# Biometrics

- Something you are
- A characteristic of the body
- Presumed unique and invariant over time

# Common Biometrics

- Fingerprint
- Iris scan
- Retinal scan
- Hand geometry
- Facial recognition

# Fingerprints

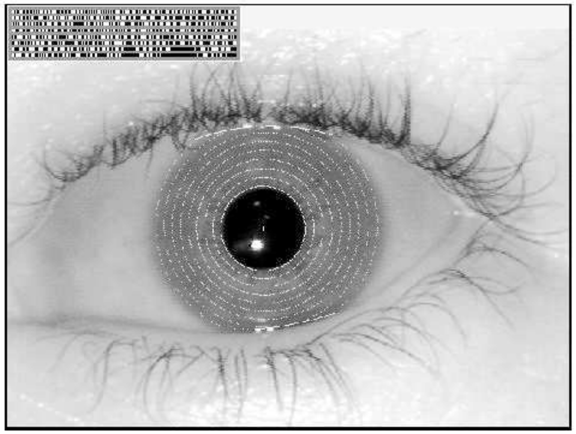
- Uniqueness well-established (not an idle issue; Bertillon measurement were once thought unique)
  ☞Fingerprints are *congenital*, not genetic
- Lots of backup fingers
- Commodity hardware available; even built in to some newer laptops
- But — bad connotations; fingerprints have traditionally been associated with criminals

# Fingerprint Recognition

- Image recognition technology
- Find significant features
- Does *not* match entire image

# Iris Scans

- Considered one of the most accurate biometrics
- Uses patterns in the iris of the eye that form after birth
- Hard part in some applications: finding the eye
- People do not like to stare into scanners



# Retinal Scan

- Looks at patterrn of blood vessels inside the eye
- Must put eye up to scanner
- Most people *really* dislike scanners that shine things into their eyes. “You’re going to shine a *what* into my eye?!”
- Falling out of favor compared to iris scans

# Hand Geometry

- Requires somewhat fussy hand-positioning
- Relatively easy to use; few acceptability issues
- Used at Disney World; formerly used by U.S. Immigration

# Facial Recognition

- Not very accurate yet
- Relies on geometry of key features — eye spacing, ears, etc.
- Major target market: walk-through authentication
- Some countries now prohibit smiling for passport pictures, to aid future automated recognizers

# Other Biometrics

- Voiceprint
- Typing rhythm

# Advantages of Biometrics

- You can’t forget your fingers
- You can’t lend your eyes to a friend
- You can’t fake a fingerprint
- Why aren’t they used more?
- Maybe they’re not that secure. . .

# Lenovo's Statement on Fingerprint Recognition

"Non-Embedded Security Subsystem models can be configured for fingerprint only authentication that does not also require typing in a password. *This configuration offers convenience, but security is not significantly better than using typed passwords only* [emphasis added]."

# Some Problems with Biometrics

- False accept rate
- False reject rate
- Fake body parts
- “Bit replay”
- Non-reproducibility

# False Accept Rate

- No biometric system is perfect
- Reducing false accept rate increases false reject rate
- Usual metric: what is the true accept rate for a given false accept rate?
- Substantial difference between different products
- For fingerprints, best is .994 TAR @ $10^{-4}$ FAR; .999 TAR @ $10^{-2}$ FAR
- For faces, .72 TAR @ $10^{-4}$ FAR; .90 TAR @ $10^{-2}$ FAR. (Lighting matters a lot for facial recognition.)
- All systems work much better for one-to-one match than “does this biometric match something in the database”

# False Reject Rate

- People change
- Cuts, scars, glasses, colds, bandages, etc.
- Problems in original image acquisition

# Fake Body Parts

- Thieves cut off someone's finger to steal his fingerprint-protected car (`http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm`)
- Biometric sensors have been fooled by "Gummi Bear" fingerprints, close-up pictures of face
- One solution: use "liveness" detectors — temperature, blood flow, etc.
- Another solution: use biometrics only when under observation

# Bit Replay

- Ultimately, a biometric translates to a string of bits
- If the biometric sensor is remote from the accepting device, someone can inject a replayed bit stream
- What if someone hacks a server and steals a biometric? You can't change your fingerprints. . .
- Encryption helps; so does tamper-resistance
- Relying on human observation may help even more

# Non-Reproducibility

- Biometric matching compares an image to a template or set of templates
- It is hard to reduce a biometric to a reproducible set of bits, suitable for use as a cryptographic key
- This makes it hard to use a biometric to protect locally-stored keys; you're really relying on the operating system

# Microsoft's Fingerprint Reader

- Can be used in place of login password
- Can be used for Web passwords
- But — you're warned not to use it for sensitive sites. Why not?
- Because the actual password has to be sitting on the disk somewhere, largely unprotected
- (Besides, it's probably not using high-quality fingerprint recognition; most of their clientele would notice a false negative more than a false positive.)

# Using Biometrics

- Biometrics work best in public places or under observation
- Remote verification is difficult, because verifier doesn't know if it's really a biometric or a bit stream replay
- Local verification is often problematic, because of the difficulty of passing the match template around
- Users don't want to rely on remote databases, because of the risk of compromise and the difficulty of changing one's body
- Best solution: use a biometric to unlock a local token
- Another solution: put the template on a mag stripe card in the user's possession; that supplies it to a local verification station. But how is the template authenticated?

# Certificates

- Binding of a name to a public key
- (Similarly, could sign a biometric template)
- Digitally signed by a *certificate authority* (CA)
- Typically, user generates key pair, and presents public key and proof of identity
- CA signs the certificate and gives it back
- Note: certificates are self-secured; they can be verified offline

# Who Issues Certificates?

- Identity-based: some organization, such as Verisign, vouches for your identity
  ☞Cert issuer is not affiliated with verifier
- Authorization-based: accepting site issues its own certificates
  ☞Cert issuer acts on behalf of verifier
- Identity-based certificates are better when user has no prior relationship to verifier, such as secure Web sites
- Authorization-based certs are better when verifier wishes to control access to certain resources — no need to trust external party
- CS dept web certificate at `http://www.cs.columbia.edu/~smb/classes/f05/cs-cert.txt`
- University web certificate at `http://www.cs.columbia.edu/~smb/classes/f05/cu-cert.txt`

# Things to Notice About Certificates

- Signer (the university didn't issue the department's certificate)
- Validity dates
- Algorithms (RSA, SHA1, MD5)
- Certificate usage — encryption and authentication, but *not* for issuing other certificates
- Certificate Revocation List (CRL)

# How Do You Revoke a Certificate?

- Revocation is hard! Verification can be done offline; revocation requires some form of connectivity
- Publish the URL of a list of revoked certificates
  ☞One reason for certificate expiration dates; you don't need to keep revocation data forever
- Online status checking
- STU-IIIs use flooding algorithm — works well because of comparatively closed communities

# What Certificates Do You Accept?

- Browers and (some) mailers have built-in list of CAs
- What were the listing criteria?
- Do you trust the CAs?
- What are their policies? Verisign's *Certification Practice Statement* (CPS) is at `http://www.verisign.com/repository/CPS/VeriSignCPSv3_03.15.05.pdf`. Have you read it?
- All certificate verification has to start from *trust anchors*

# Systems Considerations

- The last few problems are problems only in certain situations
- Whether or not biometrics are suitable depends on the situation
- In fact, all authentication schemes are situation-dependent
- Authentication is a *systems problem*

# Historical Note

- The Unix password scheme was designed for *time-sharing systems*
- Users logged in from dumb terminals, with no local computing power
- It was intended for an environment with little or no networking
- Do these assumptions still hold?

# Scenarios

- Parties: Prover ($P$), Verifier ($V$), Issuer ($I$)
- Issuer supplies credentials; Prover tries to log in to Verifier
- How many verifiers?
- How many different provers?
- What sort of networking is available?
- What sort of computer is P using?
- What is the relationship of P, V, and I?
- What are the adversary's powers?

# Example: Large Enterprise

- Comparatively homegenous computing environment
- P trusts own computer
- Centralized I, many Vs
- Perhaps use Kerberos
  - Uses password as cryptographic key
  - Uses centralized database of plaintext keys (but not passwords)
  - Little risk of keystroke loggers
  - Use management chain to authorize password change

# Example: Consumer ISP

- Unsophisticated user base
- Low cost is very important
- Trusted, high-speed internal network
  - Separate login and email passwords
  - Store the dial-up login password on the user's machine; maybe email password, too — must avoid help-desk calls
  - Use password hints; maybe even let customer care see part of the password or hints
  - Probably low risk of password file compromise
  - File theft may be less of a risk than keystroke loggers
  - Many Vs for login; several Vs for email. Use centralized back-end database, with no crypto

# Example: University Computer Center

- Central V database
- Wireless networking
- Very heterogenous client computers
  - Kerberos not usable; too many different client machines
  - Serious danger of eavesdropping; use encrypted logins only
  - Use back-end process to distribute password database, or use online query of it
  - Classical password file may be right

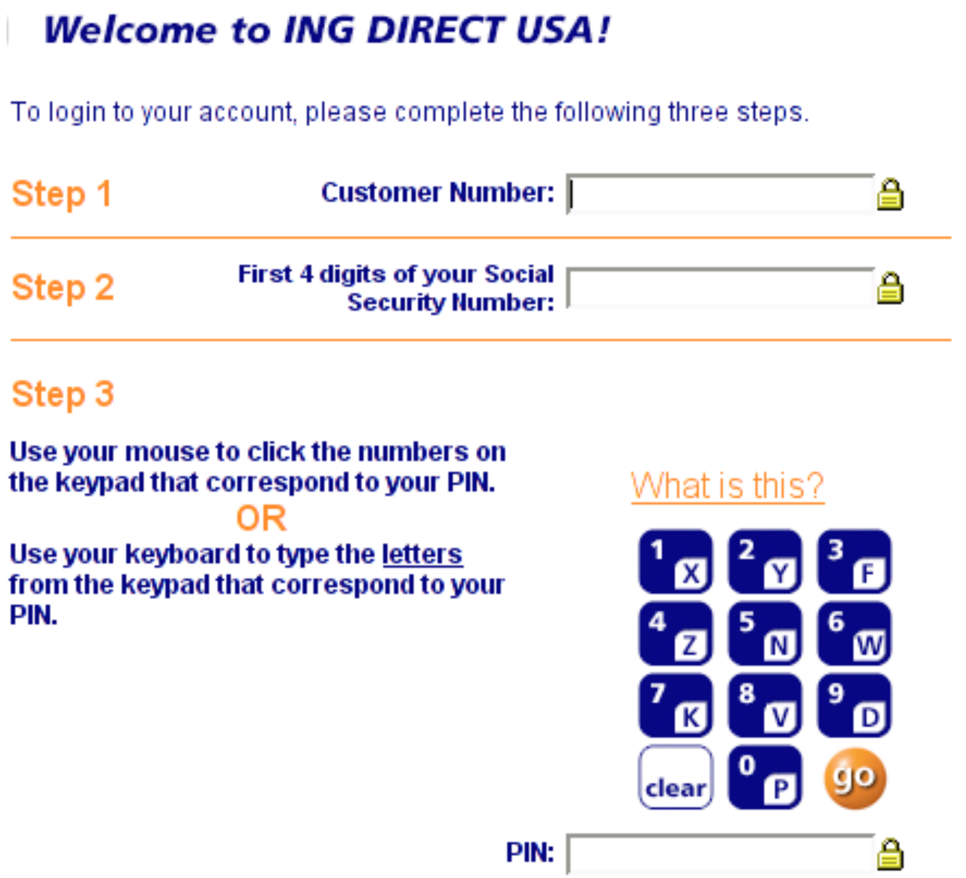
# Example: Consumer Web Site

- Low-value logins
- Can't afford customer care
- Use email addresses as login names; email password on request
- Don't worry much about compromise

# Example: Financial Services Web Site

- High-value login
- Protecting authentication data is crucial
- Customer care is moderately expensive; user convenience is important, for competitive reasons
  - Perhaps use tokens such as SecurID, but some customers don't like them
  - Do not let customer care see any passwords
  - Require strong authentication for password changes; perhaps use physical mail for communication
  - Guard against compromised end-systems

# New ING Direct Login Screen




The keypad letters are randomly chosen and change each time, to guard against keystroke loggers

# Example: Military Computer and Email Systems

- Captive user population — and they'll be there for a few years
- User training possible
- High value in some situations
- Everyone has to carry ID anyway
  - Convert dog tag to smart card containing public/private key pair
  - Use it for physical ID (Geneva Convention) and for computer login
  - Use PIN to protect private key

# The Threat Model Wasn't Right

- Prisoners of war *must* show their dog tags
- That same device can provide access to sensitive computer systems
- POWs can be "pressured" to disclose their PINs
- Result: some pilots in Iraq destroyed the chip before missions
- The designers forgot one thing: the risk of physical capture of the device *and* the device owner

# Designing Authentication Systems

- There is no one right answer
- The proper design depends on the circumstances
- The goal is *information security*
- Finding the proper balance requires good engineering