

# Security

## Lecture 11

cs193i – Internet Technologies  
Summer 2004  
Stanford University

## Administrative Stuff

- Lab #4 due today
- HW #4 due August 11
- Final exam on Friday
- Wednesday will be review

## Final Exam

- Friday (8/13) from 7-10pm in Gates B03
- Closed book
- 2 8.5x11 cheat sheets
- Cumulative
  - Emphasis on material after midterm

## Global Functions

- `escape(string)`
- `unescape(string)`
- Safe Strings  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
1234567890  
@\* - \_ + . /
- Unsafe Strings => `%20`, `%5c`, etc...

# Security

## Computer Data

- File on your own hard drive (term paper)
- File on networked file system (Leland AFS)
- Data sent to another computer (credit card number to Amazon)

## Three Considerations: What do we want?

- Privacy of our data
- Integrity of our data
- Usability of our system/data

## Three Concepts

- Confidentiality of data
- Integrity of data
- Authentication of users

Tom Anderson's cs162 Notes

## What Functionality Is Needed?

- Authentication -- who user is
- Authorization -- who is allowed to do what
- Enforcement -- make sure people do what they are supposed to do

Tom Anderson's cs162 Notes

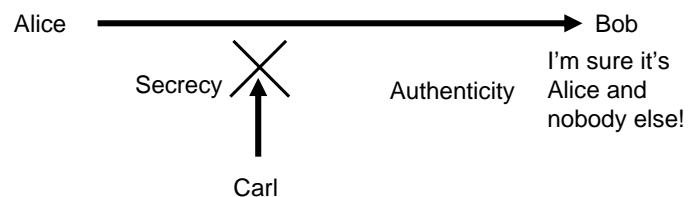
## Definitions

- Secrecy (aka Privacy, Confidentiality)
  - Diary Lock
- Authenticity
  - Hi it's Bob.
  - Prove it Dude...

## Definition Examples

- Secrecy
  - Alice sends message to Bob. Carl intercepts the message... but can't read
- Authenticity
  - Alice sends message to Bob. Bob can verify that Alice is the sender.

## The Big Picture



## Methods

- Cryptography
  - Converting messages to unreadable forms...  
Unconverting it back to the readable form
- Steganography
  - Hiding the existence of a message

## Steganography

## Null Cipher

Fishing freshwater bends and saltwater coasts  
rewards anyone feeling stressed. Resourceful  
anglers usually find masterful leapers fun and admit  
swordfish rank overwhelming anyday.

Send lawyers, guns, and money.

## Invisible Ink

- Write with lemon juice and a toothpick/ cotton  
swab. Let the paper dry.
- Heat the paper with an iron to reveal the hidden  
message.

## Cryptography

Greek: kryptos + graphein → hidden writing

## Encryption

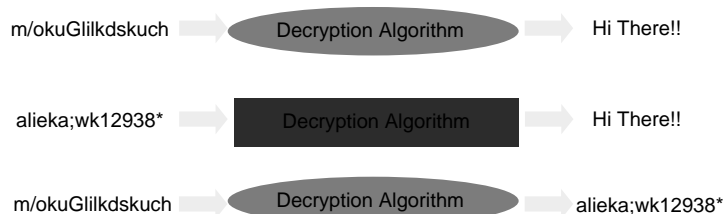
- Convert normal, readable data into obscured,  
unreadable data

Hi There!! → Encryption Algorithm → m/okuGlllkds kuch

Hi There!! → Encryption Algorithm → alieka;wk12938\*

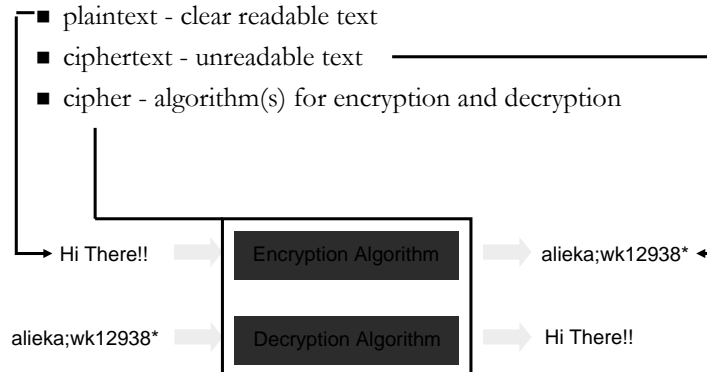
## Decryption

- Convert obscured, unreadable data into normal, readable data



## Terminology

- plaintext - clear readable text
- ciphertext - unreadable text
- cipher - algorithm(s) for encryption and decryption



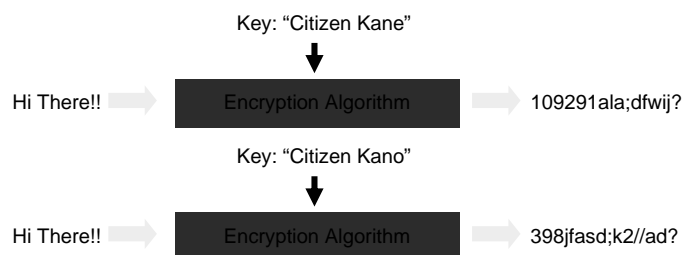
## Terminology

- Security through obscurity
  - Don't publish some details of your algorithm... assuming people won't figure it out
  - Like hiding the key under the doormat
- Once your flaw/algorithm is leaked, you're screwed

Optional Reading:  
<http://slashdot.org/features/980720/0819202.shtml>

## Terminology

- Key -- a secret piece of information that controls how the encryption algorithm works
- Different keys produce different encrypted results



## Classical Ciphers

- Monoalphabetic substitution
  - Caesar shift
- Polyalphabetic substitution
  - Jefferson

<http://www.murky.org/cryptography/classical.shtml>

## Caesar Shift

<i>PLAINTEXT</i>	a b c d e f g h i j k l m
<i>CIPHERTEXT</i>	D E F G H I J K L M N O P
<i>PLAINTEXT</i>	n o p q r s t u v w x y z
<i>CIPHERTEXT</i>	Q R S T U V W X Y Z A B C

Hello There → koorwkhuh

## Problem

- Monoalphabetic -- Same letter of plaintext always produces same letter of ciphertext
- Even though there are  $26!$  possible substitutions, monoalphabetic solutions are easy to break!
- Use frequency analysis of English language, plus some tricks...

## Breaking a Monoalphabetic Substitution

Yxdy pq yjc xzpvpyw ya icqdepzc ayjceq xq

yjcw qcc yjcuqcvrcq.

Xzexjxu Vpsdavs

## Breaking a Monoalphabetic Substitution

Yxdy pq yjc xzpvpyw ya icqdepzc ayjceq xq

yjcw qcc yjcuqcvrcq.

Xzexjxu Vpsdavs

Character Frequency: C10, Y8, Q7, X6, J5, P5, V4, D3  
A3, E3, Z3, S2, U 2, I1, R1, W2

## Breaking a Monoalphabetic Substitution

Yxdy pq yjc xzpvpyw ya icqdepzc ayjceq xq

yjcw qcc yjcuqcvrcq.

Xzexjxu Vpsdavs

Character Frequency: C10, Y8, Q7, X6, J5, P5, V4, D3  
A3, E3, Z3, S2, U 2, I1, R1, W2

Alphabet frequency: e t a o i n s r h l d c u m f p g w y b v k x j q z

## Breaking a Monoalphabetic Substitution

Yxdy pq yjc xzpvpyw ya icqdepzc ayjceq xq  
Tact is the ability to describe others as

yjcw qcc yjcuqcvrcq.  
they see themselves.

Xzexjxu Vpsdavs  
Abraham Lincoln

Character Frequency: C10, Y8, Q7, X6, J5, P5, V4, D3  
A3, E3, Z3, S2, U 2, I1, R1, W2

Alphabet frequency: e t a o i n s r h l d c u m f p g w y b v k x j q z

## Jefferson Wheel Cipher



## Computer Era

- Moore's law
- Keys breakable by brute force

## Modern Ciphers

- Bigger and bigger keys
- More and more complicated algorithms
- Based on hardcore applied mathematics... and the difficulty of factoring large (i.e. gargantuan) numbers

## Terminology

- Symmetric key cryptography
  - Caesar shift, ..., DES, AES
- Asymmetric key cryptography
  - Public/Private key schemes

## Symmetric Key Technology

- $p$  = plaintext
- $\text{crypt}()$  = encryption/decryption function
- $c$  = cipher text (unreadable)
- $k$  = key (secret; password)

## Symmetric Key Technology

- Alice wants to send a private/confidential message to Bob
- Alice computes  $c = \text{crypt}(p, k)$
- Sends  $c$  to Bob over unsecured wire
- Bob computes  $p = \text{crypt}(c, k)$

## Symmetric Key Application

- Password login
- Alice sends password to computer to prove identity (authenticity)
- Problem: Sniffing
- Solution: Challenge/response

## Shared Secret Key

- Shared secret is great... but how do we distribute it?

## Asymmetric Key Cryptography

- Instead of one key, have two
  - public key
  - private key

## Asymmetric Key Technology

- Use one key to encode/encrypt
- Use other key to decode/decrypt

## Asymmetric Key Technology

- Someone can know public key
- Computing private key from public key is very, very difficult (factoring huge number)

## Application: Secrecy

- Bob has Bob.pub, Bob.priv
- Alice has Alice.pub, Alice.priv
- Alice wants to send Bob a secret "I LUV U" note

## Application: Secrecy

- Alice finds Bob.pub from his website
- Alice computes  $c = \text{crypt}(p, \text{Bob.pub})$
- Sends  $c$  to Bob over unsecured wire
- Bob computes  $p = \text{crypt}(c, \text{Bob.priv})$

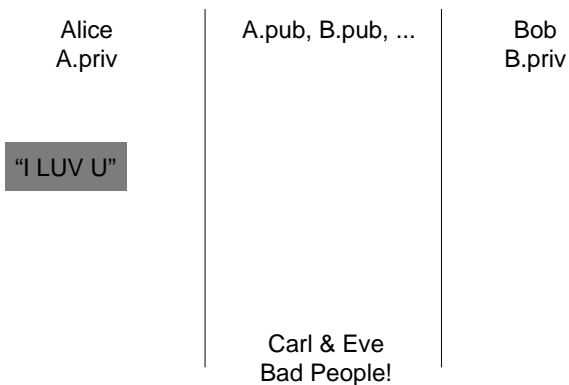
## Advantages

- Key distribution not a problem!
- Anyone can send a message to Bob
- Only Bob can decrypt!

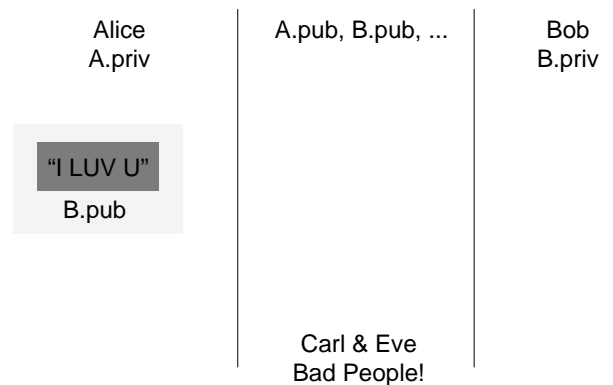
## Application: Authenticity

- Alice wants to tell Bob the message is really from her!
- Digital signature
- Alice computes  $c = \text{crypt}(p, \text{Alice.priv})$
- Alice sends  $c$  over unsecured wire
- Anyone can check that Alice is the sender... by computing  $p = \text{crypt}(c, \text{Alice.pub})$

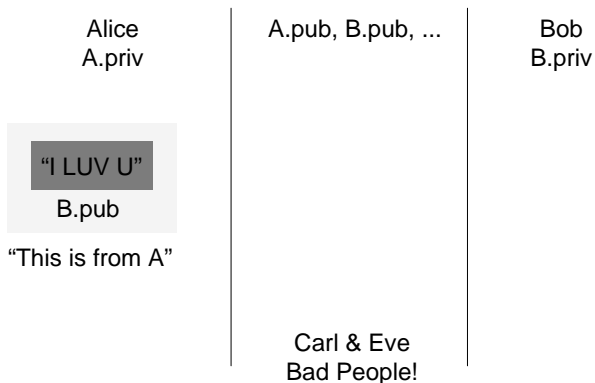
## Authenticity + Secrecy



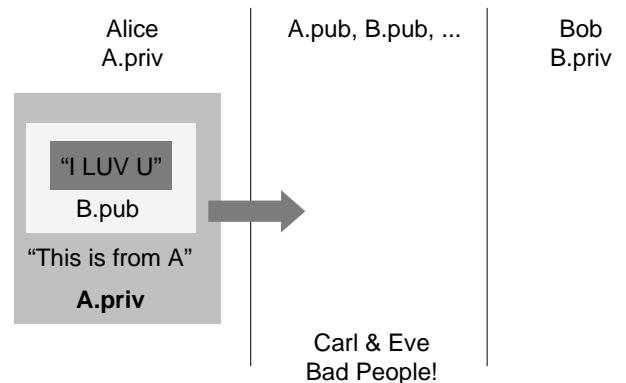
## Authenticity + Secrecy



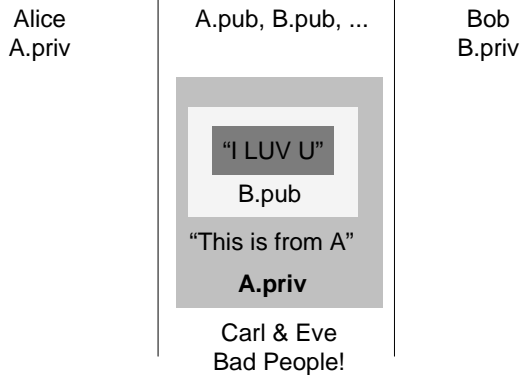
## Authenticity + Secrecy



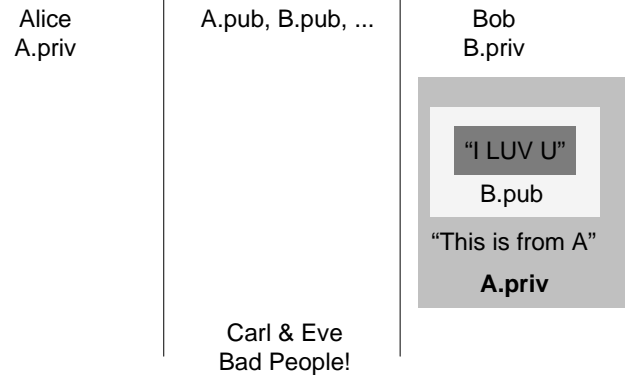
## Authenticity + Secrecy



## Authenticity + Secrecy



## Authenticity + Secrecy



## Hash Functions

- $h = \text{hash}(\text{input})$
- Every bit in input affects output
- Hash function not invertible

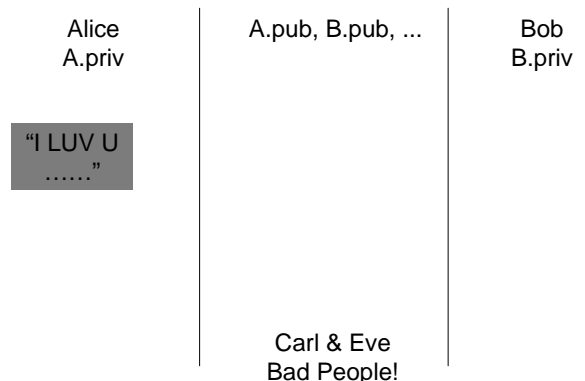
## Error Checking

- Alice wants to send a LONG message to Bob
- Alice computes  $h = \text{hash}(\$LONG\_MSG)$ ;
- Sends data to Bob, includes relatively short  $h$  at the end of message
- Bob recomputes hash.
- If match, great! Data's correct!
- If not match, either hash or data was corrupted. Resend.

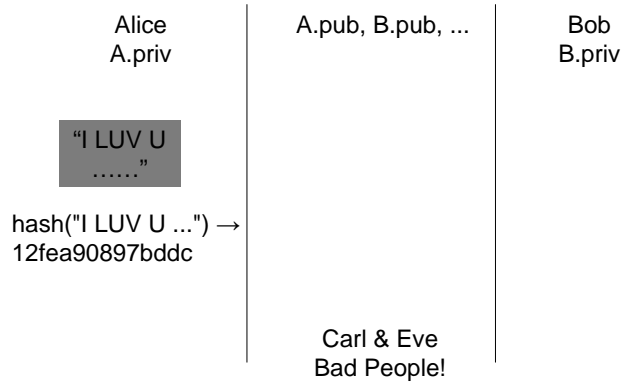
## Digital Signatures

- Bob wants to send  $\$data$  to Alice, with assurances of his identity (authenticity)
  - $h = \text{hash}(\$data)$
  - $\text{Signature} = \text{crypt}(h, \text{Bob.priv})$
- Sends these to Alice
- Alice confirms Bob's identity by
  - $h = \text{crypt}(\text{signature}, \text{Bob.pub})$
  - $h = \text{hash}(\$data)$
  - Compares!

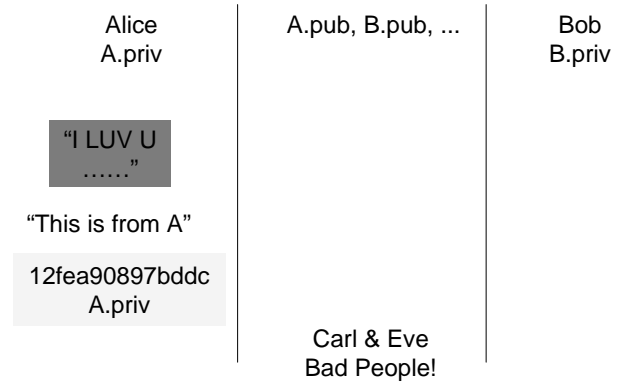
## Authenticity + Secrecy



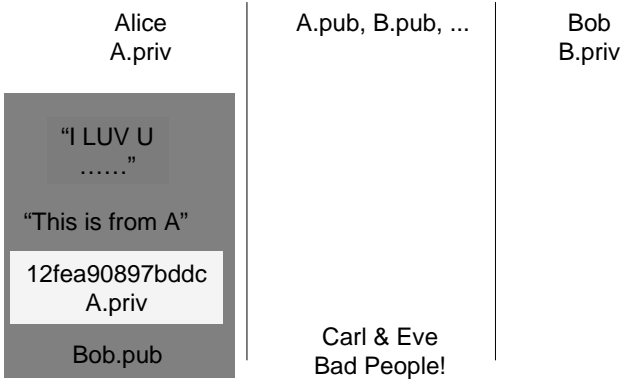
### Authenticity + Secrecy



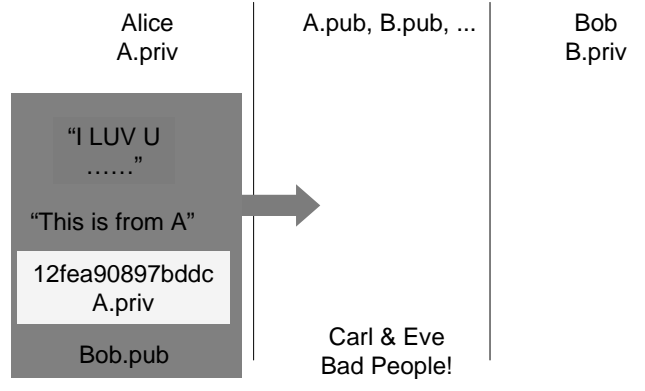
### Authenticity + Secrecy



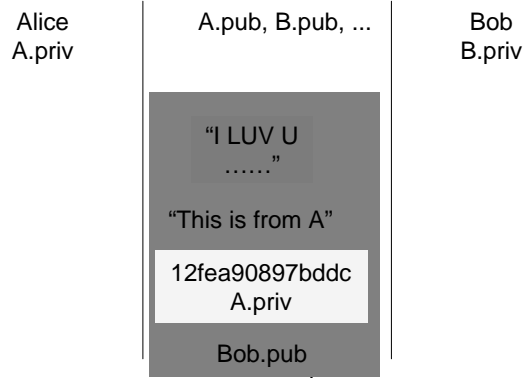
### Authenticity + Secrecy



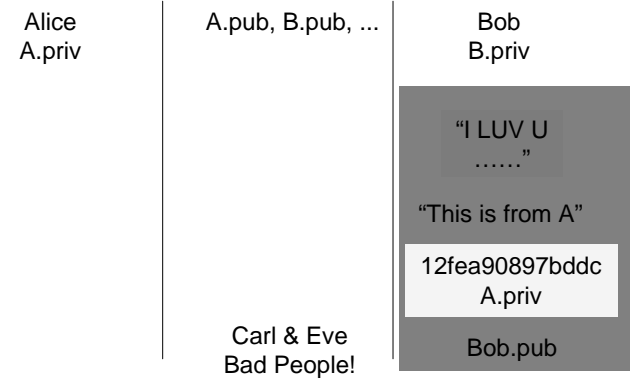
### Authenticity + Secrecy



### Authenticity + Secrecy



### Authenticity + Secrecy



## Authenticity + Secrecy

Alice  
A.priv

A.pub, B.pub, ...

Bob  
B.priv

"I LUV U  
....."

"This is from A"

12fea90897bddd  
A.priv

Carl & Eve  
Bad People!

## Authenticity + Secrecy

Alice  
A.priv

A.pub, B.pub, ...

Bob  
B.priv

"I LUV U  
....."

"This is from A"

12fea90897bddd  
A.priv

Carl & Eve  
Bad People!

## Authenticity + Secrecy

Alice  
A.priv

A.pub, B.pub, ...

Bob  
B.priv

"I LUV U  
....."

"This is from A"

12fea90897bddd  
==  
hash("I LUV U ...") →  
12fea90897bddd

Carl & Eve  
Bad People!

## Certificates

- Certificate Authority: publishes that a particular identity goes with a particular public key
- Alice gets certificate (identity  $\leftrightarrow$  public key), signed by CA
- So if you trust CA, then you can trust the public key

## SSL

- Alice connects to Bob's server
- Bob's server returns certificate (signed by VeriSign), plus something encrypted w/ Bob.priv
- Alice can verify certificate is valid
- Uses public key to decrypt token
- Bob authenticated
- Alice makes one time session key k
- Encrypts w/ Bob's public key, sends to Bob
- Now, can use symmetric key cryptography

## Symmetric vs. Asymmetric

- Symmetric faster but relies on shared secret
- Asymmetric slower but "solves" distribution-of-keys problem

## Security History

- If you write it, they will come... to attack it. :o)
- Be aware of most common attacks...
- Learn the basic tricks to writing safer code.

## CERT



## Terminology

- Vulnerability -- some buggy code that can allow bad guys to compromise your machine, or do other bad guy things
- Exploit -- some code or method to take advantage of the vulnerability

## Attack: Social Engineering

- Tricking a naïve person into revealing sensitive data (i.e. his/her password)
  - Hi this is your bank. We need your PIN to fix your account ASAP!
  - Hi this is Amazon. Your order #2333 didn't go through because your credit card was rejected. Tell us another credit card's info, and your order will be good.
  - Dumpster-diving for username & passwords on paper

## Bottom Line

- People are the weakest link
- Educate people about computer/Internet Security

## Attack: Traffic Sniffing

- Looking at packets on the wire, reading off passwords, etc...
- Problem for authentication mechanisms with cleartext passwords

## Traffic Sniffing

- (Somehow) compromise a machine. This is the hard part.
- Set ethernet "promiscuous" mode
- Install a root kit
  - hides hacker activity
  - key logger
  - packet sniffer
  - recompiled versions of programs (passwd, ls)

## Attack: Spoofer

- One person (hacker) successfully masquerades as another (normal user)

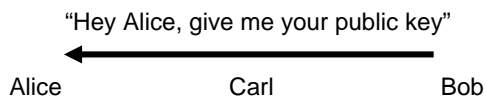
## IP Spoofing

- Rewrite headers in IP packets to say they are from someone else
- Launch some other attack. Spoofed IPs prevent good guys from finding you.

## Man In the Middle Attack (Spoofing)

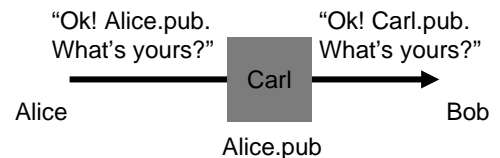


## Man In the Middle Attack (Spoofing)

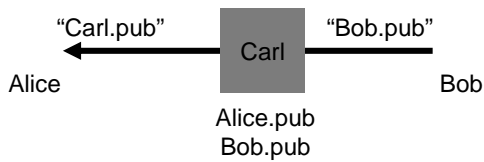


SSL-Like Example

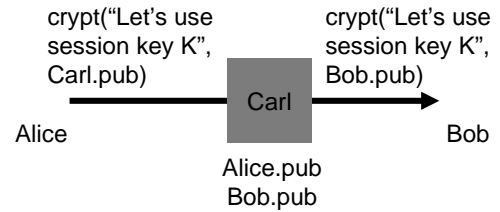
## Man In the Middle Attack (Spoofing)



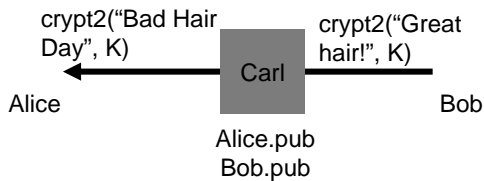
### Man In the Middle Attack (Spoofing)



### Man In the Middle Attack (Spoofing)



### Man In the Middle Attack (Spoofing)



### Verify Authenticity

- Through digital signatures
- And Certificate Authorities

### Attack: DNS Poisoning

- DNS server accepts and uses incorrect info from host with no authority
- Future requests given the incorrect info from that server's cache

### Attack: Password Guessing

- How long is YOUR password?
- Ways to break
  - Dictionary attack (words, names, dates)
  - Brute force
- Solutions
  - Freeze/Turn off account if too many incorrect logins?
  - Wait 2 seconds before logging in/displaying error.

## Passwords

- What if your website froze accounts if too many incorrect logins?
- Hacker can still attack your sites users!
- By purposefully guessing login/passwords incorrectly, so that your system locks all accounts!

## Solutions

- Longer passwords
- Other forms of authentication
  - Biometric
  - Physical key/card based

## Attack: Denial of Service

- Make the service unavailable
- Flood of incoming traffic  
(SYN flood, Malformed Packets)
- Use robot to launch DOS on server. Hard to trace identity of attacker.
- Distributed DOS (DDOS)
  - Take over many machines, launch attack simultaneously from many locations

## Smurf DOS

- Bad guy sends ping packets to IP broadcast addresses, source IP spoofed of course
- All hosts on that network perform an ICMP echo reply (reply to the ping)
- Potentially hundreds of replies per packet, can bring network down

## External Executables

- Don't trust other people's code
- If Carl can run code on Alice's computer... then Carl can take it over
- Internet Explorer, Safari Vulnerabilities
- "Reflections on Trusting Trust", Ken Thompson  
(<http://www.acm.org/classics/sep95/>)

## Attack: Trojan Horse

- Greek allusion (also, remember Monty Python?)
- Innocent looking program, does something malicious
  - OpenSexyPics.exe, Readme.txt.exe
- "recent Trojans include programs disguised as fixes to common computer viruses and those promising free pornographic images."

## Attack: Buffer Overflow

- Bad guy sends a huge, over-sized request to a naïvely implemented (aka buggy) program, overflowing the input buffer
- May overwrite data in memory (and/or) program code
- May overwrite the return address on the stack of a program in C, so that the procedure call returns somewhere else

## How To Avoid Buffer Overflow

- Write code carefully
- Limit input size; read in small chunks as opposed to reading in whole input
- Use better languages (read: Java)

## Attack: Worm

- Self replicating/Spreading computer program

## Example

- Morris Worm -- buffer overflow attack on UNIX finger and other programs...
- Robert Tappan Morris, Jr. (CMU student) launched it on Nov 2, 1988 from an MIT computer
- Intended to just spread, but a `_bug_` in his code infected computers multiple times, so that computers FROZE after a while
- Infected 6000 UNIX workstations
- CERT created in response to Morris
- Morris now a MIT faculty member

## Worms and their Payloads

- Infect computer; send emails to other people... to spread the worm
- Infect computer; install a backdoor program to let bad guy log in... to send mass spam, send more worms, etc

## Blaster Worm

- Exploited a buffer overflow in Windows's RPC service
- Programmed to SYN flood windowsupdate.com on August 15 to prevent patches

## Attack: Computer Virus

- Attaches itself to a host, another computer program
- Tries to infect other executable files it finds
- When run, it damages resources, files, etc...

## Timeline of Viruses and Worms

- May 2004 -- Sasser
  - Delta Airlines canceled many flights, computers down from Sasser
- January 2004 -- MyDoom
  - Attacked MS & SCO Group websites with DDOS

[http://en.wikipedia.org/wiki/Timeline\\_of\\_notable\\_computer\\_viruses\\_and\\_worms](http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms)

## Timeline of Viruses and Worms

- 2003 August: Sobig and Blaster
- 2001: Code Red attacks IIS
- 2000: VBS/"I Love You" Worm
- 1999: Melissa Worm

[http://en.wikipedia.org/wiki/Timeline\\_of\\_notable\\_computer\\_viruses\\_and\\_worms](http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms)