CSE 484: Computer Security and Privacy

Wrapup

Spring 2024

David Kohlbrenner

dkohlbre@cs

Thanks to Franzi Roesner, Dan Boneh, Dieter Gollmann, Dan Halperin, David Kohlbrenner, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Logistics

- Part B feedback is in progress, we'll have it to you by Monday at the latest.
- No late days on Part C
- There is an additional EC assignment for finding new, exploitable, bugs in tinyserv

Course feedback!

Please fill it out!

https://uw.iasystem.org/survey/290594

Security, Professionally

Looking for more security?

- CSE 490 Cryptography
- CSE 481S Security Capstone
- CSE 564 Graduate Computer Security

- CTFs
  - Batman's Kitchen is the UW team
  - Try picoctf (aimed at newcomers)

- Internships
  - Medium-size security companies are your best bet

Many avenues

- General software engineering
- Security engineering
- Incident response
- Network operations
- Penetration testing
- Misc security contracting
- Etc.

Many avenues

- General software engineering
- Security engineering
- Incident response
- Network operations
- Penetration testing
- Misc security contracting
- Etc.
- Security research + academia!

Security Research

Research

- Usability of security tools
- Studies on security&privacy expectations
- Studies on abuse and misinformation
- Finding new types of vulnerabilities
- Building tools for safer systems

Side channels

```
bool key[16];
bool msg[16];
bool ciphertxt[16];
```

Ciphertext is the msg XOR the key

```
bool key[16];
bool msg[16];
bool ciphertxt[16];
```

```
for(int i=0; i<16; i++){
      ciphertxt[i] = msg[i] ^ key[i];
}
```

```
bool key[16];
bool msg[16];
bool ciphertxt[16];
```

```
for(int i=0; i<16; i++){
      if (key[i] == 0){
            ciphertxt[i] = msg[i];
            sleep(0);
      }
      else{ // key[i] == 1
            ciphertxt[i] = ~msg[i];
            sleep(1);
      }
}
```

```
for(int i=0; i<16; i++){
      ciphertxt[i] = msg[i] ^ key[i];
}
```

```
bool key[16];
bool msg[16];
bool ciphertxt[16];



for(int i=0; i<16; i++){
      if (key[i] == 0){
            ciphertxt[i] = msg[i];
            sleep(0);
      }
      else{ // key[i] == 1
            ciphertxt[i] = ~msg[i];
            sleep(1);
      }
}
```

```
bool key[16];
bool msg[16];
bool ciphertxt[16];



for(int i=0; i<16; i++){
      if (key[i] == 0){
            ciphertxt[i] = msg[i];
      }
      else{ // key[i] == 1
             ciphertxt[i] = ~msg[i];
      }
}
```

Side-channels: conceptually

- A program's implementation (that is, the final compiled version) is different from the conceptual description

- Side-effects of the difference between the implementation and conception can reveal unexpected information
  - Thus: Side-channels

Detour: Covert-channels

- We'll see many unusual ways to have information flow from thing A to thing B
- If this is an *intentional* usage of side effects, it is a covert channel
- *Unintentional* means it is a side-channel
- The same *mechanism* can be used as a covert-channel, or abused as a side-channel

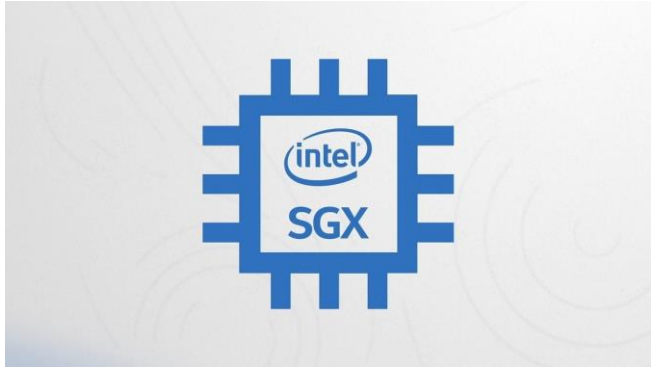
Side Channel Attacks

- Most commonly discussed in the context of cryptosystems
- But also prevalent in many contexts
    - E.g., we discussed the TENEX password implementation
    - E.g., we discussed browser fingerprinting

Why should we care about side-channels?

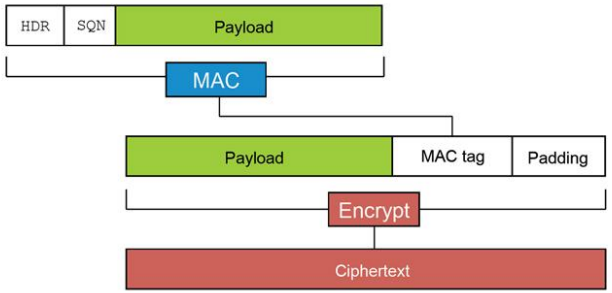
- Compromises happen via ‘simple’ methods
  - Phishing
  - Straight-forward attacks

- Embedded systems *do* see side-channel attacks
  - “Triangulation” attacks
- “High Security” systems *do* see side-channel attacks



Timing side-channels: round 2

- Cryptographic implementations fall down
  - #1 target for timing attacks
  - Extremely common to find vulnerabilities



- "Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS, and Other Systems"
  - Was very far from the last paper on the topic

Attacking cryptography with side-channels

- ANY leakage is bad
  - E.g. 1 bit of key leaking is ‘catastrophic’
- Cryptographic implementations are complex
  - Many layers of protocols

Example Timing Attacks

- **RSA:** Leverage key-dependent timings of modular exponentiations
  - https://www.rambus.com/timing-attacks-on-implementations-of-diffie-hellman-rsa-dss-and-other-systems/
  - http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf

- **Block Ciphers:** Leverage key-dependent cache hits/misses

Cache side-channels

Cache side-channels

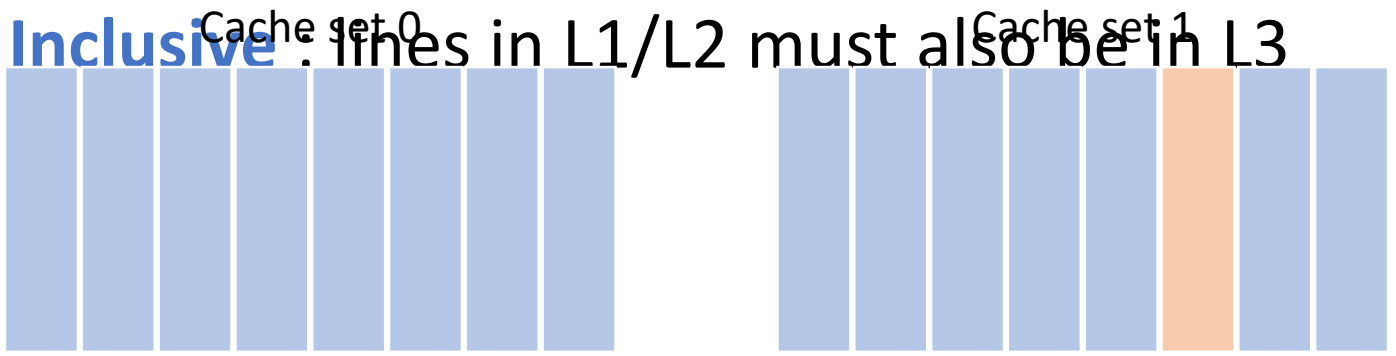
- **Idea**: The cache's current state implies something about prior memory accesses

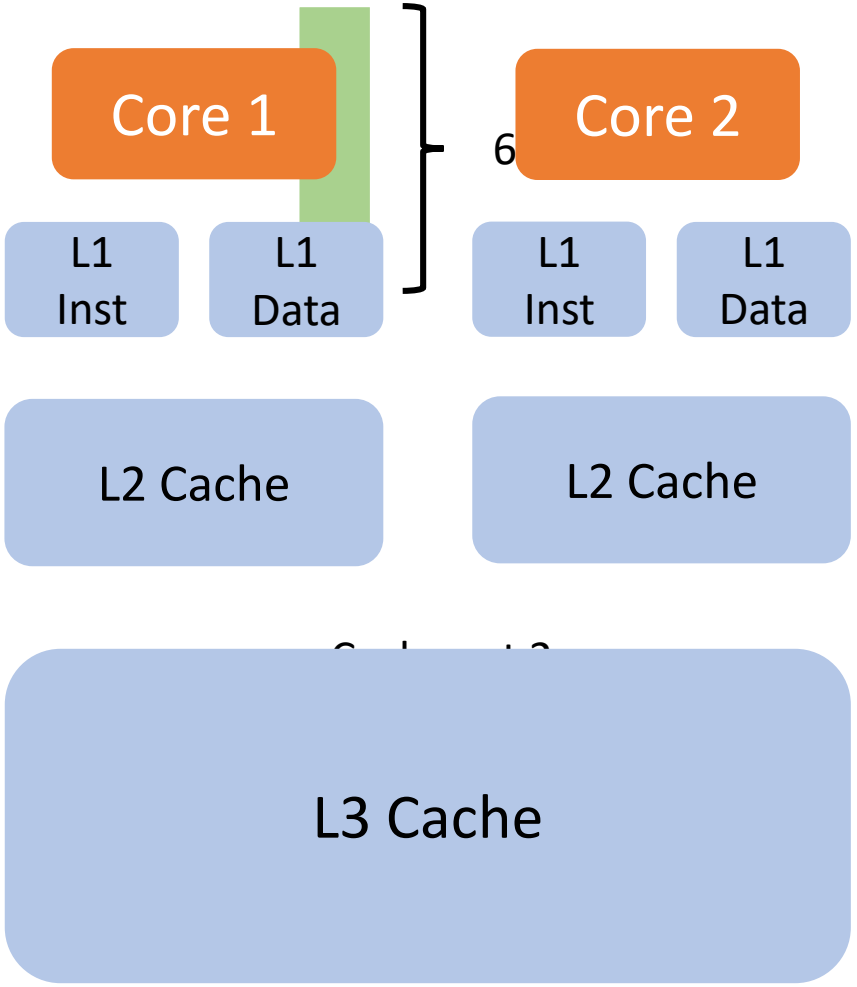
- **Insight**: Prior memory accesses can tell you a lot about a program!

Many thanks to Craig Disselkoen for the animations.

Cache Basics

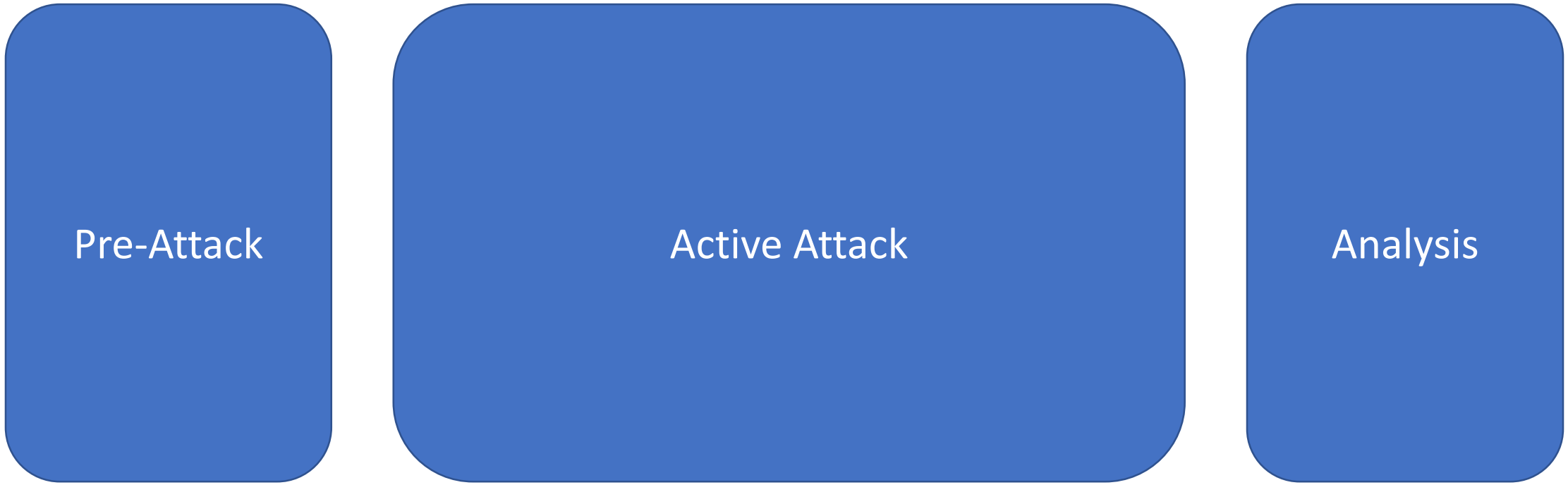
- **Cache lines** : fixed-size units of data
- **Cache set** : holds multiple cache lines
- **Set index** : assigns cache line to cache set
- **Eviction** : removing cache lines to make room
- **L1, L2, L3** : different levels of cache
- **Inclusive** : lines in L1/L2 must also be in L3



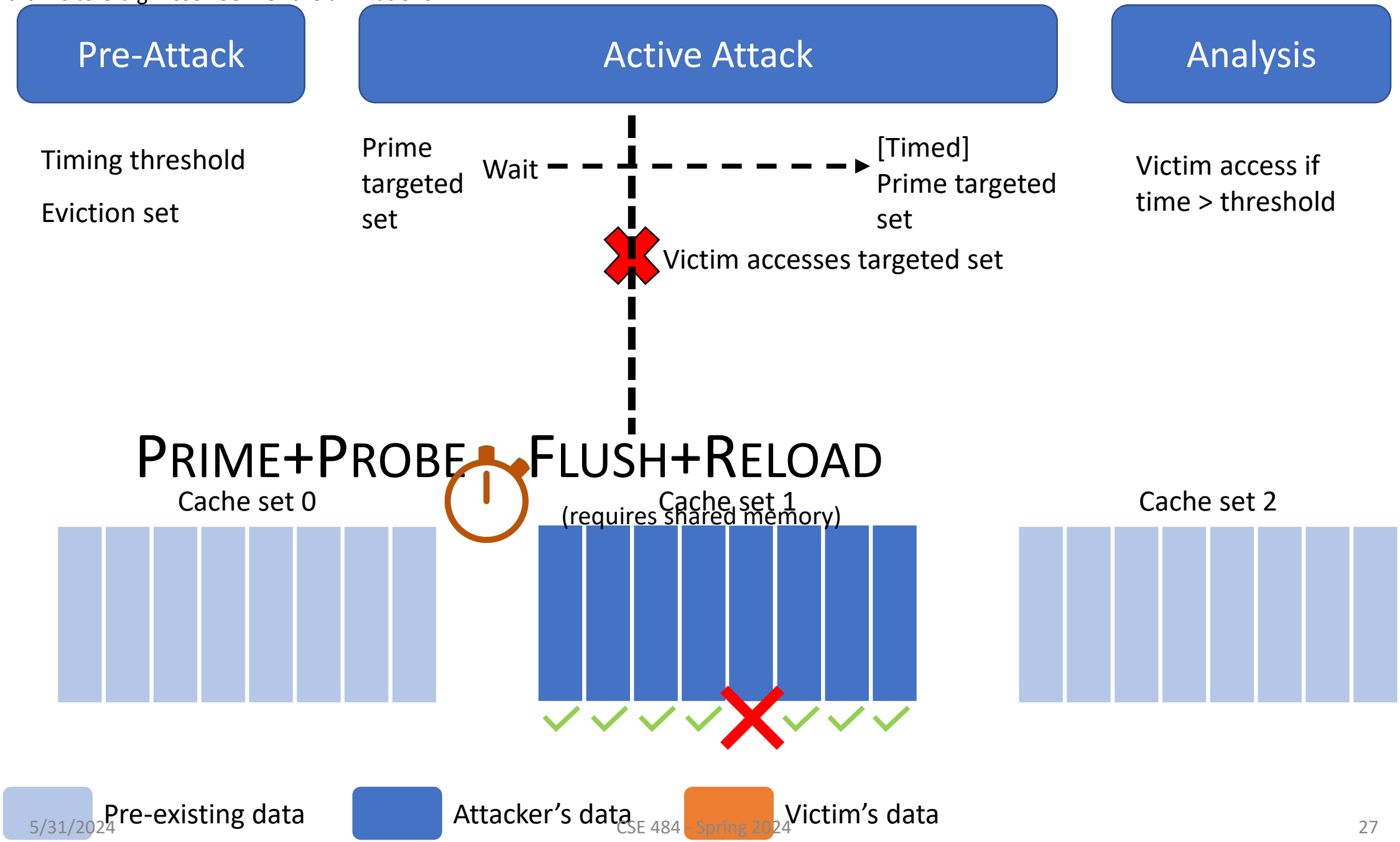


Many thanks to Craig Disselkoen for the animations.

Cache Attacks: Structure



Many thanks to Craig Disselkoen for the animations.



FLUSH + RELOAD

- Even simpler!
- Kick line L out of cache
- Let victim run
- Access L
  - Fast? Victim touched it
  - Slow? Victim didn’t touch it

Cache attacks wrapup

- Cache attacks are a core element of many side-channels
- Generally “assumed to work” these days
- New variations/tricks/mitigations published constantly
- Randomized caches are the current hotness

WRAP-UP

This Quarter

- Overview of:
  - Security mindset
  - Software security
  - Cryptography
  - Web security
  - Web privacy
  - Authentication
  - Mobile platform security
  - Usable security
  - Anonymity
  - Side channels
  - Security for emerging tech

Lots We Didn't Cover…

- Really deep dive into any of the above topics
- (Most) Network security
- (Most) Traditional OS security
- (Most) Recent attacks/vulnerabilities
- (Most) Specific protocols (e.g., SSL/TLS, Kerberos)
- Access control
- Spam
- ML Security/Privacy
- Malware / Bots / Worms
- Social engineering
- Cryptocurrencies (e.g., Bitcoin)
- Other emerging technologies
- …

Thanks for a great quarter! Hang in there.

- Stay in touch

  I'm always happy to answer questions or point you in directions on S&P ☺

- Not ready to be done?
  - CSE 490 Cryptography
  - CSE 481S Security Capstone
  - CSE 564 Graduate Computer Security

- Please fill out course evaluation:
  - https://uw.iasystem.org/survey/290594