Algoritmi di Crittografia

Corso di Laurea Magistrale in Informatica

A.A. 2018/2019

1. Elliptic curve cryptography (2)
   - EC cryptography over finite fields

Algoritmi di Crittografia

1. Elliptic curve cryptography (2)
   - EC cryptography over finite fields

The fields $\mathbf{Z}_p$

- If p is prime, the set $\{0, 1, \ldots, p-1\}$ with modular addition and multiplication is a *field*
- This means that it is a group under addition, with neutral element 0, and that it is also a group under multiplication, with neutral element 1
- The distributive property of multiplication also holds
- Polynomial factorization (i.e., the decomposition of a polynomial into a product of irreducible factors) exists and is unique over any field, hence also over $\mathbf{Z}_p$
- This is a well-known fact that has crucial importance for EC cryptography

Elliptic curves over $\mathbf{Z}_p$

- When moving from the field of real numbers to $\mathbf{Z}_p$, geometric intuitions do not help us any more
- Elliptic curves over $\mathbf{Z}_p$ appears as illustrated below

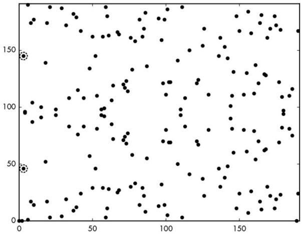

Figura: Elliptic curve $y^2 = x^3 - 4x$ over $\mathbf{Z}_{191}$: Source: J.P. Aumasson, Serious Cryptography, No Starch Press (2018)

Elliptic curves over $\mathbf{Z}_p$ (cont.d)

- However, what we have learned about the group structure and operations still holds
- In particular, let $E_p(x,y) = y^2 - x^3 - ax - b$ be a curve over $\mathbf{Z}_p$ and consider the "line" through $A = (x_A, y_A)$ and $B = (x_B, y_B)$, $x_A \neq x_B$

$$y = mx + q$$

with $m = \frac{y_B - y_A}{x_B - x_A} = (y_B - y_A)(x_B - x_A)^{-1}$ and $q = y_A - x_A m$

- The polynomial equation

$$p(x) = x^3 - m^2x^2 + (a - 2mq)x + b - q^2 = 0$$

has clearly x_A and x_B as solution

- But then, thanks to factorization, we can write $p(x) = (x - x_A)(x - x_B)r(x)$ and $r(x)$ must have degree 1
- Since $\mathbf{Z}_p$ is a field, $r(x)$ has a solution, which is the x-coordinate of the third "intersection point"

ECs suitable for cryptographic applications

- Let $E = E(a, b)$ be an elliptic curve. From all we have seen, we may conclude that the rational points $P \in E$ form a group G_E under point addition
- Whether or nor G_E is cyclic depends on the curve
- But we need a cyclic group
- Recall the definition of scalar multiplication over elliptic curves:

$$kP = \underbrace{P + P + \ldots + P}_{k-\text{times}}$$

- If G is cyclic, then there exists a point P that is a *generator*
- This means that, for any point Q on the curve, there is an integer k such that $kP = Q$
- Finding the number k (given the generator, or *base point* P) is called the *elliptic curve discrete logarithm problem* (*ECDLP*)

ECs suitable for cryptographic applications (cont.d)

- Often, the group of rational points is not a cyclic group
- However, the group of rational points always includes cyclic subgroups (trivially...)
- Clearly, not any cyclic subgroup is suitable for cryptographic purposes
- The subgroup must be large, for obvious reasons
- Also, the order of the subgroup must be divisible by a large prime (recall DH protocol)

A simple example

- Consider the curve
$$y^2 = x^3 - 4x$$
over the base field $\mathbf{Z}_{13}$
- Besides $\mathcal{O}$, the points on the curve are:
```
[(0, 0), (1, 6), (1, 7), (2, 0), (4, 3),
 (4, 10), (5, 1), (5, 12), (6, 6),
 (6, 7), (7, 4), (7, 9), (8, 5), (8, 8),
 (9, 2), (9, 11), (11, 0), (12, 4), (12, 9)]
```
- A generator for the maximal subgroup is the point $(4, 3)$ and the maximal subgroup itself is
```
[(4, 3), (1, 6), (9, 2), (12, 9), (0, 0),
  (12, 4), (9, 11), (1, 7), (4,10)]
```
besides $\mathcal{O}$

Choosing a suitable curve

- Let E be an alliptic curve over F_p and let N be the number of points of E; clearly E must be large
- A trivial estimate is $N \leq 2p + 1$ elements, i.e. $2p$ pairs (x, y) in addition to $\mathcal{O}$
- A better bound can be devised by Hasse's theorem

$$|N - (p + 1)| \leq 2\sqrt{p}$$

- Counting the exact number of points is possible but not trivial
- The fastest known algorithm to compute N is due to René Schoof
- Its complexity is $\tilde{O}(\log^5 p)$
- For the values of p needed in cryptography (e.g., $p \approx 2^{256}$) the cost of Schoof's is very high

The difficulty of the ECDLP

- Solving the ECDLP for a suitably chosen curve is considered very difficult
- An advantage of ECDLP over the classical DLP proposed by Diffie and Hellman is that same levels of security can be obtained using much smaller numbers
- Given points P and Q over E, such that $Q = kP$, a strategy to determine k consists of finding a "collision" among different linear combinations of P and Q
- In fact, suppose you find pairs a, b and c, d such that:

$$aP + bQ = cP + dQ$$

- Since we know that $Q = kP$, by substituting and rearranging we get

$$(a + bk)P = (c + dk)P$$

The difficulty of the ECDLP (cont.d)

- This means that (since P is a generator of the cyclic subgroup G_P) $a + bk \equiv c + dk$ modulus the size of the subgroup G_P
- Simple algebra then gives

$$k = (c - a)(b - d)^{-1} \mod |G_p|$$

- An adaptation of the birthday paradox shows that, if p is of the order of 2^n, then the expected number of attempts before finding a collision is $2^{n/2}$
- The downside is that, with smaller numbers, the size of the encrypted messages is consequently smaller

DH protocol on Elliptic Curves

- The DH protocol carries “easily” over EC, at least from a mathematical viewpoint
- Alice and Bob want to share a secret value (to be used as, or transformed into a symmetric key) over an insecure channel
- As before, some parameters must be publicly known: the particular curve $E(x, y) = y^2 - x^3 - ax - b$, the underlying field $\mathbf{Z}_p$, and the cyclic subgroup generator P (a point on the curve)
- Independently, Alice and Bob pick secret values k_A and k_B
- After that, they compute and release the public information $Q_A = k_A P$ and $Q_B = k_B P$, respectively
- Upon receipt, Bob computes $S = k_B Q_A = k_B k_A P$ while Alice computes $S = k_A Q_B = k_A k_B P$
- The secret is then the x-coordinate of S

Digital signature with EC

- Signing with ECC is little more involved
- Besides the curve parameters, Alice (here the signer) publishes her public key $P_A = kG$, where k is the corresponding secret key
- Let M be the message to be signed
- As a first step Alice computes $h = \texttt{SHA256}(M) \bmod |G_P|$
- Then she picks random integer r in the range $[1, |G_P| - 1]$, determines the point $rG = (x, y)$ on the curve, and computes $d = x \bmod |G_P|$
- As a last computation step, Alice computes the secret quantity $s = (h + dk)r^{-1} \bmod |G_P|$
- Alice sends the pair (d, s) to Bob (we assume that message confidentiality is not an issue)

Signature verification

- Upon receiving (d, s), Bob computes the value $w = s^{-1} \bmod |G_P|$ and then

$$\begin{aligned} u &= hw \\ &= hr(h + dk)^{-1} \bmod |G_P| \end{aligned}$$

as well as

$$\begin{aligned} v &= dw \\ &= dr(h + dk)^{-1} \bmod |G_P| \end{aligned}$$

- Using these quantities, Bob computes the linear combination $uG + vP$, which turns out to be rG

Signature verification (cont.d)

- In fact, since $P = kG$ is Alice's public key, we have (omitting the mod reductions for clarity)

$$\begin{aligned} uG + vP &= uG + vkG \\ &= (u + vk)G \\ &= (hr + drk)(h + dk)^{-1}G \\ &= r(h + dk)(h + dk)^{-1}G \\ &= rG \end{aligned}$$

- Bob now checks that the x-coordinate of $uG + vP$ coincides with d thus completing the verification