

- Terminology
- Classical Ciphers and their Cryptanalysis
- Public and Private Key Encryption
- Steganography
- Cryptology and other computational models

*CSCI283 Fall 2003 Lecture 2*  
*GWU*

# Terminology

## Some terminology

A *sender encrypts* a *plaintext* message to get *ciphertext* which is sent to the *receiver* who *decrypts* it to obtain the plaintext.

$$E(P) = C$$

$$D(C) = P$$

$$D(E(P)) = P;$$

$$D \circ E = I \Rightarrow E \text{ one-to-one}$$

For the application of *secret communication* between two parties, it should not be possible for an eavesdropper to decrypt the message. i.e D should be easy for the (legitimate) receiver, not for anyone else.

## Some terminology - contd.

- *Cipher*: is the *cryptographical algorithm*/mathematical function used to encrypt
- A *restricted* cipher is one whose security depends on keeping the algorithm secret.

Inadequate, because doing so does not provide a systematic way of simulated attack/vulnerability analysis by external experts - which typically improves security .

## Some terminology - contd.

- A *key* is used as a parameter in some ciphers. The security of ciphers that use keys is based on keeping the key(s), and not the cipher, secret.

$$E_{K_1}(P) = C; D_{K_2}(C) = P$$

- *Keyspace*: set of all possible keys.
- *Cryptosystem*: algorithm + all ciphertexts + all plaintexts + all keys

## Some terminology - contd.

- It should be possible for the receiver to *authenticate* (determine with certainty) the sender of a message and to ascertain the *integrity* of the message (i.e. determine that it is really the message that originated from the sender). We see later how to achieve this.
- *Non-repudiation*: It should not be possible for a sender to claim later that she did not send a particular message. We see later how to achieve this.

# Some terminology - Cryptanalysis

- *Cryptanalysis* is an (usually vulnerability) analysis of a cipher.
- Loss of key through means other than cryptanalysis (storage of key in an insecure fashion, for example) is a *compromise*.
- An attempt at cryptanalysis is an *attack*

Kerckhoff's assumption is that *security resides entirely in the key*, i.e. cipher not restricted in any way.

This assumption is useful for external/open vulnerability analysis of different ciphers and for determining their security.

# Cryptanalysis - types of attacks

- *Known-plaintext*:  $m$  and  $c$  known

When a known message/expected message is encrypted, as in file headers in known file-types (jpeg, tiff)

- *Chosen-plaintext*:  $m$  chosen by attacker

Attacker manages to make naïve encrypter encrypt a chosen message

- *Adaptive-chosen-plaintext*:  $m$  chosen by attacker as attack proceeds
- *Chosen-key*:  $k$  chosen

# Cryptanalysis - types of attacks – contd.

- *Ciphertext-only*:  $c$  known

Any eavesdropping/wire tapping/message interception

- *Chosen-ciphertext*:  $c$  chosen by attacker

(as when the attacker has access to the decryption, for example DVD players for watermarking, or decrypting of a message encrypted with a public key)

- *Rubber-hose* (Physical threat to key-holder)

# Cipher Security - security categories

A cipher is *unconditionally secure* if

- the probability distribution (likelihood) on the message is unchanged by knowing the ciphertext
- the ciphertext reveals no information about the plaintext
- no matter how much ciphertext the attacker gets, there is not enough information to obtain the plaintext
- Note: these do not depend on what is computationally feasible

A one-time pad is unconditionally secure.

# Example

Message is a binary string

Key is a binary string of same length

Encryption is XOR of the two strings

If probability of key bit being 1 is  $p$  and each key bit is independent of the other, for what values of  $p$  is the cipher unconditionally secure?

# Cipher Security - security categories

- A *brute-force* attack is one where the attacker tries *all possible keys* and thus breaks a system that is not unconditionally secure.
- A cipher is *computationally secure* if it is computationally infeasible to
  - break the cipher with any amount of ciphertext (in particular, brute force attack is infeasible)
  - obtain any information about the plaintext from the ciphertext
- A cipher is *provably secure* if breaking it is shown to be equivalent to solving a hitherto unsolved problem

# Example

Message is a binary string of length  $n$

Key is a binary string of same length

Encryption is XOR of the two strings

Key is computationally generated using a  $b$ -bit RNG

Is the cipher unconditionally secure?

Is it computationally secure?

# Cipher Security - expense of breaks

- How does the expense of breaking the cipher compare to the value of the encrypted information?
- Expenses can be measured in different ways:
  - data complexity: amount of data required for attack
  - processing complexity: number of instructions required
  - storage requirements: amount of memory neededCan trade one of above for other, etc.
- Expenses change with time as more computational power is available and as new methods for breaking ciphers are developed.

# A good cipher

- Qualities:
  - Cost of encryption, decryption and storage/memory requirements are reasonable and consistent with risk analysis and value of protected data
  - No restrictions on keys and data algorithm works for
  - Ciphertext should not be longer than plaintext
- Development
  - Based on sound mathematics
  - Evaluated by (external, unbiased) experts
  - Stood the test of time

# Some classical ciphers and their cryptanalysis

# Caesar cipher or Shift cipher

also see Lecture 1 notes and Homework 1

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
DEFGHIJKLMNOPQRSTUVWXYZABC

$E(A) = D$ ; Key = 3 (or Key = C)

$E(M) = M \oplus 3 \pmod{26}$

$D(C) = C - 3 \pmod{26}$

$E_{\text{Key}}(\text{symbol}) = \text{symbol} \oplus \text{Key} \pmod{\text{alphabet size}}$

$D_{\text{key}}(\text{symbol}) = \text{symbol} - \text{Key} \pmod{\text{alphabet size}}$

# Shift cipher - cryptanalysis

- Deciphering exactly one symbol in the ciphertext is enough to break the cipher. Serious weakness.
- Can decipher by targeting specific statistical properties of the language of the message – for example, single-lettered words in english can only be “a” or “l”
- Can decipher easily by brute-force, need to try only 26 keys.

# Shift cipher – weaknesses and strengths

- Strengths:
  - Computationally efficient to encrypt and decrypt
  - No storage requirements
  - Ciphertext not longer than plaintext
- Weaknesses:
  - Vulnerable to brute force: a given ciphertext can correspond to only 26 messages (or messages equal to the length of the alphabet)
  - Even more vulnerable when the language has statistical properties, because some keys will be quickly apparent as unlikely/impossible given ciphertext

# Shift cipher - Lessons learnt

- Need cipher that takes more keys than length of language alphabet, so brute force is more difficult
- Key should not be determinable from decrypting a single symbol
- How about something like the Caesar cipher, but with each letter using a different key instead of the same one? (addresses above issues)

# Substitution cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	J	M	Z	U	V	Y	W	R	D	B	U	N	J	O	X	A	E	S	L	P	T	F	G	H	I

A letter goes to another one.

The key is the table: 26! Keys

Brute force could be expensive

Each time a letter appears in the message it encrypts to the same letter in the ciphertext

# Substitution cipher - cryptanalysis

lxr rwq zoazqgr sfuqb bqabq virw gxlkiz uqnb,  
vwqjq ir blsgkn sqfab fggkniay rwq gjicfrq  
rjfabmojsfrioa mijbr fad rwqa rwq gxlkiz oaq.  
wq wfcq aorqd rwfr f sfeoj gjolkqs virw gjicfrq  
uqnb ib rwq bwqqj axslqj om uqnb f biaykq  
xbqj wfb ro brojq fad rjfzu. virw gxlkiz uqnb,  
oakn rvo uqnb fjq aqqdqd gqj xbqj: oaq gxlkiz  
fad oaq gjicfrq. Kqr xb bqq vwfr dimmejqazq  
rwib sfuqb ia rwq axslqj om uqnb aqqdqd.

# Substitution cipher - cryptanalysis

- a 22
- b 24
- c 4
- d 9
- e 2
- f 21
- g 13
- h
- i 20
- j 16
- k 10
- l 8
- m 6
- n 9
- o 15
- p
- q 51
- r 28
- s 9
- t
- u 9
- v 7
- w 16
- x 10
- y 2
- z 8

# Frequency of occurrence

- English (every 1000)

E 127

T 91

A 82

O 75

I 70

N 67

S 63

H 61

R 60

D 43

L 40

C 28

U 28

M 24

W 23

F 22

G 20

Y 20

P 19

B 15

V 10

K 8

J 2

Q 1

X 1

Z 1

- Ciphertext

q 51

r 28

b 24

a 22

f 21

i 20

j 16

w 16

o 15

g 13

x 10

k 10

d 9

u 9

n 9

s 9

l 8

z 8

v 7

m 6

c 4

e 2

y 2

h 0

t 0

p 0

$$q = E$$

l<sub>x</sub>r r<sub>w</sub>E z<sub>o</sub>a<sub>z</sub>E<sub>g</sub>r s<sub>f</sub>u<sub>E</sub>b b<sub>E</sub>a<sub>b</sub>E v<sub>i</sub>r<sub>w</sub> g<sub>x</sub>l<sub>k</sub>i<sub>z</sub> u<sub>E</sub>n<sub>b</sub>, v<sub>w</sub>E<sub>j</sub>E i<sub>r</sub>  
b<sub>l</sub>s<sub>g</sub>k<sub>n</sub> s<sub>E</sub>f<sub>a</sub>b f<sub>g</sub>g<sub>k</sub>n<sub>i</sub>a<sub>y</sub> r<sub>w</sub>E g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E r<sub>j</sub>f<sub>a</sub>b<sub>m</sub>o<sub>j</sub>s<sub>f</sub>r<sub>i</sub>o<sub>a</sub> m<sub>i</sub>j<sub>b</sub>r  
f<sub>a</sub>d r<sub>w</sub>E<sub>a</sub> r<sub>w</sub>E g<sub>x</sub>l<sub>k</sub>i<sub>z</sub> o<sub>a</sub>E. v<sub>E</sub> w<sub>f</sub>c<sub>E</sub> a<sub>o</sub>r<sub>E</sub>d r<sub>w</sub>f<sub>r</sub> f s<sub>f</sub>e<sub>o</sub>j  
g<sub>j</sub>o<sub>l</sub>k<sub>E</sub>s v<sub>i</sub>r<sub>w</sub> g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E u<sub>E</sub>n<sub>b</sub> i<sub>b</sub> r<sub>w</sub>E b<sub>w</sub>E<sub>E</sub>j a<sub>x</sub>s<sub>l</sub>E<sub>j</sub> o<sub>m</sub>  
u<sub>E</sub>n<sub>b</sub> f b<sub>i</sub>a<sub>y</sub>k<sub>E</sub> x<sub>b</sub>E<sub>j</sub> w<sub>f</sub>b r<sub>o</sub> b<sub>r</sub>o<sub>j</sub>E f<sub>a</sub>d r<sub>j</sub>f<sub>z</sub>u. v<sub>i</sub>r<sub>w</sub> g<sub>x</sub>l<sub>k</sub>i<sub>z</sub>  
u<sub>E</sub>n<sub>b</sub> o<sub>a</sub>k<sub>n</sub> r<sub>v</sub>o u<sub>E</sub>n<sub>b</sub> f<sub>j</sub>E a<sub>E</sub>E<sub>d</sub>E<sub>d</sub> g<sub>E</sub>j x<sub>b</sub>E<sub>j</sub>: o<sub>a</sub>E g<sub>x</sub>l<sub>k</sub>i<sub>z</sub>  
f<sub>a</sub>d o<sub>a</sub>E g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E. k<sub>E</sub>r x<sub>b</sub> b<sub>E</sub>E v<sub>w</sub>f<sub>r</sub> d<sub>i</sub>m<sub>m</sub>e<sub>j</sub>E<sub>a</sub>z<sub>E</sub> r<sub>w</sub>i<sub>b</sub>  
s<sub>f</sub>u<sub>E</sub>b i<sub>a</sub> r<sub>w</sub>E a<sub>x</sub>s<sub>l</sub>E<sub>j</sub> o<sub>m</sub> u<sub>E</sub>n<sub>b</sub> a<sub>E</sub>E<sub>d</sub>E<sub>d</sub>.

# Digram/Trigram occurrence

- Digram

TH	TO	IT
<i>HE</i>	NT	AR
IN	HA	<i>TE</i>
<i>ER</i>	ND	<i>SE</i>
AN	OU	HI
<i>RE</i>	<i>EA</i>	OF
<i>ED</i>	NG	
ON	AS	
<i>ES</i>	OR	
ST	TI	
<i>EN</i>	IS	
AT	<i>ET</i>	

- Trigram

THE  
ING  
AND  
HER  
*ERE*  
*ENT*  
THA  
NTH  
WAS  
*ETH*  
FOR  
DTH

$$q = E$$

l<sub>x</sub>r r<sub>w</sub>E zoa<sub>z</sub>E<sub>g</sub>r s<sub>f</sub>u<sub>E</sub>b<sub>b</sub>E<sub>a</sub>b<sub>b</sub>E virw g<sub>x</sub>l<sub>k</sub>iz u<sub>E</sub>n<sub>b</sub> v<sub>w</sub>E<sub>j</sub>E ir b<sub>l</sub>s<sub>g</sub>k<sub>n</sub> s<sub>E</sub>f<sub>a</sub>b  
 f<sub>g</sub>g<sub>k</sub>n<sub>i</sub>a<sub>y</sub> r<sub>w</sub>E g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E r<sub>j</sub>f<sub>a</sub>b<sub>m</sub>o<sub>j</sub>s<sub>f</sub>r<sub>i</sub>o<sub>a</sub> m<sub>i</sub>j<sub>b</sub>r f<sub>a</sub>d r<sub>w</sub>E<sub>a</sub> r<sub>w</sub>E g<sub>x</sub>l<sub>k</sub>iz o<sub>a</sub>E.  
 v<sub>E</sub> w<sub>f</sub>c<sub>E</sub> a<sub>o</sub>r<sub>E</sub>d<sub>d</sub> r<sub>w</sub>f<sub>r</sub> f s<sub>f</sub>e<sub>o</sub>j g<sub>j</sub>o<sub>l</sub>k<sub>E</sub>s virw g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E u<sub>E</sub>n<sub>b</sub> i<sub>b</sub> r<sub>w</sub>E b<sub>w</sub>E<sub>E</sub>j  
 a<sub>x</sub>s<sub>l</sub>E<sub>j</sub> o<sub>m</sub> u<sub>E</sub>n<sub>b</sub> f b<sub>i</sub>a<sub>y</sub>k<sub>E</sub> x<sub>b</sub>E<sub>j</sub> w<sub>f</sub>b r<sub>o</sub> b<sub>r</sub>o<sub>j</sub>E f<sub>a</sub>d r<sub>j</sub>f<sub>z</sub>u. Virw g<sub>x</sub>l<sub>k</sub>iz  
 u<sub>E</sub>n<sub>b</sub>, o<sub>a</sub>k<sub>n</sub> r<sub>v</sub>o u<sub>E</sub>n<sub>b</sub> f<sub>j</sub>E a<sub>E</sub>E<sub>d</sub>E<sub>d</sub> g<sub>E</sub>j x<sub>b</sub>E<sub>j</sub>: o<sub>a</sub>E g<sub>x</sub>l<sub>k</sub>iz f<sub>a</sub>d o<sub>a</sub>E  
 g<sub>j</sub>i<sub>c</sub>f<sub>r</sub>E. k<sub>E</sub>r x<sub>b</sub> b<sub>E</sub>E v<sub>w</sub>f<sub>r</sub> d<sub>i</sub>m<sub>m</sub>e<sub>j</sub>E<sub>a</sub>z<sub>E</sub> r<sub>w</sub>i<sub>b</sub> s<sub>f</sub>u<sub>E</sub>b<sub>b</sub> i<sub>a</sub> r<sub>w</sub>E a<sub>x</sub>s<sub>l</sub>E<sub>j</sub> o<sub>m</sub>  
 u<sub>E</sub>n<sub>b</sub> a<sub>E</sub>E<sub>d</sub>E<sub>d</sub>.

En 6 Ej 6 Ed 5 Ea 2 Eb 2 Er 1 Ef 1 Es 1 Eg 1

ER ED ES EN EA ET

uE 8 wE 8 aE 5 bE 5 rE 4 kE 3 jE 3 dE 2 zE 2 gE 1 vE 1 cE IE 1 sE 1

HE RE TE SE

TAOI NSHRD

r b a f i j w o g x k d

j=R; d = D; b or a = S; w = H;

$$q = E; j=R; w=H; d=D$$

lxr *rHE* zoazEgr sfuEb bEabE virH gxlkiz uEnb *vHERE*  
 ir blsgkn sEfab fggkniay *rHE* gRicfrE rRfabmoRsfrioa  
 miRbr fad *rHEa rHE* gxlkiz oaE. vE HfcE aorEd rHfr f  
 sfeoR gRolkes virH gjicfrE uEnb ib *rHE* bHEER  
 axslER om uEnb f biaykE xbER Hfb ro broRE fad  
 rRfzu. HirH gxlkiz uEnb, oakn rvo uEnb fRE aEEEdEd  
 gER xbER: oaE gxlkiz fad oaE gRicfrE. kEr xb bEE  
 vHfr dimmeREazE rHib sfuEb ia *rHE* axslER om  
 uEnb aEEEdEd.

TAOI NS  
 r b af i og  
*r = T*

q = E; j=R; w=H; r=T; d=D

ixT THE zONzEgr MAuES SENSE WITH gxlklz uEnS  
WHERE IT SIMgkn MEANS AggknINy THE gRIcATE  
TRANSFORMATION FIRST AND THEN THE gxlklz  
ONE. WE HAVE NOTED THAT A MAJOR PROIKEM  
WITH PRIVATE uEnS IS THE SHEER NxMIER OF  
uEnS A SlaykE xSER HAS TO STORE AND TRAzU.  
WITH gxlklz uEnS, ONkn TWO uEnS ARE NEEDED  
gER xSER: ONE Pxiklz AND ONE PRIVATE. kET xS  
SEE WHAT DimmeRENzE THIS sAuESIN THE  
NxBIER OF uEnS NEEDED.

O NS

b a og

v=W; i=l; f=A; b=S; o=O; m=F; a=N; s=M; c=V; g=P; e=J;

# Substitution cipher - cryptanalysis

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
f l z d q m y w i e u k s a o g t j b r x c v h n p

BUT THE CONCEPT MAKES SENSE WITH PUBLIC KEYS WHERE IT SIMPLY MEANS APPLYING THE PRIVATE TRANSFORMATION FIRST AND THEN THE PUBLIC ONE. WE HAVE NOTED THAT A MAJOR PROBLEM WITH PRIVATE KEYS IS THE SHEER NUMBER OF KEYS A SINGLE USER HAS TO STORE AND TRACK. WITH PUBLIC KEYS ONLY TWO KEYS ARE NEEDED PER USER ONE PUBLIC AND ONE PRIVATE. LET US SEE WHAT DIFFERENCE THIS MAKES IN THE NUMBER OF KEYS NEEDED.

# Substitution cipher – cryptanalysis algorithm

- Look for “a”/”I”
- Compute frequency of single letters; compare to that of English
- Compute frequency of digrams, compare to that of English
- Compute frequency of trigrams, compare to that of English
- Etc.

# Substitution cipher – strengths and weaknesses

- Strengths:
  - Not vulnerable to brute force attacks
  - Encryption and decryption requires low computational overhead, though more than Shift cipher
  - Ciphertext not longer than plaintext
- Weaknesses:
  - Vulnerable to statistical attack if language/message has statistical structure
  - Requires storage of key table

# Substitution cipher – lessons learnt

- In spite of  $26!$  possible keys, can break, because of structure of message
- Can we make message without statistical structure?
- Examples?

Images in well-compressed form. What about zip files?

Can we use a method where the symbol does not get mapped to the same symbol each time?

# One-time pad

- Key as long as message
- A symbol of the key is added to the corresponding symbol of the message
- If the key symbols are random, and not related to one another, this is perfectly secure – i.e. not vulnerable to a ciphertext only attack.
- However, if key symbols are related, it can be insecure
- Main disadvantage: difficult to ensure both sender and receiver have an identical, random key, and are synchronized

# Non-computational random number sources

- From students:
    - Physically measurable quantities such as the energy emitted from a light bulb
- Cannot be used as a one-time pad key because difficult for sender to communicate it to a receiver
- (Really nice suggestion) Publicly available, physically measurable quantities such as the temperature in London as available from a specific source

# Characteristics of a good cipher: Confusion and Diffusion

- Confusion: unknown symbols
- Diffusion: changes in a single plaintext symbol should affect many ciphertext symbols

So far we have studied only confusion; i.e. *stream ciphers*:

Single plaintext symbol encrypted independent of others,  
hence fast

Low error propagation (diffusion), hence also vulnerable to  
addition of spurious messages

# Block ciphers

Symbols are encrypted in blocks, where the ciphertext corresponding to a plaintext symbol depends on the entire block:

- Slower than stream

- Higher error propagation (diffusion), hence less vulnerable to addition of spurious messages

Not easily vulnerable to statistical attacks

# Permutation/scrambling/transposition ciphers

THIS IS VERY EASY TO BREAK

T	V	Y	A		
H	I	E	E	B	K
I	S	R	A	T	R
S	Y	S	O	E	

t v y ahiee bkisratr s ysoe

Cryptanalysis: try the few possible column lengths

# Product (combination) of ciphers

- Encryption using a cipher and a key followed by another encryption by another cipher and another key.
- Example: previous scrambled message encrypted with substitution cipher

t v y ahiee bkisratr s ysoe

r c n fwiqq luibjfrj b nboq

cryptanalysis? Frequency analysis, descrambling

# Standard private key protocols

## Some terminology - contd.

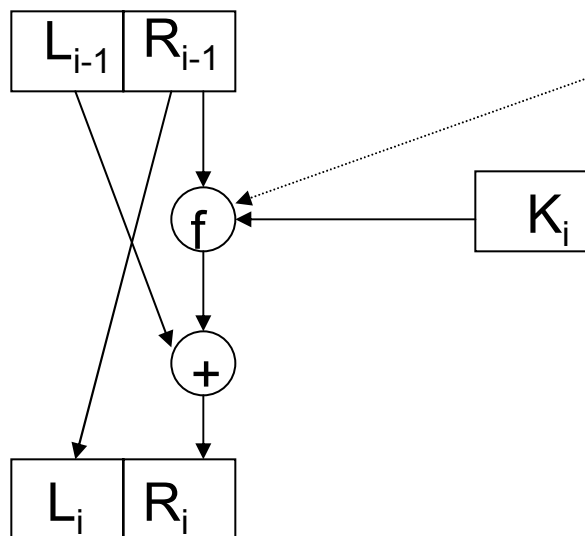
- *Symmetric/Private key algorithms:*
  - decryption key can be calculated from encryption key and vice versamost often the two keys are identical.  
Security depends on keeping the keys secret.

$$K_{\text{encryption}} = K_{\text{decryption}} \text{ or } f(K_{\text{encryption}}) = K_{\text{decryption}}$$

# DES (Data Encryption Standard)

NBS (now NIST), '77; reviewed every 5 years

- DES acts on 64-bit plaintext  $x$  (16 iterations)
  - $x_0 = P(x) = L_0R_0$
  - $L_i = R_{i-1}; R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
  - $K_i$  permuted subset of 48 bits from 56-bit key,  $K$



$f$  is substitution and scrambling; i.e. confusion and diffusion; of  $R_{i-1} + K_i$

## Need and search for another standard symmetric cipher

- 64 bit DES key too small; specially in the future
- NIST announced an open request for encryption algorithms (in contrast, DES design decisions were hush-hush)
- Responses from all over the world
- Methods openly described and thoroughly tested
- Chosen on the basis of security, cost, implementation complexity
- Won by 2 Belgian cryptographers (Daemen and Rijmen), working independently of any company (DES was based on an IBM cipher, *Lucifer*)

# AES (Advanced Encryption Standard) NIST, 2001

- Key length 128 bits (10 rounds) [can also be 192 bits (12 round), and 256 bits (14 round)]

Message =

Byte0 Byte1 Byte2 Byte3 Byte4 Byte5 Byte5 Byte7 Byte8 Byte9 ByteA ByteB ByteC ByteD ByteE ByteF

STATE = Byte0 Byte1 Byte2 Byte3  
Byte4 Byte5 Byte5 Byte7  
Byte8 Byte9 ByteA ByteB  
ByteC ByteD ByteE ByteF

# AES (Advanced Encryption Standard) NIST, 2001

- Key length 128 bits (10 rounds), 192 bits (12 round), and 256 bits (14 round)

## Algorithm Sketch

- $STATE \leftarrow \text{plaintext XOR ROUNDKEY}$
- For all except last round
  - $STATE \leftarrow \text{Substitution Cipher (STATE)}$
  - $STATE \leftarrow \text{Permutation (STATE)}$
  - $STATE \leftarrow \text{Matrix Multiplication (STATE)}$
  - $STATE \leftarrow STATE \text{ XOR ROUNDKEY}$
- Last round is as above, leaves out Matrix Multiplication
- Ciphertext is STATE

# Public key encryption

## Some terminology - contd.

- *Asymmetric/Public key algorithms:*

- the keys are distinct
- Message encrypted by one key can be decrypted by other
- calculation of one key from another is (computationally) “hard” or infeasible.
- One of the keys,  $K_{\text{public}}$  is usually made public.

Security depends entirely on the “hardness” of determining the non-public/private key.

$$K_{\text{public}} \neq K_{\text{private}} \text{ and}$$

If  $f(K_{\text{public}}) = K_{\text{private}}$ ,  $f$  computationally infeasible

# Asymmetric/public key use

All messages that can be decrypted by  $k_{public}$  can be assumed to have originated from the owner of public key  $k_{public}$  (and encrypted with  $k_{private}$ )

this use is to avoid fraud/provides authentication

All messages that are encrypted with  $k_{public}$  can only be decrypted with  $k_{private}$

this use is for secret transmission.

# RSA (after inventors: Rivest, Shamir, Adleman, winners of the Turing Award for 2002)

## *Choice of keys*

- Choose large primes  $p$  and  $q$  - about 100 digits each
- Define  $n = pq$
- Find  $e$  relatively prime to  $(p-1)(q-1)$   
(for example,  $e$  a prime larger than  $p-1$  or  $q-1$ )
- Find  $d$  such that

$$x^{ed} = x \text{ mod } n$$

(It is known how to do this if  $p$  and  $q$  are known. It is not known how to do this otherwise)

$(e, n)$  is encryption key,  $(d, n)$  is decryption key

# RSA: Use

Encryption

$$c = m^e \bmod n$$

Decryption

$$c^d \bmod n = m^{ed} \bmod n = m \bmod n$$

Given one key  $e$ , determining the other requires solving for  $d$ :

$$x^{ed} = x \bmod n \quad \forall x$$

# Example

$$p = 5, q = 11; n = 55$$

$e$  relatively prime to  $4 \times 10 = 40$

Thus fewer than 55 values of  $e$

Choose a prime larger than 4 and 10:  $e = 13$

$d = ?$  (using math and knowing  $p$  and  $q$ , can determine it is 37)

Fewer than 55 values of  $d$

One key is (13, 55); the other is (37, 55)

# Example - contd

Using (13, 55) as public, (37, 55) is private

Encrypt:

2

$$2^{13} \bmod 55 = 8192 \bmod 55 = 52$$

Know public key is 13, need find 13<sup>th</sup> root of 52 mod 55

Exactly one number is the 13<sup>th</sup> root of 8192 mod 55

Hence not perfectly secure

However, don't know how to calculate the 13<sup>th</sup> root without knowing the private key – method could be computationally secure

Decrypt:

$$52^{37} = 2 \bmod 55$$

# Example – contd.

In what way is RSA different from one-time pad?

*Given the ciphertext:*

One Time Pad

Cannot make the one-time pad key public

Without knowing the key of the one-time pad, message could be anything.

RSA

Without knowing the private key, the message is exactly one value which we do not know a feasible way to compute using current computers.

## Example – contd.

*Given the ciphertext:*

One Time Pad

Given a possible key we cannot tell if it were the correct one

RSA

Given a possible private key we could check if it were the correct one and determine the message if it were

# Example – contd.

*Given the ciphertext*

One Time Pad

Given a number, without knowing key of one-time pad, cannot say whether it is the message or not

Brute force attack impossible

RSA

Given a number, without knowing private key, can tell if it is the message in RSA or not

Need a large enough number of possible messages to make brute force infeasible using RSA

# Security of RSA for current model of computing

- Mathematical soundness

There are no published computationally feasible methods, using models of current computers, for

- finding one key from the other without factoring  $n$  (i.e. finding  $p$  and  $q$ )
- finding  $p$  and  $q$  from  $n$ , i.e. for factoring

- External analysis and test of time

RSA was made public in 1978. Serious security glitches have not yet been found

# Not a security guarantee

If a feasible method for factoring is found, RSA can be easily broken.

No proof that a feasible method for factoring cannot be found

Attempts to break RSA result in progress in computational number theory

For example: a polynomial-time algorithm for determining if a number is prime (primality testing) had not been published till late 2001/early 2002 (due to Manindar Agarwal and his undergrads, IIT-Kanpur)

# Steganography

- Hiding a message so that an attacker cannot even detect the existence of the message
- Example: watermarking; application: anti-piracy
  - Hide a message in media
  - So that it is imperceptible
  - So that any operation that retains the perceptual quality of the media cannot remove the message
  - Not known to be solvable (all current methods are breakable)
  - Requires an understanding of human perception/AI

# Another security model: Captcha

<http://www.captcha.net>

Completely Automated Public Turing test to tell  
Computers from Humans

A Turing test is a test a computer needs to pass to be  
considered “intelligent”

Captcha tests are generated, administered and graded  
by a computer but cannot be passed by one.

# Uses of Captcha

- Captcha-based security techniques depend on the hardness of problems in AI. Attempts to break Captcha-based security results in progress in AI
- E.g.: Yahoo and other online portals use it to tell humans from bots
- Can be used to prevent spam, automated form-filling, fake electronic voting

# Another computational model: Quantum Computing

- Peter Shor was the first to demonstrated a feasible way to factor using quantum computational models.
- If quantum computers can be built, RSA will be easily broken.

# References

- Bruce Schneier, *Applied Cryptography*
- Douglas Stinson, *Cryptography Theory and Practice*