ECE 459/559
Secure & Trustworthy Computer Hardware Design

Introduction to Cryptography
Part II

Garrett S. Rose
Spring 2017

THE UNIVERSITY OF TENNESSEE
KNOXVILLE

Recap

- Basics of cryptology:
  - Cryptography – art/science of securing messages
  - Cryptanalysis – art/science of breaking ciphertext
- Introduction to substitution ciphers
  - Characters of plaintext *P* replaced with other letters in same alphabet using encryption E()
  - Ex.: Caesar Cipher – letters shifted by fixed amount
- Caesar cipher is easily attacked via exhaustive search or statistical analysis

Caesar's Problem

- Conclusion: Key is too short
  - 1-character key – monoalphabetic substitution
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well by short key – Looks too much like “regular” English letters
- Solution: May the key longer
  - N-character ($n \geq 2$) – polyalphabetic substitution
  - Makes exhaustive search much more difficult
  - Statistical frequencies concealed much better

Summary

- More on Substitution Ciphers – key of *N* characters
  - Polyalphabetic substitution ciphers
  - Vigenere Tableaux cipher
- Transposition Ciphers
- Basics of Block Ciphers
- Data Encryption Standard (DES)

Polyalphabetic Substitution

- *Somewhat* flatten (diffuse) the frequency distribution of letters by combining high and low distributions
- Example – 2-key substitution:

	A	B	C	D	E	F	G	H	I	J	K	L	M
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	a	f	k	p	u	z	e	j	o	t	y	d	i

- Question: How are Key1 and Key2 defined?

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Polyalphabetic Substitution

	A	B	C	D	E	F	G	H	I	J	K	L	M
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	a	f	k	p	u	z	e	j	o	t	y	d	i

- Answer:
  - Key1 – start at 'a', skip 2, take next, skip 2, … (circular)
  - Key2 – start at 'n', (2nd half of alphabet), skip 4, take next, skip 4, take next, … (circular)

Polyalphabetic Substitution

	A	B	C	D	E	F	G	H	I	J	K	L	M
Key1:	a	d	g	j	m	p	s	v	y	b	e	h	k
Key2:	n	s	x	c	h	m	r	w	b	g	l	q	v

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key1:	n	q	t	w	z	c	f	i	l	o	r	u	x
Key2:	a	f	k	p	u	z	e	j	o	t	y	d	i

- Plaintext: `TOUGH STUFF`
- Ciphert: `ffirv zfjpm`

  use n (=2) keys in turn for consecutive P characters in P
- Note:
  - Different characters mapped into same: T, O → f
  - Same characters mapped to different: F → p, m
  - 'f' most frequent in C (0.30); in English: f(f) = 0.02 << f(e) = 0.13

Vigenère Tableaux

	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	p
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	0
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	1
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	2
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	3
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	4
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	5
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	6
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	7
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	8
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	9
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	10
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	11
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	12
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	13
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	14
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	15
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	16
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	17
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	18
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	19
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	20
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	21
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	22
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	23
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	24
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	25

Note:

Row A – shift 0 (a → a); Row B – shift 1 (a → b); … Row Z – shift 25 (a → z)

Vigenère Tableaux Example

- Key: `EXODUS`
- Plaintext P: `YELLOW SUBMARINE FROM YELLOW RIVER`
- Extended keyword (re-applied to mimic words in P):

  `YELLOW SUBMARINE FROM YELLOW RIVER`

  `EXODUS EXODUSEXO DUSE XODUSE XODUS`

- Ciphertext:

  `cbxoio wlppujmks ilgq vsofhb owyyj`

Vigenère Tableaux Example

- Key: `EXODUS`
- Plaintext P: `YELLOW SUBMARINE FROM YELLOW RIVER`
- Extended keyword (re-applied to mimic words in P):

  `YELLOW SUBMARINE FROM YELLOW RIVER`

  `EXODUS EXODUSEXO DUSE XODUSE XODUS`
- Ciphertext:

  `cbzoio wlppujmks ilgq vsofhb owyyj`
- Explanation:
  - Character from P indexes row
  - Character from extended key indexes column

  e.g.: row Y and column e → 'c'
row E and column x → 'b'
row L and column o → 'z'

Transposition Ciphers

- Rearrange letters in plaintext to produce ciphertext
- Examples 1a and 1b: Columnar transposition
  - Plaintext: HELLO WORLD
  - Transposition onto:

(a) 3 columns:

```
HEL
LOW
ORL
DXX (xx – padding)
```

(b) 2 columns:

```
HE
LL
OW
OR
LD
```

- Ciphertext (read column by column):
  - (a) hlodeorxlwlx (b) hloolelwrd
- Keys: (a) key = 3; (b) key = 2

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Transposition Ciphers

- Example 2: Rail-Fence Cipher
  - Plaintext: HELLO WORLD
  - Transposition into 2 rows (rails) column by column:

```
HLOOL
ELWRD
```

- Ciphertext: (familiar?)

```
hloolelwrd
```

- Key: Number of rails key = 2

Product Ciphers

- A.k.a. combination ciphers
- Built of multiple blocks, each is:
  - Substitution

    or:
  - Transposition
- Example: two-block product cipher
  - $E_2(E_1(P, K_{E1}), K_{E2})$
- Product cipher might *not* necessarily be stronger than its individual components used separately!
  - Might not be even as strong as individual components

Criteria for "Good" Ciphers

- "Good" depends on intended application
  - Substitution
    - Ciphertext C hides chars of plaintext P
    - If key > 1 char, C dissipates high frequency chars
  - Transposition
    - C scrambles text => hides n-grams for n > 1
  - Product ciphers
    - Can do all of the above
  - What is more important for your app?

    What facilities available to sender/receiver?
    - E.g., no supercomputer support on the battlefield

Criteria for "Good" Ciphers

- Commercial Principles of Sound Encryption Systems
  - Sound mathematics
    - Proven vs. not broken so far
  - Verified by expert analysis
    - Including outside experts
  - Stood the test of time
    - Long-term success is not a guarantee
    - Still… Flaws in many E's discovered soon after release
- Examples of popular commercial encryption:
  - DES, RSA, AES

DES = Data Encryption Standard
RSA = Rivest-Shamir-Adelman
AES = Advanced Encryption Standard

Stream and Block Ciphers

- Stream Ciphers
- Problems with stream ciphers
- Block ciphers
- Pros / cons for stream and block ciphers

Stream Ciphers

- <u>*Stream Cipher – 1 character from P → 1 character for C*</u>
- Example: Polyalphabetic cipher
  - P and K (repeated 'EXODUS'):

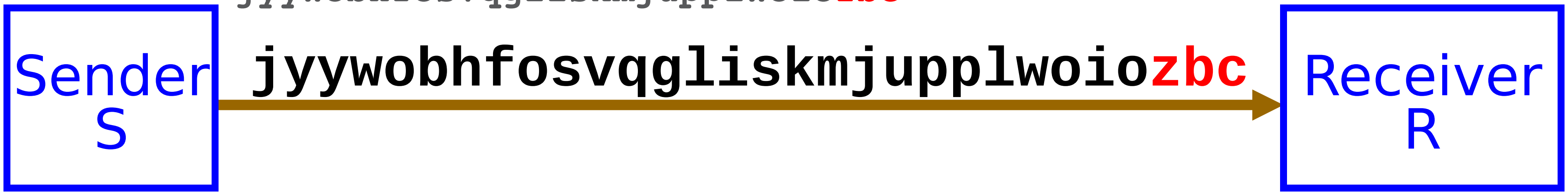
    YELLOWSUBMARINEFROMYELLOWRIVER

    EXODUSEXODUSEXODUSEXODUSEXODUS
  - Encryption:

    (1) E(Y, E) → c (2) E(E, X) → b (3) E(L, O) → z
  - C (using Vigenere Tableaux):

    cbzoiowlppujmksilgqvsofhbowyyj
  - C sent in left to right order:

    jyywobhfosvqgliskmjupplwoiozbc



Stream Ciphers

- Example: Polyalphabetic cipher
  - C as received (right to left order):

  - C and K (repeated 'EXODUS'):

    cbzoiowlppujmksilgqvsofhbowyyj

    EXODUSEXODUSEXODUSEXODUSEXODUS

  - Decryption:

    (1) D(c, E) → Y (2) D(b, X) → E (3) D(z, O) → L

  - Decrypt C:

    YEL ...

Problems with Stream Ciphers

- *Dropping a character from key results in wrong decryption*
- Example:
  - P and K (repeated 'EXODUS'):

    YELLOWSUBMARINEFROMYELLOWRIVER

    EODUSEXODUSEXODUSEXODUSEXODUSE **(missing first X)**
  - Encryption:

    (1) E(Y, E) → c (2) E(E, O) → s (3) E(L, D) → o

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Problems with Stream Ciphers

- C as received (in right to left order):

  ... osc

  - C and correct K ('EXODUS' for decryption):

    cso ...

    EXO ...

  - Decryption:

    (1) D(c, E) → Y (2) D(s, X) → V (3) D(o, O) → A

  - Decrypted:

    YVA ... → Wrong!

    - We know it's wrong, receiver might not know, yet!

Stream Cipher Problem Could be Recoverable…

- If receiver had more characters decoded, could detect that sender dropped a key character and recover
  - E.g., suppose receiver decoded:

    `YELLOW SUBMAZGTR ...`

  - Could guess, that 2nd word should really be

    `SUBMARINE`

  - Receiver would know that sender dropped a character after "SUBMA"
  - Could go back 4 characters and recalibrate… essentially "resynchronize" the decruption

Block Ciphers

- Can do better than relying on recovery for stream ciphers
  - Solution: Block Ciphers
- Block cipher:
  1 *block* of characters from P → 1 *block* of characters for C
  - Example of block cipher: columnar transposition
  - Block size = “o(message length)” (informally)

Block Ciphers

- Can do better than relying on recovery for stream ciphers
  - Solution: Block Ciphers
- Block cipher:
  1 *block* of characters from P → 1 *block* of characters for C
  - Example of block cipher: columnar transposition
  - Block size = "o(message length)"   (informally)

Block Ciphers

- Why block size = “o(message length)” ?
- Receiver must wait for almost entire C to come through before decoding some characters near beginning of C
- E.g., for P = “HELLO WORLD”, block size is “o(10)”
- Suppose Key = 3 (3 columns):

```
HEL
LOW
ORL
DXX
```

- C as sent:

Block Ciphers – Example

- C as received (right to left order): lwlxroedolh
- Receiver knows: K=3, block size = 12
  => 4 rows
  - Knows characters will be sent in the order: 1st-4th-7th-10th—2nd-5th-8th-11th—3rd-6th-9th-12th
- Receiver must wait for at least:
  - 1 chars of C to decode 1st char of P ('h')
  - 5 chars of C to decode 2nd char of P ('he')
  - 9 chars of C to decode 3rd, 4th, & 5th chars of P ('hello')
  - 10 chars of C to decode 6th, 7th, & 8th chars of P ('hello wor')
  - etc.

123
456
789
abc

a=10
b=11
c=12

Block Ciphers

- Informally, we might call ciphers such as above example columnar transposition cipher “weak-block” ciphers
  - Receiver can get some (even most) but not all chars of P before entire C is received
  - <u>Stronger</u>: receiver must wait for entire block to get any of P

- For “weak-block cipher, receiver must wait for at least:
  - 1 chars of C to decode 1st char of P ('h')
  - 5 chars of C to decode 2nd char of P ('he')
  - 9 chars of C to decode 3rd, 4th, & 5th chars of P ('hello')
  - 10 chars of C to decode 6th, 7th, & 8th chars of P ('hello wor')
  - etc.

Pros / Cons for Stream Ciphers

- ✔ Low delay for decoding individual symbols
  - Can decode as soon as received
- ✔ Low error propagation
  - Error in E(c1) does not affect E(c2)
- ✖ Low diffusion
  - Each char separately encoded =>
    carries over its frequency information (1 to 1 correspondence)
- ✖ Susceptibility to malicious insertion / modification
  - Adversary can fabricate new message from pieces of broken messages, even if E unknown

Pros / Cons for Block Ciphers

- ✔ High diffusion
  - Frequency of char from P diffused over (a few chars of) a block of C
- ✔ Immune to insertion
  - Impossible to insert a char into a block without easy detection (block size would change)
  - Impossible to modify and char in a block without easy detection (if checksums are used)
- ✖ High delay for decoding individual characters
  - For example, 'hello worldxx' above – some E can't decode even the 1st char before all chars of a block are received
- ✖ High error propagation
  - Error affects the block, not just single char

DES
(Data Encryption Standard)

Background & History of DES

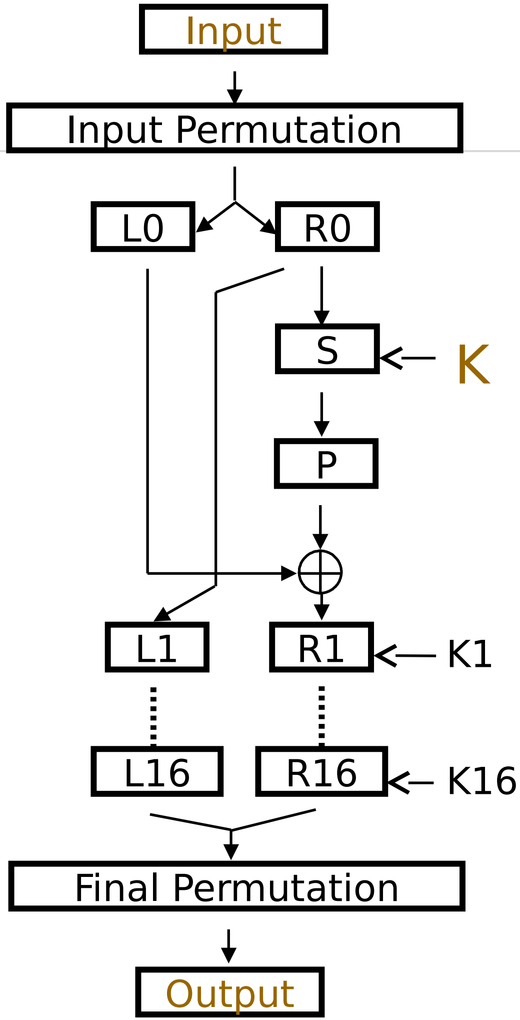
- Early 1970's – NBS (National Bureau of Standards) recognized general public need for secure crypto system
  - NBS – part of US government
  - Now: NIST – National Institute of Standards & Technology
- Idea: "Encryption for the masses"
- Existing US government crypto were not meant to be made public
  - E.g., DoD, State Department
- Problems with proliferation of commercial encryption devices
  - Incompatible
  - Not extensively tested by independent body

Background & History of DES

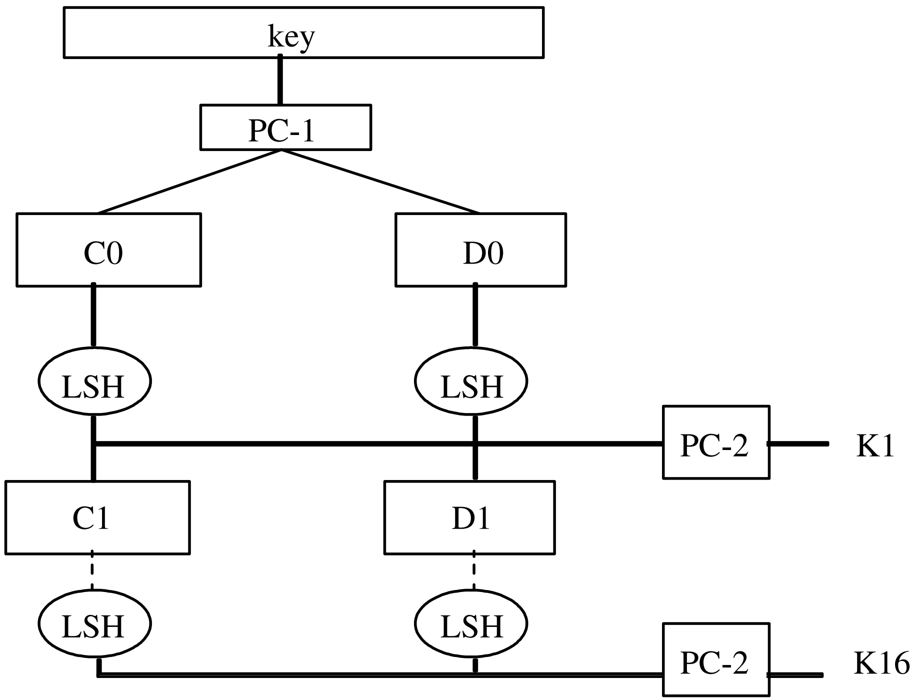
- 1972 – NBS calls for proposals for a public crypto system
  - Criteria: Highly secure / easy to understand / publishable / available to all / adaptable to diverse applications / economical / efficient to use / able to be validated / exportable
  - In truth: Not *too* strong (appease NSA, etc.)
- 1974 – IBM proposed its Lucifer
  - DES was ultimately based on Lucifer
  - Tested by NSA (National Security Agency) and general public
- Nov. 1976 – DES adopted as US standard for sensitive but unclassified data and communication
  - Later adopted by ISO (International Standards Organization)
  - Official name: DEA – Data Encryption Algorithm / DEA-1 abroad

Basic Structure of DES

- Input: 64 bits (a block)
- L_i / R_i – left/right half of input block for iteration i (32 bits each) – subject to substitution S and permutation P
- K – user-supplied key
- K_i – round key:
  - 56 bits + 8 unused (unused for E but often used for error checking)
- Output: 64 bits (a block)
- Note: R_i becomes L_{i+1}
- All basic ops simple logic:
  - Left shift, XOR



Generation of Round Keys



- Key – user-supplied key (input)
- PC-1, PC-2 – permutation tables
  - PC-2 also extracts 48 of 56 bits
- K1 – K16 – round keys (outputs)
  - Length(K_i) = 48
- Ci / Di – confusion / diffusion (?)
- LSH – left shift (rotation) tables

Problems with DES

- Diffie, Hellman 1977 prediction: “In a few years, technology would allow DES to be broken in days”
- Key length is fixed at 56
  - 2^{56} keys ~ 10^{15} keys
  - “Becoming” too short for faster computers
    - 1997: 3,500 machines could crack it in 4 months
    - 1998: special “DES cracker” HW cracked it in 4 days
- Design decisions not public
  - Suspected of having backdoors
  - Speculation: To facilitate government access?

Double DES

- Use double DES encryption: C = E(K2, E(K1, P))
- Expected to multiply difficulty of breaking the encryption
  - Not true!
  - In general, 2 encryptions are not better than one
  - Only doubles the attacker's work

Triple DES

- Not exactly C = E(K3, E(K2, E(K1, P)))
- A few tricks are used:
  - D not E in the second step, K1 used twice (in steps 1 & 2)
- It is:

  C = E(K1, D(K2, E(K1, P)))

  P = D(K1, E(K2, D(K1, C)))
- Doubles the effective key length
- 112-bit is quite strong, even for today's computers

Security of DES

- So, is DES insecure?
- No, not yet
  - 1997 attack required a lot of cooperation
  - The 1998 special-purpose machine still very expensive
  - Triple DES still beyond reach of these two attacks
- But …
  - In 1995, NIST (formerly NBS) began a search for a new, stronger encryption standard
  - Led to the AES contest