Discrete-log and elliptic-curve based cryptography

Jean-Sébastien Coron

University of Luxembourg

April 26, 2025

Discrete-log and elliptic-curve based cryptography

- Previous lecture: discrete-log based group
  - The multiplicative group $\mathbb{Z}_p^*$
  - ElGamal encryption: security proof
  - Diffie-Hellman key exchange
  - Schnorr signature scheme
- Elliptic-Curve Cryptography
  - Defines an alternative group, with generally shorter keys.
  - El-Gamal over ECC
- Pairing-based cryptography
  - Application to identity-based encryption.
- How to hash into elliptic-curves.
  - Icart's function

Discrete-log and elliptic-curve based cryptography

- Previous lecture: discrete-log based group
  - The multiplicative group $\mathbb{Z}_p^*$
  - ElGamal encryption: security proof
  - Diffie-Hellman key exchange
  - Schnorr signature scheme
- Elliptic-Curve Cryptography
  - Defines an alternative group, with generally shorter keys.
  - El-Gamal over ECC
- Pairing-based cryptography
  - Application to identity-based encryption.
- How to hash into elliptic-curves.
  - Icart's function

- Previous lecture: discrete-log based group
  - The multiplicative group $\mathbb{Z}_p^*$
  - ElGamal encryption: security proof
  - Diffie-Hellman key exchange
  - Schnorr signature scheme
- Elliptic-Curve Cryptography
  - Defines an alternative group, with generally shorter keys.
  - El-Gamal over ECC
- Pairing-based cryptography
  - Application to identity-based encryption.
- How to hash into elliptic-curves.
  - Icart's function

Discrete-log and elliptic-curve based cryptography

- Previous lecture: discrete-log based group
  - The multiplicative group $\mathbb{Z}_p^*$
  - ElGamal encryption: security proof
  - Diffie-Hellman key exchange
  - Schnorr signature scheme
- Elliptic-Curve Cryptography
  - Defines an alternative group, with generally shorter keys.
  - El-Gamal over ECC
- Pairing-based cryptography
  - Application to identity-based encryption.
- How to hash into elliptic-curves.
  - Icart's function

The multiplicative group $\mathbb{Z}_p^*$

- Let p be a prime integer.
  - The set $\mathbb{Z}_p^*$ is the set of integers modulo p which are invertible modulo p.
  - The set $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$ for the operation of multiplication modulo p.
- Generators of $\mathbb{Z}_p^*$:
  - There exists $g \in \mathbb{Z}_p^*$ such that any $h \in \mathbb{Z}_p^*$ can be uniquely written as $h = g^x \pmod{p}$ with $0 \leq x < p-1$.
  - The integer x is called the *discrete logarithm* of h to the base g, and denoted $\log_g h$.

Elliptic Curves

- Defines a new group different from $\mathbb{Z}_p^*$
  - Security based on the Elliptic Curve Discrete Logarithm Problem (ECDLP)
  - Advantage: shorter keys
- Elliptic-curve equation over $\mathbb{Z}_p$:
  - $y^2 = x^3 + ax + b$ where $a, b \in \mathbb{Z}_p$
- Group structure
  - The set of points together with $\mathcal{O}$ can define a group structure, where $\mathcal{O}$ is the point at infinity.

EC: addition formula in char $\neq 2, 3$

- The group law is defined geometrically by point addition and point doubling
- Let $P = (x_1, y_1) \neq \mathcal{O}$ and $Q = (x_2, y_2) \neq \mathcal{O}$. Then $P + Q = (x_3, y_3)$ with:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned}$$

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

- $P = (x_1, y_1) \neq \mathcal{O} \Rightarrow -P = (x_1, -y_1)$

Computing a multiple of a point

- **Double-and-add Algorithm:**
  input P and $d = (d_{\ell-1}, \ldots, d_0)$
  output $Q = dP$

  $Q \leftarrow P$
  for i from $\ell - 2$ downto 0 do
  $\quad Q \leftarrow 2Q$
  $\quad$ if $d_i = 1$ then $Q \leftarrow Q + P$
  output Q

- Complexity of computing $Q = dP$
  - $\mathcal{O}(\log d)$ operations

Computing the group order

- Ordinary elliptic-curves
  - $y^2 = x^3 + ax + b \pmod{p}$
  - Let n be the number of points, including $\mathcal{O}$.
  - We must have $n = k \cdot q$ where q is a large prime.
  - then work in subgroup of order q.
- Computing the group order n:
  - Schoof's algorithm.
  - Schoof-Elkies-Atkin algorithm.
  - or use standardized curves.

EC El-Gamal encryption

- Key generation
  - Let $\mathbb{G}$ be an elliptic curve subgroup of prime order q and G a generator of $\mathbb{G}$.
  - Let $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_q$. Let $H = \alpha G$.
  - Public-key : (G, H). Private-key : α
- Encryption of m :
  - Let $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$
  - Output $c = (rG, (rH)_x \oplus m)$ where $(rH)_x$ denotes the x coordinate of rH.
- Decryption of $c = (C_1, c_2)$
  - Output $m = (\alpha C_1)_x \oplus c_2$

- Pairing-based cryptography
  - Special bilinear map between two groups to build advanced cryptographic protocols.
  - A function $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ where $\mathbb{G}$ and $\mathbb{G}_1$ are groups of prime order q.
  - $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}$.
  - Can be constructed from elliptic curves using the Weil or Tate pairing.
- Applications
  - Identity-Based Encryption (IBE), short signatures, broadcast encryption...

Bilinear map

- Bilinear map :
  - Let $\mathbb{G}$ and $\mathbb{G}_1$ be groups of order q, for a large prime q. Let g be a generator of $\mathbb{G}$.
  - Bilinear map: function e such that

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$$

- Properties of bilinear map
  - Bilinear: $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}$.
  - Non-degenerate: $e(g, g) \neq 1$.
  - Computable: there exists an efficient algorithm to compute $e(h_1, h_2)$ for any $h_1, h_2 \in \mathbb{G}$.

Implementation of bilinear map

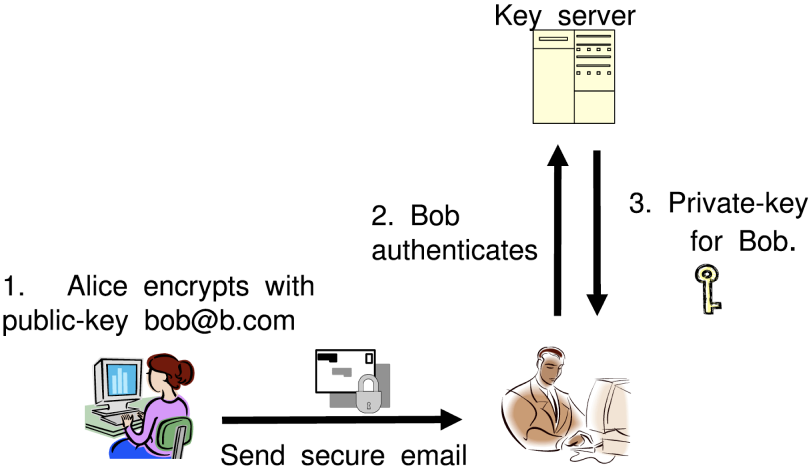
- Weil pairing or Tate pairing over an elliptic curve.
  - Let p be a large prime with $p = 2 \pmod 3$. Consider the Elliptic-Curve:
  $$E/\mathbb{F}_p : y^2 = x^3 + 1$$
  - The curve satisfies $\#E(\mathbb{F}_p) = p + 1$.
- Definition of the Weil Pairing
$$e(P, Q) = \frac{f_P(\mathcal{A}_Q)}{f_Q(\mathcal{A}_P)}$$
- Computing the Weil pairing
  - Using Miller's algorithm.
  - Algorithm in $\mathcal{O}(\log p)$ arithmetic operations mod p => $\mathcal{O}(\log^3 p)$ elementary operations.

Application of the elliptic-curve pairing

- Identity-Based Encryption (IBE)
  - Concept invented in 1984 by Adi Shamir.
  - First practical realization in 2001 by Boneh and Franklin, based on bilinear pairing operation over an elliptic-curve.
- Principle:
  - IBE allows for a party to encrypt a message using the recipient's identity as the public-key.
  - The corresponding private-key is provided by a central authority.

IBE

- Alice sends an email to Bob using his identity as the



IBE

- Principle
  - Alice encrypts her email using Bob's email address `bob@b.com` as the public-key.
  - Bob receives the message. Bob contacts the key server, authenticates and obtains his private key.
  - Bob can use his private-key to decrypt the message.
  - The private-key can be used to decrypt any future message sent to Bob by Alice or any other user.
- Advantages
  - Avoids the need to distribute PK certificates.
  - Users can use their email address as their identity
- Drawback
  - The key server can decrypt any communication

Definition of IBE

- Setup
  - Output: system public parameters *params*, and private master-key *master-key*.
- Keygen
  - Input: *params*, *master-key* and identity v.
  - Output: private key d_v for v.
- Encrypt
  - Input: message m, identity v and *params*.
  - Output: ciphertext c.
- Decrypt
  - Input: *params*, ciphertext c and private-key d_v.
  - Output: plaintext m.

The Boneh-Franklin IBE scheme

- We describe the basic scheme, which achieves only CPA security
  - Based on bilinear map: $e(g^a, h^b) = e(g, h)^{ab}$
- Setup
  - Let $\mathbb{G} = \langle g \rangle$ of prime order p. Let $H_1 : \{0, 1\}^* \to \mathbb{G}$ a hash function.
  - Generate random $a \in \mathbb{Z}_p$. Let $h = g^a$.
  - Public: (g, h). Secret: a.
- Keygen
  - Let v be an identity. Private-key $d_v = H_1(v)^a$

Boneh-Franklin

- Encryption
  - Generate a random $r \in \mathbb{Z}_p$.

$$C = \left(g^r,\ m \oplus H_2\big(e(H_1(v), h)^r\big)\right)$$

- Decryption
  - To decrypt $C = (c_1, c_2)$ using $d_v = H(v)^a$, compute:

$$m = H_2\big(e(d_v, c_1)\big) \oplus c_2$$

- Why decryption works
  - Using the bilinearity of e

$$e(H_1(v), h)^r = e(H_1(v), g^a)^r = e(H_1(v)^a, g^r) = e(d_v, c_1)$$

Security of Boneh-Franklin

- The security of the Boneh-Franklin scheme can be proven secure
  - in the random oracle model
  - under the BDH assumption.
- BDH assumption
  - BDH problem: given (g, g^a, g^b, g^c), output $e(g, g)^{abc}$.
  - BDH assumption: there is no efficient algorithm that solves the BDH problem.

Hashing into elliptic-curves

- Hashing into Elliptic Curves
  - Boneh-Franklin IBE: $Q_{id} = H_1(id)$ on the curve.
  - Password based authentication protocols (SPEKE, PAK).
- Boneh-Franklin: super-singular curve
  - Special curve with special operation: pairing.
  - Hashing is easy.
  - But larger parameters are required.
- How to hash into ordinary curves ?

SPEKE

- Simple Password Exponential Key Exchange (Jablon, 1996)
  - Let pw be a password shared by Alice and Bob
  - Let E be the subgroup of an elliptic curve of order q.
- Protocol
  - Alice sends $A = a.H(pw)$ to Bob, where $a \leftarrow \mathbb{Z}_q$
  - Bob sends $B = b.H(pw)$ to Alice, where $b \leftarrow \mathbb{Z}_q$
  - Alice computes $K = a.B = ab.H(pw)$
  - Bob computes $K = b.A = ab.H(pw)$

Try and Increment

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Try and Increment:
  Input: u an integer. We can take $u = H(m)$.
  Output: Q, a point of $E_{a,b}(\mathbb{F}_p)$.
  1. For $i = 0$ to $k - 1$
     1. Set $x = u + i$
     2. If $x^3 + ax + b$ is a quadratic residue in $\mathbb{F}_p$, then return $Q = (x, (x^3 + ax + b)^{1/2})$
  2. end For
  3. Return $\perp$
- Timing attack
  - The number of trials varies with the input, timing side-channel leaks information about the input

Supersingular Elliptic Curve

- Supersingular curve:

$$E : y^2 = x^3 + 1 \pmod{p}$$

  - with $p = 2 \pmod{3}$
  - It has $p + 1$ points.
- Hashing into E:
  - Let $y = H(m)$
  - Let $x = (y^2 - 1)^{1/3}$
  - Return $P = (x, y)$
- p must be large because of MOV attack (at least 512 bits)

Hashing into Ordinary Curves

- Elliptic curve:

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

- Icart's function
  - Published by Thomas Icart at CRYPTO 2009
  - Deterministic function into E
  - Requires $p = 2 \pmod 3$
  - Essentially one exponentiation in $\mathbb{F}_p$
- Shallue-Woestijne-Ulas algorithm
  - Deterministic algorithm into E (but requires a test)
  - Does not require $p = 2 \pmod 3$
  - Essentially one exponentiation in $\mathbb{F}_p$

Hashing into Ordinary Curves

- Elliptic curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

- Icart's function
  - Published by Thomas Icart at CRYPTO 2009
  - Deterministic function into E
  - Requires $p = 2 \pmod{3}$
  - Essentially one exponentiation in $\mathbb{F}_p$
- Shallue-Woestijne-Ulas algorithm
  - Deterministic algorithm into E (but requires a test)
  - Does not require $p = 2 \pmod{3}$
  - Essentially one exponentiation in $\mathbb{F}_p$

Icart's Function

- Elliptic curve with $p = 2 \pmod 3$:

$$E_{a,b} : y^2 = x^3 + ax + b \pmod p$$

- Icart's function: (we can have $u = H(m)$)

$$\begin{array}{rcl} f_{a,b} : \mathbb{F}_p & \mapsto & E_{a,b} \\ u & \mapsto & (x, y) \end{array}$$

$$\begin{array}{rcl} x & = & \left(v^2 - b - \frac{u^6}{27}\right)^{(2p-1)/3} + \frac{u^2}{3} \\ y & = & ux + v \end{array}$$

$$v = \frac{3a - u^4}{6u}.$$

Icart's Function

- Elliptic curve with $p = 2 \pmod 3$:

$$E_{a,b} : y^2 = x^3 + ax + b \pmod p$$

- Icart's function: (we can have $u = H(m)$)

$$\begin{aligned} f_{a,b} : \mathbb{F}_p &\mapsto E_{a,b} \\ u &\mapsto (x, y) \end{aligned}$$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{(2p-1)/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

$$v = \frac{3a - u^4}{6u}.$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$
\begin{aligned}
x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\
y &= ux + v
\end{aligned}
$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$
\begin{aligned}
x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\
y &= ux + v
\end{aligned}
$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$
\begin{aligned}
x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\
y &= ux + v
\end{aligned}
$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod p$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$
\begin{aligned}
x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\
y &= ux + v
\end{aligned}
$$

Why it Works

- $E_{a,b} : y^2 = x^3 + ax + b \pmod{p}$
- Let $y = ux + v$ with u, v two parameters
- $u^2x^2 + 2uvx + v^2 = x^3 + ax + b$
- $x^3 - u^2x^2 + (a - 2uv)x + b - v^2 = 0$
- $(x - u^2/3)^3 + x(a - 2uv - u^4/3) = v^2 - b - u^6/27$
- We want: $a - 2uv - u^4/3 = 0$
  - We take $v = (3a - u^4)/(6u)$
- We get: $(x - u^2/3)^3 = v^2 - b - u^6/27$

$$\begin{aligned} x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3} \\ y &= ux + v \end{aligned}$$

Conclusion

- Discrete-logarithm based cryptography
  - Foundation of many classical protocols (ElGamal, Diffie-Hellman, Schnorr).
- Elliptic-curve cryptography
  - Provides similar security with much shorter keys, based on the ECDLP assumption.
- Pairing-based cryptography
  - Enables new applications such as Identity-Based Encryption (IBE).
- Hashing into elliptic curves