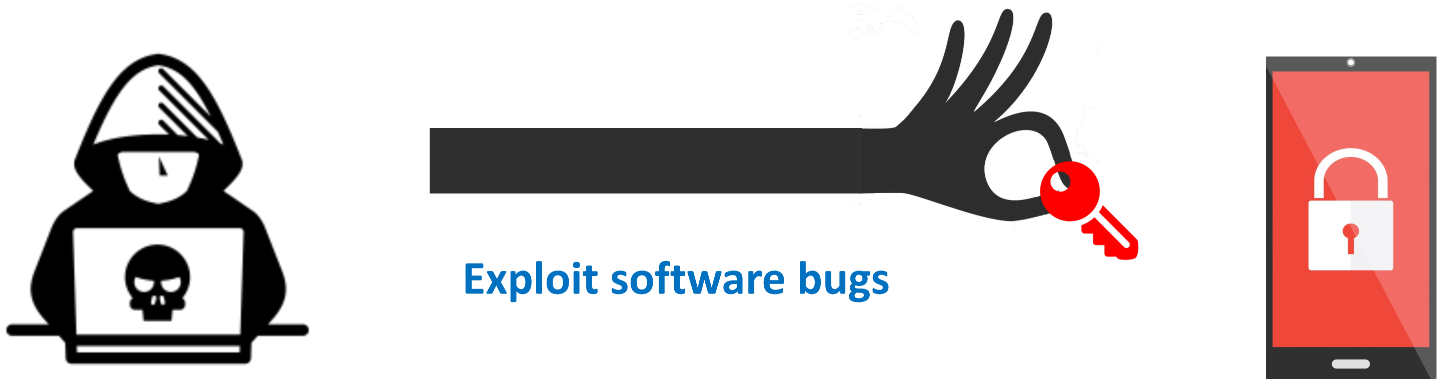
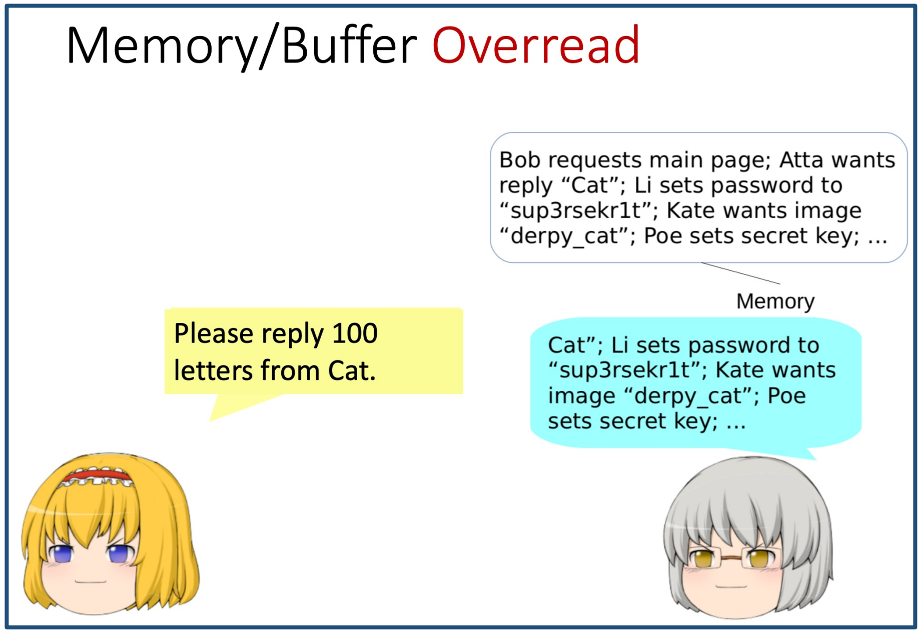
Side Channel Attack

Shuai Wang

香 港 科 技 大 學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

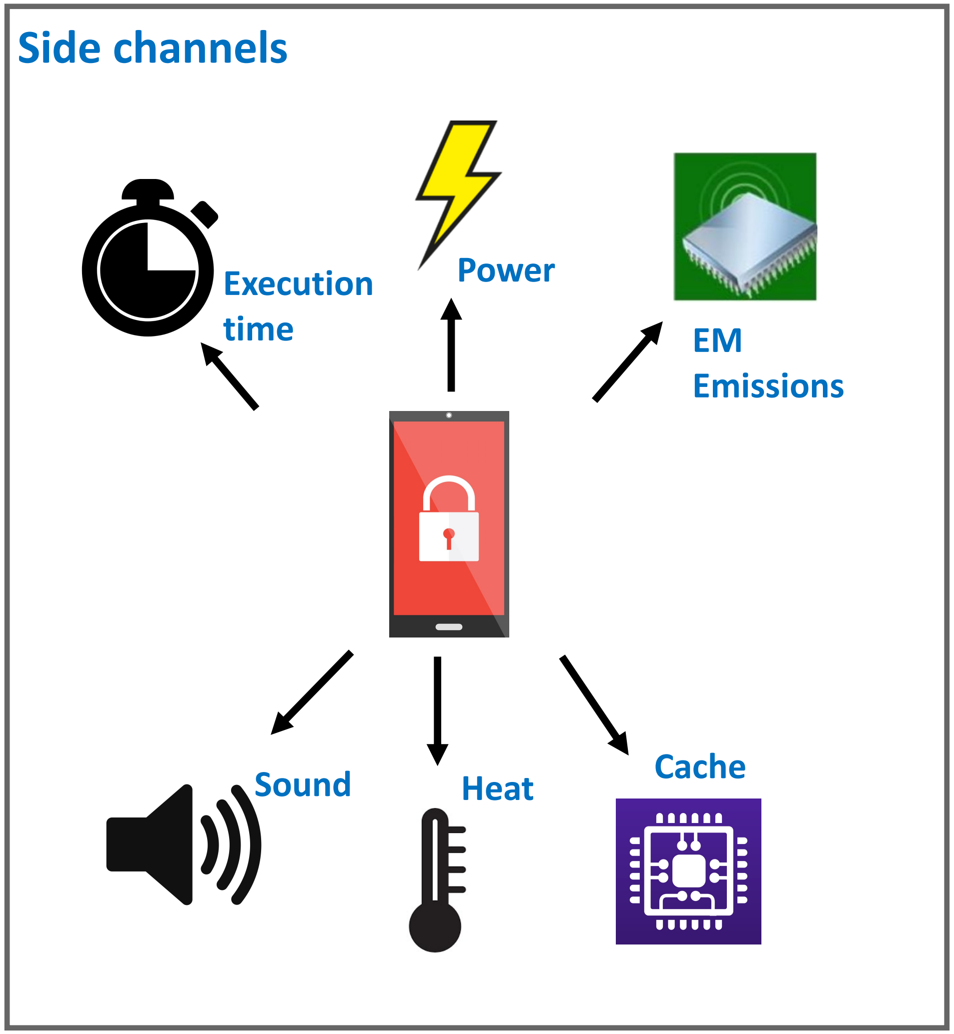
Steal Secrets from Software Systems




Memory/Buffer Overread
Bob requests main page; Atta wants reply “Cat”; Li sets password to “sup3rsekr1t”; Kate wants image “derpy_cat”; Poe sets secret key; …
Memory
Please reply 100 letters from Cat.
Cat”; Li sets password to “sup3rsekr1t”; Kate wants image “derpy_cat”; Poe sets secret key; …

Steal Secrets from Software Systems

Steal Secrets through Side Channels



Infer secrets via secret-dependent physical information.

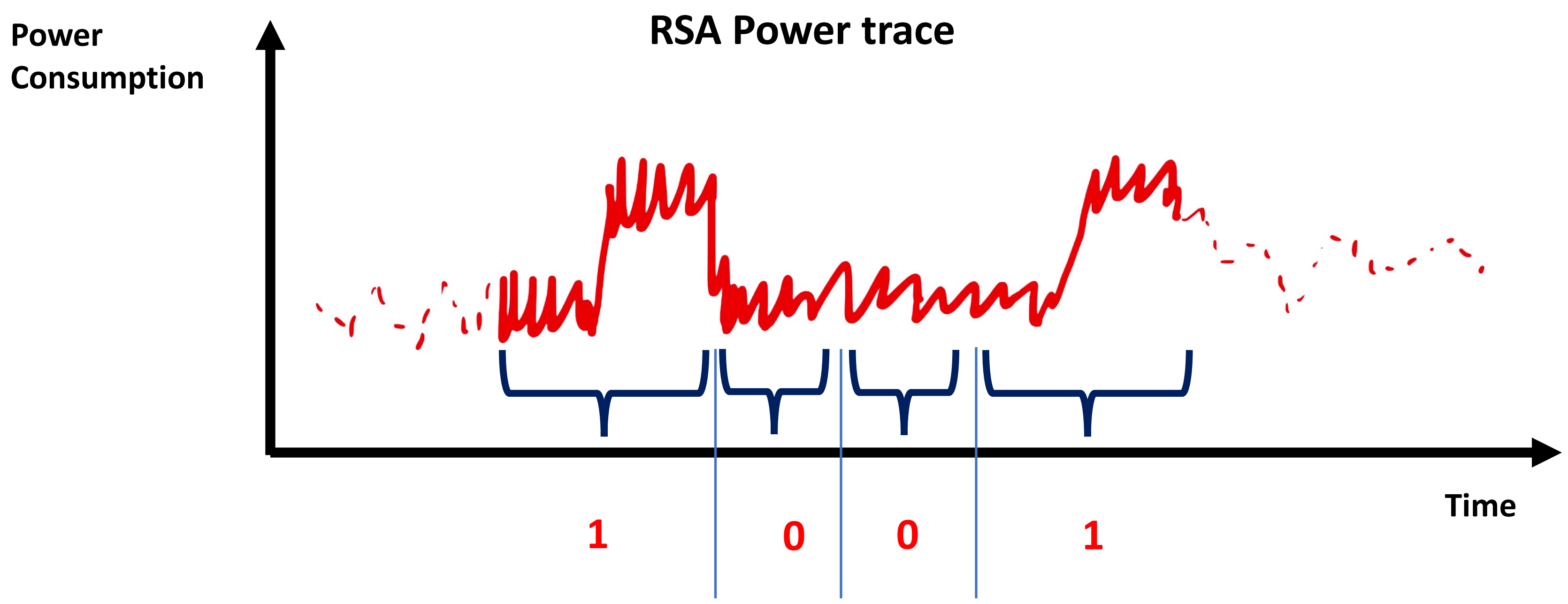
Side Channel

- **Three aspects** to form a side channel attack.
  - Secret dependent program information flow
  - Information flow affects physical environment
  - Physical environment is exploitable by adversarial.
    - **Exploitation** → not our major focus today; could be (more than) one lecture..
    - Just some high-level ideas how on it can be exploited…

Timing Side Channel

```
if (k is 1) then
   // slow branch
else
   // fast branch
```

Break RSA with Power Side Channel

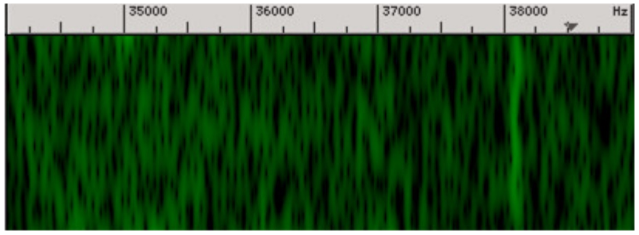


1/0 is different bits in the RSA private key. ← see explanations later.

Break RSA with Sound Side Channel

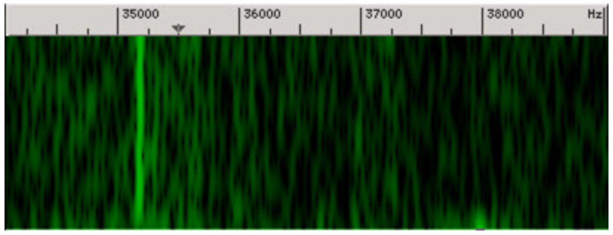
Bzzzzzzzzz

Sound spectrogram



key bit == 0

Sound spectrogram



key bit == 1

https://www.tau.ac.il/~tromer/acoustic/

Side Channel

- **Three aspects** to form a side channel attack.
    - Secret dependent program information flow
    - Information flow affects physical environment
    - Physical environment is exploitable by adversarial.

Let’s see how it works to exploit RSA.

Timing Side Channel Attack on RSA

The Scenario of RSA in Timing Channel Attack

- Alice’s public key: (N,e)
- Alice’s private key: d
- Attacker wants to find d
- Attacker can send any message M to Alice and Alice will respond with $M^d \bmod N$
  - That is, Alice signs M and sends result to the attacker
- **Threat model**: the attacker can precisely time Alice’s computation of $M^d \bmod N$

The standard RSA computation is too slow

Square & multiply: a popular optimization of the RSA implementation.

- Consider M^d mod N
- Square & multiply is used for M^d mod N
- And the implementation of mod:

```
mod(x,N)
if x >= N
    x = x % N
end if
return x
```

Decryption with Square & Multiply

```
for bit in k
    x = mod(x^2,N)
    if (bit is 1) then
        x = mod(x*M,N)
endfor
```

Timing Attack

- If bit = 0 then
  - $x = \mathrm{mod}(x^2, N)$
- If bit = 1 then
  - $x = \mathrm{mod}(x^2, N)$
  - $x = \mathrm{mod}(x*M, N)$
- Computation time differs in each case
  - What is the implication?
    - Exploitable!

Decryption with Square & Multiply

```
for bit in k
   x = mod($x^2$,N)
   if (bit is 1) then
      x = mod(x*M,N)
endfor
```

bit is 0: square	**fast**
bit is 1: square + multiply	**slow**

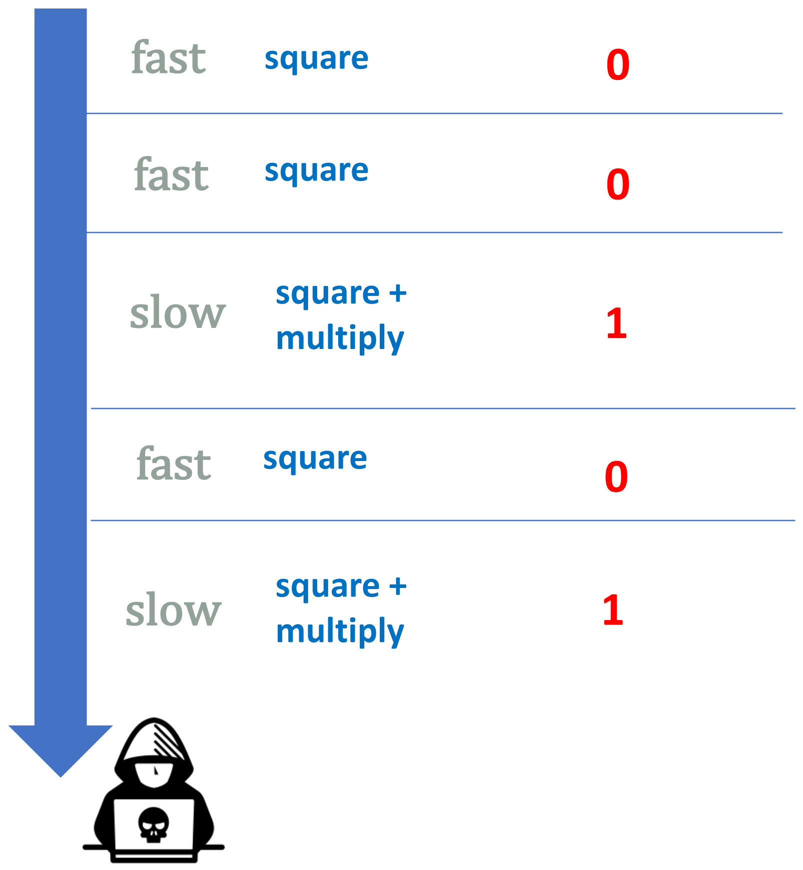
Break RSA with Timing Side Channel

Decryption with Square & Multiply

```
for bit in k
    x = mod(x^2,N)
    if (bit is 1) then
        x = mod(x*M,N)
endfor
```

bit is 0: square	**fast**
bit is 1: square + multiply	**slow**

RSA Timing trace



Obtaining such a timing trace typically requires a considerable number of trials (but don't need factor large numbers which is good). Your reading materials today.

Cache side channel attack and How it works to exploit RSA

```
if (k is 1) then
   // branch 1
else
   // branch 2
```

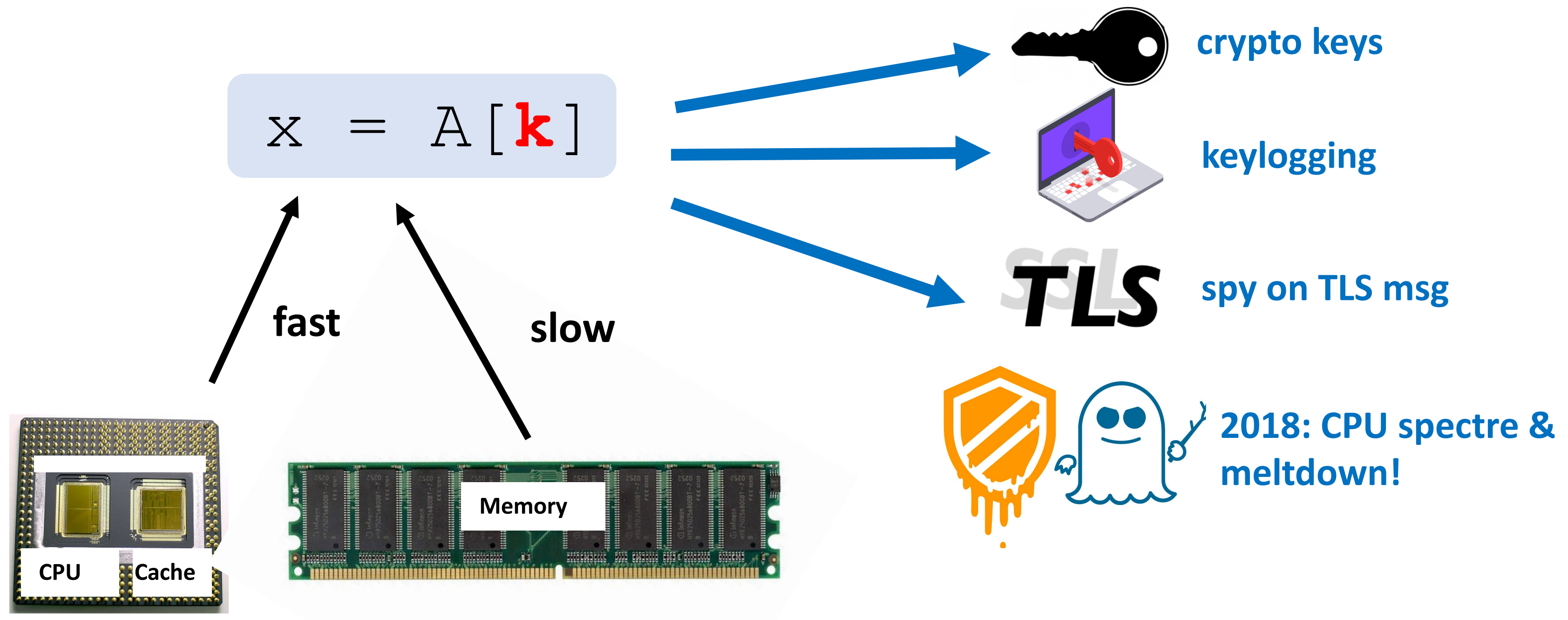
$\rightarrow$

```
x = A[k]
```

Side channel attack can be more subtle…

Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. 1996

Cache-Based Side Channel

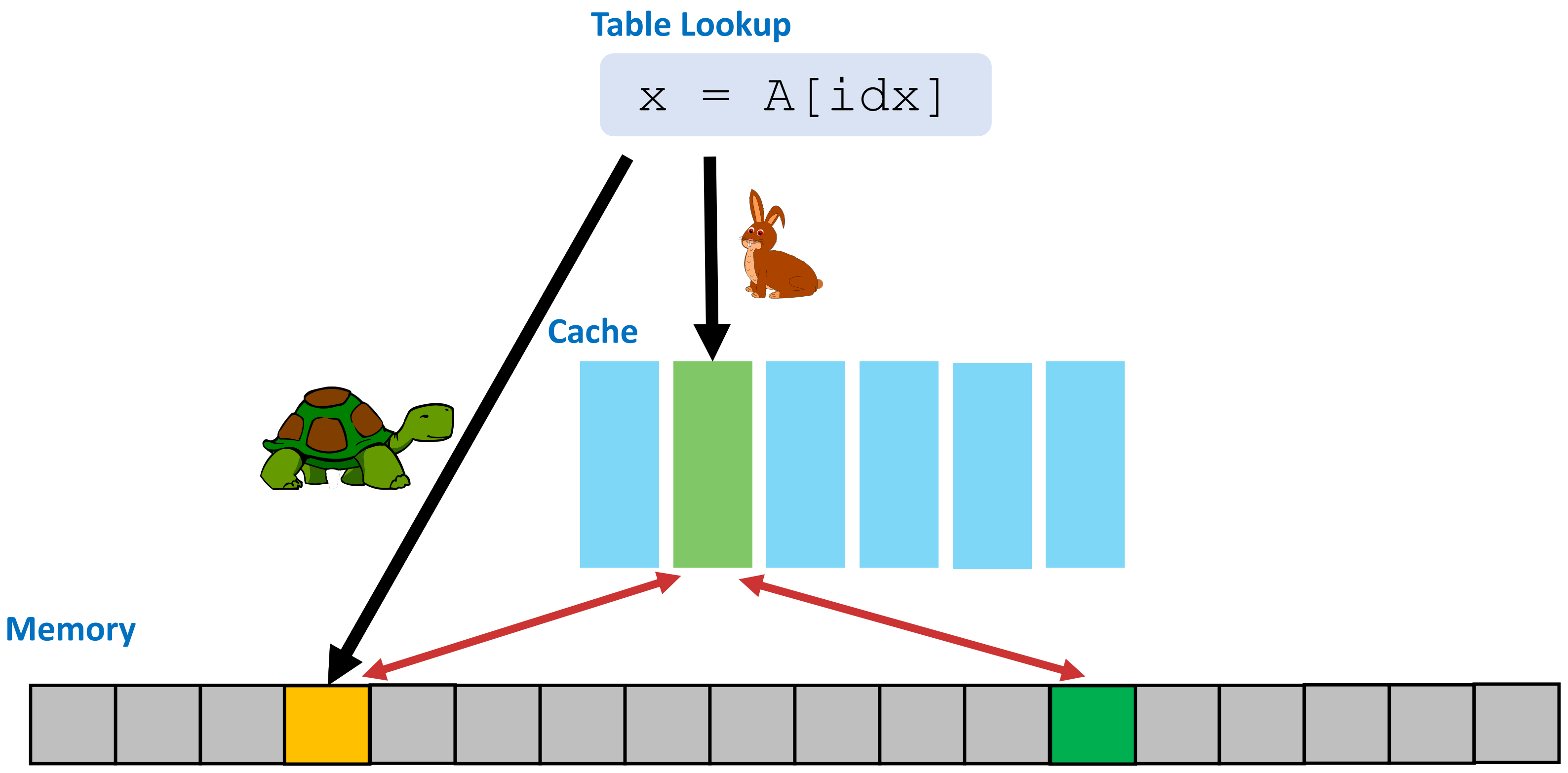


Fundamental principle of cache behavior:

- Accessing data stored in the cache is measurably faster than accessing the data from main memory

[Aciicmez et al., CT-RSA '07], [Osvik et al., CT-RSA '06], [Gullasch, IEEE S&P '11] [Zhang et al., CCS '12] [Yarom et al., USENIX Sec. '14] [Liu et al., IEEE S&P '15] [Yarom et al., CHES '14] [Yarom et al., CHES '16] [Disselkoen et al., USENIX Sec. '17] ...

Cache Basics



WARNING: Simplified

Cache Basics

- **Cache lines**: minimal storage units of a cache

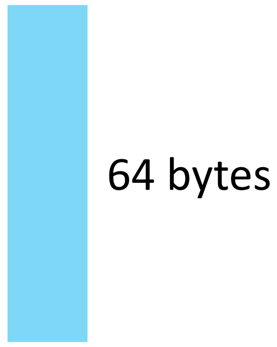
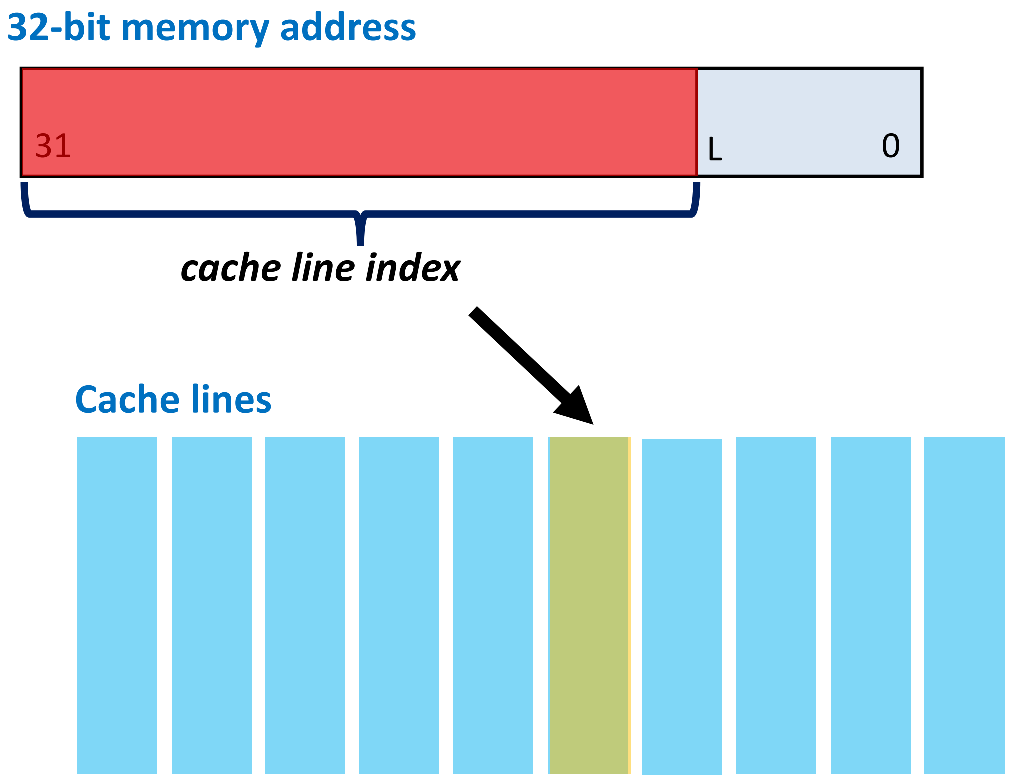


Table Lookup

```
x = A[idx]
```



WARNING: Simplified

Flush & Reload

It's possible to indirectly infer which cache unit is accessed by a victim program.

- Here cache unit can be cache line, cache bank, etc.

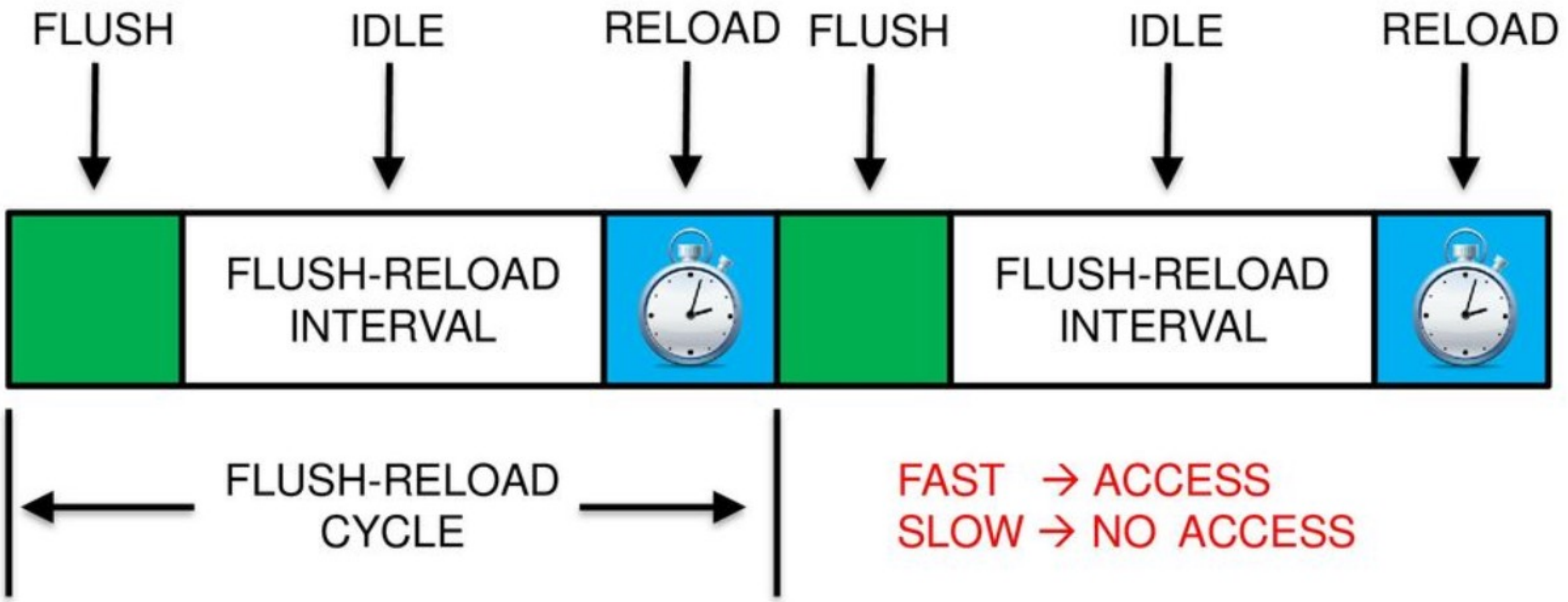
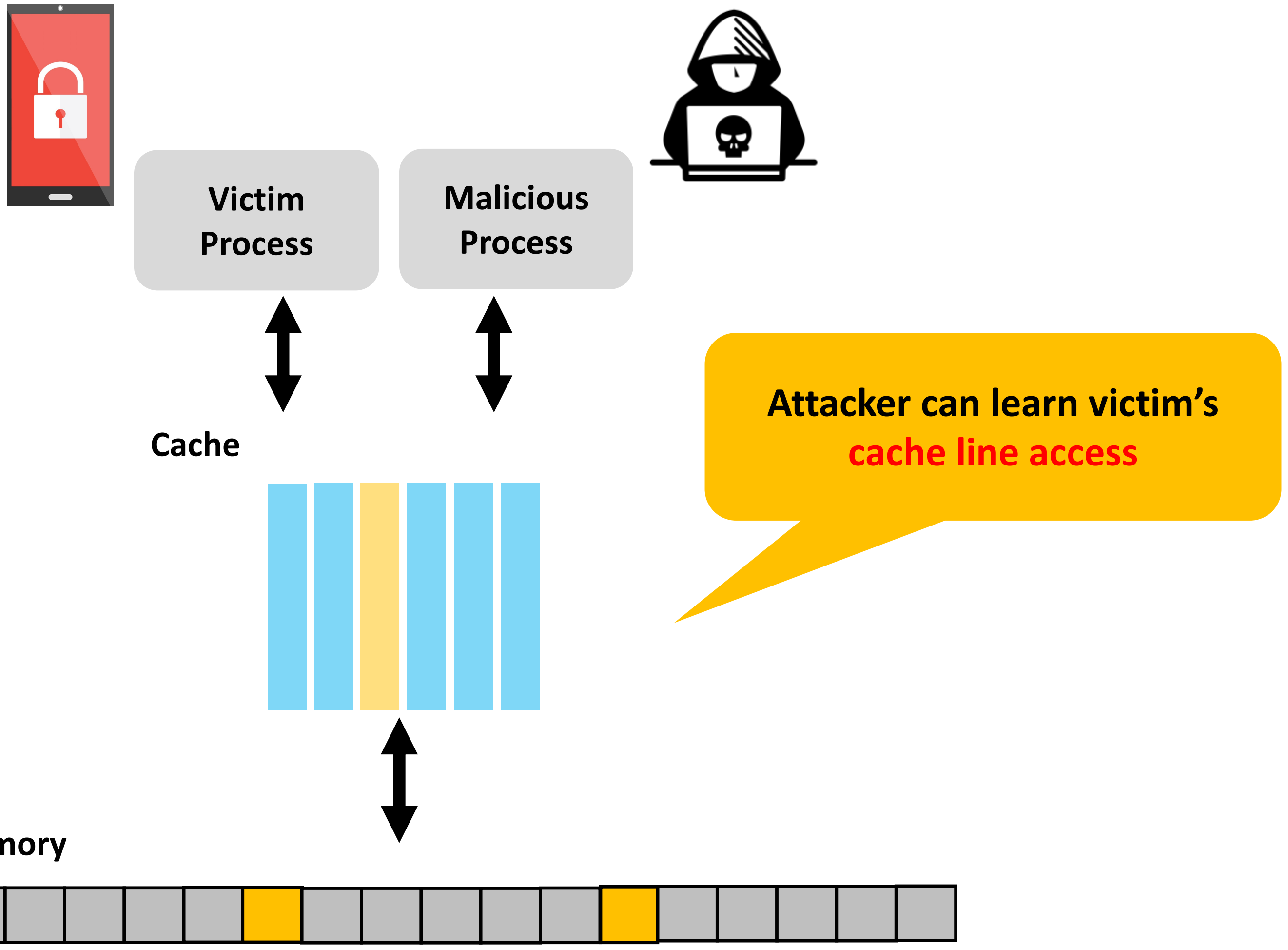
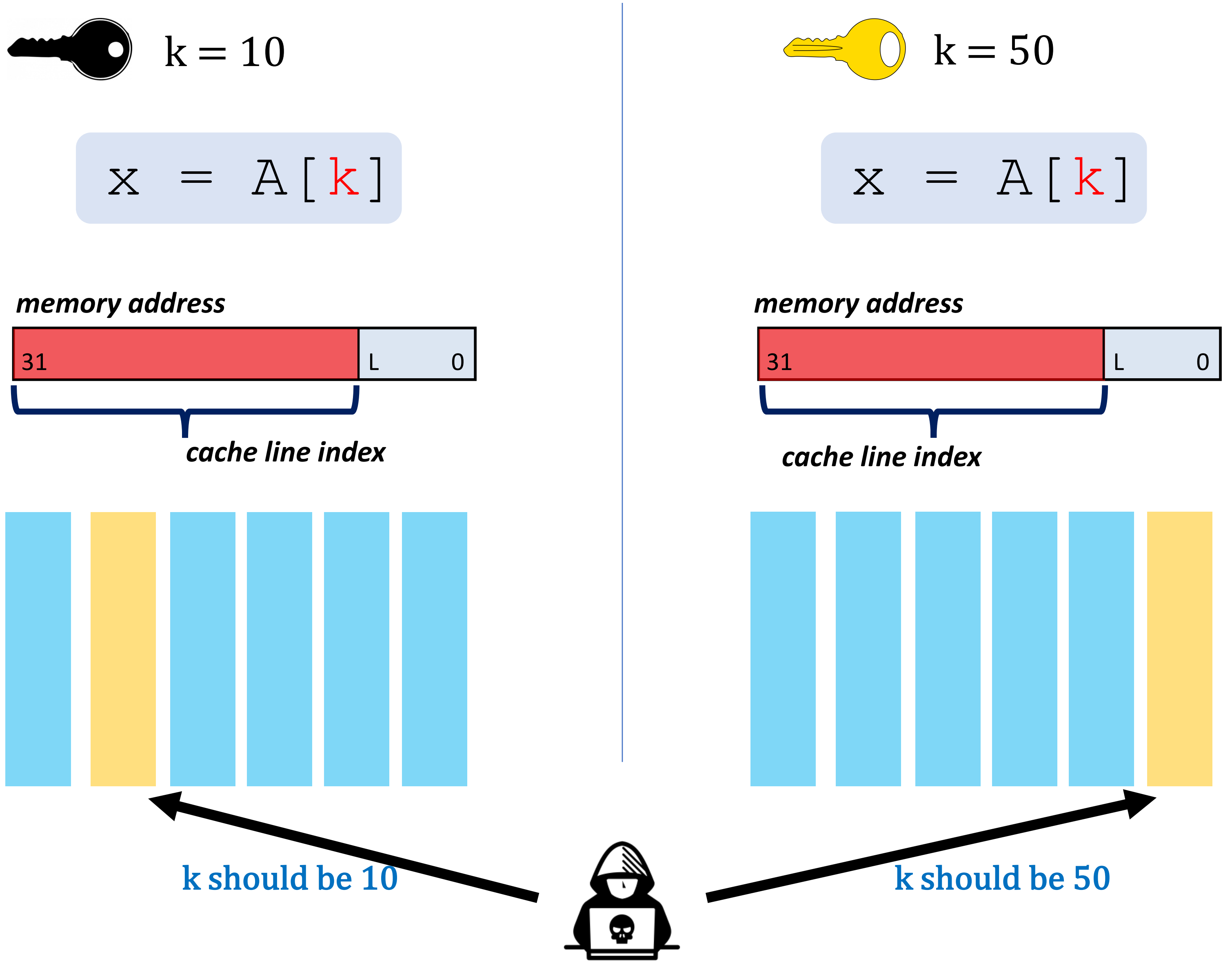


Figure from: *Return-Oriented Flush-Reload Side Channels on ARM and Their Implications for Android Devices*

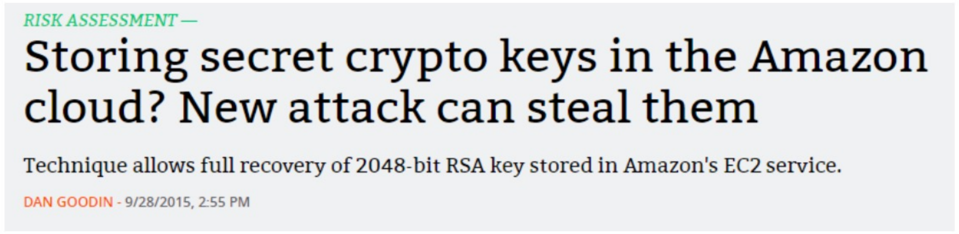
Threat Model



Cache-Based Side Channel Attack



Cache side channels are very powerful and practical, especially in the era of cloud computing

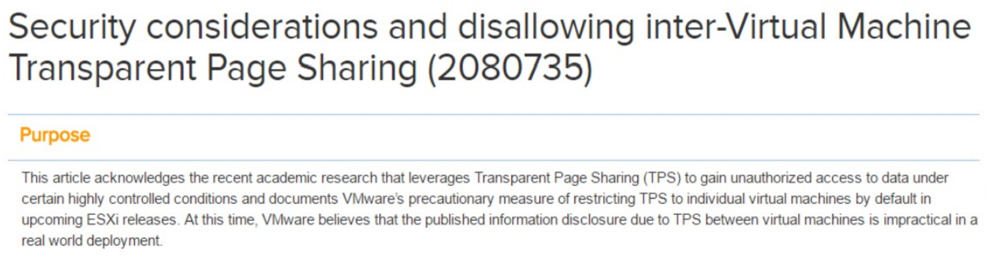

RISK ASSESSMENT —

Storing secret crypto keys in the Amazon cloud? New attack can steal them

Technique allows full recovery of 2048-bit RSA key stored in Amazon's EC2 service.

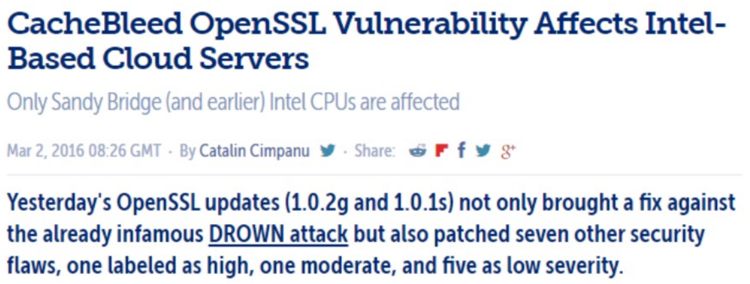
DAN GOODIN - 9/28/2015, 2:55 PM

"You must be kidding, cache attacks are not practical!"


Security considerations and disallowing inter-Virtual Machine Transparent Page Sharing (2080735)

Purpose

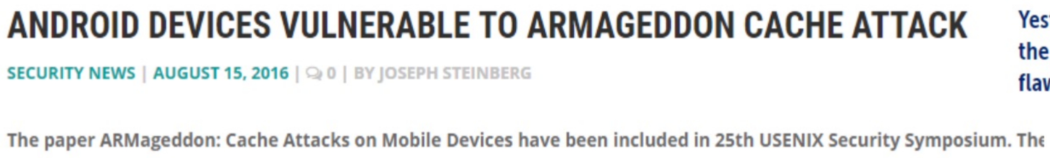
This article acknowledges the recent academic research that leverages Transparent Page Sharing (TPS) to gain unauthorized access to data under certain highly controlled conditions and documents VMware's precautionary measure of restricting TPS to individual virtual machines by default in upcoming ESXi releases. At this time, VMware believes that the published information disclosure due to TPS between virtual machines is impractical in a real world deployment.


CacheBleed OpenSSL Vulnerability Affects Intel-Based Cloud Servers

Only Sandy Bridge (and earlier) Intel CPUs are affected

Mar 2, 2016 08:26 GMT · By Catalin Cimpanu · Share:

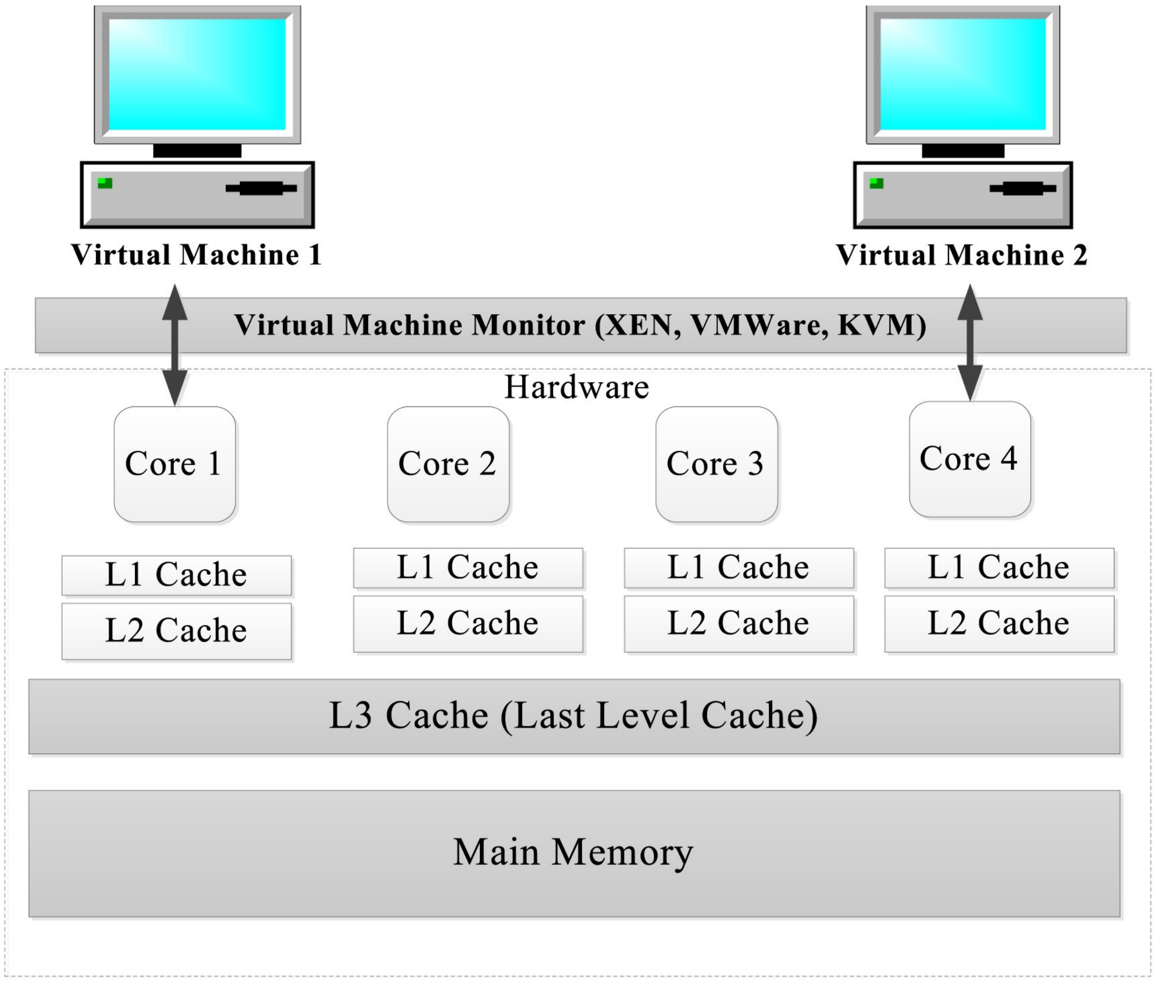
Yesterday's OpenSSL updates (1.0.2g and 1.0.1s) not only brought a fix against the already infamous DROWN attack but also patched seven other security flaws, one labeled as high, one moderate, and five as low severity.


ANDROID DEVICES VULNERABLE TO ARMAGEDDON CACHE ATTACK

SECURITY NEWS | AUGUST 15, 2016 | 0 | BY JOSEPH STEINBERG

The paper ARMageddon: Cache Attacks on Mobile Devices have been included in 25th USENIX Security Symposium. The

Why?



Because different VM instances share the same cache!

- VM instances can run either on the same core (share L1/L2 cache), or cross cores (share L3 cache)

Optimization of RSA Decryption

```
r = 1;
for i from n−1 to 0 {
    r = r∗r mod p;
    if (k[i] == 1) {
        r = r∗b mod p;
    }
}
```

square + multiply-based modular exponentiation.

small window size of key

```
r = 1;
for i from 0 to (n-1)/W {
    r = r * r mod p;
    idx = window_ith(key, i);
    if (idx != 0) {
        t = table[idx];
        r = r * t mod p;
    }
}
```

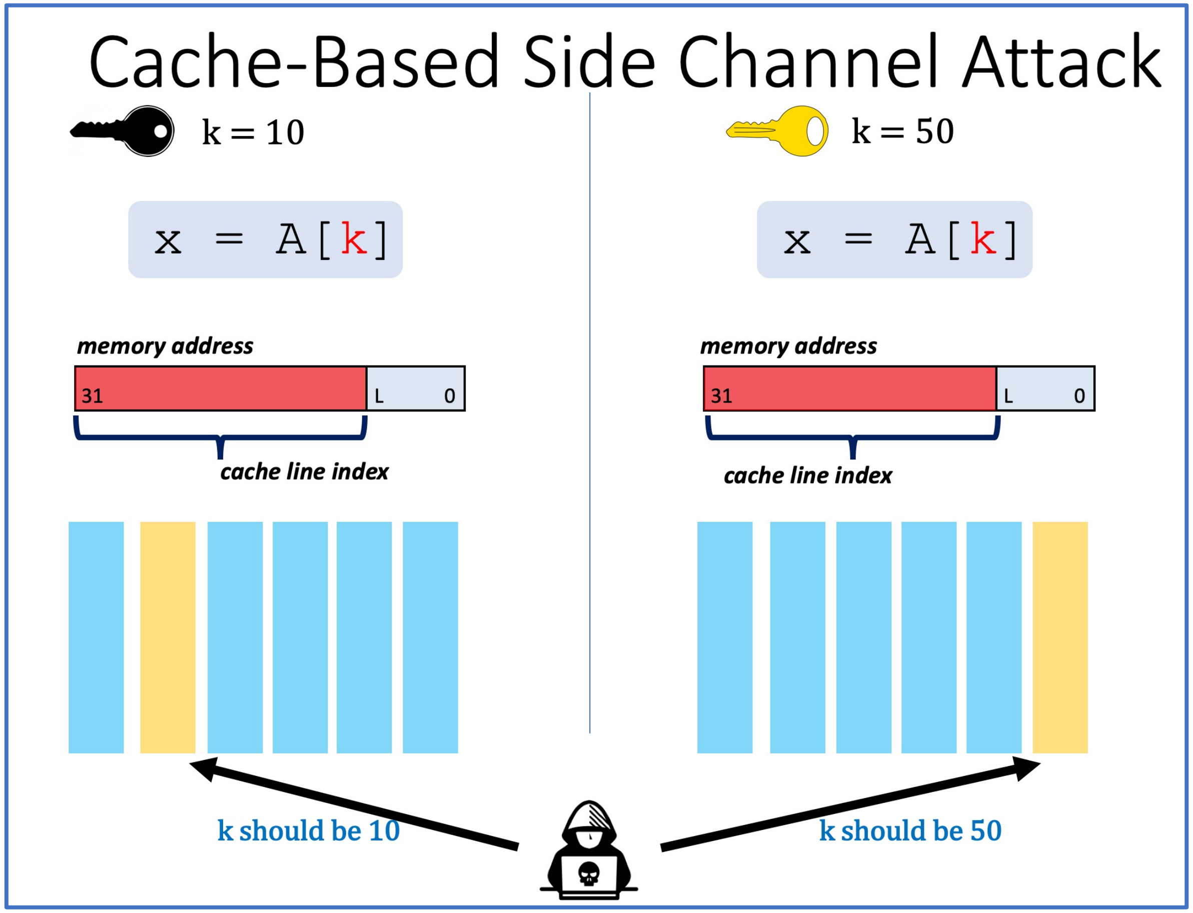
secret key

sliding-window-based modular exponentiation.

Leakage due to secret-dependent control flow

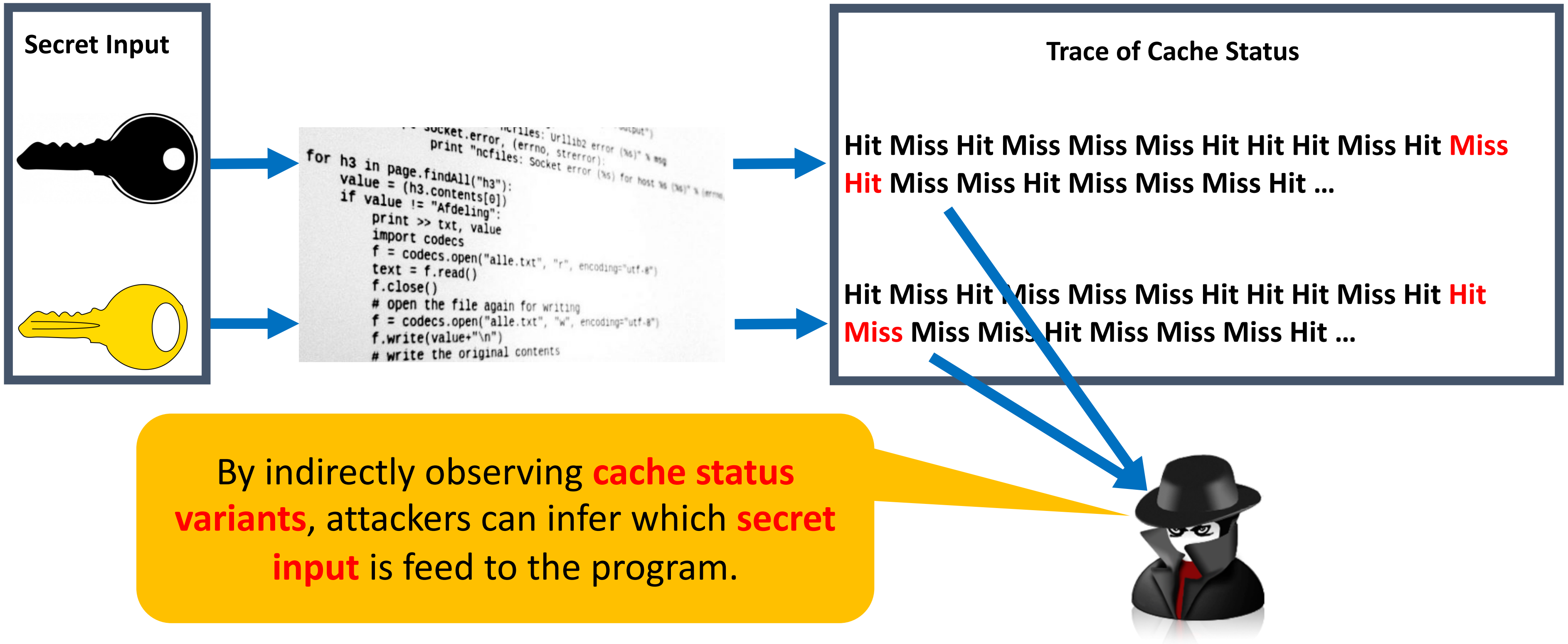
Leakage due to secret-dependent memory access

Cache-based Side Channel Attack: A Strong Threat Model



A powerful attacker can observe the accessed cache line.

Cache-based Side Channel Attack: A Weaker Threat Model



```
                                ncfiles: Urllib2 error (%s)" % msg
           socket.error, (errno, strerror):
                print "ncfiles: Socket error (%s) for host %s (%s)" % (errno,
for h3 in page.findAll("h3"):
    value = (h3.contents[0])
    if value != "Afdeling":
        print >> txt, value
        import codecs
        f = codecs.open("alle.txt", "r", encoding="utf-8")
        text = f.read()
        f.close()
        # open the file again for writing
        f = codecs.open("alle.txt", "w", encoding="utf-8")
        f.write(value+"\n")
        # write the original contents
```

Hit Miss Hit Miss Miss Miss Hit Hit Hit Miss Hit Miss Hit Miss Miss Hit Miss Miss Miss Hit …

Hit Miss Hit Miss Miss Miss Hit Hit Hit Miss Hit Hit Miss Miss Miss Hit Miss Miss Miss Hit …

By indirectly observing **cache status variants**, attackers can infer which **secret input** is feed to the program.

A less powerful attacker can observe hit/miss info instead of a particularly accessed cache line.

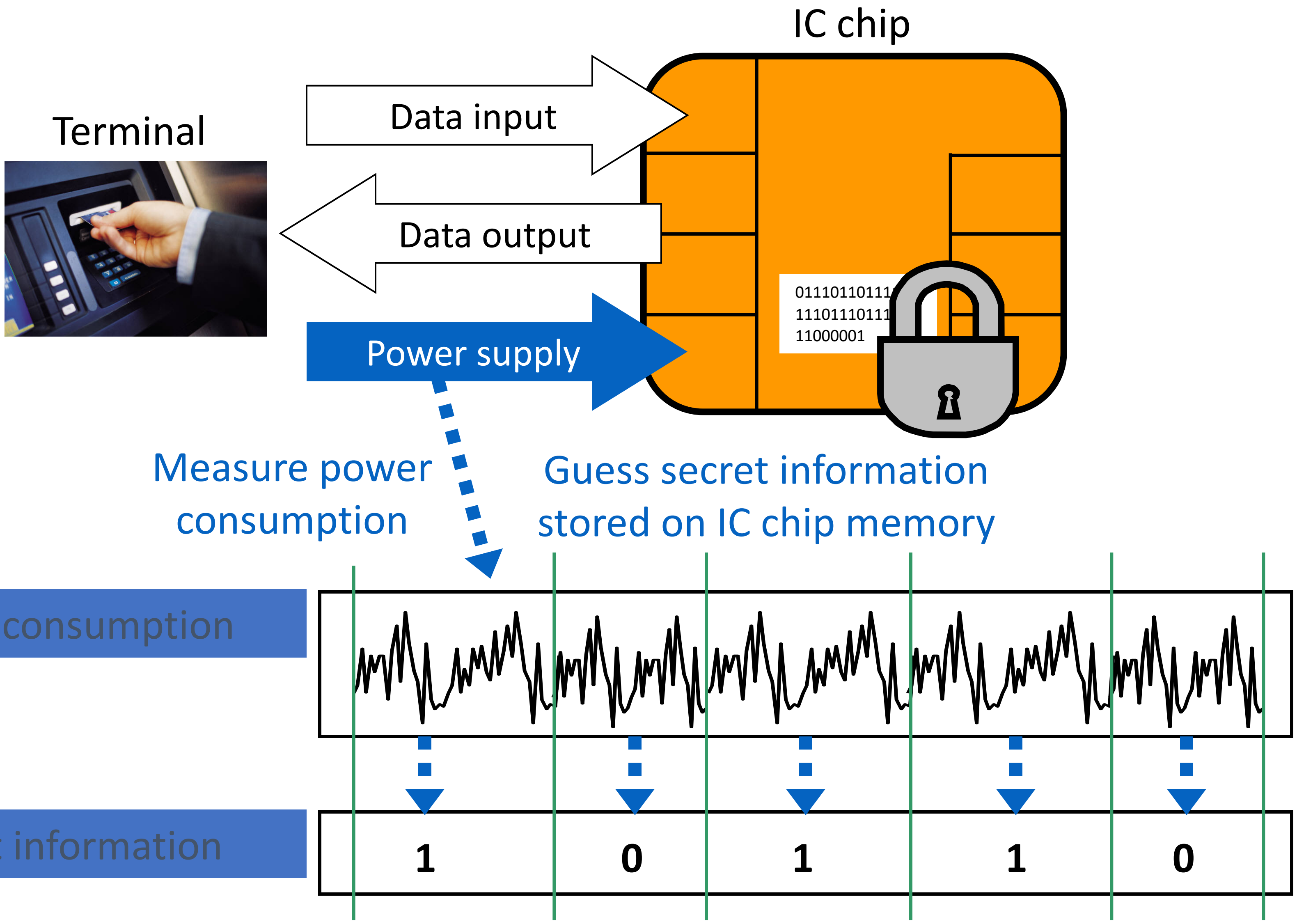
Power side channel attack and How it works to exploit RSA

Simplified!

Simple Power Analysis (SPA)

- Directly interprets the power consumption of the device
- Looks for the operations taking place and also the key!
- Assumption: attacker can get a power trace, representing a set of power consumptions across a cryptographic process

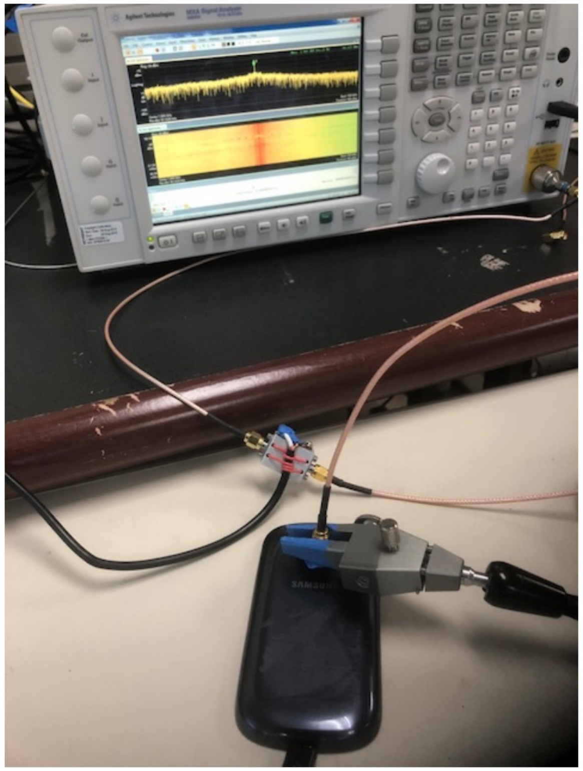
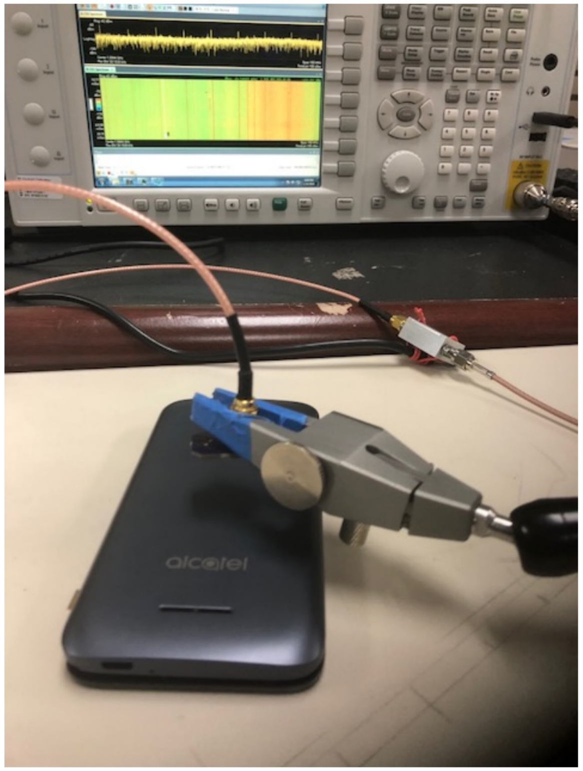
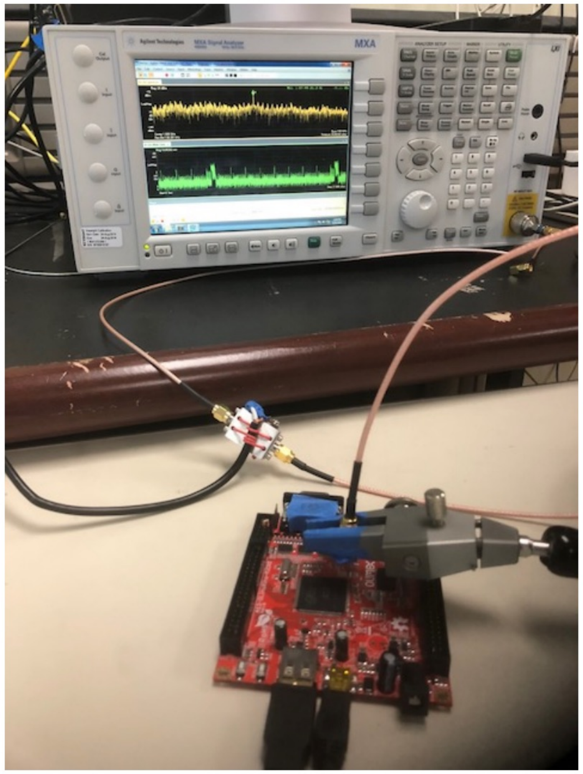
Simple Power Analysis



With SPA, one power trace is sufficient to reveal the secret

This slide is from Mark Stamp.

Simple Power Analysis (SPA)

POC attacks on mobile phone and embedded devices.