Campus 无线装订本
A5 40页
WCN-CNB3430
6 937748 309949
MADE IN CHINA

国誉商业(上海)有限公司
http://www.kokuyo.cn/st/
上海市奉贤区人杰路128号
TEL：400-820-0798　FAX：021-3255-8508
产地：上海市　QB/T1438·2007合格
A5 210×148mm

● 采用日本进口纸张，
纸质细滑，牢固不掉页。

Campus | A5 7mm×24行 40页

Cryptograph

Campus®

A5 点线本 | 7mm×24行 | 40页

KOKUYO

Classic Cryptographic Systems

Compression & Encryption

{ Data is not Information
  Information is not Wisdom

Compression and Encryption are about manipulating information

[Difference]

Compression: Extract information from the data and encode as efficiently as possible with a public algorithm

Encryption: Diffuse a key into the information as much as possible & encode with a public algorithm
(most systems diffuse the key into the data. DES does this and the DES cracks relay on this to break into the code)

DES: Data encryption standard is a symmetric-key algorithm for the encryption of electronic data.

The best approach

The best way to practically transmit secure data safely is to:
① Compress the data ⇒ ② Encrypt ⇒ ③ Add error detection & recovery.
Cryptanalysis relies (ultimately) on exploiting redundancies the plaintext (data)
- Compression removes these.

Encryption is usually slow
- Compressing first (which is fast) means that you can pipeline the information flow.
- Data is constructed by the compressor, so the volume of data into the encryption system is less.

Key Terminology

Plaintext: Message in "clear" form
隐写术 Steganography: Conceals (隐藏) the existance of the message. The message itself may or may not be encoded.
Cryptography: Message in plain view, but the meaning is concealed.
Cipher: Operates on groups of characters. Ciphertext = f(plaintext, key)
(密码, 暗号)

Caesar Cipher

Map a plain alphabet to a single shifted cipher alphabet: { P abcdefghijklmnopqrstuvwxyz / C defghijklmnopqrstuvwxyzab

Thus to encipher a plaintext of "veni vidi vici", simply look up the character in the plaintext and pick out the corresponding ciphertext character: Veni ⇒ YHQL Vidi ⇒ YLGL Vici ⇒ YLFL (会来会见会征服)

A variation of this approach is to start the cipher alphabet with a keyword. e.g. ROME and then complete the reminder of the cipher alphabet as before

The trouble with the monoalphabetic ciphers is that they can be easily cracked using frequency analysis.

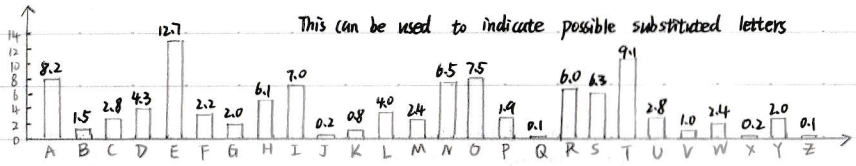
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
X X X X Y X X X X Z X X X X T X X X X W

$P(X) = -\log_2 \frac{16}{20}$

$P(Y) = -\log_2 \frac{1}{20}$

Frequency Analysis

In English, the frequency of each letter within a piece of standard text is typically:



e t a o i n s h r d l c

u m w f g y p b v k j x q z

Basic Cryptanalysis

Cryptanalysis usually involves a variety of techniques

① for example, frequency analysis can indicate possible letters, but this can be enhanced by cribs

②A "crib" is a known sequence of letters or words - e.g. in English the letter U almost always follows Q.

③We can be also make guesses about common words based on the context of letters

Unfortunately a strength of a short message is difficult of applying frequency analysis. Moreover, cipher maker usually employ some additional tricks on the text 以 for example: break up the text into words that do not match the original word lengths - or even a stream of characters

Take the following ciphertext:

JUWWTDRU MDH ZPIU FDWIUT GZKF FHQFGKGHGKDR SKAZUC PRT GZUJDCT GZPG MDH RUUT VDC GZU PRFJUC KF PWAZPQUG

(the most common letter)

If we carry out a basic frequency analysis the most common letter in the message is "U" - could this be "E"

The second most frequently letter is "G", could this be "T"?

Note that 3 letter sequence tZe - could Z be h? - it happens more than once.

Some words can be easily inferred: thPt is clearly that, so P is a

We can return to our frequency analysis and note that the letters e t h a have been allocated. The next most commonly used letter is D. D→o?

We can also notice the sequence JeWW. Pairs of letters are unusual, and we have a clue letter that is follows "e". Possible letters could be eLL, eMM, eNN, eOO, eTT, eSS.

We can make a guess that LL looks more likely: W→L

We now can observe another interesting word: alAhaQet A→p? Q→b?

The next most common letters are N, R and I

The Take the triplet aRIT (3个字母的单词) ar? an? T→d?

There are three letters left, C, F & K, that occurs frequently, and the most common letter not used thus far are S, R, I

Take the text thKF, K could be a vowel (元音), therefore K→i?

Now we have the word thiF - F could be S or R - but S makes a proper word, thus F→s C→r

By this stage, you could probably 破解 the rest intuitively, but we can still replace another letter straight away.

Another simple word to identify (big words are useful as they have fewer common denominators) sHbstitHtion H→u

well done you have solved this substitution cipher and the word that you need for the answer is alphabet.

Some cryptanalysis Tips

1. The vowels A E I O are normally high frequency, U is moderate and Y is low frequency
   Try to analyze a cipher in terms of vowels and consonants 辅音 alone before committing values to letters
2. Letters contacting low frequency letters are usually vowels.
3. Letters showing a variety of contact are usually vowels
4. In repeated diagrams (pairs of letters), one letter is usually a vowel.
5. In reversed diagrams - eg VX, XV, one letter is normally a vowel.
6. Doubled consonants are usually bordered by vowels and vice versa.
7. It is unusual to find more than five consonants in a row.
8. Vowels do not often contact one another.

Polyalphabetic Ciphers

P	abcd	用两种对应的字母表相间加密
C1	fzbv ...	
C2	go xb ...	

System: Alternate between the cipher alphabets C1 and C2

Plaintext	C1	C2	Ciphertext	
H	a		a	
E		f	f	The same letter L in the plaintext does not appear
L	p		p	as the same letter in the ciphertext
L		a	a	
O	d		d	

Vigenère Cipher provide a easy way (does not need to communicate with the receiver)

The idea is to have 26 cipher alphabets all shifted by a different amount from the plaintext alphabet, this is managed using a Vigenère square:

行

列	A B C D E F
A	A B C D E F...
B	B C D E F G...
C	C D E F G H...
D	D E F G H I...
E	E F G H I J...

① Consider the set of all 26 Caesar ciphers {Ca, Cb, Cc, ..., Cz}

② Choose a key, e.g. deceptive

③ Encrypt the plaintext letter by letter using Cd, Ce, Cc, Ce, Cp, Ct, Ci, Cv, Ce

④ Repeat from the start after the last Ce

⑤ Decryption simply works in reverse.

[Example]

key:	d e c e p t i v e	通过key找对应的列
plaintext:	w e a r e d i s c	通过原文找对应的行
ciphertext:	Z I C V T W Q N G	选出对应行的字母

Vigenere Cipher : Cryptanalysis

Basic approach
① Find the length of the key
② Divide the message into that many simple substitution encryptions.
③ Use frequency analysis to solve the resulting simple substitution.

How to find the key
① Kasisky test
② Index of coincidence (Friedman)

Kasisky Test

[Concept] plaintext words separated by multiples of the key length are encoded in the same way

Key : CAT CATCAT CAT ··· CAT ··· TCA
PT : THE ··· THE THE
CT : VHX ··· VHX WJE

It's worth nothing if the distance between the repeated patten is not a multiple of the key, this test will reveal nothing.
Use the gcd to apply frequency analysis then.

[Algorithm] ① Search for pairs of identical segments of length at least 3
② Record distances between the two segment $\Delta 1, \Delta 2, \cdots$
③ The key length (m) should be divided drive the gcd $(\Delta 1, \Delta 2 \cdots)$ (Greatest Common Divisor)

Index of Coincidence (Friedman)

IC is defined to be the probability that two randomly selected letters from a text will be identical.
E.g.
For a truely random text, the chance of pulling out an A is 1/26. The probability of pulling out two As simultaneously is 1/26 * 1/26. In general, the chance of drawing any pair of letters is 26 * 1/26 * 1/26 = 1/26 = 0.0385

The formula for the index of coincidence for any text is:

$$IC = \frac{\sum_{i=a}^{z} f_i(f_i-1)}{N(N-1)} \approx \frac{\sum_{i=a}^{z} f_i^2}{N^2} = \sum_{i=a}^{z} P_i^2$$

N is the total number of letters in the text
f_i is the frequency of letter i in the text (i.e. the number of occurences)
P_i is the probability of letter i in the text

Campus

Tabulate

Frequency Analysis

If a ciphertext is a mono-alphabetic substitution then it will have similar frequency (albeit not the same letters) as the standard alphabet.
There is a measure called 'Index of Coincidence' which can be calculated to see whether our cryptogram frequencies are similar to normal English

{ Normal written English has an IC of around 0.065
{ Entirely random text has an IC near 1/26 = 0.038

$\sum_{i=a}^{z} P_i^2$

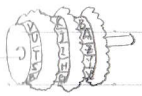
IC is unique for each language, for English text, IC is around 0.065
You can use this metric to find the key length of a Vigenere cipher, as follows
Assume you have a text $Y = y_1 y_2 y_3 \cdots y_n$

y_1 y_{m+1} ··· y_{n-m+1}
y_2 y_{m+2} ··· y_{n-m+2}
··· ··· ···
y_m y_{2m} ··· y_n

① Make a guess of the key length (e.g. m)
② Arrange the text into a matrix with m rows
③ Calculate IC for each row ($Y_1 - Y_m$)
④ If the width of the matrix is a multiple of the actual key length then: $IC(Y_i) \approx 0.065 \quad \forall 1 \le i \le m$
⑤ Repeat the process again until you find the correct key

Rotor Cipher Machines

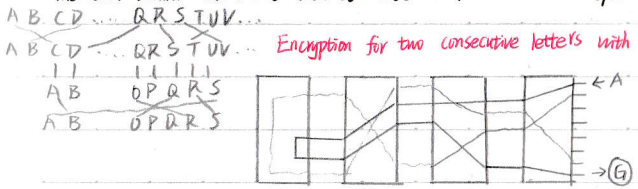
{ With 3 cylinders, $K = 26^3 = 17,576$
{ With 5 cylinders, $K = 26^5 = 12 \times 10^6$



[Operation Principles]
There are rotating disks with an array of electrical contacts on either side. The rotors advance positions after encrypting each letter, which change the substitution. this produces a complex polyalphabetic substitution cipher which changes for every letter

The Enigma Machine

The Enigma Machine was based on rotors whose movement produced ever-changing alphabetic substitution.
The Enigma machine worked by using a series of substitution ciphers where the connections were made electrically
The idea behind the rotors was to shift each substitution alphabet in addition to making the connections



Encryption for two consecutive letters with the right rotor moving one position between them

按键对应实际的字母
Plugboard settings $\frac{n!}{(n-2m)! \times m! \times 2^m}$ $(n=26)$
1→11 ↑ 11→? ↓
When m=11, max. value

The complexity of the Enigma Machine

Possible choice of rotors * Possible starting positions * number of plugboard settings
[Eg] Assume 6 rotors, 10 plugboard connections

⇒ Rotors = 6×3×4 = 120
Initial position: $26^3 = 17576$
Plugboard settings: $\frac{26!}{6! \times 10! \times 2^{10}} = 150,738,274,937,250$

⇒ Total number : $3.179 * 10^{21}$

KOKUYO

Transposition Ciphers

Transposition (换项) or permutation (置换) ciphers from the second basic building block of ciphers.
The core idea is to rearrange the order of basic units (letters / bytes / bits) without altering their actual values.
One of the earliest transposition encryption device was the Scytale (密码棒)
The scytale consists of a ribbon (纸条) wrapped around a dowel (木棒) of a particular diameter and length

[Encryption] The secret message was written on the ribbon while the ribbon was wrapped on the rod
The ribbon was then removed and transported to the other field commander who had an identical rod.
If the ribbon was intercepted (截取) it looks like jumble of letters. 杂乱

[Decryption] The recipient was a rod of the same diameter on which he wraps the parchment to read the message.

Rail Fence cipher 栅栏密码

The message letters out diagonally over a number of rows (rails) then read off cipher row by row.
The key for the rail fence cipher is just the number of rails.

[EXAMPLE]

defend the east wall of the castle ⇒ Cipher: dnetlhseedheswloteateftaafci (read off along the rows)

```
3 →   d . . . n . . . e . . . t . . . l . . . h . . . s . . .
rails . e . e . d . h . e . s . w . i . o . t . e . a . t . e
      . . f . . . t . . . a . . . a . . . f . . . c . . . i .
```

Colummar Transposition Cipher 柱状换位密码

Encryption: the message is written out in rows of a fixed length and then read out again coloum by coloum. Both the width of the rows and the permutation of the columns are usually defined by a keyword.

[Example]

Encrpt the message M below using the keyword Destiny M = {Attack postponed until two}

① The message is written out in rows of a fixed length 7 indicated by the number of letters in the key word "Destiny". The empty spaces at the end are usually filled with random letters

Alphabetical Order	1	2	5	6	3	4	7
K	D	E	S	T	I	N	Y
PT	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	f	g	x	n	u

② The columns are read according to the alphabetical order of the letters in the key word.

③ Therefore the cipher text will be

AODW TSUO COIX KNLN TTNF APTG PETU

Coloum Transposition Cipher: Cryptanalysis

Trsao rih oivo geeth tootn bfsafme cepogep ihte tcen

The number of letter in this cipher is 45 (without spaces). The key length should be a divisor of 45 {3, 5, 9, 15}

[Assume the key length is 5]

```
t i o m p      toimp
r v o e i      rovei
s o t c h
a g h e t
o e b p e
r e f c t
i t s o c
h h a y e
o t f e n  ⇒ often
```

⇒ To improve is to change to be perfect is to change often

Fitness Metrics

A fitness Metrics is a measure to determine how similar a piece of text is to English text.
[E.g]
① Single letters frequency
② Digrams statistic: the probability of a sequence of two letters
③ Trigrams statistic: the ... three
④ Quadgrams statistic: four

A piece of text very similar to English will get a high fitness score while a text of random chars will get low

Coloum Transposition cipher — Cryptanalysis Methods

[Brute Force (暴力攻击) attacks assuming Short key words]
– Try all possible short words (up to 9 letters) and look for anagrams (字谜), patterns, possible combinations for words

Dictionary attacks
① Compile a list of dictionary words including place names, famous people, mythological names, historical names and so on (1,000,000 dictionary words would be a good comprehensive target.
② Generate a text file of possible keys
③ Try to decrypt the ciphertext with all possible dictionary words and record the keyword that resulted in plaintext with the highest fitness

No.

Date

[Hill Climbing Method]

1. Assume the key length is N=10, then choose a random starting keyword of this length. This is called the parent key.
2. Apply the parent key and measure the fitness of the resulting text.
3. Generate a child key by making random swaps in the parent keyword.
4. Apply the child key measure the fitness of the resulting text, if the latter is higher than fitness obtained in step 2, than replace the parent with the child that beat it.
5. Rank different decryption keys until you find the one that produces deciphered text with the fewest rare sequences.
6. If the decryption key is not found after several hundred tunes, choose another random word of length N and go back to step 2.
7. If the key is still not found increment N by 1 and repeat the whole process.

Product Ciphers

Ciphers using only substitutions or transpositions are not secure because of language characteristics

The fore number of ciphers in succession to make harder

① Two substitutions make a more complex substitution
② Two transpositions make more complex transposition.
③ Substitution followed by a transposition makes a new much harder cipher.

Product Ciphers is bridge from classical to modern ciphers

Steganography

Difference between Steganography & Cryptography

Cryptography: although encrypted and unreadable, the existence of data is not hidden.

Steganography: no knowledge of the existence of the data

steganography Principles

A Steganosystem consists of

① A cover-object is an original unaltered medium (text, image or video)
② Embedding process in which the sender hides a message by embedding it into a cover-text, usually using a key, to obtain a stego-object
③ Stego-object
④ Recovering process in which the receiver extracts the hidden message from the stegotext using the key

Security requirement: stego-objects should be indistinguishable from cover-objects

No.

Date

Symmetric Cryptographic Systems

△ Discuss the basic principles of discrete probability and information theory

Quantifying Information

Shannon (1948): "a signal that is totally predictable carries no information"

In essence, the difference in the signal between what is predicted and actual received values is a measure of its information content or entropy.

Following Shannon the unit is the bit

A system contains N-bits of information if it contains 2^N possible characters.

Information and Entropy

$$H(x) = \sum_i P_i \log_2 \frac{1}{P_i}$$

Consider an information source with a set of symbols a_i, where each symbol occurs in the data with P_i

Information theory uses entropy as a measure of how much information is encoded in a message

How much can we compress

The main theorem proved by Shannon says essentially that the minimum number of bits needed to binary code a message is given as follows:

$$B = n \cdot H(x)$$

{ n: the number of symbols in the message
{ H(x): the entropy of the message

[EXAMPLE]

Let's consider the following message: A B A E S E D A

$P_A = 0.375$ $P_B = 0.125$ $P_E = 0.25$ $P_D = 0.125$ $P_S = 0.125$

⇒ $H(x) = 2.15$ Bits/symbol ⇒ $B = 2.15 \times 8 = 17.2$ bits (8: number of symbols)

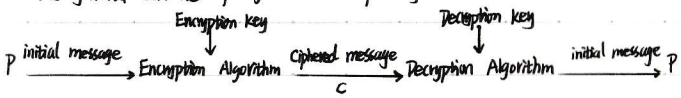
Assuming we are using standard 8-bit ASCII characters to encode this message, we will need $8 \times 8 = 64$ bits

The difference between the 17.2 bits and 64 bits used to encode the message is redundant data

△ Describe the principle of Shannon perfect secrecy.

Cryptography: Main Concepts

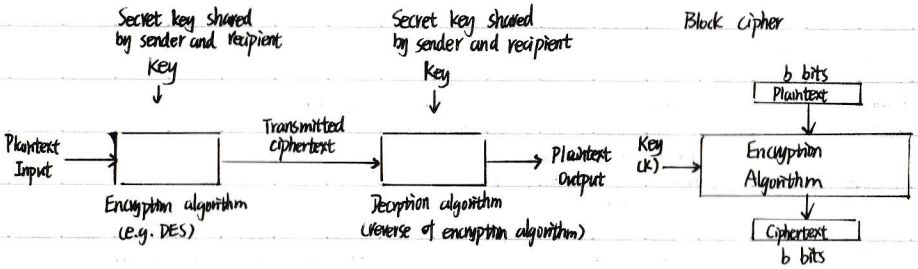
The general scheme of symmetric enciphering



The encryption algorithm uses encryption key to transform a plaintext into a ciphertext

The decryption algorithm uses decryption key to transform a cipher into a plaintext

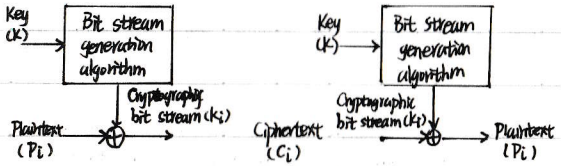
Shared key system: Both sender and receiver use the same key which must remain private. These systems are called symmetric.
Example: DES, Triple-DES, Twofish, Rijindael.



Stream Ciphers vs. Block Ciphers

Block ciphers: in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 128 bits or more is used. For example: AES

Stream cipher: in which the plaintext is encrypted/decrypted one bit or one byte at a time. For examples includes, XOR cipher, RC4 A5/1, A5/2



Stream Cipher using Algorithmic Bit Stream Generator

Problems with Shared Key Systems
① Compromised key means interceptors (攻撃) can decrypt any ciphertext they have acquired. Keys can be changed frequently to limit damage.
② Distribution of keys is problematic: keys must be transmitted securely e.g. distribute key in pieces over separate channels.

One important principle in cryptography is Kirchhoff's Principle, which was documented in his book in (1883)
"The cipher must not depend on the secrecy of the mechanism, it must not matter if it falls into the hands of the enemy".

What parts of a crypto system must be kept secret?
① Encryption Algorithm
② Decryption Algorithm
② Keys ← ✗
④ Ciphertext

Symmetric Cipher Definition: A cipher is defined as over (K, M, C) is a pair of algorithm (E, D) where
$E: K \times M \to C$
$D: K \times C \to M$
Such that: E and D satisfy the following consistency equation:
$\forall m \in M, k \in K: D(k, E(k, m)) = m$

What is a secure cipher?
- Shannon (1949) basic idea: cipher text reveals no info about plain text
Shannon Perfect Secrecy:
- A cipher (E, D) over (K, M, C) has perfect secrecy if:
$\forall m_1, m_2 \in m$. (length(m_1) = length(m_2)) and $\forall c \in C$
$Pr[E(k, m_1) = c] = Pr[E(k, m_2) = c]$
Where k is uniform in K ($k \xleftarrow{R} K$)

This means given CT, I want be able to deduce whether the message is m1 or m2 or any other m, therefore the most powerful attacker can learn nothing about the plaintext from the cipher text. This means there is no cipher text only attack (but other attacks may be possible)

△ Verify the security of traditional ciphers

Shift Cipher Review

① Mathematically, map the message letters to numbers { a, b, c, d, ..., x, y, z / 0, 1, 2, 3, ..., 23, 24, 25
② Encrypt by shifting each letter in the message by K positions (k is the key)
$c = E_k(p) = (p + k) \bmod 26$ E.g. m="HELLO"; k=1; c="IFMMP"

Does shift Cipher satisfies Shannon perfect secrecy for messages with a length ≥2?
Take m_1 = "AC", m_2 = "AZ", and c = "BD"
Now assuming all keys have the same probability: $Pr[E(k, m_1) = c] = \frac{1}{26}$
However, for all $k \in K$ we have $E(k, m_2) \neq c$, and hence
$Pr[E(k, m_2) = c] = 0$
And so the perfect secrecy requirement is violated, and the shift cipher is prone to cipher text only attack

Perfect security: an encryption algorithm is perfectly secure if a ciphertext produced using it provides no information about the plaintext without knowledge of the key. If $\mathcal{E}$ is a perfectly secure encryption function, for any fixed message m there must exists for each ciphertext c at least one key k such that $c = E_k(m)$.

m=0101100
C=01000000
k=0011100

One Time Pad Review

OTP is defined over the following sets $\begin{cases} M = C = \{0,1\}^n \\ k = \{0,1\}^n \end{cases}$

Encryption: $c = E(k,m) = k \text{ xor } m$

Decryption: $m = D(k,c) = k \text{ xor } c$

Is one time pad secure?

For OTP is defined over the following set:

$M = C = \{0,1\}^n$, $K = \{0,1\}^n \Rightarrow$

$\forall m, c$ if $E(k,m) = c$ this means $c = k \text{ xor } m$ therefore $k = c \text{ xor } m$

The size of $\{k \in K : E(k,m) = c\} = 1$

Therefore $\forall m, c$

$$Pr[E(k,m) = c] = \frac{\{k \in K : E(k,m) = c\}}{|K|} = \frac{1}{|K|}$$

Therefore OTP has perfect secrecy.

The bad news: Shanon has proved later that in order for a cipher to achieve the perfect secrecy than $|K| \geq |M|$

This means the key must be at least as long as the message.

[Quiz] Is one time pad secure?
→ Let $m \in M$ and $c \in C$. How many OTP keys map m to c?
a) None
b) 1
c) 2 ← Should be, I guess
d) Depends on m

Shannon's notion of perfect secrecy is too strong to be useful
Shannon's notion of perfect secrecy is only related to ciphertext attacks

[Solution]

Relax this notion from information-theoretic secrecy to computational secrecy. While information-theoretic secrecy required that every given a cipher text, every plaintexts are exactly as likely, the computational secrecy notion will ask only that no efficient algorithm can tell, given a ciphertext, and, say, and two messages that could potentially be plaintexts corresponding to this ciphertext, whether one of these messages is more likely than the order to be the actual plaintext. Other possible attacks should also be considered.

Shannon Perfect Secrecy:

A cipher (E,D) over (K,M,C) has perfect secrecy if:

$\forall m_1, m_2 \in M$, (length(m_1) = length(m_2)) and $\forall c \in C$

$Pr[E(k,m_1) = c] = Pr[E(k,m_2) = c]$ where $k \xleftarrow{R} K$

Computational Perfect Secrecy:

A cipher (E,D) over (K,M,C) has perfect secrecy if:

$\forall m_1, m_2 \in M$, (length(m_1) = length(m_2)) and $\forall c \in C$

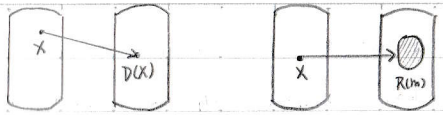
$Pr[E(k,m_1) = c] - Pr[E(k,m_2) = c]$ is "negligible" where $k \xleftarrow{R} K$

but also need adversary to exhibit $m_1, m_2 \in M$ explicitly

Both notions assume that the attacker can only observe the ciphertexts

541031
1479173

△Explain how to construct a secure block cipher



Deterministic vs Random Functions

A function D is said to be deterministic if it always produces the same output of the input does not change
On the other hand a randomized function R may produce different outputs for the same inputs

PRPs and PRFs

Essentially, a truely random function would just be composed of a lookup table filled with uniformly distributed random entries.

伪随机函数 伪随机置换

1. Pseudo Random Function (PRF) defined over (K,X,Y)
   $F: K \times X \to Y$ (Y不能是X内, 不然是 $K \times Y \to X$)
   such that exists "efficient" algorithm to evaluate $F(k,x)$

伪随机排列

2. Pseudu Random Permutation (PRP) defined over (K, X):
   $E: K \times X \to X$
   such that: ① Exists "efficient" deterministic algorithm to evaluate $E(k,x)$
   ② The function $E(k,\cdot)$ is one-to-one
   ③ Exists "efficient" inversion algorithm $D(k,y)$

PRF and PRP

A pseudo random permutation family is a collection of pseudorandom permutation, where a specific permutation may be chosen using a key

PRP: The reason it is a X and not a Y means that the output from PRP must be one-to-one. That is, the PRP must be a deterministic algorithm

PRPs Example

PRPs is sometimes called a Block cipher

Examples PRPs: 3DES, AES, ...

AES: $K \times X \to X$ where $K = X = \{0,1\}^{128}$

3DES: $K \times X \to X$ where $X = \{0,1\}^{64}$, $K = \{0,1\}^{168}$

Functionally, any PRP is also a PRF
→ A PRP is a PRF where X=Y and is efficiently invertible

Semantic Security of PRP (one-time key)

Definition: E is semantically secure if for all "efficient" A $Adv[A,E]$ is negligible. Where:

$Adv[A,E] = |EXP_A[E(k,m_1)] - EXP_A[E(k,m_2)]|$

for all explicit $m_0, m_1 \in M$ (length(m_1) = length(m_2))

EXP the experiment that an adversary performs on encrypted messages

In other words: the attacker can not distinguish between encrypted messages.

Secure PRPs

Quiz: Let $P: K \times X \to \{0,1\}^8$ be a secure PRP

Is the following Y a semantically secure PRP?

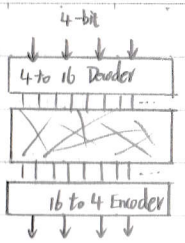
$$Y(k,x) = \begin{cases} 00100010 1 & \text{if } x = 0 \\ P(k,x) & \text{otherwise} \end{cases}$$

⇒ a) No, it is easy to distinguish Y from a random permutation
b) Yes, an attack on Y would also break P
c) It depends on P

No.

Date

How to construct an ideal Block cipher



One solution is to use a truely random permutation.
Each block may be viewed as a gigantic character. The "alphabet" consists of 2^N gigantic characters

Each particular cipher is a one-to-one mapping from the plaintext "alphabet" to the ciphertext "alphabet".
There are $2^N!$ such mappings
A secret key indicates which mapping to use

An ideal block cipher would allow us to use any of these 2^N mappings. The key space would be extremely large.
This would require a key of $\log_2(2^N!)$ bits. If $N=64$, $\log_2(2^N!) \approx N \times 2^N \approx 10^{21}$ bits of "10" GB infeasible! 不可行

Outline the basic operation principles of 3DES

Shannon's Confusion and Diffusion 混淆与扩散

混淆用于掩盖明文与密文之间的关系
扩散通过将明文冗余度分散到密文中使之分散开来，即将单个明文或密钥位的影响尽可能扩大到更多的密文中去（扩散最简单方法：置换(换位)）

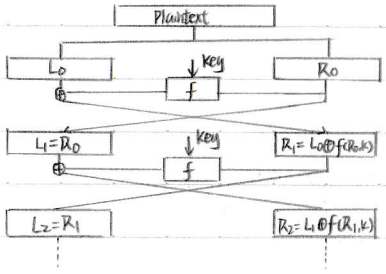
A cipher needs to completely obscure statistical properties of original message.
Claude Shannon suggested using both Substitution and Permutation blocks to thwart(挫败) cryptanalysis based on statistical analysis. He introduced two concepts:

1. diffusion: - dissipates the redundancy of the plaintext by spreading it out over the cipher text. One of the easiest technique to achieve this is permutation 排列 置换
2. confusion: Obscures the relationship between the ciphertext and the plaintext in order to hide any statistical patterns. One of the easiest technique to achieve this is substitution.

DES Core Idea (Feistel Structure)

Horst Feistel proposed an approach in order to avoid the complexity problem of idea block cipher. His approach is based on Shannon's diffusion and confusion
Feistel's approach is based on the concept of "invertible 可逆 product cipher" whereby the execution of two or more simple ciphers in sequence leads to a much more cryptographically secure solution than any of the component ciphers used. He employed the two primitive cryptographic operation: substitution and permutation.

[Feistel Structure]



$$\begin{cases} R_i = F_i(R_{i-1}) + L_{i-1} \\ L_i = R_{i-1} \end{cases}$$

Without reversing the function but reversing the order of the key

Claim: for all $f_1, \dots, f_d: \{0,1\}^n \to \{0,1\}^n$
Feistel network $F: \{0,1\}^{2n} \to \{0,1\}^{2n}$ is invertible
In other word prove that give R_i, L_i
you can find R_{i-1}, L_{i-1}

$$\Rightarrow \begin{cases} R_i = F_i(R_{i-1}, K) + L_{i-1} \\ L_i = R_{i-1} \end{cases}$$

This means: $R_{i-1} = L_i$

$L_{i-1} = ?$ $L_{i-1} = R_i + F_i(R_{i-1}, K)$

No.

Date

[Operation Principles] The inputs to the algorithm are the plaintext (of length 2w bits) and a key K.
The plaintext is split into two halves L and R, and the data is then passed through n "rounds" of processing and then recombined to produce the ciphertext.

Each round has the same structure, the left halve of the data has a substitution performed. This requires a "round function" F to be performed on the right half of the data and then XORd with the left half. Finally a permutation is performed that requires the interchange of the two halves of the data.

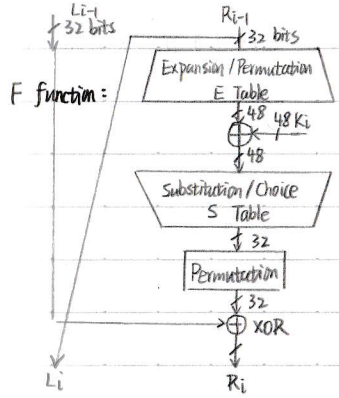
Data Encryption Standard Algorithm
③ Number of rons : 16
① Block Size : 64
② Length of key : 56

The implementation of a Feistel Cipher

The implementation of a Feistel Cipher has the following key parameters:

① Block size: A large block size generally means greater security, but reduced speed. 64 bit block sizes are very heavily used as being a reasonable trade-off although AES now uses 128 bits
② Key size: The same trade-off applies as for block size. Generally 64 bits is now considered adequate and 128 bits is preferred.
③ Number of rounds: Each round adds additional security. A ~~sign~~ single round is inadequate, but 16 is considered standard.
④ Subkey generation: the more complex this algorithm is, the more secure the overall system will be.
⑤ Round function: Greater complexity again means greater resistence to cryptanalysis,
Specific to DES: ① The design of the F function ② How round keys are derived from the main key.

DES Coarse Structure

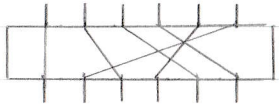


The L and R each have 32 bits, and the round key K 48 bits
The F function, on input R and K, produces 32 bits:
$F(R,K) = P(S(E(R) \oplus K))$

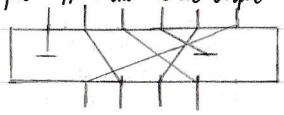
Where E: expends 32 bits to 48 bits
S: shrinks it back to 32 bits
P: permutes the 32 bits

DES Fine Structure Blocks

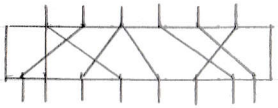
Permutations: Fixed known mapping 32-32 bits



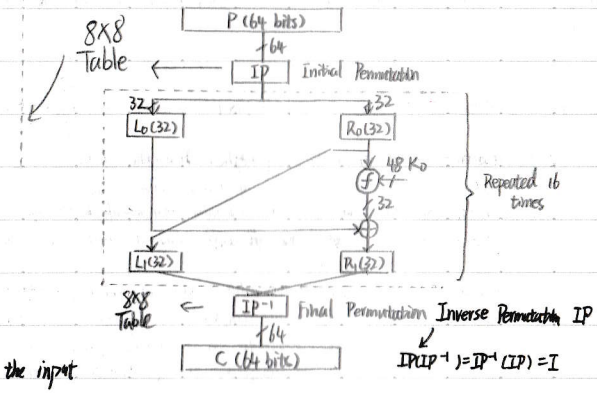
Compression Permutations (Permuted Choice): Fixed known subset of 56 bit input mapped onto 48 bit output



Expansion permutation:
→ 32-bit data shuffled and mapped (both operations fixed and known) onto 48-bit by duplicating 16 input bits.
→ This makes diffusion quicker



Substitution:
48 bits of data are divided into eight blocks of 6 bits. There are eight S-boxes each map 6 to 4 bits



$IP(IP^{-1}) = IP^{-1}(IP) = I$

Initial Permutation IP

This table specifies the input permutation on a 64-bit block. The meaning is as follows: the first bit of the output is taken from the 58th bit of the input; the second bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input

The F Function Expansion function (E) 6×8 table (Column, Row)

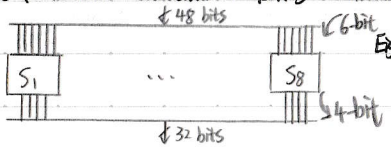
48 bits

Note that some bits from the input are duplicated at the output, therefore the 32-bit half-block is expanded to

The F Function Permutation P 4×8 table

The P permutation shuffles the bits of a 32-bit half-block

The F Function Substitution Boxes Mini-Number



Eight S-boxes each map 6 to 4 bits

$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Each S-box is specified as a 4×16 table
- each row is a permutation of 0-15
- outer bits 1 & 6 of input are used to select one of the four rows
- inner 4 bits of input are used to select a column

All the eight boxes are different

For example, $S_1(101010) = 6 = 0110$.

6 → 0101 → 5
1 → 2

2 unknowns → 2 equation (二元一次方程組)

The F Function — a bad S-box choice

If the S function is linear, then the entire DES cipher may became linear: this means there exists a fixed binary matrix such that:

$$DES(k,m) = 64 \underbrace{\boxed{A}}_{832} \cdot \underbrace{\begin{bmatrix} m \\ K_1 \\ K_2 \\ \vdots \\ K_{16} \end{bmatrix}}_{832} = \boxed{C} \pmod 2$$

← A set of linear equations

Given enough pairs of {c;m}, it is easy to calculate the secret key.

The F function S-boxes and P-box choice

Choosing the S-boxes and P-box at random would result in an insecure block cipher as they may be close to linear functions which will allow key recovery after $\approx 2^{24}$ outputs) using one of the cryptanalysis techniques.
Therefore, a number of rules have been set for the choice of S-boxes
- No output bit should be close to a linear function of the input bits...

DES Design Controversy

Issues	Criticisms
① Secret Design Criteria	Design criteria of round function and key schedules were secret. (although actual design public). However, Fear of trapdoors (後門) has been proved baseless.
② Weak keys	Certain DES keys are weak (encryption and decryption has same effect) Few such keys and their use easily avoided.
③ Inadequate Key Length	56 bits an inadequate key length. Criticized even in 1975 Unsubstantiated claims that NSA insisted on the "small" key length

Strengths of DES

Avalanche effect (雪崩): ① A small change in the plaintext or in the key results in a significant change in the ciphertext.
② An evidence of high degree of diffusion and confusion
③ A desirable property of any encryption algorithm

DES exhibits a strong avalanche effect
- Changing 1 bit in the plaintext affects 34 bits in the ciphertext on average.
- 1-bit change in the key affects 35 bits in the ciphertext on average.

Decryption is performed by exactly the same procedure, except that the keys K1, ..., K16 are used in reverse order.

one bit input change maps to one bit output change → Not secure

Attacks on DES

[Exercise] Suppose that we have a machine consisting of one million (10^6) processors, each of which can test one million (10^6) keys per second. How long is it likely to take before we find a DES key during an exhaustive key search?

$$\frac{2^{56}}{10^6 \times 10^6} / 3600 = 20h$$

No. 10 / 11

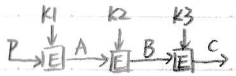
Date

Multiple Encryption with DES

DES is not secure enough. The once large key space, 2^{56}, is now too small.
In 2001, NIST published the Advanced Encryption Standard (AES) as an alternative.
But users in commerce and finance are not ready to give up on DES. [Solution] Use multiple DES with multiple keys

Triple DES with Two Keys

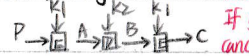
A straightforward implementation would be: $C = E_{K1}(E_{K2}(E_{K3}(P)))$



In practice: $C = E_{K1}(D_{K2}(E_{K1}(P)))$, Why?

It's for backwards compatibility with normal DES
If you use the same key for the first two operations, they cancel out and you are left with a ciphertext encrypted under a single DES key

Also referred as DED encryption

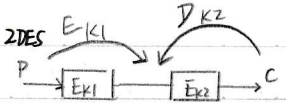


Double DES

Consider 2-DES with two keys: $C = E_{K2}(E_{K1}(P))$
Decryption: $P = D_{K1}(D_{K2}(C))$
Key length: $56 \times 2 = 112$ bits This should have thwarted brute-force attacks? — Wrong

Meet-in-the-Middle Attack on 2DES

2-DES $C = E_{K1}(E_{K2}(P))$

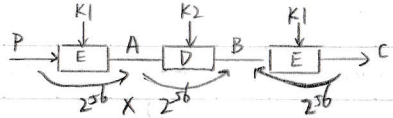


2^{56} steps → ← 2^{56} steps

$2^{56} + 2^{56} = 2^{57}$

Given a known pair (P, C), attacks as follows:
- Encryption P with all 2^{56} possible keys for K1
- Decrypt C with all 2^{56} possible keys for K2
- If $E_{K1}(P) = D_{K2}(C)$, try the keys on another (P', C')
- If words, (K1', K2') = (K1, K2) with high probability
- Takes $O(2^{57})$ steps; not much more than attacking 1-DES

Meet-in-the-Middle Attack on 3DES



1. For each possible key for K1, encrypt P to produce a possible value for A
2. Using this A, and C, attack the 2DES to obtain a pair of keys (K2, K1')
3. If K1' = K1, try the key pair (K1, K2) on another (C', P')
4. If it works, (K1, K2) is the key pair with high probability
5. It takes $O(2^{56} \times 2^{56}) = O(2^{111})$ steps on average.

10101111_2
↓
coefficient of x to power of 7
x^7

$x^7 + x^6$
$x^7 + 0 + x^5$
↓
0 (XOR 同士=0)

No. 11

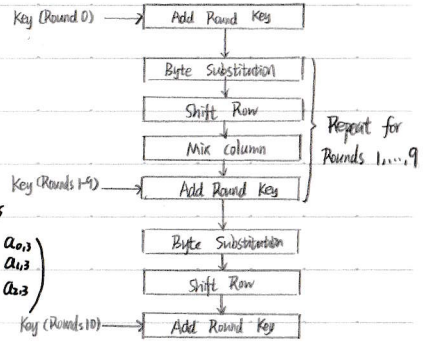
Date

△ Encrypt and Decrypt using Advance Encryption Standard Algorithm (AES)

AES Encryption Algorithm

AES is based on substitution-permutation network and not on Feistel network.

Description is done by applying the inverse function of each step (i.e. all functions must be invertible).



Each block consists of 128 bits, and these are divided into 16 bytes
Each of the operations acts upon these 8 bit bytes in State

State is a variable holding the current data block.

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & \cdots & \end{pmatrix}$$

Each $(a_{i,j})$ is an 8-bit byte, viewed as elements of the Galois field $GF(2^8)$ 有限域
Eight bits in a byte is considered as a polynomial:

$$b_7 \cdot x^7 + b_6 \cdot x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 \quad \text{e.g. } 10010011_2 = x^7 + x^4 + 1$$

Different arithematic operations
Addition is XOR (⊕)
Multiplication is modular using a prime polynomial $(x^8 + x^4 + x^3 + x + 1)$ i.e. (100011011). To multiply two elements, their ~~performance~~ representative polynomials are multiplied then divided by the prime polynomial above. The answer is the remainder of the division

AES - Byte Substitution

Each byte a in the data block in the state matrix is substituted with another byte $Sub(a) = Aa^{-1} + b$
Where: a^{-1} is the multiplicative inverse of the a in $GF(2^8)$ mod $(x^8 + x^4 + x^3 + x + 1)$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Note, zero, which has no inverse, is set to zero

$x^8 + x^4 + 1 \;|\; x$

Semantic Security of PRP (one-time-key)

Definition : E is semantically secure if for all "efficient" Adversary , ~~AEA~~, $Adv[A,E]$ is negligible

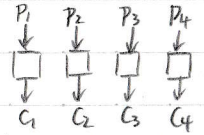
$m_0, m_1 \in M$, $length(m1) = length(m2)$

$$Adv[A,E] = |EXP_A[E(k,m1)] - EXP_A[E(k,m2)]|$$

Exp the experiment that an adversary performs on encrypted messages

The attacker cannot distinguish between encrypted messages.

How not to use a block cipher?



The plaintext is broken into blocks , $P_1, P_2, P_3, \ldots$

Each block is encrypted independently of the other blocks $C_i = E_K(P_i)$

For a given key, this mode behaves like we have a gigantic codebook , in which each plaintext block has an entry , hence the name Electronic Code Book

[Example]

Consider the two messages :

$m1$ = "hey hey", $m2$ = "hey you" which are encrypted using ECB mode : $E(k,m) = \{c1, c2\}$.

Assume we have an attacker with the following experiment :

$EXP_A[E(k,m)]$ = if $c1 = c2$ output 0 , else output 1

What is the advantage of the attack

$Adv[A,E] = |EXP_A[E(k,m1)] - EXP_A[E(k,m2)]| = ?$

Electronic Code Book Weakness

- Repetitive information contained in the plaintext may show in the ciphertext , if aligned with blocks
- If the same message is encrypted (with the same key) and sent twice , their cipher texts are the same. So an attacker can learn that two encrypted files are the same , two encrypted packets are the same , etc. This may lead to significant attacks when message space M is small.
- You can only use ECB securely if the message length is equal to the data block length and you change the key for each message

How to use a block cipher securely without changing the key

- When the key is used to encrypt many block , the adversary can see many ciphertexts with same key.
- This may make the system prone to attacks such as chosen-plaintext attack (CPA) where in the attacker can obtain the encryption of arbitrary messages of his choice and deduce the secret key
- In both DES and AES one pair of (c,m) is enough to find the key though exhaustive search.

Semantic Security of PRP (many-time key)

Definition: E is semantically ~~security~~ secure under CPA if for all "efficient" Attackers.

$Adv[A,E]$ is negligible, where:

$Adv[A,E] = |EXP_A[E(k,m_1)] - EXP_A[E(k,m_2)]|$

Exp is an experiment that an adversary performs on encrypted messages, to deduce their values.

※ $Adv[A,E]$ represents the attacker ability to distinguish between the encryption of different messages for all explicit $m_0, m_1 \in M$ ($length(m_1) = length(m_2)$)

In other words: An encryption function E can only be semantically secure if the attackers cannot distinguish between encrypted messages, in which case whatever experiment they carry their advantage will be negligible ($Adv(A,E) \leq 0$)

One way to achieve this is to ensure that the encryption algorithm produce different outputs for the same message, for each encryption round.

How to use a block cipher securely ?

If secret key is to be used multiple times than given the same plaintext message twice, encryption must produce different outputs. There are a number of methods which are approved by NIST:

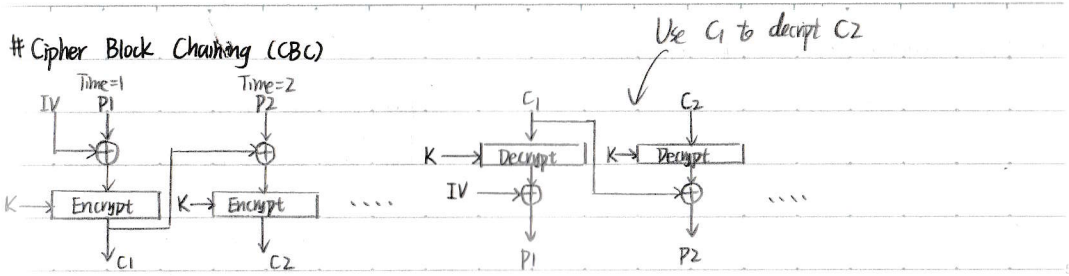
① Cipher Block Chaining mode (CBC)
② Counter mode (CTR)
③ Cipher feedback mode (CFB)
④ Output feedback mode (OFB)

Adv: advantage of adversary

ed: 1

2. bit/s

Cipher Block Chaining (CBC)



The plaintext is broken into blocks $P_1, P_2, P_3, \ldots$
Each plaintext block is XORed (chained) with the previous ciphertext block before encryption (hence the name):

$$\begin{cases} C_i = E_K(C_{i-1} \oplus P_i) \\ C_0 = IV \end{cases}$$

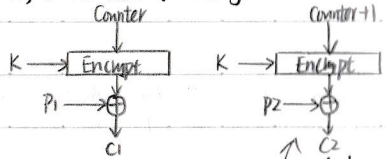
Use the Initial Vector to start the process

Decryption: $P_i = C_{i-1} \oplus D_K(C_i)$
Application: general block-oriented transmission. Not only depend on current ciphertext, but also depend on previous

[Remarks]

The same key is used for all blocks (prone to CPA)
Error propagate
Initialization vector (IV) must be known to both the sender & receiver. Typically, IV is either a fixed value or is sent encrypted in ECB mode before the rest of message.
Serial processing (slow!)

Counter Mode (CTR)



[Encryption]:

$T_1 = IV$
$T_i = T_{i-1} + 1$
$C_i = P_i$ XOR $E_K(T_i)$

A counter T is initialized to some IV and then incremented by 1 for each subsequent plaintext block.

C_2 can be obtained independent of C_1

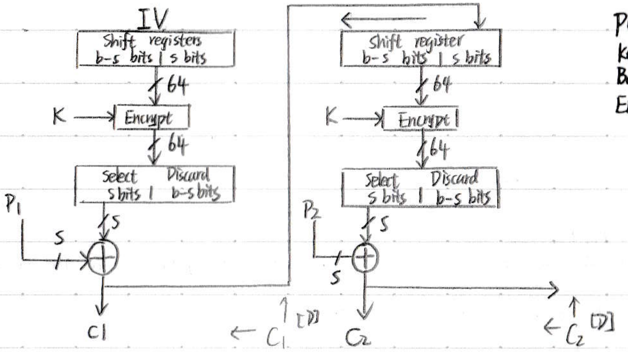
[Remarks]

Weaknesses:
① The same key is used for all blocks (prone to CPA)
② IV should not be reused.

Strengths:
① Needs only the encryption algorithm
② Fast encryption/decryption; blocks can be processed (encrypted or decrypted) in parallel
good for high speed links.

Use the first ciphertext to generate the second ciphertext

Cipher Feedback (CFB) Mode Encryption



Plaintext blocks : $P_1, P_2, \ldots$

Key : k

Basic idea : construct key stream $k_1, k_2, k_3, \ldots$

Encryption :

$$\begin{cases} C_0 = IV \\ K_i = E_k(C_{i-1}), \text{ for } i \geq 1 \\ C_i = P_i \oplus K_i, \text{ for } i \geq 1 \end{cases}$$

[Decryption]

Generate key stream K1, K2, K3, K4, ... the same way as for encryption
Then decrypt each ciphertext segment as :

$$P_i = C_i \oplus K_i$$

[Remark on CFB]

- The block cipher is used as a stream cipher
- Appropriate when data arrives in bits / bytes
- s can be any value, a common value is $s=8$
- A cipher segment depends on the current and all preceding plaintext segments
- A corrupted ciphertext segment during transmission will affect the current and next several plaintext segments

Output Feedback is more reliable

Types of attacks

The objective of the following attacks is to systematically recover plaintext from ciphertext, or even more drastically, to deduce the decryption key.

[A ciphertext-only attack]

It is one where the adversary (or cryptanalyst) tries to deduce the decryption key or plaintext or plaintext by only observing ciphertext. Any encryption scheme vulnerable to this type of attack is considered to be completely insecure (e.g. frequency analysis on a shift cipher).

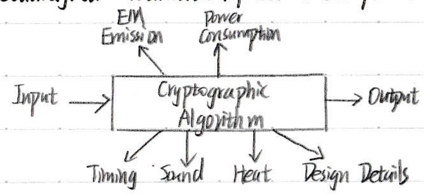
[A known-plaintext attack]

It is one where the adversary has a quantity of plaintext and corresponding ciphertext. This type of attack is typically only marginally more difficult to mount. (e.g. meet in the middle attack on DES).

Another variation of this is : chosen-plaintext attack where the adversary chooses plaintext and is then given corresponding ciphertext. Subsequently, the adversary uses any information deduced in order to recover plaintext corresponding to previously unseen cipher text.

[Side Channel Attacks]

These are based on the analysis of information leaked from the physical implementation of the system such as electromagnetic radiation, power consumption, sounds.



用 $\begin{cases} 0 \to 0,1 \\ 1 \to 1,0 \end{cases}$ 去防止这种攻击，在 this case, the power consumption may be balanced, but you need to implement the algorithm

No.

Date

Number Theory

△ Classify computational problems according to their complexity

Computational Complexity Theory

Computional complexity theory is a branch of the theory of computation in mathematics that focus on classifying computational problems according to their inherent difficulty. 固有的困难

A computational problem is said to be a task that is in principle amenable to being originally 检验
solved by a computer in other words, the problem may be solved by automatic application of mathematical steps, such as an algorithm

The Big O notation is used to classify algorithms by how they respond to changes in input size in terms of their processing time or working space requirements
Big O notation is useful when analyzing algorithms for complexity
(eg. this notation can be used indicate the relationship between the size of the input and the number of steps needed to execute an algorithm on a space constrained machine).
Example:
$T(n) = n^2+n+2 \Rightarrow$ We can state $T(n) \in O(n^2)$ $n^2 \gg n+2$ if n is big
大O符号是用于描述函数渐近行为的数学符号(用另一个函数来描述一个函数数量级的渐近上界.

Big O notation is a mathematical notation that describes the limiting behaviour of a function when the argument tends towards a particular value or infinity. In computer science, big O notation is used to classify algorithms according to how their running time or space requirements grow as the input size grows

QUIZ

Computation Problems
① Adding two N-digit numbers $n+n \rightarrow O(n)$
② Multiply two N-digit numbers $n*n \rightarrow O(n^2)$
③ Cracking a n-letter Transposition Cipher by Brute Force Search $n!$
④ Cracking a n-letter shift by Brute Force Search $O(26^n)$

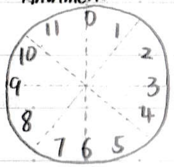
数论：给定一个正整数m，如果两个整数a和b满足(a-b)能够被m整除，那么就称a与b对模m同余，记作 $a \equiv b \pmod m$ 读作 a同余b模m

No.

Date

Modular Arithmetic Basics

Modular Arithmetic



Normal Arithmetic

Important number systems

- $\mathbb{Z}$: the set of all integers $0, \pm1, \pm2$
- 有理数 $\mathbb{Q}$: the set of all rational number a/b $(a, b \in \mathbb{Z}, b \neq 0)$
- 实数 $\mathbb{R}$: the set of all real numbers
- 复数 $\mathbb{C}$: the set of all complex numbers $a+bj$ $(a, b\ \mathbb{R})$

$\mathbb{Z}_{12} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$

$a = b \pmod n$

$a = b + k * n$

where k is ANY integer

$13 = 1 \pmod{12}$

$26 = 2 \pmod{12}$

$13 = 1 + 1 * 12$

$26 = 2 + 2 * 12$

$20 \% 6 = 8$ $20 - 8 = 12$ $20 \% 3 = 2$, $8 \% 3 = 2$ $20 \% 6 = 2$, $8 \% 6 = 2$ $20 \% 4 = 0$, $8 \% 4 = 0$

The mod congruence

一致

Definition: Let $a1, a2$ be integers and b be a positive integer. We say that $a1$ is congruent to $a2$ module b (denoted by $a1 \equiv a2 \pmod b$)

if $(a2 - a1)$ is a multiple of b $\Rightarrow$ Equivalently: $a1 \bmod b = a2 \bmod b$.

The (mod) congruence is useful for manipulating expressions involving the mod function. It lets us view modular arithmetic relative a fixed base, as creating a number system inside of which all the calculations can be carried out.

[Example 1]

Calculate the mode congruence of 113 in $\mathbb{Z}_{24}$

$113 \bmod 24 \Rightarrow 24 \overline{)113}$, quotient 4, -96, remainder 17 → mode congruence   This means $113 = 17$ in $\mathbb{Z}_{24}$

Tips: using your calculator divide 113 by 24, then the multiply the fractional part of the answer by 24

[模逆元]

一整数a对同余n之模逆元是指满足以下公式的整数 b

$a^{-1} \equiv b \pmod n$

或：

$ab \equiv 1 \pmod n$.

[Eg] 求整数3对同余 11的模逆元素x

$x \equiv 3^{-1} \pmod{11}$

↳ $3x \equiv 1 \pmod{11}$

在整数范围 $\mathbb{Z}_{11}$内，可以找到满足该同余式的 x值为4

$3(4) = 12 \equiv 1 \pmod{11}$

并且 在整数范围$\mathbb{Z}_{11}$内不存在其它满足此同余式的值

故，整数3对同余 11的模逆元素为4.

$3x - 1 = 11k$

$x = 4$

Multiplication and Addition in $\mathbb{Z}_n$

[Example] Compute $100 + 30$ in $\mathbb{Z}_{24}$

We do the addition as normal $100 + 30 = 130$, then using a calculator we find $130/24 = 5.4166667$

Then we multiply the fractional part (0.4166667) by 24, so the answer is 10

Check that $130 = 5 * 24 + 10$

[Exercise] Compute 392×514 in $\mathbb{Z}_{1024}$

$392 \times 514 = 201488$ $201488 / 1024 = 196.7656$ $0.7656 \times 1024 = 783.9744 \approx 784$

$201488 = 196 * 1024 + 784$

Modular Inversion

A modular inverse multiplicative inverse of an integer a is an integer x (模逆元) such that the product ax is congruent to 1 with respect to the modulus m. $ax \equiv 1 \pmod m$

基本原理

Over the ~~relations~~ rationales, inverse of 2 is $\frac{1}{2}$. What about $\mathbb{Z}_n$?

Definition: The inverse of x in $\mathbb{Z}_n$ is an element y in $\mathbb{Z}_n$ such as $x \cdot y = 1$, y is denoted x^{-1}

[Example] What is the multiplicative inverse of $x = 3$ in $\mathbb{Z}_7$

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
$\frac{1}{3}$ → 3	0	3	6	2	5	(1)	4
$\frac{1}{4}$ → 4	0	4	(1)	5	2	6	3
5	0	5	3	1	6	4	(2) → $(6 \times 5) \% 7 = 2$
6	0	6	5	4	3	2	(1) → $(6 \times 6) \% 7 = 1$

↙ 余数表

$\frac{3}{4}$ in $\mathbb{Z}_7$ $\frac{4}{3}x = 7k + 1 = 1, 8, 15, 22, 29, 36$

$\frac{3}{4} = 3 \times \frac{1}{4} = 3 \times 2 = 6 = 6$ in $\mathbb{Z}_7$

if $a > 0$ $ax - 1 = 7k$

$a < 0$ $\frac{1}{a}x - 1 = 7k$

Division in $\mathbb{Z}_n$

[Example] Compute $2/3$ in $\mathbb{Z}_7$

↙用它的倒数找出与之乘积mod 7余1的数

$2/3 = 2 * 1/3 = 2 * 5 = 10 = 3$ in $\mathbb{Z}_7$

$\frac{2}{3}x \equiv 1$

$2 * \frac{1}{3}x \equiv 1$

$2 * 3x \equiv 1$

$\frac{3}{2}x - 1 = 7k$

$x = 10 = 3$ in $\mathbb{Z}_7$

$\frac{3}{2} \times 10 = 15 \equiv 1$ in $\mathbb{Z}_7$

Do all elements have an inverse in $\mathbb{Z}_n$?

Lemma: (重点) for all integers x, y there exist integers a, b such that $ax + by = \gcd(x, y)$

Theorem: x in $\mathbb{Z}_n$ has an inverse if and only if $\gcd(x, n) = 1$

x与n互素

Campus   KOKUYO

No.

Date

欧几里得

Euclid's Algorithm

where r_0, r_1 are positive integer and $r_0 > r_1$

This algorithm is based on the simple observation that: $\gcd(r_0, r_1) = \gcd(r_0 - r_1, r_1)$.

$\gcd(77, 44) = 11 \Rightarrow \gcd(77-44, 44) = 11$

$\gcd(r_0, r_1) = \gcd(r_1, r_0 \bmod r_1)$

便于寻找较大数的公约数

We can apply this process iteratively to find gcd for large numbers:

$\gcd(r_0, r_1) = \gcd(r_0 - r_1, r_1) = \gcd(r_0 - 2r_1, r_1) = \cdots = \gcd(r_0 - mr_1, r_1)$ as $r_0 - mr_1 > 0$

To reduce the number of steps we use the maximum value of m, in this case

$r_0 - mr_1 = r_0 \bmod r_1$

$\Rightarrow \gcd(r_0, r_1) = \gcd(r_0 \bmod r_1, r_1)$

Since $r_0 \bmod r_1 < r_1$, we need to swap them

$\gcd(r_0, r_1) = \gcd(r_1, r_0 \bmod r_1)$.

大数放前面?

[Example] Calculate gcd (60, 22)

$60 = 22 \times 2 + 16$	$\gcd(60, 22) = \gcd(22, 16)$	$\gcd(22, 60 \bmod 22)$
$22 = 16 + 6$	$\gcd(22, 16) = \gcd(16, 6)$	$\gcd(6, 22 \bmod 16)$
$16 = 2 \times 6 + 4$	$\gcd(16, 6) = \gcd(6, 4)$	$\gcd(6, 16 \bmod 6)$
$6 = 4 + 2$	$\gcd(6, 4) = \gcd(4, 2)$	$\gcd(4, 6 \bmod 4)$
$4 = 2 \times 2 + 0$	$\gcd(4, 2) = \gcd(2, 0)$	$\gcd(2, 4 \bmod 2)$

5 in Z_{31}

$31 = 5(6) + 1 \mid 1 = 31 - 5(6)$

$5 = 1(5)$

$-6 \equiv 25 \pmod{31}$

$1 = 5 - 1(4) = 5 - 4(31 - 5(6))$

$= 5 - 31(4) + 5(24)$

$= 5(25) - 31(4)$

$1 \equiv 5(25) \bmod 31$

124 125

Extend Euclidean Algorithm (扩展欧几里德算法)

A modular multiplicative inverse of a modulo m can be found using the extend Euclidean algorithm

The Euclidean algorithm determines the greatest common divisor(gcd) of two integers, say a and m.

If a has a multiplicative inverse modulo m, this gcd must be 1.

$ax + my = \gcd(a, m) = 1 \Rightarrow ax - 1 = (-y)m \Rightarrow ax \equiv 1 \pmod{m}$

Modular Inversion using Extended Euclidean Algorithm

How to find an inverse of an element in Z_n

① Check if x has an inverse in Z_n ($\gcd(x, N)$) must be equal to 1

② Find a, b such that $a \cdot x + b \cdot y = 1$ (this can be done using the Extended Euclidean algorithm)

③ a is the inverse of x in Z_n

Finding an a multiplicative inverse using the Extended Euclidean Algorithm

[Example] Find the multiplicative inverse of 8 mod 11, using the Euclidean Algorithm.

[Solution] Organize our work carefully. To the Euclidean Algorithm in the left coloum. It will verify that $\gcd(8, 11) = 1$. Then solve for the remainders in the right coloum before back

原理见左方

$11 = 8(1) + 3 \mid 3 = 11 - 8(1)$

$8 = 3(2) + 2 \mid 2 = 8 - 3(2)$

$3 = 2(1) + 1 \mid 1 = 3 - 2(1)$

$2 = 1(2)$

$ax + by = \gcd(x, y) = 1$

Now, reverse the process using the equation on the right

$1 = 3 - 2(1)$

$1 = 3 - (8 - 3(2))(1) = 3 - (8 - 3(2)) = 3(3) - 8$

$1 = (11 - 8(1))(3) - 8 = 11(3) - 8(4) = 11(3) + 8(-4)$

Therefore $1 \equiv 8(-4) \bmod 11$

This can be written as

$1 \equiv 8(7) \bmod 11$ Hence 7 is the inverse of 8 mod 11

$8 \times 7 = 56 \equiv 1 \bmod 11$

Division in Z_n

Compute $10/8$ in Z_n: $\frac{10}{8} = 10 \cdot \frac{1}{8} = 10 * 7 = 70 = 4$ in Z_{11}

[Example] Find the multiple inverse of 9 mod 11

$11 = 9(1) + 2 \mid 2 = 11 - 9(1)$

$9 = 2(4) + 1 \mid 1 = 9 - 2(4)$

$2 = 1(2)$ 2对应上式 $2 = 11 - 9(1)$，注意!!! 用 $(11 - 9(1))(4)$ 替代 $2(4)$

Reverse

$1 = 9 - 2(4)$

$1 = 9 - (11 - 9(1))(4) = 9 - (11(4) - 9(4)) = 9(5) - 11(4) = 9(5) + 11(-4)$

$1 \equiv 9(5) \bmod 11$

7 mod 11

$11 = 7(1) + 4 \mid 4 = 11 - 7(1)$

$7 = 4(1) + 3 \mid 3 = 7 - 4(1)$

$4 = 3(1) + 1 \mid 1 = 4 - 3(1)$

$3 = 1(3)$

Reverse

$1 = 4 - 3(1)$

$1 = 4 - (7 - 4(1)) = 4(2) - 7(1)$

$1 = 11(2) - 7(3) = 7(-3) + 11(2)$

$1 \equiv 7(8)$

Fermat's little theorem states that if p is a prime number, then for any integer a, the number $a^p - a$ is an integer multiple of p. $a^p \equiv a \pmod p$

$a^{p-1} \equiv 1 \pmod p$

No.

Date

Modular Inversion using Fermat's Little theorem (费马小定理)

Fermet Little Theorem: Let p be a prime

$\forall x \in (\mathbb{Z}_p)^*: x^{p-1} \equiv 1$ in $\mathbb{Z}_p$

Where $(\mathbb{Z}_N)^*$ = (set of invertible elements in $\mathbb{Z}_n$

[Example]: $p=5$, $3^4 = 81 = 1$ in $\mathbb{Z}_5$ $\quad p=3: 2^4=16 \ 3^4=81 \ 4^4=256$ $\quad p=7: 2^6=64 \ 3^6=729 \ 4^6 \% 7 = 1$

How to use Fermat's theorem to find a modular inverse in $\mathbb{Z}_p$ (If modulo is a prime num, x的逆元为 x^{p-2})

[Theorem] $\forall x \in (\mathbb{Z}_p)^*: x^{p-1} \equiv 1$ in $\mathbb{Z}_p$ (p is a prime)

[Solution] $\forall x \in (\mathbb{Z}_p)^* \Rightarrow x \cdot x^{p-2} \equiv 1 \Rightarrow \boxed{x^{-1} \equiv x^{p-2}}$ in $\mathbb{Z}_p$

Another way to compute inverses, but less efficient than Euclid time $= O(n^3)$

Modular Inversion using Fermat's Little theorem

[Example] Compute $\frac{10}{8}$ in $\mathbb{Z}_{11}$ (11 is a prime number, Fermet little theorem can be applied)

$10/8 = 10 \cdot \frac{1}{8} = 10 \cdot 8^{-1}$

$8^{-1} = 8^{11-2} = 8^9 = 134217728 = 7$ in $\mathbb{Z}_{11}$ $\quad x^{-1} \equiv x^{p-2}$

Therefore $10/8 = 10 \cdot 7 = 70 = 4$ in $\mathbb{Z}_{11}$

根: $3^{\frac{1}{7}} \equiv ?$ 指 $?^7 \equiv 3 \pmod n$

No.

Date

Computing roots in $\mathbb{Z}_p$ $\quad x^e \equiv c$ in $\mathbb{Z}_p$

Definition: Let p be a prime and $c \in \mathbb{Z}_p$. s.t.
Let $x \in \mathbb{Z}_p$ s.t. $x^e = c$ in $\mathbb{Z}_p$. x is called an e'th root of c.

Example:
$7^{\frac{1}{3}} = 6$ in $\mathbb{Z}_{11}$ $\quad 1^{\frac{1}{3}} = 1$ in $\mathbb{Z}_{11}$
$3^{\frac{1}{3}} = 5$ in $\mathbb{Z}_{11}$ $\quad 2^{\frac{1}{3}} = ?$

When does $c^{\frac{1}{e}}$ in $\mathbb{Z}_p$ exist? Can we compute it efficiently?

$7 \equiv 6^3$ $\quad 1 \equiv 1^3$
$3 \equiv 5^2$ $\quad 2 \equiv C^2 \Rightarrow C^2 - 2 = 11k \rightarrow$ ($C^2 =$) 2, 13, 24, 35, 46, 57, 68, 79 ... X
↳ $3 \equiv C^2 \rightarrow C^2 - 3 = 11k \rightarrow 3, 14, (25) \rightarrow C = 5$

Computing roots in $\mathbb{Z}_p$ $\quad (\gcd(e, p-1) = 1)$

仅当 e 与 p-1 互素时可以用

$\forall c: c^{\frac{1}{e}} = c^d$ 想求 $\frac{1}{e}$ 的根，先找 e 在 p-1 中的逆元

Case 1: if $\gcd(e, p-1) = 1$
Theorem: if $\gcd(e, p-1) = 1$, d is the inverse of e in $\mathbb{Z}_{p-1}$ ($d = e^{-1}$ in $\mathbb{Z}_{p-1}$), than $c^{\frac{1}{e}} = c^d$ in $\mathbb{Z}_p$
To compute the eth root of C in this case:
① Find the modular inverse of e in $\mathbb{Z}_{p-1}$ (let us call it d)
② Compute $c^{\frac{1}{e}} = c^d$ in $\mathbb{Z}_p$

Case 2: if $\gcd(e, p-1) \neq 1$
In this case the problem of finding an inverse is harder (e.g. computing the square root ($e \equiv 2$) modular odd prime, in this case $\gcd(2, p-1) \neq 1$)

[Example] Compute $7^{\frac{1}{11}}$ in $\mathbb{Z}_{17}$ { First we check if $\gcd(e, p-1) = 1$ / $\gcd(e, p-1) = \gcd(11, 16)$ therefore:

$16 = 11 + 5 \quad | \ 5 = 16 - 11$
$11 = (2)5 + 1 \quad | \ 1 = 11 - 2*5$
$5 = 5(1)$

Therefore $\gcd(11, 16) = \gcd(1, 0) = 1$ so the previous theorem applied
Now reverse the process using the equations on the right in order to compute the inverse of 11 in $\mathbb{Z}_{16}$

$1 = 11 - 2*5$
$1 = 11 - 2(16 - 11)$
$1 = 3(11) - 2(16)$
Therefore $1 \equiv 3*11 \mod 16$
Hence 3 is the multiplicative inverse of 11 mod 16

$7^{\frac{1}{11}} = 7^3 = 3$ in $\mathbb{Z}_{17}$

$7^{\frac{1}{11}}$ in $\mathbb{Z}_{17}$ $\quad e=11$ $\quad p=17$ $\quad p-1=16$
$\gcd(16, 11)$
$16 = 11(1) + 5 \quad 5 = 16 - 11(1)$
$11 = 5(2) + 1 \quad 1 = 11 - 5(2)$
$5 = 1(5)$
Reverse
$1 = 11 - 5(2) = 11 - (16 - 11(1))(2)$
$= 11 - 16(2) + 11(2)$
$= 11(3) - 16(2)$
↓ 互质
$7^{\frac{1}{11}} = 7^{(3)} \equiv 3$ in $\mathbb{Z}_{17}$

Computing roots in Z_p ($\gcd(e, p-1) \neq 1$)

Special case: computing the square root in Z_p

Quadratic residue (Q.R.): an element x in Z_p is said to be a quadratic residue (Q.R.) if it has a square root in Z_p

[Example]: in Z_{11}: $\sqrt{1} = \{1, 10\}$, $\sqrt{4} = \{2, 9\}$, $\sqrt{9} = \{3, 8\}$, $\sqrt{5} = \{4, 7\}$, $\sqrt{3} = \{5, 6\}$

In this case: $\{1, 4, 9, 5, 3\}$ are quadratic residues.

$9 = 3^2 \bmod 11$, $9 = 8^2 \bmod 11$

$3 = 5^2 \bmod 11$, $3 = 6^2 \bmod 11$

Computing the square root modular prime

How can we tell which elements are Q.R

Euler's theorem: Let p be an odd prime, if x in $(Z_p)^*$ is a Q.R. then $x^{\frac{p-1}{2}} = 1$ in Z_p

Example:

in Z_{11}: $1^5\ 2^5\ 3^5\ 4^5\ 5^5\ 6^5\ 7^5\ 8^5\ 9^5\ 10^5$ $\{1, 3, 4, 5, 9\}$ is Q.R.

$= 1\ -1\ 1\ 1\ 1\ -1\ -1\ -1\ 1\ -1$

This theory is very useful for computing the order of elliptic curve groups

Case 1: $p = 3 \pmod 4$ (mod 4 余 3 的情况)

Theorem: if $c \in (Z_p)^*$ is Q.R. Then $\sqrt{c} = c^{\frac{p+1}{4}}$ in Z_p

Proof: $[c^{\frac{p+1}{4}}]^2 = c^{\frac{p+1}{2}} = \underbrace{c^{\frac{p-1}{2}}}_{1} \cdot c = \underbrace{(c^{p-1})^{\frac{1}{2}}}_{1} \cdot c = c$ in Z_p

Case 2: $p = 1 \pmod 4$

In this case finding the square root can be done using a randomized algorithm with run time $\approx O(\log^3 p)$

[Example for case 1]

Compute $\sqrt{6}$ in Z_{43} given 6 is a QR in Z_{43}

$p = 43 = 3 \bmod (4)$

Therefore $\sqrt{6} = 6^{\frac{43+1}{4}} = 6^{11} = 36$ $\quad 36 = 7$, $-36 = 7$, $36 = 36$

Check: $36 * 36 = 6$ in Z_{43}

$7 \times 7 = 49 = 6$ in Z_{43}

$5^3 = 125 \% 11 = 4$ $\quad 4, 7$

Groups

Let G be a non-empty set, and let $*$ be a binary operation on G.
This means that for every two points $a, b \in G$, a value $a * b$ is defined.
We say that G is a group if it has the following properties:

① Closure: $\forall a, b \in G$ then $(a * b) \in G$ 封闭性

② Associativity: $\forall a, b, c \in G$ then $(a * b) * c = a * (b * c)$ 结合性

③ Identity: there exists $e \in G$ such that $a * e = a = e * a$ for all $a \in G$ 单一性

④ Invertability: for every $a \in G$ there exists $a_i \in G$ such that $a * a_i = e = a_i * a$ 可逆性

[Example]

The number systems 整 Z, 理实 Q, R, 复 C and Z_n are groups under addition with $* = +$, $e = 0$ and $a_i = -a$.

[Quiz]

Let N be a positive number. Prove that Z_N is a group under addition modulo N.

[Solution]

Addition modulo N: $a, b \to a + b \bmod N$

① Closure: $a, b \in Z_N \Rightarrow a + b \bmod N \in Z_N$

② Associative: $((a+b) \bmod N) + c) \bmod N = (a + (b + c \bmod N)) \bmod N$

③ Identity: $a + 0 \equiv 0 + a \equiv a \pmod N$

④ Inverse: Inverse of a is $-a \equiv N - a \pmod N$

[Quiz]

Prove that $Z^*_{12} = \{1, 5, 7, 11\}$ is a group under multiplication modulo 12

[Solution]

① Closure: $a, b \in Z^*_{12} \Rightarrow ab \bmod 12 \in Z^*_{12}$. That is

互素 $\gcd(a, 12) = \gcd(b, 12) = 1 \Rightarrow \gcd(ab \bmod 12, 12) = 1$

Check: $5 \cdot 7 \bmod 12 = 35 \bmod 12 = 11 \in Z^*_{12}$

If $a, b \in Z^*_{12}$, $ab \bmod 12$ can never be 3!

$5 \times 5 = 25 \bmod 12 = 1$, $5 \times 7 = 35 \bmod 12 = 11$, $5 \times 11 = 55 \bmod 12 = 7$, $7 \times 11 = 77 \bmod 12 = 5$

② Associative: $((ab \bmod 12)c) \bmod 12 = (a(bc \bmod 12)) \bmod 12$

Check: $(5 \cdot 7 \bmod 12) \cdot 11 \bmod 12 = (35 \bmod 12) \cdot 11 \bmod 12 = 1$

$5(7 \cdot 11 \bmod 12) \bmod 12 = 5 \cdot (77 \bmod 12) \bmod 12 = 5 \cdot 5 \bmod 12 = 1$

③ Identity: 1 is the identity element because $a \cdot 1 \equiv 1 \cdot a \equiv a \pmod{12}$ for all a.

④ Invertability: $\forall a \in Z^*_{12}$, $\exists a^{-1} \in Z^*_{12}$ such that $a \cdot a^{-1} \bmod 12 = 1$

Check: 5^{-1} is the $x \in Z^*_{12}$ satisfying $5x \equiv 1 \pmod{12}$

No.

Date

Group Order

The order of a group G is its size |G|, meaning the number of elements in it.

Example:

the order of $(\mathbb{Z}_{12}, +)$ is 12

[Quize]

What is the order of $(\mathbb{Z}^*_{12}, *)$

阿贝尔群/交换群

Abelian Groups (Commutativity)

Definition: A group G is said to be commutative (or abelian) if $(a*b)=(b*a)$ for all $a,b \in G$

Example:

The sets of non-zero elements in Q, R and C are all commutative groups under multiplication.

Definition: A group G is said to be cyclic if it has a generator g, and element $g \in G$ such that every element $a \in G$ has the form $a=g^i$ (or ig in additive notation) for some integer i.

Example:

Z is cyclic, since every element has the form $1+1+\cdots+1$

However, Q, R and C are not cyclic.

Euler's generalization of Fermat

Euler's ϕ function: For an integer N, we define $\phi(N) = |(\mathbb{Z}_N)^*|$

Example: $\phi(12) = |\{1,5,7,11\}| = 4$

$\phi(7) = 6$

欧拉函数

对正整数 n，欧拉函数是 $\varphi(n)$ 是小于或等于 n 的正整数中与 n 互质的数的数目

How to compute Euler Totient Function $\varphi(N)$.

For $N=p$ (p prime) $\varphi(N) = N-1$

For $N=p \cdot q$ (p, q prime) $\varphi(N) = (p-1)(q-1)$

Example: $\varphi(37) = 36$

$\varphi(21) = (3-1)\times(7-1) = 2\times 6 = 12$

37

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

12 13 14 15 16 17 18 19 20

A discrete logarithm is an integer k exponent solving the equation $b^k = g$, where b and g are elements of a finite group

当模 m 有原根时，设 L 为模 m 的一个原根，则 $N = L^k \pmod m$ 时，

$Ind_L N \equiv k \pmod{\phi(m)}$，此处 $Ind_L N$ 为 N 以整数 L 为底，模 $\phi(m)$ 时的离散对数值

No.

Date

Euler's generalization of Fermat

质数 与模互素的元素的集合

Fermat Little Theorem: Let p be a prime $\forall x \in (\mathbb{Z}_p)^*: x^{p-1} = 1$ in $\mathbb{Z}_p$

Euler's Theorem: a generalisation of Fermat's Theorem

$\forall x \in (\mathbb{Z}_N)^*: x^{\varphi(N)} \bmod N = 1$

Example: $5^{\phi(12)} = 5^4 = 625 = 1$ in $\mathbb{Z}_{12}$

This theorem forms the basis of the RSA cryptosystem

△ Discuss a number of intractable problems in modular arithmetic

棘手的

Discrete Logarithm Problem

Example: Compute the $D\log_2$ for elements of $\mathbb{Z}_{13}$

$y = 2^x$: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

$x = D\log_2(y)$: 0, 1, 4

Definition: Fix a prime $p > 2$ and g in $(\mathbb{Z}_p)^*$ of order q.

Consider the function: $y \mapsto g^{*x}$ in $\mathbb{Z}_p$

Now, consider the inverse function:

$D\log_g(g^x) = x$ where x in $\{0, \ldots, q-2\}$

Given g, x it is relatively easy to compute y

Given g, y it is hard to compute x (Discrete log problem)

Best known Algorithm is General number field sieve: with a run time of the order: $e^{O(\sqrt[3]{(\ln N)})}$

No.

Date

Asymmetric Cryptographic Systems based on Trapdoor Functions 陷门函数

Cryptographic can be classified into two distinct categories:
① Shared key Systems
② Public key systems

Problem with shared key systems

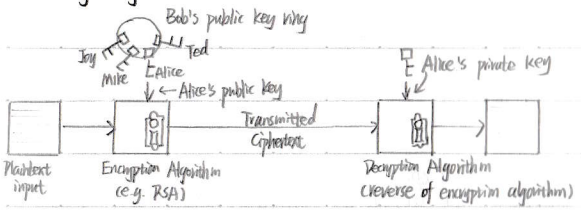
截获者

① Compromised key means interceptors can decrypt any ciphertext they have acquired. Keys can be changed frequently to limit damage.
② Distribution of keys is problematic: keys must be transmitted securely e.g. distribute key in pieces over separate channels

△ Describe the principle of public key crypto systems

Priciples of Cryptographic Systems

[Public Key Systems]



Public Key Systems

Also known as asymmetric-key systems.
Each user has a pair of keys: a public key and a private key.
The public key is used for encryption
The key is known to public (i.e. other users of the systems).
The private key is used for decryption
The key is only known to the owner.

△ Define trapdoor functions

Trapdoor Functions (TDF)

① Construct a pair of functions f, f^{-1}
② Given x, evaluation of $f(x)$ is trivial 平凡,不重要
③ Given only f and y, evaluation of $f^{-1}(y)$ computationally difficult
④ But, given f^{-1} & y, evaluation of $f^{-1}(y)$ is trivial.

A trapdoor function is a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information, called trapdoor.
In mathematical terms, if f is a trapdoor function, then there exists some secret information y, such that given $f(x)$ and y, it is easy to compute x. Consider a padlock and its key. It is trivial to change the padlock from open to closed without using the key, by pushing the shackle into the lock mechanism. Opening the padlock easily, however, requires the key to be used. Here the key is the trapdoor.

Trapdoor Function (TDF)

Definition: a trapdoor function $X \to Y$ is a triple of efficient algorithms (G, F, F^{-1})

① A key generation algorithm (G): randomized algorithm outputs a key pair (pk, sk)

② An Encryption function $(pk, \cdot)$: deterministic algorithm that defines a function $X \to Y$

③ A Decryption Function: $F^{-1}(sk, \cdot)$: defines a function $Y \to X$ that inverts $F(pk, \cdot)$

In other words:

$\forall (pk, sk)$ output by G

$\forall x \in X : F^{-1}(sk, F(pk, x)) = x$

What is a secure TDF?

Intuitively: 直观地 (G, F, F^{-1}) is secure if $F(pk, \cdot)$ is a "one-way" function: i.e. TDF can be evaluated, but cannot be inverted without sk.

It is assumed that the adversary has access to:
① An Encrypted message (c)
② The public key of the intended recipient (pk)
③ The trapdoor function F.
He can try to decrypt the message by finding the inverse function F^{-1} and applying it on the cipher text

Mathematically: (G, F, F^{-1}) is a secure TDF if for all efficient A:

$A(C, F, pk) \to p'$

$Adv[A, F] = Pr[p = p'] <$ negligible

How not to use a Trapdoor Function (TDF)

- Never encrypt by applying F directly to plaintext:

E(pk, m):

output $c \leftarrow F(pk, m)$

D(sk, c):

Output $F^{-1}(sk, c)$

- Problems

① Deterministic: cannot be semantically secure!

② Many attacks exist.

△Outline the principle of RSA ~~based ciphers~~ algorithm

Number Theory Review

Let $N = p \cdot q$ where p, q are prime numbers.

$Z_N = \{0, 1, 2, \dots, N-1\}$; $Z_N^* = \{$invertible elements in $Z_N\}$ 记 Z_N^* 指完全由可逆元素

$x \in Z_N$ is invertible $\Leftrightarrow \gcd(x, N) = 1$ (与模互素)

Number of elements in $(Z_N)^*$ is $\varphi(N) = (p-1)(q-1) = N - p - q + 1$ $(N = p \cdot q)$

How to compute Euler Totient Function $\varphi(N)$:
- For $N = p$ (p prime) $\varphi(N) = N - 1$
- For $N = p \cdot q$ (p, q prime) $\varphi(N) = (p-1)(q-1)$

Euler's Theorem: a generalisation of Fermat's Theorem

$\forall x \in (Z_N)^*: x^{\varphi(N)} \bmod N = 1$

The RSA as TDF Function

Basic key generation algorithm
① Choose random primes p, q
② Set $N = p \cdot q$
③ Choose integers e, d s.t. $e \cdot d = 1 \pmod{\varphi(N)}$ $(e \cdot d \equiv 1)$
④ Output $pk = (N, e)$, $sk = (p, q, d)$

[Encryption Algorithm]

$F(pk, x): (Z_N)^* \to (Z_N)^*: RSA(x, e) = x^e$

[Decryption Algorithm]

$F(sk, y): (Z_N)^* \to (Z_N)^*: RSA(y, d) = y^d$

Prove that $y^d = x$

Chosen integers $ed = k \cdot \varphi(N) + 1 = 1 \pmod{\varphi(N)}$

$\forall x \in (Z_N)^*: x^{\varphi(N)} \bmod N = 1$

$$y^d = (x^e)^d = x^{ed} = x^{(k\varphi(N)+1)} = (x^{(\varphi(N))})^k \cdot x = x$$

RSA Example

Key generation:
① Select primes: $p = 17$ & $q = 11$
② Compute $n = p \cdot q = 17 \times 11 = 187$
③ Compute $\varphi(n) = (p-1)(q-1) = 16 \times 10 = 160$
④ Select $\gcd(e, 160) = 1$; choose $e = 7$
⑤ Determine d: $de = 1 \bmod 160$ and $d < 160$ Value is $d = 23$
since $23 \times 7 = 161 = 10 \times 160 + 1$
⑥ ~~Public~~ Publish public key: $pk = (N, e) = (7, 187)$ 187 7
⑦ Keep secret ~~key~~ private key: $sk = (p, q, d) = (17, 11, 23)$

Encry the message $M = 88$, using this scheme

$M = 88 < 187$

Encryption:

$C = 88^7 \bmod 187 = 11$

Decryption

$M = 11^{23} \bmod 187 = 88$

△ Evaluate the security of RSA-based ciphers

RSA Security

To invert the RSA one-way function (without d) attack must compute

$$y = (x^e) \pmod N \Rightarrow x = \sqrt[e]{y} \pmod N$$

However, if an adversary could factorize N she would know p and q and hence also $\phi(n)$. Since e is public knowledge she could then find d, and easily compute $x = y^d$

RSA ~~depend~~ Security depends on the fact that it is so far very difficult to factorize large integers (很大的数很难进行因式分解)

Brute Force Approach for finding p and q:
① Find $\sqrt{n}$
② It is clear that either $p \le \sqrt{n}$ or $q \le \sqrt{n}$
③ Divide n by each of the primes until a factor is found.

However the number of primes $\le \sqrt{n}$ are about $\frac{2\sqrt{n}}{\ln(n)}$
For large numbers n very ~~diffi~~ inefficient methods

$F_1 = 5$
$F_2 = 17$
$F_3 = 257$
$F_4 = 65537$

A Fermat number is a positive integer of the form: $F_i = 2^{(2^i)} + 1 \quad i \ge 1$
Factorization of Fermat numbers can show the progress that has been made in the area of factorization
$F_5 = 4294967297 = 641 \cdot 6700417$
F_6 ... F_7 = 39 digits ... F_8 (78 digits)
As of 2015 all numbers up to F_{11} have been factored.

Best known algorithm is number field sieve: with a run time of the order: $e^{O\sqrt[3]{\ln N}}$

Current Record: RSA-768 (232 digits) (two years and 100s of machines)

Factorising a 1024-bit integer.?

Choosing the primes p and q

There are some precautions which need to be taken in choosing the primes p and q. For instance, they should not be too close together, for otherwise it might be possible to factorize n by Fermat factorization

Fermat's factorization method is based on the representation of an odd integer as the difference of two squares:

$$N = i^2 - j^2 = (i-j)(i+j)$$

In deed, if N can be factored:

$$N = p \cdot q$$

This can be written as:

$$N = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$$

So, if p and q are very close: $\frac{p-q}{2} \approx 0$ and so $\sqrt{n} \approx \frac{p+q}{2}$

In this case, factorising N will be too easy as follows:
① Choose integers i slightly larger than $\sqrt{n}$
② Test whether $i^2 - N$ is a perfect square j^2 for some integer j.
③ If i can be found then: $N = i^2 - j^2 = (i-j)(i+j)$

not good

[Example] $N = 11413$ p-q is small ✗

larger
107 is sightly ~~smaller~~ than $\sqrt{n}$
So we try $i = 107$, therefore $i^2 - N = 11449 - 11413 = 36 = 6^2$
Thus:

$$N = 117^2 - 6^2 = (107-6)(107+6) = 101 * 113$$

$N = 2027$
$\sqrt{N} \approx 45$
$45^2 - 2027 = 2$
$(45+2)(45-2) = 2027$

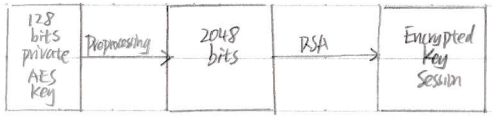
Textbook RSA

Semantically secure encryption system?

Textbook RSA encryption:
{ public key: (N, e) Encrypt: $c \leftarrow m^e \text{ in } Z_N$
{ secret key: (N, d) Decrypt: $c^d \rightarrow m$

How to use RSA encryption in practice

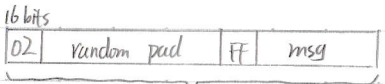
Never use textbook RSA
Example: RSA in practice (RSA-2048 bits encryption of 128 bit AES Private key)



Main questions:
How should the preprocessing be done?
Can we argue about security of resulting system?

PKCS 1 v1.5

PKCS 1 mode 2 : (encryption)



Resulting value is RSA decrypted
Widely deployed, e.g. in HTTPS

Implementation attacks

Timing attacks :

- It is developed in mid-1990's to exploit timing variations in operations e.g. multiplying by small vs large number. It can infer operand size based on ~~time~~ timing taken.
- In RSA based system, the time it takes to compute $y^d \pmod N$ can expose d.
- Countermeasures are based on making the times taken by operations for example by adding random operation to a loop or random keys.

对抗措施

Power attacks :

- It exploits the variation of power consumption of a cipher hardware under different input values to infer the key. For example by analyzing the power consumption of a smartcard while it is computing $c^d \pmod N$ one can expose d.
- Countermeasures include masking the actual power consumption of the circuit during computation through the use of redundant operations. other module methods are based on the use of power balanced logic which consume the same amount of power regardless of inputs.

Exercise :

In a public key system using RSA, you intercept the ciphertext $y=9$ sent to a user whose public key is $e=5$, $n=35$. What is the plaintext x?

$n=35=5\times7=p\times q$.
$\phi(n)=4\times6=24$
$e=5$
$ed \bmod 24 = 1$
$\Rightarrow d=5$

$c=9$.
$m=c^d \bmod n = 9^5 \bmod 35 = 4$
$x=4$

13 $de=6k+1$
$13d=6k+1=1, 61, 121, 181$

In a public-key system using RSA, you intercept the ciphertext $y=30$ sent to a user whose public key is $e=13$, $n=77$. What is the plaintext x?

$n=77=7\times11$
$\phi(n)=6\times10=60$
$de=1 \bmod 60$
$d=37$ where $37\times13=481=60\times8+1$
$m=c^d \bmod n = 30^{37} \bmod 77 = 2$

$x^{13} \bmod 77 = 30$
$x=2$

Matlab mod 函数超出精度范围无法使用
用 ECS 网上自带 Modular Arithmetic Calculator

No.

Date

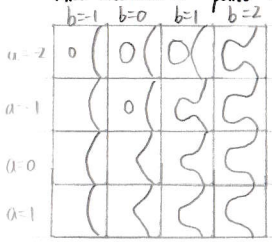
Elliptic Curve Cryptography

△Perform computation on Elliptic Curve Groups over real number

What is an Elliptic Curve?

An elliptic curve E is the graph of an equation of the form $y^2 = x^3 + ax + b$

Also include a "point at infinity"



Restrictions:

Curves, where the polynominal x^3+ax+b has a double root, cannot be used to construct a cryptographic system. This condition is equivalent to

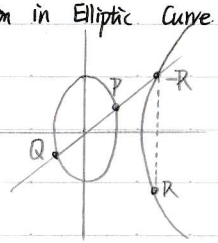
$4a^3 + 27b^2 \neq 0$

This condition ensures the elliptic curve equation has three distinct roots

One of the main reasons for the importance of elliptic curves is that they have a natural abelian group structure. 自然阿贝尔群结构

To understand the group operations, it is useful to start with elliptic curves over the real field F=R.

Addition in Elliptic Curve



ezplot ('y^2 = x^3 -10*x +10') Matlab Function

Given P and Q as two distinct points: on an elliptic curve, and the P is not -Q

Given Addition: To add two distinct points P and Q, a line is drawn through the two points. This line will intersect the elliptic curve in exactly one more point, call -R. The point -R is reflected in the x-axis to the point R.

By definition: addition in an elliptic curve group is $P+Q=R$

E.g. $y^2 = x^3 - 9x + 0$ when $y=0$

-R Q P

$x = 3, -3, 0 \rightarrow 0-3 = -3$ ✓

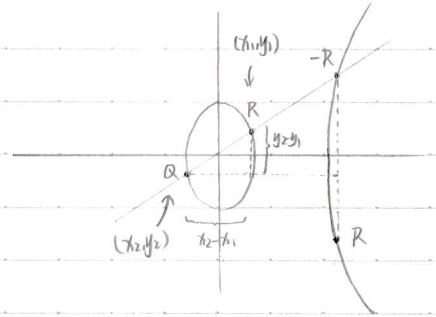
Algebraic Addition in Elliptic Curve

Given point P and Q $\begin{cases} P = (x_1, y_1) \\ Q = (x_2, y_2) \end{cases}$

$P+Q = R = (x_R, y_R)$

$$\begin{cases} x_R = \lambda^2 - x_1 - x_2 \\ y_R = -y_1 + \lambda(x_1 - x_R) \end{cases}$$

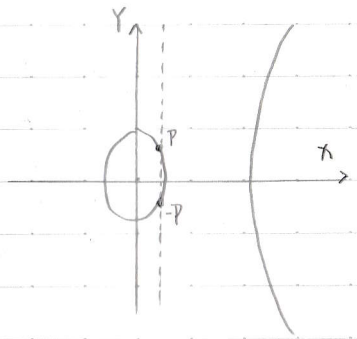
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$



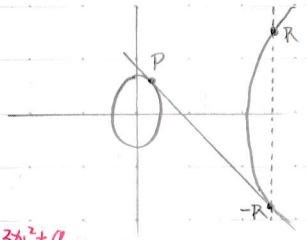
No.

Date

Addition in Elliptic Curve



- Adding the points P and -P on an elliptic curve.
- The line through P and -P is a vertical line which does not intersect the elliptic curve at a third point; thus the points P and -P cannot be added as previously. It is for this reason that the elliptic curve group includes the point at infinity O.
- By definition
  $P + (-P) = O$
  As a result of this equation, $P + O = P$ in the elliptic curve group. O is called the additive identity of the elliptic curve group.



- Doubling the point:

[Case 1: $y_P \neq 0$]

To double the point, draw a tangent line to the curve is drawn at the point P. This line interacts the elliptic curve at exactly one other point, -R.
-R is reflected in the x-axis to R.

$P + P = 2P = R$

$P = (x_1, y_1)$
$2P = R = (x_R, y_R)$ $\Rightarrow$ $\begin{cases} x_R = \lambda^2 - 2x_1 \\ y_R = -y_1 + \lambda(x_1 - x_R) \end{cases}$ $\Rightarrow \lambda = \frac{3x_1^2 + a}{2y_1}$

- Given P and Q as two distinct points $(x_1 \neq x_2)$

$\begin{cases} P = (x_1, y_1) \\ Q = (x_2, y_2) \end{cases}$ $P + Q = R = (x_R, y_R)$

$\begin{cases} x_R = \lambda^2 - x_1 - x_2 \\ y_R = -y_1 + \lambda(x_1 - x_R) \end{cases}$ $\Rightarrow$ $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$

[Case 2: $y_P = 0$]

In this case the tangent line to the elliptic curve at P is vertical and does not intersect the elliptic curve at any other point.
By definition:

$2P = O$ for such a point P.

Note: To find 3P in this situation, one can add $2P + P$. This becomes $P + O = P$. Thus $3P = P$

$3P = P$, $4P = O$, $5P = P$, $6P = O$, $7P = P$, etc.

No.

Date

Groups

Let G be a non-empty set, and let * be a binary operation on G. This means that for every two points $a, b \in G$, value $a * b$ is defined.
We say that G is a group if it has the following properties:

① Closure: $\forall a, b \in G$ the $(a * b) \in G$
② Associativity: $\forall a, b, c \in G$ then $(a * b) * c = a * (b * c)$
③ Identity: there exists $e \in G$ such that $a * e = a = e * a$ for all $a \in G$
④ Invertability: for every $a \in G$, there exists $a_i \in G$, such that $a * a_i = e = a_i * a$.

Abelian Groups

A group is said to be commutative (or abelian) if $(a * b) = (b * a)$ for all $a, b \in G$

Abelian Groups on Elliptic Curves.

Theorem: Any elliptic curve E over a filed F is an abelian group under the operation + defined above

where:

$$E = E(F) = \{(x, y) \mid x, y \in F,\ y^3 = x^2 + ax + b\} \cup \{-\infty, +\infty\}$$

Provided:

$$4a^3 + 27b^2 \neq 0$$

Proof:

① Closure: For P and Q on the curve, point P+Q always exist and is on the curve.
② Associativity: $P + (Q + R) = (P + Q) + R$ (associativity, hard to prove but holds)
③ Identity: $P + O = P$
④ Invertability: $P + -P = O$
⑤ Commutativity: $P + Q = Q + P$ (obvious)

Note: although we have used the field F=R to provide motivation and to enable visualization, we can in fact use the same formulae to define an abelian group structure over any field.

△ Perform computation on Elliptic Curve Groups over finite fields (Z_p)

Elliptic Curves over Z_p

[Definition]

Consider a prime ($p>3$), we define the elliptic curve E over Z_p as follows

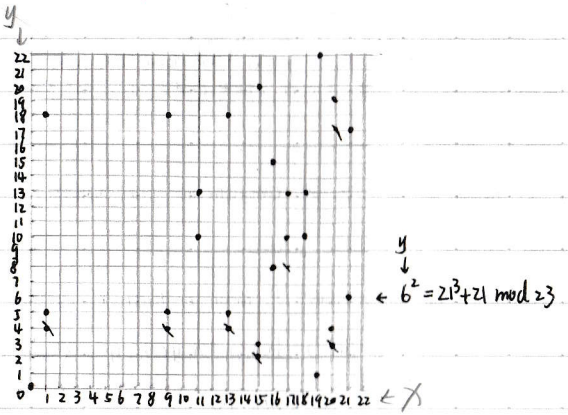
$E = E(F) = \{(x,y) \mid x,y \in F,\ y^2 = x^3 + ax + b \pmod p\} \cup \{-\infty, +\infty\}$

Such that:

$4a^3 + 27b^2 \neq 0 \pmod p$

[Example]

E: $y^2 = x^3 + x$ over Z_{23}



[Exercise] find all points on the elliptic curve below

$y^2 = x^3 + 2x + 1$ over Z_5

y	0	1	2	3	4
4					✓
3			✓	✓	
2			✓		✓
1	✓				
0					

x	0	1	2	3	4
	1	4	13	34	73
	1	4	3	4	3

[Solution]

First check that validity of the equation: $4a^3 + 27b^2 = 59 = 4 \bmod 5$

Then we calculate the points on the elliptic curve

$x=0,\ y^2=1 \Rightarrow y = 1,4 \in Z_5$ ($y^2 \bmod 5 == x^3+2x+1 \bmod 5$)

$x=1,\ y^2=4 \Rightarrow y = 2,3 \in Z_5$

$x=2,\ y^2=13=3 \Rightarrow$ no solution

$x=3,\ y^2=34=4 \Rightarrow y = 2,3 \in Z_5$

$x=4,\ y^2=73=3 \Rightarrow$ no solution

Then points on the elliptic curve are

$(0,1)\ (0,4)\ (1,2)\ (1,3)\ (3,2)\ (3,3)$ and the point at infinity: O

Properties of Elliptic Curves

The order of an Elliptic Curve E over a finite field Z_p is denoted $|E|$ and it refers to the number of points on the curve plus O.

Hasse's theorem gives the upper and lower bounds for $|E|$ as follows:

For an elliptic curve E curve over a finite filed Z_p

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}$$

Thus, taking a large field guarantees a large elliptic curve

Elliptic Curves over Z_p

Find all points on the elliptic curve below

E: $y^2 = x^3 + x + 6$ over Z_{11}

We can also solve such a question by squaring all elements in Z_{11}. This will help all elements which has a square root in Z_{11} (i.e. quadratic residue)

Y	Y^2	x	x^3+x+6	y
0	0	0	6	no
1	1	1	8	no
2	4	2	5	yes 4,7
3	9	3	3	y 5,6
4	5	4	8	~~yes~~ no
5	3	5	4	y 2,9
6	3	6	8	
7	5	7	8	y 2,9
8	9	8	4	y 3,8
9	4	9	7	
10	1	10	4	y 2,9

Point Addition Example Elliptic curves over Z_p

[Exercise] Given E below and $P = (2,7)$, find $2P$

E: $y^2 = x^3 + x + 6$ Z_{11}

Doubling the Points

Case 1: $y_P \neq 0$

2014: Q4-d

Working in Z_{11}, $a=1$, $b=6$ $\quad y^2 = x^3 + x + 6$

$P = (x_1, y_1) = (2,7)$

$\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{3x2^2+1}{2x7} = \frac{13}{14} \pmod{11} = \frac{2}{3} \bmod 11$

$13 \equiv 2 \bmod 11$, $14 \equiv 3 \bmod 11$

$3^? \equiv 1 \bmod 11$, $? = 4$

We need to find the multiplicative inverse of 3 in Z_{11}

This can be doing using EEA and the answer is 4, Therefore $\lambda = \frac{2}{3} \bmod 11 = 2*4 = 8 \bmod 11$

$2P = R = (x_R, y_R)$

$x_R = \lambda^2 - 2x_1 = 8^2 - 2*2 = 64 - 4 = 60 = 5 \bmod 11$

$y_R = -y_1 + \lambda(x_1 - x_R) = -7 + 8(2-5) = -7 - 24 = -31 = 2 \bmod 11$

Therefore $2P = (5,2)$

△ Construct key exchange protocols based on elliptic curve

Euler Theorem

Euler Theorem: $(\mathbb{Z}_p)^*$ is a cyclic group, that is

$\exists g \in (\mathbb{Z}_p)^*$ such that $\{1, g, g^2, g^3, \cdots, g^{p-2}\} = (\mathbb{Z}_p)^*$ g is called a generator of $(\mathbb{Z}_p)^*$

[Example]

$\{2^0, 2^1, 2^2, 2^3\} = \{1, 2, 4, 3\} = (\mathbb{Z}_5^*)^*$ 2 is a generator of $(\mathbb{Z}_5)^*$

$\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\} = (\mathbb{Z}_7^*)^*$ 3 is a generator of $(\mathbb{Z}_7)^*$

$\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4\}$ 2 is not a generator of $(\mathbb{Z}_7)^*$ 也在 1mb 才为 generator

The order of an element in a group

For $g \in (\mathbb{Z}_p)^*$ the set $\{1, g, g^2, g^3, \cdots\}$ is called the group generated by g, denoted $\langle g \rangle$

The order of $g \in (\mathbb{Z}_p)^*$ is the size of $\langle g \rangle$

$\text{ord}_p(g) = |\langle g \rangle| = (\text{smallest } a > 0 \text{ s.t. } g^a = 1 \text{ in } \mathbb{Z}_p)$

Example $(\mathbb{Z}_7)^*$

$\text{ord}_7(3) = 6$ $\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\} = (\mathbb{Z}_7^*)^*$

$\text{Ord}_7(2) = 3$ $\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4\}$

$\text{ord}_7(1) = 1$ $\{1^0, 1^1, 1^2, 1^3, 1^4, 1^5\} = \{1\}$

$\text{ord}_p(g)$ ← generator, ↑ modulo

The order of an element in an Elliptic Curve Group

Definition: For $P \in E(\mathbb{F}_q)$ the set $\{O, P, 2P, 3P, \cdots (m-1)P\}$ is called the group generated by P, denoted $\langle P \rangle$
Where $mP = O$, beyond mP, this pattern repeats in a cycle of length m.

The order of $P \in E(\mathbb{F}_q)$ is the size of $\langle P \rangle$

[Example]

x	x^3+2
0	2
1	3
2	1
3	4
4	1

y	y^2
0	0
1	1
2	4
3	4
4	1

(2,1) (2,4) (2,-1)
(3,2) (3,3) (3,-2)
(4,1) (4,4) (4,-1)

E: $y^2 = x^3 + 2$ in $\mathbb{Z}_5$

The elements of this group are $\{(2,0), (3,\pm 2), (4,\pm 1), O\}$.
If we take $P = (4,1)$ we find that
$2P = (3,-2)$, $3P = (2,0)$, $4P = (3,2)$, $5P = (4,-1)$ $6P = O$ The order of P is 6
Therefore P is a generator and E is a cyclic group

② $P = (4,1)$ $\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{48+0}{2} = 24 \equiv 4 \bmod 5$ $\begin{cases} x_R = \lambda^2 - 2x_1 = 16 - 8 = 8 \equiv 3 \bmod 5 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -1 + 4(4-3) = 3 \end{cases}$ $\Rightarrow 2P = (3,-2)$ ✓

③ $P = (4,1)$, $Q = (3,-2)$ $P+Q = R = (x_R, y_R)$
$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3}{1} = 3$ $\begin{cases} x_R = \lambda^2 - x_1 - x_2 = 9 - 7 = 2 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -1 + 2(4-2) = 5 \equiv 0 \bmod 5 \end{cases}$ $\Rightarrow 3P = (2,0)$ ✓

两个点 (3,3) 不在椭圆曲线上
必须化成 $3 \equiv (-2)$

④ $2P = (3,-2)$ $2P + 2P = 4P$ $\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{27}{6} = \frac{9}{2} = 9 \times \frac{1}{2} = 9 \times 3 = 27 \equiv 2 \bmod 5$
$\begin{cases} x_R = \lambda^2 - 2x_1 = 2 \times 2 - 6 = -2 = 3 \bmod 5 \\ y_R = -y_1 + \lambda(x_1 - x_R) = 2 + 2(3-3) = 2 \end{cases}$ $\Rightarrow 4P = (3,2)$ ✓

⑤ $P = (4,1)$
$Q = 4P = (3,2)$
$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = -1$ $\begin{cases} x_R = \lambda^2 - x_1 - x_2 = 1 - 7 = -6 = 4 \bmod 5 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -1 - (4-4) = -1 \end{cases}$ $\Rightarrow 5P = (4,-1)$ ✓

③ $P = (4,1)$
$Q = (3,3)$
$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-2}{1} = 3$ $\begin{cases} x_R = \lambda^2 - x_1 - x_2 = 9 - 4 - 3 = 2 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -1 + 3(4-2) = 5 \equiv 0 \end{cases}$

RSA : factoring large integers

No.

ECC :

Date

Elliptic Curve Discrete Logarithm Problem

[Problem] Given two points P and Q in an elliptic curve E over a finite field Z_p find an integer i satisfying $Q = i \cdot P$

- The security of ECC depends on how difficult it is to determine i given iP and P This is refered to as the elliptic curve logarithm problem (ECDLP)
- One of the fastest known technique's to ~~slove~~ solve ECDLP is called Pollard rho method
- Compared to factoring integers or polynomials, one can use much smaller numbers for equivalent levels of security.

Elliptic Curve Diffie-Hell Key Exchage

Public knowledge : A group $E(Z_p)$ and a point g of order n.

BOB		ALICE
Choose secret $0<b<n$		Choose secret $0<a<n$
Compute $Q_{Bob} = bg$		Compute $Q_{Alice} = ag$
Send Q_{Bob} to Bob	$\longrightarrow$	to Alice
	$\longleftarrow$	Send Q_{Alice}
Compute bQ_{Alice}		Compute aQ_{Bob}

Bob and Alice have the shared value $bQ_{Alice} = abg = aQ_{Bob}$

[Example]

Given $E: y^2 = x^3 + 7x + 3 \pmod{37}$ and point $(2,5) \Rightarrow b=3$

$10 \times 26 \equiv 1$

① Alice's secret: A=4

② Bob's secret: B=7

③ Alice sends Bob : $4(2,5) = (7,32)$

④ Bob sends Alice $7(2,5) = (18,35)$

⑤ Alice computes : $4(18,35) = (22,1)$

⑥ Bob computes: $7(7,32) = (22,1)$

$P=(x_1, y_1)$ $\lambda = \frac{3x_1^2+a}{2y_1} = \frac{12+7}{10} = \frac{19}{10} = 19 \times 26 = 13 \bmod 37$

$\begin{cases} x_R = \lambda^2 - 2x_1 = 165 = 17 \bmod 37 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -5 + 13(2-17) = 22 \bmod 37 \end{cases}$

$2P = (17,22)$

$\frac{30}{21}$ 30×37

$P=(17,22)$ $\lambda = \frac{3x_1^2+a}{2y_1} = \frac{874}{44} = \frac{437}{22} = 437 \times 32 = 35 \bmod 37$

$\begin{cases} x_R = \lambda^2 - 2x_1 = 35^2 - 34 = 7 \bmod 37 \\ y_R = -y_1 + \lambda(x_1 - x_R) = -22 + 35(17-7) = 32 \bmod 37 \end{cases}$

$4(2,5) = 2(17,22) = (7,32)$

Comparable Key Sizes for Equivalent Security

Symmetric scheme (key size in bits)	ECC-based scheme (size of n in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

△Describe the properties of cyclic elliptic curve groups

Cyclic Elliptic Curve Groups

For cryptographic purposes, it would be good to take E to be an elliptic curve which is a cyclic group, and P to be a generator for E, so that every element of E is a multiple of P.

There is a very well-developed theorey of finite abelian groups, and one of its consequences is the following useful characterisation of cyclic groups :

Theorem: A finite abelian group G is cyclic if and only if, for each prime p divisy $|G|$, it has fewer than p^2-1 elements of order p (in which case it has exactly p-1 of them)

此定

Lemma: If the order of an elliptic curve E over a finite field F_q denoted $|E|$ is a prime number than the group is cyclic and every element is a generator

From a cryptographic viewpoint this will be the best choice to build a system

No.

Date

Campus

KOKUYO

No.

Date

Message Authenticity (真实性) and Digital Signatures

△ Explain the meaning of message integrity (消息完整性)

Message Integrity

Message integrity is achieved by ensuring that a received message has actually originated from the intended party, and was not modified even if an attacker controls the channel.
Stand error-correction techniques not enough
It aims to achieve integrity (完整性) not confidentiality (机密性)
Application examples:
- Protecting operating systems from viruses.
- Ensuring the integrity of banks transections

Privacy (隐私) does not imply authenticity (可靠性)

正交关注点
- Secrecy and integrity are orthogonal concerns
- Possible to have either one without the other
- Encryption does not (in general) provide integrity

Bob: $c \leftarrow Enc_k(m)$, $m_1, m_2, \dots m_n$ → Attacker: $c_1, c_2, \dots c_n$ → $c_1\ c_2 \dots c_n'$ → Alice: $m' \leftarrow Dec_k(c')$, $m_1\ m_2 \dots m_n'$

△ Describe the security requirement of message authentication codes (MACs).

Message Authentication Code (MAC)

A message authentication code is defined by two algorithms:
① Signing Algorithm (Tag Generation): takes as input a message m and a key k and output a tag t
② Verification Algorithm: takes key k, message m, and tag t as input; outputs 1 ("accept") or 0 ("reject")

Bob: Signing Algorithm, $tag \leftarrow S(k,m)$ → $\{m, tag\}$ → Alice: Verification Algorithm, $V(k,m,tag) = \{1,0\}$

Definition: MAC $\Pi = (S,V)$ defined over (K,M,T) is a pair of algorithm:
$S(k,m)$ outputs t in T
$V(k,m,t)$ outputs '1' or '0'
Consistency Condition: For all m's and corresponding k's
$V(m, S(k,m)) = 1$

MAC Security Definition (Why does integrity requires a secret key)

Adversary's power: Chosen Message Attack

Bob: $t \leftarrow S(k,m)$ ⇄ ($m_1\ m_2\ m_3 \dots m_p$ / $t_1\ t_2\ t_3 \dots t_p$) ⇄ Adversary → (m', t')

A MAC $= (S,V)$ is said to be secure if for all "efficient" A:
$Adv[A, MAC] = Pr[V(k, m', t') = 1]$ is "negligible"

伪造
['fɔ:dʒ(ə)rɪ]

What does Forgery means?

If an attacker A is able to produce a pair (m', t'), such that m' did not originate from the sender, and $V(m', k, t') = 1$. Such a pair is called "Forgery" and the attacker is said to have forged.

Attack Model: Chosen-Message Attack

To construct a secure MAC, we assume worst case scenario, in this case: the attacker is assumed to be able to induce 引导 the sender to authenticate 鉴定 message of the attacker's choice.

Security Goal: Existential unforgeability (存在性不可伪造)

A message authentication code is said to be secure if and only if MAC is able to detect any attempt by the adversary to modify the transmitted data.

Attacker should not be able to produce a new valid (message, tag) pair or even produce a new tag for an old message.

[Exercise] Consider $MAC = (S, V)$, such that $t = S(k, m)$ is always 6 bits long
Is this a secure MAC against Chosen Message Attack?

Replay attacks

攻击者发 or fraudulently(欺骗性) repeated or delayed.
is a form of network attack in which a valid data transmission is maliciously 有敌意的

Replay attacks is when an attacker re-send old messages which have valid tags.
Note that replay attacks are not prevented by the use of MACs
No stateless mechanism can prevent them.
Need to protect against replay attacks at a higher level.

△ Construct a MAC based on symmetric ciphers

PRPs and PRFs

Pseudo Random Function (PRF) defined over (K, X, Y): $F: K \times X \to Y$ such that exists "efficient" algorithm to evaluate $F(k, x)$

Pseudo Random Permutation (PRP) defined over (K, X): $E: K \times X \to X$
such that:
① Exists "efficient" deterministic algorithm to evaluate $E(k, x)$
② The function $E(k, \cdot)$ is one to one
③ Exists "efficient" inversion algorithm $D(k, y)$

Constructing a MAC from a secure PRF

A MAC can be constructed from a secure PRF as follows:
For a PRF $F: K \times X \to Y$ define a $MAC = (S, V)$ as:
$S(k, m) := F(k, m)$
$V(k, m, t)$: output 1 if $t = F(k, m)$ and 0 otherwise

Bob $\xrightarrow{(m, t)}$ Alice

Signing Algorithm
$t \leftarrow F(k, m)$

Verification Algorithm
$V(k, m, t) = 1$ only if $t = F(k, m)$

E.g. You can use AES-128 algorithm to construct a MAC for 16-byte message

[Exercise] Suppose $F: K \times X \to Y$ is a secure PRF with $Y = \{0, 1\}^8$
Is the derived MAC a secure MAC system?

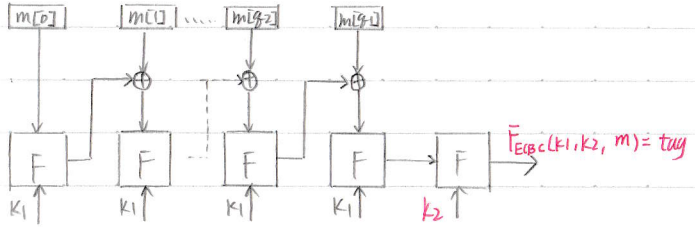
How to generate a tag for long messages

Goal: given a PRF for short message (AES) construct a PRF for long messages
Two main constructions used in practice:
{ ECBC-MAC
HMAC
Both convert a small-MAC into a big-MAC

ECBC-MAC

Definition: Let $F: K \times X \to X$ be a PRP, m is a long message which is divided into q segment $[0] \ldots m[q-1]$, we define a new MAC as follows:

$S(k1, k2, m) = F_{ECBC}(k_1, k_2, m)$
$V(k1, k2, m, t)$: output '1' if $t = F_{ECBC}(k1, k2, m)$ and '0' otherwise.



chosen message attack: adversary can ask for the ciphertext of arbitrary plaintext message.

No.

Date

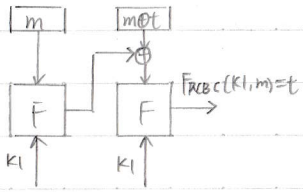
Why the last encryption step in ECBC-MAC? without k_2 $F \to F_{ECBC}$

Suppose we define the MAC with only one key as follows

$S(k_1, m) = F_{RCBC}(k_1, m)$ ← call this scheme raw CBC

Simple Attack on RAW CBC

① Adversary chooses an arbitrary one-block message $m \in X$
② He requests tag from m. Get $t = F_{RCBC}(k, m)$ (Chosen Message Attack)
③ He simply outputs t as MAC forgery for the 2-block message $(m, t \oplus m)$



Proof: $F_{RCBC}(k, (m, t \oplus m)) =$

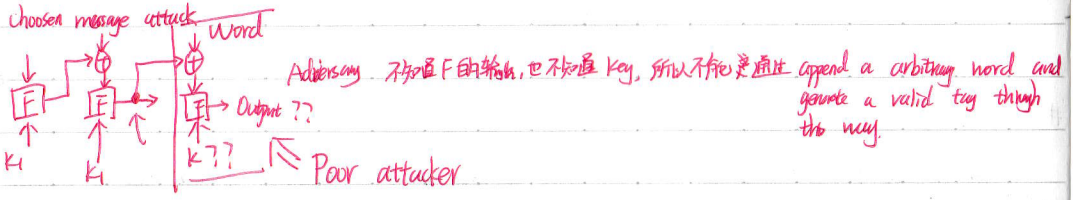
$F(k, F'(k, m) \oplus (t \oplus m)) =$

$F(k, t \oplus (t \oplus m)) =$

$F(K, m) = t$

Hashed Message Authentication Code (HMAC)

This is another approach to constructing a secure MAC for variable length messages
Mostly widely used MAC on Internet



Adversary 不知道 F 的结构, 也不知道 key, 所以不能 append a arbitrary word and generate a valid tag through the way.

Smart Adversary ⇒ evaluate the function by using choosen message attack

Campus

No.

Date

△Explain the principles of Hash Functions

Hash Functions 散列函数

A hash function: maps arbitrary length inputs to a short, fixed-length digest:

$H: \{0,1\}^N \to \{0,1\}^n$ where N is much larger than n.

Properties required of a hash function depend on its applications.
A classical application: To create a one-way password file such that users passwords are stored in as (username, password), but as (username, h(password)).
the serve does not really know your password (you need to change it). if any hacker

Security Requirements

pre-images

Definition: pre-image: if $y = H(x)$, x is a pre-image of y. Each hash value typically has multiple
A collision is a pair of distinct inputs x, x' such that $H(x) = H(x')$

Hash functions used for secure applications must be:

① Pre-image resistant:
if it is computationally infeasible to find a pre-image of a hash value (given a value y it is infeasible to find an x such that $h(x) = y$)

② Collision resistant:
if it is computationally infeasible to find a collision
- Given x and $h(x)$, infeasible to find $y \neq x$ such that $h(y) = h(x)$
- Infeasible to find any x and y, with $x \neq y$ such that $h(x) = h(y)$

Hash functions Applications Examples

① To build message authentication codes
② To create a one-way password file in order to store hash of password not actual password.
③ For instruction detection and virus detection by creating hashes of files on system and monitoring these for any changes.

Let $H: \{0,1\}^N \to \{0,1\}^8$

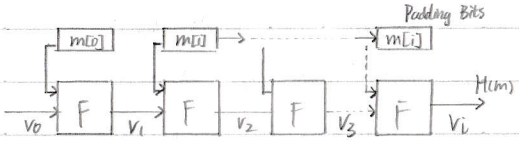
Consider $X \in \{0,1\}^{8N}$, such $X = (x_0, x_1, \dots, x_{N-1})$, each x_i is a byte
We define: $H(X) = x_0 + x_1 + x_2 + \dots + x_{n-1}$

Is this secure? No because it is easy to find collision
Eg. assuming $N=2$

$X_1 = (00000000, 11111111)$ $X_2 = (11111111, 00000000)$
$H(X_1) = H(X_2) = 11111111$

KOKUYO

Collision Resistant Hash Function (Merkle - Damaged Scheme)



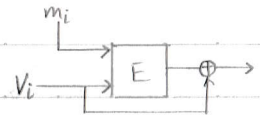
F could be a block cipher or something else.

Operation Principles

The message is broken into blocks of size K
Padding bits are appended to the last block if its size is smaller than K
If the message happens to be a multiple of K, then an extra padding block will be added.
⇒ Padding bits contain a series of 1000 which indicate the end of the message (①) and also it (②) includes the length of the message
The initial vector V0 is fixed.
F is a collision resistant compression function. encryption is naturally a collision-resistant (AES or DES eg) which means plaintext gives different ciphertext

Theorem: If F is collision-resistant, then is H collision-resistant

Collision Resistant Compression Function



The Davies-Meyer compression function:

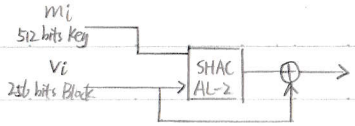
Consider a block cipher $E: (K \times \{0,1\}^n) \to \{0,1\}^n$

The Davies-Meyer compression function (F) is constructed as $F(V,m) = E(m,H) \oplus H$

Secure Hash Algorithm (SHA)

Example: SHA-256

It is based on Merkle-Damaged Scheme
It uses Davies-Meyer compression function with a block cipher called: SHACAL-2



Sender
m,t

Receiver
m,t
↓ recomputes
t'
↓
t=t'? Compare

confirm message transmitted from the sender?
shares a key used for comparing the tag.

Keyed Hash Function as MACs

Hash Functions are used to construct a message authentication code for variable length messages: because secure hash functions are generally faster and widely available.
To achieve authenticity such MACs should include a key along with message so original proposal was as follows:

$$S(k,m) = H(k \| m))$$

Original construction suffered from weaknesses, which eventually led to the creation of HMAC

HMAC Design Objectives (as defined by RFC 2104)

① To use, without modifications, hash functions
② To allow for easy replace-ability of embedded hash function.
③ To preserve original performance of hash function without significant degradation.
④ To use and handle keys in a simple way.
⑤ To have well understood cryptographic analysis of authentication mechanism strength.

Hash Message Authentication Codes (HMAC)

It is specified as Internet standard RFC2014
It uses hash function on the message:

where: opad, ipad are specified padding constants.

$$HMAC: S(k,m) = H(k \oplus opad \| H(k \oplus ipad \| m))$$

Any hash function can be used
eg. MD5, SHA-1, RIPEMD-160, Whirlpool.

Security of MACs from Hash Functions

Why collision resistance is necessary for security:
Example: consider an MAC: $S(k,m) = H(m,k) = t$

Assuming H is not collision resistant, then: S is insecure under the following a 1-chosen message attack

- an adversary finds: $m_0 \neq m_1$, s.t. $H(k,m_0) = H(k,m_1)$
- he asks for $t \leftarrow S(k,m_0)$
- he outputs (m_1, t) as forgery

Attacks on Hash Functions

- Brute Force Attack

Consider a hash function: $H: \{0,1\}^* \to \{0,1\}^n$?

Compute $H(x_1), \dots, H(x_{2^n+1})$

This attack guarantees finding a collision in time $O(2^n)$ hashes

The Birthday Problem

Quiz: What is the probability that at least two of k randomly selected people have the same birthday? (Same month and day, but not necessarily the same year.). It is assumed nobody was born on February 29 and people's birthday are equally distributed over the other 365 days of the year.

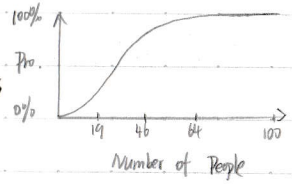
Solution: In a room of k people, consider q: the probability that all people have different birthdays

$$q = \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \frac{362}{365} \cdots \frac{365-(k-1)}{365}$$

$$q = \frac{365!/(365-k)!}{365^k}$$

p: the prob. at the least two of them have the same birthdays

$$p = 1 - q$$



The birthday paradox 生日悖论

Assumption: Consider a set of independent identically distributed integers of size B. Let $r_1, \dots, r_n \in \{1, \dots, B\}$ a subset from B of size n, such that all of its elements are randomly selected.

Theorem: when $n = 1.2 \times B^{\frac{1}{2}}$ then $Pr[\exists i \neq j : r_i = r_j] \geq \frac{1}{2}$

Consider q: the probability that all integers have different values

$$q = Pr[\forall i \neq j : r_i \neq r_j] = \left(\frac{B-1}{B}\right)\left(\frac{B-2}{B}\right) \cdots \left(\frac{B-n+1}{B}\right) = \prod_{i=1}^{n-1}\left(1 - \frac{i}{B}\right)$$

Let p: the probability that at least two of them have the same values

$$p = 1 - q = = 1 - \prod_{i=1}^{n-1}\left(1 - \frac{i}{B}\right)$$

Recall $1 - x \leq e^{-x}$

Therefore:

$$p = 1 - \prod_{i=1}^{n-1}\left(1 - \frac{i}{B}\right) \geq 1 - \prod_{i=1}^{n-1} e^{-\frac{i}{B}} = 1 - e^{\frac{-1}{B}\sum_{i=1}^{n-1} i} \geq 1 - e^{\frac{-n^2}{2B}}$$

When $n = 1.2 \times B^{\frac{1}{2}}$

$$p \geq 1 - e^{\frac{n^2}{2B}} = 1 - e^{-0.72} = 0.53$$

Attacks on Hash Functions

Birthday attack: Let $H: M \to \{0,1\}^n$ be hash function $(|M| >> 2^n)$

Choose $2^{\frac{n}{2}}$ random messages in M: $m_1, \dots, m_{2^{\frac{n}{2}}}$
For $i = 1, \dots, 2^{\frac{n}{2}}$ compute $t_i = H(m_i) \in \{0,1\}^n$
Look for a condition $(t_i = t_j)$. If not found, go back to step 1.

This attacks can find a collision in time $O(2^{n/2})$ hashes
It should be noted here that a quantum computer can find a collision in time $O(2^{\frac{n}{3}})$ hashes

Example of Hash Functions

Function	Digest Size (Bits)	Implementation Speed	Brute Force Attack Time	Birthday Attack Time
SHA-265	256	Fast	$O(2^{256})$	$O(2^{128})$
SHA-512	512	Fast	$O(2^{512})$	$O(2^{256})$
Whirlpool	512	Slow	$O(2^{512})$	$O(2^{256})$

No.

Date

Outline the principles of Digital Signatures

Digital Signature

A digital signature on a message is additional data which provide:

1. Data origin authentication of the signer
   A digital signature validates the message in the sense that assurance is provided about the integrity of the message and of the identity of the entity that signed the message.

2. Non-repudiation 不可抵賴

   A digital signature can be stored by anyone who receives the signed message as evidence that the message was sent and of who sent it. This evidence could later be presented to a third party who could use the evidence to resolve any dispute (爭議) that relates to the contents and/or origin of the message

Digital Signature Systems

Definition : a digital signature system (DSS) consists of a triple of efficient algorithm (G, S, V)

① A key generation algorithm that generate a private key at random from a set of possible private keys and corresponding public key. (pk, sk)
② A signing signing algorithm that, given a message and a private key, produces a signature. $S(sk, m) \to t$
③ A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity. $V(pk, m, t)$ outputs '0' or '1'.

The RSA Digital Signature Scheme

Basic Key Generation Algorithm:
① Choosen random primes p, q
② Set $N = p \cdot q$
③ Choose integers e, d s.t. $e \cdot d = 1 \pmod{\varphi(N)}$
④ Output $pk = (N, e)$, $sk = (p, q, d)$

The RSA Digital Signature Scheme

[RSA signing algorithm]
For each message x compute a signature y as follows
$F(sk, x) : (\mathbb{Z}_N)^* \to (\mathbb{Z}_N)^* : y = RSA(x, d) = x^d$

[RSA Signature Verification Algorithm]
For each received message y
① Compute $F(pk, y) : (\mathbb{Z}_N)^* \to (\mathbb{Z}_N)^* : RSA(y, e) = y^e$
② If $y^e = x$ then output 1 else output 0

Campus

No.

Date

Security of RSA digital signature scheme

An example of an existential forgery attack

[Simple Attack]
If Bob signed two messages $y_1 = x_1^d$, $y_2 = x_2^d$
Then the signature for $x_3 = x_1 \cdot x_2$ can be easily ~~found~~ forged as $y_3 = y_1 \cdot y_2$

[Countermeasure]
Use collision resistant hash function before signing messages.

Non-non repudiation

Alice orders 100 shares of stock from Bob
Alice computes MAC using symmetric key
Stock drops, Alice claims she did not order
Can Bob prove that Alice placed the order?
No! Since Bob also knows symmetric key, he could have forged message.
Problem: Bob knows Alice placed the order, but he cannot prove it

Non repudiation 否認,推卸責任

Alice orders 100 shares of stock from Bob
Alice signs order with her private key
Stock drops, Alice claims that she did not order
Can Bob prove that Alice placed the order?
Yes! Only someone with Alice's private key could have signed the order.
This assumes ~~that~~ Alice's private key is not stolen (revocation problem)

KOKUYO

No.

Date

same x

opite y

Euler's theorem $x^{\frac{p-1}{2}}$ calculate the order of p is a prime

Cyclic group: Cyclic

① order = prime ? Every element is a generator

not a prime number :

$|G|$

/ ↳ Only one of them is order 2

2

$p + -p = \infty$

No.

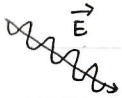
Date

Quantum Cryptography

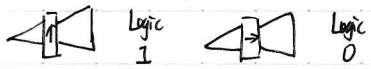
Polarization of photons can be though of as the direction of oscillation of the electric field associated to a light wave.

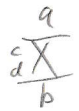
Representation of polarized photons :

- horizontally : →
- vertically : ↑
- diagonally : ↗ and ↖



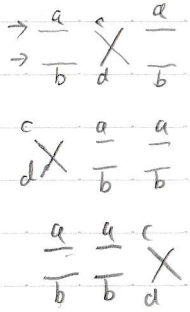
Information can be encoded on each photon by giving it a particular polarization by passing it through a filter.





$\delta^0 = b - a$

$\delta^1 = d - c$



up_delay = a + c + b

bot_delay = b + d + a

Δt = bot_delay − up_delay

= b + d + a − a − c − b

= b − a + d − c + a − b

= $\delta^0 + \delta^1 - \delta^0$

up = c + b + b

bot = d + a + a

Δt = bot − up = d − c + a − b + a − b

= $\delta^1 - \delta^0 - \delta^0$

up = a + a + c

bot = b + b + d

up − bot − up = $\delta^0 + \delta^0 + \delta^1$