Cryptography

Module in Autumn Term 2016
University of Birmingham

Lecturers: Mark D. Ryan and David Galindo

Slides originally written by Eike Ritter
Based on material developed by Volker Sorge

What is Cryptography

Encryption essential for security on the internet
Confidentiality, integrity, privacy cannot be guaranteed otherwise
Works in principle as follows:

- Alice and Bob share a secret key. HOW??
- Alice uses secret key to scramble data: encryption
- Alice sends scrambled data to Bob
- Bob decrypts data with secret key, gets message back

Course content

Lecture course will explain basic cryptographic algorithms
Will also reason about their security
Will explain how to use the algorithms properly

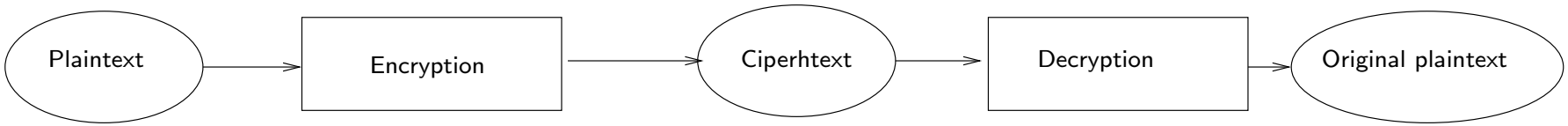
Kinds of cryptography

- Transposition: permutes components of a message
- Substitution: replacing components. Two main ways:
  - Codes: algorithms for substitution of entire words (working on meaning)
  - Ciphers: algorithms substituting bits, bytes or blocks

Ciphers are easiest to use and mathematically well understood
$\Rightarrow$ will concentrate on those

Terminology

Plaintext	Message before encryption
Encryption	Process of scrambling a message
Ciphertext	An enciphered message
Decryption	Process of unscrambling a message



Transposition Cipher

Used already since antiquity
Example: Rail Fence Cipher

- Key: Column size
- Encryption: Arrange message in columns of fixed size (the key). Add dummy text to fill the last column. Ciphertext consists of rows.
- Decryption: Calculate row size by dividing message length by the key. Arrange message in rows of this size. Plaintext consists of columns.

Security of Transposition Cipher

Is this cipher secure?

Informal answer: No.

Given any ciphertext, attacker tries all possible values for the key.

For a message of size n there are at most n possibilities for the key, hence attacker will obtain plaintext.

Precise formulation of security

Use game between two parties:

- Attacker(A): Aim is to obtain plaintext for given ciphertext
- Challenger(C): provides the challenge for the attacker

Moves of the game:

- C selects message length n and chooses a key k.
- C chooses message m and sends encrypted message $\mathrm{Enc}_k(m)$ to A
- A does some computations and eventually outputs a message

A wins the game if A's output is essentially the same as m.
(Note: A doesn't have key!)

A has probability of at least $\frac{1}{n}$ of winning this game for any message.
$\Rightarrow$ Protocol insecure.

Precise formulation of security

Use game between two parties:

- Attacker(A): Aim is to obtain plaintext for given ciphertext
- Challenger(C): provides the challenge for the attacker

Moves of the game:

- C selects message length n and chooses a key k.
- C chooses message m and sends encrypted message $\mathrm{Enc}_k(m)$ to A
- A does some computations and eventually outputs a message

A wins the game if A's output is essentially the same as m.
(Note: A doesn't have key!)

A has probability of at least $\frac{1}{n}$ of winning this game for any message.
$\Rightarrow$ Protocol insecure.

Permutations

A permutation describes the re-arrangement of the elements of an ordered list into a one-to-one correspondence of itself
Permutation is therefore a function from $\{1, \ldots, n\}$ to itself which is one-to-one.
Example: reordering of $(1, 2, 3)$ to $(2, 3, 1)$.
Two notations used

- Array notation: Write the re-ordered list below the original one, here $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$
- Write down the cycles. The first cycle is the list of numbers obtained by applying the permutation first to 1, then to the result and so on. Stop when 1 appears again. The other cycles are obtained by starting with the lowest number not appearing in the previous cycle and applying the same recipe. Cycles of length 1 are omitted. Example would be (123).

Operations on permutations

- There is the identity which maps any number to itself
- Two permutations can be composed, resulting in another permutation
- The inverse of a permutation s is the permutation t such that s composed with t is the identity.

Monoalphabetic substitution cipher

- Key: permutation of the alphabet
- Encryption: Apply the permutation
- Decryption: Apply the inverse permutation

Here is one way to choose the key:

- Choose keyword (or keyphrase)
- remove all duplicate letters from keyword
- start cipher-alphabet with letters from duplicate-free keyword
- and the end of the codeword continue with next unused letter of alphabet following last letter in codeword
- continue filling in letters in alphabetical order leaving out already used letters

Security

How difficult is it for the attacker to break this cipher?
Have $26! \approx 2^{86}$ permutations
But: Have other tools available, eg frequency analysis
Frequency of letter occurrence varies dramatically amongst letters
In English text, 12.7% of all letters are "e", and 0.2% of all letters are "x".

Enigma machine

Encryption was mechanised at the beginning of 20th century

Famous example: Enigma machine (used by German military in WW2)

consisted of keyboard, plug board, three rotors and reflector

Enigma machine

Encryption method:

- Letters from keyboard are substituted using plugboard with substitution cipher
- In next step, each rotor applies fixed substitution to the letters
- Key point: rotors are dynamic: rotors advance after each letter
- Message passes through the reflector, which applies one more permutation and applies all three rotors in opposite direction.

Successfully broken by scientist in Bletchley Park (Turing)

Also initiated the development of modern day computers

Modular arithmetic

Definition

- We say two numbers $a, b \in \mathbb{Z}$ are congruent modulo $n \in Z$, written $a \equiv b (\text{mod } n)$, if $a - b$ is divisible by n
- If $0 \leq a \leq n$, we write $[a]_n$, called the residue class of a modulo n, for the set of all numbers b such that $a \equiv b (\text{mod } n)$.
- We define addition, subtraction and multiplication on residue classes by

$$\begin{aligned} [a]_n + [b]_n &= [c]_n \quad \text{if } (a+b) \equiv c (\text{mod } n) \\ [a]_n - [b]_n &= [c]_n \quad \text{if } (a-b) \equiv c (\text{mod } n) \\ [a]_n * [b]_n &= [c]_n \quad \text{if } (a*b) \equiv c (\text{mod } n) \end{aligned}$$

Probability

Will use discrete probabilities

Definition

Let U be a finite set. A *probability distribution* P is a function $P\colon U \to [0,1]$ such that

$$\sum_{u \in U} P(u) = 1$$

We denote by $|U|$ the size of U (the number of elements in U)

Example

Let U be a finite set. The *uniform distribution* is the probability distribution P defined by

$$P(u) = \frac{1}{|U|}$$

Probabilities, continued

Definition

Let $P : U \to [0, 1]$ be a probability distribution.

- An *event* A is a subset of U.
- The probability of an event A, written $P[A]$, is defined as

$$P[A] = \sum_{u \in A} P(u)$$

Bitstrings

We write $\{0,1\}^n$ for the set of all sequences of n bits.
Have important operation $\oplus$ on bitstrings:
$\oplus$ is addition modulo 2 on each bit

One-time pad

First cipher which is secure

Message and keys are bitstrings

- Key: Random bitstring $k_1, \ldots, k_n$, as long as message $m_1, \ldots, m_n$
- Encryption: $k_1 \oplus m_1, \ldots, k_n \oplus m_n$
- Decryption of ciphertext $c_1, \ldots, c_n$: $k_1 \oplus c_1, \ldots, k_n \oplus c_n$

Precise formulation of cipher algorithm

Definition

Let $\mathcal{K}$, $\mathcal{M}$ and $\mathcal{C}$ be three sets, called keys, messages and ciphertexts. A *cipher* over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ is a pair of efficient algorithms $(E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}, \mathcal{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M})$ such that for all $m \in \mathcal{M}$ and $k \in \mathcal{K}$

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

Security of one-time pad

One-time pad satisfies very strong notion of security: Attacker cannot learn any information by looking only at ciphertexts
Formalised by:

Definition

A cipher (E, D) over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ satisfies *perfect security* if for any length n all messages m_1 and m_2 of length n and all ciphertext c

$$P[E(k, m_1) = c] = P[E(k, m_2) = c]$$

where P is the uniform distribution over keys of length n.

Theorem

The one-time pad satisfies perfect security.

Proof.

For randomly-chosen m, c and n,

$$P[E(k, m) = c] = \frac{1}{2^n}$$