Timing Attacks

Side Channel Attacks
-Timing
-Power
-Source
-Light

Usability and Security
-Depending on Users
-Role Playing

Modular Exponentiation
Modexp(m,e,n) // m^e mod n
let e_ℓ, …, e_0 // the bits of e

Square and Multiply
let acc = 1
for i=0 … ℓ
    if e_i==1
        acc=acc*m mod n
    m=m^2 mod n
return acc

m^e mod n = m ^ ( 2^0e_0 + 2^1e_1 + 2^2e_2 + … 2^ℓ e_ℓ ) mod n
          = m ^ e_0 * (m^2)^e_1 * (m^4)^e_2 *** (m^2^ℓ)^e_ℓ mod n

Power Analysis (Smart Cards)
Timing Attack against Square and Multiply

Some reductions take longer than others
- fast reductions
- slow reductions
- If attacker knows \( acc \cdot m \), he can predict whether reduction is fast or slow.

**Attacker**

<table>
<thead>
<tr>
<th>input</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_0 )</td>
<td>( t_0 )</td>
</tr>
<tr>
<td>( m_1 )</td>
<td>( t_1 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( t_2 )</td>
</tr>
</tbody>
</table>
Suppose \( e = 1 \)
-can compute \( \text{acc} \) and match end of round 0
-Predict whether round 1 will be fast or slow for each message \( m_i \).

\[
\text{let } f_1 = \text{average time of “fast” messages} \]
\[
s_1 = \text{average time of “slow” messages} \]

Two Cases
1. \(|s_1-f_1|\) is small and \(|s_0-f_0|\) is small => \( e_1 = 0 \)
2. Suppose \(|s_b-f_b|\) is large and \(|s_b-f_b|\) is small => \( e_1 = 1 \) and \( e_0 = b \)

Repeat for rounds 2 ... \( \ell \)

**Safeguard against Timing Attacks**
-modify the code

\[
\text{let acc} = 1 \\
\text{for } i = 0 \ldots \ell \\
\text{if } e_i == 1 \\
\quad \text{acc} = \text{acc} \times m \mod n \\
\text{else} \\
\quad \text{tmp} \times m \mod n \\
\text{m} = \text{m}^2 \mod n \\
\text{return acc} \\
\]

**RSA Blinding**
To compute \( m^e \mod n \)
1. pick random \( r \)
2. compute \( x = (r \times m)^e \mod n \)
3. compute \( y = r^e \mod n \)
4. return \( x/y \mod n \)

**Light Attacks**
Can determine what is displayed on the screen by recording the brightness change since the monitor updates one pixel at a time. To prevent such an attack would be to update all pixels at once.