Cipher Techniques

Road Map

- Basic Terminology
- Cryptosystem
- Classical Cryptography
- Algorithm Types and Modes
- Data Encryption Standard
- Other Stream & Block Ciphers

Basic Terminology

- **plaintext** - the original message
- **ciphertext** - the coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - the study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - the field of both cryptography and cryptanalysis

Cryptosystem

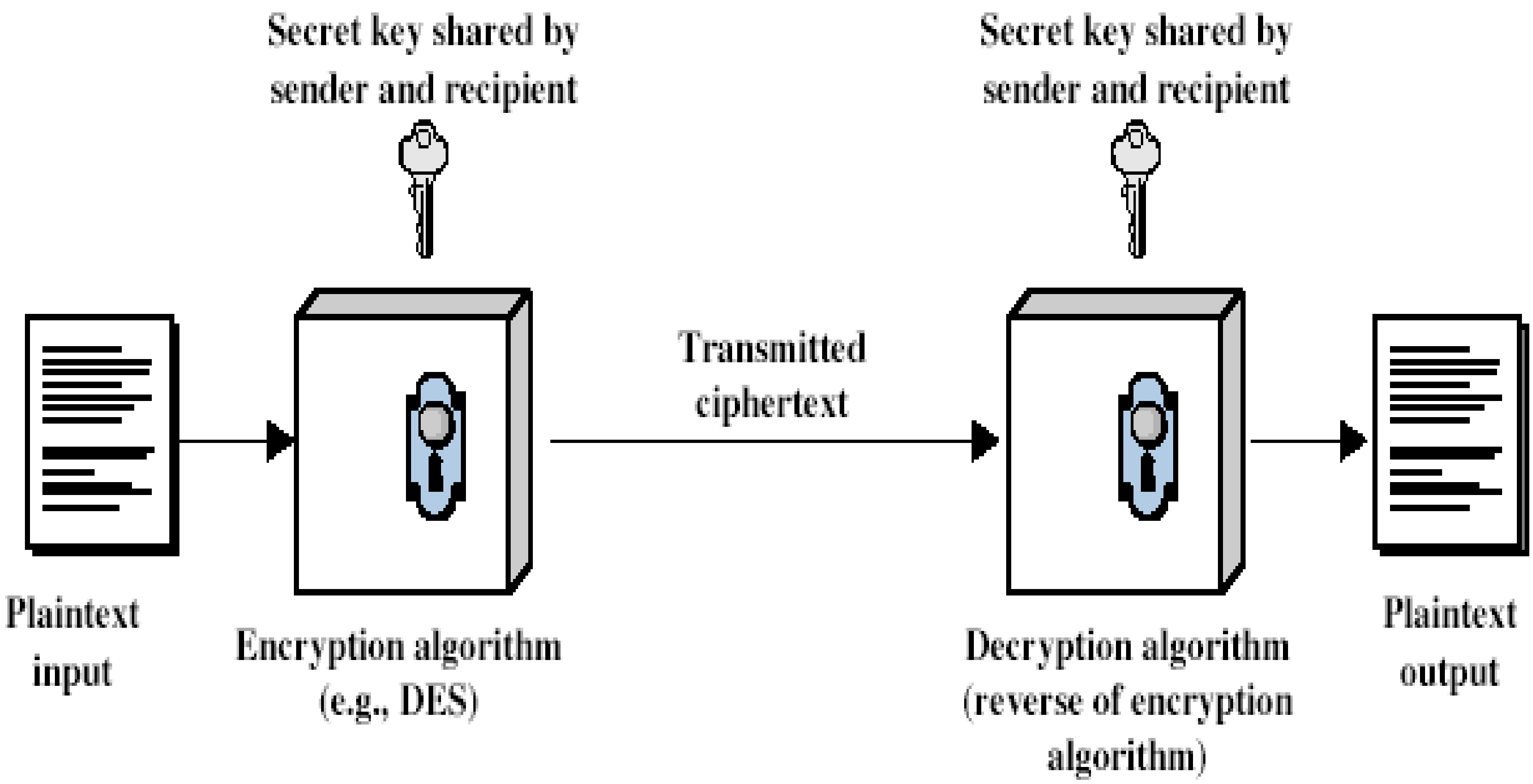
A ***cryptosystem*** is a five-tuple $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$, where the following are satisfied:

1. $\mathcal{P}$ is a finite set of possible ***plaintexts***.
2. $\mathcal{C}$ is a finite set of possible ***ciphertexts***.
3. $\mathcal{K}$, the ***key space***, is a finite set of possible ***keys***
4. $\forall K \in \mathcal{K}$, $\exists E_K \in \mathcal{E}$ (encryption rule), $\exists D_K \in \mathcal{D}$ (decryption rule).
   Each E_K: $\mathcal{P} \to \mathcal{C}$ and D_K: $\mathcal{C} \to \mathcal{P}$ are functions such that $\forall x \in \mathcal{P}$, $D_K(E_K(x)) = x$.

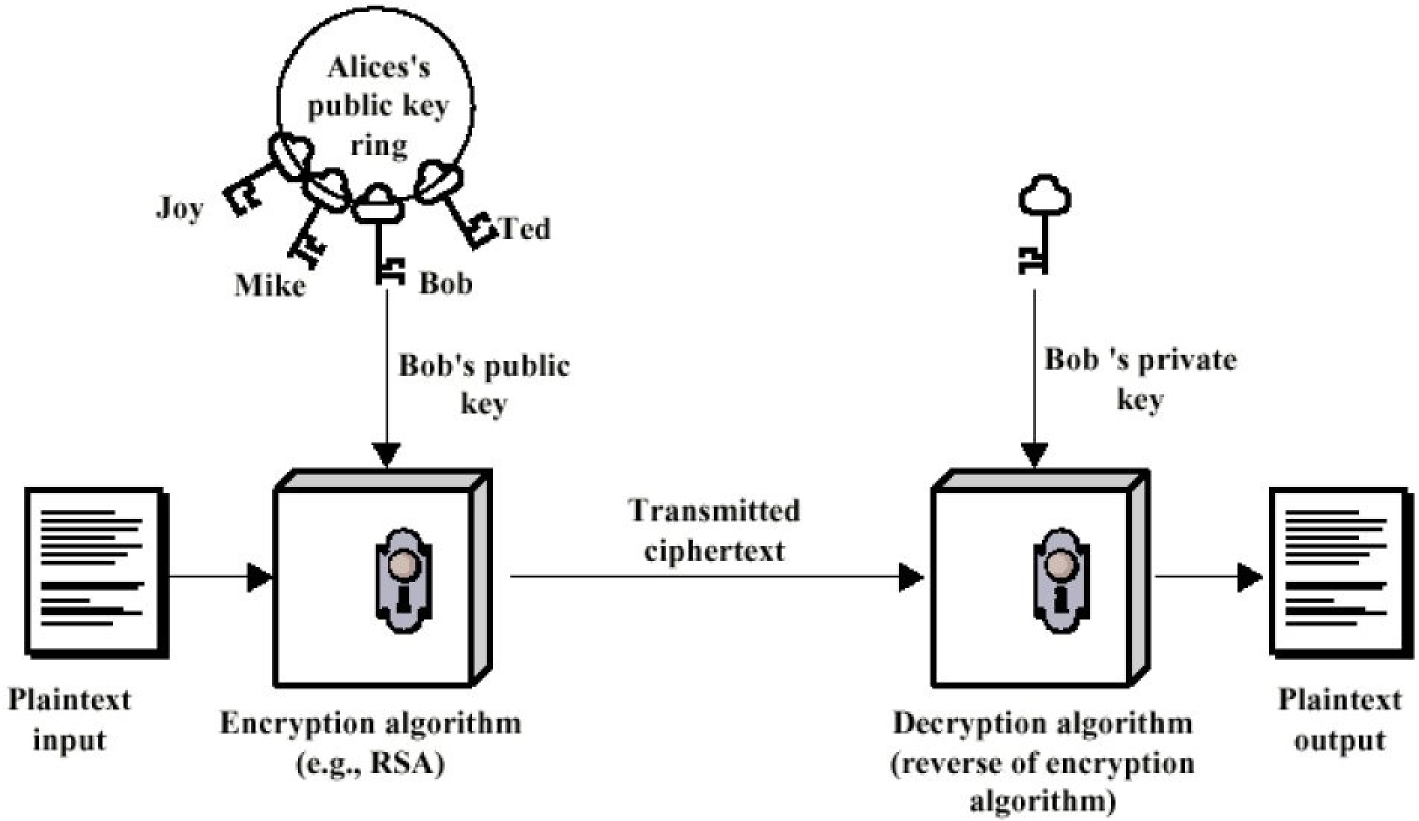
Cryptography

- Cryptography
  - Symmetric / private key / single key
  - Asymmetric / public-key / two - key

Symmetric Cryptography



Asymmetric Cryptography



Requirements

- Two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver

    $Y = E_K(X)$

    $X = D_K(Y)$
- assume encryption algorithm is known
- implies a secure channel to distribute key

Symmetric cryptography

- Transposition Techniques
- Substitution techniques
  - Caesar Cipher
  - Monoalphabetic Cipher
  - Polyalphabethic Cipher
  - Playfair Cipher

Types of Cryptanalytic Attacks

adversary needs strongest attack

- **ciphertext only**
  - only know algorithm / ciphertext, statistical, can identify plaintext, or worse: the key
- **known plaintext**
  - know/suspect plaintext & ciphertext to attack cipher
- **chosen plaintext**
  - select plaintext and obtain ciphertext to attack cipher
- **chosen ciphertext**
  - select ciphertext and obtain plaintext to attack cipher
- **chosen text**
  - select either plaintext or ciphertext to en/decrypt to attack cipher

adversary's attacks can be weaker

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to size of key space
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

Transposition Ciphers

- Consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- write message letters out diagonally over a number of rows
- then read off cipher row by row
- eg. write message out as:

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

- giving ciphertext

```
MEMATRHTGPRYETEFETEOAAT
```

Row Transposition Ciphers

- a more complex scheme
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

```
Key:        4 3 1 2 5 6 7
Plaintext:  a t t a c k p
            o s t p o n e
            d u n t i l t
            w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Classical Substitution Ciphers

- where letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- earliest known substitution cipher
- by Julius Caesar
- first attested use in military affairs
- replaces each letter by 3rd letter after it
- example:

```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

Caesar Cipher

- can define transformation as:

  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
  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

- mathematically give each letter a number

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- then have Caesar cipher as:

  $C = \mathrm{E}(p) = (p + k) \bmod (26)$

  $p = \mathrm{D}(\mathrm{C}) = (\mathrm{C} - k) \bmod (26)$

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improving security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
  - (I and J aren't distinguished)
- fill in letters of keyword (sans duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

```
MONAR
CHYBD
EFGIK
LPQST
UVWXZ
```

Encrypting and Decrypting

- plaintext encrypted two letters at a time:
  1. each letter is replaced by the one in its row in the column of the other letter of the pair, eg. “hs" encrypts to "BP", and “ea" to "IM" or "JM" (as desired). Except when that doesn’t work!
  2. if a pair is a repeated letter, insert a filler like 'X', eg. "balloon" transformed to "ba lx lo on"
  3. if both letters fall in the same row, replace each with letter to right (wrapping back to start from end), eg. “ar" encrypts as "RM"
  4. if both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom), eg. “mu" encrypts to "CM"

Polyalphabetic Ciphers

- another approach to improving security is to use multiple cipher alphabets
- called **polyalphabetic substitution ciphers**
- makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

Vigenère Cipher

- simplest polyalphabetic substitution cipher is the **Vigenère Cipher**
- effectively multiple caesar ciphers
- key is multiple letters long K = k1 k2 ... kd
- i^{th} letter specifies i^{th} alphabet to use
- use each alphabet in turn
- repeat from start after d letters in message
- decryption simply works in reverse

Example

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

```
key:        deceptivedeceptivedeceptive
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Autokey Cipher

- ideally want a key as long as the message
- Vigenère proposed the **autokey** cipher
- with keyword is prefixed to message as key
- knowing keyword can recover the first few letters
- use these in turn on the rest of the message
- but still have frequency characteristics to attack
- eg. given key *deceptive*

```
key:        deceptivewearediscoveredsav
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA
```