

Encryption

Tom Chothia
Computer Security: Lecture 2

Encryption 1

- Symmetric Key Encryption Ciphers
 - Frequency Analysis
 - One time pads
 - AES, DES and 3-DES
- Block cipher modes
- Truecrypt

Caesar Cipher

- One of the first codes was used by Julius Caesar.
- The Caesar Cipher replaces each letter of the alphabet with one three to the right, i.e.
 - a becomes d,
 - b becomes e,
 -
 - z becomes c.

Using a Key

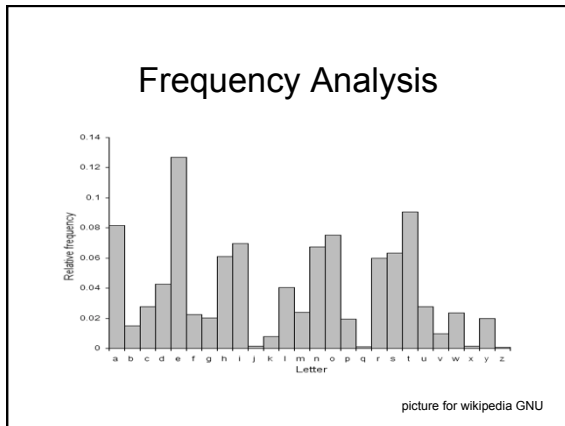
- These ciphers are easy to break because as soon as you know the scheme you can decrypt the message.
- Modern encryption schemes use a “key”.
- The scheme is public but it produces different results for each key.

Using a Key

- For instance, we can use the Caesar cipher rotating “n” rotations.
- But only 26 possible keys so you can just try them all (breaking the cipher is 26 times harder without the key).
- A better scheme replaces each letter with another letter. Here there are $26! \approx 4 \times 10^{26}$ possible keys.

Frequency analysis

- While hard to break by brute force, replacing each letter with another is easy to break using *frequency analysis*.
- Frequency analysis counts the number of times each symbol occurs and tries to draw conclusions from this.



One Time Pads

- Perfect encryption
- Needs a key as long as the message.
- XOR/add the key and the message:

Message: HELLO ALICE
 Key: SGFKPQYEIJ
 Cipher text: ALRWERKNLO

One Time Pads

- Perfect encryption
- Needs a key as long as the message.
- XOR/add the key and the message:

Message: HELLO ALICE
 Key: SGFKPQYEIJ
 Cipher text: ALRWERKNLO

Plain Text
Cipher Text

Block Ciphers

- Modern ciphers work on blocks of plain text, not just a single symbol.
- They are made up of a series of **permutations** and **substitutions** repeated on each block.
- The key controls the exact nature of the permutations and substitutions.

Advanced Encryption Standard (AES)

- AES is a state-of-the-art block cipher.
- It works on blocks of 128-bits.
- It generates 10 round keys from a single 128-bit key.
- In uses one permutation: ShiftRows and three substitutions SubBytes, MixColumns, AddRoundKey.

Modulo Arithmetic

- Arithmetic modulo "n" means that you count up to "n-1" then loop back to 0
- i.e., 0, 1, 2, ..., n-1, 0, 1, 2, ..., n-1, 0, 1, 2, ...
- $a \text{ mod } b = r$ for largest whole number k such that $a = b \cdot k + r$
- e.g. $9 \text{ mod } 4 = 1$ because $9 = 2 \cdot 4 + 1$

SubBytes

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
 $a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
 $a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
 $a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$

$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
 $b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
 $b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
 $b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

- The "SubByte" is a fixed substitution based on matrix multiplication, one byte at a type.

ShiftRows

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
 $a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
 $a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
 $a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$

$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
 $b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,0}$
 $b_{2,2}$ $b_{2,3}$ $b_{2,0}$ $b_{2,1}$
 $b_{3,3}$ $b_{3,0}$ $b_{3,1}$ $b_{3,2}$

- "ShiftRows" moves the
 - 2nd row one byte to the left,
 - the 3rd row two bytes
 - and the 4th row three bytes.

MixColumn

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
 $a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
 $a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
 $a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$

$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
 $b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
 $b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
 $b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

- "MixColumn" is a substitution of each column such that:

$$(a_0 \cdot x^3 + a_1 \cdot x^2 + a_2 \cdot x + a_3) \times (a_0 \cdot x^3 + a_1 \cdot x^2 + a_2 \cdot x + a_3) \pmod{x^4 + 1} = (b_0 \cdot x^3 + b_1 \cdot x^2 + b_2 \cdot x + b_3)$$

AddRoundKey

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
 $a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
 $a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
 $a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$

$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
 $b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
 $b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
 $b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

- "AddRoundKey" xor's the block with the 128-bit round key (which was generated from the main key).
 – $b_{i,j} = a_{i,j} \text{ xor } k_{i,j}$

AES

- AES encrypts data by first generating the round keys from the main key
- Then 9 rounds of:
 1. SubBytes
 2. ShiftRows
 3. MixColumns
 4. AddRoundKey
- Finally:
 1. SubBytes
 2. ShiftRows
 3. AddRoundKey

DES

The Data Encryption Standard (DES), was the previous standard.

Designed by IBM in early 1970's

Before it was accepted as a standard the NSA stepped in and added S-boxes and fixed the key length at 56 bits

DES

- S-boxes are a type of substitution.
- It was unclear at the time why the NSA added S-boxes to the design.
- Many believed these were a back door for the NSA.

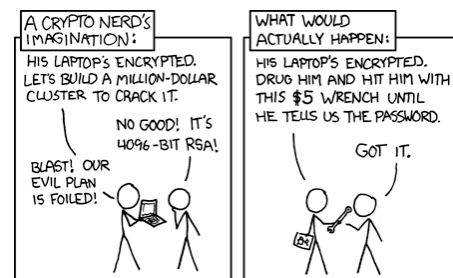
DES

- In 1990, Biham & Shamir discovered differential cryptanalysis.
- The S-boxes had made DES resistant to differential cryptanalysis.
- It seems that the NSA knew about differential cryptanalysis, at the start of the 1970s and had stepped in to protect DES.

Cost to Break DES

- 1977, Diffie and Hellman, theoretically: \$20 million, break in 1 day.
- 1993, theoretically \$1 million, in 7 hours.
- 1997, RSA Security offer \$10,000 for a real break,
 - won by a distributed computing project, at “no cost”
 - EFF (Electronic rights group) break in 56 hours for \$250,000
- 2006, COPACOBANA, general purpose brute force, break DES for \$10,000

A word about key length



3-DES

- Triple DES, was a stop gap until AES
- 3-DES takes 3 keys, K_1 , K_2 & K_3 .

$$E_{K_1 K_2 K_3}(M) = E_{K_3}(D_{K_2}(E_{K_1}(M)))$$
- Setting $K_1=K_2=K_3$ gives you DES
- Expected to be good until 2030
- Used in bank cards and RFID chips

Block Cipher Modes

- Block Ciphers can be used in a number of modes:
- **1) Electronic codebook mode (ECB)**
 - each block is encrypted individually,
 - encrypted blocks are assembled in the same order as the plain text blocks.
 - if blocks are repeated in the plain text, this is revealed by the cipher text.

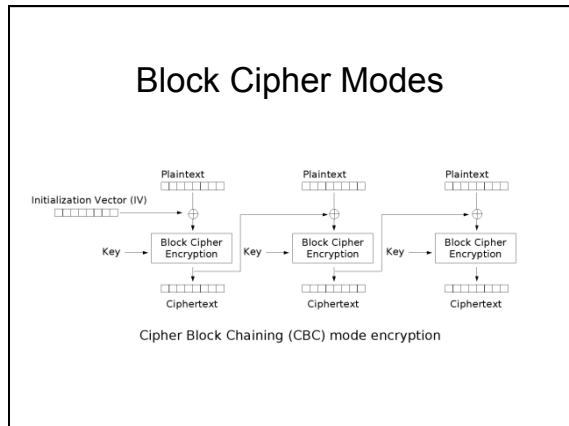
Block Cipher Modes

2) Cipher Block Chaining mode (CBC)


- each block XOR'd with previous block
- helps overcome replay attack.

- Suppose the plain text is B_1, B_2, \dots, B_n .
- $C_1 = \text{encrypt}(B_1 + \text{IV})$,
- $C_2 = \text{encrypt}(B_2 + C_1)$.
- $C_i = \text{encrypt}(B_i + C_{i-1})$.


where IV is an *initialization vector*.



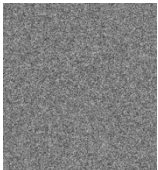
Block Cipher Modes



Original



ECB



CBC

Probabilistic Encryption

- Probabilistic encryption schemes use random elements to make every encryption different.
- E.g. adding a little bit of random plain text data to make the cipher text random.
- This extra data is then discarded when the cipher text is decrypted.
- A random IV is a good way to make encryption probabilistic.

Padding

- To hide the length of messages, short data can be padded to a fixed length.
- The padding data is discarded when the cipher text is decrypted.
- Limited by size of the fixed length.
- In modern schemes, it's done when a message is formatted for encryption, if at all.

TrueCrypt

- TrueCrypt is state-of-the-art encryption software.
- It allows you to create encrypted partition on your harddrive/USB stick
- See demo.