Cryptography

Elliptic Curve Cryptography

Prof. Dr. Heiko Knospe

TH Köln – University of Applied Sciences

August 7, 2019

Weierstrass Equation

The multiplicative group $\mathbb{Z}_p^*$ of integers modulo a prime number p is used in several cryptographic algorithms, for example in the Diffie-Hellman key exchange. A well-proven alternative is the group of points on an *elliptic curve* over a finite field. Elliptic curves are defined by *Weierstrass equations*.

Definition

Let K be a field and $a, b \in K$, then the equation

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

is called a *short Weierstrass equation*. The equation defines an affine curve in K^2. The corresponding *projective curve* has one extra point O *at infinity*.

Elliptic Curves

Let $y^2 = x^3 + ax + b$ be a Weierstrass curve over a field K and $f(x,y) = -y^2 + x^3 + ax + b$. Then $\Delta = -16(4a^3 + 27b^2)$ is called the *discriminant* of the curve. If Δ is nonzero in K, then the curve is nonsingular, i.e., the partial derivatives $D_x f = 3x^2 + a$ and $D_y f = -2y$ do not simultaneously vanish on the curve.

Definition

An *elliptic curve* E over a field K is a nonsingular projective curve defined by a Weierstrass equation. The set of points on E with coordinates in K is denoted by $E(K)$:

$$E(K) = \{(x,y) \in K \times K \mid y^2 = x^3 + ax + b\} \cup \{O\}.$$

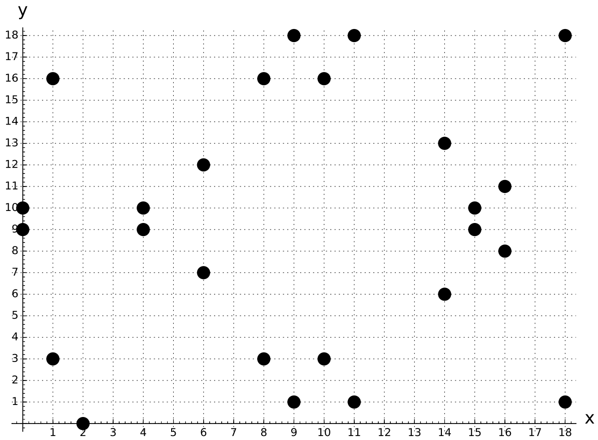
Example: Elliptic Curve over a Finite Field

Consider the elliptic curve $E : y^2 = x^3 + 3x + 5$ over $K = GF(19)$. We let SageMath find all points on the projective curve.

```
sage:   E=EllipticCurve(GF(19),[3,5])
sage:   E.points()
[(0:1:0), (0:9:1), (0:10:1), (1:3:1), (1:16:1),
 (2:0:1), (4: 9:1), (4:10:1), (6:7:1), (6:12:1),
 (8:3:1), (8:16:1), (9:1:1), (9:18:1), (10:3:1),
 (10:16:1), (11:1:1),  (11:18:1), (14:6:1),
 (14:13:1), (15:9:1), (15:10:1), (16:8:1),
 (16:11:1), (18:1:1), (18:18:1)]
```

The points on the affine curve are of the form $[x : y : 1]$. Note that there is one extra point $O = [0 : 1 : 0]$ at infinity. $E(K)$ is an abelian group of order 26 so that $E(K)$ is isomorphic to $\mathbb{Z}_{26} \cong \mathbb{Z}_{13} \times \mathbb{Z}_2$.

Example: Points of an Elliptic Curve over a Finite Field



Points on the elliptic curve $y^2 = x^3 + 3x + 5$ *over* $GF(19)$.

Group Law

A very important fact is that points in $E(K)$ *can be added*. However, addition is not the usual vector addition in K^2.

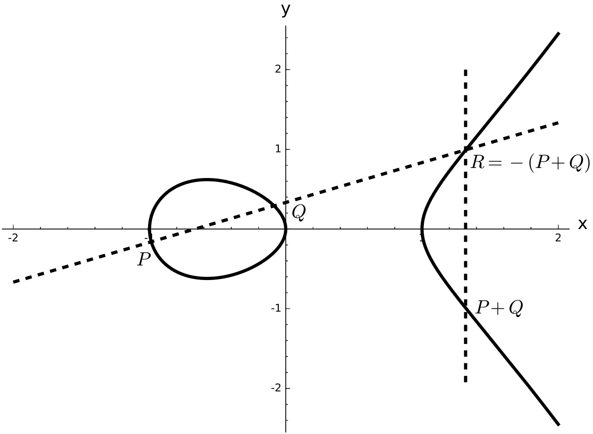
The identity element is the point O at infinity. A line through two nonsingular points P and Q intersects the elliptic curve at a third point R and we set $P + Q + R = O$.

Note: Since E is defined by a cubic equation, two solutions in K (or a double root) yield a third solution in K.

Theorem

$E(K)$ forms an abelian group with identity element O.

Addition of Points



The elliptic curve $E : y^2 + y = x^3 - x$ over the real numbers. The line through P and Q intersects the curve in R and $P + Q = -R$.

Formulas for Point Addition and Doubling

Let K be a field with $char(K) \neq 2, 3$ and E an elliptic curve over K, defined by the Weierstrass equation $y^2 = x^3 + ax + b$. Let $P = (x_1, y_1)$, $Q = (x_2, y_2) \in E(K)$, $P \neq O$, $Q \neq O$ and $P \neq -Q$. Then:

$$P + Q = (x_3, y_3),\ x_3 = m^2 - x_1 - x_2,\ y_3 = m(x_1 - x_3) - y_1,$$

$$\text{where } m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q. \end{cases}$$

m is the slope of the line through P and Q or the tangent line, if $P = Q$.

Fast Multiplication of Points

The algorithms in elliptic curve cryptography require the fast *multiplication* of points. Analogous to the square-and-multiply algorithm for a fast exponentiation, multiplying a point by a factor can be done recursively by doubling and adding points (*double-and-add algorithm*).

Example: $17 \cdot P = 2 \cdot (2 \cdot (2 \cdot (2 \cdot P)))) + P$.

Cryptographic Applications

For cryptographic use, one chooses a finite field $K = GF(p)$ or $K = GF(2^m)$, an elliptic curve E over K and a *base point* $g \in E(K)$.

The point g generates a cyclic subgroup

$$G = < g > \subset E(K)$$

of order $n = ord(g)$. n should be a large prime or at least contain a large prime factor. The cofactor is defined as $h = \frac{ord(E(K))}{n}$ and usually h is small or equal to 1.

Number of Points on an Elliptic Curve

There are efficient algorithms to compute the order of $E(K)$ if K is a finite field. Hasse's Theorem gives the *approximate number of points*:

Theorem

Let E be an elliptic curve over a finite field K of order q. Then

$$|\, q+1-ord(E(K))\,| \leq 2\sqrt{q}.$$

Hence $E(GF(q))$ has about the same size as q.

Discrete Logarithm

Definition

Let E be an elliptic curve over a finite field K, $g \in E(K)$ a base point, $G = < g >$, $n = ord(G)$ and $A \in G$. Then the unique positive integer $a < n$ such that $a \cdot g = A$ is called the *discrete logarithm* $\log_g(A)$ of A.

The security of elliptic curve cryptography relies on the hardness of the discrete logarithm (DL) problem in the group $G \subset E(K)$. The elliptic curve and the *domain parameters* must be carefully chosen, since there are less secure curves where the computation of discrete logarithms can be reduced to an easier DL problem.

Elliptic curve cryptography is widely standardized by national and international organizations (e.g., ISO, ANSI, NIST, IEEE, IETF) and one of the proposed curves is usually chosen.

Example

Consider the curve `brainpoolP256r1` (RFC 5639). The curve is defined by the Weierstrass equation $y^2 = x^3 + ax + b$ over a 256-bit field $K = GF(p)$. The base point $g = (x_g, y_g)$ generates the full group $G = E(K)$ and $n = ord(g)$ is a 256-bit prime number.

```
p =  A9FB57DBA1EEA9BC3E660A909D838D726E3BF623D52620282013481D1F6E5377
a =  7D5A0975FC2C3057EEF67530417AFFE7FB8055C126DC5C6CE94A4B44F330B5D9
b =  26DC5C6CE94A4B44F330B5D9BBD77CBF958416295CF7E1CE6BCCDC18FF8C07B6
g =  (xg,yg)
xg=  8BD2AEB9CB7E57CB2C4B482FFC81B7AFB9DE27E1E3BD23C23A4453BD9ACE3262
yg=  547EF835C3DAC4FD97F8461A14611DC9C27745132DED8E545C1D54C72F046997
n  = A9FB57DBA1EEA9BC3E660A909D838D718C397AA3B561A6F7901E0E82974856A7
h  = 1
```

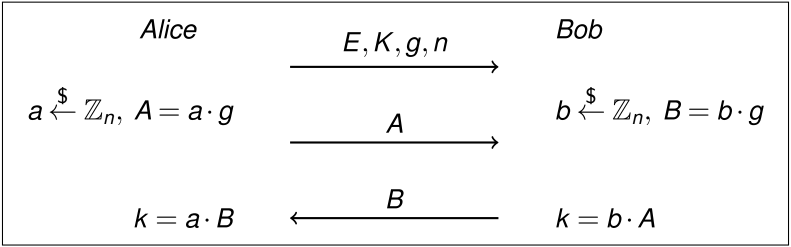
Elliptic Curve Diffie-Hellman (ECDH)

The common choice for Diffie-Hellman are subgroups of the multiplicative group $\mathbb{Z}_p^*$, but over the last decade, elliptic curves have become increasingly popular since they permit shorter keys.

For a Diffie-Hellman key exchange with elliptic curves, the communication partners (say Alice and Bob) have to agree on a finite field K, an elliptic curve E over K and a base point g generating a group G of order n. Usually, they would choose a standard curve and its domain parameters.

An eavesdropper, who only knows A and/or B as well as the elliptic curve and its domain parameters, should not be able to derive a, b or k if the computational Diffie-Hellman (CDH) problem is hard in G.

Elliptic-Curve Diffie-Hellman Key Exchange



Alice $\xrightarrow{E, K, g, n}$ *Bob*

$a \overset{\$}{\leftarrow} \mathbb{Z}_n,\ A = a \cdot g$ $\xrightarrow{A}$ $b \overset{\$}{\leftarrow} \mathbb{Z}_n,\ B = b \cdot g$

$k = a \cdot B$ $\xleftarrow{B}$ $k = b \cdot A$

Elliptic-curve Diffie-Hellman key exchange between Alice and Bob.

For a uniform output, the x-coordinate of the point k is taken as input of a key derivation or hash function.

Example

Alice and Bob agree on the elliptic curve $y^2 = x^3 + 3x + 5$ over $GF(19)$ and the base point is $g = 2 \cdot (1,3) = (18,18)$. The point g has order 13.

Alice chooses the secret key $a = 2$ and computes $A = a \cdot g = 2 \cdot (18,18) = (11,18)$. Bob chooses the secret key $b = 4$ and computes $B = b \cdot g = 4 \cdot (18,18) = (8,3)$.

Alice obtains the shared secret key by computing

$$k = a \cdot B = 2 \cdot (8,3) = (9,1).$$

Bob computes

$$k = b \cdot A = 4 \cdot (11,18) = (9,1).$$

Other Applications of Elliptic Curve Cryptography

Elliptic curves can also be used for key establishment, encryption and signatures. Elliptic-curve algorithms can replace RSA or classical Diffie-Hellman with the multiplicative group.

- Diffie-Hellman Key Encapsulation Mechanism
- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Elliptic Curve Integrated Encryption Scheme (ECIES)

Furthermore, elliptic curves can be used to factorize large integers and to attack RSA.

- Elliptic Curve Factoring Method (ECM)

Security of Elliptic Curve Cryptography

The *security* of elliptic curve schemes relies on the hardness of the discrete-logarithm (DL) problem. A major difference to the multiplicative group is that the known sub-exponential algorithms cannot be applied to elliptic curves.

If $n = ord(G)$ is a prime number, then the best known algorithms for computing discrete logarithms on elliptic curves are Babystep-Giantstep and Pollard's ρ-method for logarithms. Their complexity is $O(\sqrt{n})$. However, the elliptic curve needs to be carefully chosen in order to prevent certain types of attacks.

Comparison

The following table shows comparable strengths for different algorithms and key lengths (in bits). The table compares symmetric encryption schemes (key size), RSA (size of the modulus N), Diffie-Hellman using a subgroup of order q in $\mathbb{Z}_p^*$ (size of p, size of q) and Diffie-Hellman with elliptic curves (size of n, where n is the order of the cyclic group of points).

Symmetric Encryption	RSA	DH	ECDH
80	1024	$1024,\ 160$	$160 - 223$
112	2048	$2048,\ 224$	$224 - 255$
128	3072	$3072,\ 256$	$256 - 383$
192	7680	$7680,\ 384$	$384 - 511$
256	15360	$15360,\ 512$	$512+$

Comparable algorithm strengths for different key lengths.