

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

2015/12/08 – Séminaire Sécurité, Yaoundé

Damien Robert

Équipe LFANT, Inria Bordeaux Sud-Ouest



université
de BORDEAUX

Inria
informatics mathematics

Cryptologie à clé publique

Cryptologie :

- Chiffrement ;
- Authenticité ;
- Intégrité.

La cryptologie à clé publique est basée sur une fonction à sens unique (avec trappe)
⇒ chiffrement asymétrique, signatures, preuves sans connaissances...

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.



Contexte actuel

- [http://en.wikipedia.org/wiki/PRISM_\(surveillance_program\)](http://en.wikipedia.org/wiki/PRISM_(surveillance_program)) :
« The Prism program collects stored Internet communications based on demands made to Internet companies » (Microsoft, Yahoo!, Google, Facebook, Paltalk, YouTube, AOL, Skype, Apple...)
 - En sus de ce programme d'espionnage, utilisation de virus extrêmement sophistiqués, capables de s'infiltrer dans le firmware des disques durs : Stuxnet, Flame, Equation Group.
 - Officiellement pour lutter contre le terrorisme ;
 - Mais aussi utilisé pour l'espionnage économique :
<http://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance>
 - Les États-Unis ne sont pas les seuls à avoir des systèmes de surveillance...
<http://www.theguardian.com/world/2013/jul/04/france-electronic-spying-operation-nsa>
- ⇒ L'utilisation de la cryptographie est un enjeu de souveraineté nationale !



Quelle cryptographie ?

- Tout système ad-hoc, non basé sur des standards a été cassé (DECSS, GSM, WEP..)
- Utiliser un standard suffit-il ?

Standards :

- NIST workshop to standardize new elliptic curves ;
- IETF CFRG workgroup (Crypto Forum Research Group).



Quelle cryptographie ?

<http://blog.cryptographyengineering.com/2013/09/on-nsa.html>

Matthew Green – « The NSA has been :

- Tampering with national standards (NIST is specifically mentioned) to promote weak, or otherwise vulnerable cryptography.
- Influencing standards committees to weaken protocols.
- Working with hardware and software vendors to weaken encryption and random number generators.
- Attacking the encryption used by “the next generation of 4G phones”.
- Obtaining cleartext access to “a major internet peer-to-peer voice and text communications system”
- Identifying and cracking vulnerable keys.
- Establishing a Human Intelligence division to infiltrate the global telecommunications industry.
- decrypting SSL connections.

»



Quelle cryptographie ?

- Tout système ad-hoc, non basé sur des standards a été cassé (DECSS, GSM, WEP...)
- Utiliser un standard suffit-il ?

Standards :

- NIST workshop to standardize new elliptic curves ;
- IETF CFRG workgroup (Crypto Forum Research Group).

⇒ On a besoin d'une cryptographie issue de la recherche publique, conduite par des experts universitaires, en lien avec le milieu économique local.

But du cours : expliquer les concepts clés de la cryptologie, plutôt que de donner des « boîtes à outils » toutes prêtes.



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 : $E(m_i \oplus G(r) \parallel r \oplus H(m_i \oplus G(r)))$ 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 : $E(m_i \oplus G(r) \parallel r \oplus H(m_i \oplus G(r)))$ 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)
- 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
- Bug 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 (attack on Cipher Block Chaining)
- CRIME and BREACH (attack on compression)
- Downgrade attack : FREAK (export grade cryptography), Logjam (precomputation)
- Bugs : Heartbleed (buffer overflow), BERserk, goto fail
- Bogus certificates

Mitigating future loss of private keys : perfect forward secrecy via ephemeral Diffie-Hellman key exchange.



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.



- Générer une clé nécessite de l'aléa de qualité
- Aléa matériel ou pseudo-aléa logiciel
- Modèle d'entropie et extracteur d'aléa
- Un aléa de faible qualité peut complètement compromettre un système. (Par exemple si l'aléa dans le système de chiffrement El-Gamal a ses premiers bits prédictibles alors la système est cassé)
- Playstation (aléa constant), Clés ssh Debian (aléa = date)
- Instruction RDRAND : instruction Intel retournant un nombre aléatoire grâce à un générateur d'aléa matériel
- Theodore Ts'o : « I am so glad I resisted pressure from Intel engineers to let /dev/random rely only on the RDRAND instruction. To quote from the article below: 'By this year, the Sigint Enabling Project had found ways inside some of the encryption chips that scramble information for businesses and governments, either by working with chipmakers to insert back doors....' Relying solely on the hardware random number generator which is using an implementation sealed inside a chip which is impossible to audit is a BAD idea.»



- Générer une clé nécessite de l'aléa de qualité
- Aléa matériel ou pseudo-aléa logiciel
- Modèle d'entropie et extracteur d'aléa
- Un aléa de faible qualité peut complètement compromettre un système. (Par exemple si l'aléa dans le système de chiffrement El-Gamal a ses premiers bits prédictibles alors la système est cassé)
- Playstation (aléa constant), Clés ssh Debian (aléa = date)
- Instruction RDRAND : instruction Intel retournant un nombre aléatoire grâce à un générateur d'aléa matériel
- Theodore Ts'o : « I am so glad I resisted pressure from Intel engineers to let /dev/random rely only on the RDRAND instruction. To quote from the article below: 'By this year, the Sigint Enabling Project had found ways inside some of the encryption chips that scramble information for businesses and governments, either by working with chipmakers to insert back doors....' Relying solely on the hardware random number generator which is using an implementation sealed inside a chip which is impossible to audit is a BAD idea.»



- Générer une clé nécessite de l'aléa de qualité
- Aléa matériel ou pseudo-aléa logiciel
- Modèle d'entropie et extracteur d'aléa
- Un aléa de faible qualité peut complètement compromettre un système. (Par exemple si l'aléa dans le système de chiffrement El-Gamal a ses premiers bits prédictibles alors la système est cassé)
- Playstation (aléa constant), Clés ssh Debian (aléa = date)
- Instruction RDRAND : instruction Intel retournant un nombre aléatoire grâce à un générateur d'aléa matériel
- Theodore Ts'o : « I am so glad I resisted pressure from Intel engineers to let /dev/random rely only on the RDRAND instruction. To quote from the article below: 'By this year, the Sigint Enabling Project had found ways inside some of the encryption chips that scramble information for businesses and governments, either by working with chipmakers to insert back doors....' Relying solely on the hardware random number generator which is using an implementation sealed inside a chip which is impossible to audit is a BAD idea.»

Authentification

La cryptographie (à clé publique) permet de transformer un canal public en un canal authentifié et confidentiel, à partir d'un **premier échange authentifié**. Comment authentifier ce premier échange ?

- Certificats (TLS)
- Web of trust (GPG)
- Trust on first use (SSH)



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) fabrique 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



Dossier personnel de suivi médical embarqué

- Complétude et véracité du dossier médical
- Simplicité d'usage, Disponibilité, Accès Statistiques
- Confidentialité, Durabilité, Soutenabilité Économique



Résumé : importance de la cryptographie

- Chiffrement (pli confidentiel)
- Identification (badge cantine, carte à puce, empreinte digitale)
- Intégrité (clé RIB, scellés)
- Signature (ordre de virement) : identification, intégrité, non-répudiation
- Vie privée, anonymat, argent électronique, vote électronique, certificats, calcul distribué, stockage distribué

Applications : Développement des cartes à puces, commerce électronique, téléphonie mobile, armement, intérieur, sécurité des logiciels, sécurité des réseaux, objets interconnectés

Une opportunité pour les pays en développement

- Réseaux ;
- Objets interconnectés ;
- Puces RFID.



Résumé : importance de la formation en cryptographie

- La conception d'un cryptosystème et son implémentation doit être fait par un expert ;
- Utiliser les standards et les applications open-source mises à disposition par les experts universitaires ;
- Mais ne pas faire confiance aveugle à tous les standards ;
- Il faut des experts locaux pour faire un choix éclairé et le mieux adapté aux besoins spécifiques ;
- Seule la recherche Universitaire permet de rester à la pointe des dernières avancées scientifiques.



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
- Boîte mail, consultation de compte en banque, ...
- Avantages : simple, rapide, bien connu
- Fragilités : attaques statistiques, gestion de clés



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
- 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.

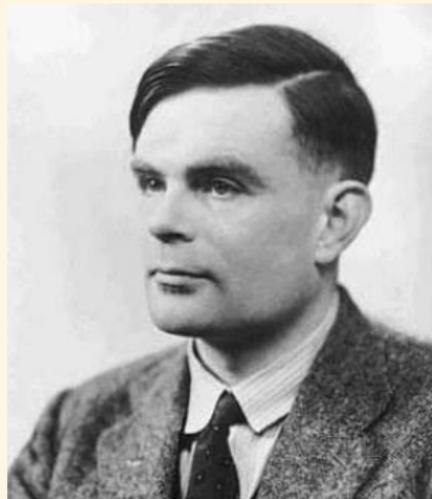


Comment faire ?

- 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

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

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 .



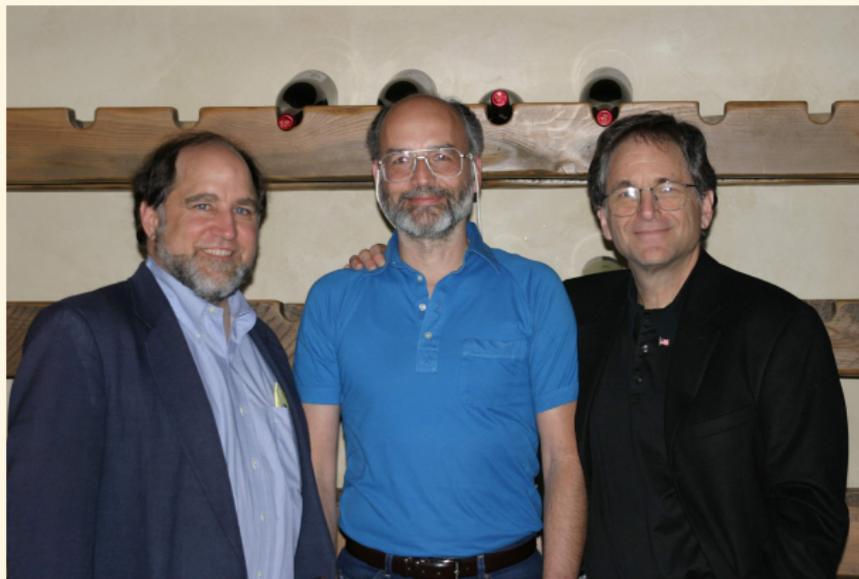
$$(p, q) \xrightarrow{\text{green}} N = pq$$

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



- 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.





Rivest, Shamir et Adleman

Protocole RSA

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

Théorème (Petit théorème de Fermat)

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



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}

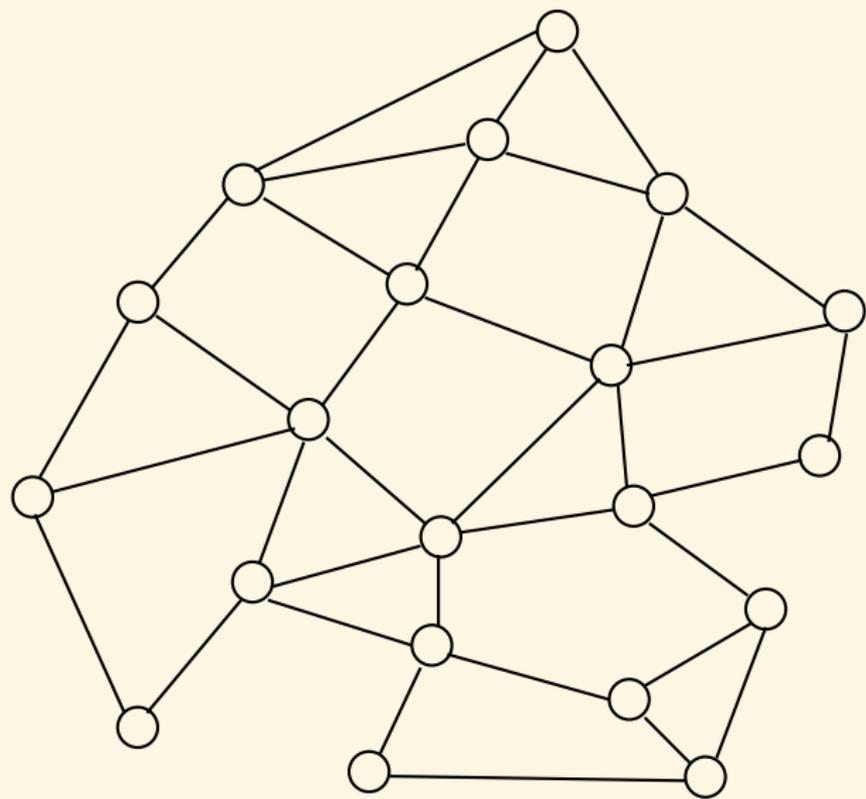


Bob

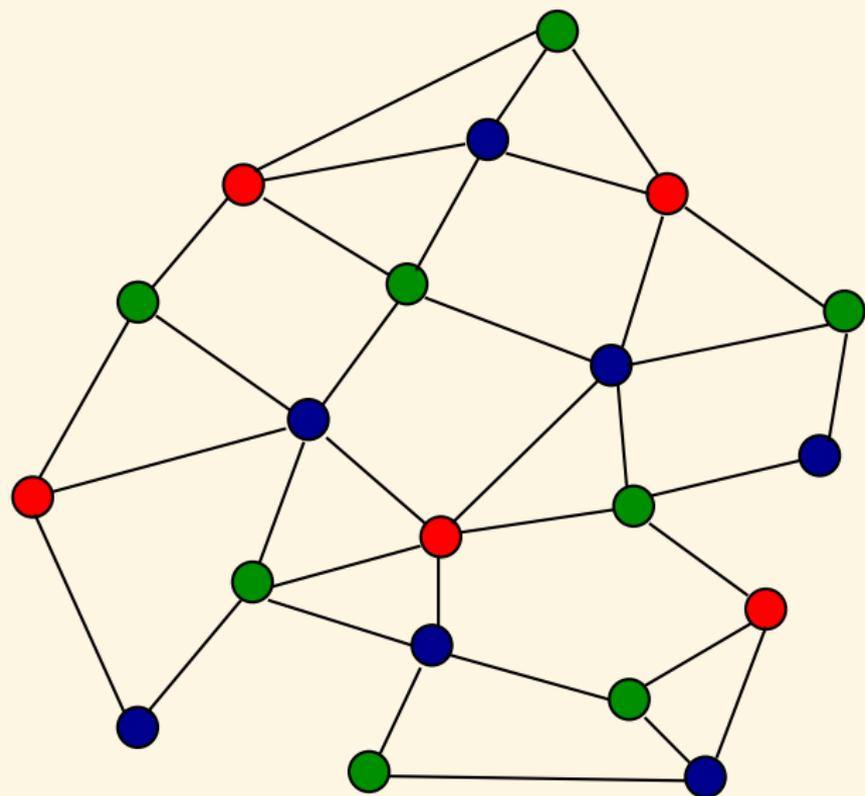
interroge Alice et se convainc qu'elle connaît bien le secret.

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

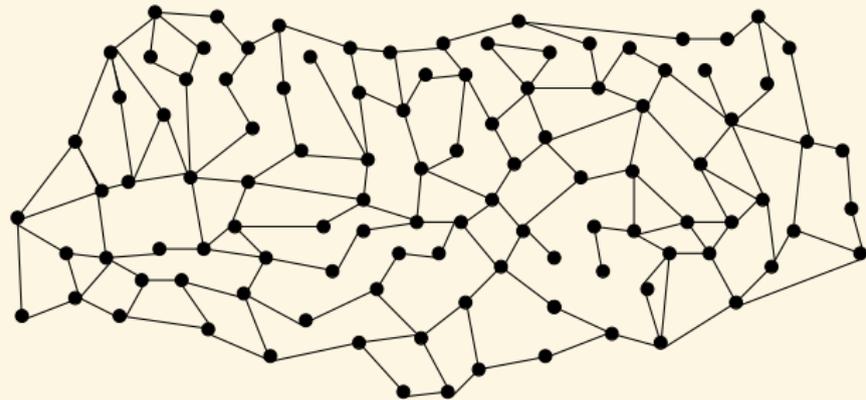
Coloriage de graphe



Coloriage de graphe



Coloriage de graphe



Zero Knowledge Proofs

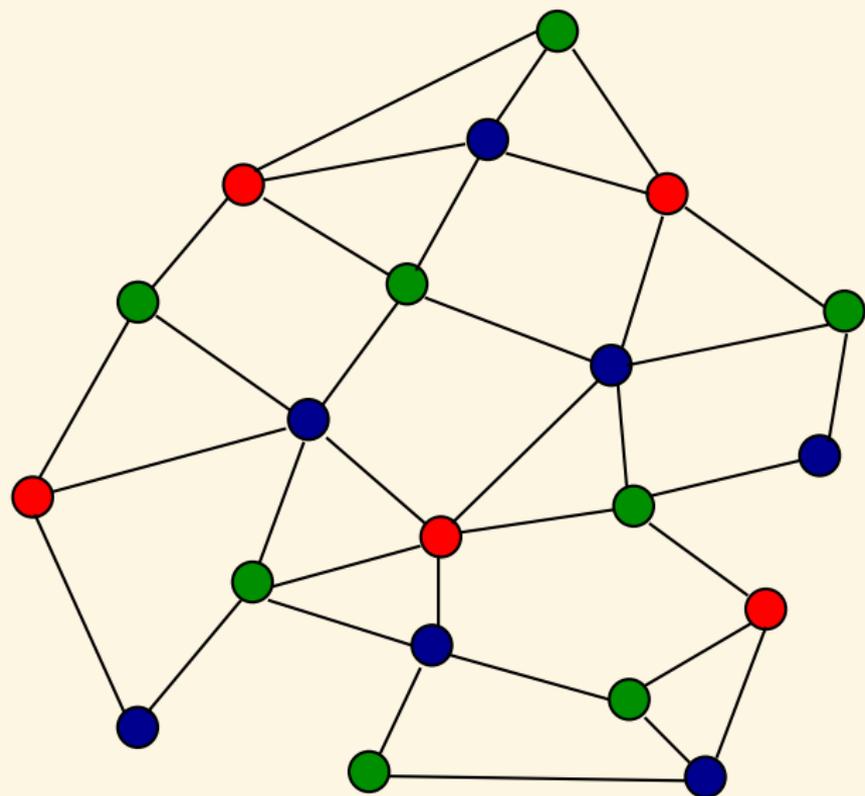


Shafi Goldwasser (1981)

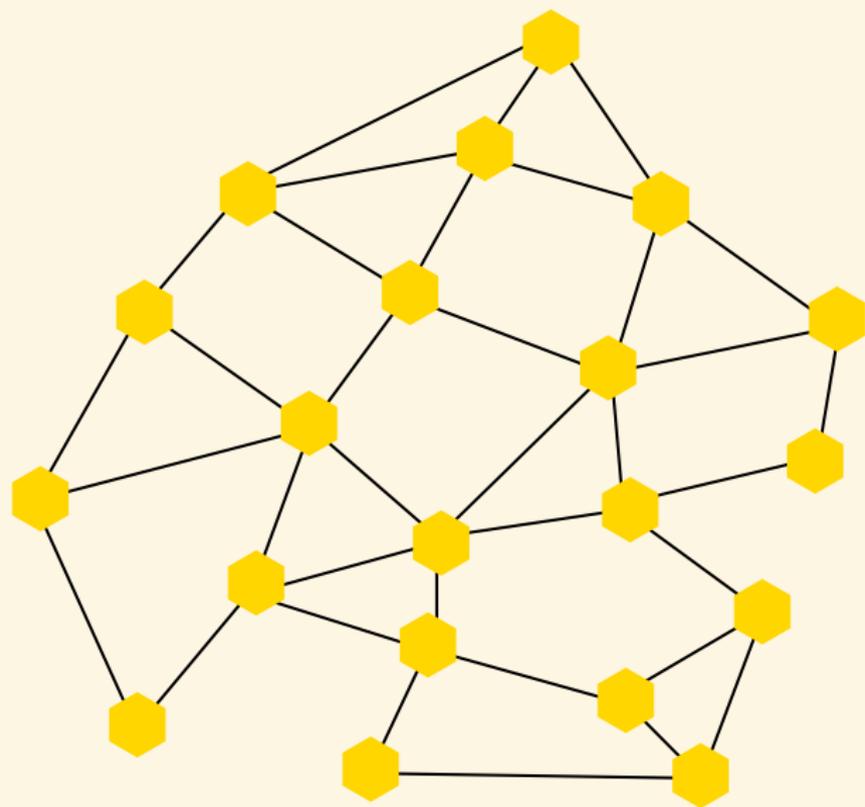


Oded Goldreich (1991)

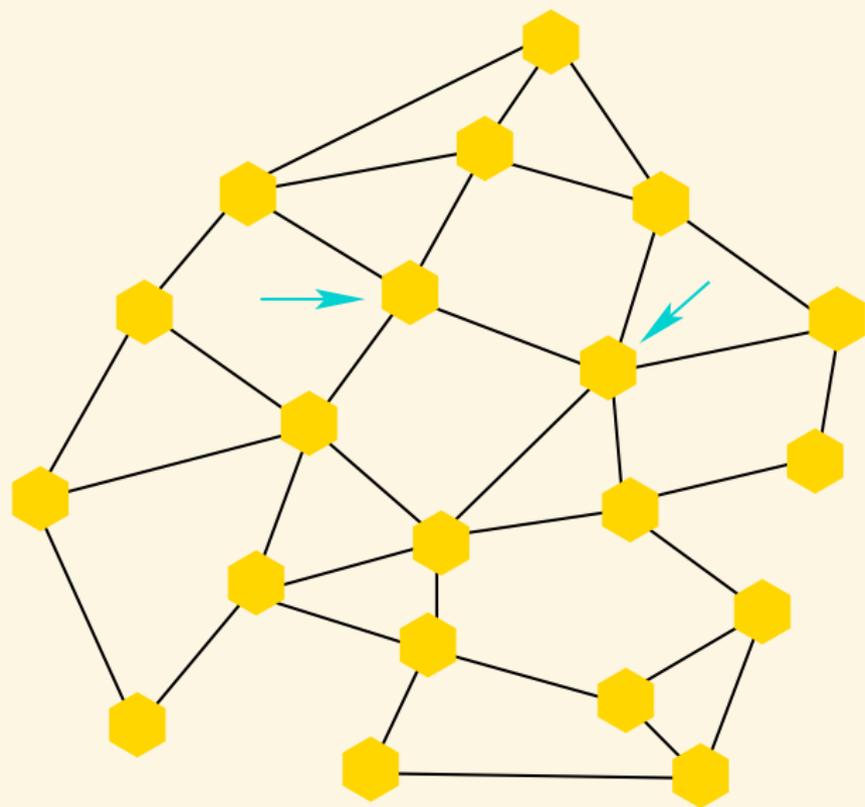
Le coloriage d'Alice (secret)



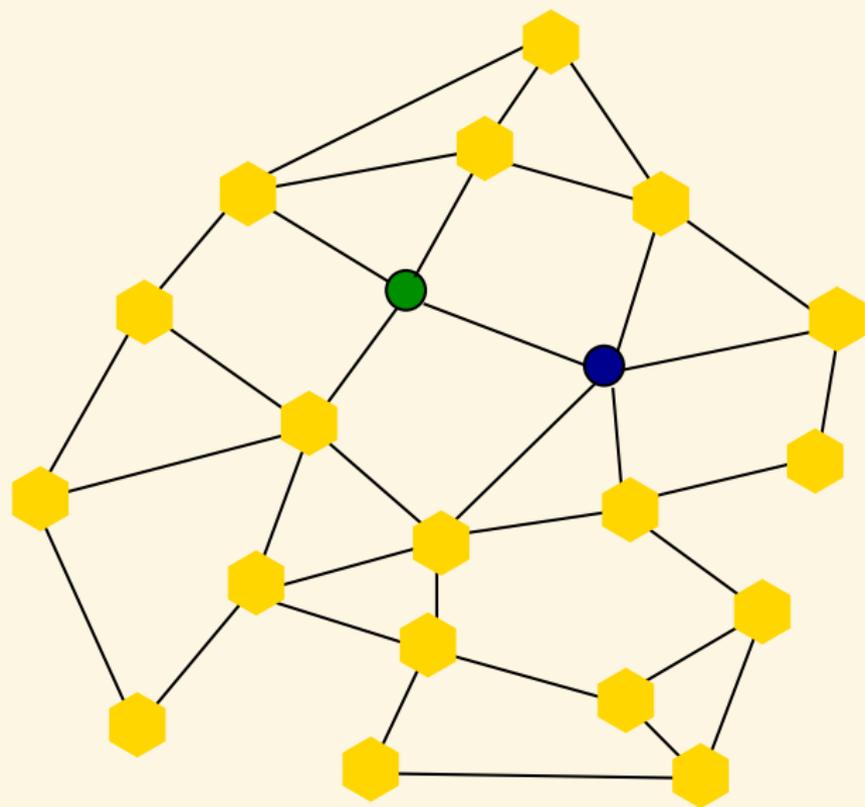
Le coloriage d'Alice caché



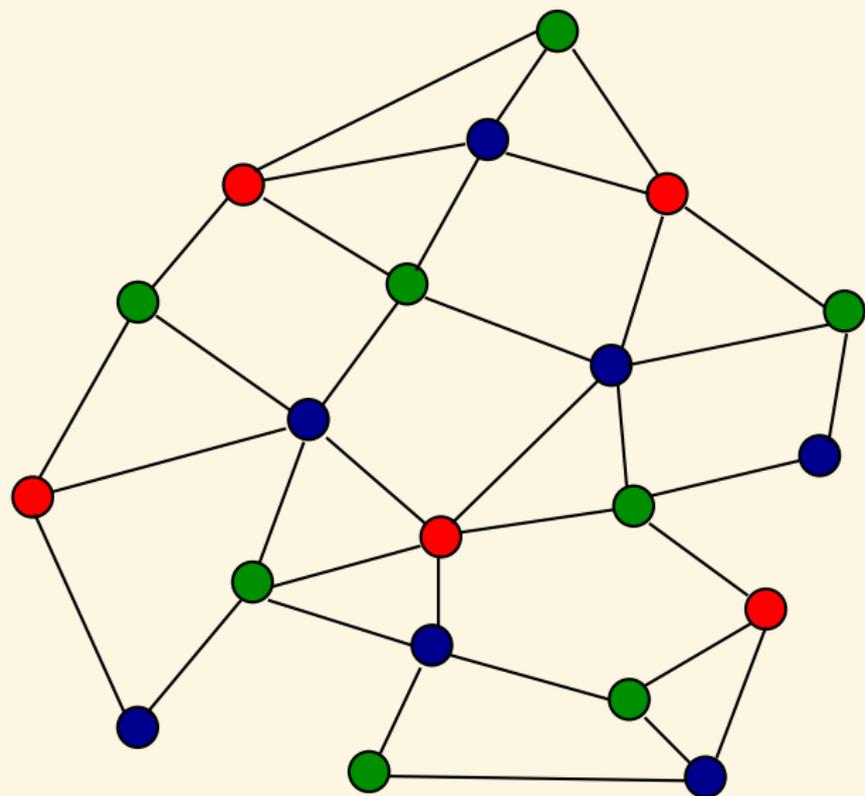
La question de Bob



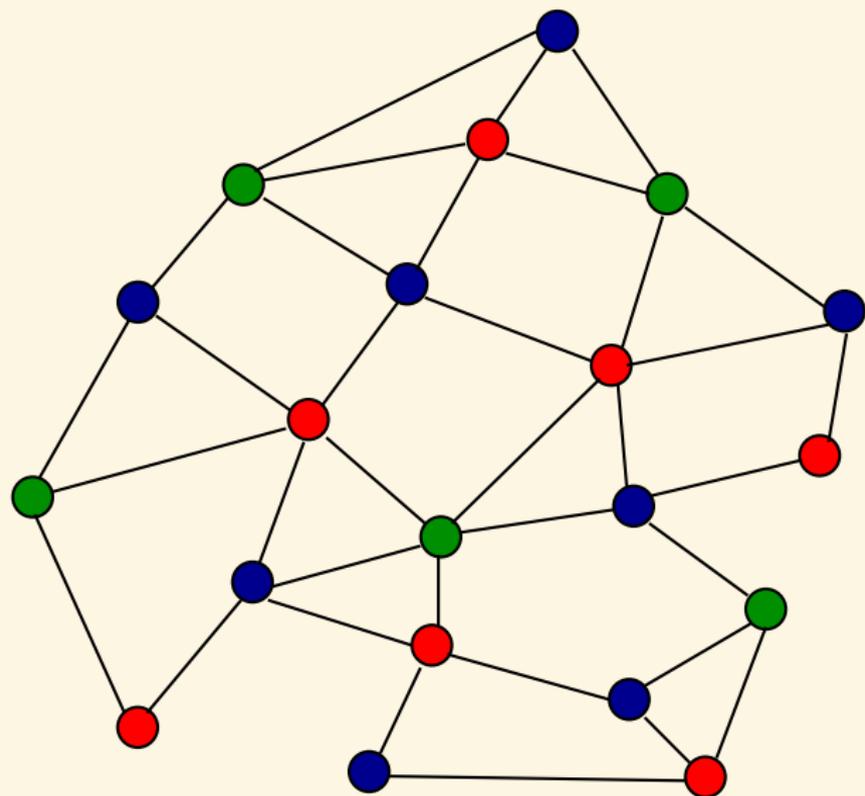
Dévoilement



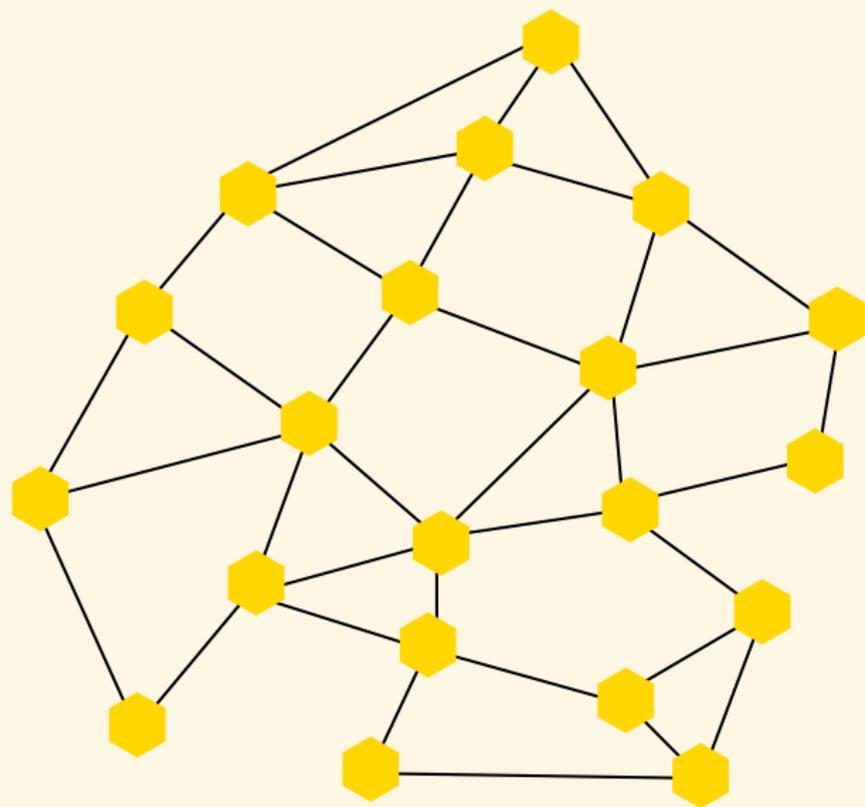
Le coloriage d'Alice (secret)



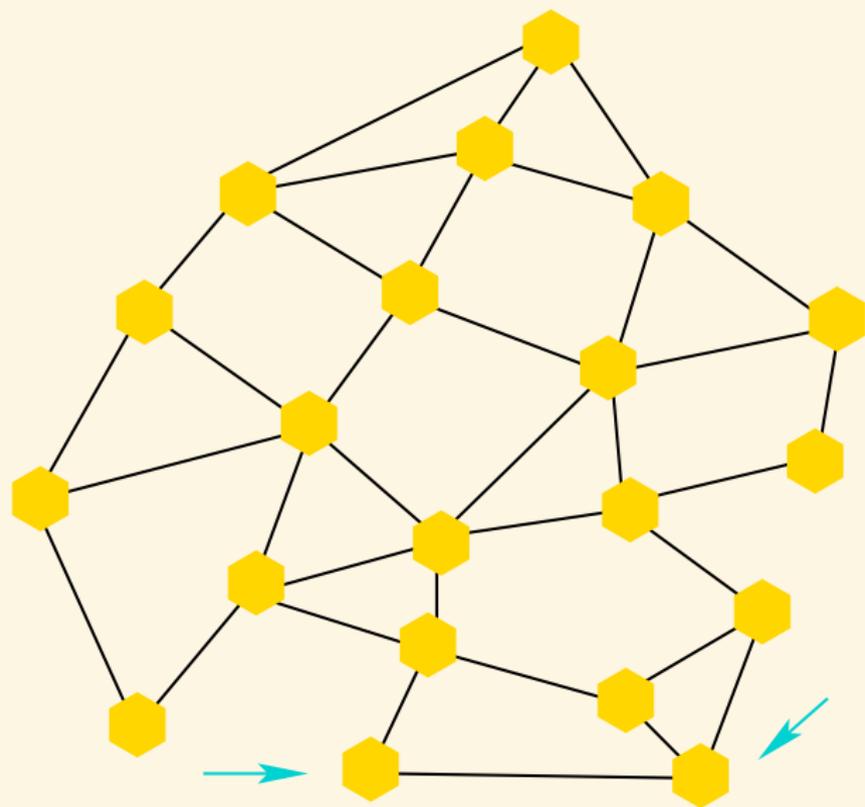
Le coloriage d'Alice permuté



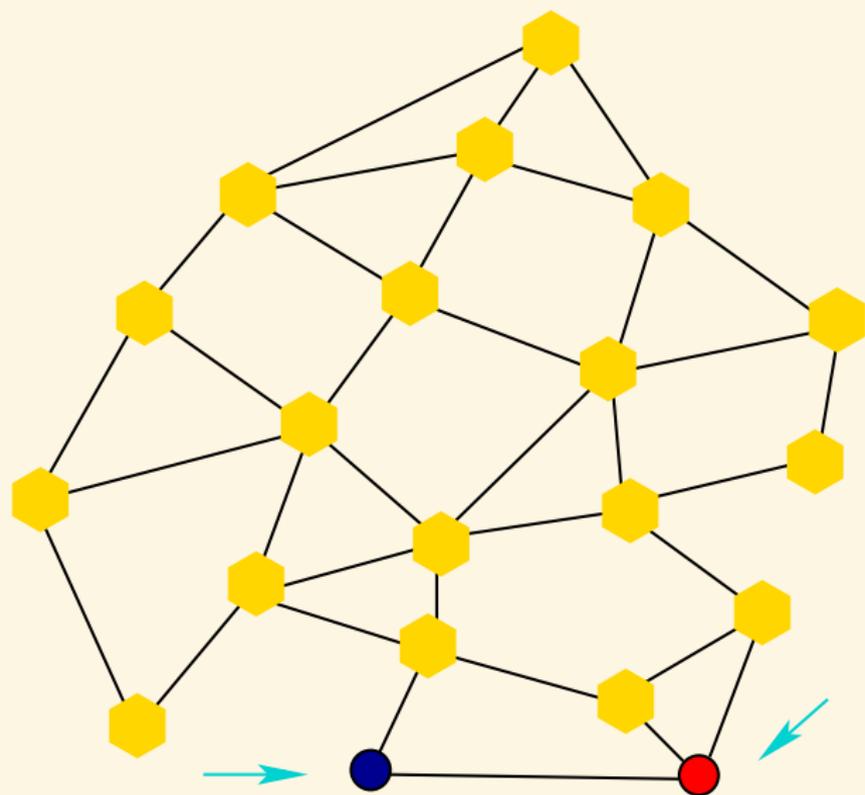
Le coloriage d'Alice caché



La deuxième question de Bob



Dévoilement



Fonctions de Hachage cryptographique

Définition

Comprime des données arbitraires en un “hash” de longueur fixée (typiquement 128 bits).

Une fonction de hachage est dite cryptographiquement sûre si

- Très difficile de trouver des collisions.
- Très difficile de retrouver m à partir de $H(m)$ (si l'on ne connaît pas m).



Mettre en gage des données

Jouer à pile ou face par téléphone.

- Alice lance une pièce
- Bob dit « tu as lancé pile »
- Alice répond « perdu c'était face ! »



Mettre en gage des données

Jouer à pile ou face par téléphone.

- Alice lance une pièce
- Alice envoie le haché de “Pile” ou “Face” à Bob.
- ...



Mettre en gage des données

Jouer à pile ou face par téléphone.

- Alice lance une pièce
- Alice envoie le haché $H(m||\text{Pile})$ ou $H(m||\text{Face})$ à Bob, où m est un message aléatoire.
- Bob dit « tu as lancé pile »
- Alice dévoile m à Bob pour qu'il puisse vérifier le résultat.

Remark

Attention : en cas de plusieurs tirages, Alice doit changer le message aléatoire m à chaque fois !



Échange de clé de Diffie Hellmann

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

- On part de $7 \bmod 61$;
- Alice choisit $17 \bmod 61$ et envoie $17 \times 7 = 58 \bmod 61$;
- Bob choisit $31 \bmod 61$ et envoie $31 \times 7 = 34 \bmod 61$;
- Le clé commune est $29 = 34 \times 17 = 58 \times 31$.

Remark

Pour retrouver la clé, Eve calcule $1/7 = 35 \bmod 61$ et retrouve $17 = 58 \times 35 \bmod 61$ 😞.



Exponentielle et logarithme discrets

$p = 7$ un nombre premier et $b = 5 \bmod 7$

x	b^x
1	$5 \bmod 7$
2	$4 \bmod 7$
3	$6 \bmod 7$
4	$2 \bmod 7$
5	$3 \bmod 7$
6	$1 \bmod 7$
7	$5 \bmod 7$

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

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

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

Exponentiation rapide : $b^x \bmod p$

$$p = 139, b = 112, x = 73$$

$$x = 2^6 + 2^3 + 1 = (1001001)_2$$

$$b^x = b b^{2^3} b^{2^6}$$

$$b_0 = b = 112 \bmod 139,$$

$$b_1 = b_0^2 = 34 \bmod 139,$$

$$b_2 = b_1^2 = b^{2^2} = 44 \bmod 139,$$

$$b_3 = (b_2)^2 = b^{2^3} = 129 \bmod 139,$$

$$b_4 = (b_3)^2 = b^{2^4} = 100 \bmod 139,$$

$$b_5 = (b_4)^2 = b^{2^5} = 131 \bmod 139,$$

$$b_6 = (b_5)^2 = b^{2^6} = 64 \bmod 139.$$

$$b^x = b_0 b_3 b_6 = 112 \times 129 \times 64 = 44 \bmod 139.$$

Pigala, *Chandah-sûtra* (avant -200).

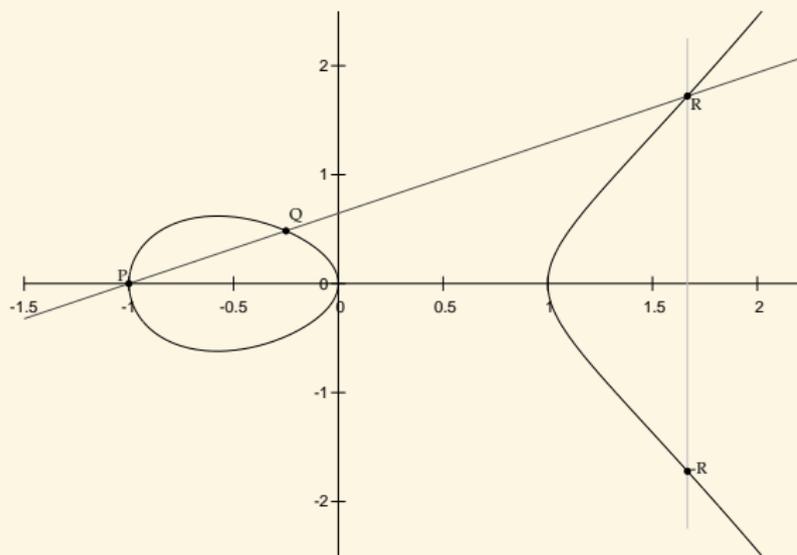


Les courbes elliptiques

Définition (char $k \neq 2, 3$)

Une courbe elliptique est une courbe plane d'équation

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



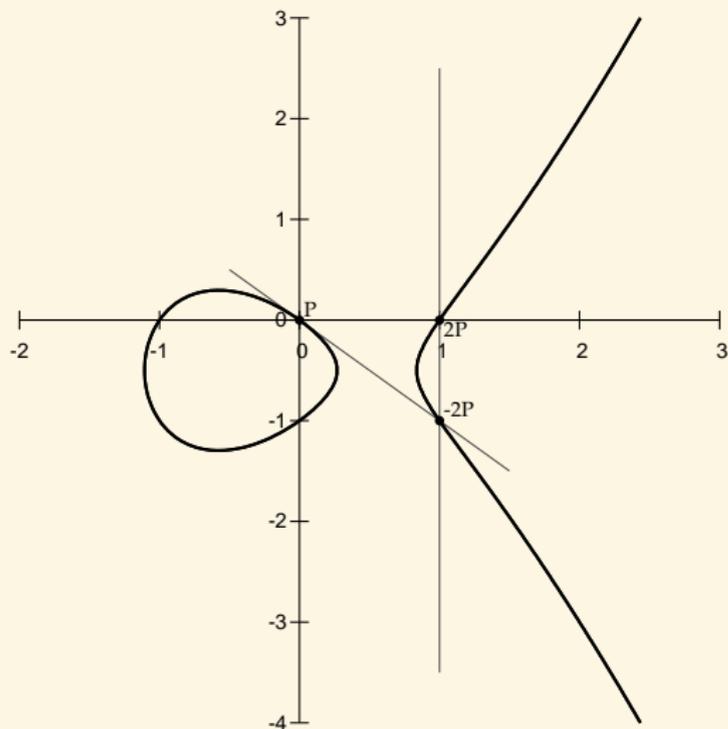
Exponentiation :

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

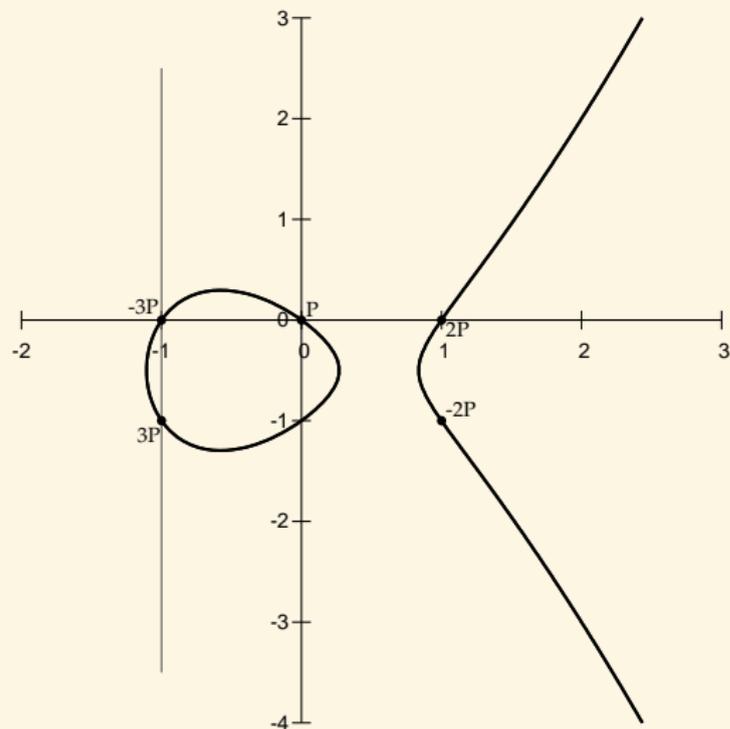
Logarithme discret :

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

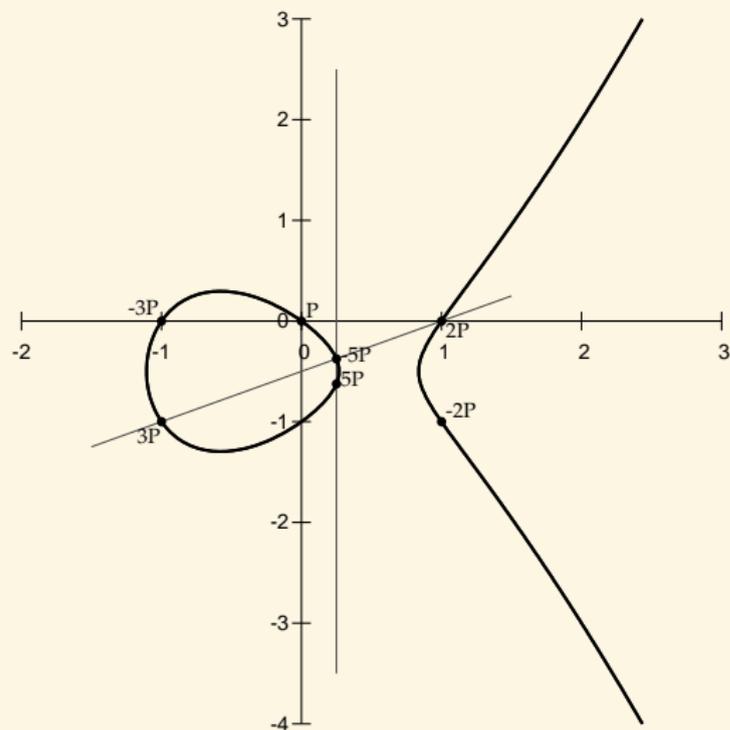
Exponentiation sur une courbe elliptique



Exponentiation sur une courbe elliptique



Exponentiation sur une courbe elliptique



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 : ℓ tel que $Q = \ell P$.
- Utilisées par la NSA ;
- Utilisées dans les passeports biométriques Européens.



Avantage des courbes elliptiques

À niveau de sécurité égale, les cryptosystèmes basés sur les courbes elliptiques, par rapport à RSA sont

- plus rapides ;
- plus compacts ;
- plus puissants.

Exemple (Couplages)

Sur une courbe elliptique, à partir d'une **clé publique** on peut générer d'autres **clé publiques**. De même pour la **clé secrète**.

⇒ Certificats anonymes.

Exemple (Fonctions de hachage)

Les graphes d'isogénies de courbes elliptiques supersingulières fournissent des fonctions de hachage cryptographiquement sûres.



Choisir une courbe elliptique

Modèle de Weierstrass ou d'Edwards, petite caractéristique, grande caractéristique, GLV, GLS, petits coefficients, coefficients aléatoires...

- Courbes du NIST ;
- Courbes Brainpool (BSI) ;
- Courbes ANSSI ;
- Courbes NUMS – Nothing Up My Sleeve (Microsoft) ;
- Curve25519 (DJB) ;
- Courbe Goldilock (Mike Hamburg).



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.

