Elliptic Curves in Cryptography

Generalized Discrete Logarithm Problem

Recall the discrete logarithm problem: Let p be a large prime, and g a generator of Z_p^*. The problem is given $g^x \bmod p$ to find x.

We can generalize this problem to any group with a generator (actually, it is not necessary that this element is the generator).

Let us look at a group G and an element $g \in G$ of large unsmooth order. We can define the generalized discrete logarithm problem: given g^x to find x.

Note that if G has a smooth order, we can divide the discrete logarithm problem to several sub-problems which can be easily solved.

Properties of Fields

A field $\mathcal{K}$ can be either an infinite field or a finite field. The number of elements in a finite field $\mathcal{K}$ must of the form $q = p^r$, where p is a prime number and r is an integer.

We define the *characteristic* $char(\mathcal{K})$ of a field $\mathcal{K}$ to be 0 in the case of infinite fields or p when the field is finite. For example:

$$char(\mathbb{R}) = 0 \qquad char(F_{11}) = 11 \qquad char(F_{2^5}) = 2$$

Another (equivalent) definition of the characteristic of a field is the minimal positive number of times we need to add 1 to itself to get 0. We define the characteristic of a field to be 0 if there is no such positive number.

General Definition of An Elliptic Curve

An *Elliptic Curve* is an algebraic structure over a field $\mathcal{K}$. The elliptic curve E contains all the points $(x, y) \in \mathcal{K}^2$ satisfying the curve's equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

The curve also contains a special point O called the "point at infinity".

The above equation can be transformed using algebraic assignments to a much simpler equation. For example, assigning $z = x + a_2/3$ transforms Equation 1 into:

$$y^2 + a_1zy + a'_3y = z^3 + a'_4z + a'_6$$

Note that this transformation assumes that 3^{-1} is defined in the field $\mathcal{K}$.

General Definition of An Elliptic Curve (cont.)

Assuming we can use the above transformations (which is always true for any infinite field, like $\mathbb{R}$) we can transform Equation 1 into:

$$y^2 = x^3 + ax + b \tag{2}$$

This equation can be achieved for any field whose characteristic is different from 2 or 3.

Note: The equation must not have multiple roots.

Over fields with characteristic 3, Equation 1 can be transformed into:

$$y^2 = x^3 + ax^2 + bx + c \tag{3}$$

Over fields with characteristic 2, Equation 1 can be written either as:

$$y^2 + cy = x^3 + ax + b \qquad \text{or} \qquad y^2 + xy = x^3 + ax^2 + b \tag{4}$$

Over finite fields, the requirement that there are no multiple roots is equivalent to the requirement that $4a^3 + 27b^2 \neq 0$ (the computation is done in the field!).

General Definition of An Elliptic Curve (cont.)

For example, we present the curve E over Z_5 as all the solutions for the equation $y^2 = x^3+x$. The points on this curve are: $(0,0)$, $(2,0)$, $(3,0)$, $(4,0)$ and of course O.

For the same equation over Z_{11} we get the following points: (0,0), (5,3), (5,8), (7,3), (7,8), (8,5), (8,6), (9,1), (9,10), (10,3), (10,8), O.

For curves of forms (2)–(4), it is easy to see that if (x, y) is on the curve, so does the point $(x, -y)$. This is obvious for any field with characteristic different from 2, as y appears in the equations only as y^2. For fields with characteristics 2, the claim is still true, as $-1 = 1$.

Addition and Negative Elements over the Reals

We first present the geometrical ideas behind points addition over a curve which is defined over $\mathbb{R}$.

Let $\mathcal{K} = \mathbb{R}$, and let P, Q be two points on E. We define the points $P + Q$ and $-P$ according to the following rules:

- If $P = O$ then $-P = O$ and $P + Q = Q$.
- If $P = (x, y) \neq O$ then $-P = (x, -y)$.
- If P and Q differ in the x-coordinate then we define the straight line $l = \overline{PQ}$. It is guaranteed that l intersects with the curve in a third point R. We define $P + Q = -R$.
- If $P = -Q$ then $P + Q = O$.
- If $P = Q$ and let l be the tangent line in P. If l intersects with the curve in another point, we define this point as R, and $P+Q = -R$. Otherwise, we define $R = O$, and then $P + Q = 2P = O$.

Addition and Negative Elements over the Reals (cont.)

We can explicitly calculate R. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ then:
if $P \neq \pm Q$

$$x_{P+Q} = \left(\frac{y_P - y_Q}{x_P - x_Q}\right)^2 - x_P - x_Q$$

and

$$y_{P+Q} = -y_P + \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_{P+Q})$$

For $P = Q$ (we do not need to deal with $P = -Q$) we get

$$x_{2P} = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \qquad \text{and} \qquad y_{2P} = -y_P + \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_{2P})$$

Elliptic Curves over Finite Fields

Let $\mathcal{K}$ be a finite field F_q of $q = p^r$ elements. We can define the curve using equations (2)–(4) according to p.

For addition of points we can use the above formulae as long as $char(\mathcal{K}) \neq 2, 3$.

For $char(\mathcal{K}) = 3$ we get the following equations:

$$x_{P+Q} = \left(\frac{y_P - y_Q}{x_P - x_Q}\right)^2 - a - x_P - x_Q, \qquad y_{P+Q} = -y_P + \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_{P+Q})$$

When P and Q are distinct, and a is the coefficient of the x^2 in the curve's equation.

When $P = Q$:

$$x_{2P} = \left(\frac{2ax_P + b}{2y_P}\right)^2 - a - 2x_P, \qquad y_{2P} = -y_P + \left(\frac{2ax_P + b}{2y_P}\right)(x_P - x_{2P})$$

where b is the coefficient of x in the curve's equation.

Elliptic Curves over Finite Fields (cont.)

For a field with characteristic 2 we have two cases. The first is when the curve is $y^2 + xy = x^3 + ax^2 + b$ (the nonsupersingular case):

For $P \neq Q$:

$$x_{P+Q} = \left(\frac{y_P + y_Q}{x_P + x_Q}\right)^2 + \frac{y_P + y_Q}{x_P + x_Q} + a + x_P + x_Q,$$

$$y_{P+Q} = \left(\frac{y_Q + y_P}{x_Q + x_P}\right)(x_P + x_{P+Q}) + x_{P+Q} + y_P$$

when P and Q are distinct.

For $P = Q$:

$$x_{2P} = x_P^2 + \frac{b}{x_P^2}, \qquad y_{2P} = x_P^2 + (x_P + \frac{y_P}{x_P})x_{2P} + x_{2P}$$

Elliptic Curves over Finite Fields (cont.)

If the curve is $y^2 + cy = x^3 + ax + b$ (the supersingular case) then:

$$x_{P+Q} = \left(\frac{y_P + y_Q}{x_P + x_Q}\right)^2 + x_P + x_Q, \qquad y_{P+Q} = \left(\frac{y_Q + y_P}{x_Q + x_P}\right)(x_P + x_{P+Q}) + y_P + c$$

when P and Q are distinct. When $P = Q$:

$$x_{2P} = \frac{x_P^4 + a^2}{c^2}, \qquad y_{2P} = \left(\frac{x_P^2 + a}{c}\right)(x_P + x_{2P}) + y_P + c$$

Elliptic Curves over Finite Fields (cont.)

The number of points on the curve can be bounded by $2q+1$.

However, as in F_q^* only half of the elements have square roots, then we expect to find only half of that number.

Let χ be the quadratic character of F_q ($\chi(x) = 1$ iff x has a square root, $\chi(0) = 0$, and $\chi(x) = -1$ otherwise). Over $q = p$ the quadratic character $\chi(x) = (\frac{x}{p})$ is the Legendre symbol.

The number of solutions for the equation $y^2 = u$ is $1 + \chi(u)$, and thus, the number of solution to the curve's equation (including the point at infinity) is

$$1 + \sum_{x \in F_q} (1 + \chi(x^3 + ax + b)) = q + 1 + \sum_{x \in F_q} \chi(x^3 + ax + b)$$

Hasse's Theorem: *Let N be the number of the points on the curve E defined over F_q. Then*

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

Elliptic Curves over Finite Fields (cont.)

The abelian group defined by the curve is not necessarily a cyclic one (there might not be a generator), but is a product of cyclic groups of the form $Z_{p^\alpha} \times Z_{p^\beta}$ where $p|N$ $(\alpha \geq 1, \beta \geq 0)$.

Type of the abelian group of F_q-points on E is a listing $(\ldots, p^\alpha, p^\beta, \ldots)_{p|N}$ of the orders of the cyclic factors. Where the cases when $\beta = 0$ are omitted. Note that finding the type of a curve is not always an easy task.

Multiplying all the values in the listing gives the number of points on the curve (as each point can be represented as a vector over the various cyclic sub-groups that compose the curve).

Elliptic Curve Diffie-Hellman

Let E be a curve, and let $P \in E/F_q$ be a point on the curve with a very large order where F_q is a field with q elements.

Each user chooses at random $k \in F_q$, and publishes $Q = kP$ as his public value.

When A and B wish to communicate, A computes $k_A Q_B$ while B computes $k_B Q_A$. It is easy to see that both results are equal to $k_A k_B P$.

Elliptic Curve Diffie-Hellman (cont.)

Note that it is not wise to use both the x value and the y value of the joint point $k_A k_B P$. As given an x value of a point, the y value has only two options (the solutions of $x^3 + ax + b$).

As we would like the shared secret to contain as few redundancy as possible, it is advised to use only the x value of the shared point.

The ElGamal Cryptosystem

The ElGamal cryptosystem is based on the discrete logarithm problem (even though breaking ElGamal might not imply solving the discrete logarithm problem).

In Z_p the scheme is as follows: given p and g the generator of Z_p^*, each user chooses at random x and publishes $y = g^x \bmod p$ as the public key.

When sending a message $m \in Z_p$ to a user with a public key y:

- Choose at random $k \in Z_{p-1}$.
- Compute $c = (c_1, c_2)$ where $c_1 = g^k$ and $c_2 = m \cdot y^k$.

To decrypt the ciphertext $c = (c_1, c_2)$ we compute $m = c_2 \cdot c_1^{-x}$.

Elliptic-Curve ElGamal

As mentioned before, we can generalize the ElGamal system to other groups besides Z_p. Over the elliptic Curve E/F_q we get the following variant:
Let E and $P \in E/F_q$ be the system parameters (just like in Elliptic Curve Diffie-Hellman).
Alice published $Q_A = k_A P$ as her public key.
Let $M \in E$ be a message m embedded into the curve. To send M to Alice, Bob chooses $l \in F_q$ at random, and sends her the two points: $(lP, M + l(Q_A))$.

To decrypt a message (R_1, R_2), Alice computes $R_3 = k_A R_1$ and $M = R_2 - R_3$.

Elliptic Curve DSA

System parameters: Let E be a curve over Z_p, $P \in E/Z_p$ a point of order q. Note that unlike DSA, q is not necessarily much smaller than p (thus we can use small p for efficient implementations).

Keys: Each user U chooses the secret key x_U at random, and publishes $Q_U = x_U P$ as his public key.

Signature:

- Alice chooses $k \in \{1, \ldots, q-1\}$ at random.
- Computes $kP = (x_1, y_1)$ and $r \equiv x_1 \bmod q$. If $r = 0$ she returns to the first step.
- Computes $s \equiv k^{-1}(H(m) + x_{Alice} r) \bmod q$. If $s = 0$ she returns to the first step.
- The signature is the pair (r, s).

Elliptic Curve DSA (cont.)

Verification:

- Bob check that $r, s \neq 0$.
- Computes $w \equiv s^{-1} \bmod q$.
- Computes $u_1 = H(m)w \bmod q$, $u_2 = rw \bmod q$.
- Computes $(x_0, y_0) = u_1 P + u_2 Q$, and $v \equiv x_0 \bmod q$.
- Test whether $v \stackrel{?}{=} r$.

The size of the signature is 320 bits for q of 160 bits.
Note that each user can choose his own curve, while all having the same underlying p (or more precisely Z_p)!

Choosing P on E

The curve's equation $y^2 = x^3 + ax + b$ must satisfy $P = (x, y)$. However, solving this problem and finding P on E is very difficult.
Therefore, we first choose P and then choose a, and finally we compute:

$$b = y^2 - x^3 - ax$$

(recall to check that $4a^3 + 27b^2 \neq 0$).
Usually we don't need to know $N = \#E$, but it is important to know it is not smooth. There are various efficient techniques to compute N and find such an E for which $\#E$ is not smooth.

Elliptic Curves in Cryptography

Elliptic curves can be used for more than just a transformation of cryptographic primitives.

In “Factoring Integers using Elliptic Curves over Finite Fields”, Lenstra demonstrates how to factor $n = p_1 \cdot p_2 \cdots p_m$ with time of $O(e^{\sqrt{\log p_1 \log\log p_1}})$ where p_1 is the smallest prime dividing n.

Another useful application of elliptic curves is finding “certified primes”. It is known that $Primes \in P$. However, since 1986, there is an elliptic curve primality testing algorithm. This algorithm outputs primes with proofs of being primes, that can be easily verified.

The algorithm can be applied after Miller-Rabin or Solovay-Strassen probabilistic algorithms have outputed “probably a prime” on a candidate p. Given this p, we can try the elliptic curve primality test, and usually get a certificate to it's primality.

It is evident that the algorithm cannot always succeed, or otherwise it was a proof that $Primes \in P$.

Elliptic Curve Primality Test

In 1986 Goldwasser and Kilian have presented an elliptic curve primality test. The test can verify that a given number p is prime, by outputting a certificate to its primality of length $\log^2 p$. The certificate can be verified using a deterministic algorithm which requires time of $\log^4 p$.

Recall Pratt's algorithm which proves that $Primes \in NP$. It is known that if n is an odd integer, than Z_n^* has a generator iff $n = p^e$ for some prime p and an integer e. Moreover, if $n = p$ then $O(g) = p - 1$, where g is the generator of Z_p^*.

Pratt's algorithm is as follows:

1. Guess a factorization of $p - 1 = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ and g a generator of Z_p^*.
2. For $1 \leq i \leq k$ check that $g^{p-1/p_i} \neq 1 \bmod p$.
3. If this succeeds, continue recursively with proving that the p_i's are primes.

Elliptic Curve Primality Test (cont.)

The elliptic curve primality test is based on a similar principles. Given a candidate p we generate random curve over F_p whose order has a sufficiently large prime factor q. The existence of such a group can be used to prove that p is prime, and now we continue to prove the primality of q recursively.

It is also based on a deterministic primality test algorithm by Adleman, Pomerance and Rumley which runs in time $O(k^{c \log \log k})$ for numbers of length k. This algorithm is used after the size of the prime we would like to check is small enough, so that it's non-polynomial time complexity will not affect the total time complexity of the algorithm.

Elliptic Curve Primality Test (cont.)

1. Set $p_0 = p; i = 0; lowerbound = 2^{(\log_2 p)^{c/\log_2 \log_2 \log_2 p}}; certificate = \phi$.
2. While $p_i \geq lowerbound$
   (a) Randomly choose $a, b \in Z_{p_i}$, and compute $\#E$ the number of points on the curve $y^2 = x^3 + ax + b$ over F_{p_i}, until $gcd(4a^3 + 27b^2, p) = 1$ and PP($\#E/2$)="probably prime".
   (b) Set $q = \#E/2$.
   (c) Randomly choose $M \in E$ until $M \neq O$ and $qM = O$.
   (d) Set $i = i + 1$ and $p_i = q$.
   (e) Append M, q, a, b to $certificate$.
3. If $APR(p_i)$="composite" repeat the entire algorithm.
4. Return($certificate$).

Elliptic Curve Primality Test (cont.)

There are three sub routines which we do not describe: $PP(x)$ a polynomial probabilistic algorithm which checks primality, $ARP(x)$ the ARP deterministic primality check algorithm, and Schoof's algorithm for counting the number of points on a curve.

We note that Schoof's algorithm is deterministic and has a running time of $\log^8 p$ operations, but has many improvements.