Radboud University

Cryptanalysis in elliptic curve cryptography

Cryptology – Autumn 2023

Monika Trimoska

December 4, 2023

Technische Universiteit Eindhoven

Elliptic curves

What is an elliptic curve?

Let $\mathbb{F}_q$ be a finite field. An **elliptic curve** over $\mathbb{F}_q$ is a curve given by an equation of the form

$$\boxed{E : y^2 = x^3 + Ax + B}$$

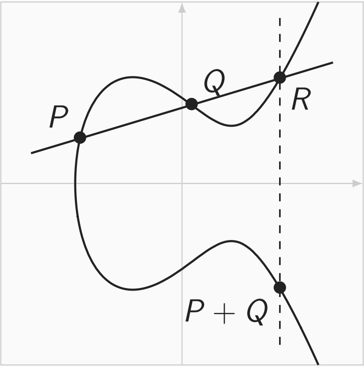
(short Weierstrass form)

with $A, B \in \mathbb{F}_q$.

- There is also a requirement that the discriminant $\Delta = 4A^3 + 27B^2$ is nonzero.
- The set of points on E with the addition law form a group.
- The group law is constructed geometrically.

The geometry of elliptic curves[1]

Adding points on an elliptic curve

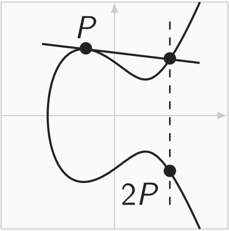


Addition $P + Q$

- Draw a line through P and Q
  $\hookrightarrow$ The line intersects the curve E at a third point R
- Draw a vertical line through R
  $\hookrightarrow$ The line intersects E in another point
- We define that point to be the sum of P and Q

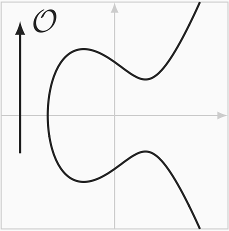
[1]Figures from the TikZ for Cryptographers library

Adding points on an elliptic curve

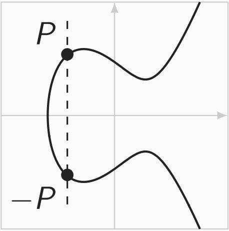


Doubling $P + P$

- Modify the first step : draw the tangent line to E at P



Neutral element $\mathcal{O}$



Inverse element $-P$

The algebra of elliptic curves

The addition law on E has the following properties:

- $P + \mathcal{O} = P$, for all $P \in E$
- Let $P \in E$. There is a point of E, denoted by $-P$, satisfying $P + (-P) = \mathcal{O}$
- $P + (Q + R) = (P + Q) + R$, for all $P, Q, R \in E$
- $P + Q = Q + P$, for all $P, Q \in E$.

Elliptic curves with points in $\mathbb{F}_p$ are finite groups

- Closure ✓
- Associativity ✓
- Identity element ✓
- Inverse element ✓

We can write down explicitly the formulas for the addition law on E.

$\hookrightarrow$

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$,
then $P_1 + P_2 = (x_3, y_3) = (\lambda^2 - x_1 - x_2, \lambda(x_3 - x_1) + y_1)$, where

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{when } P_1 \neq P_2 \\[2ex] \dfrac{3x_1^2 + a}{2y_1}, & \text{when } P_1 = P_2. \end{cases}$$

Elliptic Curve Discrete Logarithm Problem

Elliptic curve discrete logarithm problem

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given: points $P, Q \in E(\mathbb{F}_q)$

Find: an integer x such that $xP = Q$

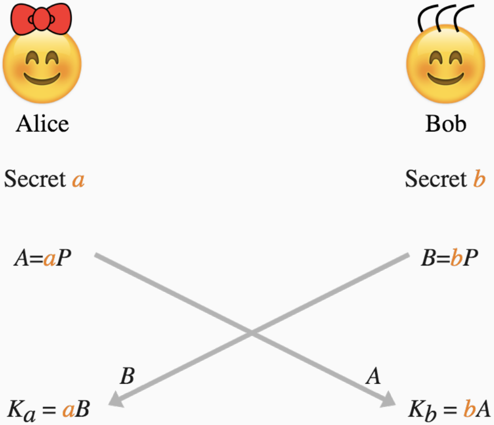
!

We can use the hardness of ECDLP only because computing multiples is easy.

$\hookrightarrow$ We can compute mP in $\mathcal{O}(\log m)$ steps by the usual Double-and-Add Method.

- First write $m = m_0 + m_1 \cdot 2 + m_2 \cdot 2^2 + \cdots + m_r \cdot 2^r$
- Then mP can be computed as $mP = m_0 P + m_1 \cdot 2P + m_2 \cdot 2^2 P + \cdots + m_r \cdot 2^r P$
- Requires r doublings (and sums)

ECDLP applications



$$K_a = abP = baP = K_b$$

Figure: Diffie-Hellman key exchange

ECDLP applications

- Diffie-Hellman key exchange
- ElGamal encryption and signatures
- Identification protocols
- Extension: Pairing-based crypto
- ECDSA used in all currently deployed cryptosystems:

Embedded SCTs

Log ID	76:FF:88:3F:0A:B6:FB:95:51:C2:61:CC:F5:87:BA:34:B4:A4:CD:BB:29:DC:68:42…
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Tue, 15 Aug 2023 07:27:07 GMT
Log ID	DA:B6:BF:6B:3F:B5:B6:22:9F:9B:C2:BB:5C:6B:E8:70:91:71:6C:BB:51:84:85:34…
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Tue, 15 Aug 2023 07:27:07 GMT
Log ID	EE:CD:D0:64:D5:DB:1A:CE:C5:5C:B7:9D:B4:CD:13:A2:32:87:46:7C:BC:EC:DE:…
Signature Algorithm	SHA-256 ECDSA
Version	1
Timestamp	Tue, 15 Aug 2023 07:27:07 GMT

ECDLP attacks

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given: points $P, Q \in E(\mathbb{F}_q)$

Find: an integer x such that $xP = Q$

Generic attacks

- Exhaustive Search
- Pollard's rho method
- Baby-step Giant-step
- Kangaroo
- **Parallel Collision Search**

Attacks on specific families

- MOV attack: using the Weil/Tate pairing
- Anomalous curves
- **Index calculus**

Parallel Collision Search

Collision search

What is a **collision**? Why does a collision help us solve the (EC)DLP?

$\hookrightarrow$ Having two different linear combinations of a random point $R \in E(\mathbb{F}_q)$

$$R = aP + bQ$$
$$R = a'P + b'Q,$$

we infer that

$$aP + bQ = a'P + b'Q$$
$$(a - a')P = (b' - b)xP,$$

and we compute

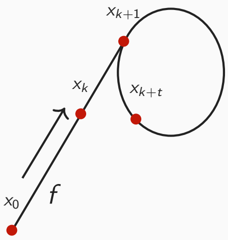
$$x = \frac{a - a'}{b' - b} \pmod{N}.$$

Collision search

Collision

Given a random map $f : S \to S$ on a finite set S of cardinality N, we call collision any pair R, R' of elements in S such that $f(R) = f(R')$.

Pollard's Rho method



- Ideally, f is a random mapping.
- Expected number of steps until the collision is found:

$$\sqrt{\frac{\pi N}{2}}.$$

Collision search

$$f(R) = \begin{cases} R + P & \text{if } R \in S_1 \\ 2R & \text{if } R \in S_2 \\ R + Q & \text{if } R \in S_3, \end{cases}$$

Property of f

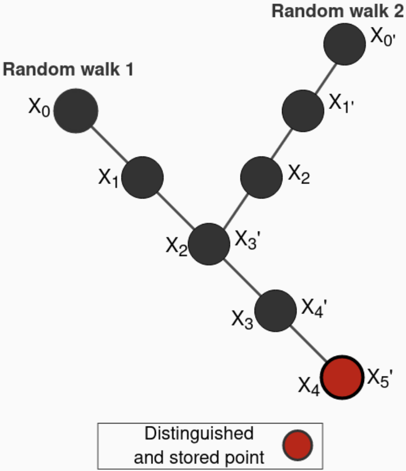
Input $(aP + bQ) \rightarrow$ Output $(a'P + b'Q)$.

(If the input of f is linear combination of P and Q, the output of f is also a linear combination of P and Q.)

Intuitively:

- Start from $R = aP + bQ$ for some random a and b
- *Walk* the random walk until we find the same point twice
  $\hookrightarrow$ To discover the collision, we need to store *all* * the points that we compute.

Parallel Collision Search



- Proposed by van Oorschot & Wiener (1996).
- Distinguished points : a set of points having an easily testable property.
  ex. The x-coordinate has 3 trailling zero bits: 10101101000.
- Only distinguished points are stored in memory.
- θ - the proportion of distinguished points in a set S.
- Complexity ? How many points do we expect to compute (store) before a collision is found ?
  $\hookrightarrow$ The Birthday Paradox

The Birthday Paradox (recall)

The birthday problem in collision search algorithms

We draw, randomly, elements from a set of size N.

How many times do we expect to draw an element before we get the same element twice?

$\hookrightarrow$ About a square root of the total number of elements.

Complexity of the Parallel Collision Search

The expected number of distinguished points calculated before a collision is found

$$E(X) = \sqrt{\frac{\pi N}{2}}$$

Time complexity (for L threads)

$$\mathcal{O}(\frac{1}{L}\sqrt{\frac{\pi N}{2}})$$

Memory complexity

$$\mathcal{O}(\theta\sqrt{\frac{\pi N}{2}})$$

Practical considerations

- Achieving perfect parallelization
- Shared memory VS Client-server setting
- Storage and lookup

Index Calculus

Index calculus attack

- Originally, a method for computing discrete logarithms in the multiplicative group of a finite field.
- Core ideas can be traced back to computation methods for discrete logs from the 19th century.
- Subexponential in $(\mathbb{Z}/p\mathbb{Z})^*$
- Core observations:
  - Any natural number can be factored into prime numbers.
  - As with the ordinary logarithm, there is a link between the multiplication of natural numbers and the addition of discrete logarithms

$$\log(q_1 \cdot \dots \cdot q_n) = \log(q_1) + \dots + \log(q_n) \pmod{p-1}$$

Index calculus attack (a toy example)

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \bmod p$

$$2^1 \equiv 2 \pmod{47}$$
$$2^7 \equiv 34 = 2 \cdot 17 \pmod{47}$$
$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$
$$2^{12} \equiv 7 \pmod{47}$$
$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

Index calculus attack (a toy example)

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$$2^1 \equiv 2 \pmod{47}$$

$$2^7 \equiv 34 = 2 \cdot 17 \pmod{47}$$

$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$

$$2^{12} \equiv 7 \pmod{47}$$

$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

Index calculus attack (a toy example)

$\boxed{28 = 2^x \pmod{47}, x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$$2^1 \equiv 2 \pmod{47}$$

$$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$$

$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$

$$2^{12} \equiv 7 \pmod{47}$$

$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

Index calculus attack (a toy example)

$\boxed{28 = 2^x \pmod{47}, x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \bmod p$

$$2^1 \equiv 2 \pmod{47}$$

$$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$$

$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$

$$2^{12} \equiv 7 \pmod{47}$$

$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

$\boxed{28 = 2^x (\bmod\ 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$$2^1 \equiv 2 \pmod{47}$$

$$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$$

$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$

$$2^{12} \equiv 7 \pmod{47}$$

$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

Index calculus attack (a toy example)

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$2^1 \equiv 2 \pmod{47}$

$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$

$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$

$2^{12} \equiv 7 \pmod{47}$

$2^{18} \equiv 25 = 5^2 \pmod{47}$

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$2^1 \equiv 2 \pmod{47}$

$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$

$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$

$2^{12} \equiv 7 \pmod{47}$

$2^{18} \equiv 25 = 5^2 \pmod{47}$

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \mod p$

$2^1 \equiv 2 \pmod{47}$

$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$

$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$

$2^{12} \equiv 7 \pmod{47}$

$2^{18} \equiv 25 = 5^2 \pmod{47}$

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

Index calculus attack (a toy example)

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Relation search phase

Find relations of the form $\prod_{j=1}^{4} R_j^{r_j} \equiv 2^r \bmod p$

$$2^1 \equiv 2 \pmod{47}$$

$$\cancel{2^7 \equiv 34 = 2 \cdot 17 \pmod{47}}$$

$$2^8 \equiv 21 = 3 \cdot 7 \pmod{47}$$

$$2^{12} \equiv 7 \pmod{47}$$

$$2^{18} \equiv 25 = 5^2 \pmod{47}$$

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

Index calculus attack (a toy example)

$\boxed{28 = 2^x \pmod{47}, x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Linear algebra phase

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

$\boxed{28 = 2^x (\text{mod } 47), x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

Linear algebra phase

2	3	5	7		
1	0	0	0	1	
0	1	0	1	8	$L_2 = L_2 - L_3$
0	0	0	1	12	
0	0	2	0	18	$L_4 = L_4 / 2$

Index calculus attack (a toy example)

$\boxed{28 = 2^x \pmod{47}, x = ?}$

Let $\mathcal{F} = \{R_1, R_2, R_3, R_4\} = \{2, 3, 5, 7\}$ be a factor base

2	3	5	7	
1	0	0	0	1
0	1	0	1	8
0	0	0	1	12
0	0	2	0	18

$L_2 = L_2 - L_3$

$L_4 = L_4 / 2$

2	3	5	7	
1	0	0	0	1
0	1	0	0	-4=42
0	0	0	1	12
0	0	1	0	9

Infer: $\log_2 2 = 1, \log_2 3 = 42, \log_2 5 = 9, \log_2 7 = 12$
$\log_2 28 = \log_2(2^2 \cdot 7) = 2\log_2 2 + \log_2 7 = 14$

Index calculus attack

Algorithm summary

Input: a finite cyclic group $(G, +)$ and two elements g, $h \in G$

Output: $x \in \mathbb{Z}$ such that $h = x \cdot g$

1. Finding an appropriate *factor base* $\mathcal{B} = \{g_1, ..., g_k\}$, such that $\mathcal{B} \subseteq G$
2. Relation search phase : find relations of the form

$$[a_i]g \; + \; [b_i]h \; = \; \sum_{j=1}^{n} [c_{ij}]g_j$$

for random integers a_i, b_i.

3. Linear algebra phase : having matrices $A = (a_i b_i)$ and $M = (c_{ij})$, find a kernel vector $v = (v_1 ... v_k)$ of the matrix M. Compute solution :

$$x = -(\sum_i a_i v_i)/(\sum_i b_i v_i)$$

Index calculus on elliptic curves

- $\log(P_1 \cdot \ldots \cdot P_n) = \log(P_1) + \cdots + \log(P_n)$ ✓
- "Prime" points **?**
- Point decomposition **?**

$\hookrightarrow$ The index calculus attack can be applied for elliptic curves over extension fields.

Index calculus on binary elliptic curves

Let $\mathbb{F}_{2^n}$ be a finite field and E be an elliptic curve defined by

$$\boxed{E : y^2 + xy = x^3 + ax^2 + b}$$

with $a, b \in \mathbb{F}_{2^n}$.

Point decomposition phase of the Index calculus algorithm

Find $P_1, \ldots, P_{m-1} \in E(\mathbb{F}_{2^n})$, such that

$$P_m = P_1 + \ldots + P_{m-1}$$

Point decomposition

Semaev's summation polynomials (2004)

*In the case of characteristic 2 and 3

$$S_2(X_1, X_2) = X_1 + X_2,$$
$$S_3(X_1, X_2, X_3) = X_1^2X_2^2 + X_1^2X_3^2 + X_1X_2X_3 + X_2^2X_3^2 + b,$$

For $m \geq 4$

$$S_m(X_1, \ldots, X_m) =$$
$$Res_X(S_{m-k}(X_1, \ldots, X_{m-k-1}, X), S_{k+2}(X_{m-k}, \ldots, X_m, X))$$

For $P_1, \ldots, P_m \in E(\mathbb{F}_{2^n})$

$$P_1 + \ldots + P_m = \mathcal{O} \iff S_m(\mathbf{x}_{P_1}, \ldots, \mathbf{x}_{P_m}) = 0$$

Weil descent

Gaudry and Diem (2008 and 2009)

Rewrite the equation $S_m(X_1, \ldots, X_m) = 0$ as a system of n equations over $\mathbb{F}_2$.

Example (trivial case of $m = 2$):

$$S_2(X_1, X_2) = 0$$

$$X_1 + X_2 = 0$$

$$(a_{1,0} + a_{1,1}t + \ldots + a_{1,n-1}t^{n-1}) + (a_{2,0} + a_{2,1}t + \ldots + a_{2,n-1}t^{n-1}) = 0$$

$$(a_{1,0} + a_{2,0}) + (a_{1,1} + a_{2,1})t + \ldots + (a_{1,n-1} + a_{2,n-1})t^{n-1} = 0$$

$$\begin{cases} a_{1,0} + a_{2,0} = 0 \\ a_{1,1} + a_{2,1} = 0 \\ \ldots \\ a_{1,n-1} + a_{2,n-1} = 0 \end{cases}$$

The system is commonly solved using Gröbner basis methods.

Gaudry and Diem (2008 and 2009)

Rewrite S_m in terms of the elementary symmetric polynomials

$$
\begin{aligned}
\mathbf{e}_1 &= \sum_{1 \leq i_1 \leq m} X_{i_1}, \\
\mathbf{e}_2 &= \sum_{1 \leq i_1, i_2 \leq m} X_{i_1} X_{i_2}, \\
&\dots \\
\mathbf{e}_m &= \prod_{1 \leq i \leq m} X_i .
\end{aligned}
$$

PDP algebraic model

Choice of a factor base : an l-dimensional vector subspace V of $\mathbb{F}_{2^n} / \mathbb{F}_2$. When $l \sim \frac{n}{m}$ the system has a reasonable chance to have a solution.

$$X_1 = a_{1,0} + \ldots + a_{1,l-1} t^{l-1}$$
$$X_2 = a_{2,0} + \ldots + a_{2,l-1} t^{l-1}$$
$$\ldots$$
$$X_m = a_{m,0} + \ldots + a_{m,l-1} t^{l-1}$$

$$\mathbf{e}_1 = e_{1,0} + \ldots + e_{1,l-1} t^{l-1}$$
$$\mathbf{e}_2 = e_{2,0} + \ldots + e_{2,2l-2} t^{2l-2}$$
$$\ldots$$
$$\mathbf{e}_m = e_{m,0} + \ldots + e_{m,m(l-1)} t^{m(l-1)}$$

PDP algebraic model

- Equations defining symmetric polynomials

$$
\begin{aligned}
&e_{1,0} = a_{1,0} + \ldots + a_{m,0} \\
&e_{1,1} = a_{1,1} + \ldots + a_{m,1} \\
&\ldots \\
&e_{m,m(l-1)} = a_{1,l} \cdot \ldots \cdot a_{m,l}.
\end{aligned}
$$

- Equations derived from the Weil descent

The system is commonly solved using Gröbner basis methods.

Summary

Today:

- Elliptic curves as finite groups
- Parallel Collision Search
- The Index Calculus attack