

Quick Intro to Computer Security

- What is computer security?
 - Securing communication
 - ▼ Cryptographic tools
 - Access control
 - ▼ User authentication
 - Computer security and usability
-
- Thanks to Mike Reiter for the slides

What Is Computer Security?

- Protecting computers against misuse and interference

- Broadly comprised of three types of properties
 - ▼ Confidentiality: information is protected from unintended disclosure
 - ▼ Integrity: system and data are maintained in a correct and consistent condition
 - ▼ Availability: systems and data are usable when needed
 - ▼ Also includes timeliness

- These concepts overlap
- These concepts are (perhaps) not all-inclusive
 - ▼ Spam?
 - ▼ “Non-business related” surfing?

Hacking

■ To be annoying

- ▼ Newsday technology writer & hacker critic found ...
 - ▼ Email box jammed with thousands of messages
 - ▼ Phone reprogrammed to an out of state number where caller's heard an obscenity loaded recorded message

[Time Magazine, December 12, 1994]

■ To be seriously annoying

- ▼ An international group attacked major companies: MCI WorldCom, Sprint, AT&T, and Equifax credit reporters
 - ▼ had phone numbers of celebrities (e.g. Madonna)
 - ▼ had access to FBI's national crime database
 - ▼ gained information on phones tapped by FBI & DEA
 - ▼ created phone numbers of their own

[PBS website report on Phonemasters (1994 – 1995)]

Hacking

■ For profit

- ▼ Hacker accessed Citibank computers and transferred \$10M to his account
- ▼ Once caught, he admitted using passwords and codes stolen from Citibank customers to make other transfers to his accounts

[PBS web site report on Vladimir Levin, 1994]

■ For extortion

- ▼ Hacker convicted of breaking into a business' computer system, stealing confidential information and threatening disclosure if \$200,000 not paid

[U.S. Dept. of Justice Press Release, July 1 2003]

Hacking

■ As a business in information

- ▼ Internet sites traffic in tens of thousands of credit-card numbers weekly
- ▼ Financial loses of over \$1B/year
- ▼ Cards prices at \$.40 to \$5.00/card – bulk rates for hundreds or thousands

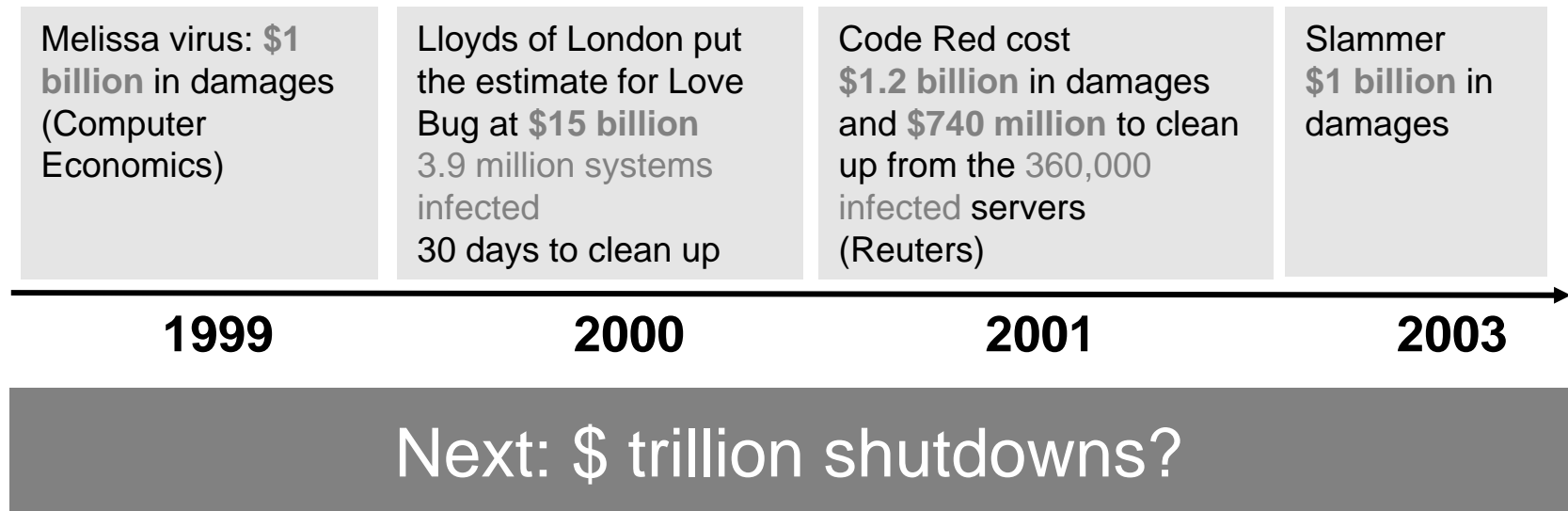
[New York Times News Service, May 13, 2002]

■ As a business for renting infrastructure

- ▼ Rent a pirated computer for \$100/hour
- ▼ Average rate in underground markets
- ▼ Used for sending SPAM, launching DDOS attacks, ...

[Technology Review, September 24, 2004]

The Costs Can Be Staggering



Types of Computer Misuse (1)

[Neumann and Parker 1989]

■ External

- ▼ Visual spying Observing keystrokes or screens
- ▼ Misrepresentation Deceiving operators and users
- ▼ Physical scavenging “Dumpster diving” for printouts

■ Hardware misuse

- ▼ Logical scavenging Examining discarded/stolen media
- ▼ Eavesdropping Intercepting electronic or other data
- ▼ Interference Jamming, electronic or otherwise
- ▼ Physical attack Damaging or modifying equipment
- ▼ Physical removal Removing equipment & storage media

Types of Computer Misuse (2)

[Neumann and Parker 1989]

■ Masquerading

- ▼ Impersonation
- ▼ Piggybacking
- ▼ Spoofing
- ▼ Network weaving

Using false identity external to computer
Usurping workstations, communication
Using playback, creating bogus systems
Masking physical location or routing

■ Pest programs

- ▼ Trojan horses
- ▼ Logic bombs
- ▼ Malevolent worms
- ▼ Viruses

Implanting malicious code
Setting time or event bombs
Acquiring distributed resources
Attaching to programs and replicating

■ Bypasses

- ▼ Trapdoor attacks
- ▼ Authorization attacks

Utilizing existing flaws
Password cracking

Types of Computer Misuse (3)

[Neumann and Parker 1989]

■ Active misuse

▼ Basic

Creating false data, modifying data

▼ Denials of service

Saturation attacks

■ Passive misuse

▼ Browsing

Making random or selective searches

▼ Inference, aggregation

Exploiting traffic analysis

▼ Covert channels

Covert data leakage

■ Inactive misuse

Failing to perform expected duties

■ Indirect misuse

Breaking crypto keys

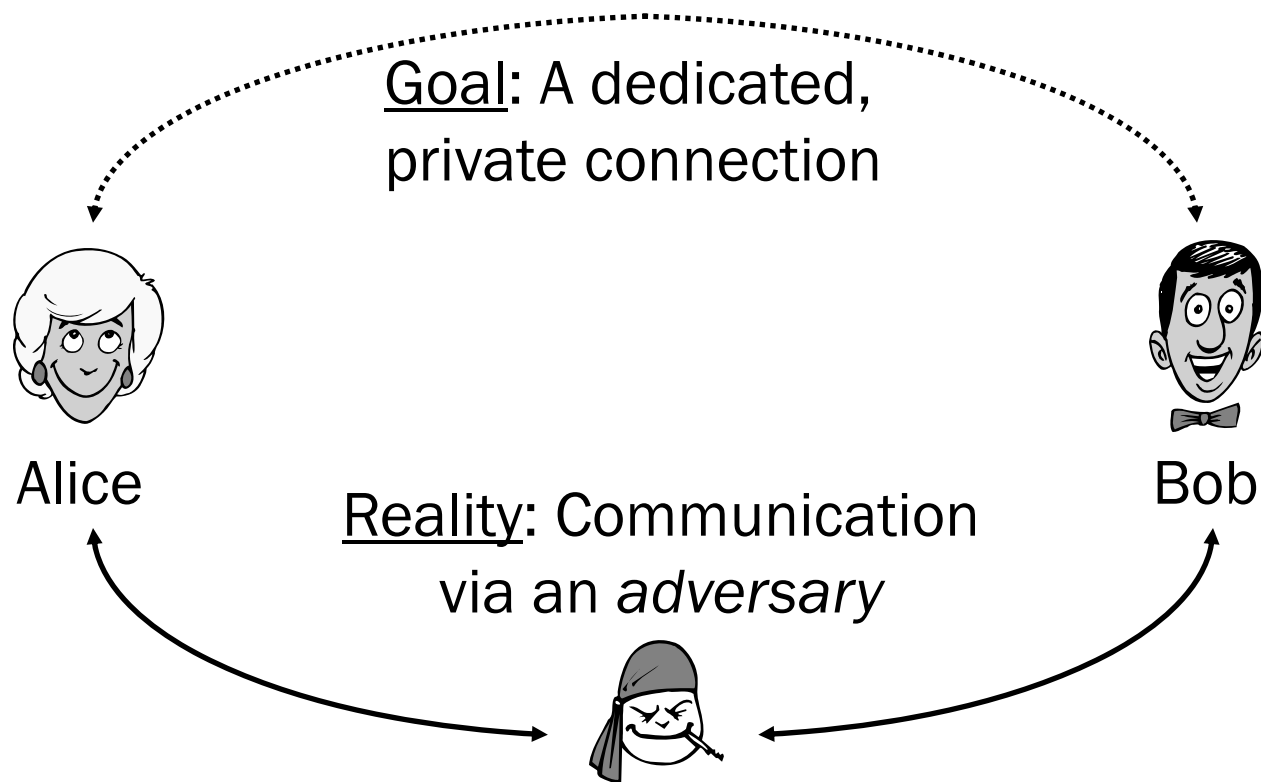
Threat Models

- **Can't protect against everything**
 - ▼ Too expensive
 - ▼ Too inconvenient
 - ▼ Not worth the effort

- **Identify the most likely ways your system will be attacked**
 - ▼ Identify likely attackers and their resources
 - ▼ Dumpster diving or rogue nation?
 - ▼ Identify consequences of possible attacks
 - ▼ Mild embarrassment or bankruptcy?
 - ▼ Design security measures accordingly
 - ▼ Accept that they will not defend against all attacks

Cryptography

- Study of techniques to communicate securely in the presence of an *adversary*
- Traditional scenario



Adversary's Goals

1. **Observe what Alice and Bob are communicating**
 - ▼ Attacks on “confidentiality” or “secrecy”
 2. **Observe that Alice and Bob are communicating, or how much they are communicating**
 - ▼ Called “traffic analysis”
 3. **Modify communication between Alice and Bob**
 - ▼ Attacks on “integrity”
 4. **Impersonate Alice to Bob, or vice versa**
 5. **Deny Alice and Bob from communicating**
 - ▼ Called “denial of service”
- **Cryptography traditionally focuses on preventing (1) and detecting (3) and (4)**

Symmetric Encryption

- A symmetric encryption scheme is a triple $\langle G, E, D \rangle$ of efficiently computable functions

- ▼ G outputs a “secret key” K

$$K \leftarrow G(\cdot)$$

- ▼ E takes a key K and “plaintext” m as input, and outputs a “ciphertext”

$$c \leftarrow E_K(m)$$

- ▼ D takes a ciphertext c and key K as input, and outputs \perp or a plaintext

$$m \leftarrow D_K(c)$$

- ▼ If $c \leftarrow E_K(m)$ then $m \leftarrow D_K(c)$

- ▼ If $c \leftarrow E_K(m)$, then c should reveal “no information” about m

Public Key Encryption

- A public key encryption scheme is a triple $\langle G, E, D \rangle$ of efficiently computable functions

- ▼ G outputs a “public key” K and a “private key” K^{-1}

$$\langle K, K^{-1} \rangle \leftarrow G(\cdot)$$

- ▼ E takes public key K and plaintext m as input, and outputs a ciphertext

$$c \leftarrow E_K(m)$$

- ▼ D takes a ciphertext c and private key K^{-1} as input, and outputs \perp or a plaintext

$$m \leftarrow D_{K^{-1}}(c)$$

- ▼ If $c \leftarrow E_K(m)$ then $m \leftarrow D_{K^{-1}}(c)$

- ▼ If $c \leftarrow E_K(m)$, then c and K should reveal “no information” about m

Message Authentication Codes

- A message authentication code (MAC) scheme is a triple $\langle G, T, V \rangle$ of efficiently computable functions

- ▼ G outputs a “secret key” K

$$K \leftarrow G(\cdot)$$

- ▼ T takes a key K and “message” m as input, and outputs a “tag” t

$$t \leftarrow T_K(m)$$

- ▼ V takes a message m , tag t and key K as input, and outputs a bit b

$$b \leftarrow V_K(m, t)$$

- ▼ If $t \leftarrow T_K(m)$ then $V_K(m, t)$ outputs 1 (“valid”)

- ▼ Given only message/tag pairs $\{\langle m_i, T_K(m_i) \rangle\}_i$, it is computationally infeasible to compute $\langle m, t \rangle$ such that

$$V_K(m, t) = 1$$

for any new $m \neq m_i$

Digital Signatures

- A digital signature scheme is a triple $\langle G, S, V \rangle$ of efficiently computable algorithms

- ▼ G outputs a “public key” K and a “private key” K^{-1}

$$\langle K, K^{-1} \rangle \leftarrow G(\cdot)$$

- ▼ S takes a “message” m and K^{-1} as input and outputs a “signature” σ

$$\sigma \leftarrow S_{K^{-1}}(m)$$

- ▼ V takes a message m , signature σ and public key K as input, and outputs a bit b

$$b \leftarrow V_K(m, \sigma)$$

- ▼ If $\sigma \leftarrow S_{K^{-1}}(m)$ then $V_K(m, \sigma)$ outputs 1 (“valid”)

- ▼ Given only K and message/signature pairs $\{\langle m_i, S_{K^{-1}}(m_i) \rangle\}_i$, it is computationally infeasible to compute $\langle m, \sigma \rangle$ such that

$$V_K(m, \sigma) = 1$$

any new $m \neq m_i$

Hash Functions

- A hash function is an efficiently computable function h that maps an input x of arbitrary bit length to an output

$$y \leftarrow h(x)$$

of fixed bit length

- ▼ Preimage resistance: Given only y , it is computationally infeasible to find any x' such that $h(x') = y$.
- ▼ 2nd preimage resistance: Given x , it is computationally infeasible to find any $x' \neq x$ such that $h(x') = h(x)$.
- ▼ Collision resistance: It is computationally infeasible to find any two distinct inputs x, x' such that $h(x) = h(x')$.

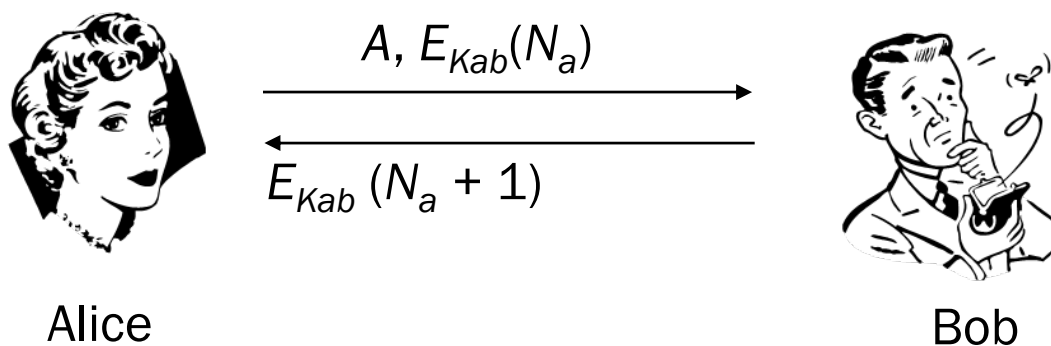
Pick the Right Tool for the Job

- Know what each tool does
 - ▼ E.g., encryption *does not* tell you who sent a message
 - ▼ E.g., digital signatures *do not* prevent a message from being tampered with

- Seems obvious, but often not true in practice

Example of Challenge-Response

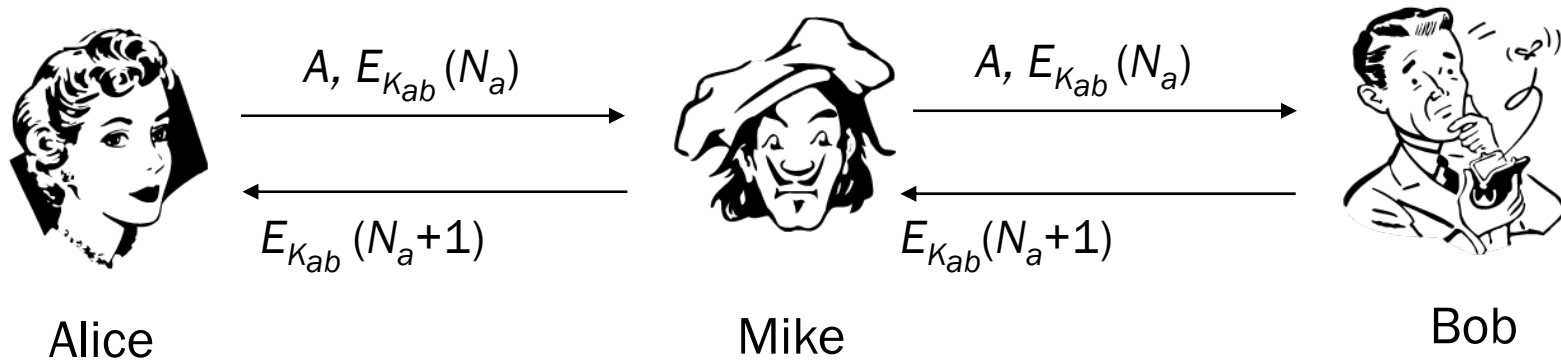
- Alice and Bob share a key K_{ab}
- Alice wishes to authenticate Bob



- Alice is now convinced she's talking to Bob
 - ▼ Should she be?

An "Attack"

- Alice and Bob share a key K_{ab}
- Alice wishes to authenticate Bob



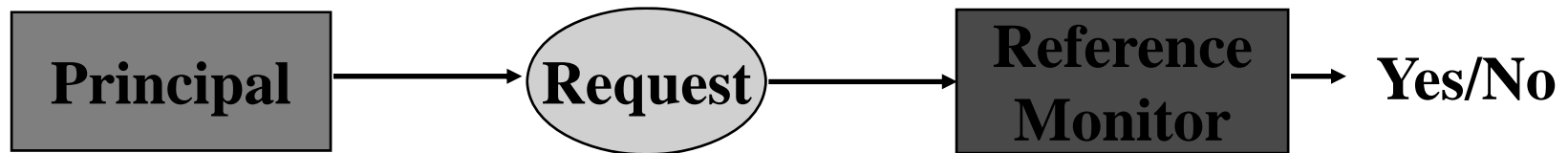
- Alice thinks she is talking to Bob
- In fact, she is talking to Mike (man-in-the-middle)

Why Is Security Hard?

- We have all these tools...
- Problems can't be solved by direct application of building blocks
 - ▼ E.g., messages often need padding before they can be encrypted
- Composing building blocks yields new vulnerabilities
 - ▼ E.g., adversary can interact with valid users in protocol, obtain information that can allow him to impersonate valid user
 - ▼ Replay (freshness attacks)
 - ▼ Insert (e.g., type flaw attacks, man-in-the-middle attacks)
 - ▼ Initiate different protocol sessions (parallel session attacks)

Access Control

- Principal makes a request for an object
- Reference monitor grants or denies the request



Ex: Editor

Send file

File server

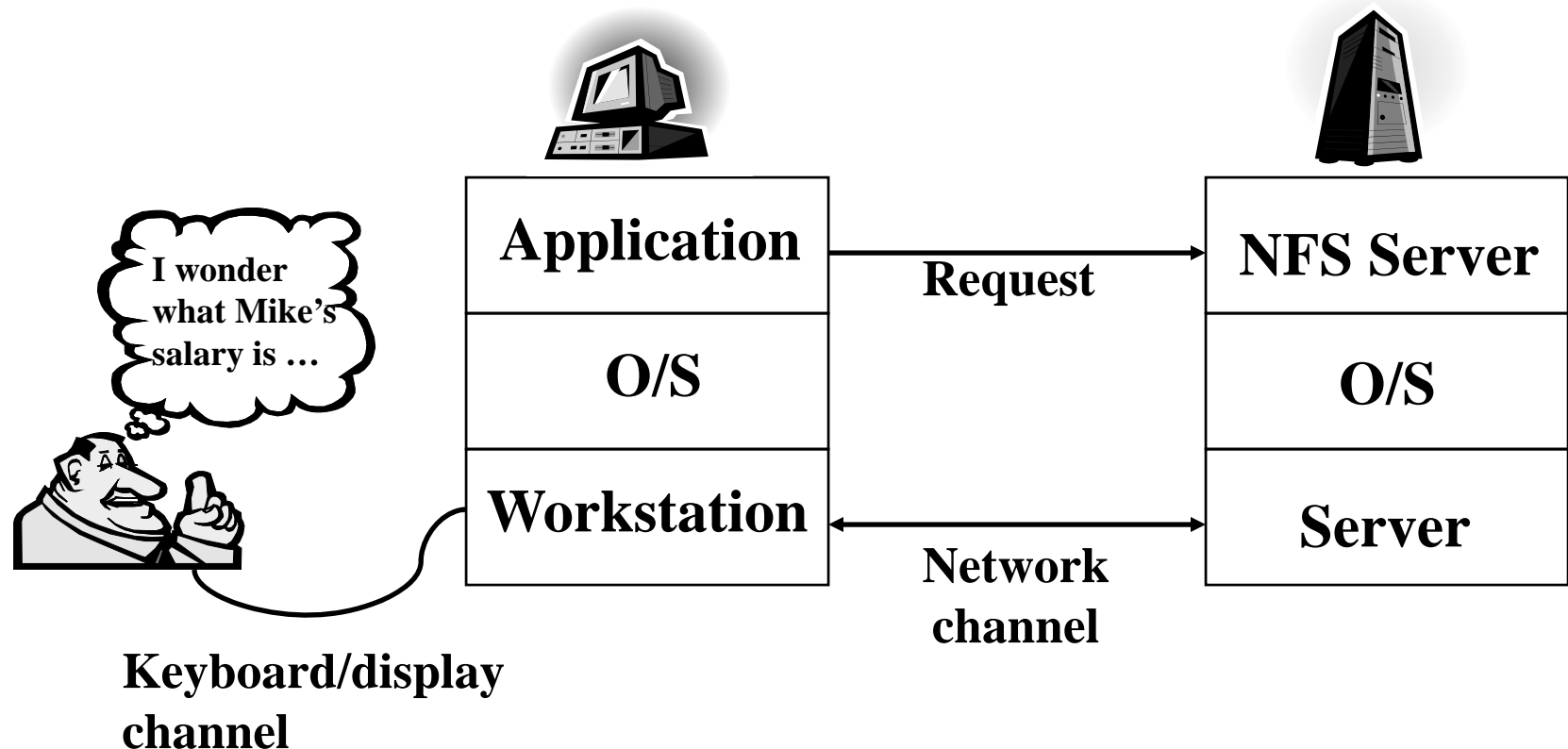
Ex: Host

Route packet

Firewall

- Authorization: Determining whether access should be allowed
 - The “decision” the reference monitor must make
- Authentication: Determining who made request

The Challenge



- Who is the request "from"?
 - The user? The workstation? The application?
 - All of the above?

User Authentication

- Typically based on one or more of
 - ▼ Something you know
 - ▼ Something you have
 - ▼ Something you “are”

- Two-factor authentication typically refers to using two of these

Something You Know

- Password or PIN
- Social security number
- Mother's maiden name
- Pet's name
- A picture

Something You Have

- Physical key
- Proximity card
- RSA SecureID token
- Smartcard or credit card
- SecureNet token
- STU-III key
- Cell phone
- ...

Something You Are

- Typically refers to biometrics
- Many options
 - ▼ Face
 - ▼ Fingerprint
 - ▼ Voiceprint
 - ▼ Iris
- Accuracy is more of an issue for biometrics than other user authentication technologies
 - ▼ False accepts: Accepting an authentication attempt by a person who is not the claimed person
 - ▼ False rejects: Rejecting an authentication attempt by the claimed person

Human-generated Cryptographic Keys

- An alternate use of passwords is to generate a repeatable cryptographic key
 - ▼ Most commonly used for file encryption
 - ▼ Particularly the encryption of other keying material
- Some research has been done to generate repeatable and strong cryptographic keys from biometric information
 - ▼ Much more work left to do, though
- Key difference is the *threat model*
 - ▼ In user authentication, a trusted monitor performs the authentication and limits the number of incorrect attempts
 - ▼ In key generation, typically there is no trusted monitor to limit attempts, and so it must be computationally intractable to break

Beyond User Authentication

- User authentication is an obvious usability issue for computer systems
 - ▼ It *requires* user interaction
- But it is not the only one, or even the most difficult one
- Currently there is significant debate in the community as to the extent other security mechanisms should be made visible to users or be hidden

Usability

Usability is the extent to which users can access the functionality of a system with effectiveness, efficiency, and satisfaction to achieve specific goals. ...

- **Effectiveness:** The degree to which a system fulfills its intended purpose and supports its users by enabling accurate and complete task performance.
- **Efficiency:** The resources expended by a system's users in achieving accurate and complete task performance.
- **User Satisfaction:** The users' perceived acceptability of the system.

Federal Aviation Administration, www.hf.faa.gov

- Note focus on “task performance” (functional properties)

Trust vs Trustworthiness

Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another.

Rousseau et al. Not so different after all: A cross-discipline view of trust. *Academy of Management Review* 32(3):393–404, 1998.

Trustworthiness ... asserts that the system does what is required—despite environmental disruption, human user and operator errors, and attacks by hostile parties—and that it does not do other things.

Schneider, ed. *Trust in Cyberspace*. Committee on Information Systems Trustworthiness, National Research Council, 1999.

Trust and Usability

- Usability promotes trust
 - ▼ Fraudsters know this well
 - ▼ Example: phishing

Trustworthiness and Usability

- **If a system is not usable, then it is not trustworthy**
 - ▼ Example: Florida ballot in 2000 U.S. presidential election
 - ▼ Example: U.S.S.R.'s Phobos 1 satellite, lost because of a single mistyped character

- **Are more usable systems more trustworthy?**
 - ▼ Not necessarily

- **Are more trustworthy devices necessarily more usable?**
 - ▼ Not necessarily, but must be usable to be trustworthy

Improving Usability and Trustworthiness

- How can we increase the combination of usability and trustworthiness?

- Two schools of thought
 - ▼ Security needs to disappear
 - ▼ Security should not disappear, but should be presented using better metaphors

The “Security Must Disappear” Argument

- **Security is hard to understand**
 - ▼ What is a “public” key?
 - ▼ Does encryption make web purchases safe?
- **Security is hard to use**
 - ▼ What is the right Java policy file?
 - ▼ Many steps needed to get a certificate
 - ▼ Try sharing a file with (only) a group of people
- **Security is annoying**
 - ▼ “I can’t get to your directory”
 - ▼ “I forgot my Amazon (Yahoo, E-Trade, ...) password”
 - ▼ “You can’t do that from behind a firewall”
- **The number of devices is exploding**
 - ▼ Most never see a professional admin, and so must be self-managing

The “Security Must Disappear” Argument

- We have made great strides on implementing invisible (or mostly invisible) security
 - ▼ SSH, SSL/TLS, VPNs
 - ▼ Automatic updates (e.g., Windows update)
 - ▼ Identity-based signatures and encryption
 - ▼ Wireless security tokens
- However, these sacrifice some security (or functionality) for the sake of invisibility in practice

The “Security Cannot Disappear” Argument

■ Invisible security

- ▼ Works only at the extremes, or at the expense of security
- ▼ Impossible in the “fuzzy” middle, where it matters
 - ▼ When is an installed/run program a virus?
- ▼ Leads to things not working for reasons the user doesn’t understand

■ “Mostly invisible” security (augmented with “Are you sure?” warnings) yields only two realistic cases

- ▼ Always heed the warning: same as invisible security
- ▼ Always ignore the warning: security is compromised

■ Users handle their own security in real life, all the time

- ▼ Vehicle, home, office keys; keys, alarms
- ▼ Cash, checks, credit cards, ATM cards, PINs, safe deposit boxes, IDs
- ▼ Purchases, transactions, contracts

The “Security Cannot Disappear” Argument

What works in security UI

■ Clear, understandable metaphors

- ▼ Abstract out the mechanism meaningfully for users
- ▼ Use physical analogs where possible

■ User-centric design

- ▼ Start with the user model, design the underlying mechanism to implement it

■ Unified security model

- ▼ Across applications: “Windows GUI for security”

■ Meaningful, intuitive user input

- ▼ Don’t assume things on the user’s behalf—figure out how to ask so that the user can answer intelligently