

---

# Biometrics; Authentication as a Systems Problem



---

# Biometrics

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

Metanote: biometrics is an area of rapid progress; some of the limitations I describe here are likely to change in the near future. Exercise: which of the problems are likely to remain difficult issues for system designers?

---

## Common Biometrics

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



---

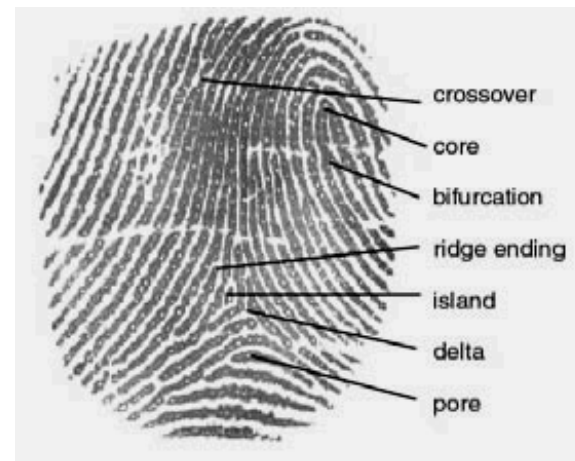
# Fingerprints

- 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 laptops and phones
- But—in some places, bad connotations; fingerprints have traditionally been associated with criminals. Easing now as fingerprint authentication becomes more common (e.g., many border crossings).

---

# Fingerprint Recognition

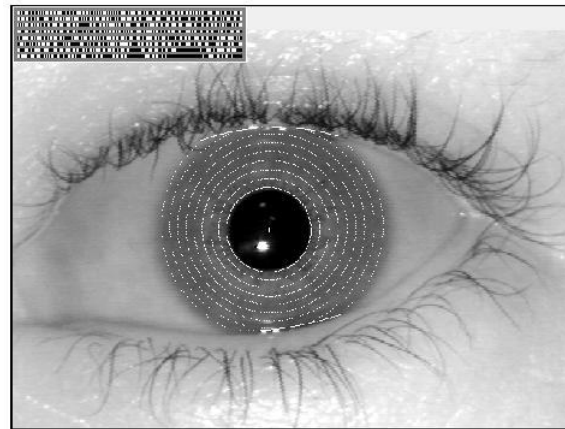
- Image recognition technology
- Find significant features
- Does *not* match entire image
- Matching isn't as easy as you see on television
- New automated systems have improved scanning speed, but there can still be accuracy issues



---

## 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 pattern of blood vessels inside the eye
- Must put eye up to laser 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 (but they've switched to fingerprints)





---

# Facial Recognition

- Not very accurate yet, but getting better
- Relies on geometry of key features—eye spacing, ears, etc.
- Major target market: walk-through authentication (and detection)
- 👉 Also: finding suspects in a crowd
- Some countries (US, UK, Germany, probably others) now prohibit smiling for passport pictures, to aid (future) automated recognizers

---

## Other Biometrics

- Voiceprint
- Typing rhythm

# Human Voice Recognition

- Press the red button to “go secure”
- Crypto magic happens, followed by the display of some hex digits
- Each party reads the hex digits to the other
- You must recognize the other party’s voice speaking those digits

👉 Will computers be able to fake that soon?



(Photo courtesy Matt Blaze)

---

## 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].”

(Note: “Embedded Security” models, which use a tamper-resistant chip, are more secure; more on that later.)

---

## Some Problems with Biometrics

- False accept rate
- False reject rate
- Fake (or “detached”) body parts
- “Bit replay”
- Non-reproducibility
- Many biometrics are *public*

---

## 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 (NIST, 2004)
- 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.)

---

# iPhone Fingerprint Recognition

- New iPhones have a fingerprint recognizer in the Home button: replace the PIN to unlock the phone
- Uses advanced technology; claimed to be immune to fake fingerprints, detached body parts, etc.
- Apple says the odds on a random finger matching are 1 in 50,000—and only five tries are allowed
- ☞  $1 - (1 - 50,000)^5 \approx \frac{1}{10,000}$  — the same as one guess at a 4-digit PIN
- The Chaos Computer Club has already shown that those claims are incorrect: use a high-resolution camera, a suitable printer, and some white glue...

---

## Is That Secure?

- Lossy mapping of fingerprint images to template; cannot reconstruct fingerprint from it
- Templates stored in secure coprocessor; communications from sensor to coprocessor are encrypted
- Data is *not* backed up in cleartext to iCloud
- PIN reentry is required after 48 hours, after failed authentication attempts, or after rebooting

---

## What is “Secure”?

- What is being protected?
- What is the threat model?
- We can't answer “is it secure?” without defining what we're trying to protect!
- ☞ Fingerprint logs into phone: probably secure enough
- ☞ Can the fingerprint data be protected? Harder
- ☞ Fingerprint processor is vouching for user's identity: hardest of all

---

## 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 tamper-resistant token or chip; store keys there
- 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?



---

## Signed Templates

- Can digitally sign a biometric template
- Medium doesn't matter; signed template is self-authenticating
- Verifier can operate offline
- But—which digital signatures should it trust?
- How do you revoke *authorization*?

---

## Systems Considerations

- The last two issues illustrate an important point: authentication doesn't stand by itself
- Whether or not biometrics are suitable depends on the situation
- How you set up your biometric authentication matters, too
- In fact, all authentication schemes are situation-dependent
- Authentication is a *systems problem*

---

## Certificates as a Systems Issue

- The basic concept—a digitally-signed binding of a name to a public key—is simple enough
- (Just as we signed a biometric template)
- But—it's more complicated than that

---

## Issuing Certificates

- Typically, user generates key pair, and presents public key and proof of identity
- CA signs the certificate and gives it back
- Note: certificates are also 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 its own resources—no need to trust external party
- See CS dept and university web certificates at  
<http://www.cs.columbia.edu/~smb/classes/f14/cs-cert.txt>  
and  
<http://www.cs.columbia.edu/~smb/classes/f14/cu-cert.txt>

---

## Things to Notice About Certificates

- Signer (the university didn't issue the department's certificate)
- Validity dates
- Algorithms (RSA, SHA1)
- Until recently, the CS department's certificate used MD5.
- (See older year's certificates at .../f07/...)
- They both use 2048-bit keys: modern standard
- Certificate usage—encryption and authentication, but *not* for issuing other certificates
- Certificate Revocation List (CRL)
- OCSP server: Online Certificate Status Protocol

---

## 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-III's use flooding algorithm—works well because of comparatively closed communities

---

## Why Revoke Certificates?

- Private key compromised
- Cancel authorization associated with certificate
- Note the difference between identity and authorization certificates here
- CA key compromised, e.g., DigiNotar



---

## What Certificates Do You Accept?

- Browsers 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/CPSv3.8.1_final.pdf`. Have you read it?
- All certificate verification has to start from *trust anchors*; these must be locally provisioned. (Firefox trusts about 200 CAs; Windows IE trusts > 300—and at least 10% are agencies of some government)

---

## The Risks of Built-in CAs

|                                                      |                          |
|------------------------------------------------------|--------------------------|
| AOL Time Warner Root Certification Authorit...       | Builtin Object Token     |
| ▼ Autoridad de Certificacion Firmaprofesional CIF... |                          |
| Autoridad de Certificacion Firmaprofesional ...      | Builtin Object Token     |
| ▼ Baltimore                                          |                          |
| Baltimore CyberTrust Root                            | Builtin Object Token     |
| Baltimore CyberTrust Code Signing Root               | Software Security Device |
| Baltimore CyberTrust Mobile Root                     | Software Security Device |
| ▼ BankEngine Inc.                                    |                          |
| bankengine                                           | Software Security Device |
| ▼ BelSign NV                                         |                          |
| BelSign Object Publishing CA                         | Software Security Device |
| BelSign Secure Server CA                             | Software Security Device |

It's amusing to read Baltimore's complex corporate history

---

## Historical Note on Passwords

- 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 his/her 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 recovery

---

## Example: Wireless Consumer ISP

- Unsophisticated user base
- Low cost is very important
- Trusted, high-speed internal network
  - Separate login and email passwords
  - Store the wireless 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
  - Reasonably 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 new password on request (but why not send out old password?)
- Don't worry much about compromise



---

## Example: Mailman Mailing List Server

- Use of password is rare (and often non-existent)
- Solution: auto-generate passwords; email them to users in the clear
- No serious resources at risk, especially for public mailing lists
- Better choice than asking users to pick a password—people *will* reuse some standard password
- But—the password may give access to the archives for closed mailing lists

---

## 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
  - Today, perhaps use smartphones as second factor
  - 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

---

## iPhone 5S

- My fingerprint, my phone
- Fingerprint database backed up via iTunes—but it's encrypted to the phone
- More convenient than (short) PIN; security is probably comparable
- Spoofing seems possible—but does it matter? What is the threat model? The attack is *targeted*; most phone locks are designed to protect against casual thieves.

# A Previous ING Direct Login Screen

## Welcome to ING DIRECT USA!

To login to your account, please complete the following three steps.

**Step 1** Customer Number:  

**Step 2** First 4 digits of your Social Security Number:  

### Step 3

Use your mouse to click the numbers on the keypad that correspond to your PIN.

OR

Use your keyboard to type the letters from the keypad that correspond to your PIN.

What is this?



PIN:  

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

---

## It's Gone Now...

- Too complicated?
- Bypassed by the hackers?
- That happened to a similar scheme in Turkey within 24 hours...

# Some Sites Still Use It

**Westpac Online**  
**Sign In**

**1. Enter Customer ID**  [Hoax email alert](#)  
Using your keyboard

**Select the type of customer you are**  
 Personal     Business

---

**2. Enter password** [? Help](#)  
Using the buttons below

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |   |   |   |
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

Hmm—letters and number keys only; no punctuation. Other sites *require* punctuation in passwords. . .

---

## 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 in 2003 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