IS53012B/A Computer Security

Dr Ida Pu

Room 10, 29 St James
Goldsmiths, University of London

2019-20 (since 2007)

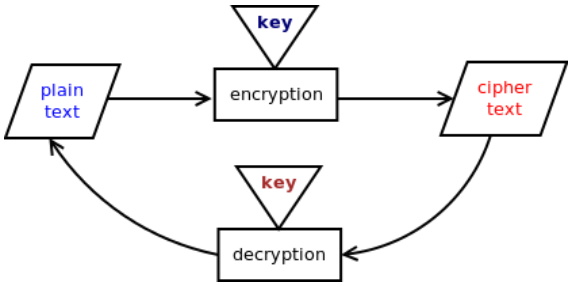
Part I

Cryptosystems Again

Cryptosystems

Include following components:

- key $k \in \mathbf{K}$, where $\mathbf{K}$ is called the *keyspace*
- how keys, e.g. (c, d), are to be chosen and distributed
- encryption and decryption algorithms $\mathbf{f}$ and $\mathbf{g}$, e.g. $c = \mathbf{f}(m, e)$, $m = \mathbf{g}(c, d)$
- alphabets $\mathbf{M}$ and $\mathbf{C}$
- method of blocking (if any).



Note:

- With the keys, one can encrypt and decrypt messages
- The security of a cryptosystem lies in the keys. If the keys are found, the cryptosystem is compromised.

Cryptosystems

Alice $\rightarrow$ **Bob**
Alice wishes to send a secret message m to **Bob**.

Assume $e = d = k$

Alice: $m \rightarrow c$

- makes sure Bob has key k
- encryption: $c = \mathbf{f}(m, k)$
- sends c over an often unsecured communication channel

Bob: $c \rightarrow m$

- receives the ciphertext c and the key k
- decryption: $m = \mathbf{g}(c, k)$
- reads m

Keys are essential

Charles is knowledgeable

- about a cryptosystem, and
- may be able to intercept messages, a m as well as c.

Charles should, however, be unable to retrieve the keys

- A cryptosystem is *symmetric* if either c or d, i.e. the encryption or decryption key, can be determined easily from knowledge of the other. Alice and Bob use *one* key which works for both encryption and decryption.
  Examples: Ciphers we have seen so far are symmetric, i.e. shift cipher, transposition cipher and one-time pads.
- Keyspace needs to be sufficiently large for a symmetric cryptosystem to survive in exhaustive attacks.

How long does it take for an exhaustive search?

This depends on

- the number of keys used
- the time required to examine each key

On average, Charles needs to investigate half number of the keys in the key space before being successful, as he may be successful at the first, second, $\cdots$, or nth attempt. So the number of attempts on average:

$$\frac{1+2+\cdots+n}{n} = \frac{n(n+1)}{2n} = \frac{n+1}{2}$$

Exhaustive search on substitution ciphers

Shift ciphers are substitution ciphers in which each plaintext letter is replaced by a ciphertext letter.

- The size of the (English) alphabet is 26
- In decryption, every ciphertext letter will be one of the 26 possible letters
- Using statistical analysis, letter frequency and letter pattern frequency can be used to find the key faster.

Counting

Consider the English alphabet (A, B, $\cdots$, Z) that is encoded using ASCII.

- In Caesar's cipher there are 26 possible keys. So the size of the key space is 26.
- For the substitution cipher there are 26! (read '26 factorial' $= 26 \times 25 \times 24 \times \cdots 2 \times 1$) possible keys $\approx$ 4 x 1026.
- A statistical analysis can shorten the search.
- A key length of 56 bits used to be considered as secure where the size of the key space was 56.
- Now a search through 256 keys is computationally feasible so keys are of lengths 128, 192 or 256 bits.

Key space

$2^{56} = 72,057,594,037,927,936$
$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$
2^{192} =6,277,101,735,386,680,763,835,789,423,207,666,416,102,
355,444,464,034,512,896
2^{256} =115,792,089,237,316,195,423,570,985,008,687,907,853,269,
984,665,640,564,039,457,584,007,913,129,639,936

How long does it take to find the key?

- Suppose the key is k bits long. Then the size of the key space is 2^k.
- On average, half of the keys $= 2^k/2 = 2^{k-1}$ needs to be investigated until the correct one is found
- Suppose $n \in [1, 10^6]$ keys can be investigated in a microsecond
- It will take Charles $2^{k-1}/n$ microseconds to find the key.

k \ n	1	10^6 (/ms)
32	36 minutes	2 milliseconds
56	1142 years	10 hours
128	5.4×10^{24} years	5.4×10^{18} years

Key management

Consider the following issues:

- Where are the keys generated and by whom?
- How are the keys generated?
- Where are the keys held once they have been generated?
- How are the keys distributed to Alice and Bob (from each other or from the Trusted Third Party TTP)?
- How often are the keys replaced?

Key generation Alice or a TTP:

- Alice generates the keys and sends one to Bob (or vice versa)
- a TTP generates the keys for Alice and Bob.

Key Storage Keys are held by Alice and Bob individually, or by a TTP.

Key Distribution The channel they are using to communicate is insecure so they cannot send the keys over the communication channel.

Key Replacement How often are the keys replaced?

one-time key such as one-time pad, a key is used only once, or the key may be used for a time period of one second or perhaps one day.

session key A key with a limited life is called a *session key*.

Chaining keys

- When Alice generates a new *session key*, and sends it to Bob first encrypting it with the old session key.
  A session key is a single-use symmetric key used for encryption all messages in one communication session.
- If Charles discovers one key then he will be able to determine all subsequent keys.

Random numbers

Random numbers are very important in cryptography. For example keys are often strings of random binary bits.

How are random numbers generated?

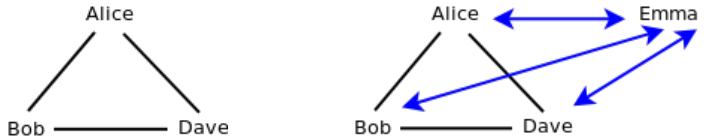
- Ideally by flipping a fair coin, but in reality by a computer programme.
- Such numbers are only pseudo-random.
- A random number generator uses some function f to generate a list of random numbers within a given range.
- Typically the next random number depends on the previous one so that

$$r_{n+1} = f(r_n)$$

The function f must be kept secret. Why?

How many keys are needed?

Example: 3 and 4 people communicate with each other using a symmetric key system:



Theorem

For n people communicating in a symmetric cryptosystem, and each pair of people sharing a key pair, a total of $n(n-1)/2$ pairs of keys is required.

So
10 people $10 \times 9/2 = 45$ key pairs
100 people $100 \times 99/2 = 4,950$ key pairs
1000 people $1000 \times 999/2 = 499,500$ key pairs

Linear Feedback Shift Register (LFSR)

Given a short key $b_1 \cdots b_n$, it is possible to generate a non-repeating stream

$$\underbrace{b_1 \cdots b_n}_{n \text{ bits}}, \underbrace{b_{n+1} \cdots b_{2n}}_{n \text{ bits}}, \cdots, \underbrace{b_{2^n-n+1} \cdots b_{2^n-1}}_{n-1 \text{ bits}}$$

where $b_{i+n} = b_i \oplus b_{i+n-1}$ and $i = 1, 2, \cdots, 2^n - n - 1$
Start with any binary key of length n and generate the next segment of the key stream by $b_i \oplus b_{i+n}$.

Example

Given 1101, we have a non-repeating stream of length $2^4 - 1 = 15$ bits.

$\mathbf{1101} \to 1\mathbf{1010} \to 11\mathbf{0101} \to 110\mathbf{1011} \to 1101\mathbf{011} \to 11010\mathbf{110} \to 110101100 \cdots$

Example

Alice (1101, m), m=1110001 $\rightarrow$1101 $\rightarrow$ Bob (1101, c)

1101 ↓ 11010 ↓ 110101 ↓ k=1101011

$c = m \oplus k = 0011010 \rightarrow$

1101 ↓ 11010 ↓ 110101 ↓ k=1101011

$m = c \oplus k = 1110001$

Risks

Let the ith bit of a ciphertext, plaintext and the key be c_i, m_i and k_i respectively. We have

$$c_i = m_i \oplus k_i$$

$$m_i = c_i \oplus k_i$$

and

$$c_i \oplus m_i = c_i \oplus (c_i \oplus k_i) = k_i$$

Whop! If he knows a section of m_i and c_i, Charlie can recover the key-stream for the section.
If $i = 1, \cdots, 4$ in the previous example, then
$m_{1-4} \oplus c_{1-4} = 1110 \oplus 0011 = 1101$.

Key Generation

- Security for a stream cipher relies on the design of the key stream generator
- A keystream must be unpredictable.
- Designing a good keystream generator is difficult.
- However, there are many applications for stream ciphers because of their speed of use, ease of implementation and the fact that one bit of corrupt ciphertext does not impact on the rest of the message.
- LFSR should *not be used alone* as a key-stream generator.
  A *valid* combining function F should be used instead:

$$\left.\begin{array}{l} LFSR_1 \rightarrow \\ LFSR_2 \rightarrow \\ \quad\vdots \\ LFSR_n \rightarrow \end{array}\right\} \rightarrow F \rightarrow \text{key stream}$$

Part II

Cryptanalysis

Cryptanalysis

Studies of methods for obtaining the meaning of encrypted information, referred to as "code breaking", or "cracking the code"

- Attempt to circumvent the security of cryptographic algorithms and protocols
- Focus on methods of attacks that primarily target weaknesses in the cryptography
- Exclude
  - bribery
  - physical coercion
  - burglary
  - keystroke logging
  - social engineering
- Changes drastically through the history of cryptography. Today it is no longer possible to have unlimited success.

Activities

- break a single ciphertext
- recognise patterns and use known algorithms
- infer the meaning of ciphertexts, noticing unusual frequencies of symbols
- deduce the key(s)
- find a weakness in implementation or environment of cryptosystems
- find general weaknesses in encryption algorithms

Breakable security

- Security relies upon some one-way functions and it would take too long to examine all possibilities
- BUT Charlie does not have to try all the possibilities!
- Charlie is knowledgeable, diligent and works with all sorts of information and tools available

Example

1. Hidden meanings of "AF" change the course of World War II
2. Mafia boss Bernardo Provenzano's notes in ciphertext "... I met 512151522 191212154", where A $\rightarrow$4, B $\rightarrow$5, $\cdots$, etc.

Common attacks

Cipher-text-only Charlie has a string of cipher text c

Known plaintext Charlie has a set of ciphertext and corresponding plain text

Chosen-plaintext (Chosen-ciphertext) Charlie can choose the plain/cipher text and construct its cipher/plain text, may be statically or adaptively

Related-key attack Like a chosen-plaintext attack, except Charlie can obtain ciphertexts under two different keys. e.g. Keys are unknown but the relationship between the keys are known.

Success of attacks

Total break Charlie deduces the secret key

Global or Instance deduction Charlie discovers additional plain texts

Information deduction Charlie gains some information about plaintext unknown previously

Distinguishing algorithm Charlie can distinguish the cipher text from a random permutation

Classical techniques

1. Frequency analysis
2. Define good mathematical problems and models
3. Probabilities and statistics

Cryptanalysis of the Caesar Cipher

Example

wklv phvvdjh lv qrw wrr kdug wr euhdn

1. frequency table: blank:7 w:4 v:4 h:3 d:3 r:3 k:2 l:2 u:2 p:1 j:1 g:1 n:1
2. try blank →blank
3. try short words
   lv, grw, wrr and wr
4. try patterns
   "xyy", e.g. see, too, (woo, gee), but ~~see~~, as "se" is not an English word.
   So wrr →TOO, and wr →TO
5. qrw →qOT
   "xOT", e.g. hot, got, cot, dot, not, so q→N
   ⋮
   So shift key = -3.

Cryptanalysis of a cryptosystem

Example

John proposes a cryptosystem that is based on one-time key pad and requires no key exchange. It works as follows: If she wants to send Bob a message m, Alice generates her key k_a, a sequence of random bits (the same length as m), computes $c = m \oplus k_a$ and sends c to Bob, where $\oplus$ represents the bitwise XOR operation. On receipt of c, Bob generates his own random bits k_b of same length, computes $d = c \oplus k_b$ and sends d to Alice. On receipt of d, Alice computes $e = d \oplus k_a$ and sends e to Bob. On receipt of e, Bob computes $e \oplus k_b$ for the last time.

How does it work?

Alice		Bob
m		
$\downarrow$		
$m \oplus k_a$	$\rightarrow\rightarrow$	$m \oplus k_a$
		$\downarrow$
$m \oplus k_a \oplus k_b$	$\leftarrow\leftarrow$	$m \oplus k_a \oplus k_b$
$\downarrow$		
$m \oplus k_a \oplus k_b \oplus k_a = m \oplus k_b$	$\rightarrow\rightarrow$	$m \oplus k_b$
		$\downarrow$
		$m \oplus k_b \oplus k_b = m$

Charlie may overhear

Alice		Charlie		Bob
m				
$\downarrow$				
$m \oplus k_a$	$\rightarrow$	$m \oplus k_a$	$\rightarrow$	$m \oplus k_a$
				$\downarrow$
$m \oplus k_a \oplus k_b$	$\leftarrow$	$m \oplus k_a \oplus k_b$	$\leftarrow$	$m \oplus k_a \oplus k_b$
$\downarrow$		$\downarrow$		
$m \oplus k_a \oplus k_b \oplus k_a$		$(m \oplus k_a) \oplus (m \oplus k_a \oplus k_b) = k_b$		
$\downarrow$				
$m \oplus k_a \oplus k_b \oplus k_a = m \oplus k_b$	$\rightarrow$	$m \oplus k_b$	$\rightarrow$	$m \oplus k_b$
		$\downarrow$		$\downarrow$
		$k_b \oplus m \oplus k_b = m$		$m \oplus k_b \oplus k_b = m$

Shannon's theory

An influential paper "Communication theory of secrecy systems", the Bell Systems Technical Journal, 1949

- Computational security: 'computationally secure' means that the best algorithm for breaking it requires N operations, where N is some specified, very large number.
- Unconditional security: means that it cannot be broken, even with infinite computational resources.