ECE568 Lecture 04:
Introduction to Cryptography

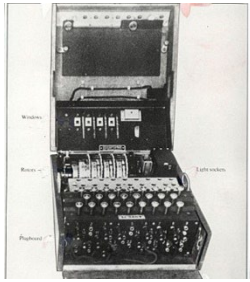
Wei Huang
Department of Electrical and Computer Engineering
University of Toronto

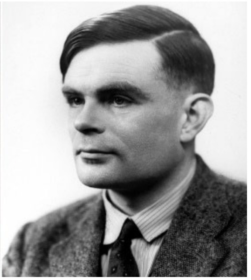
Lecture Outline

- Cryptography
  - What can it do
- Basic Ciphers:
  - Substitution Ciphers
  - Poly-Alphabetic and Periodic Ciphers
  - One-Time Pad and Vernam Ciphers
- Block Cipher vs. Stream Ciphers

Cryptographic Algorithm as Weapons

- Enigma machine and Alan Turing





- Export control, Phillip Zimmerman and PGP

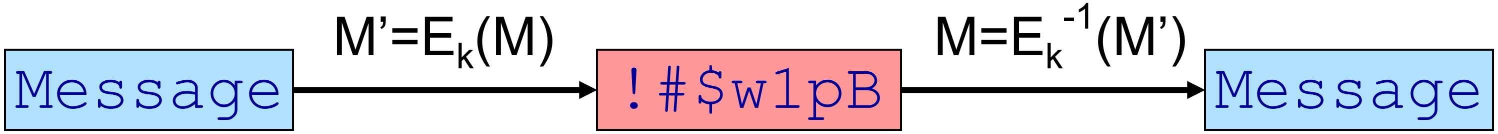
Cryptography

- Cryptography literally means “secret writing”, but in reality cryptography is capable of much more than keeping data secret.

- It’s main use is in protecting the **storage and transmission of data**. Cryptographic mechanisms are used to establish 4 properties:
  1. **Confidentiality**: Secrecy of the data (already discussed). This is provided by a set of devices called **Ciphers**.
  2. **Integrity**: The trustworthiness of the data (also already discussed). Provided by **Hashes**.
  3. **Authentication**: Allows a principal (user or machine) to prove their identity, or the origin of a piece of data. Provided by **Message Authentication Codes (MAC’s)** and **Signatures**.
  4. **Non-repudiation:** Prevent that principal from later denying that they performed the action. This can be achieved with the help of a **Trusted 3rd Party**.

Ciphers

- A cipher is an algorithm that can be used to obfuscate information so that it appears random to anyone who does not possess special information called a key.



- Ciphers are based on a class of functions called **trapdoor one-way functions.**
  - A one-way function is a function that is easy to compute, but whose inverse is difficult to compute.
  - The “trapdoor” means that given a special piece of information (the key), the inverse will also become easy to compute.
- *Interesting fact:*
  - It has never been proven (or disproven) that one-way functions really exist. Ciphers are based on functions that are believed to be one-way because no one has ever shown an easy way of computing the inverse.
- Examples of well-known one-way functions: Factoring, discrete log

Candidate one-way functions

- Factoring: $f(x,y) = x*y$, $f^{-1}(z) = \{x,y\}$
  such that $x*y = z$

- Discrete log: $f(x,y,n) = x^y \bmod n$, $f^{-1}(z,x,m) = y$
  such that $x^y \bmod n = z$

How to Use One-Way Function

Example

Kerckhoff’s Principle

The security of any given encryption system must depend only on the secrecy of the key, **K**, and not on the secrecy of the algorithm

- Algorithms are hard to change: compiled into software, wired into circuits
- An algorithm can be audited studied and publicly known without making it useless. Users just need to keep the keys secret.
- Deployed implementations need only protect the key instead of the whole algorithm

Brute-Force Attack

Why it's a bad idea…

Key Size	Number of Possible Keys	Time Required at 10^6 tests/sec	Time Required at 10^{12} tests/sec
32 bits	$2^{32} = 4.3 \bullet 10^9$	36 minutes	2.2 ms
56 bits	$2^{56} = 7.2 \bullet 10^{16}$	1142 years	10 hours
128 bits	$2^{128} = 3.4 \bullet 10^{38}$	$5.4 \bullet 10^{24}$ years	$5.4 \bullet 10^{18}$ years
168 bits	$2^{168} = 3.7 \bullet 10^{50}$	$5.9 \bullet 10^{36}$ years	$5.9 \bullet 10^{30}$ years
26 characters (permutation)	$26! = 4 \times 10^{26}$	6.4×10^{12} years	$6.4 \bullet 10^6$ years

Substitution Cipher and Caesar (Shift) Cipher

- When Caesar mounted his campaign against the Gauls (modern day France), he needed a way to communicate with his troops securely. He contrived a very simple cipher:
  - Take each letter, and replace it with the letter shifted 3 letters to the right in the alphabet. If there are no more letters, wrap around to the beginning of the alphabet. This is similar to modern day rot13 used on many UNIX systems which rotates 13 letters instead of 3. Decryption is just the inverse.

- The Caesar (shift) cipher is a particular instance of a class of ciphers called substitution ciphers.
  - Each **plain text** letter is replaced with exactly one **cipher text** letter.
  - The **key** is the mapping between plain text letters and cipher text letters.

Examples

Attacks on Ciphers

- When attacking a cipher, usually the adversary can have several goals:
  - Adversary wants to get the plain text corresponding to a cipher text
  - Adversary wants to get the key (and possibly the algorithm if it's not public)

- When analyzing the strength of ciphers, cryptographers categorize into strength against several classes of attacks (hardest to easiest):
  1. **Cipher Text only:** Adversary only has access to cipher texts.
  2. **Known Plain Text/Cipher Text Pairs:** Adversary has access to some number of plain text/cipher text pairs. The more pairs it takes to crack the cipher, the stronger the cipher is.
  3. **Chosen Plain Text/Cipher Text:** The adversary can pick a plain text and get corresponding cipher text or vice-versa. Thus, adversary can **adaptively** select plain texts/cipher texts that help her break the cipher.

Breaking Substitution Ciphers

- These ciphers are very easy to break. The weakness is that **every letter in the plain text alphabet always gets encrypted to the same letter in the cipher text alphabet.** If the attacker knows the original message is in English, then:
  - E has probability 0.12
  - TAOINSHR have probability 0.06 - 0.09
  - DL have probability ~ 0.04
  - Etc…

- By performing **frequency analysis** on the cipher text, the attacker can decode common letters. Then by matching against common English words, the attacker can recover the plain text and eventually recover they key

- These attacks are easy enough that they can (and were) done by hand. With the help of computers, they are effortless to break.

Improving Substitution Ciphers

- The downfall of the substitution cipher was that it did not hide any frequency information because every plain text letter always encrypted to the same cipher text letter. To break this dependence:
  - Instead of having one mapping, have a set of *n* mappings, and change the mapping with every character. When we have used up the *n* mappings, then start from the first mapping and repeat. This is called a **Poly-alphabetic Cipher.**

- Because of this repetition, these ciphers are sometimes called **periodic ciphers**:
  - If the attacker knows, or can somehow guess the period, an attack is possible. For small *n* this is only incrementally harder than a plain substitution cipher, but for large *n* this can be very difficult.
  - The attack requires more cipher text examples, but becomes easier if the adversary has a plain text/cipher text pair.

The Vernam Cipher or One Time Pad

- A *theoretically unbreakable* cipher exists, called a **One Time Pad (OTP)**, also called the **Vernam Cipher** after its creator. It is essentially a polyalphabetic cipher that never repeats – for every character, a randomly chosen substitution is used.
  - OTP's are implemented at the bit level (alphabet of 2 symbols). The cipher requires the key to be the same length as the message to be encrypted. The cipher text is created by computing the bitwise XOR of the plain text and the key.
  - XOR will flip the plain text bit if the key has a corresponding 1, and the leave it if the key has a corresponding 0. If the key is well chosen (i.e. random), the cipher text is the plain text with randomly flipped bits. For a message containing *n* bits of information, the OTP adds exactly *n* bits of randomness creating a completely random cipher text.
  - OTP is considered information theoretically secure.

The Vernam Cipher or One Time Pad

- The disadvantage with the OTP is that the key is as long as the message, making it impractical for most applications (bandwidth overheads of 100% are not generally acceptable).
  - It is called a OTP because each key can only be used once, thus for every message bit sent, one key bit must also be transmitted (separately). If any key is used to encrypt more than one message, the security is reduced significantly.
  - In addition, it is **malleable,** meaning that flipping a bit in the cipher text flips a bit in the plain text. As a result, it is easy to tamper with, and **must** be combined with some sort of integrity check if used.
  - Key must be random, so you need a good source of randomness.

The Vernam Cipher or One Time Pad

- How strong is OTP against:
  1. Cipher Text only attack:
  2. Known Ciphertext/Plaintext attack:
  3. Chosen Ciphertext/Plaintext attack:

Practical Ciphers

- Practical Ciphers:
  - Will have fixed length keys that are much shorter than the message (and do not depend on message length).
  - Be efficient for encryption and decryption.
  - Have cipher texts which are computationally difficult to decrypt without the key. Note: “computationally difficult” is a moving target, as computers are getting more and more powerful.

Block vs. Stream Ciphers

- Symmetric key ciphers (ciphers that use the same key to encrypt and decrypt) fall into 2 broad categories:
  - **Block Ciphers:** These encrypt a block of plain text at a time, usually 64 bits or some multiple of that. To use these, the plain text is divided into blocks and each is encrypted separately. The last block might need to be "padded" to make it a full block length.
  - **Stream Ciphers:** These are more similar to OTP's. A key is used to generate a pseudo-random sequence of bits, which are then XOR'ed with the cipher text. In this way, the plain text can be encrypted a bit at a time, making it useful for streaming applications (such as voice or video). These ciphers suffer from synchronization problems, if any bits are lost, the entire stream might have to be resent.

Lecture Outline

- Block Ciphers
  - Cipher Techniques
  - The DES Cipher
  - The AES Cipher

Block Ciphers

- Block ciphers are more commonly used than Stream Ciphers. The reasons for this are unfortunately not entirely logical:
  - In the past, most stream ciphers remained proprietary, so they could not be analyzed. People could not be confident of their security. In contrast, there are many publicly available and well-studied block ciphers.

- In a 1949 paper Communication Theory of Secrecy Systems, Claude Shannon outlined two basic techniques for a good cryptosystem: **confusion** and **diffusion.**

Properties: Confusion

Confusion: obscuring of the relationship between the plaintext and the ciphertext

- Primary goal is to make statistical analysis difficult, even if the attacker has a large number of known plaintext/ciphertext pairs
- Encoding should be **non-linear**:

$$E_k(M_1 + M_2) \neq E_k(M1) + E_k(M_2)$$

- Each character of the ciphertext should depend on the entire key

Properties: Diffusion

Diffusion: spreading the influence of individual plaintext characters over much of the ciphertext

- Each output bit is affected by many input bits
- Ideally, flipping a single bit of either the key or the plaintext should change half the output bits (*i.e.*, the probability of bit_i flipping is 0.5 for any value of i)
- Repetitive patterns in plaintext are spread over the entire ciphertext, thus hiding statistical information about the plaintext

Common Block Ciphers

- The two most common block ciphers in use are those specified by NIST (National Institute of Standards and Technology)
  - **DES: D**ata **E**ncryption **S**tandard
  - **AES: A**dvanced **E**ncryption **S**tandard. Replaced DES as the official standard encryption algorithm in 2000.
  - Both are **iterated block ciphers** meaning they work by performing the same operation over and over.

- Many others exist but there is not enough time to cover them.

Some History

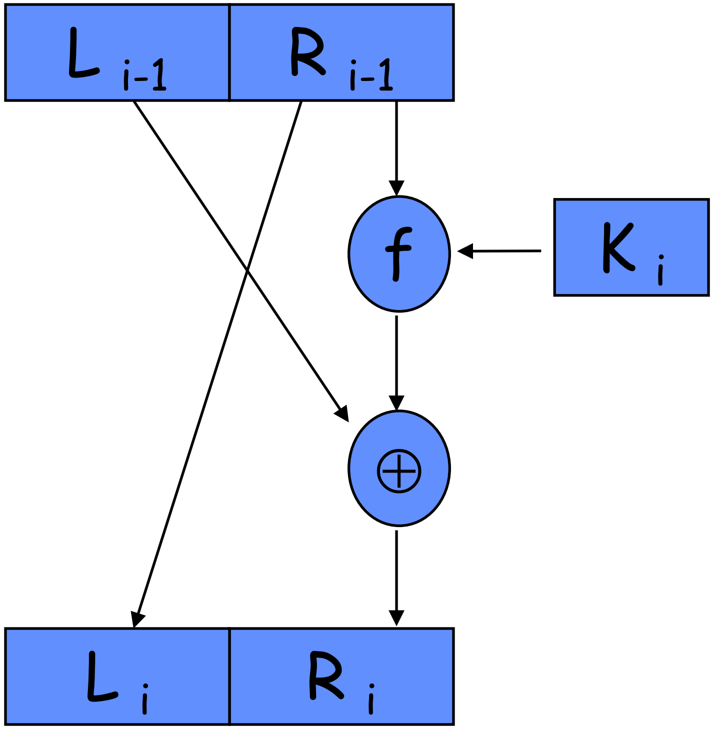
- Early in the 70's, non-military encryption was a big mess. An effort was made by NIST to standardize encryption with a reliably strong, well-studied cipher.
  - IBM produced a candidate, originally called Lucifer, which eventually became designated DES on November 23, 1976.
  - The standard provided strong security, as well as good performance. It was easy to implement and could be used in a variety of applications
  - It was evaluated and pronounced secure by the NSA. The involvement of the NSA created much distrust and speculation at the time.
- DES uses a 56-bit key and has a block length of 64-bits.
  - Key is too small to be used given modern computers. DES is **no longer** considered secure!

Structure of DES

- DES uses what is known as a **Feistel Network**:
  - The network consists of several "rounds." In each round, the input is split into a left half *L*, and a right half *R*.
  - The two halves are switched, and some computation is performed on one of the halves (each round only modifies half of the input bits).
  - Each round also includes computation with a portion of the key, called a **sub-key** $\boldsymbol{K_i}$. The output of one round becomes the input for the next.

- The DES standard specifies that this structure is repeated for **16 rounds**.

One Round of DES



Feistel Network

- Pros: it is guaranteed to produce a decryptable (i.e. invertable) function no mater what is used in *f*. Why?
- Cons: it is inefficient. Each round only modifies half of the block.

Key Schedule in DES

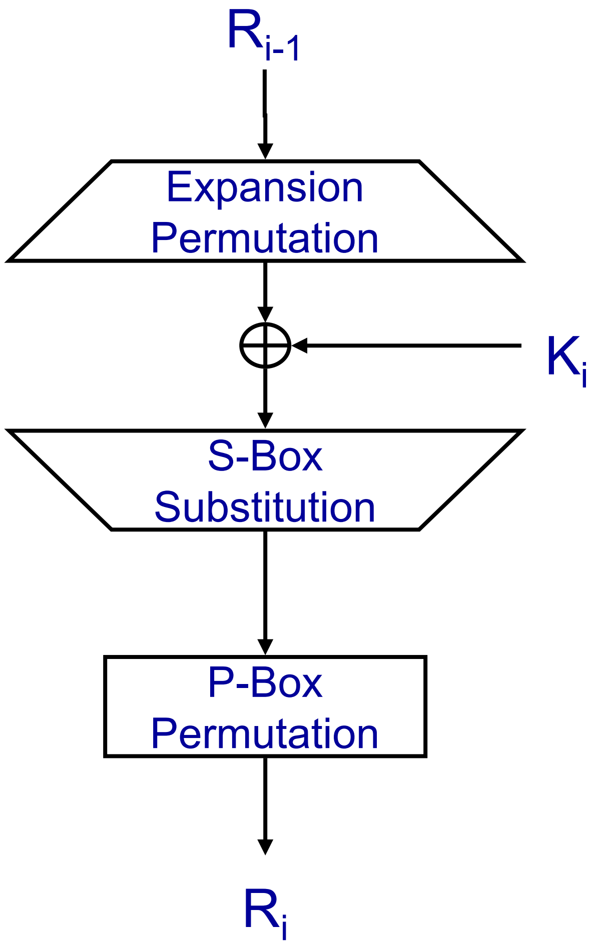
- The 56-bit key is put through a key-schedule to create 16 sub keys $\boldsymbol{K_i}$, which are used in each round of the cipher:
  1. The 56-bit key is split into two 28-bit halves
  2. Each half is shifted left by 1 or 2 bits (depending on which subkey)
  3. 24-bits are selected from each of the 2 28-bit halves (again depends on the round) to make a 48-bit subkey

- Exact number of shifts and bit selections are selected through “pseudorandom black magic”:
  - Arbitrary constant used to pick sections, the constants are selected because they have a random distribution
  - Can lookup specific details in any standard crypto reference

Computation on Each Round

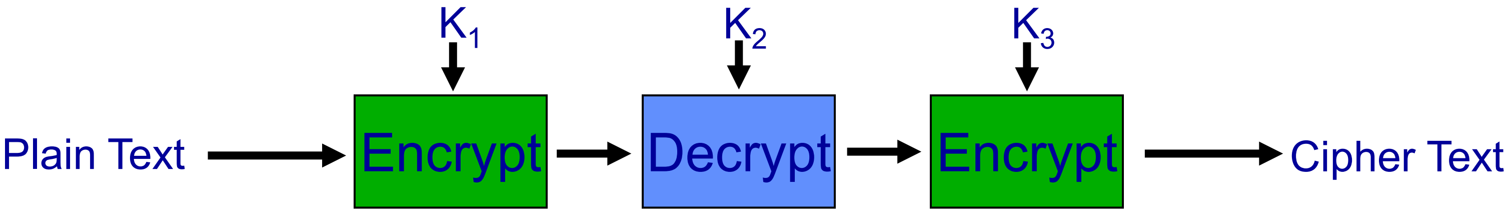
- Each *f* function contains
  1. Expansion permutation that expands 32-bit input into 48-bits
  2. XOR with a 48-bit sub-key (48 bits generated from the 56-bit key). The sub-keys are derived using a key-schedule algorithm.
  3. S-box substitution boxes that substitute and compress into 32-bit output
  4. Permutation of 32-bits

- Design of the S-boxes is very important as this is the only non-linear element in the cipher:
  - Contents are supposed to be essentially random, but were actually set by the NSA using a secret and unknown method



Problems with DES

- DES had some problems which started to emerge over time:
  - DES used a 56-bit key which was adequate at the time. However, as computers became more powerful, it became realistic to brute-force (try all 2^{56} key combinations) to decrypt cipher text without knowing the key.
  - Variants were created that used longer key lengths by running the same DES algorithms multiple times. Most applications use 3DES (triple-DES) which takes a 168-bit key and runs the algorithm 3 times using 56-bits of the key each time. Note: weaknesses mean that 3DES only has an effective key length of 112 bits.



Meet in the middle attack on 2DES

- Why did we skip 2DES and go directly to 3DES?
  - Adversary can do a known PT/CT attack that effectively reduces the key space to search by near 56-bits
  - Adversary trades off storage space to reduce computational effort.