CNS Lecture 4

Random Numbers
Steganography
Classical cryptography

Assignments ncp

Hash/hmac values are binary data, you can't use strcpy() or strcmp(), use memcpy()

In the news

Widely Deployed Software
(1) CRITICAL: Apple QuickTime Multiple Vulnerabilities
(2) HIGH: Adobe Flash Player Multiple Vulnerabilities
(3) HIGH: Microsoft Internet Explorer Compressed Content Heap Overflow(MS06-042)
(4) HIGH: Microsoft Pragmatic General Multicast Buffer Overflow (MS06-052)
(5) HIGH: Microsoft Publisher File Parsing Buffer Overflow (MS06-054)
(6) MODERATE: Cisco IOS VTP Multiple Vulnerabilities
(7) MODERATE: HP OpenView Multiple Vulnerabilities
(8) MODERATE: PHP NULL Processing Arbitrary File Overwrite
(9) LOW: Microsoft Indexing Service Cross Site Scripting Vulnerability(MS06-053)

Other Software
(10) HIGH: Multiple Products PHP File Include Vulnerabilities
(11) HIGH: Multiple Products SQL Injection Vulnerabilities
(12) HIGH: Tagger LE Remote Code Execution
(13) MODERATE: Act Networks NetPerformer FRAD Multiple Vulnerabilities
(14) MODERATE: SQL-Ledger/LedgerSMB Remote Code Execution

You are here ...

Attacks & Defenses	Cryptography	Applied crypto
• Risk assessment✓	•Random numbers	•SSH
• Viruses✓	•Hash functions✓	•PGP
• Unix security✓	MD5, SHA,RIPEMD	•S/Mime
• authentication✓	•Classical + stego	•SSL
• Network security	•Number theory	•Kerberos
Firewalls,vpn,IPsec,IDS	•Symmetric key	•IPsec
	DES, Rijndael, RC5	
	•Public key	
	RSA, DSA, D-H,ECC	

Random Numbers 3449563819347634876234734 6

- Why?
- Generating (pseudo) random numbers from a random seed
- Sources of random bits

Crypto Toolkit
secret-key crypto
public-key crypto
big-number math
random numbers
prime numbers
hash functions✓

Random Numbers

Good cryptography requires good random numbers.

- non-crypto: games, simulation
- salt, cookie, nonce (challenge)
- TCP sequence number
- used for public keys (RSA, D-H)
- per-message secrets (DSS k, PGP)
- used for secret keys (KDC/ssh session key)
- Encryption initialization vectors (IV)
- used for one-time pads or seed
- used for blinding

predictable = vulnerable

Random numbers -- definition

- equally likely to choose any element (uniform distribution)
- independence – can't infer one value in the sequence from others
- tests of random sequences (Knuth)
  - half the bits 1
  - Chi-square (bin tests), Komolgorov-Smirnov
  - spectral test
  - runs tests, n-D tests
  - bit, byte, word correlations
  - FIPS 140 tests: runs, poker, monobit
  - show that it's NOT random
  - Software: Diehard or ent
- cryptographic random numbers need to be
  - unpredictable
  - resistant to attack
    - By observing random output, can't predict next nor back-guess
    - Can't significantly influence output or initial seed

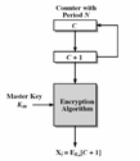
predictable, means you can guess the key

Pseudorandom numbers (PRNG)

- computer is deterministic
- rand() statistically OK, but cryptographically weak
  - based on linear congruential method
  - $R_{i+1} = (aR_i + b) \bmod n$
  - random numbers between 0 and n-1
  - eventually repeats
  - even if good seed, R_0, can solve for a,b,n if you can capture three R's
- repeatability good for testing (games/simulation), bad for crypto

Stronger PRNGs

- software linear feedback shift registers (LFSR)
- encrypted counter →
- chained encryption (OFB, DES/rijndael/etc.)
  - R_{i+1} = encrypt(key,R_i)
  - OFB's (e.g.,DES) are slow (export limits)
- repeated hash (MD5, SHA)
  - R_{i+1} = MD5(key,R_i)
  - eventually repeats $2^{\text{outputsize}}$
- Java SecureRandom -- SHA(seed,counter)
- Blum, Blum, Shub (quadratic residue) →



catch 22: need a good random seed (key)
seed needs to be unpredictable

BBS (large primes p and q, secret!)

$X_0 = s^2 \bmod n$ n =pq where p=q=3 mod 4

$X_i = (X_{i-1})^2 \bmod n$ s is relatively prime to n

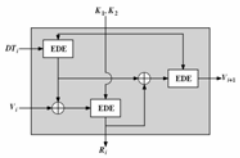
$B_i = X_i \bmod 2$ just use low order bit

i	X_i	B_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0
10	177046	0
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	137171	1
20	48060	0

p=383
q=503
s=101355

ANSI random numbers

ANSI X9.17



- V_0 64-bit initial seed
- E_k 3DES encryption with random keys K_1 K_2 (112 bits)
- DT_i high-resolution time t (max 64-bits)
- $R_n = E_k(E_k(t) \oplus V_n)$ next random number
- $V_{n+1} = E_k(E_k(t) \oplus R_n)$ next seed
- 9 DES operations
- Export controlled
- if keys are discovered, can predict all outputs! ☹

need a good random seed/key

True random numbers

Sources for a random seed

- coin toss, dice
- radioactive source
- noisy diode
- unplugged microphone (/dev/audio)
- lava lampe (video, SHA, BBS)
- FORTEZZA card has random source ?
- a few hardware sources for computers (QNG),intel i810, AMD768
  - thermal noise (resistor)
  - sampling oscillators
  - I/O device, 75 Kbs
  - Serial port devices

need to verify generator is working (FIPS 140-2)

Almost random sources

- source for a random seed
  - keystroke timing
  - mouse tracking
  - frame buffer
  - disk times
  - /dev/random
  - system status (netstat, ps, iostat)
- Netscape attack

- poll several sources
- may be biases -- use a mixing function (MD5, encryptor) to eliminate biases and "stretch" output
- Retain (and protect) random pool across reboots

Bad random sources

Netscape (SSL)

```
x = mixbits (time.tv_usec)
y = mixbits(getpid() + time.tv_sec + getppid() << 12)
seed = MD5(x,y)
nonce = MD5(seed++)
key = MD5(seed++)
```

MIT_MAGIC_COOKIE

```
key = rand() % 256
```

Kerberos v4

```
srandom(time.tv_usec ^ time.tv_sec ^ getpid() ^ gethostid() ^ counter++ )
key = random()
```

SESAME

```
key = rand()
```

Texas hold'em -- ooops

- On-line poker site used simple random number generator and guessable seed (Pascal's Randomize())
- Instead of 52! possibilities from a shuffle, less than 2^{32}
- After 5 cards revealed, you can figure out the order of the deck
- Nerd's revenge

SSH v1 random numbers

randoms.c

```
random_get_noise_from_command(state, uid, "ps laxww 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "ps -al 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "ls -alni /tmp/. 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "w 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "netstat -s 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "netstat -an 2>/dev/null");
if (time(NULL) - start_time < 30)
random_get_noise_from_command(state, uid, "netstat -in 2>/dev/null");
```

then mixes using MD5

ssh v2 uses openssl (/dev/urandom)

Truerand()

in Blaze's cryptolib

- returns about 16 bits of entropy per call
- starts a timer (16.7 ms)
- does count++ til timer expires
- shifts/XOR count into a buffer
- does it 11 times

randomness of clock skew and OS events

Java's dueling threads
Start 8 threads each doing counter[i]++
rand is XOR of 8 counters

PGP's random numbers

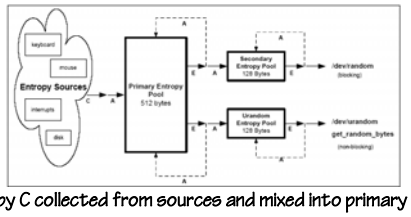
noise.c random.c randpool.c

- key strokes, high res time
- randomness retained in **randseed.bin**
- data from file pre/post washed with encryption (CFB)
- updated any time user does keyboard input
- updated with hash (MD5) of file being encrypted
- data stirred with MD5
- uses X9.17 but with IDEA
- paper on randomness of PGP IDEA keys

gpg randomness

- Random numbers for public key generation and message keys
- *cipher/random.c* based on Gutman's paper
- Uses /dev/urandom, seed file + pid, time, and clock
- Mixes pool with RIPEMD-160
- Wipes stack and prefers "secure memory" (no swap)
- Pool updated whenever key requested for encryption or secure hash (DSS k)
- State track current entropy of pool
- Application can request strong entropy (slower)
- Saves pool to file ~/.gpg/randseed.bin

UNIX /dev/random



- Entropy C collected from sources and mixed into primary pool
- Secondary pools draw entropy E from primary pool
- Whenever entropy is withdrawn some is mixed back in
- Event entropy (bits):
  keyboard(12), mouse(8), disk(3), interrupts(4)

/dev/random

- randomness from keyboard timings, mouse,interrupts, IO completion, delta times
- randomness is added to pool by each driver in kernel
  - Linux: e.g., in mouse driver routine

```
add_mouse_randomness(queue->buf[head] = inb(AUX_INPUT_PORT));
```

  - and in keyboard.c, after a key press

```
add_keyboard_randomness(scancode);
```

- mixing with primitive polynomial GF(2) and MD5
- startup is a problem (can save/restore pool)
- Available on most UNIX systems
- user can request random bytes with read to **/dev/random**
  reading reduces the available bits (4096 max) and may block!
- **/dev/urandom** for pseudo-random (recurse MD5)
- **/dev/hwrandom** if hardware random source available (AMD, Intel)

What if embedded system, cold start, little activity?

Entropy gathering daemons (EGD)

- For systems without /dev/random
- EGADS (secureosftware.com)
  - Conservative entropy estimates
    - If not enough entropy, gathers more data
  - "Tiny" variant of Yarrow, slow pool and fast pool of entropy
  - Data gathered from system mixed in pool (8 LFSR's) with UMAC nonce and counter using primitive polynomial
    - Unix info: /dev/random, df, ps
    - Windows: timestamp, sleep() jitter, performance counters,
  - Access via UNIX domain socket
  - API for random bytes, int's, double, range, various distributions (uniform, gaussian, …)
  - OpenSSL will look for EGADS daemon if no /dev/random

OpenSSL random numbers

- mix (MD5) in msg, time, /dev/random
- Windows -- hash of screen (frame buffer)
- add randomness on each connect() in SSL
- save/restore from rand file
- Used in key generation utilities
- API RAND_*(3)

```
#include <openssl/rand.h>
char buff[1024];
RAND_bytes(buff, sizeof(buff));
```


Other PRNG's

- Window's CryptGenRandom()
  - Entropy gathered from system (process and thread IDs, ticks since boot, current time, memory info, performance counters) and mixed with MD4 and RC4
- Java's SecureRandom
  - Initial entropy
    - Old version entropy from "dueling threads" or system time and Rand
    - today: /dev/random or Window's CryptGenRandom()
  - Can also set your own initial seed
  - Mix with SHA1
  - Methods to get int's, long's, double's, gaussian

```
import java.security.SecureRandom;
byte test[20];
SecureRandom crng = new SecureRandom();
crng.nextBytes(test);
```


Other applications

Source of randomness

- Netscape-1 -- time, pid, ppid
- Netscape-2 -- time, ps, netstat, env,stat,events
- Kerberos -- master key, pid, time
- Nautilus -- microphone feedback
- stel-1 -- uname, time, pid, ppid, pgrp
- stel-2 -- truerand
- ISAKMP-1 -- fixed seed, time, SHA
- ISAKMP-2 -- truerand
- GKMP -- fixed seed, time, SHA mix, X9.17
- PEM -- key strokes or file or system info

Ask how randomness is provided !

Random thoughts

- avoid math library PRNGs
- avoid linear-congruential PRNGs
- make sure your hardware sources are working (FIPS 140-2)
- use several sources
- mix/hash them into pool
- protect the pool/seed
- Optional: stretch the bits with a hash

Secret messages

- Steganography
- Classical encryption (ciphers, codebooks)
- Digital encryption

Secrecy

- Sociology/philosophy
- Secrets of love, greed, war throughout history (oral/written)
- Techniques for hiding
- Techniques for discovering

Steganography

The art of covered writing
Security through obscurity

- motivation
- history
- classical techniques
- wartime censors
- spread spectrum
- digital steganography
- pathological stego

Stego – hidden message

Crypto – visible but encrypted (gibberish)

AP

U.S. officials say Osama bin Laden is posting instructions for terrorist activities on sports chat rooms, pornographic, bulletin boards and other Web sites.

Steganography -- motivation

- conceal what you are communicating
- double meaning
- encrypted message is "random"
- sending encrypted messages might make you a spy

technique
- cover document/carrier (includes redundancy/noise)
- message (probably encrypted)
- maybe a key
- replace noise with message

history
- ancient Greece: waxed tablets
- shaved heads and tatoos
- other bodily concealment
- written text

3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told, by the 25th. Admin has improved here, though there's room for improvement still: just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours,

Figure 2.8 A Puzzle for Inspector Morse
(from *The Silent World of Nicholas Quinn*, by Colin Dexter)

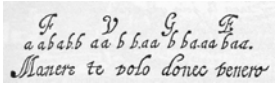
steganalysis

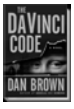
- detect/prove stegomessage
- read stegomessage
- remove/jam the stegomessage
- have a better statistical model of the cover than the sender

Classical stego

linguistic
- character marking, overwrite with a pencil
- cursive variations (Bacon)
- pin punctures
- first letter of each word (null cipher)
- letter positions on page (overlay, grille)
- drawings
- anagrams
- codes

technical
- microdots
- invisible ink
- typewriter correction ribbon
- smuggling (false bottoms)
- spread spectrum (RF)
- digital





ANAGRAMS

Rocket boys ↔ October Sky

Computer science ↔ more succinct pee

Computer security ↔ erotic cutey rumps

Red hat linux ↔ rat-held UNIX

The houses of parliament ↔ Loonies far up the Thames

Stego example

In England, during the days of Cromwell, Sir John Trevanion, a cavalier of distinction, having fallen from grace was locked up in Colcester Castle. He had every reason to believe that he would be put to death just as had been his friends and fellow Royalists, Sir Charles Lucas and Sir George Lisle. While awaiting his doom, however, he was one day handed the following letter by his jailer.

Punctuation + 3

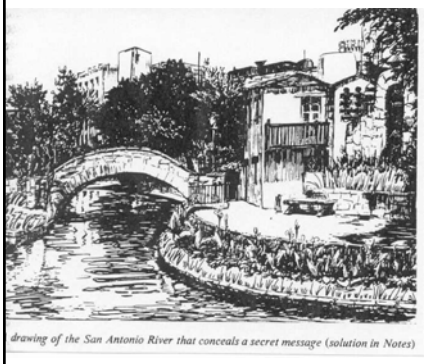
Worthie Sir John:

Hope, that is ye beste comfort of ye afflicted, cannot much, I fear me, help you now. That I would saye to you, is this only: if ever I may be able to requite that I do owe you, stand not upon asking me. 'Tis not much that I can do: but what I can do, bee ye verie sure I wille. I knowe that, if dethe comes, if ordinary men fear it, it frights not you, accounting it for a high honor, to have such a rewarde of your loyalty. Pray yet that you may be spared this soe bitter, cup. I fear not that you will grudge any sufferings: only if bie submission you can turn them away, 'tis the part of a wise man. Tell me, an if you can, to do for you anythinge that you wolde have done. The general goes back on Wednesday. Restinge your servant to command.

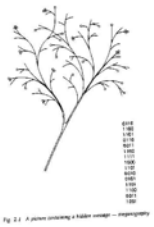
R.T.

PANEL AT EAST END OF CHAPEL SLIDES

Stego pics



drawing of the San Antonio River that conceals a secret message (solution in Notes)



A Puzzle for Lord Peter – Dorothy Sayers

- The key 7876565434321

```
12345678
ithought   h
toseethe   e
fairiesi   s
nthefiel   i
dsbutisa   t
wonlythe   t
evilelep   e
hantswit   t
htheirbl   h
ackbacks   b
```

I thought to see the fairies in the fields, but I saw only the evil elephants with their black backs. Woe! how that sight awed me! The elves danced all around and about while I heard voices calling clearly. Ah! how I tried to see—throw off the ugly cloud—but no blind eye of a mortal was permitted to spy them. So then came minstrels, having gold trumpets, harps and drums. These played very loudly beside me, breaking that spell. So the dream vanished, whereat I thanked Heaven. I shed many tears before the thin moon rose up, frail and faint as a sickle of straw. Now though the Enchanter gnash his teeth vainly, yet shall he return as the Spring returns. Oh, wretched man! Hell gapes, Erebus now lies open. The mouths of Death wait on thy end.

Stego examples

Newe Eight Weather: Tonight increasing snow. Unexpected precipitation smothers eastern towns. Be extremely cautious and use snowtires especially heading east. The highways are knowingly slippery. Highway evacuation is suspected. Police report emergency situations in downtown ending near Tuesday.

First letter of each word yields:
Newt is upset because he thinks he is President.

From WWII German spy (Kahn):
Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.

Second letter of each word yields:
Pershing sails from NY June 1.

WWII censors

- 15,000 examiners, opened million pieces of overseas mail/day
- No crossword puzzles, chess games, student grades, knitting instructions, blank pages, childish scrawls, lover's X's
- cables ordering flowers, quantity/type could be encoding
- most commercial codes forbidden
- classified ads
- radio broadcasts
- shifted the hands on a shipment of watches

Digital steganography

Did you find the hidden message?

- 2.3 MB message in a Kodak digital snapshot (2048x3072 pixels, 24-bit pixel) use 1 bit/pixel
- tools to hide PGP messages
- tools to hide messages in GIF's AU's video
- tools to hide data on disk
- tools to hide data/services on Internet
- font alterations, word/line shifts
- embed in email headers, postscript comments, html
- plus classical stuff (masks, last letter of each line, etc) in plaintext email
- covert channels

Stego file system

- having encrypted data may be incriminating, may be forced to give the key
- deniability
- hide data in file system, unallocated blocks, in unused bits of file infrastructure, spare sectors
- if you don't know name and key, can't even prove it's there
- stego file system software incriminating?

- More later (forensics)

Internet steganography

- hackers use non-standard ports for "telnet"
- shell over ICMP (the ping protocol)
- hide bits in network packet headers
  - generate rogue packets
  - hack kernel to piggy-back on legit traffic
  - unused bits or TCP/IP options
- Hide info in html, pdf, mail headers, postscript, images, audio, video …

The envelope is the message

```
%!PS-Adobe-3.0
%%Title: Microsoft Word - erika.doc
%%Creator: PSCRIPT.DRV Version 4.0
%%CreationDate: 03/07/03 10:10:37
%%BoundingBox: 13 13 599 780
%%Pages: (atend)
```

```
<html><head>
<LINK REL="SHORTCUT ICON" HREF=usflag.ico>
<title>Tom Dunigan's Home Page</title>
<!BODY bgcolor="#c0c0c0">
<BODY bgcolor="#ffffff">
<!-- Here's a comment which I can see when I edit the file
     but which is invisible to anyone reading the file via a
     browser.-->
```

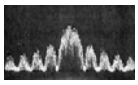
```
From galbrait@cs.utk.edu  Tue Aug  3 21:07:52 2004
Return-Path: <galbrait@cs.utk.edu>
X-Original-To: dunigan@wisp.csm.ornl.gov
Delivered-To: dunigan@wisp.csm.ornl.gov
Received: from emroute1.cind.ornl.gov (emroute1.cind.ornl.gov [160.91.4.119])
        by wisp.csm.ornl.gov (Postfix) with ESMTP id B68901B63B
        for <dunigan@wisp.csm.ornl.gov>; Tue,  3 Aug 2004 21:07:52 -0400 (EDT)
Received: from emroute1.cind.ornl.gov (localhost [127.0.0.1])
 by emroute1.cind.ornl.gov (PMDF V6.2-X27 #30899)
 with ESMTP id <0I1W004E0DTI6K@emroute1.cind.ornl.gov> for
 dunigan@wisp.csm.ornl.gov (ORCPT thd@ornl.gov); Tue,
 03 Aug 2004 21:07:51 -0400 (EDT)
Received: from VIRUSCAN-DAEMON.emroute1.cind.ornl.gov by emroute1.cind.ornl.gov
 (PMDF V6.2-X27 #30899) id <0I1W00702DSGYC@emroute1.cind.ornl.gov> for
 dunigan@wisp.csm.ornl.gov (ORCPT thd@ornl.gov); Tue,
 03 Aug 2004 21:07:32 -0400 (EDT)
```


Covert channels

- techniques to leak information from secure system
- use entities to convey information that are not normally expected to transport information
- multilevel secure OS's (CMWs), encrypted Internet tunnels
- low bandwidth
- timing, size of messages, presence or absence of objects
- traffic analysis (volume, src/dest, new activity)
- a concern for secure OS's

- generate your own noise (see Rivest winnowing/chaffing)

Spread spectrum



- Radio (RF) transmission
  - Hard to detect (looks like noise/random)
  - Hard to jam
  - Favorite of military, now in consumer wireless phones etc
- Transmitter and receiver use same pseudo-RF key
  - Frequency hopping
  - Direct sequence (modulate carrier with a pseudo-random code sequence)

Stego shortcomings

- lot of overhead to hide a few bits
- once discovered, rendered useless
- security through obscurity
  - keyed stego helps

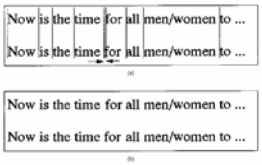
- active research: digital steganalysis

- commercial application: digital watermarking

Digital watermarks

using steganography for copy protection

- spatial alterations
  - –postscript/hardcopy
  - –line or word positioning
- each recipient assigned code word
- code corresponds to unique alteration
- Color copiers imbed a watermark throughout the copy (spread spectrum like) to identify the copy machine– traceback counterfeiting etc.



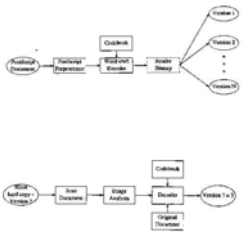
Copy protection

detection
- scan illicit document
- image analysis with original and codes
- determine whose copy was leaked
- withstands fax, xerox
- digital watermarks or hidden "signatures"

- key to "machine ID"

- key to a "dongle"

- uncopiable disks

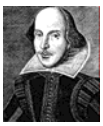


Shake & Bacon

pathological steganography

The hidden meaning …

- songs, movies
- shadows in Lion King
- The Bible Code
- Bacon wrote Shakespeare
- Nostradamus quatrains



Is your future encoded in the textbook?

predictions

- Bible Code (original hebrew) predicts current world events
- Nostradamus (16 Century) quatrains – predict Intel pentium bug

C 2,VI:

Aupres des portes & dedans deux citez
Seront deux fleaux, & onc n'apperceut VN TEL,
Faim, dedans peste, de fer hors gens boutez,
Crier secours au grand Dieu immortel.

and its translation to English (V ➔ 5 or PENTium):

Near the gates and inside two cities
Will be TWO FLAWS, and nobody noticed it [from] INTEL
Hunger, pest inside, by steel people thrown out
Cry for help to the great immortal God.

Classical crypto

- codes
- substitution ciphers
- transposition ciphers
- mechanical assists

Book: Kahn, *The Codebreakers*

- Crypto terms

  cryptology -- study of encryption and decryption

  cryptography -- secret writing, using encryption to conceal info

  cryptanalysis -- breaking encryption, code breaker

- secret key crypto (symmetric)
- public key crypto (asymmetric)

assume algorithm is known, strength is in key

Crypto timeline

```
1900 BC     first written cryptography
 500 BC     substitution cipher used by Hebrew scribes (ATBASH cipher)
  50 BC     Caesar cipher
   7 BC     scytale  (first encryption device)
 855        cipher alphabets for magic
1379        diplomatic code
1518        first book on cryptology
1585        Vigenere cipher
1790        Jefferson wheel cipher
1854        Playfair cipher
1917        Vernam cipher machine (one-time pad)
1923        Enigma machine
1948        Captain Midnight decoder ring
1970        Feistel (IBM) Lucifer cipher
1976        DES
1976        Diffie-Helman
1977        RSA
1984        ElGamal
1985        ROT13
1990        IDEA
1991        PGP, DSA
1992        SHA
1994        RC5
1999        AES ciphers
2002        quantum encryption (optical)
```

Media
Spoken
Written (1900 BC)
Telegraph (1835)
Radio/wireless (1895)
Internet (1980)
Photons (2001)

cryptanalysis

recover the message and/or key, you know the encryption algorithm

- **ciphertext only** -- cryptanalyst has only cipertext of possibly many messages
- **known plaintext** -- access to both plain and ciphertext of several messages, probable words
- **chosen plaintext** -- plaintext and ciphertext, plus attacker can choose the plaintext that gets encrypted (the "oracle")
- **chosen ciphertext** -- attacker has access to decrypting box, objective is deduce the key, have the corresponding plaintext

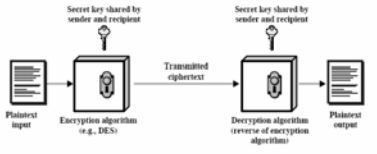
The HUMAN factor

- **rubber hose attack** -- threaten, torture, blackmail for the key
- **purchase-key attack** -- bribery (or burglary)
- **scam attack** – "excuse me, could you tell me your paseword?"
- **I'm stupid attack** – easy to guess key (name, birthdate, phonenumber, ….)

The more data you have the better, assisted by recognizable plaintext (ASCII), probable word attacks, context, human error/laziness.

Key attacks

- good guesses
- dictionary attacks
- trojan horse
- know key generation technique (bad random numbers)
- brute force



Brute force

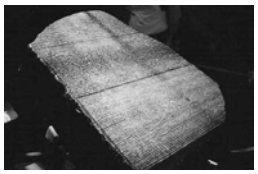
Try every key … time and space complexity

- storage is cheap, pre-calculate x encrypted with all keys
- faster CPU's, parallelism
- today: 40-bit weak, 56-bit suspect

- Time and space
  - at one million guesses/second, 56-bit key takes 1000 yrs
  - 112-bit key ➔ 10^{28} years, but universe 20×10^{9} years old!
  - store 2^{56} 64-bit messages ➔ 20 billion 1 GB drives

- only the one-time pad is unconditionally secure, make do with computationally secure.

cryptanalysis



similar problems/techniques

- ancient language translation
- hieroglyphics/rosetta stone
- extra-terrestial languages
- military use of Choctaws (1918) and Navahos (WWII, Windtalkers) for secure radio communication

codes

- form of stego, words with double meanings
- diplomatic and commercial codes
- codebook -- mapping of words or phrases to code word or "number", and reverse for two-part
- supplemented with "alphabet" to spell out words not in codebook
- commercial codes historically used to reduce telegraph costs
- book codes: 534.17 242.21 114.55
  - E.g. word 17 on page 234 of Moby Dick

Code book

One Part Code

ABABA--A, an
ABABE--Abandon-ing-s
ABABI-- Abandoned
ABABO--Abate-ing-s
ABABU--Abated
ABACI--Ability
…

Two-part code

VANOL--A, an	ABABA--It is hoped
LANEX--Abandon-ing-s	ABBCO--Shipped
STUGH--Abandoned	ACDZR--Terminated
RIZLB-- Abated	….

…

Classical ciphers

Converting plain text to cipher text

- Transposition
  - Rearrange plaintext (anagrams)
- Substitution
  - Substitute letters of the plain text with other letters/symbol
  - Mono/polyalphabetic ciphers
  - One-time pad

- Combo of both transposition and substitution
- want something you can use "in the field"

Transposition

- re-arrange characters (permute)
- plain text letters re-arranged, arrays or rail-fence

```
T H I S A S
E C R E T M
E S S A G E

teehcsirsseaatgsme
```

- use keyword to select column selection order
- double, do it again
- Scytale (pronounced SIT'alee) (7 BC)
- transposition can be represented as matrix transform
- cryptanalysis:
  frequency analysis (same letter frequencies as plain text)

- **product cipher**: transpose and substitute, (watch out for multiple substitutions)
- **superencryption**: codebook, transpose, and substitute

substitution

- **monoalphabetic** unique mapping of plaintext alphabet to ciphertext alphabet
  - Caesar, Hill, Playfair
- **polyalphabetic** plaintext mapped to ciphertext based on key to select alphabet
  - Vigenere, enigma
- **stream** keystream is generated and used to map plaintext to ciphertext
  - One-time pad

Monoalphabetic – shift ciphers

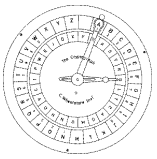
- shift 1: HAL maps to IBM (2001 Space Odyssey)
- caesar cipher (shift 3):
  - Military cipher
  - A maps to C

```
plain:  meet me after toga party
cipher: PHHW PH DIWHY WRJD SDUWB
```

  - caesar/shift: (a=0 b=1…z=25)
    encrypt $c_i = (p_i + k) \bmod 26$
    Decrypt $p_i = (c_i - k) \bmod 26$
    "key" is shift, k (how many possible keys?)
  - caesar special case of affine ciphers
- Affine (linear):
  - encrypt $c_i = (ap_i + b) \bmod 26$
  - decrypt $p_i = a^{-1}(c_i - b) \bmod 26$
    how many possible keys (a,b)?
- decoder rings, slides

- In software use mod arithmetic

Captain Midnight decoder ring 1948



Wheatstone disc 1817

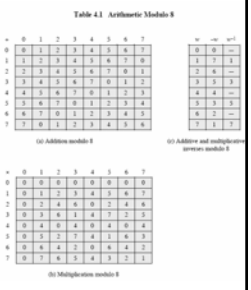
Modular arithmetic

- mod (congruence), remainder after dividing
- if a mod n = b then a = kn + b
- 14 mod 3 = 5 mod 3 = 2 (14 = 4*3 + 2)
- modular arithmetic (+,*) on non-negative integers Z_n
  - associative $a+(b+c) \bmod n = (a+b)+c \bmod n$
  - commutative $ab \bmod n = ba \bmod n$
  - distributive $a(b+c) \bmod n = (ab + ac) \bmod n$
  - reducible $ab \bmod n = ((a \bmod n)(b \bmod n)) \bmod n$

  lets us work with smaller numbers
- linear (affine) ciphers of alphabet x mod 26
  - in C x % 26
- additive inverse: (5 + 3) mod 8 = 0 or -2 = 6 mod 8
- multiplicative inverse $(x^{-1}x) \bmod n = 1$

  only numbers relatively prime to n have multiplicative inverse, use extended Euclid algorithm to find inverse

  If n is prime, every member has an inverse
- Makes encrypting/decrypting easy in software … makes cryptanalysis easy too ☹



Modular arithmetic exercises

- 19 mod 8 = ?
- 3*7 mod 8 = ? 5 + 15 mod 7 = ?
- Additive inverse: a + b mod n = 0

  x + 3 mod 8 = 0

  -2 mod 8 = ?
- Multiplicative inverse a*b mod n = 1
  - Use extended Euclidean alg to calculate (~dunigan/cns06/gcd.c)
  - Multiplicative inverse of a mod n only exists if a is relatively prime to n

    x * 5 mod 8 =1

    4/3 mod 8 = ?

    3/21 mod 26 = ?

    $5 * 15^{-1} \bmod 26$ = ?

```
Multiplicative inverses mod 26:
(1,1) (3,9)  (5,21) (7,15)
(11,19) (17,23) (25,25)
```

Slide rule ciphers

St Cyr 1880
key: position

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cryptanalysis of monoalphabetic substitution

Even with ciphertext-only

- for shift ciphers, can do brute force search (25 keys) (**assumes** plaintext is recognizable !)
- could map to any ordering of alphabet (26! keys)
- use frequency statistics of language (single letter, digrams, trigrams, words)
- assume most frequent letter in ciphertext is E, then continue using trial-and-error
- often need as little as 200 ciphertext letters

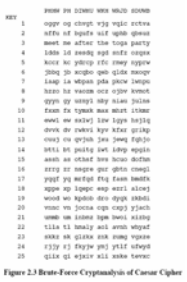


Figure 2.3 Brute-Force Cryptanalysis of Caesar Cipher

Frequency analysis

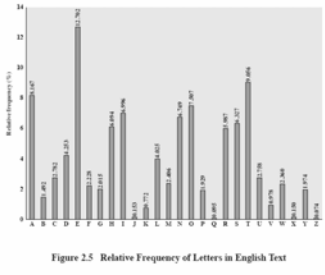


Figure 2.5 Relative Frequency of Letters in English Text

E. V. Wright's 267 pg., 1937 novel *Gadsby* had no e's !

From the introduction:

It is a story about a small town. It is not a gossipy yarn; nor is it dry, monotonous account, full of such customary fillins as "romantic moonlight casting murky shadows down a long, winding country road." Nor will it say anything about twinklings lulling distant folds; robins carolling at twilight, nor any warm glow of lamplight from a cabin.

- To hide single letter frequencies:
  - encrypt multiple letters (playfair(2), hill(m))
  - use multiple alphabets (polyalphabet)

Playfair cipher

- Wheatstone, 1854
- Encrypt two letters at a time (digrams)
  - Pair in same row, replace with letter to right (OQ→PU) (decrypt: left)
  - Pair in same column, replace with one beneath (BU→KZ) (decrypt: above)
  - Replace row letter with letter in row of other letter's column (SX → TW)
- Used in Boar War, WWI, and backup cipher in WWII
  - John F. Kennedy's PT-109 was sunk by a Japanese cruiser in the Solomon Islands. He made it to shore on Japanese-controlled Plum Pudding Island and was able to send an emergency message in Playfair from an Allied coast-watcher's hut to arrange the rescue of the survivors from his crew. (problem 2.9 in text)

```
Key: MANCHESTER
M A N C  H
E S T R  B
D F G IJ K
L O P Q  U
V W X Y  Z

Encrypt SECRET MESSAGE

SE CR ET ME (SX) SA GE
TS RI SR ED TW FS DT
```


Hill cipher

- linear algebra *mod* 26
- m x m key matrix K, m-column of plaintext P
  - encryption C = KP mod 26
  - decryption P = K^{-1} C mod 26
  - K must be invertible
    - determinant D non-zero mod 26
    - determinant has multiplicative inverse mod 26 (odd and not a a multiple of 13)
- block cipher (m letters at time)
- hides m - 1 letter frequencies
- hard to break ciphertext-only
- with known plaintext can solve for K
- easy with a computer

```
a b   inverse: d/D  -b/D
c d            -c/D  a/D
 where determinant D = ad-bc
d/D mod 26 == d*D^-1  mod 26
 use extended gcd to find D^-1
```

$$\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 26 \rightarrow \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

```
Plaintext: friday  5 17  8  3  0 24
Key: hitd   7  8 19  3
 7 8        5        (7*5 + 8*17) mod 26
19 3       17          = 171 mod 26
                       = 15
ciphertext: 15 16  2  5 10 20  pqcfku
```

```
Inverse: D=7*3 - 8*19 mod 26 = -1 = 25
 use gcd,c D^-1 mod 26  = 25
  3*25  -8*25      23   8
-19*25   7*25      19  19
(23*15 + 8*16) mod 26 = 5
```


Polyalphabetic substitution ciphers

- A set of monoalphbet substitutions based on a key as long as the message (repeat)
- hides single-char frequency
- use Vigenere or Beaufort (key is repeated)
- **vigenere:**
  encrypt $c_i = (p_i + k_i) \bmod 26$
  decrypt $p_i = (c_i - k_i) \bmod 26$
- **beaufort:**
  encrypt/decrypt $c_i = (k_i - p_i) \bmod 26$
- use homophones, map E to one of several cipher symbols

Jefferson cipher cylinder (36 discs, key is order of discs), 1790-1942
Adopted by US Army in 1922

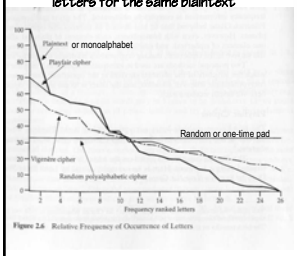
Beaufort

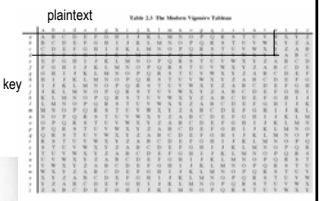
```
// c_i = (k_i - p_i) mod 26
unsigned char plain, cipher;
int count = 0;
int five = 0;

while(!feof(file_handle)){
  plain = fgetc(file_handle);
  if((plain >= 'A') && (plain <='Z')) {
         plain = plain - 'A';
         cipher = Key[count]-plain;
         if (cipher < 'A') cipher +=26;
         printf("%c", cipher);
         five = (five+1)%5;
         if(five == 0) printf(" ");
   } else if ((plain >= 'a') && (plain <='z')) {
          plain= plain -'a';
         cipher = Key[count]-plain;
         if (cipher < 'a') cipher +=26;
         printf("%c", cipher);
         five = (five+1)%5;
         if(five == 0) printf(" ");
  }
}
 ....
```


Vigenere

Given a key letter x and a plaintext letter y, the ciphertext is the letter at the intersection of row x with column y

Stength: multiple ciphertext letters for the same plaintext





Key: *deceptive*

Plaintext: *we are discovered*

```
deceptivedecept
wearediscovered
ZICVTWQNGRZGVTW
```

encrypt $c_i = (p_i + k_i)$ mod 26
decrypt $p_i = (c_i - k_i)$ mod 26

Vigenere cryptanalysis

vigenere, non-running key, cipher-text only

- kasiski method -- use repetitions in ciphertext for clue to key length

```
Plaintext:   TOBEORNOTTOBE
Key:         NOWNOWNOWNOWN
Ciphertext:  GCXRCNACPGCXR
repeats are 9 chars apart, guess period is 3 or 9.
```

- index of coincidence
- harder for running key (book, autokey)
  - still some statistics
  - probable word -- try at different offsets within ciphertext, identify key/book

Index of coincidence

Let p_i be the probability of the i^{th} letter in English text then

$$\sum_{i=0}^{25} p_i^2 = 0.065$$

For random text, this sum is 0.038.

If f_i is the number of appearances of letter i in an L-character ciphertext, then

$$IC = \frac{\sum_{i=0}^{25}\binom{f_i}{2}}{\binom{L}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{L(L-1)}$$

cryptanalysis of vigenere (fixed-size key)

Try breaking ciphertext into m columns, and calculate IC for each column.

```
m       IC
1       0.045
2       0.046 0.041
3       0.043 0.050 0.047
4       0.042 0.039 0.046 0.040
5       0.063 0.068 0.069 0.061 0.072
```

looks like keysize is 5

Now you have columns of ciphertext with monoalphabet for each column. Use mono techniques to guess key.

One-time pad

- perfect substitution cipher
- sender and receiver share a random stream (pad)
- XOR plaintext (p_i) with pad (r_i)

  $c_i = p_i \oplus r_i$

  decrypt $p_i = c_i \oplus r_i$
- must not reuse pad (why?)
- pad (key) needs to be as long as message!
- key distribution a problem or random stream generation a problem
- cheat:
  - XOR with fixed length key (repeated)
  - XOR with a running key from a book (just a polyalphabet, Vigenere, breakable)
  - XOR with pseudo-random number stream

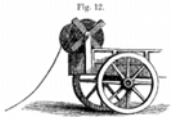
Civil War military communications

Communications usually done by couriers (horseback)

Base communications and lookouts (signal corp) via line of sight : lanterns & flags (semaphores)

Technology of the civil war

- railroads
- telegraph
  - Strung along railways
  - Easily tapped/cut
  - Mobile troops carried their cable and ran it from command post to command post



Civil War ciphers (telegraph)

- Confederates
  - Shift cipher or sometimes Vigenere (three keys), dictionary (pg,col,line)
  - Couldn't decrypt Union ciphers (published in newspapers for help)
  - Often couldn't decrypt their own (telegraph errors)
  - They lost
- Union
  - US Military Telegraph (civilian cipher clerks)
    - USMT reported to Sec. of War
  - Substitution and transposition
    - Based on code words and "routing" (word transposition)
    - Real words (vs random text) helped overcome telegraph errors
  - 10 ciphers developed during the war (developed by Anson Stager)
  - Codebooks carefully controlled (only 14 people had codebook for cipher 6)
  - 10th cipher codebook:
    12 pages of possible routes and 36 pages with 1,608 code words
  - They won ☺

Union cipher

Message:

FOR COLONEL LUDLOW:

RICHARDSON AND BROWN, CORRESPONDENTS OF THE TRIBUNE, CAPTURED AT VICKSBURG, ARE DETAINED AT RICHMOND. PLEASE ASCERTAIN WHY THEY ARE DETAINED AND GET THEM OFF IF YOU CAN.

THE PRESIDENT.

code book

COLONEL	VENUS
CAPTURED	WAYLAND
VICKSBURG	ODOR
RICHMOND	NEPTUNE
PRESIDENT	LINCOLN
4:30 PM	NELLY

ADAM

Cipher clerk adds code for time (NELLY), selects GUARD cipher and will add fill and column of NULLs. Prepends GUARD

FOR	VENUS	LUDLOW	RICHARDSON	AND
BROWN	CORRESPONDENTS	OF	THE	TRIBUNE
WAYLAND	AT	ODOR	ARE	DETAINED
AT	NEPTUNE	PLEASE	ASCERTAIN	WHY
THEY	ARE	DETAINED	AND	GET
THEM	OFF	IF	YOU	CAN
ADAM	NELLY	THIS	FILLS	UP

GUARD cipher:
5 columns, route:
Up column 1.
Down column 2.
Up column 5.
Down column 4.
Up column 3.

GUARD ADAM THEM THEY AT WAYLAND BROWN FOR KISSING VENUS CORRESPONDENTS AT NEPTUNE ARE OFF NELLY TURNING UP CAN GET WHY DETAINED TRIBUNE AND TIMES RICHARDSON THE ARE ASCERTAIN AND YOU FILLS BELLY THIS IF DETAINED PLEASE ODOR OF LUDLOW COMMISSIONER

World War I crypto

- Technology: telephone/telegraph and "wireless" – radio
  "trench codes" – code words for locations etc.
- German ADFGX cipher -- substitution and transposition, good Morse separation of A D F G X, reduce transmit errors
  A .- D -.. F ..-. G --. X -..-

1) Randomly place letters of alphabet in grid (the key)

```
  A D F G X
A f z t n i/j     send ammunition → DGFGAGDF FXXDXDGFAGAXAFAXXAAG
D g l d s b
F y v p e a                       monoalphabetic substitution
G k c u w r
X o m x q h
```

2) Write the letters out under a transposition keyword, e.g WARTHOG

```
  W A R T H O G                     A G H O R T W
  D G F G A G D                     G D A G F G D
  F F X D X D   then sort columns   F D D X X X F
  G F A G A X A                     F A A X A G G
  F A X X A A G                     A G A A X X F
```

3) Write letters out in column order

GFFADDAGADAAGXXAFXAXGXGXDFDF

Wireless Traffic Analysis

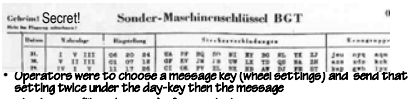
Although the French had no idea what German messages said, they recorded with diligence the strength of signal, the volume of messages, call signs, as well as any clear non-encrypted text that slipped through to determine the composition of the German forces. Their analysis resulted in a diagram of major German bases within the two weeks of the outbreak of war, which was mostly accurate.

Enigma machine

"... would require 1000 operators several million years to try all states"

- polyalphabet substitution used by Germans WWII
- use 3 out of 5 wheels, plugboard (5x4x3x26x26x26, 10^{15}) = 10^{21}
- key: which wheels, their order, their starting position, ring position, and pluge code book had settings for the day

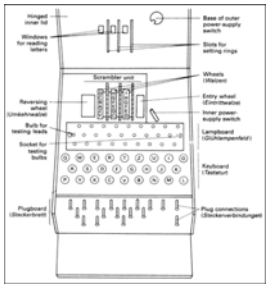
```
Plugboard Setting: (N,K) (P,T) (B,E) (J,V) (Q, C) (R,O)
Rotor order: 3-1-2      Letter ring position: 24 3 17
Rotor Key Setting: FKT
```


Geheim! Secret! Sonder-Maschinenschlüssel BGT

- Operators were to choose a message key (wheel settings) and send that setting twice under the day-key then the message
- wheels step (like odometer) after each character
- reflector allows decryption with no change to initial setup
- M209 (portable version, sort of)
- UNIX crypt -- single rotor, reflector, 256 char alphabet

- Allies (Poland and Turing) broke the code – Germans chose guessable settings, repeated plaintext -- lots of work

Enigma components



3-rotor enigma

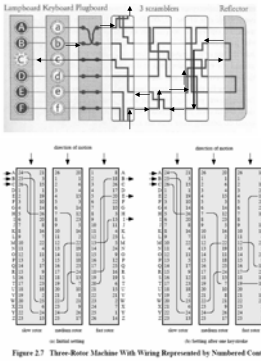
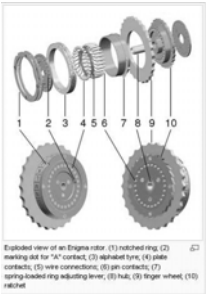


Figure 2.7 Three-Rotor Machine With Wiring Represented by Numbered Contacts



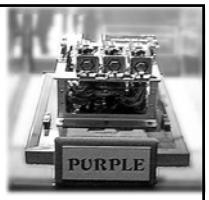
Exploded view of an Enigma rotor. (1) notched ring, (2) marking dot for "A" contact, (3) alphabet tyre, (4) plate contacts, (5) wire connections, (6) pin contacts, (7) spring-loaded ring adjusting lever, (8) hub, (9) finger wheel, (10) ratchet

Each rotor is a monoalphabet substitution

enigma applet

movie

Japanese PURPLE machine



- Used telephone stepping switches instead of rotors
- US's Friedman able to construct a machine to crack PURPLE without ever seeing the original machine (18 months)
- Japanese often sent same message encrypted with Red machine (already "broken") and PURPLE, so had both plaintext and cipher text
- Feed "known plaintext" to help decipher:
  "water desalination unit out on Iwo Jima"

Next time ...

Digital cryptography

DES, CAST, blowfish, lucifer

	Lectures
1.	Risk, viruses
2.	UNIX vulnerabilities
3.	Authentication & hashing
4.	Random #s classical crypto
5.	Block ciphers DES, RC5
6.	AES, stream ciphers RC4, LFSR
7.	**MIDTERM** ☹
8.	Public key crypto RSA, D-H
9.	ECC, PKCS, ssh/pgp
10.	PKI, SSL
11.	Network vulnerabilities
12.	Network defenses, IDS, firewalls
13.	IPsec, VPN, Kerberos, secure OS
14.	Secure coding, crypto APIs
15.	review