

Announcements

- Reading Chapter 19

Types of Software Threats

- Trojan Horse

- a program that looks like a normal program
- for example a login program written by a user
- UNIX example: never put “.” early in your path

- Trap door

- hole left by the programmers to let them into the system
- “system” password set to a default value by the vendor

- Worms

- programs that clone themselves and use resources
- Internet worm:
 - exploited several bugs and “features” in UNIX
 - .rhosts files
 - bug in finger command (overwrite strings)
 - sendmail “debug” mode to run commands

Viruses

- Most common on systems with little security
 - easy to write to boot blocks, system software
 - never run untrusted software with special privileges
- Possible to write system independent viruses
 - MS Word virus
 - uses macros to call into the OS

Access Matrix

- **Abstraction of protection for objects in a system.**
 - Rows are domains (users or groups of users)
 - Columns are objects (files, printers, etc.)
 - Items are methods permitted by a domain on an objects
 - read, write, execute, print, delete, ...
- **Representing the Table**
 - simple representation (dense matrix) is large
 - sparse representation possible: each non-zero in the matrix
 - observation: same column used frequently
 - represent groups of users with a name and just store that
 - create a default policy for some objects without a value
- **Revocation of access**
 - when are access rights checked?
 - Selective revocation vs. global

Access Matrix

	F1	F2	F3	Laser Printer	
D1	read		execute		
D2			execute	print	
D3	read, write		execute		
D4			execute		
D5		delete			

- Rows represent users or groups of users
- Columns represent files, printers, etc.

Capabilities

- Un-forgeable Key to access something
- Implementation: a string
 - I.e. a long numeric sequence for a copier)
- Implementation: A protected memory region
 - tag memory (or procedures) with access rights
 - example - x86 call gate abstraction
 - permit rights amplification

Monitoring

- Record (log) significant events
 - attempts to login to the system
 - changes to selected files or directories
- Possible to compromise the log
 - the user or software breaking in could delete all or part of the logs
 - could record logs to non-erasable storage
 - have a line printer attached to the machine
 - use WORM drives
 - send data to a secure remote host

Add new slides about

- Auditing, tripwire

Encryption: protecting info from being read

- Given a message m
 - use a key k , and function E_k to compute $E_k(m)$
 - store or send only $E_k(m)$
 - use a second key k' and function $D_{k'}$, such that
 - $D_{k'}(E_k(m)) = m$
 - E_k and $D_{k'}$ need not be kept a secret
- If $k=k'$ it's called private key encryption
 - need to keep k secret
 - example DES
- if $k \neq k'$, it's called public key encryption
 - need only keep one of them secret
 - if k' is secret, anyone can send a private message
 - if k is secret, it is possible to “sign” a message
 - still need a way to authenticate k or k' for a user
 - example RSA

Transposition Cipher

- Block of text is used to break up digrams
- To Break:
 - each letter is itself, so normal distribution of letters is seen
 - guess number of columns (verify with known plaintext)
 - order columns using trigram frequency

<u>M</u>	<u>E</u>	<u>G</u>	<u>A</u>	<u>B</u>	<u>U</u>	<u>C</u>	<u>K</u>
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwo

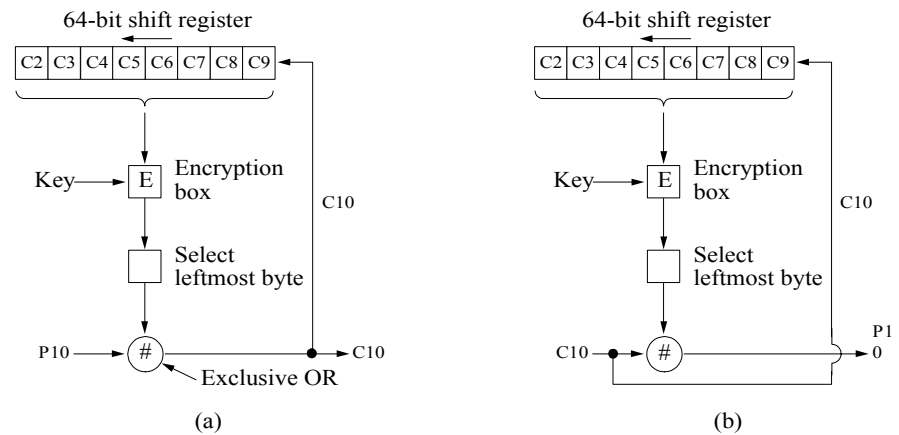
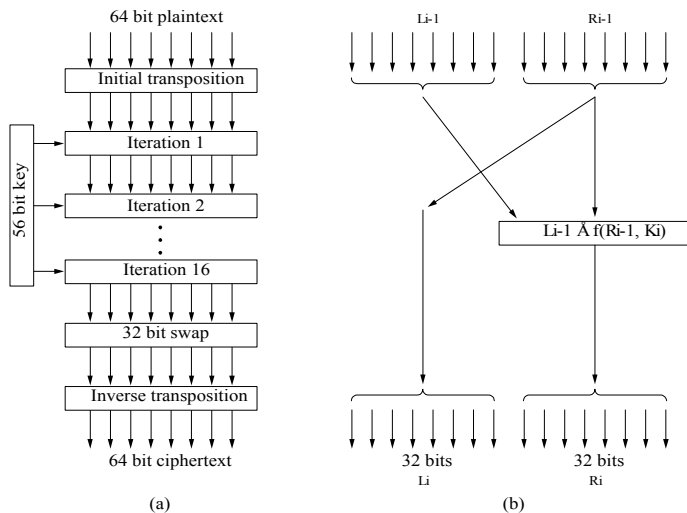
Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMAI
ESILYNTWRNNTSOWDPAEDOBUEIRICX

From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

DES

- Block cipher: uses 56 bit keys, 64 bits of data
- Uses 16 stages of substitution
- Variations
 - cipher block chaining: xor output of block n with into block n+1
 - cipher feedback mode: use 64bit shift register
 - can produce one byte at a time



From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

One Time Pad

- Key Idea: randomness in key
- Create a random string as long as the message
 - each party has the pad
 - xor each bit of the message with the a bit of the key
- Almost impossible to break
- Some practical problems
 - need to ensure key is not captured
 - a one bit drop will corrupt the rest of the message

Add slide about SSL

Add Slide about

- Protection Domains (18.2)
- Unix
 - Setuid
 - Daemon proceses