b
UNIVERSITÄT
BERN

Network Security

III. Asymmetric Encryption

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 07.03.2022 – 14.03.2022

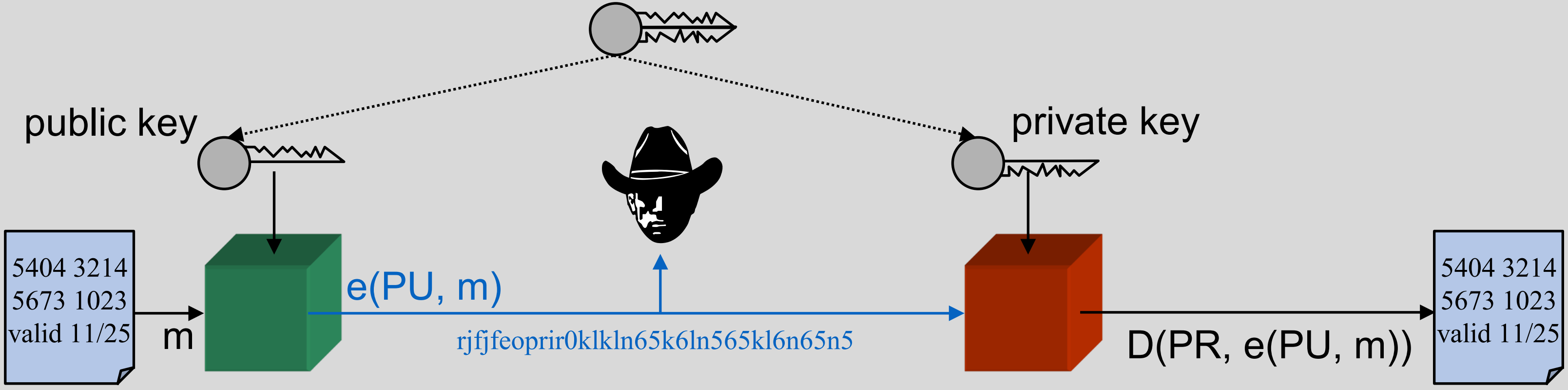
Network Security: Asymmetric Encryption

Table of Contents

b UNIVERSITÄT BERN

1. Introduction

1. Asymmetric Encryption



b
UNIVERSITÄT
BERN

1. Introduction

2. Asymmetric Encryption Problems

- Attacker knows public key, encryption scheme, and cipher text.
- Everybody can send a message to a receiver and imitate identities.
- Very computing intensive compared to symmetric encryption

UNIVERSITÄT BERN

1. Introduction

3. Asymmetric Encryption Applications

- Encryption and decryption
- Digital signatures: encrypting a (part of a) message with a private key
- Symmetric key exchange

1. Introduction

4. Asymmetric Encryption: Requirements

- Computationally easy to
  - generate a public/private key pair
  - encrypt a message C = e(PU, M)
  - decrypt a message M = D(PR, C)

- Computationally infeasible to
  - determine the private key PR
  - recover message M knowing the public key PU and ciphertext C

- 2 keys are applied in either order:
  M = D(PU, e(PR, M))
  = D(PR, e(PU, M))

1. Introduction

5. Asymmetric Encryption: Public Key Cryptanalysis

Key size must be

- large enough to make brute-force attack infeasible,
- but small enough for practical encryption and decryption.

Another attack:
find private key from public key.
Today, it has not been mathematically proven that this is infeasible for all asymmetric encryption algorithms.

2. Rivest Shamir Adleman

1. Algorithm

Key Generation

1. Selection of 2 big prime numbers p, q, e.g., 1024 bits
2. Calculate $n = p \cdot q$; $z = (p-1)\cdot(q-1)$
3. Select $e < n$, so that e and z do not have common factors.
4. Select d so that $e \cdot d \bmod z = 1$.
5. Public key: <e, n>, private key: <d, n>

- **Encryption**: $c = m^e \bmod n$
  - m: message in plaintext
  - c: message in ciphertext
- **Decryption**: $m = c^d \bmod n$
- Security is based on the fact that there are no fast algorithms for prime factorization.

UNIVERSITÄT BERN

2. RSA

2. Example

Key Generation

- $p = 7, q = 11$
- $n = 77;$
  $z = (p-1)\cdot(q-1) = 60 = 5\cdot3\cdot2\cdot2$
- $e = 7$
- $7d \bmod 60 = 1 \Rightarrow d = 43$
  $(7\cdot43 = 301, 301 \bmod 60 = 1)$
- $\langle e,n \rangle = \langle 7,77 \rangle; \langle d,n \rangle = \langle 43,77 \rangle$

Encryption

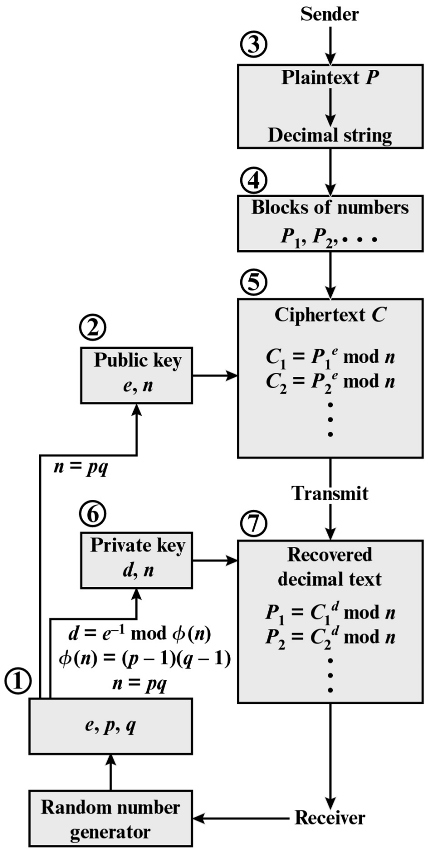
- $m = 9, c = 9^7 \bmod 77 = 37$

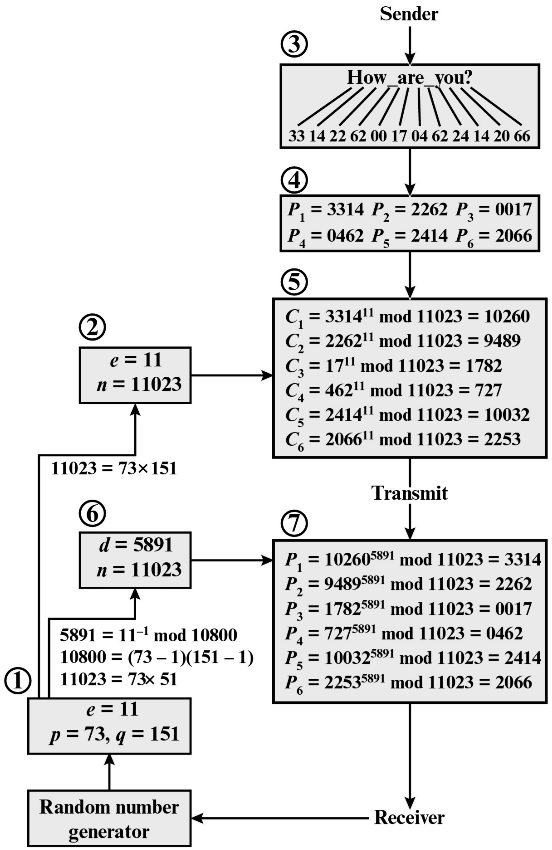
Decryption

- $m = 37^{43} \bmod 77 = 9$

2. RSA

3. Processing of Multiple Blocks





2. RSA

4.1 Security Attacks

- Brute force
- Mathematical attacks
  - Efforts to factoring the product of two primes

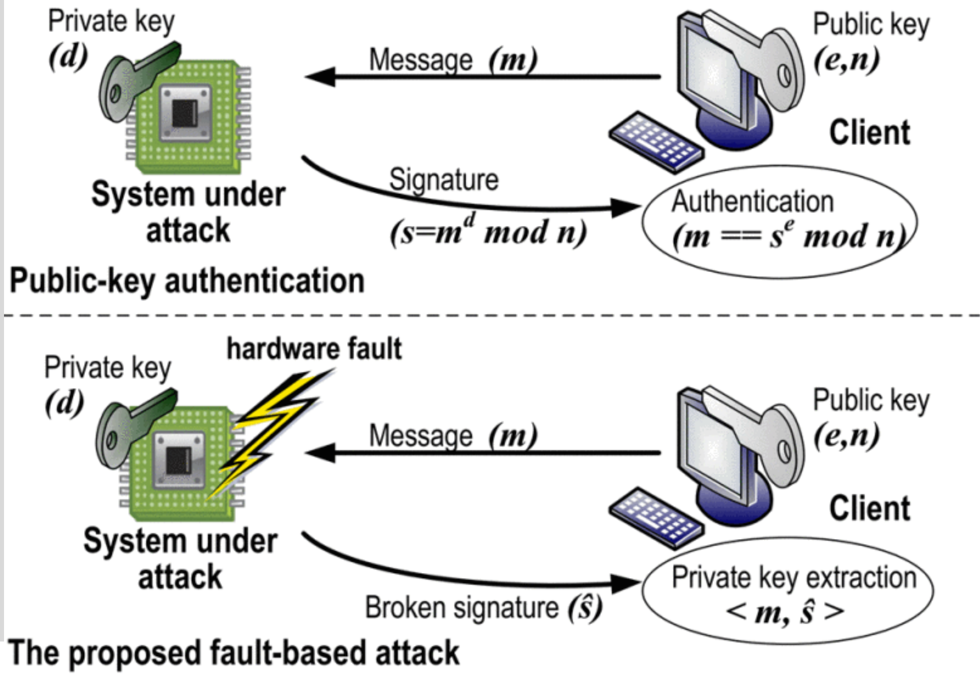
- Timing attacks
  - Measuring decryption running time dependent on data
  - Solutions
    - constant time
    - random delays
    - Blinding: multiply ciphertext before exponentiation

Default solution: large keys

2. RSA

4.2 Security Attacks

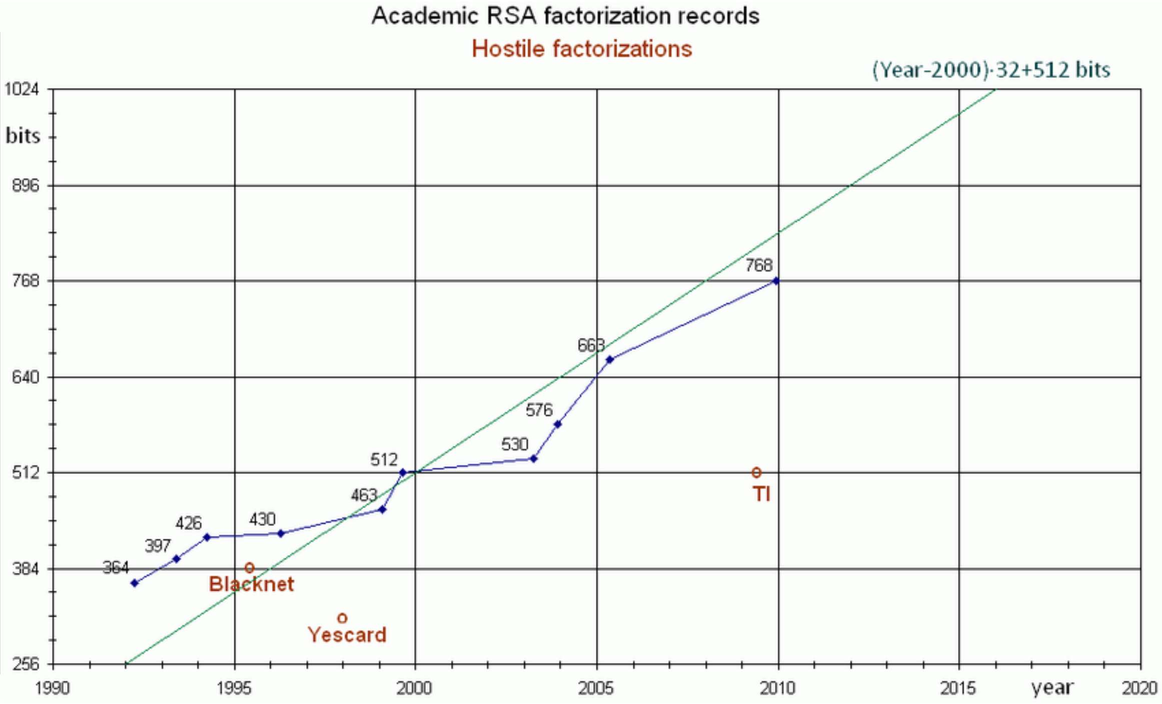
- Hardware fault-based attacks
  - inducing hardware faults in generating signatures,
    e.g., by reducing processor power
  - requires access to the hardware
- Chosen ciphertext attack
  - exploits properties of RSA algorithm
  - Adversary chooses ciphertext and is given corresponding plaintext.
  - From that the private key could be derived.



2. RSA

5. Factorization

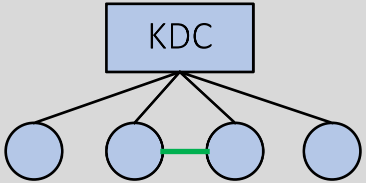
2020: RSA-250 (829 bits) was factored.



3. Key Management

1. Session Key Exchange

- Public keys → certificates
- Key Distribution Center
  - Negotiation of N keys with N clients
  - KDC calculates session keys and
    uses one of the N keys for key exchange
- Diffie Hellman Key Exchange



UNIVERSITÄT BERN

3. Key Management

2.1 Diffie Hellman Key Exchange

- A and B exchange prime p and generator g (also prime).
- A selects secret random number x (private key), calculates $n = g^x \bmod p$, and transmits n (public key) to B.
- B selects secret random number y (private key), calculates $m = g^y \bmod p$, and transmits m (public key) to A.
- Session key: $z = n^y \bmod p = m^x \bmod p = g^{xy} \bmod p$
- Security by infeasibility to compute x and y (discrete logarithm), which is much more complex than prime factorization

UNIVERSITÄT BERN

3. Key Management

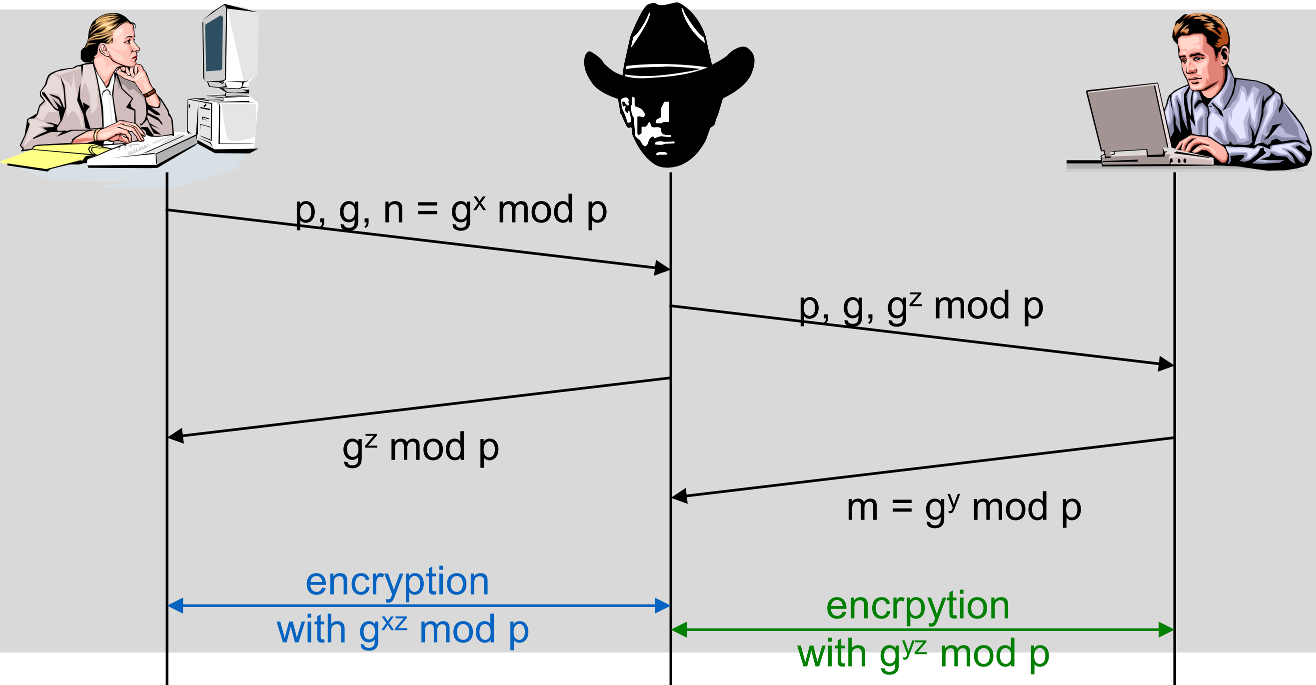
2.2 Diffie Hellman Key Exchange Example

- p = 47, g = 3, A: x = 8, B: y = 10
- A $\rightarrow$ B: (47, 3, n = 28 (= 3^8 mod 47))
- B $\rightarrow$ A: (47, 3, m= 17 (= 3^{10} mod 47))
- Key z = 17^8 mod 47 = 28^{10} mod 47 = 3^{80} mod 47 = 4

- Problem: „Man in the Middle“ attack
- Solution: Authenticated DH exchange using private or public keys

3. Key Management

3. „Man in the Middle“ Attack



4. Elliptic Curves

1. Arithmetics

- Most public-key cryptography products and standards use RSA.
- The key length for secure RSA use has increased over recent years and generates heavier processing load on applications using RSA.
- Elliptic Curve Cryptography is showing up in standardization efforts including IEEE P1363 for Public-Key Cryptography.
- ECC aims to offer equal security for a far smaller key size.

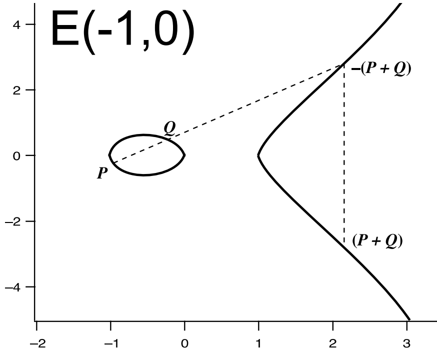
4. Elliptic Curves

2. Elliptic Curves over Real Numbers

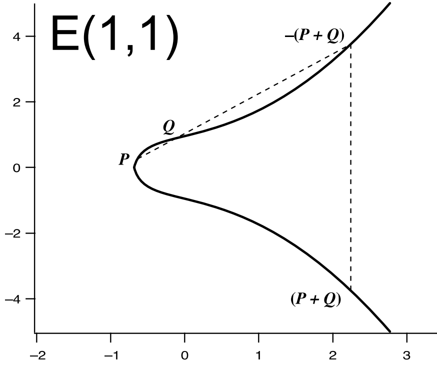
- Elliptic curves are not ellipses.
- Elliptic curves are described by cubic equations, similar to those used for calculating the circumference of an ellipse (Weierstrass equation):

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

- Here: equations of the form $y^2 = x^3 + ax + b$
- To plot such a curve, we need to compute $y = \sqrt{(x^3 + ax + b)}$
- Sets of points E(a, b) depict curves, e.g., E(-1,0), E(1,1)
- For any 3 points in such a set: the sum is O (zero point).



(a) $y^2 = x^3 - x$



(b) $y^2 = x^3 + x + 1$

UNIVERSITÄT BERN

4. Elliptic Curves

3. Addition Rules

O = -O

P + O = P

P + (-P) = O

To add two points P, Q with different x coordinates:
draw a straight line and find the point of intersection: P + Q = -R

4. Elliptic Curves

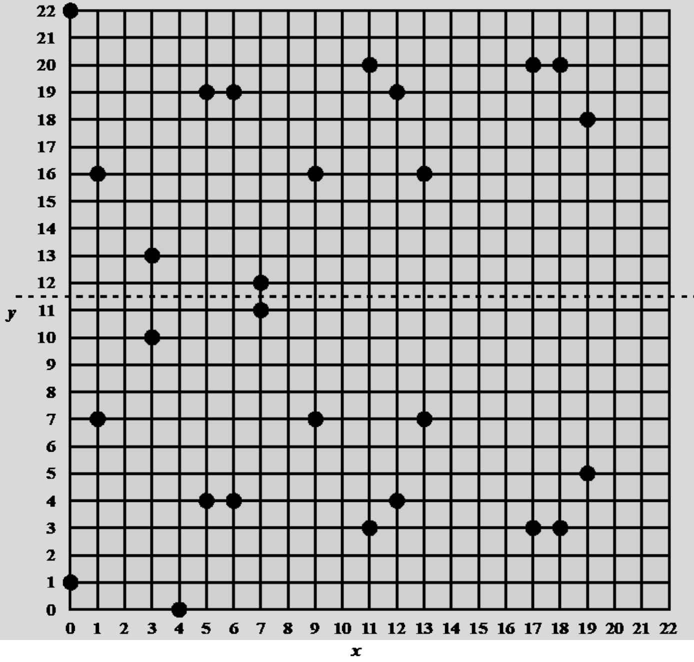
4. Elliptic Curves over Z_p

- Elliptic curve cryptography makes use of elliptic curves, in which variables and coefficients are restricted to elements of a finite field.
- Prime curve over Z_p
  - integers from 0 to p-1
  - good for software processing
- Binary curve over $GF(2^m)$
  - values in $GF(2^m)$ and calculations over $GF(2^m)$
  - good for hardware processing
- here:
  $y^2 \bmod p = (x^3 + ax + b) \bmod p$,
  which is for example satisfied by
  a=1, b=1, x=9, y=7, p=23
  - $7^2 \bmod 23 = (9^3 + 9 + 1) \bmod 23$
  - $49 \bmod 23 = 739 \bmod 23$
  - $3 = 3$
- Coefficients a, b and variables x, y are all elements of Z_p.

4. Elliptic Curves

5. Points on the Elliptic Curve $E_{23}(1,1)$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)



UNIVERSITÄT BERN

4. Elliptic Curves

6. $E_p(a,b)$ Addition Rules

1. P + O = P
2. $P = (x_P, y_P)$: $P + (x_P, -y_P) = O$
3. $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, $P \neq -Q$: $R = P + Q = (x_R, y_R)$
4. Multiplication as repeated addition, e.g., 4P = P + P + P + P

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p$$
$$y_R = (\lambda(x_P - x_R) - y_P) \bmod p$$

$$\lambda = \begin{cases} \left(\dfrac{y_Q - y_P}{x_Q - x_P}\right) \bmod p & \text{if } P \neq Q \\ \left(\dfrac{3x_P^2 + a}{2y_P}\right) \bmod p & \text{if } P = Q \end{cases}$$

4. Elliptic Curves

7. Elliptic Curves over GF(2^m)

- A finite field (Galois Field) GF(2^m) consists of 2^m elements together with addition and multiplication operations that can be defined over polynomials.
- Cubic equation appropriate for cryptographic applications is
  $y^2 + xy = x^3 + ax^2 + b$,
  x, y, a, b are elements of GF(2^m).

- Example: GF(2^4) with the irreducible polynomial $f(x) = x^4 + x + 1$
- Generator g with f(g) = 0:
  $g^4 = g + 1$, in binary: g = 0010
- $g^5 = g^4\, g = (g + 1)\, g = g^2 + g = 0110$ (XOR)

g^0=0001	g^4=0011	g^8=0101	g^{12}=1111
g^1=0010	g^5=0110	g^9=1010	g^{13}=1101
g^2=0100	g^6=1100	g^{10}=0111	g^{14}=1001
g^3=1000	g^7=1011	g^{11}=1110	g^{15}=0001

UNIVERSITÄT BERN

4. Elliptic Curves

8. Example Point on $E_2^4(g^4,1)$

$y^2 + xy = x^3 + g^4x^2 + 1$

(g^5, g^3) satisfies equation.

$(g^3)^2 + g^5\, g^3 = (g^5)^3 + g^4\, (g^5)^2 + 1$

$g^6 + g^8 = g^{15} + g^{14} + 1$

$1100 + 0101 = 0001 + 1001 + 0001$

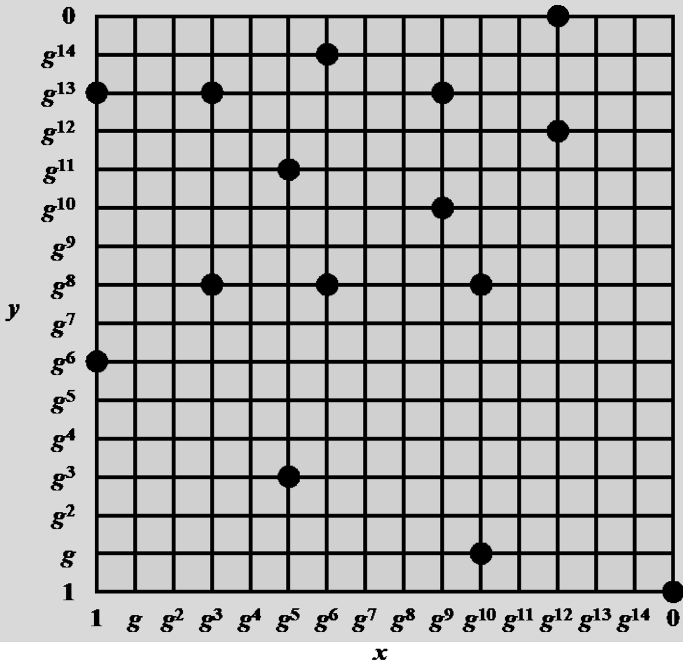
$1001 = 1001$

b UNIVERSITÄT BERN

4. Elliptic Curves

9. Points on the Elliptic Curve $E_2{}^4(g^4,1)$

$(0, 1)$	(g^5, g^3)	(g^9, g^{13})
$(1, g^6)$	(g^5, g^{11})	(g^{10}, g)
$(1, g^{13})$	(g^6, g^8)	(g^{10}, g^8)
(g^3, g^8)	(g^6, g^{14})	$(g^{12}, 0)$
(g^3, g^{13})	(g^9, g^{10})	(g^{12}, g^{12})



UNIVERSITÄT BERN

4. Elliptic Curves

10. $E_{2^m}(a,b)$ Addition Rules

1. P + O = P
2. P = (x_P, y_P): P + $(x_P, x_P + y_P)$ = O
   $(x_P, x_P + y_P)$ = -P
3. P = (x_p, y_p), Q = (x_Q, y_Q): R = P + Q = (x_R, y_R)
4. P = (x_p, y_p): R = 2P = (x_R, y_R)

$$x_R = \lambda^2 + \lambda + x_P + x_Q + a$$
$$y_R = \lambda(x_P + x_R) + x_R + y_P$$

$$\lambda = \frac{y_Q + y_P}{x_Q + x_P}$$

$$x_R = \lambda^2 + \lambda + a$$
$$y_R = x_P^2 + (\lambda + 1)x_R$$

$$\lambda = x_P + \frac{y_P}{x_P}$$

4. Elliptic Curves

11. Elliptic Curve Cryptography

To form a cryptographic system using elliptic curves, we need to find a “hard problem”, e.g.,

- factoring the product of two primes or
- taking the discrete logarithm.

- Q = k · P;
  Q, P belong to a prime curve.
- It is easy to compute Q given k, P, e.g., *100P=2(2(P+2(2(2(P + 2P)))))*
- It is hard to find k given Q, P.
- This is known as the elliptic curve logarithm problem.

b UNIVERSITÄT BERN

4. Elliptic Curves

12. Example ECC

- $E_{23}(9, 17)$ is defined by $y^2 \ mod\ 23 = (x^3 + 9x + 17)$ mod 23.
- What is the discrete logarithm k of Q = (4, 5) to the base P = (16, 5)?
- Brute-force method is to compute multiples of P until Q is found.
- *P = (16, 5); 2P = (20, 20); 3P = (14, 14); 4P = (19, 20); 5P = (13, 10); 6P = (7, 3); 7P = (8, 7); 8P = (12, 17); 9P = (4, 5).*
- discrete logarithm Q = (4, 5) to the base P = (16, 5) is k = 9.
- In reality, k would be so large as to make the brute-force approach infeasible.

UNIVERSITÄT BERN

4. Elliptic Curves

13. Analog to Diffie Hellman Key Exchange

- An attacker would need to calculate k given base point G and kG.
- Example: p=211, $E_p(0, -4)$: $y^2 = x^3-4$; G=(2, 2)
- 240 G = O
- Private key $n_A = 121$
  → public key P_A = 121 (2, 2) = (115, 48)
- $n_B = 203$ → P_B = 203 (2, 2) = (130, 203)
- Shared key = 121 · (130, 203) = 203 · (115, 48) = 121 · 203 (2, 2) = (161, 69)

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

UNIVERSITÄT BERN

4. Elliptic Curves

14. Comparable Key Sizes in Terms of Computational Efforts for Cryptanalysis

Symmetric key algorithms	Diffie-Hellman, Digital Signature Algorithm	RSA (size of n in bits)	ECC (modulus size in bits)
80	$L = 1024$ $N = 160$	1024	160–223
112	$L = 2048$ $N = 224$	2048	224–255
128	$L = 3072$ $N = 256$	3072	256–383
192	$L = 7680$ $N = 384$	7680	384–511
256	$L = 15{,}360$ $N = 512$	15,360	512+

Thanks

for Your Attention

UNIVERSITÄT
BERN

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 07.03.2022 – 14.03.2022