infini vers un espace fini de :

  128, 160, 224, 256 ou 512 bits

- Résistantes aux attaques
  - 1ère pré-image
  - 2de pré-image
  - collisions



2^264
2^(80*3,3)
10^80
nb. atomes univers

Hachage

Hachage cryptographique

SHA

Secure
Hash
Algorithm

SHA256

```
f9dcc621fcb8ac6a172c40fd3ffcbfcf20fa400e3b216d3777362ee7c22965ea
```

$$\text{SHA256}(\text{“Guess \#0”}) = \begin{matrix} 11010001010111001000101011010001 \\ 01111100101100000011100100000111 \\ 11101100101001100101110111101100 \\ 00010011101000111011111000101011 \\ 10000001001111000011001100100111 \\ 10001101011011101001000110000101 \\ 11101011111101011000101011011000 \\ 10001110100000011000011010000000 \end{matrix}$$

Hachage

Hachage cryptographique

SHA

Secure
Hash
Algorithm

La cryptographie moderne est une branche de la cryptologie qui utilise des algorithmes mathématiques pour protéger et sécuriser les informations. Elle est devenue essentielle dans le monde numérique pour garantir la confidentialité, l'intégrité et l'authenticité des données échangées. Les principaux objectifs de la cryptographie moderne sont de garantir la confidentialité, l'authenticité et l'intégrité des données. Elle est utilisée dans de nombreux domaines tels que les transactions financières, la communication en ligne, la sécurité des réseaux informatiques, et la protection des données personnelles.

SHA1 → 09843fa4bfac79788cb21b1e9feb2d6015ff0356

SHA1 → 3a1637db49251af125ebc74662eb6c9e9dda0c16

Benoit LEGER-DERVILLE
12 mars 1974
5 rue de la Villa
65432 Bourg-la-Ville

SHA1 → ba81264abab6e296cc48838ebb9df78db1970267

SHA1 → 5304418f60d04748d4aee15393248381515aa993

SHA1 → 194a0023224bf8db193346aa8b09cdfc0689f1fd

SHA1 → 7a372c801f2d5b766cdd0fca47785fa4efaa8ee3

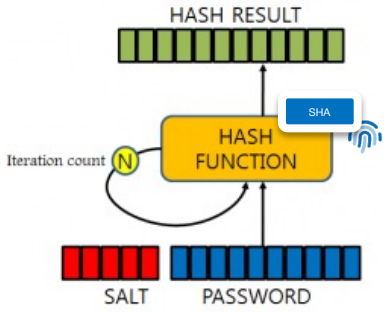
Hachage

Hachage cryptographique

SHA

Secure
Hash
Algorithm

P@55w0rD/2023 → SHA → `6967e32dfaafdde85d7d40db50cd906c9200caee`

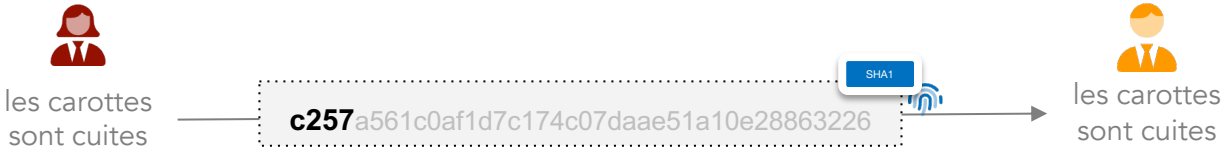


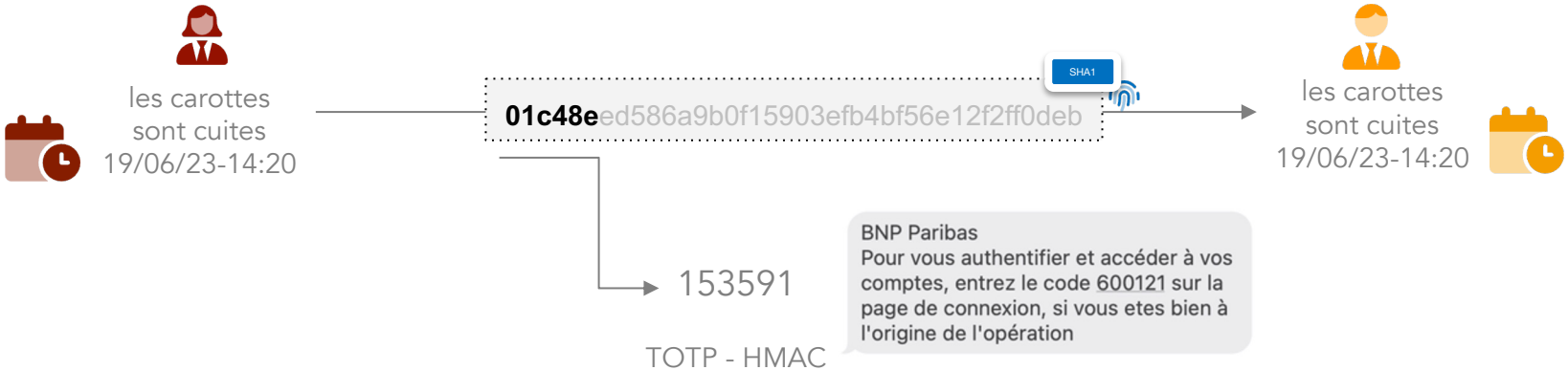
Hachage

Hachage cryptographique

SHA

Secure
Hash
Algorithm





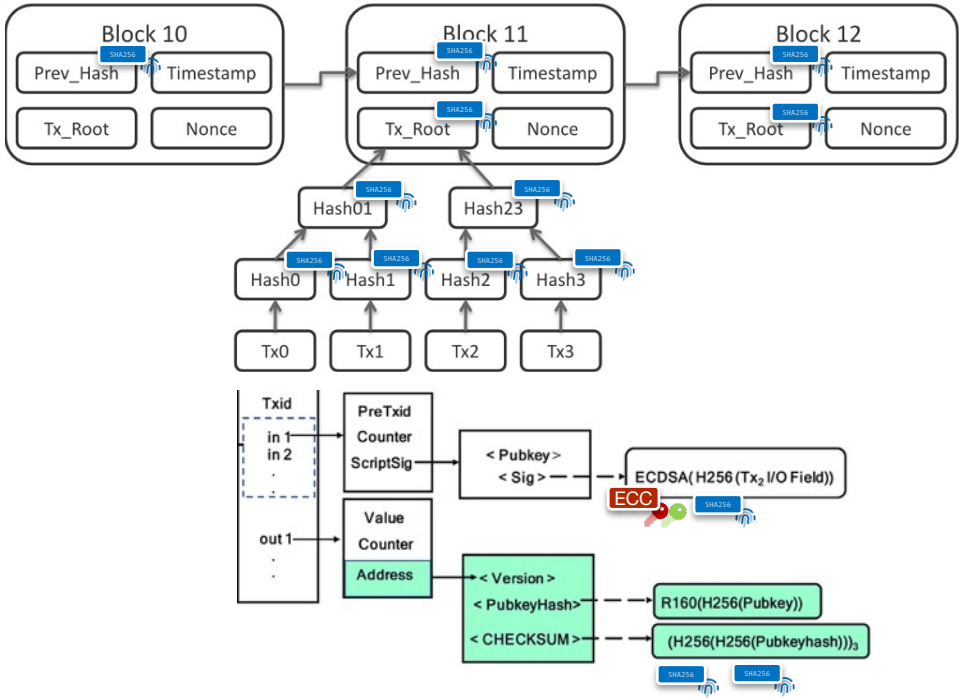
Hachage

Hachage cryptographique

SHA — Secure Hash Algorithm

SHA256

`f9dcc621fcb8ac6a172c40fd3ffcbfcf20fa400e3b216d3777362ee7c22965ea`



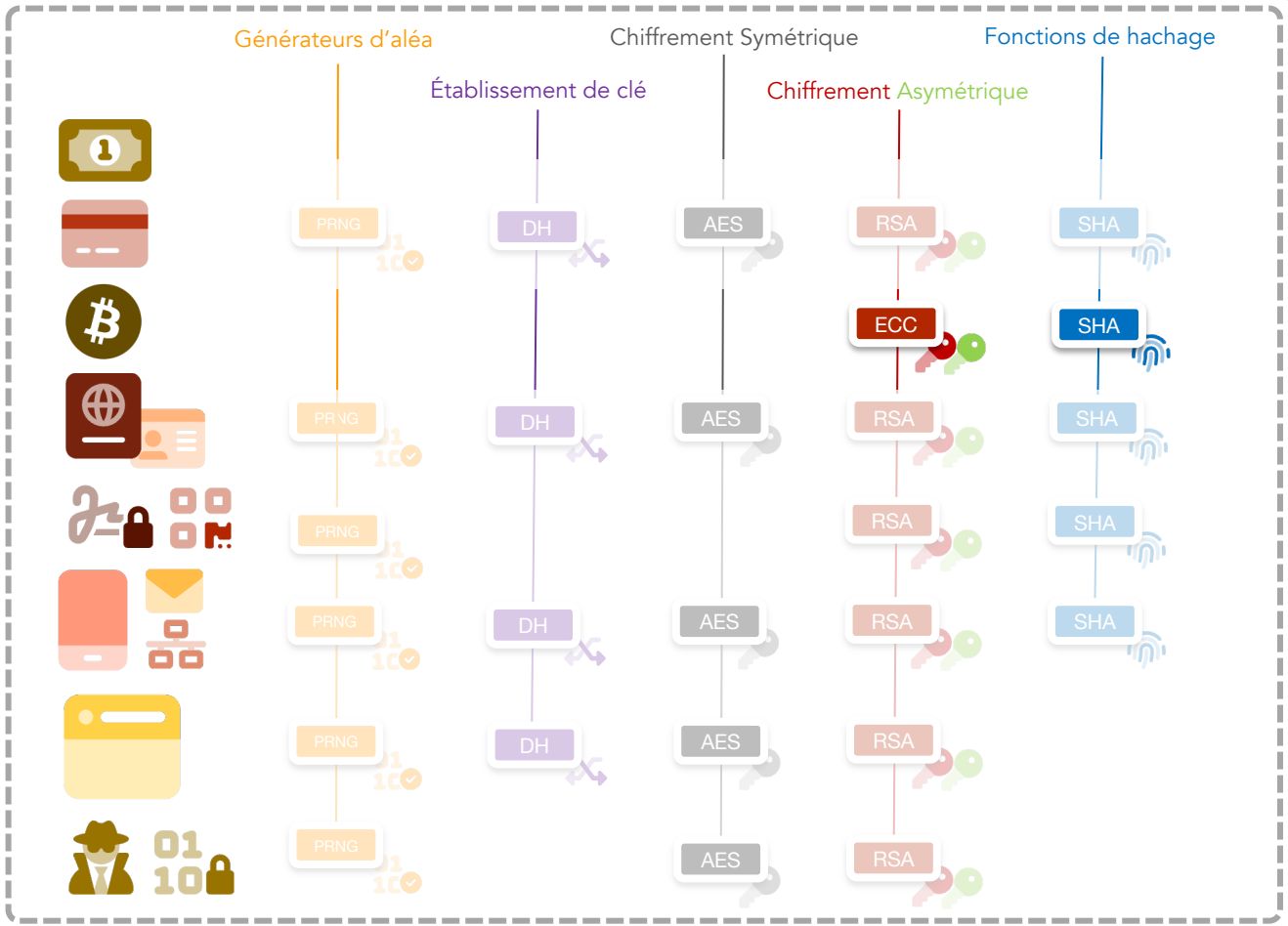
Cryptographie

La cryptographie ne se limite plus aujourd'hui à assurer la **confidentialité** des secrets

1. Confidentialité
2. Intégrité
3. Authenticité

4. Disponibilité
5. Originalité
6. Non-répudiation

7. Traçabilité
8. Preuve à divulgation nulle (zero-knowledge)



Génération d'Aléa

Pseudo Aléatoire

PRNG

Pseudo Random Number Generator

Alice — s →

91690410bec9 (graine) → PRNG → 798 (aléa)

Une source d'aléa est un processus probabiliste duquel on peut extraire une séquence de bits aléatoires. Il est difficile de s'assurer de la qualité des sorties d'une source d'aléa.

Deux approches : 1. Réaliser des tests statistiques sur les sorties de la source ; 2. Modéliser le processus stochastique de la source.

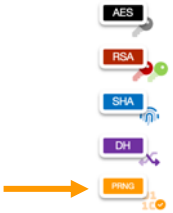
1. Confidentialité
2. Intégrité
3. Authenticité
4. Disponibilité
5. Originalité
6. Non-répudiation
7. Traçabilité

R23

Générateur d'aléa déterministe

HMAC-DRBG, Hash-DRBG et CTR-DRBG [FIPS197, ISO18033-3] sont recommandés.

Schéma	R/O	Notes
HMAC-DRBG [SP800-90A, ISO18031]	R	
Hash-DRBG [SP800-90A, ISO18031]	R	
CTR-DRBG [SP800-90A, ISO18031]	R	



1. Test de primalité : Miller-Rabin
2. RSA : $|p - q| \geq 2^{n/2 - 100}$

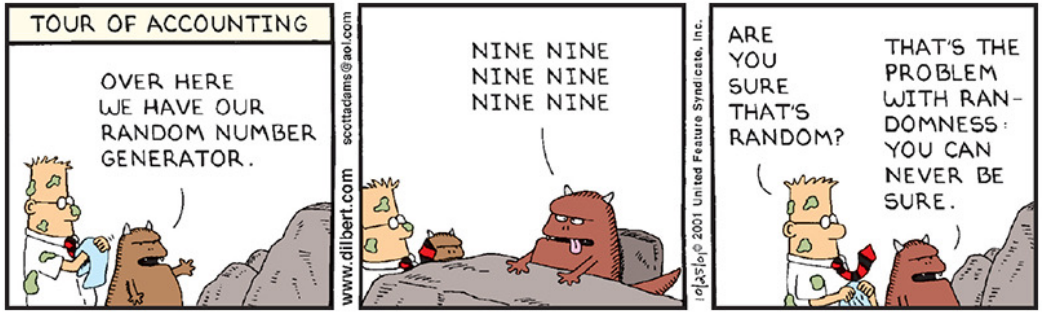
Source : Guide de sélection d'algorithmes cryptographiques - ANSSI

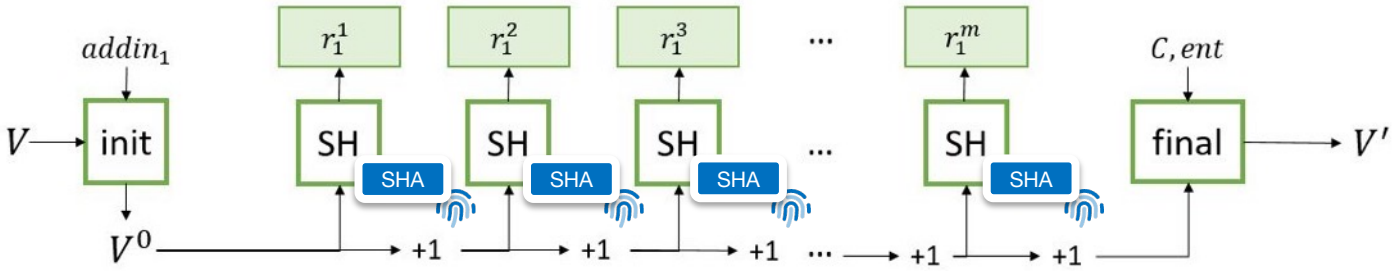
Génération d'Aléa

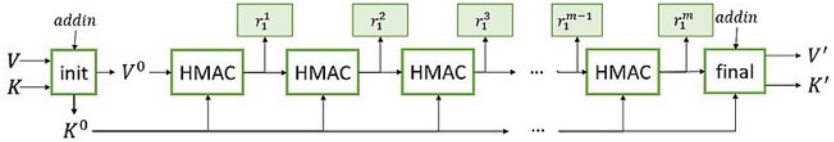
Pseudo Aléatoire

PRNG

Pseudo Random Number Generator



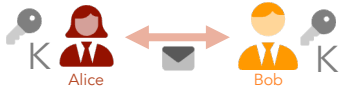




Symétrique

Chiffrement Symétrique

AES — Advanced Encryption Standard



1. Confidentialité
   - Très rapide
   - Gros volume
2. ~~Intégrité~~
3. ~~Authenticité~~
4. ~~Disponibilité~~
5. ~~Originalité~~
6. ~~Non répudiation~~
7. ~~Traçabilité~~

Un mécanisme de chiffrement symétrique permet, à l'aide d'une clé secrète K, de transformer un message clair M en un message chiffré C

R1 — Algorithmes de chiffrement par bloc

AES [FIPS197, ISO18033-3] est recommandé pour ses trois tailles de clés (128, 192 et 256 bits).

Primitive	Taille de clé	Taille de bloc	R/O	Notes
AES [FIPS197, ISO18033-3]	k = 128 bits	n = 128 bits	R	
	k = 192 bits		R	
	k = 256 bits		R	
Triple-DES [SP800-67, ISO18033-3]	k = 112 bits	n = 64 bits	O	3.1.a, 3.1.b
	k = 168 bits		O	3.1.b

R2 — Algorithmes de chiffrement par flot

L'algorithme de chiffrement par flot Chacha20 [RFC8439] est recommandé.

Primitive	R/O	Notes
ChaCha20 [RFC8439]	R	

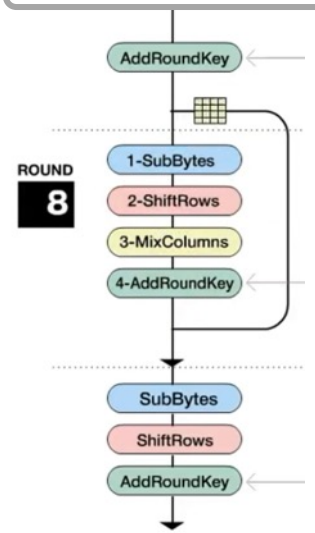
1. AES
2. Blowfish
3. DES
4. 3DES
5. IDEA
6. RC4
7. RC5
8. Serpent
9. Twofish

Source : Guide de sélection d'algorithmes cryptographiques - ANSSI

Symétrique

Chiffrement Symétrique

AES — Advanced Encryption Standard



Source : https://www.youtube.com/watch?v=gP4PqVGudtg

Chiffrement Symétrique

Advanced
Encryption
Standard

Qui écoute la planète ?

- Les USA
- Les Chinois
- Les Belges

Rijndael

Joan Daemen et Vincent Rijmen

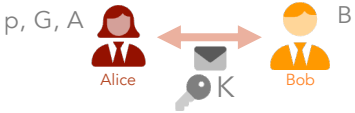
Keccak

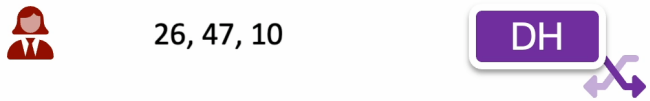
Etablissement de clé

Partage de secret

DH

Diffie Hellman





Un mécanisme d'établissement de clé permet à deux entités (ou plus) de se mettre d'accord sur une valeur de clé secrète.

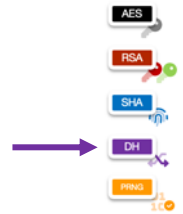
1. Confidentialité
   - Rapide
   - vuln. MitM
2. ~~Intégrité~~
3. ~~Authenticité~~
4. ~~Disponibilité~~
5. ~~Originalité~~
6. ~~Non-répudiation~~
7. ~~Traçabilité~~

R22

Mécanismes d'établissement de clé

Les mécanismes d'établissement de clé DH et EC-DH sont recommandés.

Primitive	Mécanisme	R/O	Notes
FF-DLOG	DH [SP800-56A, ISO11770-3]	R	6.4.a
EC-DLOG	EC-DH [SP800-56A, ISO11770-3]	R	6.4.a



1. ECDH

Source : Guide de sélection d'algorithmes cryptographiques - ANSSI

Partage de secret

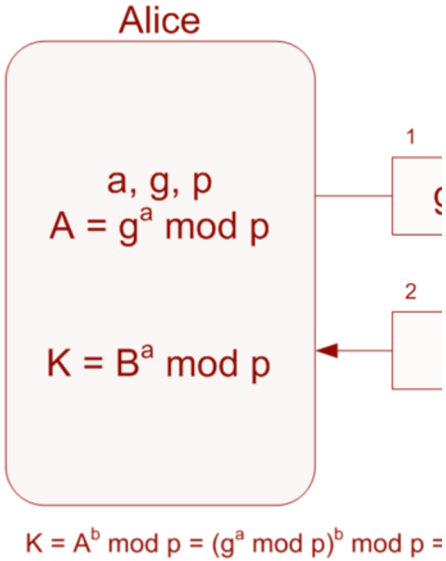
DH

Diffie
Hellman

Discrete Log Problem

$2^n = 16$

$2^n \bmod 17 = 16$

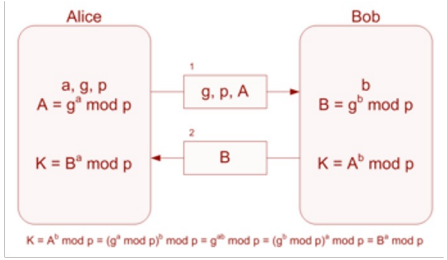


$3 \bmod 17 \equiv$

Partage de secret

DH

Diffie
Hellman



1. Confidentialité

2. Attaque MitM
   Man in the Middle

~~3. Intégrité~~

~~4. Authenticité~~

~~5. Disponibilité~~

~~6. Originalité~~

~~7. Non-répudiation~~

~~8. Traçabilité~~

G:	1112413672743230891881273653171640834687833299890125651932066327486193358080471215100797237775771467149894260123512660136402158406998369481252844452718796
N:	7781871412403125272081994420237498498848517960190636813994593624893166160780445513887821412313009361625936687452586174696909303141691170346568788390764347

Next Bob and Alice will generate two random numbers (X and Y), calculate an X value and a Y value, respectively:

Bob's X Value	39	**Alice's Y value**	86
	Bob's random value		Alice's random value
Bob's A value	156437054090510643708213487397359650653706600103085237076814872023455692501938943883924390306994339326651518339656294271164634334019120166300414013853096 1	**Alice's B value**	867562183713756921586080163515215184994404228948268998120566392949664779371069184674731003959263884633988486672884899488683959430851633340784964621708228
	A=G^x mod N		B=G^y mod N

and Bob will send his A value to Alice, and Alice will send her B value to Bob, and they now re-calculate the values to generate the **same shared key**:

Bob's Key	844421731057512944560047379488773309578989566491934683447580616259308802773327317656211099035659946951701937245204428699328935731311556825137634874050285	Alice's Key	844421731057512944560047379488773309578989566491934683447580616259308802773327317656211099035659946951701937245204428699328935731311556825137634874050285
	Key=B^x mod N		Key=A^y mod N

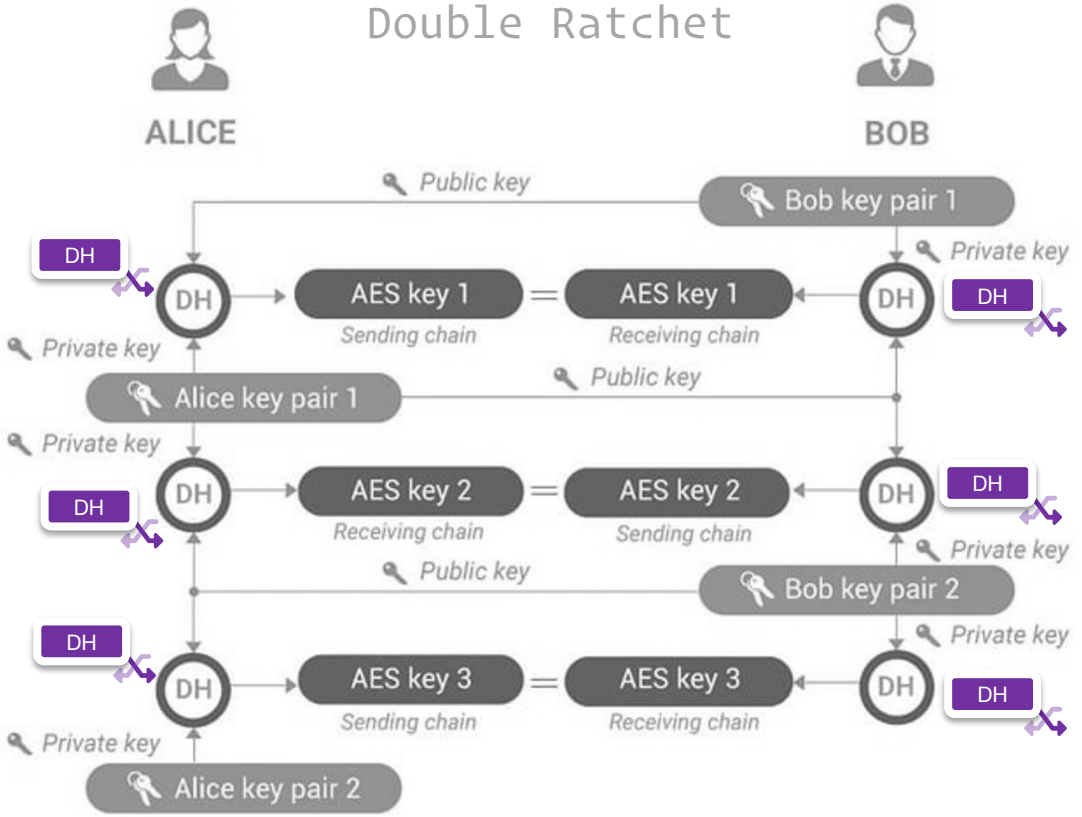
Partage de secret

DH

Diffie Hellman

- **Symmetric end-to-end encryption.** Message decrypted on all recipients' ends with the sar exchange.
- **Forward secrecy.** Unique ephemeral keys ar compromised, all your other messages rema therefore safe.
- **Independent key renewal.** The algorithm doe get new keys. It uses key derivation function
- **Plausible deniability.** If a message gets inter who has sent it.
- **No lost or out-of-order messages.** Each mes header. This way, if a message gets lost or u chain.

Double Ratchet

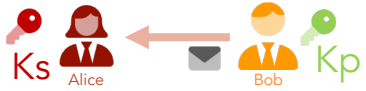


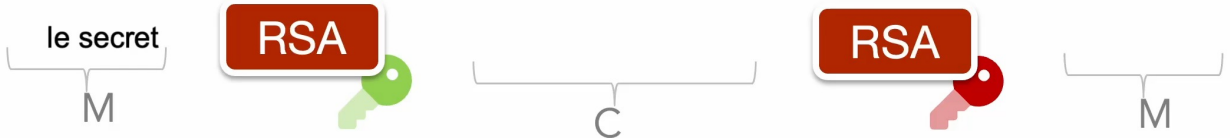
Asymétrique

Chiffrement Asymétrique

RSA

Rivest
Shamir
Adleman





L'opération publique de chiffrement transforme à l'aide de la clé publique Kp un message clair M en un message chiffré C. L'opération privée de déchiffrement permet de recalculer M à partir de C et de la clé privée Ks. Le chiffrement asymétrique permet donc à toute personne ayant accès à la clé publique de chiffrer des messages à l'intention du détenteur de la clé privée.

1. Confidentialité
   - Lent
   - Petit volume
2. ~~Intégrité~~
3. Authenticité
4. ~~Disponibilité~~
5. Originalité
6. Non-répudiation
7. ~~Traçabilité~~

R18

Paramètres de courbes elliptiques pour le DLOG

Les paramètres de courbes elliptiques P256r1, P384r1 et P512r1 de la fa pool, les paramètres de courbes elliptiques P-256, P-384 et P-521 du paramètres de courbes Curve25519 et Curve448 sont recommandés.

Familles de courbes	Courbes	R/O	Notes
Brainpool [RFC5639]	BrainpoolP256r1	R	5.3.a
	BrainpoolP384r1	R	
	BrainpoolP512r1	R	
NIST [FIPS186] (voir Annexe D.1.2)	NIST P-256	R	5.3.a
	NIST P-384	R	
	NIST P-521	R	
IETF [RFC7748]	Curve25519	R	5.3.b
	Curve448	R	

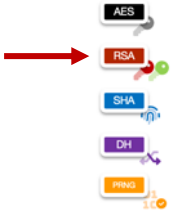
R16

Dimensionnement du schéma asymétrique RSA

On recommande d'utiliser des modules RSA d'au moins 3072 bits publics e de taille strictement supérieure à 2^{16}.

Primitive	Taille des paramètres	R/O	Notes
RSA	$n \geq 3072, \log_2(e) > 16$	R	
	$n \geq 2048, \log_2(e) > 16$	O	

1. ECC
2. Post Quantique
   - FALCON
   - SPHINCS+
   - CRYSTALS-KYBER
   - CRYSTALS-DILITHIUM



Source : Guide de sélection d'algorithmes cryptographiques - ANSSI

Asymétrique

Chiffrement Asymétrique

Rivest
Shamir
Adleman

p et q premiers
n=p.q
$\varphi(n) = (p - 1)(q - 1)$

e premier avec $\varphi(n)$
d inverse de e modulo $\varphi(n)$

$$C \equiv M^e \ (mod\ n)$$

$$M \equiv C^d \ (mod\ n)$$

- *Comme* $c=m^e \bmod n$, $c^d \bmod n = m^{ed} \bmod n$
- *Comme* $ed = 1 \bmod (p-1)(q-1)$ *il existe un entier k tel que* $ed = 1 + k(p-1)(q-1)$
- *Par conséquent* $c^d \bmod n = m^{1+k(p-1)(q-1)} \bmod n$
- *Or (théorème de Fermat)* $m^{(p-1)} \bmod p = 1$ *si m n'est pas multiple de p. Par élévation à la puissance k(q-1) puis multiplication par m on obtient :* $m^{1+k(p-1)(q-1)} \bmod p = m$ *égalité qui reste vraie (2 membres=0) si m est multiple de p*
- *Par symétrie* $m^{1+k(p-1)(q-1)} \bmod q = m$ *donc* $m^{1+k(p-1)(q-1)} - m$ *est divisible par p et q donc par pq (p et q premiers et différents)*
- *Par conséquent* $c^d \bmod n = m^{1+k(p-1)(q-1)} \bmod pq = m$

Asymétrique

Chiffrement Asymétrique

Rivest
Shamir
Adleman

$$C \equiv M^e \ (mod\ n)$$

$$M \equiv C^d \ (mod\ n)$$

p et q premiers
n=p.q
φ(n) = (p - 1)(q -1)

e premier avec φ(n)
d inverse de e modulo φ(n)

Message=hello

p=178073160948557126481066827279199589991471119317587515137880407422837840796990039076227830307920605683888462626592386879221836763255789116306115475356732510317134601972410072595829099995663862495962938839420792464181544007162357012759562578827662767529379905026780616246336287030939352827543039555774118880861

q=165913198990242931132064773335662836078668169463718043341256089400405261062088687848791945416784551815961572361115057192773107752072542696403751434980938696269955622031354160320970858076125518083081550380613322799355585425770840866287627064491097888784189682166960030796126554428589177031865285035399557919043

e=65537

d=754158844612049696625323462034471233350106979530304991655547818102473961274876871484930767002008752049805075385396933676984272196317705375990451300233291178609665541211739105149498446063417988602161801091618458620366472987817610621858053924301783257306475237413563917372230187274291021292952018564551721175629226044233740822768641591594005786809854260573238815006082225726301776815307304847004390652930521925116847168781097305937840013363364431841434819869871137449204223710920038333303046675715777184108821767096958029838594954090281987848501950948557088360341346723410869909782189658972296699953260552783260756351 3

N=295446877872795151938154245509911777273393275853174715990971590381862848945373934643480827766218229320868403524016730482327205732142228807617228805447959853981359618473272435496339767228514703323688672115137778244981514377443976013657189790405678578122389070681886176417142983940434040662778387474834276709234192372780729815997187350367415629163310148194097664461912596526751407330054681332004258072992808137891577120158661948277174748254453829033860064224494201067057447815640287748796678939831122247930143951606695548203913045284760438130510844259880483102317018508493449083312299070937455103511329012871720013602 3

Public key (e,n) Private key (d,n)

cipher=273743492279956537274854241317285273489687949926990383595577818303469365389369383595210177750765405360365710781579824329830305899759170691700224655572029292953466922904049974236228138933712837677798511174101407776261740120993841843494109034252357715708754150858729358036738829777908947650774051399543308967832986991616410972346874156044067581889268723764995402779650371468342794857706752356242882149636133999260072349301958518578960262847313752426012359065189237744326344835963663284896710050292993953459275970081116400819064990504268034858744518958991288065553666631228513430001665872030553725667373795109184098702 53

Decipher=hello

Les 5 primitives cryptographiques

Chiffrement Symétrique

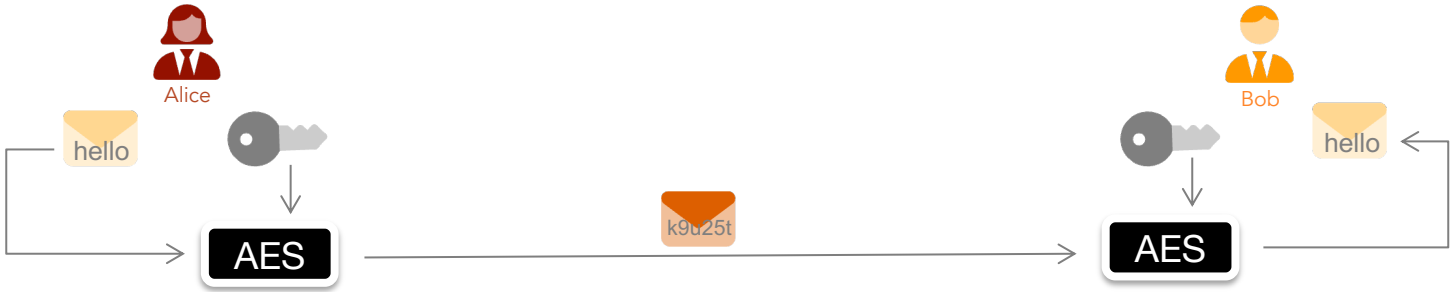
Chiffrement Asymétrique

Fonctions de hachage

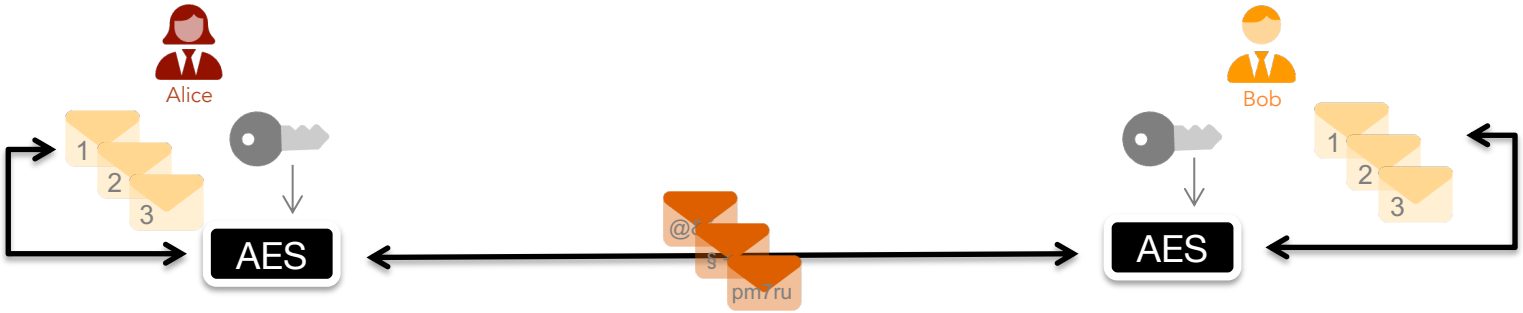
Établissement de clé

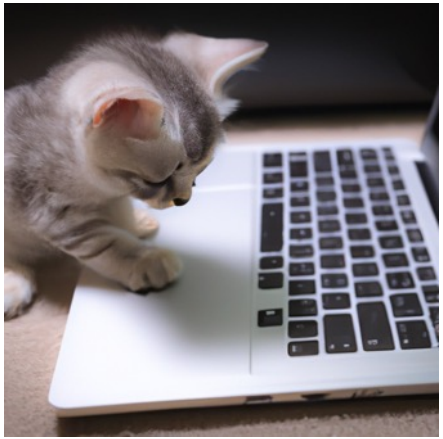
Générateurs d'aléa

Chiffrement symétrique

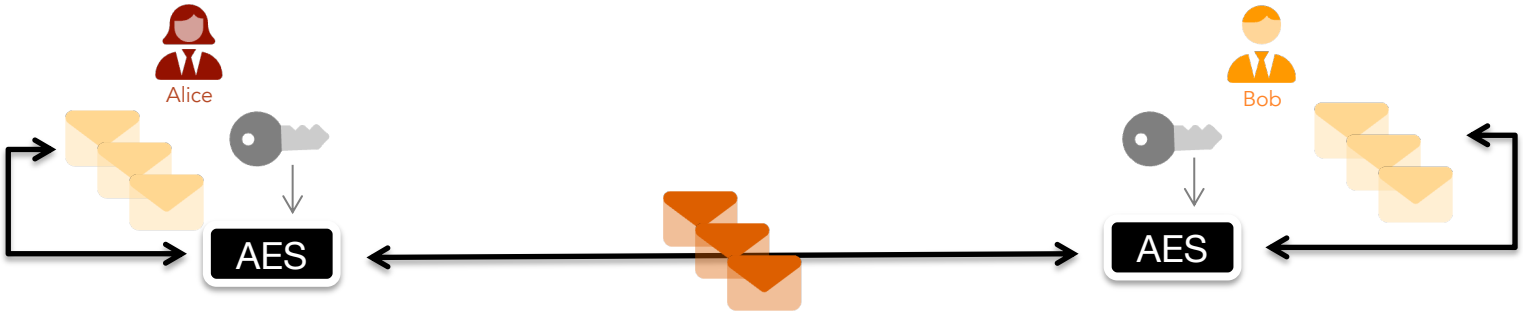


Chiffrement symétrique





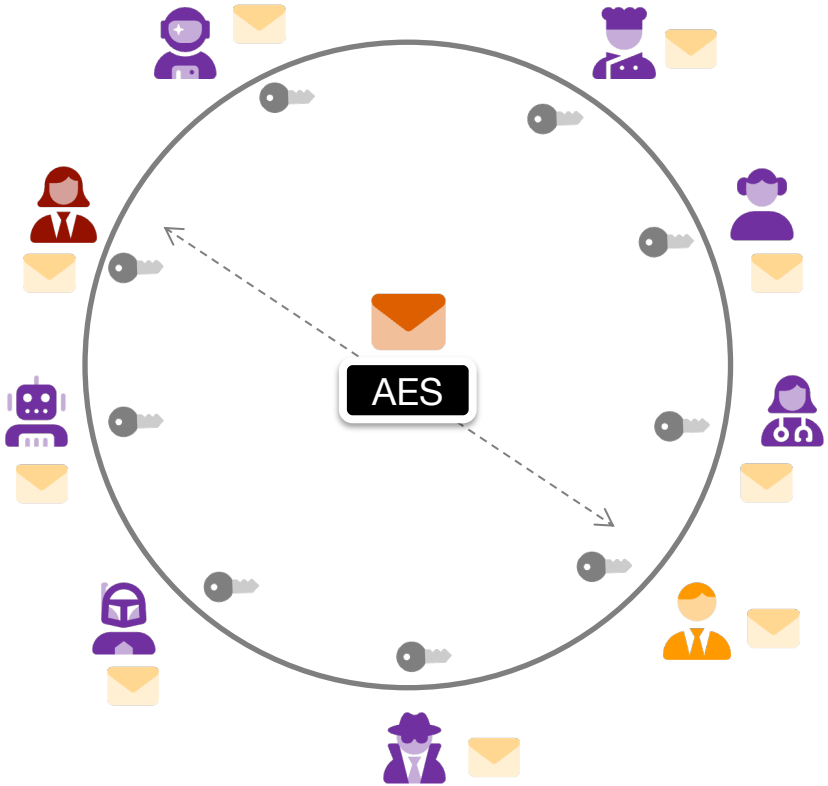
Chiffrement symétrique



- rapide et optimisé
- chiffrement de larges volumes
- usage simple et direct (dans les 2 sens : A<=>B)
- d'où vient cette unique clé symétrique ?
- établissement de la clé
- partage/échange de la clé

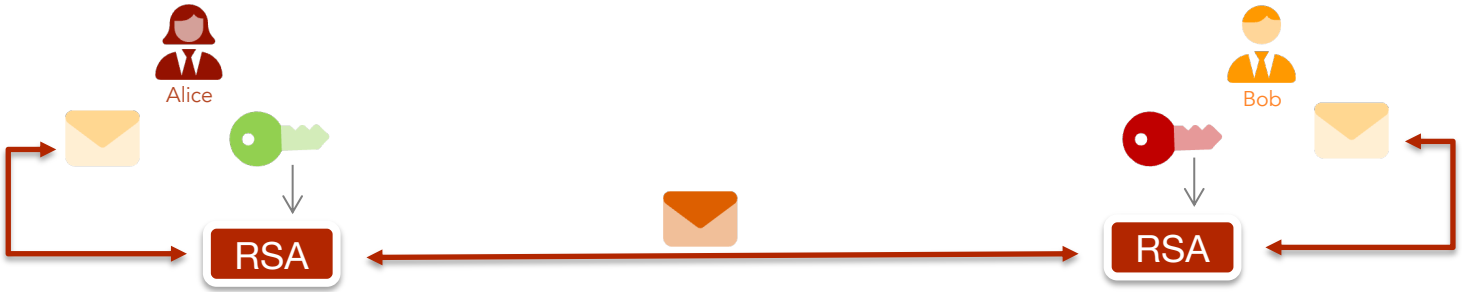
Chiffrement symétrique

le partage et le maintien confidentiel de la clé est une difficulté

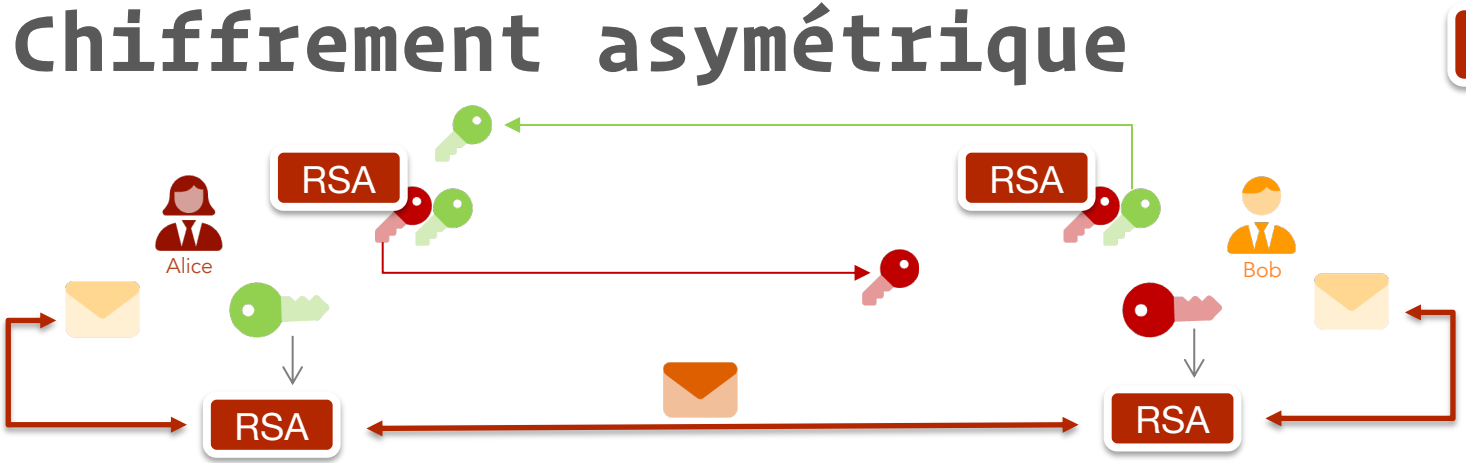


Chiffrement asymétrique

bi-clé
issu du
même
algorithme



Chiffrement asymétrique



bi-clé issu du même algorithme

- génération du bi-clé par celui qui déchiffre
- une des bi-clés est nommée clé privée
- l'autre bi-clé est nommée clé publique

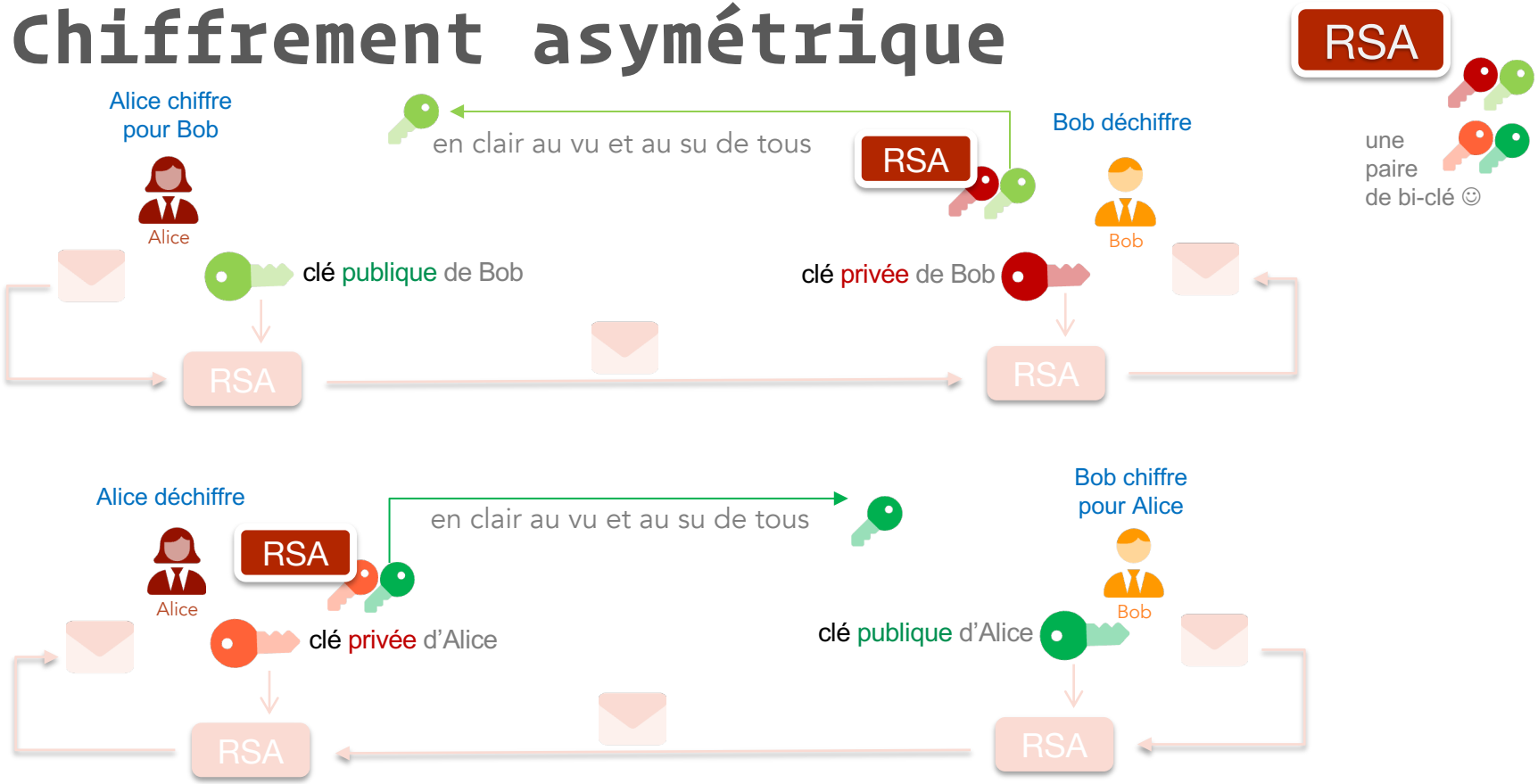
- la clé publique est envoyée en claire au vu et au su de tous

Chiffrement asymétrique

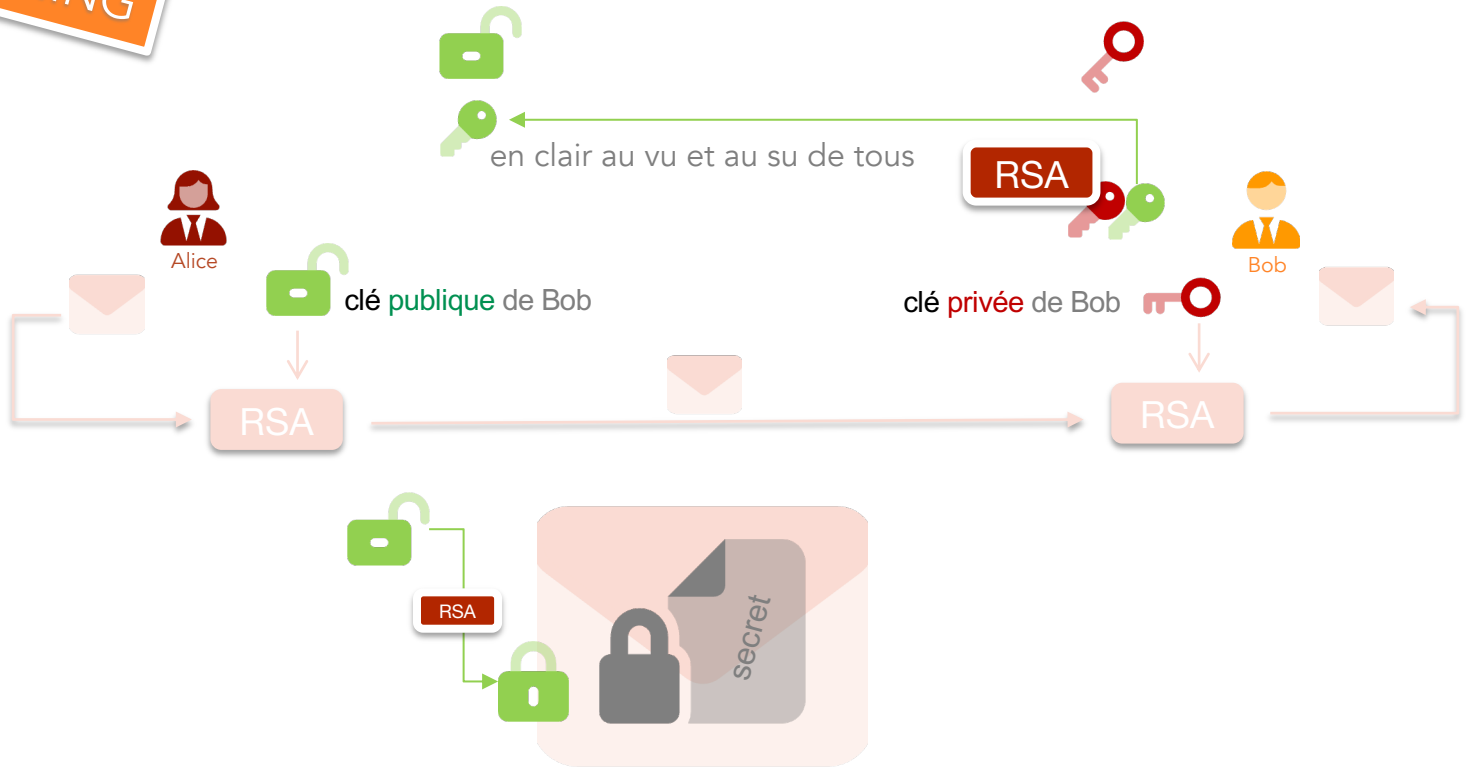
- génération du bi-clé par celui qui déchiffre
- une des bi-clé est nommée clé privée
- l'autre bi-clé est nommée clé publique

- la clé publique est envoyée en clair au vu et au su de tous
- seule la clé publique permet de chiffrer
- seule la clé privée permet de déchiffrer

Chiffrement asymétrique

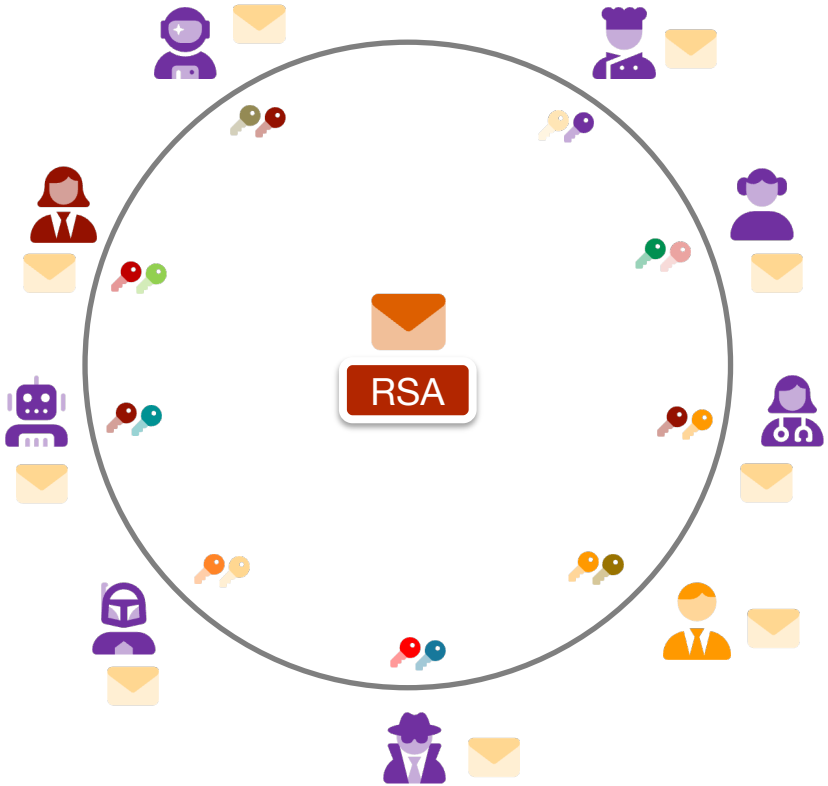


Analogie Chiffrement asymétrique



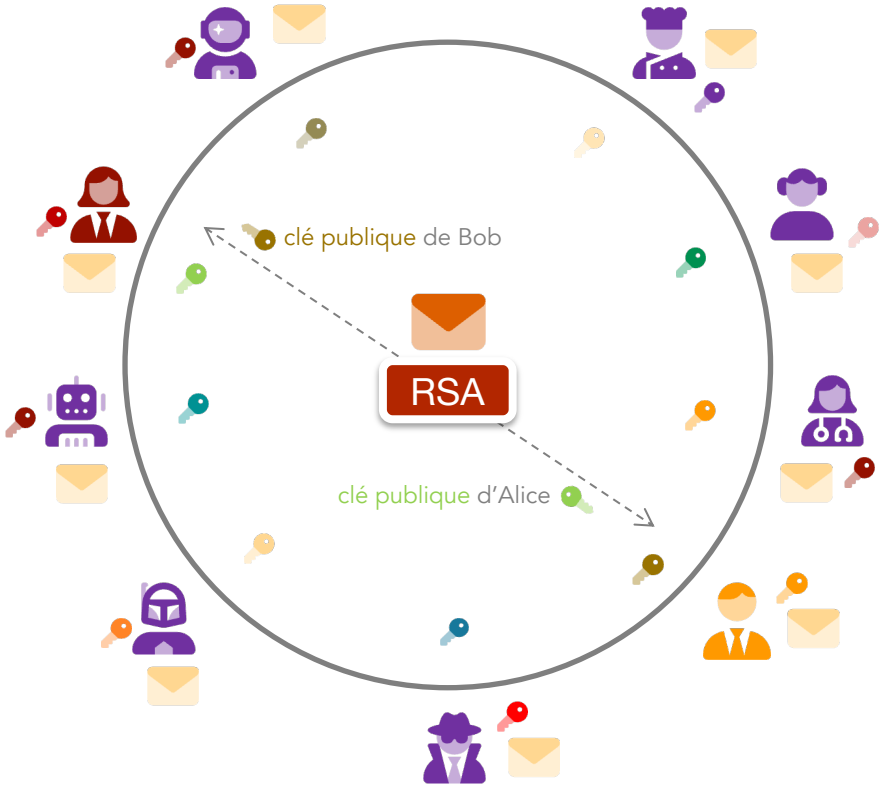
Chiffrement asymétrique

des pairs de clé avec des clés publiques partagées et diffusées



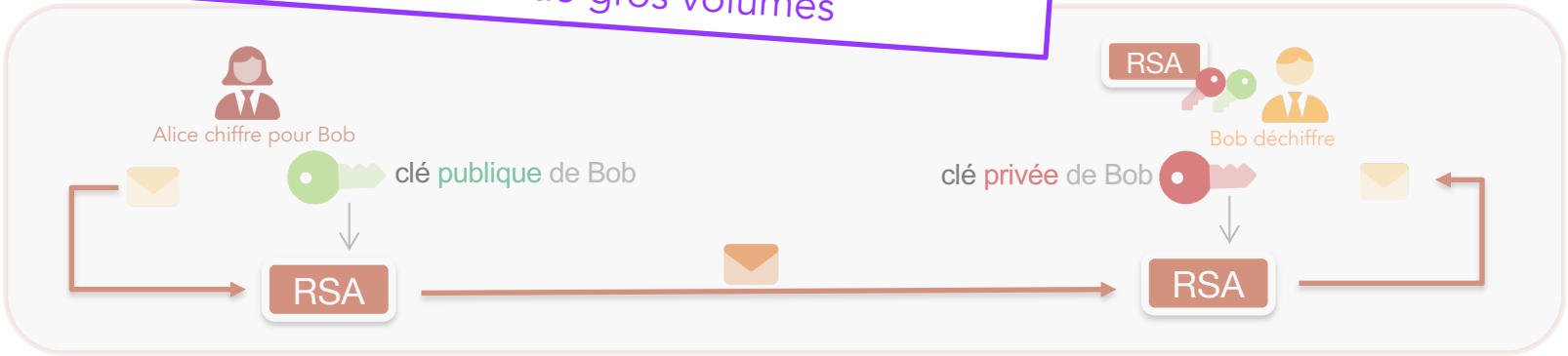
Chiffrement asymétrique

des pairs de clé avec des clés publiques partagées et diffusées



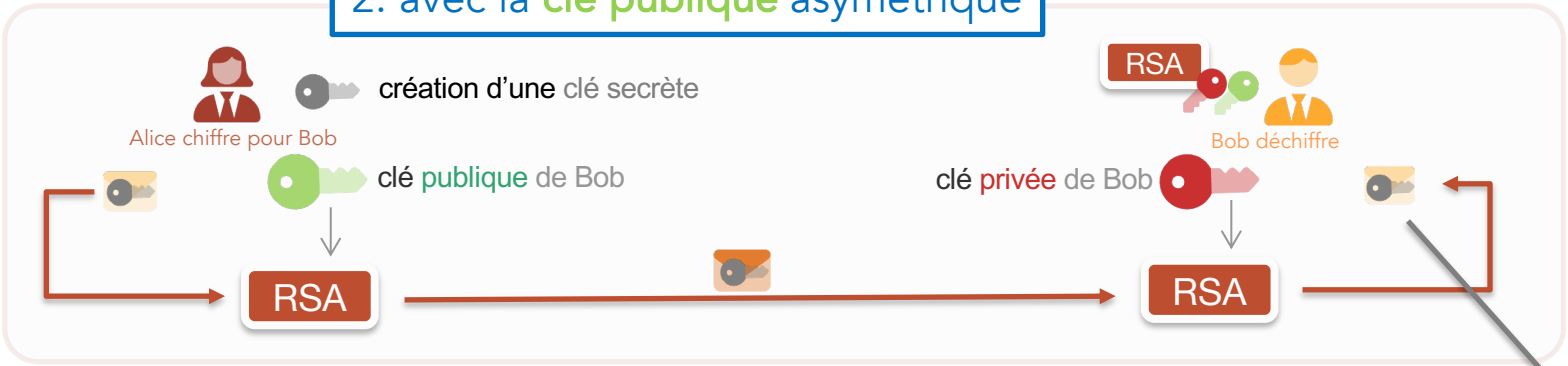
Chiffrement asymétrique

1. Beaucoup plus lent que la crypto. symétrique
2. Difficile de chiffrer de gros volumes



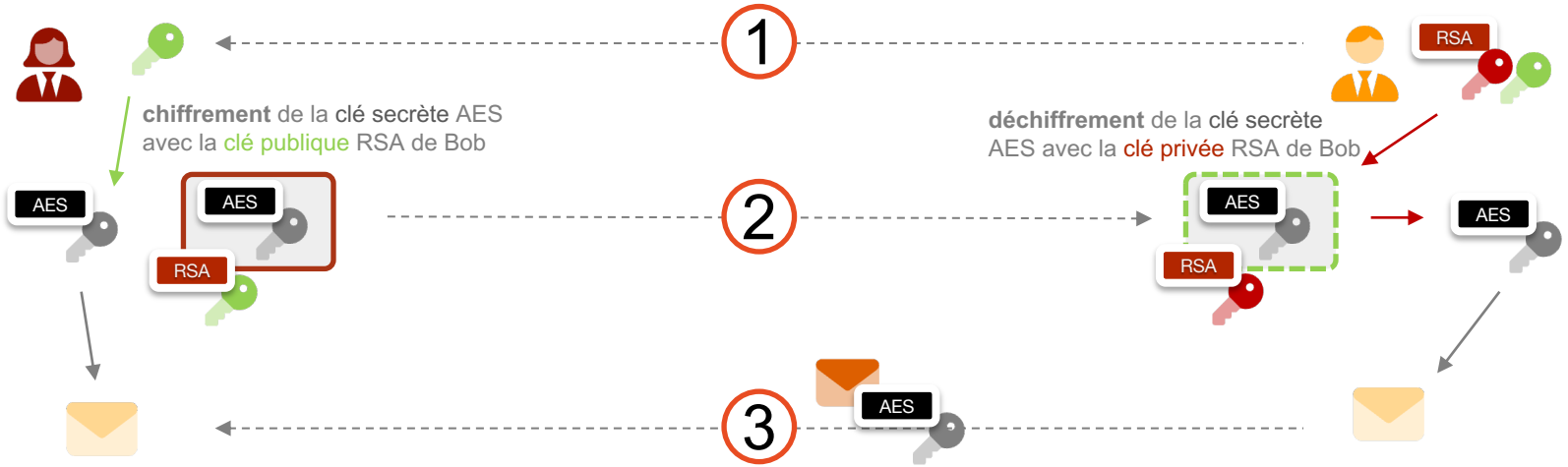
Chiffrement réel

1. Chiffrement de la clé symétrique
2. avec la clé publique asymétrique



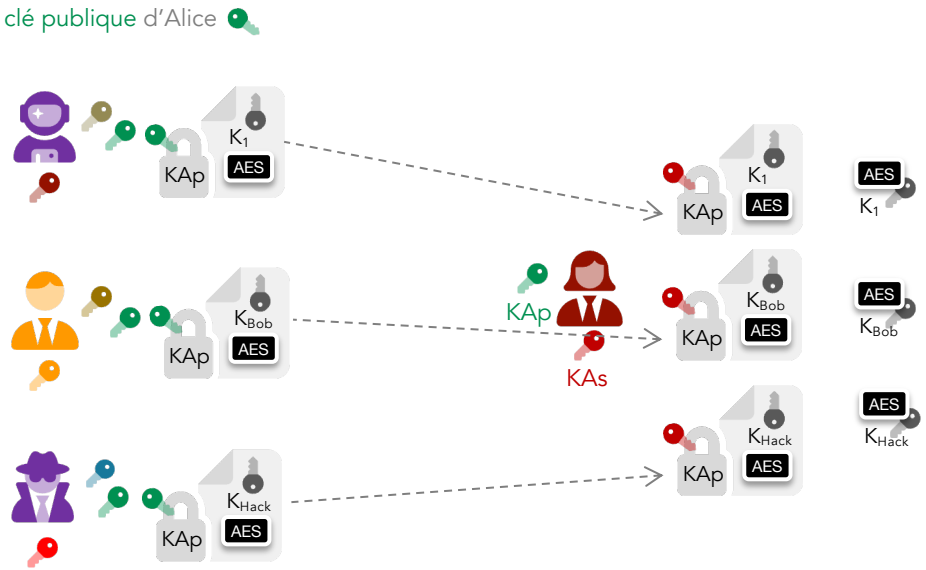
Chiffrement robuste

1. Chiffrement de la clé symétrique
2. avec la clé publique asymétrique

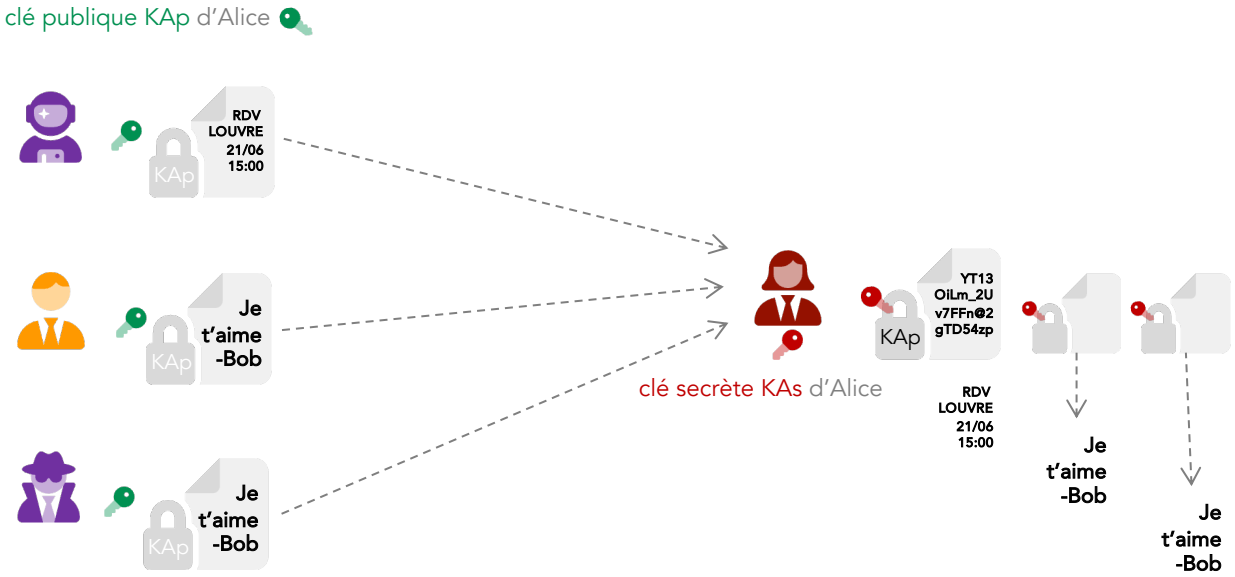


chiffrement et déchiffrement des messages avec la clé secrète AES échangée « secrètement »

Usage: chiffrement pour la Confidentialité

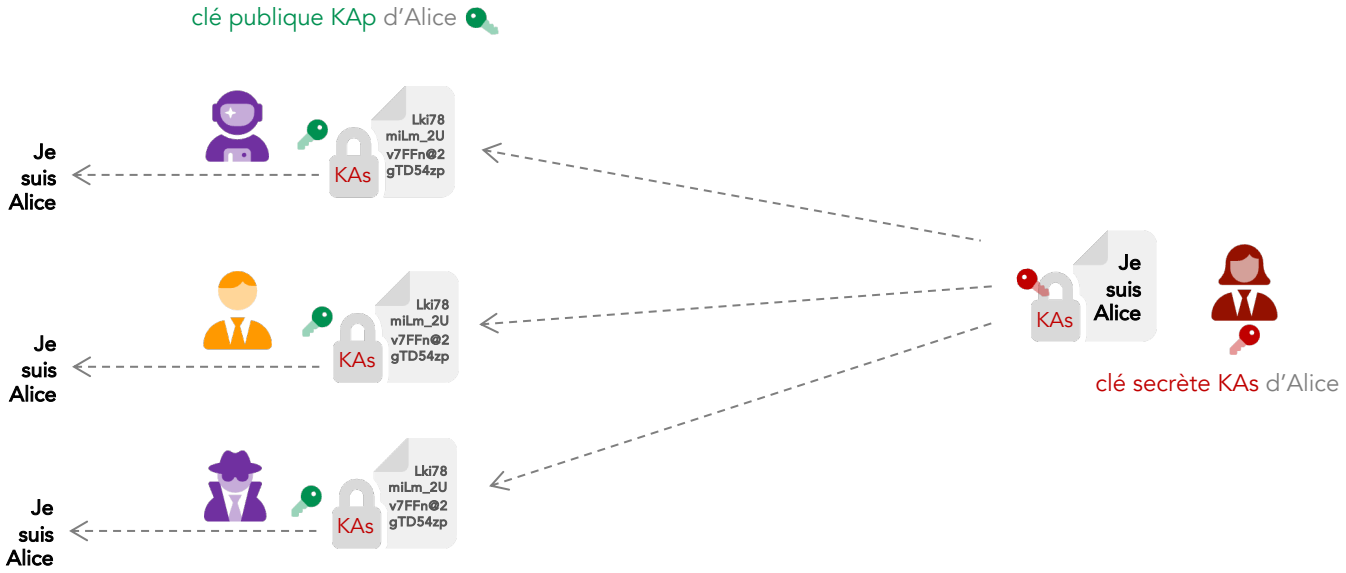


Usage: Chiffrement/Déchiffrement pour confidentialité

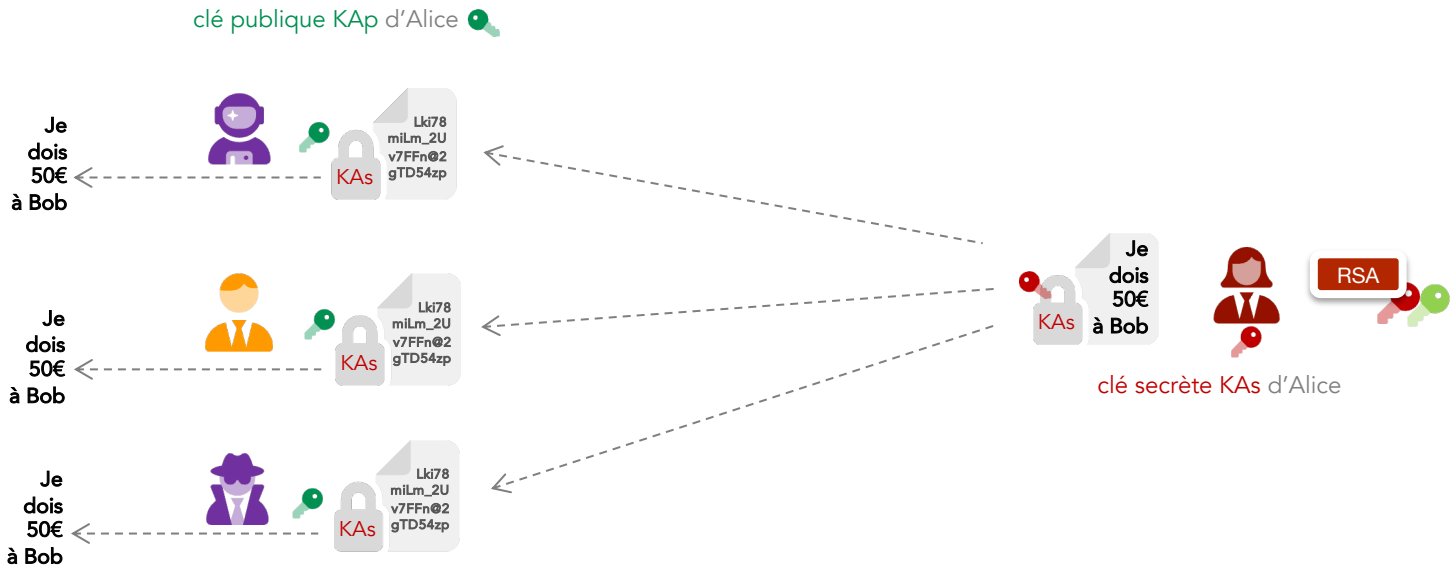


Usage: ? / ? pour ?

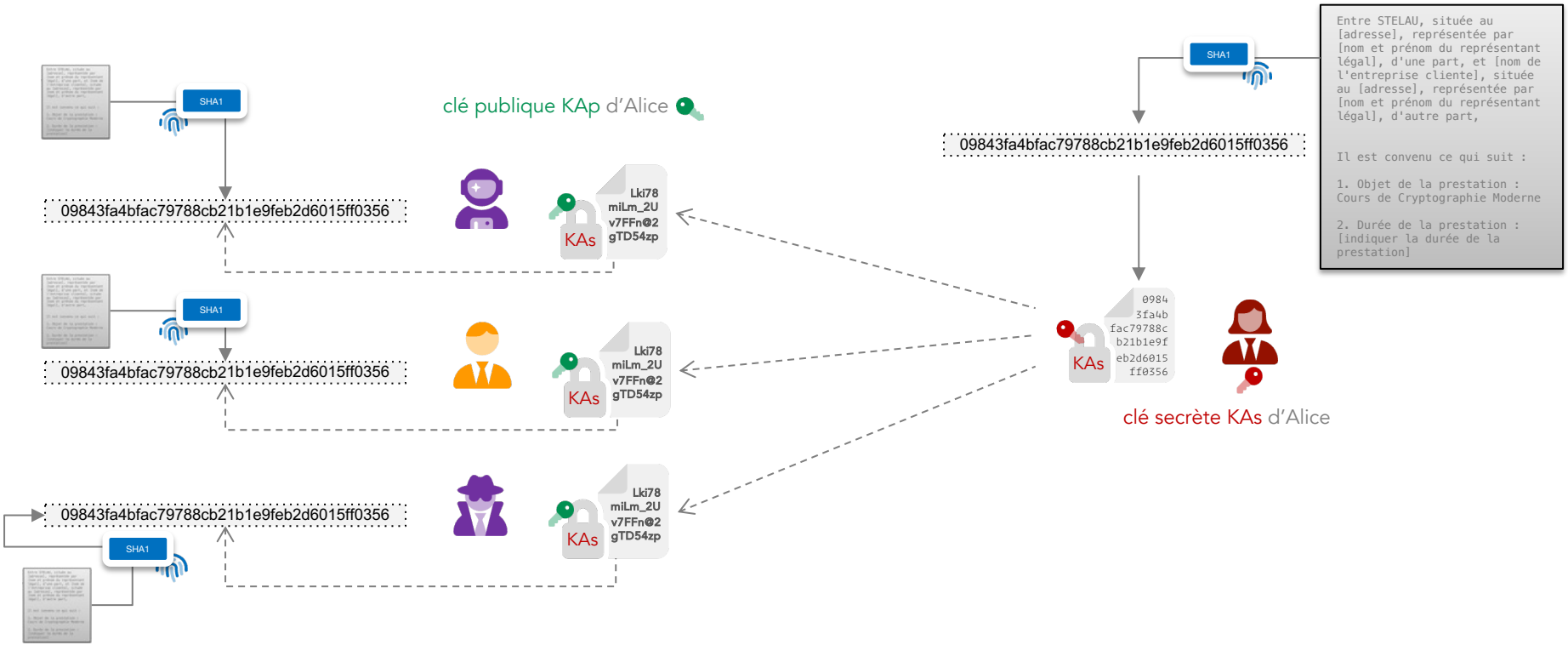
RSA



Usage: Signature / Vérification pour non-répudiation



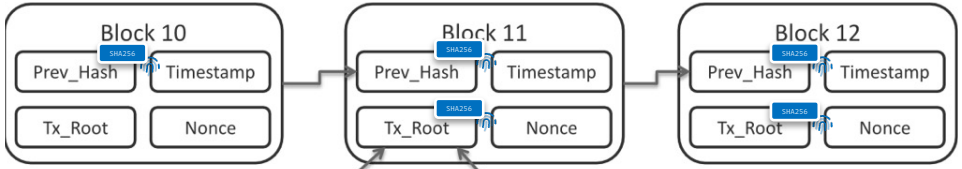
Usage: Signature / Vérification pour non-répudiation

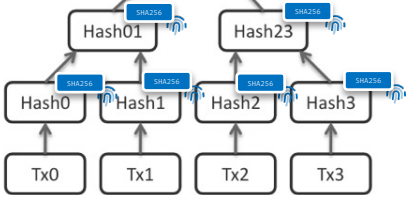


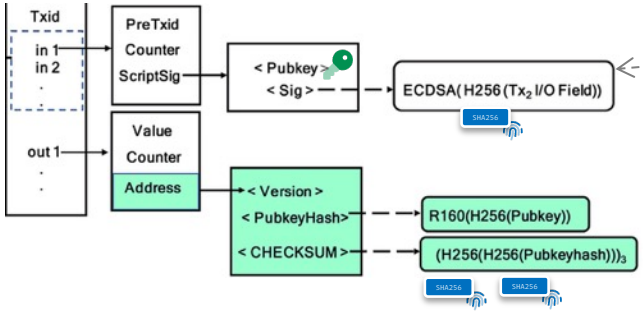
Deux usages différents

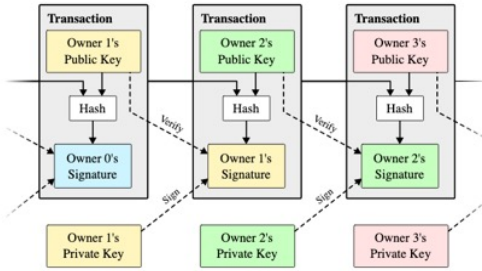
	clé privée	clé publique
chiffrement	déchiffrer	chiffrer
signature	signer	vérifier

Bitcoin : hashs et signature

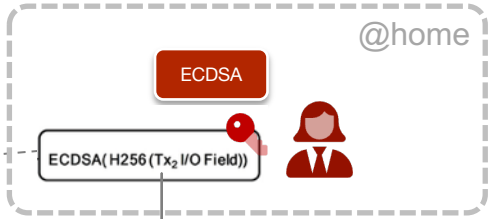








La clé privée de l'expéditeur est utilisée pour signer le hachage SHA-256 de toutes les données de la transaction, y compris les entrées, les sorties et les montants.



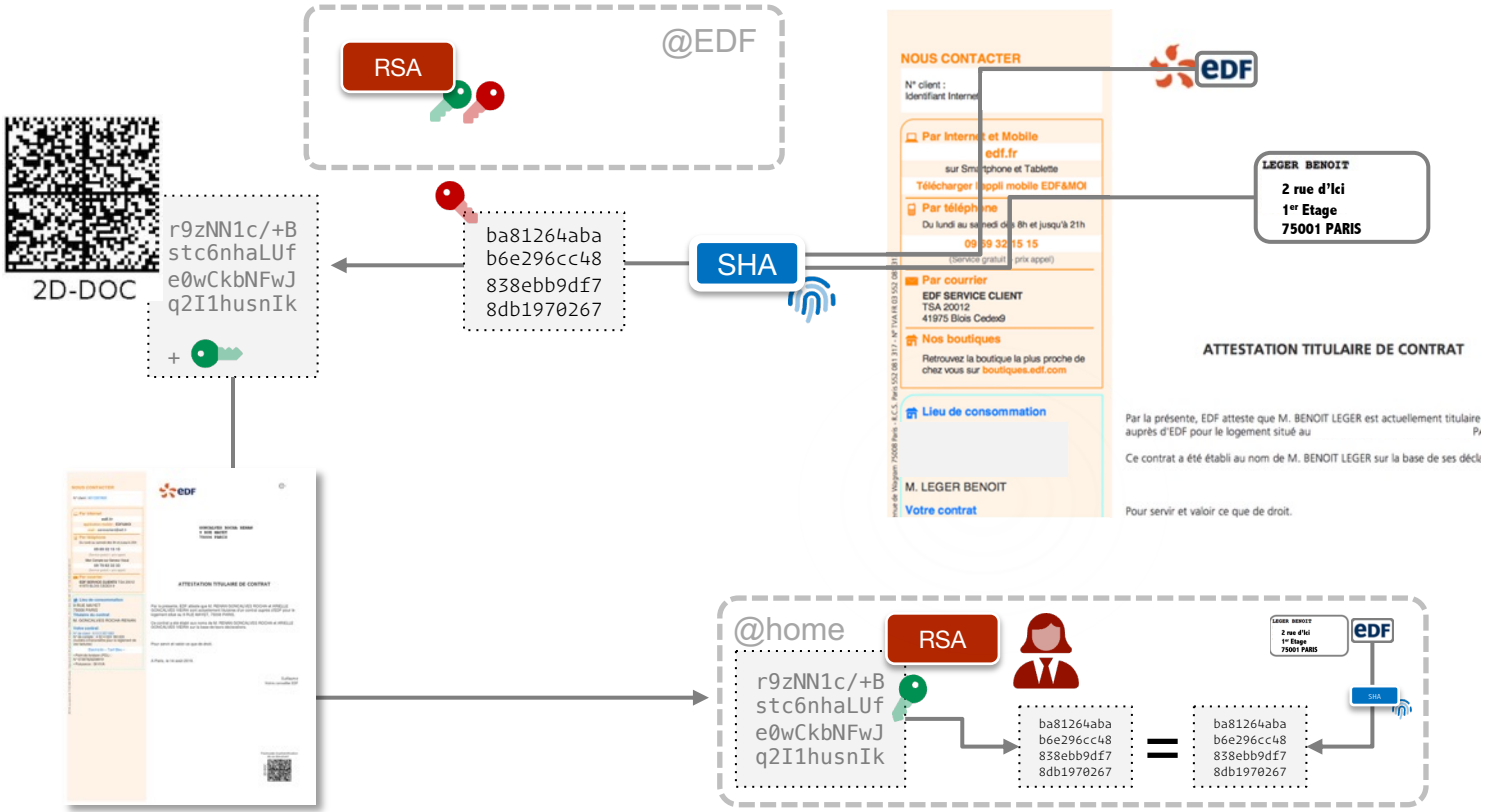
T_{x2}

clé publique d'Alice possède 2฿

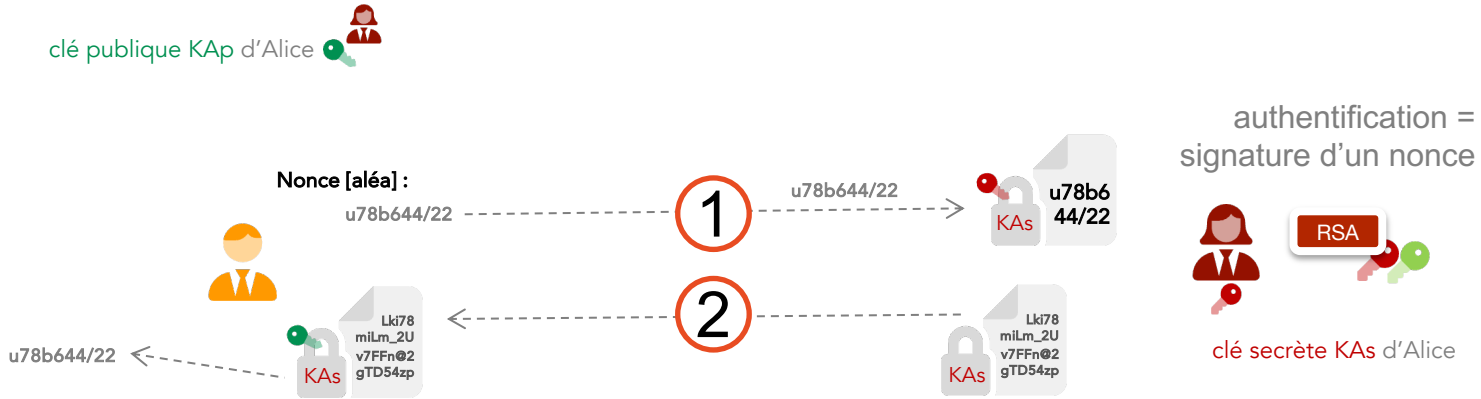
clé publique de Bob reçoit 1฿

Un Tx (ou transaction) en Bitcoin est une opération dans laquelle une ou plusieurs entrées de Bitcoin sont utilisées pour créer une ou plusieurs sorties de Bitcoin. Les entrées de Bitcoin sont des sorties de transactions antérieures qui ont été envoyées à l'adresse Bitcoin du destinataire et qui sont maintenant disponibles pour être dépensées. Les sorties de Bitcoin sont des montants de Bitcoin qui sont envoyés à une adresse Bitcoin.

CEV : Cachet Electronique Visible



Usage: Authentification



~ 100% des authentifications
sont établies sur une signature

Trois usages différents

	clé privée	clé publique
chiffrement	déchiffrer	chiffrer
signature	signer	vérifier
authentification	signer	vérifier

Les 5 primitives cryptographiques

Chiffrement Symétrique

Chiffrement Asymétrique

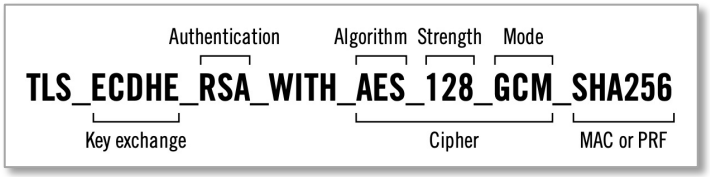
Résout la difficulté de **l'échange** de clé
+
Permet l'usage du principe de **Signature** et d'**Authentification**

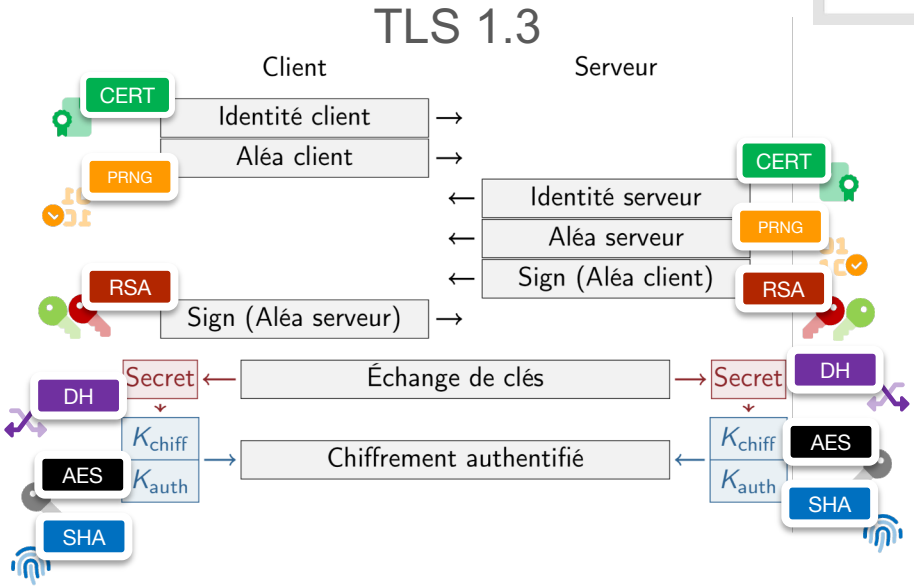
Fonctions de hachage

Établissement de clé

Générateurs d'aléa

Assemblage Crypto



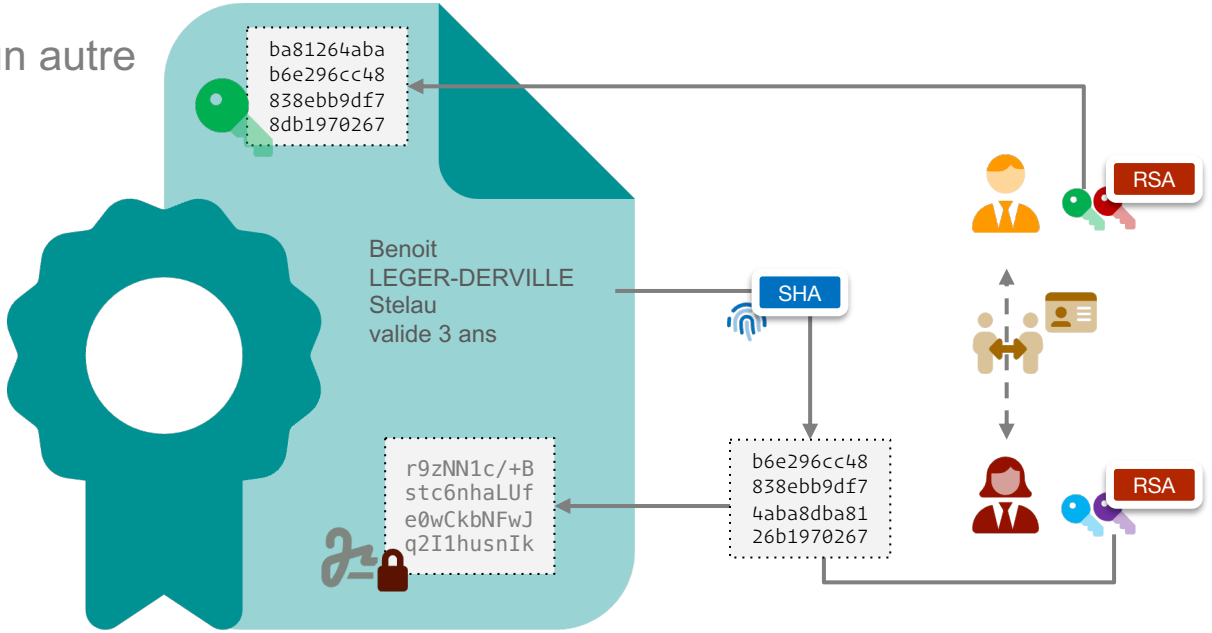


Cipher Suite Name	Auth	KX	Cipher	MAC	PRF
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	RSA	ECDHE	AES-128-GCM	-	SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDSA	ECDHE	AES-256-GCM	-	SHA384
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	RSA	DHE	3DES-EDE-CBC	SHA1	Protocol
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	RSA	AES-128-CBC	SHA1	Protocol
TLS_ECDHE_ECDSA_WITH_AES_128_CCM	ECDSA	ECDHE	AES-128-CCM	-	SHA256

Certificat électronique

Ce n'est pas une primitive cryptographique
C'est un assemblage
C'est un fichier comme un autre

1. identité
2. clé publique
3. signature de l'identité par un tiers



Very Short Crypto Story

3000 ans de crypto. symétrique

recettes militaro-diplomatiques de confusion et de diffusion

Confusion et **Diffusion**
« *tant bien que mal* »
de César à Enigma

100 ans de crypto. moderne

de Kerckhoffs ...
au crypto-système incassable

1. Principes de Kerckhoffs - 1883
2. One Time Pad - 1917

---

50 ans de crypto. asymétrique

LA véritable révolution

Résout la difficulté de **l'échange** de clé
+
Permet l'usage du principe de **Signature**

20 ans de crypto. quantique

révolution ? (ou pas)

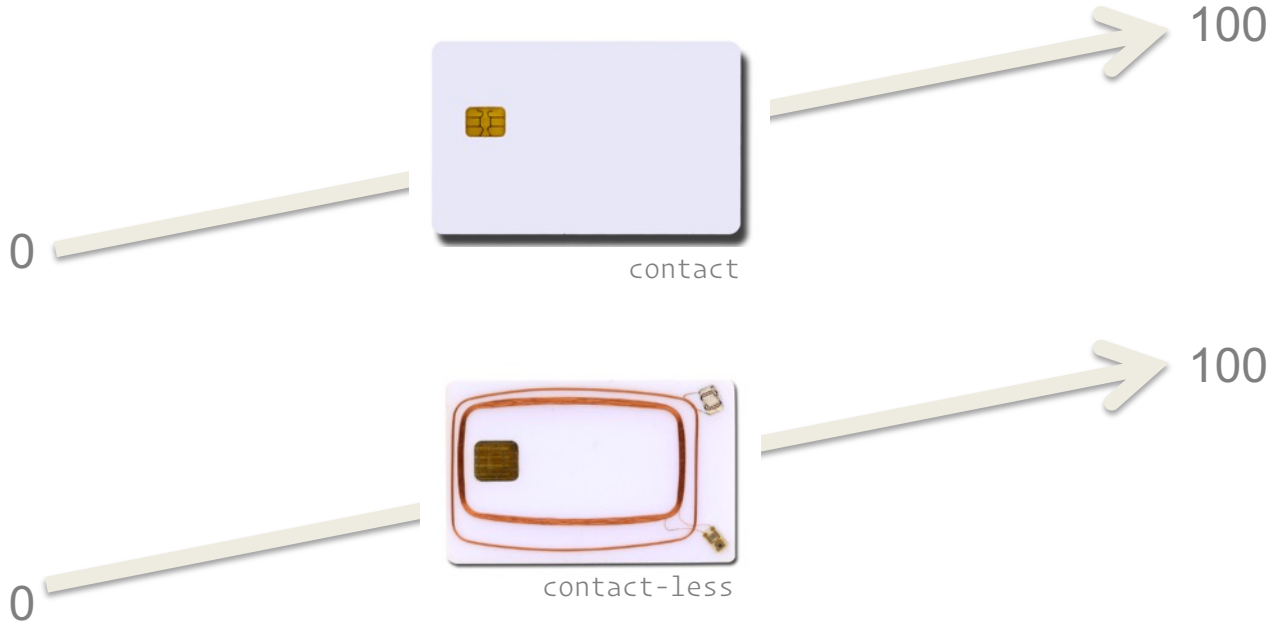
Les vrais difficultés de la cryptographie moderne

1. THEORIE : Cryptologie
   - failles théoriques = mathématiques
   - cryptologue est un métier

2. CODE : Implémentations
   - erreurs/failles/vuln = informatique
   - « Do not implement cryptography yourself ! »

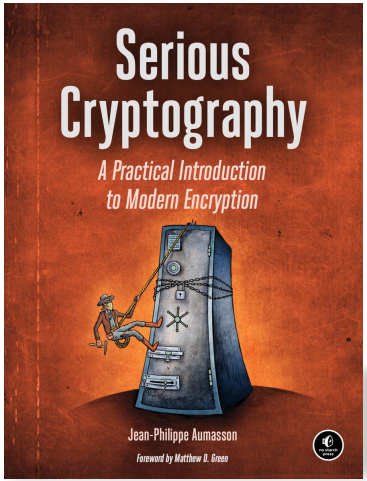
3. UX : Usages
   - mauvais usages = ignorance/pusillanimité
   - bons usages = formation

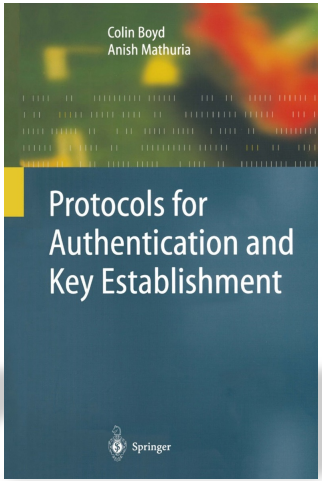
Attention : « C’est (*pas*) sécurisé ! » ne veut rien dire

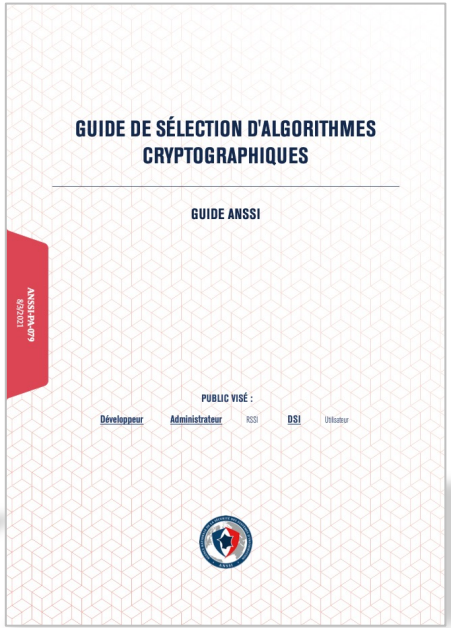


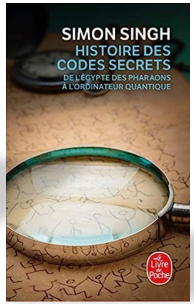
Fin

Ouvrages Cryptographiques







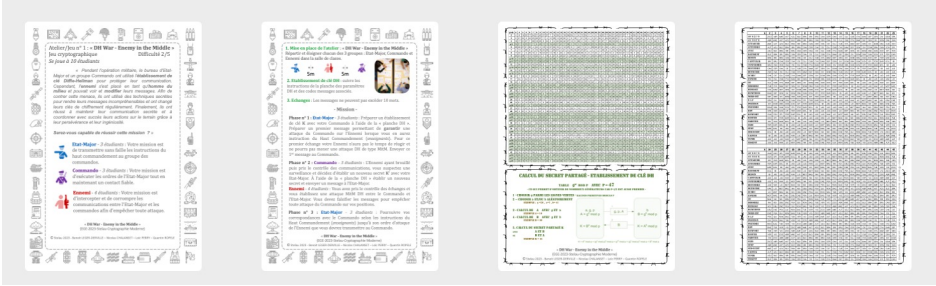


Ateliers/jeux Cryptographie Moderne

RSA

DH

Atelier 1 : DH War- Ennemy in the Middle



4 groupes
de 10 étudiants

Atelier 2 : Wannacry Nightmare Puzzle

4 groupes
de 5 étudiants

Hello !

le secret

M

C

M

Hello !

26, 47, 10

91690410bec9

graine

798

aléa