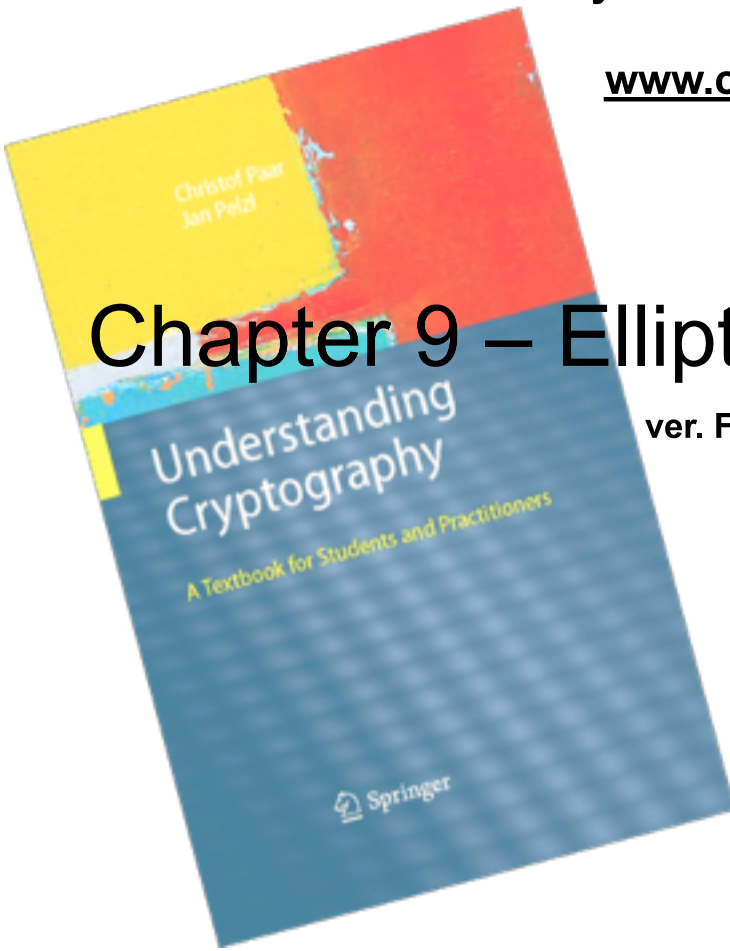
Understanding Cryptography

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



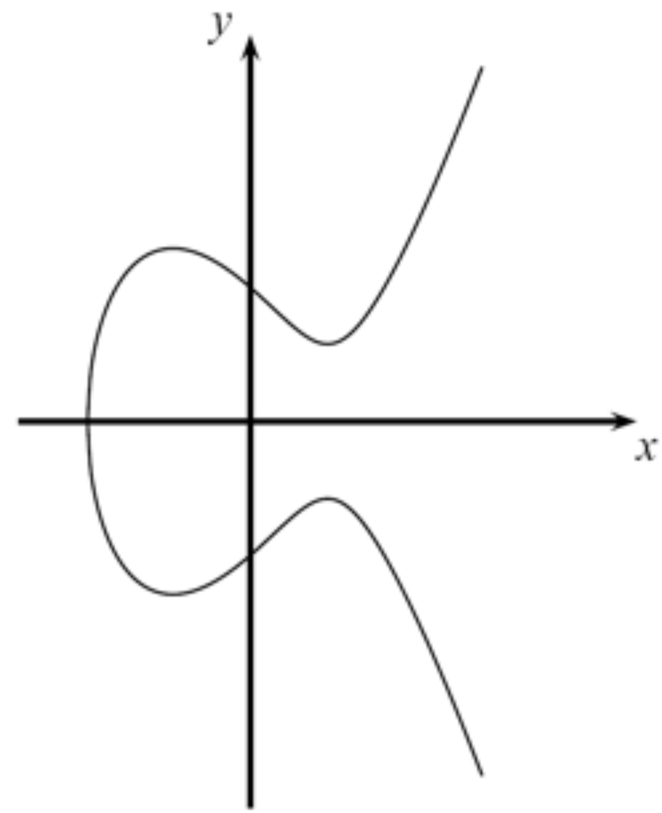
Chapter 9 – Elliptic Curve Cryptography

ver. February 2nd, 2015

These slides were prepared by Tim Güneysu, Christof Paar and Jan Pelzl
And modified by Sam Bowne

Some legal stuff (sorry): Terms of Use

Contents of this Chapter



Introduction

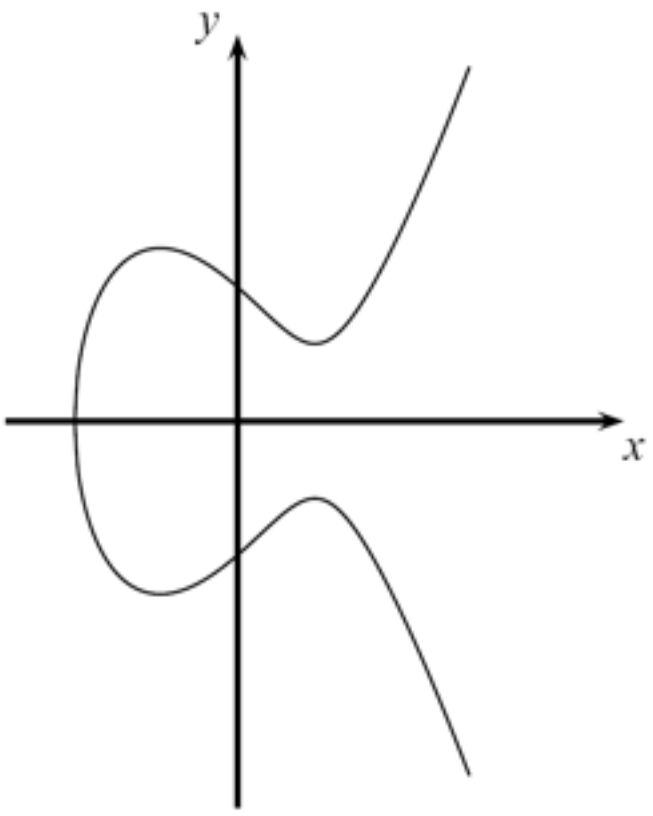
Elliptic Curve Cryptography (ECC)

- Around sinde the 1980s
- Same level of security as RSA with shorter keys
- ECC keys are 160-256 bits; RSA needs 1024-3072 bits
- ECC calculations are faster
- ECC uses less network bandwidth because signatures and keys are shorter

In this chapter, you will learn:

- The basic pros and cons of ECC vs. RSA and DL schemes.
- What an elliptic curve is and how to compute with it.
- How to build a DL problem with an elliptic curve.
- Protocols that can be realized with elliptic curves.
- Current security estimations of cryptosystems based on elliptic curves.

9.1 How to Compute with Elliptic Curves



Circle and Ellipse

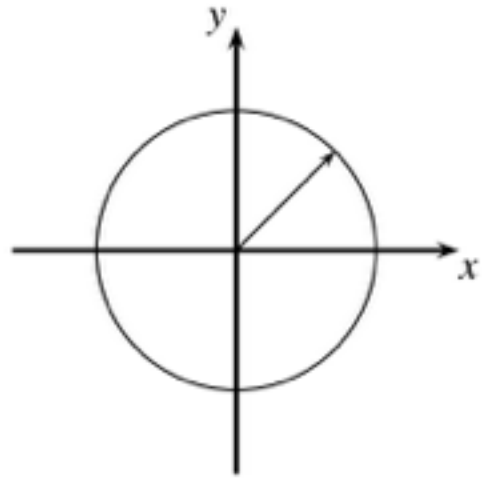


Fig. 9.1 Plot of all points (x, y) which fulfill the equation $x^2 + y^2 = r^2$ over $\mathbb{R}$

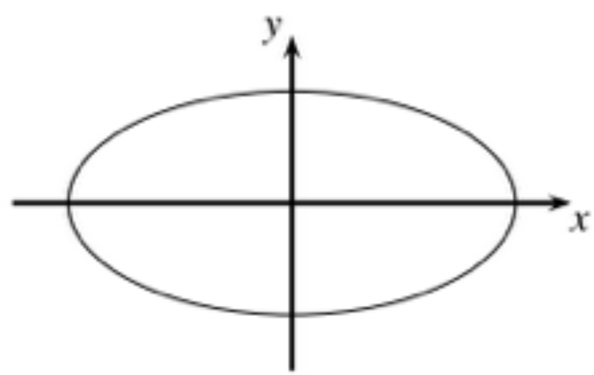


Fig. 9.2 Plot of all points (x, y) which fulfill the equation $a \cdot x^2 + b \cdot y^2 = c$ over $\mathbb{R}$

Computations on Elliptic Curves

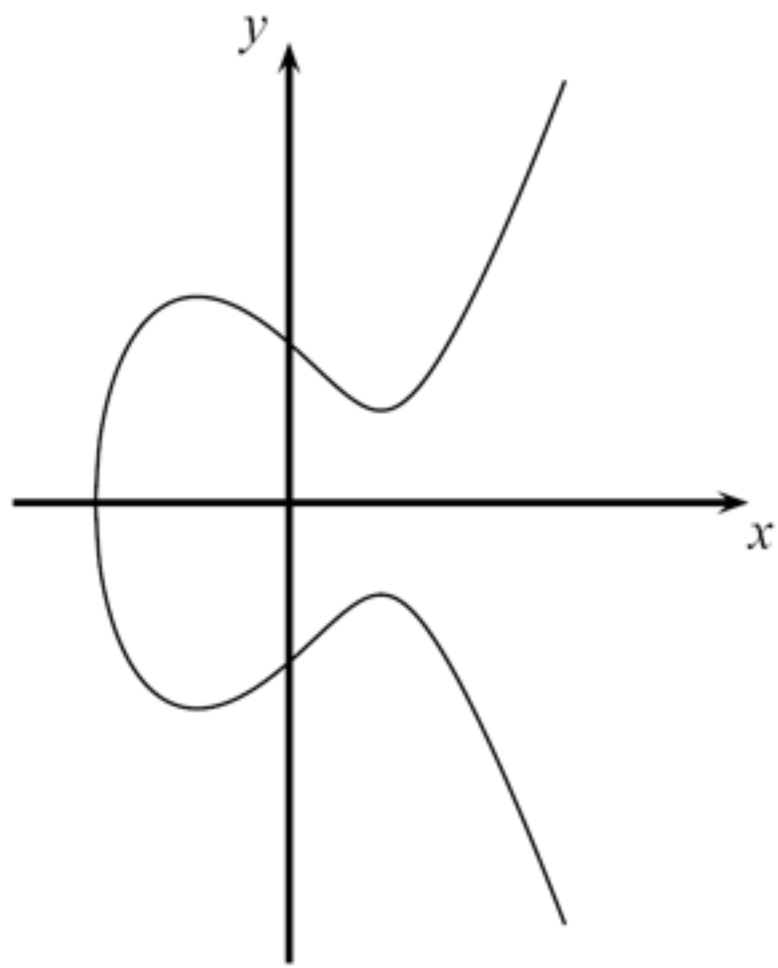
- Elliptic curves are polynomials that define points based on the (simplified) Weierstraß equation:

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

  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

Computations on Elliptic Curves (ctd.)

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

Definition 9.1.1 Elliptic Curve
The elliptic curve *over* $\mathbb{Z}_p$, $p > 3$, *is the set of all pairs* $(x,y) \in \mathbb{Z}_p$ *which fulfill*

$$y^2 \equiv x^3 + a \cdot x + b \bmod p \tag{9.1}$$

together with an imaginary point of infinity $\mathcal{O}$, *where*

$$a, b \in \mathbb{Z}_p$$

and the condition $4 \cdot a^3 + 27 \cdot b^2 \neq 0 \bmod p$.

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

Computations on Elliptic Curves (ctd.)

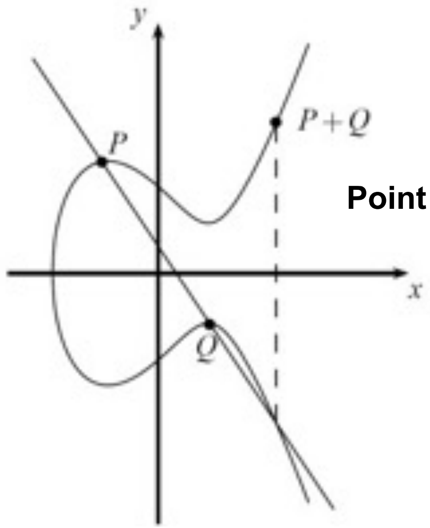
- Identity Point θ
  - *In any group, a special element is required to allow for the identity operation, i.e.,*

    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 (infinite) identity point is denoted by* θ
- Elliptic Curves 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 $-P = (x,-y)$

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*



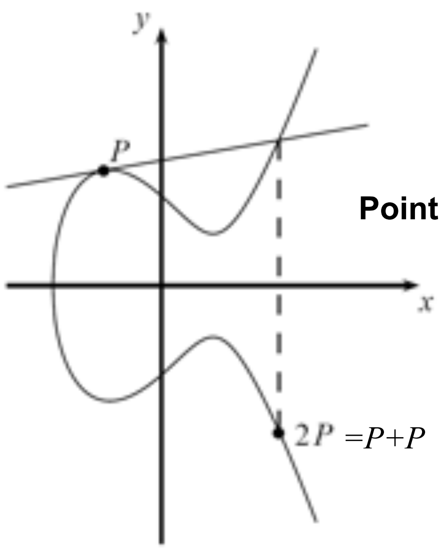
Point Addition

Elliptic Curve Point Addition and Point Doubling

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

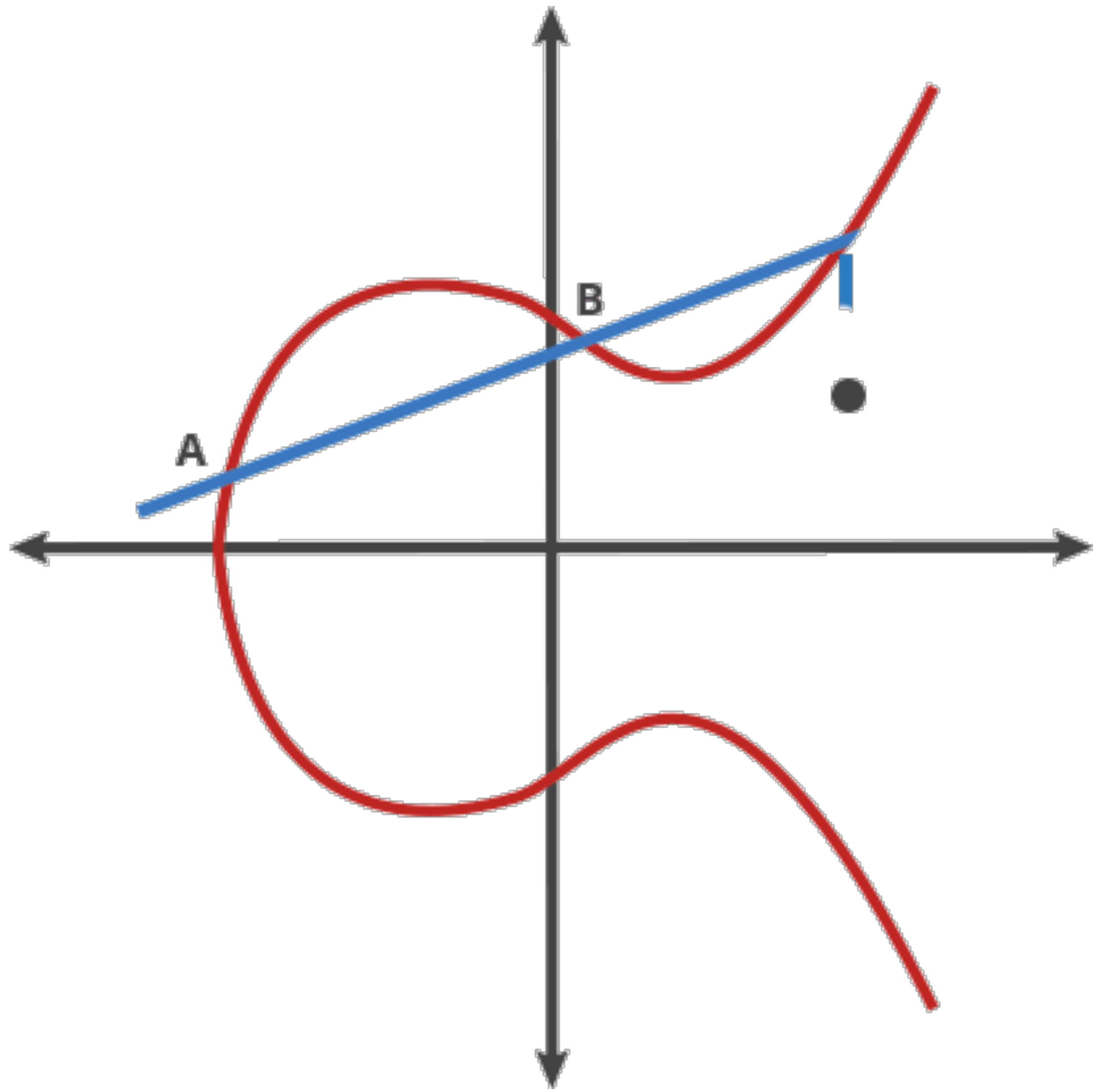
where

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



Point Doubling

Animation at Link Ch 9a



Computations on Elliptic Curves (ctd.)

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

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

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

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

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

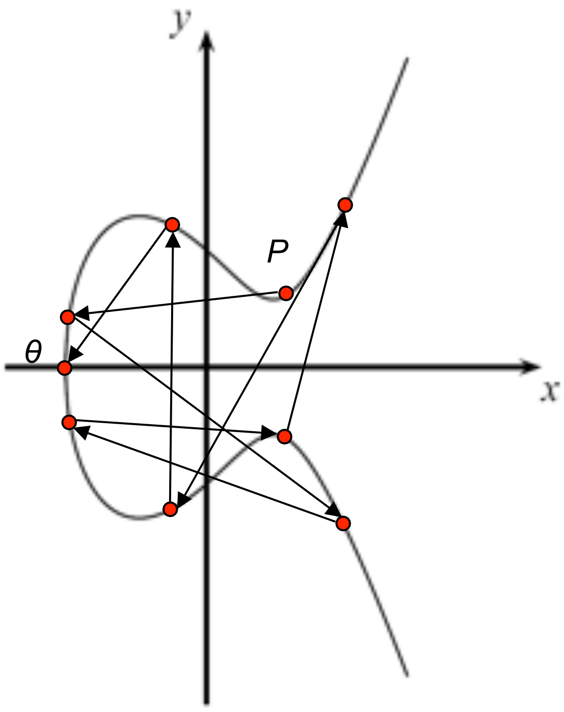
Finally $2P = (5,1) + (5,1) = (6,3)$

Computations on Elliptic Curves (ctd.)

- The points on an elliptic curve and the point at infinity θ form cyclic subgroups

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

This elliptic curve has order $\#E = |E| = 19$ since it contains 19 points in its cyclic group.



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

Theorem 9.2.2 Hasse's theorem
Given an elliptic curve E modulo 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^{160} points, a prime with a length of about 160 bits is required

Elliptic Curve Discrete Logarithm Problem

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

Definition 9.2.1 Elliptic Curved Discrete Logarithm Problem (ECDLP)
Given is an elliptic curve E. We consider a primitive element P and another element T. The DL problem is finding the integer d, where $1 \leq d \leq \#E$, such that:

$$\underbrace{P + P + \cdots + P}_{d \;\; times} = d\,P = T. \tag{9.2}$$

Elliptic Curve Discrete Logarithm Problem

$$\underbrace{P+P+\cdots+P}_{d\ \ times} = dP = T.$$

- 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
  - Known Square-and-Multiply Method can be adapted to Elliptic Curves
  - The method for efficient point multiplication on elliptic curves: Double-and-Add Algorithm

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:

1 FOR $i = t-1$ DOWNTO 0

1.1 $T = T + T \bmod n$

IF $d_i = 1$

1.2 $T = T + P \bmod n$

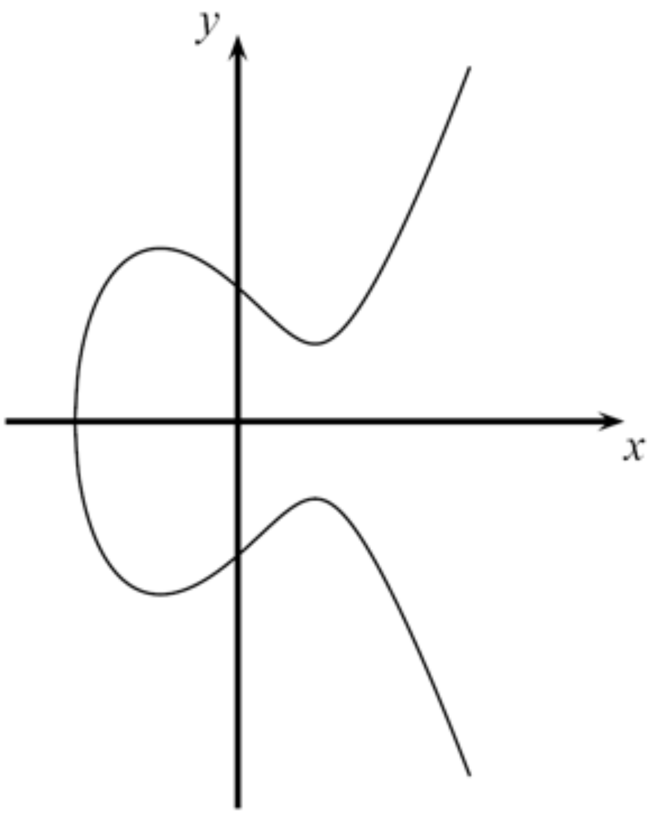
2 RETURN (T)

Example: Double-and-Add Algorithm for Point Multiplication

Step		
#0	$P = \mathbf{1}_2 P$	inital setting, bit processed: $d_4 = 1$
#1a	$P + P = 2P = \mathbf{10}_2 P$	DOUBLE, bit processed: d_3
#1b	$2P + P = 3P = 10_2 P + 1_2 P = \mathbf{11}_2 P$	ADD, since $d_3 = 1$
#2a	$3P + 3P = 6P = 2(11_2 P) = \mathbf{110_2} P$	DOUBLE, bit processed: d_2
#2b		no ADD, since $d_2 = 0$
#3a	$6P + 6P = 12P = 2(110_2 P) = \mathbf{1100_2} P$	DOUBLE, bit processed: d_1
#3b	$12P + P = 13P = 1100_2 P + 1_2 P = \mathbf{1101}_2 P$	ADD, since $d_1 = 1$
#4a	$13P + 13P = 26P = 2(1101_2 P) = \mathbf{11010_2} P$	DOUBLE, bit processed: d_0
#4b		no ADD, since $d_0 = 0$

Kahoot!

9.3 Diffie-Hellman Key Exchange with Elliptic Curves



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:

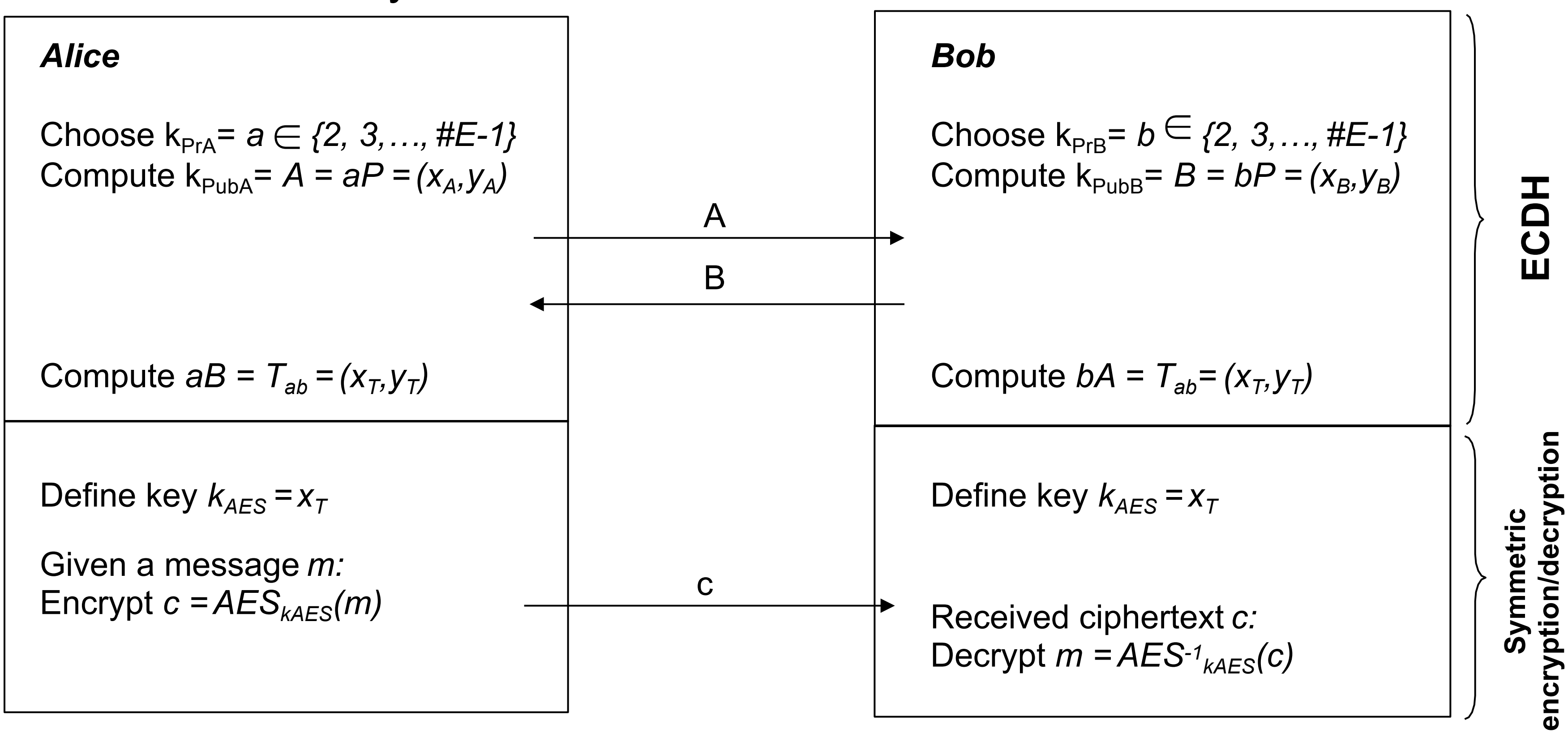
Alice		**Bob**
Choose $k_{PrA} = a \in \{2, 3, \ldots, \#E-1\}$ Compute $k_{PubA} = A = aP = (x_A, y_A)$		Choose $k_{PrB} = b \in \{2, 3, \ldots, \#E-1\}$ Compute $k_{PubB} = B = bP = (x_B, y_B)$
	$\xrightarrow{A}$	
	$\xleftarrow{B}$	
Compute $aB = T_{ab}$		Compute $bA = T_{ab}$

- Joint secret between Alice and Bob: $T_{AB} = (x_{AB}, y_{AB})$

- 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)

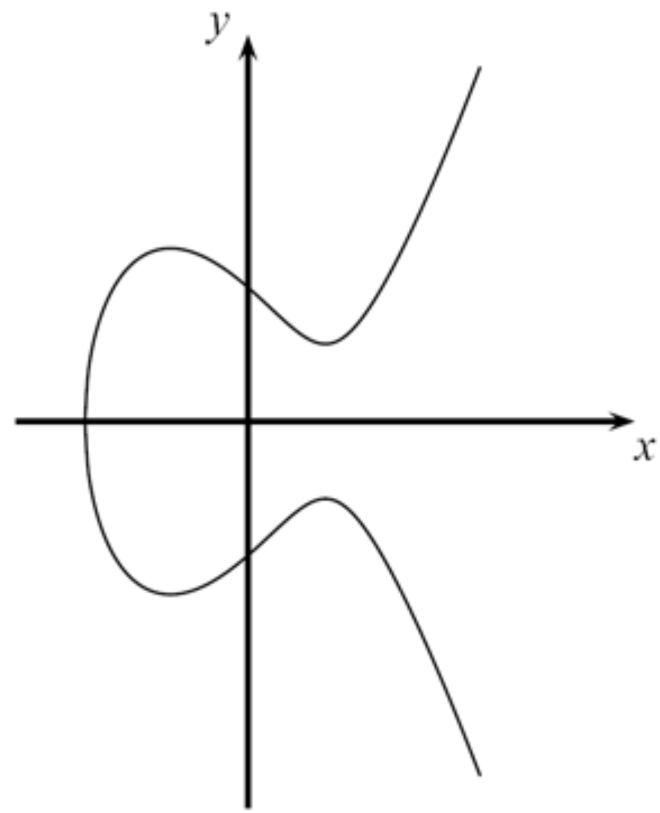
ECDH (ctd.)

- 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 is used to derive the session key

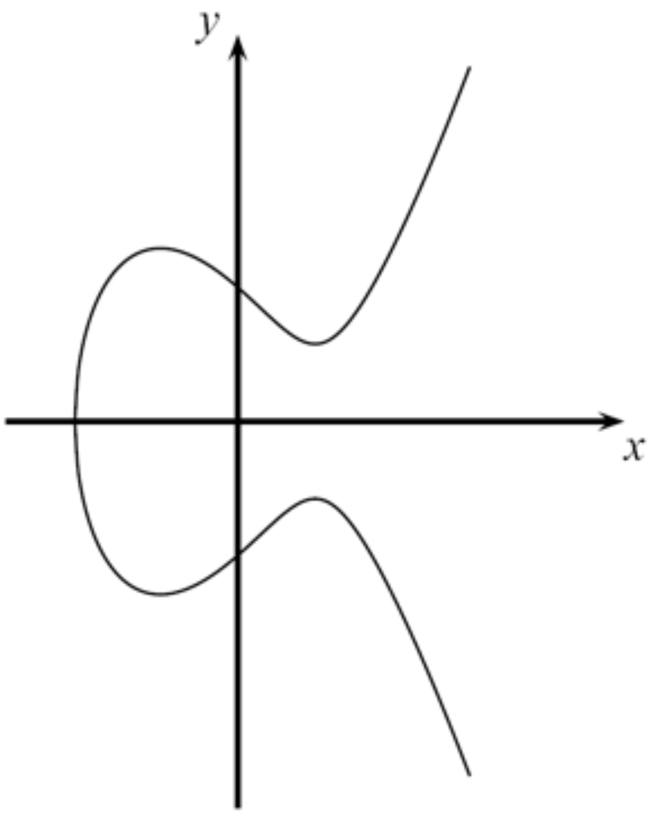
9.4 Security



Security Aspects

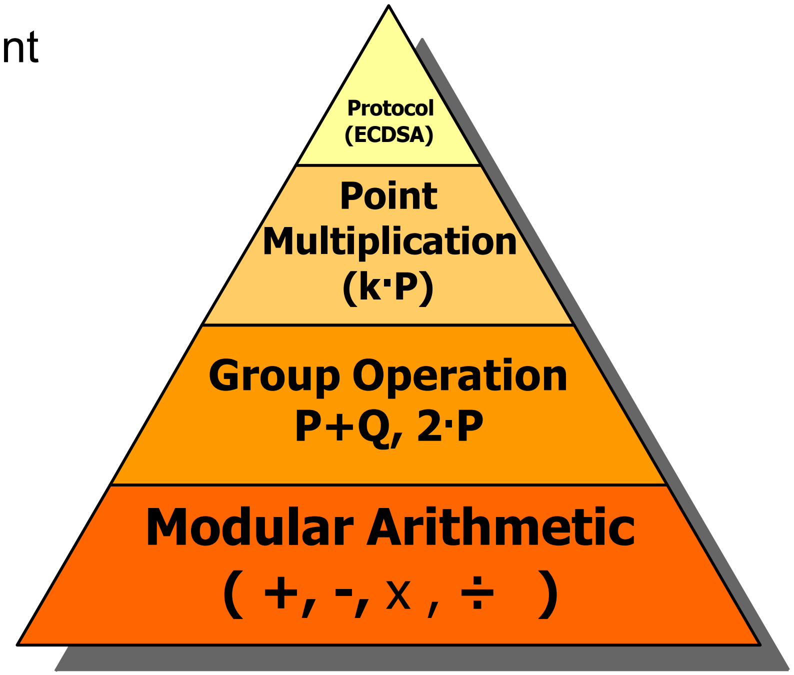
- Why are parameters signficantly smaller for elliptic curves (160-256 bit) than for RSA (1024-3076 bit)?
  - Attacks on groups of elliptic curves are weaker than available factoring algorithms or integer DL attacks
  - Best known attacks on elliptic curves are the Baby-Step Giant-Step and Pollard-Rho method
  - Number of steps required: $\sqrt{p}$
  - An elliptic curve using a prime p with 160 bits (and roughly 2^{160} points) provides a security of 2^{80} steps required by an attacker
  - An elliptic curve using a prime p with 256 bit (roughly 2^{256} points) provides a security of 2^{128} steps

9.5 Implementation in Software and Hardware



Implementations in Hardware and Software

- Computations have four layers:
  - **Basic modular arithmetic**: computationally most expensive
  - **Group operation:** point doubling and point addition
  - **Point multiplication:** Double-and-Add method
  - **Upper layer protocols:** like ECDH and ECDSA

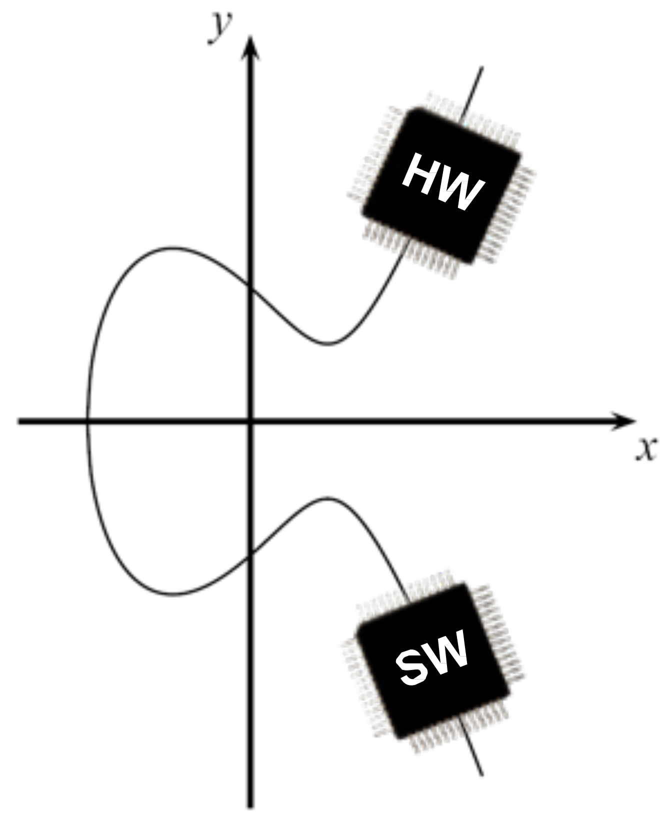


- Most efforts should go in optimizations of the modular arithmetic operations, such as
  - Modular addition and subtraction
  - Modular multiplication
  - Modular inversion

Implementations in Hardware and Software

- Software implementations
  - Optimized 256-bit ECC implementation on 3GHz 64-bit CPU requires about *2 ms* per point multiplication
  - Less powerful microprocessors (e.g, on SmartCards or cell phones) even take significantly longer (>*10 ms*)

- Hardware implementations
  - High-performance implementations with 256-bit special primes can compute a point multiplication in a few hundred microseconds on reconfigurable hardware
  - Dedicated chips for ECC can compute a point multiplication in a few tens of microseconds



Key Length

- To double the effort for an attacker, **add two bits** to ECC key length
- For RSA and DL, you must **add 20-30 bits** to double an attacker's effort

Attacks against the Discrete Logarithm Problem

Elliptic curves challenges with key sizes of 108 and 109 bits have been solved

But no solutions are known for 131-bit keys

Quantum Computers

- The existence of quantum computers would probably be the end for ECC, RSA & DL
- TEXTBOOK SAYS:
  - *At least 2-3 decades away, and some people doubt that QC will ever exist*

NIST Recommendations from 2016

SP 800-57 Part 1 Rev. 4

Recommendation for Key Management, Part 1: General

Date Published: January 2016

Supersedes: SP 800-57 Part 1 Rev. 3 (July 2012);

NIST Recommendations from 2016

Table 2: Comparable strengths

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

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 (approximately 160–256 bit vs. 1024–3072 bit), which results in shorter ciphertexts and signatures.
- In many cases ECC has performance advantages over other public-key algorithms.
- ECC is slowly gaining popularity in applications, compared to other public-key schemes, i.e., many new applications, especially on embedded platforms, make use of elliptic curve cryptography.

Kahoot!