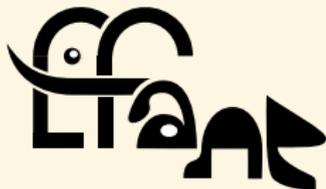


# À quoi sert la cryptologie? – Petit panorama des mathématiques de la cryptologie

2018/10/18 – Lycée Montaigne, Bordeaux

Damien Robert

Équipe LFANT, Inria Bordeaux Sud-Ouest



université  
de **BORDEAUX**

*Inria*  
informatics mathematics

- Prépa au Lycée du Parc à Lyon ;
  - Normale sup Paris ;
  - Thèse à Inria Nancy ;
  - Postdoc à Microsoft Research, Redmond ;
  - Chercheur à Inria Bordeaux.
- ⇒ Algorithmes en théorie des nombres et géométrie algébrique, applications en cryptologie.



# Cryptologie à clé publique

## Cryptologie :

- Chiffrement;
- Authenticité;
- Intégrité.

## Applications :

- Militaires;
- Vie privée;
- Communications (internet, téléphones...)
- Commerce électronique...



## Contexte Historique

- Riche histoire ; chiffrement de messages depuis l'antiquité au moins ;
- Principale application auparavant militaire ;
- Dorénavant la cryptologie joue un rôle essentiel pour garantir la sécurité des communications ;
- **Cryptanalyse** : déchiffrement d'Énigma par le groupe Ultra (Secret) à Bletchley Park lors de la seconde guerre mondiale ;
- À la fin de la guerre, vente des machines Énigma capturées par les alliés à d'autres pays.

### Exemple (Cryptanalyse d'Énigma)

- $\text{enigma}(c) = \text{plug}^{-1} \circ \text{rotor}_1^{-1} \circ \text{rotor}_2^{-1} \circ \text{rotor}_3^{-1} \circ \text{reflector} \circ \text{rotor}_3 \circ \text{rotor}_2 \circ \text{rotor}_1 \circ \text{plug}(c)$ .
  - Les rotors bougent à chaque étape.
  - Le réflecteur transpose une lettre avec une lettre distincte.
- ⇒ Permutation sans point fixe!

## Contexte Historique

- Riche histoire ; chiffrement de messages depuis l'antiquité au moins ;
- Principale application auparavant militaire ;
- Dorénavant la cryptologie joue un rôle essentiel pour garantir la sécurité des communications ;
- **Cryptanalyse** : déchiffrement d'Énigma par le groupe Ultra (Secret) à Bletchley Park lors de la seconde guerre mondiale ;
- À la fin de la guerre, vente des machines Énigma capturées par les alliés à d'autres pays.

### Exemple (Cryptanalyse d'Énigma)

- $\text{enigma}(c) = \text{plug}^{-1} \circ \text{rotor}_1^{-1} \circ \text{rotor}_2^{-1} \circ \text{rotor}_3^{-1} \circ \text{reflector} \circ \text{rotor}_3 \circ \text{rotor}_2 \circ \text{rotor}_1 \circ \text{plug}(c)$ .
  - Les rotors bougent à chaque étape.
  - Le réflecteur transpose une lettre avec une lettre distincte.
- ⇒ Permutation sans point fixe!

# Protocoles cryptographiques

- Briques de base (primitives), s'appuyant sur des objets mathématiques
- Ces primitives sont combinées pour former des algorithmes/modes opératoires (algorithme de chiffrement, algorithme de signature)
- Ces modes opératoires sont combinés pour former des protocoles (protocole de session TLS, protocole de vote)
- Ces protocoles sont implémentés en logiciel ou matériel
- Puis ils sont utilisés.



## Exemple : De RSA à un algorithme de chiffrement

- RSA permet de chiffrer un message  $m$  d'une certaine taille  $k : m \mapsto E(m)$  ;
  - Comment chiffrer un message de longueur arbitraire ?
  - Idée naturelle : découper  $m$  en messages  $m_1 \parallel m_2 \dots \parallel m_d$  de taille  $k$ , et poser  $E(m) = E(m_1) \parallel E(m_2) \dots \parallel E(m_d)$  ;
  - Problème : RSA est malléable.  $E(m \times m') = E(m) \times E(m')$ .
- ⇒ A partir de plusieurs chiffrés on peut en produire plein d'autres ;
- La solution est de chiffrer les blocs  $m_i$  avec du padding :  $E(m_i \parallel G(r) \parallel r \parallel H(m_i \parallel G(r)))$  où  $r$  est aléatoire et  $H$  et  $G$  sont deux fonctions de hachage (on a une preuve de sécurité).



## Exemple : De RSA à un algorithme de chiffrement

- RSA permet de chiffrer un message  $m$  d'une certaine taille  $k$  :  $m \mapsto E(m)$  ;
  - Comment chiffrer un message de longueur arbitraire ?
  - Idée naturelle : découper  $m$  en messages  $m_1 \parallel m_2 \dots \parallel m_d$  de taille  $k$ , et poser  $E(m) = E(m_1) \parallel E(m_2) \dots \parallel E(m_d)$  ;
  - Problème : RSA est malléable.  $E(m \times m') = E(m) \times E(m')$ .
- ⇒ A partir de plusieurs chiffrés on peut en produire plein d'autres ;
- La solution est de chiffrer les blocs  $m_i$  avec du padding :  $E(m_i \parallel G(r) \parallel r \parallel H(m_i \parallel G(r)))$  où  $r$  est aléatoire et  $H$  et  $G$  sont deux fonctions de hachage (on a une preuve de sécurité).



## Example : intégrité et chiffrement

- Comment combiner les deux briques de base que sont le chiffrement (Encrypt) et l'intégrité (MAC Message Authentication Code);
- Encrypt then MAC?
- MAC then Encrypt?
- Encrypt + Mac?



## Example : intégrité et chiffrement

- Comment combiner les deux briques de base que sont le chiffrement (Encrypt) et l'intégrité (MAC Message Authentication Code);
- **Encrypt then MAC?**
- MAC then Encrypt?
- Encrypt + Mac?



# Attaques

- Attaques sur les briques de base (très rare);
- Attaques sur l'empilement des briques en algorithmes ou protocoles;
- Attaques sur l'implémentation;
- Attaques sur l'exécution.



- Briques de base : repose sur des problèmes mathématiques bien identifiés et très étudiés (difficulté de la factorisation, logarithme discret dans les courbes elliptiques)
- Preuves de sécurité sur les algorithmes et protocoles : si un attaquant peut attaquer le protocole (avec une certaine probabilité  $p$  en temps  $T$ ), alors il peut attaquer une brique de base (avec une certaine probabilité  $p'$  en temps  $T'$ )



# Sécurité?

- Erreur dans les preuves
- Preuves justes mais modèle incorrect
- Modèle correct mais utilisé dans un autre contexte
- Réductions de sécurités inefficaces
- Bugs dans les programmes
- Attaques physiques (par canaux cachés) : mesure des impulsions électromagnétiques, du bruit, du temps de calcul, des cache miss



# Attaques sur TLS

SSL (Secure Sockets Layer) / TLS (Transport Layer Security)

- Protocole : Renegotiation attack / Version rollback attack
- BEAST (attaque sur le mode Cipher Block Chaining)
- CRIME and BREACH (attaque sur la compression)
- Downgrade attack : FREAK (export grade cryptography), Logjam (gros précalculs)
- Bugs : Heartbleed (buffer overflow), BERserk, goto fail
- Certificats mal formés

Mitiger la perte des clés privées : **perfect forward secrecy** via un échange de clés éphémères par Diffie-Hellman.



# Sécurité!

## Preuves formelles

- Des protocoles
- Des implémentations
- Des compilateurs (voir « Reflections on Trusting Trust » de Ken Thomson)
- Du matériel

## Implémentation

- En temps constant;
- Sans branches;
- Faites par des experts (bibliothèques opensource comme NaCl)

Ne jamais concevoir son propre système cryptographique ad hoc ou sa propre implémentation à moins d'être un expert.



## Quelques applications cryptographiques modernes

- Chiffrement de groupe ;
- Mise en gage ;
- Partage de secret ;
- Preuves sans divulgation de connaissance ;
- Certificats anonymes ;
- Transfert inconscient ;
- Signature de cercle ;
- Calcul multipartite sécurisé ;
- Chiffrement fonctionnel ;
- Obfuscation ;

### Exemple (Partage de secret)

- Comment partager un secret entre  $n$  personnes de telle sorte que  $m$  personnes parmi les  $n$  puissent le reconstituer ?
- Un polynôme de degré  $d$  est entièrement déterminé par sa valeur en  $d + 1$  points distincts.
- Le secret est un polynôme  $P$  de degré  $m - 1$ , chaque personne  $i$  connaît la valeur  $P(i)$ .

## Quelques applications cryptographiques modernes

- Chiffrement de groupe ;
- Mise en gage ;
- Partage de secret ;
- Preuves sans divulgation de connaissance ;
- Certificats anonymes ;
- Transfert inconscient ;
- Signature de cercle ;
- Calcul multipartite sécurisé ;
- Chiffrement fonctionnel ;
- Obfuscation ;

### Exemple (Partage de secret)

- Comment partager un secret entre  $n$  personnes de telle sorte que  $m$  personnes parmi les  $n$  puissent le reconstituer ?
- Un polynôme de degré  $d$  est entièrement déterminé par sa valeur en  $d + 1$  points distincts.
- Le secret est un polynôme  $P$  de degré  $m - 1$ , chaque personne  $i$  connaît la valeur  $P(i)$ .

# Applications cryptographiques : Bitcoin

- Monnaie électronique décentralisée
- Fichier de transaction public (blockchain)
- Signature des transactions par une courbe elliptique
- La vérification de la blockchain (et validation des nouvelles transactions) fabrication de nouveaux bitcoins.



# Applications cryptographiques : Vote électronique Belenios

- Confidentialité du vote (partage de secret)
- Résultats corrects (preuves Zero-Knowledge)
- Validation (et confidentialité) de la liste des électeurs



# L'essor du cloud computing

- Chiffrement homomorphe : l'utilisateur fournit au nuage un message chiffré  $f_K(m)$  et un programme  $P$ , et le nuage renvoie  $f_K(P(m))$ .  
Le nuage n'a rien appris sur la donnée  $m$ , ni sur le résultat!
  - L'utilisateur fournit au nuage un message chiffré  $f_K(m)$  et le chiffrement  $f_K(P)$  d'un programme  $P$ , et le nuage renvoie  $f_K(P(m))$ .  
Le nuage ne sait pas ce qu'il a calculé!
- ⇒ Une version faible utilise les couplages de courbes elliptiques ;
- ⇒ La version complète utilise des réseaux, en particulier des réseaux d'idéaux dans des corps de nombres ;
- ☹ Encore très lent.



# Cryptographie post-quantique

- RSA (basé sur la factorisation) et les courbes elliptiques sont vulnérables aux ordinateurs quantiques ;
- Problématique pour des secrets à très long terme (50 ans) : secrets défense
- Conception de nouveaux protocoles résistant aux ordinateurs quantiques
- Diffie-Hellmann : échange de clé par des opérations dans un groupe.  
Diffie-Hellmann post-quantique : échange de clé par des opérations dans un graphe.



# Chiffrement

Alice (Sophie Germain)



veut écrire

à Bob (Carl Friedrich Gauss)



# Chiffrement à clé secrète



Alice

$m = \text{QUARTIQUE}$

clé secrète de chiffrement  $K = +3$

$$c = f_K(m)$$

$c = \text{TXDUWLTXH}$  est envoyé



à Bob

$c = \text{TXDUWLTXH}$

clé secrète de déchiffrement  $K' = -K = -3$

$$m = f_K^{(-1)}(c) = f_{K'}(c)$$



# Chiffrement à clé secrète



Alice

$m = \text{QUARTIQUE}$

clé secrète de chiffrement  $K = +3$

$$c = f_K(m)$$

$c = \text{TXDUWLTXH}$  est envoyé



à Bob

$c = \text{TXDUWLTXH}$

clé secrète de déchiffrement  $K' = -K = -3$

$$m = f_{K'}^{-1}(c) = f_{K'}(c)$$



# Chiffrement à clé secrète



Alice

$m = \text{QUARTIQUE}$

clé secrète de chiffrement  $K = +3$

$$c = f_K(m)$$

$c = \text{TXDUWLTXH}$  est envoyé



à Bob

$c = \text{TXDUWLTXH}$

clé secrète de déchiffrement  $K' = -K = -3$

$$m = f_{K'}^{-1}(c) = f_{K'}(c)$$



## Chiffrement à clé secrète

- Trois étapes : création et distribution de clés, chiffrement, déchiffrement
- Avantages : simple, rapide, bien connu
- Fragilités : attaques statistiques, gestion de clés
- Le chiffrement de Vernam (One Time Pad) est **inconditionnellement sûr**, quelle que soit la puissance de calcul de l'adversaire (**Attention : nécessite une clé secrète de même taille que le message envoyé**);
- Notion de théorie de l'information (Shannon);
- Très compliqué et coûteux à mettre en place correctement.

Comment transmettre la clé secrète de manière sécurisée ?

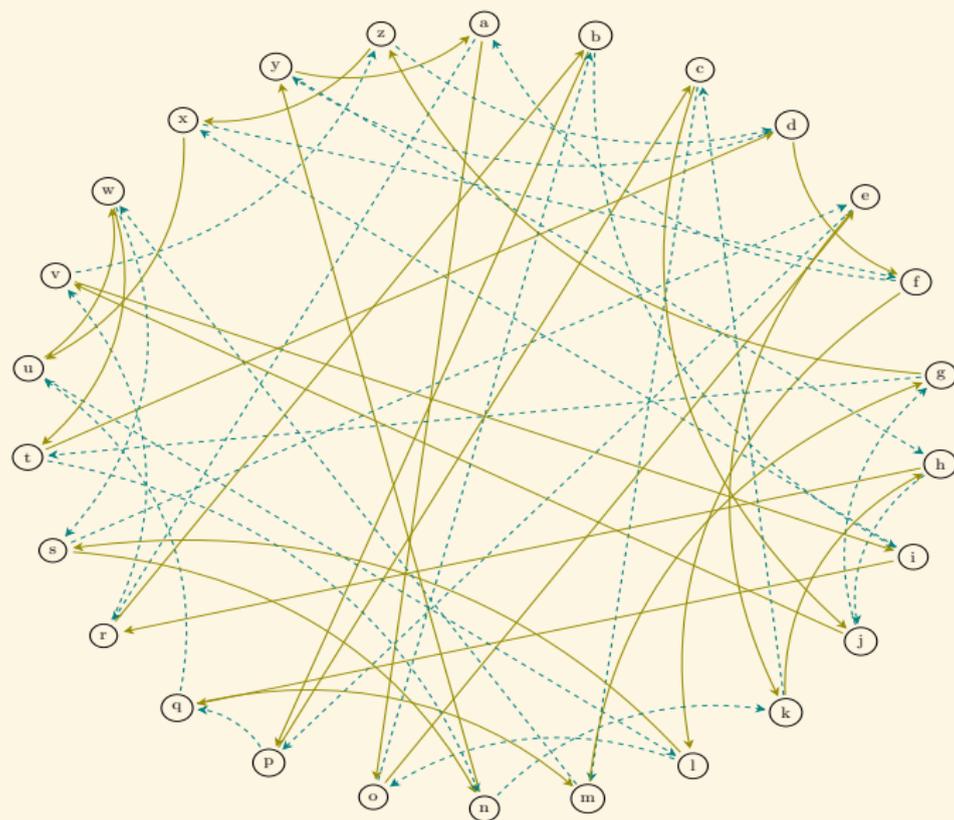


# Échange de clé

- Échanger une clé secrète commune à travers un canal public ;
- Proposé par Diffie et Hellman en 1976 ;
- Utilise des groupes ;
- Version moderne post-quantique : utilise des graphes.

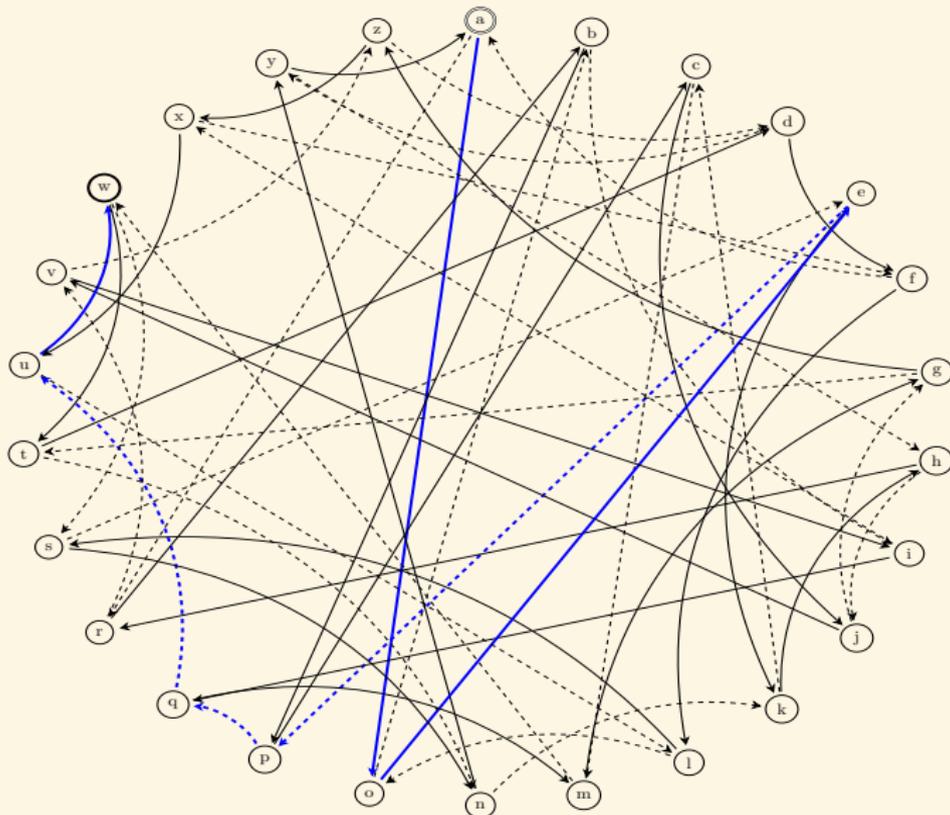


# Échange de clé par graphe



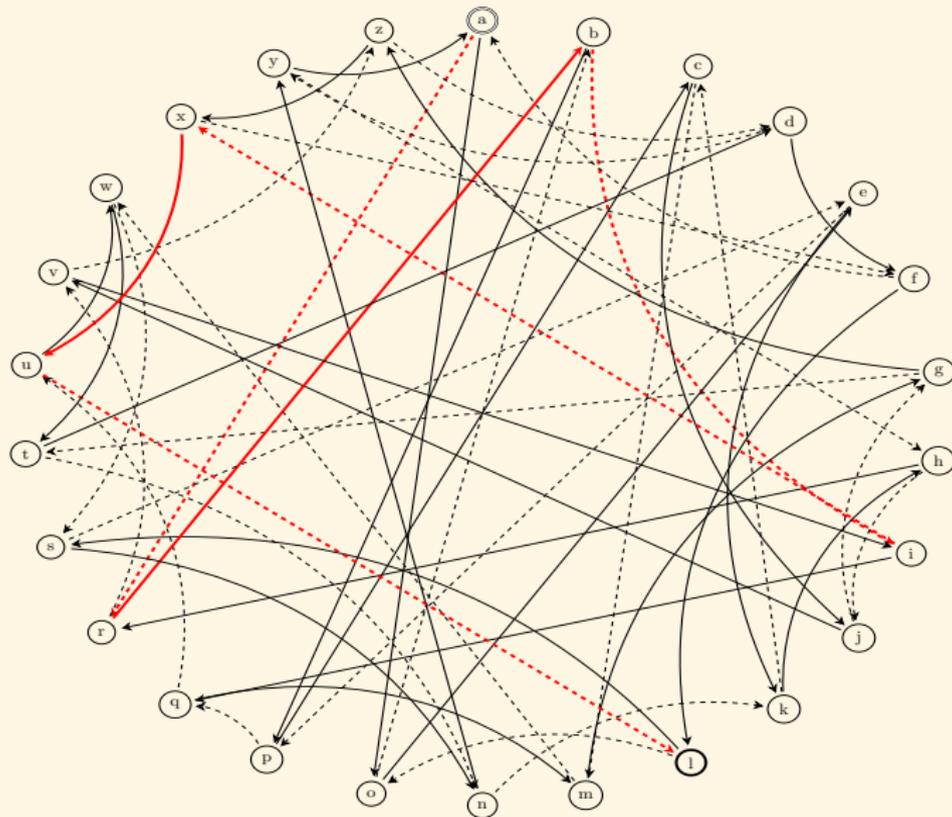
# Échange de clé par graphe

Alice part de 'a', suit le chemin 001110, et tombe sur 'w'.



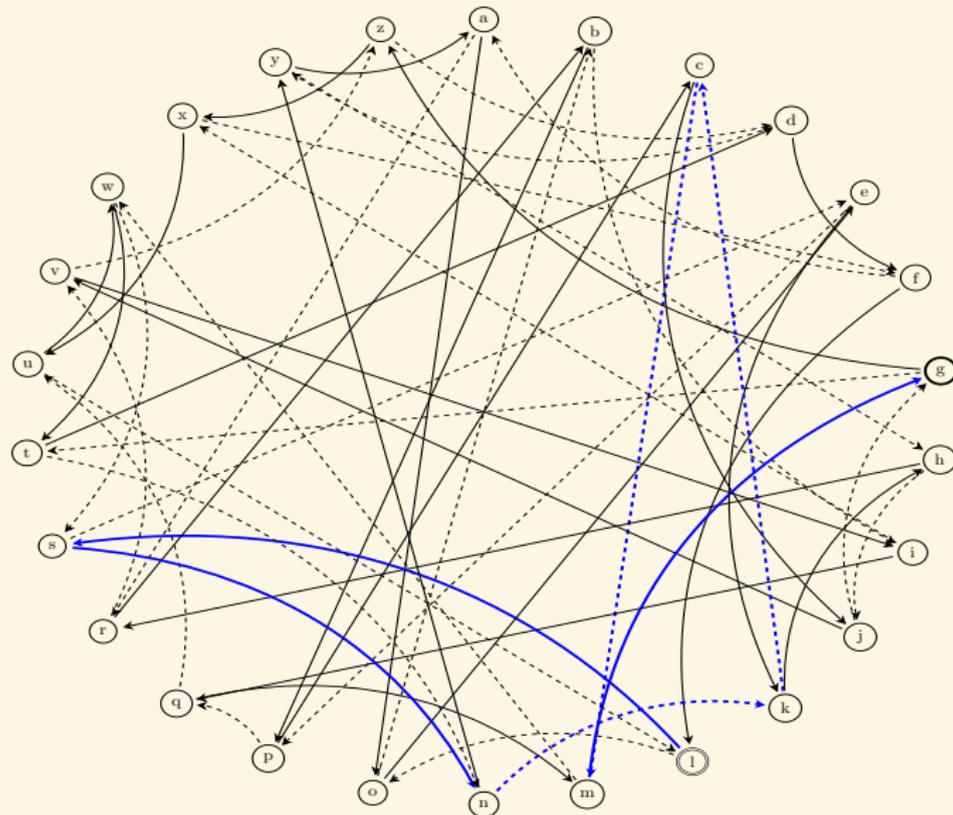
# Échange de clé par graphe

Bob part de 'a', suit le chemin 101101, et tombe sur 'l'.



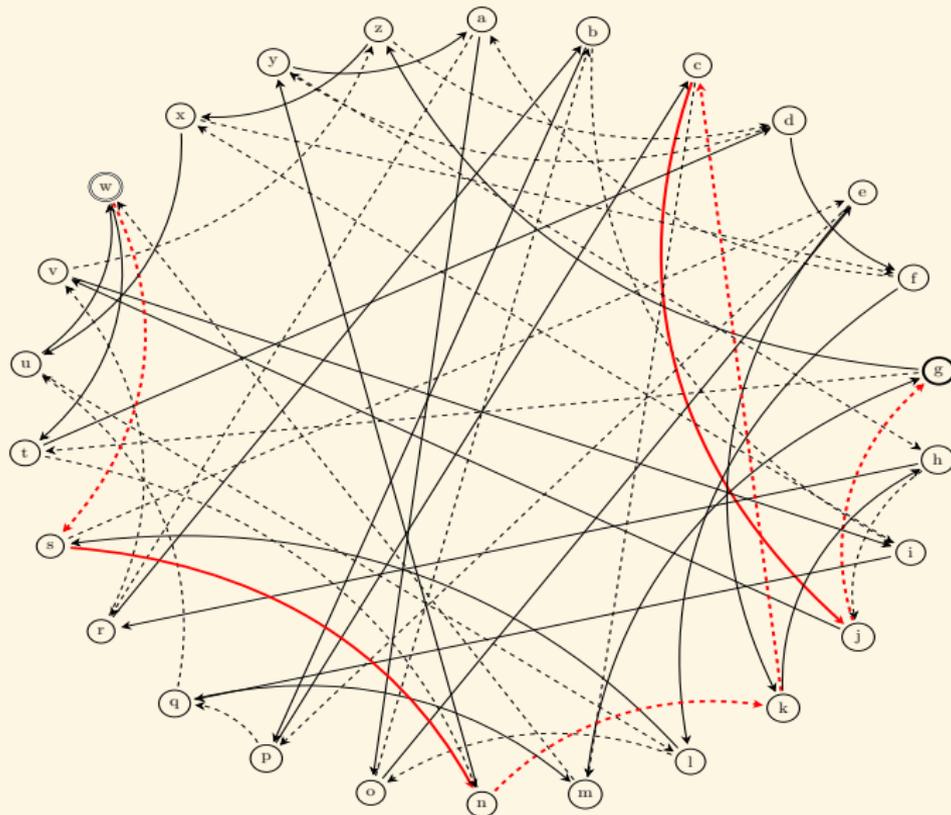
# Échange de clé par graphe

Alice part de 'l', suit le chemin 001110, et obtient 'g'.



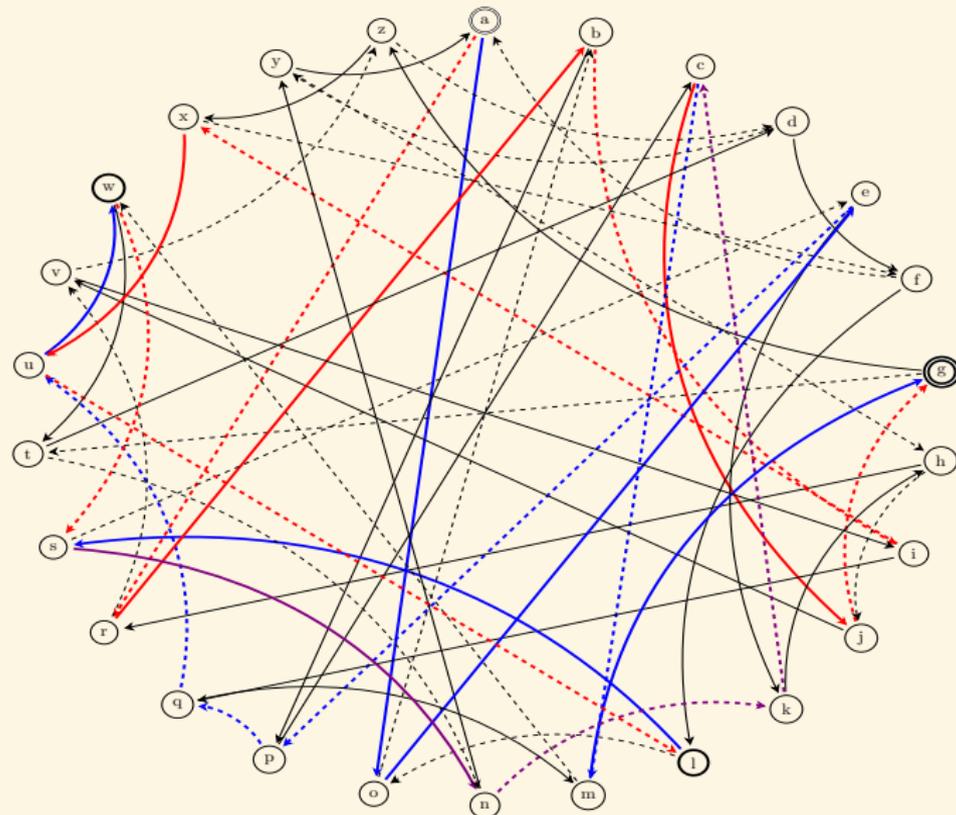
# Échange de clé par graphe

Bob part de 'w', suit le chemin 101101, et obtient 'g'.



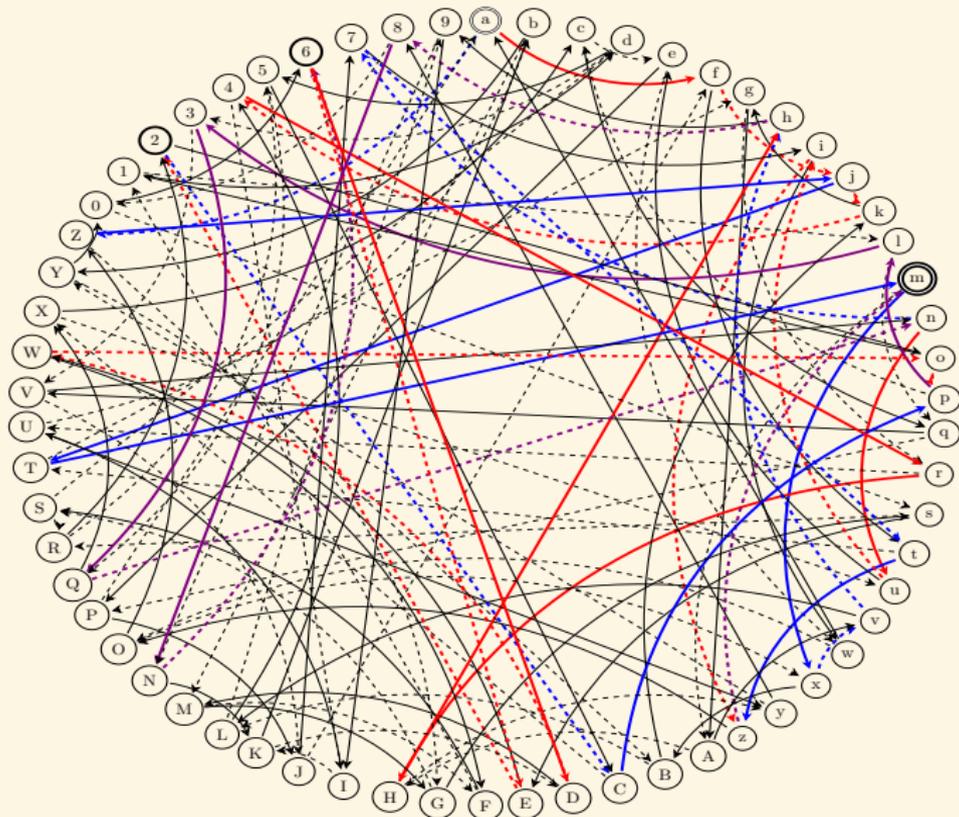
# Échange de clé par graphe

## L'échange de clé complet



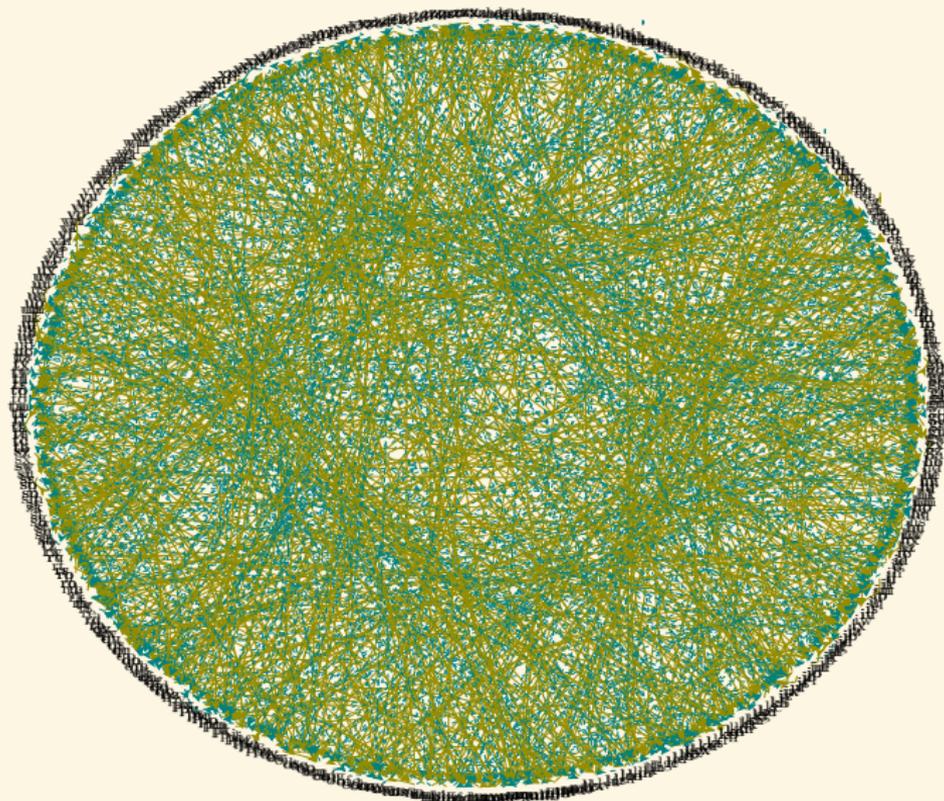
# Échange de clé par graphe

Graphe plug grand (62 noeuds)



# Échange de clé par graphe

Graphe encore plus grand (676 noeuds)



# Échange de clé par graphe

Taille cryptographique :

- Pour une sécurité de  $2^{128}$  bits, il faut un arbre avec  $n = 2^{256}$  noeuds (attaque en  $\sqrt{n}$  : on cherche un chemin en partant du point de départ et d'arrivée à la fois.)
- Il faut aussi que les arrêtes mélanges bien le graphe : en  $\log(n)$  étapes on arrive sur un noeud « uniforme » (graphe de Ramanujan).
- Le graphe ne tient pas en mémoire...Il faut un algorithme qui à partir d'un noeud donne ses voisins.



# Chiffrement à clé publique



Alice  
veut envoyer  $m$  à Bob.

Elle trouve la clé publique de chiffrement  $K_{Bob}^{pub}$  dans l'annuaire.

Elle calcule  $c = f_{K_{Bob}^{pub}}(m)$

$c$  est envoyé



à Bob

qui utilise sa clé secrète de déchiffrement  $K_{Bob}^{sec}$

$$m = f_{K_{Bob}^{pub}}^{(-1)}(c) = g_{K_{Bob}^{sec}}(c)$$



# Chiffrement à clé publique



Alice  
veut envoyer  $m$  à Bob.

Elle trouve la clé publique de chiffrement  $K_{Bob}^{pub}$  dans l'annuaire.

Elle calcule  $c = f_{K_{Bob}^{pub}}(m)$

$c$  est envoyé



à Bob

qui utilise sa clé secrète de déchiffrement  $K_{Bob}^{sec}$

$$m = f_{K_{Bob}^{pub}}^{(-1)}(c) = g_{K_{Bob}^{sec}}(c)$$



# Chiffrement à clé publique



Alice  
veut envoyer  $m$  à Bob.

Elle trouve la clé publique de chiffrement  $K_{Bob}^{pub}$  dans l'annuaire.

Elle calcule  $c = f_{K_{Bob}^{pub}}(m)$

$c$  est envoyé

à Bob



qui utilise sa clé secrète de déchiffrement  $K_{Bob}^{sec}$

$$m = f_{K_{Bob}^{pub}}^{(-1)}(c) = g_{K_{Bob}^{sec}}(c)$$



# Chiffrement à clé publique

- Trois étapes : création et publication de clés, chiffrement, déchiffrement
- Avantages : gestion de clé simplifiée, solidité mathématique
- Permet de faire des *signatures*, du chiffrement de *groupe*...
- Fragilités : plus lent, plus compliqué à implémenter

En pratique on combine les deux chiffrements : clé publique pour échanger une clé de session (secrète) qui servira à chiffrer à la volée.

Comment savoir quelle clé publique utiliser pour communiquer avec un utilisateur donné?



## Comment faire ?

- Situations asymétriques : l'un sait l'autre pas.
- Celui qui connaît le secret a un avantage (il peut déchiffrer, il peut se prouver).
- Mesurer cet avantage : **théorie de la complexité algorithmique**.
- S'appuyer sur des problèmes difficiles.



## La thèse de Turing-Church



Alan Turing



Alonzo Church

## Tests de primalité

Savoir si un entier  $P$  est premier.



Pierre de Fermat



Agrawal, Kayal et Saxena

$$T = n^{6+\epsilon(n)}$$

où  $n$  est le nombre de chiffres décimaux de  $P$ .

# Factorisation

Théorème fondamental de l'arithmétique.



Euclide



Carl Friedrich Gauss

$$N = \prod_{1 \leq i \leq l} p_i^{e_i}.$$

## Factorisation



Hendrik Lenstra



Brigitte Vallée

Factoriser un entier  $N$  prend un temps  $T = \exp(\sqrt{n})$  où  $n$  est le nombre de chiffres décimaux de  $n$ . (Algorithme heuristique :  $T = \exp(n^{1/3})$ )



$$(p, q) \xrightarrow{\text{green}} N = pq$$

$$(p, q) \xleftarrow{\text{red}} N = pq$$

- En décembre 2009, Thorsten Kleinjung et une dizaine de collègues ont factorisé un nombre de 232 décimales.
- *The sieving, which was done on many hundreds of machines, took almost two years.*
- Calculer le produit de deux nombres de 116 décimales prend 8 millièmes de secondes sur mon ordinateur portable.



## Protocole RSA



Rivest, Shamir et Adleman

# Protocole RSA

- Soit  $N = pq$  un produit de deux grands nombres premiers ;
- Soit  $e$  premier à  $\phi(N) = (p - 1)(q - 1)$  et  $d$  l'inverse de  $e$  modulo  $\phi(N)$  ;
- **Chiffrement** :  $x \mapsto x^e \pmod N$  ;
- **Déchiffrement** :  $x \mapsto x^d \pmod N$  ;

## Théorème (Petit théorème de Fermat)

$$x^{\#(\mathbb{Z}/N\mathbb{Z})^\times} = 1 \pmod N.$$



# Exponentielle et logarithme discrets

$p = 7$  un nombre premier et  $b = 5 \pmod{7}$

$x$	$b^x$
1	$5 \pmod{7}$
2	$4 \pmod{7}$
3	$6 \pmod{7}$
4	$2 \pmod{7}$
5	$3 \pmod{7}$
6	$1 \pmod{7}$
7	$5 \pmod{7}$

$y = b^x$	$x = \log_b(y)$
1	$0 \pmod{6}$
2	$4 \pmod{6}$
3	$5 \pmod{6}$
4	$2 \pmod{6}$
5	$1 \pmod{6}$
6	$3 \pmod{6}$

$$\exp_b : \mathbb{Z}/(p-1)\mathbb{Z} \rightarrow (\mathbb{Z}/p\mathbb{Z})^*$$

$$\log_b : (\mathbb{Z}/p\mathbb{Z})^* \rightarrow \mathbb{Z}/(p-1)\mathbb{Z}$$

# Exponentielle et logarithme discret : coûts algorithmiques

- Calcul de  $b^x$  :  $O(\log x)$  multiplications :
  - ▶  $b^{2x} = (b^2)^x$
  - ▶  $b^{2x+1} = b^{2x} \times b$
- Calcul de  $\log_b(g)$  (méthode générique) :  $O(\sqrt{\ell})$  opérations, où  $\ell$  est le plus grand facteur premier de  $\sqrt{\ell}$  (restes Chinois + relevé de Hensel + pas de bébés pas de géants);
- Calcul de  $\log_b(g)$  (en utilisant la structure de  $\mathbb{Z}/p\mathbb{Z}^*$  pour travailler sur des bases de facteurs de nombres friables, comme pour les algorithmes de factorisation) :  $\exp(p^{1/3})$ .



# Exponentielle et logarithme discret : coûts algorithmiques

- Calcul de  $b^x$  :  $O(\log x)$  multiplications :
  - ▶  $b^{2x} = (b^2)^x$
  - ▶  $b^{2x+1} = b^{2x} \times b$
- Calcul de  $\log_b(g)$  (méthode générique) :  $O(\sqrt{\ell})$  opérations, où  $\ell$  est le plus grand facteur premier de  $\sqrt{\ell}$  (restes Chinois + relevé de Hensel + pas de bébés pas de géants);
- Calcul de  $\log_b(g)$  (en utilisant la structure de  $\mathbb{Z}/p\mathbb{Z}^*$  pour travailler sur des bases de facteurs de nombres friables, comme pour les algorithmes de factorisation) :  $\exp(p^{1/3})$ .



# Échange de clé de Diffie Hellmann et chiffrement El Gamal

Alice et Bob veulent échanger une clé commune via un canal non sécurisé.

- On part de  $g \bmod p$ ;
- Alice choisit  $a \bmod p - 1$  et envoie  $g^a \bmod p$ ;
- Bob choisit  $b \bmod p - 1$  et envoie  $g^b \bmod p$ ;
- Le clé commune est  $g^{ab} = (g^a)^b = (g^b)^a \bmod p$ .

## Remarque

*Cas particulier d'échange de clé sur les graphes : les noeuds sont les  $g^i$ , et on a deux types d'arrêtes :  $g^i \mapsto g^{i+1}$  et  $g^i \mapsto g^{2i}$ .*

Chiffrement :

- Clé publique d'Alice :  $(g, g^a)$ , Clé secrète d'Alice :  $a$ .
- Chiffrement : Bob choisit un  $r$  aléatoire, et envoie  $(g^r, m \times g^{ar})$ ;
- Déchiffrement : Alice calcule  $g^{ar}$  ce qui lui permet de retrouver  $m$ .

## Remarque

*S'étend à n'importe quel groupe cyclique  $G = \langle g \rangle$ .*

# Échange de clé de Diffie Hellmann et chiffrement El Gamal

Alice et Bob veulent échanger une clé commune via un canal non sécurisé.

- On part de  $g \bmod p$ ;
- Alice choisit  $a \bmod p - 1$  et envoie  $g^a \bmod p$ ;
- Bob choisit  $b \bmod p - 1$  et envoie  $g^b \bmod p$ ;
- Le clé commune est  $g^{ab} = (g^a)^b = (g^b)^a \bmod p$ .

## Remarque

*Cas particulier d'échange de clé sur les graphes : les noeuds sont les  $g^i$ , et on a deux types d'arrêtes :  $g^i \mapsto g^{i+1}$  et  $g^i \mapsto g^{2i}$ .*

Chiffrement :

- Clé publique d'Alice :  $(g, g^a)$ , Clé secrète d'Alice :  $a$ .
- Chiffrement : Bob choisit un  $r$  aléatoire, et envoie  $(g^r, m \times g^{ar})$ ;
- Déchiffrement : Alice calcule  $g^{ar}$  ce qui lui permet de retrouver  $m$ .

## Remarque

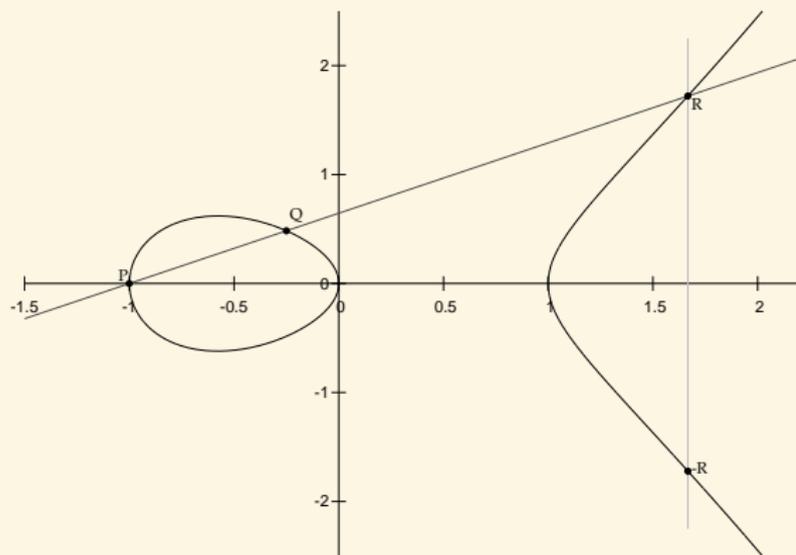
*S'étend à n'importe quel groupe cyclique  $G = \langle g \rangle$ .*

# Les courbes elliptiques

## Définition (char $k \neq 2, 3$ )

Une courbe elliptique est une courbe plane d'équation

$$y^2 = x^3 + ax + b \quad 4a^3 + 27b^2 \neq 0.$$



Exponentiation :

$$(\ell, P) \mapsto \ell P$$

Logarithme discret :

$$(P, \ell P) \mapsto \ell$$

# Utilisation des courbes elliptiques

## Exemple (ECC 160 bits)

- $E$  courbe elliptique  $y^2 = x^3 + x + 333$  sur  $\mathbb{F}_{1461501637330902918203684832716283019655932542983}$

- Clé publique :

$$P = (1369962487580788774992199588498961558341362086296, \\ 407160203592982096299905031630798490942043935021);$$

$$Q = (69569756243634326598411303228618910556938958980, \\ 1126203611660190221708449639677667925024412968395);$$

- Clé secrète :  $\ell$  tel que  $Q = \ell P$ .
- Utilisées par la NSA ;
- Utilisées dans les passeports biométriques Européens.
- Plus rapide, compact et puissant (couplages) que RSA.



# ECC contre RSA pour 128 bits de sécurité

- ECC (Curve25519) 256 bits :

AAAC3NzaC11ZDI1NTE5AAAAIMoNrNYhU7CY1Xs6v4Nm1V6oRHs/FEe8P+XaZ0PcxPzz

- RSA 3248 bits :

MIHRgIBAAKCAZcAv1GW+b5L2tmqb5bUJMrfLHgr2jga/Q/8IJ5QJqeSsB7xLVT/  
ODN3KNSPxyjaHmDNDtWgSikZvPYeyZWfLp0B0vgWdQugUGHvfg473Z0lqZk6  
1nA45XZGHUPT98p4+ghPag5JyvAVs1fCf/Vl1tBHbu/NOYIAC4F3tHP81nn+10nB  
e1lEALbdmVgTTZ5jcrRrt4IDT5a4IeI9yTe0aVdTsuJ6990hpkRvZyTou1eoxp5eV  
KQ7aIX6es9Xjnr8widZunM8rqhBw9EMmlqabnXZITPQoV3rUAnwKzDLV7E56viJk  
S2xU5+95IctYU/RTTbf3wTxnkdOqxId0MONHyBjsukXgYKxvB1fwhBKZ4tWu1iGw  
UC1iK7qLm12zJhLn4WovaxrvvtX008250xncEfYDXyU4xbRnJn+ZsT7tququfC1M  
U4MYRDwy7uj+H1EmIGul69Fw9NkuCItWI9dFpcDtSP+/1eEN7wc2F1xhDIRwer0F  
6I1P4S2wn1uQyHzsTLVdcp+rqa1AsvbwBCKL4ravEO2CEQIDAQABAOIB11wt5YoJ  
YZzk4RcnbSX/LvmwICfdmkjTKW6F1w+P4TnotCr0wPG00bDoANJoUrcXnsSqNGMgC  
015F8q9+UuDwZx4KBZm0j8IPOPzJ2nYck5dYDhyMHZDq1LJ4zJfgPQGQ5Wwq2Bwm  
2RHDhAdDtTh6YZArs/z9hAqtA9gqMPnMPcdQpIv1sH50n06zBJD8sJQA+k0XG+Y2  
GS8NakLcUV1DpNd/Q+QHkv4Aw1ge2EF8QvmKtu/9reK0BqWnm2Tpad6RtAhZwPJX  
Uhd9yiesTF6rjZ1ZcMGXUaNSRt0zD3D4zowRz2JLTCe4GkiJmtc3waN6hu1IaIzq  
boI11evqnbatqnc4rCq8sf21yZqaLUIbH41W2G3K8xMJN3iY8cgHTYneNYa+/d  
7xyNW1M09SK1HsyaPcWv98BdD+At0x/6R6YPykeR+qXJ9ETGfKw4U6iNbBQXOMB  
kZb1Ry8vFMH8vsYIzh8Edg6aq00S5cU57KiD5/Gc8KuqI6vmf21cCdCa487kCgW6  
cGXQ2bLZGYBiMZfFO01pCQECgcwA5ZU3/8yS0duNhsDz3sgC2u40HwHUBxu50Ua  
a5t4CoUY9iuf7b7qhbEcvdLgI01XA5xo+r4p0xgbLVdUTsRR1mrDM2+wrCjjwXCw  
pAFMR12Rr72yLUC7N0WnNcoUshrNL4x/1j8T4WLRcannpXcor+/kn1rdwLEbrRc+  
zRTAdJlMPT4kwJehTE9Mzw2/03GX3MeLzVzJk1zvpCGw20N/2Yqjs++v5hXoHPs  
21y6y6/FV097dvCtcf7NahS04Jsjubfnj0Mx89AUNZsCgcwA1DfabCGJ5Ckmq+mg  
2q91DPjz6r29wmBtYyT20oZ2kd4QBHR0p0t59yG4bvdRqZG/Dr5LJdVdVFRkroh1  
/JJ7rLz/ZBQCLRS5t7/G2B0kBDOMMM+02wR60CTmxUhmGvs0ZWRp5Kkha5PSvZa  
Wau2CN3mXNK72RL3RFUvuhNynkOEj50au1RaGgpZ0B0JTKYI9nffbeSxU+DV8MC  
gcwA18be28T15FXy+/IGQ3EBHFucTItDQQA2Ew/8pTfK+z0kr9yYISsKXUuaSk  
+skghkPcruGw8LgabH4GT/zGu+1H4btyekSBxeCtFqTtpED1WJOWD2ozi7NXSjd  
YrhF+VCcMCA7ekOqS5HjkmT4XMO/wPab4VFEKzLnHzQ1cZB3ke7/4/0HnD5cIE7  
vWNeRcDvdYRggT+wBX+Y6bXP142Smj8uyu1oDmpmR5ZUCnTdqT408K/RT0x4jCeC  
CUHgv5rVil107bS4CdkCgctXvnQwCzmvvVrV744Tftuh81TwHnqGwaA/LKU3wW9  
T/x9ba1uHFHXkaWvRba61LiCDGPsYM4hwTYokqYnfbC2rvOWO6rtnX1P1An3y61V  
ovQfgDeNiFmIyvniPPEm0JZA+QnburLYwOx4DgwYvyBnpa18WP08c3L/J4hkWlm  
Pc30DJ0XUumLevAnCv0jvgSfW8NenSVfzw+KToIEKaP0rWfJZUWDA479vY6D  
UNwRjPntYIwtSAV+PpRvINk0ZeHamW9H+D1cWkBy2euc93gruYdtFej/biGASD  
+Hh... (truncated)



# Identification par mot de passe



Alice  
mot de passe d'Alice **BELOTE**

**BELOTE** est envoyé



à Bob  
mot de passe de Bob **REBELOTE**

**REBELOTE** est envoyé à Alice



# Identification par mot de passe

- Alice et Bob doivent convenir d'un mot de passe secret partagé (question secrète)
- Avantage : simple
- Fragilités : risque de réutilisation e.g. par un tiers, gestion de mots de passe



# Identification sans divulgation de connaissance



Alice  
connaît un secret  $S_{Alice}$

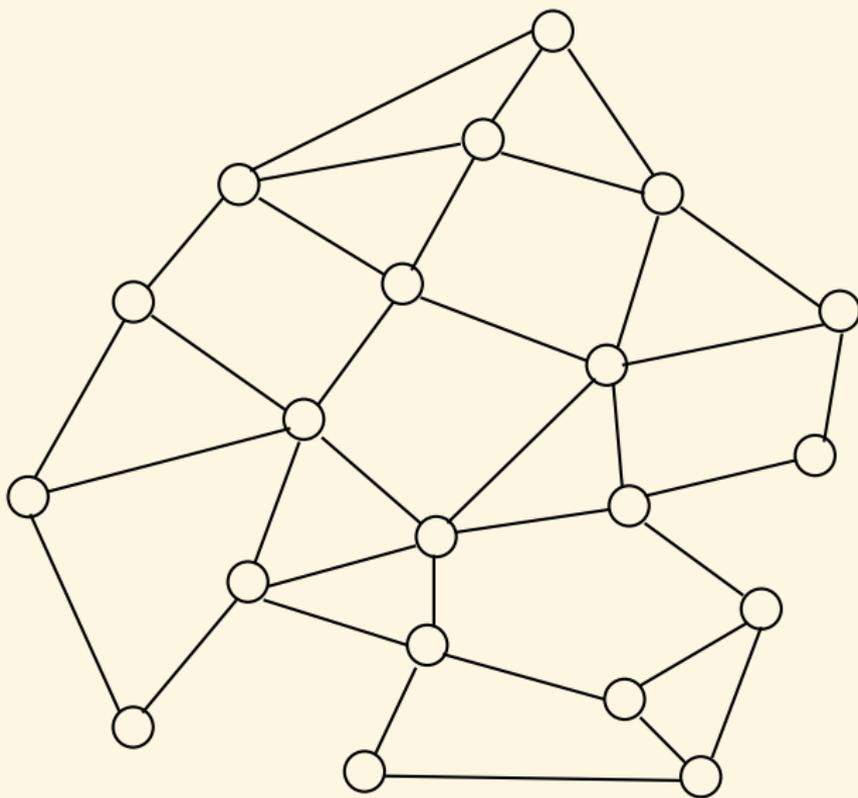


Bob

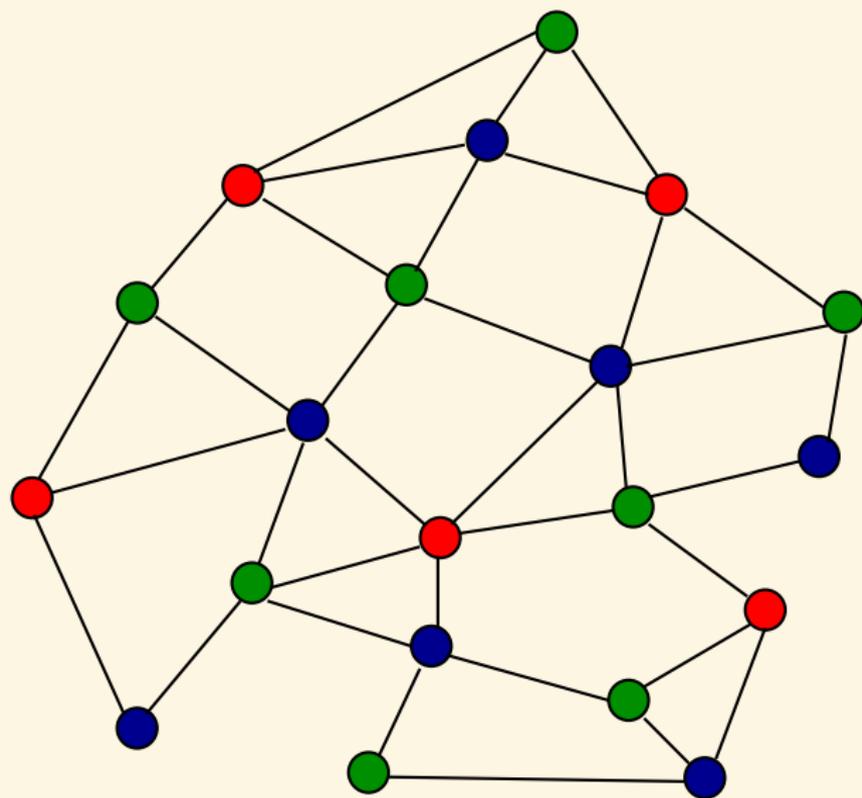
interroge Alice et se convainc qu'elle connaît bien le secret.

À la fin de l'échange, Bob n'a rien appris sur ce secret!

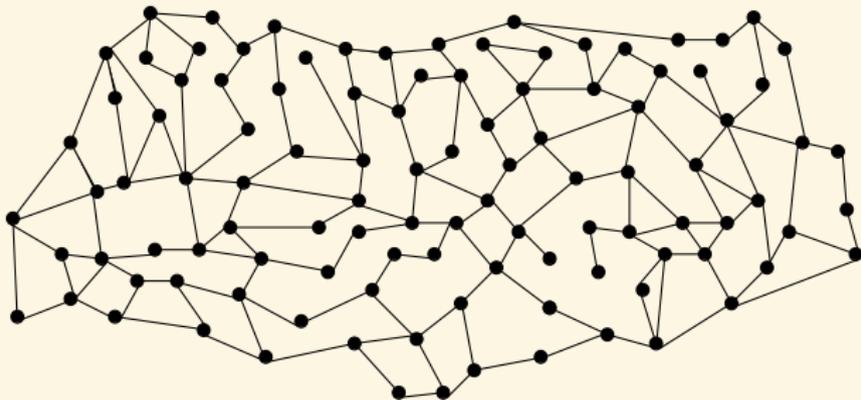
## Coloriage de graphe



## Coloriage de graphe



# Coloriage de graphe



# Zero Knowledge Proofs

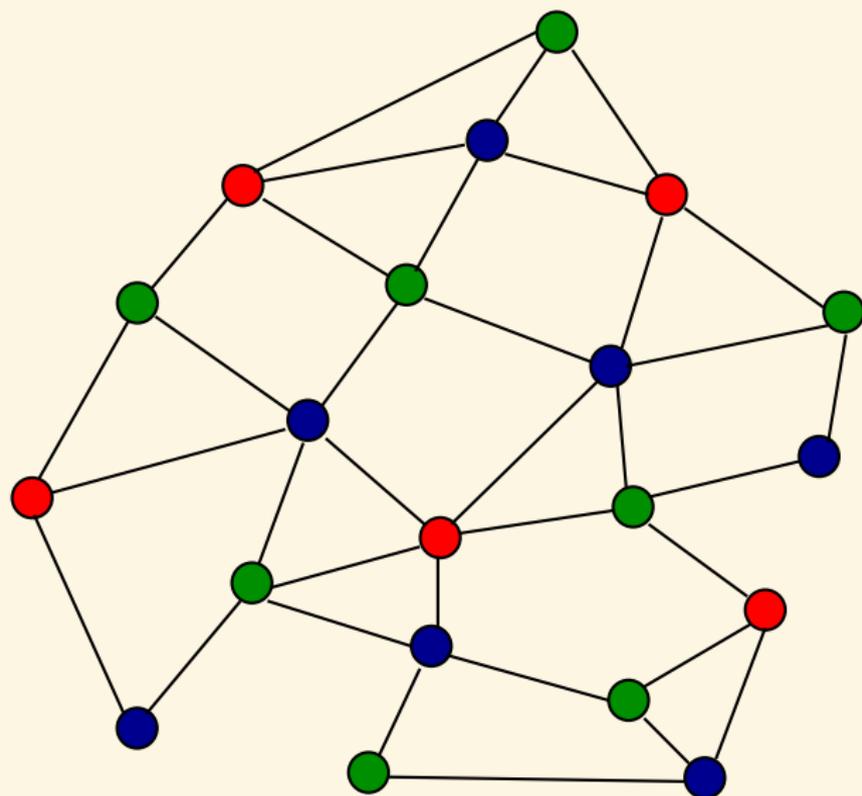


Shafi Goldwasser (1981)

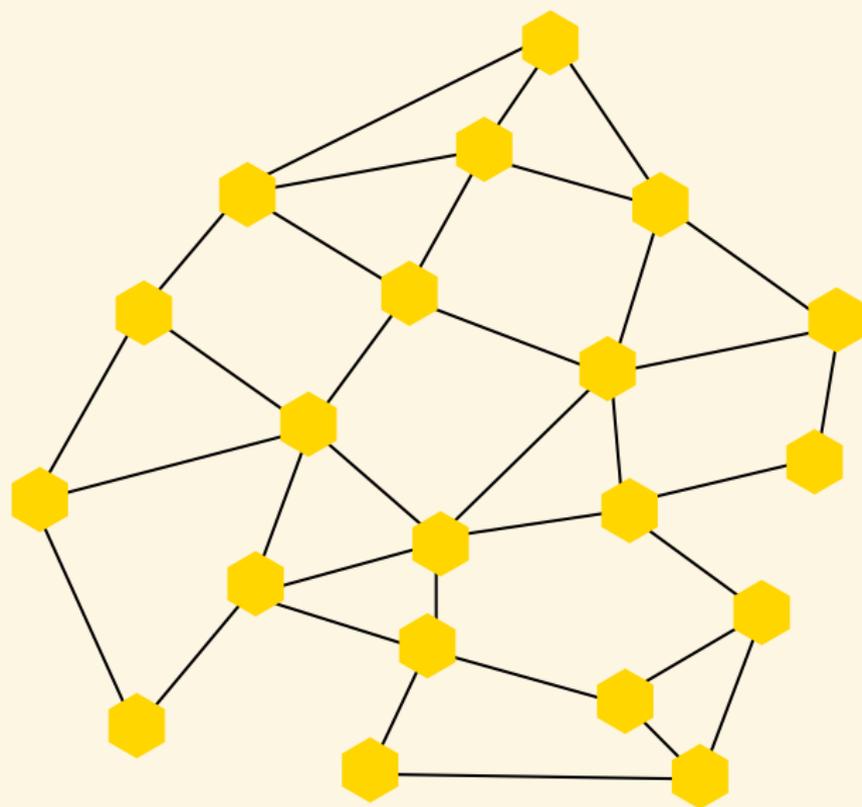


Oded Goldreich (1991)

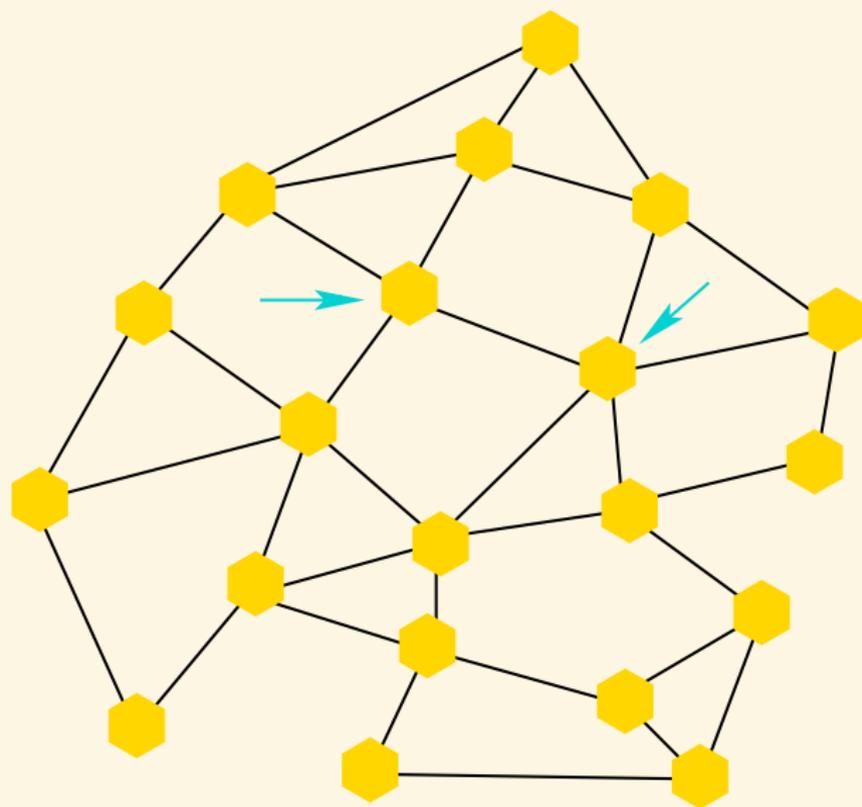
## Le coloriage d'Alice (secret)



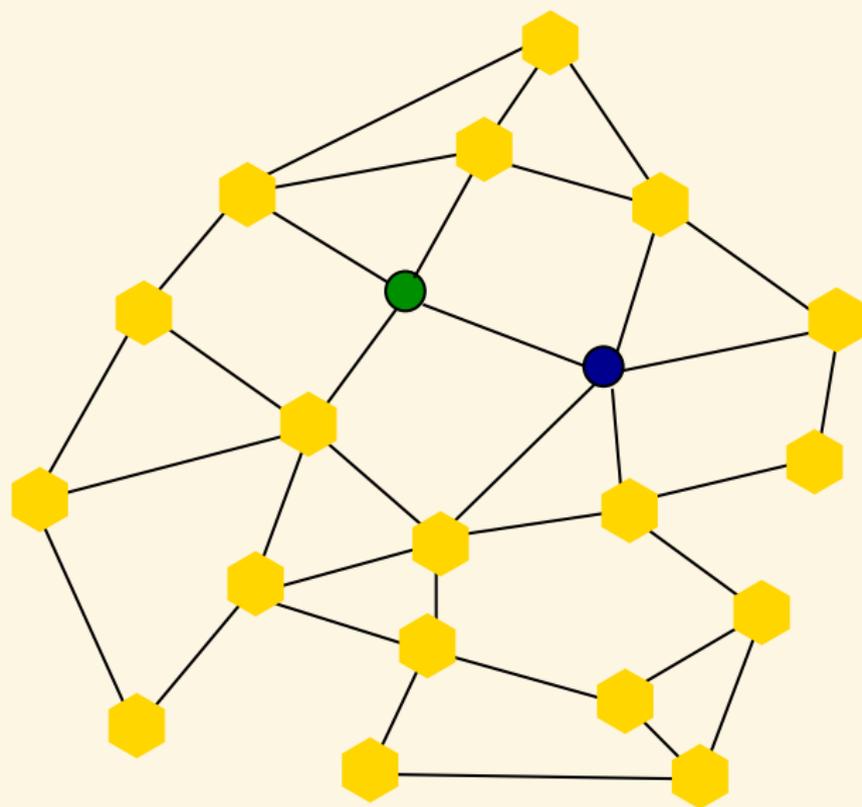
## Le coloriage d'Alice caché



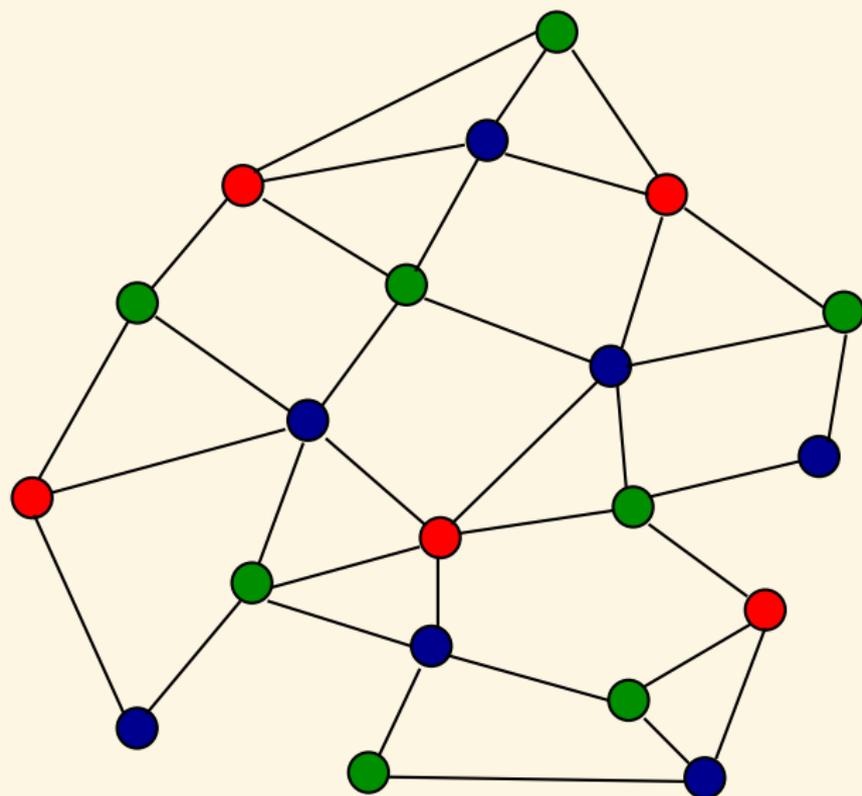
## La question de Bob



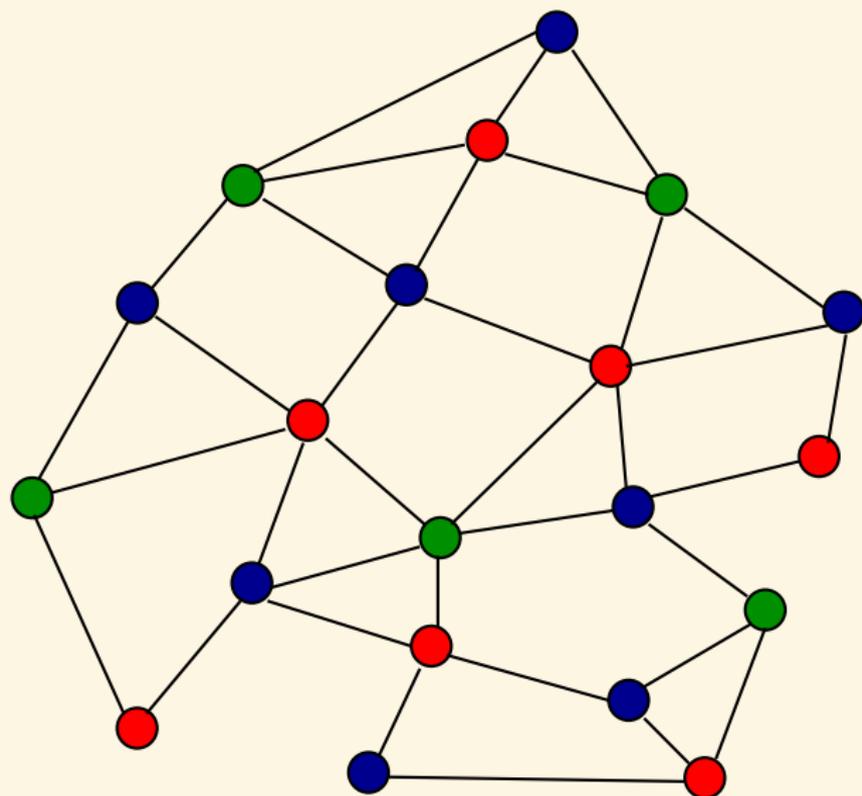
# Dévoilement



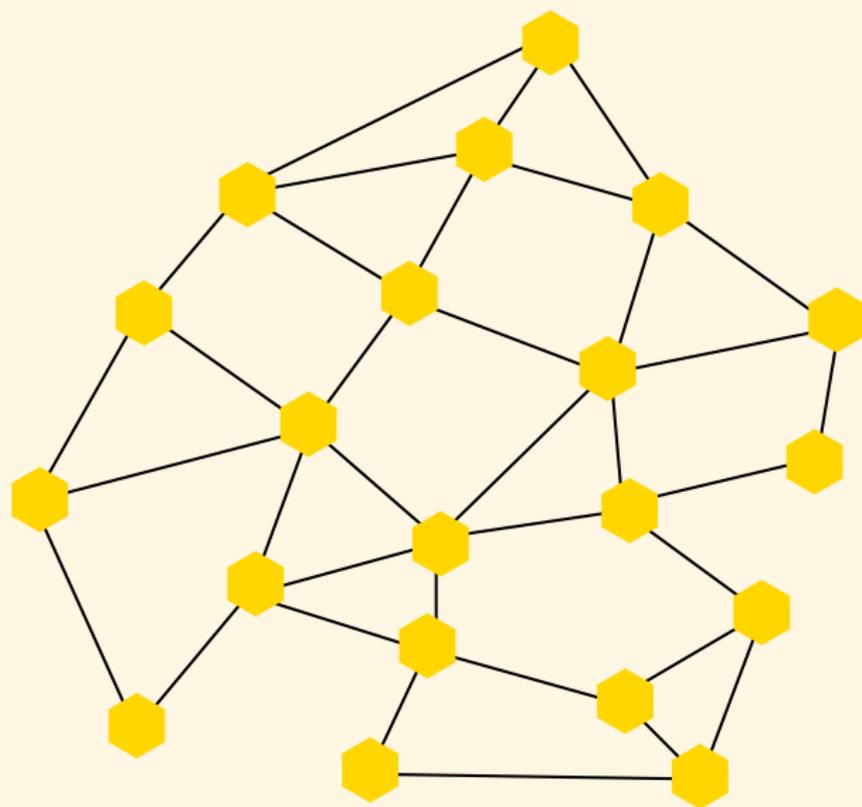
## Le coloriage d'Alice (secret)



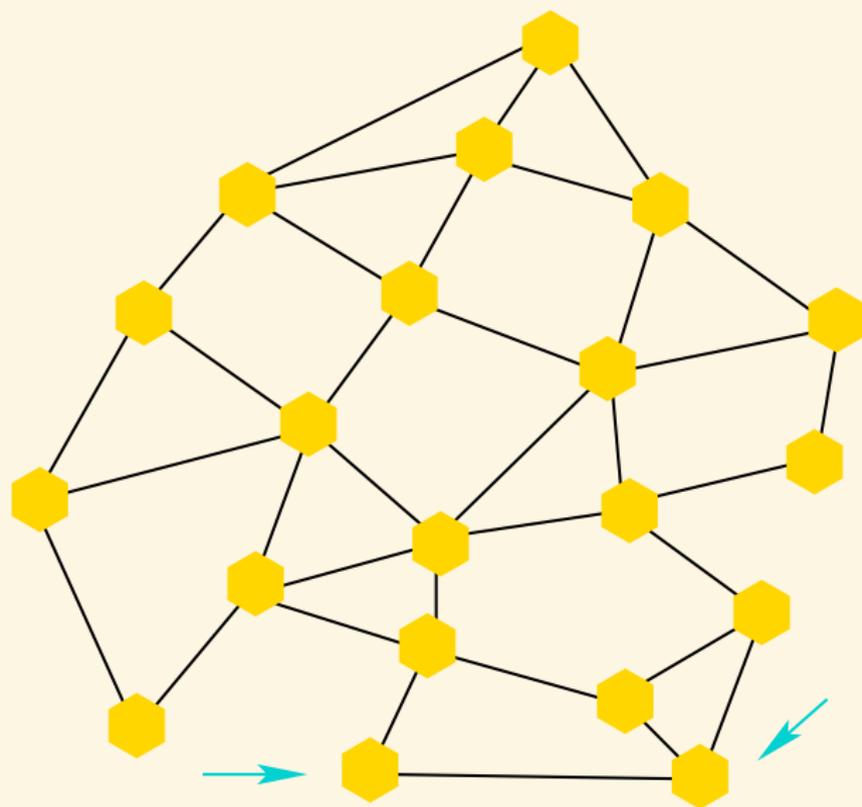
## Le coloriage d'Alice permuté



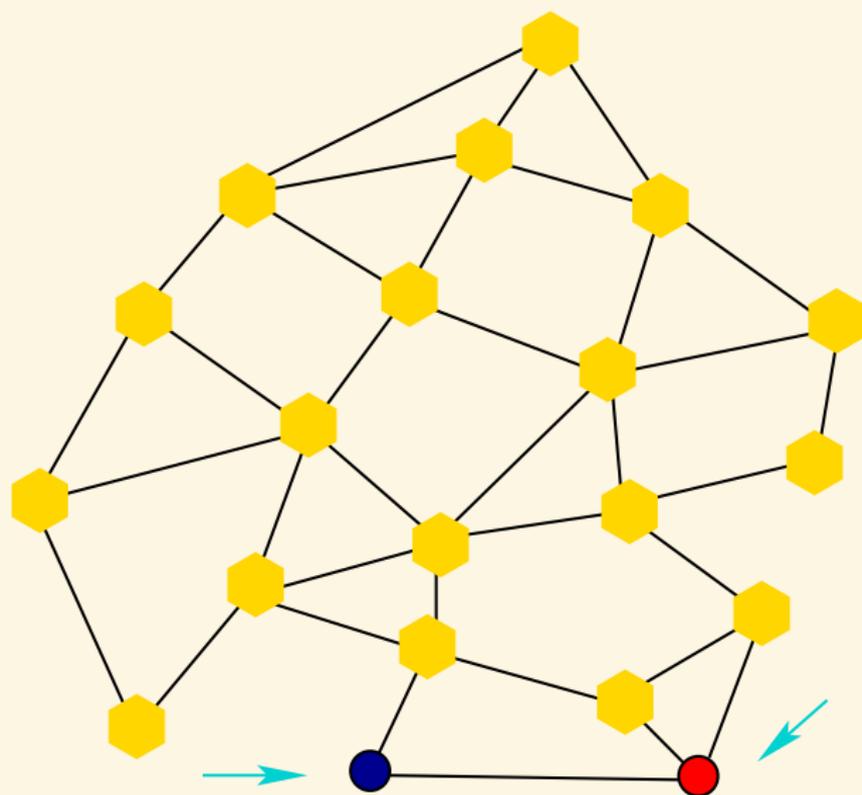
## Le coloriage d'Alice caché



## La deuxième question de Bob



# Dévoilement



## Carrés dans $\mathbb{Z}/p\mathbb{Z}$

- Soit  $p > 2$  un nombre premier.  $(\mathbb{Z}/p\mathbb{Z}^*, \times)$  est un groupe cyclique d'ordre  $p - 1$  ;
- Il y a  $(p - 1)/2$  carrés et  $(p - 1)/2$  non carrés ;
- Si  $x \in \mathbb{Z}/p\mathbb{Z}^*$  alors  $x$  est un carré si et seulement si  $x^{\frac{p-1}{2}} = 1$ .
- Il existe des algorithmes efficaces pour calculer des racines carrés.

### Démonstration.

- Comme  $\mathbb{Z}/p\mathbb{Z}$  est un corps, l'équation  $X^2 - a$  a au plus deux solutions. On vérifie qu'il y a deux solutions distinctes si  $a$  est un carré (non nul), et 0 sinon. Il y a donc  $(p - 1)/2$  carrés non nuls. (Plus rapide :  $x \rightarrow x^2$  est un morphisme du groupe multiplicatif de  $\mathbb{Z}/p\mathbb{Z}$  et de noyau  $\pm 1$ .)
- Par le petit théorème de Fermat  $x^{p-1} = 1$  pour  $x \in \mathbb{Z}/p\mathbb{Z}^*$ . Ainsi, si  $y = x^2$  est un carré,  $y^{\frac{p-1}{2}} = 1$ . Donc les carrés sont solutions de  $X^{\frac{p-1}{2}} = 1$ , or il y a au plus  $(p - 1)/2$  solutions, donc toutes les solutions sont des carrés.



## Carrés dans $\mathbb{Z}/p\mathbb{Z}$

- Soit  $p > 2$  un nombre premier.  $(\mathbb{Z}/p\mathbb{Z}^*, \times)$  est un groupe cyclique d'ordre  $p - 1$  ;
- Il y a  $(p - 1)/2$  carrés et  $(p - 1)/2$  non carrés ;
- Si  $x \in \mathbb{Z}/p\mathbb{Z}^*$  alors  $x$  est un carré si et seulement si  $x^{\frac{p-1}{2}} = 1$ .
- Il existe des algorithmes efficaces pour calculer des racines carrés.

### Démonstration.

- Comme  $\mathbb{Z}/p\mathbb{Z}$  est un corps, l'équation  $X^2 - a$  a au plus deux solutions. On vérifie qu'il y a deux solutions distinctes si  $a$  est un carré (non nul), et 0 sinon. Il y a donc  $(p - 1)/2$  carrés non nuls. (Plus rapide :  $x \rightarrow x^2$  est un morphisme du groupe multiplicatif de  $\mathbb{Z}/p\mathbb{Z}$  et de noyau  $\pm 1$ .)
- Par le petit théorème de Fermat  $x^{p-1} = 1$  pour  $x \in \mathbb{Z}/p\mathbb{Z}^*$ . Ainsi, si  $y = x^2$  est un carré,  $y^{\frac{p-1}{2}} = 1$ . Donc les carrés sont solutions de  $X^{\frac{p-1}{2}} = 1$ , or il y a au plus  $(p - 1)/2$  solutions, donc toutes les solutions sont des carrés.



## Symbole de Legendre

- Symbole de Legendre :

$$\left(\frac{x}{p}\right) = \begin{cases} 0 & \text{si } x = 0 \pmod{p} \\ 1 & \text{si } x \text{ est un carré} \\ -1 & \text{si } x \text{ n'est pas un carré} \end{cases}$$

- $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \pmod{p}$ ;
- Multiplicativité :  $\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right) \left(\frac{y}{p}\right)$ ;
- Réciprocité quadratique ( $p, q$  premiers  $> 2$ ) :

$$\left(\frac{p}{q}\right) \left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2} \frac{q-1}{2}}.$$

## Carrés dans $\mathbb{Z}/n\mathbb{Z}$

- Soit  $n = pq$  un nombre RSA, par le théorème chinois :

$$(\mathbb{Z}/n\mathbb{Z}^*, \times) = (\mathbb{Z}/p\mathbb{Z}^* \times \mathbb{Z}/q\mathbb{Z}^*, \times);$$

- Il y a donc 4 racines carrés de  $-1$  : qui correspondent via l'isomorphisme à

$$(1, 1), (-1, -1), (1, -1), (-1, 1).$$

- Si on connaît la factorisation de  $n$ , on peut calculer des racines carrés en ce ramenant aux racines carrés modulo  $p$  et modulo  $q$ .
- Réciproquement, savoir calculer des racines carrés permet de factoriser  $n$ .  
(Preuve : si  $x_1$  et  $x_2$  sont deux racines carrés de  $y$ , alors il y a une chance sur deux que le pgcd de  $x_1 - x_2$  et de  $n$  donne  $p$  ou  $q$ ).



## Symbole de Jacobi

- Symbole de Jacobi : si  $n$  est impair, le symbole de Jacobi est l'extension du symbole de Legendre par multiplicativité de l'argument du bas :

$$\left(\frac{x}{n_1 n_2}\right) = \left(\frac{x}{n_1}\right) \left(\frac{x}{n_2}\right);$$

- Réciprocité quadratique générale :

$$\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2} \frac{n-1}{2}} \left(\frac{n}{m}\right) \quad (m \text{ et } n \text{ impairs et premiers})$$

avec les équations supplémentaires  $\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$ ,  $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$  ;

⇒ Le symbole de Jacobi peut être calculé en temps polynomial ;

- Test de primalité : si  $\left(\frac{x}{n}\right) \neq x^{\frac{n-1}{2}}$  alors  $n$  n'est pas premier (et réciproquement si  $n$  n'est pas premier alors au moins la moitié des  $x$  premiers à  $n$  seront des témoins de non primalité).



## Pile ou face

- Soit  $n = pq$  un nombre RSA,  $(\mathbb{Z}/n\mathbb{Z}^*, \times) = (\mathbb{Z}/p\mathbb{Z}^* \times \mathbb{Z}/q\mathbb{Z}^*, \times)$ ;
- $\left(\frac{x}{n}\right) = \left(\frac{x}{p}\right) \left(\frac{x}{q}\right)$  donc si  $x$  est premier à  $n$ ,  $\left(\frac{x}{n}\right) = 1$  quand  $x$  est un carré modulo  $n$  (=carré modulo  $p$  et carré modulo  $q$ ) **ou** quand  $x$  n'est un carré ni modulo  $p$  ni modulo  $q$ ;
- Calcul de  $\left(\frac{x}{n}\right)$  : **temps polynomial**;
- Décider si  $x$  est un vrai carré (et calculer la racine carré) ou un faux carré : **factorisation de  $n$**
- $x \mapsto x^2$  est une fonction à sens unique!

### Pile ou face :

- Bob choisit  $n = pq$  et envoie  $x$  tel que  $\left(\frac{x}{n}\right) = 1$ ;
- Alice répond “vrai carré” ou “faux carré”;
- Bob envoie  $p$  et  $q$  pour qu’Alice puisse vérifier si elle avait raison ou non.



# Identification sans divulgation d'informations

- Clé secrète d'Alice :  $p, q, s \bmod n = pq$ ;
- Clé publique d'Alice :  $n = pq, r = s^2$ ;

## Identification Zero Knowledge :

- Alice choisit  $u \bmod n$  aléatoirement, calcule  $z = u^2$  et envoie  $t = zr = u^2s^2$  à Bob;
- Bob choisit soit
  - ▶ De vérifier  $z$  : il demande  $u$  à Alice et vérifie que  $z = u^2$ ;
  - ▶ De vérifier  $t$  : il demande  $us$  à Alice et vérifie que  $t = (us)^2$ .
- Un usurpateur va produire soit un faux  $u$  soit un faux  $t$  et a une chance sur deux de se faire attraper.
- Bob va demander plusieurs tours de vérification (par exemple 30) Si Alice donne toujours la bonne réponse, soit elle connaît le secret, soit elle est très chanceuse (probabilité  $1/2^{30}$ ).



# Identification sans divulgation d'informations

- Clé secrète d'Alice :  $p, q, s \bmod n = pq$ ;
- Clé publique d'Alice :  $n = pq, r = s^2$ ;

## Identification Zero Knowledge :

- Alice choisit  $u \bmod n$  aléatoirement, calcule  $z = u^2$  et envoie  $t = zr = u^2s^2$  à Bob;
- Bob choisit soit
  - ▶ De vérifier  $z$  : il demande  $u$  à Alice et vérifie que  $z = u^2$ ;
  - ▶ De vérifier  $t$  : il demande  $us$  à Alice et vérifie que  $t = (us)^2$ .
- Un usurpateur va produire soit un faux  $u$  soit un faux  $t$  et a une chance sur deux de se faire attraper.
- Bob va demander plusieurs tours de vérification (par exemple 30) Si Alice donne toujours la bonne réponse, soit elle connaît le secret, soit elle est très chanceuse (probabilité  $1/2^{30}$ ).



# Algorithmes quantiques : calcul de période

## Période

Soit  $f : \mathbb{Z}/N\mathbb{Z} \rightarrow X$  une fonction périodique de période  $r$ . But : trouver  $r$ .

Algorithme classique pour trouver  $r : O(N)$ . Algorithme quantique :  $O(\log N^2)$ .

## Démonstration.

- Transformée de Fourier rapide quantique pour calculer  $\hat{f}$  (en superposition)
- Une mesure de l'état de  $\hat{f}$  donne un multiple  $\lambda.r$
- On recommence pour trouver différents multiples  $\lambda_i.r$ , puis on prend leur pgcd ; avec une grande probabilité on trouve  $r$ .



- Application : factorisation en temps polynomial.
- Preuve : si  $x \in (\mathbb{Z}/N\mathbb{Z})^*$ , et que son ordre est  $k$  (pair), alors  $x^{k/2}$  est une racine carré de  $x$ , donc on peut calculer des racines carrés en temps polynomial, donc factoriser  $N$  en temps polynomial.



# Algorithmes quantiques : calcul de période

## Période

Soit  $f : \mathbb{Z}/N\mathbb{Z} \rightarrow X$  une fonction périodique de période  $r$ . But : trouver  $r$ .

Algorithme classique pour trouver  $r : O(N)$ . Algorithme quantique :  $O(\log N^2)$ .

## Démonstration.

- Transformée de Fourier rapide quantique pour calculer  $\hat{f}$  (en superposition)
- Une mesure de l'état de  $\hat{f}$  donne un multiple  $\lambda.r$
- On recommence pour trouver différents multiples  $\lambda_i.r$ , puis on prend leur pgcd ; avec une grande probabilité on trouve  $r$ .



- Application : factorisation en temps polynomial.
- Preuve : si  $x \in (\mathbb{Z}/N\mathbb{Z})^*$ , et que son ordre est  $k$  (pair), alors  $x^{k/2}$  est une racine carré de  $x$ , donc on peut calculer des racines carrés en temps polynomial, donc factoriser  $N$  en temps polynomial.



## Algorithmes quantiques : le sous-groupe caché

- Généralisation : le problème du sous-groupe caché :

$$f : G \rightarrow X$$

But : retrouver le plus grand sous-groupe  $H$  tel que

$$f : G \rightarrow G/H \rightarrow X$$

- Algorithme quantique pour résoudre le HSP sur un groupe abélien fini en temps **polynomial**. (Si  $G$  est cyclique il s'agit de trouver une période, pour un groupe abélien celà revient à trouver plusieurs périodes indépendantes.)
- Utilise la transformée de Fourier (rapide) multidimensionnelle ( $\hat{G}$  : groupe des caractères de  $G$ ).
- Casse le logarithme discret : si  $h = g^x$ , alors  $(n, m) \mapsto h^n g^{-m}$  a pour périodes  $\langle (1, x) \rangle$ .



## Étendre l'échange de clé de Diffie-Hellmann

- Si  $G$  est un groupe abélien agissant sur  $X$ .
- On fixe un point base  $x \in X$ .
- Alice choisit un secret  $a \in G$  et envoie  $a.x$ ;
- Bob choisit un secret  $b \in G$  et envoie  $b.x$ ;
- La clé commune est  $ab.x = ba.x \in X$ .

### Exemple

Échange de clé sur le graphe de Cayley d'un groupe abélien.



## Algorithmes quantiques : le problème du décalage caché

- $G$  agit sur  $X$ ,  $f, g$  deux fonctions  $X \rightarrow Y$  telles que

$$\exists s \in G \mid \forall x \in X, f(x) = g(s.x).$$

- But : retrouver  $s$ .
  - Algorithme quantique polynomial si  $G$  est cyclique ;
  - Algorithme quantique sous-exponentiel si  $G$  est abélien ;
  - Pas d'algorithmes sous-exponentiels connus si  $G$  n'est pas abélien ;
- ⇒ Échange de clé post-quantique : Graphe d'isogénies de courbes elliptiques supersingulières.
- Le graphe correspondant est un graphe de Ramanujan sur lequel agit un groupoïde non commutatif.

