CRYPTOGRAPHY

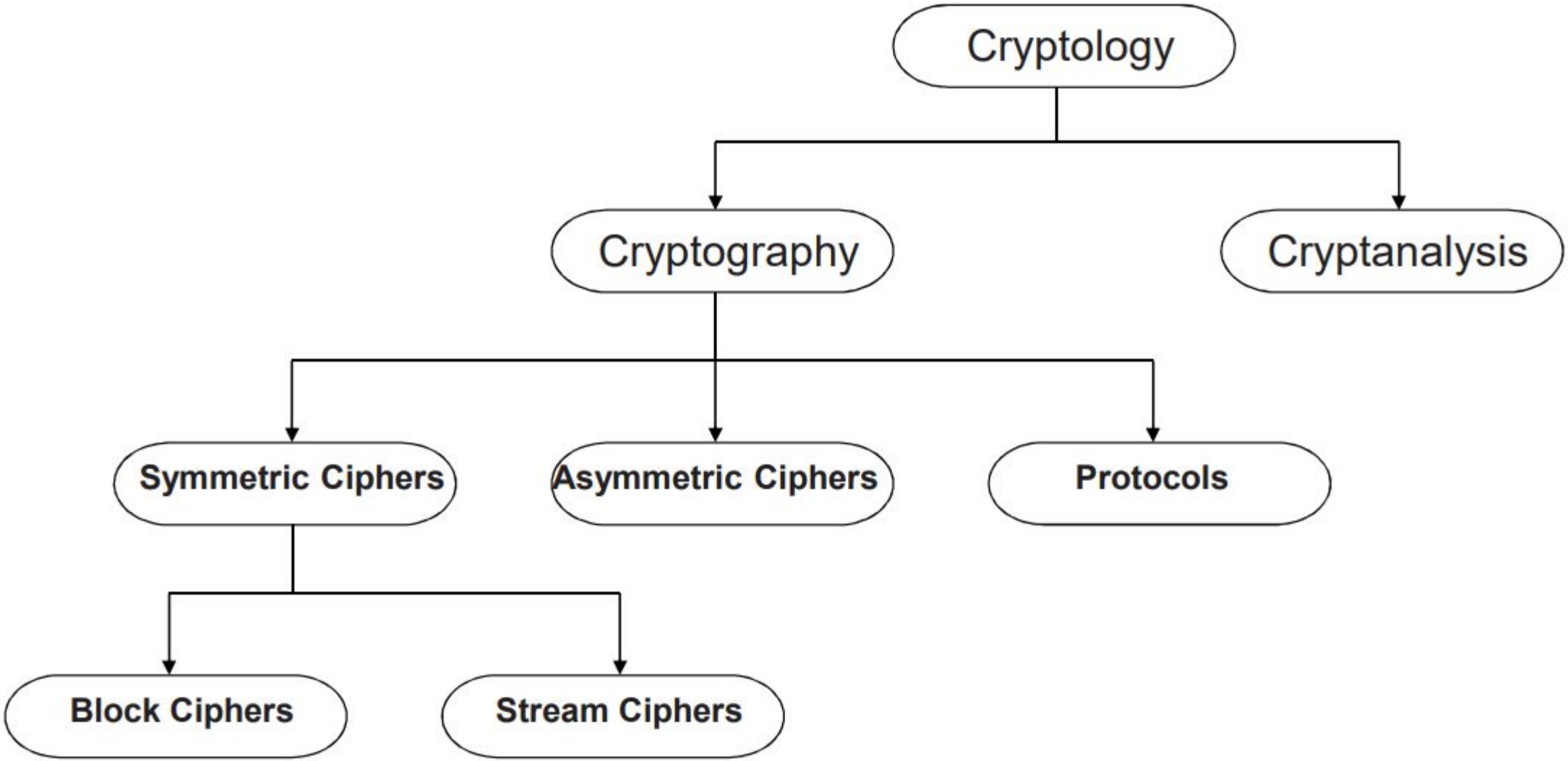
A.K.M. MEHEDI HASAN
Lecturer
CSE,BUET

Cryptography

- Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents.

ABC (meaningful message)-> ZYX(cipher)

Classification of the Field of Cryptology



- **Ancient Crypto:** Early signs of encryption in Eqypt in ca. 2000 B.C.
  Letter-based encryption schemes (e.g., Caesar cipher) popular ever since.

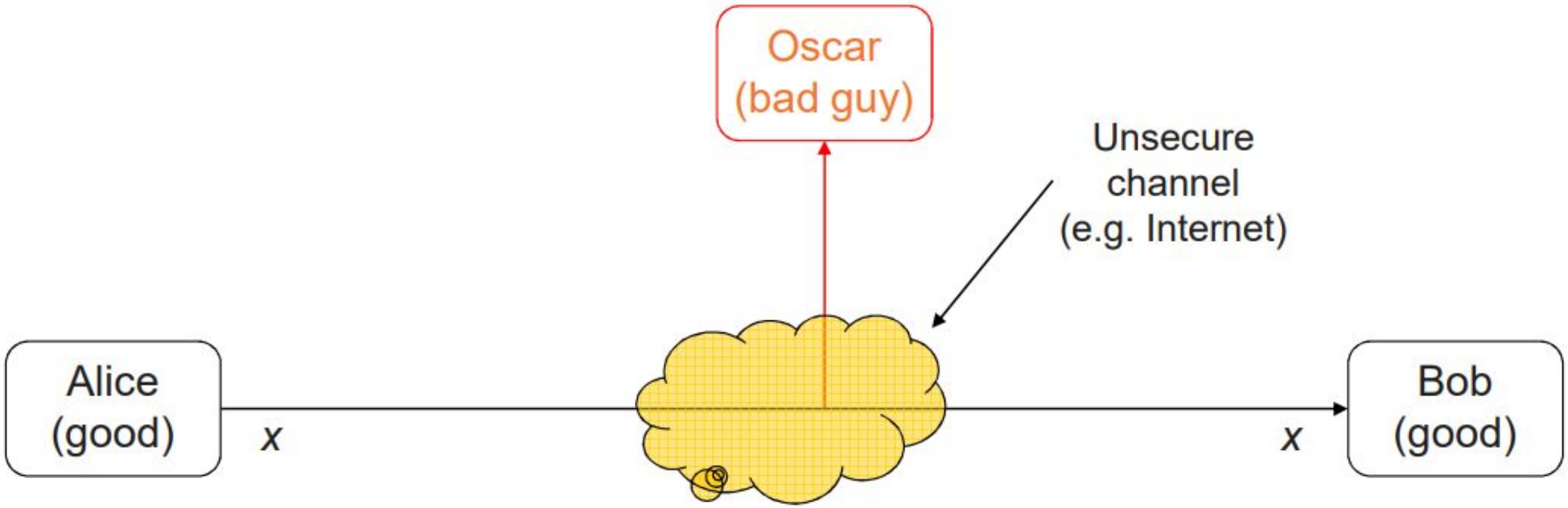
- **Symmetric ciphers:** All encryption schemes from ancient times until 1976 were symmetric ones.

- **Asymmetric ciphers:** In 1976 public-key (or asymmetric) cryptography was openly proposed by Diffie, Hellman and Merkle.

- **Hybrid Schemes:** The majority of today‘s protocols are hybrid schemes, i.e., the use both
  - symmteric ciphers (e.g., for encryption and message authentication) and
  - asymmetric ciphers (e.g., for key exchange and digital signature).

Symmetric Cryptography

- Alternative names: **private-key**, **single-key** or **secret-key** cryptography.



- **Problem Statement:**

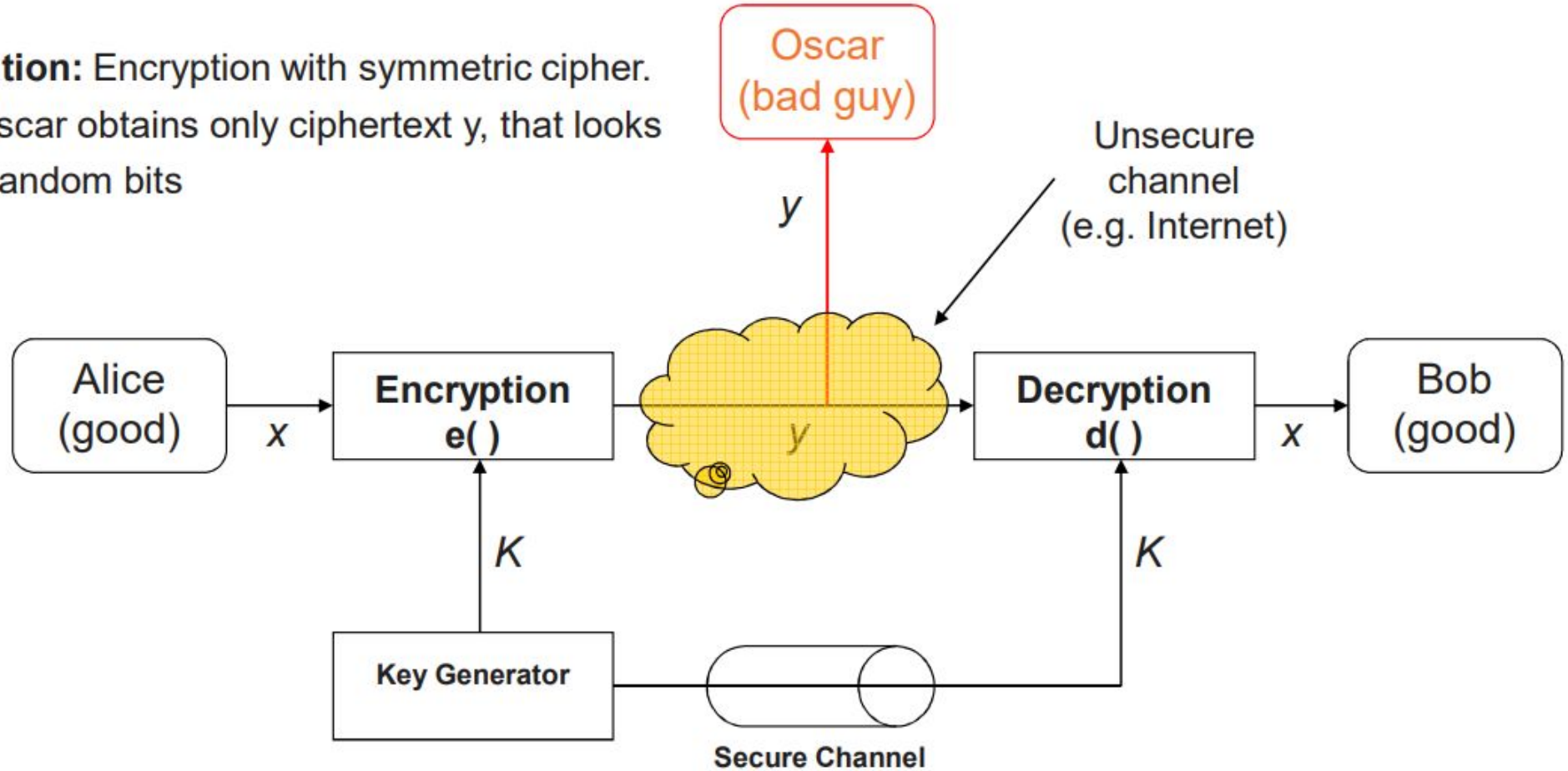
  1) Alice and Bob would like to communicate via an unsecure channel (e.g., WLAN or Internet).

  2) A malicious third party Oscar (the bad guy) has channel access but should not be able to understand the communication.

Symmetric Cryptography

Solution: Encryption with symmetric cipher.

⇒ Oscar obtains only ciphertext y, that looks like random bits



- x is the. **plaintext**
- y is the **ciphertext**
- *K* is the **key**
- Set of all keys {*K*1, *K*2, ...,*Kn*} is the **key space**

Symmetric Cryptography

- Encryption equation $\mathbf{y = e_K(x)}$
- Decryption equation $\mathbf{x = d_K(y)}$

- Encryption and decryption are inverse operations if the same key K is used on both sides:

$$d_K(y) = d_K(e_K(x)) = x$$

- Important: The key must be transmitted via a **secure channel** between Alice and Bob.
- The secure channel can be realized, e.g., by manually installing the key for the Wi-Fi Protected Access (WPA) protocol or a human courier.
- However, the system is only secure if an attacker does not learn the key K!

⇒ **The problem of secure communication is reduced to secure transmission and storage of the key K.**

Things to be remembered

- Never ever develop your own crypto algorithm unless you have a team of experienced cryptanalysts checking your design.
- Do not use unproven crypto algorithms or unproven protocols.
- Attackers always look for the weakest point of a cryptosystem. For instance, a large key space by itself is no guarantee for a cipher being secure; the cipher might still be vulnerable against analytical attacks.
- Key lengths for symmetric algorithms in order to thwart exhaustive key-search attacks:
  - 64 bit: insecure except for data with extremely short-term value
  - 128 bit: long-term security of several decades, unless quantum computers become available (quantum computers do not exist and perhaps never will)
  - 256 bit: as above, but probably secure against attacks by quantum computers.

What is AES?

- AES is an encryption standard chosen by the National Institute of Standards and Technology(NIST), USA to protect classified information. It has been accepted world wide as a desirable algorithm to encrypt sensitive data.
- It is a block cipher which operates on block size of 128 bits for both encrypting as well as decrypting.
- Each Round performs similar operations.

Why AES?

- In 1990's the cracking of DES algorithm became possible.
- Around 50 hours of brute-forcing allowed to crack the message.
- NIST started searching for new feasible algorithm and proposed its requirement in 1997.
- In 2001 Rijndael algorithm designed by Rijment and Daemon of Belgium was declared as the winner of the competition.
- It met all Security, Cost and Implementation criteria.

How Does AES work?

- AES basically repeats 4 major functions to encrypt data. It takes 128 bit block of data and a key and gives a ciphertext as output.The functions are:
  - Substitute Bytes
  - Shift Rows
  - Mix Columns
  - Add Key

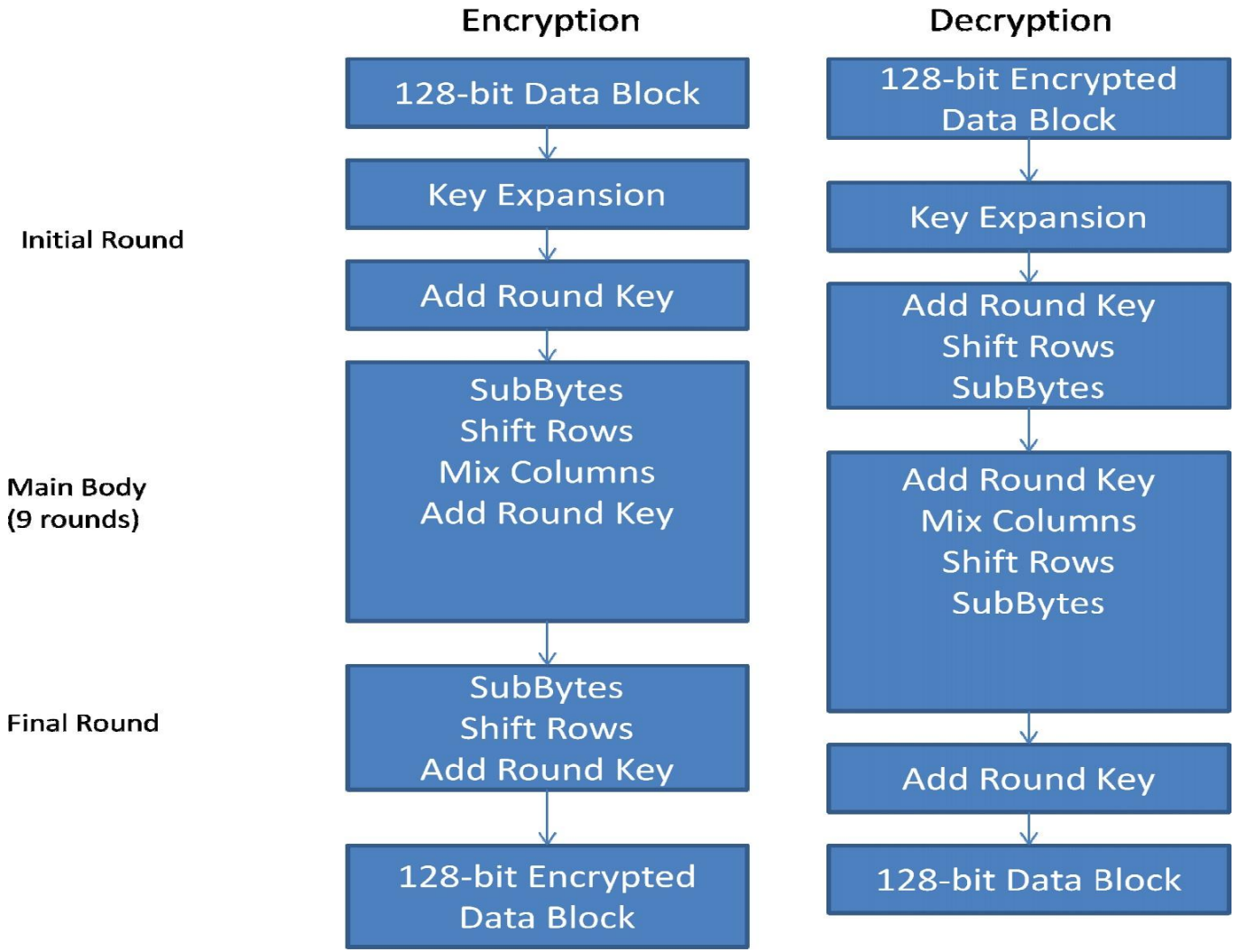
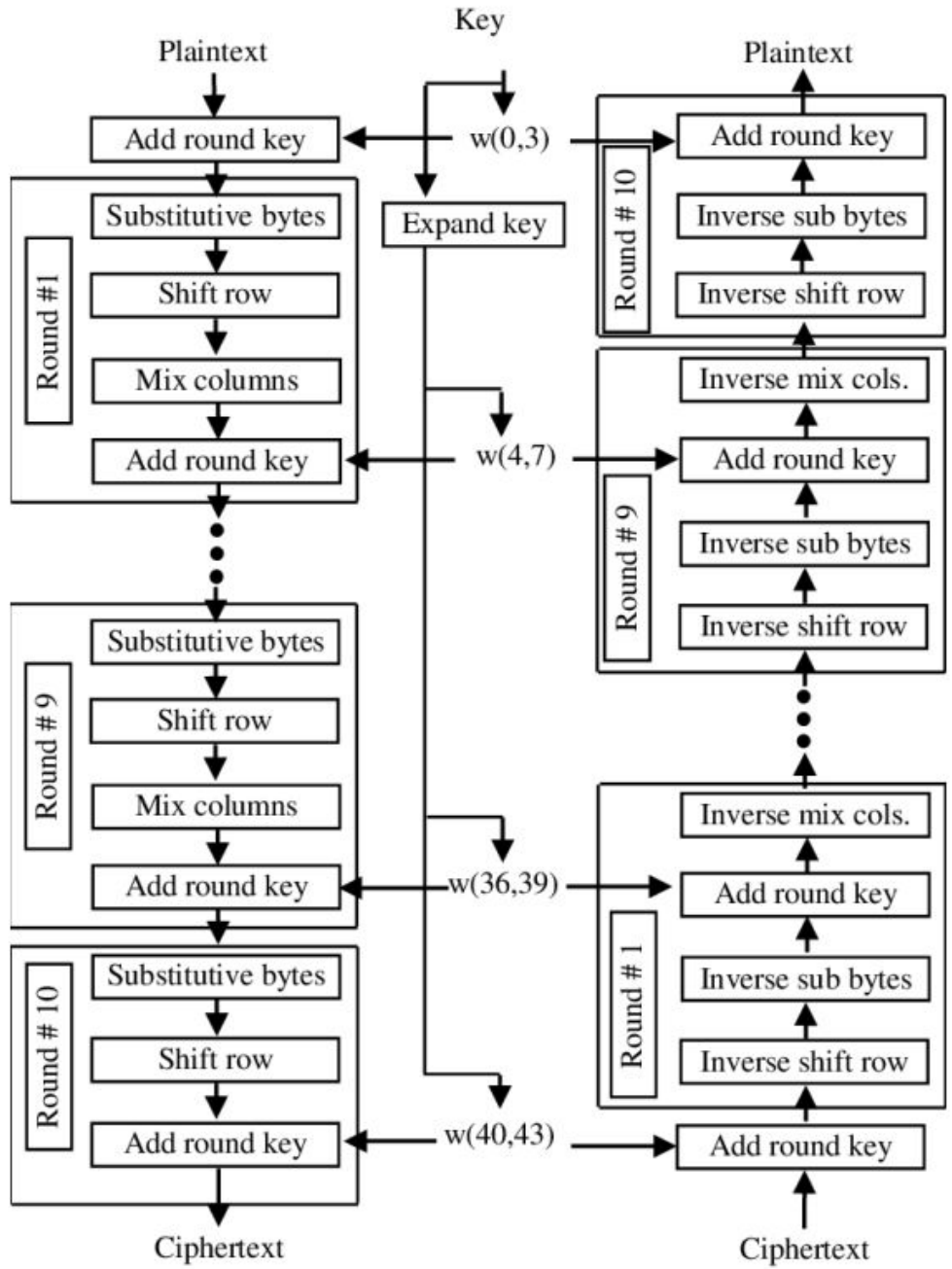
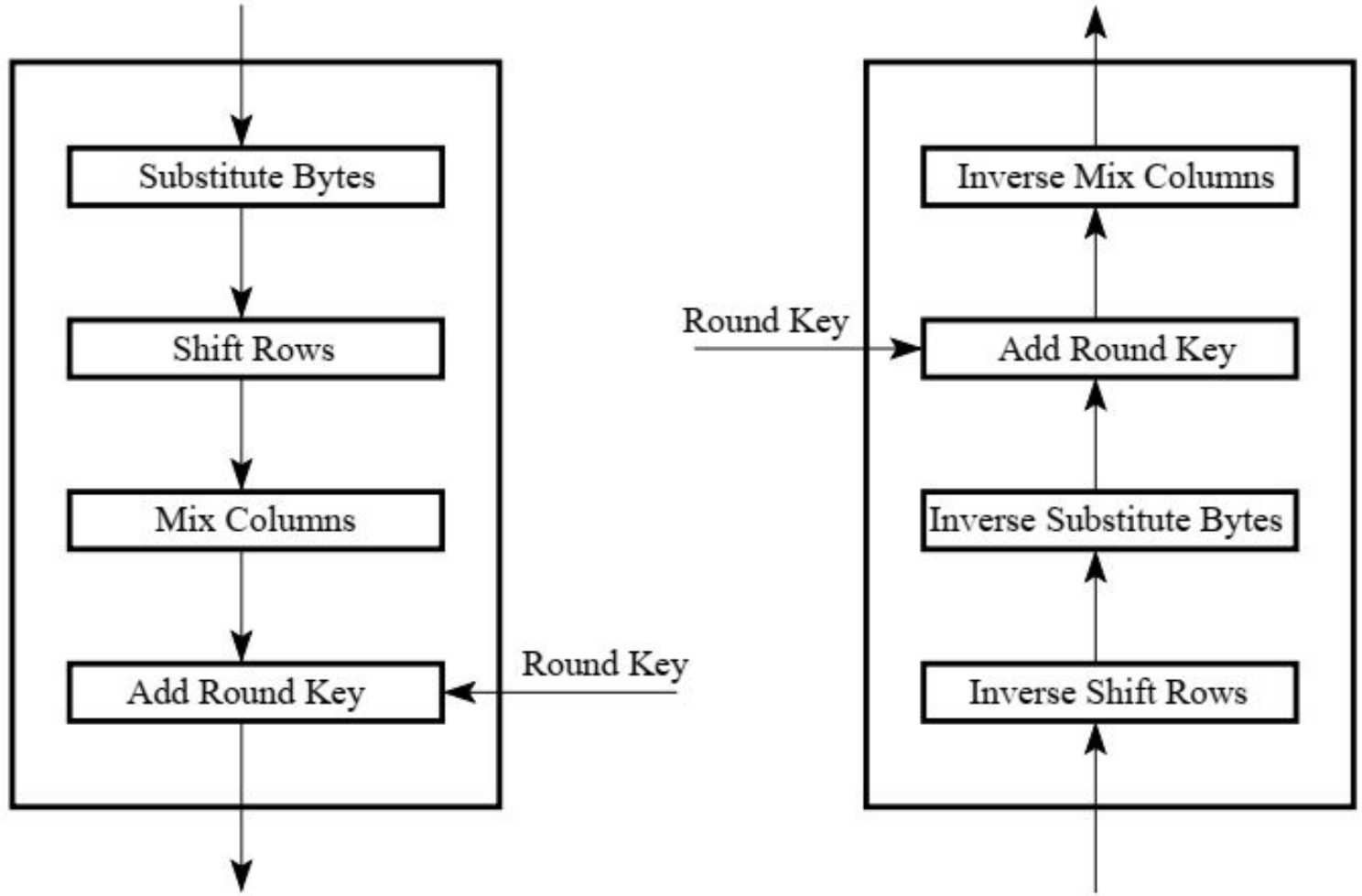


Figure 1 (Encryption on the left, Decryption on the right)


Key
Plaintext
Add round key
w(0,3)
Expand key
Round #1
Substitutive bytes
Shift row
Mix columns
Add round key
w(4,7)
Round # 9
Substitutive bytes
Shift row
Mix columns
Add round key
w(36,39)
Round # 10
Substitutive bytes
Shift row
Add round key
w(40,43)
Ciphertext
Plaintext
Round # 10
Add round key
Inverse sub bytes
Inverse shift row
Round # 9
Inverse mix cols.
Add round key
Inverse sub bytes
Inverse shift row
Round # 1
Inverse mix cols.
Add round key
Inverse sub bytes
Inverse shift row
Add round key
Ciphertext

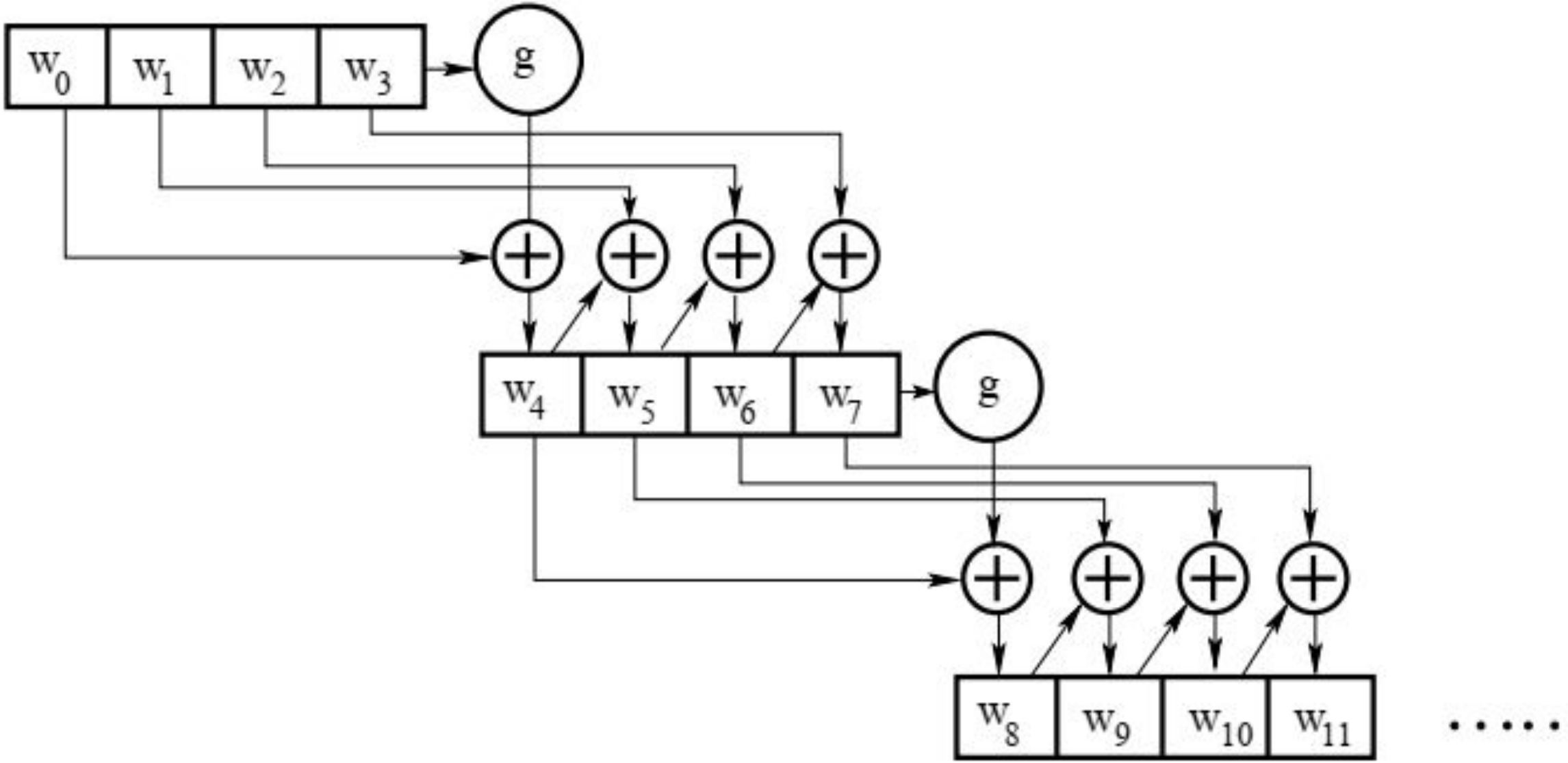

Substitute Bytes
Shift Rows
Mix Columns
Add Round Key
Round Key
Encryption Round
Inverse Mix Columns
Round Key
Add Round Key
Inverse Substitute Bytes
Inverse Shift Rows
Decryption Round

Analysis of Steps: Key Expansion

- **Key Expansion:** In the key expansion process the given 128 bits cipher key is stored in a [4]x[4] byte matrix (16*8=128 bits) and then the four column words of the key matrix is expanded into a schedule of 44 words (44*4=176 bytes) resulting in 11 round keys.

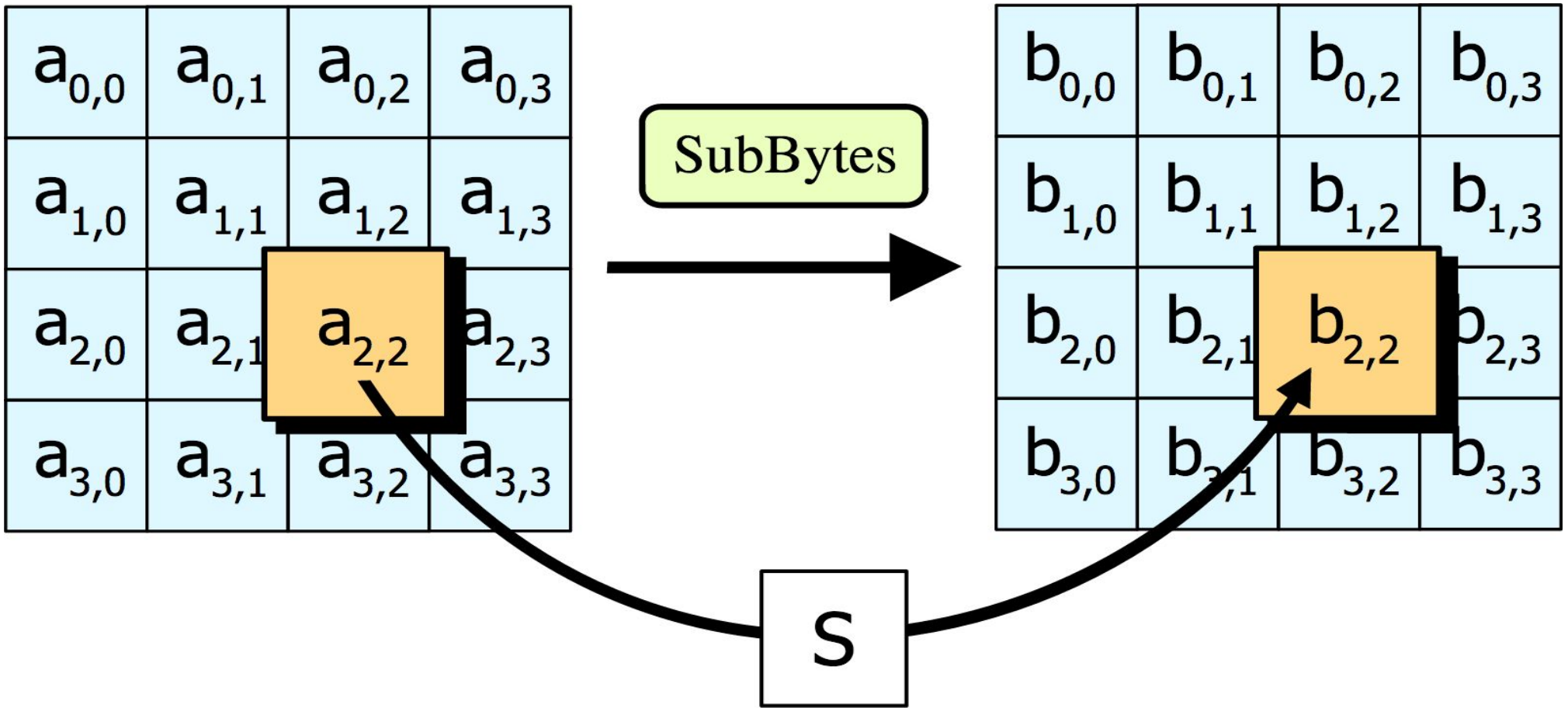
- Number of round keys = N_r + 1. Where N_r is the number of rounds (which is 10 in case of 128 bits key size). So, here, number round keys = 11.

Analysis of Steps: Key Expansion



Analysis of Steps: Substitute Bytes

SubBytes: Each element of the matrix is replaced by an element of the S-box matrix.



Analysis of Steps: Substitute Bytes

SubBytes: Each element of the matrix is replaced by an element of the S-box matrix.

- The S-box is a special lookup table which is constructed from Galois fields.
- The Generating function used in this algorithm is GF(2^8).
  - i.e., 256 values are possible
- The elements of the S-box are written in hexadecimal system.

Analysis of Steps: Substitute Bytes

AES S-box

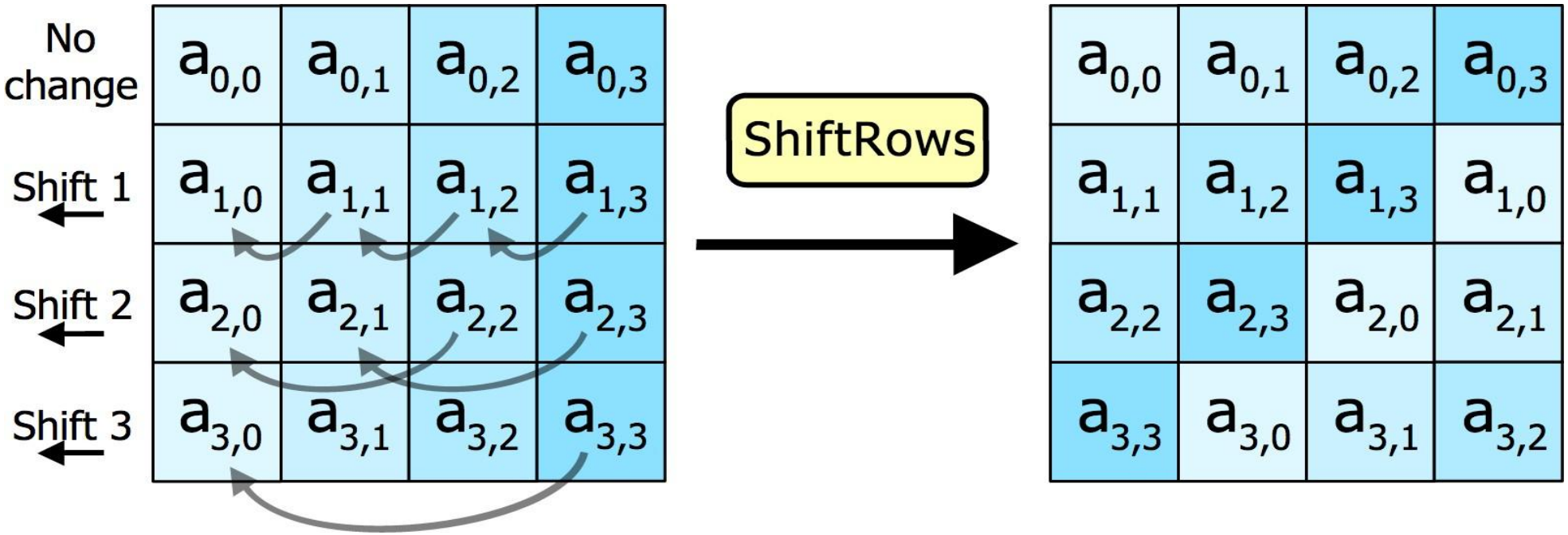
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Inverse S-box

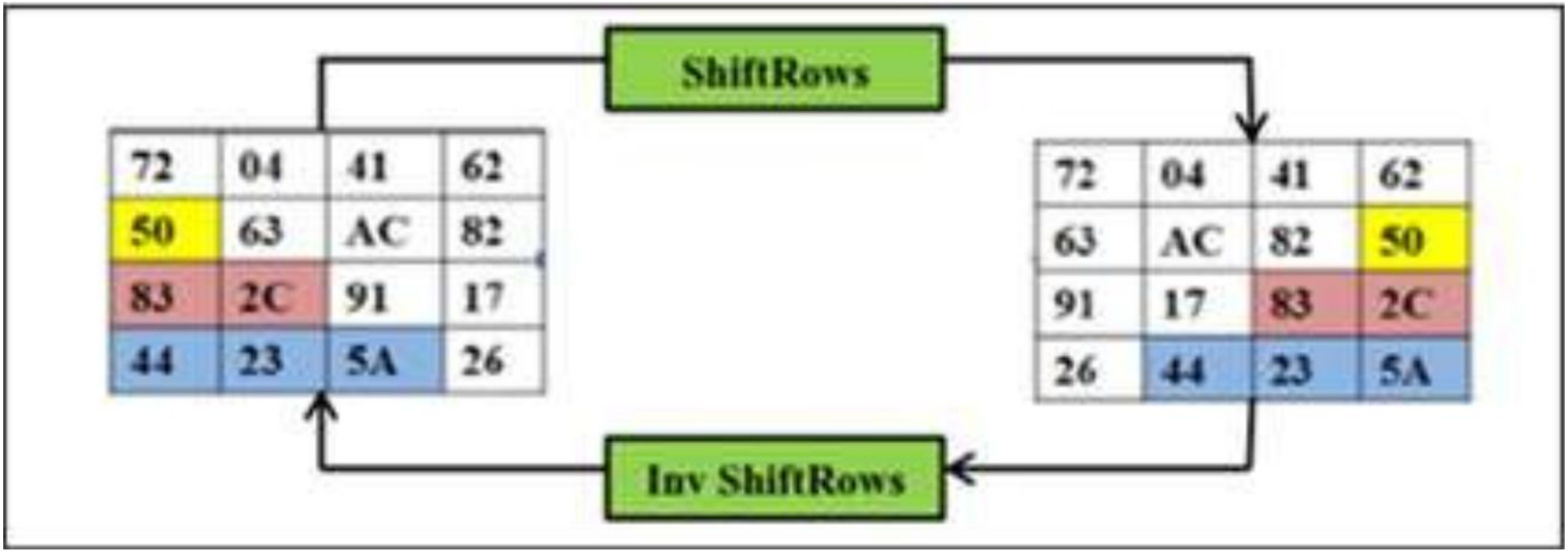
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Analysis of Steps: Shift Rows

Shift Rows: In this step rows of the block are cylindrically shifted in left/right direction. The first row is untouched, the second one is shift by one, third one by two and the fourth one by three.



Analysis of Steps: Shift Rows



Analysis of Steps: Mix columns

Mix columns: This is the most important part of the algorithm. It causes the flip of bits to spread all over the block. In this step the block is multiplied with a fixed matrix. The multiplication is a field multiplication in galois field.

For each row there are 16 multiplication, 12 XORs and a 4 byte output.

Analysis of Steps: Mix columns

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} s_{0.0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1.0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2.0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3.0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0.0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1.0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2.0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3.0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Analysis of Steps: Mix columns

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

$$\begin{aligned} s'_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\ s'_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j}) \end{aligned}$$

Predefine Matrix **State Array** **New State Array**

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

$$\begin{aligned} &(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\} \\ &\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\} \\ &\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\} \\ &(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\} \end{aligned}$$

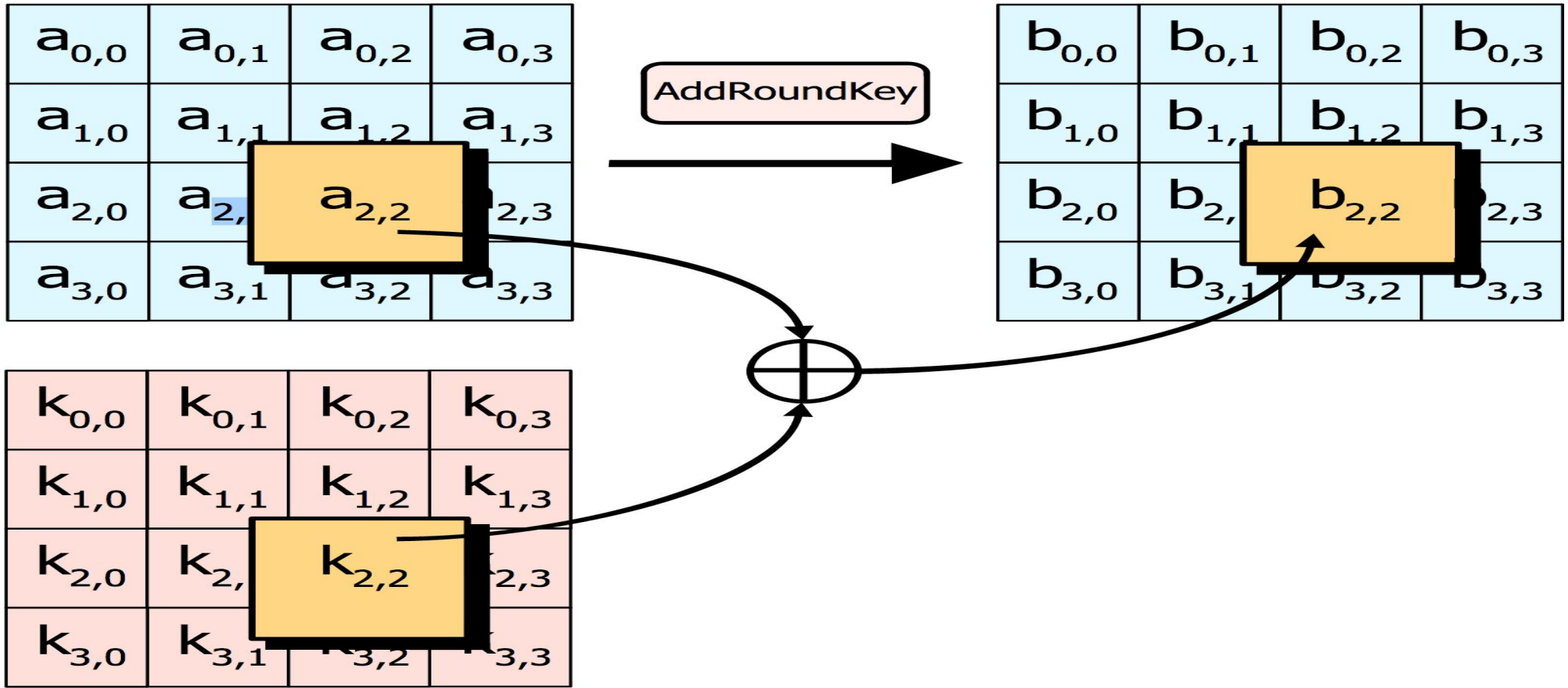
47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Analysis of Steps: Mix columns

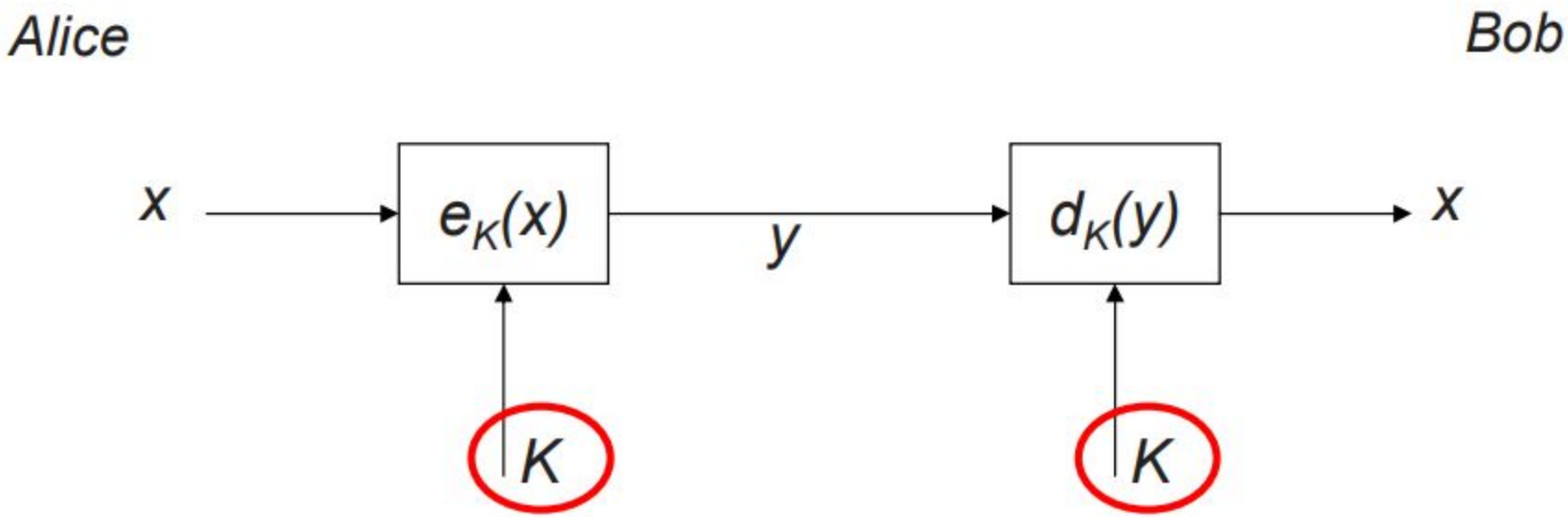
$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} s_{0.0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1.0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2.0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3.0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0.0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1.0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2.0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3.0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Analysis of Steps: Add round key

Add round key: In this step, each byte is XOR-ed with corresponding element of the key matrix.



Symmetric Cryptography revisited



Two properties of symmetric (secret-key) crypto-systems:

- The **same secret key *K*** is used for encryption and decryption
- Encryption and Decryption are very similar (or even identical) functions

Symmetric Cryptography: Shortcomings

- Symmetric algorithms, e.g., AES or 3DES, are very secure, fast & widespread **but**:

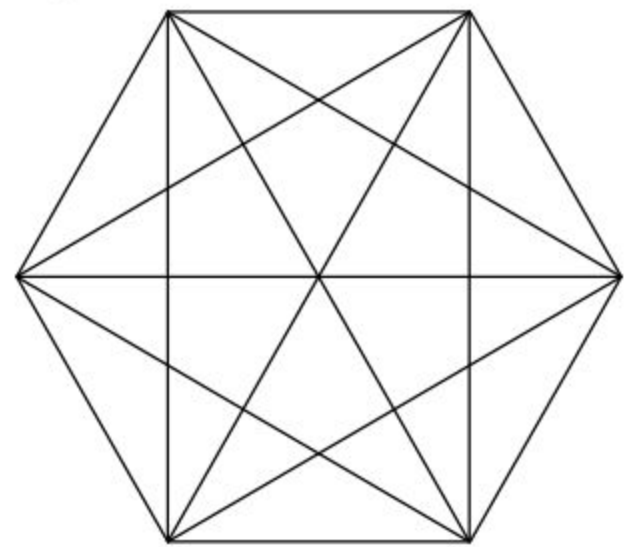
- Key distribution problem: The secret key must be **transported securely**
- Number of keys: In a network, each pair of users requires an individual key

→ *n* users in the network require $\frac{n \cdot (n-1)}{2}$ keys, each user stores *(n-1)* keys

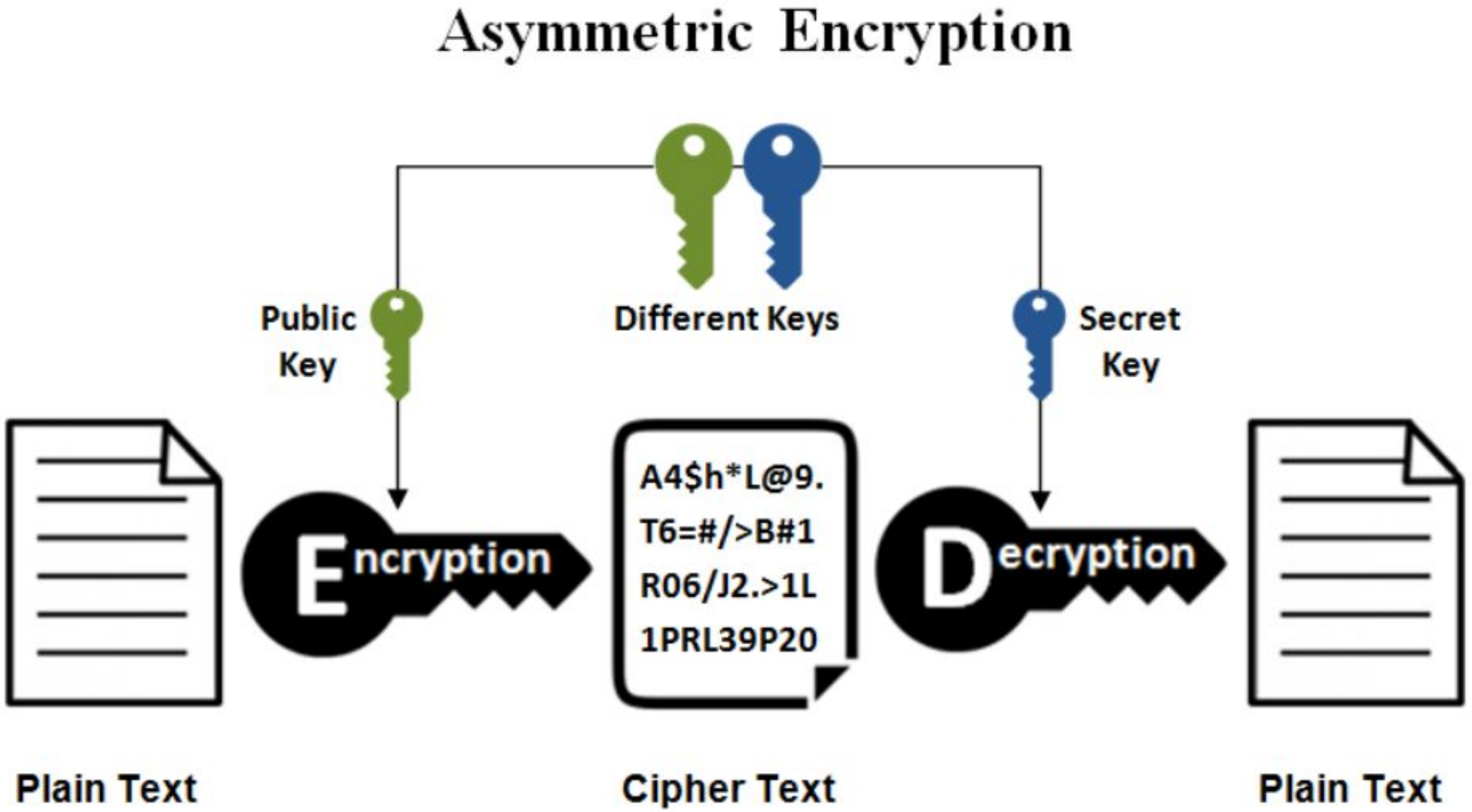
Example:

6 users (nodes)

$\frac{6 \cdot 5}{2} = 15$ keys (edges)

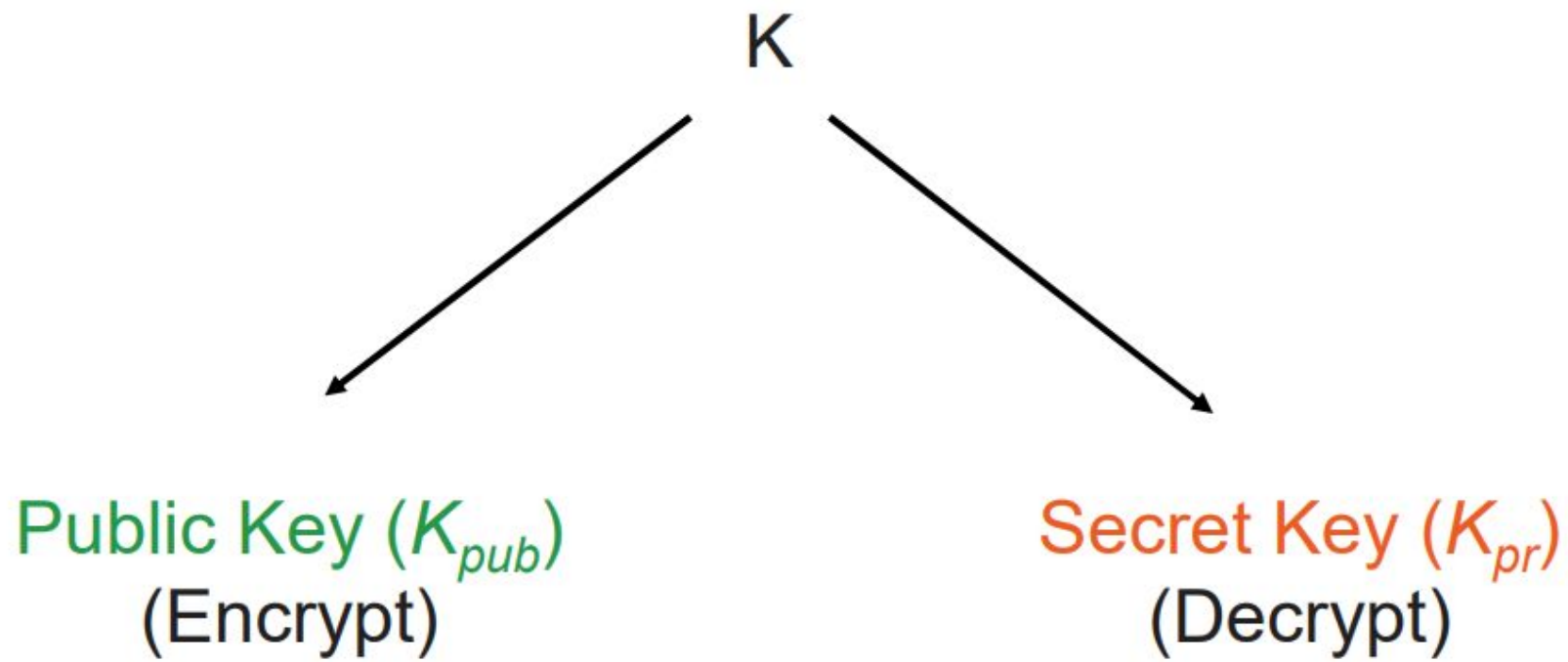


Asymmetric Cryptography



Asymmetric (Public-Key) Cryptography

Principle: “Split up” the key



→ During the key generation, a key pair K_{pub} and K_{pr} is computed

Security Mechanisms of Public-Key Cryptography

Here are main mechanisms that can be realized with asymmetric cryptography:

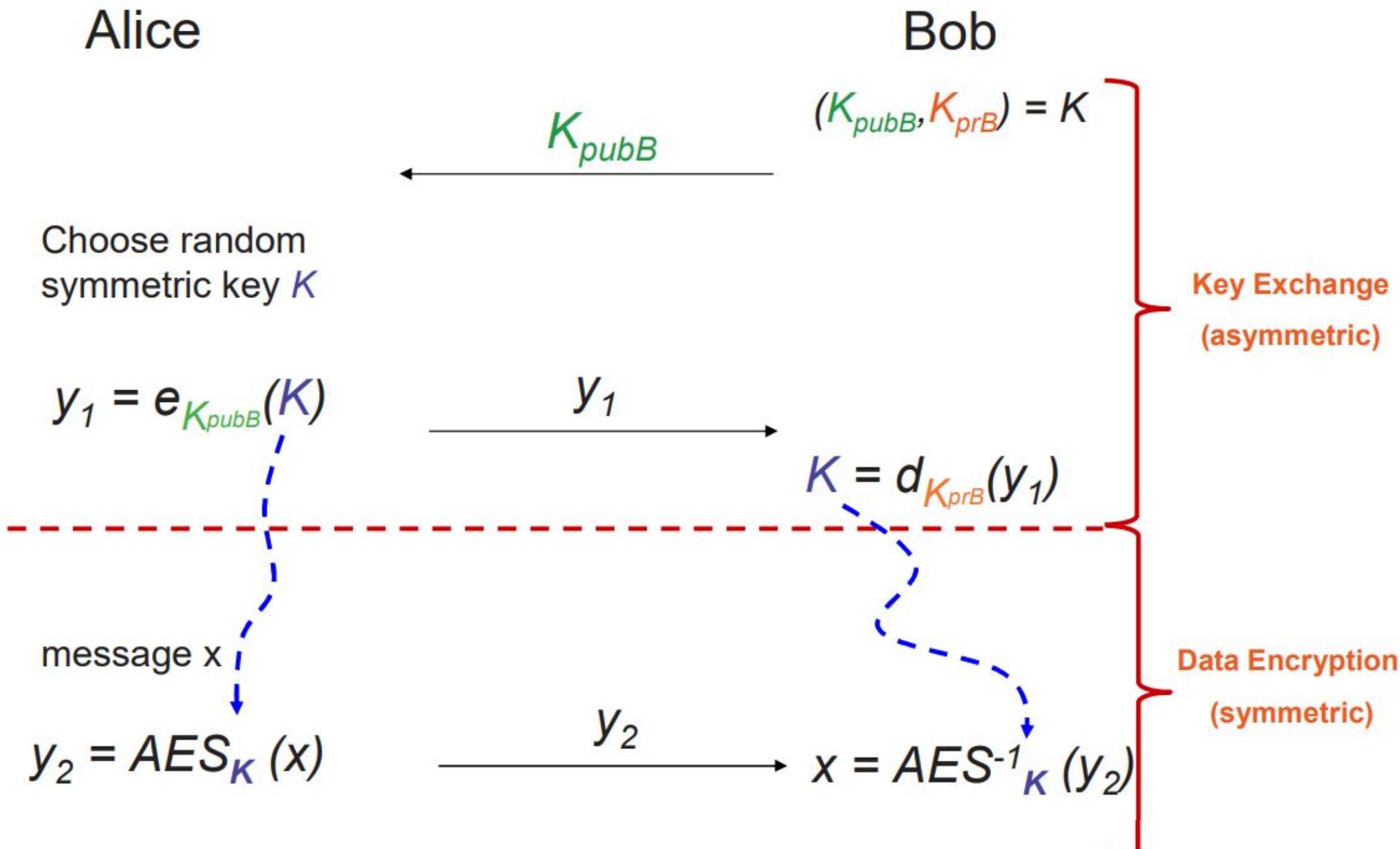
- **Key Distribution** (e.g., Diffie-Hellman key exchange, RSA) without a pre-shared secret (key)
- **Nonrepudiation and Digital Signatures** (e.g., RSA, DSA or ECDSA) to provide message integrity
- **Identification**, using challenge-response protocols with digital signatures
- **Encryption** (e.g., RSA / Elgamal)
  Disadvantage: Computationally very intensive
  (1000 times slower than symmetric Algorithms!)

In practice: **Hybrid systems**, incorporating asymmetric and symmetric algorithms

1. **Key exchange** (for symmetric schemes) and **digital signatures** are performed with (slow) **asymmetric** algorithms

2. **Encryption** of data is done using (fast) symmetric ciphers, e.g., **block ciphers or stream ciphers**

Example: Hybrid protocol with AES as the symmetric cipher



How to build Public-Key Algorithms

Asymmetric schemes are based on a **„one-way function“** ***f()***:

- Computing $y = f(x)$ is computationally easy
- Computing $x = f^{-1}(y)$ is computationally infeasible

One way functions are based on **mathematically hard problems**.

Three main families:

- **Factoring integers** (RSA, ...):
  Given a composite integer *n*, find its prime factors
  (Multiply two primes: easy)

- **Discrete Logarithm** (Diffie-Hellman, Elgamal, DSA, …):
  Given *a, y* and *m,* find *x* such that $a^x = y \bmod m$
  (Exponentiation a^x : easy)

- **Elliptic Curves (EC)** (ECDH, ECDSA): Generalization of discrete logarithm

Note: The problems are considered mathematically hard, but no proof exists (so far).

Key Lengths and Security Levels

Symmetric	*ECC*	*RSA, DL*	*Remark*
64 Bit	128 Bit	≈ 700 Bit	Only short term security (a few hours or days)
80 Bit	160 Bit	≈ 1024 Bit	Medium security (except attacks from big governmental institutions etc.)
128 Bit	256 Bit	≈ 3072 Bit	Long term security (without quantum computers)

The RSA Cryptosystem

- Martin Hellman and Whitfield Diffie published their landmark public-key paper in 1976
- Ronald Rivest, Adi Shamir and Leonard Adleman proposed the asymmetric RSA cryptosystem in1977
- Until now, RSA is the most widely use asymmetric cryptosystem although elliptic curve cryptography (ECC) becomes increasingly popular
- RSA is mainly used for two applications
  - Transport of (i.e., symmetric) keys
  - Digital signatures

Key Generation

- Like all asymmetric schemes, RSA has set-up phase during which the private and public keys are computed

Algorithm: RSA Key Generation

Output: public key: $k_{pub} = (n, e)$ and private key $k_{pr} = d$

1. Choose two large primes p, q
2. Compute $n = p * q$
3. Compute $\Phi(n) = (p-1) * (q-1)$
4. Select the public exponent $e \in \{1, 2, \ldots, \Phi(n)-1\}$ such that $\gcd(e, \Phi(n)) = 1$
5. Compute the private key d such that $d * e \equiv 1 \bmod \Phi(n)$
6. **RETURN** $k_{pub} = (n, e)$, $k_{pr} = d$

Remarks:

- Choosing two large, distinct primes p, q (in Step 1) is non-trivial
- $\gcd(e, \Phi(n)) = 1$ ensures that e has an inverse and, thus, that there is always a private key d

Example: RSA with small numbers

ALICE

Message $x = 4$

BOB

1. Choose $p = 3$ and $q = 11$
2. Compute $n = p * q = 33$
3. $\Phi(n) = (3-1) * (11-1) = 20$
4. Choose $e = 3$
5. $d \equiv e^{-1} \equiv 7 \bmod 20$

$K_{pub} = (33,3)$ ←

$y = x^e \equiv 4^3 \equiv 31 \bmod 33$

$y = 31$ →

$y^d = 31^7 \equiv 4 = x \bmod 33$

Implementation aspects

- The RSA cryptosystem uses only one arithmetic operation (modular exponentiation) which makes it conceptually a simple asymmetric scheme
- Even though conceptually simple, due to the use of very long numbers, RSA is orders of magnitude slower than symmetric schemes, e.g., DES, AES
- When implementing RSA (esp. on a constrained device such as smartcards or cell phones) close attention has to be paid to the correct choice of arithmetic algorithms

RSA is typically exposed to these analytical attack vectors

- **Mathematical attacks**
  - The best known attack is factoring of n in order to obtain $\Phi(n)$
  - Can be prevented using a sufficiently large modulus n
  - The current factoring record is 664 bits. Thus, it is recommended that n should have a bit length between 1024 and 3072 bits

- RSA is the most widely used public-key cryptosystem
- RSA is mainly used for key transport and digital signatures
- The public key *e* can be a short integer, the private key *d* needs to have the full length of the modulus *n*
- RSA relies on the fact that it is hard to factorize *n*
- Currently 1024-bit cannot be factored, but progress in factorization could bring this into reach within 10-15 years. Hence, RSA with a 2048 or 3076 bit modulus should be used for long-term security
- A naïve implementation of RSA allows several attacks, and in practice RSA should be used together with padding

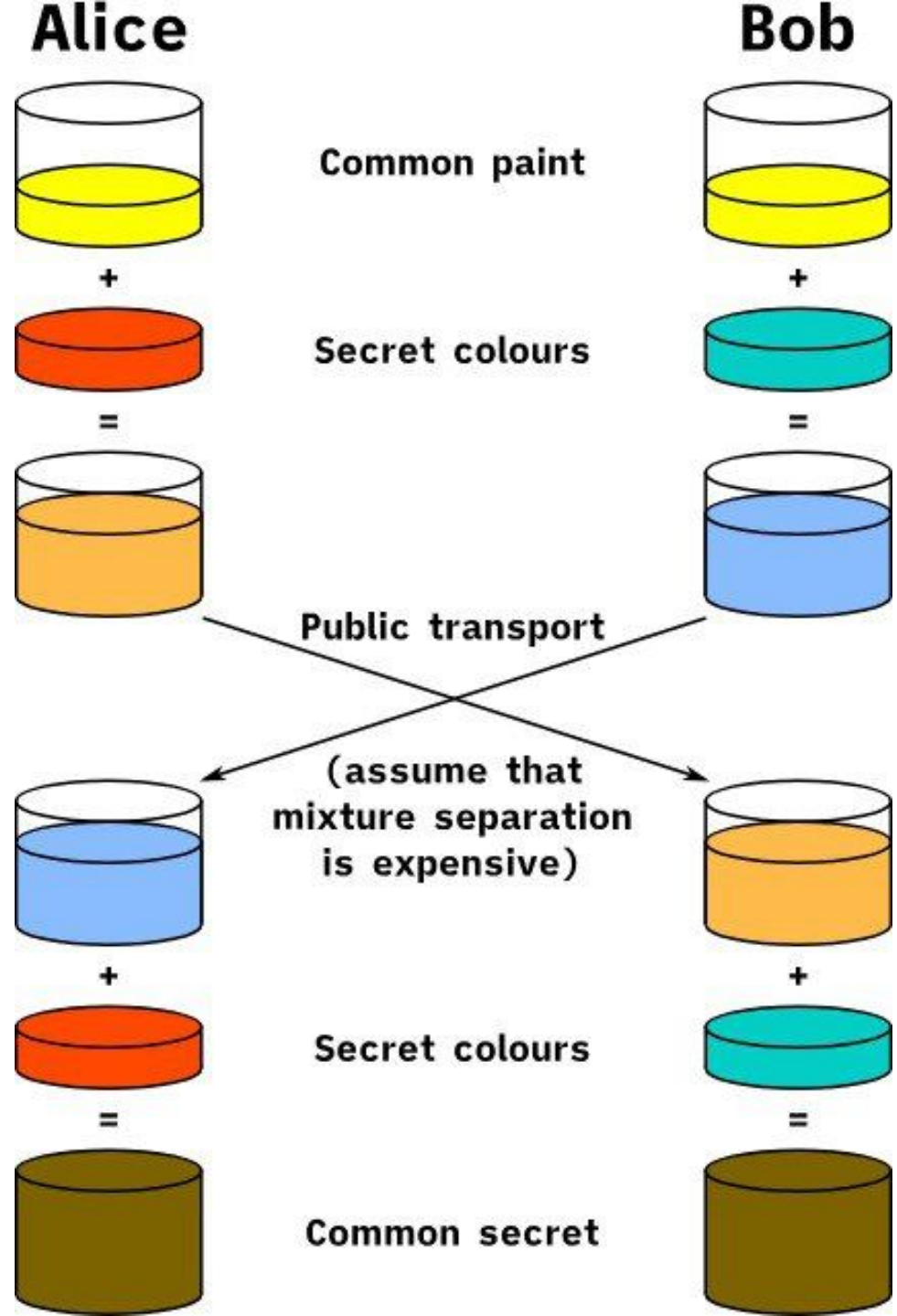
Diffie–Hellman Key Exchange: Overview

- Proposed in 1976 by **Whitfield Diffie and Martin Hellman**
- **Widely used**, e.g. in Secure Shell (SSH), Transport Layer Security (TLS), and Internet Protocol Security (IPSec)
- The Diffie–Hellman Key Exchange (DHKE) is a key exchange protocol and **not** used for encryption

  (For the purpose of encryption based on the DHKE, ElGamal can be used.)

Diffie–Hellman Key Exchange: Set-up

1. Choose a large prime *p*.
2. Choose an integer α ∈ {2,3, . . . , *p*−2}.
3. Publish *p and* α.



Diffie–Hellman Key Exchange

Alice		Bob
Choose random private key $k_{prA}=a \in \{1,2,\ldots,p\text{-}1\}$		Choose random private key $k_{prB}=b \in \{1,2,\ldots,p\text{-}1\}$
Compute corresponding public key $k_{pubA}= A = \alpha^a \bmod p$	$A \rightarrow$	
	$\leftarrow B$	Compute correspondig public key $k_{pubB}= B = \alpha^b \bmod p$
Compute common secret $k_{AB} = B^a = (\alpha^a)^b \bmod p$		Compute common secret $k_{AB} = A^b = (\alpha^b)^a \bmod p$

We can now use the joint key k_{AB} for encryption, e.g., with AES

$y = AES_{kAB}(x)$ $\xrightarrow{y}$ $x = AES^{-1}_{kAB}(y)$

Diffie–Hellman Key Exchange: Example

Domain parameters p=29, α=2

Alice		Bob
Choose random private key $k_{prA} = a = 5$		Choose random private key $k_{prB} = b = 12$
Compute corresponding public key $k_{pubA} = A = 2^5 = 3 \bmod 29$	A →	
	← B	Compute correspondig public key $k_{pubB} = B = 2^{12} = 7 \bmod 29$
Compute common secret $k_{AB} = B^a = 7^5 = 16 \bmod 29$		Compute common secret $k_{AB} = A^b = 3^{12} = 16 \bmod 29$

Proof of correctness:

Alice computes: $B^a = (\alpha^b)^a \bmod p$
Bob computes: $A^b = (\alpha^a)^b \bmod p$

i.e., Alice and Bob compute the same key k_{AB}!

The Discrete Logarithm Problem

Discrete Logarithm Problem (DLP) in Z_p^*

- Given is the finite cyclic group Z_p^* of order $p-1$ and a primitive element $\alpha \in Z_p^*$ and another element $\beta \in Z_p^*$.
- The DLP is the problem of determining the integer $1 \leq x \leq p-1$ such that $\alpha^x \equiv \beta \bmod p$
- This computation is called the **discrete logarithm problem (DLP)**

$$x = log_\alpha \beta \bmod p$$

- Example: Compute x for $5^x \equiv 41 \bmod 47$

Attacks against the Discrete Logarithm Problem

Summary of records for computing discrete logarithms in Z_p^*

Decimal digits	Bit length	Date
58	193	1991
68	216	1996
85	282	1998
100	332	1999
120	399	2001
135	448	2006
160	532	2007

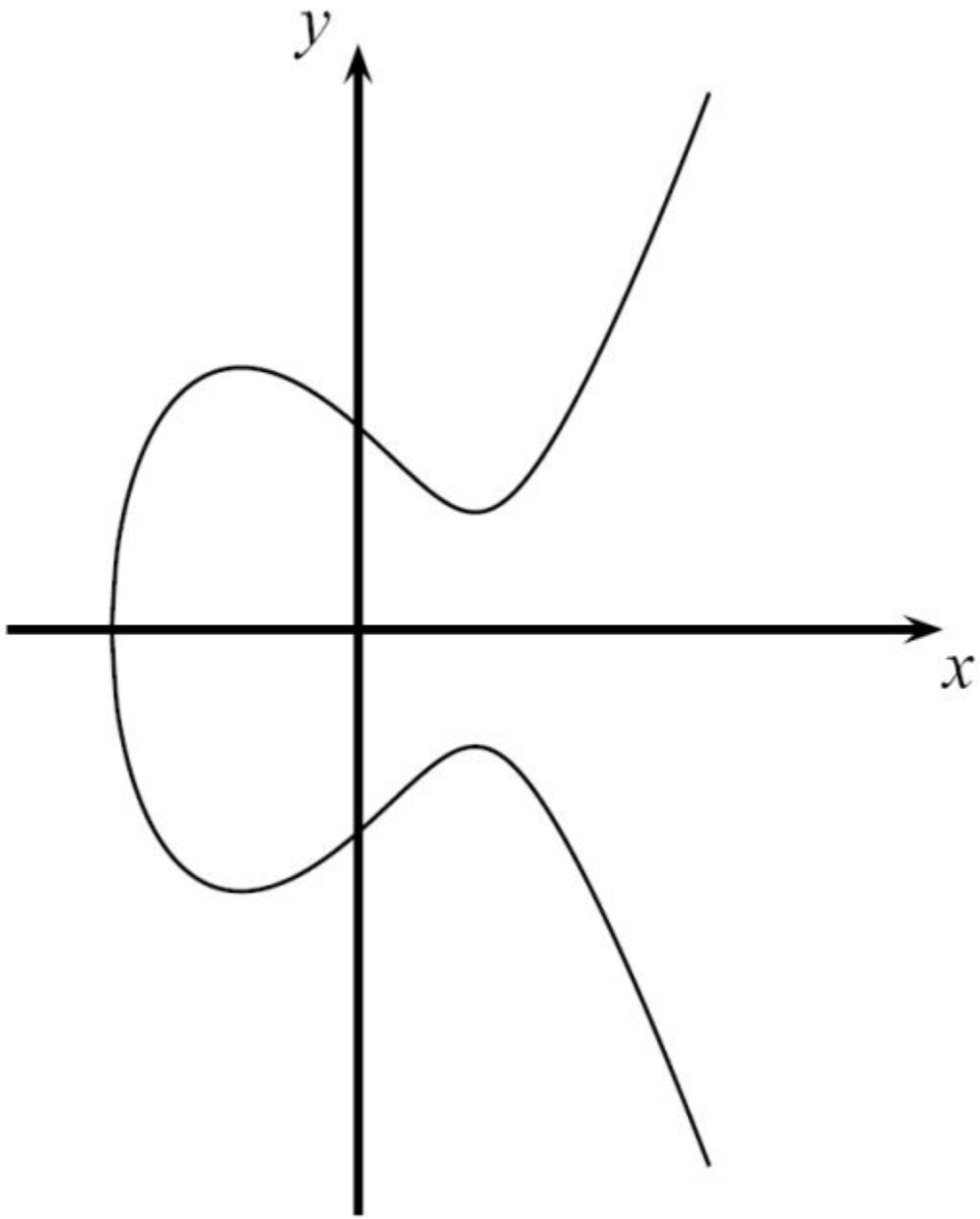
In order to prevent attacks that compute the DLP, it is recommended to use primes with a length of at least 1024 bits for schemes such as Diffie-Hellman in Z_p^*

Security of the classical Diffie–Hellman Key Exchange

- Which information does Oscar have?
  - α, p
  - $k_{pubA} = A = \alpha^a \bmod p$
  - $k_{pubB} = B = \alpha^b \bmod p$
- Which information does Oscar want to have?
  - $k_{AB} = \alpha^{ba} = \alpha^{ab} = \bmod p$
  - This is kown as Diffie-Hellman Problem (DHP)
- The only known way to solve the DHP is to solve the DLP, i.e.
  1. Compute $a = log_\alpha A \bmod p$
  2. *Compute* $k_{AB} = B^a = \alpha^{ba} = \bmod p$

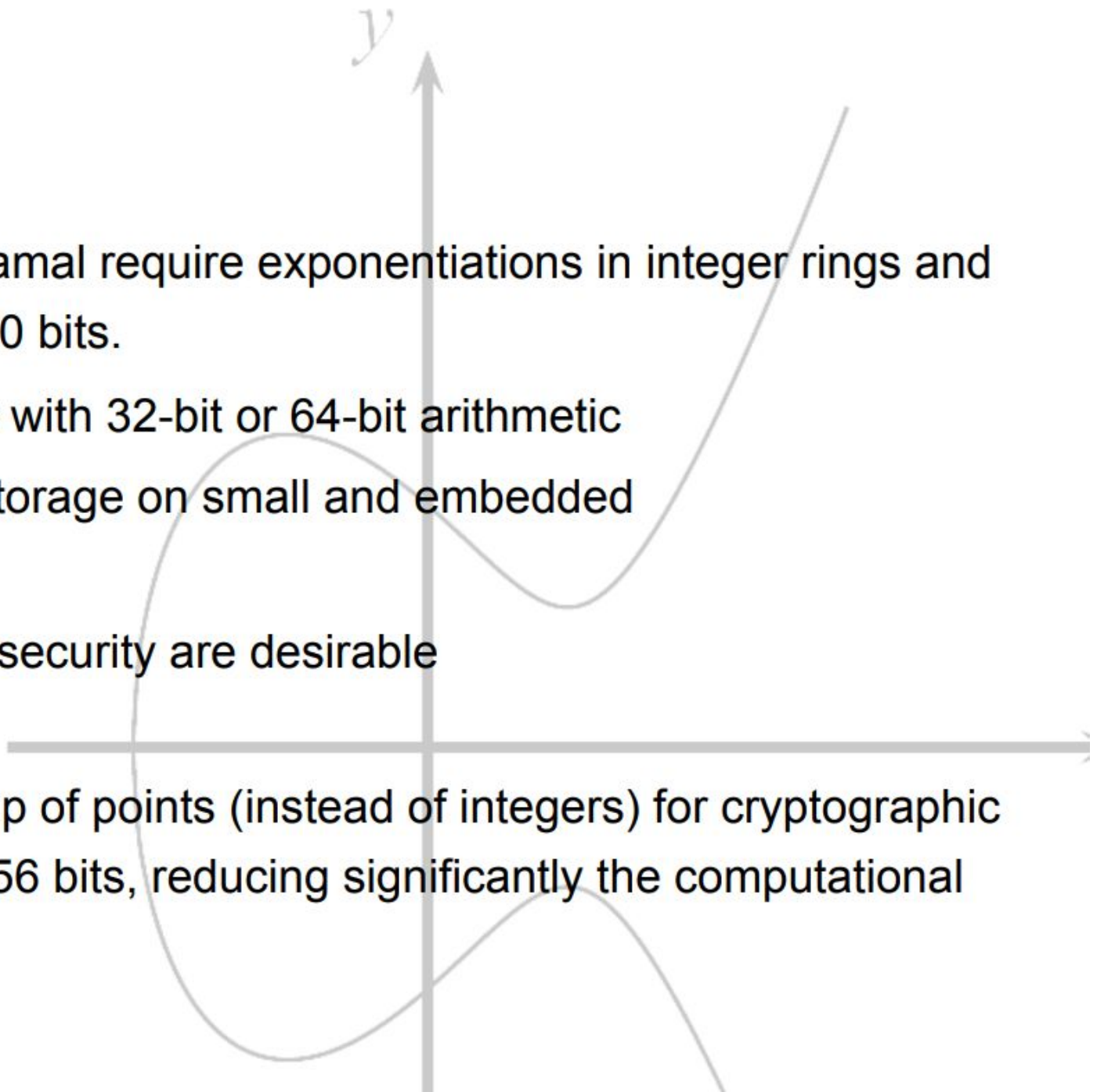
  It is conjectured that the DHP and the DLP are equivalent, i.e., solving the DHP implies solving the DLP.
- To prevent attacks, i.e., to prevent that the DLP can be solved, choose $p > 2^{1024}$

- The Diffie–Hellman protocol is a widely used method for key exchange. It is based on cyclic groups.
- The discrete logarithm problem is one of the most important one-way functions in modern asymmetric cryptography. Many public-key algorithms are based on it.
- For the Diffie–Hellman protocol in Z_p^*, *the prime p should be at least 1024 bits* long. This provides a security roughly equivalent to an 80-bit symmetric cipher.
- For a better long-term security, a prime of length 2048 bits should be chosen.



Elliptic Curve Cryptography

Motivation



- **Problem:**
  Asymmetric schemes like RSA and Elgamal require exponentiations in integer rings and fields with parameters of more than 1000 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 160-256 bits, reducing significantly the computational effort.

Computations on Elliptic Curves

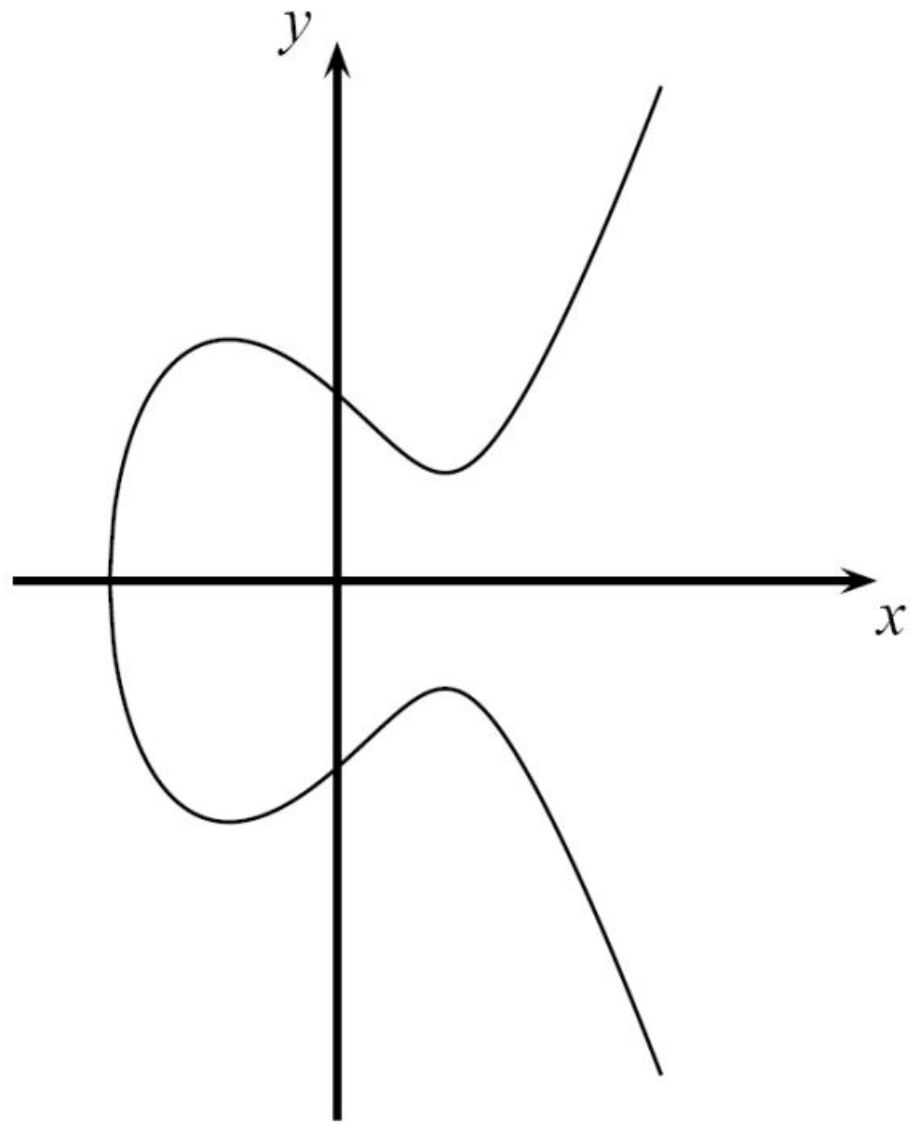
- Elliptic curves are polynomials that define points based on the (simplified) Weierstraß equation:

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

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.)

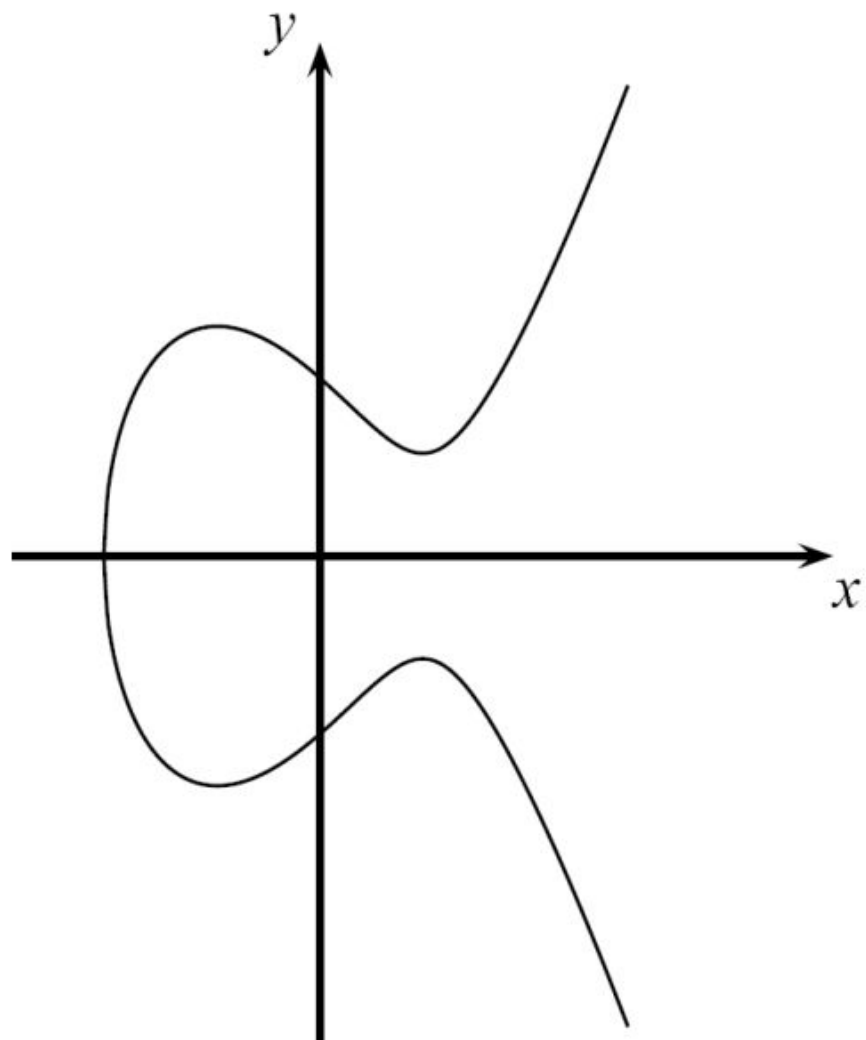
- In cryptography, we are interested in elliptic curves module a prime *p*:

> ***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 of 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

We will consider only those elliptic curves that have no multiple roots - which is equivalent to the condition

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



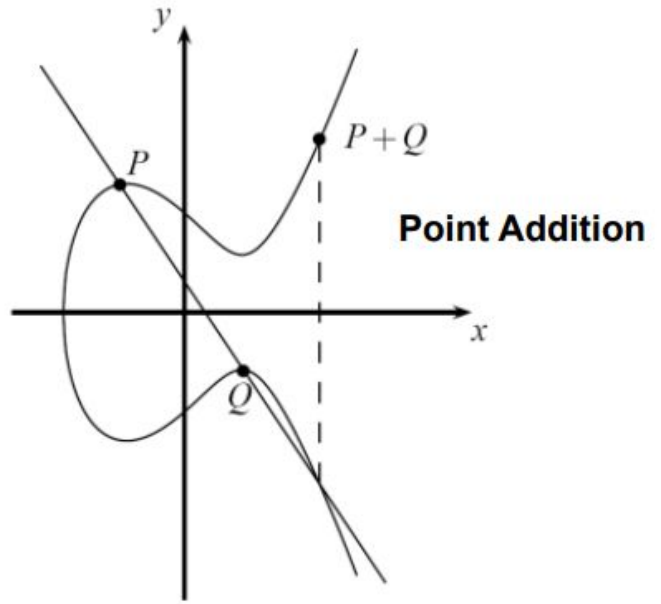
Computations on Elliptic Curves (ctd.)

- 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

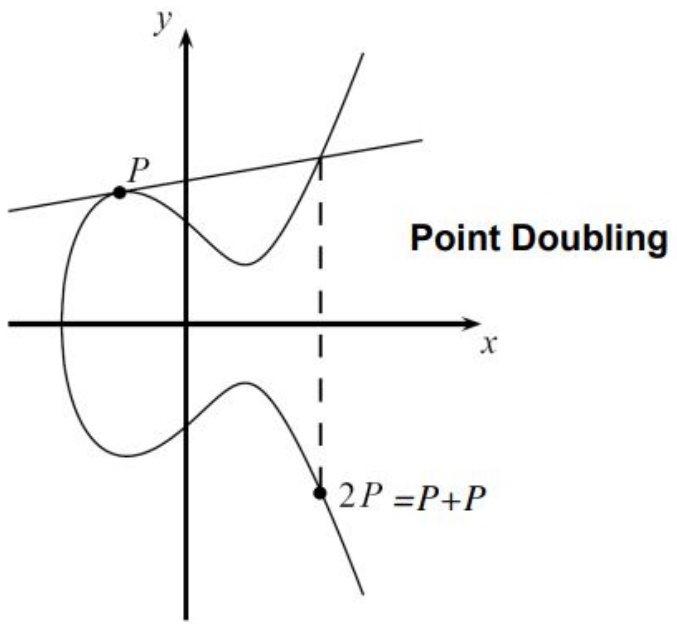
$x_3 = s^2 - x_1 - x_2 \bmod p$ and $y_3 = s(x_1 - x_3) - y_1 \bmod p$

where

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



Point Addition



Point Doubling

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

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

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

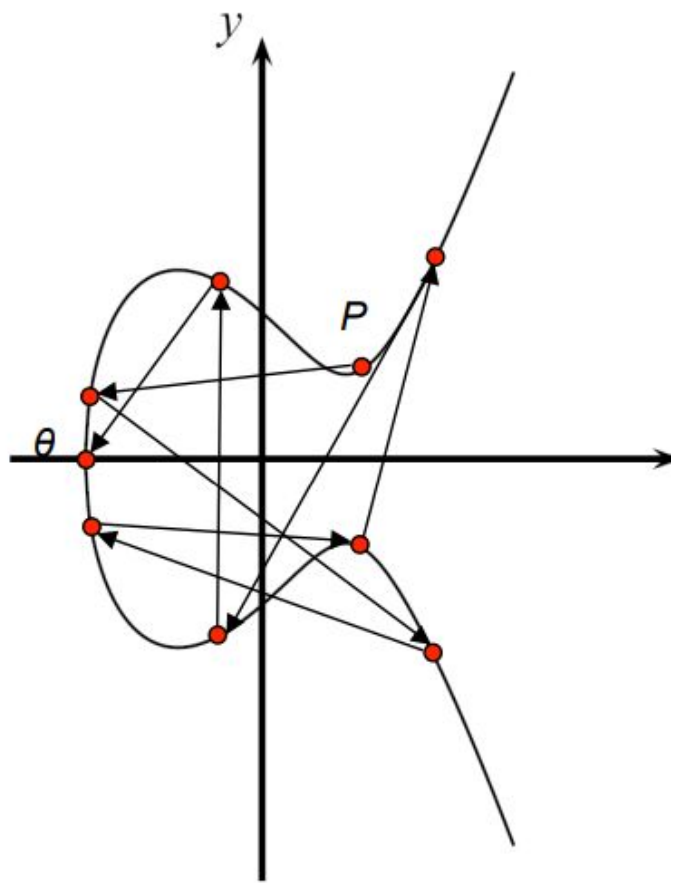
$$x_3 = s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 159 \equiv 6 \bmod 17$$

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

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

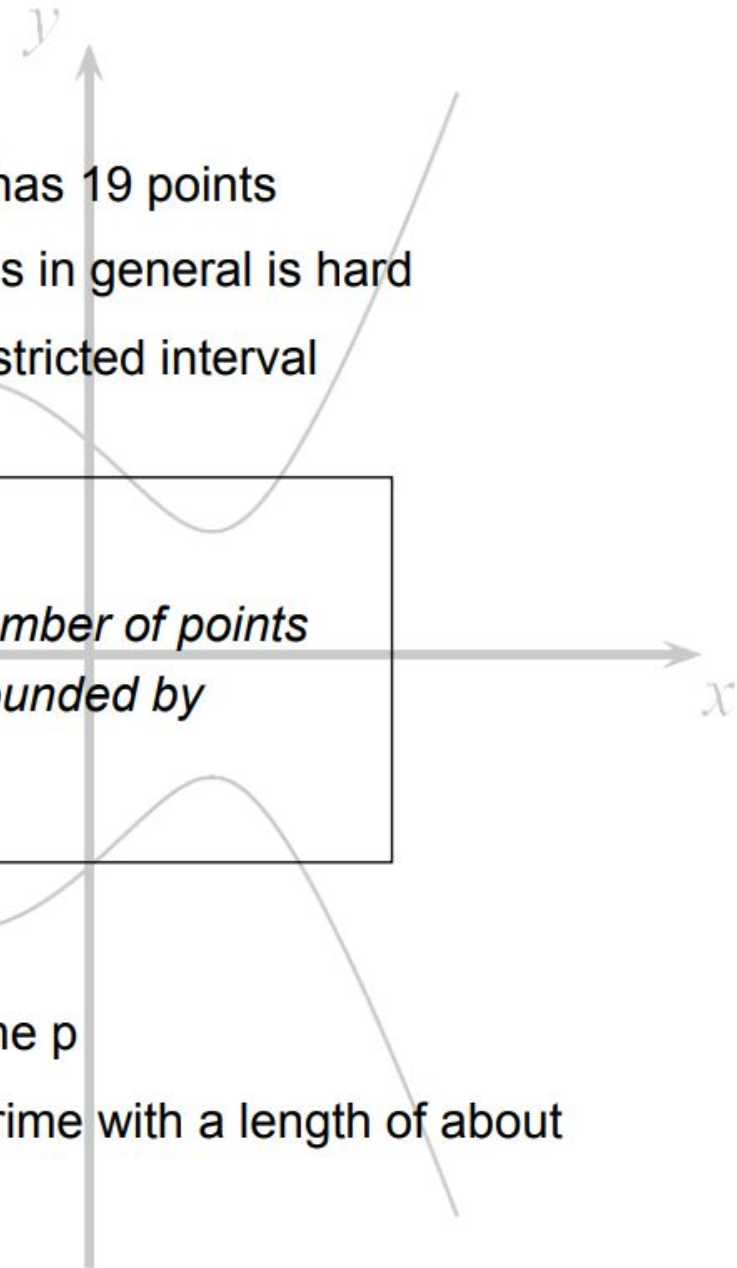
- The points on an elliptic curve and the point at infinity θ form cyclic subgroups

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



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:* $y^2 = x^3+2x+2 \ mod \ 17$ 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^{160} points, a prime with a length of about 160 bits is required

How to calculate 26P?

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 $d_i = 1$

    $T = T + P \bmod n$

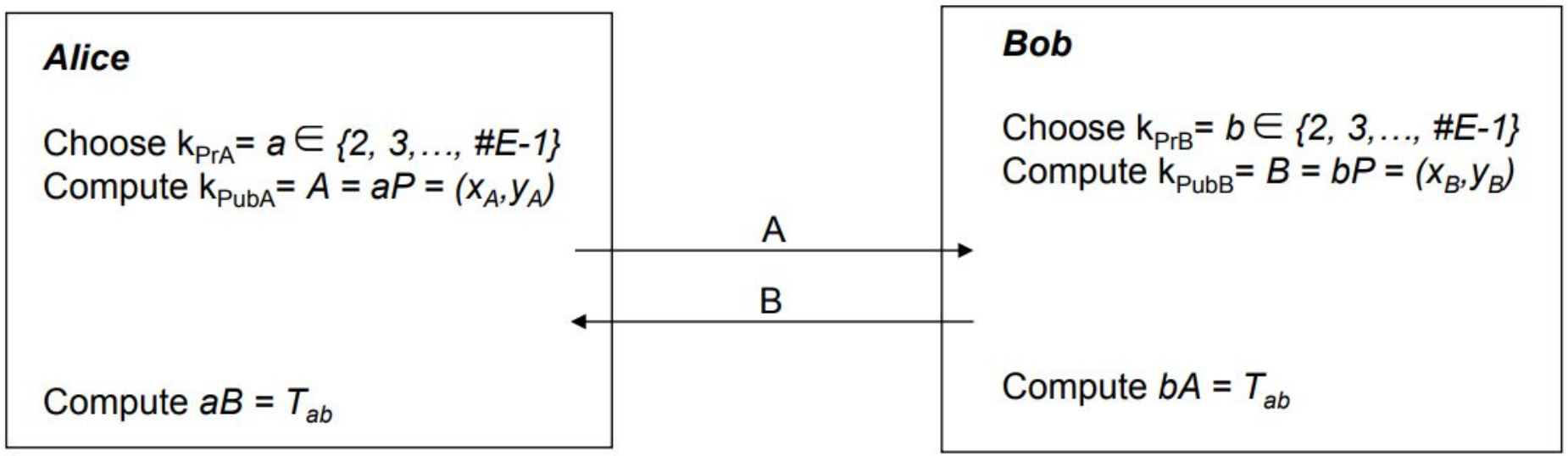
RETURN (T)

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 ($d_0 = 0$)

The Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

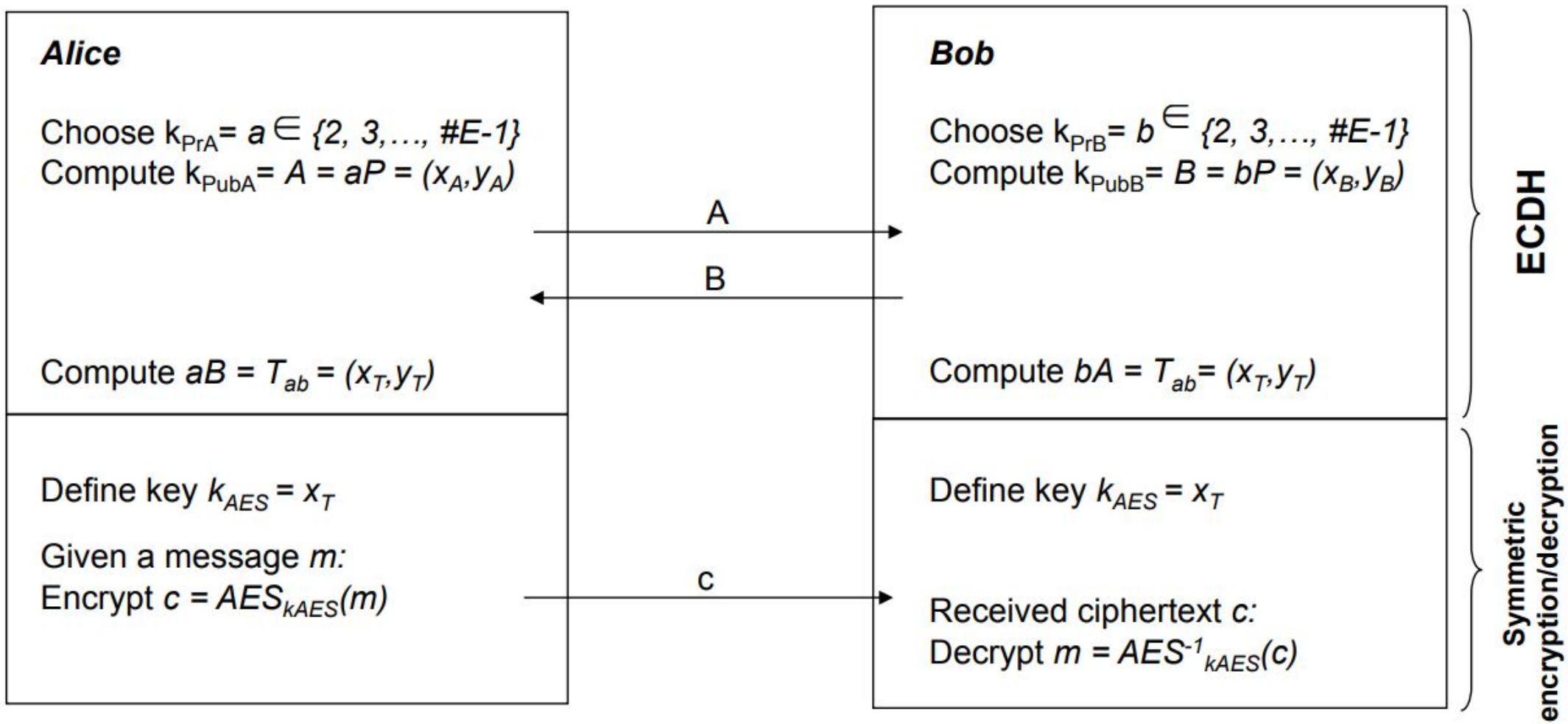
- Given a prime *p*, a suitable elliptic curve *E* and a point $P=(x_P,y_P)$
- The Elliptic Curve Diffie-Hellman Key Exchange is defined by the following protocol:



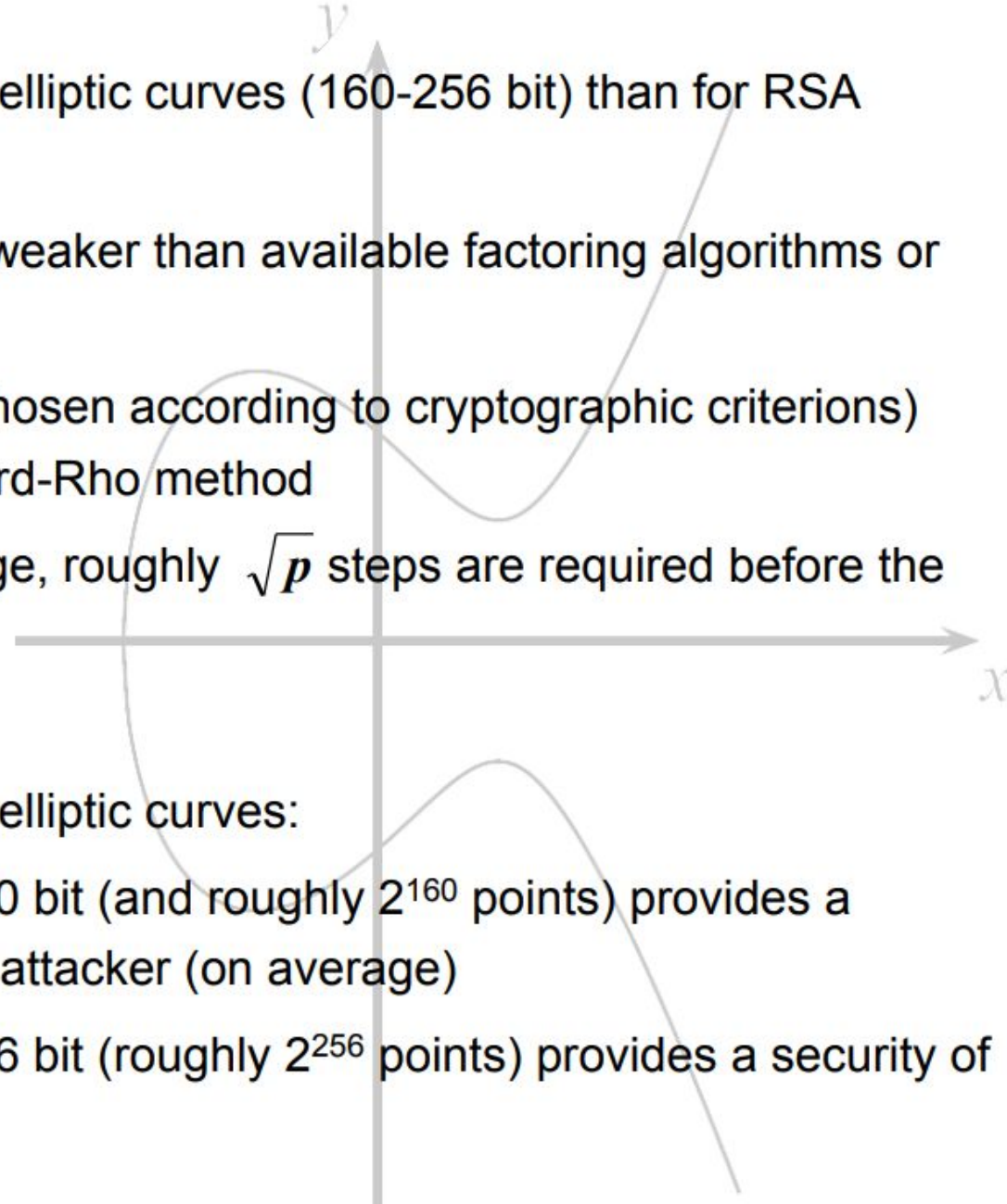
- Joint secret between Alice and Bob: $T_{AB} = (x_{AB}, y_{AB})$
- Proof for correctness:
  - Alice computes aB=a(bP)=abP
  - Bob computes bA=b(aP)=abP since group is associative
- One of the coordinates of the point T_{AB} (usually the x-coordinate) can be used as session key (often after applying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



Security Aspects



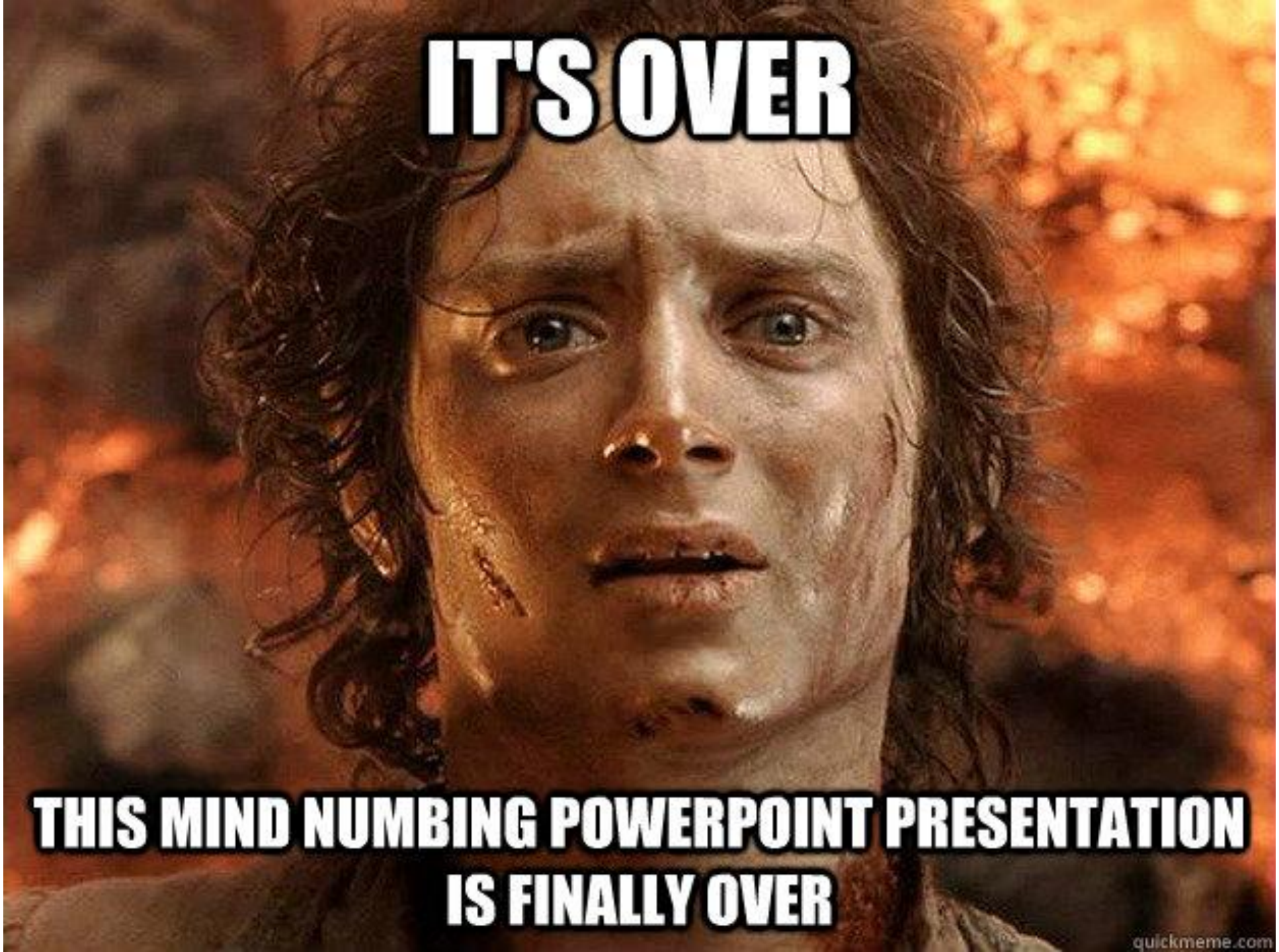
- Why are parameters signficantly smaller for elliptic curves (160-256 bit) than for RSA (1024-3076 bit)?
  - Attacks on groups of elliptic curves are weaker than available factoring algorithms or integer DL attacks
  - Best known attacks on elliptic curves (chosen according to cryptographic criterions) 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 160 bit (and roughly 2^{160} points) provides a security of 2^{80} steps that required by an attacker (on average)
  - An elliptic curve using a prime p with 256 bit (roughly 2^{256} points) provides a security of 2^{128} steps on average

- 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.

Links to follow

- https://crypto-textbook.com/slides.php
- https://www.youtube.com/watch?v=Q4X4OZTwftg&list=PLxP0p--aBHmIAeOBX1lkpTn-wAbAimg8-
- https://www.youtube.com/watch?v=NF1pwjL9-DE&t=1s
- https://www.youtube.com/watch?v=NmM9HA2MQGI


IT'S OVER
THIS MIND NUMBING POWERPOINT PRESENTATION IS FINALLY OVER
quickmeme.com