LA CRYPTO POUR LES DEVS

Késako?

➤ http://talks.m4dz.net/crypto-pour-les-devs/

M4DZ

Paranoïd Web Dino & Tech Evangelist

m4dz.net | @m4d_z | PGP 0xD4627C417D969710

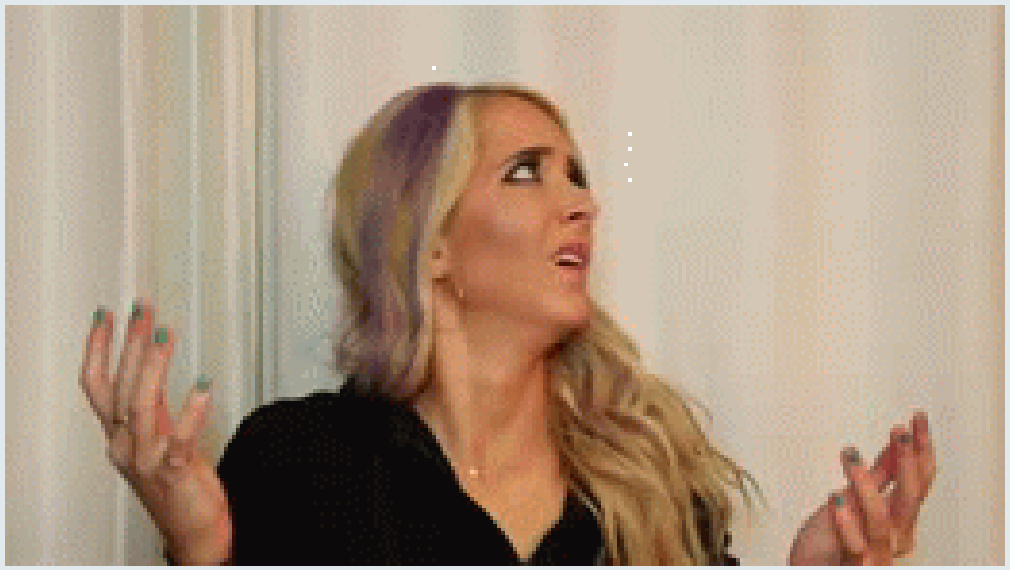
www.alwaysdata.com

La responsabilité du développeur

La limite des mots de passe : l'interface CC

- The Scary Truth About Your Passwords - Lastpass blog - 2014
- Worst passwords of 2017, Top 100 - Splashdata - 2017

m'en fous, on n'héberge pas de données sensibles



Une seule solution

Chiffrer

(ou crypter, troll detected)

- guerre de l'information
- tracking et recroisement
- identité numérique

AUCUNE DONNÉE SENSIBLE NE DEVRAIT CIRCULER OU ÊTRE STOCKÉE EN CLAIR

CHIFFRER ?

Chiffrement

Plus question de reculer

NSA / Prism / Loi Renseignement vs GPDR / ePrivacy

La crypto, ça n'est pas…

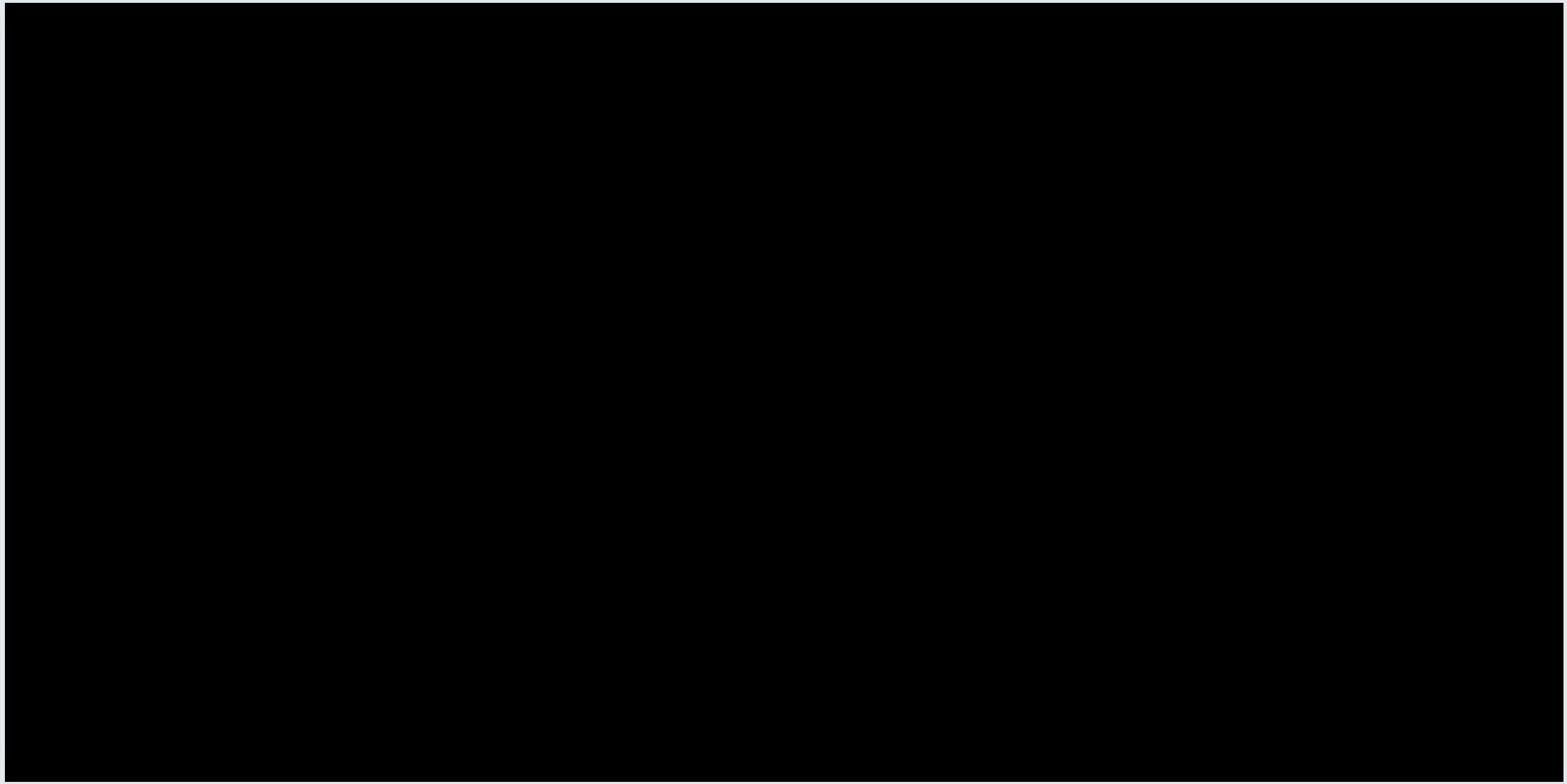
- l'authentification
- la sécurité
- la révocation

Objectif :
Protéger des informations sensibles

La crypto, c'est :

- Hash
- Encryption
- Échange de clés
- Signature

Seule la clé est importante



CRYPTOGRAPHIE & CRYPTANALYSE

Il était une fois…

Le Code César

Le chiffre de Vigenère

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

```
Plaintext:  ATTACKATDAWN
Key:        LEMONLEMONLE
Ciphertext: LXFOPVEFRNHR
```

LA FAILLE : LES RÉPÉTITIONS

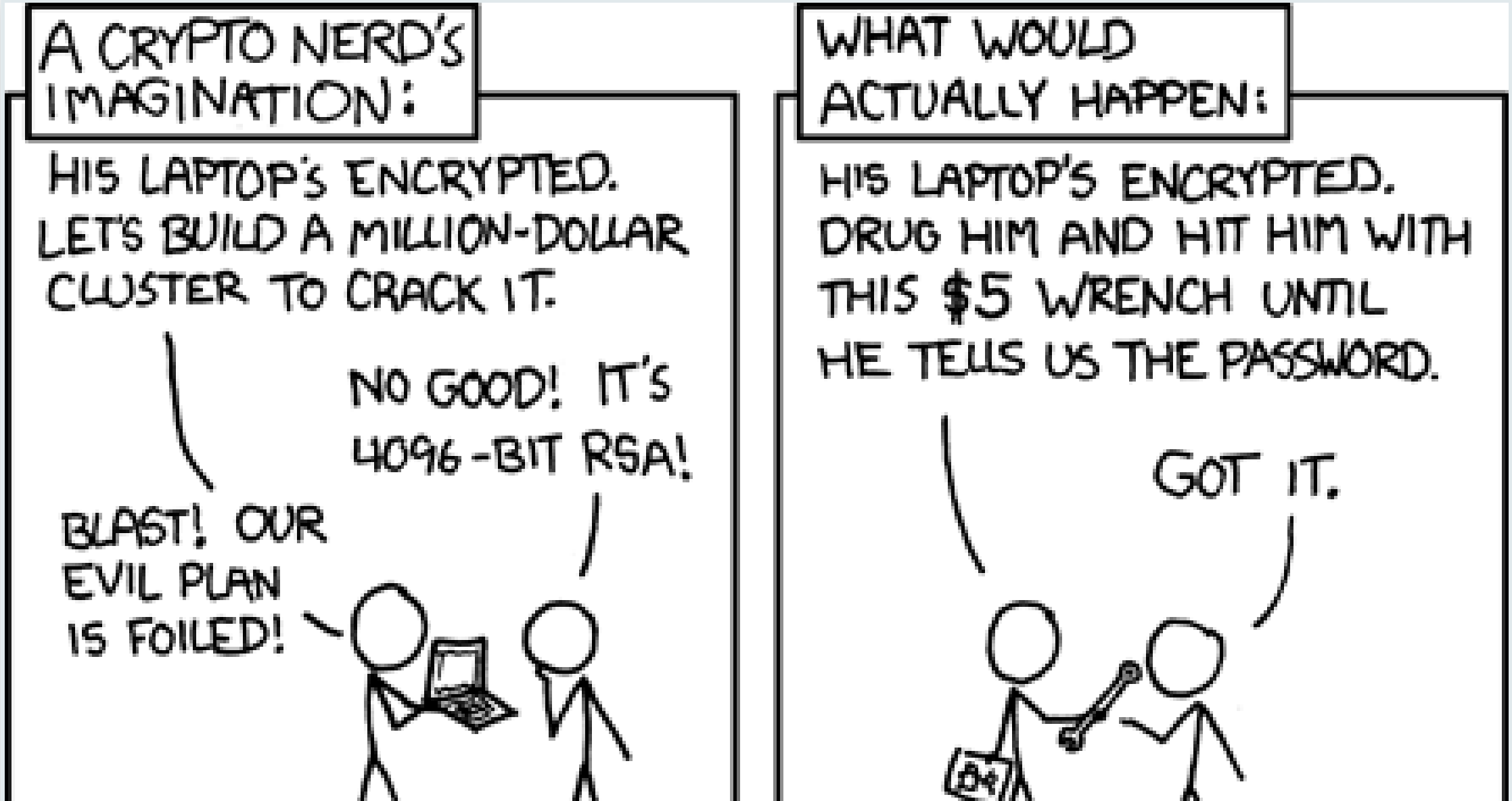
L'analyse des fréquences rend caduques toutes protections qui utiliseraient un dénominateur commun

Enigma

LA FAILLE : L'ESPIONNAGE ET L'ATTAQUE PAR FORCE BRUTE

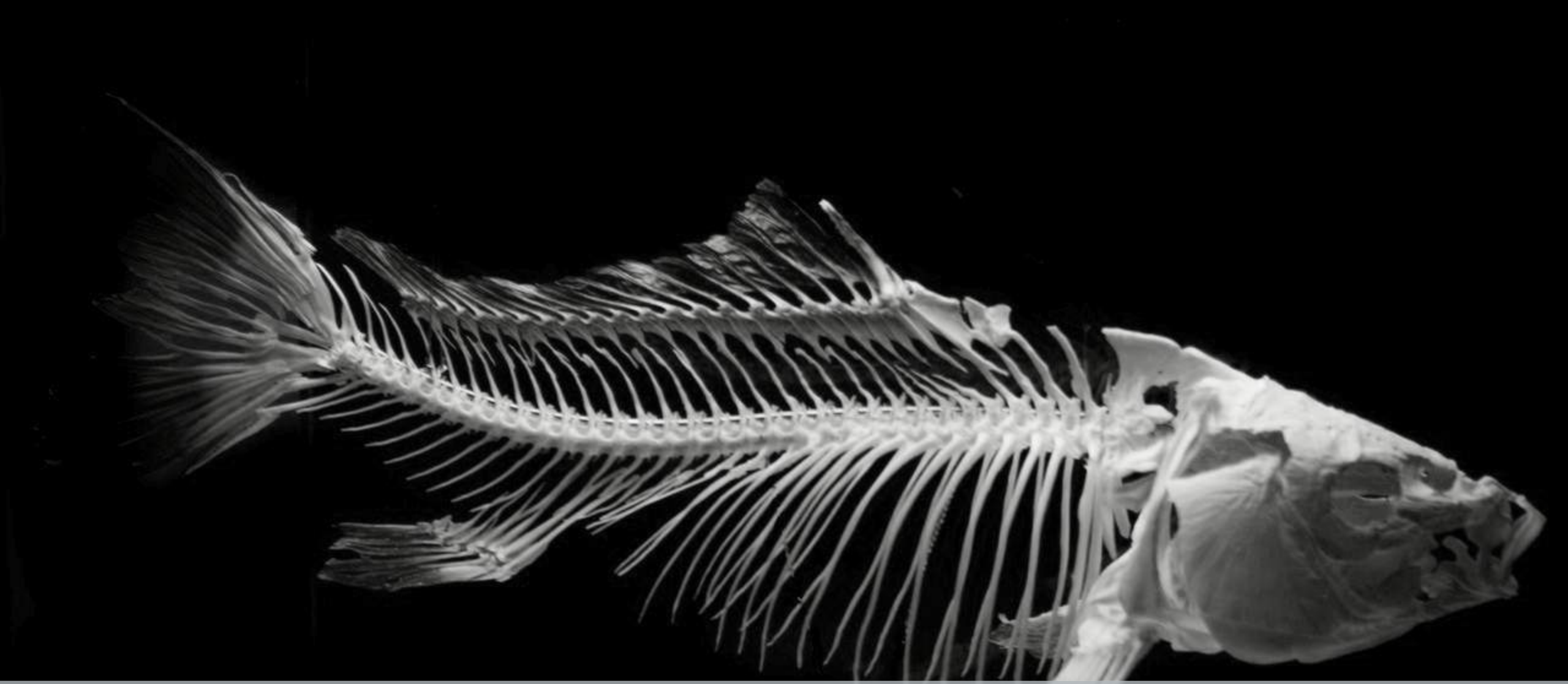
Aucun système ne peut être suffisamment robuste pour résister éternellement à une attaque

La protection des clés est essentielle



Le chiffrement numérique :

TOKEN UNIQUE



HACHAGE, SALAGE & ENTROPIE

Hachage : obfuscation des données 👍

PROBLÈME

les rainbow / lookup / reverse-lookup tables

Saler

- ajoute de l'entropie
- supprime les risques de répétition
- doit être **unique** et **aléatoire**

On ne fait pas…

```
md5(sha1(password))
md5(md5(salt) + md5(password))
sha1(sha1(password))
sha1(str_rot13(password + salt))
md5(sha1(md5(md5(password) + sha1(password)) + md5(password)))
```

On fait :

- pseudo-aléatoire CSPRNG : le salt
- dérivation PBKDF2 (SHA256) / Bcrypt / Scrypt sur `[salt+pasword]` (+ entropie)
- stockage du résultat et des paramètres

Salted Password Hashing - Doing it Right

UN HACHAGE SANS RÉPÉTITION ET EN EXÉCUTION LENTE LIMITE SA SURFACE D'ATTAQUE

SYMÉTRIQUE VS ASYMÉTRIQUE

Chiffrement par bloc

- ~~DES (Data Encryption Standard)~~
- AES (Advanced Encryption Standard)
- IDEA
- BlowFish

Chiffrement par flux (Stream Cipher)

- ~~RC4~~
- ChaCha20 ?
- Panama ?

PROBLÈME

Les machines ne sont pas aléatoires

- besoin de données imprévisibles
- méthodes crypto CSPRNG

  (pas `/dev/urandom` directement, utilisez les méthodes des libs crypto)
- IV (Vecteur d'Initialisation)

  (bytes-block utilisés en initialisation d'un algo de chiffrement pour assurer son caractère unique)

Padding, Random, IV

- ~~ECB (Electronic Code Book)~~
- ~~CBC (Cipher Block Chaining)~~
- AEAD (Authenticated Encryption with Associated Data)

https://blog.cloudflare.com/padding-oracles-and-the-decline-of-cbc-mode-ciphersuites/

PROBLÈME

Une clé peut être compromise : une clé symétrique doit nécessairement circuler

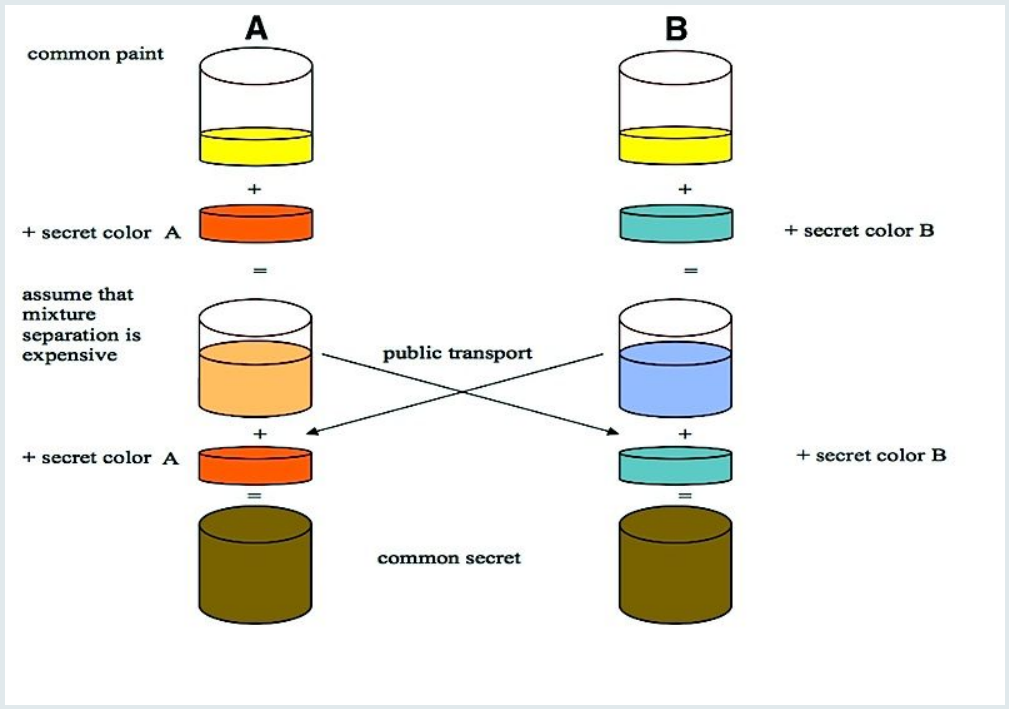
Bob & Alice échangent leurs clés

Clés, certificats, signatures & chiffrement

Clé symétrique

- clé unique pour toutes les opérations
- rapide
- sensible sur la clé

Diffie-Hellman



- clé publique commune
- secret partagé

PGP / GnuPG

- clés asymétriques (RSA) sur clé symétrique (~~IDEA~~ AES)
- chiffre (clé publique) et signe (clé privée)
- utilise l'entropie fournie par l'utilisateur

PGP / GnuPG

- clés asymétriques (RSA) sur clé symétrique (~~IDEA~~ AES)
- chiffre (clé publique) et signe (clé privée)
- utilise l'entropie fournie par l'utilisateur
- **la Crypto pour tous** (https://ssd.eff.org/fr)

Signature

- asymétrique inversée
- pas de sécurisation

Signature

- asymétrique inversée
- pas de sécurisation
- **identification**

Certificats

- authentifie un client auprès d'un tiers de confiance
- assure la révocation

Certificats

- authentifie un client auprès d'un tiers de confiance
- assure la révocation
- 🔗 **Let's Encrypt**

Les standards

- X.509
- PKCS
- PCIDSS


PROTÉGER

Le réseau

- ~~SSL~~ / TLS
- Confidentialité persistante

Les accès :
Password Hash

Les données

- RSA
- Symétrique encapsulé
- Boitiers HSM

WEBCRYPTO À LA RESCOUSSE (?)

Côté backend

- PyCrypto
- RbNaCl
- Node.js Crypto module
- PHP Mcrypt

Et côté front ?

The WG Spec

before reading

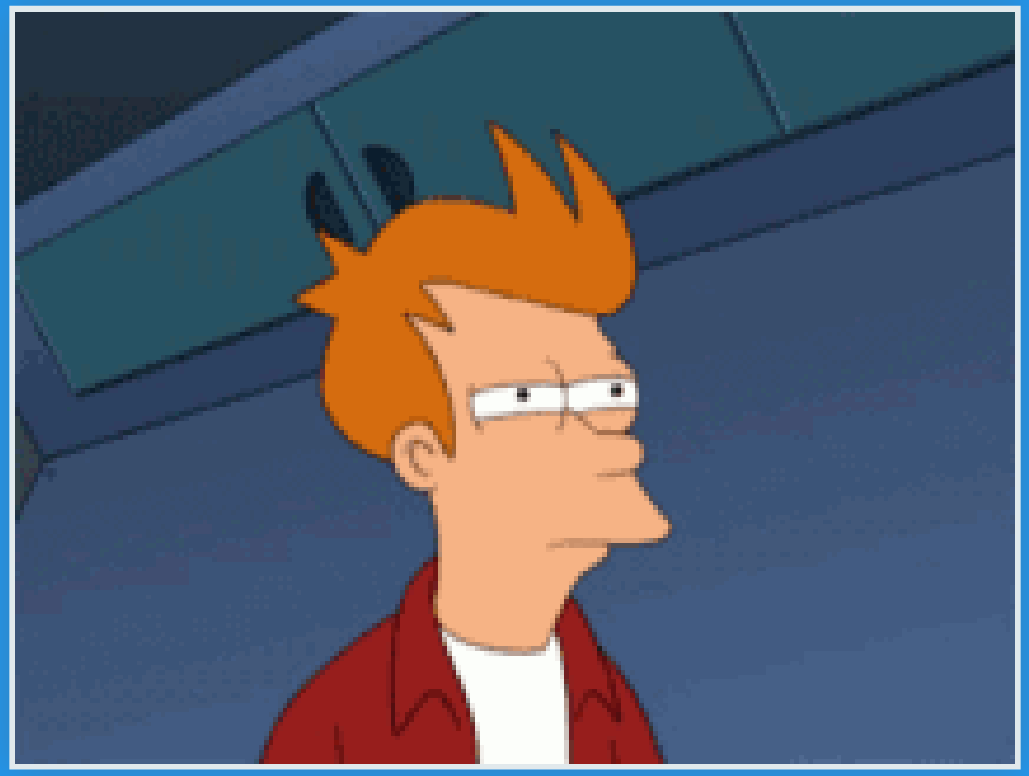
The WG Spec

after reading

Current Status

- Recommandation (26 janvier 2017)
- Spec obscure pour les néophytes

```
window.crypto.subtle
```



```
window.crypto.subtle.encrypt(/* ... */)
.then(function(encrypted){
  //returns an ArrayBuffer containing the encrypted data
  console.log(new Uint8Array(encrypted))
})
.catch(function(err){
  console.error(err)
})
```

WebCrypto API

- n'utilise que des Promises
- ne traite qu'avec des sources binaires (ArrayBuffers)

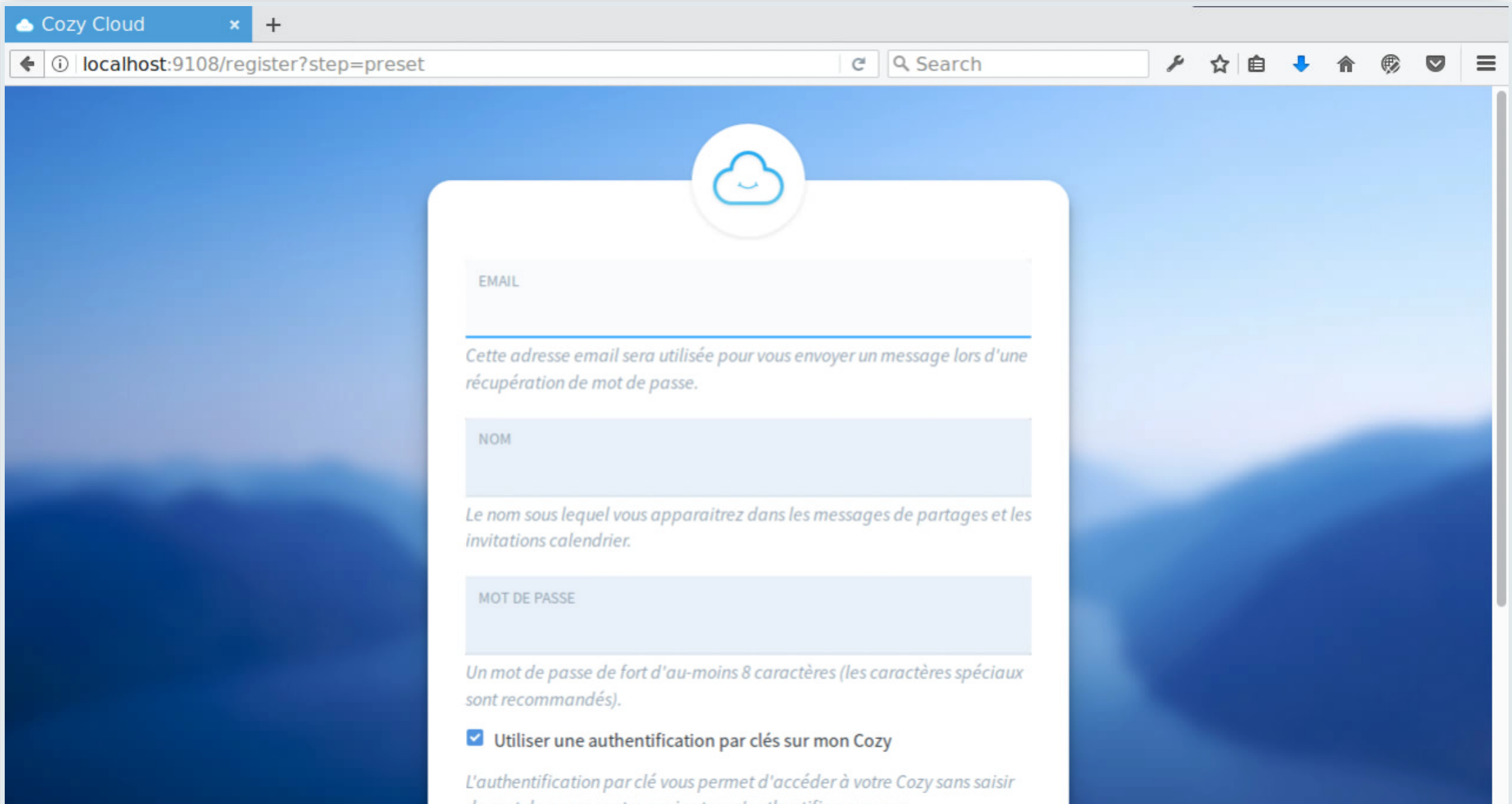
Point Bonus

- RSASSA-PKCS1-v1_5 / RSA-OAEP
- AES-CBC / AES-GCM / AES-KW
- HMAC
- SHA-256 / SHA-384 / SHA-512

les navigateurs n'implémentent que les algos qu'ils estiment nécessaires

Comme avec `canPlayType`

Démo : password less



BROWSER : DES LIBS BASÉES SUR WEBCRYPTOAPI

https://gist.github.com/jo/8619441

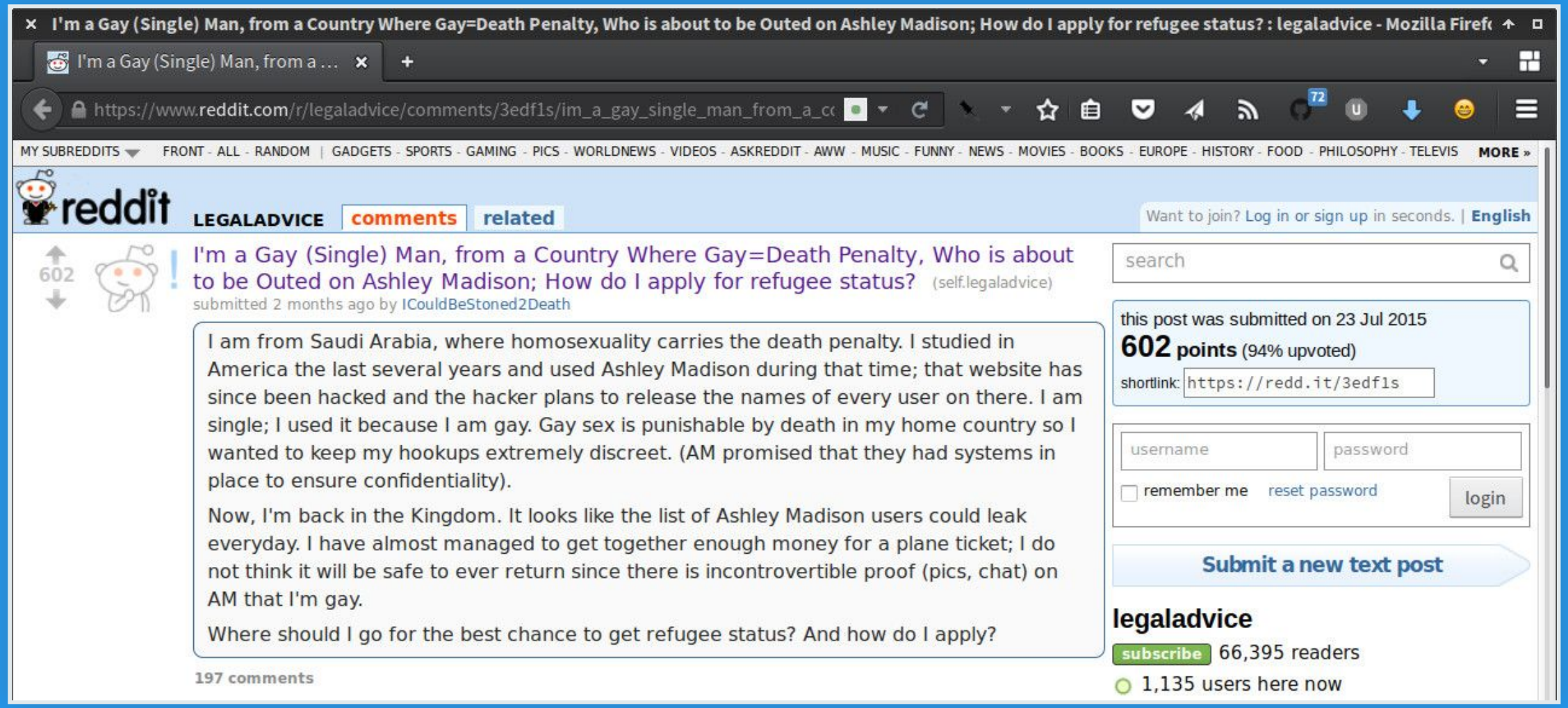
Alors, on fait quoi ?

Ne jouez pas les apprentis sorciers

N'oubliez jamais que :

- la sécurité est inversement proportionelle à la simplicité d'utilisation
- toute sécurité a un coût

On arrête d'avoir peur, et on protège



QUESTIONS ?

➤ **http://talks.m4dz.net/crypto-pour-les-devs/**