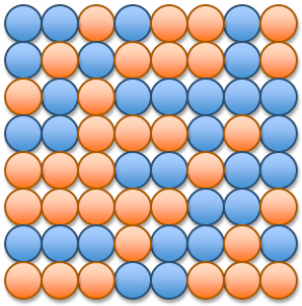


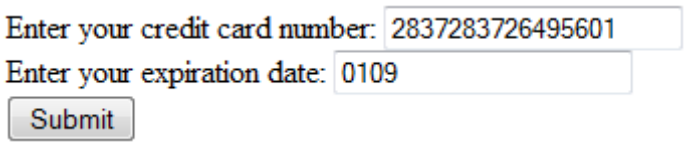
UNIT 13B
The Internet: Encryption

Data Privacy

- Suppose you send an email to a friend.
  - Who has access to that email?
  - What if you encrypt the message so it is private?
  - Can someone who intercepts the email decrypt it?
  - Who would be against email encryption?
- On the Internet and any computer network, any transmitted data can be intercepted and copied.

A Shady Example

- I want to make a purchase online and click a link that takes me to http://www.sketchystore.com/checkout.jsp
- What I see in my browser:



A Shady Example (cont'd)

- When I press SUBMIT, I send this:

```
POST /purchase.jsp HTTP/1.1
Host: www.sketchystore.com
User-Agent: Mozilla/4.0
Content-Length: 48
Content-Type: application/x-www-form-
  urlencoded
userid=tom&creditcard=2837283726495601&
  exp=0109
```

A Shady Example (cont'd)

- If this information is sent unencrypted, who has access to my credit card number?
  - Other people who can connect to my wireless ethernet?
  - Other people physically connected to my wired ethernet?
- When I send a letter through the mail, it passes through the hands of many mail carriers. What keeps them from reading my mail?
  - What if I send a postcard?
- Packets are passed from router to router.
  - All those routers have access to my data.

Encryption

- We need to encrypt (encode) our data so others can't understand it (easily) except for the person who is supposed to receive it.
- Simple encryption scheme:
- Shift every letter forward by 1

  A → B, B → C, ..., Z → A
- Example:
  MESSAGE → NFTTBHF
- Can you decrypt TFDSFU?

Caesar Cipher

- Shift forward *n* letters.
- For example, shift forward 3 letters:
  A ⟶ D, B ⟶ E, ..., Z ⟶ C
  - This is a Caesar cipher using a **key** of 3.
- MESSAGE ⟶ PHVVDJH
- How can we decode this:
  DEEDUSEKBTFEIIYRBOTUSETUJXYI

Caesar Cipher (cont'd)

DEEDUSEKBTFEIIYRBOTUSETUJXYI	QRRQHFRXOGSRVVLEOBGHFRGHWKLV
EFFEVTFLCUGFJJZSCPUVTFUVKYZJ	RSSRIGSYPHTSWWMFPCHIGSHIXLMW
FGGFWUGMDVHGKKATDQVWUGVWLZAK	STTSJHTZQIUTXXNGQDIJHTIJYMNX
GHHGXVHNEWIHLLBUERWXVHWXMABL	TUUTKIUARJVUYYOHREJKIUJKZNOY
HIIHYWIOFXJIMMCVFSXYWIXYNBCM	UVVULJVBSKWVZZPISFKLJVKLAOPZ
IJJIZXJPGYKJNNDWGTYZXJYZOCDN	VWWVMKWCTLXWAAQJTGLMKWLMBPQA
JKKJAYKQHZLKOOEXHUZAYKZAPDEO	WXXWNLXDUMYXBBRKUHMNLXMNCQRB
KLLKBZLRIAMLPPFYIVABZLABQEFP	XYYXOMYEVNZYCCSLVINOMYNODRSC
LMMLCAMSJBNMQQGZJWBCAMBCRFGQ	YZZYPNZFWOAZDDTMWJOPNZOPESTD
MNNMDBNTKCONRRHAKXCDBNCDSGHR	ZAAZQOAGXPBAEEUNXKPQOAPQFTUE
NOONECOULDPOSSIBLYDECODETHIS	ABBARPBHYQCBFFVOYLQRPBQRGUVF
OPPOFDPVMEQPTTJCMZEFDPEFUIJT	BCCBSQCIZRDCGGWPZMRSQCRSHVWG
PQQPGEQWNFRQUUKDNAFGEQFGVJKU	CDDCTRDJASEDHHXQANSTRDSTIWXH

- How long would it take a computer to try all 25 shifts?

Vigenère Cipher

- Shift different amount for each letter.

```
   ABCDEFGHIJKLMNOPQRSTUVWXYZ
A  ABCDEFGHIJKLMNOPQRSTUVWXYZ
B  BCDEFGHIJKLMNOPQRSTUVWXYZA
C  CDEFGHIJKLMNOPQRSTUVWXYZAB
D  DEFGHIJKLMNOPQRSTUVWXYZABC
E  EFGHIJKLMNOPQRSTUVWXYZABCD
F  FGHIJKLMNOPQRSTUVWXYZABCDE   etc.
```



```
   ABCDEFGHIJKLMNOPQRSTUVWXYZ
A  ABCDEFGHIJKLMNOPQRSTUVWXYZ
B  BCDEFGHIJKLMNOPQRSTUVWXYZA
C  CDEFGHIJKLMNOPQRSTUVWXYZAB
D  DEFGHIJKLMNOPQRSTUVWXYZABC
E  EFGHIJKLMNOPQRSTUVWXYZABCD
F  FGHIJKLMNOPQRSTUVWXYZABCDE
...
```

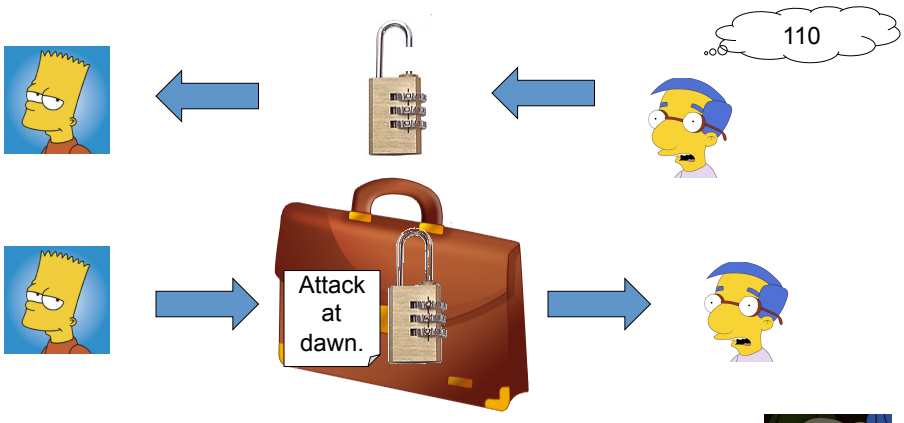
- Pick a secret key: DECAF
- Write Key over and over: DECAFDECAFDE
- Message: ATTACKATDAWN
- Encrypted: DXVAHNEVDFZR

Vigenère Cipher (cont'd)

- If you don't know the key, how could you decrypt the message?
- What makes a good key?
- How do you get the secret key to the receiver?

"Secure communication was practical only for people who could arrange to meet beforehand, or who had access to a prior method of secure communication (such as military couriers) for carrying the key between them. If Internet communications had to proceed on this assumption, electronic commerce never could have gotten off the ground."
(from *Blown To Bits*)

Secure Transmission: Locks



What if someone steals the lock?
What if someone steals the locked briefcase?

Locks

- Milhouse knows the combination to one set of locks, which can be used to send messages that only Milhouse can read.
- Bart knows the combination to another set of locks, which can be used to send messages that only Bart can read.
- This works because locks are easy to open if you know the combination, and locks are hard to open if you don't know the combination.

Locks

- How do you open a lock, if you don't know the combination?
  - If there are 3 digits, how many combinations do we need to try? (worst case)
- Suppose someone can crack my 3-digit combo lock in 15 minutes, by trying every combination. Do I give up on combo locks? No, I use more digits!
  - How long to crack a 6-digit lock at this rate? 10 days
  - How long to crack a 12-digit lock at this rate? 30,000 years
- Locks on the Internet: Public key encryption

RSA Encryption

- Current encryption technique for transmitting data on the Internet
  - Named after its inventors: Rivest, Shamir and Adleman
  - The URL at the top of the browser will begin with https://
  - The information you send using the HTTPS protocol is more secure than any encrypted military order sent during World War I, World War II, The Korean War, or The Vietnam War.

How RSA works

- First, we must be able to represent any message as a single number.
- For example:

A T **T** A **C** K **A** T **D** A **W** N

0120**20**01**03**11**01**20**04**01**23**14

Public and Private Keys

- Every receiver has a public key (*e*, *n*) and a private key (*d*, *n*).
- The transmitter encodes a (numerical) message *M* into an encrypted message *C* using the receiver's public key:

  M^e modulo $n \rightarrow C$
- The receiver decodes the encrypted message *C* to get the original message *M* using the private key (which no one else knows).

  C^d modulo $n \rightarrow M$

Example

- Milhouse's Public Key: (3, 33) (his "padlock") (e = 3, n = 33)
- Milhouse's Private Key: (7, 33) (his "combo") (d = 7, n = 33)
  - Usually these are really huge numbers with many hundreds of digits!
- Bart wants to send the message 4 to Milhouse
  - Bart encrypts the message using Milhouse's public key (e and n):

    4^3 modulo 33 $\rightarrow$ 31 ... Bart sends 31
- Milhouse receives the encoded message 31
  - Milhouse decrypts the message using his private key (d and n): 31^7 modulo 33 $\rightarrow$ 4

Simple Example: Computing e, n and d

• p and q are (big) random primes.	$p = 3$, $q = 11$
• $n = p \times q$	$n = 3 \times 11 = 33$
• r = $(p - 1)(q - 1)$	r = $2 \times 10 = 20$
• e is small and relatively prime to r	$e = 3$
• d, such that: $ed \mod \varphi = 1$	$3d \mod 20 = 1$ $d = 7$

Usually the primes are huge numbers--hundreds of digits long.

Cracking RSA

- Everyone knows (e, n). Only Bart knows d.
- If we know e and n, can we figure out d?
  - If so, we can read secret messages to Bart.
- We **can** determine d from e and n.
  - Factor n into p and q.
    $n = p \times q$
    $\varphi = (p - 1)(q - 1)$
    $ed = 1 \pmod{\varphi}$
  - We know e (which is public), so we can solve for d.

Cracking RSA (cont'd)

- How do you factor *n* ?
  - Try dividing n by 2, 3, 4, 5, ...
    (There are better factoring algorithms, but they're not significantly faster than this.)
- Suppose someone can factor my 5-digit *n* in 1 millisecond, by dividing by every number less than *n*.
- Do I give up on RSA?
  - No, use more digits!

RSA is safe (for now)

- Suppose someone can factor my 5-digit *n* in 1 ms, by dividing by every number less than *n*.
- At this rate, to factor a 10-digit number would take 2 minutes.
- At this rate, to factor a 15-digit number would take 4 months.
- At this rate, to factor a 20-digit number would take 30,000 years.
- At this rate, to factor a 25-digit number would take 3 billion years.
- We're safe with RSA! (or so we assume...)