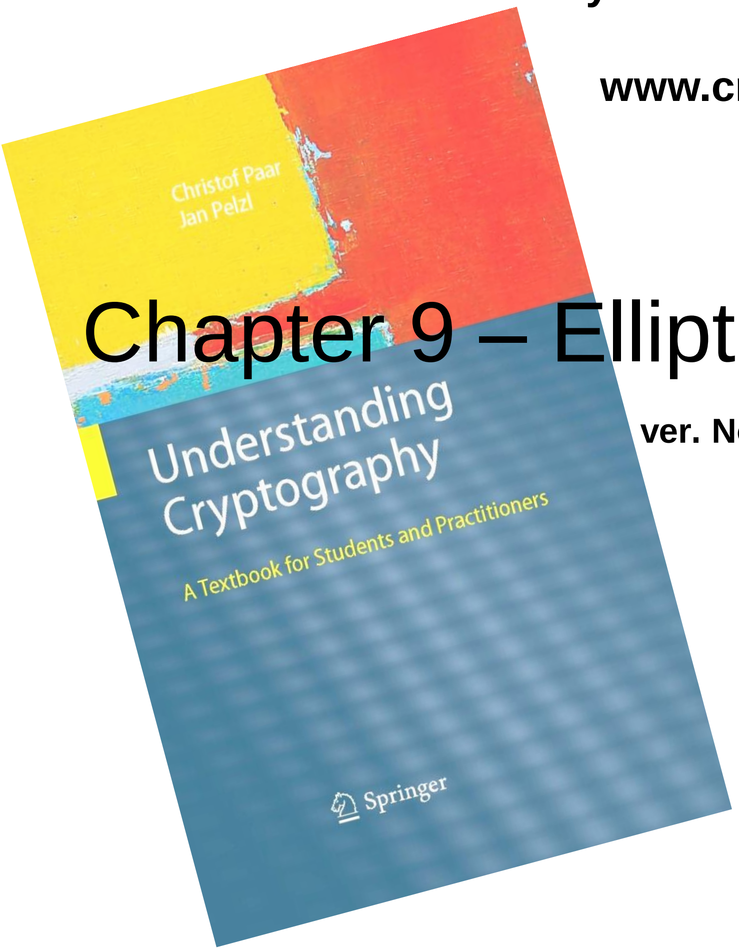
# Understanding Cryptography

**by Christof Paar and Jan Pelzl**

**www.crypto-textbook.com**




# Chapter 9 – Elliptic Curve Cryptography

**ver. November 10, 2024**

**These slides were originally prepared by Tim Güneysu, Christof Paar and Jan Pelzl. Later, they were modified by Tomas Fabsic for purposes of teaching I-ZKRY at FEI STU.**

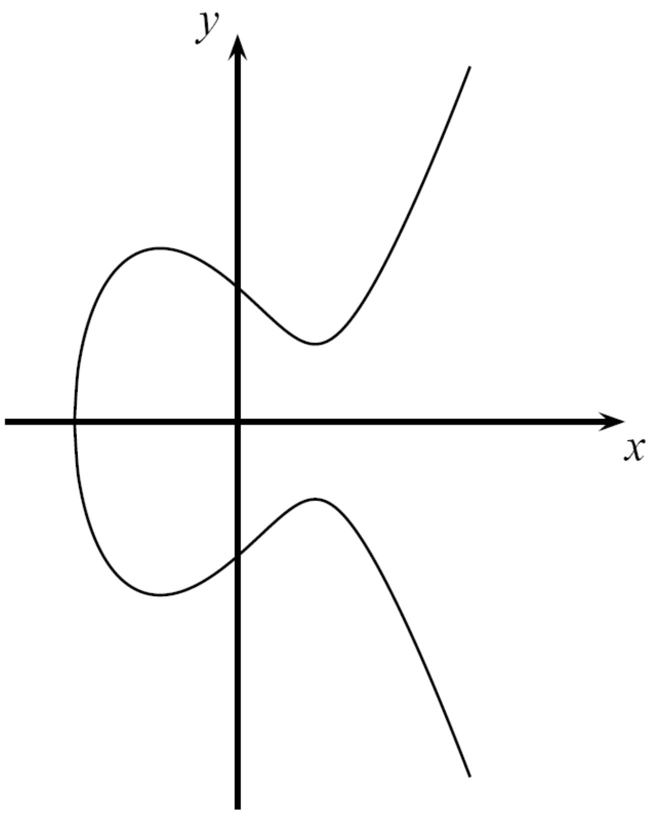
# Homework

- Read Chapter 9.
- Solve problems from the exercise set no. 8 and submit them to AIS by **18.11.2024 23:59**.

## Some legal stuff (sorry): Terms of Use

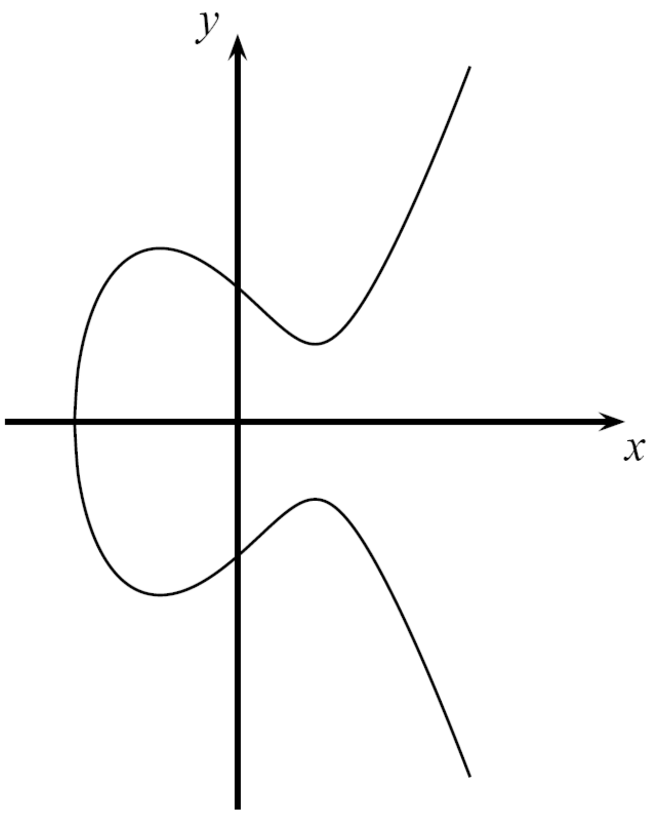
## Content of this Chapter

- Introduction
- Computations on Elliptic Curves
- The Elliptic Curve Diffie-Hellman Protocol
- Security Aspects



# Content of this Chapter

- **Introduction**
- Computations on Elliptic Curves
- The Elliptic Curve Diffie-Hellman Protocol
- Security Aspects

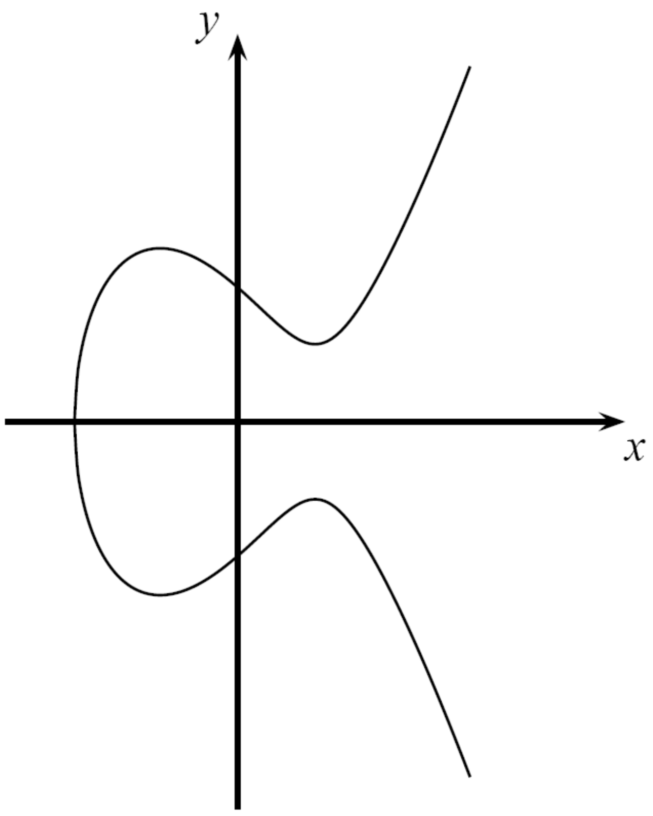


## Motivation

- **Problem:**
  Asymmetric schemes like RSA and Elgamal require exponentiations in integer rings and fields with parameters of more than 2000 bits.
  - High computational effort on CPUs with 32-bit or 64-bit arithmetic
  - Large parameter sizes critical for storage on small and embedded
- **Motivation:**
  Smaller field sizes providing equivalent security are desirable
- **Solution:**
  Elliptic Curve Cryptography uses a group of points (instead of integers) for cryptographic schemes with coefficient sizes of 256-512 bits, reducing significantly the computational effort.

## Content of this Chapter

- Introduction
- **Computations on Elliptic Curves**
- The Elliptic Curve Diffie-Hellman Protocol
- Security Aspects



# Elliptic Curves over R

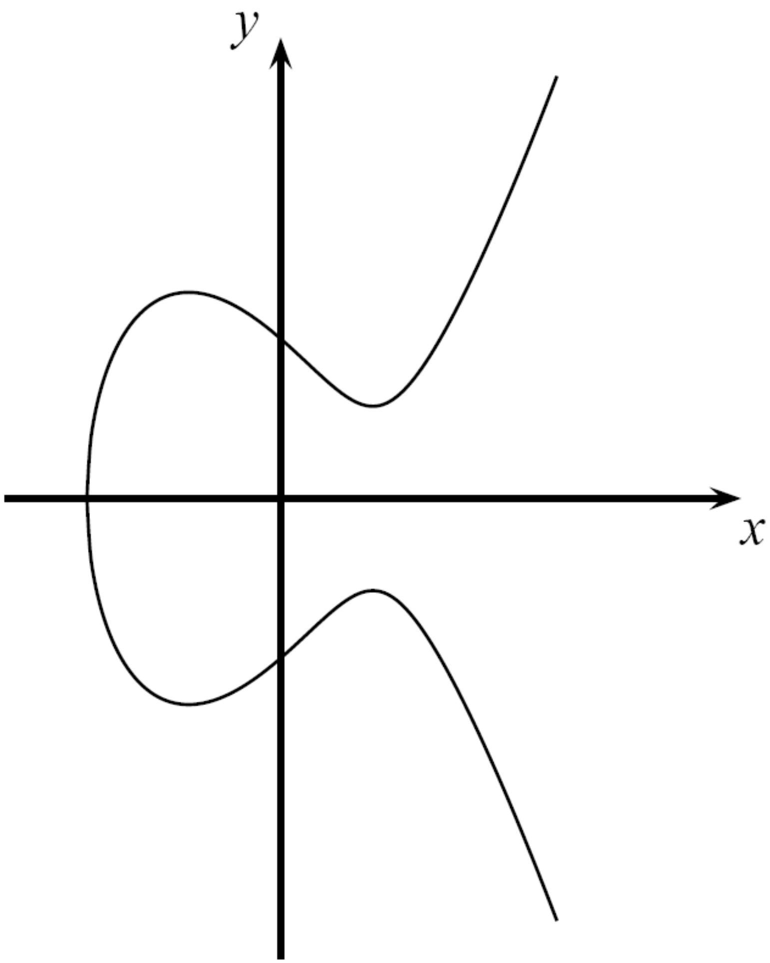
- Elliptic curves are polynomials that define points based on the (simplified) Weierstraß equation:

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

for parameters a,b that specify the exact shape of the curve

- On the real numbers and with parameters a, b $\in$ R, an elliptic curve looks like this →

- Elliptic curves can not just be defined over the real numbers *R* but over many other types of finite fields.




**Example**: $y^2 = x^3 - 3x + 3$ over $R$

# Elliptic Curves over Zp

- In cryptography, we are interested in elliptic curves module a prime *p*:

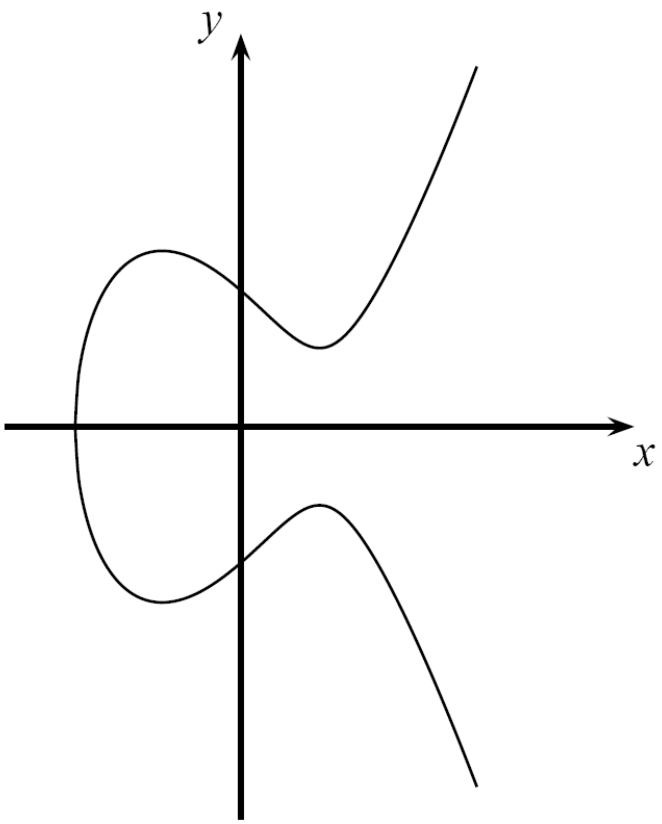
**_Definition: Elliptic Curves over prime fields_**

*The elliptic curve over $Z_p$, $p>3$ is the set of all pairs $(x,y) \in Z_p$ which fulfill*

$$y^2 = x^3 + ax + b \bmod p$$

*together with an imaginary point at infinity θ, where $a,b \in Z_p$ and the condition*

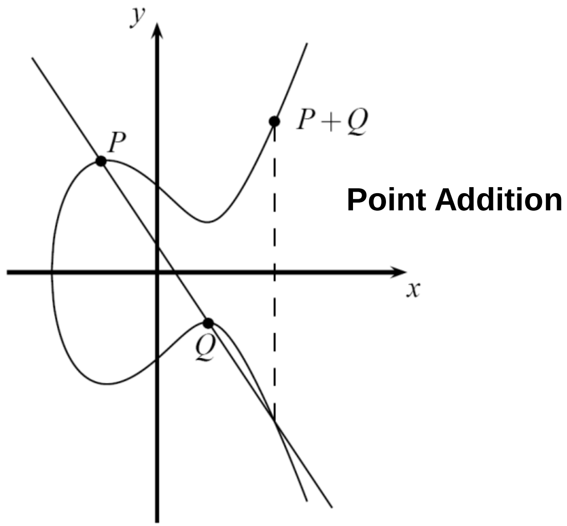
$$4a^3+27b^2 \neq 0 \bmod p.$$

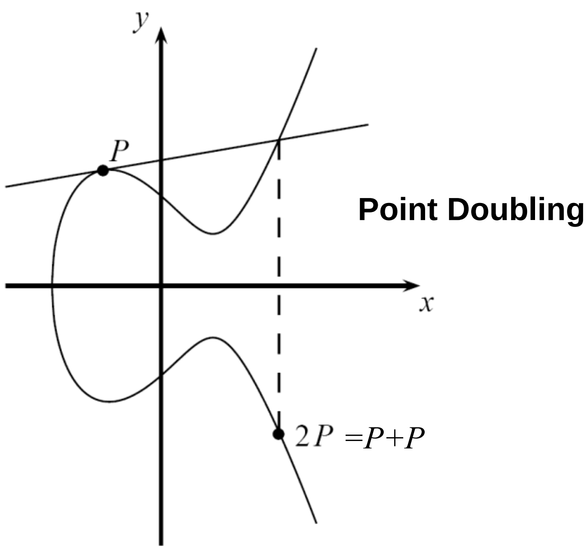



- Note that $Z_p = \{0,1,\ldots, p-1\}$ is a set of integers with modulo p arithmetic

# Computations on Elliptic Curves (ctd.)

- Generating a *group of points* on elliptic curves based on point addition operation *P+Q = R, i.e.,* $(x_P,y_P)+(x_Q,y_Q) = (x_R,y_R)$
- Geometric Interpretation of point addition operation
  - *Draw straight line through P and Q; if P=Q use tangent line instead*
  - *Mirror third intersection point of drawn line with the elliptic curve along the x-axis*
- Elliptic Curve Point Addition and Doubling Formulas




$x_3 = s^2 - x_1 - x_2 \bmod p$ and $y_3 = s(x_1 - x_3) - y_1 \bmod p$

*where*

$$s = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p & \text{; if } P \neq Q \text{ (point addition)} \\ \dfrac{3x_1^2 + a}{2y_1} \bmod p & \text{; if } P = Q \text{ (point doubling)} \end{cases}$$

# Computations on Elliptic Curves (ctd.)

- **Example**: Given $E: y^2 = x^3+2x+2 \bmod 17$ and point $P=(5,1)$

  **Goal:** Compute $2P = P+P = (5,1)+(5,1) = (x_3,y_3)$

$$s = \frac{3x_1^2 + a}{2y_1} = (2 \cdot 1)^{-1}(3 \cdot 5^2 + 2) = 2^{-1} \cdot 9 \equiv 9 \cdot 9 \equiv 13 \bmod 17$$

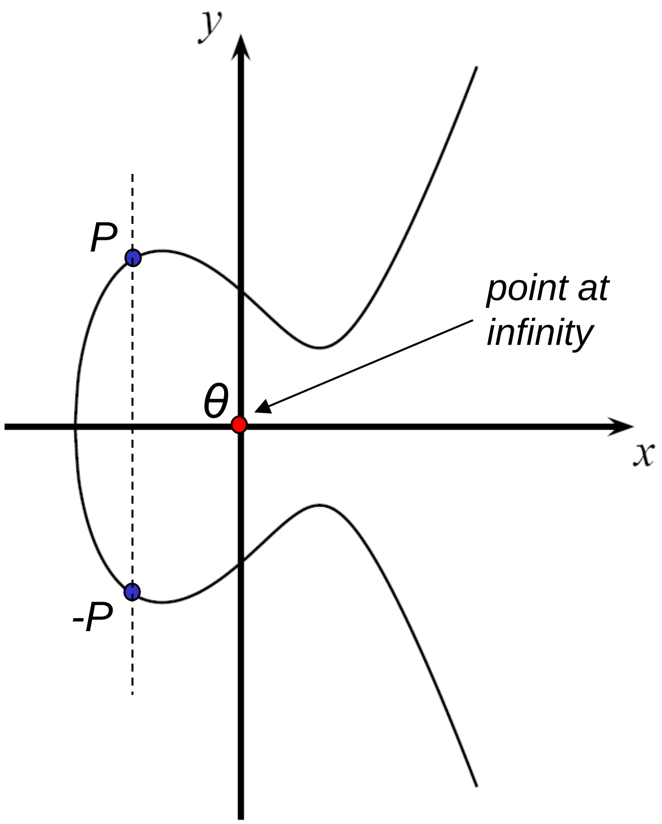
$$x_3 = s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 159 \equiv 6 \bmod 17$$

$$y_3 = s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 \equiv 3 \bmod 17$$

**Finally $2P = (5,1) + (5,1) = (6,3)$**

# Point at infinity and inverses

- Some special considerations are required to convert elliptic curves into a group of points
  - *In any group, a special element is required to allow for the identity operation, i.e., given* $P \in E$*:* $P + \theta = P = \theta + P$
  - *This identity point (which is not on the curve) is additionally added to the group definition*
  - *This identity point is called* ***point at infinity****, and is denoted by θ in this presentation (In the book, it is denoted by caligraphic O).*




- Elliptic Curve are symmetric along the *x*-axis
  - Up to two solutions *y and -y* exist for each quadratic residue *x* of the elliptic curve
  - For each point *P =(x,y),* the inverse or negative point is defined as *-P =(x,-y)*

## Elliptic curves as groups

**Theorem 9.2.1** *The points on an elliptic curve together with $\mathcal{O}$ form a group with cyclic subgroups. Under certain conditions* all *points on an elliptic curve form a cyclic group.*

*Note: The caligraphic O in the theorem represents the point at infinity.*

## Elliptic curves as groups: Example

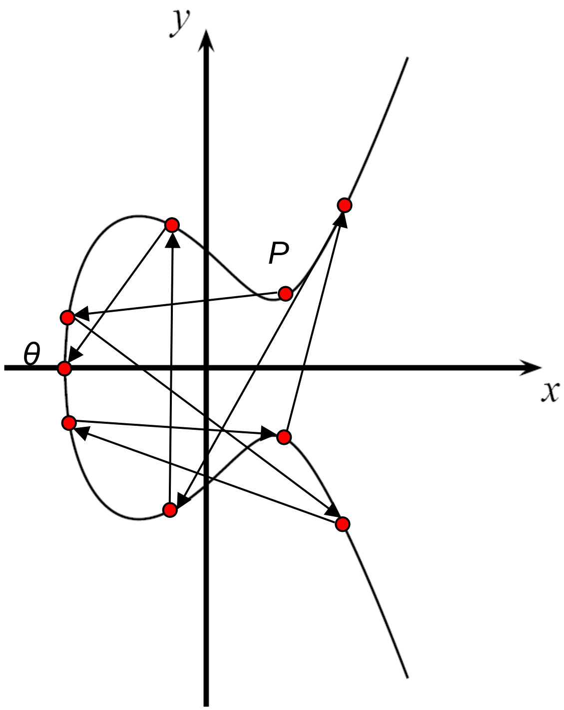
*Let E:* $y^2 = x^3+2x+2 \bmod 17$.

*Then E forms a cyclic group of order #E = |E| = 19.*

*Let P=(5,1).*

*Then P belongs to E, and:*

| | |
|---|---|
| $2P = (5,1)+(5,1) = (6,3)$ | $11P = (13,10)$ |
| $3P = 2P+P = (10,6)$ | $12P = (0,11)$ |
| $4P = (3,1)$ | $13P = (16,4)$ |
| $5P = (9,16)$ | $14P = (9,1)$ |
| $6P = (16,13)$ | $15P = (3,16)$ |
| $7P = (0,6)$ | $16P = (10,11)$ |
| $8P = (13,7)$ | $17P = (6,14)$ |
| $9P = (7,6)$ | $18P = (5,16)$ |
| $10P = (7,11)$ | $19P = \theta$ |



# Number of Points on an Elliptic Curve

- How many points can be on an arbitrary elliptic curve?
  - Consider previous example: $E: y^2 = x^3+2x+2 \bmod 17$ has 19 points
  - However, determining the point count on elliptic curves in general is hard
- But Hasse's theorem bounds the number of points to a restricted interval

> ***Definition: Hasse's Theorem:***
>
> *Given an elliptic curve module p, the number of points on the curve is denoted by #E and is bounded by*
>
> $$p+1-2\sqrt{p} \leq \#E \leq p+1+2\sqrt{p}$$

- **Interpretation:** The number of points is „close to" the prime p
- **Example:** To generate a curve with about $2^{256}$ points, a prime with a length of about 256 bits is required

# Elliptic Curve Discrete Logarithm Problem

- Cryptosystems rely on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP)

> ***Definition: Elliptic Curve Discrete Logarithm Problem (ECDLP)***
>
> *Given a primitive element P and another element T on an elliptic curve E. The ECDL problem is finding the integer d, where 1 ≤ d ≤ #E such that*
>
> $$\underbrace{P + P + \ldots + P}_{d \text{ times}} = dP = T.$$

- Cryptosystems are based on the idea that *d* is large and kept secret and attackers cannot compute it easily
- If *d* is known, an efficient method to compute the point multiplication *dP* is required to create a reasonable cryptosystem
  - By adapting Square-and-Multiply Method Elliptic Curves we obtain an efficient method for point multiplication on elliptic curves: **Double-and-Add Algorithm**

# Double-and-Add Algorithm for Point Multiplication

- **Double-and-Add Algorithm**

**Input**: Elliptic curve *E,* an elliptic curve point *P* and *a* scalar *d* with bits $d_i$

**Output**: $T = d\,P$

**Initialization**:

$T = P$

**Algorithm**:

FOR $i = t-1$ DOWNTO 0

  $T = T + T \bmod n$

  IF $d_i = 1$

    $T = T + P \bmod n$

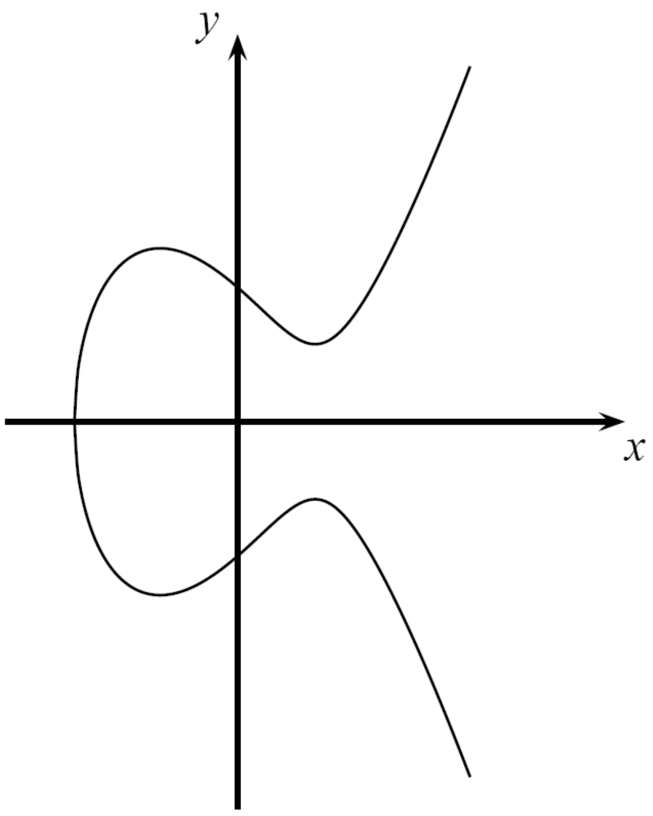
RETURN ($T$)

**Example**: $26P = (11010_2)P = (d_4 d_3 d_2 d_1 d_0)_2\, P$.

| Step | | |
|---|---|---|
| #0 | $P = \mathbf{1}_2 P$ | inital setting |
| #1*a* | $P+P = 2P = \mathbf{10}_2 P$ | DOUBLE (bit $d_3$) |
| #1*b* | $2P+P = 3P = 10^2\, P+1_2 P = \mathbf{11}_2 P$ | ADD (bit $d_3$=1) |
| #2*a* | $3P+3P = 6P = 2(11_2 P) = \mathbf{110_2} P$ | DOUBLE (bit $d_2$) |
| #2*b* | | no ADD ($d_2 = 0$) |
| #3*a* | $6P+6P = 12P = 2(110_2 P) = \mathbf{1100_2} P$ | DOUBLE (bit $d_1$) |
| #3*b* | $12P+P = 13P = 1100_2 P+1_2\, P = \mathbf{1101}_2 P$ | ADD (bit $d_1$=1) |
| #4*a* | $13P+13P = 26P = 2(1101_2 P) = \mathbf{11010_2} P$ | DOUBLE (bit $d_0$) |
| #4*b* | | no ADD ($d_0 = 0$) |

## Content of this Chapter

- Introduction
- Computations on Elliptic Curves
- **The Elliptic Curve Diffie-Hellman Protocol**
- Security Aspects



# The Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

- Given a prime *p*, a suitable elliptic curve *E* and a point $P=(x_P,y_P)$
- The Elliptic Curve Diffie-Hellman Key Exchange is defined by the following protocol:

***Alice***

Choose $k_{PrA} = a \in \{2, 3, \ldots, \#E-1\}$
Compute $k_{PubA} = A = aP = (x_A, y_A)$

***Bob***

Choose $k_{PrB} = b \in \{2, 3, \ldots, \#E-1\}$
Compute $k_{PubB} = B = bP = (x_B, y_B)$

A → (Alice to Bob)

B ← (Bob to Alice)

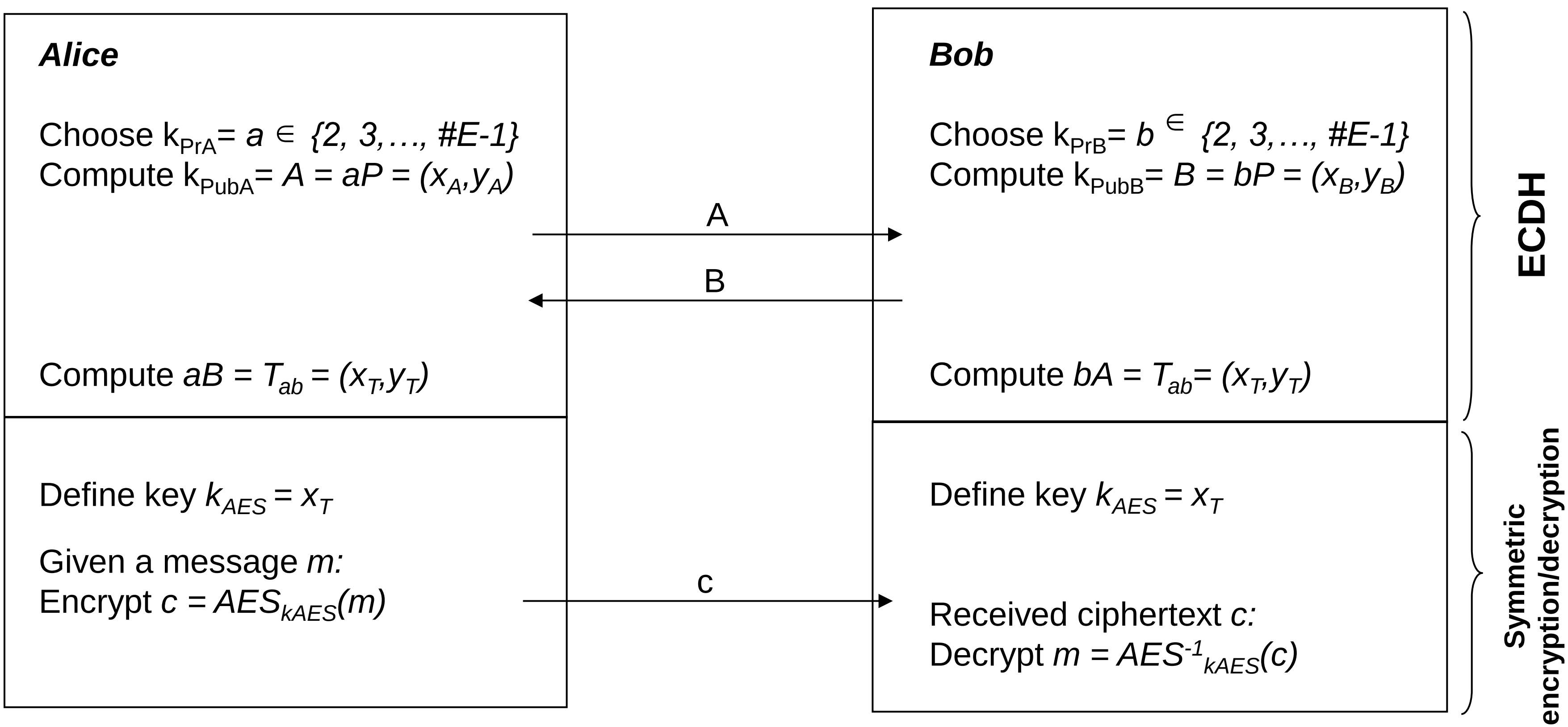
Alice: Compute $aB = T_{ab}$

Bob: Compute $bA = T_{ab}$

- Joint secret between Alice and Bob: $T_{AB} = (x_{AB}, y_{AB})$
- Proof for correctness:
  - Alice computes aB=a(bP)=abP
  - Bob computes bA=b(aP)=abP since group is associative
- One of the coordinates of the point $T_{AB}$ (usually the x-coordinate) can be used as session key (often after applying a hash function)

# The Elliptic Curve Diffie-Hellman Key Exchange (ECDH) (ctd.)

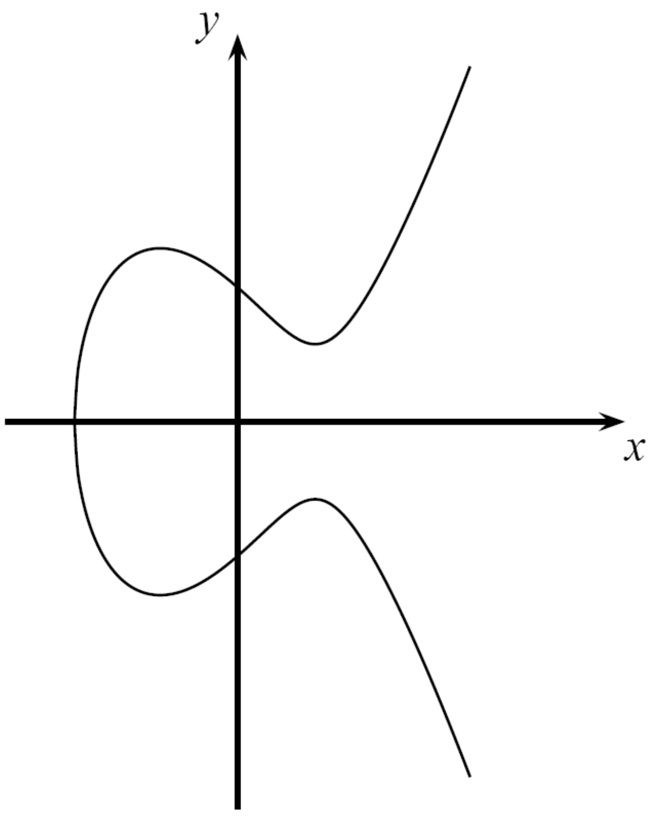
- The ECDH is often used to derive session keys for (symmetric) encryption
- One of the coordinates of the point $T_{AB}$ (usually the x-coordinate) is taken as session key




- In some cases, a hash function (see next chapters) is used to derive the session key

## Content of this Chapter

- Introduction
- Computations on Elliptic Curves
- The Elliptic Curve Diffie-Hellman Protocol
- **Security Aspects**



## Security Aspects

- Why are parameters signficantly smaller for elliptic curves (e.g. 256 bit) than for RSA (e.g. 3076 bit)?
  - Attacks on groups of elliptic curves are weaker than available factoring algorithms or integer DL attacks
  - Best known attacks on elliptic curves (chosen according to cryptographic criterions) are the Baby-Step Giant-Step and Pollard-Rho method
  - Complexity of these methods: on average, roughly $\sqrt{p}$ steps are required before the ECDLP can be successfully solved

- Implications to practical parameter sizes for elliptic curves:
  - An elliptic curve using a prime p with 256 bit (and roughly $2^{256}$ points) provides a security of $2^{128}$ steps that required by an attacker (on average)

## Lessons Learned

- Elliptic Curve Cryptography (ECC) is based on the discrete logarithm problem. It requires, for instance, arithmetic modulo a prime.
- ECC can be used for key exchange, for digital signatures and for encryption.
- ECC provides the same level of security as RSA or discrete logarithm systems over $Z_p$ with considerably shorter operands (e.g. 256 bit vs. 3072 bit), which results in shorter ciphertexts and signatures.
- In many cases ECC has performance advantages over other public-key algorithms.