Introduction to Cryptography

m0leCon 2023 Workshops

Contacts

Discord:	@rising0

Other resources

BOOKS:

- *A Graduate Course in Applied Cryptography* (D. Boneh, V. Shoup)
- *Real World Cryptography* (D. Wong)

MORE CHALLENGES:

- **CryptoHack**

USEFUL TOOL:

- **Cyberchef** (basically magic for lazy people)

Where is cryptography?

Nowadays cryptography is found **anywhere**

- Internet communications (SSL, HTTPS...)
- Mobile networks (e.g. GSM)
- Messaging applications (e.g. Signal, WhatsApp)
- Legal documentations (digital signatures)
- Credit-card transactions over Internet
- Blockchains
- ... many more!

What is cryptography?

- Protect informations
- Secure communications in presence third parties
- Endpoint authentication
- Verify message integrity

CONFIDENTIALITY

AUTHENTICATION

DATA INTEGRITY

NON-REPUDIATION

Main concepts

- Building blocks of cryptography are called **primitives**
- **Protocol** - step-by-step procedure all participants agree on aimed at specific functions. In cryptography protocols are build combining different primitives together
- **Encryption (E)** - process of transforming plaintext (comprehensible message) to ciphertext (incomprehensible)
- **Decryption (D)** - process of transforming ciphertext to plaintext. Reverse operation of the encryption
- The pair E,D is called **cipher**

Main concepts

- **Secret** - additional parameter to E and D known only by the interested parts in the communication (kA, kB)

  E(pt, kA)   D(ct, kB)

- **Symmetric key**

  kA = kB

- **Asymmetric key (public/private)**

  kA != kB

General problem

Our basic symmetric cipher

Alice and Bob agree on $E(k,m) = km$ & $D(k,c) = c/k$

Alice and Bob choose $k = 2$

Alice sends $m = 7$ as $c = 2 \times 7 = 14$

Bob receives the encrypted message and recovers $m = 14/2 = 7$

Eve doesn’t know E, D and k, only sees 14 going from Alice to Bob

Substitution ciphers
Transposition ciphers

Substitution ciphers

Caesar's Cipher (ROT-x)

Encryption and decryption are basically shift operations of *x* positions across the printable characters set. The key is *x*

e.g. ROT-13

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NOPQRSTUVWXYZABCDEFGHIJKLM

```
meet me there
```

↓

```
zrrg zr gurer
```

Substitution ciphers
Caesar's Cipher (ROT-x)

LIMITS

- only 25 possible shifts, brute-forcing is easy
- given the number of shifts, every letter is substituted with the same letter. Possibility to check frequency of characters or guess the key by knowing parts of the plaintext

Substitution ciphers
Vigenère, polyalphabetic cipher

Shift ciphers logic is extended using as a key a string. Each letter of the key is considered as its position number in the alphabet (e.g. A = 0, B = 1, ...)

Letter frequency can be disguised as each letter is rotated of a number of positions determined by the key

```
      meet me there
KEY:  supe rs ecret
-------------------
      eytx dw xjvvx
```

Substitution ciphers
Vigenère, polyalphabetic cipher

LIMITS

- if len(message) > len(key) the key is simply repeated to match len(message).

  Possible weakness! Knowing the key length the problem can be splitted into len(key) different Caesar ciphers individually breakable

Substitution ciphers
Hill Cipher

Each symbol of the message is 1-1 mapped to a set of numbers modulo X, where X is the total number of symbols (e.g. printable characters)

Encryption and decryption operations are matrix multiplications (modulo X) with a secret matrix key

e.g. mapping A-Z -> 0-25

$$\begin{matrix} F \\ L \\ A \\ G \end{matrix} \quad \begin{pmatrix} 5 \\ 11 \\ 0 \\ 6 \end{pmatrix}$$

$$\begin{pmatrix} 10 & 4 & 21 & 14 \\ 2 & 7 & 18 & 11 \\ 7 & 10 & 1 & 19 \\ 23 & 12 & 3 & 6 \end{pmatrix} \begin{pmatrix} 5 \\ 11 \\ 0 \\ 6 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 9 \\ 8 \end{pmatrix} \pmod{26}$$

Transposition ciphers
Column transposition

The symbols of the plaintext are not substituted. The ciphertext is a permutation of the symbols of the plaintext obtained through a complex algorithm

example: encryption of the message *PIANTARE IL CAMPO DIETRO LA COLLINA*

```
V E T R I N A
- - - - - - -
7 2 6 5 3 4 1
- - - - - - -
P I A N T A R
E I L C A M P
O D I E T R O
L A C O L L I
N A D V Y I Q
```

Output:

```
RPOII IDAAT ATLAM RLNCE OALIC PEOLN
```

Kerchoff’s principle

“The cryptographic key should be the only secret: it would be foolish to rely on our enemies not to discover what algorithms we use because they most likely will. Instead, let’s be open about them.”

XOR cipher

eXclusive OR operation

XOR is one of the main boolean binary operators. Generally represented with the following symbols

^

⊕

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

XOR properties

$a \oplus (b \oplus c) = (a \oplus b) \oplus c$ associativity

$a \oplus b = b \oplus a$ commutativity

$a \oplus a = 0$ identity element

$a \oplus 0 = a$ self inverse

$a \oplus b \oplus a = b$ self elimination

One-Time Pad (OTP)

Using the XOR operator we can build the following **secure** function

$$E(k,m) = m \oplus k = c$$

- If *k* is chosen randomly the attacker has no information on the message m (this is called **perfect secrecy**)

Attacks on OTP: known plaintext key recovery

- Attacker can request the encryption of a given message
- Knowing (*m*,*c*), the key can be easily recovered

$$k = m \oplus c$$

Attacks on OTP: key reuse

c_1, c_2 encrypted with the same key k

$$c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2$$

Attacks on OTP: crib-dragging

c_1, c_2 encrypted with the same key *k*

If we know parts of m_1 and parts of m_2 we can retrieve pieces of k

Useful tools for these attacks are

- https://github.com/CameronLonsdale/MTP (Many-Time Pad attack)
- https://github.com/hellman/xortool

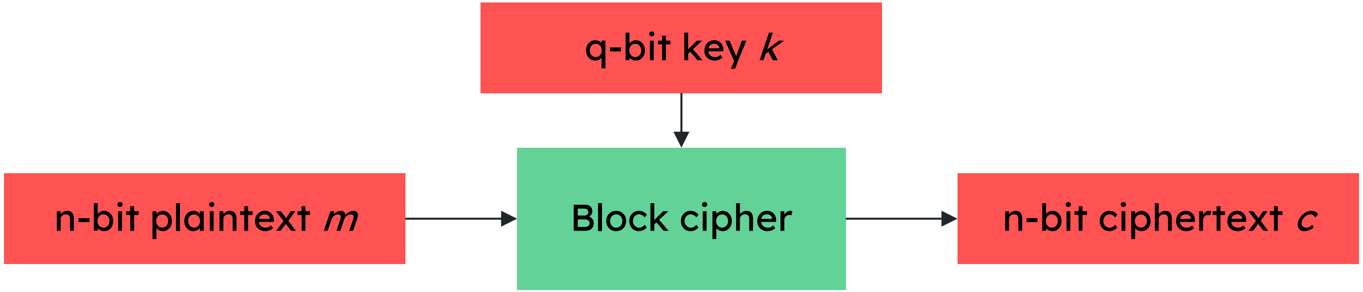
Block ciphers

What is a block cipher?

Algorithm that allows the encryption of blocks of **fixed length**, called **block size**, using a shared secret key *k* (**symmetric key**)

The cipher is defined by choosing E (encryption function) and the inverse operation D (decryption function)

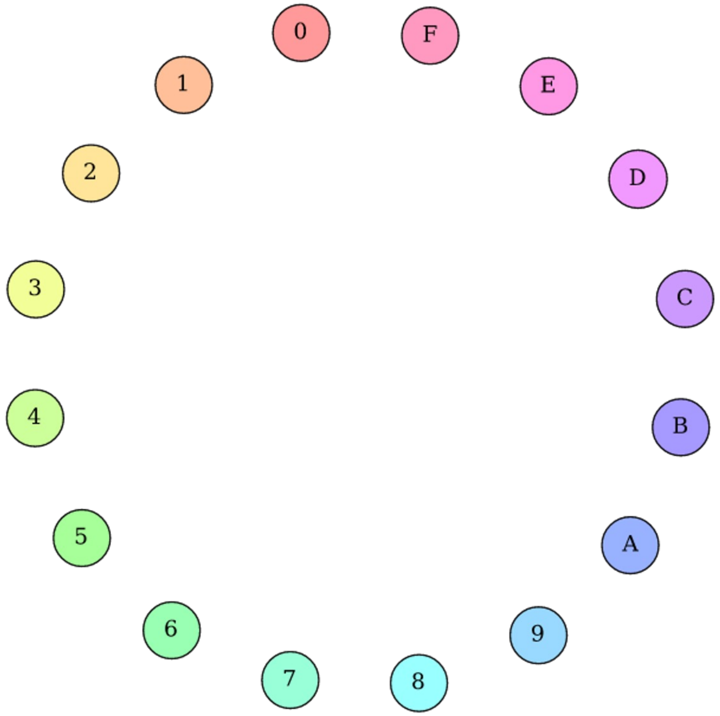
c = E(m,k) m = D(c,k)



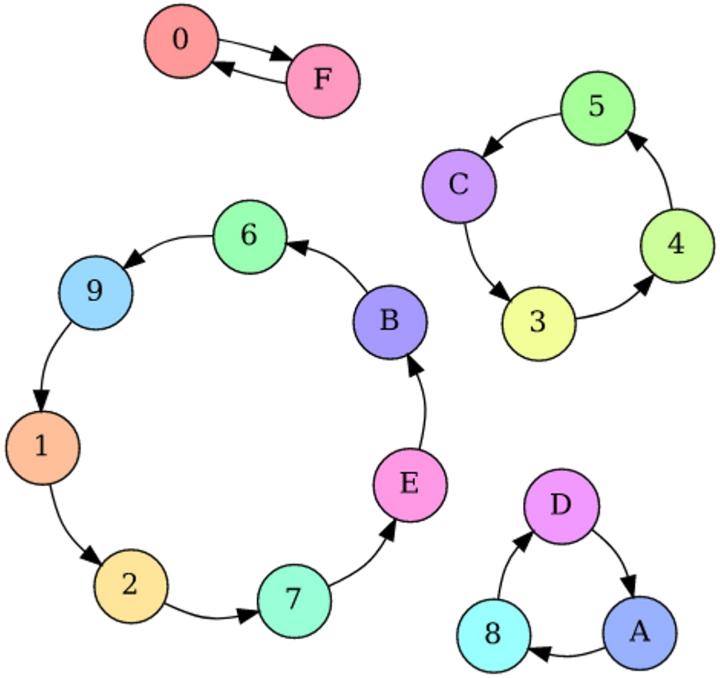
Keyed permutation

- Block ciphers can be seen as large substitution tables implementing a permutation of the *n*-dimensional space of blocks
- The key identifies one of the possible $2^n!$ permutations

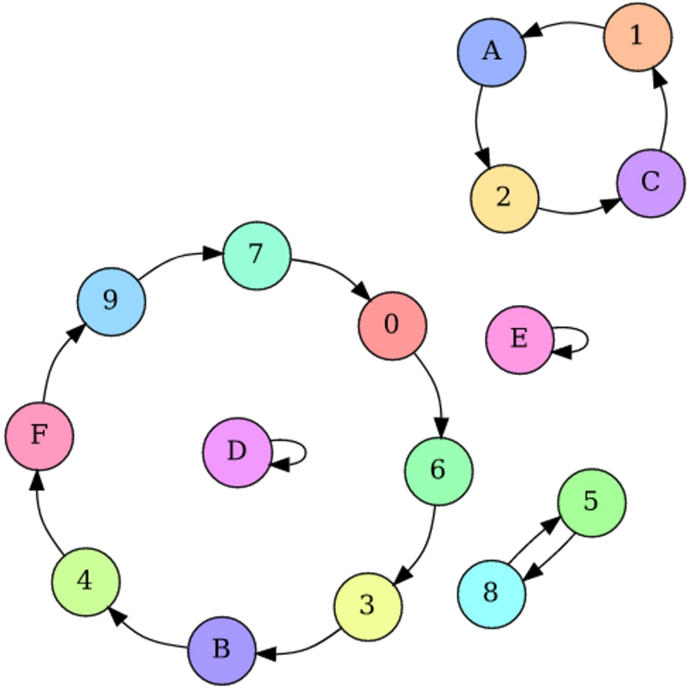
Keyed permutation



Keyed permutation



Keyed permutation



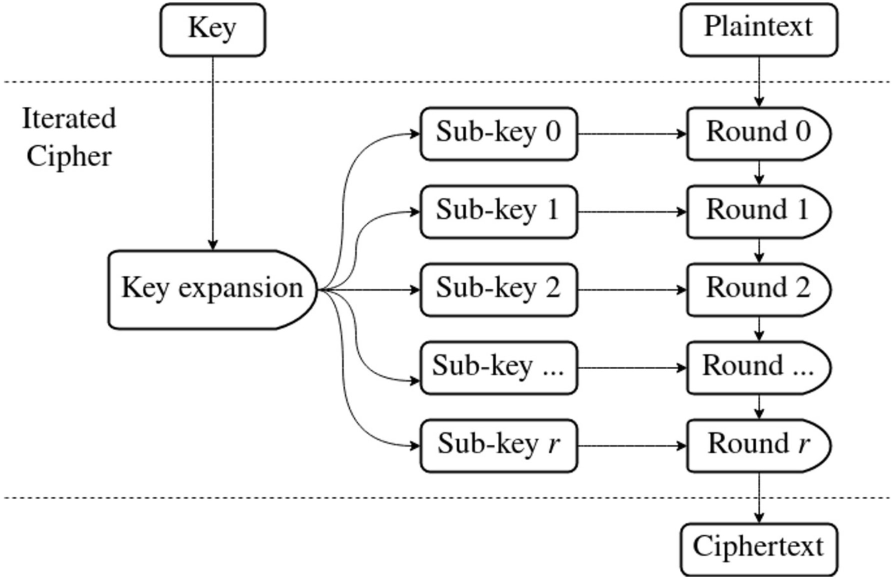
Common construction

All modern block ciphers are designed and implemented as **iterated ciphers**, made of

- **key schedule algorithm**, to generate subkeys from the master key
- **round function**, iterated with the different subkeys

Does iteration increase security? There are heuristic evidences of incremented security, but not every function is good for iteration (e.g. linear functions)

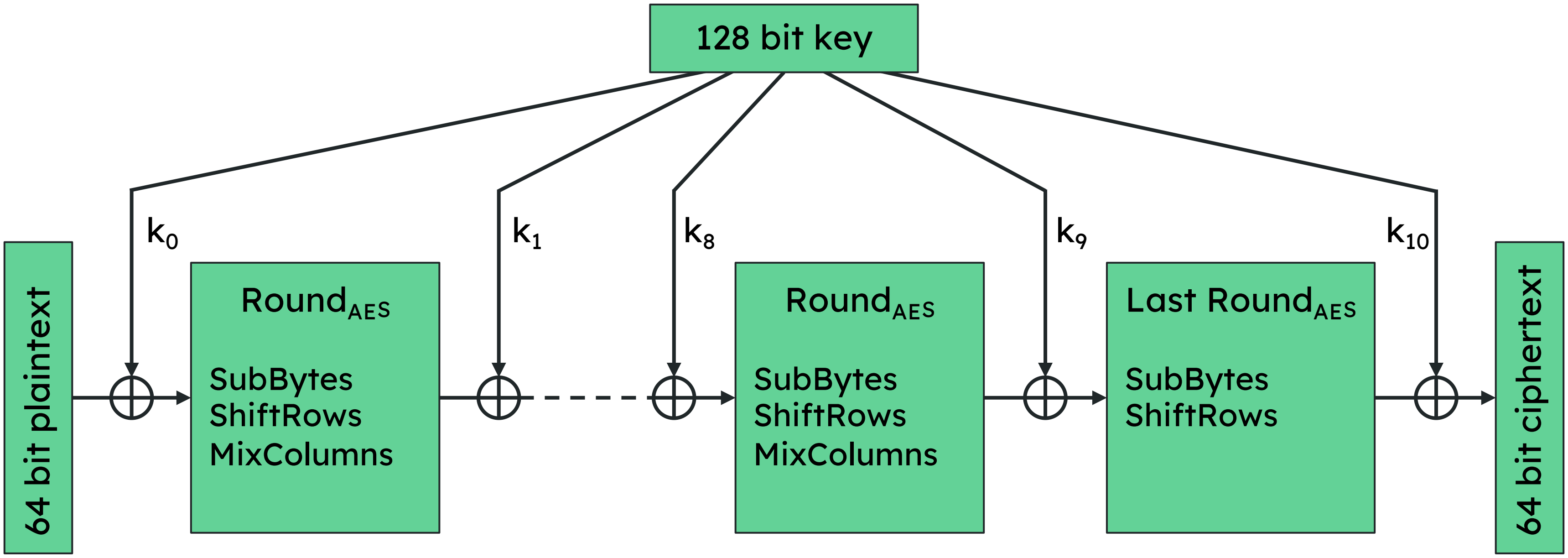
Common construction



AES

Advanced
Encryption Standard

AES - Structure



Padding

Operation of **extending** the plaintext to match the desired length required by the underlying block cipher in use

We remove the very restrictive requirement $n = mb, m \in \mathbb{N}$

- First idea: add null bytes (`0x00`) at the end
  Problem: difficult to remove correctly the padding after the decryption
- **PKCS#5** & **PKCS#7** standards: value of added bytes matches the number of added bytes

  Example: 3 bytes missing, the padding will be `0x03 0x03 0x03`
  N.B. if the message has the correct size a whole new block is created

Remaining problems with block ciphers

1. What if a message is larger than a block?
2. How are symmetric keys shared?

Modes of operation

We can use the already defined block ciphers in some way to extend their capabilities to messages longer than the blocksize!

Main modes of operation:

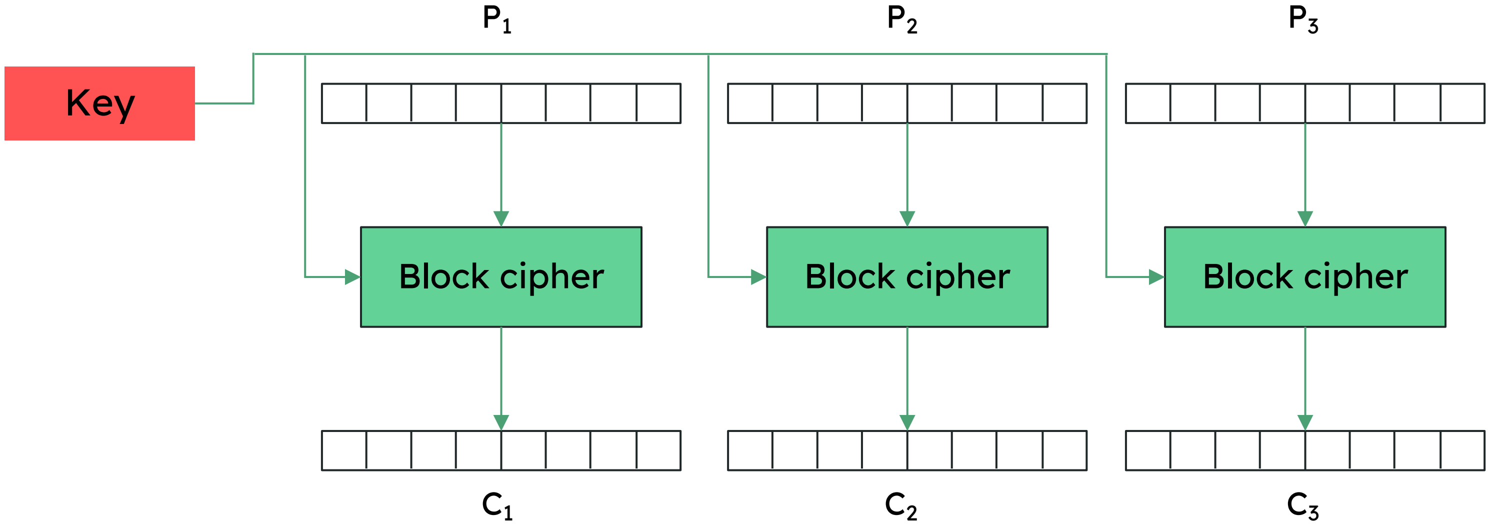
- **Electronic Code Book (ECB)**
- **Cipher Block Chaining (CBC)**
- **Counter (CTR)**

[Other modes: Cipher FeedBack (CFB), Output FeedBack (OFB), Galois Counter Mode (GCM), ...]

ECB Mode

- Given a message of length *n* and a block cipher with blocksize *b*
- Suppose that $n = mb, m \in \mathbb{N}$

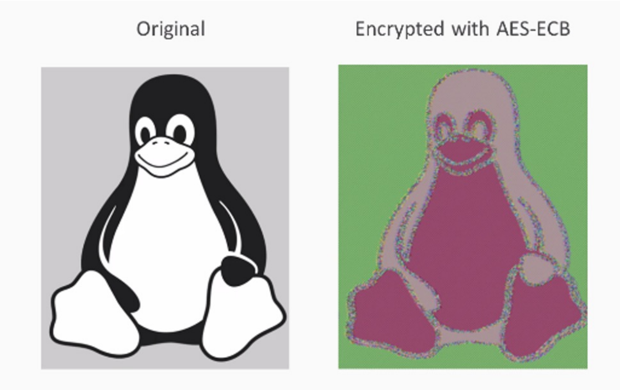
Let's split the message in n/b parts p_1, p_2, ... and being encrypt them with the same key to c_1, c_2, ...



ECB - Issues

- Equal blocks of plaintext are converted to equal block of ciphertext
- Global structure of message is preserved (information about the plaintext from the ciphertext!)

An example using images



ECB - Oracle Attack

Scenario: we are given an **oracle** that computes and returns the following

$$C = ECB(key, P \| S)$$

- P is a chosen plaintext
- S is a secret we try to recover
- || performs the concatenation of P and S

ECB - Oracle Attack

Attack **strategy**

- Send a message with len(P) == blocksize - 1, save the result
- Bruteforce the last byte of the message until the match with the saved result is found
- And so on, one byte at a time

With AES-128 key bruteforce takes 2^{128} tries, ECB Oracle takes 256 * len(S) tries

Introduction to Cryptography

m0leCon 2023 Workshops