Crypto – B/B mindmeld

- history
- symmetric
- hashes/macs/1-way etc
- public key
- summary

Crypto – B/B mindmeld

- history
- symmetric
- hashes/macs/1-way etc
- public key
- summary

a little bibliography please

- Zimmerman Telegram – Barbara Tuchman
  - how one nation state can go wrong with crypto twice in one century (this is only WWI)
- Secrets and Lies. Bruce Schneier
  - why crypto may not solve your problems
- The Codebreakers. David Kahn.
  - the book … (WW II got added post declassification)
- Cryptography Decrypted. Mel/Baker
  - or if you are a hard case, Applied Crypto, although …

history

- up to WW II it was alphanumeric in the west
- stream ciphers (a letter at a time)
- block ciphers (a block of letters at a time)
- WW II changed everything (computers)
- but there are still some very basic principles to cryptoanalysis that have lasted

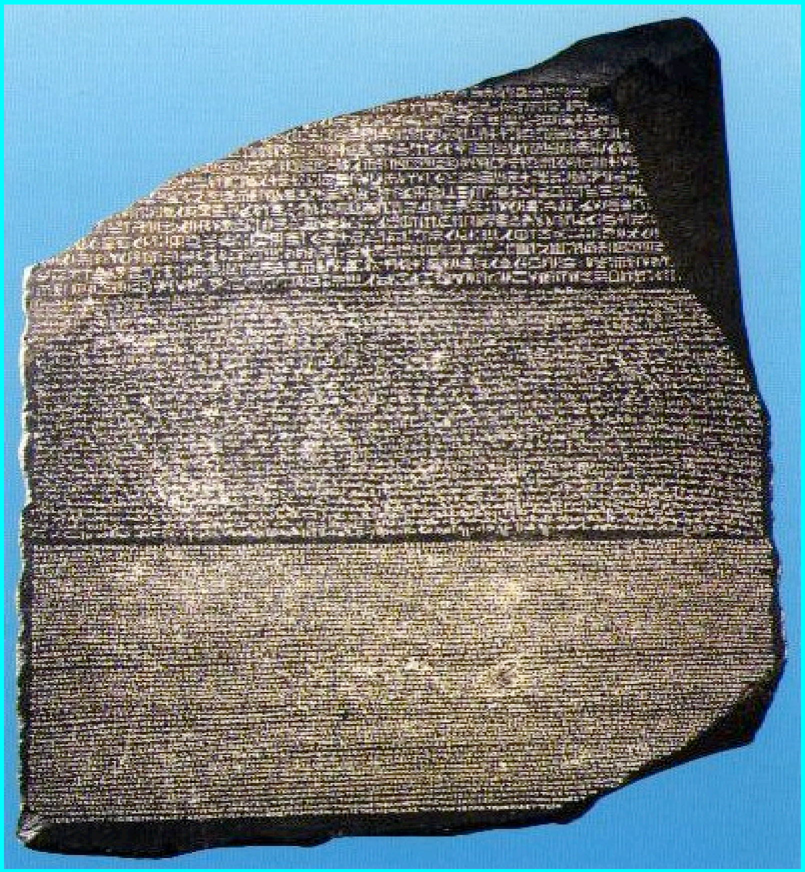
fundamental defs

- cryptography
  - sometimes secret writing wasn't meant to be secret
  - sometimes it is
  - alphabetic historically, now bits/bytes/blocks
- cryptoanalysis
  - decoding the secret writing without the keys
  - hey! chocolate for your password?

policy considerations

- think about this as we talk about what are basically mechanisms
- what threats exist in this space?
- what might policy considerations thus be for:
  - a. govt. spy agency (NSA or CIA or MI5?)
    - bond, james bond AND his laptop?
  - b. hospital
  - c. computer technology company
  - d. university

Rosetta stone - solved

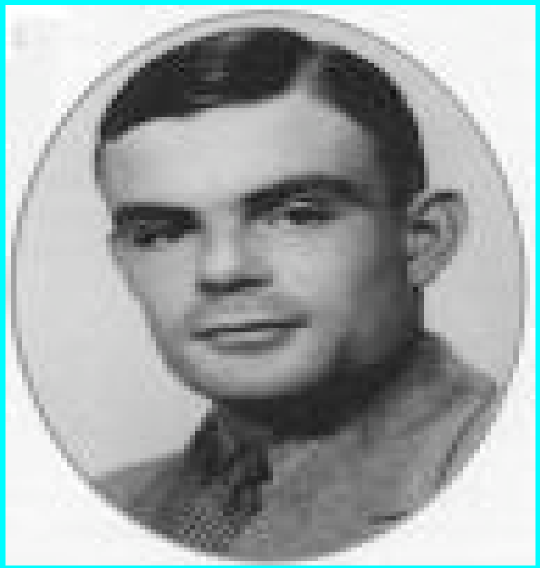


linear A – not solved

oh yes – the enigma machine

and its natural enemy – the bombe

brought to you by this man (and friends)



Hmmm… any impact on Computer Science?

Cryptosystem

- Quintuple (E, D, M, K, C)
  - M set of plaintexts
  - K set of keys
  - C set of ciphertexts
  - E set of encryption functions e: M × K → C
  - D set of decryption functions d: C × K → M

Example

- Example: Cæsar cipher
  - M = { sequences of letters }
  - K = { $i \mid i$ is an integer and $0 \leq i \leq 25$ }
  - E = { $E_k \mid k \in$ K and for all letters m,

    $E_k(m) = (m + k) \bmod 26$ }
  - D = { $D_k \mid k \in$ K and for all letters c,

    $D_k(c) = (26 + c - k) \bmod 26$ }
  - C = M

J. Caesar – total idiot?

great Caesar's ghost …

isn't this the same as the legendary rot13?

what was he thinking anyway?

Attacks

- Opponent whose goal is to break cryptosystem is the *adversary*
  - Assume adversary knows algorithm used, but not key
- Three types of attacks:
  - *ciphertext only*: adversary has only ciphertext; goal is to find plaintext, possibly key
  - *known plaintext*: adversary has ciphertext, corresponding plaintext; goal is to find key
  - *chosen plaintext*: adversary may supply plaintexts and obtain corresponding ciphertext; goal is to find key

Basis for Attacks

- Mathematical attacks
  - Based on analysis of underlying mathematics
- Statistical attacks
  - Make assumptions about the distribution of letters, pairs of letters (digrams), triplets of letters (trigrams), *etc*.
    - Called *models of the language*
  - Examine ciphertext, correlate properties with the assumptions.

Classical Cryptography

- Sender, receiver share common key
  - Keys may be the same, or trivial to derive from one another
  - Sometimes called *symmetric cryptography*
- Two basic types
  - Transposition ciphers
  - Substitution ciphers
  - Combinations are called *product ciphers*

modern functional version

- M is a message
- K is the shared secret key
- we have 2 functions
  - e(K,M) -> cybermsg -> d(K,C) -> plaintext
- so the sticky wicket is what?

Transposition Cipher

- Rearrange letters in plaintext to produce ciphertext
- Example (Rail-Fence Cipher)
  - Plaintext is `HELLO WORLD`
  - Rearrange as

```
HLOOL
ELWRD
```

  - Ciphertext is `HLOOL ELWRD`

Attacking the Cipher

- Anagramming (rearrange letters in word)
  - If 1-gram frequencies match English frequencies, but other *n*-gram frequencies do not, probably transposition
  - Rearrange letters to form *n*-grams with highest frequencies

Example

- Ciphertext: `HLOOLELWRD`
- Frequencies of 2-grams beginning with H
  - HE 0.0305
  - HO 0.0043
  - HL, HW, HR, HD < 0.0010
- Frequencies of 2-grams ending in H
  - WH 0.0026
  - EH, LH, OH, RH, DH ≤ 0.0002
- Implies E follows H

Example

- Arrange so the H and E are adjacent

```
HE
LL
OW
OR
LD
```

- Read off across, then down, to get original plaintext

Substitution Ciphers

- Change characters in plaintext to produce ciphertext
- Example (Cæsar cipher)
  - Plaintext is `HELLO WORLD`
  - Change each letter to the third letter following it (X goes to A, Y to B, Z to C)
    - Key is 3, usually written as letter 'D'
  - Ciphertext is `KHOOR ZRUOG`

Attacking the Cipher

- Exhaustive search
  - If the key space is small enough, try all possible keys until you find the right one
  - Cæsar cipher has 26 possible keys
- Statistical analysis
  - Compare to 1-gram model of English

Statistical Attack

- Compute frequency of each letter in ciphertext:

G	0.1	H	0.1	K	0.1	O	0.3
R	0.2	U	0.1	Z	0.1		

- Apply 1-gram model of English
  - Frequency of characters (1-grams) in English is on next slide

Character Frequencies

a	0.080	h	0.060	n	0.070	t	**0.090**
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	**0.130**	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

Statistical Analysis

- $f(c)$ frequency of character c in ciphertext
- $\varphi(i)$ correlation of frequency of letters in ciphertext with corresponding letters in English, assuming key is i
  - $\varphi(i) = \Sigma_{0 \le c \le 25} f(c)p(c - i)$ so here,
    $\varphi(i) = 0.1p(6 - i) + 0.1p(7 - i) + 0.1p(10 - i) + 0.3p(14 - i) + 0.2p(17 - i) + 0.1p(20 - i) + 0.1p(25 - i)$
    - $p(x)$ is frequency of character x in English

Correlation: $\varphi(i)$ for $0 \leq i \leq 25$

i	$\varphi(i)$	i	$\varphi(i)$	i	$\varphi(i)$	i	$\varphi(i)$
0	0.0482	7	0.0442	13	0.0520	19	0.0315
1	0.0364	8	0.0202	14	0.0535	20	0.0302
2	0.0410	9	0.0267	15	0.0226	21	0.0517
3	0.0575	10	0.0635	16	0.0322	22	0.0380
4	0.0252	11	0.0262	17	0.0392	23	0.0370
5	0.0190	12	0.0325	18	0.0299	24	0.0316
6	0.0660					25	0.0430

The Result

- Most probable keys, based on φ:
  - $i = 6$, $\varphi(i) = 0.0660$
    - plaintext `EBIIL TLOLA`
  - $i = 10$, $\varphi(i) = 0.0635$
    - plaintext `AXEEH PHKEW`
  - $i = 3$, $\varphi(i) = 0.0575$
    - plaintext `HELLO WORLD`
  - $i = 14$, $\varphi(i) = 0.0535$
    - plaintext `WTAAD LDGAS`
- Only English phrase is for $i = 3$
  - That's the key (3 or 'D')

Cæsar's Problem

- Key is too short
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well
    - They look too much like regular English letters
- So make it longer
  - Multiple letters in key
  - Idea is to smooth the statistical frequencies to make cryptanalysis harder

Vigènere Cipher

- Cæsar cipher, but use a phrase
- Example
  - Message `THE BOY HAS THE BALL`
  - Key `VIG`
  - Encipher using Cæsar cipher for each letter:

```
key    VIGVIGVIGVIGVIGV
plain  THEBOYHASTHEBALL
cipher OPKWWECIYOPKWIRG
```

Relevant Parts of Tableau

	G	*I*	*V*
A	G	I	V
B	H	J	W
E	L	M	Z
H	N	P	C
L	R	T	G
O	U	W	J
S	Y	A	N
T	Z	B	O
Y	E	H	T

- Tableau shown has relevant rows, columns only
- Example encipherments:
  - key V, letter T: follow V column down to T row (giving “O”)
  - Key I, letter H: follow I column down to H row (giving “P”)

Useful Terms

- *period*: length of key
  - In earlier example, period is 3
- *tableau*: table used to encipher and decipher
  - Vigènere cipher has key letters on top, plaintext letters on the left
- *polyalphabetic*: the key has several different letters
  - Cæsar cipher is monoalphabetic

Attacking the Cipher

- Approach
  - Establish period; call it *n*
  - Break message into *n* parts, each part being enciphered using the same key letter
  - Solve each part
    - You can leverage one part from another

Establish Period

- Kaskski: *repetitions in the ciphertext occur when characters of the key appear over the same characters in the plaintext*
- Example:

```
key    VIGVIGVIGVIGVIGV
plain  THEBOYHASTHEBALL
cipher OPKWWECIYOPKWIRG
```

Note the key and plaintext line up over the repetitions (underlined). As distance between repetitions is 9, the period is a factor of 9 (that is, 1, 3, or 9)

One-Time Pad

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable
  - Why? Look at ciphertext `DXQR`. Equally likely to correspond to plaintext `DOIT` (key `AJIY`) and to plaintext `DONT` (key `AJDY`) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

so at this point we have certain algorithms for symmetric encryption

- typically these algorithms do the bulk work
- as public key is too slow
- DES, 3-DES
- IDEA
- BLOWFISH
- SKIPJACK
- AES
- keys are different lengths

Overview of the DES

- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
- A product cipher
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits
- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key

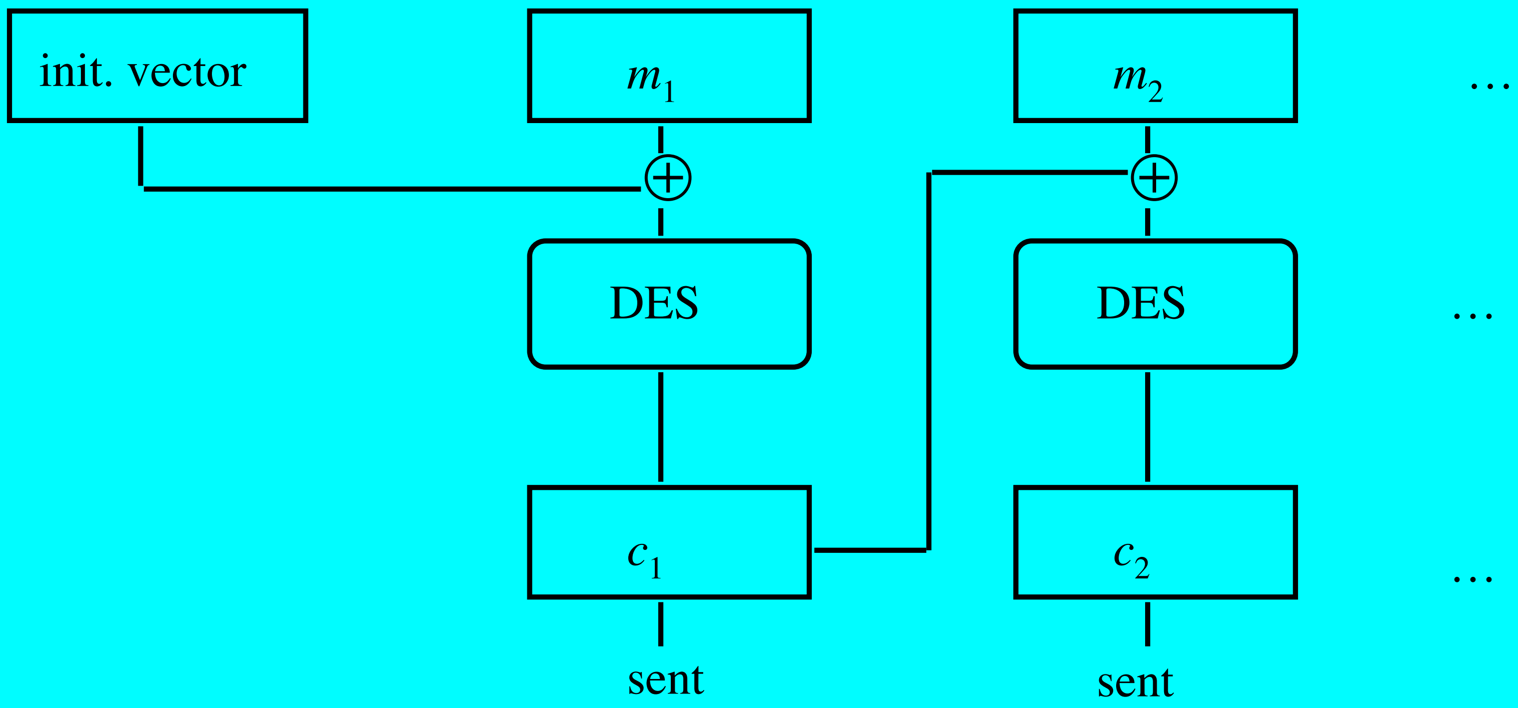
Controversy

- Considered too weak from day one
  - Diffie, Hellman said in a few years technology would allow DES to be broken in days
    - Design using 1999 technology published
  - Design decisions not public
    - S-boxes may have backdoors

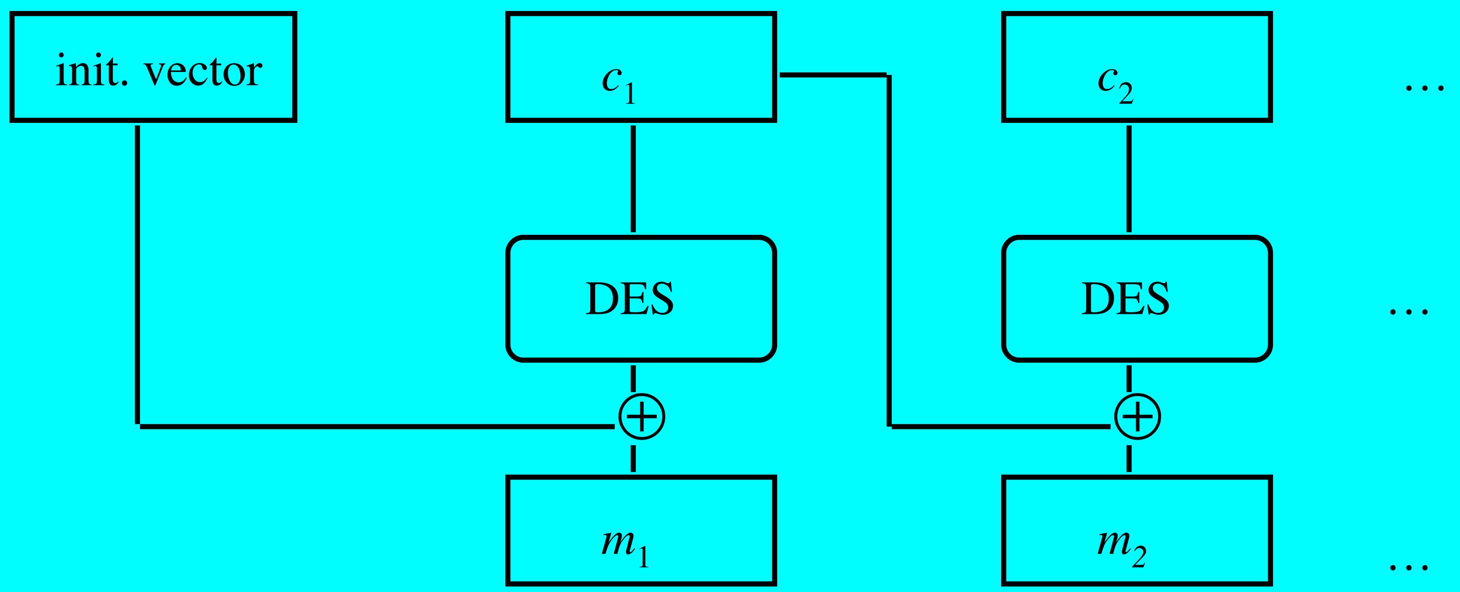
DES Modes

- Electronic Code Book Mode (ECB)
  - Encipher each block independently
- Cipher Block Chaining Mode (CBC)
  - Xor each block with previous ciphertext block
  - Requires an initialization vector for the first one
- Encrypt-Decrypt-Encrypt Mode (2 keys: k, k')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}^{-1}(\mathrm{DES}_k(m)))$
- Encrypt-Encrypt-Encrypt Mode (3 keys: k, k', k'')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}(\mathrm{DES}_{k''}(m)))$

CBC Mode Encryption



CBC Mode Decryption



Self-Healing Property

- Initial message
  - `3231343336353837 3231343336353837`
    `3231343336353837 3231343336353837`
- Received as (underlined 4c should be 4b)
  - `ef7c4cb2b4ce6f3b f6266e3a97af0e2c`
    `746ab9a6308f4256 33e60b451b09603d`
- Which decrypts to
  - `efca61e19f4836f1 3231333336353837`
    `3231343336353837 3231343336353837`
  - Incorrect bytes underlined
  - Plaintext “heals” after 2 blocks

Current Status of DES

- Design for computer system, associated software that could break any DES-enciphered message in a few days published in 1998
- Several challenges to break DES messages solved using distributed computing
- NIST selected Rijndael as Advanced Encryption Standard, successor to DES
  - Designed to withstand attacks that were successful on DES

pros/cons of symm. encryption

- pros
  - faster than public key/asymmetric usually
- cons
  - key distribution is not scalable
    - more people know secret, less of a secret
    - "Ben Franklin rule"
  - export laws have been a problem
  - Moore's law may eat a few bits a year

let us generalize a bit

- if you do cryptoanalysis,
- you don't know Key, you only know ciphertext (CC)
- we assume you know E and D (know alg)
- there is some set of K1 … KN that you can guess
- any function that say reduces the odds is good
- if you get the odds down you can bruteforce solve it with a computer
- one common design flaw: reduce the entropy of the system by making keys easy for users

Cryptographic Checksums

- Mathematical function to generate a set of k bits from a set of n bits (where $k \leq n$).
  - k is smaller then n except in unusual circumstances
- Example: ASCII parity bit
  - ASCII has 7 bits; 8th bit is "parity"
  - Even parity: even number of 1 bits
  - Odd parity: odd number of 1 bits

Example Use

- Bob receives “10111101” as bits.
  - Sender is using even parity; 6 1 bits, so character was received correctly
    - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
  - Sender is using odd parity; even number of 1 bits, so character was not received correctly

Definition

- Cryptographic checksum $h: A \rightarrow B$:
  1. For any $x \in A$, $h(x)$ is easy to compute
  2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$
  3. It is computationally infeasible to find two inputs x, $x' \in A$ such that $x \neq x'$ and $h(x) = h(x')$
     – Alternate form (stronger): Given any $x \in A$, it is computationally infeasible to find a different $x' \in A$ such that $h(x) = h(x')$.

functional forms

- md(msg) -> bit string of length N (128, 160)
- md(shared secret, msg) -> bit string
  - Alice can send the bits to Bob can use the shared secret to prove what exactly?
- consider M where M = M1, M2, M3
- we can skip M2 and generate a bit string
- md(M1), md(M3) -> bit string and expect Bob to know the same order

Collisions

- If $x \neq x'$ and $h(x) = h(x')$, x and x' are a *collision*
  - Pigeonhole principle: if there are n containers for n+1 objects, then at least one container will have 2 objects in it.
  - Application: if there are 32 files and 8 possible cryptographic checksum values, at least one value corresponds to at least 4 files

some example uses

- a MD is used as a hash
  - reduce message M of arbitrary length to N bits
- an integrity check
  - file F has an integrity check published for it
  - with public-key, we sign the integrity check
- with a shared secret we get a authentication system

note a very interesting idea lurking

- a one-way function
  - given some math function we can compute and not be able to figure out the inputs
- MD functions are not the only examples
- given x, and f(x)->z and you have z
- good luck figuring out x
- this is fundamentally important

Keys

- Keyed cryptographic checksum: requires cryptographic key
  - DES in chaining mode: encipher message, use last *n* bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
- Keyless cryptographic checksum: requires no cryptographic key
  - MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru

HMAC

- Make keyed cryptographic checksums from keyless cryptographic checksums
- *h* keyless cryptographic checksum function that takes data in blocks of *b* bytes and outputs blocks of *l* bytes. k' is cryptographic key of length *b* bytes
  - If short, pad with 0 bytes; if long, hash to length *b*
- *ipad* is 00110110 repeated *b* times
- *opad* is 01011100 repeated *b* times
- HMAC-$h(k, m) = h(k' \oplus opad \parallel h(k' \oplus ipad \parallel m))$
  - $\oplus$ exclusive or, $\parallel$ concatenation

Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone
    - Public key, private key inverses
- Idea
  - Confidentiality: encipher using public key, decipher using private key
  - Integrity/authentication: encipher using private key, decipher using public one

there are really 5 possible functions here

- encryption, decryption
- signing and verification
  - public key certificates (later chapter)
  - interesting because public key is PUBLIC
- session-key generation
  - generate a key to use for awhile
  - avoid distribution of shared secrets

Requirements

1. It must be computationally easy to encipher or decipher a message given the appropriate key
2. It must be computationally infeasible to derive the private key from the public key
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack

RSA

- Exponentiation cipher
  thus 1-way
- Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*

Security Services

- Confidentiality
  - Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

More Security Services

- Integrity
  - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message enciphered with private key came from someone who knew it

Warnings

- Encipher message in blocks considerably larger than the examples here
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning
    - Example: reverse enciphered message of text ON to get NO

Diffie-Hellman

- public key but doesn’t do signing/encryption
- allows 2 sides to create shared secrets
  - that can be used with MD and bulk sym. enc. to encode messages/pkts
- basis of many session key algorithms
- DH exchange however must be authenticated a priori to prevent MITM

like so:

- Alice/Bob a priori agree on 2 public #s
  - p: a large prime
  - g, where g < p
- pre-compute:
  - Alice Bob
  - S(a) = f(random) S(b) = f(random)
  - T(a) = g**S(a) mod p T(b) = g**S(b) mod p
- Alice sends T(a) to Bob, and Bob sends T(b) to A

DH, part 2

- post-compute of shared secret key material
  - Alice Bob
  - S(secret) – T(b) ** S(a) mod p

    S(secret) = T(a) ** S(b) mod p
- never mind the proof:
  - S(secret) gives the same number of shared secret bits on both sides
  - can be used with MD or symmetric enc. algorithm

pros/cons of public-key crypto

- usually quite slow in software
  - more likely do this:
  - create random key N for bulk encryption
  - encrypt M with N using E(M,N)
  - now encrypt N with public key crypto
- thus in networking protocols some combo of algorithms is likely
  - AES, SHA, RSA (or something)
- existence of session-key alg, or signatures a PRO!

Key Points

- Classical cryptosystems encipher and decipher using the same key
  - Or one key is easily derived from the other
- Public key cryptosystems encipher and decipher using different keys
  - Computationally infeasible to derive one from the other
- Cryptographic checksums provide a check on integrity
  - used for authentication, session-key generation and in point of fact are very useful

policy considerations

- .so given
  - an enterprise
  - a govt. security agency
  - it's WWII: the US 101st Airborne Division in Bastogne during the Battle of the Bulge
  - a hospital worried about S/OX or HIPPA
  - a university
- what policy considerations may exist re say crypto in and of itself?

and furthermore

- what can crypto do and what can it NOT do?
- what is key escrow and did you think about that in your policy considerations?
- do you allow users to bring laptops on site, and insist on encryption between your home and branch campuses?