XOR Cipher

Rocco DE NICOLA
IMT Lucca

Gaspare FERRARO
CINI Cybersecurity National Laboratory

Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided "as is" without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Goal

- Introduce the concept of cryptanalysis
- Present the parameters needed to define «safe» a cipher
- Classify cryptographic attacks based on their metodologies
- Give the definition of perfect encryption
- Present XOR cipher and One-Time-Pad as an example of a perfect cipher

Prerequisites

- Lecture:
  - *CR_1.1 – Introduction to cryptography and classical ciphers*

Outline

- Cryptanalysis
- Attacker’s model and techniques
- Perfect encryption
- XOR Cipher and One-Time-Pad

Outline

- Cryptanalysis
- Attacker’s model and techniques
- Perfect encryption
- XOR Cipher and One-Time-Pad

Cryptanalysis

- *Cryptanalysis* is a set of techniques, set up
  - to test the robustness of the algorithm and of the key by trying possible attacks against it
  - to break the code and infer the key from the available ciphertext or decrypt the ciphertext without knowing key
- Two kinds of attacks:
  - Analytic
  - Brute-force

Safe encryption

- A symmetrical encryption pattern is safe if:
  - The sender and the receiver receive and keep the key safely (no attacker must intercept the key)
  - The encryption algorithm is robust, i.e., an attacker in possession of a certain number of ciphertexts, but not of the encryption key, is unable to infer the plaintext or the key
- It is assumed that the algorithm is known and that it is impractical to decipher messages by having only ciphertexts (Kerckhoffs's principle)

Kerckhoffs's principle

- *The encryption scheme* is not secret
  - The attacker knows the encryption scheme
  - The only secret is the *key*
  - The key must be chosen at random and kept secret
- Some arguments in favor of this principle:
  - Easier to keep secret a *key* rather than an *algorithm*
  - Easier to change a *key* than to change an *algorithm*
  - Simplifies standardization:
    - Ease of deployment
    - Public validation

Cryptanalysis application

- Cryptanalysis techniques can be applied starting from different “hypotheses” about the information possessed by the attacker:
  - Not knowing anything, not even the algorithm
  - Knowing some ciphertexts and the algorithm
  - Knowing also some plaintexts

Outline

- Cryptanalysis
- Attacker’s model and techniques
- Perfect encryption
- XOR Cipher and One-Time-Pad

Attacker’s Knowledge

- *Ciphertext only*: A collection of ciphertexts
- *Known plaintext*: A collection of ciphertexts and one or more pairs <plaintext, ciphertext>
- *Chosen plaintext*: A collection of <plaintext, ciphertext> pairs with plaintexts selected by the attacker
- *Chosen ciphertext*: A collection of <plaintext, ciphertext> pairs with ciphertexts selected by the attacker
- *Chosen text*: Two collections of pairs, <plaintext, ciphertext> one with chosen text and the other with chosen ciphertext

Cryptonalitic Attacks

- The attacker tries:
  - to deduce the key used from a specific plain text, to compromise all future and past messages encrypted with that key
  - to guess the plain text from the encrypted text
- The attacker leverages on:
  - the knowledge of the encryption algorithm
  - some knowledge of the general characteristics of plaintext
  - (possibly) some sample pairs of <plaintext, ciphertext>

Brute force attacks

- The attacker:
  - Tries all possible keys on some ciphertexts until an intelligible translation into plaintext is obtained
  - On average, half of all possible keys must be tried to achieve success
- The attacker must have:
  - Some degree of knowledge of the expected plaintext
  - Some means to automatically distinguish plain texts from ciphered texts

Levels of security

- Unconditional security: no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- Computational security: given limited computing resources (e.g., the time needed for calculations is greater than the age of universe), the cipher cannot be broken
- Quantum Computers might change the scene: it might be possible to create specific algorithms for them that dramatically reduce the time needed to break cryptographic algorithms

Shannon's principles

- How to evaluate if a cipher is secure enough?
- Shannon defined two key concepts:
  - *Confusion*:
    - Makes relationship between ciphertext and key as complex as possible
    - The key must be well distributed in the ciphertext
    - Every bit of the ciphertext should depend on every bit of the key
  - *Diffusion*:
    - Dissipates statistical structure of plaintext over bulk of ciphertext
    - The plaintext must be well distributed in the ciphertext
    - Every bit of the ciphertext should depend on every bit of the plaintext
- Avalanche effect: changing 1 bit of the (plaintext, key) should change every bit of the ciphertext with a probability of 50%

Outline

- Cryptanalysis
- Attacker’s model and techniques
- Perfect encryption
- XOR Cipher and One-Time-Pad

Perfect secrecy

- Perfect secrecy is based on the idea that, for any two messages m1, m2 and any ciphertext c, the probability of obtaining c as the result of the encryption of m1 or m2 is the same
- Symmetric encryption algorithms rely on substitutions and transpositions. Even for the best of those currently in use, it is not known whether there can be an efficient cryptanalytic procedure that can reverse these transformations without knowing the encryption key
- Asymmetric encryption algorithms depend on mathematical problems that are thought to be difficult to solve. There is no proof that these problems are hard, and a mathematical breakthrough could make systems vulnerable to attack

Outline

- Cryptanalysis
- Attacker’s model and techniques
- Perfect encryption
- XOR Cipher and One-Time-Pad

Exclusive Or (XOR)

- Boolean operation that returns true only when inputs differ
- Typically represented by the symbol ⊕ or with ^ in many programming language
- Truth table:
  - Same inputs: $0 \oplus 0 = 1 \oplus 1 = 0$
  - Different input: $0 \oplus 1 = 1 \oplus 0 = 1$
- Some properties:
  - Identity element: $A \oplus 0 = A$
  - Self-inverse: $A \oplus A = 0$
  - Commutative: $A \oplus B = B \oplus A$
  - Associative: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- We can construct a simple cipher with this operation

Exclusive Or (XOR)

- Boolean operation th urns true only when inputs differ
- Typically represented ymbol ⊕ or with ^ in many programming language
- Truth table:
  - Same inputs: $0 \oplus 0 = 1$
  - Different input: $0 \oplus 1 =$
- Some prop
  - Identity e
  - Self-inve
  - Commut
  - Associative: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- We can construct a simple cipher with this operation

Further details can be found in the lectures:

- *HW_S_0.2.1 – Logic Gates & Flip-Flops (slide 40-on)*
- *HW_S_0.2.3 – Linear Feedback Shift Registers - LFRSs*

XOR Cipher

- The XOR cipher is a simple encryption algorithm in which each character of a message m is encrypted by applying the bitwise XOR operation $\oplus$ with a fixed key k, formally:
  - $m = m_1, m_2, m_3, \ldots$ and $k = k_1, k_2, k_3, \ldots$ we have $c = (m_1 \oplus k_1), (m_2 \oplus k_2), (m_3 \oplus k_3), \ldots$
- For properties of the XOR operation, the decryption coincide with encryption
- By using a constant repeating key, a simple XOR cipher can trivially be broken using frequency analysis
- XOR cipher is vulnerable to known-plaintext attack, as given a pair (m, c) we recover the key as $k = m \oplus \ c$
- Messages encrypted with XOR cipher are also easily malleable as we can alter portion of the ciphertext to obtain predictable changes in the message

Perfect encryption with OTP

- *One-Time Pad* (OTP) is a perfect encryption technique that cannot be cracked, but requires a pre-shared key with at least the same size as the message (the one-time pad)
- Is equivalent to a XOR Cipher where $|k| = |m|$
- If the key is truly random and never reused (in whole or in part), the resulting ciphertext will be impossible to decrypt
- Any cipher scheme, to guarantee perfect secrecy, must use keys with effectively the same requirements as OTP keys (slightly impractical!)

Many-Time Pad (MTP)

- If the same key k is used to encrypt with One-Time Pad two or more different messages m_1, m_2, ..., into c_1, c_2, ..., we can perform a Many-Time Pad attack
- We know that:
  - $c_1 = m_1 \oplus k,\ c_2 = m_2 \oplus k \rightarrow c_1 \oplus c_2 = m_1 \oplus m_2$
- We can exploit statistical information from pairs of XORed message $m_i \oplus m_j$, examples can be found on the article:
  - https://www.thecrowned.org/the-one-time-pad-and-the-many-time-pad-vulnerability
- Useful tool to perform Many-Time Pad attack interactively:
  - https://github.com/CameronLonsdale/MTP

XOR Cipher

Rocco DE NICOLA
IMT Lucca
Gaspare FERRARO
CINI Cybersecurity National Laboratory
Matteo ROSSI
Politecnico di Torino

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it