

# 7. Elliptic Curve Digital Signature Algorithm (ECDSA)

## A. Elliptic Curves over Finite Field

Let

- $p > 3$  be a prime and  $\mathbb{F} = GF(p^n)$  be a finite field.
- $x^3 + ax + b$  where  $a$  and  $b \in \mathbb{F}$ , a cubic polynomial with no multiple roots.

An **elliptic curve**  $E$  over  $\mathbb{F}$  is the set of points  $(x, y)$  with  $x, y \in \mathbb{F}$  which satisfy the equation

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

together with a single element denoted by  $\mathcal{O}$  and called the “**point at infinity**”.

## Additional Law

For  $P = (x_1, y_1) \in E$  and  $Q = (x_2, y_2) \in E$ , then

- 1  $-P = (x_1, -y_1)$
- 2  $P + Q = R = (x_3, y_3)$  where the coordinates of the point  $R$  are defined by

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1$$

where

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

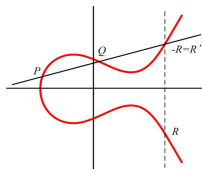


Figure: Point Addition

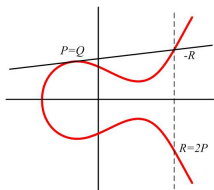


Figure: Point Doubling

## Example

Consider an elliptic curve

$$E : y^2 = x^3 + x + 6 \text{ over } GF(11)$$

To find all points  $(x, y)$  of  $E$  for

each  $x \in GF(11)$ , compute

$z = x^3 + x + 6 \pmod{11}$  and

determine whether  $z$  is a

**quadratic residue** (QR).

If so, solve  $y^2 = z$  in  $GF(11)$ . We

can find there are totally 13 points

on this curve.

$x$	$x^3 + x + 6$	QR?	$y$
0	6	no	–
1	8	no	–
2	5	yes	4, 7
3	3	yes	5, 6
4	8	no	–
5	4	yes	2, 9
6	8	no	–
7	4	yes	2, 9
8	9	yes	3, 8
9	7	no	–
10	4	yes	2, 9

## Example (continued)

There are 13 points in the group.

So, it is **cyclic** and **any point** other  $\mathcal{O}$  is generator.

Let  $P = (2, 7)$ . We can compute  $2P = (x_2, y_2)$  as follows.

$$\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{3 \cdot 2^2 + 1}{2 \cdot 7} = \frac{13}{14} = 2 \cdot 3^{-1} = 2 \cdot 4 = 8 \pmod{11}$$

$$x_2 = \lambda^2 - 2x_1 = 8^2 - 2 \cdot 2 = 5 \pmod{11}$$

$$y_2 = (x_1 - x_2)\lambda - y_1 = (2 - 5) \cdot 8 - 7 = 2 \pmod{11}$$

Therefore, we obtain  $2P = (5, 2)$ .

## Example (continued)

Let  $3P = P + 2P = (x_3, y_3)$ . Then we can compute  $3P$  as follows.

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2 - 7}{5 - 2} = 2 \pmod{11}$$

$$x_3 = \lambda^2 - x_1 - x_2 = 2^2 - 2 - 5 = 8 \pmod{11}$$

$$y_3 = (x_1 - x_3)\lambda - y_1 = (2 - 8) \cdot 2 - 7 = 3 \pmod{11}$$

Hence, we obtain  $3P = (8, 3)$ .

Similarly, we can also compute the cyclic group generated by  $P$ .

$$\begin{array}{cccc} P = (2, 7) & 2P = (5, 2) & 3P = (8, 3) & 4P = (10, 2) \\ 5P = (3, 6) & 6P = (7, 9) & 7P = (7, 2) & 8P = (3, 5) \\ 9P = (10, 9) & 10P = (8, 8) & 11P = (5, 9) & 12P = (2, 4) \\ 13P = P + 12P = 2P + 11P = 3P + 10P = \dots = \mathcal{O} \end{array}$$

## B. EC-DSA(IEEE P1363/D4, 1998)

### a. System set-up

- 1 Choose  $p$ , a prime, and  $n$ , an integer,  $f(x)$ , an irreducible polynomial over  $GF(p)$  of degree  $n$ , generating finite field  $GF(p^n)$  with the defining polynomial  $f(x)$ , and assume that  $\alpha$  is a root of  $f(x)$  in  $GF(p^n)$ .
- 2 Generate a **non-supersingular** curve  $E$  over  $GF(p^n)$ .
- 3 Choose a point  $P = (x, y)$  on  $E$  of order  $q$  which is **prime**.
- 4 **Converting function:**

$$c(x) : GF(p^n) \rightarrow Z_{p^n}$$

which is given by

$$c(x) = \sum_{i=0}^{n-1} c_i p^i \in Z_{p^n}, \text{ for } x = \sum_{i=0}^{n-1} c_i \alpha^i \in GF(p^n), 0 \leq c_i < p$$

## b. Key Generation

- 1 Private key  $d$ , which is an integer, randomly selected as  $0 < d < q$ .
- 2 Public key  $Q$ , which is a point on  $E$  and computed by  $Q = dP = (x_d, y_d)$ .

### c. Signing for a message $m$

- 1 Generate a **ont-time** key pair  $(k, R)$  in the following way: randomly choose  $k$ :  $0 < k < q$  and compute a point  $R = kP = (x_k, y_k)$ .
- 2 Compute  $r$  in the following way: generating an integer by using the converting function for converting  $x_k$  into a  $p$ -ary number:

$$x_k = \sum_{i=0}^{n-1} c_{i,k} \alpha^i \text{ converting into } c(x_k) = r = \sum_{i=0}^{n-1} c_{i,k} p^i, 0 \leq i < p.$$

- 3 Compute  $s$ :

$$h(m) = dr + ks \pmod{q},$$

where  $h(x)$  is the cryptographic hash function.

The pair  $(r, s)$  is a **signature** of the message  $m$ .

## d. Verifying

- 1 Compute numbers:

$$t = s^{-1} \bmod q$$

$$t_1 = h(m) \cdot t \bmod q$$

$$t_2 = t \cdot r \bmod q$$

- 2 Compute a point on  $E$  by using the system key  $P$  and user's public key  $Q$ :

$$t_1 P - t_2 Q = (x_k, y_k)$$

- 3 By using the converting function to compute the integer  $c(x_k)$  and check if  $r = c(x_k) \bmod q$ . If this is true, then  $(r, s)$  is accepted as a valid signature of the message  $m$ .

# 8. Identity-based Cryptography from Bilinear Pairing

## A. Identity-based Cryptography

The difference between “an identity-base cryptosystem” and “a traditional cryptosystem” include the following four aspects:

- How to **generate** a key
- How to **authenticate** the key
- How to **distribute** the key
- How to **use** the key

## B. Comparison of Three Different Cryptosystems

- Symmetric Key Cryptosystems
  - ▶ The sender and the receiver share a **secret key** through a **authenticated and secret channel**.
- Public Key Cryptosystems
  - ▶ A **Certificate Authority (CA)** issues a certificate for each user.
  - ▶ The certificate binds user's public key and his/her identity.
  - ▶ The sender has to obtain the receiver's public key and certificate through a **authenticated channel** before starting secure communications.
- Identity-based Cryptosystems
  - ▶ A user's **identity** is his/her **public key**.
  - ▶ A public key could be any personal information, such as an email address, a phone number, a post address, a photo, etc.

## C. Pairing and Bilinear Groups

- Let  $\mathbb{G}_1, \mathbb{G}_2$  and  $\mathbb{G}_T$  be **cyclic groups of prime order  $q$**
- Let  $P_1$  be a generator of  $\mathbb{G}_1$  and  $P_2$  is a generator of  $\mathbb{G}_2$
- Let  $\hat{e}$  be a map  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ , which is called a **pairing**
- The pairing has the following properties
  - ▶ **Bilinear**: For all  $P \in \mathbb{G}_1$ , all  $Q \in \mathbb{G}_2$  and all  $a, b \in Z$  we have  $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
  - ▶ **Non-degenerate**:  $\hat{e}(P_1, P_2) \neq 1$
  - ▶ **Computable**: There exists an efficient algorithm to compute  $\hat{e}(P, Q)$  for all  $P \in \mathbb{G}_1$  and  $Q \in \mathbb{G}_2$

## D. Pairing Based Hard Problems

- **Bilinear Diffie-Hellman (BDH) assumption:**
  - ▶ For  $a, b, c \in_R Z_q^*$ , given  $(aP_i, bP_j, cP_k)$ , for some values of  $i, j, k \in \{1, 2\}$ , computing  $\hat{e}(P_1, P_2)^{abc}$  is hard.
- **Decisional BDH (DBDH) assumption:**
  - ▶ For  $a, b, c, r \in_R Z_q^*$ , differentiating  $(aP_i, bP_j, cP_k, \hat{e}(P_1, P_2)^{abc})$  and  $(aP_i, bP_j, cP_k, \hat{e}(P_1, P_2)^r)$ , for some values of  $i, j, k \in \{1, 2\}$ , is hard.
- **Bilinear DH Inversion (k-BDHI) assumption:**
  - ▶ For and integer  $k$ , and  $a \in_R Z_q^*$ , given  $(aP_i, a^2P_i, \dots, a^kP_i)$  for  $i \in \{1, 2\}$ , computing  $\hat{e}(P_1, P_2)^{1/a}$  is hard.

The above three are well-known assumptions, which are used to analyze security of pairing-based cryptographic mechanisms.

## E. The Boneh-Franklin Identity-based Encryption (IBE) Scheme

### a. Key Generation

- **Master Private Key:**  $s \in_R \mathbb{Z}_q^*$
- **Master Public Key:**  $P$  is a generator of  $\mathbb{G}_1$ ,  $P_{pub} = sP$
- **User Public Key:**
  - ▶  $ID$  is an identity date string
  - ▶  $H_1$  is a hash function (MapToPoint) –  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$
  - ▶ Public key:  $Q_{ID} = H_1(ID) \in \mathbb{G}_2$
- **User Private Key:**  $D_{ID} = sQ_{ID} \in \mathbb{G}_2$

## b. Encryption and Decryption

### 1 Hash functions:

- ▶  $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$
- ▶  $H_3 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{Z}_q^*$
- ▶  $H_4 : \{0, 1\}^n \rightarrow \{0, 1\}^n$

### 2 Encryption: $\text{Encrypt}(m) \rightarrow C$

- ▶  $\sigma \in_R \{0, 1\}^*$ ,  $r = H_3(\sigma, m)$ ,  $g_{ID} = \hat{e}(Q_{ID}, P_{pub})$
- ▶  $C = (U, V, W) = (rP, \sigma \oplus H_2(g_{ID}^r), m \oplus H_4(\sigma))$

### 3 Decryption: $\text{Decrypt}(U, V, W) \rightarrow (m \text{ or "invalid"})$

- ▶  $\sigma = V \oplus H_2(\hat{e}(D_{ID}, U))$ ,  $m = W \oplus H_4(\sigma)$ ,  $r = H_3(\sigma, m)$
- ▶ If  $U = rP$ , return  $m$ ; else return "invalid"

## F. The Hess Identity-based Signature Scheme (ISO/IEC 14888-3)

- Key generation is similar to IBE
  - ▶ Master Public Key:  $P, sP$ ; Master Private Key:  $s$
  - ▶ Signer's Private Key:  $sQ$  where  $Q = H_1(ID)$
- Hash functions:
  - ▶  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$
  - ▶  $H_2 : \{0, 1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$
- **Sign on message  $m$ :** Signature is  $(h, S)$ 
  - ▶  $k \in_R \mathbb{Z}_q^*$
  - ▶  $T = \hat{e}(sQ, P)^k$
  - ▶  $h = H_2(m, T)$
  - ▶  $S = (k - h)sQ$
- **Verify  $(h, S)$ :**
  - ▶  $T = \hat{e}(s, P)\hat{e}(Q, sP)^h$
  - ▶ Accept the signature if and only if  $h = H_2(m, T)$

## G. The Cha-Cheon Identity-based Signature Scheme (ISO/IEC 14888-3)

- Key generation is similar to IBE
  - ▶ Master Public Key:  $P, sP$ ; Master Private Key:  $s$
  - ▶ Signer's Private Key:  $sQ$  where  $Q = H_1(ID)$
- Hash functions:
  - ▶  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$
  - ▶  $H_2 : \{0, 1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$
- Sign on message  $m$ : Signature is  $(T, S)$ 
  - ▶  $r \in_R \mathbb{Z}_q^*$
  - ▶  $T = rQ$
  - ▶  $h = H_2(m, T)$
  - ▶  $S = (r + h)sQ$
- Verify  $(T, S)$ :
  - ▶  $h = H_2(m, T)$
  - ▶ Accept the signature if and only if  $\hat{e}(P, S) = \hat{e}(sP, T + hQ)$

## G. The Smart-Chen-Kudla Identity-based Key Agreement Scheme

- Key generation:
  - ▶ Master Public Key:  $P, sP$ ; Master Private Key:  $s$
  - ▶ User  $X$ 's Private Key:  $sQ_X$  where  $Q_X = H_1(ID)$

$$A$$

$$sQ_A, a \in_R Z_q^*$$

$$B$$

$$sQ_B, b \in_R Z_q^*$$

$$\begin{array}{c} \xrightarrow{t_A = aP} \\ \xleftarrow{t_B = bP} \end{array}$$

$$\begin{aligned} K_{AB} &= abP \parallel \hat{e}(sQ_A, t_B) \hat{e}(Q_B, asP) \\ &= abP \parallel \hat{e}(bQ_A + aQ_B, sP) \end{aligned}$$

$$\begin{aligned} K_{BA} &= abP \parallel \hat{e}(sQ_B, t_A) \hat{e}(Q_A, bsP) \\ &= abP \parallel \hat{e}(bQ_A + aQ_B, sP) \end{aligned}$$

## H. The Chen-Kudla Identity-based Key Agreement Scheme

- Key generation:
  - ▶ Master Public Key:  $P, sP$ ; Master Private Key:  $s$
  - ▶ User  $X$ 's Private Key:  $sQ_X$  where  $Q_X = H_1(ID)$

$$A \\ sQ_A, a \in_R Z_q^*$$

$$B \\ sQ_B, b \in_R Z_q^*$$

$$\begin{array}{c} \xrightarrow{t_A = aQ_A} \\ \xleftarrow{t_B = bQ_B} \end{array}$$

$$\begin{aligned} K_{AB} &= \hat{e}(sQ_A, t_B + aQ_B) \\ &= \hat{e}(Q_A, Q_B)^{s(a+b)} \end{aligned}$$

$$\begin{aligned} K_{BA} &= \hat{e}(t_A + bQ_A, sQ_B) \\ &= \hat{e}(Q_A, Q_B)^{s(a+b)} \end{aligned}$$