Classical Cryptography

CSG 252     Lecture 1

September 16, 2008

Riccardo Pucella
Goals of Classical Cryptography

- Alice wants to send message $X$ to Bob
- Oscar is on the wire, listening to all communications
- Alice and Bob share a key $K$
- Alice encrypts $X$ into $Y$ using $K$
- Alice sends $Y$ to Bob
- Bob decrypts $Y$ back to $X$ using $K$

- Want to protect message $X$ from Oscar
- Much better: protect key $K$ from Oscar
Shift Cipher

- Given a string M of letters
  - For simplicity, assume only capital letters of English
  - Remove spaces

- Key k: a number between 0 and 25

- To encrypt, replace every letter by the letter k places down the alphabet (wrapping around)

- To decrypt, replace every letter by the letter k places up the alphabet (wrapping around)

- Example: k=10, THISISSTUPID ➔ DRSCSCCCDEZSN
Definition of Cryptosystem

A cryptosystem is a tuple \((P,C,K,E,D)\) such that:

1. \(P\) is a finite set of possible plaintexts
2. \(C\) is a finite set of possible ciphertexts
3. \(K\) is a finite set of possible keys (keyspace)
4. For every \(k\), there is an encryption function \(e_k \in E\) and decryption function \(d_k \in D\) such that \(d_k(e_k(x)) = x\) for all plaintexts \(x\).

Encryption function assumed to be injective

Encrypting a message:

\[x = x_1 \ x_2 \ ... \ x_n \rightarrow e_k(x) = e_k(x_1) \ e_k(x_2) \ ... \ e_k(x_n)\]
Properties of Cryptosystems

- Encryption and decryption functions can be efficiently computed.
- Given a ciphertext, it should be difficult for an opponent to identify the encryption key and the plaintext.
- For the last to hold, the key space must be large enough! Otherwise, may be able to iterate through all keys.
Shift Cipher, Revisited

$P = \mathbb{Z}_{26} = \{0, 1, 2, \ldots, 25\}$

Idea: $A = 0, B = 1, \ldots, Z = 25$

$C = \mathbb{Z}_{26}$

$K = \mathbb{Z}_{26}$

$e_k = ?$

Add $k$, and wraparound...
Modular Arithmetic

### Congruence

- $a, b$: integers  $m$: positive integer
- $a \equiv b \pmod{m}$ iff $m$ divides $a-b$
- $a$ congruent to $b$ modulo $m$

**Examples:** $75 \equiv 11 \pmod{8}$  $75 \equiv 3 \pmod{8}$

- Given $m$, every integer $a$ is congruent to a unique integer in $\{0,...,m-1\}$
- Written $a \pmod{m}$
- Remainder of $a$ divided by $m$
Modular Arithmetic

\( Z_m = \{ 0, 1, \ldots, m-1 \} \)

Define \( a + b \) in \( Z_m \) to be \( a + b \pmod{m} \)

Define \( a \times b \) in \( Z_m \) to be \( a \times b \pmod{m} \)

Obeys most rules of arithmetic

+ commutative, associative, 0 additive identity
x commutative, associative, 1 mult. identity
+ distributes over \( \times \)

Formally, \( Z_m \) forms a ring

For a prime \( p \), \( Z_p \) is actually a field
Shift Cipher, Formally

- \( P = \mathbb{Z}_{26} = \{0, 1, 2, \ldots, 25\} \) (where A=0, B=1, ..., Z=25)
- \( C = \mathbb{Z}_{26} \)
- \( K = \mathbb{Z}_{26} \)
- \( e_k(x) = x + k \pmod{26} \)
- \( d_k(y) = y - k \pmod{26} \)

Size of the keyspace? Is this enough?
Affine Cipher

Let’s complicate the encryption function a little bit

$K = \mathbb{Z}_{26} \times \mathbb{Z}_{26}$ (tentatively)

$e_k(x) = (ax + b) \mod 26$, where $k=(a,b)$

How do you decrypt?

Given $a, b,$ and $y$, can you find $x \in \mathbb{Z}_{26}$ such that

$(ax+b) \equiv y \pmod{26}$?

or equivalently: $ax \equiv y-b \pmod{26}$?
**Affine Cipher**

**Theorem:** $ax \equiv y \pmod{m}$ has a unique solution $x \in \mathbb{Z}_m$ iff $\gcd(a,m)=1$

- In order to decrypt, need to find a unique solution
- Must choose only keys $(a,b)$ such that $\gcd(a,26)=1$
- Let $a^{-1}$ be the solution of $ax = 1 \pmod{m}$
- Then $a^{-1}b$ is the solution of $ax = b \pmod{m}$
Affine Cipher, Formally

\[ P = C = \mathbb{Z}_{26} \]

\[ K = \{ (a,b) \mid a,b \in \mathbb{Z}_{26}, \gcd(a,26)=1 \} \]

\[ e_{(a,b)}(x) = ax + b \pmod{26} \]

\[ d_{(a,b)}(y) = ? \]

What is the size of the keyspace?

\[ (\text{Number of } a\text{'s with } \gcd(a,26)=1) \times 26 \]

\[ \varphi(26) \times 26 \]
Substitution Cipher

- $P = \mathbb{Z}_{26}$
- $C = \mathbb{Z}_{26}$
- $K = \text{all possible permutations of } \mathbb{Z}_{26}$
  - A permutation $P$ is a bijection from $\mathbb{Z}_{26}$ to $\mathbb{Z}_{26}$
- $e_k(x) = k(x)$
- $d_k(x) = k^{-1}(x)$

Example
- Shift cipher, affine cipher

Size of keyspace?
Cryptanalysis

• Kerckhoff’s Principle:
  • The opponent knows the cryptosystem being used
  • No “security through obscurity”

• Objective of an attacker
  • Identify secret key used to encrypt a ciphertext

• Different models are considered:
  • Ciphertext only attack
  • Known plaintext attack
  • Chosen plaintext attack
  • Chosen ciphertext attack
Cryptanalysis of Substitution Cipher

- Statistical cryptanalysis
  - Ciphertext only attack
- Again, assume plaintext is English, only letters
- Goal of the attacker: determine the substitution
- Idea: use statistical properties of English text
Statistical Properties of English

Letter probabilities (Beker and Piper, 1982): $p_0, ..., p_{25}$
A: 0.082, B: 0.015, C: 0.028, ...

More useful: ordered by probabilities:
E: 0.120
T, A, O, I, N, S, H, R: [0.06, 0.09]
D, L: 0.04
C, U, M, W, F, G, Y, P, B: [0.015, 0.028]
V, K, J, X, Q, Z: < 0.01

Most common digrams: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST...

Most common trigrams: THE, ING, AND, HER, ERE, ENT,...
Statistical Cryptanalysis

General recipe:

- Identify possible encryptions of E (most common English letter)
  - T,A,O,I,N,S,H,R: probably difficult to differentiate
- Identify possible digrams starting/finishing with E (-E and E-)
- Use trigrams
- Find ‘THE’
- Identify word boundaries
Polyalphabetic Ciphers

- Previous ciphers were **monoalphabetic**
  - Each alphabetic character mapped to a unique alphabetic character
  - This makes statistical analysis easier
- Obvious idea
  - Polyalphabetic ciphers
  - Encrypt multiple characters at a time
Vigenère Cipher

Let $m$ be a positive integer (the key length)

$P = C = K = Z_{26} \times \ldots \times Z_{26} = (Z_{26})^m$

For $k = (k_1, \ldots, k_m)$:

- $e_k(x_1, \ldots, x_m) = (x_1 + k_1 \pmod{26}, \ldots, x_m + k_m \pmod{m})$
- $d_k(y_1, \ldots, y_m) = (y_1 - k_1 \pmod{26}, \ldots, y_m - k_m \pmod{m})$

Size of keyspace?
Cryptanalysis of Vigenère Cipher

Thought to thwart statistical analysis, until mid-1800

Main idea: first figure out key length \( m \)

Two identical segments of plaintext are encrypted to the same ciphertext if they are \( \delta \) position apart, where \( \delta = 0 \pmod{m} \)

Kasiski Test: find all identical segments of length \( \geq 3 \) and record the distance between them: \( \delta_1, \delta_2, \ldots \)

\( m \) divides \( \gcd(\delta_1, \delta_2, \ldots) \)
We can get further evidence for the value of m as follows.

The index of coincidence of a string $X = x_1...x_n$ is the probability that two random elements of $X$ are identical.

Written $I_c(X)$

Let $f_i$ be the # of occurrences of letter $i$ in $X$; $I_c(X) = ?$

For an arbitrary string of English text, $I_c(X) \approx 0.065$

If $X$ is a shift ciphertext from English, $I_c(X) \approx 0.065$

For $m=1,2,3,...$ decompose ciphertext into substrings $y_i$ of all $m^{th}$ letters; compute $I_c$ of all substrings.

$I_c$s will be $\approx 0.065$ for the right $m$

$I_c$s will be $\approx 0.038$ for wrong $m$
Then what?

- Once you have a guess for m, how do you get keys?
- Each substring $y_i$:
  - Has length $n' = n/m$
  - Encrypted by a shift $k_i$
  - Probability distribution of letters: $f_0/n', \ldots, f_{25}/n'$
  - $f_{0+k_i \pmod{26}}/n', \ldots, f_{25+k_i \pmod{26}}/n'$ should be close to $p_0, \ldots, p_{25}$
- Let $M_g = \sum_{i=0,\ldots,25} p_i \left( f_{i+g \pmod{26}} / n' \right)$
  - If $g = k_i$, then $M_g \approx 0.065$
  - If $g \neq k_i$, then $M_g$ is usually smaller
15 minutes break
**Hill Cipher**

- A more complex form of polyalphabetic cipher
- Again, let \( m \) be a positive integer
- \( P = C = (\mathbb{Z}_{26})^m \)

To encrypt:  \( \text{ (case } m=2\text{)} \)

- Take linear combinations of plaintext \( (x_1, x_2) \)
- E.g., \( y_1 = 11 \times x_1 + 3 \times x_2 \pmod{26} \)
  \( y_2 = 8 \times x_1 + 7 \times x_2 \pmod{26} \)

- Can be written as a matrix multiplication \( \pmod{26} \)
Hill Cipher, Continued

- \( K = \text{Mat} (\mathbb{Z}_{26}, m) \) (tentatively)

- \( e_k (x_1, \ldots, x_m) = (x_1, \ldots, x_m) k \)

- \( d_k (y_1, \ldots, y_m) = ? \)

- Similar problem as for affine ciphers

- Want to be able to reconstruct plaintext

- Solve \( m \) linear equations (mod 26)

- I.e., find \( k^{-1} \) such that \( kk^{-1} \) is the identity matrix

- Need a key \( k \) to have an inverse matrix \( k^{-1} \)
Cryptanalysis of Hill Cipher

- Much harder to break with ciphertext only
- Easy with known plaintext
- Recall: want to find secret matrix $k$
- Assumptions:
  - $m$ is known
  - Construct $m$ distinct plaintext-ciphertext pairs
    - $(X_1, Y_1), \ldots, (X_m, Y_m)$
  - Define matrix $Y$ with rows $Y_1, \ldots, Y_m$
  - Define matrix $X$ with rows $X_1, \ldots, X_m$
  - Verify: $Y = X \cdot k$
  - If $X$ is invertible, then $k = X^{-1} \cdot Y$!
Stream Ciphers

The cryptosystems we have seen until now are block ciphers
Characterized by $e_k(x_1, ..., x_n) = e_k(x_1), ..., e_k(x_n)$
An alternative is stream ciphers
Generate a stream of keys $Z = z_1, ..., z_n$
Encrypt $x_1, ..., x_n$ as $e_{z_1}(x_1), ..., e_{z_n}(x_n)$
Stream ciphers come in two flavors
- Synchronous stream ciphers generate a key stream from a key independently from the plaintext
- Non-synchronous stream ciphers can depend on plaintext
A **synchronous stream cipher** is a tuple \((P,C,K,L,E,D)\) and a function \(g\) such that:

- \(P\) and \(C\) are finite sets of plaintexts and ciphertexts
- \(K\) is the finite set of possible keys
- \(L\) is a finite set of keystream elements
- \(g\) is a keystream generator, \(g(k)=z_1z_2z_3\ldots, z_i\in L\)

For every \(z\in L\), there is \(e_z\in E\) and \(d_z\in D\) such that

\(d_z(e_z(x)) = x\) for all plaintexts \(x\)

Vigenère Cipher as a Stream Cipher

\[ P = C = L = \mathbb{Z}_{26} \]

\[ K = (\mathbb{Z}_{26})^m \]

\[ e_z(x) = x + z \pmod{26} \]

\[ d_z(y) = y - z \pmod{26} \]

\[ g(k_1, \ldots, k_m) = k_1 k_2 \ldots k_m k_1 k_2 \ldots k_m k_1 k_2 \ldots k_m \ldots \]

This is a periodic stream cipher with period \( m \)

\[ z_{i+m} = z_i \text{ for all } i \geq 1 \]
Linear Feedback Cipher

Here is a way to generate a synchronous stream cipher

- Take $P = C = L = Z_2 = \{0, 1\}$ (binary alphabet)
- Note that addition mod 2 is just XOR
- $K = (Z_2)^{2m}$
- A key is of the form $(k_1, ..., k_m, c_0, ..., c_{m-1})$
- $e_z(x) = x + z \pmod{2}$  $d_z(y) = y - z \pmod{2}$
- $g(k_1, ..., k_m, c_0, ..., c_{m-1}) = z_1z_2z_3...$ defined as follows:
  - $z_1 = k_1, ..., z_m = k_m;$
  - $z_{i+m} = \sum_{j=0,...,m-1} c_jz_{i+j} \pmod{2}$
- If $c_0, ..., c_{m-1}$ are carefully chosen, period of the keystream is $2^{m-1}$
- Advantage: can be implemented very efficiently in hardware
  - For fixed $c_0, ..., c_{m-1}$
Cryptanalysis of Linear Feedback Cipher

- Just like Hill cipher, susceptible to a known plaintext attack
  - And for the same reason: based on linear algebra

- Given m, and pairs \(x_1, x_2, \ldots, x_n\) and \(y_1, y_2, \ldots, y_n\) of plaintexts and corresponding ciphertexts

- Suppose \(n \geq 2m\)

- Note that \(z_i = x_i + y_i \pmod{2}\) by properties of XOR

- This gives \(k_1, \ldots, k_m\); remains to find \(c_0, \ldots, c_{m-1}\)

- Using \(z_{i+m} = \sum_{j=0,\ldots,m-1} c_j z_{i+j} \pmod{2}\), we get m linear equations in m unknowns \((c_0, \ldots, c_{m-1})\), which we can solve
Autokey Cipher

A simple example of a non-synchronous stream cipher

\[ P = C = K = L = Z_{26} \]

\[ e_z(x) = x + z \pmod{26} \]

\[ d_z(x) = x - z \pmod{26} \]

The keystream corresponding to key \( k \) is

\[ z_1 = k \]

\[ z_i = x_{i-1} \text{ for all } i \geq 2. \]

where \( x_1, x_2, x_3, \ldots \) is the sequence of plaintext

What's the problem?