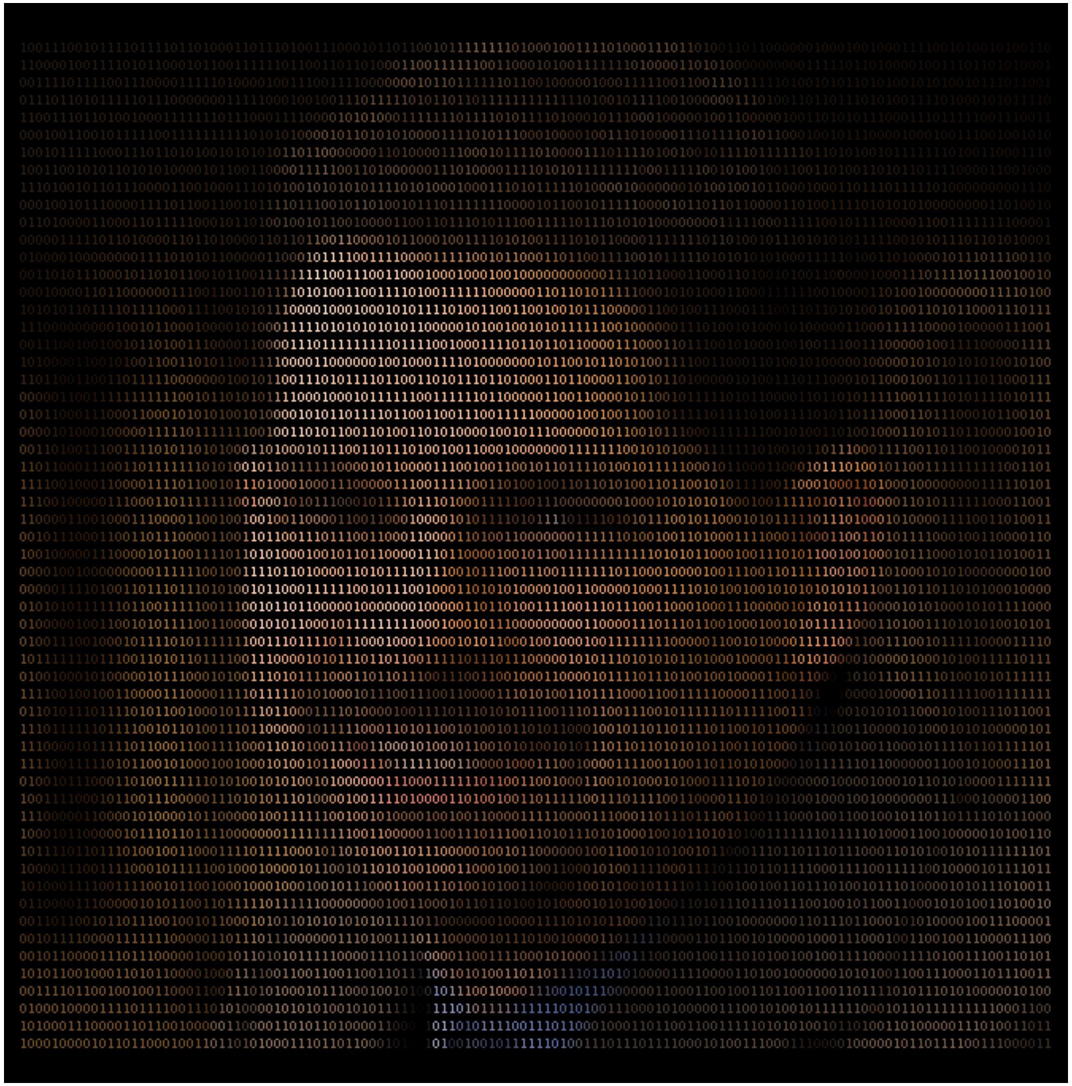
PROTECTING DIGITAL INFORMATION

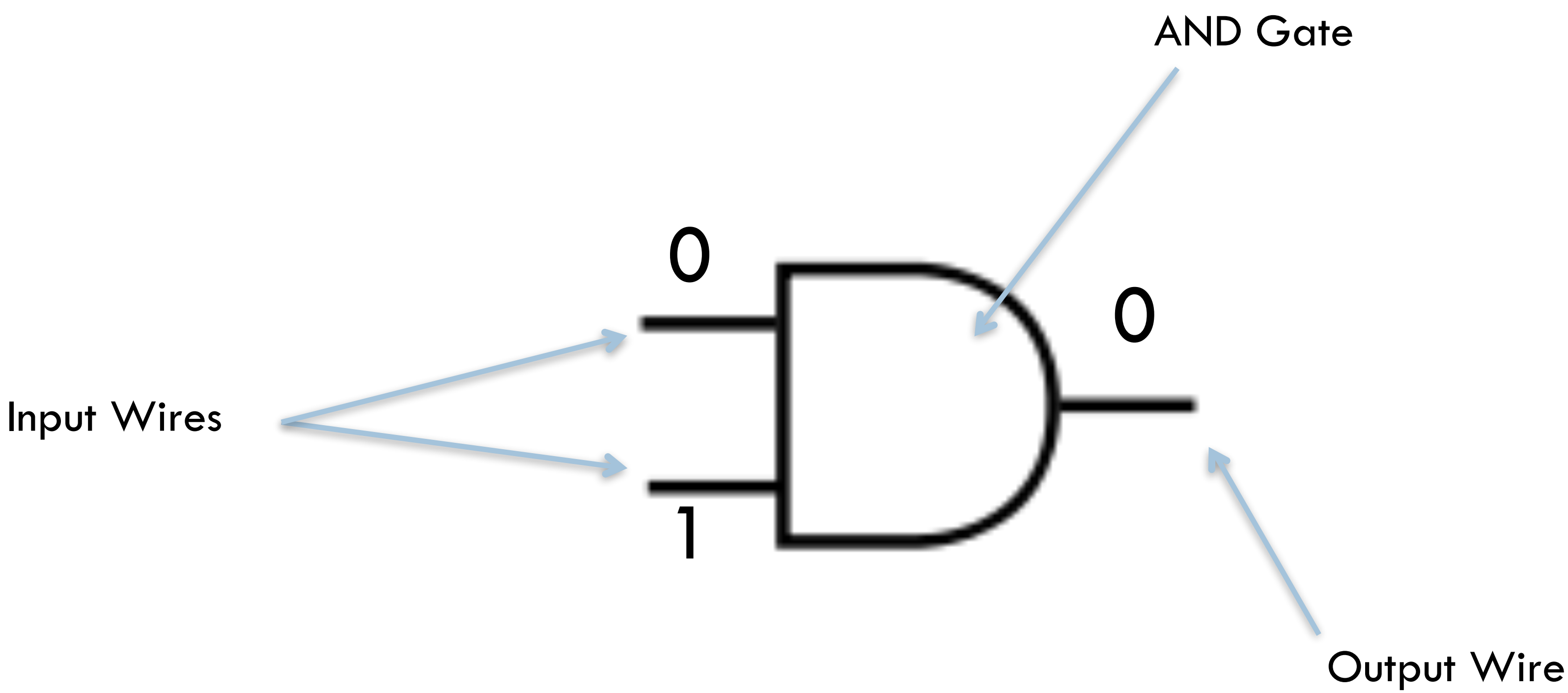
Roadmap: Fall 2017

But First, An Aside: This is Misleading

This is More Like It

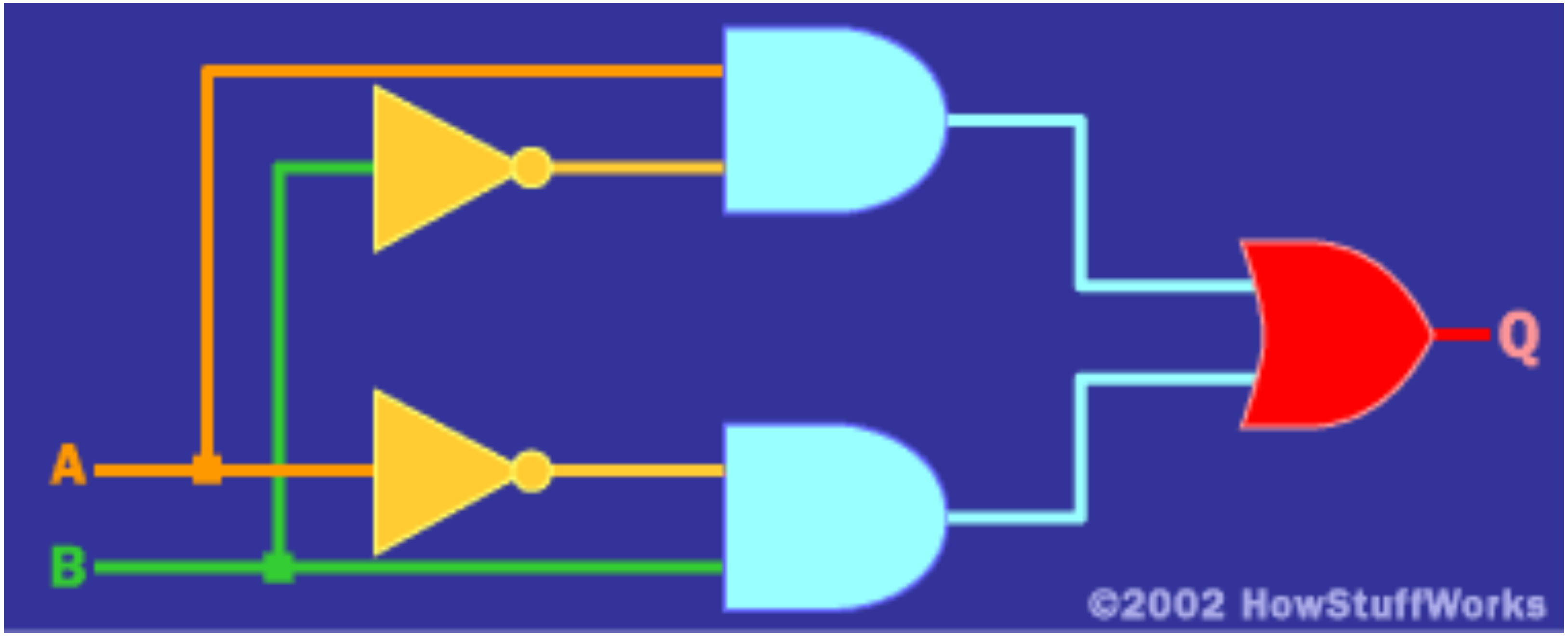


Inside the Computer: Gates



0's & 1's represent low & high voltage, respectively, on the wires

Inside the Computer: Gates



All logic performed inside the computer is performed on zeros and ones, and results are stored as zeros and ones.

The Decimal Number System

- Deci- (ten)

- Base is ten
  - first (rightmost) place: ones (i.e., 10^0)
  - second place: tens (i.e., 10^1)
  - third place: hundreds (i.e., 10^2)
  - …

- Digits available: 0, 1, 2, …, 9 (ten total)

Example: your favorite number…

$8{,}675{,}309 = 8 \times 10^6 + 6 \times 10^5 + \ldots + 9 \times 10^0$

The Binary Number System

- Bi- (two)
  - bicycle, bicentennial, biphenyl
- Base two
  - first (rightmost) place: ones (i.e., 2^0)
  - second place: twos (i.e., 2^1)
  - third place: fours (i.e., 2^2)
  - …
- Digits available: 0, 1 (two total)

Example

- $8,675,309_{10}$

  $=$

  $100001000101111111101101_2$

- Fewer available digits in binary:
  more space required for representation

Converting Binary to Decimal

- For each 1, add the corresponding power of two

- $1010010111101_2 = 1 \text{ x } 2^{12} + 0 \text{ x } 2^{11} + 1 \text{ x } 2^{10} + 0 \text{ x } 2^9 + \ldots + 1 \text{ x } 2^2 + 0 \text{ x } 2^1 + 1 \text{ x } 2^0 = 5309_{10}$

Now You Get The Joke

THERE ARE 10 TYPES OF PEOPLE IN THE WORLD:

THOSE WHO CAN COUNT IN BINARY

AND THOSE WHO CAN'T

More About Binary

- How many different things can you represent using binary:

- with only one slot (i.e., one bit)? 2
- with two slots (i.e., two bits)? $2^2 = 4$
- with three bits? $2^3 = 8$
- with *n* bits? 2^n

Representing Different Information

- So far, everything has been a natural number
  - What about decimal numbers? Negative numbers?

- What about characters? Punctuation?

- Idea:
  - put all the characters, punctuation in order
  - assign a unique number to each
  - done! (we know how to represent numbers)

ASCII: American Standard Code for Information Interchange

1 ┌	33 !	65 A	97 a	129 ▯	161 ¡	193 Á	225 á
2 ┐	34 "	66 B	98 b	130 ‚	162 ¢	194 Â	226 â
3 └	35 #	67 C	99 c	131 ƒ	163 £	195 Ã	227 ã
4 ┘	36 $	68 D	100 d	132 „	164 ¤	196 Ä	228 ä
5 |	37 %	69 E	101 e	133 …	165 ¥	197 Å	229 å
6 –	38 &	70 F	102 f	134 †	166 ¦	198 Æ	230 æ
7 •	39 '	71 G	103 g	135 ‡	167 §	199 Ç	231 ç
8 ◘	40 (	72 H	104 h	136 ˆ	168 ¨	200 È	232 è
9	41)	73 I	105 i	137 ‰	169 ©	201 É	233 é
10	42 *	74 J	106 j	138 Š	170 ª	202 Ê	234 ê
11 ♂	43 +	75 K	107 k	139 ‹	171 «	203 Ë	235 ë
12 □	44 ,	76 L	108 l	140 Œ	172 ¬	204 Ì	236 ì
13	45 -	77 M	109 m	141 ▯	173 -	205 Í	237 í
14 ♫	46 .	78 N	110 n	142 Ž	174 ®	206 Î	238 î
15 ☼	47 /	79 O	111 o	143 ▯	175 ¯	207 Ï	239 ï
16 ┼	48 0	80 P	112 p	144 ▯	176 °	208 Đ	240 ð
17 ◄	49 1	81 Q	113 q	145 '	177 ±	209 Ñ	241 ñ
18 ↕	50 2	82 R	114 r	146 '	178 ²	210 Ò	242 ò
19 ‼	51 3	83 S	115 s	147 "	179 ³	211 Ó	243 ó
20 ¶	52 4	84 T	116 t	148 "	180 ´	212 Ô	244 ô
21 ┴	53 5	85 U	117 u	149 •	181 µ	213 Õ	245 õ
22 ┬	54 6	86 V	118 v	150 –	182 ¶	214 Ö	246 ö
23 ┤	55 7	87 W	119 w	151 —	183 ·	215 ×	247 ÷
24 ↑	56 8	88 X	120 x	152 ˜	184 ¸	216 Ø	248 ø
25 ├	57 9	89 Y	121 y	153 ™	185 ¹	217 Ù	249 ù
26 →	58 :	90 Z	122 z	154 š	186 º	218 Ú	250 ú
27 ←	59 ;	91 [	123 {	155 ›	187 »	219 Û	251 û
28	60 <	92 \	124 |	156 œ	188 ¼	220 Ü	252 ü
29	61 =	93]	125 }	157 ▯	189 ½	221 Ý	253 ý
30	62 >	94 ^	126 ~	158 ž	190 ¾	222 Þ	254 þ
31	63 ?	95 _	127 ▯	159 Ÿ	191 ¿	223 ß	255 ÿ
32	64 @	96 `	128 €	160	192 À	224 à	

The Problem with ASCII

- What about Greek characters? Chinese?
- UNICODE: use 16 bits
- How many characters can we represent?

The Problem with ASCII

- What about Greek characters? Chinese?
- UNICODE: use 16 bits
- How many characters can we represent?
- $2^{16} = 65{,}536$

You Control The Information

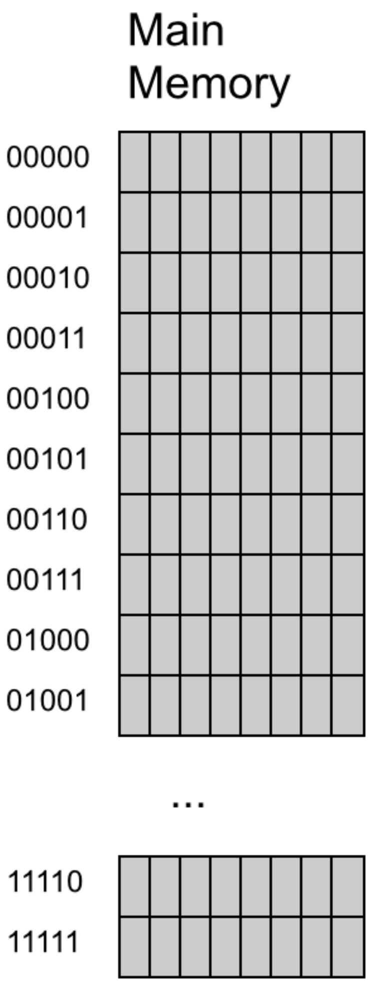
- What is this?   01001101

You Control The Information

- What is this? 01001101
- Depends on how you interpret it:
- $01001101_2 = 77_{10}$
- 01001101_2 = 'M'
- 01001101_{10} = one million one thousand one hundred and one
- 01001101 = a font code for a Microsoft Word document
- When information stored in computers, one must be clear on both representation <u>and</u> interpretation

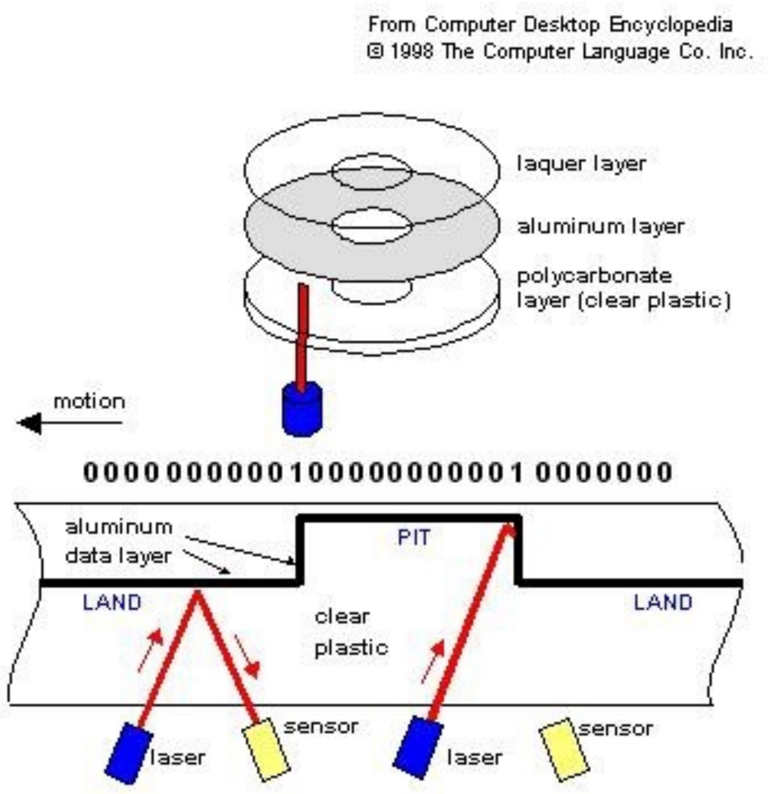
So What Does Memory Look Like?

- First, a little terminology:
  - A single one or zero is called a *bit*
    - Short for "binary digit"
  - 8 bits is a *byte*
    - My laptop has roughly 500 billion bytes of memory
- Every byte of memory has an address (so we know which byte of memory we are using/discussing)
  - See example at right

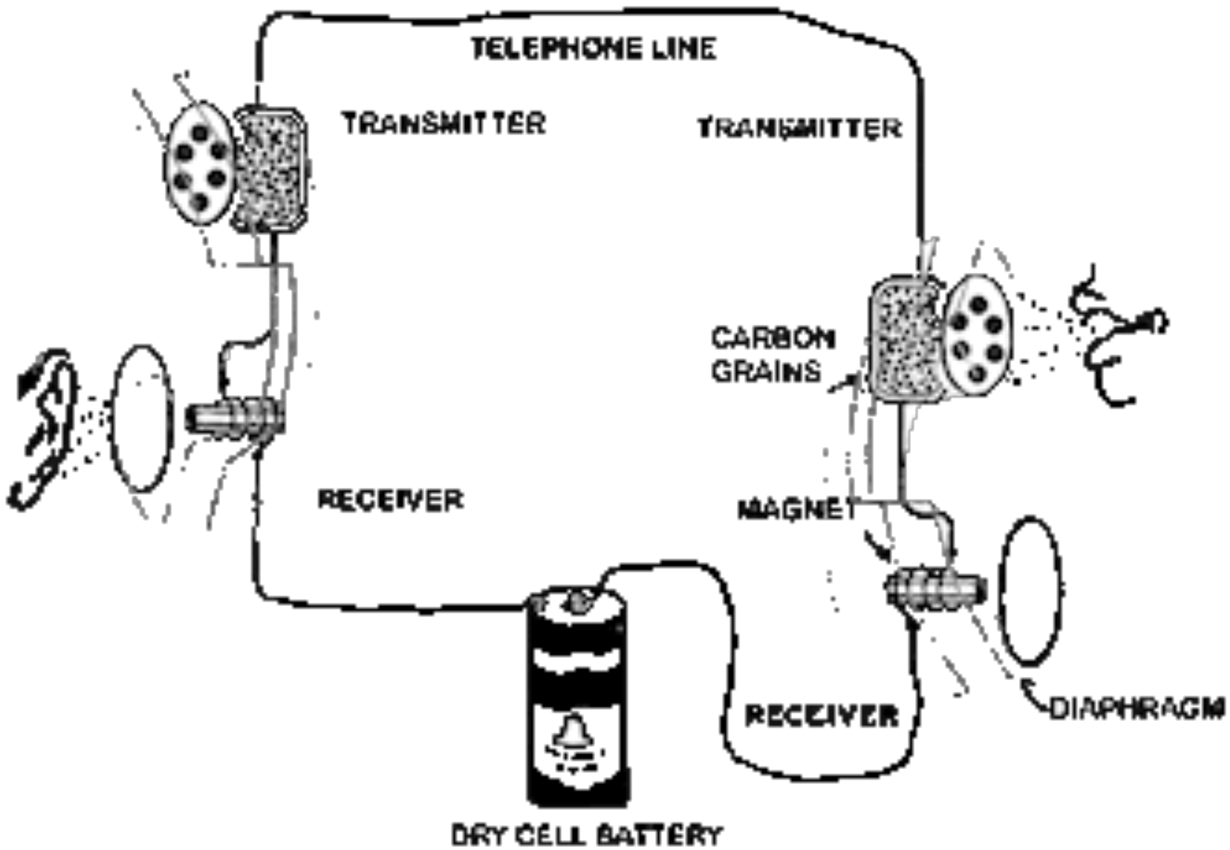


Why Just 0 and 1?

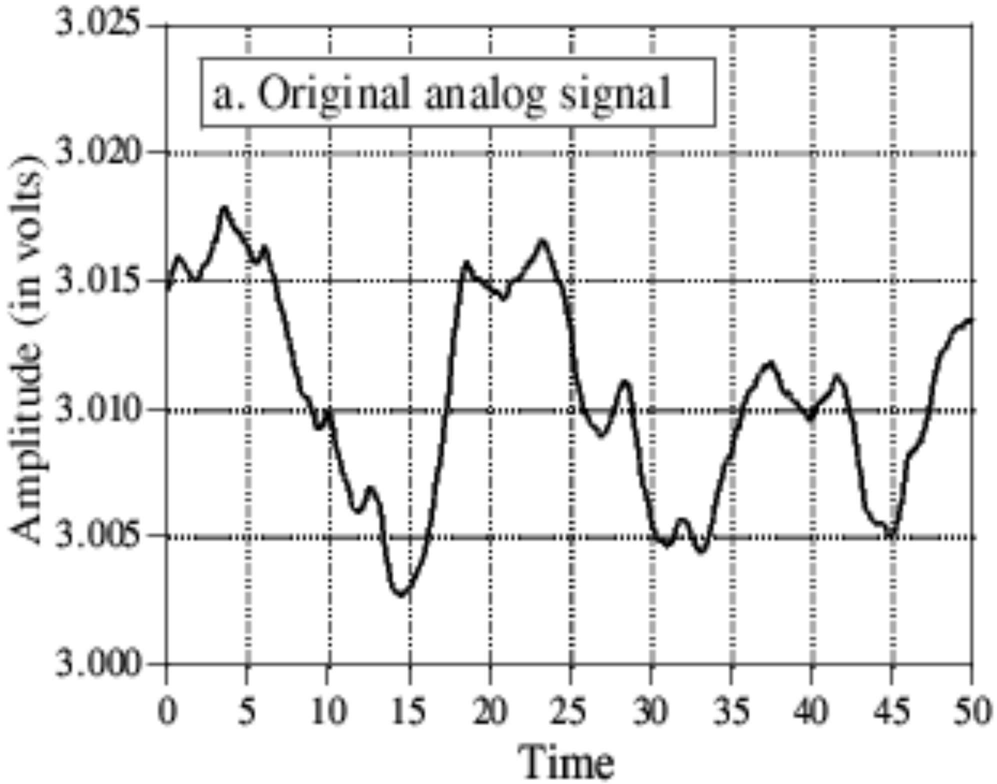
- Easy to represent
  - low voltage vs high voltage
  - Reflective pit vs non-reflective pit
  - N/S orientation of magnetic element vs S/N orientation of magnetic element



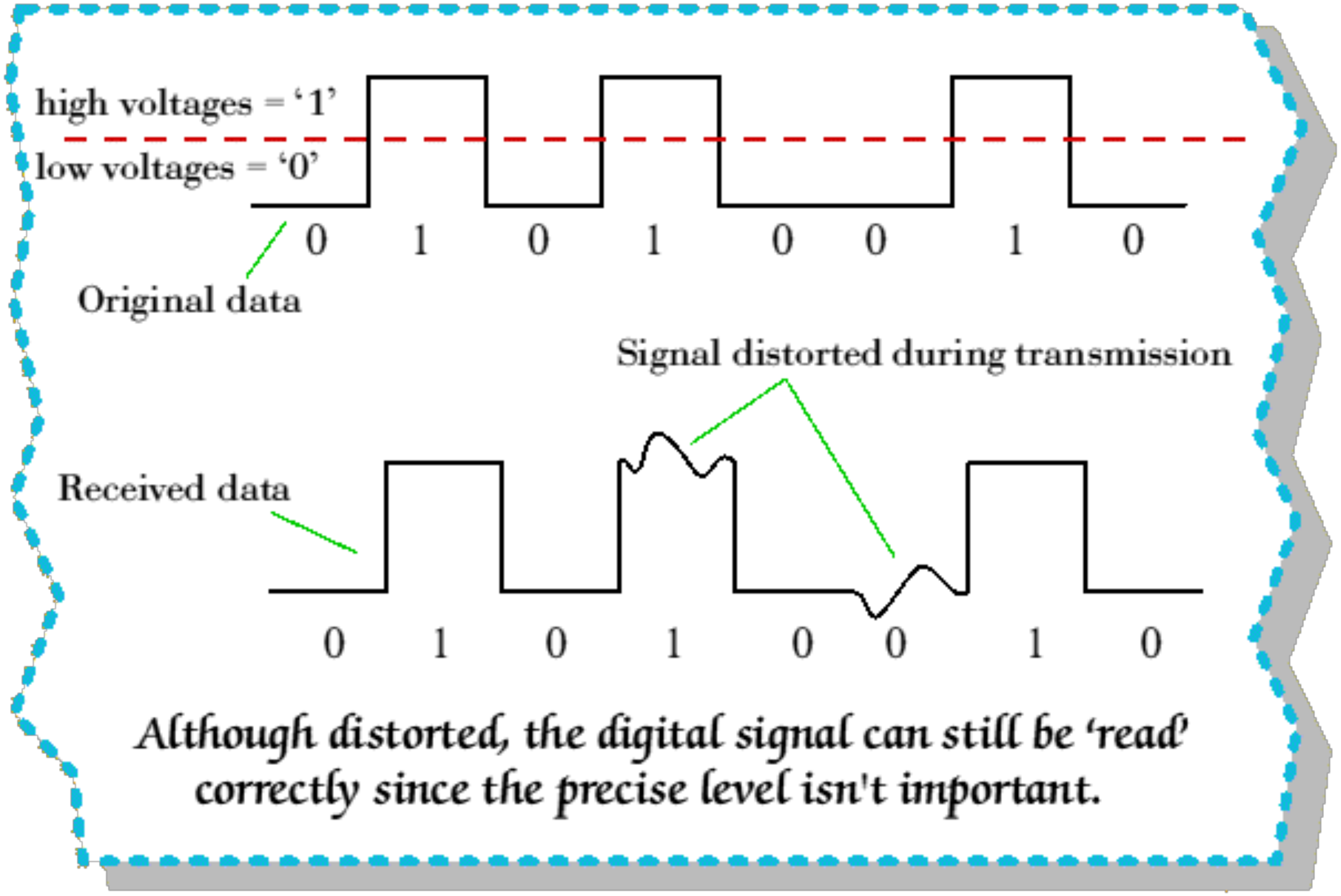
What's so Great about Digital?



What's so Great about Digital?



What's so Great about Digital?



This is another reason why we use only binary — easier signal recovery!
In reality, all sort of error correcting codes are used to aid in this

But Back to the Primary Issue

How do we protect stored data?
One answer: Encryption

Definition

- Cryptology is the study of secret writing
- Concerned with developing algorithms which may be used:
  - To conceal the content of some message from all except the sender and recipient (*privacy* or *secrecy*), and/or
  - Verify the correctness of a message to the recipient (*authentication* or *integrity*)
- The basis of many technological solutions to computer and communication security problems

Terminology

- *Plaintext*: The original intelligible message
- *Ciphertext*: The transformed message
- *Cipher*: An algorithm for transforming an intelligible message into one that is unintelligible

Terminology (cont).

- *Key*: Some critical information used by the cipher, known only to the sender & receiver
- *Encrypt*: The process of converting plaintext to ciphertext using a cipher and a key
- *Decrypt*: The process of converting ciphertext back into plaintext using a cipher and a key

- *Cryptanalysis*: The study of principles and methods of transforming an unintelligible message back into an intelligible message ***without knowledge of the key!***

Concepts

- Encryption: Mapping plaintext to ciphertext using the specified key:

$$C = E_K(P)$$

- Decryption: Mapping ciphertext to plaintext using the specified key:

$$P = E_K^{-1}(C) = D_K(C)$$

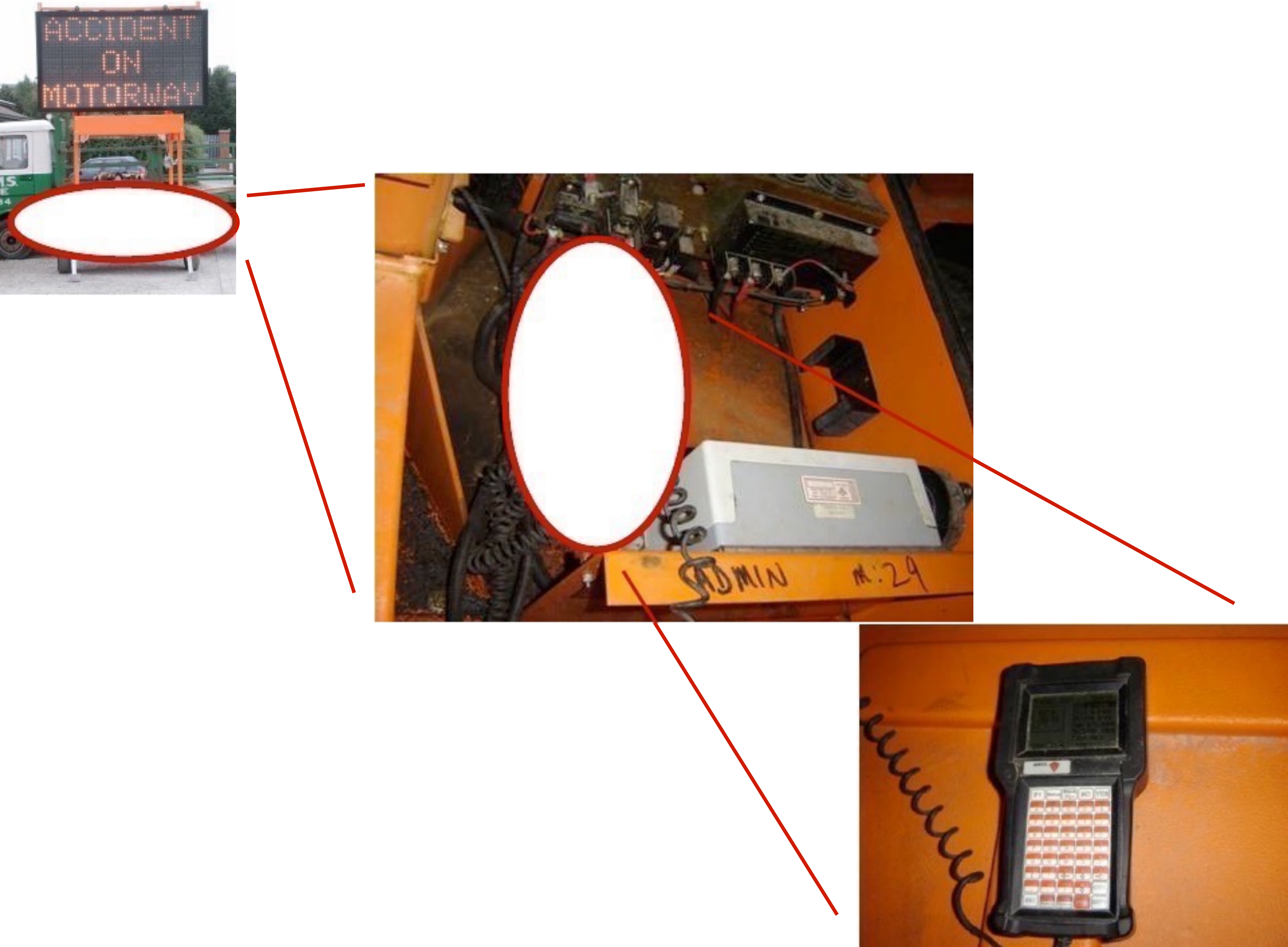
Concepts (cont.)

- *Key*: Is the parameter which selects which exact transformation is used, and is selected from a *keyspace* $\mathcal{K}$
- We usually assume the cryptographic system is public, and only the key is secret information
  - Why?

Concepts (cont.)

- *Key*: Is the parameter which selects which exact transformation is used, and is selected from a *keyspace* $\mathcal{K}$
- We usually assume the cryptographic system is public, and only the key is secret information
  - Why?
  - Because if the security of your system is based on the adversary not knowing how your system works, history shows you'll be greatly disappointed — called "security through obscurity"
  - Instead: build system so securely that even if the adversary has the blueprints to the system (but not the key), he/she still can't break in!


ACCIDENT
ON
MOTORWAY


ACCIDENT
ON
MOTORWAY
ADMIN


CAUTION!
ZOMBIES!
AHEAD!!!

RUN FOR
COLD
CLIMATES

THE END
IS NEAR!
!!!!!!!!


RAPTORS
AHEAD
CAUTION
MARSH

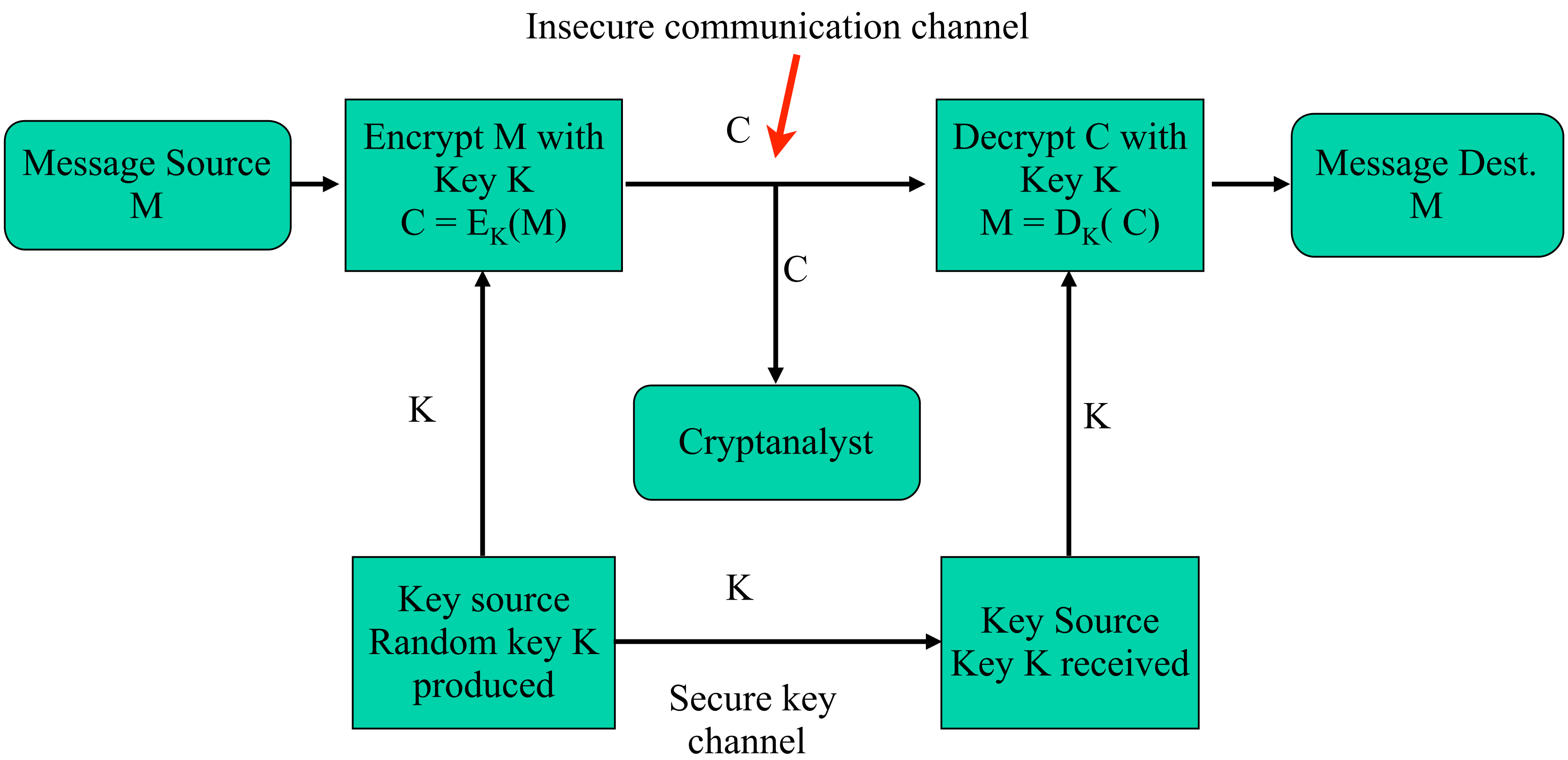

TRAPPED
IN SIGN
FACTORY


SEND
HELP!

Rough Classification

- Symmetric-key encryption algorithms
  - Sender and recipient (typically) share same key
  - Fast
  - Key management issues (how do you get same key to both)
- Public-key encryption algorithms
  - Sender and recipient use different keys
  - Much slower
  - Different key management issues (we'll discuss briefly)
- Digital signature algorithms — works like a signature
- Hash functions — used to guarantee that document has not been changed in transit, and that document was sent by person who claims to have sent it

Symmetric-Key Encryption System



All "traditional" encryption algorithms are symmetric key

Exhaustive Key Search

- Always theoretically possible to simply try every key
  - So keys are chosen long enough so that this is not computationally feasible

- Most basic attack, directly proportional to key size

- *Assumes attacker can recognize when plaintext is found!!*

Exhaustive Key Search

- Fastest Supercomputer (Wikipedia): As per June 2012, IBM Sequoia
  - 16.31 Petaflops = 16.31×10^{15} FLOPS
- Number of FLOPS required per key check
  - Optimistically estimated at 1000
- Number of key checks per second
  - $16.31 \times 10^{15} / 1000 = 16.31 \times 10^{12}$
- Number of seconds in a year
  - 31,536,000
- Number of years to crack 128-bit AES $= 6.61 \times 10^{17}$

Example: The Caeser Cipher

- 2000 years ago Julius Caesar used a simple substitution cipher, now known as the *Caesar cipher*
  - First attested use in military affairs (e.g., Gallic Wars)
- Concept: replace each letter of the alphabet with another letter that is k letters after original letter
- Example: replace each letter by 3rd letter after

L FDPH L VDZ L FRQTXHUHG

I CAME I SAW I CONQUERED

General Caesar Cipher

- Can use any shift from 1 to 25
  - I.e. replace each letter of message by a letter a fixed distance away
- Specify *key letter* as the letter that plaintext A maps to
  - E.g. a key letter of F means A maps to F, B to G, ... Y to D, Z to E, I.e. shift letters by 5 places
- Hence have 26 (25 useful) ciphers
  - Hence breaking this is easy. Just try all 25 keys one by one.

Mixed Monoalphabetic Cipher

- Rather than just shifting the alphabet, could shuffle (jumble) the letters arbitrarily
- Each plaintext letter maps to a different random ciphertext letter
- Key is 26 letters long

```
Plain:       ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:      DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext:   IFWEWISHTOREPLACELETTERS
Ciphertext:  WIRFRWAJUHYFTSDVFSFUUFYA
```

Security of Mixed Monoalphabetic Cipher

- With a key of length 26, now have a total of 26! ~ 4 x 10^{26} keys
  - A computer capable of testing16.31 x 10^{12} keys every second would take more than 777,677 years to test them all.
  - On average, expect to take more than 388,000 years to find the key.

- With so many keys, might think this is secure…but you'd be wrong (your laptop could probably break it in under a minute)

Security of Mixed Monoalphabetic Cipher

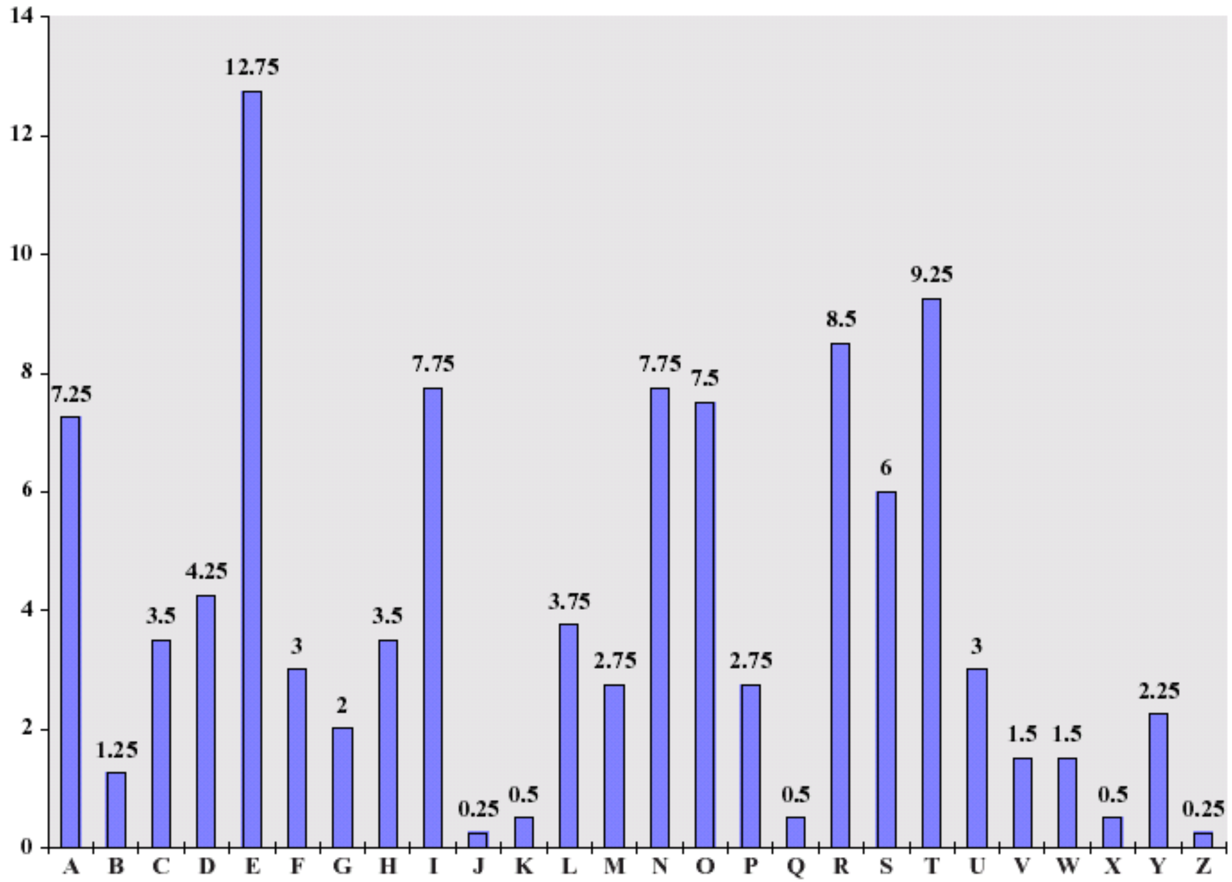
- Variations of the monoalphabetic substitution cipher were used in government and military affairs for many centuries into the middle ages
- The method of breaking it, *frequency analysis* was discovered by Arabic scientists
- All monoalphabetic ciphers are susceptible to this type of analysis

Language Redundancy and Cryptanalysis

- Human languages are redundant
- Letters in a given language occur with different frequencies.
  - Ex. In English, letter e occurs about 12.75% of time, while letter z occurs only 0.25% of time.
- In English the letters e is by far the most common letter
- So, calculate frequencies of letters occurring in ciphertext and use this as a guide to guess at the letters. This greatly reduces the key space that needs to be searched.

Language Redundancy and

- Tables of single, double, and triple letter frequencies are also available



Other Languages

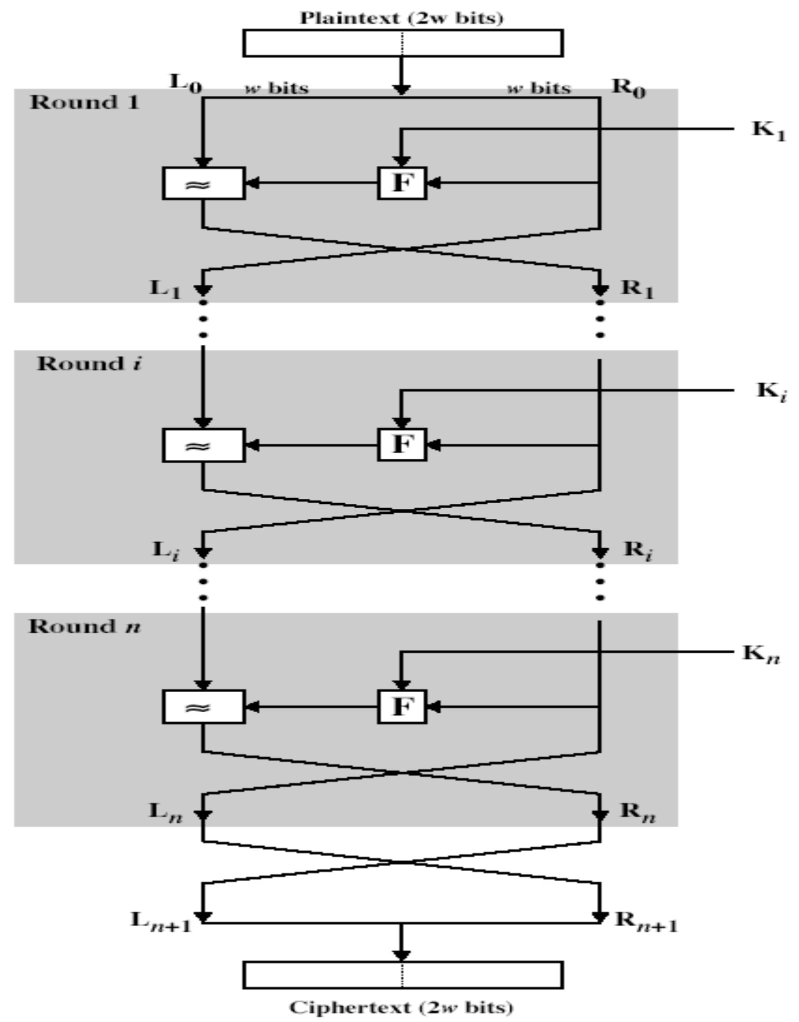
- Natural languages all have varying letter frequencies
- Languages have different numbers of letters (cf. Norwegian)
- Can take sample text and count letter frequencies
- Seberry (1st Ed) text, Appendix A has counts for 20 languages. Hits most European & Japanese & Malay

Polyalphabetic Ciphers

- Might guess that one approach to improving security is to use multiple cipher alphabets, hence the name polyalphabetic ciphers
- Makes cryptanalysis harder since have more alphabets to guess and because flattens frequency distribution
- Use a key to select which alphabet is used for each letter of the message
  - ith letter of key specifies ith alphabet to use
- Use each alphabet in turn
- Repeat from start after end of key is reached

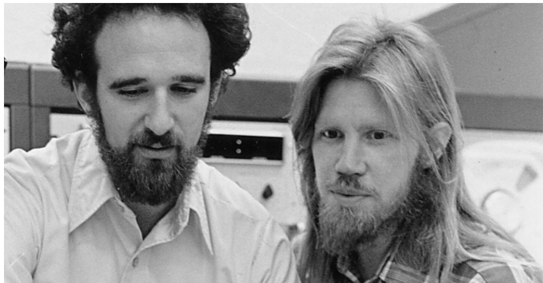
- Bottom line: straight substitution ciphers are not secure!

General Symmetric Cipher Structure



Public Key Cryptography

- Absolutely remarkable idea
  - So remarkable, that when the paper ("New Directions in Cryptography", Diffie and Hellman) proposing it was submitted, it was rejected four times (because cryptographers at the time thought it was not possible)
  - Interesting side note: paper published in 1976. It was later revealed that both MI6 and the NSA had discovered this independently almost 8 years earlier.



Public Key Cryptography

- Among the things it makes possible
  - Two people, Alice and Bob, in a crowded room can shout information to each other for all to hear. At the end, Alice and Bob will share a secret key that no one else in the room knows!
  - A person can send an encrypted message to a person they have never met, without having previously exchanged encryption keys!

Public Key Cryptography

- Key idea (no pun intended): split the encryption key
- A person, say Alice, who wishes to perform encryption has a key consisting of two parts: a public part and a private part, $<K_{pu}, K_{pr}>$
- The public part is published for all the world to see
- The private part is known only to Alice

Public Key Cryptography

- A message encrypted with K_{pu} can only be decrypted with K_{pr} and vice-versa!
  - So Bob can send a message that only Alice can read by encrypting with K_{pu}
  - And any message that can be decrypted using K_{pu} could only have been encrypted by Alice (because she is the only one who knows K_{pr})
- How? Various ciphers exist. Most are slow, because encryption and decryption involve calculations using very large numbers
  - So in practice, public key used to encrypt only small amounts of data…

Public Key Cryptography

- …like a symmetric encryption key!
- In practice: public key cryptography is used to transmit symmetric keys between parties that have more data to encrypt
  - Your web browsers do this all the time (e.g., the lock icon): generate a random symmetric key, use public key crypto to transmit the key, then both parties use this symmetric key with a symmetric cipher to encrypt/decrypt the real data that needs to be sent

Cryptography

- But there's a problem with all crypto
- Bruce Schneier: Strong cryptography is very powerful when it is done right, but it is not a panacea. Focusing on the cryptographic algorithms while ignoring other aspects of security is like defending your house not by building a fence around it, but by putting an immense stake into the ground and hoping that the adversary runs right into it. Smart attackers will just go around the algorithms.

Cryptography

- Among the issues:
  - Storing data in encrypted form makes it difficult (and/or inefficient) to do many of the things that people and organizations like to do with their data
    - Search it
    - Perform analytics on it
    - In general, use it
- So when using it, it really needs to be stored in plaintext
  - And any good adversary knows this

Cryptography

- Important Note: This does NOT mean that data should never be stored encrypted!
  - Backups can be safely stored encrypted
  - Old data should be stored encrypted
    - Typically rarely used, but most definitely NOT worthless
      - E.g., Financial transaction records, credit card numbers, legal files
    - Data that is required, by law, to be retained
      - See Sarbanes-Oxley, various sunshine laws, etc.

Cryptography

- And cryptography is extremely useful for keeping data confidential while it is being transmitted
- Finally, there has been a great deal of research done over the past fifteen or so years on ways of encrypting so that operations (e.g., searching, sorting, whatever) can be performed on the encrypted data
  - Remarkable result: Dr. Craig Gentry (currently at IBM, formerly Stanford graduate student) showed in his 2010 doctoral dissertation how to encrypt data in such a way that any operation can be performed on the encrypted data! (This will win Turing Award)
    - His method is not practical at this time (nor soon)