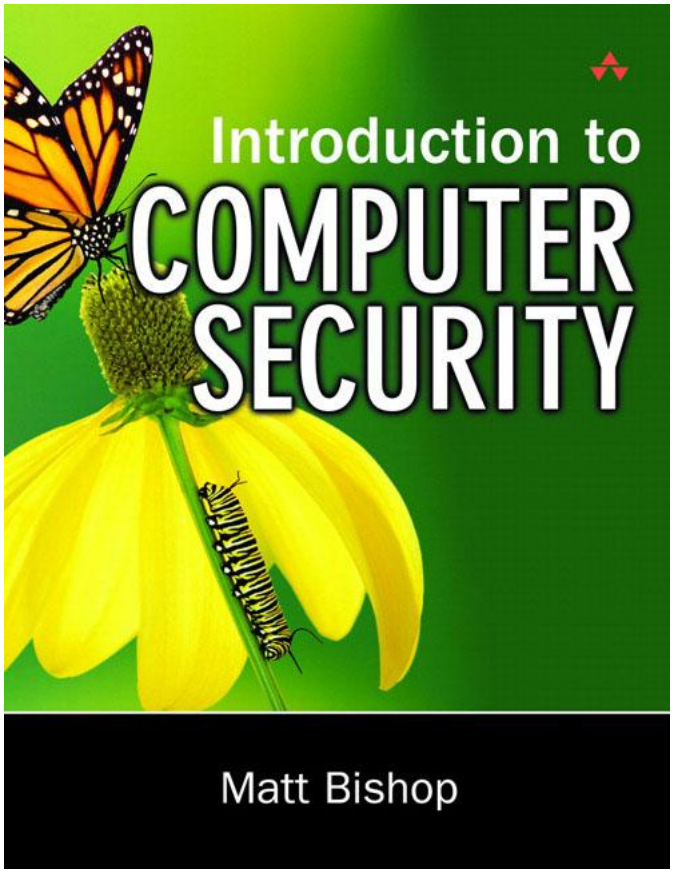
These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

Cryptography I

Basic Cryptography - Ch 8



Adnan Gutub
gutub@kfupm.edu.sa

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia

Chapter 8: Basic Cryptography

Classical Cryptography

Public Key Cryptography

Cryptographic Checksums

Overview

What is Cryptography?

Classical Cryptography

- Cæsar cipher
- Vigènere cipher
- Block (Hill) ---------- not in book
- DES
- AES ---------- not in book

Public Key Cryptography

- Diffie-Hellman
- RSA
- ECC ---------- not in book

Cryptographic Checksums

- HMAC

Cryptography

- Word "crypto graphy" comes from two Greek words meaning "secret writing" = art & science of covering meaning
- Privacy and security needed while communicating over insecure media (internet)
- In past, Cryptography was heavily used for military to keep sensitive information secret from enemies (e.g. Caesar cipher)
- Nowadays, with the technologic progress as our dependency on electronic systems has increased we need more sophisticated techniques.
- Cryptography provides *most* of the methods and techniques for a *secure communication*

Terminology

Cryptology

- All-inclusive term used for the study of secure communication over non-secure channels and related problems.

Cryptography

- The process of designing systems to realize secure communications over non-secure channels

Cryptoanalysis

- The discipline of breaking the cryptographic systems

Coding Theory

- Deals with representing the information using codes. It covers: compression, secrecy, and error-correction. Recently, it is predominantly associated with error-correcting codes which ensures the correct transmissions over noisy-channels.

Cryptography

depends on:

- mathematics & usage of digital systems

Inter-disciplinary study of three fields:

- Mathematics
- Computer Science
- Electrical Engineering

} Computer Engineer

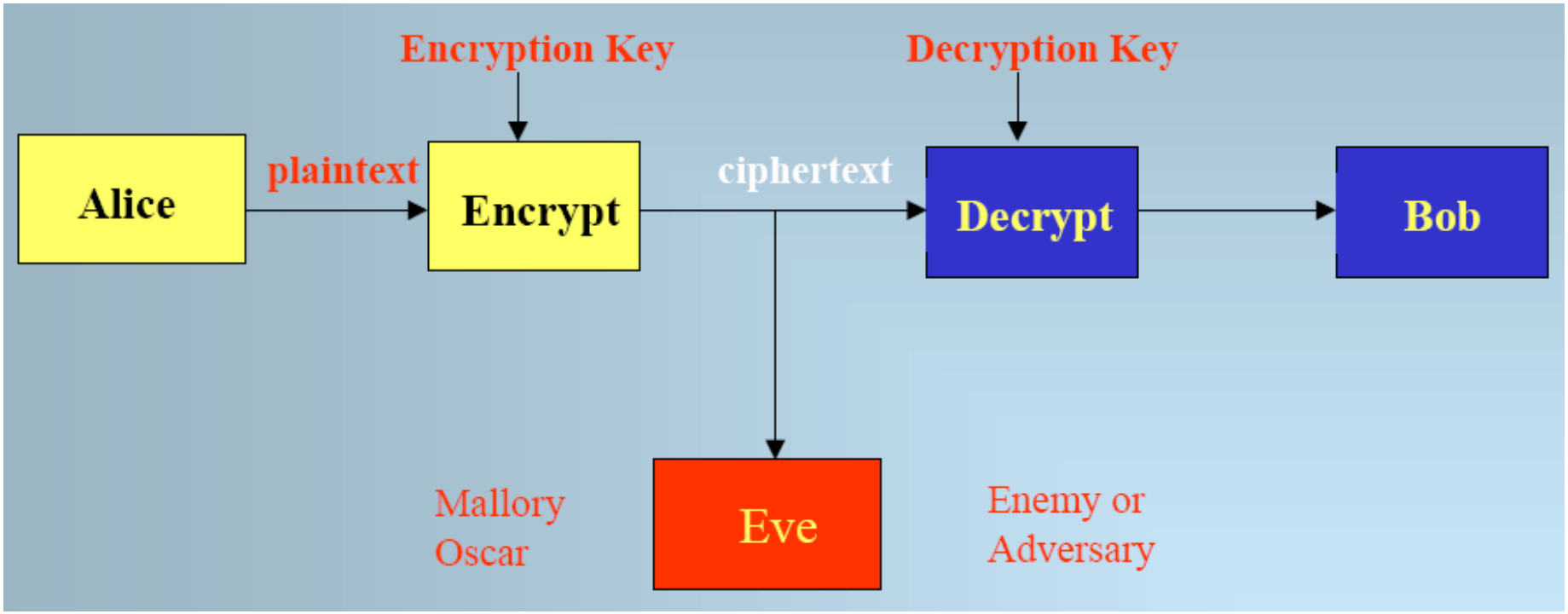
The importance of crypto-analysis

- Without having a complete understanding of crypto-analysis (or crypto-analytic techniques) it is impossible to design *good* (secure, unbreakable) cryptographic systems

Other Disciplines

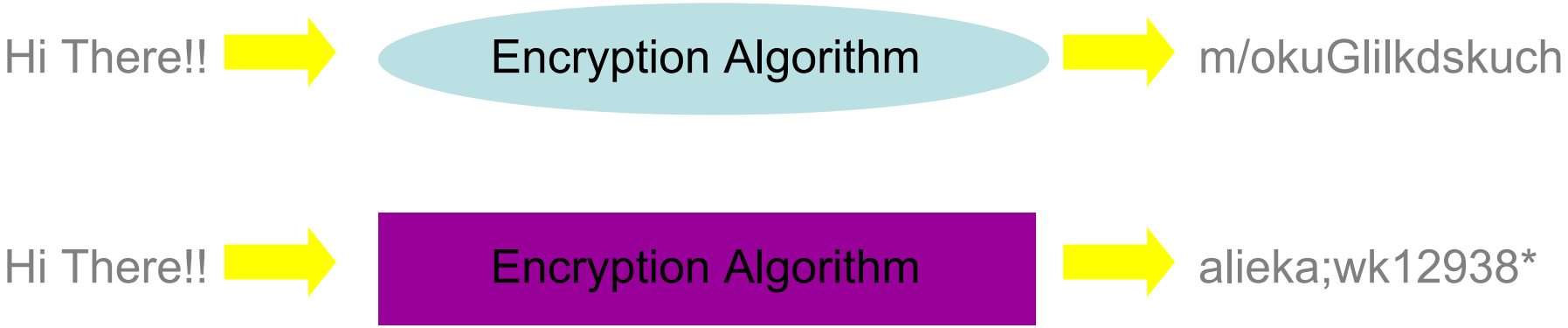
- It makes use of other disciplines such as error-correcting codes, compression

Secure Communication



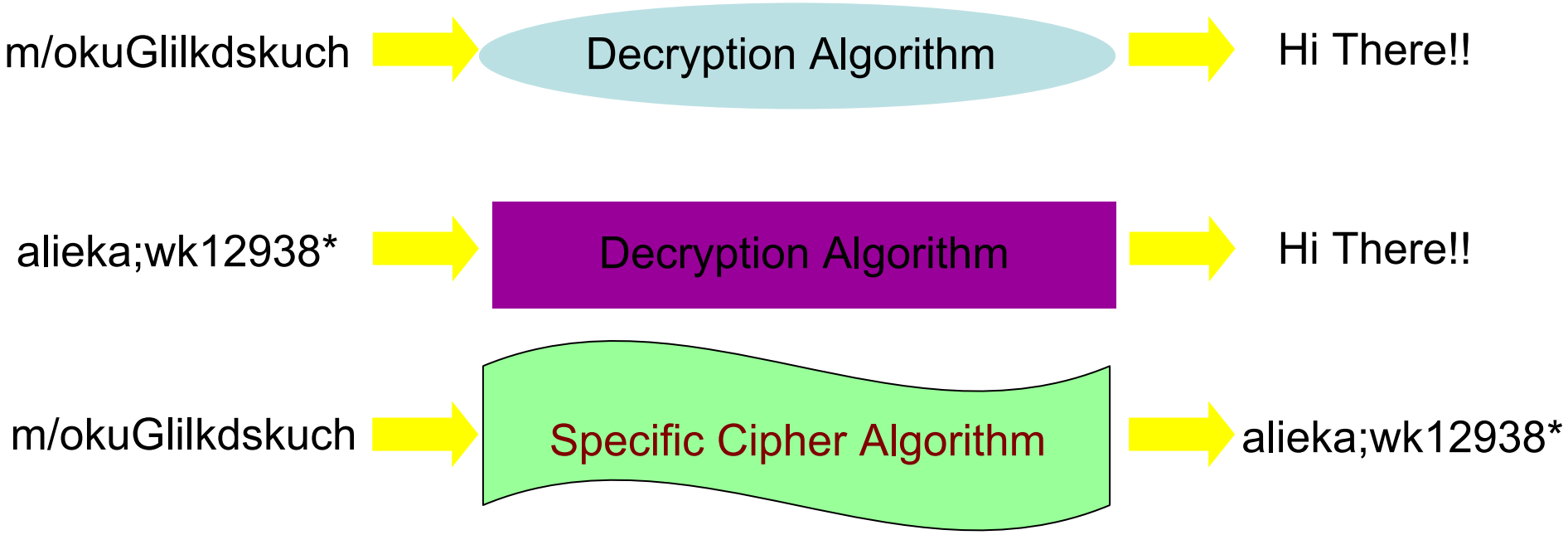
Encryption

Convert normal, readable data into obscured, unreadable data



Decryption

Convert obscured, unreadable data into normal, readable data

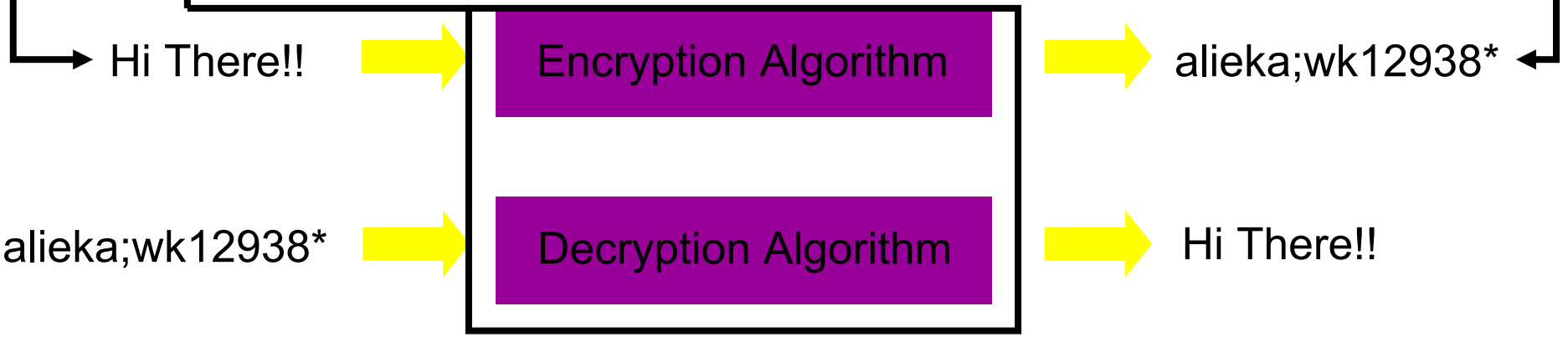


Terminology

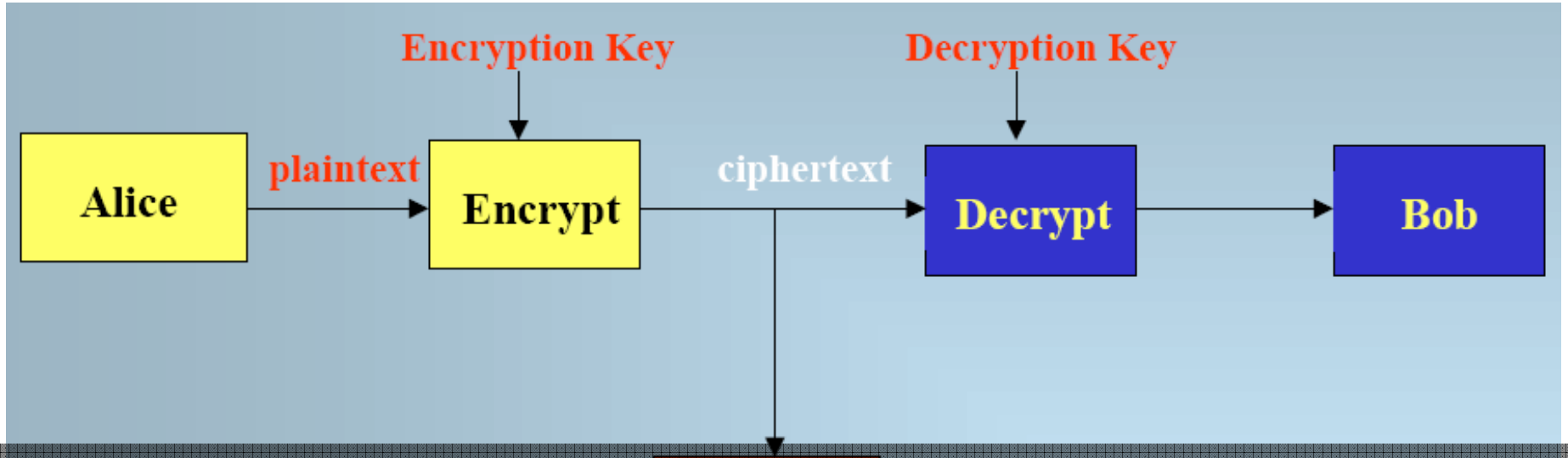
plaintext - clear readable text

ciphertext - unreadable text

cipher - algorithm(s) for encryption and decryption



Eve's Goals
Secure Communication



Mallory

- Read the message
- Modify the contents of the message
- Figure out Alice's key
  - Bob will think Alice sent the altered message
  - Read all messages encrypted with this key
- Impersonate Alice
  - communicate with Bob who thinks he is communicating with Alice

Attack Means: Cryptanalysis

Opponent whose goal is to break cryptosystem is the *adversary*

- Assume adversary knows algorithm used, but not key

Ciphertext only

- Alice has only a copy of ciphertext

Known Plaintext

- Eve *has* a copy of ciphertext and the corresponding plaintext and tries to figure out the key

Chosen Plaintext

- Eve *can* have a ciphertext corresponding to a sample plaintext which she believes is useful to figure the key

Chosen Ciphertext

- Eve can have a plaintext corresponding to a sample ciphertext which she believes is useful to figure the key

Basis for Attacks & Cryptanalysis

Mathematical attacks

- Based on analysis of underlying mathematics

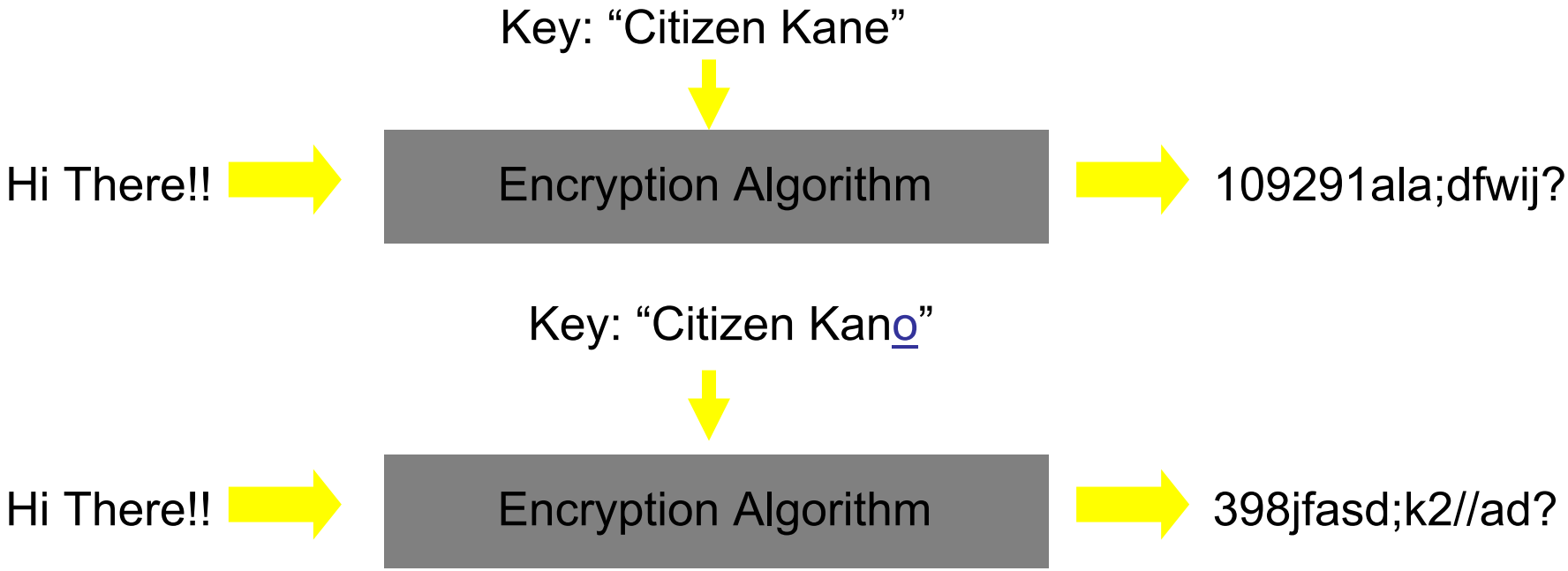
Statistical attacks

- Make assumptions about the distribution of letters, pairs of letters (digrams), triplets of letters (trigrams), *etc.*
  - Called *models of the language*
- Examine ciphertext, correlate properties with the assumptions.

Terminology: Key

Key -- a secret piece of information that controls how the encryption algorithm works

Different keys produce different encrypted results



Terminology

Security through obscurity

- Don't publish some details of your algorithm... assuming people won't figure it out
- Like hiding the key under the doormat

Once your flaw/algorithm is leaked, you're screwed

Kerckhkoffs Principle

Complete knowledge of the Algorithm

- While assessing the strength of a cryptosystem, one should always assume that the enemy knows the cryptographic algorithm used

The security of the system, therefore, should be based on

- the quality (strength) of the algorithm but not its obscurity or darkness
- the key space (or key length)

Computer Era

Moore's law and its implications

Keys breakable by brute force

Modern Ciphers

- Bigger and bigger keys
- More and more complicated algorithms
- Based on hardcore applied mathematics... and the difficulty of factoring large numbers

Cryptosystem - formally

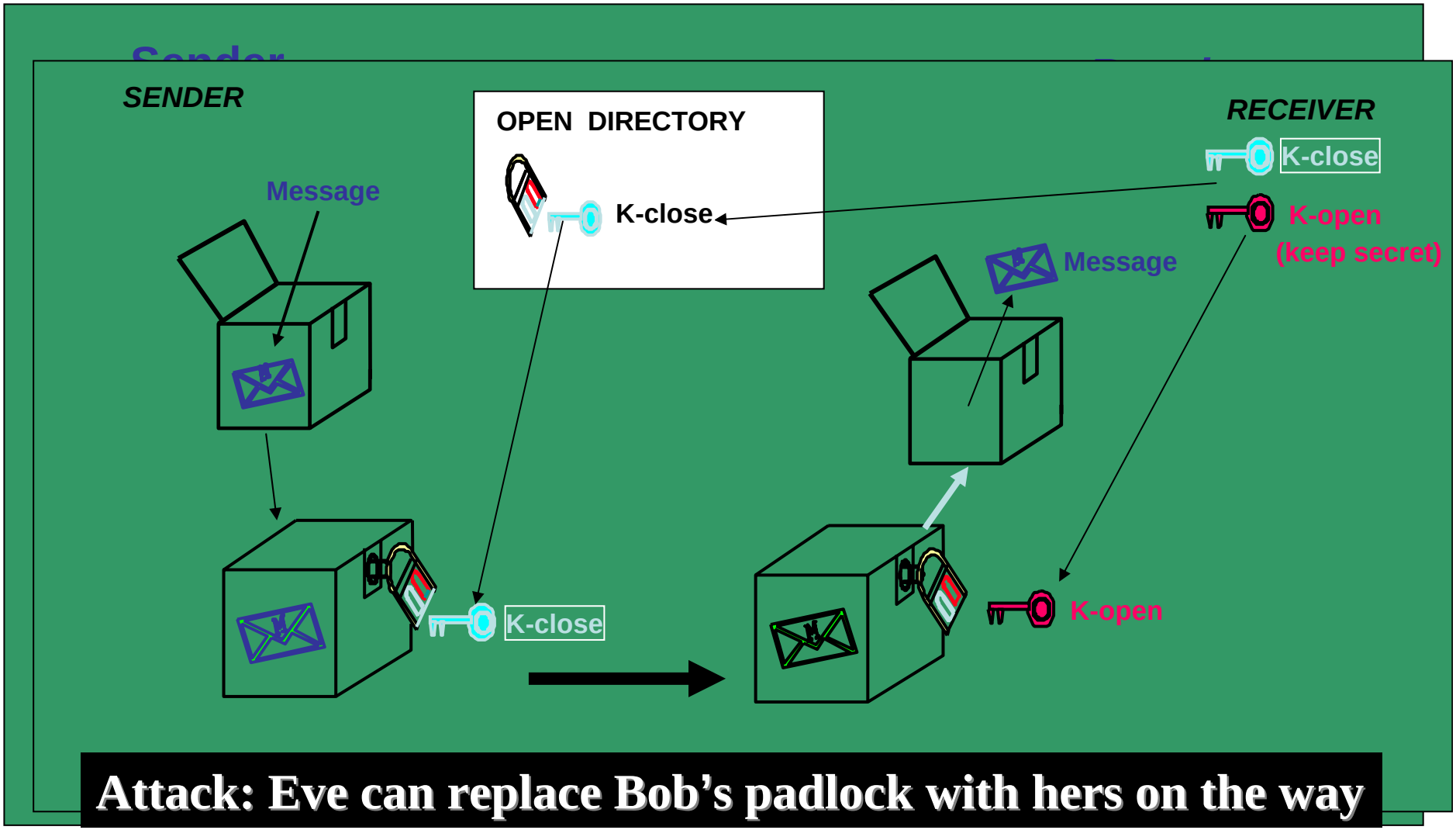
Cryptosystem is a quintuple $(\mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{K}, C)$

- $\mathcal{M}$ set of plaintexts
- $\mathcal{K}$ set of keys
- C set of ciphertexts
- $\mathcal{E}$ set of encryption functions e: $\mathcal{M} \times \mathcal{K} \rightarrow C$
- $\mathcal{D}$ set of decryption functions d: $C \times \mathcal{K} \rightarrow \mathcal{M}$

Example: Cæsar cipher

- $\mathcal{M}$ = { sequences of letters }
- $\mathcal{K}$ = { i | i is an integer and $0 \leq i \leq 25$ }
- $\mathcal{E}$ = { E_k | $k \in \mathcal{K}$ and for all letters m, $E_k(m) = (m + k) \bmod 26$ }
- $\mathcal{D}$ = { D_k | $k \in \mathcal{K}$ and for all letters c, $D_k(c) = (26 + c - k) \bmod 26$ }
- $C = \mathcal{M}$

Terminology



Attack: Eve can replace Bob's padlock with hers on the way

Back to Classical Cryptography

Sender, receiver share common key

- Keys may be the same, or trivial to derive from one another
- Sometimes called *symmetric cryptography,* single-key, shared-key, etc.

Two basic types

- Substitution ciphers
- Transposition ciphers
- Combinations are called *product ciphers*

Classic Cryptography

Substitution (Caesar)

Transposition

Enigma Machine

Vigenere

Block (Hill)

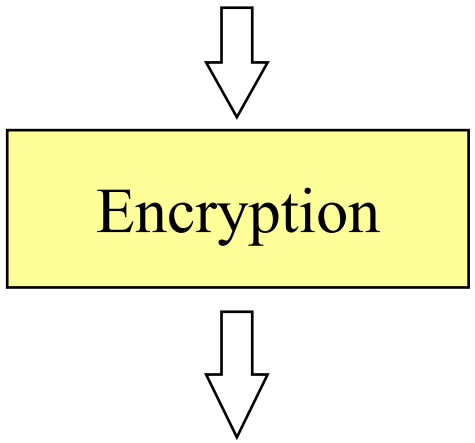
Vernam (one time pad)

DES

AES

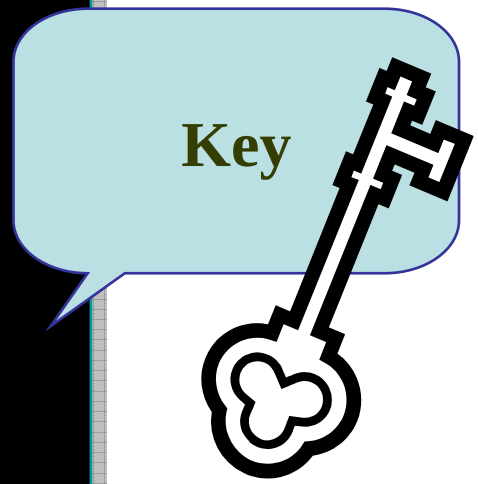
Substitution

NUCLEAR



13 20 2 11 4 0 17

A	=>	0
B	=>	1
C	=>	2
D	=>	3
E	=>	4
.		.
.		.
.		.
X	=>	23
Y	=>	24
Z	=>	25

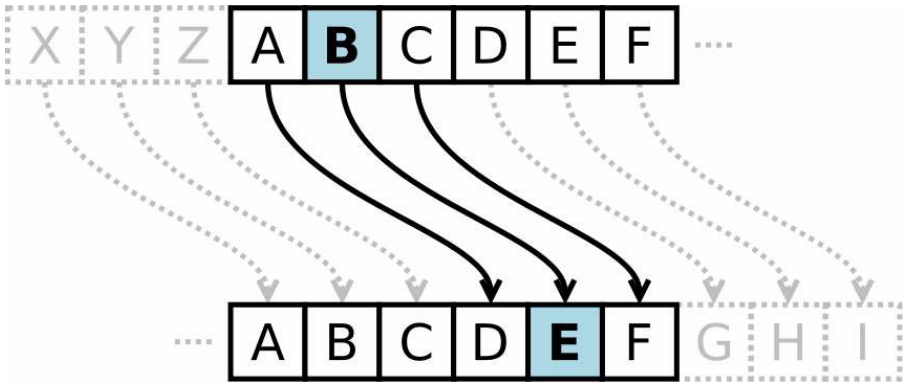


Substitution Ciphers

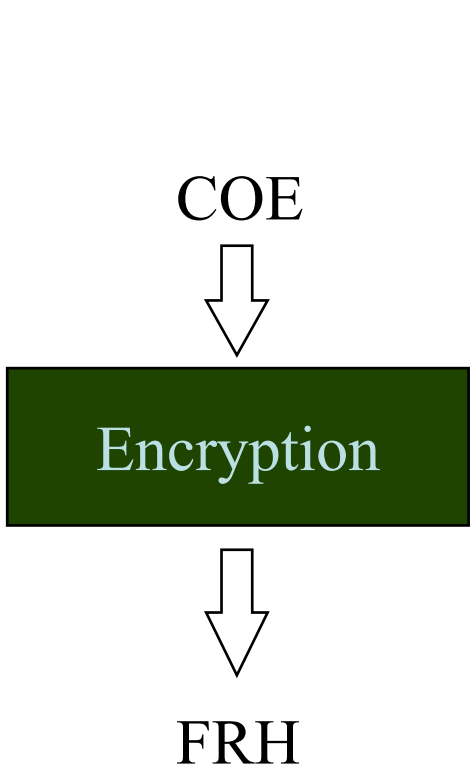
Change characters in plaintext to produce ciphertext

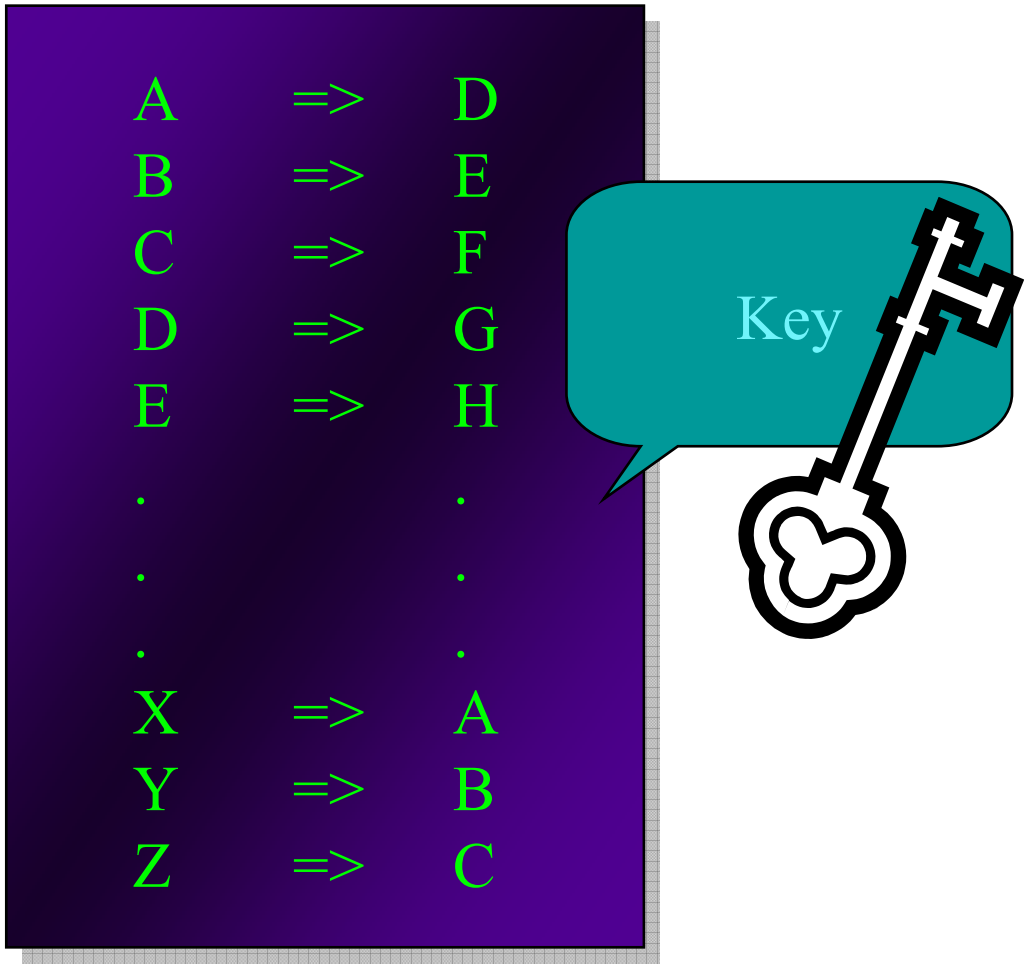
Example (Cæsar cipher)

- Plaintext is `HELLO WORLD`
- Change each letter to the third letter following it
- (A goes to D, X goes to A, Y to B, Z to C)
  - Key is 3, usually written as letter 'D'
- Ciphertext is `KHOOR ZRUOG`



Substitution (Caesar)





Caesar Cipher: Shift by 3

PLAINTEXT	a	b	c	d	e	f	g	h	i	j	k	l	m
CIPHERTEXT	D	E	F	G	H	I	J	K	L	M	N	O	P

PLAINTEXT	n	o	p	q	r	s	t	u	v	w	x	y	z
CIPHERTEXT	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Hello There → khoor wkhuh

Advance Substitution (Random)

Key

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

NUCLEAR → **Encryption** → ???????

Problem

Monoalphabetic

- Same letter of plaintext always produces same letter of ciphertext

Even though there are 26!

- possible substitutions, monoalphabetic solutions are easy to break!

Security

- There are n! different substitutions on an alphabet with n letters
- Assume n = 26 letters
- n = 26
- n! = 403291461126605635584000000 = $=4\times10^{26}$ keys
- Trying all possibilities at 1 nanosecond per key requires ????

Attacking Substitution Cipher

Exhaustive search

- If the key space is small enough, try all possible keys until you find the right one
- Cæsar cipher has 26 possible keys

Statistical analysis

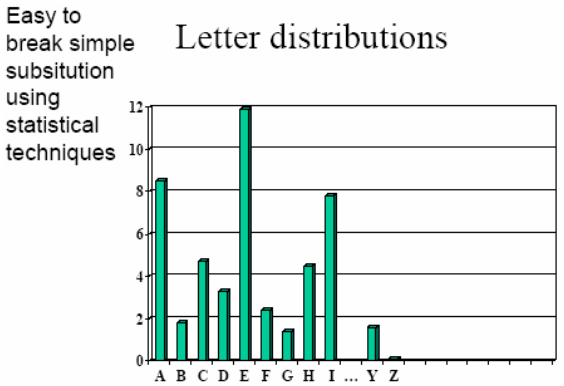
- Compare to 1-gram model of English

Statistical Attack

a	0.080	h	0.060	n	0.070	t	0.090
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	0.130	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

English Characters Frequency (Denning 1982)

Easy to break simple subsitution using statistical techniques



Breaking a Monoalphabetic Substitution

```
X ydis pq yjc xzpvpyw ya icqdepzc ayjceq xq
A tact is the ability to describe others as

yjcw qcc yjcuqcvrcq.
they see themselves.

                                Xzexjxu Vpsdavs
                                Abraham Lincoln
```

Character Frequency: c-10, y-8, q-7, x-6, j-5, p-5, v-4, d-3
a-3, e-3, z-3, s-2, u-2, w-2, i-1, r-1

Alphabet frequency: e t a o i n s r h l d c u m f p g w y b v k x j q z

Transposition (Permutation)

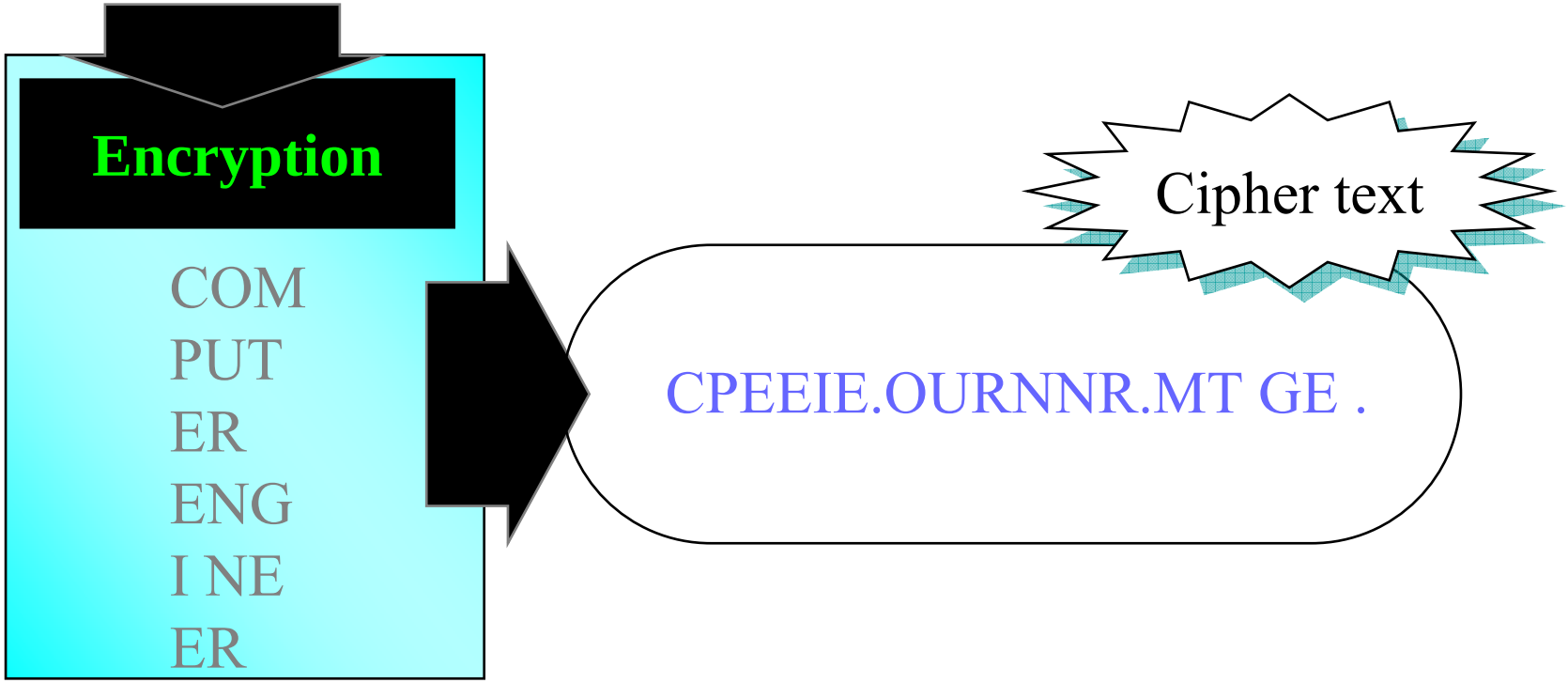
Substitution reserves places

But

Transposition reserves content

Transposition (Permutation)

COMPUTER ENGINEER



Attacking Transposition Cipher

How ???

We will leave this as a HW question !!!

Enigma Machine

Germany- World War 1

Encryption: Keys are typed in normally

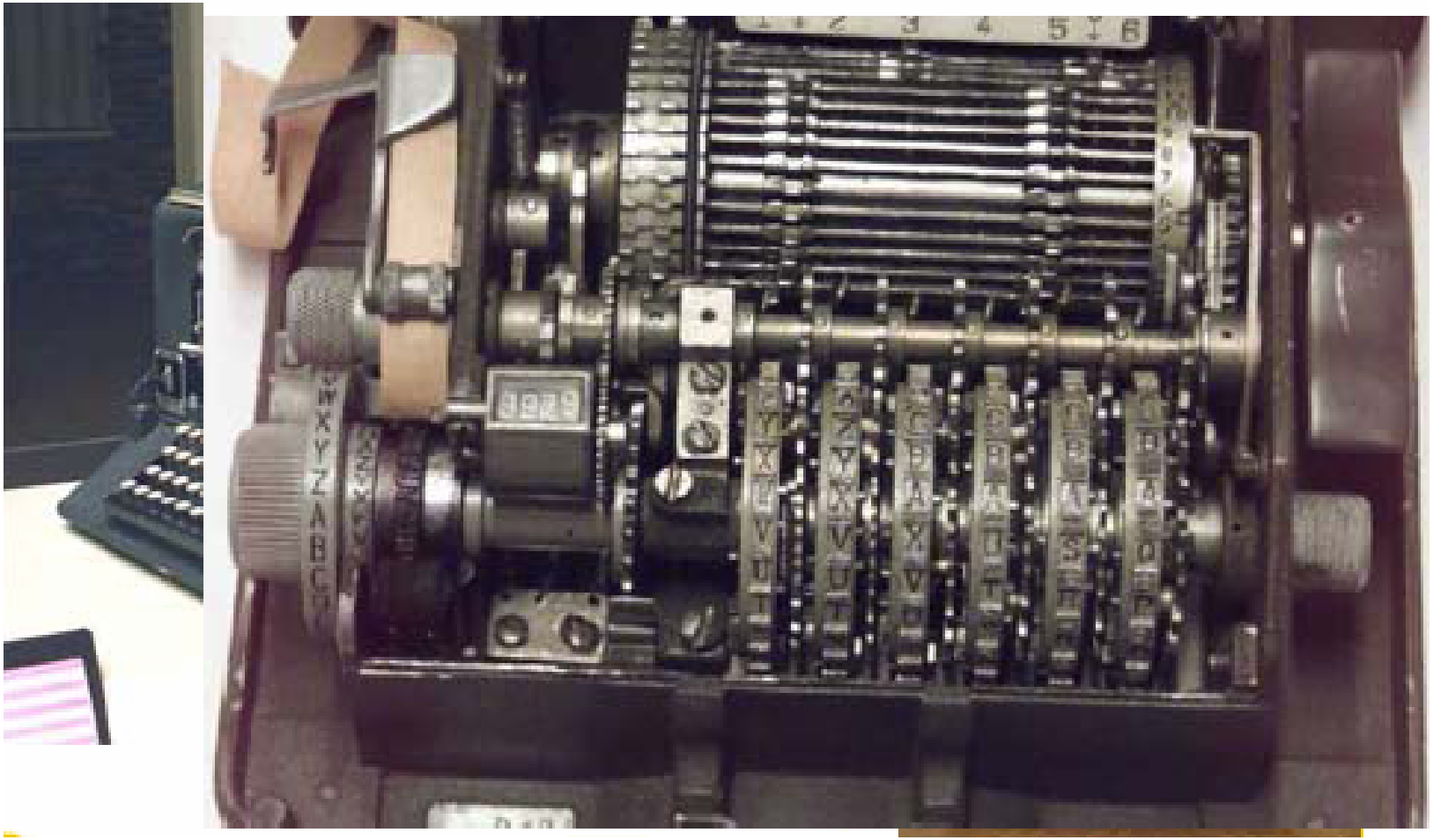
Machine output: Cipher text - encrypted message typed on paper

Decryption: Normal typing cipher text –
Machine output: Plain text on paper

Keys: Mechanical rotors

Wheel Cipher

Mechanical: Hagelin C38



Vigènere Cipher

Like Cæsar cipher, but use *phrase*

Example

- Message:
- THE BOY HAS THE BALL
- Key:
- VIG
- Encipher using Cæsar cipher for each letter:

```
key     VIGVIGVIGVIGVIGV
plain   THEBOYHASTHEBALL
cipher  OPKWWECIYOPKWIRG
```

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

Vigènere Cipher: Useful Terms

period: length of key

- In earlier example, period is 3

tableau: table used to encipher and decipher

- Vigènere cipher has key letters on top, plaintext letters on the left

polyalphabetic: the key has several different letters

- Cæsar cipher is monoalphabetic

Vigenere Cipher

Vigenere Cipher encrypts *m* alphabetic characters at a time

each plaintext element is equivalent to *m* alphabetic characters

key *K* is a *keyword* that associate with an alphabetic string of length *m*

Attacking Vigènere Cipher

Approach

- Establish period; call it *n*
- Break message into *n* parts,
  - each part being enciphered using the same key letter (Caesar cipher)
- Solve each part as a Caesar cipher!
  - You can influence one part from another
  - observe two identical segments in Ciphertext each of length at least three, then there is a good chance that they do correspond to identical segments of plaintext.

Establish Period

Kaskski: *repetitions in the ciphertext occur when characters of the key appear over the same characters in the plaintext*

Example:

```
key     VIGVIGVIGVIGVIGV
plain   THEBOYHASTHEBALL
cipher  OPKWWECIYOPKWIRG
```

the key and plaintext line up over the repetitions (underlined).

distance between repetitions is 9,

the period is a factor of 9 (that is, 1, 3, or 9)

Vigenere Cipher Secrecy

number of possible keywords of length m ➔ 26^m

if $m = 5$, then the keyspace has size exceeding 1.1×10^7.

This is already large enough to preclude exhaustive key search by hand (but not by computer).

having keyword length m, an alphabetic character can be mapped to one of m possible alphabetic characters (assuming that the keyword contains m distinct characters).

Such a cryptosystem is called *polyalphabetic*.

In general, cryptanalysis is more difficult for polyalphabetic than for monoalphabetic cryptosystems.

Block ciphers

Substitution ciphers: changing one letter in the plaintext changes exactly one letter in the ciphertext.

- This greatly facilitates finding the key using frequency analysis.

Block ciphers: prevents this by encrypting a block of letters simultaneously.

Many of the modern (symmetric) cryptosystems are block ciphers.

DES operates on 64 bits of blocks

AES uses blocks of 128 bits (192 and 256 are also possible).

Example: Hill Cipher (1929)

The key is an $n \times n$ matrix whose entries are integers in Z_{26}.

Block cipher: Hill cipher

Encryption: vector-matrix multiplication

Example: Let n=3, key matrix *'M'* be

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 11 & 9 & 8 \end{pmatrix}$$

assume the plaintext is ABC=(0,1,2)

$$(0,1,2) \times \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 11 & 9 & 8 \end{pmatrix} \equiv (26,23,22) \bmod 26 = (0,23,22) \Rightarrow AXW(ciphertext)$$

Decryption:

$$\begin{pmatrix} 22 & 5 & 1 \end{pmatrix}$$

in

$$(0,23,22) \times \begin{pmatrix} 22 & 5 & 1 \\ 6 & 17 & 24 \\ 15 & 13 & 1 \end{pmatrix} \equiv (468,677,574) \bmod 26 = (0,1,2) \Rightarrow ABC(plain-text)$$

Hill Cipher

If we change one letter in the plaintext, all the letters of the ciphertext will be affected.

Example:

Let the plaintext be ABB instead of ABC then the ciphertext is

$$(0,1,1)\times\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 11 & 9 & 8\end{pmatrix}\equiv(15,14,14)\bmod 26=(15,14,14)\Rightarrow POO(ciphertext)$$

Another Example

Use Key:

$$M = \begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}$$

Decryption Key:

$$N = \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix}$$

Hill Cipher Attack

Ciphertext:

- Hill Cipher is more difficult to break with a ciphertext-only attack.

Plaintext + Ciphertext:

1. Opponent has determined the value of *m*
2. Compute the key

Properties of Good Cryptosystems

Diffusion: one character change in the plaintext should effect as many ciphertext characters as possible.

Confusion: The key should not relate to the ciphertext in a simple way.

Shannon (1949)

One-Time Pad (Vernam Cipher)

Vernam in 1918, proposed the one-time pad, which is a provably secure cryptosystem.

Messages are represented as a binary string (a sequence of 0's and 1's using a coding mechanism such as ASCII coding.)

The key is a truly random sequence of 0's and 1's of the same length as the message.

The encryption is done by adding the key to the message modulo 2, bit by bit as *exclusive OR*, $\oplus$ (XOR).

One-time pad

Secret-key encryption scheme (symmetric)

- Encrypt plaintext by XOR with sequence of bits
- Decrypt ciphertext by XOR with same bit sequence

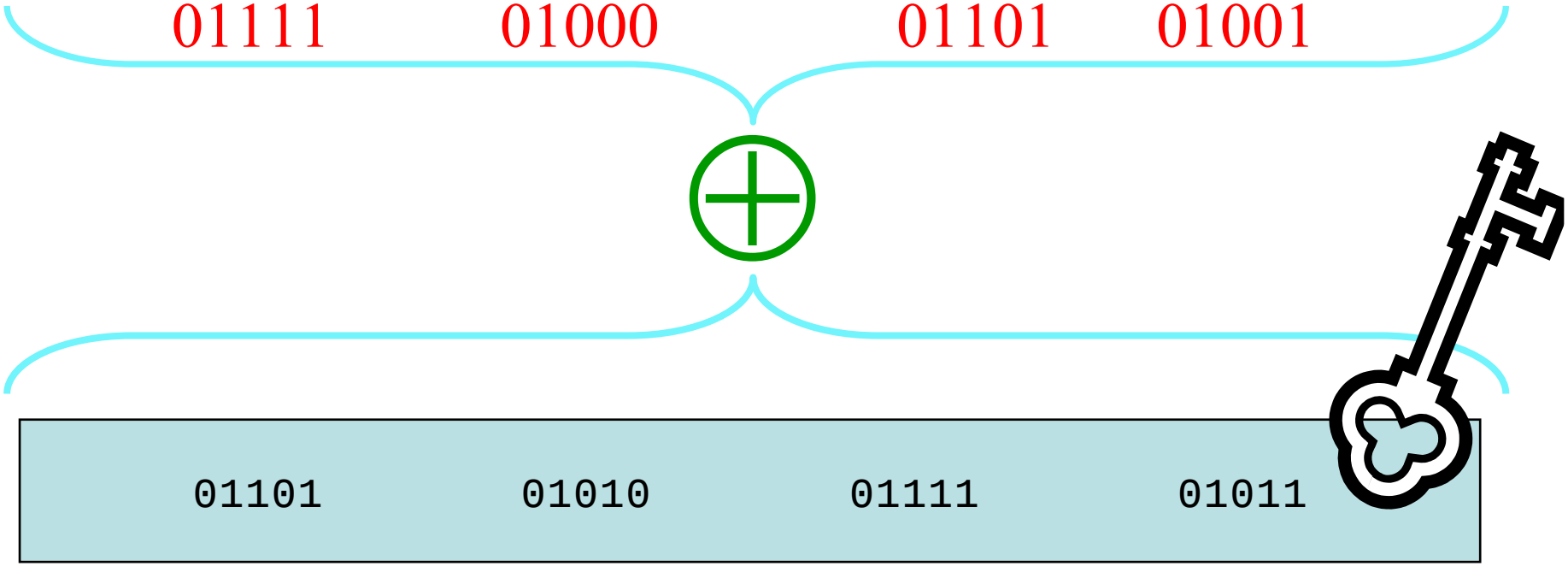
Scheme for pad of length *n*

- Set P of plaintexts: all n-bit sequences
- Set C of ciphertexts: all n-bit sequences
- Set K of keys: all n-bit sequences
- Encryption and decryption functions

encrypt(key, text) = key $\oplus$ text (bit-by-bit)

decrypt(key, text) = key $\oplus$ text (bit-by-bit)

One Time Pad: Example Unconditional Secure



00010 00010 00010 00010

Cipher???

One time pad (Vernam Cipher) Why - unconditional secure?

- General: C = (P+K) mod 26 ; P = (C-K) mod 26
  - C,P,K ∈ [0,25]; A=0, B=1,….Z=25
- Consider Ciphertext: C = XHGRQ
  - Key = AAAAA ⇒ P = XHGRQ
  - Key = VAYEK ⇒ P = CHINA
  - Key = EZANZ ⇒ P = TIGAR
  - ……
  - Key = ZZZZZ ⇒ P = YIHSR
- Conclusion: for *every* 5-character *plaintext* there is a 5-character key which maps the ciphertext to the plaintext

One-Time Pad

A Vigenère cipher with a random key at least as long as the message

- Provably unbreakable
  - Why? Look at ciphertext `DXQR`. Equally likely to correspond to plaintext `DOIT` (key `AJIY`) and to plaintext `DONT` (key `AJDY`) and any other 4 letters
- Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
  - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

Evaluation of one-time pad

Advantages

- Easy to compute encrypt, decrypt from key, text
- As hard to break as possible
  - This is an information-theoretically secure cipher
  - Given ciphertext, all possible plaintexts are equally likely, assuming that key is chosen randomly

Disadvantage

- Key is as long as the plaintext
  - How does sender get key to receiver securely?

Security of one-time pad systems relies on the condition that keys are generated using truly random sources

Idea for stream cipher: use pseudo-random generators for key...

Randomness & Pseudo-randomness

Randomness: **C**losely related to *unpredictability*

Pseudo-randomness : sequences appears random to a computationally bounded adversary

Cryptosystems need random unpredictable numbers for

One-time pad

Secret key for DES, AES, etc.

Primes p, q for RSA

Private key for ECC

Challenges used in challenge based identification systems

True random number generation (RNG)

Requires a naturally occurring source of randomness

(randomness exists in the nature)

Hardware based random number generators (RNG)

exploit the randomness which occurs in some physical phenomena

- Elapsed time between emission of particles during radioactive decay
- Thermal noise from a semiconductor diode or resistor
- Frequency instability of a free running oscillator
- The amount which a metal insulator semiconductor capacitor is charged during a fixed period of time.

The first two are subject to observation and manipulation by adversaries.

Software base RNG

1. The system clock
2. Elapsed time between keystrokes or mouse movement
3. Content of input/output buffer
4. User input
5. OS values such as system load and network statistics.

All of them are subject to observation and manipulation.

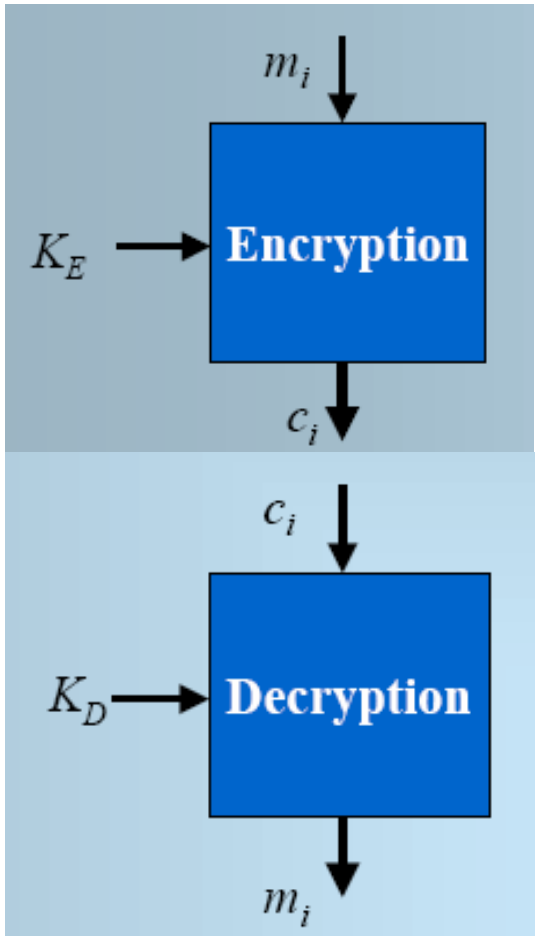
Individually these sources are very "weak".

The randomness can be increased by combining the outputs of these sources using a complex mixing function (e.g. hashing the concatenation of the output bits).

Still, not quite secure!

Revisit Block cipher

- Is function which maps *n*-bit plaintext blocks to *n*-bit ciphertext blocks; *n* is called the *blocklength*.
- It may be viewed as a simple substitution cipher with a large character size.
- The function is parameterized by a *k*-bit key *K*.
- $K_D = F(K_E)$



Data Encryption Standard (DES)

In 1975, the NBS: National Bureau of Standards (later NIST) released DES (Data Encryption Standard) and a free license for its use.

Standard widely used in banking industry since 1977 (should be replaced in 2000).

Biham & Shamir in 1990, showed an efficient cryptoanalysis method (*differential*) to attack DES.

Data Encryption Standard (DES)

  - Most widely used block cipher in world
  - Encrypts 64-bit data using 56-bit key
  - Has widespread use
  - Has been considerable controversy over its security
- IBM developed Lucifer cipher
  - by team led by Feistel in late 60s
  - used 64-bit data blocks with 128-bit key

- in 1973 NBS issued request for proposals for a national cipher standard

- IBM submitted their revised Lucifer which was eventually accepted as the DES

DES

Was designed to encipher sensitive but unclassified data

A block cipher (64 bits):

- encrypts blocks of 64 bits using a 64 bit key
- outputs 64 bits of ciphertext

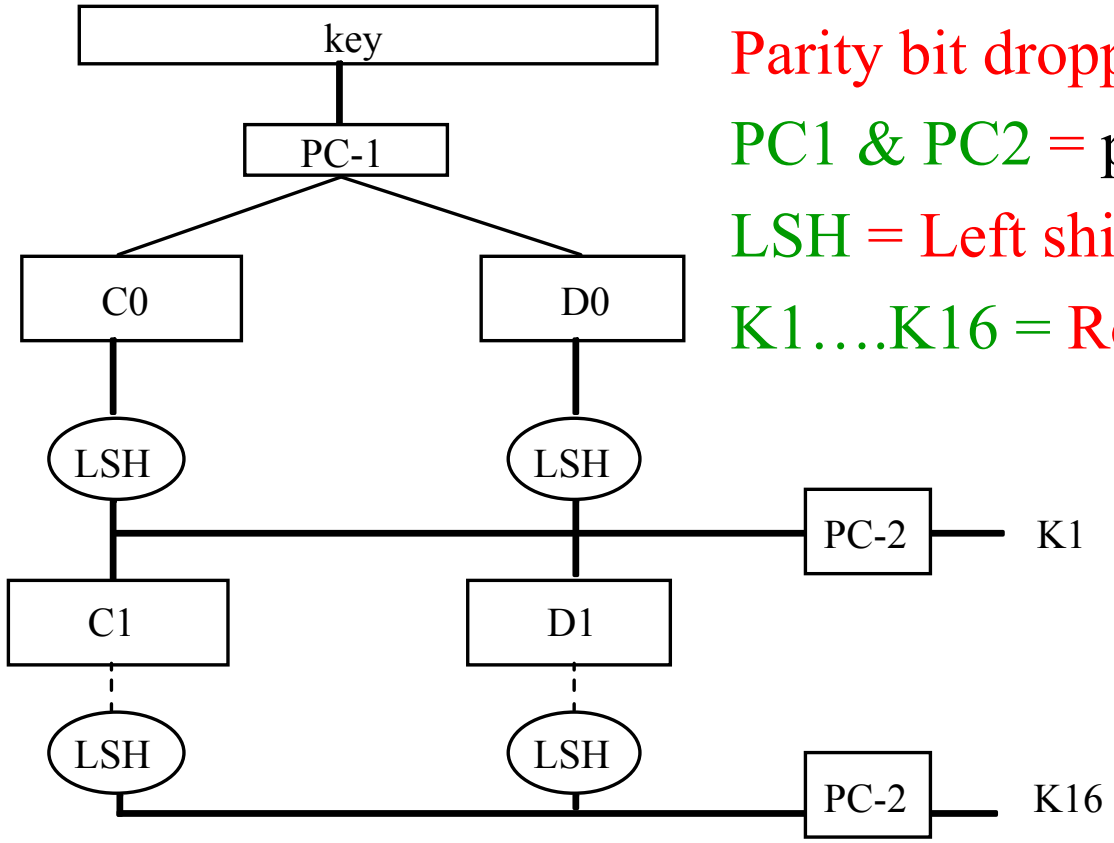
A product cipher

- basic unit is the bit
- performs both substitution and transposition (permutation) on the bits

Cipher consists of 16 rounds (iterations) each with a round key *generated* from the user-supplied key

Round key = 48 bits

Generation of Round Keys



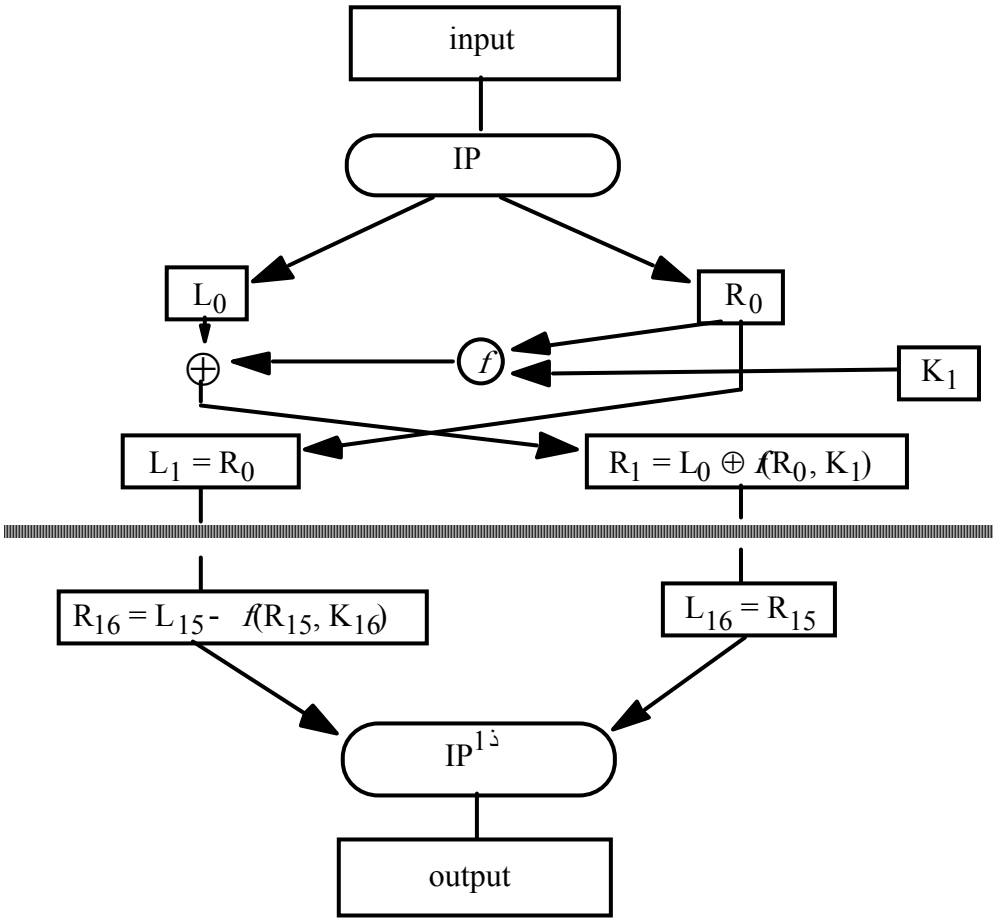
Parity bit dropped ⇒ 56 bits

PC1 & PC2 = permutation tables

LSH = Left shift (rotations)

K1….K16 = Round keys = 48 bits each

Encipherment



Input = 64 bits

Output of rounds 1 = input of round 2

Round input is partitioned into L & R = 32 bits each

R is to be exnded to 48 bits.

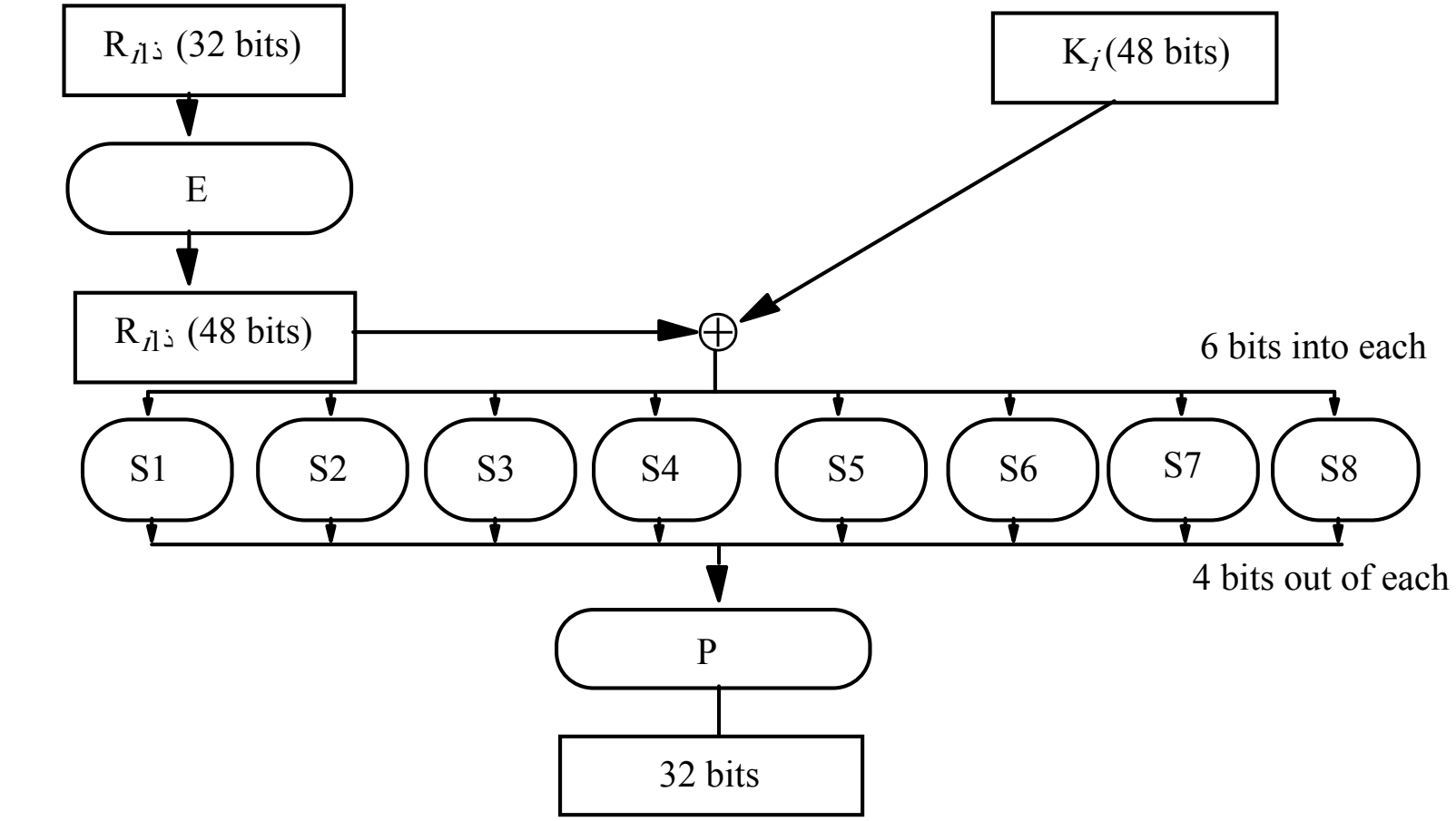
f runs on R & K = 48 bits producing 32 bits output to be XOR'd with L

XORing Output (32 bits) ⇒ expanded to 48 bits ⇒ new R

Previous R ⇒ new L

Strength of DES is in Function *f*

The *f* Function



Controversy

Considered too weak

- Diffie, Hellman said in a few years technology would allow DES to be broken in days
  - Key is too short
  - DES Break Design was published
- Design decisions not public
  - S-boxes may have backdoors

Undesirable Properties

- 4 weak keys
  - They are their own inverses
- 12 semi-weak keys
  - Each has another semi-weak key as inverse
- Complementation property
  - $\text{DES}_k(m) = c \Rightarrow \text{DES}_k(m') = c'$
- S-boxes exhibit irregular properties
  - Distribution of odd, even numbers non-random
    - DES did not randomize input sufficiently
  - Outputs of fourth box depends on input to third box
    - After five rounds, it can be noticed that each output bit depended on every key input bit

Differential Cryptanalysis

Biham & Shamir 1990

A chosen ciphertext attack

- Requires 2^{47} plaintext, ciphertext pairs
  - Much fewer than several trail-error approaches by others

Revealed several properties

- Small changes in S-boxes reduce the number of pairs needed ----- weakened the cipher --- reducing attacks effort
- Making every round key independent does not delay attack

Linear cryptanalysis improves result

- Requires 2^{43} plaintext, ciphertext pairs

DES Modes

Electronic Code Book Mode (ECB)

- Using DES directly - Encipher each block independently (rarely used)

Cipher Block Chaining Mode (CBC)

- XOR each block with previous ciphertext block
- Requires an initialization vector for the first one

Triple DES: used by many financial institutions

- Encrypt-Decrypt-Encrypt Mode (2 keys: k, k')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}^{-1}(\mathrm{DES}_k(m)))$
- Encrypt-Encrypt-Encrypt Mode (3 keys: k, k', k'')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}(\mathrm{DES}_{k''}(m)))$

Electronic Codebook (ECB)
Mode of operation

Plaintext P is broken into n-bit blocks, i.e. $P = P_1\ P_2\ \ldots\ P_L$

Ciphertext consists of the blocks $C = C_1\ C_2\ \ldots\ C_L$

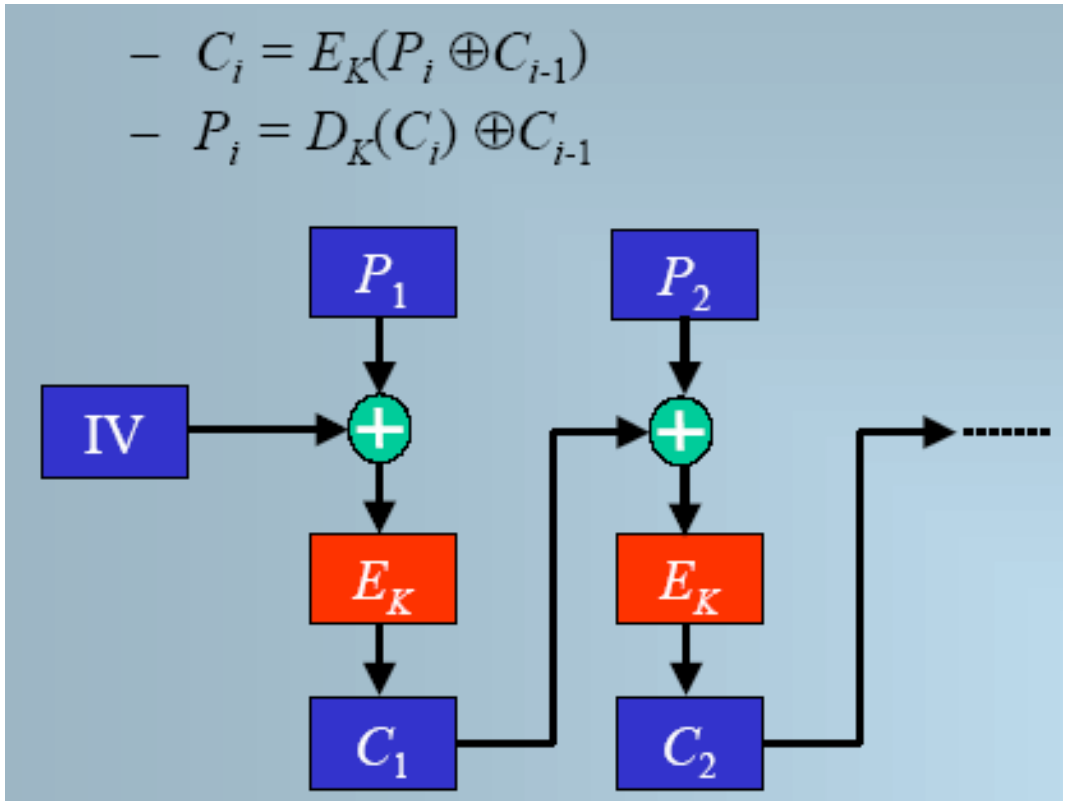
Where $C_i = E_K(P_i)$ for $i = 1,2, \ldots, L$.

- Identical plaintext blocks (under the same key) results in identical ciphertext.
- Each block is encrypted independently of others.
- Malicious block substitutions does not affect decryption of other blocks.
- Errors in a single block do not propagate to other blocks.
- Not recommended for messages of more than one block.

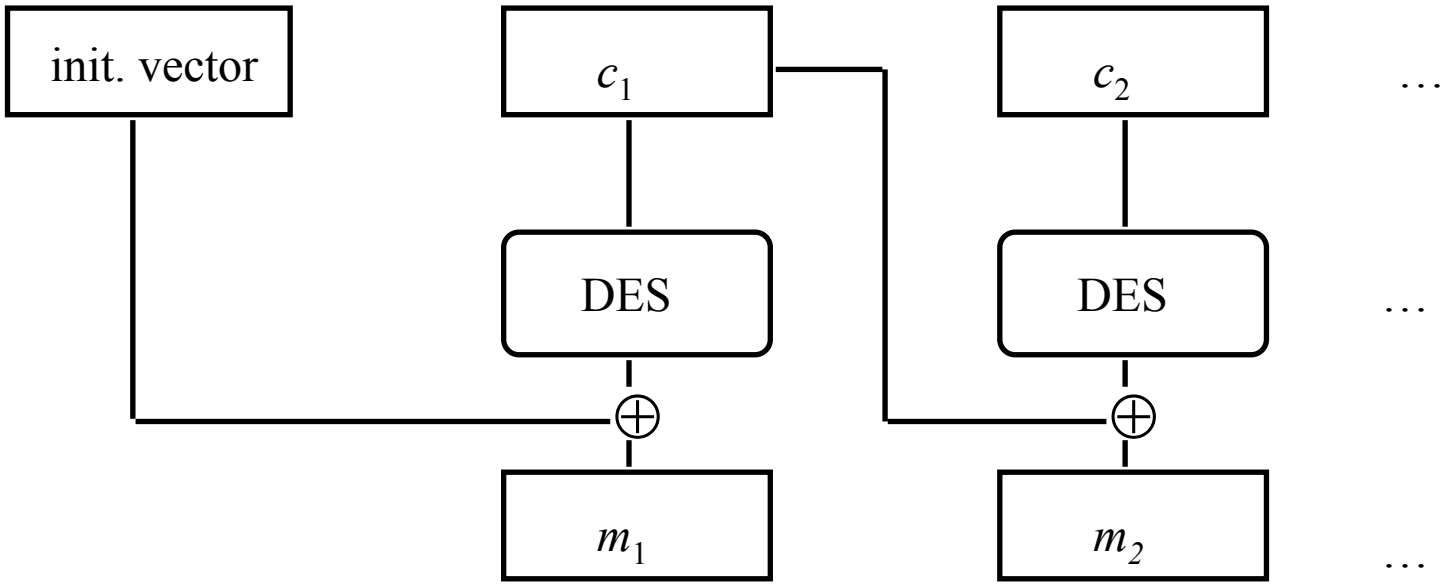
Cipher Block Chaining (CBC)

Mode of operation

- Encryption of a block depends on encryption of previous blocks.
- Needs 64-bits initial vector (IV)
- No error-recovery after an error in plaintext
- Error propagation & Self-synchronizing (Self healing property)
  - if a block is altered ⇒ error propagation does not go beyond two blocks



CBC Mode Decryption



Self-Healing Property

Initial message

- 3231343336353837 3231343336353837
  3231343336353837 3231343336353837

Received as (underlined 4c should be 4b)

- ef7c4cb2b4ce6f3b f6266e3a97af0e2c
  746ab9a6308f4256 33e60b451b09603d

Which decrypts to

- efca61e19f4836f1 3231333336353837
  3231343336353837 3231343336353837
  - Incorrect bytes underlined
  - Plaintext “heals” after 2 blocks

Other alternatives

Triple DES:

- $C = E_{k1}(D_{k2}(E_{k1}(P)))$ provides 112-bit security.
- $C = E_{k3}(E_{k2}(E_{k1}(P)))$ provides 112-bit security.

DESX:

- $C = k_3 \oplus E_{k2}(E_{k1}(P))$
- Fairly secure

Rijndael (AES):

- was elected as the Advanced Encryption Standard (AES) out of 15 candidate algorithms.

History of Rijndael AES

Successor to DES

- The AES selection is administered by NIST
- Unlike DES, AES selection was an open process.
  - 1997, NIST called for candidates to replace DES.
  - Requirements were
    - Block cipher with 128-bit block size
    - Support for 128, 192, 256 bits of key sizes
    - Efficient software and hardware implementation.

Cryptographic community was asked to comment on five finalists:

MARS(IBM), RC6(RSA), Rijndael, Serpent, Twofish

- NIST chose Rijndael as AES in 2000.

- Likely to be the most commonly used algorithm in the next decade.
- for more information www.nist.gov/aes

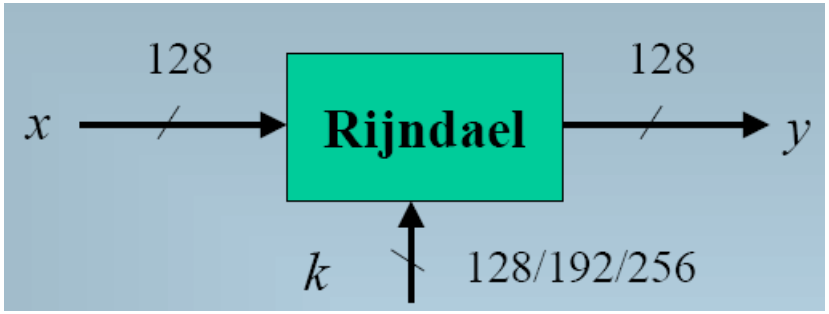
Speeds of the five finalists

Algorithm	Pentium Pro 200 Mhz Mbit/s	FPGA hardware Gbit/s
MARS	69	-
RC6	105	2.4
Rijndael	71	1.9
Serpent	27	4.9
Twofish	95	1.6

Rijndael Overview

Block size is also variable (128/192/256)

# of rounds is a function of key length:



Key length (in bits)	#of rounds n_r
128	10
192	12
256	14

Rijndael overview

Rijndael is not a Feistel cipher.

- Feistel ciphers do not encrypt the whole block in each iteration. This explains why Rijndael has fewer # of rounds.

Rijndael has three basic steps (or so called layers):

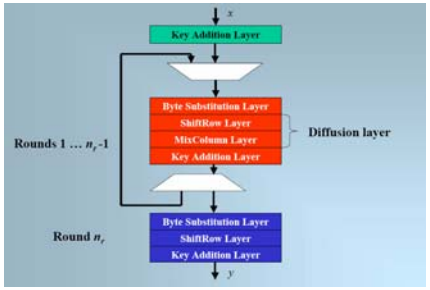
- **Key Addition Layer**: XORing the block with the round key.
- **Byte Substitution Layer:** 8-by-8 substitution (s-box).

Nonlinear operation (confusion).

- **Diffusion Layer:** provides the diffusion of the bits of a block. Linear diffusion layer.

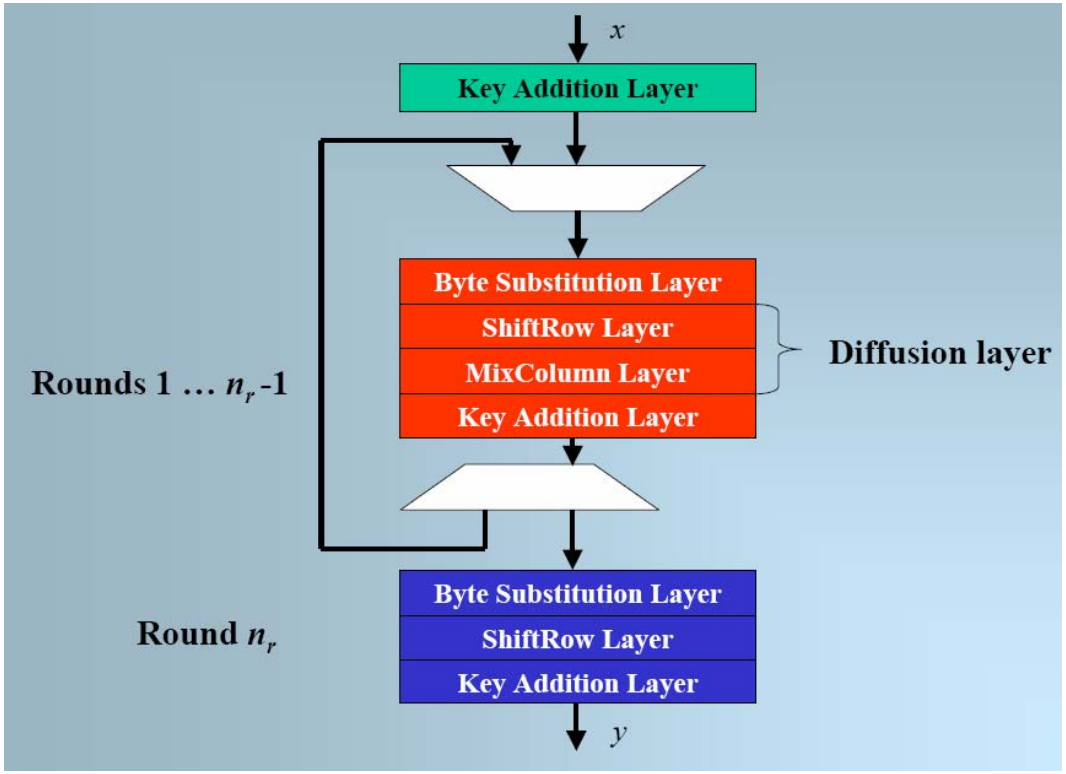
ShiftRow Layer

MixColumn Layer

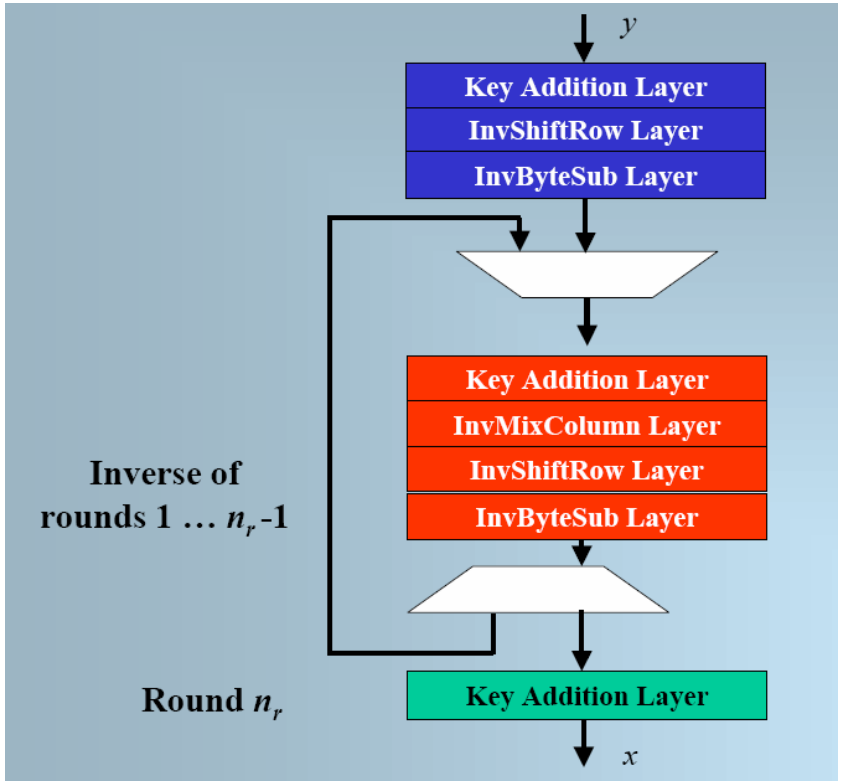


Rijndael AES Crypto operation

Encryption



Decryption



Remarks

In every round, each bit in the block are treated uniformly

- This has the effect of diffusing the input bits faster
- After two rounds each of the 128 output bits depends on each of the 128 input bits.

S-box is constructed using a very simple algebraic mapping, $x \rightarrow x^{-1}$ in GF(2^8).

- The mapping is highly nonlinear.
- Its simplicity removes any suspicions about a certain trapdoor which is believed to exist in DES for years.

The MixColumn layer causes diffusion in the byte level.

Key scheduling also utilizes highly nonlinear Byte Substitution mapping.

No known attacks are better than brute force for seven or more rounds (Rijndael makes use of at least 10 rounds).

Public Key Cryptography

Two keys

- Idea: Diffie & Hellman ~ 1976 > 30 years

– *Private key* known only to individual

– *Public key* available to anyone

- Public key, private key inverses

Idea

– Confidentiality: encipher using public key, decipher using private key

– Integrity/authentication: encipher using private key, decipher using public one

Requirements

- It must be computationally
  - easy to encipher or decipher a message given the appropriate key
  - infeasible to derive the private key from the public key
  - infeasible to determine the private key from a chosen plaintext attack

Overview of Public Key Cryptosystem

- Integer factorization problems (RSA)
- Discrete Logarithm problems (Diffie-Helman, ElGamal)
- Elliptic Curve Cryptosystems

Algorithm family	Bit length
Integer Factorization (IF)	1024
Discrete Logarithm (DL)	1024
Elliptic curves (EC)	160
Block cipher	80

RSA

- 1978 @ MIT: Rivest Shamir Adleman = RSA
  - 2 years after Diffie Helman idea was proposed
- Exponentiation cipher
- Based on *Integer Factorization* problem
  - Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*
- Its patent expired in 2000.

Background

Totient function $\phi(n)$

- Number of positive integers less than *n* and relatively prime to *n*
  - *Relatively prime* means with no factors in common with *n*

Example: $\phi(10) = 4$

- 1, 3, 7, 9 are relatively prime to 10

Example: $\phi(21) = 12$

- 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

RSA Algorithm

Choose: $p, q \in$ positive distinct *large primes*

Compute: $n = p \times q$

n = encryption/decryption modulus $\rightarrow$ computations in Z_n

Compute: $\varphi(n) = (p - 1)(q - 1)$

Choose randomly: $e \in Z_{\varphi(n)}^*$

$\rightarrow$ gcd($\varphi(n)$,e)=1, (e has an inverse mod $\varphi(n)$)

Find $d = e^{-1} = ??\ mod\ \varphi(n)$

- Compute d such that ed mod $\phi(n) = 1$

Encryption: $c = x^e\ mod\ n$ where $x < n$

Decryption: $x = c^d\ mod\ n$

n,e are made public but p,q,d are secret

Example: Confidentiality

Encryption

Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$

Alice chooses $e = 17$, making $d = 53$

Bob wants to send Alice secret message HELLO (07 04 11 11 14)

- $07^{17} \bmod 77 = 28$
- $04^{17} \bmod 77 = 16$
- $11^{17} \bmod 77 = 44$
- $11^{17} \bmod 77 = 44$
- $14^{17} \bmod 77 = 42$

Bob sends 28 16 44 44 42

Cont. Example: Confidentiality

Decryption

Alice receives 28 16 44 44 42

Alice uses private key, $d = 53$, to decrypt message:

- $28^{53} \bmod 77 = 07$
- $16^{53} \bmod 77 = 04$
- $44^{53} \bmod 77 = 11$
- $44^{53} \bmod 77 = 11$
- $42^{53} \bmod 77 = 14$

Alice translates message to letters to read HELLO

- No one else could read it, as only Alice knows her private key and that is needed for decryption

Example: Integrity/Authentication

Signing

Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$

Alice chooses $e = 17$, making $d = 53$

Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)

- $07^{53} \bmod 77 = 35$
- $04^{53} \bmod 77 = 09$
- $11^{53} \bmod 77 = 44$
- $11^{53} \bmod 77 = 44$
- $14^{53} \bmod 77 = 49$

Alice sends 35 09 44 44 49

Example: Integrity/Authentication

Verifying Signature

Bob receives 35 09 44 44 49

Bob uses Alice's public key, $e = 17$, $n = 77$, to decrypt message:

- $35^{17} \bmod 77 = 07$
- $09^{17} \bmod 77 = 04$
- $44^{17} \bmod 77 = 11$
- $44^{17} \bmod 77 = 11$
- $49^{17} \bmod 77 = 14$

Bob translates message to letters to read HELLO

- Alice sent it as only she knows her private key, so no one else could have enciphered it
- If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly

Authenticity + Confidentiality

Alice A.priv	A.pub, B.pub, ...	Bob B.priv
“Hello”		
	Carl & Eve Bad People!	

Authenticity + Confidentiality

Alice A.priv	A.pub, B.pub, ...	Bob B.priv
“Hello” B.pub		
	Carl & Eve Bad People!	

Authenticity + Confidentiality

Alice A.priv	A.pub, B.pub, ...	Bob B.priv

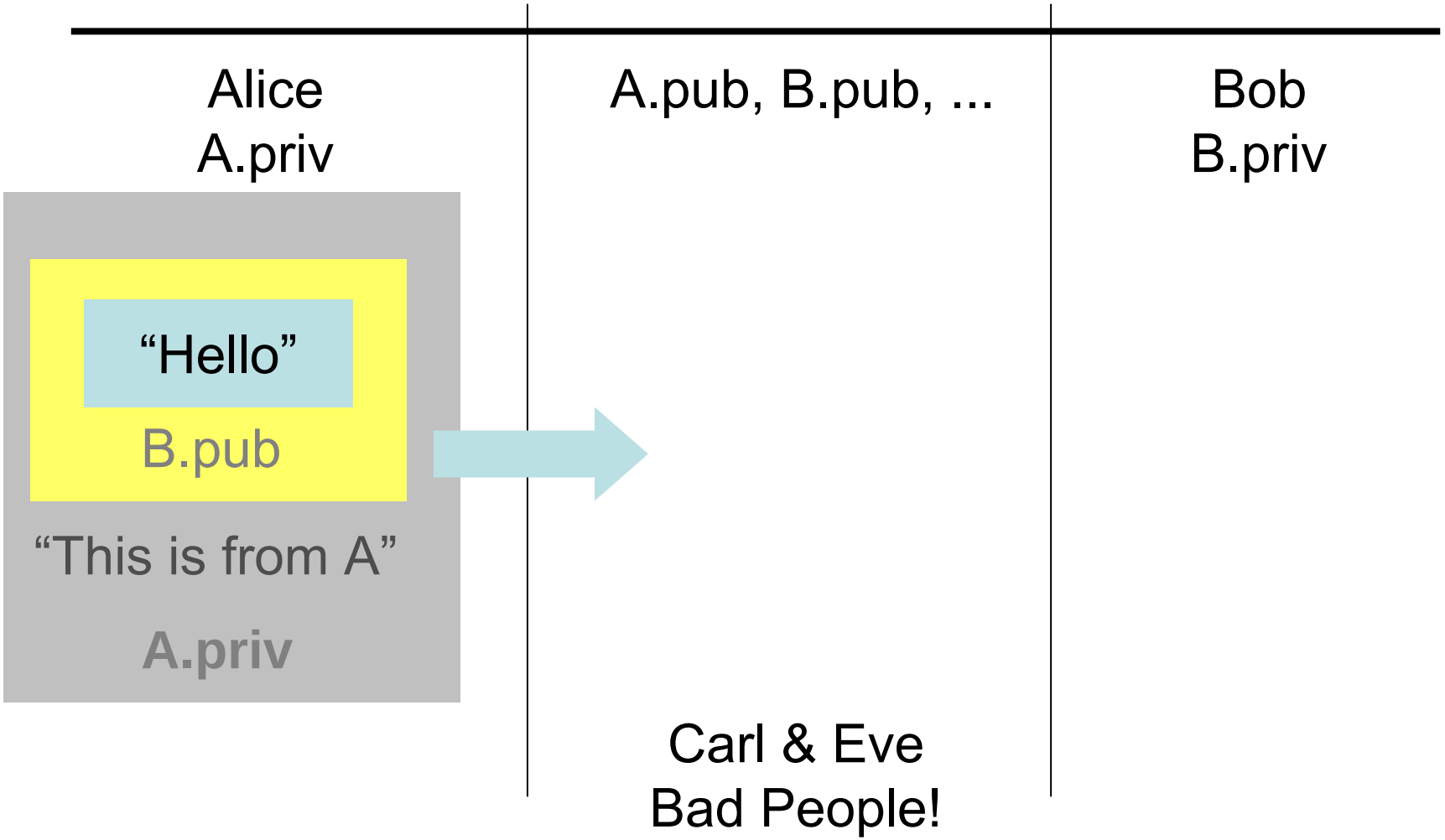
"Hello"

B.pub

"This is from A"

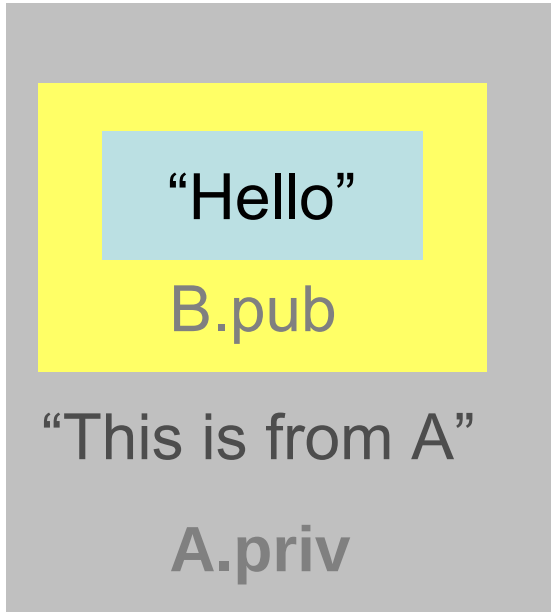
Carl & Eve
Bad People!

Authenticity + Confidentiality



Authenticity + Confidentiality

Alice	A.pub, B.pub, ...	Bob
A.priv		B.priv



Carl & Eve
Bad People!

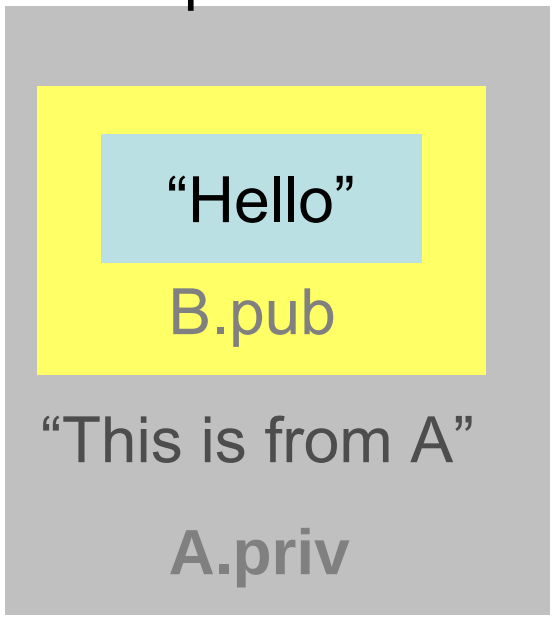
Authenticity + Confidentiality

Alice
A.priv

A.pub, B.pub, ...

Carl & Eve
Bad People!

Bob
B.priv



Example: Both

Alice wants to send Bob message HELLO both enciphered and authenticated (integrity-checked)

- Alice's keys: public (17, 77); private: 53
- Bob's keys: public: (37, 77); private: 13

Alice enciphers HELLO (07 04 11 11 14):

- $(07^{53} \bmod 77)^{37} \bmod 77 = 07$
- $(04^{53} \bmod 77)^{37} \bmod 77 = 37$
- $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
- $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
- $(14^{53} \bmod 77)^{37} \bmod 77 = 14$

Alice sends 07 37 44 44 14

RSA keys …. Example (simple)

$p = 11$, $q = 5$ ➔ $n = 55$

$\varphi(n) = 10 \times 4 = 40 \qquad = 2^3 \times 5$

an integer e can be used as an encryption exponent if and only if e is not divisible by 2, 5

We do not need to factor $\varphi(n)$ to get e

Just verify: $\gcd(\varphi(n), e) = 1$ (Euclidean algorithm)

Assume: $e = 7$ (public key)

Extended Euclidean algorithm $\Rightarrow$ $e^{-1} = ??\ mod\ 40$

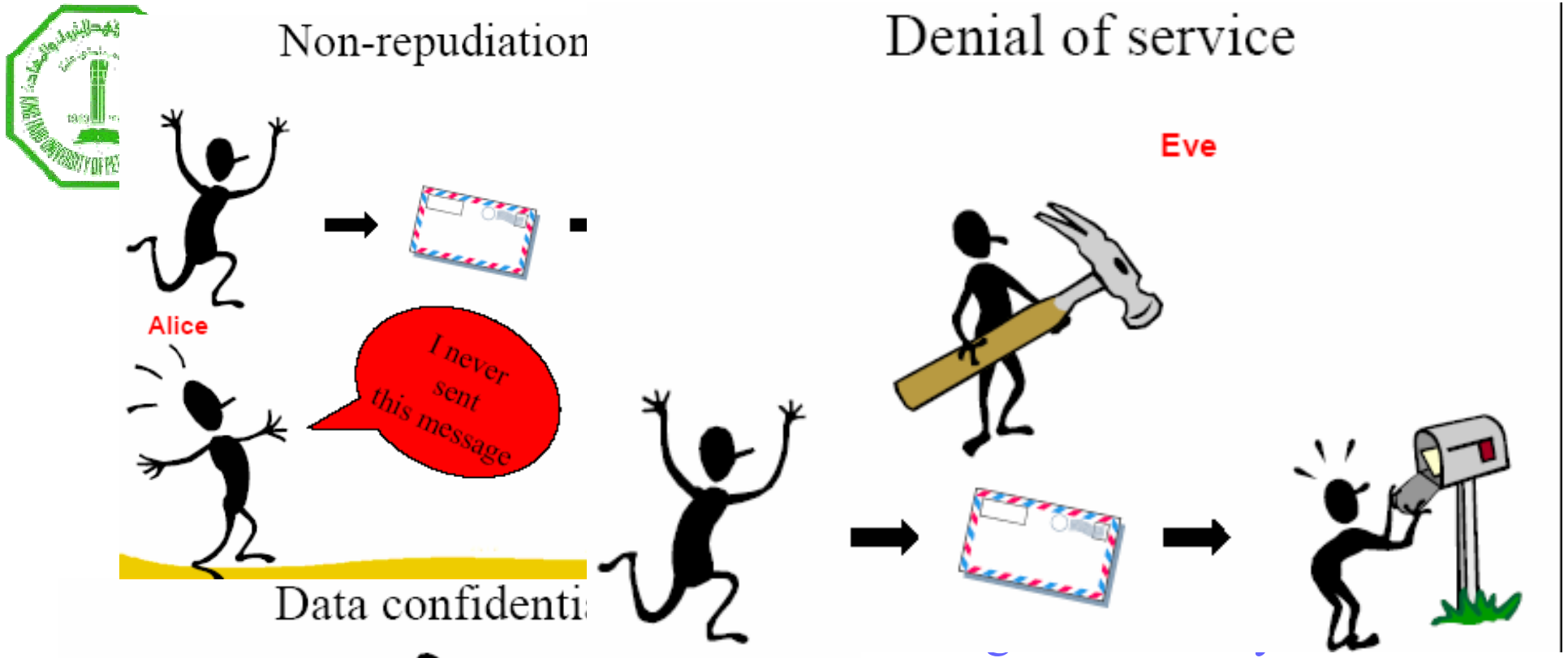
Secret exponent key: *23*

other pares: $e=3,\ e^{-1}=??$ $e=9,\ e^{-1}=??$ $e=11,\ e^{-1}=??$ $e=13,\ e^{-1}=??$ $e=17,\ e^{-1}=??$ $e=19,\ e^{-1}=??$

$\boldsymbol{Z_{40}^{*}=\{1,3,7,9,11,13,17,19,21,23,27,29,31,33,37,39\}}$

$e=3,\ e^{-1}=27$ $e=13,\ e^{-1}=37$ $e=17,\ e^{-1}=33$

$e=e^{-1}= \{9, 11, 19, 21, 29, 31, 39\}$


Non-repudiation
Denial of service
Alice
I never sent this message
Eve
Data confidenti
Eve
Entity authentication
Hello, I am Alice

Other Cryptographic Applications

Digital Signatures

- allows electronically sign (personalize) the electronic documents, messages and transactions

Identification

- is capable of replacing password-based identification methods with more powerful (secure) techniques

Key Establishment

- To communicate a key to your correspondent (or perhaps actually mutually generate it with him) whom you have never physically met before

Secret Sharing

- Distribute the parts of a secret to a group of people who can never exploit it individually

Other Cryptographic Applications

E-commerce

- carry out the secure transaction over an insecure channel like Internet

E-cash

- The cash can be sent securely through computer networks
- The cash cannot be copied and reused
- The spender of the cash can remain anonymous
- The transaction can be done *offline*
- The cash transferred to others
- A piece of cash can be divided into smaller amounts

Games

- Flipping coins over the phone

Electronic Voting

Warnings

Encipher message in blocks considerably larger than the examples here

- If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
- Attacker cannot alter letters, but can rearrange them and alter message meaning
  - Example: reverse enciphered message of text ON to get NO

Elliptic Curve Cryptography (ECC)

What is Elliptic Curve Cryptography (ECC)?

- ECC:cryptography technique based on *elliptic curve theory* that can be used as faster, smaller, and more efficient cryptosystem.

Who introduced it and when?

- Victor Miller and Neal Koblitz independently, around 1985

What is the basic principle?

- Obtain same level of security as conventional cryptosystems but with much smaller key size

Why use ECC?

How do we analyze Cryptosystems?

- How difficult is the underlying problem that it is based upon?
  - RSA – Integer Factorization
  - ElGamal - DSA – Discrete Logarithms
  - ECC - Elliptic Curve Discrete Logarithm problem
- How do we measure difficulty?
  - We examine the algorithms used to solve these problems

Benefits of ECC

Same benefits of the other cryptosystems:

confidentiality, integrity, authentication and non-repudiation but…

Shorter key lengths

- Encryption, Decryption and Signature Verification speed up
- Storage and bandwidth savings

Applications of ECC

Many devices are small and have limited storage and computational power

Where can we apply ECC?

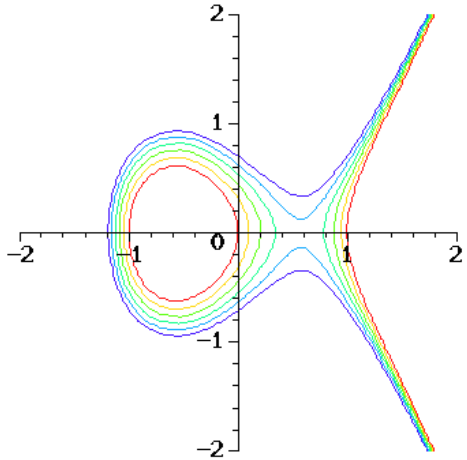
- Wireless communication devices
- Smart cards
- Web servers that need to handle many encryption sessions
- Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems

Security Equivalent key sizes

Symmetric	ECC	DH/DSA/RSA
80	163	1024
128	283	3072
192	409	7680
256	571	15360

Elliptic Curves

An Elliptic Curve is such an alternate cyclic group. The group consists of all points of the form: $y^2 = x^3 + ax + b$. Where x, y, a, and b are all elements of a field F.

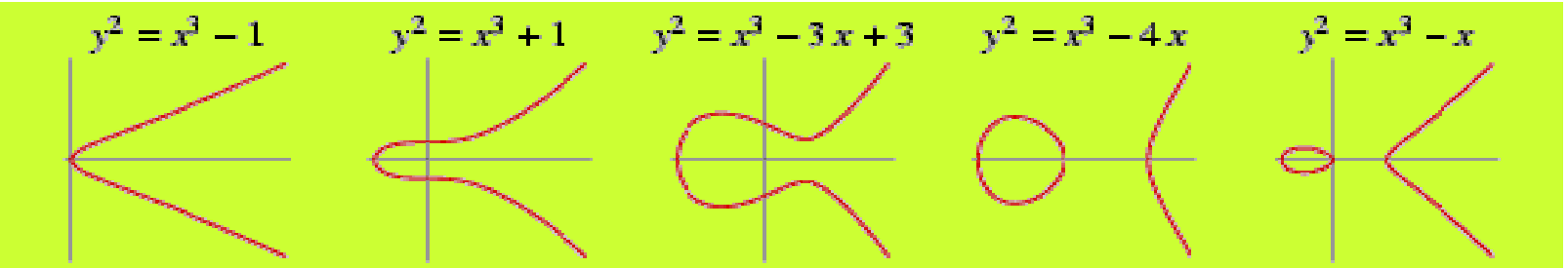


General form of a EC

An *elliptic curve* is a plane curve defined by an equation of the form

$$y^2 = x^3 + ax + b$$

Examples



Elliptic Curves over GF(p)

GF(p): Modulo *p* operation

An elliptic curve ***Group*** over a finite field defines

- a **set of points** (x, y) that satisfy the elliptic curve equation, together with the "***point at infinity***" (***O***), the EC equation is given by:
  - ***GF(p)*:** $y^2 = x^3 + ax^2 + b$
    - $a, b \in GF(p)$, and
    - $4a^3 + 27b^2 \neq 0 \pmod{p}$

ECC Encryption/Decryption

- **Public Information**
  - Elliptic Curve ***E***, and the **base point** $P = (x_p, y_p)$.

Receiver

Choose a random Private key $\boldsymbol{k_A}$ and DECLARE $\boldsymbol{k_A P}$ as a Public key

Compute $\boldsymbol{k_A C_1}$ $(= \boldsymbol{k_A k_B P})$

Retrieve the message by computing:

$$M = C_2 - k_A k_B P = C_2 + (- k_A k_B P)$$

Sender

Message M is embedded into E.

Choose a random Private key $\boldsymbol{k_B}$

Compute:

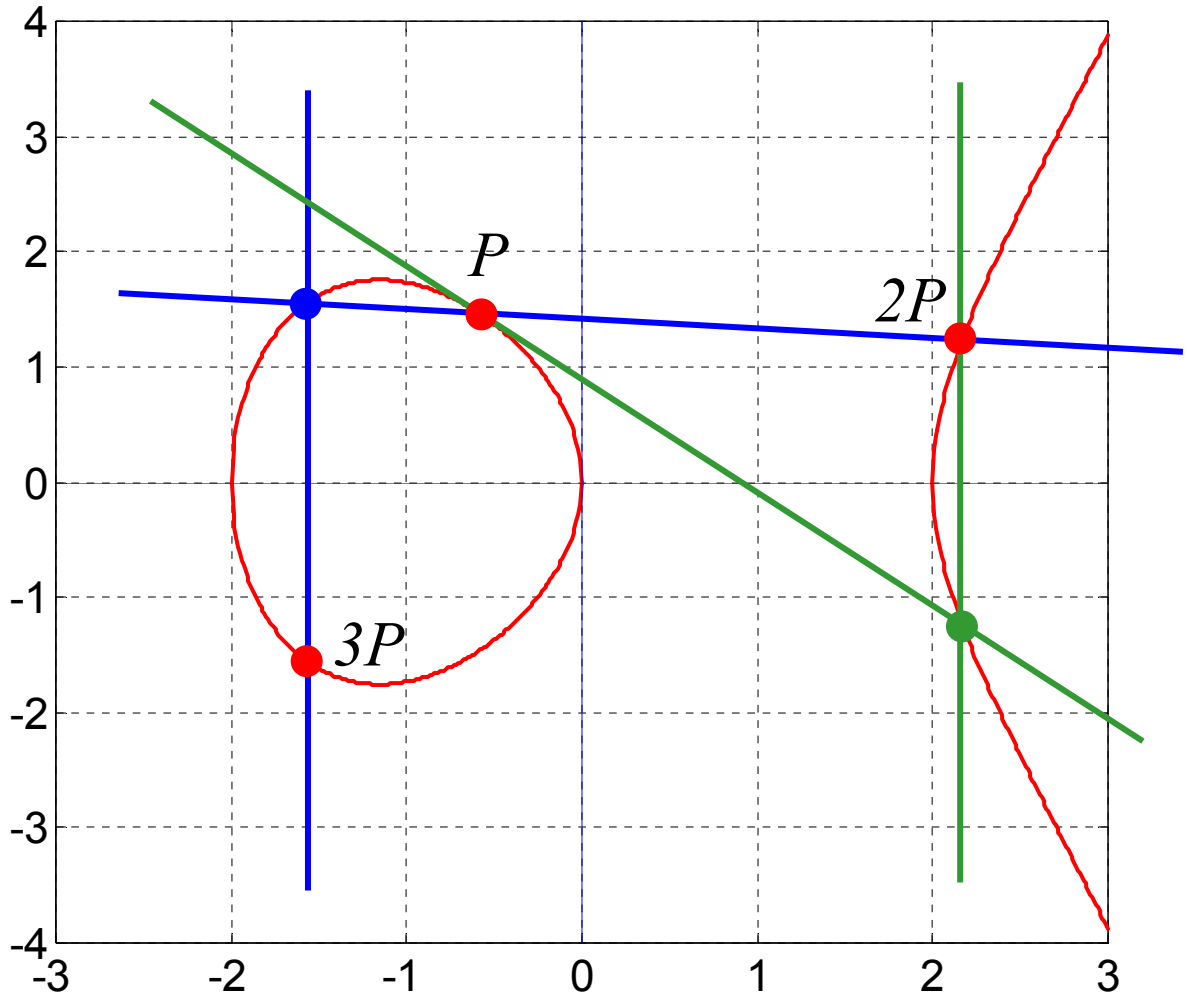
- $\boldsymbol{k_B k_A P}$.
- $C_1 = \boldsymbol{k_B P}$
- $C_2 = (x_m, y_m) + k_B k_A P$

Send (C_1 , C_2) as the encrypted message.

Scalar Multiplication

Also called point multiplication

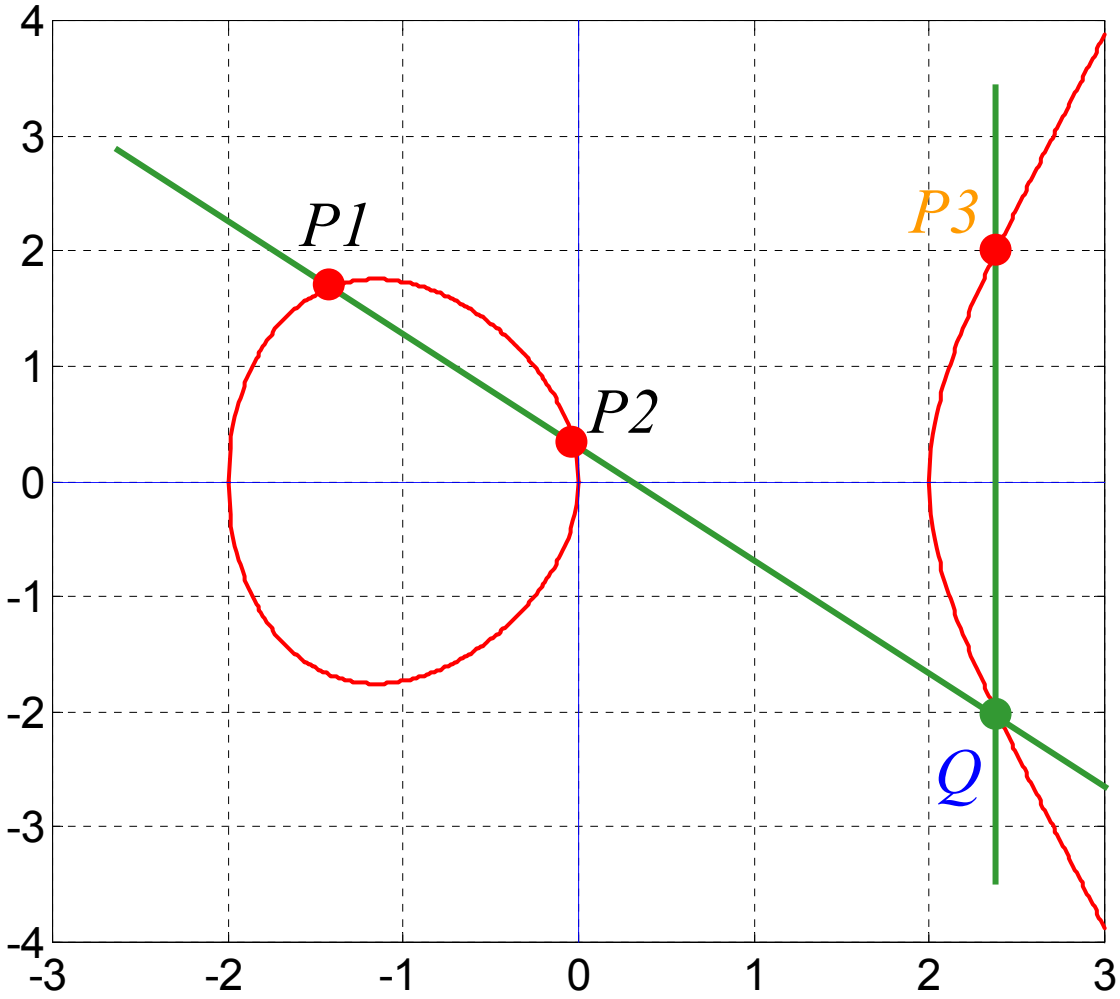
- $KP = P + P + P + \ldots + P$ (K times)
- Where K is an integer.



Point Addition

Adding 2 EC Points P1 & P2:

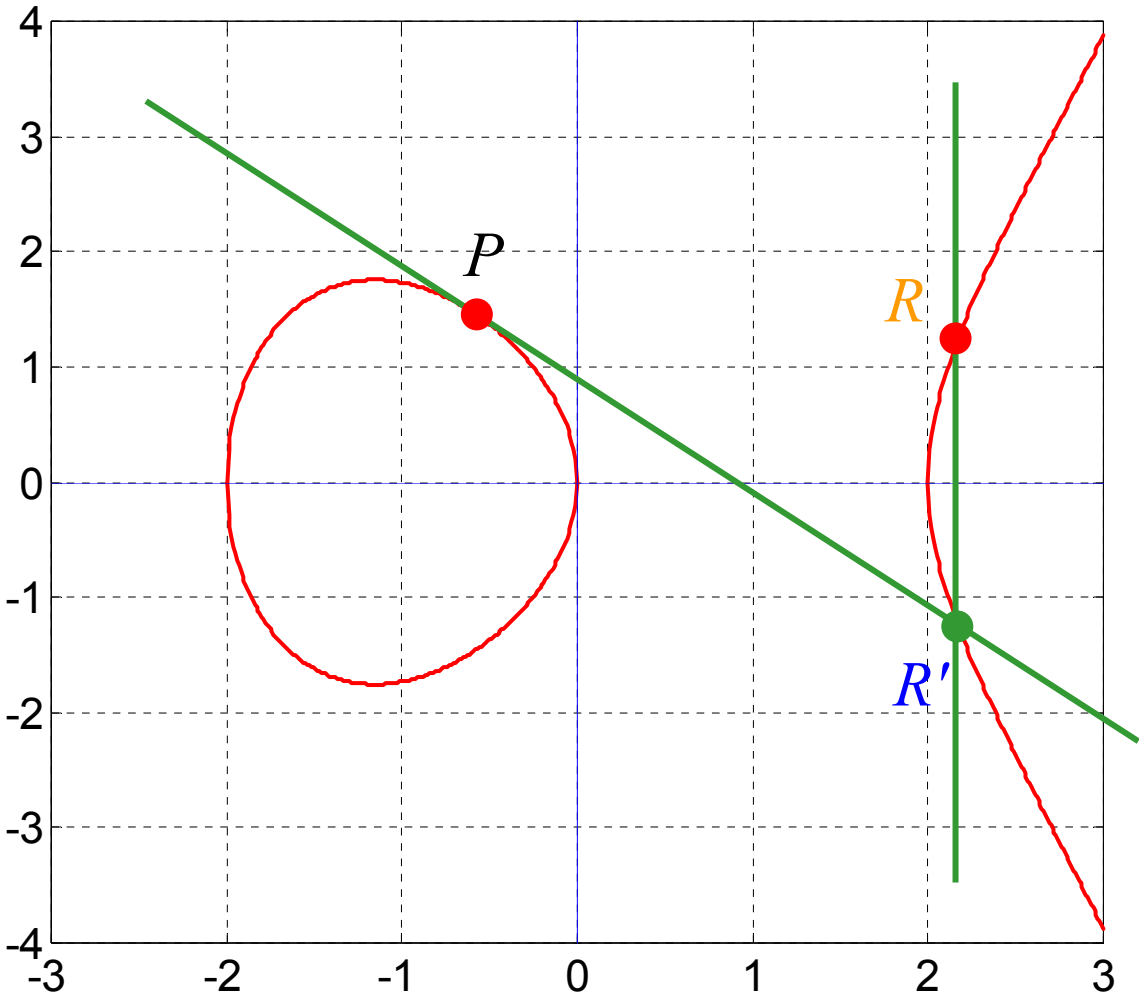
- Draw straight line connecting P1 and P2
- Line intersects the EC at *Q*
- The *point* P3 = P1+P2 is the replica point of Q wrt x-axis.
- P1 + P2 = P3



Point Doubling

What if $P_1=P_2=P$?

- P + P → 2P
- → point doubling
- Draw a tangent line through *P*,
- Tangent intersects the EC at R′,
- The *point R* =**2P** is the replica of *R′* wrt x-axis.
- P+P = 2P = R



ECC Remarks

- ECC provide same security as RSA with much less computations
- ECC maps the message into a point on the elliptic curve
- ECC Encryption ciphers the message point into another point as the cipher message.
- ECC Decrypts the cipher message (point) back to the original message point using another key
- ECC can be applied for almost all crypto applications efficiently

Cryptographic Checksums

Mathematical function to generate a set of k bits from a set of n bits (where $k \leq n$).

- k is smaller then n except in unusual circumstances

Example: ASCII parity bit

- ASCII has 7 bits; 8th bit is "parity"
- Even parity: even number of 1 bits
- Odd parity: odd number of 1 bits

Example Use

Bob receives “10111101” as bits.

- Sender is using even parity; 6 bits of 1’s, so character was received correctly
  - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
- Sender is using odd parity; even number of 1 bits, so character was not received correctly

Definition

Cryptographic checksum h: $A \rightarrow B$:

1. For any $x \in A$, $h(x)$ is easy to compute
2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$
3. It is computationally infeasible to find two inputs x, $x' \in A$ such that $x \neq x'$ and $h(x) = h(x')$
   - Alternate form (stronger): Given any $x \in A$, it is computationally infeasible to find a different $x' \in A$ such that $h(x) = h(x')$.

Collisions

If $x \neq x'$ and $h(x) = h(x')$, x and x' are a *collision*

- Pigeonhole principle: if there are n containers for n+1 objects, then at least one container will have 2 objects in it.
- Application: if there are 32 files and 8 possible cryptographic checksum values, at least one value corresponds to at least 4 files

Keys

- Keyed cryptographic checksum: requires cryptographic key
  - DES in chaining mode: encipher message, use last *n* bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
- Keyless cryptographic checksum: requires no cryptographic key
  - MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru

HMAC

Make keyed cryptographic checksums from keyless cryptographic checksums

h keyless cryptographic checksum function that takes data in blocks of *b* bytes and outputs blocks of *l* bytes. *k′* is cryptographic key of length *b* bytes

- If short, pad with 0 bytes; if long, hash to length *b*

ipad is 00110110 repeated *b* times

opad is 01011100 repeated *b* times

HMAC-$h(k, m) = h(k' \oplus opad \,||\, h(k' \oplus ipad \,||\, m))$

- $\oplus$ exclusive or, || concatenation

Key Points

Two main types of cryptosystems: classical and public key

Classical cryptosystems encipher and decipher using the same key

- Or one key is easily derived from the other

Public key cryptosystems encipher and decipher using different keys

- Computationally infeasible to derive one from the other

Cryptographic checksums provide a check on integrity