Introduction to elliptic curve cryptography
Key exchange and signatures

Sorina Ionica

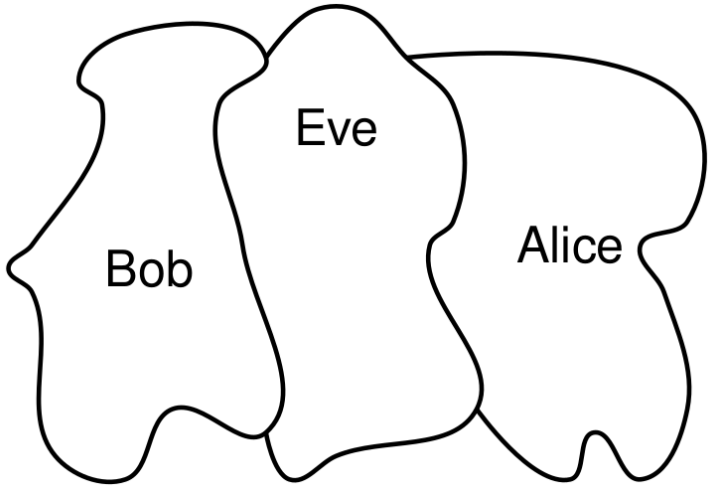
Université de Picardie Jules Verne, France

May 14, 2018

Cryptographie à clé publique

Il était une fois...

Dans l'Empire des Sept Couronnes Perdues, le vieux roi meurt.
Il laisse son royaume à ses enfants légitimes, Alice et Bob.

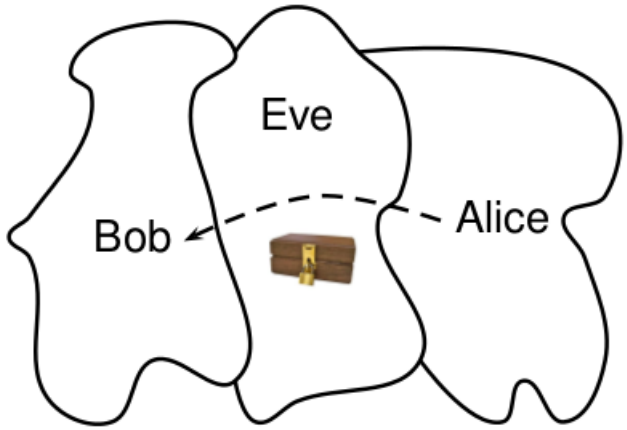


Il était une fois...

Eve se prépare d'attaquer sur la partie Ouest de Bob. Alice découvre son plan malign et doit informer Bob.

Elle envoie une lettre à Bob par son meilleur émissaire.

Alice met l'enveloppe dans un coffre-fort, elle ferme le cadenas et garde la clé.



Il était une fois

Bob reçoit le coffre-fort, met son cadenas et garde sa clé. Il renvoie le coffre à Alice.

Alice reçoit le coffre-fort, enlève son cadenas et renvoie le coffre-fort à Bob.

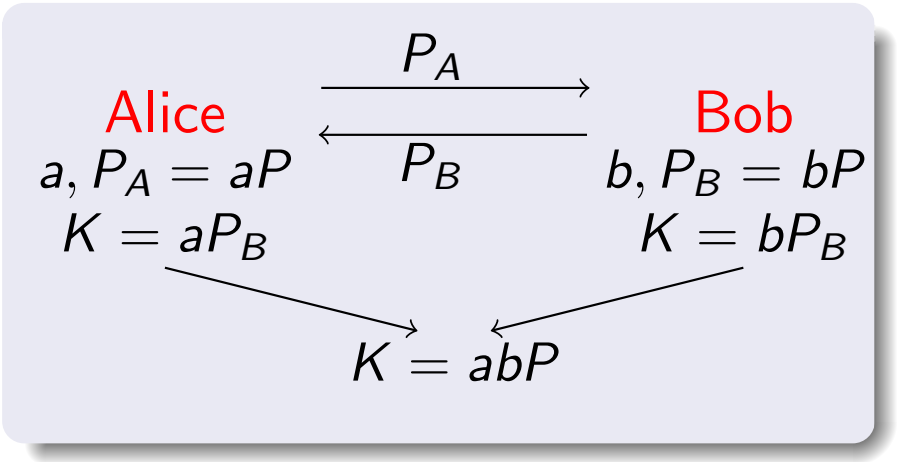
Bob reçoit le coffre-fort, ouvre son cadenas avec sa clé et récupére la lettre.

Paradigme de la clé publique

- Tout utilisateur dispose d'une pair clé publique/clé secrète.
- La clé publique est utilisée pour le chiffrement.
- La clé secrète est utilisée pour le déchiffrement.
- La sécurité repose sur un problème mathématique difficile: pour retrouver la clé secrete à partir de la clé publique on ne connaît pas d'algorithme autre qu'exponentiel.

Public key cryptography and groups

- Sharing a common secret over an insecure channel
- Diffie-Hellman Key Exchange : $(\mathbb{G}, +, P)$ public



Security: the Discrete Logarithm Problem (DLP) in $\mathbb{G}$

- Given $P, Q \in \mathbb{G}$ find (if it exists) λ such that

$$Q = \lambda P$$

Rational points and Hasse's theorem

The group of rational points

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 | F(x, y) = 0\}.$$

Hasse's theorem says that :

$$q + 1 - 2\sqrt{q} \leq E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}$$

Elliptic curves over $\mathbb{F}_q$ are groups with $\mathcal{O}(q)$ elements.

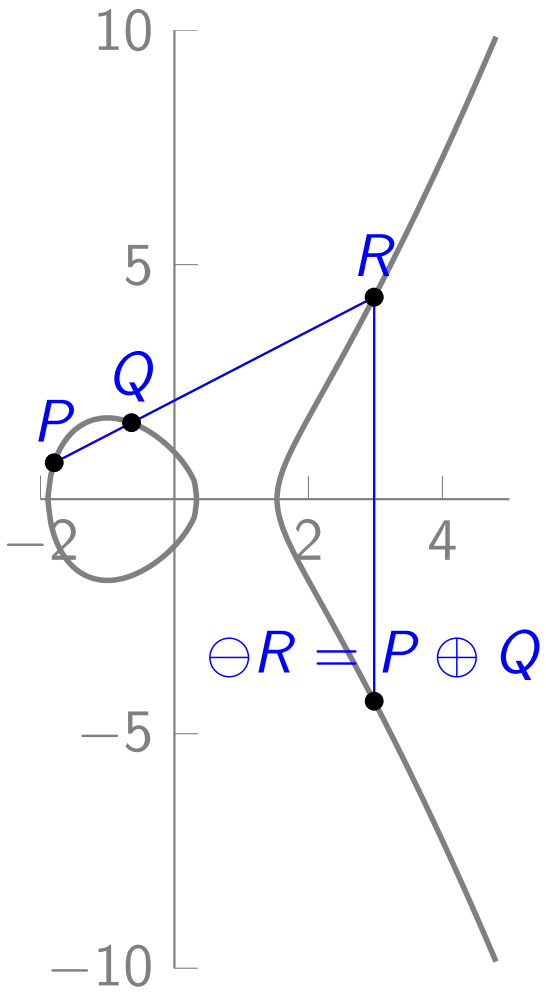
```
sage: E=EllipticCurve(GF(103),[2,5])
sage: print E.order()
```

Generating curves in cryptography

- Point counting via the SEA algorithm $O((\log q)^6)$
- Take random curves until a group with prime order is found.

```
sage: p=115292150460737127 7
sage: for a in range(2^10):
sage:     for b in range(2^10):
sage:       E=EllipticCurve(GF(p),[a+1,b])
sage:         if is_prime(E.order()):
sage:             print E
```

Efficient group law



$$E : y^2 = x^3 + ax + b,\ a, b \in \mathbb{F}_q$$

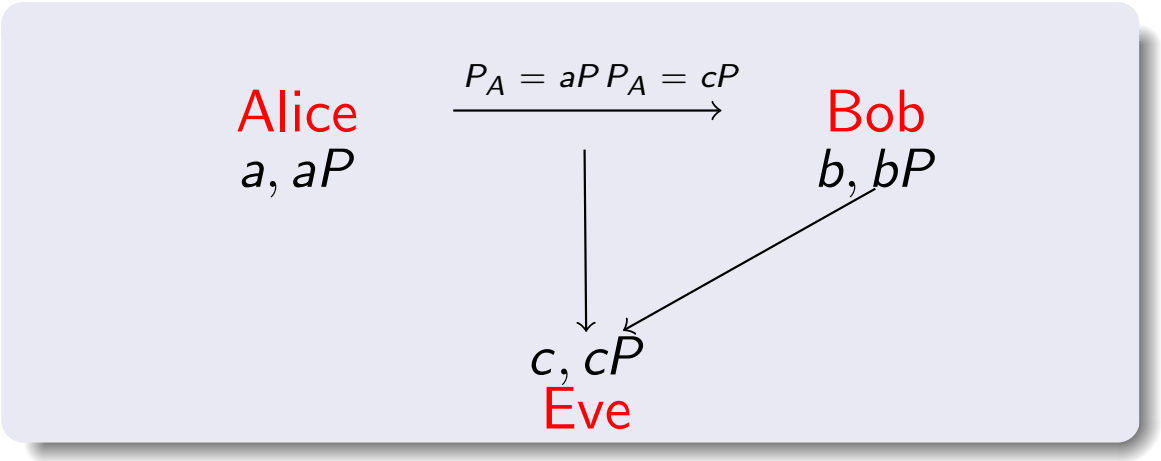
If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ with $Q \neq -P$, we denote by

$$\lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} & \text{if } P \neq Q, \\ \frac{3x_P^2 + a}{2y_P} & \text{if } P = Q. \end{cases}$$

The coordinates of $P + Q$ are

$$\begin{aligned} x_{P+Q} &= \lambda^2 - x_P - x_Q, \\ y_{P+Q} &= \lambda(x_{P+Q} - x_P) + y_P. \end{aligned}$$

Man-in-the-middle attack



- Eve intercepts Alice's message and remplaces it by another one, calculated from secret key.
- She also gets Bob's message for Alice et computes a common secret with him,letting him believe that she is Alice.
- In a similar way, she makes Alice believe that she is Bob.

Signatures

- *Authentic*: it can only be issued by the author of the message
- *Impossible to reproduce*: It depends on the message.
- *Vérifiable* by everyone $\Rightarrow$ the public key paradigm

Application : authentification certificates

Signatures

Key generation

Generate a pair public/secret key.

Sign algorithm

Use the secret key to sign messages.

Verification algorithm

Use the public key to verify the authenticity of the message.

Schnorr signatures

Key generation

Generate elliptic curve with $r|\#E(\mathbb{F}_q)$, r large.

Secret key $x \in \mathbb{Z}/r\mathbb{Z}$.

Public key : the curve E, a point P of order r and $Q = xP$.

Both parties agree on a hash function $H : \{0,1\}^* \to \mathbb{Z}/r\mathbb{Z}$.

Schnorr signatures

Sign algorithm

Input: A message $m \in \{0,1\}^*$ and a private key $x \in \mathbb{Z}/r\mathbb{Z}$.

Output: A Schnorr signature $(s, e) \in (\mathbb{Z}/r\mathbb{Z})^2$.

1. Set $k = random(\mathbb{Z}/r\mathbb{Z})$.
2. Set $R = kP$.
3. Set $e = H(m||R)$.
4. Let $s = k - xe \pmod{r}$
5. Return (s, e).

Schnorr signatures

Sign algorithm

Input: A message $m \in \{0,1\}^*$ and a private key $x \in \mathbb{Z}/r\mathbb{Z}$.

Output: A Schnorr signature $(s, e) \in (\mathbb{Z}/r\mathbb{Z})^2$.

1. Set $k = random(\mathbb{Z}/r\mathbb{Z})$.
2. Set $R = kP$.
3. Set $e = H(m||R)$.
4. Let $s = k - xe \pmod{r}$
5. Return (s, e).

Verify algorithm

Input: A signature $(s, e) \in (\mathbb{Z}/r\mathbb{Z})^2$, a message $m \in \{0,1\}^*$ and a public key $Q \in \mathbb{G}$.

Output: **True** or **False**

1. Let $R' = sP + eQ$
2. Let $e' = H(m||R')$
3. If $e' = e$ then return **True**
   else return **False**