Cryptographie et réseaux euclidiens

Alice Pellet--Mary

CNRS et université de Bordeaux

Maths, science et société

12 octobre 2022

université de BORDEAUX

Qu'est-ce que la cryptographie ?

Cryptographie $\approx$ science des codes secrets

Qu'est-ce que la cryptographie ?

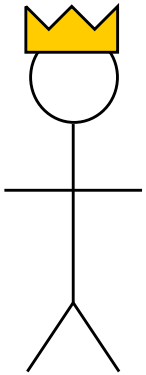
Cryptographie $\approx$ science des codes secrets

Applications

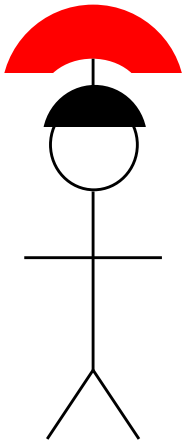
- Espionnage, armée...
- Payements sécurisés sur Internet, communications par mail...
- Vote électronique.
- Faire des calculs sur des données sensibles. (e.g., données médicales)
- ...

Chiffrement de César

Contexte (fictif)

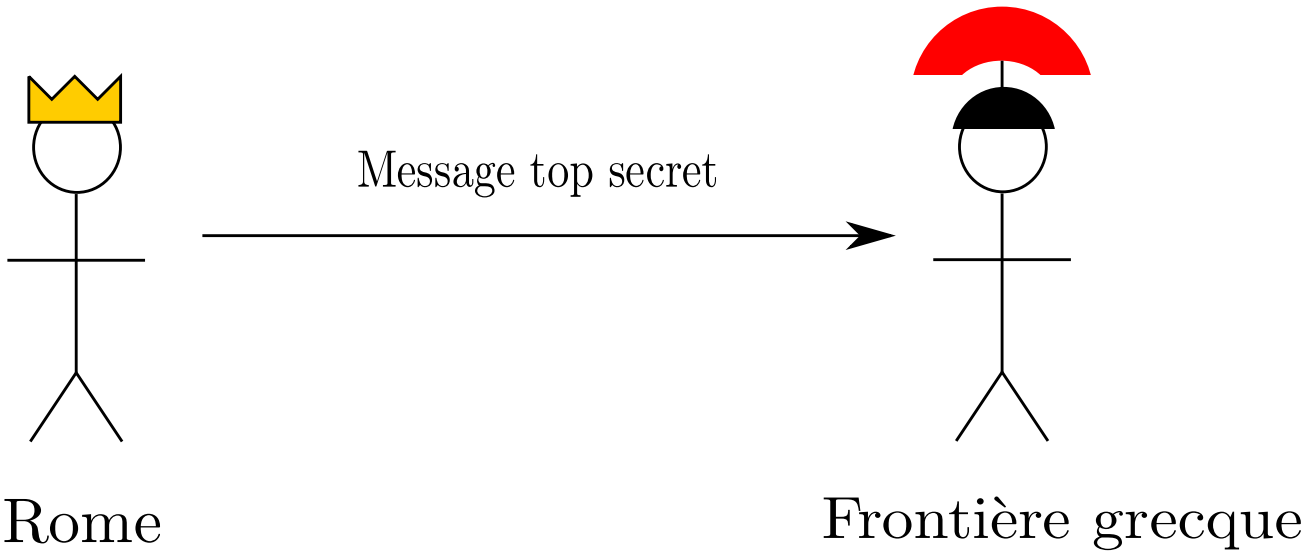


Rome

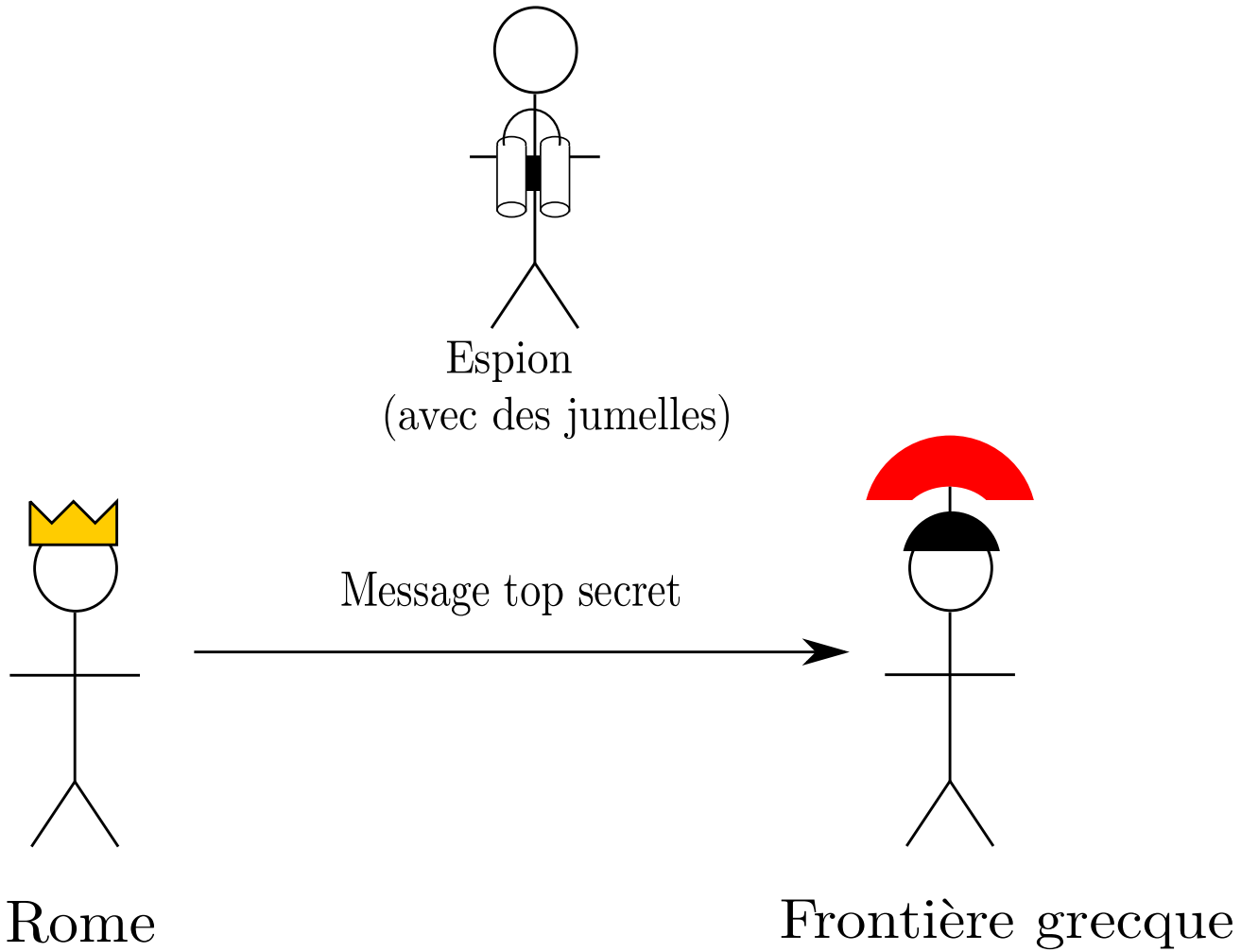


Frontière grecque

Contexte (fictif)



Contexte (fictif)



Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Exemple :

A v e C e s a r

Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Exemple :

A v e C e s a r
D

Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Exemple :

A v e C e s a r
D y

Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Exemple :

A v e C e s a r
D y h

Chiffrement de César

Idée

Décaler toutes les lettres de l'alphabet de 3.

Lettre d'origine	a	b	c	d	e	f	g	h	i	j	k	l	m
Lettre après décalage	d	e	f	g	h	i	j	k	l	m	n	o	p

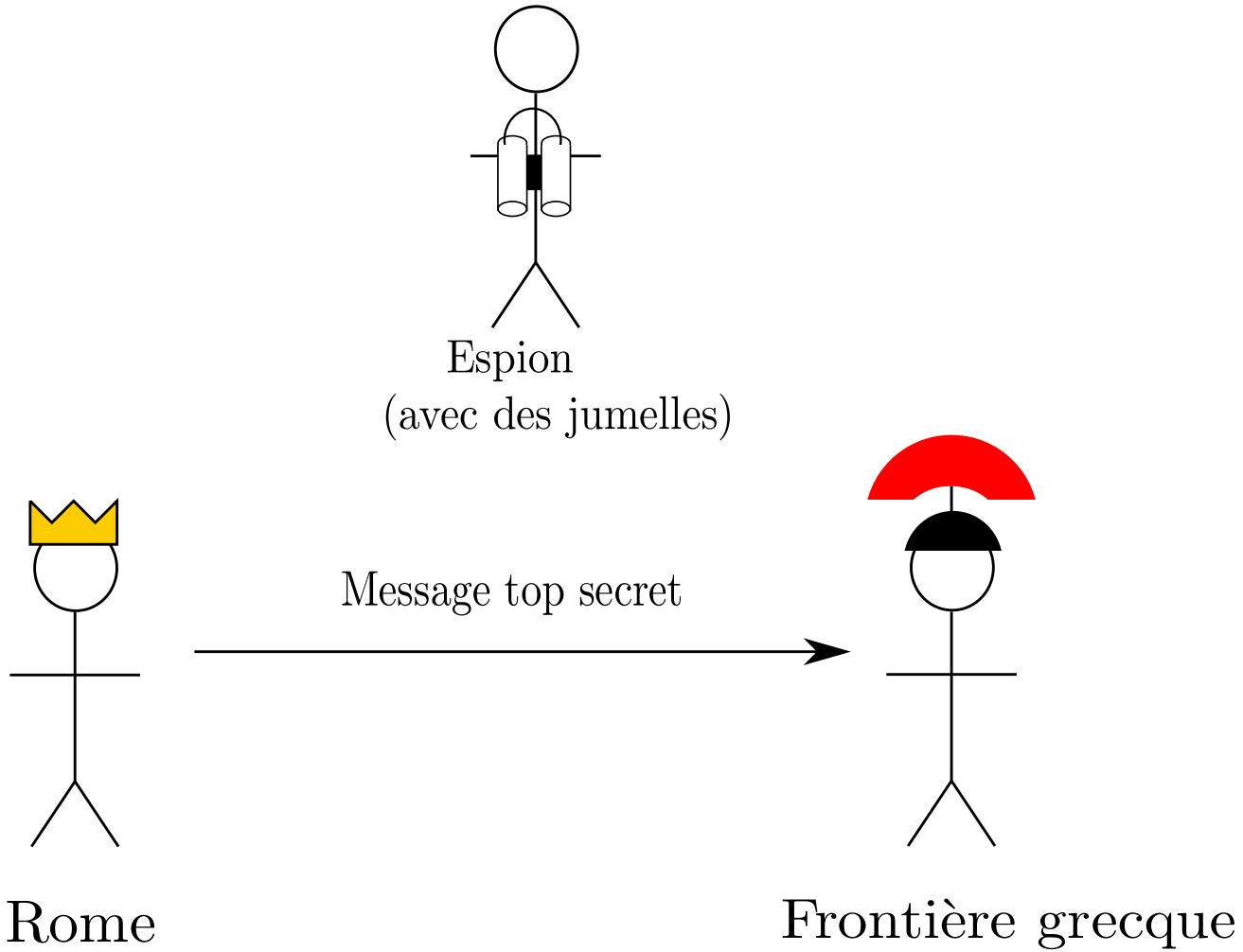
Lettre d'origine	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettre après décalage	q	r	s	t	u	v	w	x	y	z	a	b	c

Exemple :

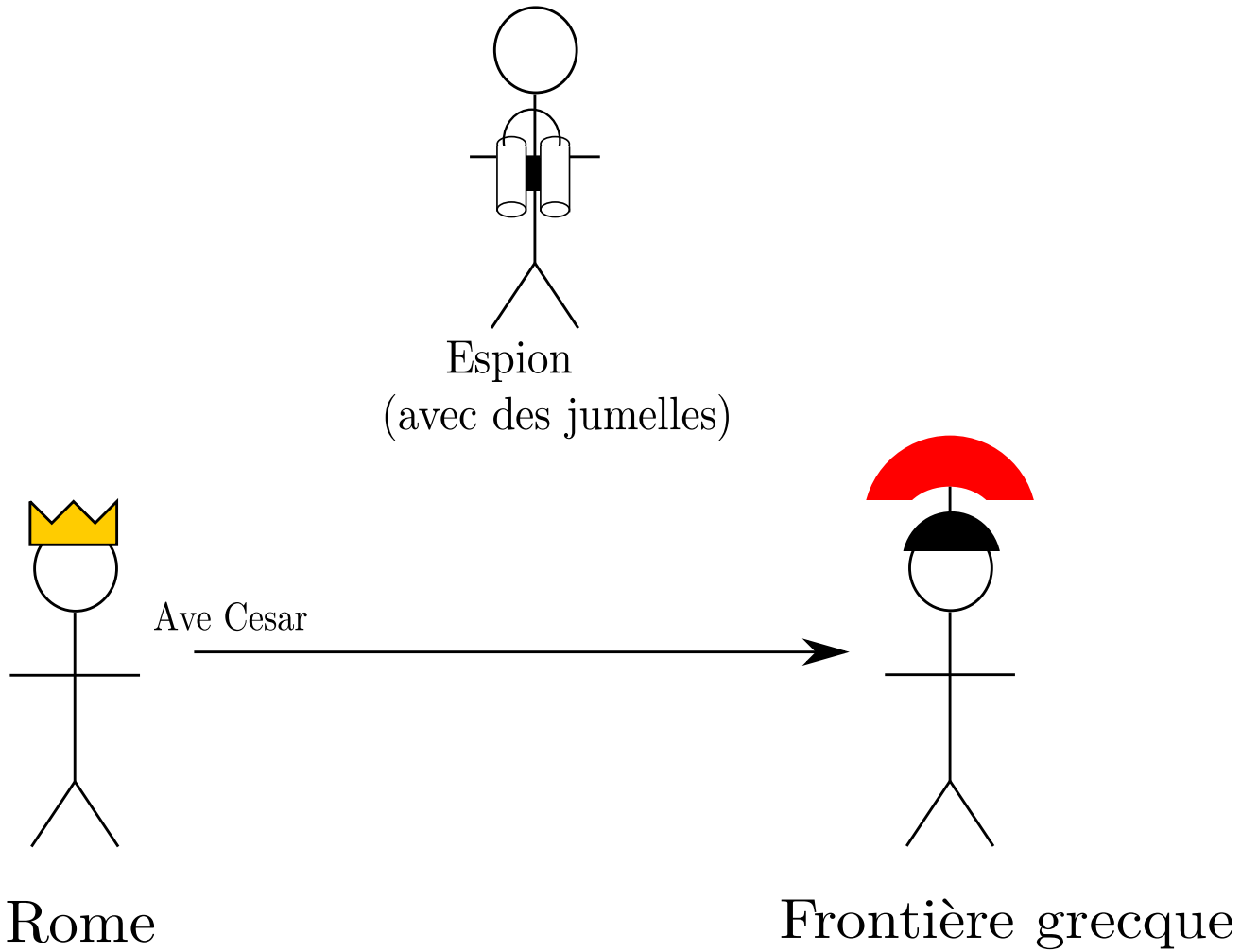
A v e C e s a r

D y h F h v d u

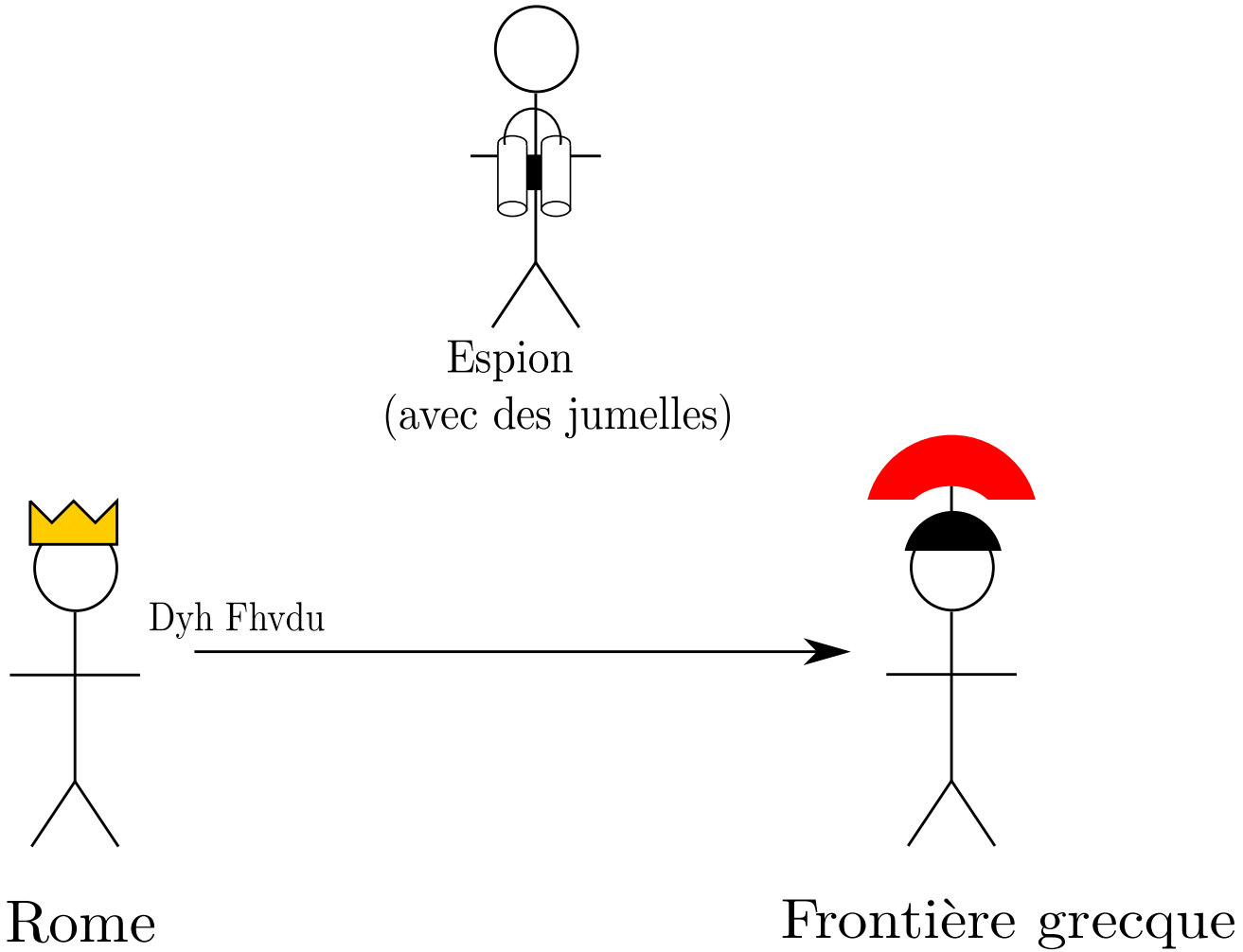
Retour à l'histoire



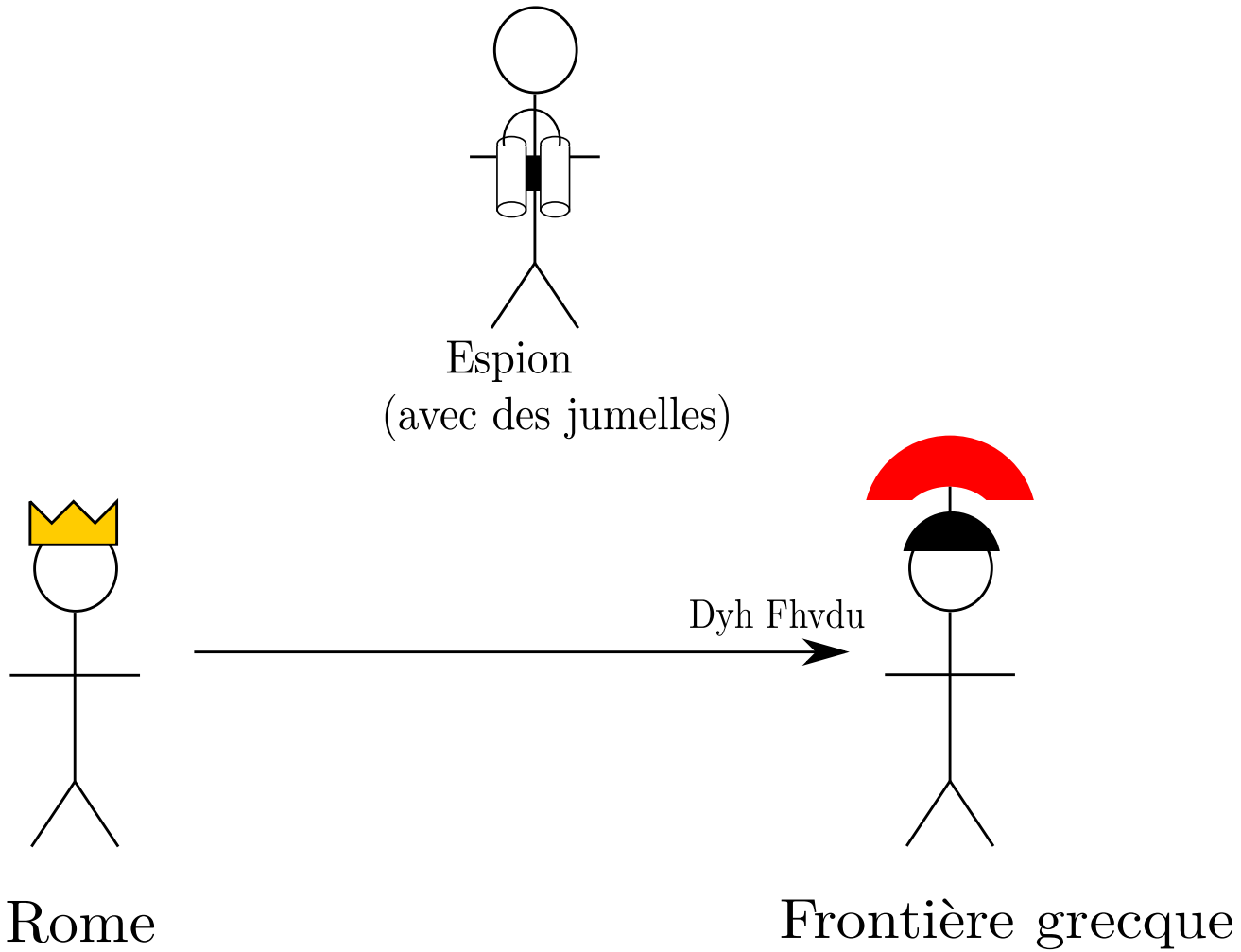
Retour à l'histoire



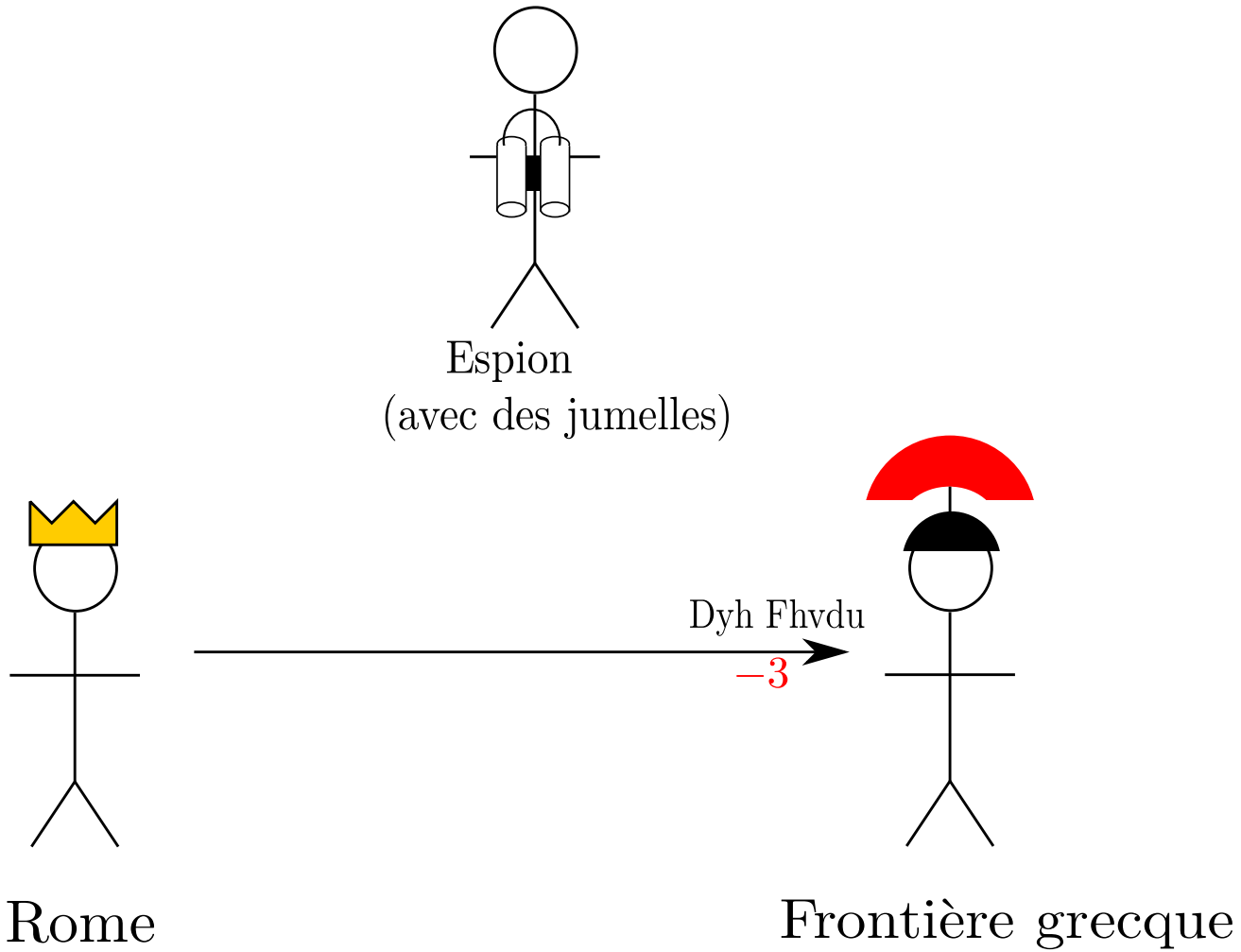
Retour à l'histoire



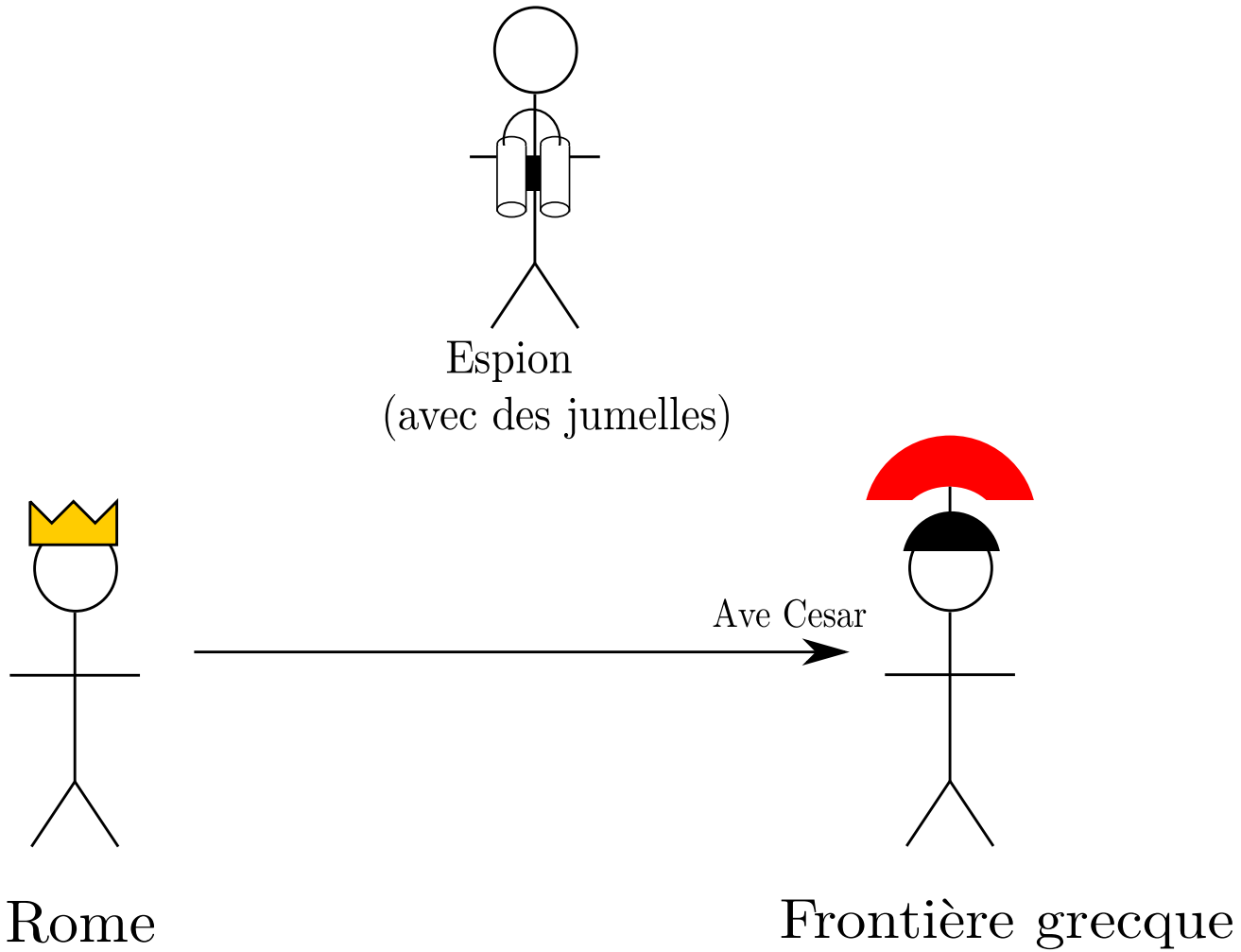
Retour à l'histoire



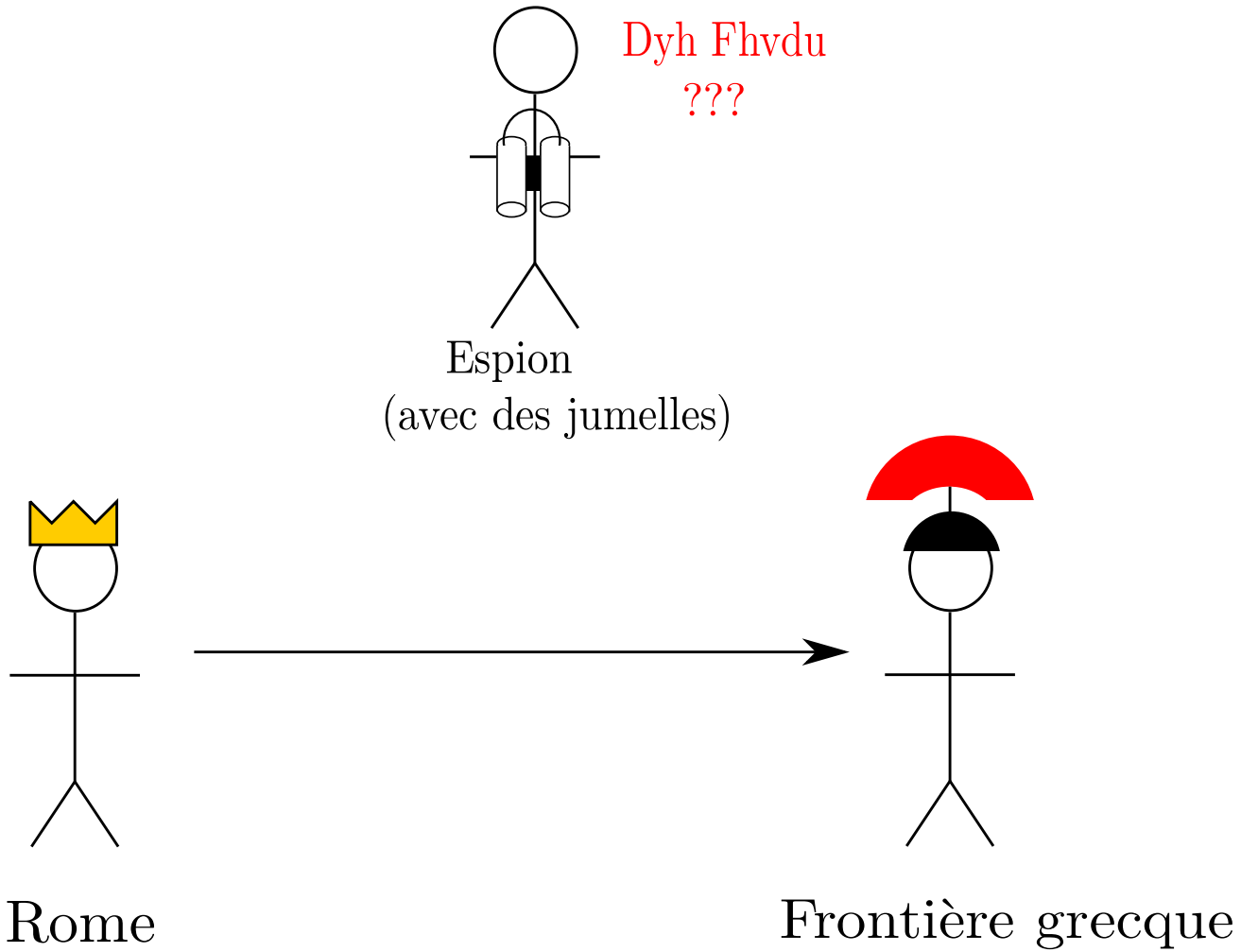
Retour à l'histoire



Retour à l'histoire



Retour à l'histoire



Oui mais...

Problème

Et si un jour un espion grec découvre le principe ? Faut-il inventer un nouveau code secret ?

Oui mais...

Problème

Et si un jour un espion grec découvre le principe ? Faut-il inventer un nouveau code secret ?

Non. On peut simplement changer le décalage. Par exemple, décaler de 7 lettres au lieu de 3.

Oui mais...

Problème

Et si un jour un espion grec découvre le principe ? Faut-il inventer un nouveau code secret ?

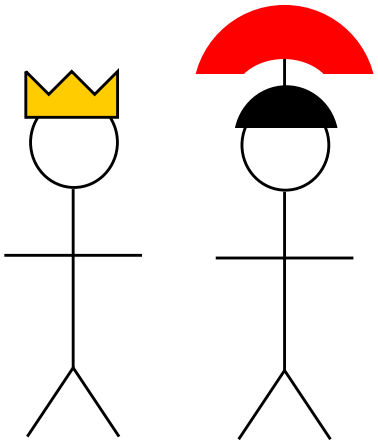
Non. On peut simplement changer le décalage. Par exemple, décaler de 7 lettres au lieu de 3.

Nouveau protocole

César et son général choisissent un entier k entre 1 et 25. Cet entier k est appelé *clé secrète*. Ils décalent ensuite les lettres de k positions pour chiffrer leurs messages.

Même si les Grecs connaissent le protocole, tant qu'ils ne connaissent pas la clé secrète k, ils ne peuvent pas déchiffrer les messages chiffrés.

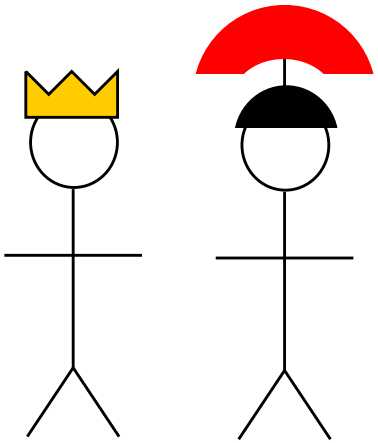
Retour à l'histoire



Rome

Frontière grecque

Retour à l'histoire

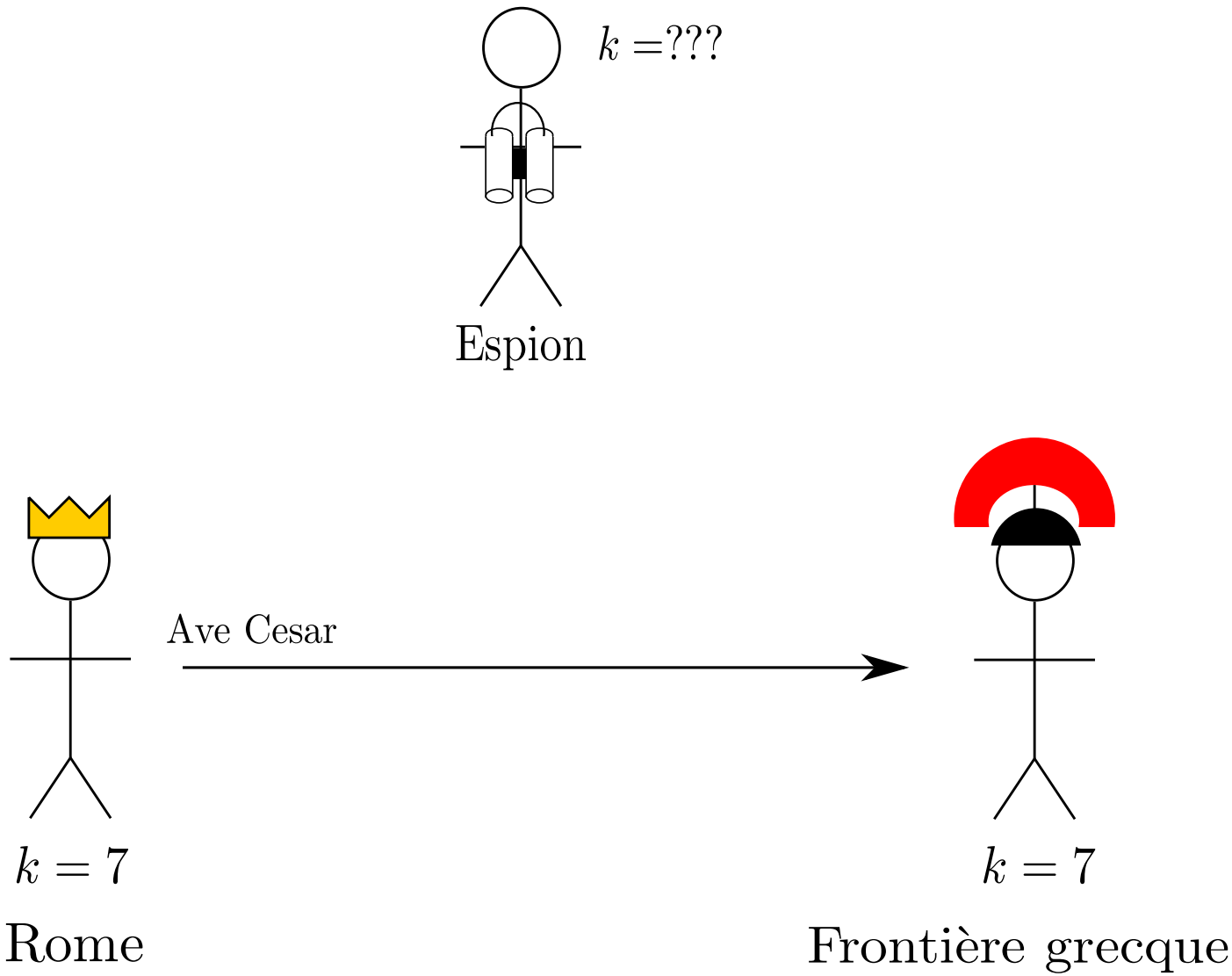


$k = 7$

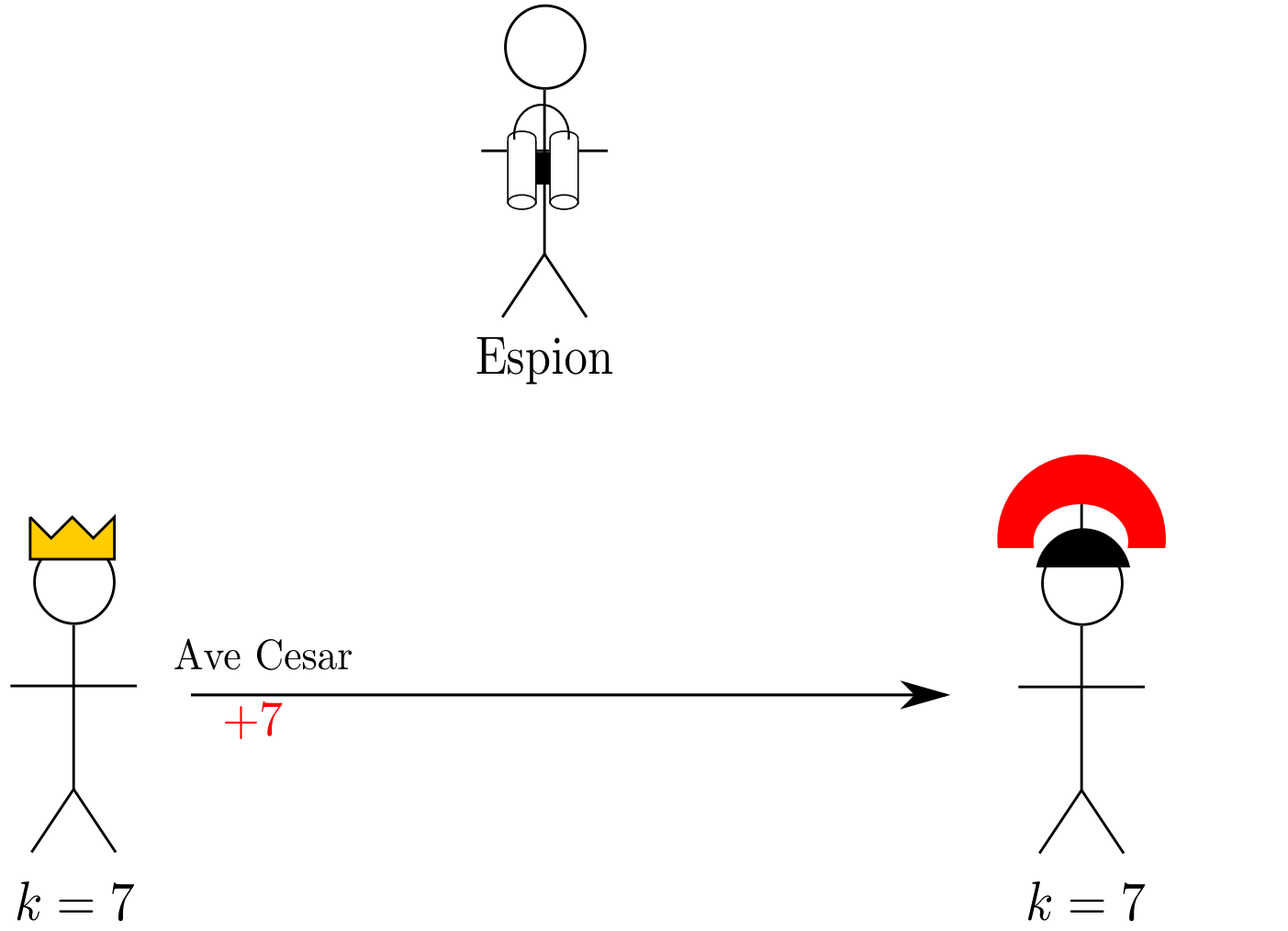
Rome

Frontière grecque

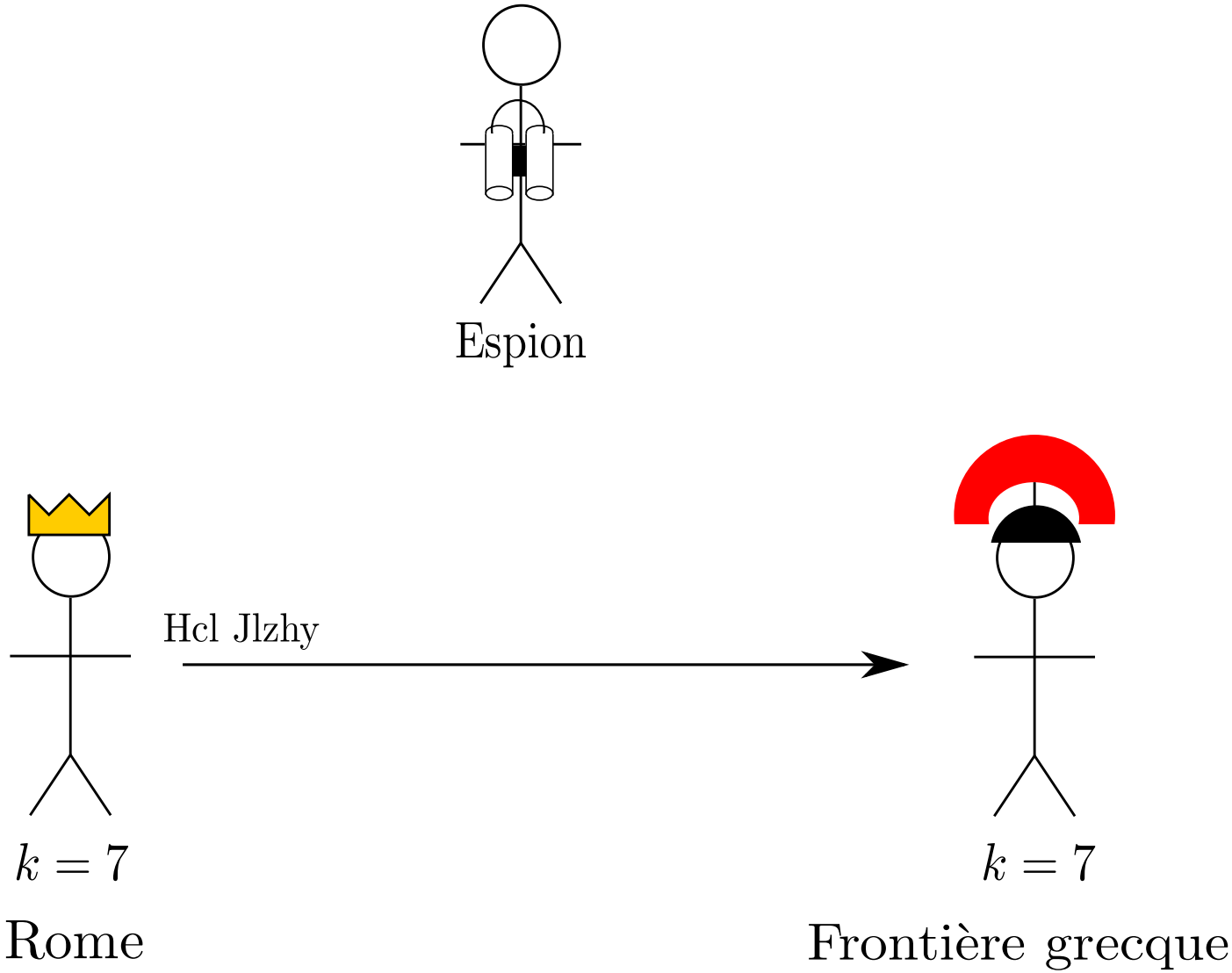
Retour à l'histoire



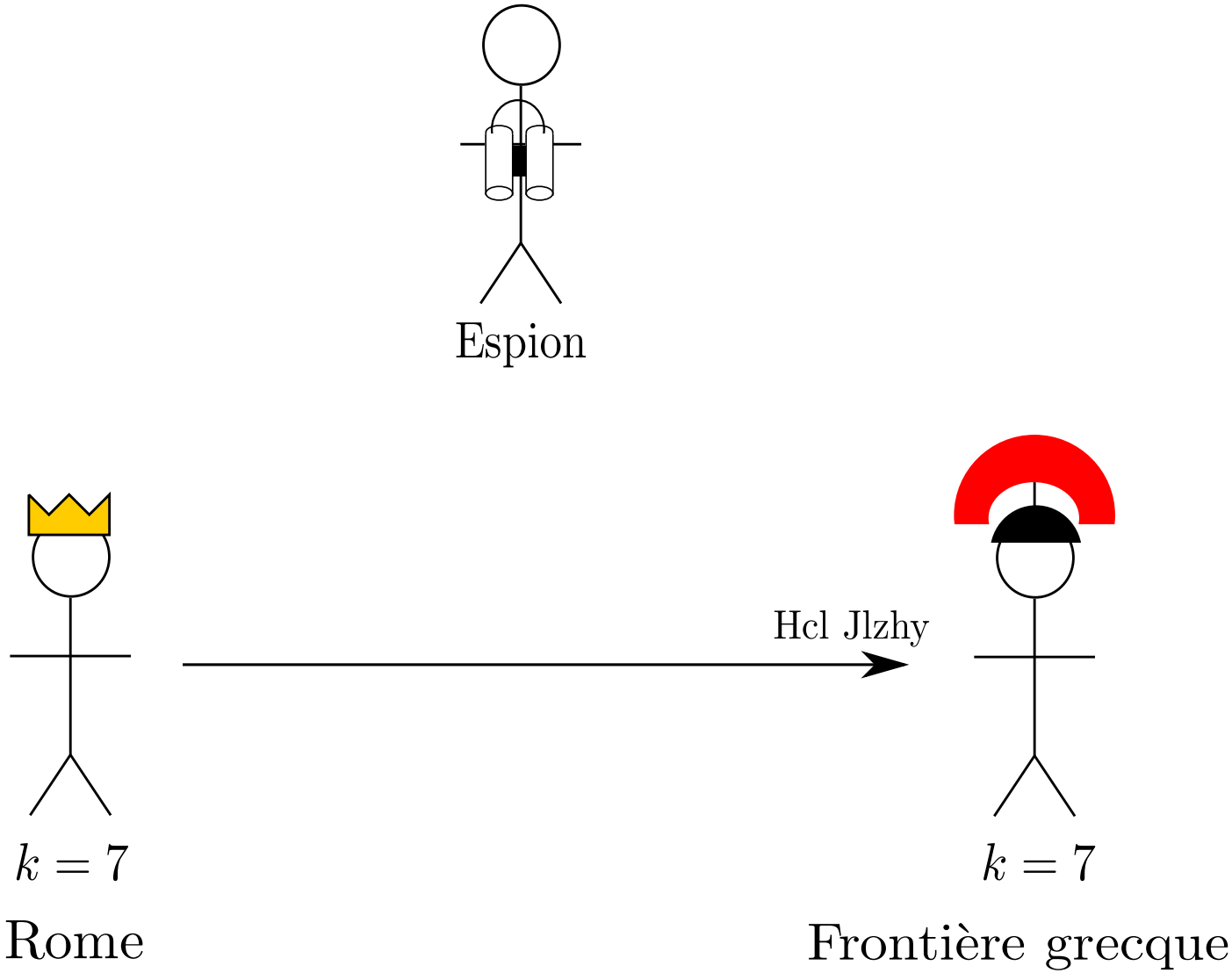
Retour à l'histoire



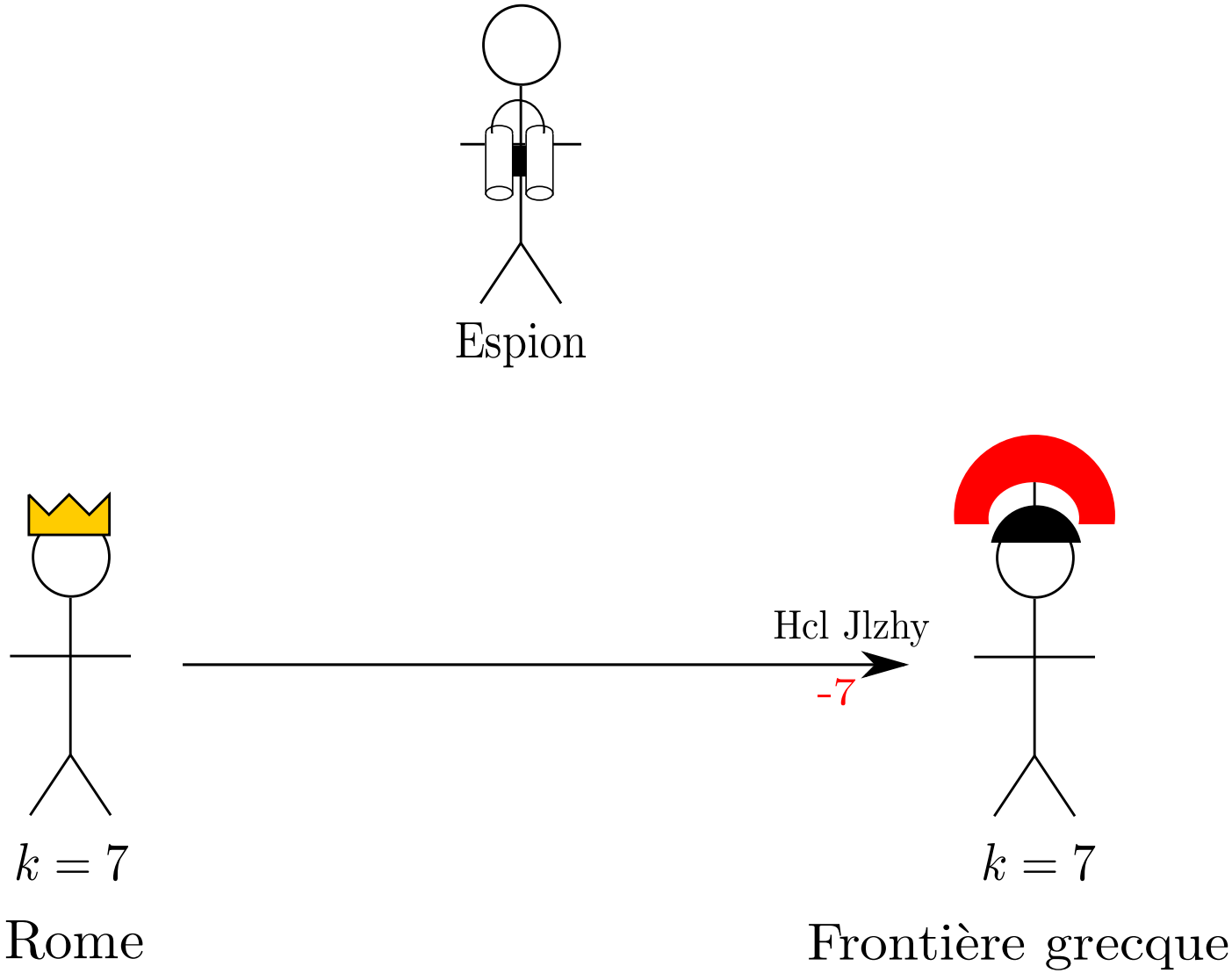
Retour à l'histoire



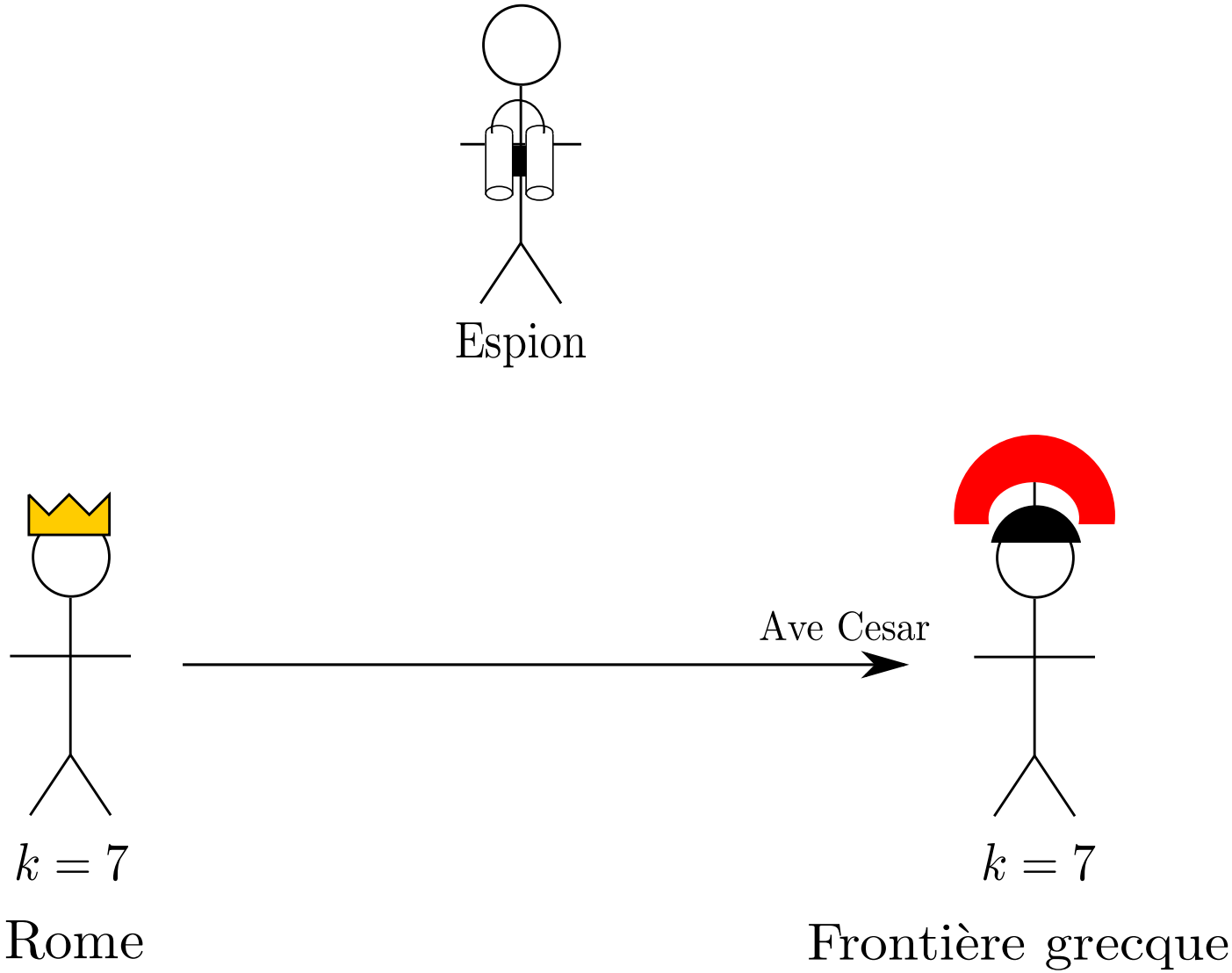
Retour à l'histoire



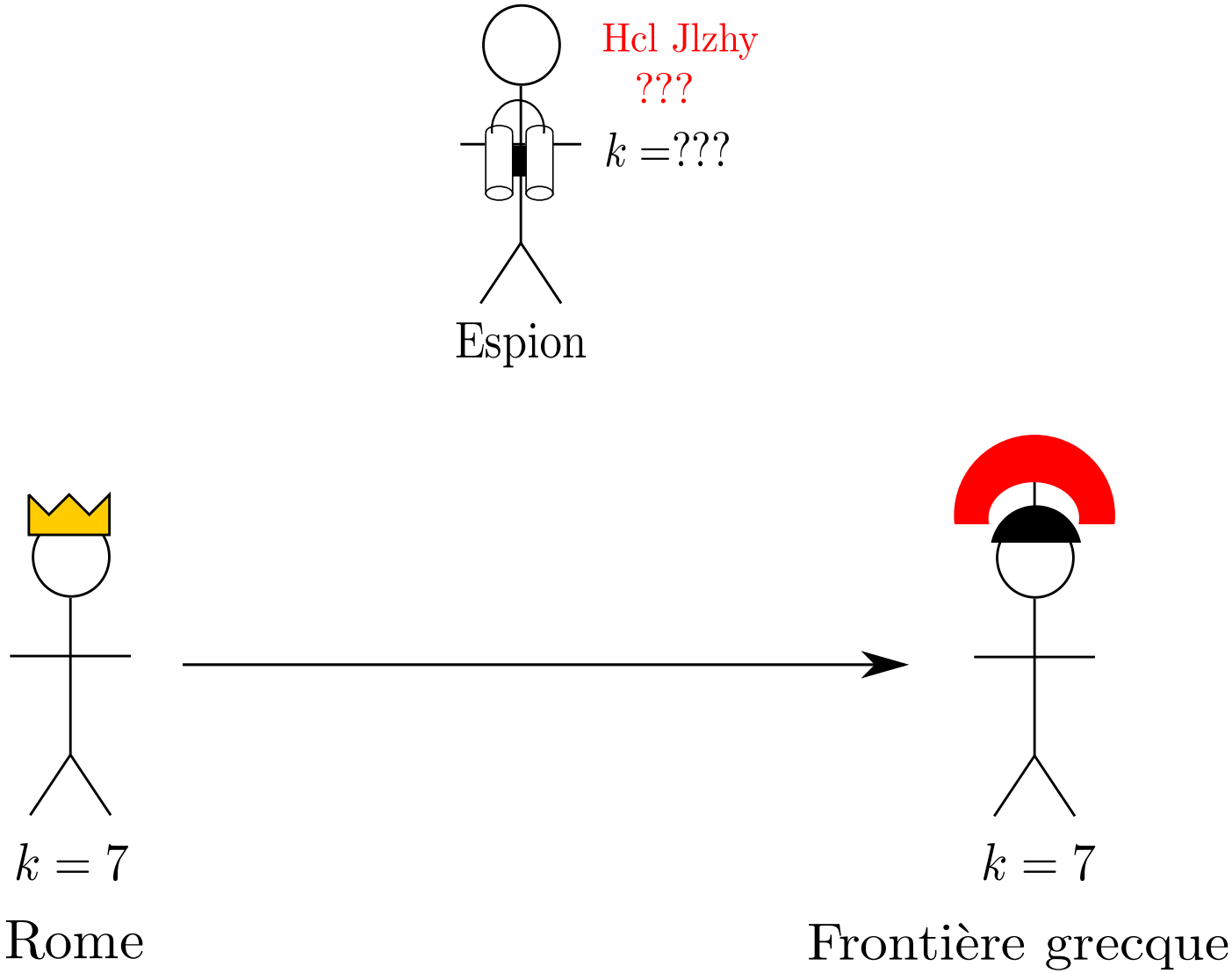
Retour à l'histoire



Retour à l'histoire



Retour à l'histoire



Les problèmes restants

- On peut tester les 25 possibilités de décalage.
- Analyse des fréquences des lettres.

Les problèmes restants

- On peut tester les 25 possibilités de décalage.
- Analyse des fréquences des lettres.

Aujourd'hui, on sait faire mieux que César, par exemple

- la machine Enigma, utilisée pendant la seconde guerre mondiale

- le protocole AES (utilisé par exemple pour https)

Les problèmes restants

- On peut tester les 25 possibilités de décalage.
- Analyse des fréquences des lettres.

  Aujourd'hui, on sait faire mieux que César, par exemple
  - la machine Enigma, utilisée pendant la seconde guerre mondiale

  - le protocole AES (utilisé par exemple pour https)

- Il faut que César et son général se rencontrent.
  $\Rightarrow$ achats sur Internet ?

Chiffrement asymétrique

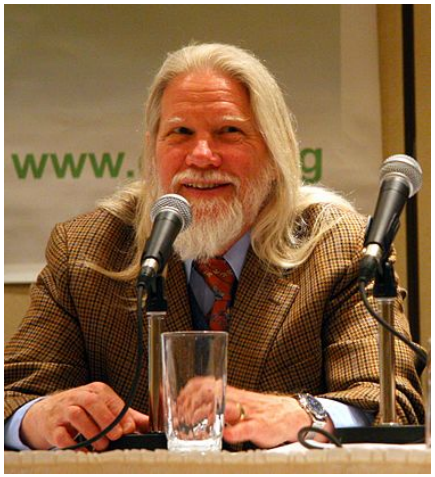
Question

César et son général peuvent-ils échanger des messages chiffrés sans s'être rencontrés avant ?

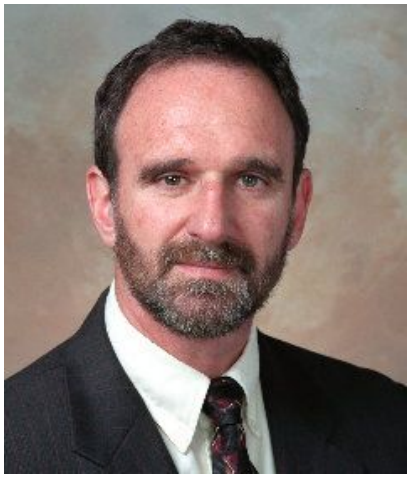
Question

César et son général peuvent-ils échanger des messages chiffrés sans s'être rencontrés avant ?

Oui. C'est la cryptographie *asymétrique* (= à clé publique)
Première construction proposée par Whitfield Diffie et Martin Hellman en 1976.

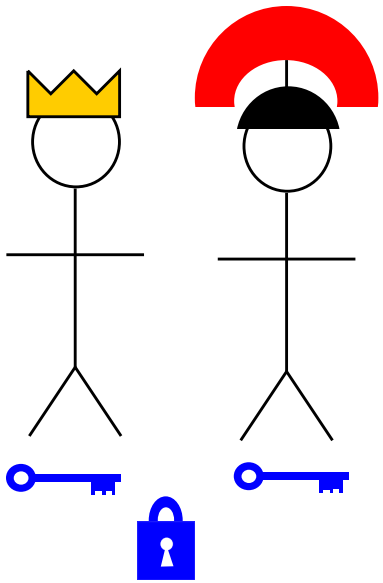


Whitfield Diffie



Martin Hellman

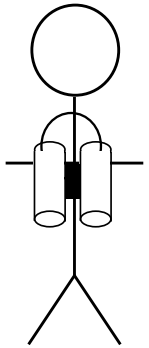
Chiffrement symétrique (par exemple chiffrement de César)



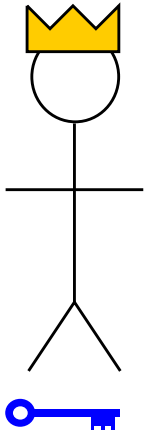
Rome

Frontière grecque

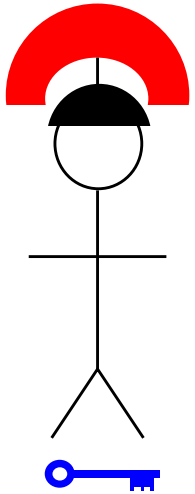
Chiffrement symétrique (par exemple chiffrement de César)



Espion

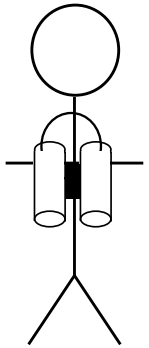


Rome

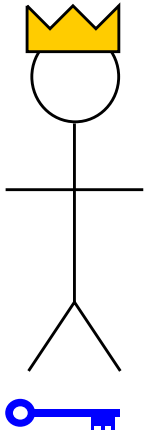


Frontière grecque

Chiffrement symétrique (par exemple chiffrement de César)

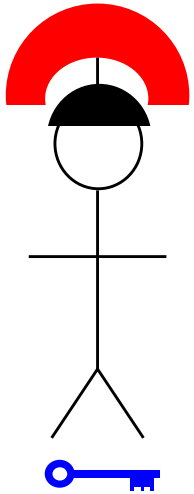


Espion



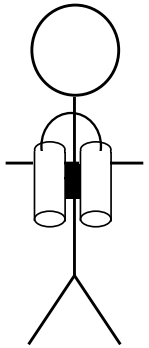
Ave Cesar

Rome

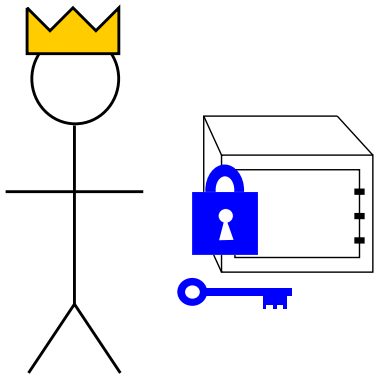


Frontière grecque

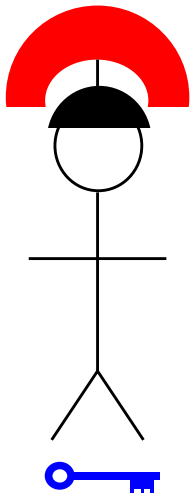
Chiffrement symétrique (par exemple chiffrement de César)



Espion

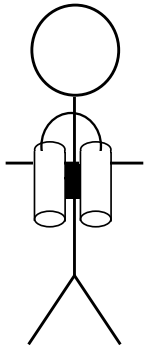


Rome

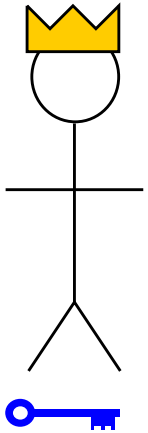


Frontière grecque

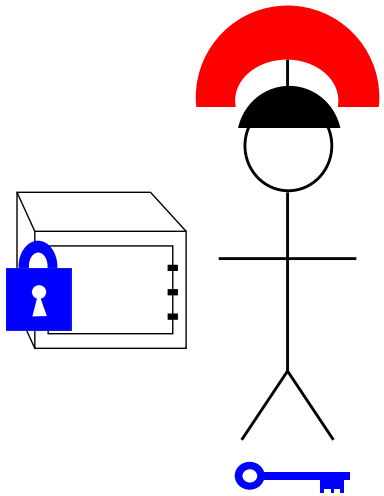
Chiffrement symétrique (par exemple chiffrement de César)



Espion

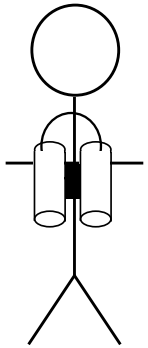


Rome

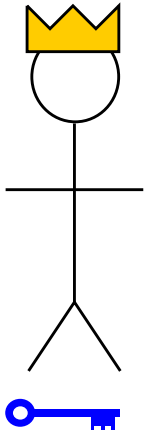


Frontière grecque

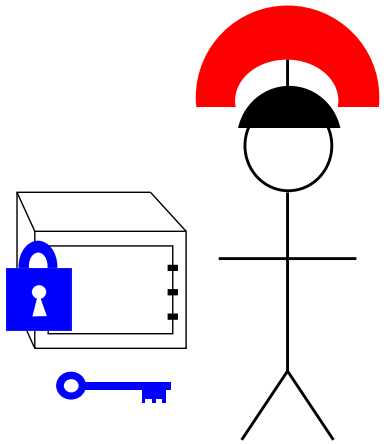
Chiffrement symétrique (par exemple chiffrement de César)



Espion

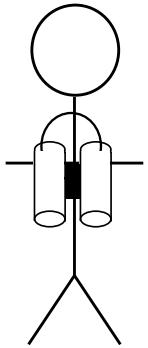


Rome

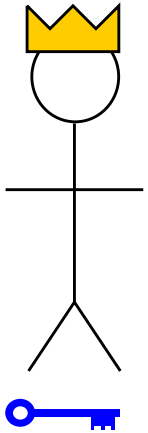


Frontière grecque

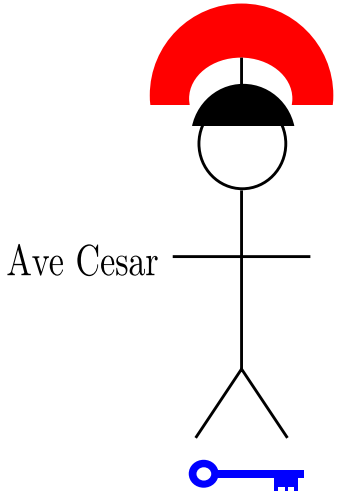
Chiffrement symétrique (par exemple chiffrement de César)



Espion

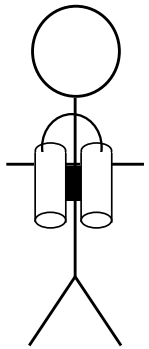


Rome

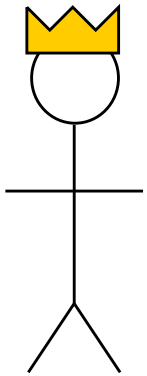


Frontière grecque

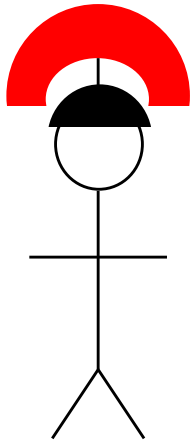
Chiffrement asymétrique



Espion

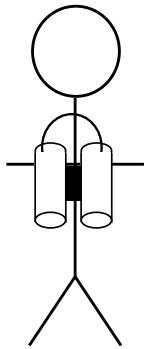


Rome

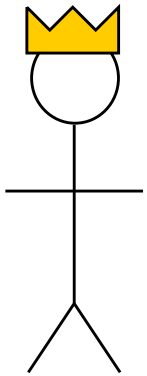


Frontière grecque

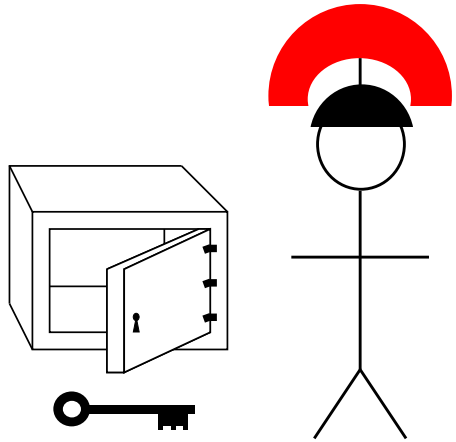
Chiffrement asymétrique



Espion

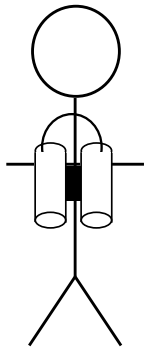


Rome

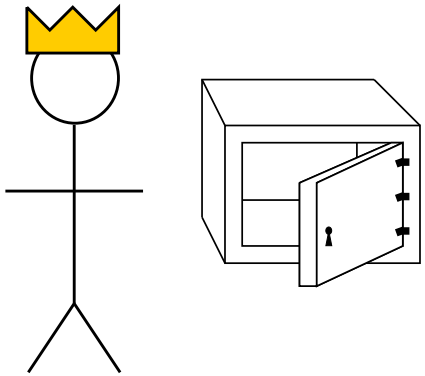


Frontière grecque

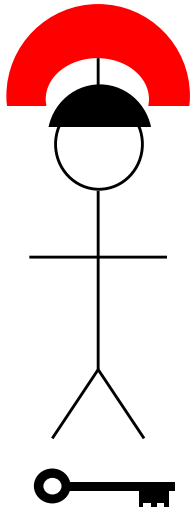
Chiffrement asymétrique



Espion

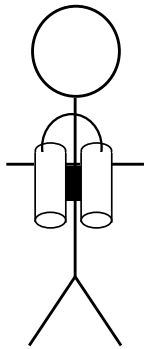


Rome

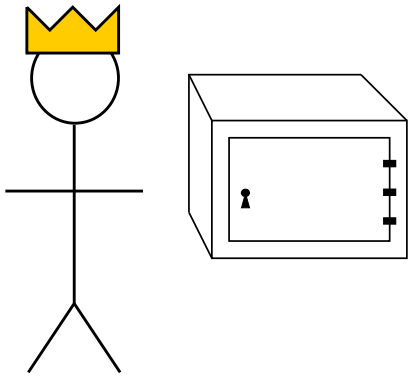


Frontière grecque

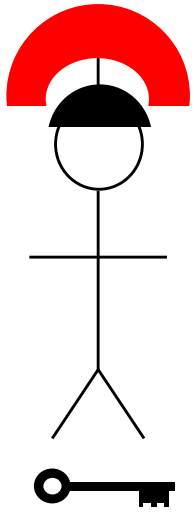
Chiffrement asymétrique



Espion

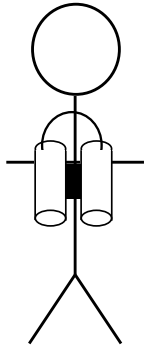


Rome

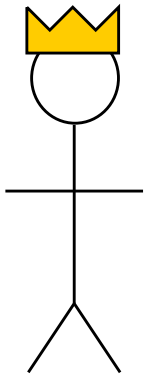


Frontière grecque

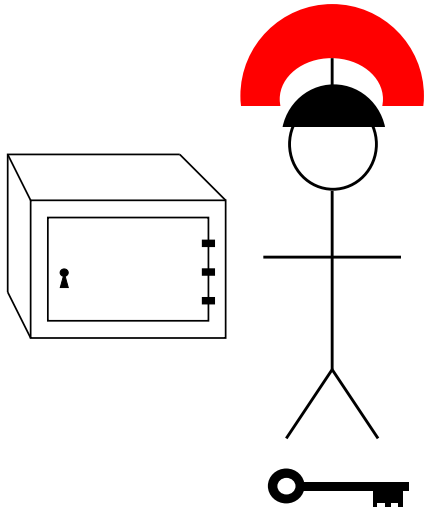
Chiffrement asymétrique



Espion

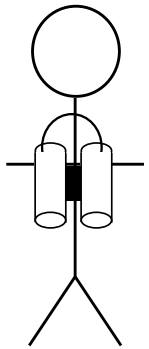


Rome

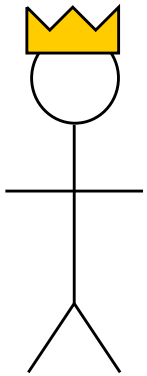


Frontière grecque

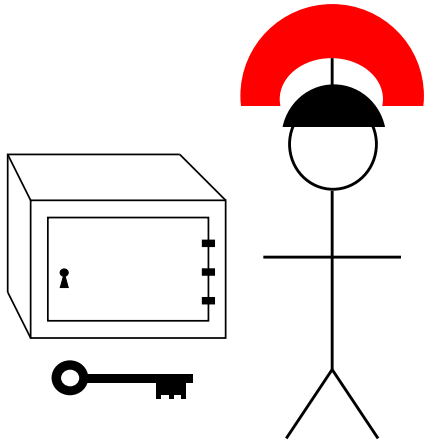
Chiffrement asymétrique



Espion

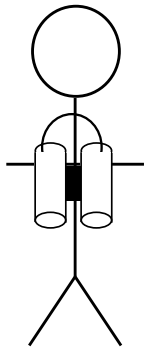


Rome

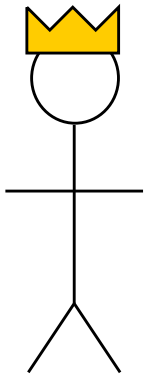


Frontière grecque

Chiffrement asymétrique



Espion

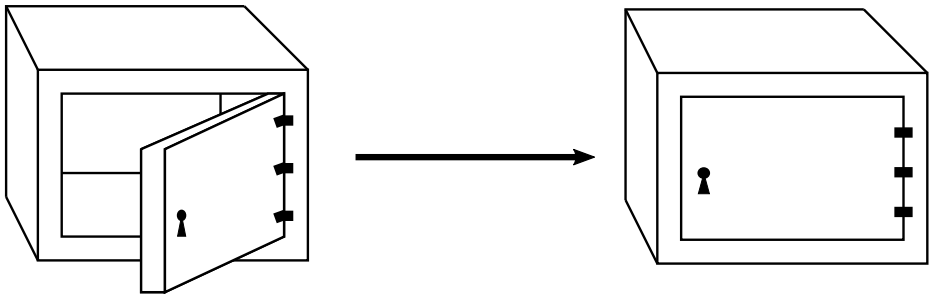


Rome

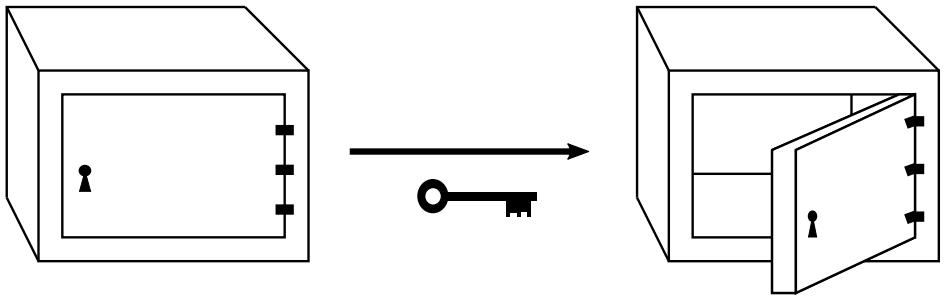
Frontière grecque

En pratique, comment ça marche ?

Les boîtes sont remplacées par des maths.



Facile



Difficile
(besoin de la clé)

Objectif : trouver un problème de maths facile dans un sens et difficile dans l'autre.

Factorisation

Factoriser

Factoriser un nombre, c'est l'écrire comme produit de deux nombres plus petits.

$6 = 2 \times 3, \qquad 16 = 4 \times 4 = 2 \times 8, \qquad 291149 = ???$

Factorisation

Factoriser

Factoriser un nombre, c'est l'écrire comme produit de deux nombres plus petits.

$6 = 2 \times 3, \qquad 16 = 4 \times 4 = 2 \times 8, \qquad 291149 = 811 \times 359$

Factorisation

Factoriser

Factoriser un nombre, c'est l'écrire comme produit de deux nombres plus petits.

$$6 = 2 \times 3, \qquad 16 = 4 \times 4 = 2 \times 8, \qquad 291149 = 811 \times 359$$

On a trouvé un bon problème de maths

Factoriser est un problème *difficile*, mais multiplier c'est *facile*.

Factorisation

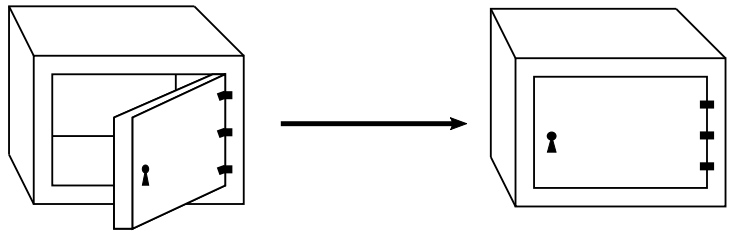
Factoriser

Factoriser un nombre, c'est l'écrire comme produit de deux nombres plus petits.

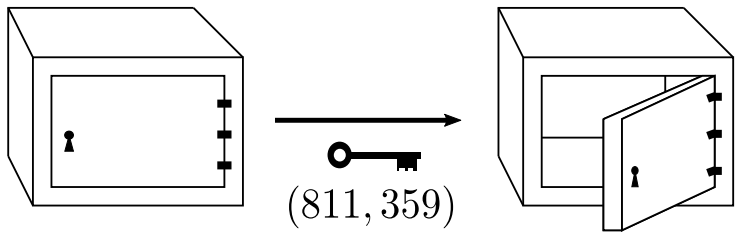
$6 = 2 \times 3, \qquad 16 = 4 \times 4 = 2 \times 8, \qquad 291149 = 811 \times 359$

On a trouvé un bon problème de maths

Factoriser est un problème *difficile*, mais multiplier c'est *facile*.



$811 \times 359 \to 291149$
Facile



$291149 \to 811 \times 359$
Difficile

Factorisation

Factoriser

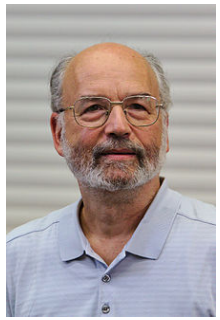
Factoriser un nombre, c'est l'écrire comme produit de deux nombres plus petits.

$6 = 2 \times 3, \qquad 16 = 4 \times 4 = 2 \times 8, \qquad 291149 = 811 \times 359$

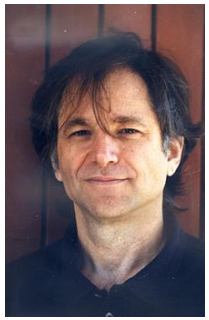
On a trouvé un bon problème de maths

Factoriser est un problème *difficile*, mais multiplier c'est *facile*.

Ronald Rivest



Adi Shamir



Leonard Adleman

Chiffrement RSA (1977)

Symétrique vs asymétrique

Chiffrement asymétrique:

- pas besoin d'interaction physique
- mais plus lent que le chiffrement symétrique

Symétrique vs asymétrique

Chiffrement asymétrique:

- pas besoin d'interaction physique
- mais plus lent que le chiffrement symétrique

En pratique: on utilise du chiffrement asymétrique pour échanger une clé secrète commune, puis on utilise du chiffrement symétrique pour discuter.

Formalisons un peu

La cryptographie vue de loin

Protocoles cryptographiques (à clé publique)

chiffrement signature vote électronique ...

La cryptographie vue de loin

Protocoles cryptographiques (à clé publique)

chiffrement signature vote électronique ...

codes correcteurs réseaux euclidiens isogénies

factorisation logarithme discret ...

Problèmes (supposés) difficiles

La cryptographie vue de loin

Protocoles cryptographiques (à clé publique)

chiffrement signature vote électronique ...

codes correcteurs réseaux euclidiens isogénies

~~factorisation~~ ~~logarithme discret~~ ...

Problèmes (supposés) difficiles

même avec un ordinateur quantique

La cryptographie vue de loin

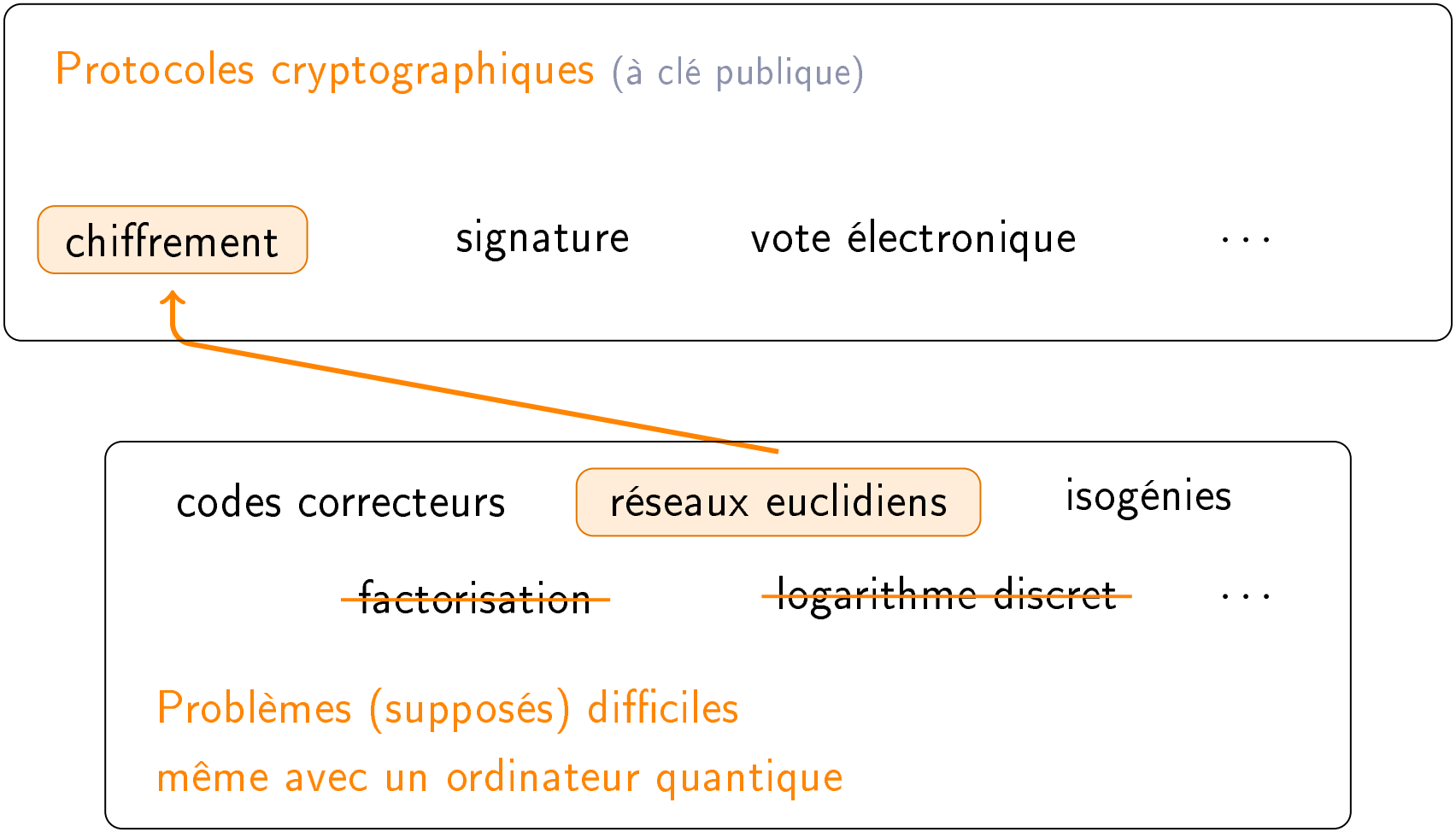


Schéma de chiffrement

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

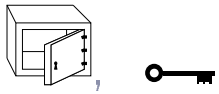
Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$

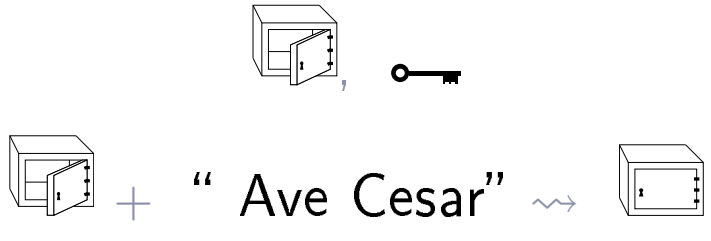


Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)



Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)
- $\texttt{Dec}(sk, c) \rightsquigarrow m' \in \{0,1\}$

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)
- $\texttt{Dec}(sk, c) \rightsquigarrow m' \in \{0,1\}$

Correction: pour tout $(pk, sk) \leftarrow \texttt{Setup}()$, et pour tout $m \in \{0,1\}$

$$\texttt{Dec}(sk, \texttt{Enc}(m, pk)) = m$$

Sécurité

T-Attaquant: algorithme (probabiliste) qui termine en temps $\leq T$

Sécurité

T-Attaquant: algorithme (probabiliste) qui termine en temps $\leq T$

On définit l'avantage d'un T-attaquant $\mathcal{A}$ pour différentes attaques:

Sécurité

T-Attaquant: algorithme (probabiliste) qui termine en temps $\leq T$

On définit l'avantage d'un T-attaquant $\mathcal{A}$ pour différentes attaques:

- Key recovery: retrouver sk à partir de pk

$$\mathsf{Adv}_{\mathrm{KR}}(\mathcal{A}) := \Pr_{(pk,sk)\leftarrow \texttt{KeyGen()}}\left(\mathcal{A}(pk) = sk\right)$$

Sécurité

T-Attaquant: algorithme (probabiliste) qui termine en temps $\leq T$

On définit l'avantage d'un T-attaquant $\mathcal{A}$ pour différentes attaques:

- Key recovery: retrouver sk à partir de pk

$$\mathrm{Adv}_{\mathrm{KR}}(\mathcal{A}) := \Pr_{(pk,sk)\leftarrow \mathtt{KeyGen()}} \Big(\mathcal{A}(pk) = sk\Big)$$

- Indistinguabilité: distinguer un chiffrer de 0 et un chiffré de 1

$$\mathrm{Adv}_{\mathrm{Ind}}(\mathcal{A}) := \Big|\Pr\left(\mathcal{A}(pk, \mathtt{Enc}(1, pk)) = 1\right) - \Pr\left(\mathcal{A}(pk, \mathtt{Enc}(0, pk)) = 1\right)\Big|$$

Sécurité

T-Attaquant: algorithme (probabiliste) qui termine en temps $\leq T$

On définit l'avantage d'un T-attaquant $\mathcal{A}$ pour différentes attaques:

- Key recovery: retrouver sk à partir de pk

$$\mathrm{Adv}_{\mathrm{KR}}(\mathcal{A}) := \Pr_{(pk,sk)\leftarrow \mathtt{KeyGen()}}\Big(\mathcal{A}(pk) = sk\Big)$$

- Indistinguabilité: distinguer un chiffrer de 0 et un chiffré de 1

$$\mathrm{Adv}_{\mathrm{Ind}}(\mathcal{A}) := \Big|\Pr\left(\mathcal{A}(pk, \mathtt{Enc}(1, pk)) = 1\right) - \Pr\left(\mathcal{A}(pk, \mathtt{Enc}(0, pk)) = 1\right)\Big|$$

- il existe aussi des modèles où l'attaquant peut
  - mesurer le temps de `Enc`
  - lire certains bits de mémoire pendant l'exécution de `Enc`
  - changer certains bits de mémoire pendant l'exécution de `Enc`

Sécurité (suite)

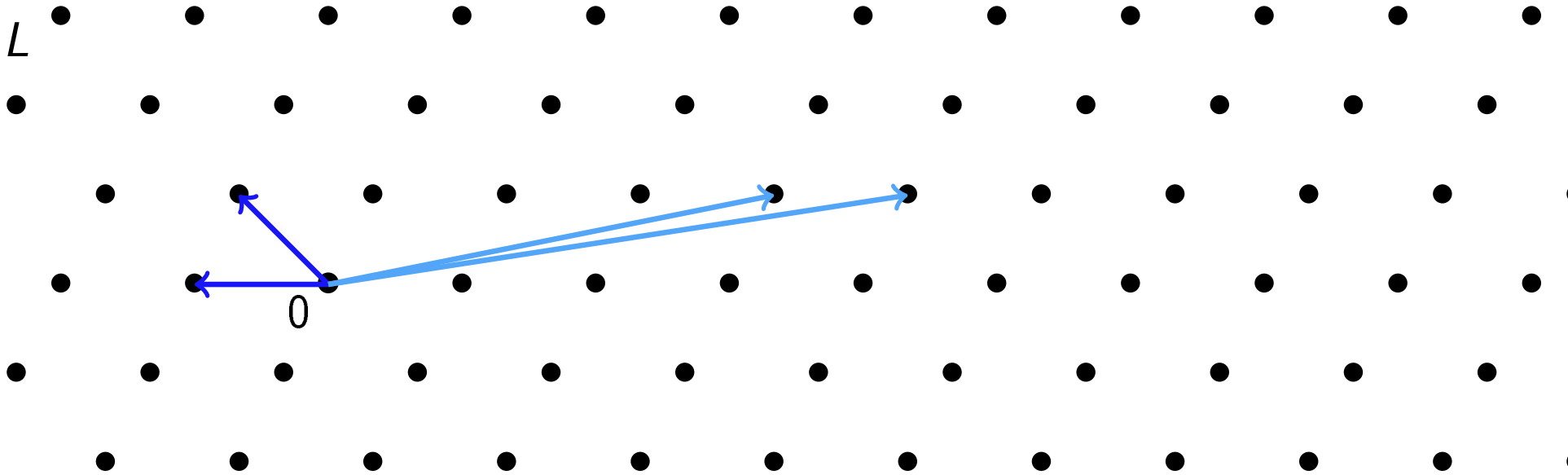
Exemple d'énoncés de sécurité:

"Pour $T = 2^{128}$, il n'existe pas de T-attaquant $\mathcal{A}$ contre le schéma de chiffrement $\mathcal{C} = (\mathtt{Setup}, \mathtt{Enc}, \mathtt{Dec})$ tel que $\mathrm{Adv}_{\mathrm{Ind}}(\mathcal{A}) \geq 2^{-128}$."

"Pour $T = 2^{80}$, il n'existe pas de T-attaquant $\mathcal{A}$ contre le schéma de chiffrement $\mathcal{C} = (\mathtt{Setup}, \mathtt{Enc}, \mathtt{Dec})$ tel que $\mathrm{Adv}_{\mathrm{KR}}(\mathcal{A}) \geq 2^{-80}$."

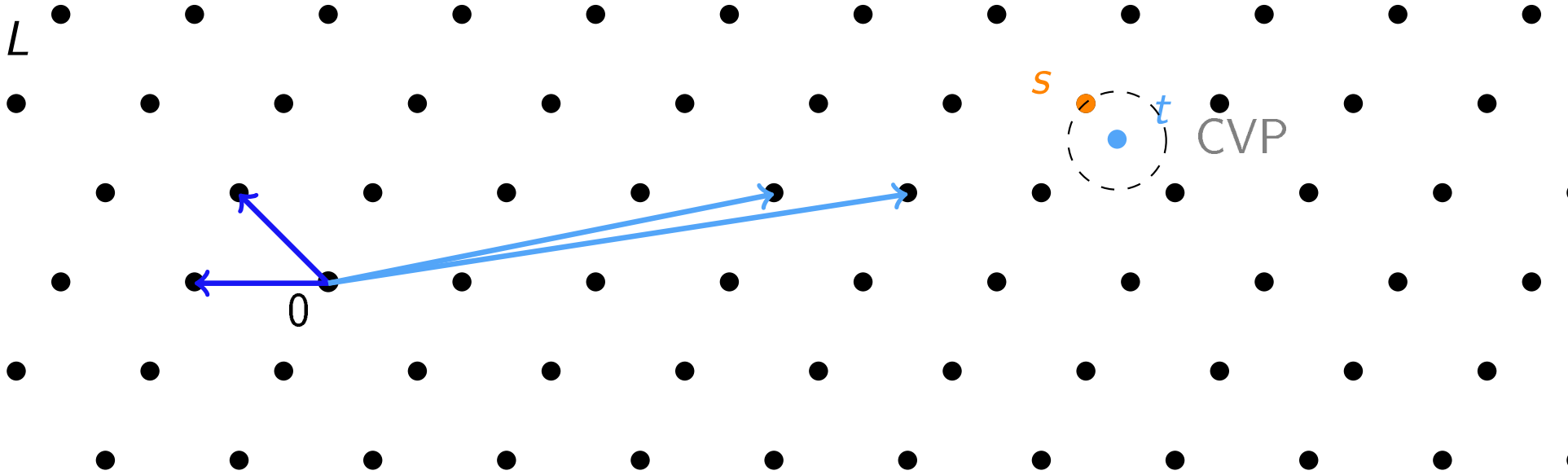
Réseaux euclidiens

Réseaux euclidiens



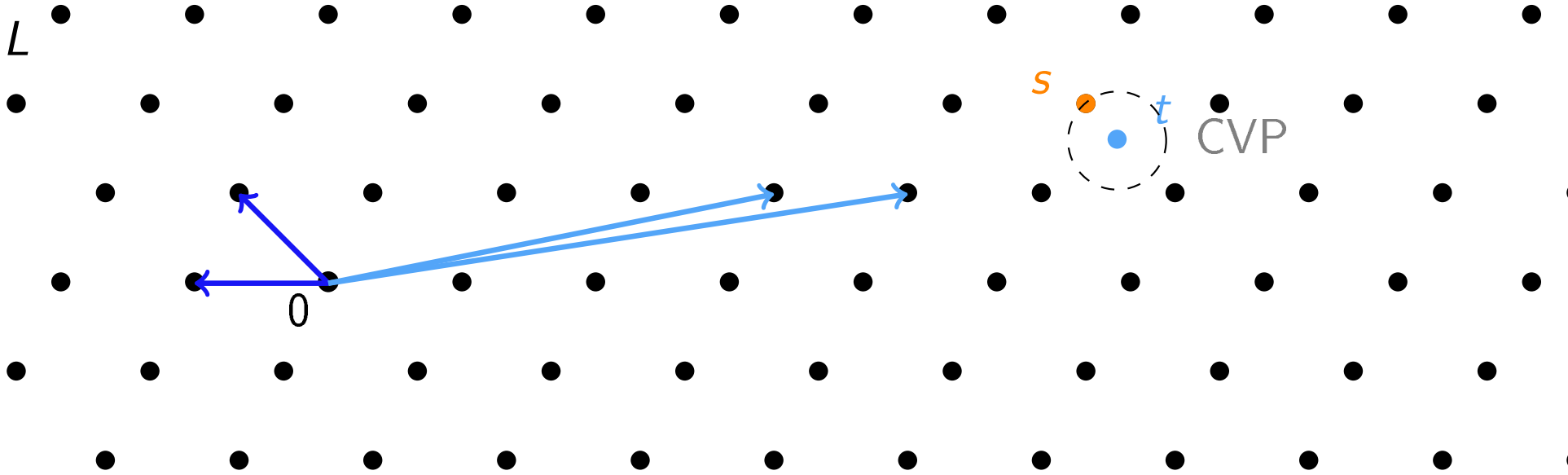
- $L = \{Bx \mid x \in \mathbb{Z}^n\}$ est un réseau (euclidien)
- $B \in \mathrm{GL}_n(\mathbb{R})$ est une base
- n est la dimension (ou le rang) de L

Exemple de problème difficile



CVP : Problème du plus proche vecteur (Closest Vector Problem)

Exemple de problème difficile



CVP : Problème du plus proche vecteur (Closest Vector Problem)

- Construire t à partir de s: facile (tout le temps)
- Retrouver s à partir de t:
  - facile si on a une bonne base
  - difficile si on a seulement une mauvaise base

Facile ? Difficile ?

Problème facile: il existe un algorithme polynomial qui résout le problème

Problème difficile: on ne connaît pas d'algorithme polynomial

Facile ? Difficile ?

Problème facile: il existe un algorithme polynomial qui résout le problème

Problème difficile: on ne connaît pas d'algorithme polynomial

Polynomial en quoi ?

Facile ? Difficile ?

Problème facile: il existe un algorithme polynomial qui résout le problème

Problème difficile: on ne connaît pas d'algorithme polynomial

Polynomial en quoi ?

Pour les réseaux, on mesure la complexité en fonction de la dimension n.
$\Rightarrow$ le problème est difficile quand la dimension est grande

Facile ? Difficile ?

Problème facile: il existe un algorithme polynomial qui résout le problème

Problème difficile: on ne connaît pas d'algorithme polynomial

Polynomial en quoi ?

Pour les réseaux, on mesure la complexité en fonction de la dimension n.
$\Rightarrow$ le problème est difficile quand la dimension est grande

Exemples concrets:

- $n \leq 170 \rightsquigarrow$ quelques jours sur un super calculateur
- $n \geq 500 \rightsquigarrow$ des milliards d'années (même avec un super calculateur)

Construction du schéma de chiffrement

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)
- $\texttt{Dec}(sk, c) \rightsquigarrow m' \in \{0,1\}$

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)
- $\texttt{Dec}(sk, c) \rightsquigarrow m' \in \{0,1\}$

Correction: $\texttt{Dec}(sk, \texttt{Enc}(m, pk)) = m$ pour tout $m \in \{0,1\}$

Chiffrement à clé publique

Espace des messages: $\{0,1\}$

Un schéma de chiffrement consiste en 3 algorithmes (probabilistes):

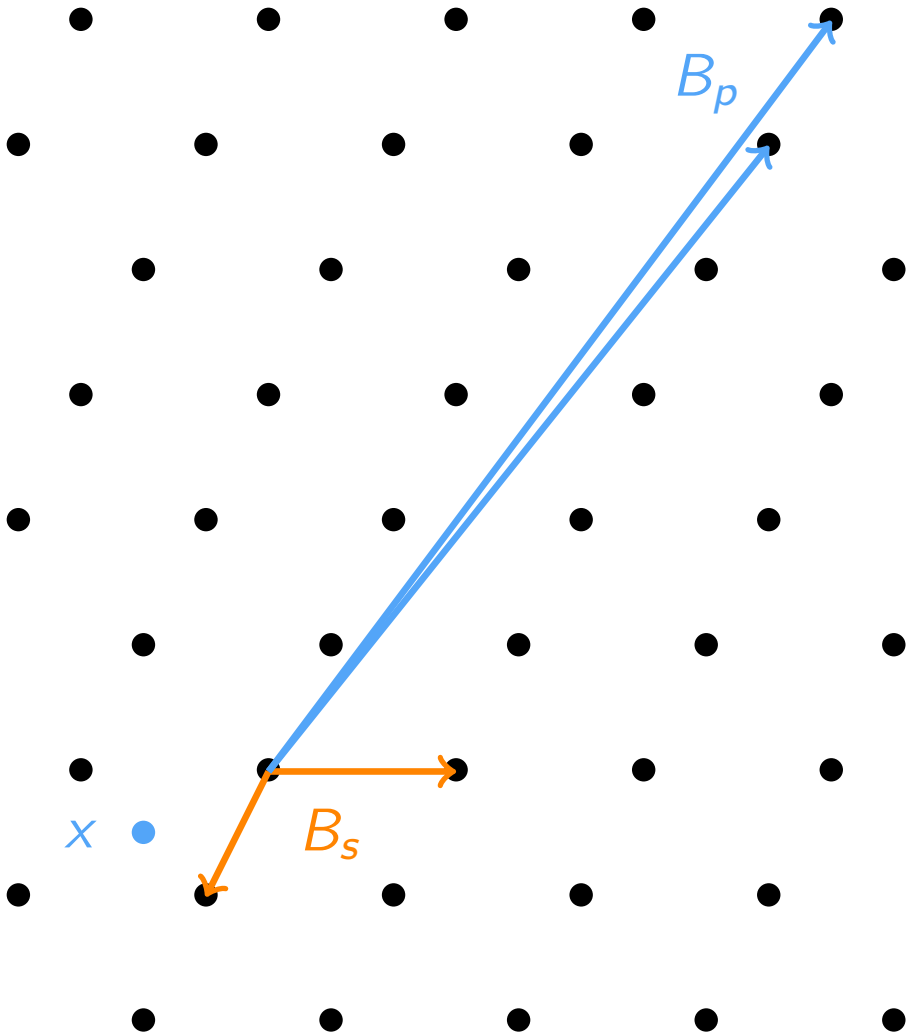
- $\texttt{Setup}() \rightsquigarrow (pk, sk)$
- $\texttt{Enc}(pk, m) \rightsquigarrow c$ (m message $\in \{0,1\}$)
- $\texttt{Dec}(sk, c) \rightsquigarrow m' \in \{0,1\}$

Correction: $\texttt{Dec}(sk, \texttt{Enc}(m, pk)) = m$ pour tout $m \in \{0,1\}$

Sécurité: Pour tout T-attaquant $\mathcal{A}$

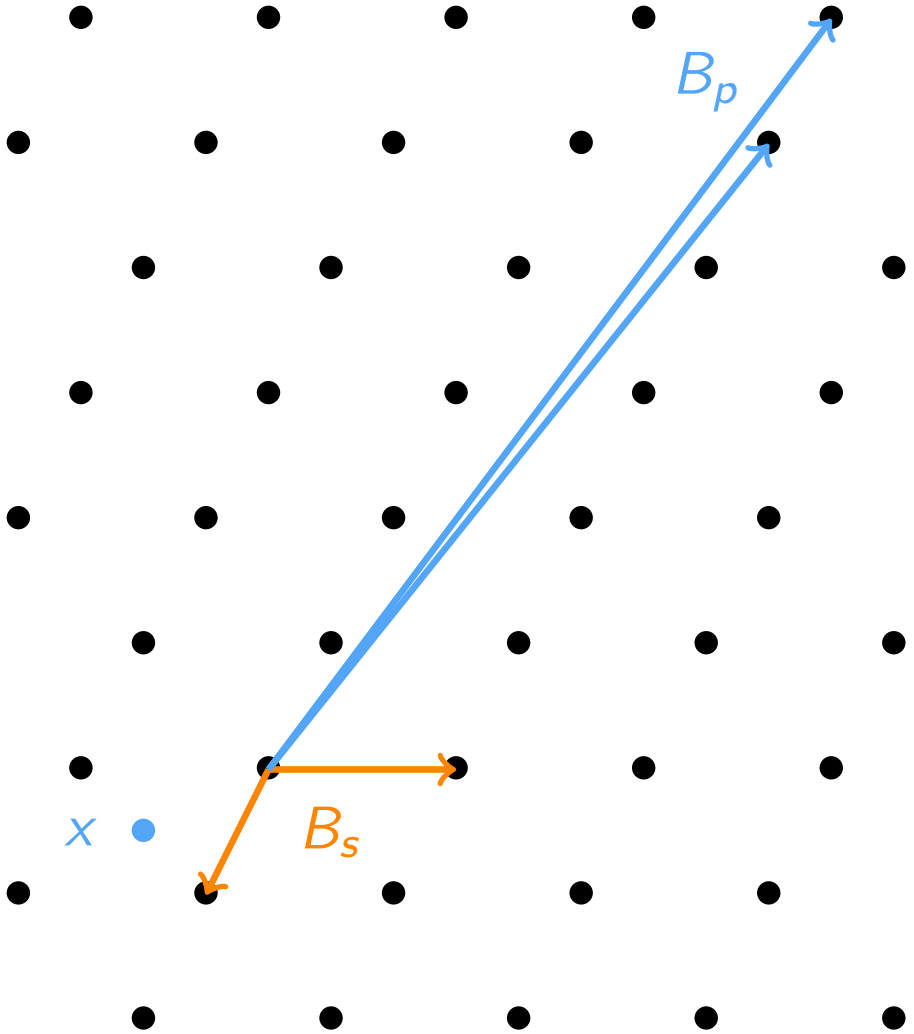
$$\left| \Pr\left(\mathcal{A}(pk, \texttt{Enc}(1, pk)) = 1\right) - \Pr\left(\mathcal{A}(pk, \texttt{Enc}(0, pk)) = 1\right) \right| \leq \varepsilon$$

Chiffrement à base de réseaux



$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

Chiffrement à base de réseaux

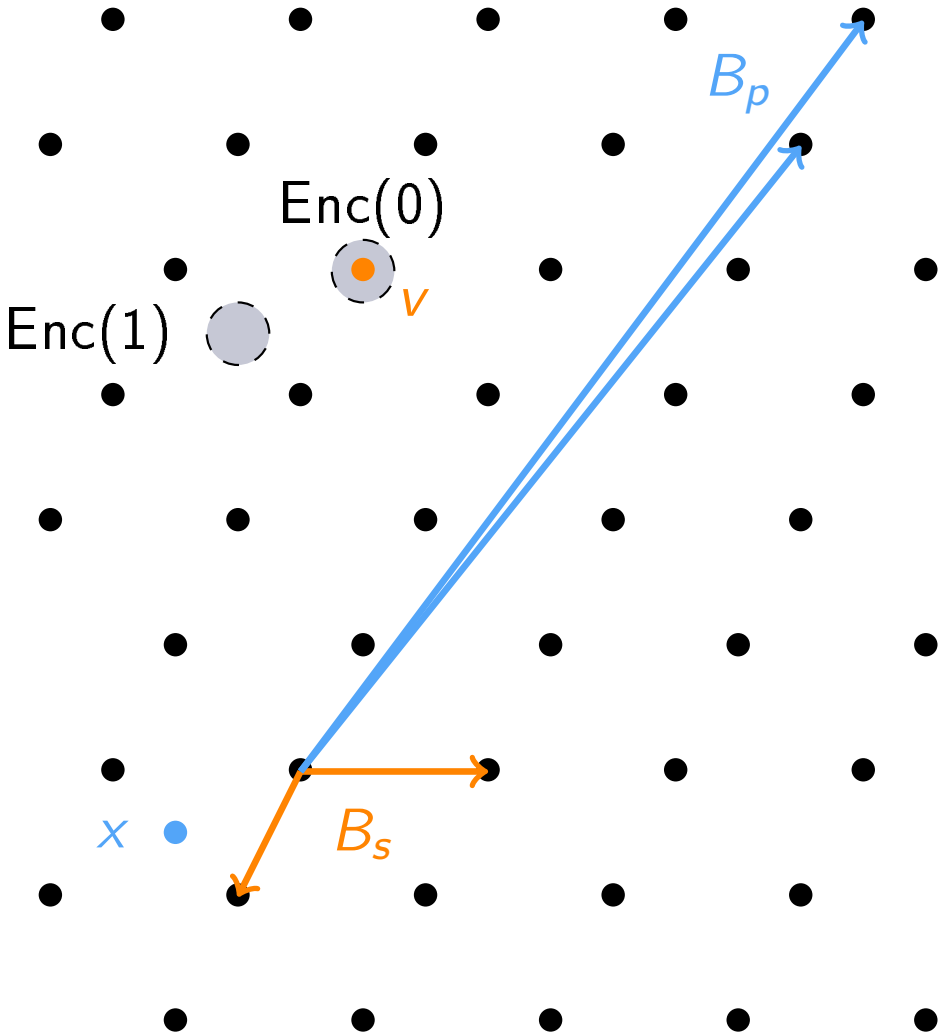


$\mathrm{pk} = (B_p, x)$

$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Chiffrement à base de réseaux



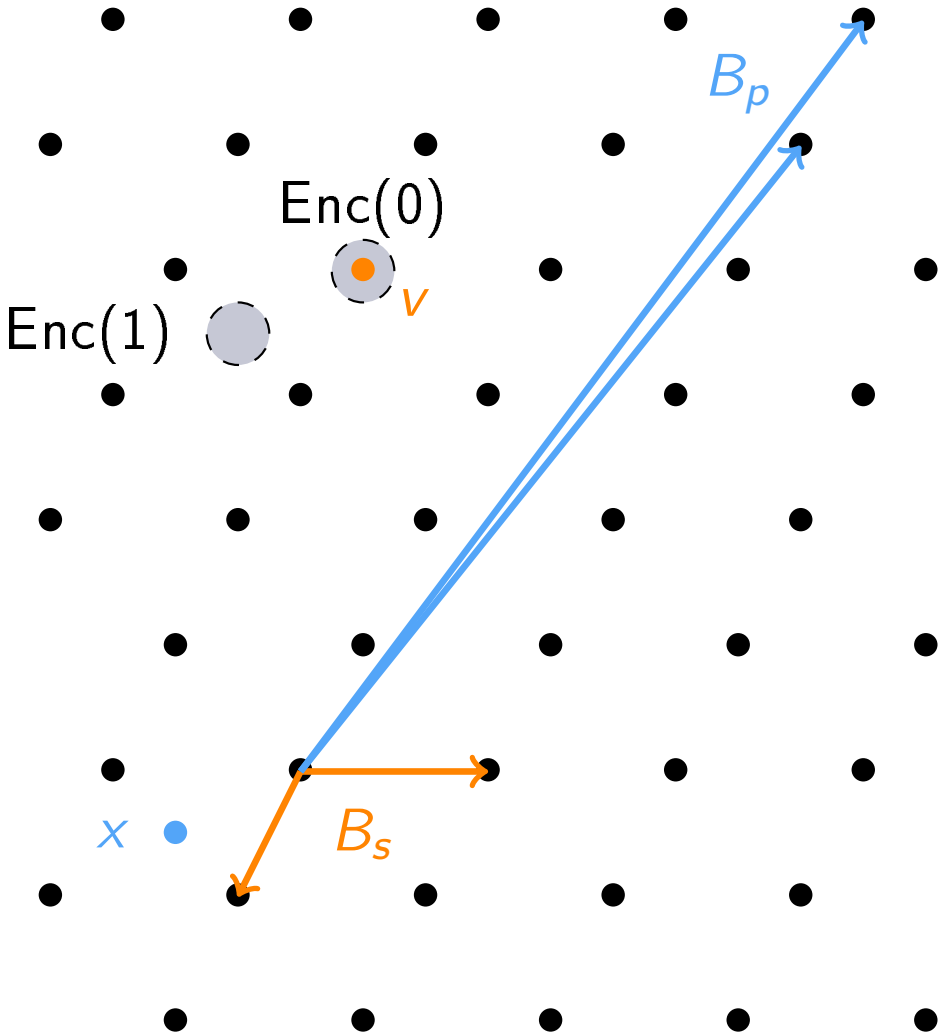
$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

$\mathtt{Enc}(m, \mathrm{pk})$:

- générer $v \in L$ aléatoirement
- générer $e \in \mathbb{R}^n$ petit
- renvoyer $c = v + e + m \cdot x$

Chiffrement à base de réseaux



$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

$\mathtt{Enc}(m, \mathrm{pk})$:

- générer $v \in L$ aléatoirement
- générer $e \in \mathbb{R}^n$ petit
- renvoyer $c = v + e + m \cdot x$

$\mathtt{Dec}(c, \mathrm{sk})$:

- trouver $w \in L$ le plus proche de c
- si c est très proche de w, renvoyer $m = 0$
- sinon renvoyer $m = 1$

Preuve de sécurité

Théorème

Le schéma de chiffrement précédent est correct.
S'il existe un T-attaquant $\mathcal{A}$ tel que $\mathrm{Adv}_{\mathrm{Ind}}(\mathcal{A}) \geq \varepsilon$, alors il existe un algorithme $\mathcal{B}$ qui résout dec-CVP en temps T avec probabilité $\geq \varepsilon$.
(le schéma de chiffrement et "sûr" à condition que le problème dec-CVP soit difficile.)

dec-CVP: étant donné une mauvaise base B_p d'un réseau L et $t \in \mathbb{R}^n$ t.q.

- soit t est proche d'un point de L;
- soit $t - x$ est proche d'un point de L (avec x comme sur la diapo précédente)

déterminer dans quel cas on est.

Preuve de sécurité

Théorème

Le schéma de chiffrement précédent est correct.
S'il existe un T-attaquant $\mathcal{A}$ tel que $\mathrm{Adv}_{\mathrm{Ind}}(\mathcal{A}) \geq \varepsilon$, alors il existe un algorithme $\mathcal{B}$ qui résout dec-CVP en temps T avec probabilité $\geq \varepsilon$.
(le schéma de chiffrement et "sûr" à condition que le problème dec-CVP soit difficile.)

dec-CVP: étant donné une mauvaise base B_p d'un réseau L et $t \in \mathbb{R}^n$ t.q.

- soit t est proche d'un point de L;
- soit $t - x$ est proche d'un point de L (avec x comme sur la diapo précédente)

déterminer dans quel cas on est.

idée de la preuve au tableau

Un (autre?) problème mathématique difficile: ISIS

ISIS

Inhomogeneous short integer solution (ISIS):

Paramètres: q un nombre premier, $m \geq n$ des entiers

Entrée: $A \in \mathbb{Z}^{m \times n}$ et $y \in \mathbb{Z}^n$

Sortie: $x \in \mathbb{Z}^m$ tel que $x \cdot A = y \bmod q$ et $\|x\|$ aussi petit que possible

ISIS

Inhomogeneous short integer solution (ISIS):

Paramètres: q un nombre premier, $m \geq n$ des entiers

Entrée: $A \in \mathbb{Z}^{m \times n}$ et $y \in \mathbb{Z}^n$

Sortie: $x \in \mathbb{Z}^m$ tel que $x \cdot A = y \bmod q$ et $\|x\|$ aussi petit que possible

Existence d'une solution: Si A de rang n mod q $\rightsquigarrow$ $\exists$ une solution

ISIS

Inhomogeneous short integer solution (ISIS):

Paramètres: q un nombre premier, $m \geq n$ des entiers

Entrée: $A \in \mathbb{Z}^{m \times n}$ et $y \in \mathbb{Z}^n$

Sortie: $x \in \mathbb{Z}^m$ tel que $x \cdot A = y \bmod q$ et $\|x\|$ aussi petit que possible

Existence d'une solution: Si A de rang n mod q $\rightsquigarrow$ $\exists$ une solution

Unicité: $\{x \cdot A \,|\, x \in (\mathbb{Z}/q\mathbb{Z})^m\}$ est un espace vectoriel affine de dimension $m - n$ $(\text{si } \mathrm{rk}(A) = n)$ $\rightsquigarrow$ q^{m-n} solutions potentielles

ISIS

Inhomogeneous short integer solution (ISIS):

Paramètres: q un nombre premier, $m \geq n$ des entiers

Entrée: $A \in \mathbb{Z}^{m \times n}$ et $y \in \mathbb{Z}^n$

Sortie: $x \in \mathbb{Z}^m$ tel que $x \cdot A = y \bmod q$ et $\|x\|$ aussi petit que possible

Existence d'une solution: Si A de rang n mod q $\rightsquigarrow$ $\exists$ une solution

Unicité: $\{x \cdot A \,|\, x \in (\mathbb{Z}/q\mathbb{Z})^m\}$ est un espace vectoriel affine de dimension $m - n$ (si $\mathrm{rk}(A) = n$) $\rightsquigarrow$ q^{m-n} solutions potentielles

Calcul d'une solution:

- Calculer $x \in \mathbb{Z}^m$ tel que $x \cdot A = y \bmod q$ $\rightsquigarrow$ facile (pivot de Gauss)
- Trouver x petit parmi les q^{m-n} possibilités $\rightsquigarrow$ difficile si $m \gg n$
  (aussi difficile que de résoudre le problème du plus proche vecteur dans un réseau)

Conclusion

Conclusion

Pour faire de la crypto à clé publique on a besoin de problèmes

- faciles dans un sens
- difficiles dans l'autre sens si on n'a pas la clé secrète
- faciles dans l'autre sens si on a la clé secrète

Conclusion

Pour faire de la crypto à clé publique on a besoin de problèmes

- faciles dans un sens
- difficiles dans l'autre sens si on n'a pas la clé secrète
- faciles dans l'autre sens si on a la clé secrète

Comment construire ce genre de problèmes ?

- avec des nombres premiers
- avec des corps finis
- avec des courbes elliptiques
- avec des réseaux euclidiens
- avec des codes correcteurs d'erreur
- ...

Conclusion

Pour faire de la crypto à clé publique on a besoin de problèmes

- faciles dans un sens
- difficiles dans l'autre sens si on n'a pas la clé secrète
- faciles dans l'autre sens si on a la clé secrète

Comment construire ce genre de problèmes ?

- avec des nombres premiers
- avec des corps finis
- avec des courbes elliptiques
- avec des réseaux euclidiens
- avec des codes correcteurs d'erreur
- ...

Questions ?