Simple Attack on Crypto Systems

Man-in-the-Middle Attacks

- The attacker modifies the ciphertext to $C'$
- When the receiver decrypts the $C'$ he obtains $M'$ which is different from the message $M$

Figure 1.13: A man-in-the-middle attack where the adversary modifies the ciphertext and the recipient decrypts the altered ciphertext into an incorrect message.
Simple Attack on Crypto Systems

Man-in-the-middle attack

- The attacker modifies the digital signature $S$ into $S'$ together with the encryption of $M'$ using the recipient public key.
- When the recipient tries to verify the signature $S'$, he obtains the message $M'$ by encrypting $S'$.

![Diagram of man-in-the-middle attack](image)

**Figure 1.13:** A man-in-the-middle attack where the adversary modifies the ciphertext and the recipient decrypts the altered ciphertext into an incorrect message.
Simple Attack on Crypto Systems

• Brute-Force Decryption Attack
  – Suppose we have a message of length n-bits
  – Using a t-byte array to hold the message M
  – Using 8-bit ASCII encoding, we have a total of number of possible plaintext t-byte array \( (2^8)^t = 2^n \)
  – Each character in English hold 1.25 bit of information
    • The t-byte array correspond to English text is \( (2^{1.25})^t \) and in term of n \( 2^{0.16n} \)

Figure 1.14: Natural-language plaintexts are a very small fraction of the set of possible plaintexts. This fraction tends to zero as the plaintext length grows. Thus, for a given key, it is hard for an adversary to guess a ciphertext that corresponds to a valid message.
Simple Attack on Crypto Systems

• **Brute-Force Decryption Attack**
  – In a natural language that uses an alphabet instead of ideograms, there is a constant $\alpha$, with $0 < \alpha < 1$ such that there are $2^{\alpha n}$ texts among all $n$-bits array.
  – The fraction of valid messages out of all possible $n$-bit plaintexts is about
    \[
    \frac{2^{\alpha n}}{2^n} = \frac{1}{2^{(1-\alpha)n}}
    \]
  – Adversary tries all possible decryption keys.
  – Using K-bit as the decryption key.
  – There are $2^k$ possible plaintext.
  – Each plaintext has a probability of
    \[
    \frac{1}{2^{(1-\alpha)n}}
    \]
    being a valid message.
Simple Attack on Crypto Systems

• Brute-Force Decryption Attack
  – Adversary tries all possible decryption keys
  – Using K-bit as the decryption key
  – There are $2^k$ possible plaintexts
  – Each plaintext has a probability of $\frac{1}{2^{(1-\infty)n}}$ being a valid text message
  – The expected number of plaintexts corresponding to valid text message is

\[
\frac{2^k}{2^{(1-\infty)n}}
\]
Message Authentication Codes

- Allows for Alice and Bob to have data integrity, if they share a secret key.

- Given a message M, Alice computes $H(K || M)$ and sends M and this hash to Bob.

\[ \text{Sender} \quad \text{Attacker} \quad \text{Recipient} \]

Communication channel

\[ \text{message M} \xrightarrow{h} 6B34339 \quad \text{MAC} \xrightarrow{4C66809} \quad \text{MAC} \xrightarrow{(\text{attack detected})} 87F9024 \quad \text{computed MAC} \xrightarrow{\text{shared secret key}} \quad \text{message M'} \]
Digital Certificates

- **Certificate Authority (CA)** digitally signs a binding between an identity and the public key for that identity.
Passwords

• A short sequence of characters used as a means to authenticate someone via a secret that they know.

• Userid: __________________

• Password: ________________
How a password is stored?

User

Dog124

hash function

Password file

Butch: ASDSA 21QW3R50E ERWWER323 ...
...
Strong Passwords

What is a strong password

- UPPER/lower case characters
- Special characters
- Numbers

When is a password strong?

- Seattle1
- M1ke03
- P@$w0rd
- TD2k5secV
Password Complexity

A fixed 6 symbols password:

Numbers
\[10^6 = 1,000,000\]

UPPER or lower case characters
\[26^6 = 308,915,776\]

UPPER and lower case characters
\[52^6 = 19,770,609,664\]

32 special characters (\&, %, $, £, “, |, ^, §, etc.)
\[32^6 = 1,073,741,824\]

94 practical symbols available
\[94^6 = 689,869,781,056\]

ASCII standard 7 bit \(2^7 = 128\) symbols
\[128^6 = 4,398,046,511,104\]

Odd characters make passwords safer
Password Length

26 UPPER/lower case characters = 52 characters
10 numbers
32 special characters
=> 94 characters available

5 characters: $94^5 = 7,339,040,224$
6 characters: $94^6 = 689,869,781,056$
7 characters: $94^7 = 64,847,759,419,264$
8 characters: $94^8 = 6,095,689,385,410,816$
9 characters: $94^9 = 572,994,802,228,616,704$

Longer passwords are better
Password Validity: Brute Force Test

Password does not change for 60 days
how many passwords should I try for each second?

5 characters: 1,415 PW /sec
6 characters: 133,076 PW /sec
7 characters: 12,509,214 PW /sec
8 characters: 1,175,866,008 PW /sec
9 characters: 110,531,404,750 PW /sec
Secure Passwords

A strong password includes characters from at least three of the following groups:

<table>
<thead>
<tr>
<th>Group</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase letters</td>
<td>a, b, c, ...</td>
</tr>
<tr>
<td>Uppercase letters</td>
<td>A, B, C, ...</td>
</tr>
<tr>
<td>Numerals</td>
<td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</td>
</tr>
<tr>
<td>Non-alphanumeric (symbols)</td>
<td>( ) ` ~ ! @ # $ % ^ &amp; * - + =</td>
</tr>
<tr>
<td>Unicode characters</td>
<td>€, Γ, f, and λ</td>
</tr>
</tbody>
</table>

Use pass phrases eg. "I re@lly want to buy 11 Dogs!"
Social Engineering

• **Pretexting**: creating a story that convinces an administrator or operator into revealing secret information.

• **Baiting**: offering a kind of “gift” to get a user or agent to perform an insecure action.

• **Quid pro quo**: offering an action or service and then expecting something in return.