Review

• Chapter 1: Basic Concepts and Terminology
  – Integrity, Confidentiality, Availability, Authentication, and Accountability
  – Types of threats: active vs. passive, insider/outsider
  – Lots of terminology and general concepts

• Chapter 2: Basic Cryptographic Tools
  – Symmetric key encryption and secure hashing
  – Public key cryptography
  – Random Numbers

• Chapter 3 – User Authentication
Chapter 3

User Authentication
RFC 2828 defines user authentication as: “The process of verifying an identity claimed by or for a system entity.”
"On the Internet, nobody knows you're a dog."
Authentication Process

- fundamental building block and primary line of defense
- basis for access control and user accountability

- identification step
  - presenting an identifier to the security system

- verification step
  - presenting or generating authentication information that corroborates the binding between the entity and the identifier
**User Authentication**

the four means of authenticating user identity are based on:

<table>
<thead>
<tr>
<th>something the individual knows</th>
<th>something the individual possesses (token)</th>
<th>something the individual is (static biometrics)</th>
<th>something the individual does (dynamic biometrics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• password, PIN, answers to prearranged questions</td>
<td>• smartcard, electronic keycard, physical key</td>
<td>• fingerprint, retina, face</td>
<td>• voice pattern, handwriting, typing rhythm</td>
</tr>
</tbody>
</table>
Password Authentication

- widely used line of defense against intruders
  - user provides name/login and password
  - system compares password with the one stored for that specified login

- the user ID:
  - determines that the user is authorized to access the system
  - determines the user’s privileges
  - is used in discretionary access control
Password Vulnerabilities

- offline dictionary attack
- password guessing against single user
- workstation hijacking
- electronic monitoring
- specific account attack
- popular password attack
- exploiting user mistakes
- exploiting multiple password use
Countermeasures

- controls to prevent unauthorized access to password file
- intrusion detection measures
- rapid reissuance of compromised passwords
- account lockout mechanisms
- policies to inhibit users from selecting common passwords
- training in and enforcement of password policies
- automatic workstation logout
- policies against similar passwords on network devices
Use of Hashed Passwords

Figure 3.1 UNIX Password Scheme
UNIX Implementation

original scheme
- up to eight printable characters in length
- 12-bit salt used to modify DES encryption into a one-way hash function
- zero value repeatedly encrypted 25 times
- output translated to 11 character sequence

now regarded as inadequate
- still often required for compatibility with existing account management software or multivendor environments
Improved Implementations

- much stronger hash/salt schemes available for Unix
- recommended hash function is based on MD5
  - salt of up to 48-bits
  - password length is unlimited
  - produces 128-bit hash
  - uses an inner loop with 1000 iterations to achieve slowdown
- OpenBSD uses Blowfish block cipher based hash algorithm called Bcrypt
  - most secure version of Unix hash/salt scheme
  - uses 128-bit salt to create 192-bit hash value
Password Cracking

- **dictionary attacks**
  - develop a large dictionary of possible passwords and try each against the password file
  - each password must be hashed using each salt value and then compared to stored hash values

- **rainbow table attacks**
  - pre-compute tables of hash values for all salts
  - a mammoth table of hash values
  - can be countered by using a sufficiently large salt value and a sufficiently large hash length
## Observed Password Lengths

<table>
<thead>
<tr>
<th>Length</th>
<th>Number</th>
<th>Fraction of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55</td>
<td>.004</td>
</tr>
<tr>
<td>2</td>
<td>87</td>
<td>.006</td>
</tr>
<tr>
<td>3</td>
<td>212</td>
<td>.02</td>
</tr>
<tr>
<td>4</td>
<td>449</td>
<td>.03</td>
</tr>
<tr>
<td>5</td>
<td>1260</td>
<td>.09</td>
</tr>
<tr>
<td>6</td>
<td>3035</td>
<td>.22</td>
</tr>
<tr>
<td>7</td>
<td>2917</td>
<td>.21</td>
</tr>
<tr>
<td>8</td>
<td>5772</td>
<td>.42</td>
</tr>
<tr>
<td>Total</td>
<td>13787</td>
<td>1.0</td>
</tr>
</tbody>
</table>
## Passwords Cracked from a Sample Set of 13,797 Accounts

<table>
<thead>
<tr>
<th>Type of Password</th>
<th>Search Size</th>
<th>Number of Matches</th>
<th>Percentage of Passwords Matched</th>
<th>Cost/Benefit Ratioa</th>
</tr>
</thead>
<tbody>
<tr>
<td>User/account name</td>
<td>130</td>
<td>368</td>
<td>2.7%</td>
<td>2.830</td>
</tr>
<tr>
<td>Character sequences</td>
<td>866</td>
<td>22</td>
<td>0.2%</td>
<td>0.025</td>
</tr>
<tr>
<td>Numbers</td>
<td>427</td>
<td>9</td>
<td>0.1%</td>
<td>0.021</td>
</tr>
<tr>
<td>Chinese</td>
<td>392</td>
<td>56</td>
<td>0.4%</td>
<td>0.143</td>
</tr>
<tr>
<td>Place names</td>
<td>628</td>
<td>82</td>
<td>0.6%</td>
<td>0.131</td>
</tr>
<tr>
<td>Common names</td>
<td>2239</td>
<td>548</td>
<td>4.0%</td>
<td>0.245</td>
</tr>
<tr>
<td>Female names</td>
<td>4280</td>
<td>161</td>
<td>1.2%</td>
<td>0.038</td>
</tr>
<tr>
<td>Male names</td>
<td>2866</td>
<td>140</td>
<td>1.0%</td>
<td>0.049</td>
</tr>
<tr>
<td>Uncommon names</td>
<td>4955</td>
<td>130</td>
<td>0.9%</td>
<td>0.026</td>
</tr>
<tr>
<td>Myths and legends</td>
<td>1246</td>
<td>66</td>
<td>0.5%</td>
<td>0.053</td>
</tr>
<tr>
<td>Shakespearean</td>
<td>473</td>
<td>11</td>
<td>0.1%</td>
<td>0.023</td>
</tr>
<tr>
<td>Sports terms</td>
<td>238</td>
<td>32</td>
<td>0.2%</td>
<td>0.134</td>
</tr>
<tr>
<td>Science fiction</td>
<td>691</td>
<td>59</td>
<td>0.4%</td>
<td>0.085</td>
</tr>
<tr>
<td>Movies and actors</td>
<td>99</td>
<td>12</td>
<td>0.1%</td>
<td>0.121</td>
</tr>
<tr>
<td>Cartoons</td>
<td>92</td>
<td>9</td>
<td>0.1%</td>
<td>0.098</td>
</tr>
<tr>
<td>Famous people</td>
<td>290</td>
<td>55</td>
<td>0.4%</td>
<td>0.190</td>
</tr>
<tr>
<td>Phrases and patterns</td>
<td>933</td>
<td>253</td>
<td>1.8%</td>
<td>0.271</td>
</tr>
<tr>
<td>Surnames</td>
<td>33</td>
<td>9</td>
<td>0.1%</td>
<td>0.273</td>
</tr>
<tr>
<td>Biology</td>
<td>58</td>
<td>1</td>
<td>0.0%</td>
<td>0.017</td>
</tr>
<tr>
<td>System dictionary</td>
<td>19683</td>
<td>1027</td>
<td>7.4%</td>
<td>0.052</td>
</tr>
<tr>
<td>Machine names</td>
<td>9018</td>
<td>132</td>
<td>1.0%</td>
<td>0.015</td>
</tr>
<tr>
<td>Mnemonics</td>
<td>14</td>
<td>2</td>
<td>0.0%</td>
<td>0.143</td>
</tr>
<tr>
<td>King James bible</td>
<td>7525</td>
<td>83</td>
<td>0.6%</td>
<td>0.011</td>
</tr>
<tr>
<td>Miscellaneous words</td>
<td>3212</td>
<td>54</td>
<td>0.4%</td>
<td>0.017</td>
</tr>
<tr>
<td>Yiddish words</td>
<td>56</td>
<td>0</td>
<td>0.0%</td>
<td>0.000</td>
</tr>
<tr>
<td>Asteroids</td>
<td>2407</td>
<td>19</td>
<td>0.1%</td>
<td>0.007</td>
</tr>
<tr>
<td>TOTAL</td>
<td>62727</td>
<td>3340</td>
<td>24.2%</td>
<td>0.053</td>
</tr>
</tbody>
</table>

*Computed as the number of matches divided by the search size. The more words that need to be tested for a match, the lower the cost/benefit ratio.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>password</td>
<td>password</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
</tr>
<tr>
<td>2</td>
<td>123456</td>
<td>123456</td>
<td>password</td>
<td>password</td>
<td>password</td>
<td>password</td>
<td>password[10]</td>
</tr>
<tr>
<td>3</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>123456789</td>
</tr>
<tr>
<td>4</td>
<td>qwerty</td>
<td>abc123</td>
<td>qwerty</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>qwerty</td>
</tr>
<tr>
<td>5</td>
<td>abc123</td>
<td>qwerty</td>
<td>abc123</td>
<td>qwerty</td>
<td>12345</td>
<td>football</td>
<td>12345</td>
</tr>
<tr>
<td>6</td>
<td>monkey</td>
<td>monkey</td>
<td>123456789</td>
<td>123456789</td>
<td>123456789</td>
<td>qwerty</td>
<td>123456789</td>
</tr>
<tr>
<td>7</td>
<td>12345678</td>
<td>letmein</td>
<td>111111</td>
<td>1234</td>
<td>football</td>
<td>1234567890</td>
<td>letmein</td>
</tr>
<tr>
<td>8</td>
<td>letmein</td>
<td>dragon</td>
<td>12345678</td>
<td>baseball</td>
<td>1234</td>
<td>12345678</td>
<td>12345678</td>
</tr>
<tr>
<td>9</td>
<td>trustno1</td>
<td>111111</td>
<td>iloveyou</td>
<td>dragon</td>
<td>12345678</td>
<td>princess</td>
<td>football</td>
</tr>
<tr>
<td>10</td>
<td>dragon</td>
<td>baseball</td>
<td>adobe123[a]</td>
<td>football</td>
<td>baseball</td>
<td>1234</td>
<td>iloveyou</td>
</tr>
<tr>
<td>11</td>
<td>baseball</td>
<td>iloveyou</td>
<td>123123</td>
<td>12345678</td>
<td>welcome</td>
<td>login</td>
<td>admin</td>
</tr>
<tr>
<td>12</td>
<td>111111</td>
<td>trustno1</td>
<td>admin</td>
<td>monkey</td>
<td>1234567890</td>
<td>welcome</td>
<td>welcome</td>
</tr>
<tr>
<td>13</td>
<td>iloveyou</td>
<td>12345678</td>
<td>1234567890</td>
<td>letmein</td>
<td>abc123</td>
<td>solo</td>
<td>monkey</td>
</tr>
<tr>
<td>14</td>
<td>master</td>
<td>sunshine</td>
<td>letmein</td>
<td>abc123</td>
<td>111111</td>
<td>abc123</td>
<td>login</td>
</tr>
<tr>
<td>15</td>
<td>sunshine</td>
<td>master</td>
<td>photoshop[a]</td>
<td>111111</td>
<td>1qa2wsx</td>
<td>admin</td>
<td>abc123</td>
</tr>
<tr>
<td>16</td>
<td>ashley</td>
<td>123123</td>
<td>1234</td>
<td>mustang</td>
<td>dragon</td>
<td>121212</td>
<td>starwars</td>
</tr>
<tr>
<td>17</td>
<td>bailey</td>
<td>welcome</td>
<td>monkey</td>
<td>access</td>
<td>master</td>
<td>flower</td>
<td>123123</td>
</tr>
<tr>
<td>18</td>
<td>passwdord</td>
<td>shadow</td>
<td>shadow</td>
<td>shadow</td>
<td>monkey</td>
<td>passwdord</td>
<td>dragon</td>
</tr>
<tr>
<td>19</td>
<td>shadow</td>
<td>ashley</td>
<td>sunshine</td>
<td>master</td>
<td>letmein</td>
<td>dragon</td>
<td>passwdord</td>
</tr>
<tr>
<td>20</td>
<td>123123</td>
<td>football</td>
<td>12345</td>
<td>michael</td>
<td>login</td>
<td>sunshine</td>
<td>master</td>
</tr>
<tr>
<td>21</td>
<td>654321</td>
<td>jesus</td>
<td>password1</td>
<td>superman</td>
<td>princess</td>
<td>master</td>
<td>hello</td>
</tr>
<tr>
<td>22</td>
<td>superman</td>
<td>michael</td>
<td>princess</td>
<td>696969</td>
<td>qwertyuiop</td>
<td>hottie</td>
<td>freedom</td>
</tr>
<tr>
<td>23</td>
<td>qazwsx</td>
<td>ninja</td>
<td>azerty</td>
<td>123123</td>
<td>solo</td>
<td>love</td>
<td>whatever</td>
</tr>
<tr>
<td>24</td>
<td>michael</td>
<td>mustang</td>
<td>trustno1</td>
<td>batman</td>
<td>passwdord</td>
<td>zaq1zaq1</td>
<td>qazwsx</td>
</tr>
<tr>
<td>25</td>
<td>Football</td>
<td>passwdord</td>
<td>000000</td>
<td>trustno1</td>
<td>starwars</td>
<td>passwdord1</td>
<td>trustno1</td>
</tr>
</tbody>
</table>
Through 20 years of effort, we've successfully trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess.
Password File Access Control

can block offline guessing attacks by denying access to encrypted passwords

- make available only to privileged users
- shadow password file
  - a separate file from the user IDs where the hashed passwords are kept

vulnerabilities

- weakness in the OS that allows access to the file
- accident with permissions making it readable
- users with same password on other systems
- access from backup media
- sniff passwords in network traffic
Password Cracking

“John the Ripper” demo
Available on Kali Linux
and other
Linux distributions
Password Selection Techniques

**User Education**

- Users can be told the importance of using hard to guess passwords and can be provided with guidelines for selecting strong passwords.

**Computer Generated Passwords**

- Users have trouble remembering them.

**Reactive Password Checking**

- System periodically runs its own password cracker to find guessable passwords.

**Proactive Password Checking**

- User is allowed to select their own password, however the system checks to see if the password is allowable, and if not, rejects it.
- Goal is to eliminate guessable passwords while allowing the user to select a password that is memorable.
Proactive Password Checking

- **Bloom filter**
  - used to build a table based on dictionary using hashes
  - check desired password against this table

- **password cracker**
  - compile a large dictionary of passwords not to use

- **rule enforcement**
  - specific rules that passwords must adhere to
Bloom Filters

- Probabilistic Data Structure using a bit map. Hash the word n times and place a 1 at bit index for each hash
- By definition, Bloom filter can check for if a value is ‘possibly in the set’ or ‘definitely not in the set’.

Image Credit: GeeksforGeeks
Figure 3.2 Performance of Bloom Filter
How big should the bit map be? How many hashes?

Suppose we have a dictionary of 1 million words and we wish to have a 0.01 probability of rejecting a password not in the dictionary. If we choose six hash functions, the required ratio is $R = 9.6$.

Therefore, we need a hash table of $9.6 \times 10^6$ bits or about $1.2 \text{ Mbytes}$ of storage. In contrast, storage of the entire dictionary would require on the order of $8 \text{ MBytes}$. Thus, we achieve a compression of almost a factor of 7.

Furthermore, password checking involves the straightforward calculation of six hash functions and is independent of the size of the dictionary, whereas with the use of the full dictionary, there is substantial searching.
Other Authentication Methods

Tokens, Smart Cards, Biometrics
Token Authentication

• object user possesses to authenticate, e.g.
  – embossed card
  – magnetic stripe card
  – memory card
  – smartcard
# Types of Cards Used as Tokens

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Defining Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embossed</td>
<td>Raised characters only, on front</td>
<td>Old credit card</td>
</tr>
<tr>
<td>Magnetic stripe</td>
<td>Magnetic bar on back, characters on front</td>
<td>Bank card</td>
</tr>
<tr>
<td>Memory</td>
<td>Electronic memory inside</td>
<td>Prepaid phone card</td>
</tr>
<tr>
<td>Smart Contact</td>
<td>Electronic memory and processor inside</td>
<td>Biometric ID card</td>
</tr>
<tr>
<td>Contactless</td>
<td>Electronic contacts exposed on surface, Radio antenna</td>
<td></td>
</tr>
<tr>
<td></td>
<td>embedded inside</td>
<td></td>
</tr>
</tbody>
</table>
Coinbase Demo with Google Authenticator
Memory Cards

- can store but do not process data
- the most common is the magnetic stripe card
- can include an internal electronic memory
- can be used alone for physical access
  - hotel room
  - ATM
- provides significantly greater security when combined with a password or PIN
- drawbacks of memory cards include:
  - requires a special reader
  - loss of token
  - user dissatisfaction
Smartcard

• credit-card like
• has own processor, memory, I/O ports
  – wired or wireless access by reader
  – may have crypto co-processor
  – ROM, EEPROM, RAM memory
• executes protocol to authenticate with reader/computer
• also have USB dongles
Smart Card Dimensions

The smart card chip is embedded into the plastic card and is not visible. The dimensions conform to ISO standard 7816-2.

Typical chip layout:
- RAM
- EEPROM
- ROM
- CPU
- Crypto coprocessor
Communication Initialization between a Smart Card and a Reader

Figure 3.4 Communication Initialization between a Smart Card and a Reader

Source: Based on [TUNS06].

ATR = Answer to reset
PTS = Protocol type selection
Biometric Authentication

• attempts to authenticate an individual based on unique physical characteristics
• based on pattern recognition
• is technically complex and expensive when compared to passwords and tokens
• physical characteristics used include:
  • facial characteristics
  • fingerprints
  • hand geometry
  • retinal pattern
  • iris
  • signature
  • voice
Common Biometrics

- Fingerprint
- Iris
- Face
- Signature
- Voice Print
Uncommon Biometrics

- DNA
- Gait
- Retina
- Ear
Fingerprints

- Analysis based on discrete features
  - Crossover
  - Island
  - Etc.

- Discrimination power based on combinatorics
  - More matches, more confidence
Fingerprints (II)

- Oldest biometric technology
  - Trained experts / court-approved
  - Automatic data base retrieval

- Advantages
  - Reliable, unique (even identical twins)
  - Inexpensive scanners
  - Cooperative subjects → good fingerprints
  - Non-cooperative subjects → latent prints

- Disadvantages
  - 5% of world population has no usable fingerprints
  - Mask-able (gloves / abrasion)
  - Can be faked
Iris

- Analysis based on discrete features
  - Polar striations
  - Also neoplasms, etc.

- Discrimination power based on combinatorics
  - Similar to fingerprints
  - Infra-red lighting
  - Otherwise dark-eyed people can’t be matched
Iris

• New biometric technology
• Advantages
  – Reliable / unique (even identical twins)
  – Relatively inexpensive scanners
• Disadvantages
  – No human experts (hard to audit)
  – Cooperative subjects with active sensors only
  – Behavior over time is unclear
Uses of Biometrics

- Forensics (post-hoc identity)
  - Non-cooperative subjects
  - Latent / accidental data
  - Identity search
- Verification (security)
  - Cooperative subjects
  - Verify/reject a single identity
- Intelligence / Surveillance
  - Non-cooperative subjects
  - Biometrics at a distance
  - Watch list
Fearless Predictions

• Currently...
  1. Forensics: Fingerprints, DNA
  2. Security: Fingerprint, Signature, Iris, 2D Face
  3. Intelligence: Human face recognition

• In the near future...
  – Forensics: DNA, Fingerprints, Face
  – High-end Security: Iris, 3D Face
  – Low-end security: Fingerprint, 2D Face
  – Intelligence: Face, gait, ear...
Biometric Authentication

- authenticate user based on one of their physical characteristics
Figure 3.5  Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.
Operation of a Biometric System
Biometric Accuracy

- never get identical templates
- problems of false match / false non-match
Biometric Measurement Operating Characteristic Curves

Figure 3.8  Idealized Biometric Measurement Operating Characteristic Curves. Different biometric application types make different trade-offs between the false match rate and the false nonmatch rate. Note that system A is consistently inferior to system B in accuracy performance. [JAIN00]
Biometric Accuracy

- can plot characteristic curve
- pick threshold balancing error rates
Remote User Authentication

• authentication over network more complex
  – problems of eavesdropping, replay
• generally use challenge-response
  – user sends identity
  – host responds with random number
  – user computes $f(r, h(P))$ and sends back
  – host compares value from user with own computed value, if match user authenticated
• protects against a number of attacks
Password Protocol

- **Example of a challenge-response protocol**

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\leftarrow {r, h(), f()}$</td>
<td>random number \ h(), f(), functions</td>
</tr>
<tr>
<td>$P'$ password</td>
<td>$f(r', h(P')) \rightarrow$</td>
<td></td>
</tr>
<tr>
<td>$r'$, return of $r$</td>
<td>$\leftarrow \text{yes/no}$</td>
<td>if $f(r', h(P')) = f(r, h(P(U)))$ then yes else no</td>
</tr>
</tbody>
</table>

(a) Protocol for a password

- **user transmits identity to remote host**
- **host generates a random number (nonce)**
- **nonce is returned to the user**
- **host stores a hash code of the password**
- **function in which the password hash is one of the arguments**
- **use of a random number helps defend against an adversary capturing the user’s transmission**
Token Protocol

- **user transmits identity to the remote host**
- **host returns a random number and identifiers**
- **token either stores a static passcode or generates a one-time random passcode**
- **user activates passcode by entering a password**
- **password is shared between the user and token and does not involve the remote host**

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td>$r$, random number $h()$, $f()$ functions</td>
</tr>
<tr>
<td>$P' \rightarrow W'$ password to passcode via token $r'$, return of $r$</td>
<td>$f(r', h(W')) \rightarrow$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\leftarrow$ yes/no</td>
<td>if $f(r', h(W')) = f(r, h(W(U)))$ then yes else no</td>
</tr>
</tbody>
</table>

(b) Protocol for a token

• **Example of a token protocol**
Static Biometric Protocol

- **user transmits an ID to the host**
- **host responds with a random number and the identifier for an encryption**
- **client system controls biometric device on user side**
- **host decrypts incoming message and compares these to locally stored values**
- **host provides authentication by comparing the incoming device ID to a list of registered devices at the host database**

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td>$r$, random number $E()$, function</td>
</tr>
<tr>
<td></td>
<td>$\leftarrow { r, E() }$</td>
<td></td>
</tr>
<tr>
<td>$B^i \rightarrow BT^i$ biometric device $r^i$, return of $r$</td>
<td>$E(r', D', BT') \rightarrow$ $E^{-1}(E'(r', P', BT') = (r', P', BT')$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\leftarrow$ yes/no</td>
<td>if $r' = r$ and $D' = D$ and $BT' = BT(U)$ then yes else no</td>
</tr>
</tbody>
</table>

(c) Protocol for static biometric

- **Example of a static biometric protocol**
Dynamic Biometric Protocol

- Host provides a random sequence and a random number as a challenge
- Sequence challenge is a sequence of numbers, characters, or words
- User at client end must then vocalize, type, or write the sequence to generate a biometric signal
- The client side encrypts the biometric signal and the random number
- Host decrypts message and generates a comparison

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td>$r$, random number $x$, random sequence challenge $E()$, function</td>
</tr>
<tr>
<td>$B', x' \rightarrow BS'(x')$ $r'$, return of $r$</td>
<td>$E(r', BS'(x')) \rightarrow$</td>
<td>$E^{-1}E(r', BS'(x')) = (r', BS'(x'))$ extract $B'$ from $BS'(x')$</td>
</tr>
<tr>
<td>$\leftarrow$ yes/no</td>
<td>if $r' = r$ and $x' = x$ and $B' = B(U)$ then yes else no</td>
<td></td>
</tr>
</tbody>
</table>

(d) Protocol for dynamic biometric
<table>
<thead>
<tr>
<th>Attacks</th>
<th>Authenticators</th>
<th>Examples</th>
<th>Typical defenses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client attack</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>Guessing, exhaustive search</td>
<td>Large entropy; limited attempts</td>
</tr>
<tr>
<td>Token</td>
<td></td>
<td>Exhaustive search</td>
<td>Large entropy; limited attempts, theft of object requires presence</td>
</tr>
<tr>
<td>Biometric</td>
<td></td>
<td>False match</td>
<td>Large entropy; limited attempts</td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>Plaintext theft, dictionary/exhaustive search</td>
<td>Hashing; large entropy; protection of password database</td>
</tr>
<tr>
<td>Token</td>
<td></td>
<td>Passcode theft</td>
<td>Same as password; 1-time passcode</td>
</tr>
<tr>
<td>Biometric</td>
<td></td>
<td>Template theft</td>
<td>Capture device authentication; challenge response</td>
</tr>
<tr>
<td><strong>Eavesdropping, theft, and copying</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>&quot;Shoulder surfing&quot;</td>
<td>User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication</td>
</tr>
<tr>
<td>Token</td>
<td></td>
<td>Theft, counterfeiting hardware</td>
<td>Multifactor authentication; tamper resistant/evident token</td>
</tr>
<tr>
<td>Biometric</td>
<td></td>
<td>Copying (spoofing) biometric</td>
<td>Copy detection at capture device and capture device authentication</td>
</tr>
<tr>
<td><strong>Replay</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>Replay stolen password response</td>
<td>Challenge-response protocol</td>
</tr>
<tr>
<td>Token</td>
<td></td>
<td>Replay stolen passcode response</td>
<td>Challenge-response protocol; 1-time passcode</td>
</tr>
<tr>
<td>Biometric</td>
<td></td>
<td>Replay stolen biometric template response</td>
<td>Copy detection at capture device and capture device authentication via challenge-response protocol</td>
</tr>
<tr>
<td><strong>Trojan horse</strong></td>
<td>Password, token, biometric</td>
<td>Installation of rogue client or capture device</td>
<td>Authentication of client or capture device within trusted security perimeter</td>
</tr>
<tr>
<td><strong>Denial of service</strong></td>
<td>Password, token, biometric</td>
<td>Lockout by multiple failed authentications</td>
<td>Multifactor with token</td>
</tr>
</tbody>
</table>
Authentication Security Issues

- Client Attacks
- Host Attacks
- Eavesdropping
- Replay
- Trojan Horse
- Denial-Of-Service
**Authentication Security Issues**

- **eavesdropping**: Adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary.
- **host attacks**: Directed at the user file at the host where passwords, token passcodes, or biometric templates are stored.
- **Trojan horse**: An application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric.
- **client attacks**: Adversary attempts to achieve user authentication without access to the remote host or the intervening communications path.
- **replay**: Adversary repeats a previously captured user response.
- **denial-of-service**: Attempts to disable a user authentication service by flooding the service with numerous authentication attempts.
Practical Application: Iris Biometric System

The existing information technology (IT) structure provides capability for remote transactions. It allows access either by PIN or iris biometric (for higher valued transactions).

Figure 3.11 Multichannel System Architecture Used to Link Public- and Personal-use Iris Identification Devices via the Internet. The system uses each customer's PIN (personal identification number), iris code, and CIN (customer identification number) to validate transactions. [NEGI00]
**Figure 3.10b**

**Token Protocol**

- **user transmits identity to the remote host**
- **host returns a random number and identifiers**
- **token either stores a static passcode or generates a one-time random passcode**
- **user activates passcode by entering a password**
- **password is shared between the user and token and does not involve the remote host**

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\leftarrow { \ r, h(), f() }$</td>
<td>$r$, random number $h()$, $f()$, functions</td>
</tr>
<tr>
<td>$P' \rightarrow W'$</td>
<td>$f(r', h(W')) \rightarrow$</td>
<td></td>
</tr>
<tr>
<td>$P' \rightarrow W'$</td>
<td>password to passcode via token $r'$, return of $r$</td>
<td>$\leftarrow \text{yes/no}$</td>
</tr>
<tr>
<td></td>
<td>$\leftarrow \text{yes/no}$</td>
<td>if $f(r', h(W')) = f(r, h(W(U)))$ then yes else no</td>
</tr>
</tbody>
</table>

(b) Protocol for a token

- Example of a token protocol
### Static Biometric Protocol

**Example of a static biometric protocol**

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td>$r$, random number $E()$, function</td>
</tr>
<tr>
<td>$B' \rightarrow BT'$ biometric device $r'$, return of $r$</td>
<td>$E(r', D', BT') \rightarrow$</td>
<td>$E^{-1}E(r', p', BT) = (r', p', BT)$</td>
</tr>
<tr>
<td>$\leftarrow$ yes/no</td>
<td>if $r' = r$ and $D' = D$ and $BT' = BT(U)$ then yes else no</td>
<td></td>
</tr>
</tbody>
</table>

- **user transmits an ID to the host**
- **host responds with a random number and the identifier for an encryption**
- **client system controls biometric device on user side**
- **host decrypts incoming message and compares these to locally stored values**
- **host provides authentication by comparing the incoming device ID to a list of registered devices at the host database**
Dynamic Biometric Protocol

- **Example of a dynamic biometric protocol**

- **host** provides a random sequence and a random number as a challenge

- **sequence challenge** is a sequence of numbers, characters, or words

- **user** at client end must then vocalize, type, or write the sequence to generate a biometric signal

- **the client side** encrypts the biometric signal and the random number

- **host** decrypts message and generates a comparison

<table>
<thead>
<tr>
<th>Client</th>
<th>Transmission</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$, user</td>
<td>$U \rightarrow$</td>
<td>$r$, random number $x$, random sequence challenge $E()$, function $E^{-1}E(r', BS(x')) = (r', BS(x'))$ extract $B'$ from $BS(x')$</td>
</tr>
<tr>
<td>$B'$, $x'$ $\rightarrow BS(x')$</td>
<td>$E(r', BS(x')) \rightarrow$</td>
<td>if $r' = r$ and $x' = x$ and $B' = B(U)$ then yes else no</td>
</tr>
<tr>
<td>$r'$, return of $r$</td>
<td>$\leftarrow { r, x, E() }$</td>
<td>$\leftarrow$ yes/no</td>
</tr>
<tr>
<td>Attacks</td>
<td>Authenticators</td>
<td>Examples</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td><em>Client attack</em></td>
<td>Password</td>
<td>Guessing, exhaustive search</td>
</tr>
<tr>
<td></td>
<td>Token</td>
<td>Exhaustive search</td>
</tr>
<tr>
<td></td>
<td>Biometric</td>
<td>False match</td>
</tr>
<tr>
<td><em>Host attack</em></td>
<td>Password</td>
<td>Plaintext theft, dictionary/exhaustive search</td>
</tr>
<tr>
<td></td>
<td>Token</td>
<td>Passcode theft</td>
</tr>
<tr>
<td></td>
<td>Biometric</td>
<td>Template theft</td>
</tr>
<tr>
<td><em>Eavesdropping, theft, and copying</em></td>
<td>Password</td>
<td>&quot;Shoulder surfing&quot;</td>
</tr>
<tr>
<td></td>
<td>Token</td>
<td>Theft, counterfeiting hardware</td>
</tr>
<tr>
<td></td>
<td>Biometric</td>
<td>Copying (spoofing) biometric</td>
</tr>
<tr>
<td><em>Replay</em></td>
<td>Password</td>
<td>Replay stolen password response</td>
</tr>
<tr>
<td></td>
<td>Token</td>
<td>Replay stolen passcode response</td>
</tr>
<tr>
<td></td>
<td>Biometric</td>
<td>Replay stolen biometric template response</td>
</tr>
<tr>
<td><em>Trojan horse</em></td>
<td>Password, token, biometric</td>
<td>Installation of rogue client or capture device</td>
</tr>
<tr>
<td><em>Denial of service</em></td>
<td>Password, token, biometric</td>
<td>Lockout by multiple failed authentications</td>
</tr>
</tbody>
</table>
Authentication Security Issues

- **eavesdropping**
  adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary

- **host attacks**
  directed at the user file at the host where passwords, token passcodes, or biometric templates are stored

- **denial-of-service**
  attempts to disable a user authentication service by flooding the service with numerous authentication attempts

- **Trojan horse**
  an application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric

- **client attacks**
  adversary attempts to achieve user authentication without access to the remote host or the intervening communications path

- **replay**
  adversary repeats a previously captured user response
Authentication Security Issues

- client attacks
- host attacks
- eavesdropping
- replay
- trojan horse
- denial-of-service
Practical Application
Case Study: ATM Security

(a) Point-to-point connection to processor

Issuer-owned ATM
Issuer (e.g., bank)
Processor (e.g., Fidelity)
EFT exchange e.g., Star, VISA

Issuer-owned ATM
Internet
Issuer (e.g., bank)
Processor (e.g., Fidelity)
EFT exchange e.g., Star, VISA

Issuer’s internal network
Summary

• introduced user authentication
  – using passwords
  – using tokens
  – using biometrics

• remote user authentication issues

• example application and case study
Summary

- four means of authenticating a user’s identity
  - something the individual knows
  - something the individual possesses
  - something the individual is
  - something the individual does
- vulnerability of passwords
  - offline dictionary attack
  - specific account attack
  - popular password attack
  - password guessing against single user
  - workstation hijacking
  - exploiting user mistakes
  - exploiting multiple password use
  - electronic monitoring
- hashed password and salt value
- password file access control
- password selection strategies
  - user education
  - computer generated passwords
  - reactive password checking
  - proactive password checking
- Bloom filter
- token based authentication
  - memory cards
  - smart cards
- biometric authentication
- remote user authentication
  - password protocol
  - token protocol
  - static biometric protocol
  - dynamic biometric protocol