Network Security Protocols

Radia Perlman
(radia.perlman@sun.com)

Logistics

- One lecture/week
  - interactive
  - class participation counts
- problem sets
- quizzes
- project: of your choice, suggestions will be given in class. Tip: do it early, when I have fewer to look at!

Logistics

- Book: “Network Security: Private Communication in a Public World”, Kaufman, Perlman, Speciner; Prentice Hall, ISBN 0-13-046019-2
- Prerequisites: Nothing specific, but “mathematical sophistication”, some algorithms, some networking

What This Course is About

- Focus on network protocols
- Cryptography, especially practical issues, and intuition.
- How to design a secure protocol
- Recognizing snake oil and common flaws
- Conceptual overview of standards and deployed systems
- Possible research topics

Outline

- Introduction
- Cryptography
- Authentication
- Standards and Deployed Systems
  - PKI, Secure Email, Kerberos, SSL, IPsec, Web (HTTP, cookies)

Section: Introduction

- What is secure communication?
- What can the intruders do?
- Operating system issues: viruses, active content
- Legal Issues: patents, export controls

Intruders: What Can They Do?

- Eavesdrop
- Send Messages
- Impersonate an address and lie in wait
- Replay recorded messages
- Modify messages in transit
- Write malicious code and trick people into running it

Some Examples to Motivate the Problems

- Sharing files between users
  - File store must authenticate users
  - File store must know who is authorized to read and/or update the files
  - Information must be protected from disclosure and modification on the wire
  - Users must know it's the genuine file store (so as not to give away secrets or read bad data)

Examples cont’d

- Electronic Mail
  - Send private messages
  - Know who sent a message (and that it hasn’t been modified)
  - Non-repudiation - ability to forward in a way that the new recipient can know the original sender
  - Anonymity

Examples cont'd

- Electronic Commerce
  - Pay for things without giving away my credit card number to an eavesdropper or phony merchant
  - Buy anonymously
  - Merchant wants to be able to prove I placed the order

Sometimes goals conflict

- nonrepudiation vs plausible deniability
- privacy vs company (or govt) wants to be able to see what you're doing
- avoiding false positives vs false negatives
- safety vs functionality
- losing data vs disclosure (copies of keys)
- denial of service vs preventing intrusion

Other threats

- Denial of service: Used to be ignored… "it would be illogical"
- Traffic analysis

Quick Overview

- We’re going to need cryptography
- It allows you to prove you know a secret without divulging it
- If Alice and Bob share a secret:
  - Bob can know he’s talking to Alice
  - Bob can encrypt a message for Alice
  - Bob can know nobody has tampered with a message from Alice

Overview cont’d

- Public key crypto allows Alice to prove to Bob she knows her secret without Bob knowing her secret
- Securely distributing keys (secret key systems like Kerberos, vs PKIs)
- Session protocols (e.g., SSH, SSL, IPSEC)
- Email (e.g., S/MIME, PGP)

Overview cont'd

- There are lots of variants of schemes, because multiple independent organizations simultaneously worked on the problems
- There are a few basic crypto tricks that are the basis of all these protocols
- We cover the toolkit of crypto tricks
- Then we explain the specifics of the alphabet-soup of protocols

Active Content: Threat or Menace?

- If you run a program I wrote, it can do things with your rights behind your back
  - Read your private files and send them to me
  - Delete or mangle files you have rights to access
  - Send email composed by me but sent (proveably!) by you
  - Authorize me to do things later (if you can add me to ACLs)

Active Content: What were they thinking!? (or… why can’t I only run software from trustworthy sources?)

- Bandwidth and Storage Efficiency
  - A program to generate a graphic could be much smaller than the bitmap (particularly for animations)
- Extensibility
  - Application designer can add capabilities not envisioned by the platform designer
- Push computation out to the client
  - Where CPU cycles tend to be cheaper

...and even if the source is Trustworthy

- The program must be bug-free or it might introduce security problems
- How do you know this is genuine? (e.g. when downloading from the web)
- You’re not just trusting that source, but all sources they’ve ever trusted...

Digital Pests

- Trojan horse: malicious code hidden inside an otherwise useful program (waiting for someone with interesting privileges to run it)
- Virus: malicious code hidden inside a program that when run “reproduces” by installing copies of itself inside programs the person running it has permission to modify

Digital Pests

- Worm: A program that replicates over a network by finding nodes willing to accept copies and run them
- Trapdoor: An undocumented entry point intentionally written into a program
- Logic Bomb: malicious code triggered on a future event
- Letter Bomb: malicious code executed upon opening an email message

Spreading Pests

- Booting from an infected floppy disk
- Loading and executing infected software from the Internet or other untrusted source
- Extracting and running untrustworthy code from an email message
- Displaying a Postscript or Word file
- Email with “autolaunch” capability
- Bugs (e.g., bounds checking)

What Protection Is There?

- Decent operating systems
- Interpreted languages in “sandboxes”
- Digitally signed content
- Content scanners (at WS or Firewall)
- Connectivity restrictions (through Firewall)
- Educating users
- Genetic diversity

Legal Issues Past (hopefully)

- All Public Key cryptographic algorithms were patented until September 1997.
- RSA patent expired September 20, 2000
- Patents in general a real problem.
- Export controls
- Usage controls

Section: Cryptography

- Three kinds of cryptographic algorithms
  - Secret Key Cryptography (DES, IDEA, RCx, AES)
  - Public Key Cryptography (RSA, Diffie-Hellman, DSS)
  - Message Digests (MD4, MD5, SHA-1)

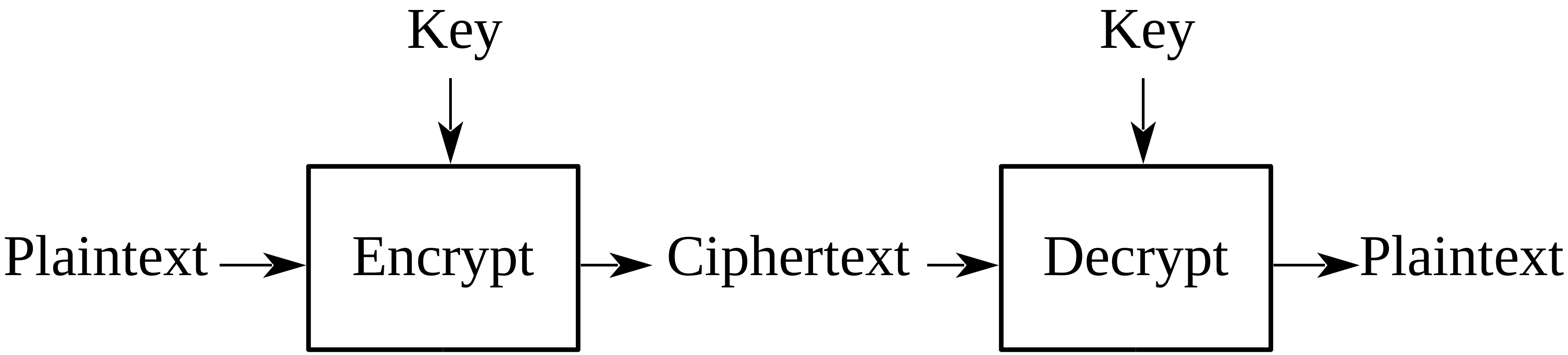
Secret Key Cryptography

- Originally a way to keep secret data private
  - Encode a message using a secret “key”
  - A long and colorful history
- Today, it has many uses
  - Privacy
  - Authentication
  - Data Integrity

What is Encryption?

- You and I agree on a secret way to transform data
- Later, we use that transform on data we want to pass over an unsafe communications channel
- Instead of coming up with new transforms, design a common algorithm customized with a "key"

Secret Key Encryption for Privacy



How Secure is Encryption?

- An attacker who knows the algorithm we're using could try all possible keys
- Security of cryptography depends on the limited computational power of the attacker
- A fairly small key (e.g. 64 bits) represents a formidable challenge to the attacker
- Algorithms can also have weaknesses, independent of key size

How do we know how good an algorithm is?

- A problem of mathematics: it is very hard to prove a problem is hard
- It's never impossible to break a cryptographic algorithm - we want it to be as hard as trying all keys
- Fundamental Tenet of Cryptography: *If lots of smart people have failed to solve a problem then it probably won't be solved* (soon)

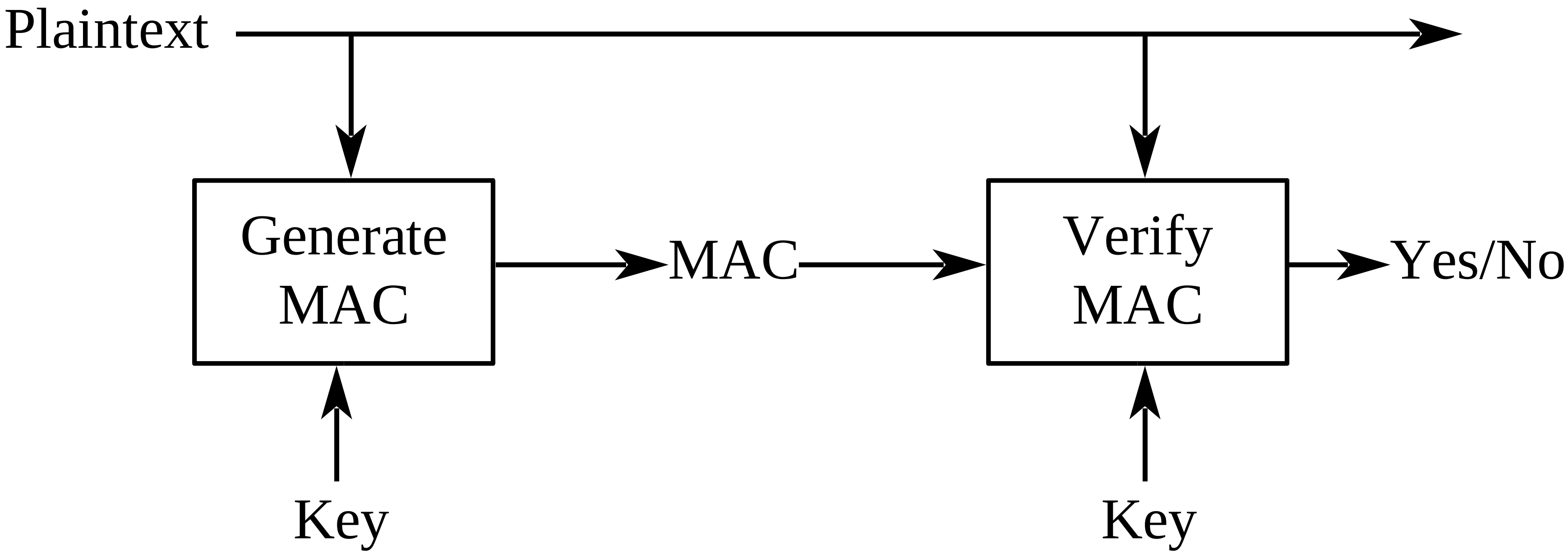
To Publish or Not to Publish

- If the good guys break your algorithm, you’ll hear about it
- If you publish your algorithm, the good guys provide free consulting by trying to crack it
- The bad guys will learn your algorithm anyway
- Today, most commercial algorithms are published; most military algorithms are not

Uses of Cryptography

- Transmitting secret data over an insecure channel
- Storing secret data on an insecure medium
- Message integrity checksum/authentication code (MIC/MAC)
- Authentication: “challenge” the other party to encrypt or decrypt a random number

Secret Key Integrity Protection



Challenge / Response Authentication

Alice (knows K)

Bob (knows K)

I'm Alice →

Pick Random R
Encrypt R using K
(getting C)

← If you're Alice, decrypt C

R →

Secret Key Algorithms

- DES (Data Encryption Standard)
  - 56 bit key (+ 8 parity bits) controversial!
  - Input and output are 64 bit blocks
  - slow in software, based on (sometime gratuitous) bit diddling
- IDEA (International Data Encryption Algorithm)
  - 128 bit key
  - Input and output are 64 bit blocks
  - designed to be efficient in software

Secret Key Algorithms

- Triple DES
  - Apply DES three times (EDE) using K1, K2, K3 where K1 may equal K3
  - Input and output 64 bit blocks
  - Key is 112 or 168 bits
- Advanced Encryption Standard (AES)
  - New NIST standard to replace DES.
  - Public Design and Selection Process. Rijndael.
  - Key Sizes 128,192,256. Block size 128.

Secret Key Algorithms

- RC2 (Rivest’s Cipher #2)
  - Variable key size
  - Input and output are 64 bit blocks
- RC4 (Rivest’s Cipher #4)
  - Variable key size
  - Extremely efficient
  - Stream cipher - one time use keys
- Many other secret key algorithms exist
- It is hard to invent secure ones!
- No good reason to invent new ones

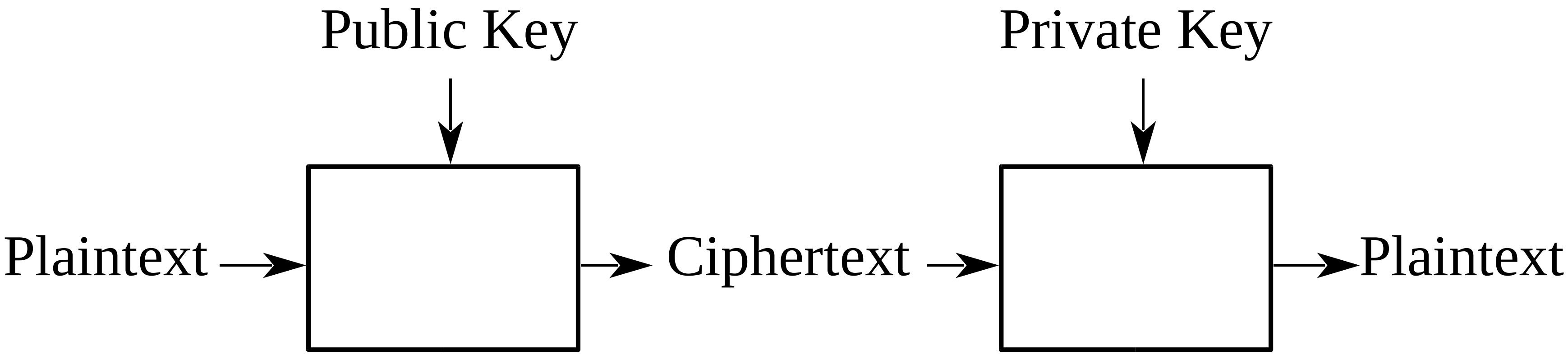
XOR (Exclusive-OR)

- Bitwise operation with two inputs where the output bit is 1 if exactly one of the two input bits is one
- (B XOR A) XOR A) = B
- If A is a “one time pad”, very efficient and secure
- Common encryption schemes (e.g. RC4) calculate a pseudo-random stream from a key

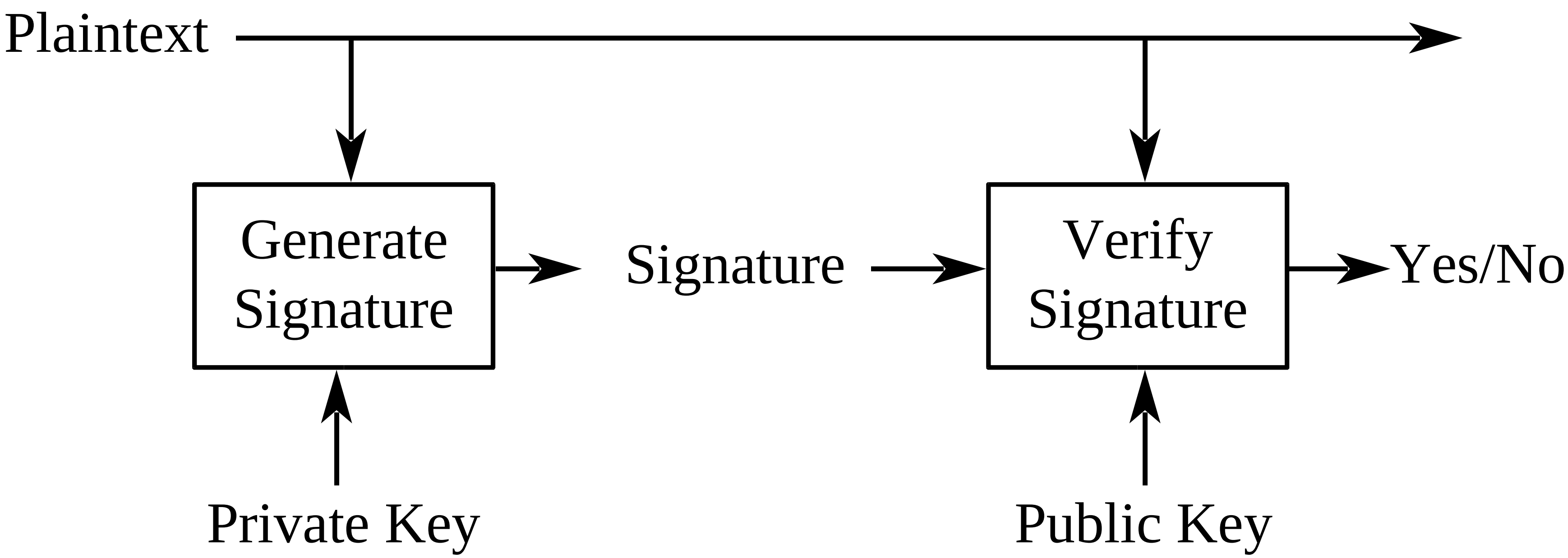
Public Key Cryptography

- Two keys per user: a private key and a public key. The keys reverse each other's effects.
- Encrypt a message for Alice using her public key
- Decryption requires her private key
- Generating Digital Signatures requires the private key
- Verifying them requires the public key

Public Key Encryption for Privacy



Public Key Integrity Protection



Public Key Authentication

Alice (knows A's private key)

Bob (knows A's public key)

I'm Alice

Pick Random R
Encrypt R using A's public key (getting C)

If you're Alice, decrypt C

Decrypt C

R

Message Digest Functions

- Also known as cryptographic hashes
- Non-reversible function
- Takes an arbitrary size message and mangles it into a fixed size digest
- It should be impossible to find two messages with the same MD, or come up with a message with a given MD
- Useful as a shorthand for a longer thing

Message Digest Functions

Message Digest Functions

- MD2, MD4, and MD5 used to be most popular. SHA-1 taking over
- All produce 128 bit digests
- MD4 and MD2 were recently “broken” and MD5 has significant weaknesses
- SHA-1 was proposed by the U.S. government. It produces a 160 bit digest
- Message digests are not difficult to design, but most are not secure

Combining Cryptographic Functions for Performance

- Public key cryptography is slow compared to hashes and secret key cryptography
- Public key cryptography is more convenient & secure in setting up keys
- Algorithms can be combined to get the advantages of both

Hybrid Encryption

Instead of:

Message

Encrypted with Alice's Public Key

Use:

Randomly Chosen K

Encrypted with Alice's Public Key

+

Message

Encrypted with Secret Key K

Hybrid Signatures

Instead of:

Message

Signed with Bob’s Private Key

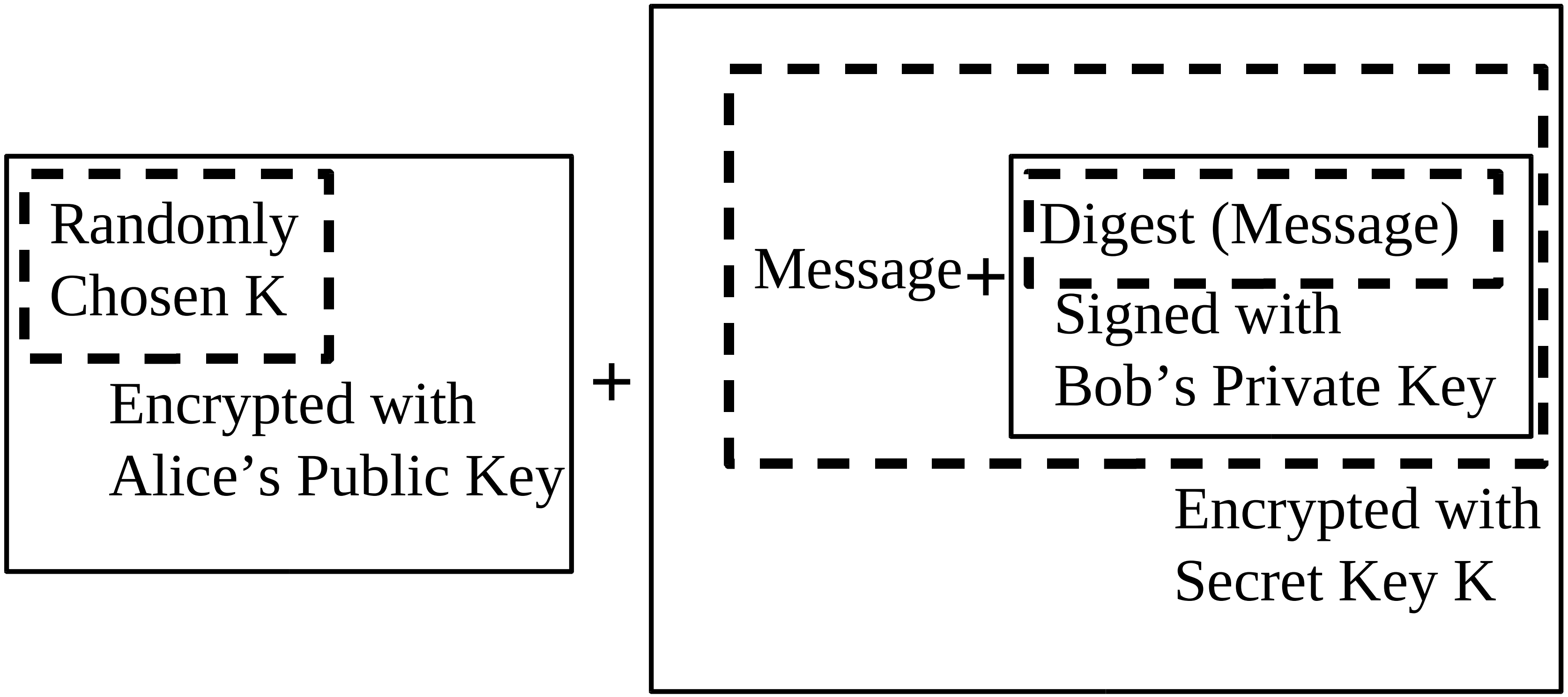
Use:

Message +

Digest (Message)

Signed with Bob’s Private Key

Signed and Encrypted Message



Section: Authentication

- Non-cryptographic authentication
- Special problems with people
- Cryptographic authentication
- Key distribution: KDCs and CAs

Non-Cryptographic Network Authentication

- Password based
  - Transmit a shared secret to prove you know it (e.g. cellular phones)
- Address based
  - If your address on a network is fixed and the network makes address impersonation difficult, recipient can authenticate you based on source address
  - UNIX .rhosts and /etc/hosts.equiv files

Authentication of People

- What you know
- What you have
- What you are

What You Know...

- Mostly this means passwords
  - Subject to eavesdropping
  - Subject to theft of password database
  - Subject to on-line guessing
  - Subject to off-line guessing
- How can you force people to choose good passwords?

What You Have...

- Passive Devices (physical key, mag stripe card)
- Smart Cards
  - PIN activated memory
  - Display on card (no reader necessary)
  - Display and keyboard on card
  - Special reader w/electrical connection
    - Crypto on the card
    - Secret never leaves the card
    - PCMCIA, SmartDisk, ISO format

What You Are...

- Biometric Devices
  - Retinal/Iris Scanners
  - Signature Verifiers
  - Fingerprint Readers
- Limitations
  - Expensive
  - Users hate them
  - Not useful for network authentication (though possibly as an adjunct)

Biometrics

- Lots of false positives and false negatives
- Easier to verify a claimed identity than search for a match

On-Line Password Guessing

- If guessing must be on-line, password need only be mildly unguessable
- ATM machine eats card after 3rd wrong PIN
- Military: they arrest you after one wrong attempt
- Computers
  - Lock out account after ‘n’ tries
  - Process attempts slowly
  - Audit failed attempts and alert an administrator

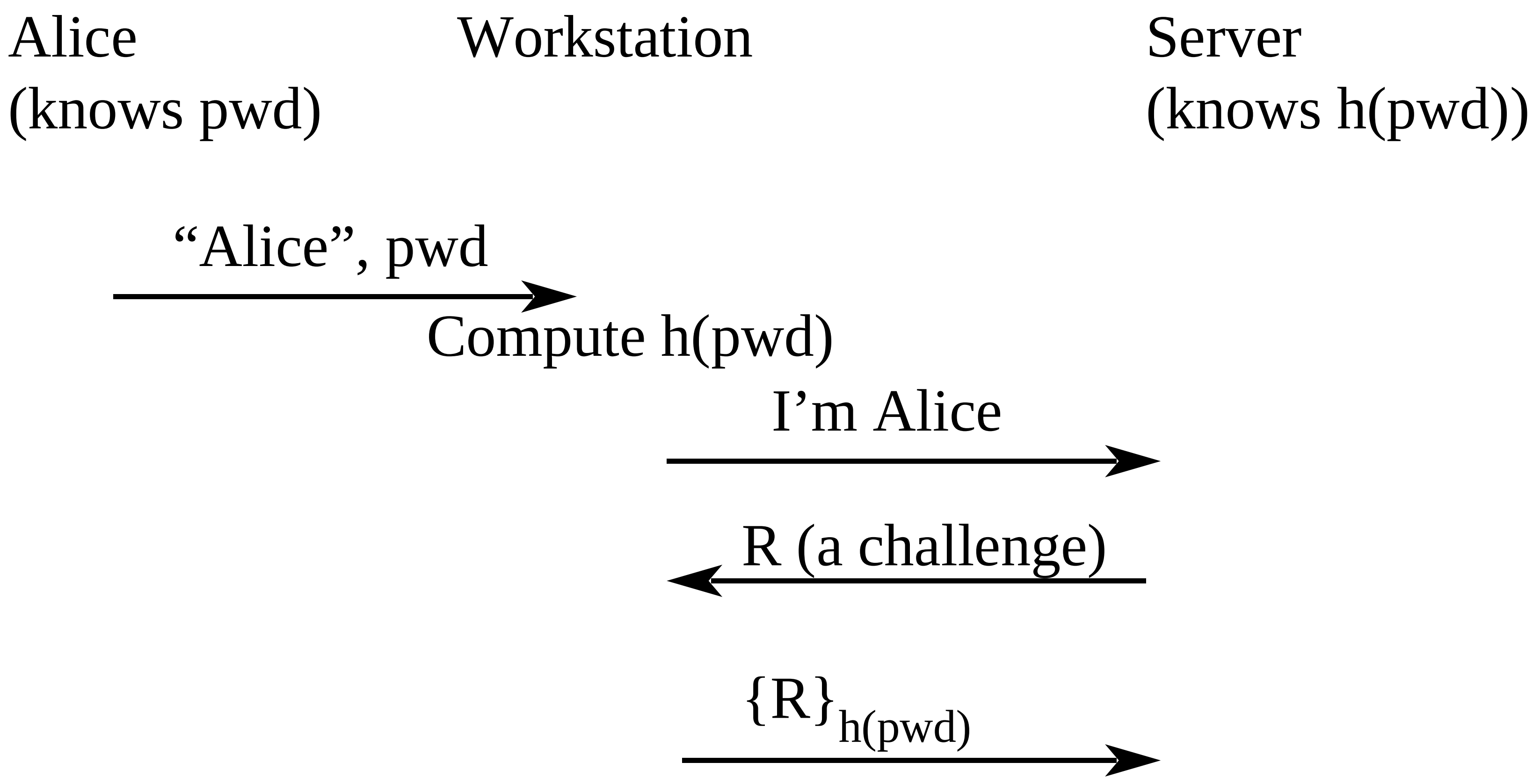
Off-Line Password Guessing

- If a guess can be verified with a local calculation, passwords must survive a very large number of guesses
- Unix password database was world readable and held one-way hashes of passwords
- Once you read the database, you can take it back to all the Crays in your basement and have them guess passwords

Passwords as Secret Keys

- A password can be converted to a secret key and used in a cryptographic exchange
- An eavesdropper can often learn sufficient information to do an off-line attack
- Most people will not pick passwords good enough to withstand such an attack

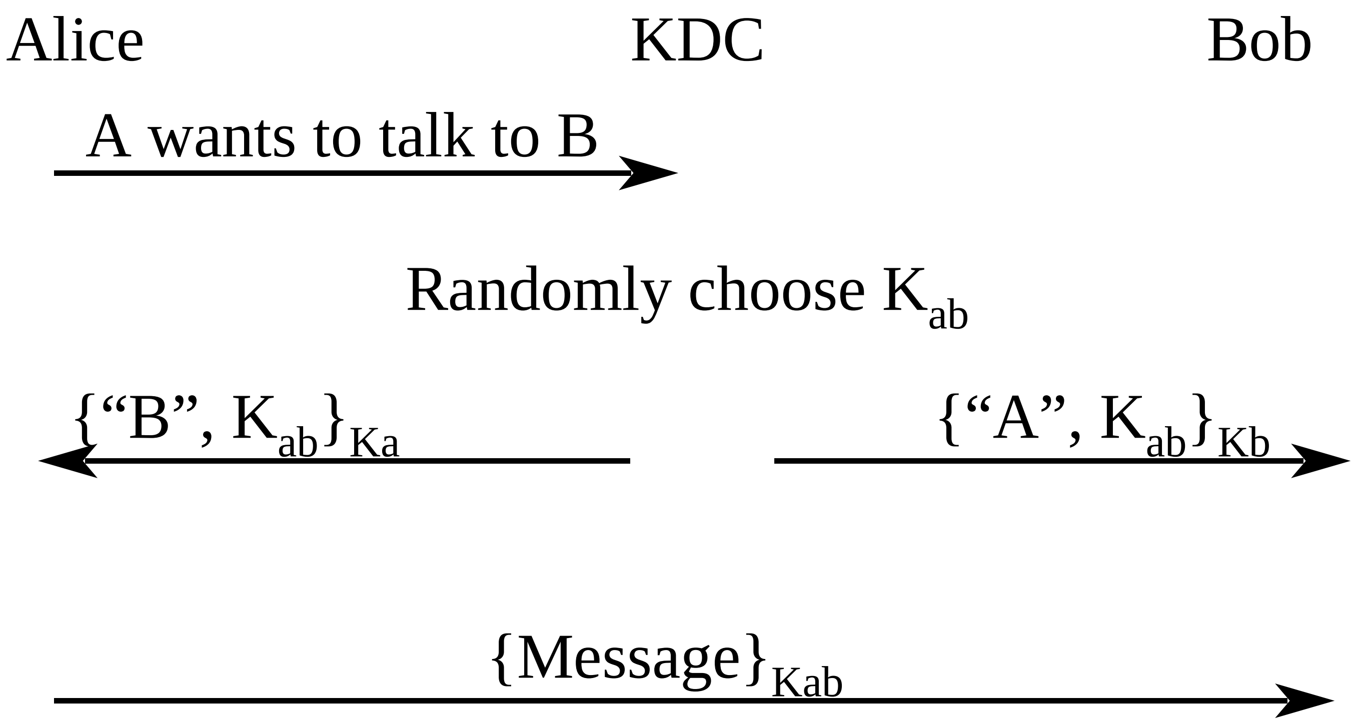
Sample Protocol



Key Distribution - Secret Keys

- What if there are millions of users and thousands of servers?
- Could configure n^2 keys
- Better is to use a Key Distribution Center
  - Everyone has one key
  - The KDC knows them all
  - The KDC assigns a key to any pair who need to talk

Key Distribution - Secret Keys



Key Distribution - Public Keys

- Certification Authority (CA) signs "Certificates"
- Certificate = a signed message saying "I, the CA, vouch that 489024729 is Radia's public key"
- If everyone has a certificate, a private key, and the CA's public key, they can authenticate

KDC vs CA Tradeoffs

- Stealing the KDC database allows impersonation of all users and decryption of all previously recorded conversations
- Stealing the CA Private keys allows forging of certificates and hence impersonation of all users, but not decryption of recordings
- Recovering from a CA compromise is easier because user keys need not change

KDC vs CA Tradeoffs

- KDC must be on-line and have good performance at all times
- CA need only be used to create certificates for new users
  - It can be powered down and locked up, avoiding network based attacks
- CA's work better interrealm, because you don't need connectivity to remote CA's

KDC vs CA Tradeoffs

- Public Key cryptography is slower and (used to) require expensive licenses
- The “revocation problem” levels the playing field somewhat

Authorization with ACLs

- ACL lists who has access
- Easier with groups, and wildcarded names (*@Sun.com)
- Groups might be members of groups
- Might be slow to verify if someone is a member of a deep group

Authorization with Capabilities

- Suggested for OS world, never caught on
- Idea is your certificate says what you're allowed to do, not who you are
- "Capabilities are the access control mechanism for the future and always will be"

Authorization Today

- Server keeps membership list of groups. ACL cannot list a group not stored on that server
- KDC keeps track of groups for each user, stores in ticket (DCE, Win2K)

Electronic Mail Security: What might you want?

- Privacy
- Authentication
- Integrity
- Non-repudiation

Complications with email

- finding someone's key
- distribution lists
- store-and-forward
- text-only email infrastructure

Non-Repudiation vs Plausible Deniability

- Non-Repudiation: ability to prove to 3rd party the message came from sender
- Plausible deniability protects sender. The receiver knows who sent it but can’t prove it to a 3rd party
- Non-Repudiation easy with public key
- Plausible deniability easy with secret key
- Can do vice versa, but difficult

Firewalls

- Paranoid (i.e. sensible) conn. to Internet
- Sits between your net and Internet and protects you, somehow
- Packet filter (limited access to your net to outsiders)
- Application gateway (also outsiders)
- Encrypted tunnel: full access to “insiders”

Firewalls

- real art is knowing specifically what they should do (what rules it should have)
- depends on applications
- easiest: just let everything through
- alternative: run everything over http

Intrusion Detection Systems

- Look at traffic
- Let you know if anything out of the ordinary is happening
- Real art: recognizing bad stuff and not setting off false alarms

Future topics

- secret key crypto and tricks
- public key algorithms (Diffie-Hellman, RSA, some number theory)
- PKI issues
- authorization
- strong password protocols
- details of Kerberos, IPsec, SSL