CIS 551 / TCOM 401

Computer and Network Security

Spring 2006

Lecture 8

Announcements

- MIDTERM HAS BEEN POSTPONED:
  - Midterm is now one week from today. (2/14/2006)
  - If this causes problems for you, see me after class.

Κρυπτογραφία (Cryptography)

- From the Greek "kryptos" and "graphia" for “secret writing”

- Confidentiality
  - Obscure a message from eaves-droppers
- Integrity
  - Assure recipient that the message was not altered
- Authentication
  - Verify the identity of the source of a message
- Non-repudiation
  - Convince a 3rd party that what was said is accurate

Terminology

Plaintext $\xrightarrow{\text{encryption}}$ Ciphertext $\xrightarrow{\text{decryption}}$ Plaintext

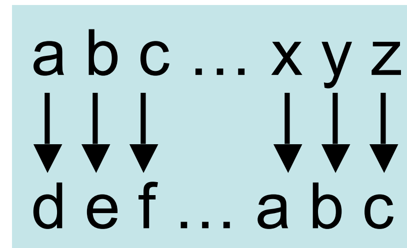
- Cryptographer
  - Invents cryptosystems
- Cryptanalyst
  - Breaks cryptosystems
- Cryptology
  - Study of crypto systems
- Cipher
  - Mechanical way of encrypting text
- Code
  - Semantic translation: “eat breakfast tomorrow” = “attack on Thursday” (or use Navajo!)

Kinds of Cryptographic Analysis

- Goal is to recover the key (& algorithm)
- Ciphertext only attacks
  - No information about content or algorithm
  - Very hard
- Known Plaintext attacks
  - Full or partial plaintext available in addition to ciphertext
- Chosen Plaintext attacks
  - Know which plaintext has been encrypted
- Algorithm & Ciphertext attacks
  - Known algorithm, known ciphertext, recover key

The Caesar Cipher

- Purportedly used by Julius Caesar (c. 75 B.C.)
  - Add 3 mod 26



- Advantages
  - Simple
  - Intended to be performed in the field
  - Most people couldn't read anyway
- Disadvantages
  - Violates "no security through obscurity"
  - Easy to break (why?)

Monoalphabetic Ciphers

- Also called *substitution* ciphers
- Separate *algorithm* from the *key*
  - Add N mod 26
  - rot13 = Add 13 mod 26

- General monoalphabetic cipher
  - Arbitrary permutation π of the alphabet
  - Key is the permutation

$$\begin{array}{cccc} a & b & c & d \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \pi(a) & \pi(b) & \pi(c) & \pi(d) \end{array}$$

Example Cipher

	`a b c d e f g h i j k l ...`
π	`z d a n c e w i b f g h ...`

Plaintext: **`he lied`**

Ciphertext: **`ic hbcn`**

Cryptanalysis of Monoalphabetic Ciphers

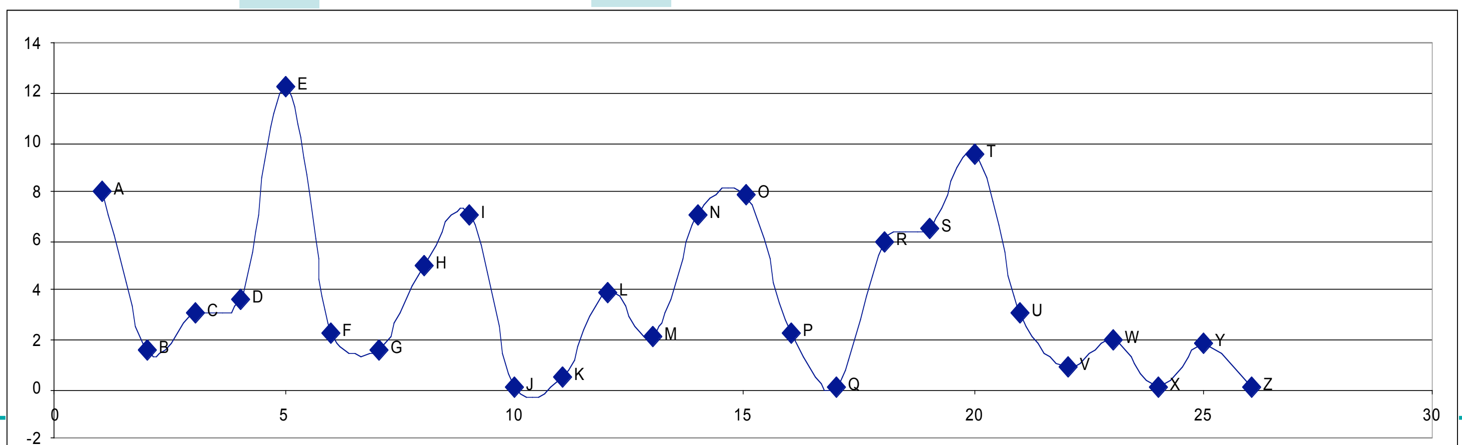
- Brute force attack: try every key
  - N! Possible keys for N-letter alphabet
  - 26! ≈ 4 x 10^{26} possible keys
  - Try 1 key per μsec … 10 trillion years

- …but (!) monoalphabetic ciphers are *easy* to solve

  - One-to-one mapping of letters is bad
  - Frequency distributions of common letters

Order & Frequency of Single Letters

E	12.31%	L	4.03%	B	1.62%
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		



Monoalphabetic Cryptanalysis

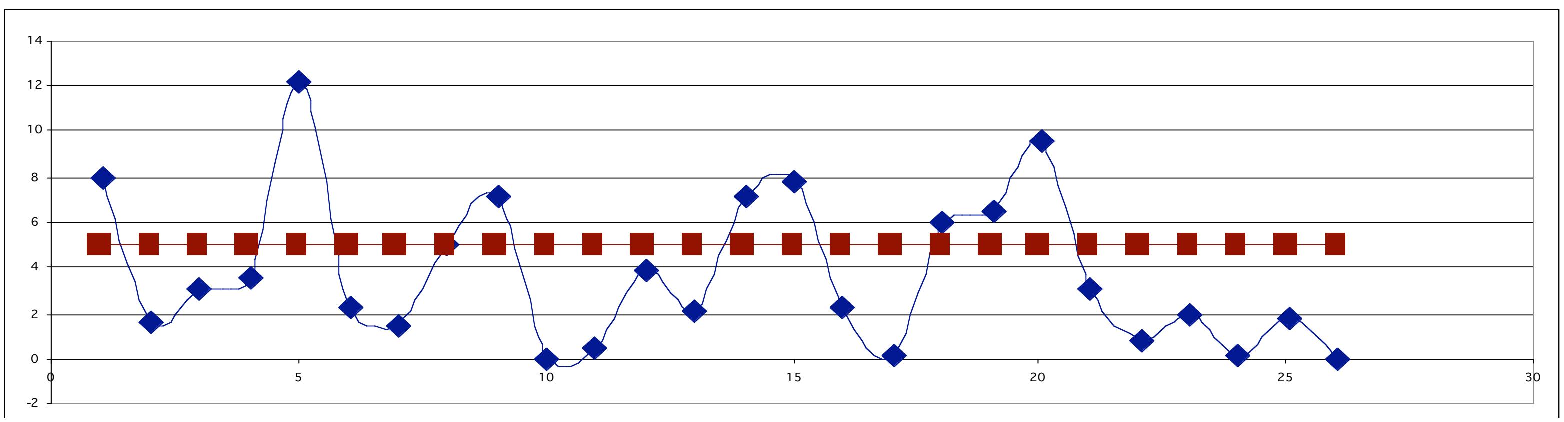
- Count the occurrences of each letter in the cipher text
- Match against the statistics of English

- Most frequent letter likely to be “e”
- 2^{nd} most frequent likely to be “t”
- etc.

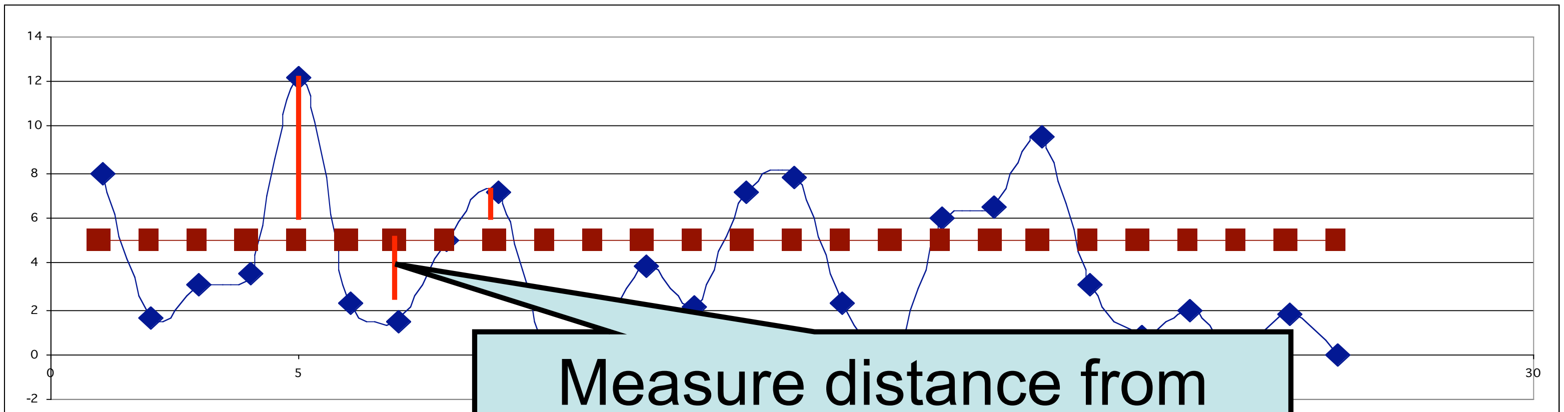
- Longer ciphertext makes statistical analysis more likely to work…

Desired Statistics

- Problems with monoalphabetic ciphers
  - Frequency of letters in ciphertext reflects frequency of plaintext
- Want a single plaintext letter to map to multiple ciphertext letters
  - “e” $\rightarrow$ “x”, “c”, “w”
- Ideally, ciphertext frequencies should be flat



Variance: Measure of “roughness”



Measure distance from “flat” dist.

$$\text{Var} = \sum_{\alpha = a}^{\alpha = z} (\text{prob}(\alpha) - 1/26)^2$$

$$= \ldots$$

$$= \left(\sum_{\alpha = a}^{\alpha = z} \text{prob}(\alpha)^2 \right) - 1/26$$

Polyalphabetic Substitutions

- Pick k substitution ciphers
  - $\pi_1\ \pi_2\ \pi_3\ \ldots\ \pi_k$
  - Encrypt the message by rotating through the k substitutions

m	**e**	**s**	**s**	**a**	**g**	**e**
$\pi_1(\mathbf{m})$	$\pi_2(\mathbf{e})$	$\pi_3(\mathbf{s})$	$\pi_4(\mathbf{s})$	$\pi_1(\mathbf{a})$	$\pi_2(\mathbf{g})$	$\pi_3(\mathbf{e})$
q	**a**	**x**	**o**	**a**	**u**	**v**

- Same letter can be mapped to multiple different ciphertexts
  - Helps smooth out the frequency distributions
  - *Diffusion*

Perfect Substitution Ciphers

$$
\begin{array}{r l}
 & p_1\ p_2\ p_3\ \ldots\ p_n \\
\oplus & \underline{b_1\ b_2\ b_3\ \ldots\ b_n} \\
 & c_1\ c_2\ c_3\ \ldots\ c_n
\end{array}
$$

- Choose a string of random bits the same length as the plaintext, XOR them to obtain the ciphertext.
- Perfect Secrecy
  - Probability that a given message is encoded in the ciphertext is unaltered by knowledge of the ciphertext
  - Proof: Give me any plaintext message and any ciphertext and I can construct a key that will produce the ciphertext from the plaintext.

One-time Pads

- Another name for Perfect Substitution
- Actually used by US agents in Russia
  - Physical pad of paper
  - List of random numbers
  - Pages were torn out and destroyed after use
- Vernam Cipher
  - Used by AT&T
  - Random sequence stored on punch tape
- Not practical for computer security…

Problems with “Perfect” Substitution

- Key is the same length as the plaintext
  - Sender and receiver must agree on the same random sequence
  - Not any easier to transmit key securely than to transmit plaintext securely
- Need to be able to generate many truly random bits
  - Pseudorandom numbers generated by an algorithm aren’t good enough for long messages
- Can’t reuse the key

Computational Security

- Perfect Ciphers are *unconditionally secure*
  - No amount of computation will help crack the cipher (i.e. the *only* strategy is brute force)

- In practice, strive for *computationally security*
  - Given enough power, the attacker could crack the cipher (example: brute force attack)
  - But, an attacker with only *bounded resources* is extremely unlikely to crack it
  - Example: Assume attacker has only polynomial time, then encryption algorithm that can't be inverted in less than exponential time is secure.

Kinds of Industrial Strength Crypto

- Shared Key Cryptography
- Public Key Cryptography
- Cryptographic Hashes

- All of these aim for computational security
  - Not all methods have been proved to be intractable to crack.

Shared Key Cryptography

- Sender & receiver use the same key
- Key must remain private
- Also called *symmetric* or *secret key* cryptography
- Often are *block-ciphers*
  - Process plaintext data in blocks
- Examples: DES, Triple-DES, Blowfish, Twofish, AES, Rijndael, …

Shared Key Notation

- Encryption algorithm

  E : key x plain → cipher

  Notation: K{msg} = E(K, msg)

- Decryption algorithm

  D : key x cipher → plain

- D inverts E

  D(K, E(K, msg)) = msg

- Use capital “K” for shared (secret) keys

- Sometimes E is the same algorithm as D

Secure Channel: Shared Keys

Alice

Bart

K_{AB}\{Hello!\} →

← K_{AB}\{Hi!\}

K_{AB}

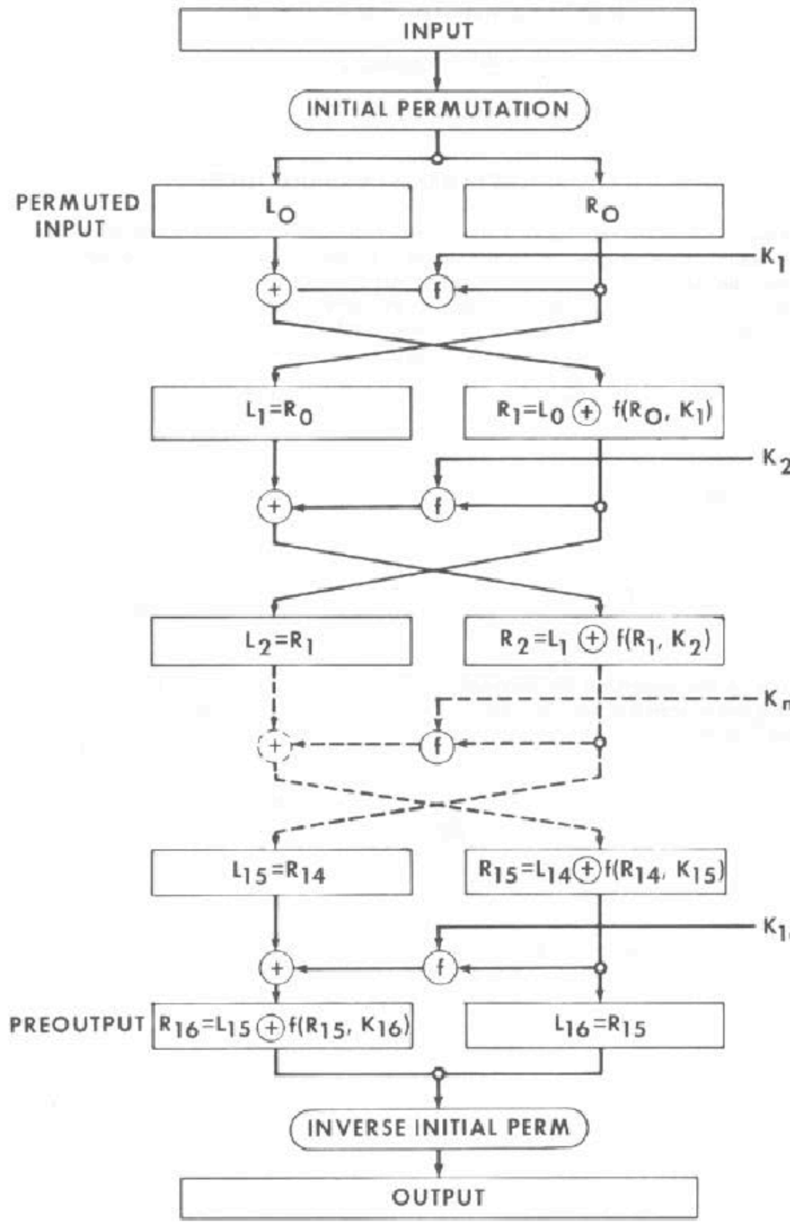
K_{AB}

Data Encryption Standard (DES)

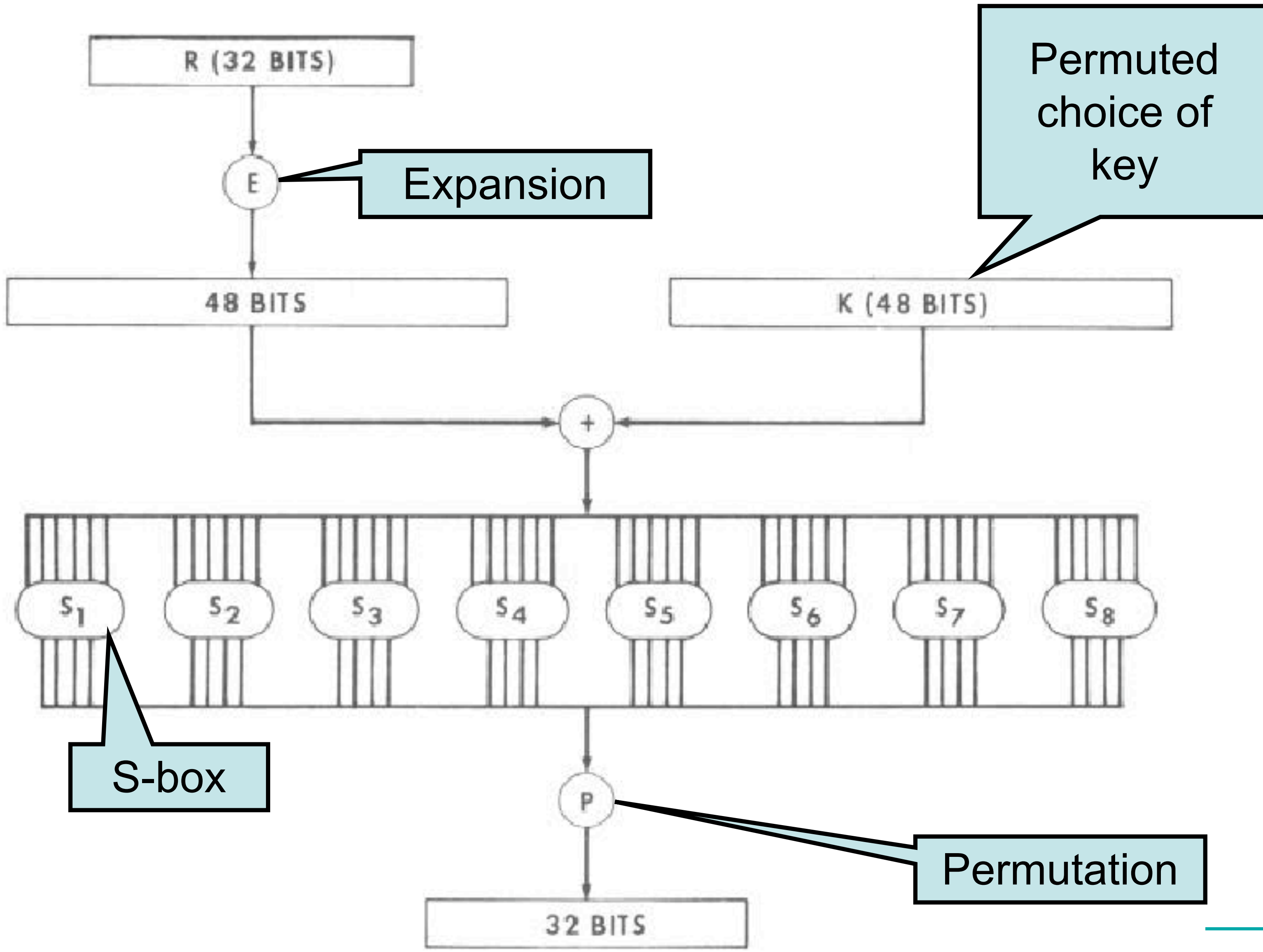
- Adopted as a standard in 1976
- Security analyzed by the National Security Agency (NSA)
  - http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
- Key length is 56 bits
  - padded to 64 bits by using 8 parity bits
- Uses simple operators on (up to) 64 bit values
  - Simple to implement in software or hardware

- Input is processed in 64 bit blocks
- Based on a series of 16 *rounds*
  - Each cycle uses permutation & substitution to combine plaintext with the key

DES Encryption

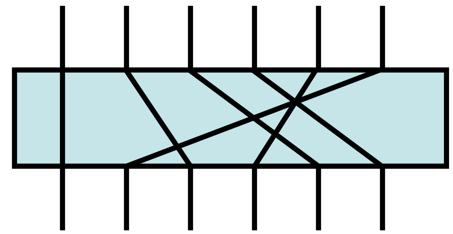


One Round of DES (f of previous slide)

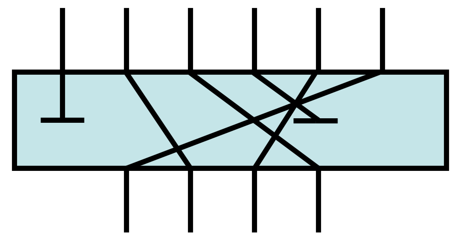


Types of Permutations in DES

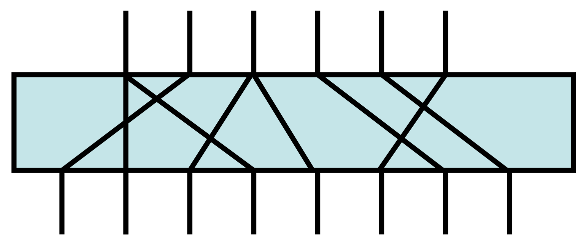
Permutation



Permuted
Choice



Expansion
Permutation



DES S-Boxes

- Substitution table
- 6 bits of input replaced by 4 bits of output
- Which substitution is applied depends on the input bits

- Implemented as a lookup table
  – 8 S-Boxes
  – Each S-Box has a table of 64 entries
  – Each entry specifies a 4-bit output

DES Decryption

- Use the same algorithm as encryption, but use $k_{16} \dots k_1$ instead of $k_1 \dots k_{16}$
- Proof that this works:
  - To obtain round j from j-1:

$$(1) \quad L_j = R_{j-1}$$
$$(2) \quad R_j = L_{j-1} \oplus f(R_{j-1}, k_j)$$

  - Rewrite in terms of round j-1:

$$(1) \quad R_{j-1} = L_j$$
$$(2) \quad L_{j-1} \oplus f(R_{j-1}, k_j) = R_j$$
$$L_{j-1} \oplus f(R_{j-1}, k_j) \oplus f(R_{j-1}, k_j) = R_j \oplus f(R_{j-1}, k_j)$$
$$L_{j-1} = R_j \oplus f(R_{j-1}, k_j)$$
$$L_{j-1} = R_j \oplus f(L_j, k_j)$$

Problems with DES

- Key length too short: 56 bits
  - www.distributed.net broke a DES challenge in 1999 in under 24 hours (parallel attack)
- Other problems
  - Bit-wise complementation of key produces bit-wise complemented ciphertext
  - Not all keys are good (half 0’s half 1’s)
  - Differential cryptanalysis: Carefully choose pairs of plaintext that differ in particular known ways (e.g. they are complements)

Block Cipher Performance

Algorithm	Key Length	Block Size	Rounds	Clks/Byte
Twofish	variable	128	16	18.1
Blowfish	variable	64	16	19.8
Square	128	128	8	20.3
RC5-32/16	variable	64	32	24.8
CAST-128	128	64	16	29.5
DES	56	64	16	43
Serpent	128,192,256	128	32	45
SAFER (S)K-128	128	64	8	52
FEAL-32	64, 128	64	32	65
IDEA	128	64	8	74
Triple-DES	112	64	48	116

Advanced Encryption Standard (AES)

- National Institute of Standards & Technology NIST
  - Computer Security Research Center (CSRC)
  - http://csrc.nist.gov/
  - http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
- Uses the Rijndael algorithm
  - Invented by Belgium researchers
    Dr. Joan Daemen & Dr. Vincent Rijmen
  - Adopted May 26, 2002
  - Key length: 128, 192, or 256 bits
  - Block size: 128, 192, or 256 bits

Problems with Shared Key Crypto

- Compromised key means interceptors can decrypt any ciphertext they've acquired.
  - Change keys frequently to limit damage
- Distribution of keys is problematic
  - Keys must be transmitted securely
  - Use couriers?
  - Distribute in pieces over separate channels?
- Number of keys is $O(n^2)$ where n is # of participants
- Potentially easier to break?