



Cryptanalysis in elliptic curve cryptography

Cryptology – Autumn 2022

Monika Trimoska

December 6, 2022

Institute for Computing and Information Sciences
Radboud University

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

$$E : y^2 = x^3 + Ax + B$$

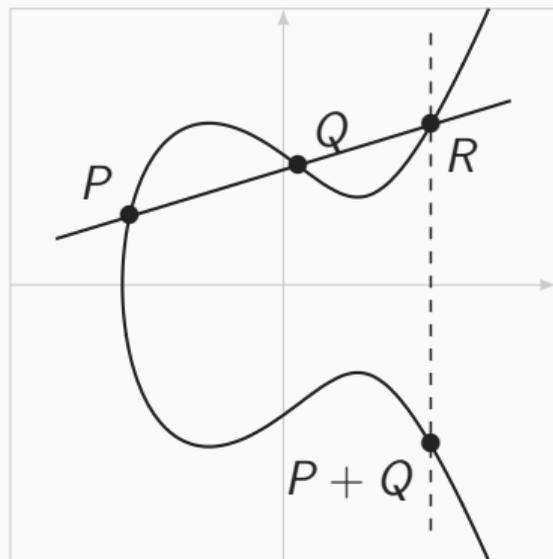
(short Weierstrass form)

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 curves¹

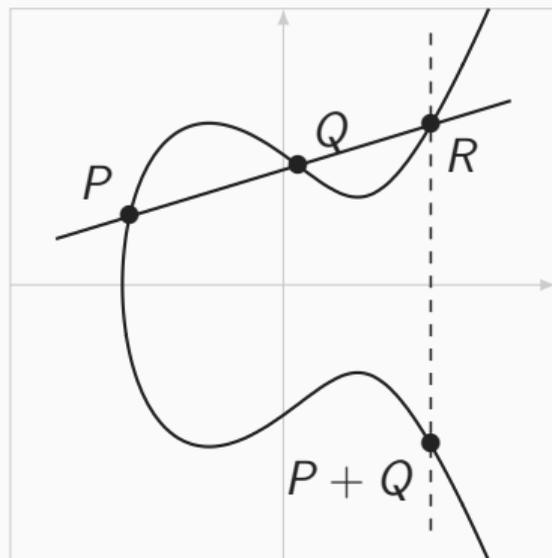
Adding points on an elliptic curve



Addition $P + Q$

¹Figures from the TikZ for Cryptographers library

Adding points on an elliptic curve



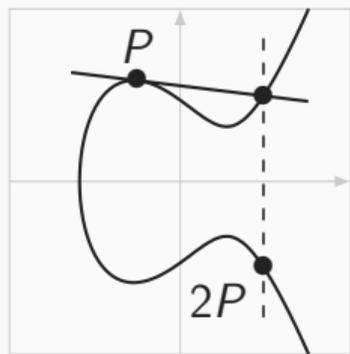
Addition $P + Q$

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

¹Figures from the TikZ for Cryptographers library

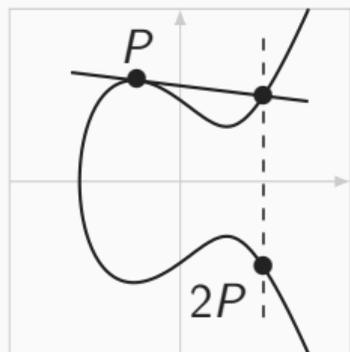
The geometry of elliptic curves

Adding points on an elliptic curve



Doubling $P + P$

Adding points on an elliptic curve

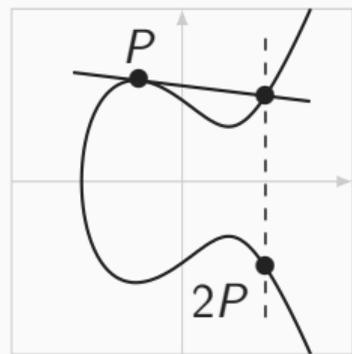


Doubling $P + P$

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

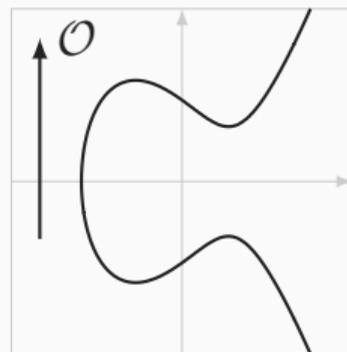
The geometry of elliptic curves

Adding points on an elliptic curve

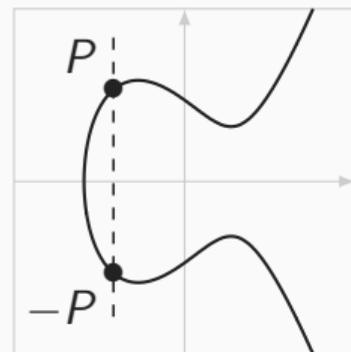


Doubling $P + P$

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



Neutral element \mathcal{O}



Inverse element $-P$

The 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 $P, Q \in E$.

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

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

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

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

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 .

↪

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

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

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 $xP = Q$

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**.

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

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**.

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

- First write $m = m_0 + m_1 \cdot 2 + m_2 \cdot 2^2 + \dots + m_r \cdot 2^r$
- Then mP can be computed as $mP = m_0P + m_1 \cdot 2P + m_2 \cdot 2^2P + \dots + m_r \cdot 2^rP$
- Requires r doublings (and sums)

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

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

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

↪ 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

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

$$(a - a')P = (b' - b)Q,$$

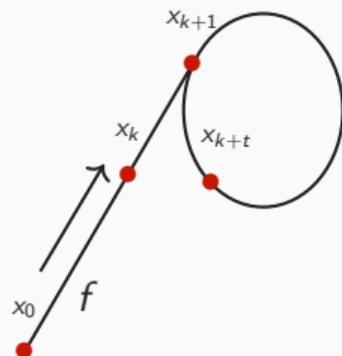
and we compute

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

Collision

Given a random map $f : S \rightarrow S$ 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:

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

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

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 .)

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

Property of f

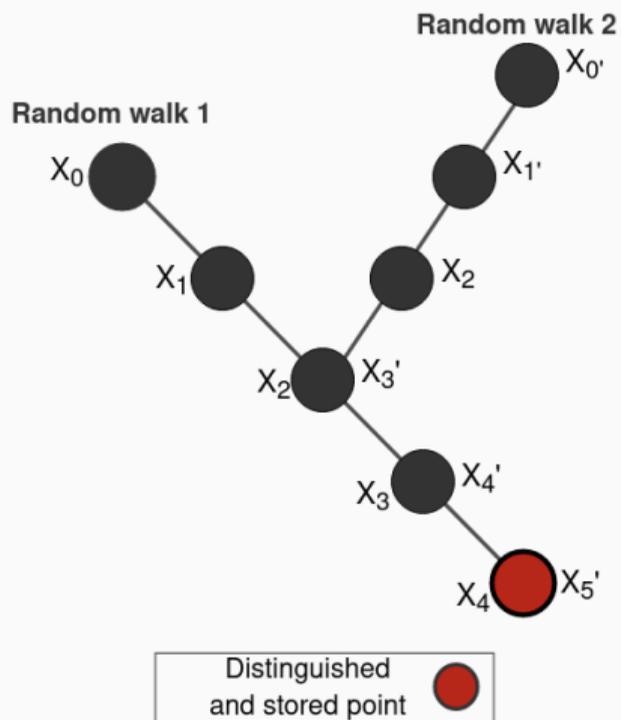
Input $(aP + bQ) \rightarrow$ Output $(a'P + b'Q)$.

(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:

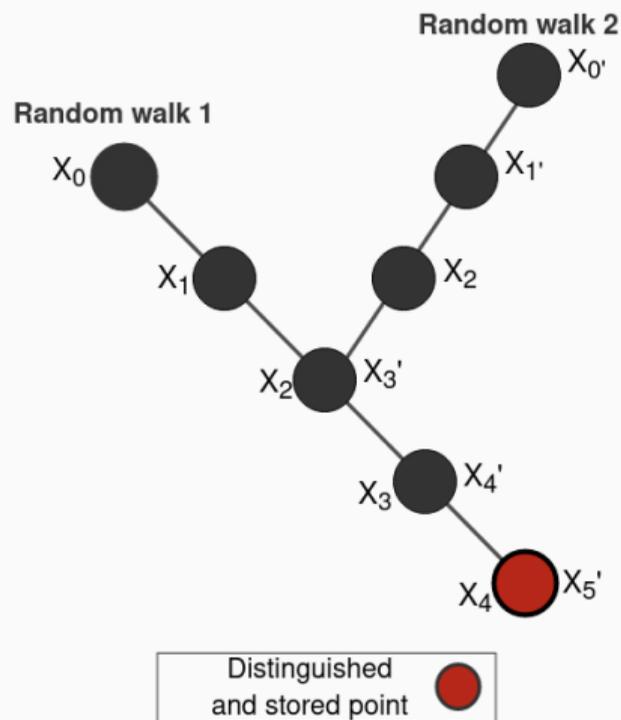
- Start from $R = aP + bQ$ for some random a and b
- *Walk* the random walk until we find the same point twice
 - \hookrightarrow To discover the collision, we need to store *all* * the points that we compute.

Parallel Collision Search



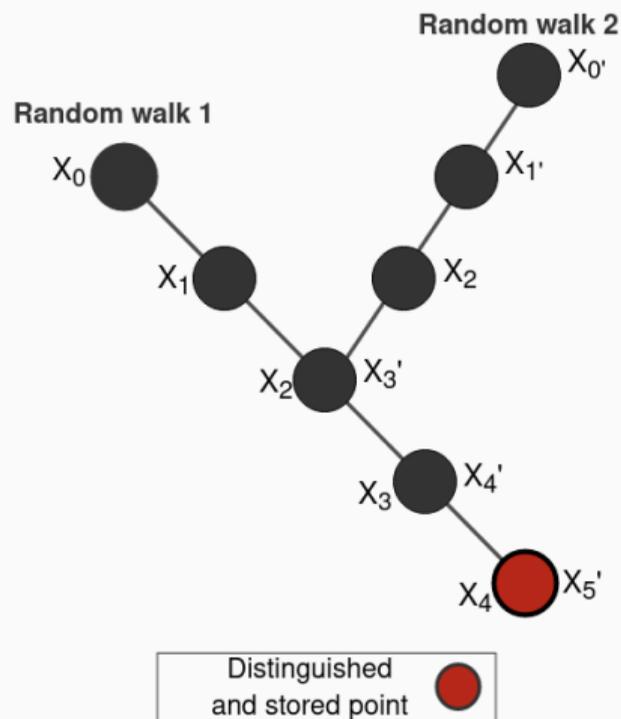
- Proposed by van Oorschot & Wiener (1996).

Parallel Collision Search



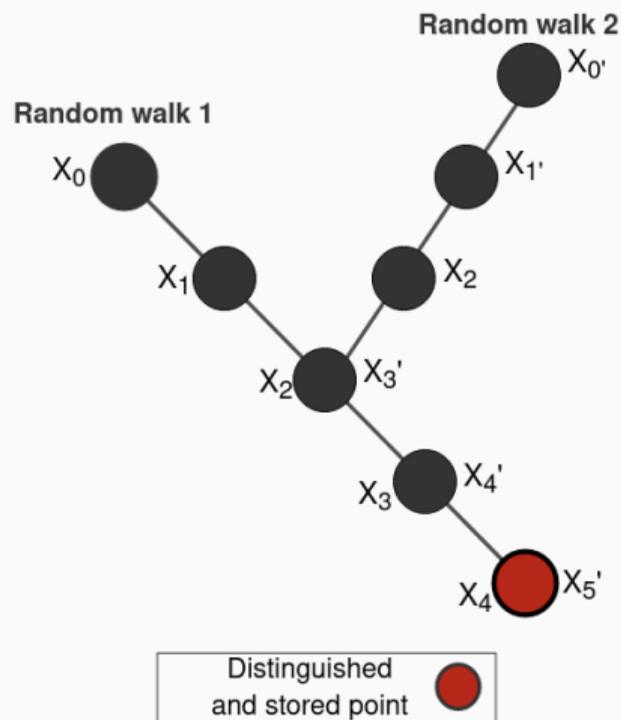
- Proposed by van Oorschot & Wiener (1996).
- **Distinguished points** : a set of points having an easily testable property.
ex. The x -coordinate has 3 trailing zero bits:
10101101**000**.
- Only distinguished points are stored in memory.

Parallel Collision Search



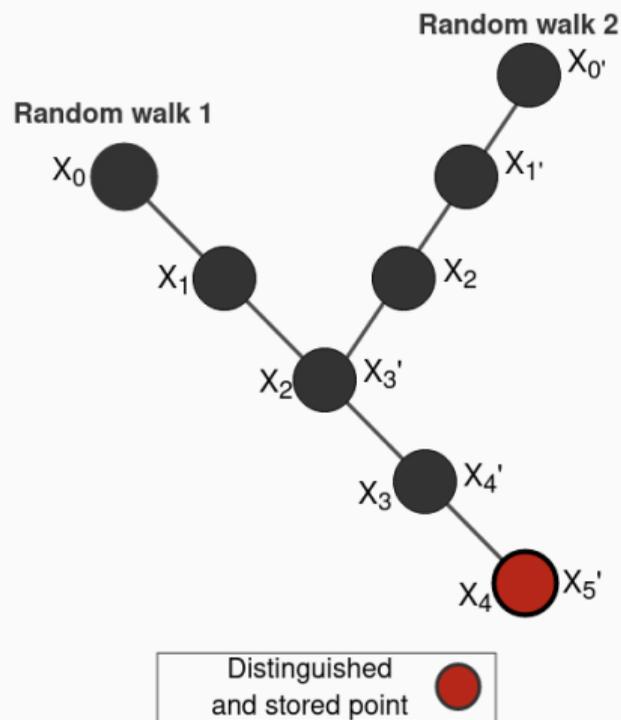
- Proposed by van Oorschot & Wiener (1996).
- **Distinguished points** : a set of points having an easily testable property.
ex. The x -coordinate has 3 trailing zero bits:
10101101**000**.
- Only distinguished points are stored in memory.
- θ - the **proportion** of distinguished points in a set S .

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 trailing zero bits:
10101101**000**.
- 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 ?

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 trailing zero bits:
10101101**000**.
- 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 ?
↪ The Birthday Paradox

The Birthday Paradox

The Birthday Paradox

What is the probability that, in a set of N randomly chosen people, at least two will share a birthday?

The Birthday Paradox

What is the probability that, in a set of N randomly chosen people, at least two will share a birthday?

How big should N be to get a probability of 50% ?

The Birthday Paradox

What is the probability that, in a set of N randomly chosen people, at least two will share a birthday?

How big should N be to get a probability of 50% ?

↪ $N = 23$

The Birthday Paradox

What is the probability that, in a set of N randomly chosen people, at least two will share a birthday?

How big should N be to get a probability of 50% ?

↪ $N = 23$

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?

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?

- Probability that there is no collision when the first element is drawn : **1**

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?

- Probability that there is no collision when the first element is drawn : **1**
- Probability that there is no collision when the second element is drawn : **$1 - \frac{1}{N}$**

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?

- Probability that there is no collision when the first element is drawn : 1
- Probability that there is no collision when the second element is drawn : $1 - \frac{1}{N}$
- Probability that there is no collision when the third element is drawn : $1 - \frac{2}{N}$
- ...

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?

$$P(X > T) = \left(1 - \frac{1}{N}\right)\left(1 - \frac{2}{N}\right) \dots \left(1 - \frac{T-1}{N}\right)$$

Using a first-order Taylor approximation $e^x \approx 1 + x$, this simplifies to

$$\begin{aligned} P(X > T) &\approx e^{-\frac{1}{N}} \cdot e^{-\frac{2}{N}} \dots e^{-\frac{T-1}{N}} \approx \\ &\approx e^{-(1+2+\dots+(T-1))/N} \approx \\ &\approx e^{-\frac{T(T-1)}{2N}}. \end{aligned}$$

↪ For $P(X > T) \approx 63\%$, we get $T \approx 1.41\sqrt{N}$

Complexity of the Parallel Collision Search

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

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

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)

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

Memory complexity

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

Index Calculus

- Originally, a method for computing discrete logarithms in the multiplicative group of a finite field.

- 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.

- 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 $(\mathbb{Z}/p\mathbb{Z})^*$

- 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)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

Index calculus attack (a toy example)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

Index calculus attack (a toy example)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

Index calculus attack (a toy example)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

Index calculus attack (a toy example)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

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)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

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)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

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)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

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)

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

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 $\prod_{j=1}^4 R_j^{r_j} \equiv 2^r \pmod{p}$

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

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

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

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

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

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)

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

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

Index calculus attack (a toy example)

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

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	$L_2 = L_2 - L_3$
0	0	0	1	12	
0	0	2	0	18	$L_4 = L_4 / 2$

Index calculus attack (a toy example)

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

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

2	3	5	7			2	3	5	7		
1	0	0	0	1		1	0	0	0	1	
0	1	0	1	8	$L_2 = L_2 - L_3$	0	1	0	0	-4=42	
0	0	0	1	12		0	0	0	1	12	
0	0	2	0	18	$L_4 = L_4 / 2$	0	0	1	0	9	

Index calculus attack (a toy example)

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

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

2	3	5	7			2	3	5	7		
1	0	0	0	1		1	0	0	0	1	
0	1	0	1	8	$L_2 = L_2 - L_3$	0	1	0	0	-4=42	
0	0	0	1	12		0	0	0	1	12	
0	0	2	0	18	$L_4 = L_4 / 2$	0	0	1	0	9	

Infer: $\log_2 2 = 1, \log_2 3 = 42, \log_2 5 = 9, \log_2 7 = 12$

$$\log_2 28 = \log_2(2^2 \cdot 7) = 2\log_2 2 + \log_2 7 = 14$$

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* $\mathcal{B} = \{g_1, \dots, g_k\}$, 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 $v = (v_1 \dots v_k)$ of the matrix M . Compute solution :

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

- $\log(P_1 \cdot \dots \cdot P_n) = \log(P_1) + \dots + \log(P_n)$

- $\log(P_1 \cdot \dots \cdot P_n) = \log(P_1) + \dots + \log(P_n)$ ✓
- "Prime" points

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

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

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

↪ The index calculus attack can be applied for elliptic curves over extension fields.

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

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

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

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

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

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

Point decomposition phase of the Index calculus algorithm

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

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

Semaev's summation polynomials (2004)

*In the case of characteristic 2 and 3

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

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

For $m \geq 4$

$$S_m(X_1, \dots, X_m) =$$

$$\text{Res}_X(S_{m-k}(X_1, \dots, X_{m-k-1}, X), S_{k+2}(X_{m-k}, \dots, X_m, X))$$

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

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

Gaudry and Diem (2008 and 2009)

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

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

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

$$X_1 + X_2 = 0$$

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

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

Gaudry and Diem (2008 and 2009)

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

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

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

$$X_1 + X_2 = 0$$

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

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

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

The system is commonly solved using **Gröbner basis** methods.

Today:

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

Today:

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

Next time:

- Isogeny-based cryptography