Classical Cryptology

Diego F. Aranha

Institute of Computing
UNICAMP

Introduction

Objectives:

- Revisit classical cryptosystems.
- Analyze their resistance against simple attacks.

Introduction

Objectives:

- Revisit classical cryptosystems.
- Analyze their resistance against simple attacks.

Hidden intentions:

- Detect and justify what **not** to use in practice.
- Construct a preliminary model of a secure cryptosystem.

Notation

Sets:

- Alphabet of definition $\mathcal{A}$.
- Plaintext space $\mathcal{M}$.
- Ciphertext space $\mathcal{C}$.
- Key space $\mathcal{K}$.

Algorithms:

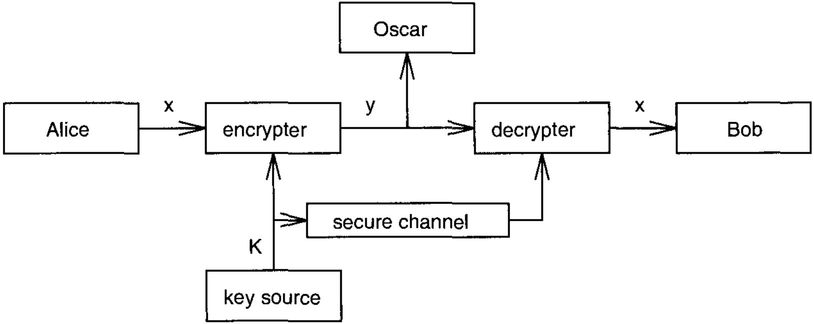
- Bijection $E_e : \mathcal{M} \to \mathcal{C}$ parameterized by key $e \in \mathcal{K}$.
- Bijection $D_d : \mathcal{C} \to \mathcal{M}$ parameterized by key $d \in \mathcal{K}$.

Cryptosystem

A **cryptosystem** is given by
$\{\{E_e : e \in \mathcal{K}\}, \{D_d : d \in \mathcal{K}\} | \forall e \in \mathcal{K}, \exists d \in \mathcal{K}, D_d = E_e^{-1}\}$.

Consistency: $\forall m \in \mathcal{M}, D_d(E_e(m)) = m$.

Cryptosystem



Symmetric block cipher:

- Encryption: $y_i = Enc_k(x_i), 1 \leq i \leq n$
- Decryption: $x_i = Dec_k(y_i), 1 \leq i \leq n$

Modular arithmetic

Let a, b integers and m a positive integer:

- $a \equiv b \pmod{m}$ iff $m|(b - a)$, that is, $a \bmod m = b \bmod m$.
- We say that a is **congruent** to b with relation to modulo m.
- The notation $(a \bmod m) > 0$ is used to denote the **remainder** of the division of a by m.
- We say that $(a \bmod m)$ is the value of a **reduced** modulo m.

Example:

- $101 \bmod 7 = 3$.
- $-101 \bmod 7 = 4$.

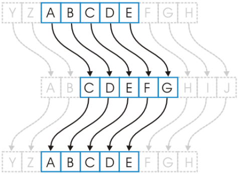
Important: Notice that not all programming languages use this convention!

Modular arithmetic

Define $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ equipped with the operations $(+, \times)$ modulo m:

- $(\mathbb{Z}_m, +)$ is an **Abelian** group.
- $(\mathbb{Z}_m, +, \times)$ is a **ring**.
- Addition is closed, associative, has identity and inverse.
- Multiplication is closed, commutative, associative, distributive and has identity.

Shift cipher (Caesar's cipher)



Definition

It is a block cipher where each letter is replaced by the letter after k positions.

Formalization:

- The key is the number k.
- The permutation is given by $\pi(m_i) = (m_i + k) \bmod |\mathcal{A}|$.

Observations:

- What is the key size?

Shift cipher

Criptanalysis: exhaustive search of the m keys! In average, test only $\frac{m}{2}$ keys.

Desirable properties for a block cipher:

- Functions for encryption Enc_k and decryption Dec_k should be computable in polynomial time.
- An attacker who captures ciphertext should not be able to systematically recover plaintext in reasonable time.
- Key space should resist exhaustive search of an attacker with polynomial computational power.

Monoalphabetic substitution ciphers

Definition

A **monoalphabetic cipher** $E_\pi : \mathcal{M} \rightarrow \mathcal{C}$ is a rule to replace each letter x_i from message x with $\pi(x_i)$, where π defines a permutation in the alphabet of definition.

Formalization:

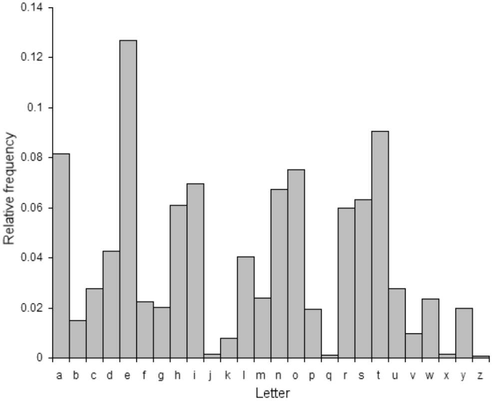
- The key is the permutation $\pi : \mathcal{A} \rightarrow \mathcal{A}$.
- The encryption function is $E_\pi(m) = (\pi(x_1), \pi(x_2), \ldots, \pi(x_{|x|})$.
- The decryption function is $D_\pi(y) = (\pi^{-1}(y_1), \pi^{-1}(y_2), \ldots, \pi^{-1}(y_{|y|})$

Observations:

- The key space has size $(|\mathcal{A}|!)$.
- In general, the key has size $|\mathcal{A}|$.

Monoalphabetic substitution ciphers

Cryptanalysis: Frequency analysis!



Note: Shift cipher is a special case of a monoalphabetic substitution that uses only m from the $m!$ possible keys.

Introduction to Number Theory

1. The **congruence** $ax \equiv b \pmod{m}$ has a single solution $x \in \mathbb{Z}_m$ for any $b \in \mathbb{Z}_m$ iff $gcd(a, m) = 1$.

2. Let $a \in \mathbb{Z}_m$ with $gcd(a, m) = 1$. Then there is a single element a^{-1} such that $aa^{-1} \equiv a^{-1}a \equiv 1 \pmod{m}$. This element is called **multiplicative inverse**.

3. When $gcd(a, m) = 1$, integers a and m are called **relatively prime** or **co-prime**. The number of integers co-prime to m in $\mathbb{Z}_m$ is denoted by Euler's **totient function** $\phi(m)$.

Affine cipher

Definition

It is a special case of a monoalphabetic substitution that applied a linear function such that $Enc_{k=(a,b)}(x) = (ax + b) \bmod m$.

Example: For $m = 26 = 2 \cdot 13$, values a such that $gcd(a, m) = 1$ are $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$.

Parameter b can be any element from $\mathbb{Z}_m$. In this case, the affine cipher has only $12 \cdot 26$ valid keys.

Affine cipher

Definition

It is a special case of a monoalphabetic substitution that applied a linear function such that $Enc_{k=(a,b)}(x) = (ax + b) \bmod m$.

Example: For $m = 26 = 2 \cdot 13$, values a such that $gcd(a, m) = 1$ are $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$.

Parameter b can be any element from $\mathbb{Z}_m$. In this case, the affine cipher has only $12 \cdot 26$ valid keys.

Formalization:

- The key is the pair of integers (a, b).
- Encryption is given by $y_i = Enc_{a,b}(x_i) = (ax_i + b) \bmod m$.
- Decryption?

Affine cipher

Definition

It is a special case of a monoalphabetic substitution that applied a linear function such that $Enc_{k=(a,b)}(x) = (ax + b) \bmod m$.

Example: For $m = 26 = 2 \cdot 13$, values a such that $gcd(a, m) = 1$ are $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$.

Parameter b can be any element from $\mathbb{Z}_m$. In this case, the affine cipher has only $12 \cdot 26$ valid keys.

Formalization:

- The key is the pair of integers (a, b).
- Encryption is given by $y_i = Enc_{a,b}(x_i) = (ax_i + b) \bmod m$.
- Decryption is given by $x_i = Dec_k(y_i) = a^{-1}(y_i - b) \bmod m$.

Affine cipher

How many keys are valid for a generic affine cipher where $a, b \in \mathbb{Z}_m$?

Affine cipher

How many keys are valid for a generic affine cipher where $a, b \in \mathbb{Z}_m$?
Answer: The number of valid keys is $m \cdot \phi(m)$.

Let

$$m = \prod_{i=1}^{n} p_i^{e_i},$$

where primes p_i are pairwise distinct with $e_i > 0$. Then

$$\phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - {p_i}^{e_i - 1})$$

.
Example: If $60 = 2^2 \cdot 3 \cdot 5$, $\phi(60) = (4-2) \cdot (3-1) \cdot (5-1) = 16$.

Let n be the product of two primes p, q. What is the value of $\phi(n)$?

Affine cipher

How many keys are valid for a generic affine cipher where $a, b \in \mathbb{Z}_m$?
Answer: The number of valid keys is $m \cdot \phi(m)$.

Let

$$m = \prod_{i=1}^{n} p_i^{e_i},$$

where primes p_i are pairwise distinct with $e_i > 0$. Then

$$\phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1})$$

.
Example: If $60 = 2^2 \cdot 3 \cdot 5$, $\phi(60) = (4 - 2) \cdot (3 - 1) \cdot (5 - 1) = 16$.

Let n be the product of two primes p, q. What is the value of $\phi(n)$?
Answer: $\phi(n) = (p - 1)(q - 1)$.

Polyalphabetic substitution cipher

Definition

A **polyalphabetic substitution** maps disjoint sets of letter with different permutations π_i.

Formalization:

- The key is the set of permutatons $\Pi = (\pi_1, \pi_2, \ldots, \pi_t)$.
- Encryption is given by $Enc_\Pi(x) = (\pi_1(x_1), \pi_2(x_2), \ldots, \pi_t(x_t))$.
- Decryption is analogous.

Observation: The frequency of symbols is distorted!

Polyalphabetic shift cipher (Vigenère)

Definition

A **polyalphabetic shift cipher** is a special case of a polyalphabetic substitution cipher where each permutation is a shift cipher.

Formalization:

- Define $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_m)^t$.
- The key is the set $k = (k_1, k_2, \ldots, k_t)$.
- Encryption is given by $y = Enc_k(x) = (x_1 + k_1, x_2 + k_2, \ldots, x_t + k_t)$.
- Decryption is given by $x = Dec_k(y) = (y_1 - k_1, y_2 - k_2, \ldots, y_t - k_t)$.

Note: How big is the key space?

Polyalphabetic shift cipher (Vigenère)

Definition

A **polyalphabetic shift cipher** is a special case of a polyalphabetic substitution cipher where each permutation is a shift cipher.

Formalization:

- Define $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_m)^t$.
- The key is the set $k = (k_1, k_2, \ldots, k_t)$.
- Encryption is given by $y = Enc_k(x) = (x_1 + k_1, x_2 + k_2, \ldots, x_t + k_t)$.
- Decryption is given by $x = Dec_k(y) = (y_1 - k_1, y_2 - k_2, \ldots, y_t - k_t)$.

Note: The key space is m^t.

Hill cipher (1929)

Definition

It is a polyalphabetic substitution cipher that divides the characters in blocks of t letters and applies t linear combinations of the t letters.

Example: If $t = 2$, a plaintext element can be written as (x_1, x_2) and a ciphertext element as $(y_1 = 11x_1 + 3x_2 \bmod m, y_2 = 8x_1 + 7x_2 \bmod m)$.

Formalization:

- The key $k = K$ is the matrix of linear combinations.
- Encryption is given by $y = Enc_k(x) = xK$.
- Decryption?

Important: What is the restriction on matrix K?

Hill cipher (1929)

Definition

It is a polyalphabetic substitution cipher that divides the characters in blocks of t letters and applies t linear combinations of the t letters.

Example: If $t = 2$, a plaintext element can be written as (x_1, x_2) and a ciphertext element as $(y_1 = 11x_1 + 3x_2 \bmod m, y_2 = 8x_1 + 7x_2 \bmod m)$.

Formalization:

- The key $k = K$ is the matrix of linear combinations.
- Encryption is given by $y = Enc_k(x) = xK$.
- Decryption is given by $x = Dec_k(y) = yK^{-1}$.

Important: Matrix K needs to be invertible in $\mathbb{Z}_m$.

Transposition cipher

Definition

A **transposition** $Enc_\theta : \mathcal{M} \to \mathcal{C}$ changes the position i of each letter x_i in message x by $\theta(i)$, where θ defines a permutation in the set $\{1, 2, \ldots, n\}$, $n = |x|$.

Formalization:

- The key is the permutation $\theta : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$.
- Encryption is given by $Enc_\theta(x) = (x_{\theta(1)}, x_{\theta(2)}, \ldots, x_{\theta(n)})$.
- Decryption is given by $Dec_\theta(y) = (y_{\theta^{-1}(1)}, y_{\theta^{-1}(2)}, \ldots, y_{\theta^{-1}(n)})$.

Observations:

- The key space has size $n!$ and the key in general is smaller than n.

Important: The transposition cipher is a special case of the Hill cipher!

Stream cipher (Vernam, 1919)

Definition

A **stream cipher** is a polyalphabetic shift cipher where the number of shifts is identical to the plaintext length.

Formalization:

- The key is the string $(k_1, k_2, \ldots, k_n)$.
- Each letter defines a different shift.

Classification:

- **Periodic**: se $k_{i+d} = k_i$ para $d < n, i \geq 1$.
- **Synchronous**: the key stream is constructed independently from the plaintext.
- **Asynchronous**: the key stream is derived from the plaintext and ciphertext.

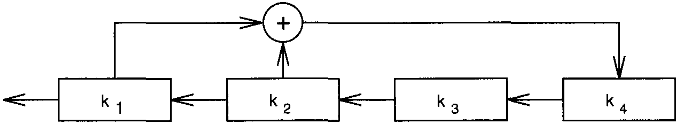
Stream cipher

Comments:

- A block cipher can be seen as a stream cipher with constant key stream!
- The Vigenère cipher is periodic with period t.
- Stream ciphers are usually instantiated in the binary case ($\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_2$).

Stream cipher

Synchronous example: *Linear Feedback Shift Register*



Asynchronous example: Auto-key cipher

- The key stream k is $k_1 = r, k_i = x_{i-1}, i \geq 2$.
- Encryption is given by $y_i = Enc_k(x_i) = x_i + k_i \bmod m$.
- Decryption is given by $x_i = Dec_k(y_i) = y_i - k_i \bmod m$.
- Key space again has size m.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.
5. **Polyalphabetic shift cipher**: index of coincidences and frequency analysis.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.
5. **Polyalphabetic shift cipher**: index of coincidences and frequency analysis.
6. **Hill cipher**: known-plaintext attack.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.
5. **Polyalphabetic shift cipher**: index of coincidences and frequency analysis.
6. **Hill cipher**: known-plaintext attack.
7. **Transposition cipher**: solution of anagram, known-plaintext attack.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.
5. **Polyalphabetic shift cipher**: index of coincidences and frequency analysis.
6. **Hill cipher**: known-plaintext attack.
7. **Transposition cipher**: solution of anagram, known-plaintext attack.
8. **LFSR**: solution of linear equations.

Cryptanalysis

1. **Shift cipher**: exhaustive search in $\mathcal{K}$.
2. **Monoalphabetic substitution cipher**: frequency analysis.
3. **Affine cipher**: frequency analysis and linear equations.
4. **Polyalphabetic substitution**: partition and frequency analysis.
5. **Polyalphabetic shift cipher**: index of coincidences and frequency analysis.
6. **Hill cipher**: known-plaintext attack.
7. **Transposition cipher**: solution of anagram, known-plaintext attack.
8. **LFSR**: solution of linear equations.
9. **Auto-key cipher**: exhaustive search in $\mathcal{K}$.