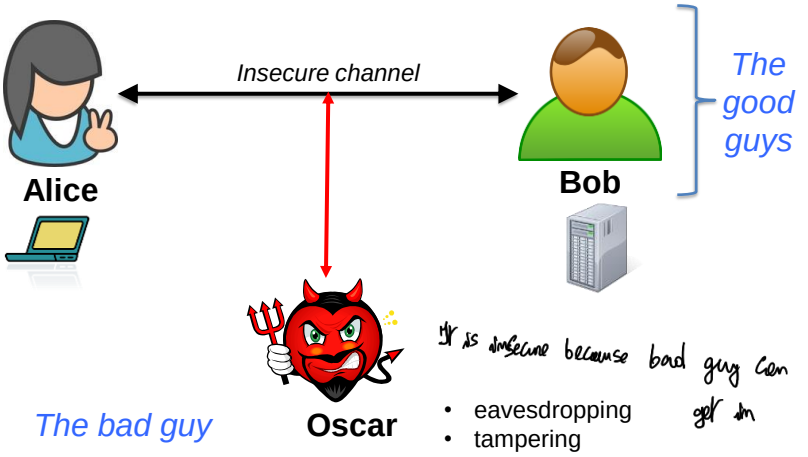
Symmetric Encryption

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
gianluca.dini@unipi.it

1

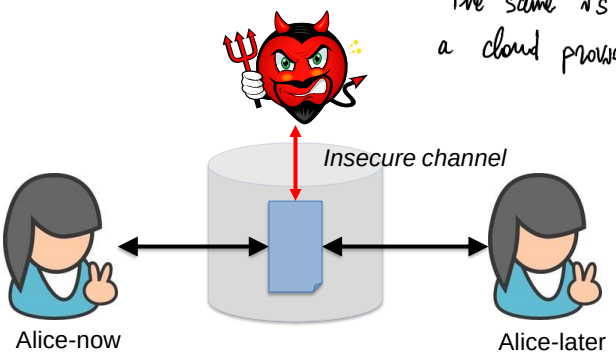
Main characters



2

Variation: storage. Alice sends a message to Alice tomorrow. The laptop storing it is a channel. The same is for a cloud provider.

Encrypted files



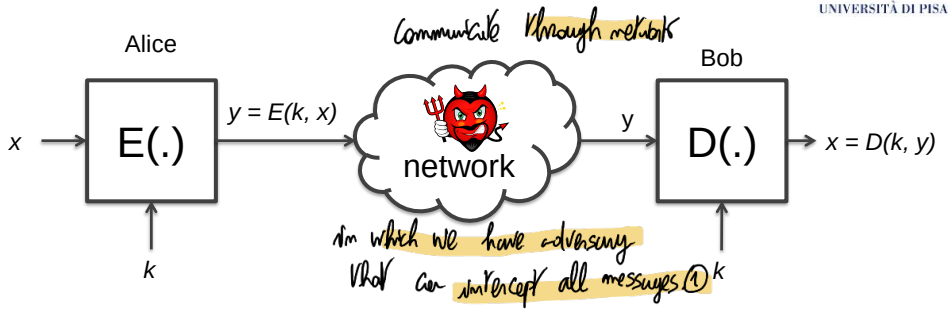
- Analogous to secure communication
- *Alice-now* sends an encrypted message to *later*

March 22 FoC - Symmetric Encryption 3

3

Communication model is:

The model



Communicate through network

in which we have adversary that can intercept all messages ① k

- *E*, *D*: cipher *k*: shared secret key (128 bits)
- *x*, *y*: plaintext, ciphertext

- Encryption algorithm is publicly known
  - Never use proprietary algorithm

March 22 FoC - Symmetric Encryption 4

4 ① Alice and Bob use 2 algorithms to secure communication (E: encryption, D: decryption)
Encryption component takes the plaintext and key as input and produces the ciphertext.

Bob uses decryption algorithm that takes ciphertext and key as input, which produces the original plaintext.

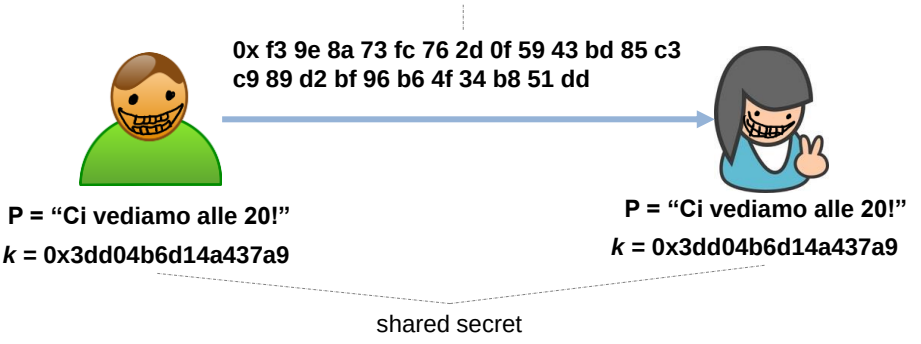
Note that E and D are publicly known (adversary knows them).

REQUIREMENT: Key is secret, System is secure as long as key is secret.

How can Alice and Bob get a key? Meet in person, but we want the network to achieve this secret.

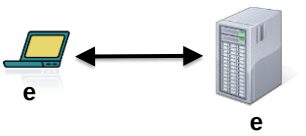
NOTE: If Bob wants to reply both Alice and Bob need E, D.

Example: DES (CBC)



5

Example: SSL



- **Handshake protocol**
  - establish a **shared secret key** by means of public key cryptography
    - 2nd part of the course
- **Record protocols**
  - use **shared secret key** to transmit data to ensure confidentiality and integrity
    - 1st part of the course

6

Cipher definition

Keygen.

- **(DEF)** A cipher, or encryption scheme, defined over **(K, P, C)** is a triple of "efficient" algs (Gen, Enc, Dec) s.t.
  - Gen: $\mathbf{Z}^+ \rightarrow \mathbf{K}$
  - Enc: $\mathbf{P} \times \mathbf{K} \rightarrow \mathbf{C}$; Dec: $\mathbf{C} \times \mathbf{K} \rightarrow \mathbf{P}$
  - Enc may be randomized; Dec is always deterministic
  - Equivalent notations
    - Enc(k, x), $\text{Enc}_k(x)$, E(k, x), $E_k(x)$
    - The same for Dec

Properties of a cipher

- Correctness
  - For all p in **P** and k in **K**, D(k, E(k, p)) = p
- Security (informal)
  - A symmetric cipher is secure iff for each pair (p, c), with p ∈ **P** and c ∈ **C**, then
  - given the ciphertext c, it is "difficult" to determine the corresponding plaintext p without knowing the key k, and vice versa
  - given a pair of ciphertext c and plaintext p, it is "difficult" to determine the key k, unless it is used just once

An historical example

UNIVERSITÀ DI PISA

Mono-alphabetic substitution

Cleartext alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key	J	U	L	I	S	C	A	E	R	T	V	W	X	Y	Z	B	D	F	G	H	K	M	N	O	P	Q

P = "TWO HOUSEHOLDS, BOTH ALIKE IN DIGNITY,
IN FAIR VERONA, WHERE WE LAY OUR SCENE"

("Romeo and Juliet", Shakespeare)

We don't encrypt the original text. We group into blocks of 5 letters. To not give information about words and their length.

P' = "TWOHO USEHO LDSBO THALI KEIND IGNIT
YINFA IRVER ONAWH EREWE LAYOU RSCEN E"

C = "HNZEZ KGSEZ WIGUZ HEJWR VSRYI RAYRH
PRYCJ RFMSF ZYJNE SFSNS WJPZK FGLSY S"

First Attack

UNIVERSITÀ DI PISA

- Brute force attack (exhaustive key search)
  - Oscar has ciphertext (y) and some plaintext (x)
  - Oscar tries all possible keys
    - **for each** k in K
      if (y == E(k, x)) **return** k
- The attack is *always possible*
- The attack may be more complicated because of *false positives* (later)

An historical example

- Mono-alphabetic substitution
  - The key is a permutation of the alphabet
  - Encryption algorithm
    - Every cleartext character having position *p* in the alphabet is substituted by the character having the same position *p* in the key
  - Decryption algorithm
    - Every ciphertext character having position *p* in the key is substituted by the character having the same position *p* in the (cleartext alphabet)
- Number of keys ≈ 26! ≈ 4×10^{26}
  - number of seconds since the Universe birth!

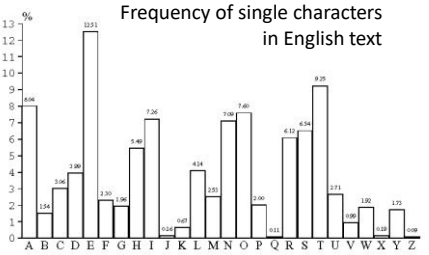
An historical example

- Brute force attack is practically infeasible given the enormous key space
- Brute force attack considers the cipher as a black box
- The monoalphabetic substitution algorithm is subject to an analytical attack which analyzes the internals of the algorithm

An historical example

- The monoalphabetic-substitution cipher maintains the redundancy that is present in the cleartext
- It can be "easily" crypto-analized with a ciphertext-only attack based on language statistics

Frequency of single characters in English text



An historical example

- The following properties of a language can be exploited
  - The frequency of letters
  - Generalize to pairs or triples of letters
  - Frequency of short words
    - If word separators (blanks) have been identified

Lesson learned

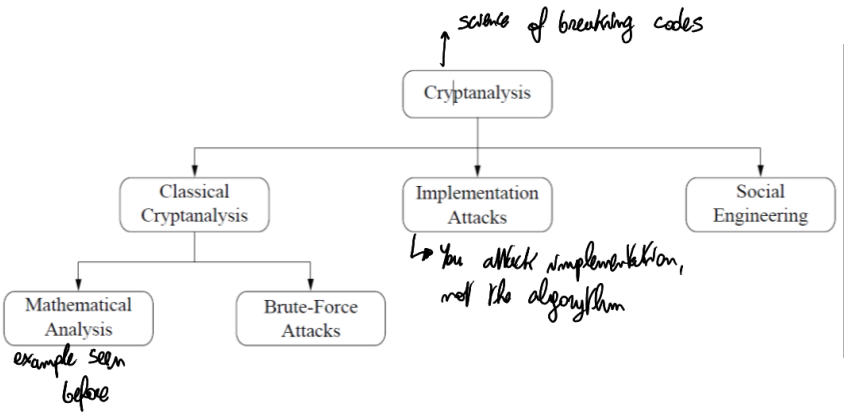
UNIVERSITÀ DI PISA

- Good ciphers should hide statistical properties of the encrypted plaintext
- The cyphertext symbols should appear to be random
- A large key space alone is not sufficient for strong encryption function (necessary condition)

15

Crptanalysis

UNIVERSITÀ DI PISA



16

Attack Complexity

When we talk about "difficult", we mean: either takes a lot of time or you have to consider the complexity of attack:

- Attack complexity is the dominant of:
  - Data complexity
    - Expected number of input data units required (How many inputs you need: ciphertext and plaintext)
  - Storage complexity
    - Expected number of storage units required (Amount of storage your attack needs to perform an attack)
  - Processing complexity
    - Expected number of operations required to processing input data and/or fill storage with data (try all the possible keys or to exploit statistical property)

If those complexities are exponential or sub-exponential is good. We will see.

17

Types of attacks

- Attacks are classified according to what information an adversary has access to
  - ciphertext-only attack (the least strong) We always assume this is possible always
  - known-plaintext attack: attack in which adversary is able to have pairs of plaintexts and cyp.
  - chosen-plaintext attack (the strongest)
    - ↳ the adversary is the one to choose the plaintext to receive ciphertext.
- Fact.
  - A cipher secure against CPAs is also secure against the others
- Best practice.
  - It is customary to use ciphers resistant to a CPA even when mounting that attack is not practically feasible

18

1. **Storage Complexity**
   This refers to how much memory (RAM, disk space, etc.) an attack requires. Some attacks need to store large precomputed tables (like rainbow tables for cracking hashes), while others may work with minimal storage but trade it for increased computation.

2. **Data Complexity**
   This measures how much input data (e.g., plaintext-ciphertext pairs, known messages, intercepted ciphertexts) an attack needs to be successful. Some attacks require a huge amount of intercepted encrypted traffic, while others may work with just a few chosen or known plaintexts.

3. **Processing Complexity**
   This is the amount of computational power needed to carry out the attack. It's typically measured in operations like XORs, multiplications, or hash evaluations. Some attacks may require only polynomial-time computations, while others (like brute force) are exponential in nature.

Kerchoff's principle (19th century)

- Kerchoff's maxim
  - A cryptosystem should be secure even if everything about the system, except the key, is public knowledge
- Shannon's maxim
  - The enemy knows the system
- Pros
  - Maintaining security is easier
    - Keys are small secrets
      - Keeping small secrets, it's easier than keeping large secrets
      - Replacing small secrets, once possibly compromised, is easier than replacing large secrets (cheaper)

19

Security through Obscurity

- Security through Obscurity
  - Attempt to use secrecy of design or implementation to provide security
- History shows that it doesn't work
  - GSM/A1 disclosed by mistake
  - RC4 disclosed deliberately
  - Enigma disclosed by intelligence
  - ... many others...

In case of compromission, fixing faulty algorithms like in GMS it can take a lot of money, time and reputation

- Defense in Depth
  - Solely relaying on StO is a poor design decision
  - StO is a valid secondary measure

20

- knows the system
- Bruice Schneier's maxim
  - The fewer and simpler the secrets that one must keep to ensure system security, the easier it is to maintain system security.

Security through Obscurity

- "Hiding security vulnerabilities in algorithms, software, and/or hardware decreases the likelihood they will be repaired and increases the likelihood that they can and will be exploited by evil-doers. Discouraging or outlawing discussion of weaknesses and vulnerabilities is extremely dangerous and deleterious to the security of computer systems, the network, and its citizens." – S.M. Bellovin and R. Bush, *Security Through Obscurity Considered Dangerous*, Internet Engineering Task Force (IETF), February 2002.

21

Symmetric Encryption

EXERCISES

22

Shift Cipher (Caesar Cipher)

- Shift every plaintext letter by a fixed number of positions (the key) in the alphabet with wrap around
- Ex.
  - PT = «ATTACK»
  - K = 17
  - CT = "RKKRTB"

23

Shift Cipher (Caesar Cipher)

- Letters are encoded as numbers
  - A $\rightarrow$ 0, B $\rightarrow$1, C $\rightarrow$ 2, …, Z $\rightarrow$ 25
- PT, CT and K are elements of the ring $\mathbb{Z}_{26}$
  - Encryption: $y = x + k \bmod 26$
  - Decryption: $x = y - k \bmod 26$
  - EX.
    - PT (x) = «ATTACK» => 0 19 19 0 2 10
    - K = 17
    - CT (y) = 17 10 10 17 19 1 => "RKKRTB"

24

Shift Cipher (Caesar Cipher)

- Possible attacks
  - Brute force attack
    - Small key space: 26 possible keys
  - Anlytical attack
    - Letter frequency analysis

25

Affine cipher

- Definition
  - Let a, b, x, y $\in \mathbb{Z}_{26}$
  - Encryption: $y = a \cdot x + b \bmod 26$
  - Decryption: $x = a^{-1} (y - b) \bmod 26$
  - With k = (a, b) and gcd(a, 26) = 1

a^{-1} exists if gcd(a, 26) in modular arythmetic.

greatest common denominator

- Example
  - Plaintext: «ATTACK» => 0, 19, 19, 0, 2, 10
  - k = (9, 13)
  - Ciphertext: 13, 2, 2, 13, 5, 25 => «NCCNFZ»

26

Affine cipher

- Attacks
  - Brute force attack
    - Key space = (#values for a) × (#values for b) = 12 × 26 = 312
  - Analytical attack
    - Letter frequency analysis

a cannot assume all 26 values

27

Reader

- Understanding Cryptography, Section 1.4 “Modular Arithmetic and More Historical Ciphers”

28