

Side Channel Attacks

Brumley/Boneh

- Work againsts CRT-optimized modular exponent algorithm
- Montgomery multiplication optimization
- Sliding Windows
- Karatsuba multiplication $n^{1.7}$
 - Over campus network
 - E.g. Coloced servers

Counter Measures

1. RSA blinding

- Server does not want to leak computation time
- To blind, pick a random number r and compute:
 - $x = rm \bmod N$
 - $r^d m^d = x^d \bmod N$
 - $m^d = x^d / r^d \bmod N$
 - RSA blinding requires additional computation of: 1 multiplication, 2 exponents, and 1 divide
 - **Benefit:** Because attacker does not know the random number thus no timing attack will work.
 - **Cost:** work of 3 exponentiation (assuming multiplication are cheap and division are equivalent of exponentiation)

2. Fixed time delay for response

- **Benefit:** timing attack will not work
- **Cost:** Waste CPU cycles

Local Side Channel Attacks

- CPU usage/temperature
- Pagefault
 - RSA involves two main operations: multiplication and exponentiation
 - If multiplication generates a lot of pagefaults, we can observe the number of pagefaults
- Cache lines
 - If the cache is indexed by physical address, then the cache does not need to be flushed during context switch.
 - If we know RSA use a specific cache line index for multiplication, attacker set to index with specific value and allow the RSA to run for one iteration. Attacker load another

value and compare the time it takes to load. If fast, the cache is not flushed. If slow, the cache is flushed and a multiplication occurred.

Cache

Physical Address	value

- Branch prediction registers
 - BPR stores all the address of the branch target.
 - In RSA multiplication, if the last bit of the key = 1, branch occurs.

CRT Monitor Attack

In an office with windows, often the CRT monitors are arranged such that it is not facing the window. However, an attack is still possible simply by observing the light reflection off the wall.

Idea: elections are excited and lit up fast and fade away slowly.

Keystroke Sound

Each key on keyboard generate different sound, building model of keyboard sound.

Typing Habit

- Measure inter-keystroke time
 - Hand 0.1 sec
 - Nothing 0.01 sec
 - Finger 0.2 sec
- In SSH, you see packets go by assuming each keystroke = separate packet observe now close each packet get set.
- Two letter model for more precise estimate
- Generate table of diagram frequencies
 - Build a Hidden Markup Model