ELEMENTS DE CRYPTOGRAPHIE

De l’antiquité à la cryptographie moderne

Avant de commencer 1/2 «Coder l'information»

* On ne sait bien mémoriser que des «bits» (interrupteurs)
  * entiers : écriture binaire

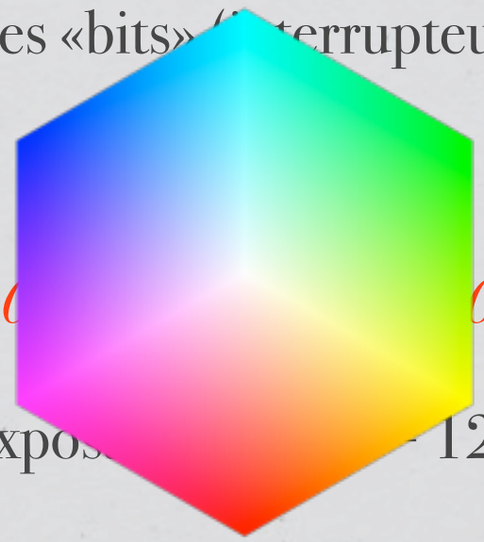
  *ex: 135 = 10000111_2 = 1*128+0*64+0*32+0*16+0*8+1*4+1*2+1*1*

  * réels ➛ flottants (mantisse, exposant) (12.47 = 1247e-4)
  * caractères ➛code ASCII ('A' = 65)
* Et de l'information ?
  * couleur d'un point d'une image: rgb(126,142,135)
  * son : tableau de fréquences entières
* Bilan: du point de vue de la machine, l'information c'est une suite de bits…

Avant de commencer 1/2 «Coder l'information»

* On ne sait bien mémoriser que des «bits» (interrupteurs)
  * entiers : écriture binaire

*ex: 135 = 10000111_2 = 1*128+0*64+0*32+0*16+0*8+1*4+1*2+1*1*

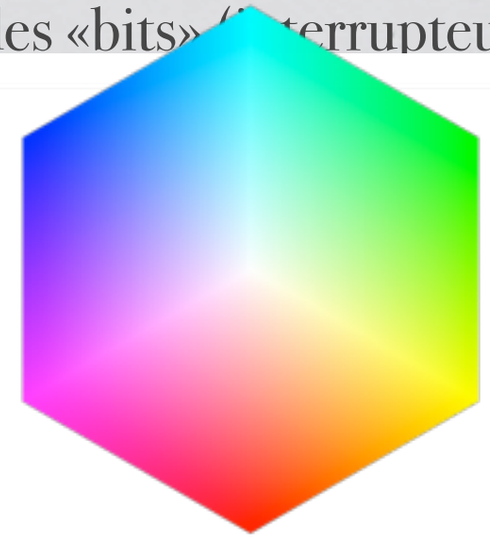


  * réels ➛ flottants (mantisse, exposant) (1247e-4)
  * caractères ➛code ASCII ('A' = 65)
* Et de l'information ?
  * couleur d'un point d'une image: rgb(126,142,135)
  * son : tableau de fréquences entières
* Bilan: du point de vue de la machine, l'information c'est une suite de bits…

Avant de commencer 1/2 «Coder l'information»

* On ne sait bien mémoriser que des «bits» (interrupteurs)

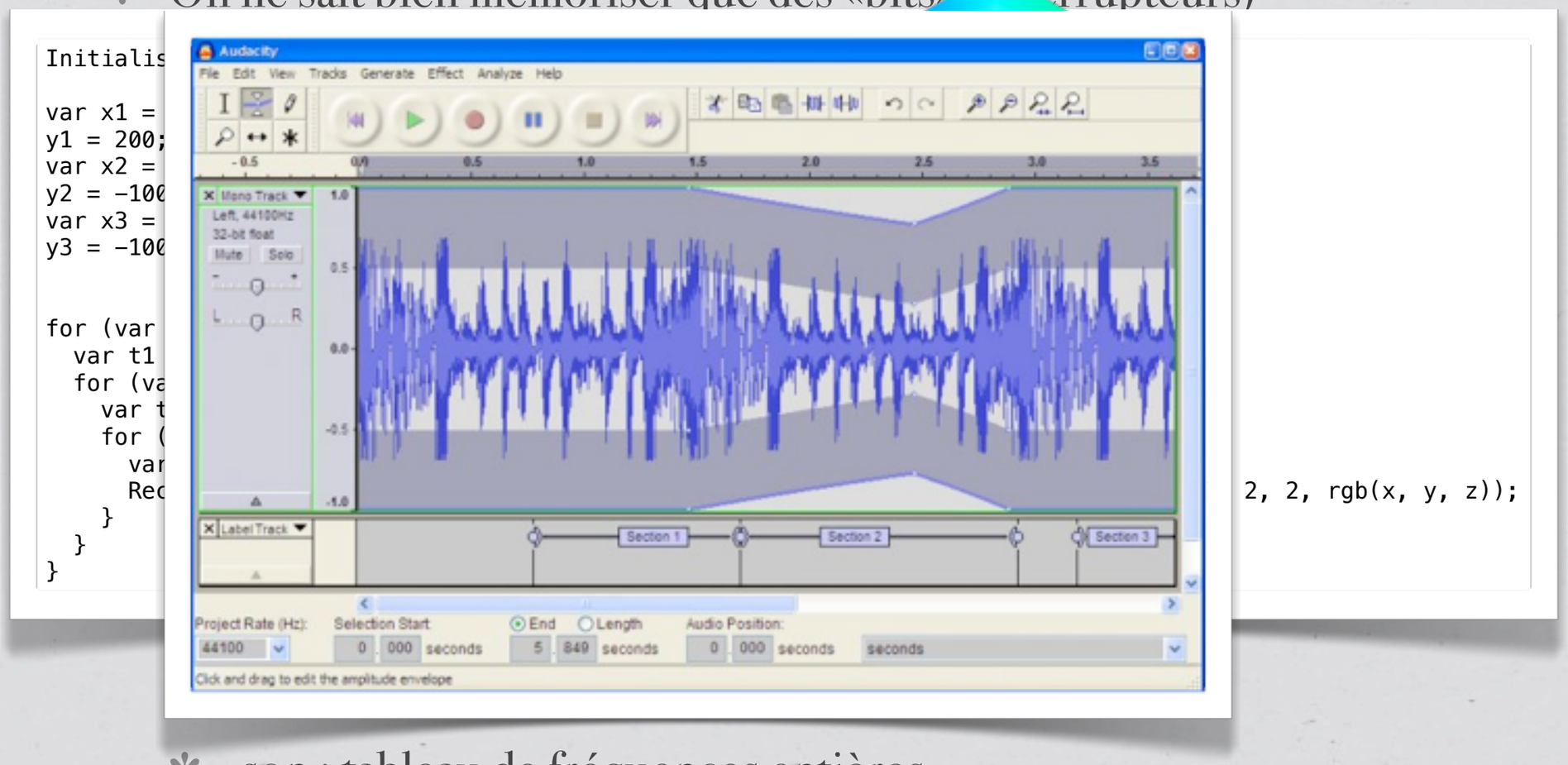
```
Initialiser();

var x1 = 0,
y1 = 200;
var x2 = 200*Math.sqrt(3)/2,
y2 = -100;
var x3 = -x2,
y3 = -100;


for (var x = 0; x < 256; x += 2) {
  var t1 = x / 256;
  for (var y = 0; y < 256; y += 2) {
    var t2 = y / 256;
    for (var z = 0; z < 256; z += 2) {
      var t3 = z / 256;
      RectanglePlein(500+t1 * x1 + t2 * x2 + t3 * x3, 250+t1 * y1 + t2 * y2 + t3 * y3, 2, 2, rgb(x, y, z));
    }
  }
}
```



  * couleur d'un point d'une image: rgb(126,142,135)
  * son : tableau de fréquences entières
* Bilan: du point de vue de la machine, l'information c'est une suite de bits…

Avant de commencer 1/2 «Coder l'information»

* On ne sait bien mémoriser que des «bits» (interrupteurs)

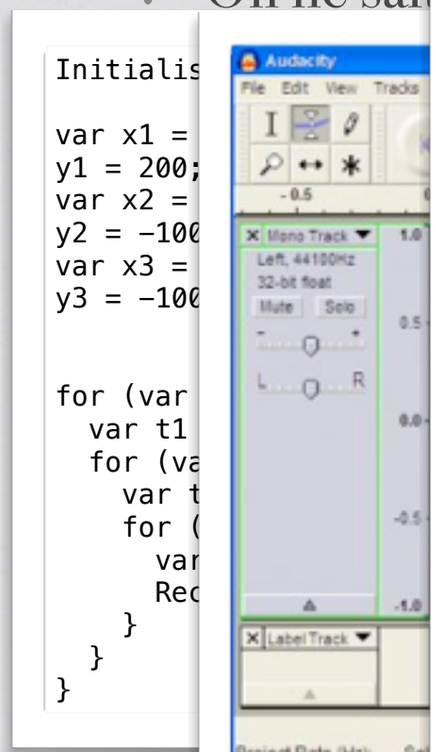


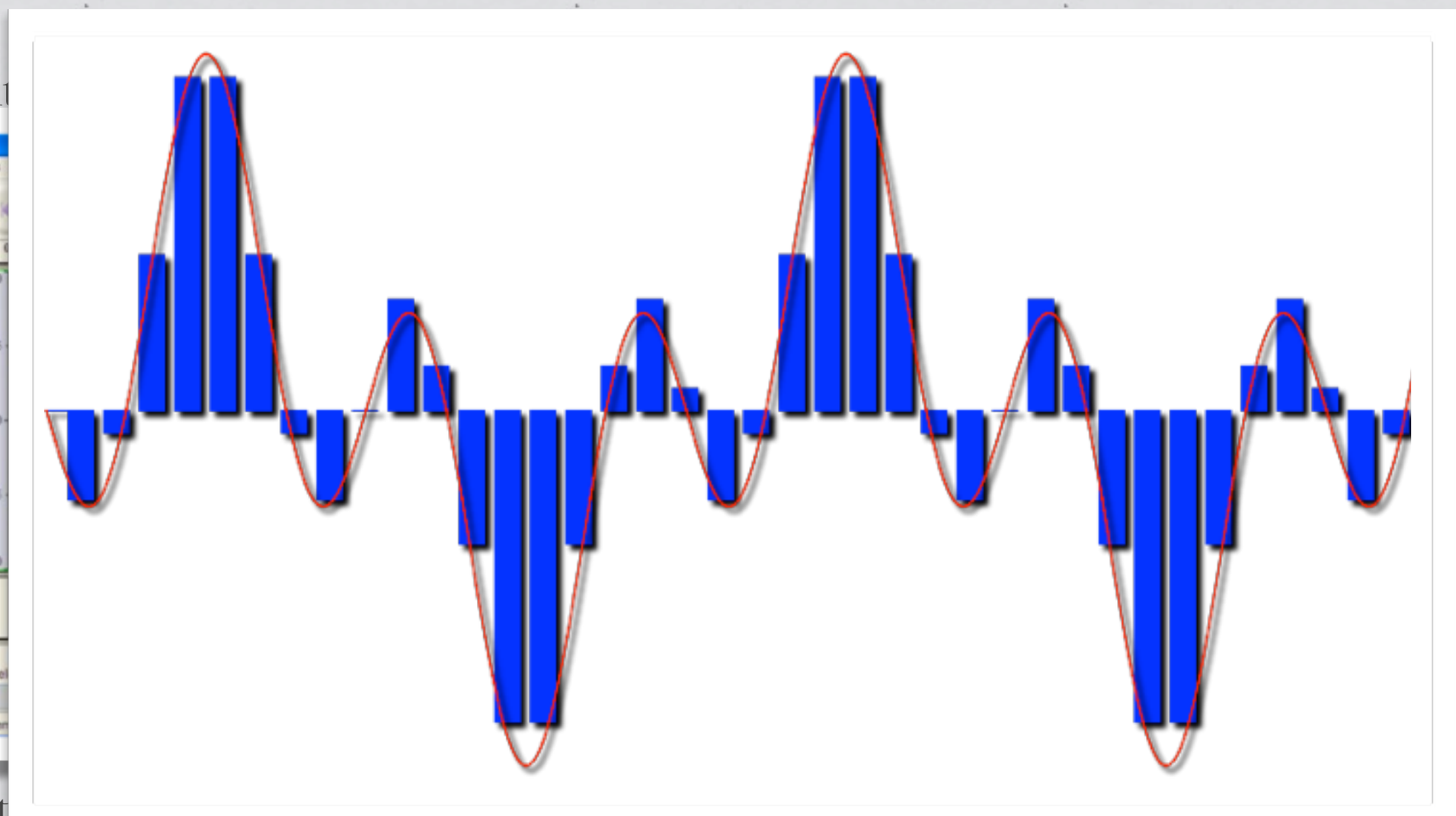
* son : tableau de fréquences entières

* Bilan: du point de vue de la machine, l'information c'est une suite de bits…

Avant de commencer 1/2 «Coder l'information»

* On ne sait





* son : t

* Bilan: du point de vue de la machine, l'information c'est une suite de bits…

Avant de commencer 2/2
La «théorie de l'information»

On s'intéresse à la manière de transmettre l'information (pas à l'information elle-même).

* Préoccupation n° 1: que l'information envoyée soit la même que celle qui est reçue (ou qu'on se rende compte qu'elle est différente). Détection et correction d'erreurs (ex: un CD rayé fonctionne toujours car il y a réparation des erreurs).

* Préoccupation n° 2: que l'information se transmette rapidement. Techniques de compression (ex: mp3, images sur internet jpeg mais surtout vidéo, TNT (on transmet 6 chaînes parfaitement là où avant on n'en transmettait qu'une).

* Préoccupation n° 3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

Avant de commencer 2/2
La «théorie de l'information»

Bit de parité : 01011010 01111101 11011011 00010010

↯ transmission avec erreurs

01111010 01111101 11011011 00000010

Erreurs de transmission dans les blocs 1 et 4

Il faut ajouter 1 bit pour 7 bits (taille + 14%)

de compression (ex: mp3, images sur internet jpeg mais surtout vidéo, TNT (on transmet 6 chaînes parfaitement là où avant on n'en transmettait qu'une).

* Préoccupation n°3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

Avant de commencer 2/2
La «théorie de l'information»

On s'intéresse à la manière de transmettre l'information (pas à l'information elle-même).

* Préoccupation n° 1: que l'information envoyée soit la même que celle qui est reçue (ou qu'on se rende compte qu'elle est différente). Détection et correction d'erreurs (ex: un CD rayé fonctionne toujours car il y a réparation des erreurs).

* Préoccupation n° 2: que l'information se transmette rapidement. Techniques de compression (ex: mp3, images sur internet jpeg mais surtout vidéo, TNT (on transmet 6 chaînes parfaitement là où avant on n'en transmettait qu'une).

* Préoccupation n° 3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

Avant de commencer 2/2
La «théorie de l'information»

Compression: petit calcul autour de la vidéo HD (16 millions de couleurs):

- image 1024*1080*(8*3) = 33 000 000 bits par images
- 30 images par secondes -> 1 000 000 000 bits par seconde = 1 Gbits par seconde = 125 Mo/s
- Contenance d'un CD: 700Mo = 700 000 000 = 5 600 000 000 bits -> 5,6 s. vidéo non compressée
- Contenance d'un DVD: 8Go -> 1 minute de vidéo non compressée
- Le Parrain : 168min -> 168 DVD si non compressé or, en divx, Le Parrain tient sur 1 CD.

- débit TNT : 24 000 000 bits par seconde maxi mais limité à 8 000 000 bits/s à cause des erreurs d'émission sur le canal.

-> sans compression, pas de DVD
-> sans compression et correction d'erreur, pas de TNT
-> sans compression et correction d'erreur et cryptographie, pas de canal+ sur TNT.

* Préoccupation n°3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

Avant de commencer 2/2
La «théorie de l'information»

On s'intéresse à la manière de transmettre l'information (pas à l'information elle-même).

* Préoccupation n° 1: que l'information envoyée soit la même que celle qui est reçue (ou qu'on se rende compte qu'elle est différente). Détection et correction d'erreurs (ex: un CD rayé fonctionne toujours car il y a réparation des erreurs).

* Préoccupation n° 2: que l'information se transmette rapidement. Techniques de compression (ex: mp3, images sur internet jpeg mais surtout vidéo, TNT (on transmet 6 chaînes parfaitement là où avant on n'en transmettait qu'une).

* Préoccupation n° 3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

Avant de commencer 2/2
La «théorie de l'information»

On s'intéresse à la manière de transmettre l'information (pas à l'information elle-même).

Un concept important : la «**redondance**»

Mon nom est: J r mie B URDON

Mon téléphone est: 2 51 12 8 25

Retirer de la redondance pour compresser

Ajouter de la redondance pour repérer les erreurs

Utiliser la redondance pour attaquer un système crypto

* Préoccu… …le qui est reçue (c… …t correct… …éparation des erre…

* Préoccu… …chniques de com… …éo, TNT (on tra… …qu'une).

* Préoccupation n°3: que certaines informations soient protégées (ex: codes de cartes bleues, certaines conversations,...). Cryptographie !!!

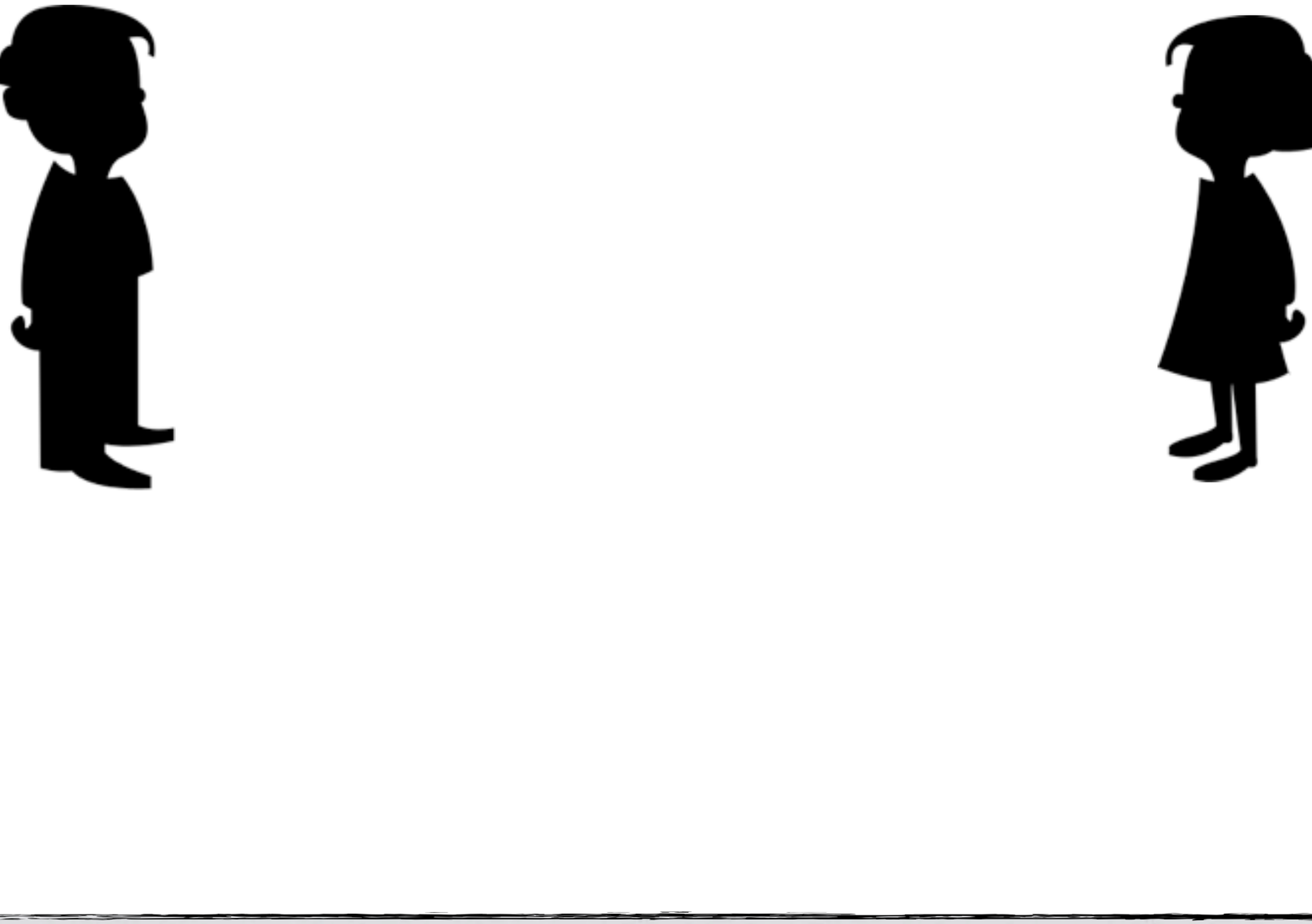
Plan

1. Historique, terminologie et premières définitions
   - Quelques techniques pour camouflage
   - Quelques attaques
2. Cryptographie «moderne»
   - Cryptographie vs cryptanalyse
   - Sécurité et chiffrement

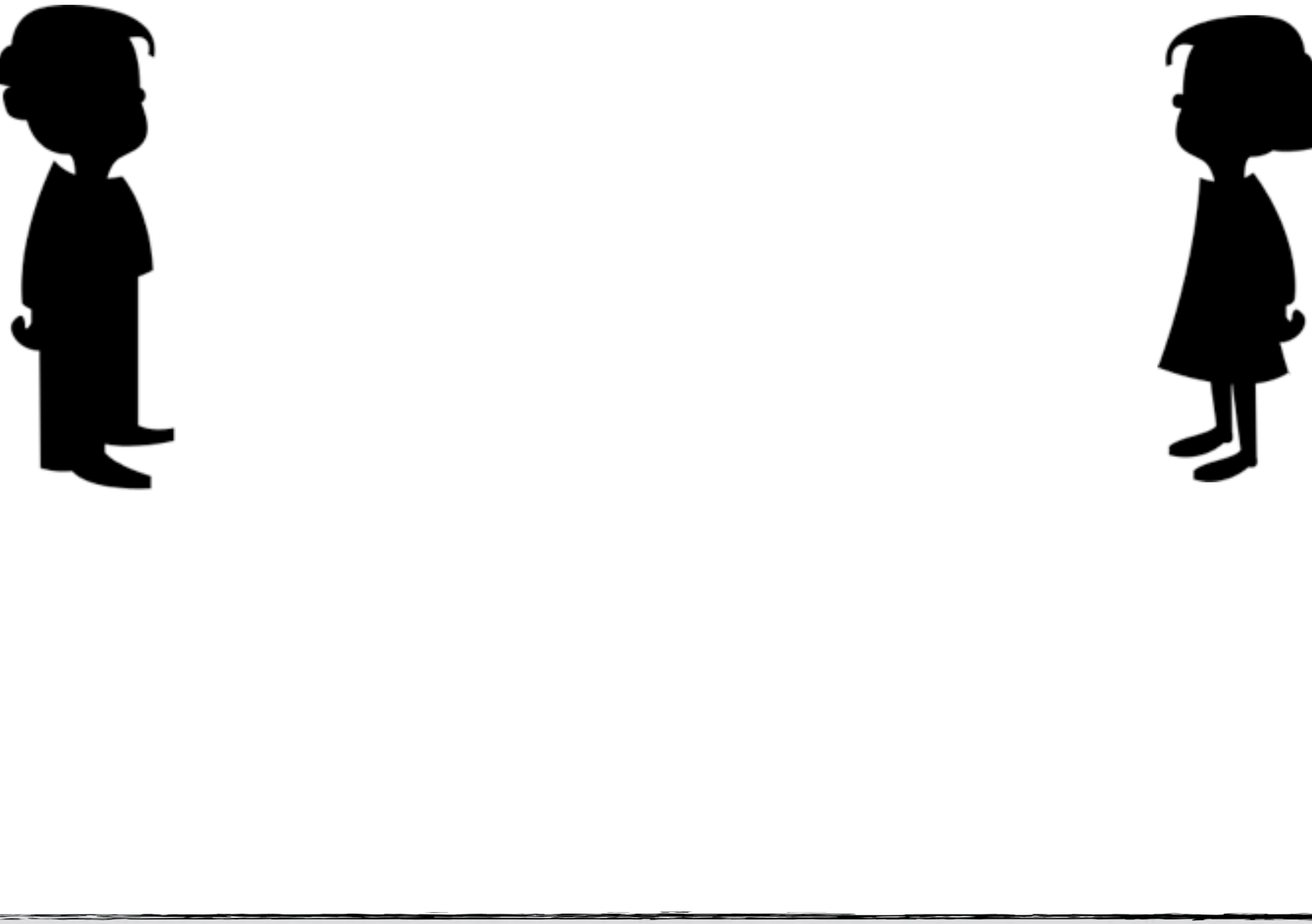
Bref historique

- Depuis l'antiquité, on trouve des traces de méthodes destinées à chiffrer des messages
- Longtemps cantonnée à des fins militaires
- La cryptographie
  - se «démocratise» avec le développement massif de l'informatique personnelle
  - doit s'adapter à des applications de plus en plus diverses (chiffrement de texte, de son ou de video; authentification; signature électronique; vote électronique,. . .)
  - subit la croissance exponentielle de la puissance de calcul des ordinateurs

Petit schéma général



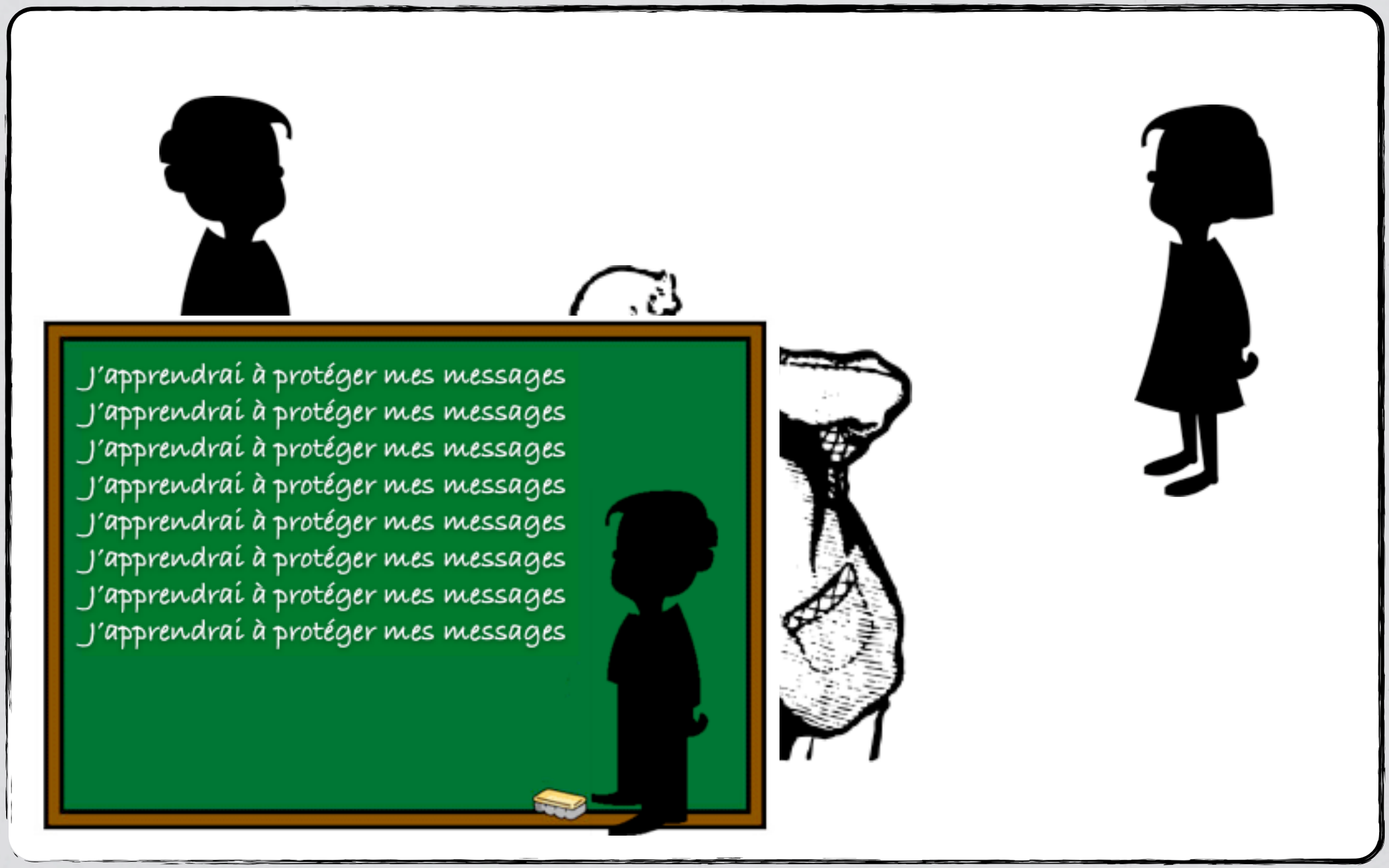
Petit schéma général



Petit schéma général

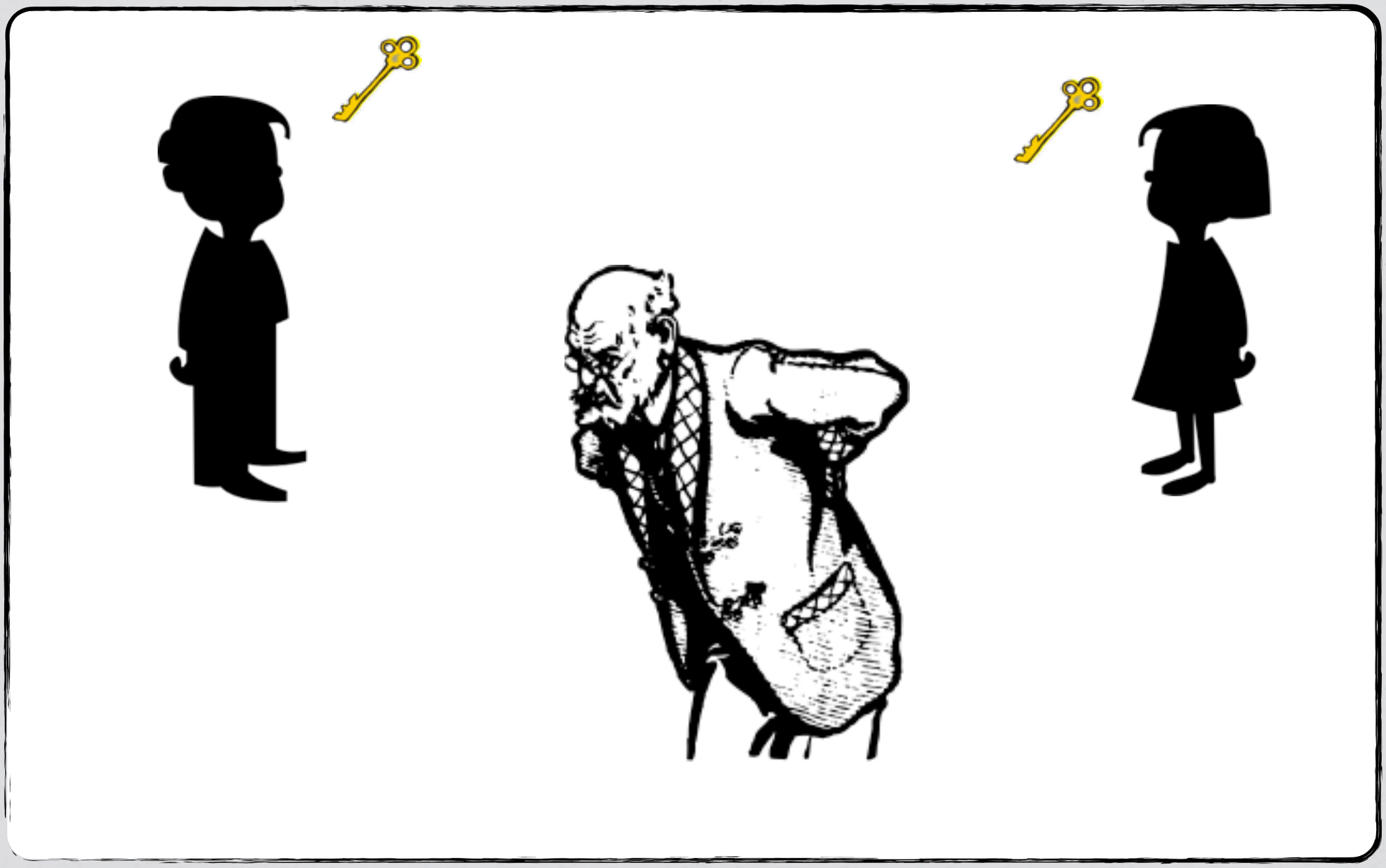
Petit schéma général

Petit schéma général

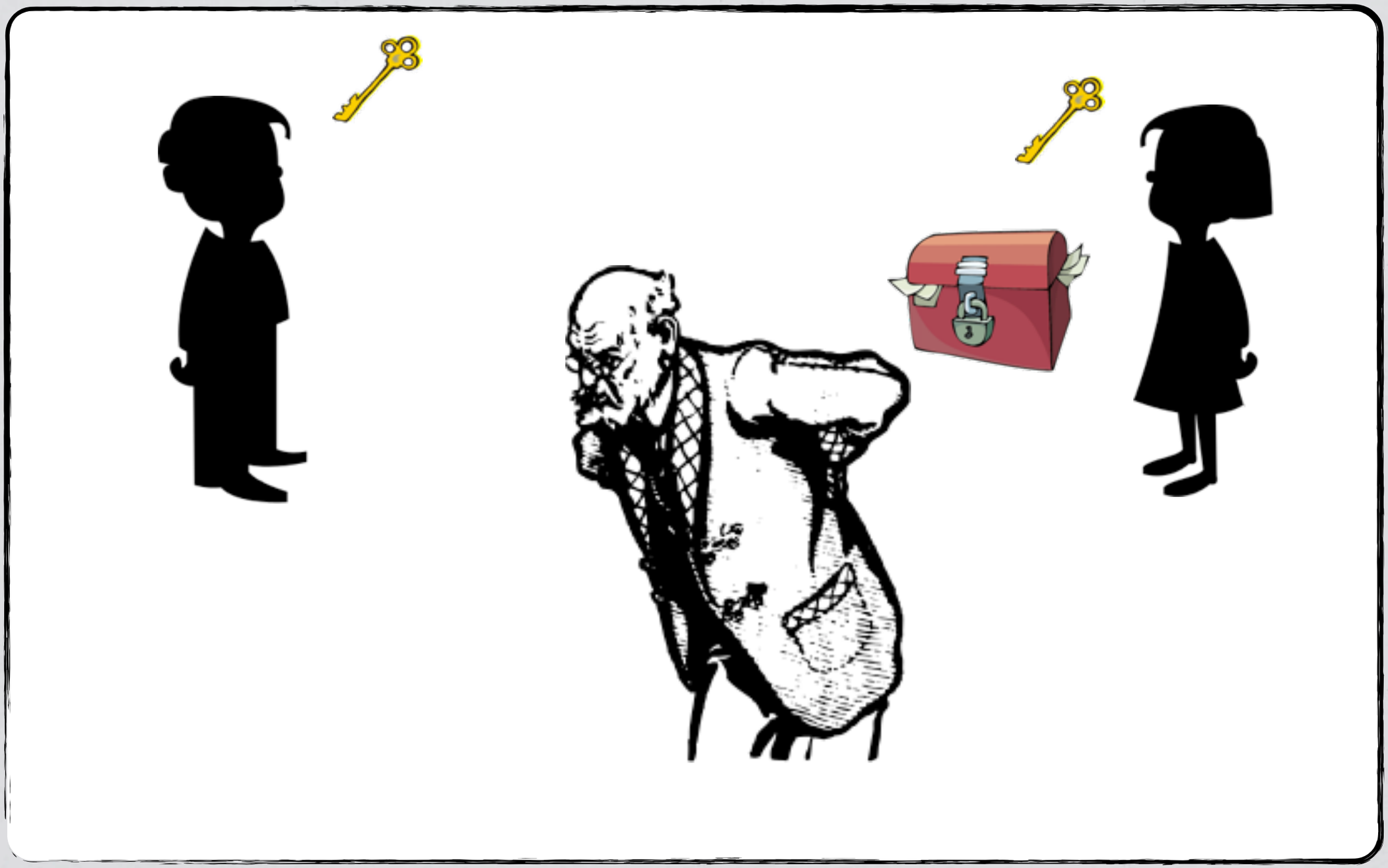


Petit schéma général

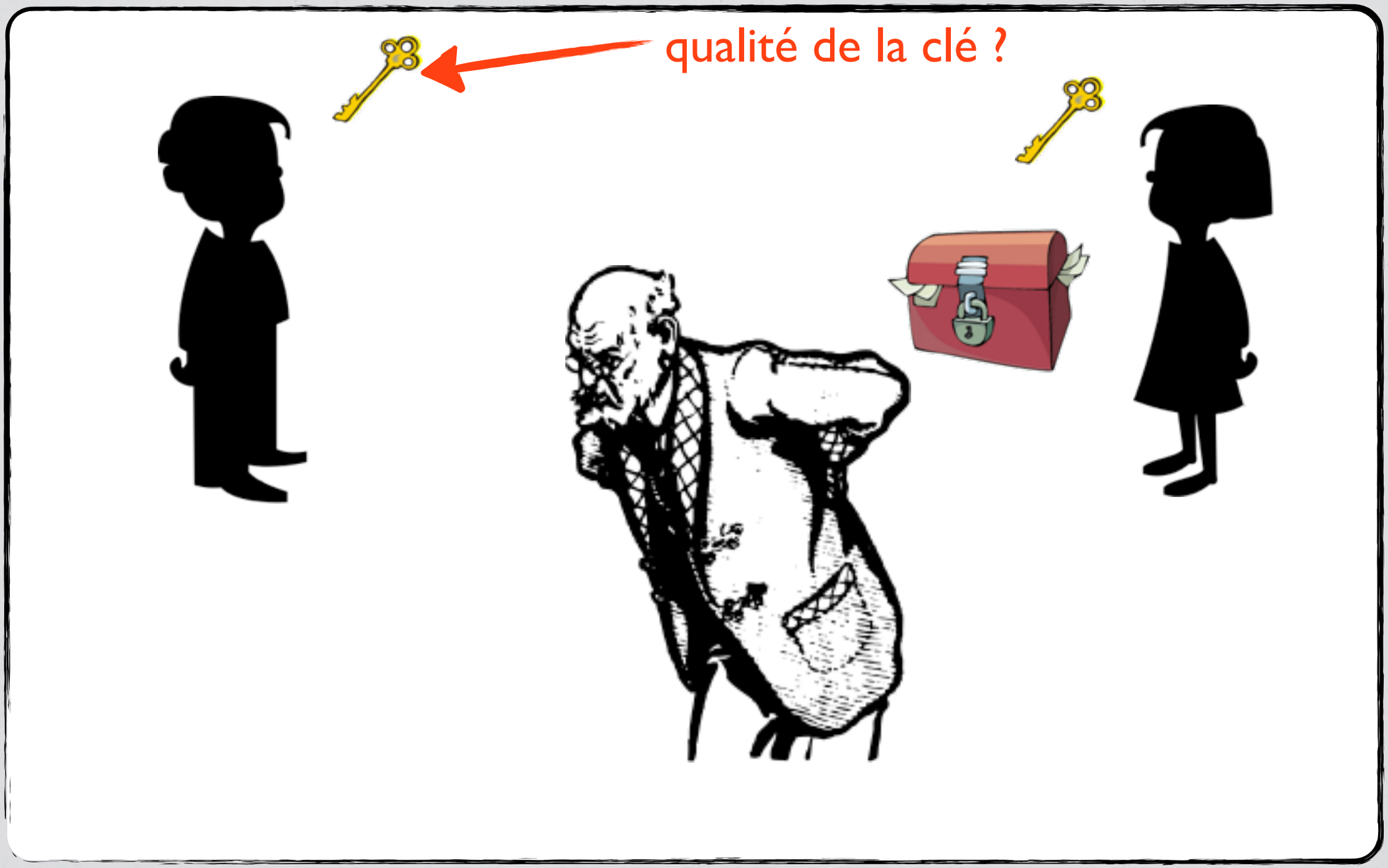
Petit schéma général



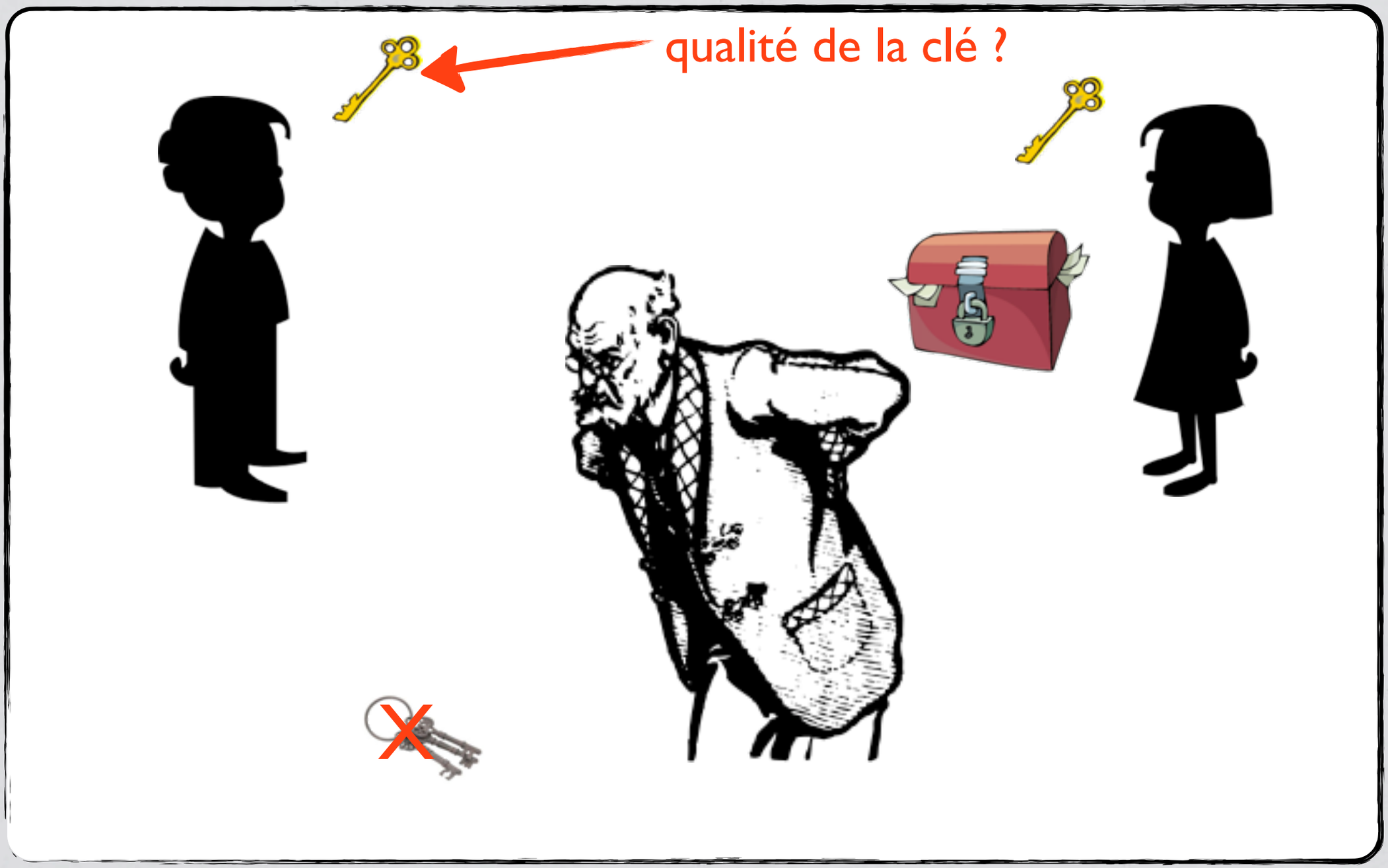
Petit schéma général



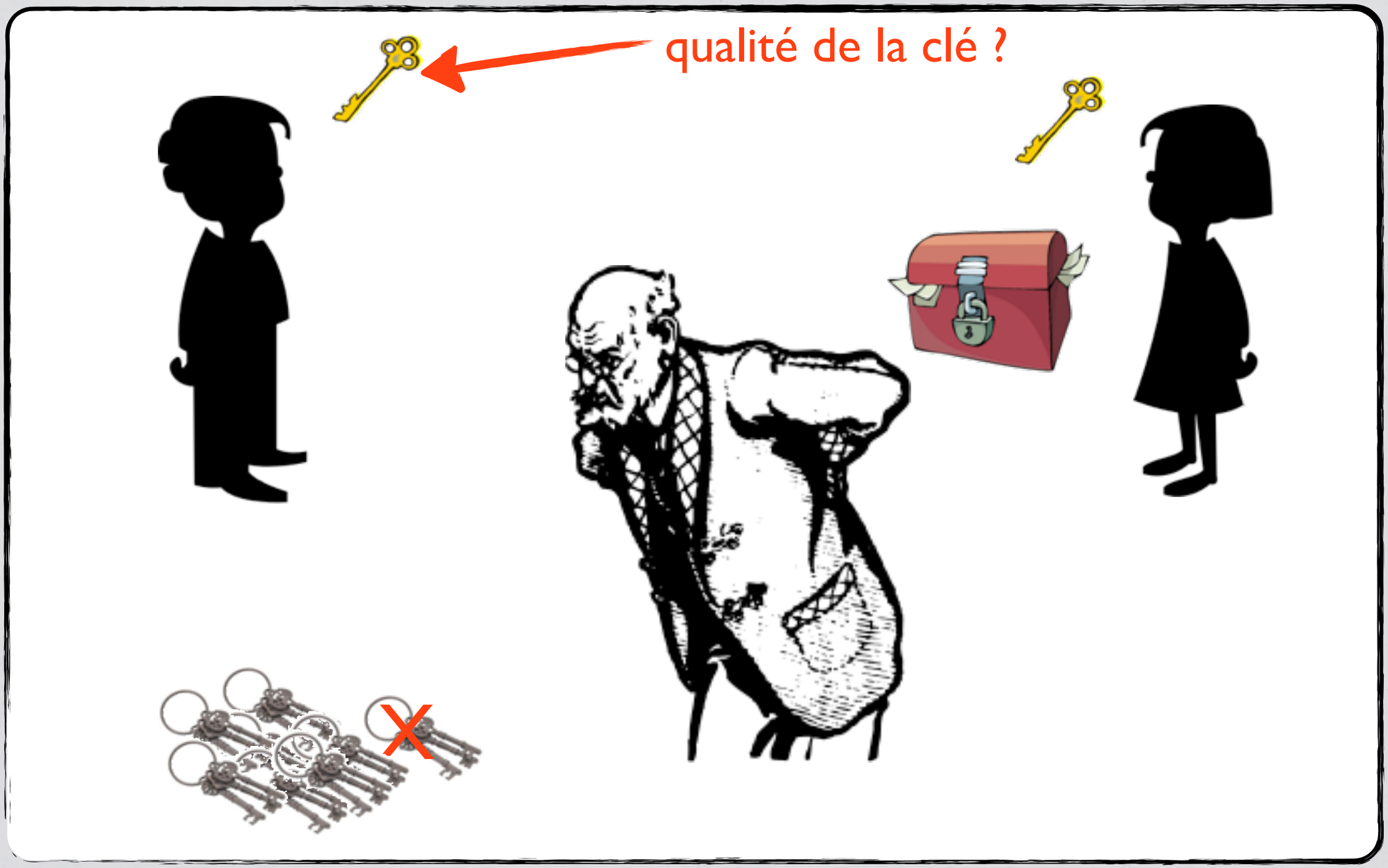
Petit schéma général



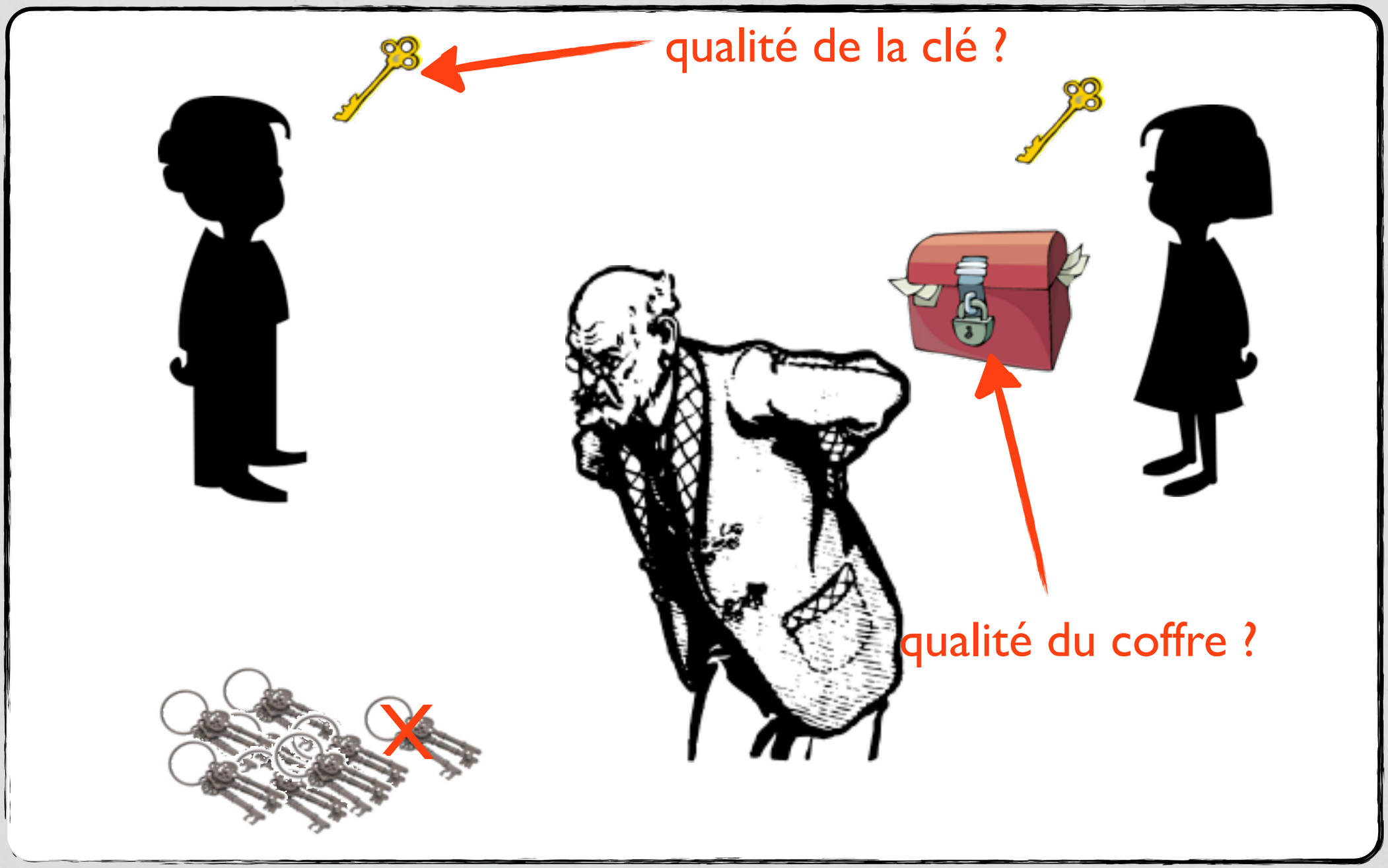
Petit schéma général



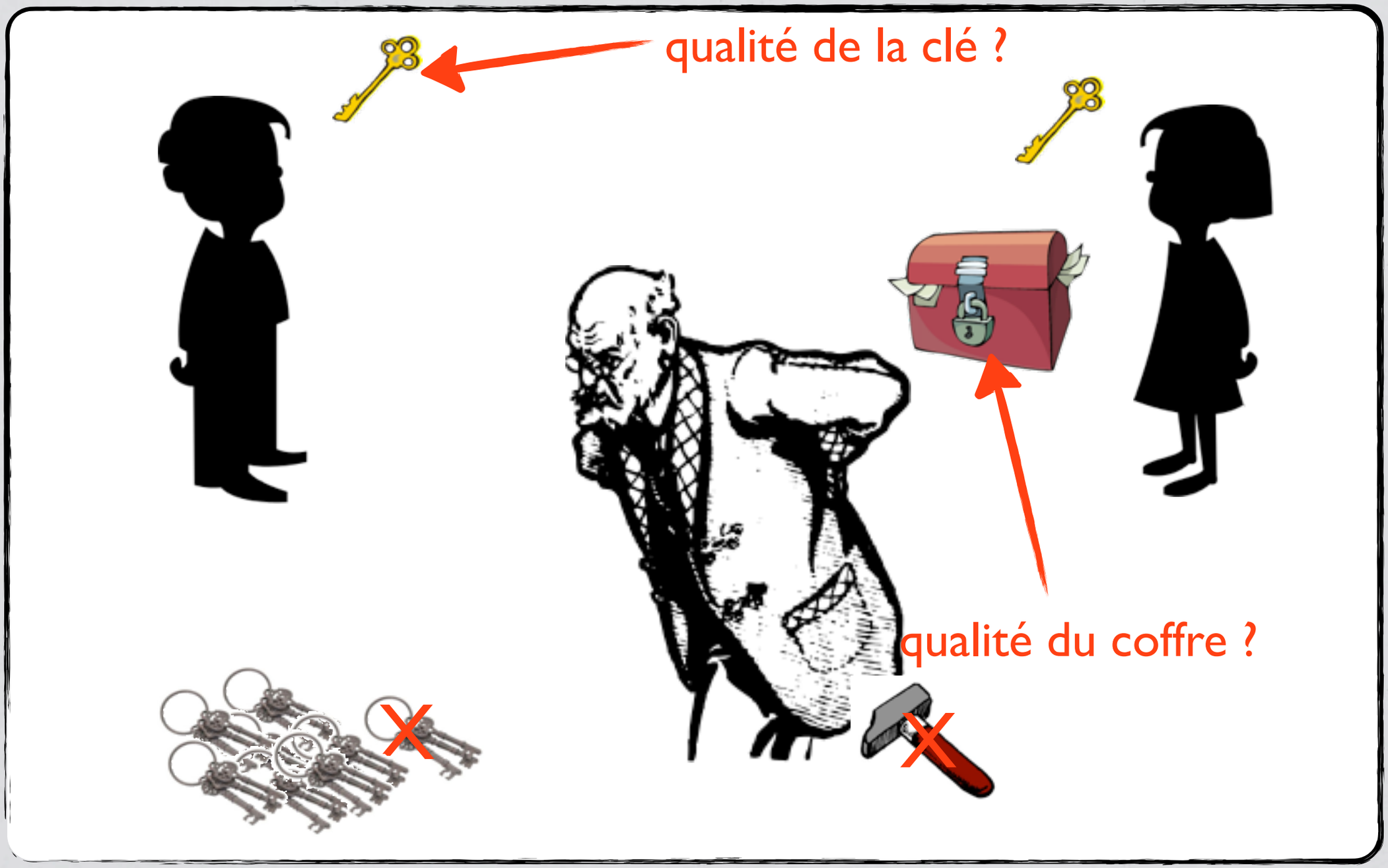
Petit schéma général



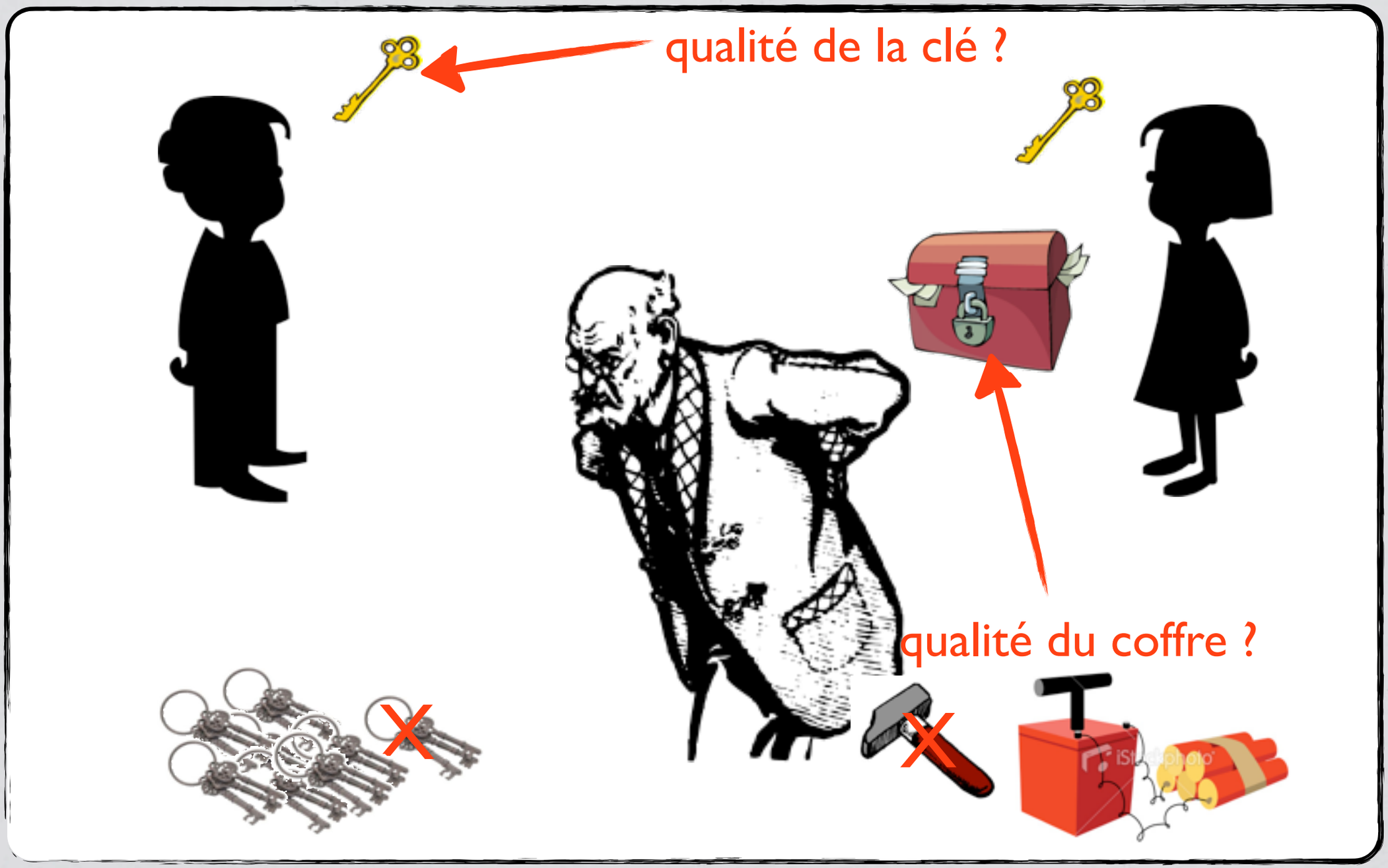
Petit schéma général



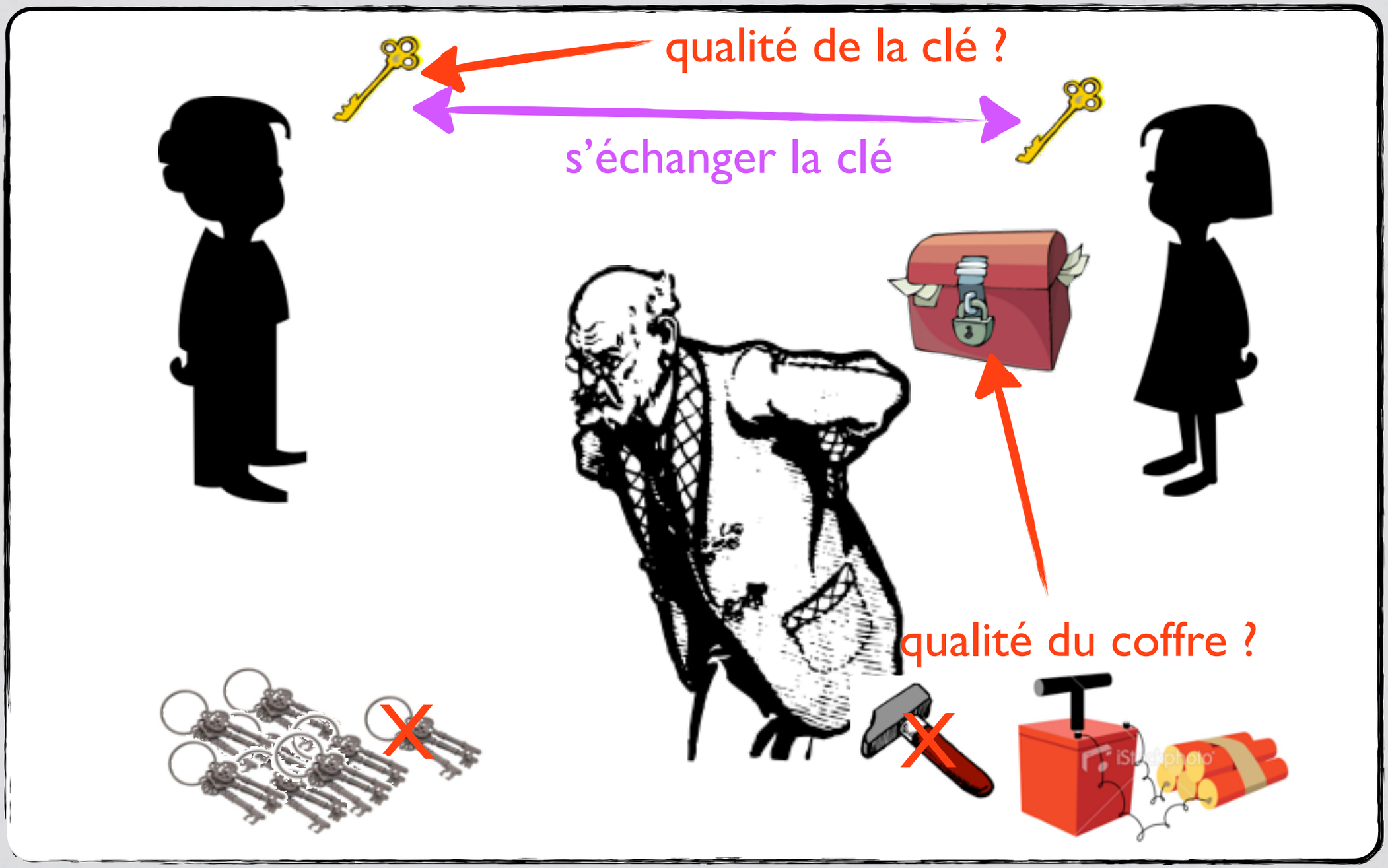
Petit schéma général



Petit schéma général



Petit schéma général

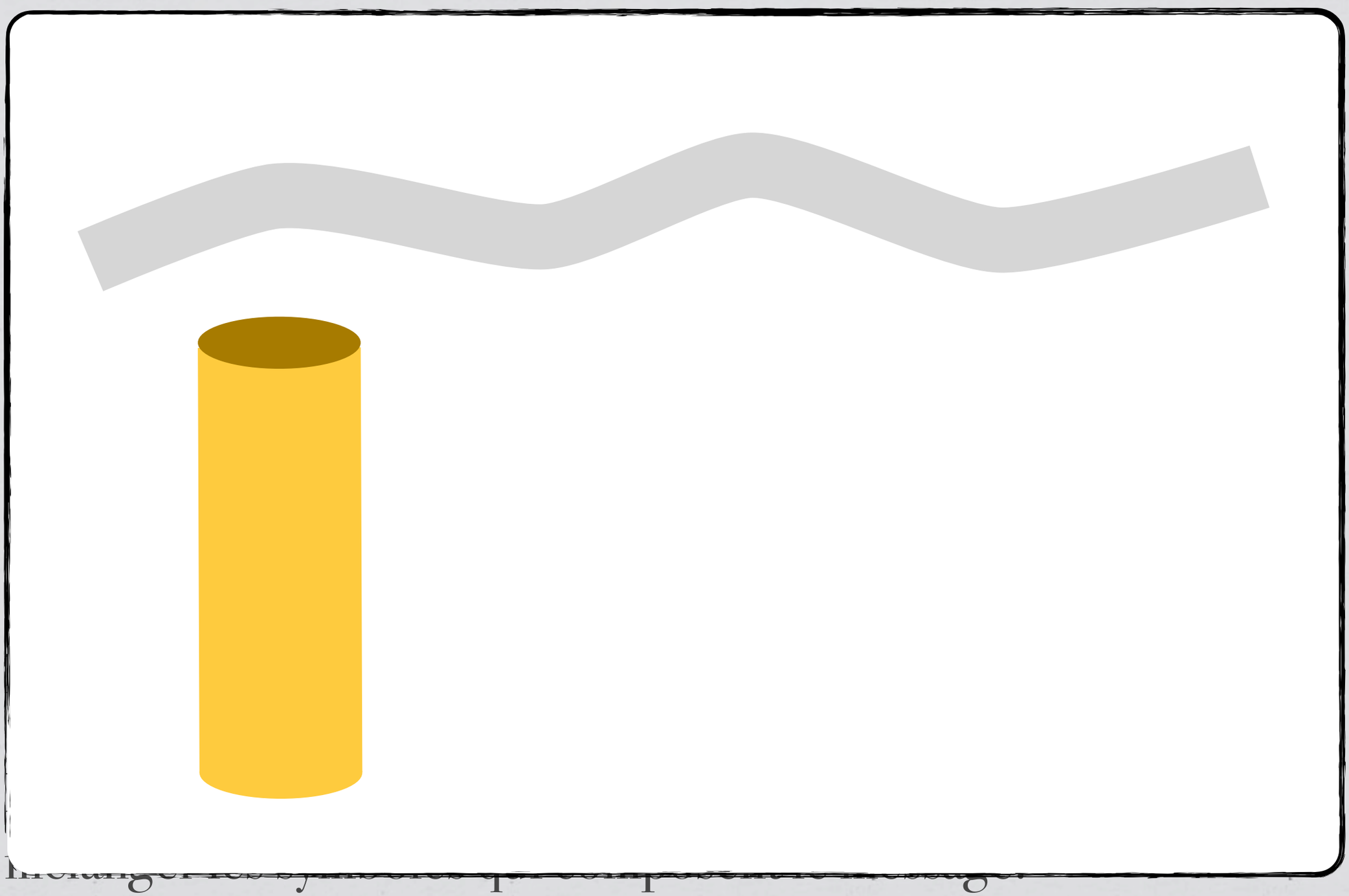


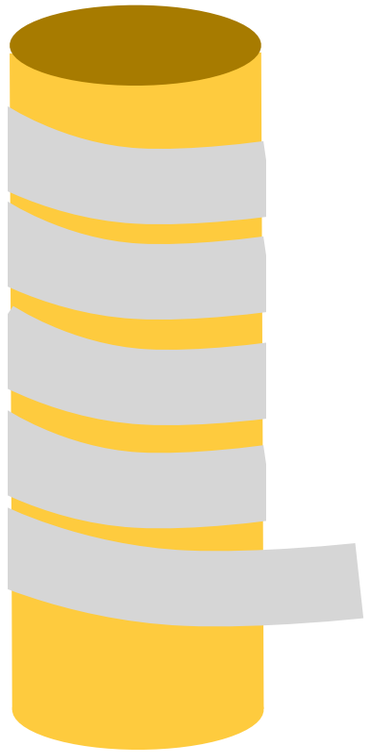
Le mélange Grec 500 av JC: *le transcodage*

Dès l'antiquité, les Grecs échangeaient des messages chiffrés obtenus en mélangeant les lettres du message.

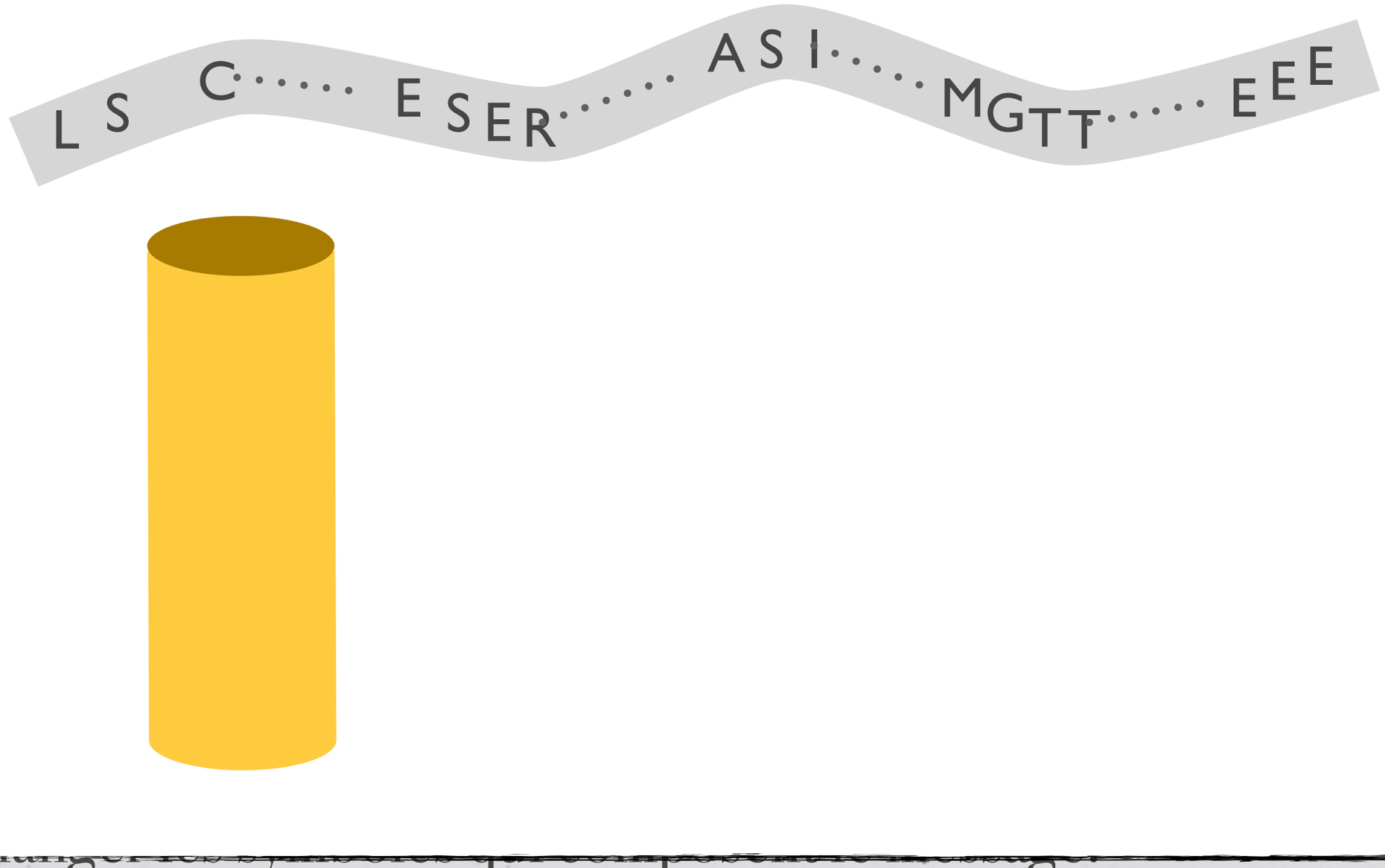
Le mélange était obtenu en enroulant des bandes de papier sur un cylindre et en y inscrivant le message verticalement. Pour lire le message, il fallait posséder un cylindre du même diamètre.

Le transcodage est une technique de cryptage qui consiste à mélanger les symboles qui composent le message.






L S C
E S E R
A S I
M G T T
E E E


L S C E S E R A S I M G T T E E E

L S C····· E SER····· A S I····· MGTT····· E E E

La clé, c’est le diamètre du tuyau

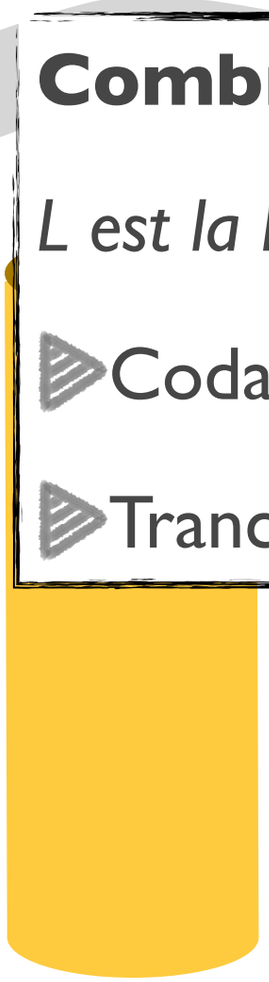
(ou plutôt le nombre de symboles

à passer dans la lecture)

Combien de codages différents (clés) ?

L est la longueur du message

- Codage «grec» : *L* clés
- Trancodage général: *L!* clés



(ou plutôt le nombre de symboles

à passer dans la lecture)

Jules Verne

ppelée l'une après l'autre, et forma l'incompréhensible succession des mots uivants :

mm.rnlls esreuel seecJde
sgtssmfunteief niedrke
kt,samn atrateS Saodrrn
emtnael nuaect rrilSa
Atvaar .nscrc ieaabs
ccdrmi eeutul frantu
dt,iac oseibo KediiY

Quand ce travail fut terminé, mon oncle prit vivement la feuille sur laquelle je venais d'écrire, et il l'examina longtemps avec attention.

Jules Verne

ppelée l'une après l'autre, et forma l'incompréhensible succession des mots uivants :

mm.rnlls esreuel seecJde
sgtssmfunteief niedrke
kt,samn atrateS Saodrrn
emtnael nuaect rrilSa
Atvaar .nscrc ieaabs
ccdrmi eeutul frantu
dt,iac oseibo KediiY

Quand ce travail fut terminé, mon oncle prit vivement la feuille sur laquelle je venais d'écrire, et il l'examina longtemps avec attention.

Il était conçu en ces termes :

In Sneffels Yoculis craterem kem delibat umbra Scartaris Julii intra calendas descende, audas viator, et terrestre centrum attinges. Kod feci. Arne Saknussem.

Ce qui, de ce mauvais latin, peut être traduit ainsi :

Descends dans le cratère du Yocul de Sneffels que l'ombre du Scartaris vient caresser avant les calendes de Juillet, voyageur audacieux, et tu parviendras au centre de la Terre. Ce que j'ai fait. Arne Saknussemm.

Le codage de César 40 av JC : *la substitution*

Il s'agit d'un codage mono-alphabétique défini par une rotation circulaire des lettres

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

Texte clair : `coder des messages, qu'elle drole d'idée !`

Texte codé : `frghu dhv phvvdjhv, tx'hooh guroh g'lghh !`

- Décrit par Suétone (écrivain romain) dans *la vie des douze Césars*
- Sa sécurité est très faible puisqu'au plus 26 essais suffisent à retrouver le message initial
- Simple d'utilisation : utilisé par les sudistes et même par l'armée russe en 1915. Utilisé aussi sur les forums USENET sous le nom de ROT-13 pour éviter qu'un message soit lu involontairement (ex: devinettes)
- La substitution (ou codage mono-alphabétique) est une technique de cryptage qui consiste à permuter les lettres de l'alphabet.

Le codage de César 40 av JC : *la substitution*

Il s'agit d'un codage mono-alphabétique défini par une rotation circulaire des lettres

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

Tex

Tex

- I

- S al

- S é

aus

inv

- La substitution (ou codage mono-alphabétique) est une technique de cryptage qui consiste à permuter les lettres de l'alphabet.

Combien de codages différents (clés) ?

Indépendant de la longueur du message

- Codage «César» : 26 clés
- Substitution générale: $26!=4.10^{26}$ clés

Un humain qui traite 100000 clés par seconde mettra plus de 10^{12} siècles (l'univers à $1,37.10^{8}$ siècles) pour toutes les tester et pourtant ce n'est pas un bon cryptosystème !

Chiffre de Vigénère: *Substitution poly-alphabétique*

On utilise un mot clé et des décalages de César. Par exemple, lorsque la clé est ‘FEU’, on utilise alternativement les décalages :

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

BONJOUR est codé par GSHOSOW !

La longueur de la clé est une donnée importante !

Chiffre de Vigénère: *Substitution poly-alphabétique*

On utilise un mot clé et des décalages de César. Par exemple, lorsque la clé est 'FEU', on utilise alternativement les décalages :

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

Combien de codages différents (clés) ?

Indépendant de la longueur du message, K= longueur de la clé

- Codage «Vigénère» : 26^K clés
- En général: $(26!)^K$

BC

La longueur de la clé est une donnée importante !

Lithanie de Trithène : *Stéganographie*

L'abbé de Trithène (1499) est le premier à écrire un ouvrage *Polygraphiae* traitant de cryptographie

C'est le premier à introduire le principe de stéganographie.

Le message est caché dans un texte ou une image a priori anodin pour une personne non informée.

Lithanie de Trithène : *Stéganographie*

Exemple : une lithanie de Trithène

Dans la félicité à perpétuité,
Dans son royaume à perpétuité,
En paradis à perpétuité,
Ainsi qu'en toute éternité.
Dans la gloire à perpétuité,
Mais dans son règne
Sempiternel, toujours dans la félicité,
Tant dans la lumière que dans la béatitude,
Et toujours dans la gloire à perpétuité,
Mais dans son règne

L’abbé d… un ouvrage
Polygrap…

C’est le p…ographie.

Le messa…a priori anodin
pour une…

Lithanie de Trithène : *Stéganographie*

Exemple : une lithanie de Trithène

Dans la félicité E
Dans son royaume à perpétuité,
En paradis à perpétuité,
Ainsi qu'en toute éternité.
Dans la gloire à perpétuité,
Mais dans son règne
Sempiternel, toujours dans la félicité,
Tant dans la lumière que dans la béatitude,
Et toujours dans la gloire à perpétuité,
Mais dans son règne

L'abbé d… un ouvrage *Polygrap…*

C'est le p…ographie.

Le messa…a priori anodin pour une…

Lithanie de Trithène : *Stéganographie*

L'abbé d… un ouvrage
Polygrap…

C'est le p… ographie.

Le messa… a priori anodin
pour une…

Exemple : une lithanie de Trithène

Dans la félicité E
Dans son royaume E
En paradis E
Ainsi qu'en toute éternité.
Dans la gloire E
Mais dans son règne
Sempiternel, toujours dans la félicité,
Tant dans la lumière que dans la béatitude,
Et toujours dans la gloire E
Mais dans son règne

Lithanie de Trithène : *Stéganographie*

Exemple : une lithanie de Trithène

R E
D T me E
N E
Ain Z éternité.
L E
Mai S gne
F O R
Tan M ère que dar U
Et toujours L E
Mai S gne

L’abbé d… un ouvrage *Polygrap…*

C’est le p…nographie.

Le messa…a priori anodin pour une…

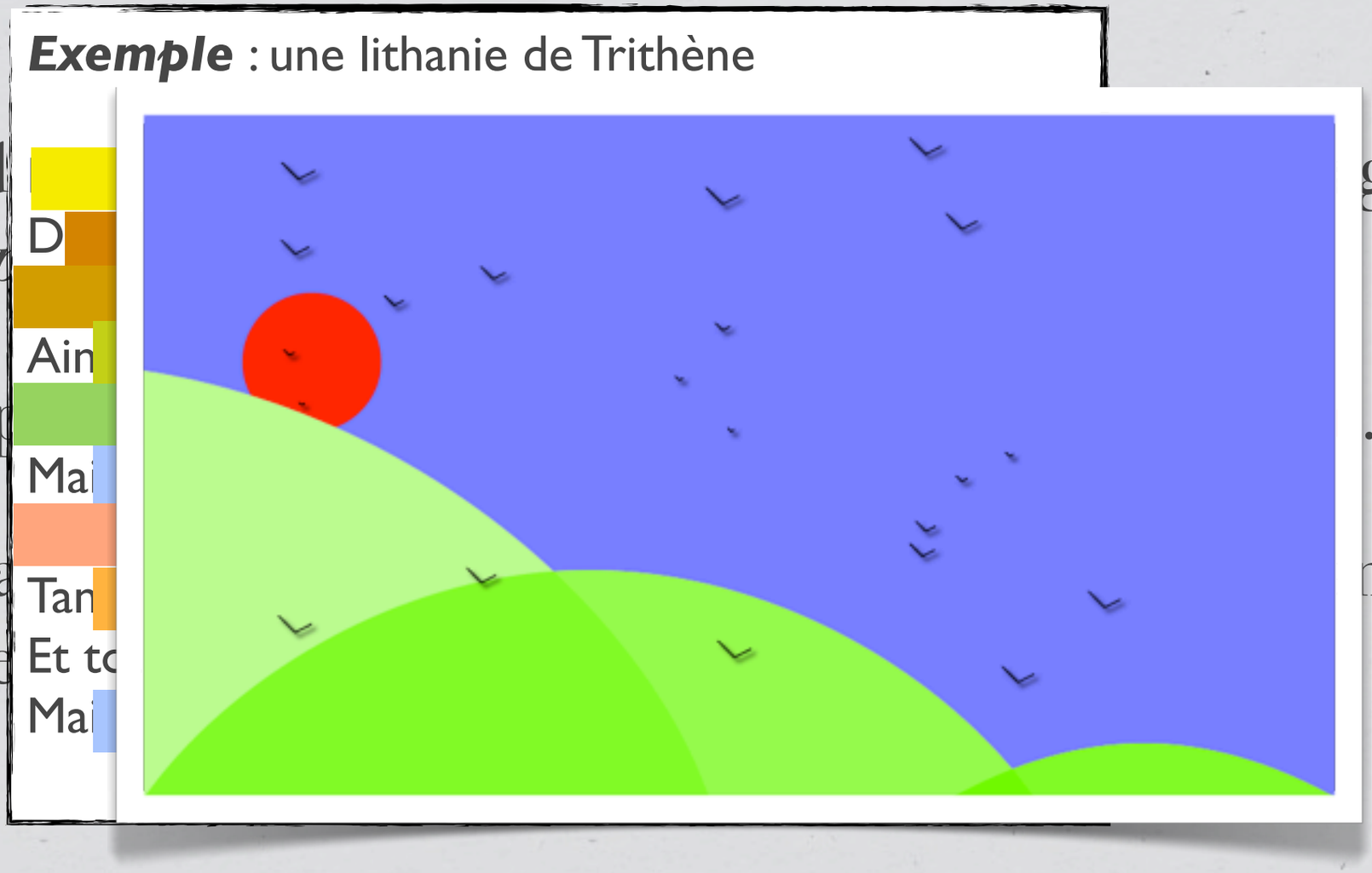
Lithanie de Trithène : *Stéganographie*

L’abbé d ge
Polygrap

C’est le p .

Le messa hodin
pour une

Exemple : une lithanie de Trithène



Lithanie de Trithène : *Stéganographie*

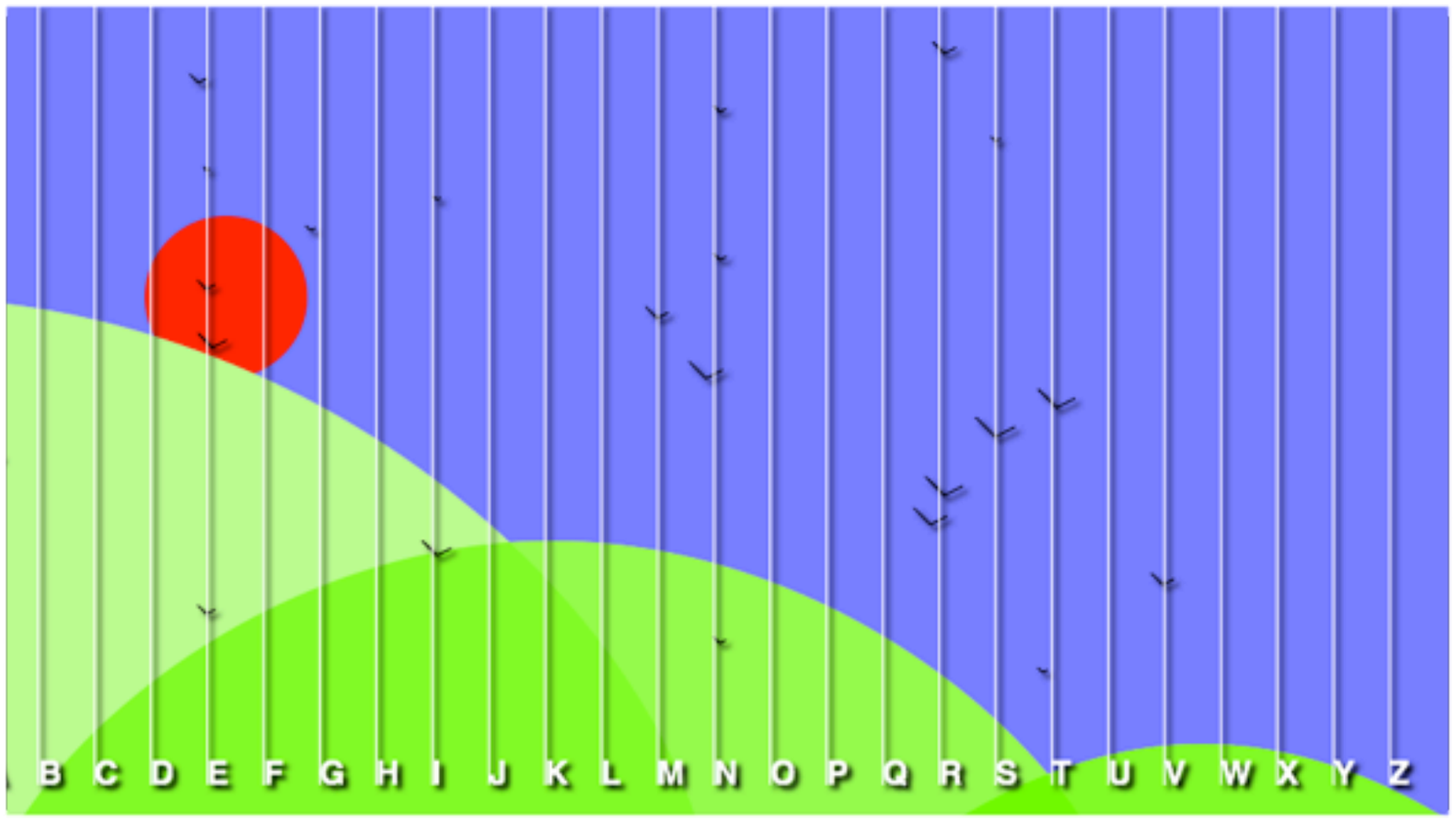
L'abbé d
Polygrap

C'est le p

Le messa din
pour une

Exemple : une lithanie de Trithène

D
Ain
Mai
Tan
Et to
Mai



La machine Enigma (1945)

C’est une machine (plusieurs modèles) de cryptage de message utilisée par l’armée allemande pendant la seconde guerre mondiale. L’histoire dit que les alliés, en se procurant les plans de la machine, on réussi à décrypter près de 18000 messages échangés entre les différentes division allemandes, ce qui a évidemment contribué fortement à leur victoire.

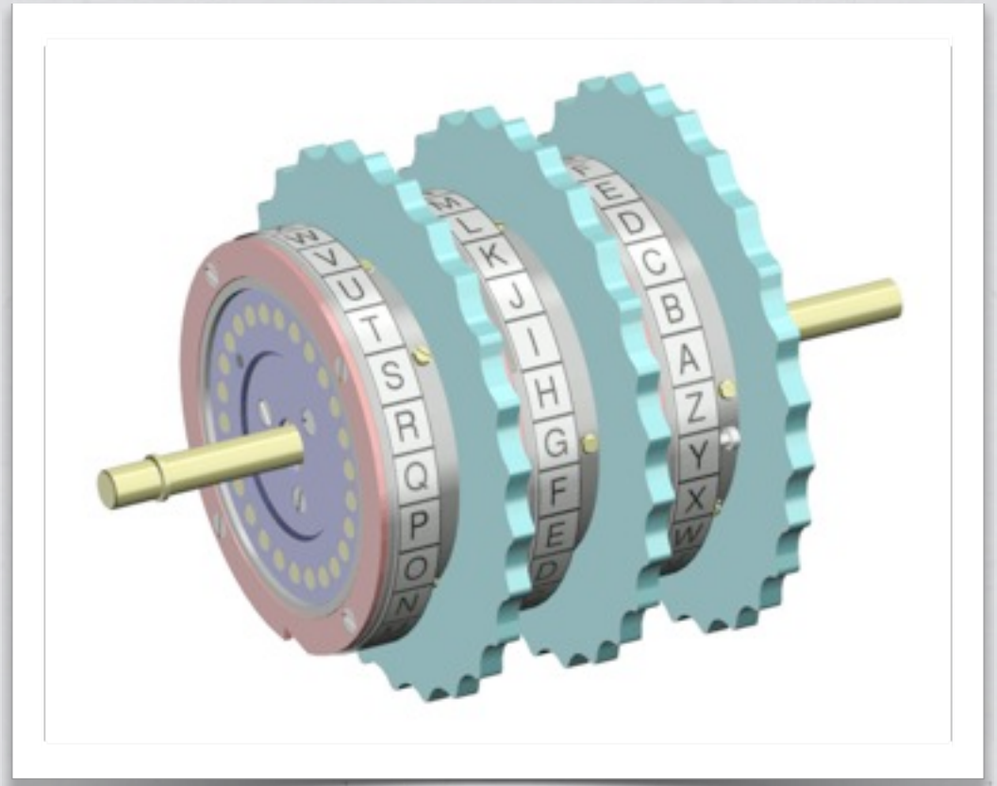
La m

)45)

C’est une mach lisée par
l’armée alleman dit que les
alliés, en se pro r près de
18000 message s, ce qui a
évidemment co

Enigma : PRINCIPES

C’est une machine à rotors

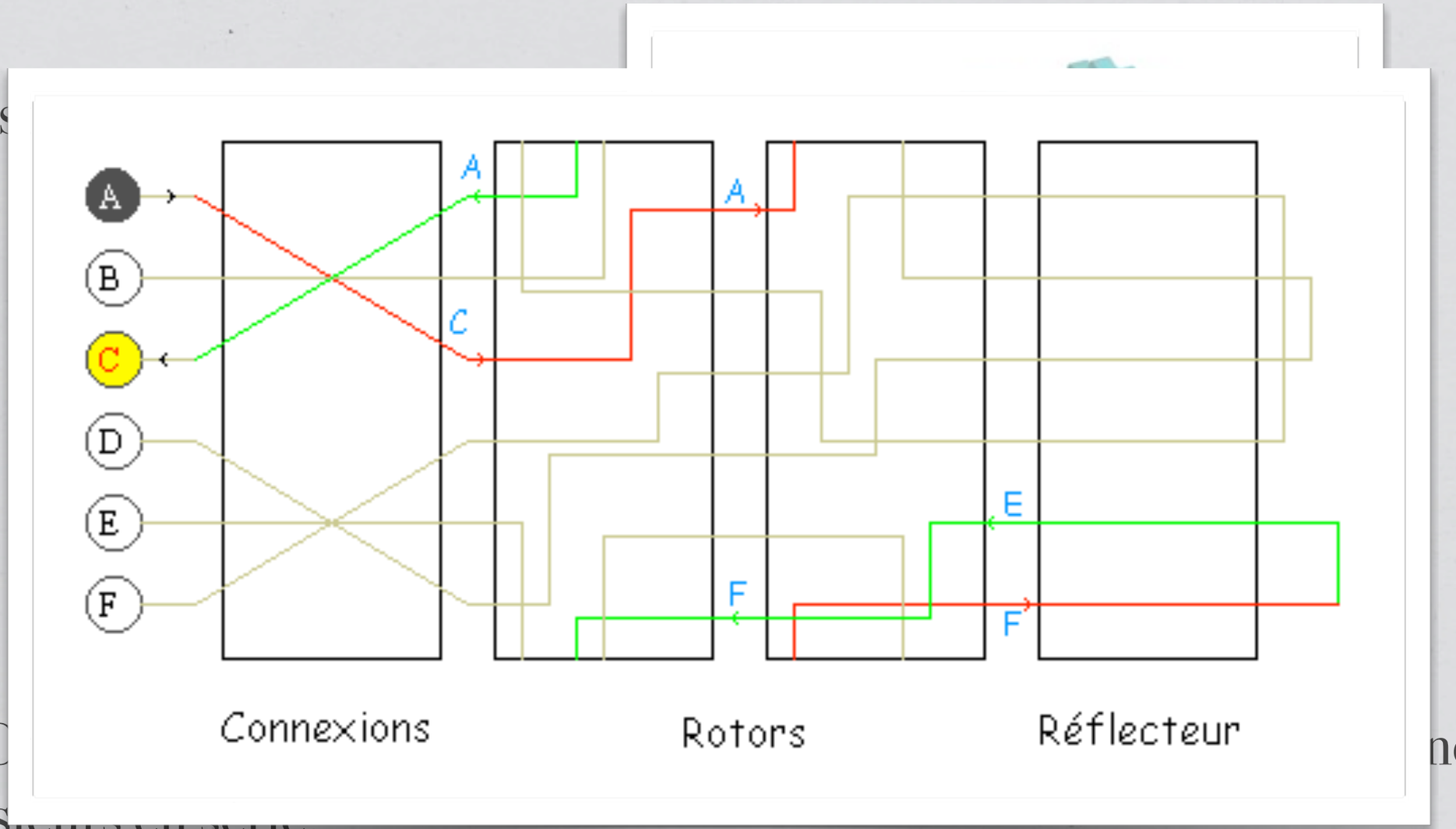


- Chaque rotor est une substitution mono-alphabétique. il y en a donc plusieurs en série.

- A chaque codage de lettre, le rotor tourne et la substitution change.

Enigma : PRINCIPES

C'es



- C nc plusieurs en série.
- A chaque codage de lettre, le rotor tourne et la substitution change.

Enigma : PRINCIPES

C'é

- plusieurs en série.

- A chaque codage de lettre, le rotor tourne et la substitution change.

Combien de codages différents (clés) ?

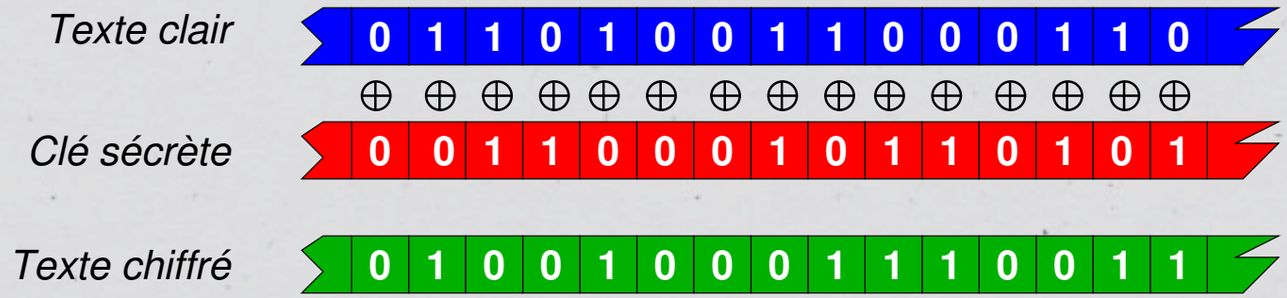
- la position des 6 fiches du tableau de connexion
- l'ordre des rotors
- la position initiale des rotors

En tout : 10^{16} clés !

Un humain qui traite 100000 clés par seconde mettra plus de 30 siècles pour toutes les tester et pourtant ce n'est pas un bon cryptosystème !

Système de Vernam et téléphone rouge

C’est un système à clé secrète partagée entre deux parties.



C’est le seul cryptosystème prouvé (mathématiquement) inconditionnellement sûr (par Shannon) si la clé est parfaitement aléatoire et n’est utilisée qu’une fois.

Il a longtemps été utilisé pour crypter les conversations du téléphone rouge (clé envoyée entre les pays par valise diplomatique).

Il marque le début de la recherche «moderne» en cryptographie.

Principes de Kerckhoffs (fin XIXe)

Traité de cryptographie militaire

1. Une information codée ne doit en aucun cas pouvoir être déchiffrée sans la connaissance de sa clé.

2. Les interlocuteurs ne doivent pas subir de dégâts au cas où le système de codage serait dévoilé.

3. La clé doit être simple et modifiable à souhait.

4. Les cryptogrammes doivent être transportables, c'est-à-dire télégraphiables.

5. L'appareil de codage et les documents doivent être transportables.

6. Le système doit être simple d'utilisation.

7. Le système de chiffrage doit être au préalable examiné par des experts.

Principes de Kerckhoffs (fin XIXe)

Traité de cryptographie militaire

1. Une i… …ns la conna…
2. Les in… …de codag…
3. La clé…
4. Les c… …iables.
5. L'app…
6. Le système doit être simple d'utilisation.
7. Le système de chiffrage doit être au préalable examiné par des experts.

TOUS LES CRYPTOSYSTEMES MODERNES

RESPECTENT CES PRINCIPES (SAUF QUELQUES CRYPTOSYSTEMES MILITAIRES)

Rq: Maxime de Shannon: *l'adversaire connaît le système.*

Quelques attaques sur les cryptosystèmes

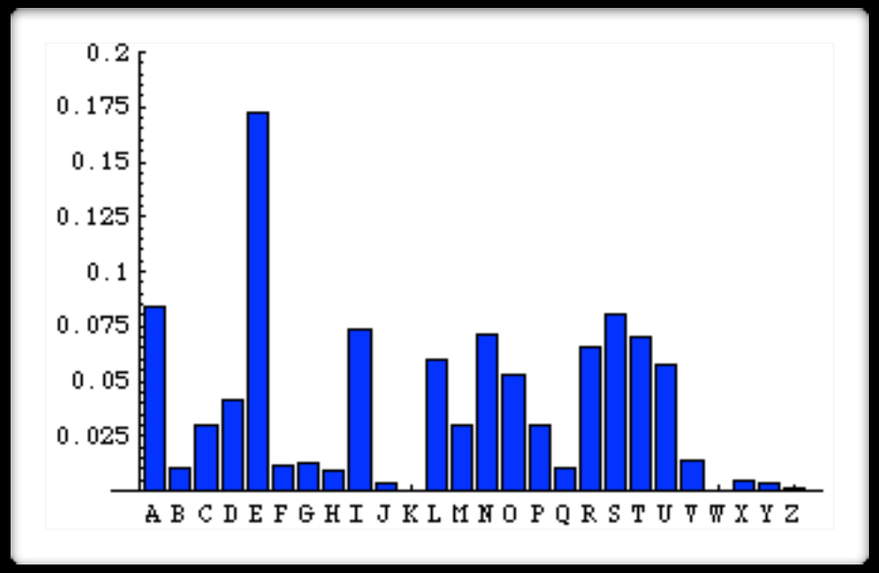
L’attaque exhaustive ou *force brute*

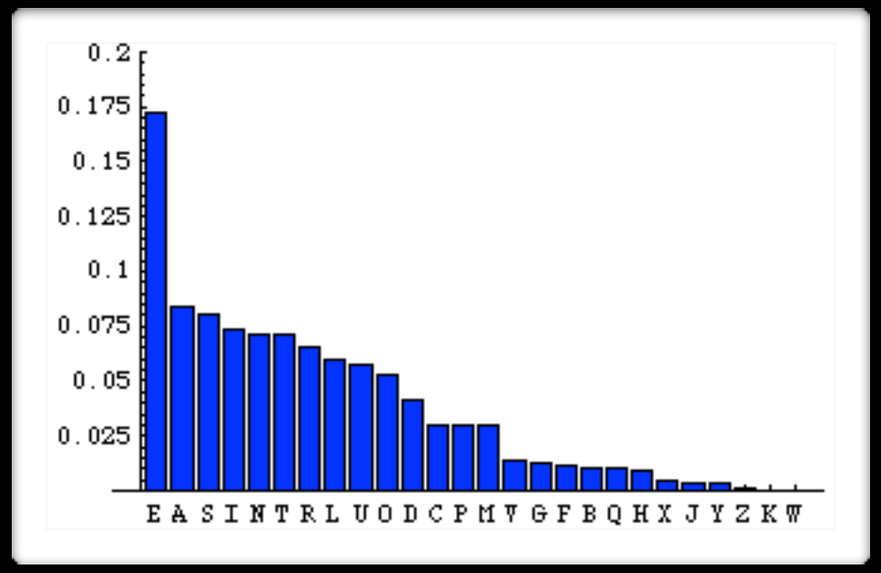
Elle consiste à tester toutes les clés possibles afin de retrouver le message clair.

- Dans le cas du transcodage, si on mélange les *n* symboles qui composent le message, il y a *n!* possibilités (mais toutes les permutations ne sont pas aussi bonnes). Mais dès que l'on connait un peu plus la règle de transcodage, le nombre de tests à réaliser chute (*n* dans le cas du mélange à la grecque).
- Dans le cas de la substitution mono-alphabétique, il faut a priori tester toutes les permutations de symboles $26!=2^{86}$ mais on peut supposer que le message devient lisible dès qu'on a bien placé la moitié des symboles (voire moins). Quand on connaît la règle de substitution, ce nombre chute (26 pour le codage de César,...)

L'attaque par analyse des fréquences

Elle s'adapte très bien au attaques sur les substitutions mono-alphabétiques.





Les fréquences des lettres du français sont plus ou moins fixes d'un texte à l'autre.

Il faut que le message codé soit suffisamment long pour que les fréquences aient un sens «probabiliste» (loi des grands nombres). Dans les autres cas, on n'obtient quand même une bonne indication.

Autres attaques classiques

Technique du mot probable (cribbing) : on suppose qu'on mot courant de la langue française (susceptible d'apparaître plusieurs fois dans le message) sera toujours représenté par la même suite de symbole.

Test de Friedman : Calcul d'un nombre qui permet de déterminer si un message est codé par substitution monoalphabétique ou polyalphabétique et donne une méthode pour décrypter les messages codés par le chiffre de Vigénère.

Méthode de Babbage/Kasiski : repérage des séquences de lettres qui se répètent dans le texte.

CRYPTOGRAPHIE MODERNE
CRYPTOGRAPHIE VS CRYPTANALYSE

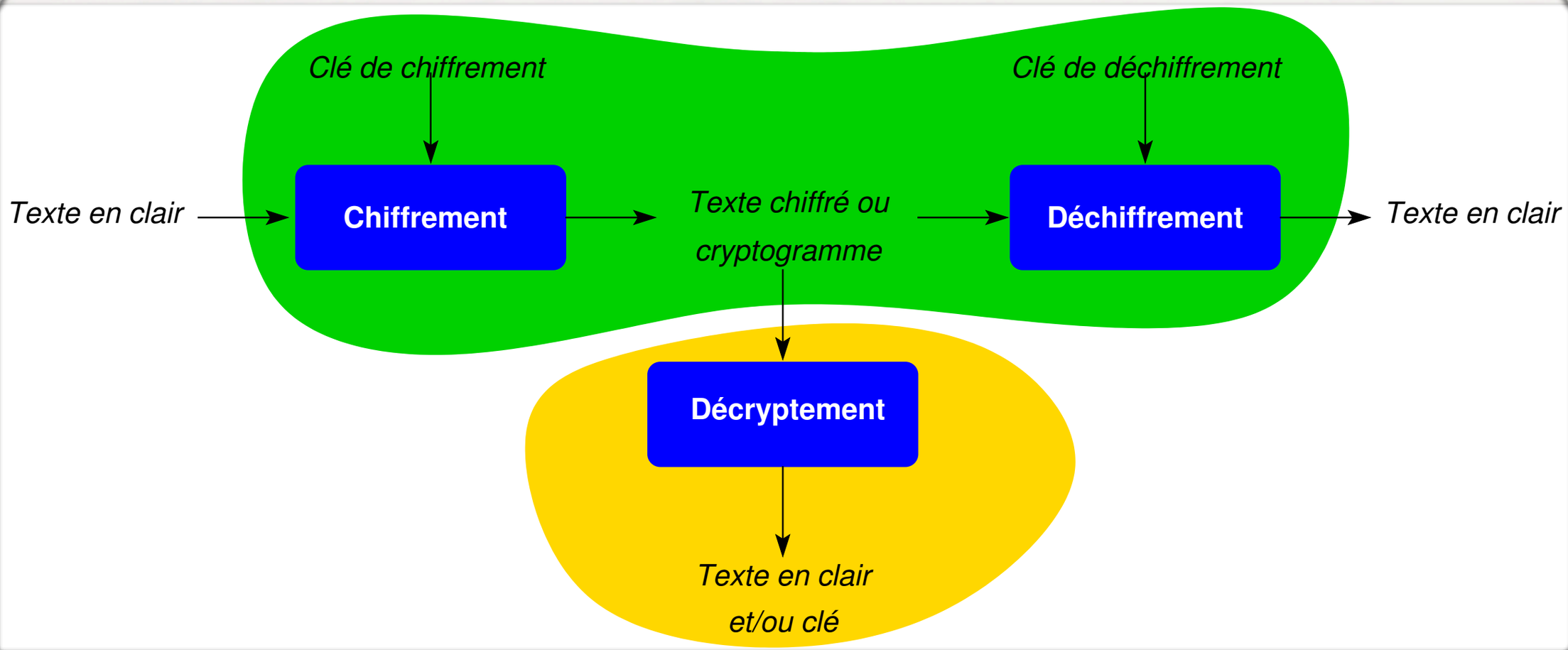
Cryptologie

La **cryptologie** est une science qui comporte deux branches:

La **cryptographie** est l'étude des méthodes qui permettent de transmettre des données de manière confidentielle. Elle consiste en la mise au point de procédés pour chiffrer et déchiffrer des messages.

La **cryptanalyse** est l'étude des procédés cryptographiques en vue de leurs trouver d'éventuelles faiblesses. Ceci afin de décrypter les messages codés sans connaître le secret qui permet de le faire facilement.

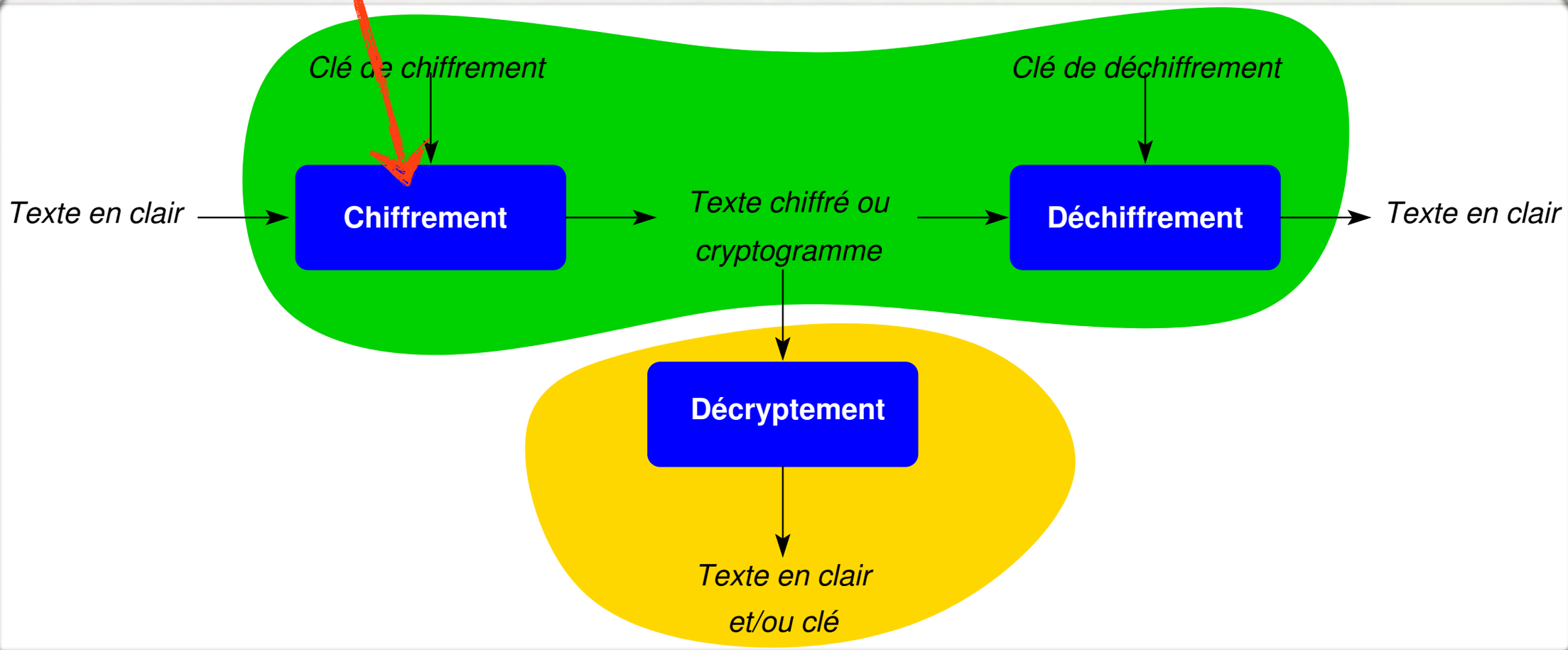
Cryptologie



facilement.

Cryptologie

fonction Chiffrement(clair,clé) : chaîne

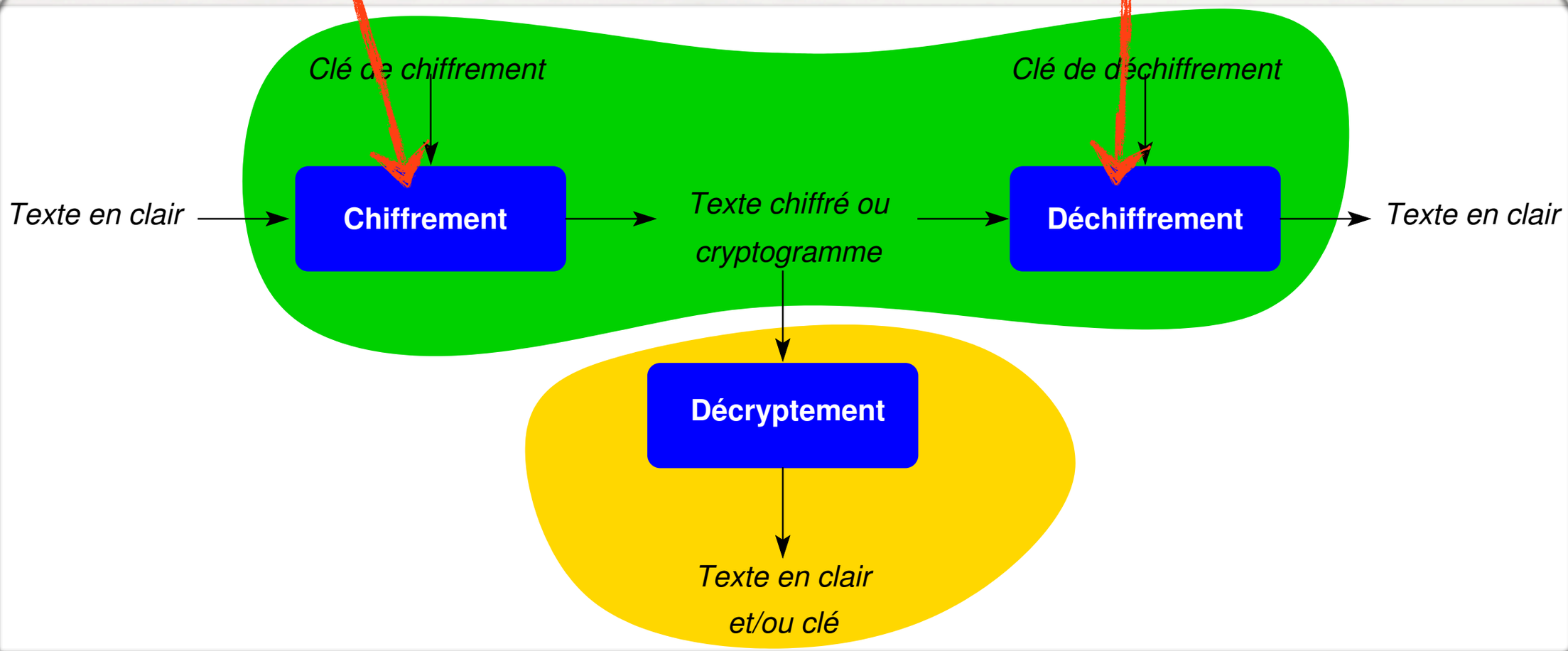


facilement.

fonction Dechiffrement(codé,clé) : chaîne

fonction Chiffrement(clair,clé) : chaîne

Cryptologie

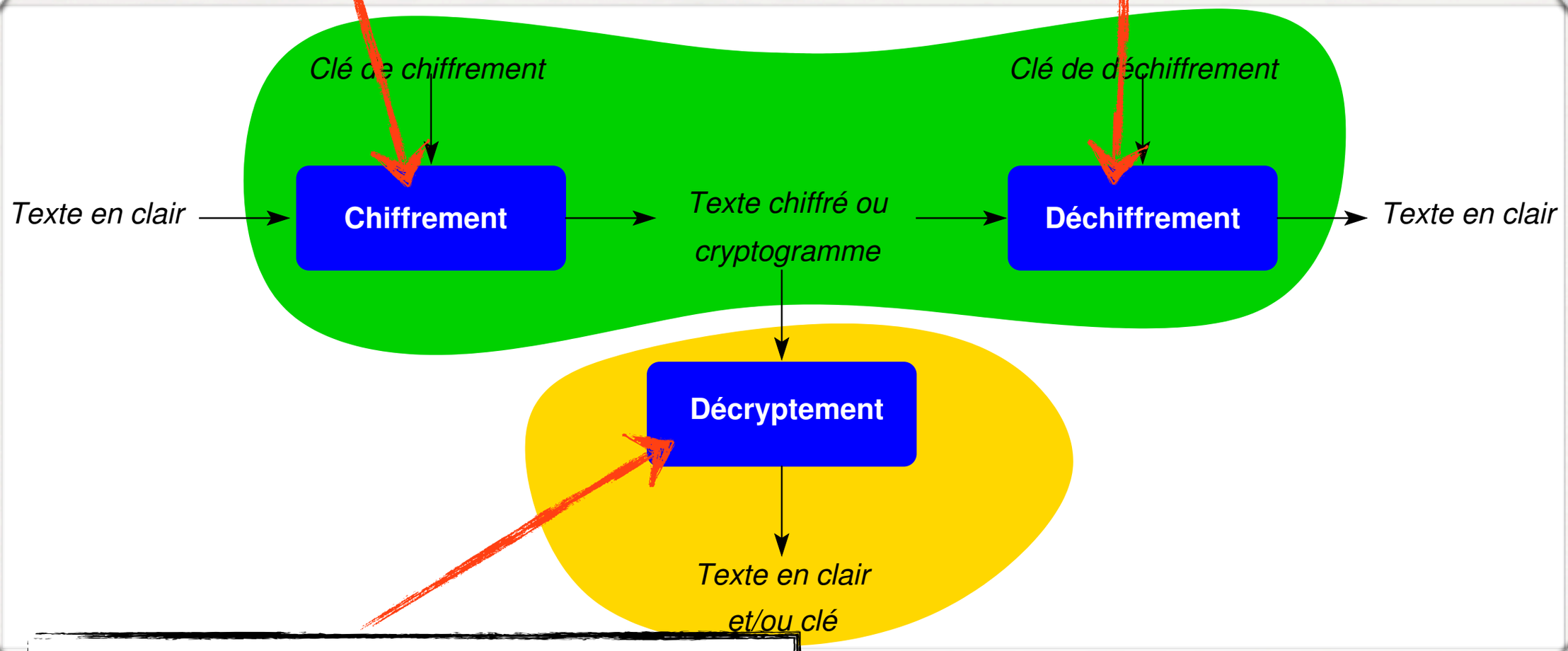


facilement.

Cryptologie

fonction Chiffrement(clair,clé) : chaîne

fonction Dechiffrement(codé,clé) : chaîne



fonction Decryptement(codé) : chaîne

Sécurité et chiffrement

Mécanismes et services de sécurité

But de la cryptographie moderne : fournir un certain nombre de **services de sécurité**

- Confidentialité
- Intégrité
- Authentification de l'origine des données ou d'un tiers
- Non-répudiation
- Preuves à apport nul de connaissance

Moyens mis en oeuvre : **mécanismes de sécurité** construits au moyen d'outils cryptographiques (fonctions, algorithmes, générateurs aléatoires, protocoles,...)

- Chiffrement
- Scellement et signature
- Protocoles d'authentification mutuelle avec échange de clés

Sécurité et chiffrement

Mécanismes et services de sécurité

But de la cryptographie moderne : fournir un certain nombre de **services de sécurité**

- Confi
- Intégr
- Authe
- Non-r
- Preuv

Les utilisations de la cryptographie mélangent

en général plusieurs services de sécurité qui sont

basés sur plusieurs mécanismes de sécurité !!!

Moyens mis en oeuvre : **mécanismes de sécurité** construits au moyen d'outils cryptographiques (fonctions, algorithmes, générateurs aléatoires, protocoles,...)

- Chiffrement
- Scellement et signature
- Protocoles d'authentification mutuelle avec échange de clés

Vocabulaire

- **Confidentialité** = Assurer que les données concernées ne pourront être dévoilées qu'aux personnes autorisées.
- **Intégrité** = Assurer que les données ne seront pas altérées (intensionnellement ou non) pendant leur transmission ou leur stockage.
- **Authentification/Identification** = Prouver l'origine d'une donnée ou l'identité d'une personne.
- **Signature (non-répudiation)** = Permet à une personne de prendre part à un contrat avec impossibilité de renier ensuite ses engagements.

Confidentialité et algorithmes de chiffrement

La confidentialité est le problème de base de la cryptographie, il se résout par la notion de chiffrement.

Deux types d'algorithmes:

- **Algorithmes symétriques ou à clé secrète**
  - Plus rapides donc préférés pour le chiffrement de données}
  - Les clés doivent résister à une attaque exhaustive (>56 bits, 128 bits).
- **Algorithmes asymétriques ou à clé publique**
  - Echange de clés secrètes
  - Signature
  - La clé publique donne une info supplémentaire pour casser la clé (>512 bits, 1024 bits).

Confidentialité et algorithmes de chiffrement

La confidentialité est le problème de base de la cryptographie, il se résout par la notion de chiffrement.

Deux types d'algorithmes:

- **Algorithmes symétriques ou à clé secrète**

  1000 personnes veulent se parler, il faut 499500 clés !!!

  1000000 personnes veulent se parler, il faut 500 milliards de clés !!!

- **Algorithmes asymétriques ou à clé publique**
  - Echange de clés secrètes
  - Signature
  - La clé publique donne une info supplémentaire pour casser la clé (>512 bits, 1024 bits).

Confidentialité et algorithmes de chiffrement

La confidentialité est le problème de base de la cryptographie, il se résout par la notion de chiffrement.

Deux types d'algorithmes:

- **Algorithmes symétriques ou à clé secrète**

1000 personnes veulent se parler, il faut 499500 clés !!!

1000000 personnes veulent se parler, il faut 500 milliards de clés !!!

- **Algorithmes asymétriques ou à clé publique**

1000 personnes veulent se parler, il faut 1000 clés !!!

1000000 personnes veulent se parler, il faut 1000000 de clés !!!

  - La clé publique donne une info supplémentaire pour casser la clé (>512 bits, 1024 bits).

Algorithmes symétriques (ou à clé privée)

Algorithmes symétriques (ou à clé privée)

Alice veut envoyer le message M à Bob

Alice

K

Message M

Bob

K

Algorithmes symétriques (ou à clé privée)

Alice calcule C=encode(M,K) et l'envoie à Bob

Algorithmes symétriques (ou à clé privée)

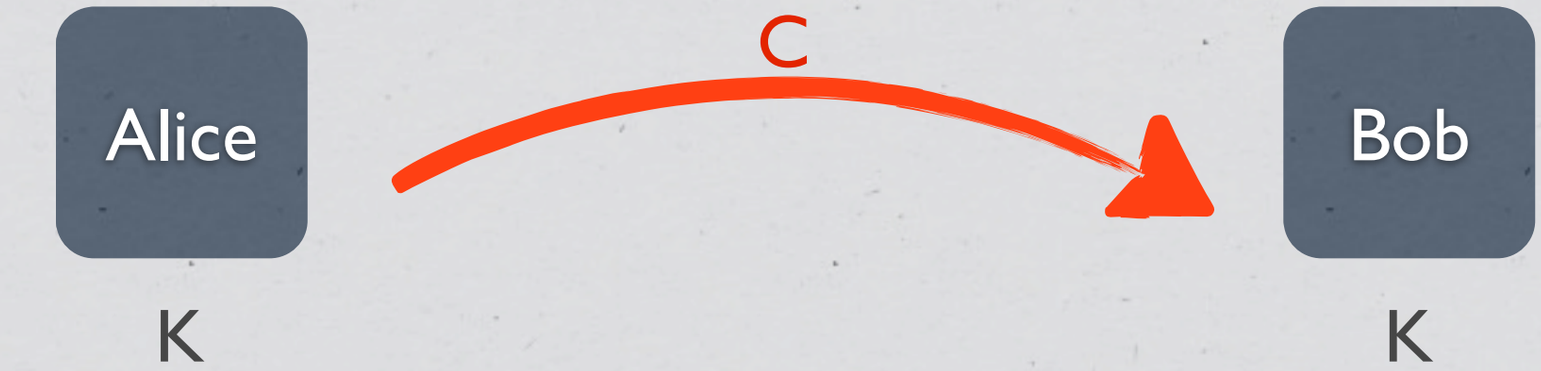
Bob calcule M=decode(C,K) et lit le message

Algorithmes symétriques (ou à clé privée)

Bob calcule M=decode(C,K) et lit le message

Algorithmes symétriques (ou à clé privée)

Bob calcule M=decode(C,K) et lit le message



Cryptosystèmes «utilisés»:

Avant 2000: DES, clés de 56 bits (chaîne de 7 caractères)

Depuis: AES, clés de 128 à 256bits (chaînes de 16 ou 32 caractères)

Algorithmes symétriques (ou à clé privée)

Si on est capable de tester 1000000 de clés par secondes pour DES

Il faut 2200 ans pour décoder le message....

MAIS si on branche 10000 ordinateurs qui testent chacun

une partie des clés, il ne faut que 80 jours !!!

Message ??

Cryptosystèmes «utilisés»:

Avant 2000: DES, clés de 56 bits (chaîne de 7 caractères)

Depuis: AES, clés de 128 à 256bits (chaînes de 16 ou 32 caractères)

S’échanger une clé protocole de Diffie-Hellman

Alice

Bob

S'échanger une clé protocole de Diffie-Hellman

Alice et Bob se mettent d'accord sur deux nombres
p grand nombre premier et g nombre inférieur à p

Alice

p,g

Bob

p,g

S'échanger une clé protocole de Diffie-Hellman

Alice choisit un nombre secret a et calcule
$A=g^a$ modulo p

Alice

p,g,a,A

Bob

p,g

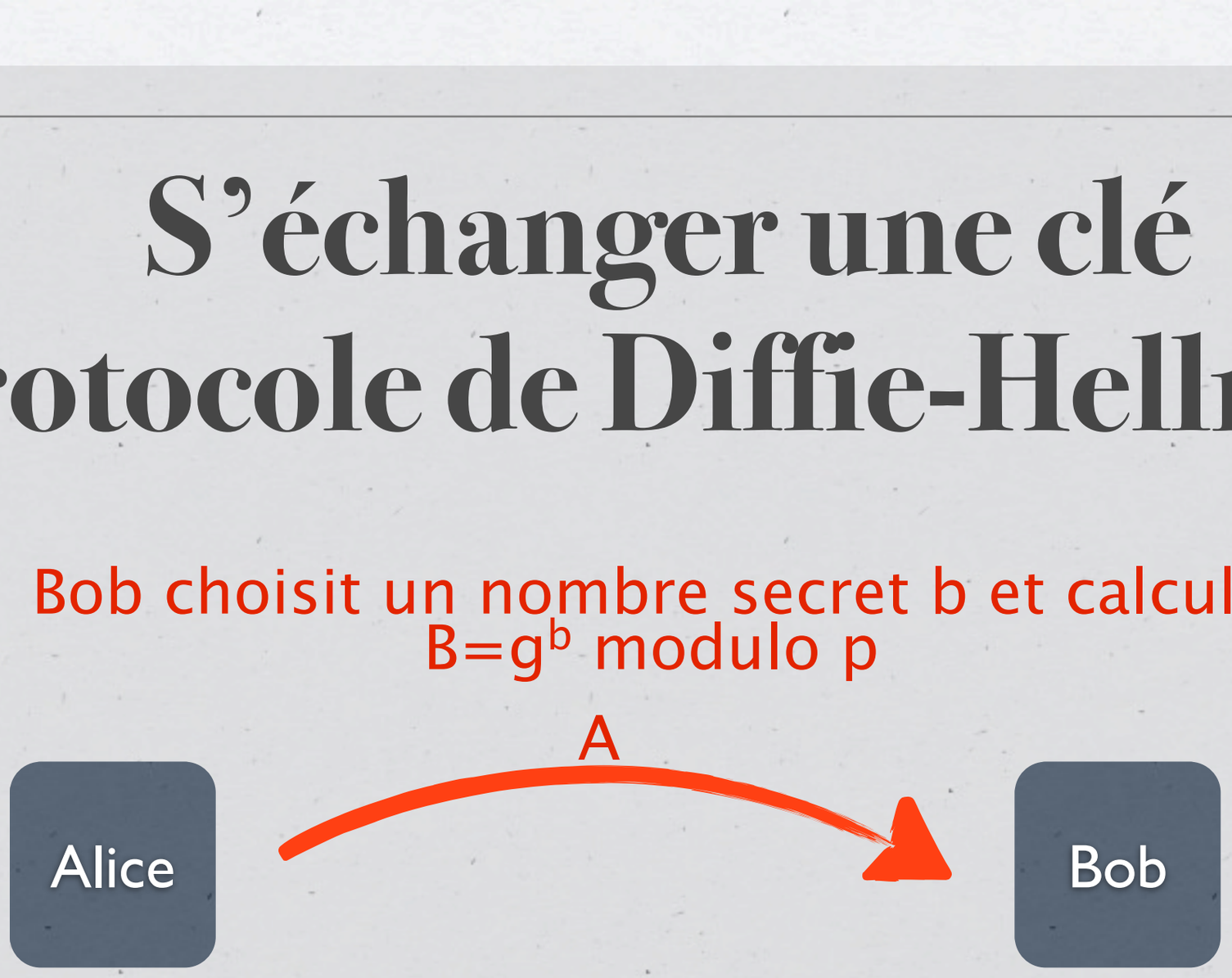
S'échanger une clé protocole de Diffie-Hellman

Alice envoie A à Bob, a reste secret

S'échanger une clé protocole de Diffie-Hellman

Bob choisit un nombre secret b et calcule
$B=g^b$ modulo p

A

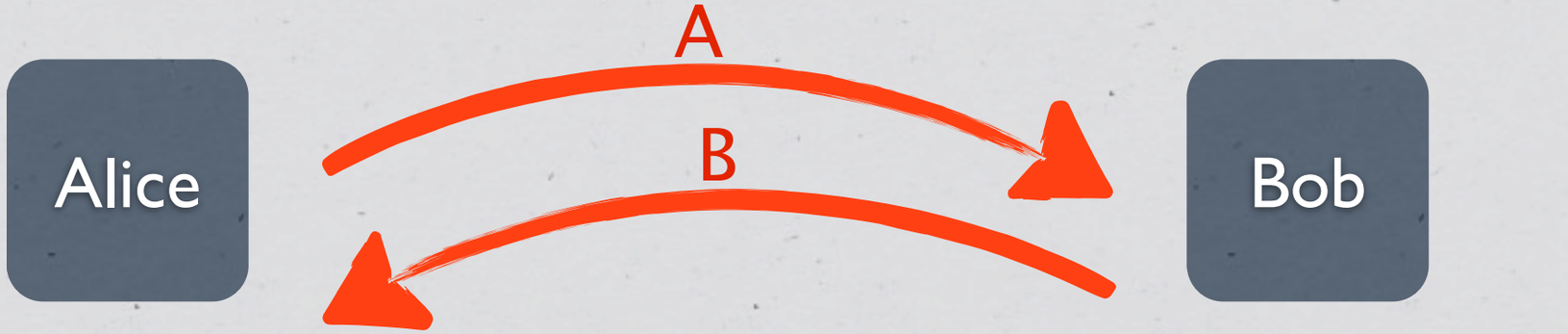


p,g,a,A

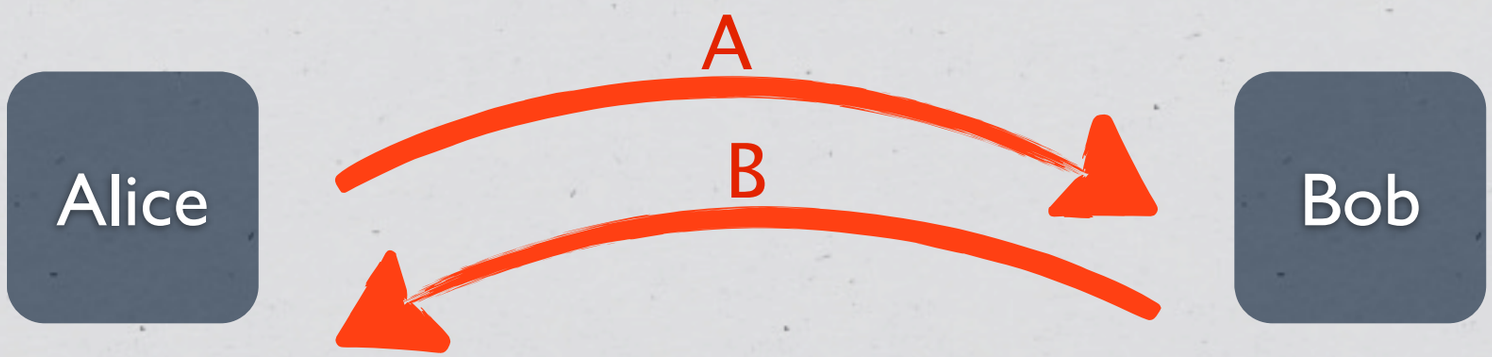
p,g,b,B,A

S'échanger une clé protocole de Diffie-Hellman

Bob envoie B à Alice, b reste secret



S'échanger une clé
protocole de Diffie-Hellman



La clé est $K=g^{ab}$ modulo p
Alice la calcule (sans connaître b) car $K=B^a$ modulo p
Bob la calcule (sans connaître a) car $K=A^b$ modulo p
Il est matériellement impossible (si p grand) de calculer K
si on ne connaît pas a et b

S'échanger une clé protocole de Diffie-Hellman

Exercice: l'attaque de l'homme au milieu. Imaginons que Colin se place au milieu entre toutes les conversations entre Alice et Bob (il se fait passer pour Bob vis-a-vis d'Alice et pour Alice vis-a-vis de Bob. Quelles opérations Colin peut-il faire pour pouvoir écouter toutes les conversations chiffrées qui suivront cet échange de clé ?

La clé est K=g^ab modulo p
Alice la calcule (sans connaître b) car $K=B^a$ modulo p
Bob la calcule (sans connaître a) car $K=A^b$ modulo p
Il est matériellement impossible (si p grand) de calculer K si on ne connaît pas a et b

Algorithmes asymétriques (ou à clé publique)

Alice

(e_A, d_A, n_A)

Bob

(e_B, d_B, n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire

Alice: (e_A, n_A)

Bob: (e_B, n_B)

Alice

(e_A, d_A, n_A)

Bob

(e_B, d_B, n_B)

Avancé

Général | Onglets | Contenu | Applications | Vie privée | Sécurité | Sync | Avancé

Général | Réseau | Mises à jour | Chiffrement

Protocoles

☑ Utiliser SSL 3.0 ☑ Utiliser TLS 1.0

Certi

Lorsq

Af

Pé

Gestionnaire de certificats

Vos certificats | Personnes | Serveurs | Autorités | Autres

Vous possédez des certificats enregistrés identifiant ces autorités de certification :

Nom du certificat	Périphérique de sécurité
▼ VeriSign, Inc.	
Thawte SGC CA	Sécurité personnelle
Thawte SGC CA - G2	Sécurité personnelle
VeriSign, Inc.	Sécurité personnelle
VeriSign, Inc.	Sécurité personnelle
VeriSign Class 3 Public Primary Certification Authority - G5	Builtin Object Token
Verisign Class 3 Public Primary Certification Authority	Builtin Object Token
Verisign Class 1 Public Primary Certification Authority	Builtin Object Token
Verisign Class 2 Public Primary Certification Authority	Builtin Object Token

Voir… | Modifier la confiance… | Importer… | Exporter… | Supprimer ou ne plus faire confiance…

?

Av

Alice

Bob

Algorithmes asymétriques (ou à clé publique)

Annuaire

Alice: (e_A, n_A)

Bob: (e_B, n_B)

Alice

(e_A, d_A, n_A)

Bob

(e_B, d_B, n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice veut parler à Bob, elle récupère (e_B,n_B)

Alice

(e_A,d_A,n_A)

Bob

(e_B,d_B,n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice veut parler à Bob, elle récupère (e_B,n_B)

Alice

(e_A,d_A,n_A)

Message clair M

Bob

(e_B,d_B,n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice calcule $C=F(M,(e_B,n_B))$ et l'envoie à Bob

Alice

(e_A,d_A,n_A)

Message clair M

Bob

(e_B,d_B,n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice calcule $C=F(M,(e_B,n_B))$ et l'envoie à Bob

Alice

(e_A,d_A,n_A)

Bob

(e_B,d_B,n_B)

En fait F(M,(e,n)) est une fonction dite à trappe, c'est à dire qu'il est facile de calculer l'inverse de F quand on connaît la trappe (ici d) mais très difficile quand on ne la connaît pas

ex (RSA): $C=F(M,(e,n))=M^e$ *modulo n*, avec $ed=1$ *modulo* $\varphi(n)$

connaissant C,e,n, impossible de retrouver M, mais si on connaît d,

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice calcule $C=F(M,(e_B,n_B))$ et l'envoie à Bob

Alice

(e_A,d_A,n_A)

Message clair M

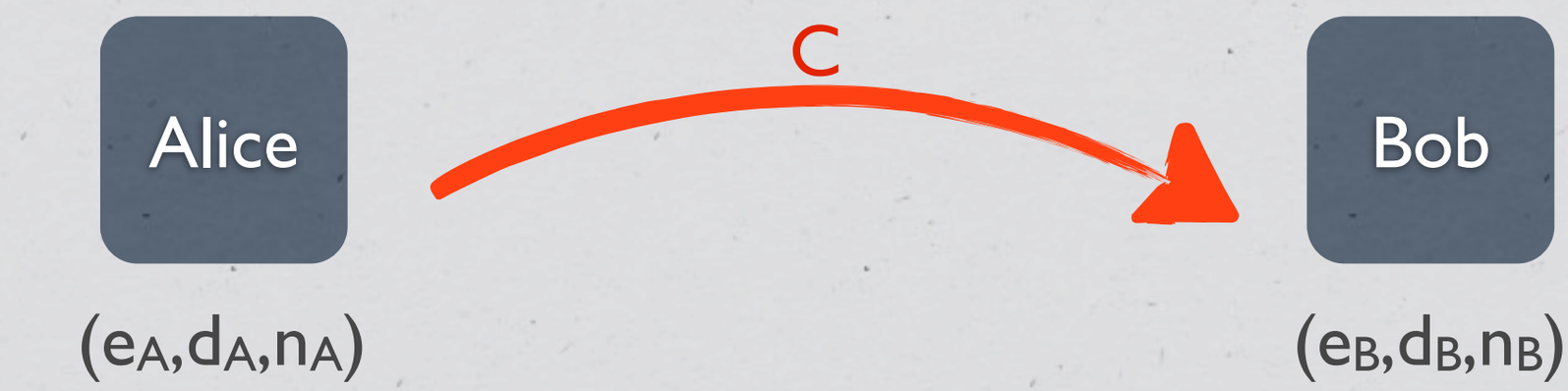
Bob

(e_B,d_B,n_B)

Algorithmes asymétriques (ou à clé publique)

Annuaire
Alice: (e_A,n_A)
Bob: (e_B,n_B)

Alice calcule $C=F(M,(e_B,n_B))$ et l'envoie à Bob



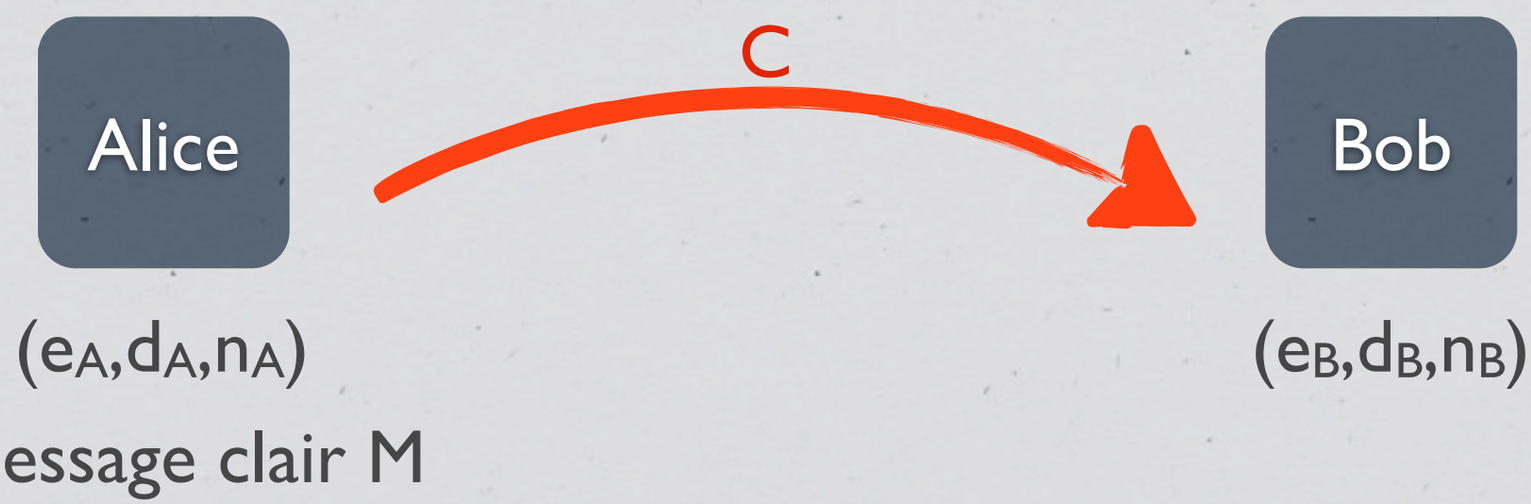
Algorithmes asymétriques (ou à clé publique)

Annuaire

Alice: (e_A,n_A)

Bob: (e_B,n_B)

Bob calcule $M=F^{-1}(C,(d_B,n_B))$ et lit le message !



RSA générer (e,d,n)

Comment générer un triplet valide et qui assure un bon niveau de sécurité ?

1.Tirer au hasard deux très grand nombres premiers p et q (100 à 200 chiffres).
On pose $n=pq$ et $\varphi(n)=(p-1)(q-1)$

2.Choisir au hasard e (petit entier tel que $PGCD(e,\varphi(n))=1$).

3.Calculer d tel que $ed=1$ *modulo* $\varphi(n)$ [facile: algorithme d'Euclide généralisé]

Toute la sécurité de RSA repose sur le fait qu'il est très difficile de retrouver p et q à partir de n (problème de la factorisation de grands entiers, pas d'autre méthode que d'essayer tous les nombres, donc matériellement difficile si p et q sont grands 2^{128})

RSA générer (e,d,n)

Comment générer un triplet valide et qui assure un bon niveau de sécurité ?

1.Ti
On

2.C

Si on est capable de tester 1000000 de nombres par secondes

Il faut 10^{23} siècles pour factoriser n

3.Calculer *d* tel que *ed=1 modulo* $\varphi(n)$ [facile: algorithme d'Euclide généralisé]

Toute la sécurité de RSA repose sur le fait qu'il est très difficile de retrouver p et q à partir de n (problème de la factorisation de grands entiers, pas d'autre méthode que d'essayer tous les nombres, donc matériellement difficile si p et q sont grands 2^{128})

Compétition de factorisation RSA

La **compétition de factorisation RSA** fut une compétition mise en avant par la société RSA Security jusqu'en mai 2007. Mise en place le 18 mars 1991, son but était d'encourager la recherche dans la théorie calculatoire des nombres et dans la difficulté pratique de la mise en facteurs de grands entiers. Ils publièrent une liste de nombres semi-premiers connus comme les nombres RSA dotés d'une récompense financière pour les factorisations réussies pour certains d'entre eux. Le plus petit d'entre eux, un nombre à 100 chiffres décimaux, appelé RSA-100, fut factorisé en quelques jours, mais beaucoup de nombres plus grands n'ont pas encore été factorisés et sont supposés le rester pendant encore plusieurs dizaines d'années.

Sommaire [masquer]

Il faut 10^{25} siècles pour factoriser n

2.C

3.Calculer *d* tel que *ed=1 modulo* $\varphi(n)$ [facile: algorithme d'Euclide généralisé]

Toute la sécurité de RSA repose sur le fait qu'il est très difficile de retrouver p et q à partir de n (problème de la factorisation de grands entiers, pas d'autre méthode que d'essayer tous les nombres, donc matériellement difficile si p et q sont grands 2^{128})

Compétition de factorisation RSA

La **compétition de factorisation RSA** fut une compétition mise en avant par la société RSA Security jusqu'en mai 2007. Mise en place le 18 mars 1991, son but était d'encourager la recherche dans la théorie calculatoire des nombres et dans la difficulté pratique de la mise en facteurs de grands entiers. Ils publièrent une liste de nombres semi-premiers connus comme les nombres RSA dotés d'une récompense financière pour les factorisations réussies pour certains d'entre eux. Le plus petit d'entre eux, un nombre à 100 chiffres décimaux, appelé RSA-100, fut factorisé en quelques jours, mais beaucoup de nombres plus grands n'ont pas encore été factorisés et sont supposés le rester pendant encore plusieurs dizaines d'années.

Sommaire [masquer]

Utilité [modifier]

Cette compétition n'était pas seulement intéressante du point de vue de la théorie des nombres, mais aussi d'un sens très pratique — comme trouver une solution est plus ou moins la même chose que de casser une clé publique RSA. L'algorithme de clé publique RSA est une pierre angulaire de beaucoup de protocoles cryptologiques — incluant certains utilisés par les systèmes financiers. Les progrès de cette compétition donnaient une indication pour savoir la taille des clés encore sûres, et pour combien de temps. Comme les laboratoires RSA sont un fournisseur de produits basé sur l'algorithme RSA, la compétition était utilisée par eux comme un stimulant pour la communauté pour attaquer le noyau de leurs solutions — entre autres pour prouver sa force.

Il faut 10^{25} siècles pour factoriser n

2 C

isé]

t q à
le que
128)

Compétition de factorisation RSA

La **compétition de factorisation RSA** fut une compétition mise en avant par la société RSA Security jusqu'en mai 2007. Mise en place le 18 mars 1991, son but était d'encourager la recherche dans la théorie calculatoire des nombres et dans la difficulté pratique de la mise en facteurs de grands entiers. Ils publièrent une liste de nombres semi-premiers connus comme les nombres RSA dotés d'une récompense [illegible] eux, un nomb[illegible] beaucoup de [illegible] encore plusie[illegible]

Compétition	Prix	Statut	Date de factorisation	Par
RSA-576	USD 10 000	Factorisé	3 décembre 2003	J. Franke et al.
RSA-640	USD 20 000	Factorisé	2 novembre 2005	F. Bahr et al.
RSA-704	USD 30 000	Annulé	-	-
RSA-768	USD 50 000	Factorisé	15 janvier 2010	Divers organismes
RSA-896	USD 75 000	Annulé	-	-
RSA-1024	USD 100 000	Annulé	-	-
RSA-1536	USD 150 000	Annulé	-	-
RSA-2048	USD 200 000	Annulé	-	-

Sommaire

2 C[illegible]

Utilité [mo[illegible]

Cette compé[illegible] aussi d'un se[illegible] casser une c[illegible] de protocoles[illegible] cette compét[illegible] de temps. Co[illegible] compétition était utilisée par eux comme un stimulant pour la communauté pour attaquer le noyau de [illegible]128)
leurs solutions — entre autres pour prouver sa force.

Signature numérique

Cahier des charges

* calculable par le signataire quel que soit le message M (doit dépendre du signataire et de M)
* tout individu tout pouvoir vérifier la signature
* elle doit être infalsifiable
* l'expéditeur ne peut pas dire que sa signature a été imitée.

Un exemple de mise en œuvre, signature par RSA

* Alice veut signer un message (rappel ses clés sont (e_A,d_A,n_A))
* Alice calcule:

$$S=Signature(M, (d_A,n_A))=M^{d_A} \text{ modulo } n_A$$

* pour vérifier la signature (sur le message qui a été transmit à part par Alice à Bob)

$$ver(M,S, (e_A,n_A)) = vrai \Leftrightarrow S^{e_A} = M \text{ modulo } n_A$$

Conclusion

Tout le reste va reposer sur une suite de protocoles

- ✓ Comment envoyer un message très long (découpage par blocs) ?
- ✓ Comment bien partager un secret ?
- ✓ Comment générer du bon (pseudo-)aléatoire ?
- ✓ Comment signer un message ?

Avec des problèmes algorithmiques à résoudre

- ✓ Comment générer des grands nombres premiers ?
- ✓ Comment trouver et calculer rapidement des fonctions à trappe, dont celle de RSA ?
- ✓ etc,etc,...

Conclusion

Les craintes autour de la sécurité. Que croire ?

- des primitives de cryptographie **fiables** (crypter signer)

- des protocoles qui gèrent des services (ex: vote électronique), en général il y a des preuves «mathématiques» de bon fonctionnement et de résistance aux attaques connues (respect d'un cahier des charges). **~fiables**

- des machines «commerciales» basées sur ces protocoles. Difficile d'en vérifier le fonctionnement sauf en épluchant le code (qui est souvent un secret d'entreprise donc non diffusé). **Pas fiable**

✓Comment trouver et calculer rapidement des fonctions à trappe, dont celle de RSA ?

✓etc,etc,...

Quelques exercices

* Ecrire la fonction qui calcule le pgcd de deux nombres par l'algorithme d'Euclide.
* Soient p=5 et q=7. calculer n, $\varphi(n)$. On choisit e=13, calculer d.
* Ecrire l'algorithme d'exponentiation modulaire rapide