A Survey on SGX Enclave Privileged Side-Channel Attacks

Amin Fallahi
Ph.D. Student

Syracuse University
Department of Electrical Engineering and Computer Science

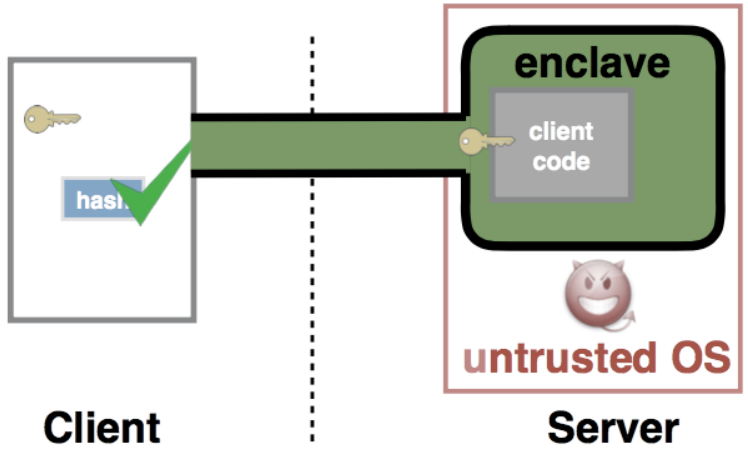
MAY 3, 2018

Contents

- Introduction
- Attacks
- Defenses
- Analysis
- Conclusion

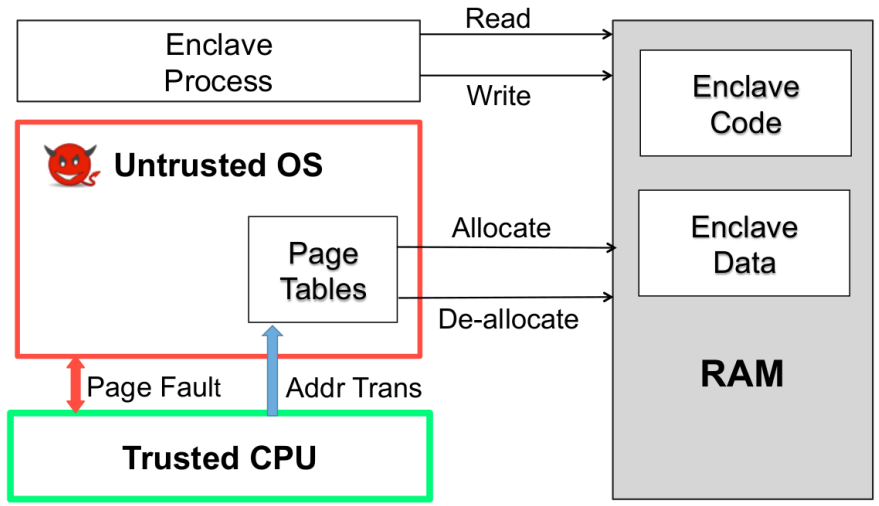
Introduction

- Computing environment
  - Shared resources executing user programs
- Security inside the cloud
  - A hardware solution needed
- Intel SGX
  - Secure user code and data in the public cloud
  - Protecting data inside the enclave
- Privileged Operating System
  - Kernel space, hardware control



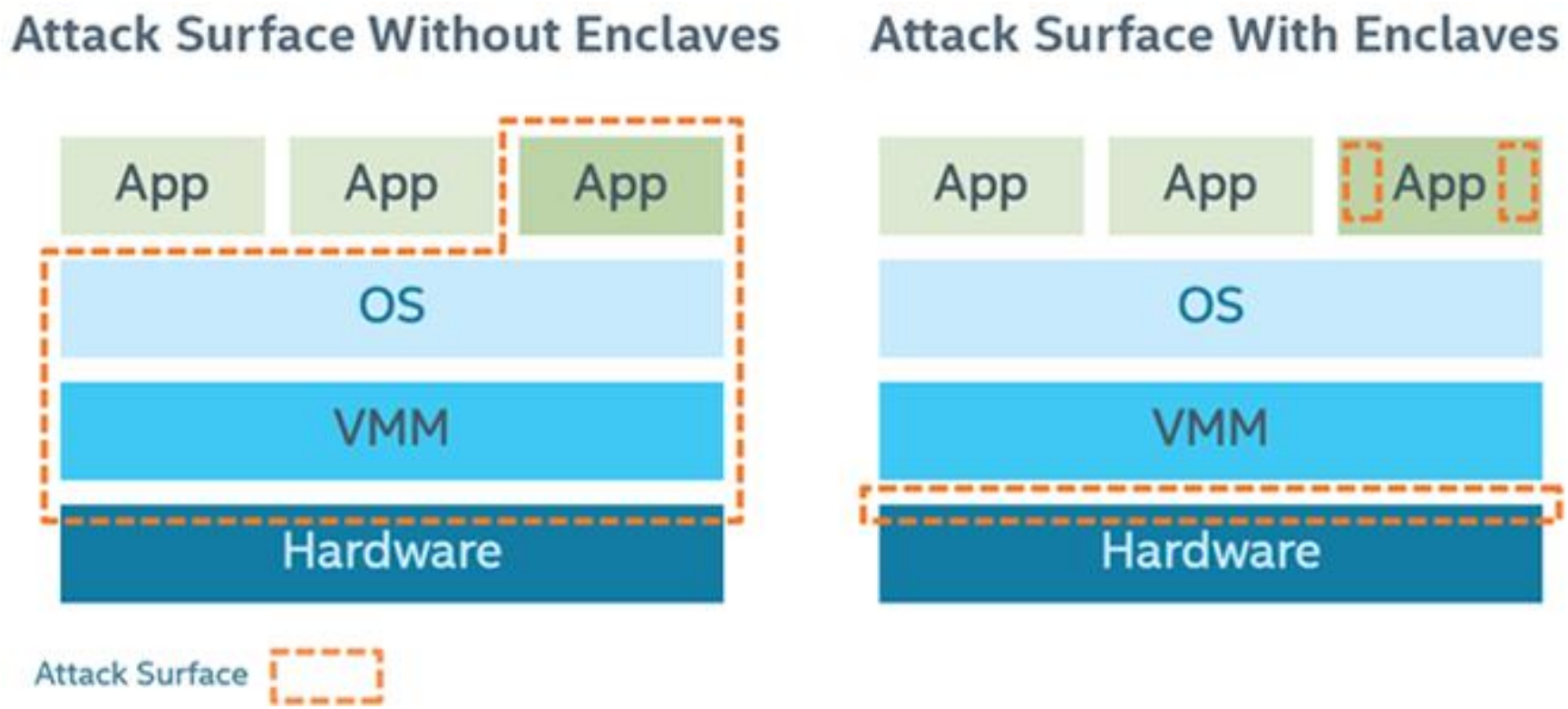
Introduction

- Side-Channels
  - Attacks by analyzing system behavior
  - Cache attacks: Monitoring cache accesses
  - Timing attacks: Measuring time between computations
  - Page-fault attack: Monitoring page-faults and analyzing the pattern



Introduction

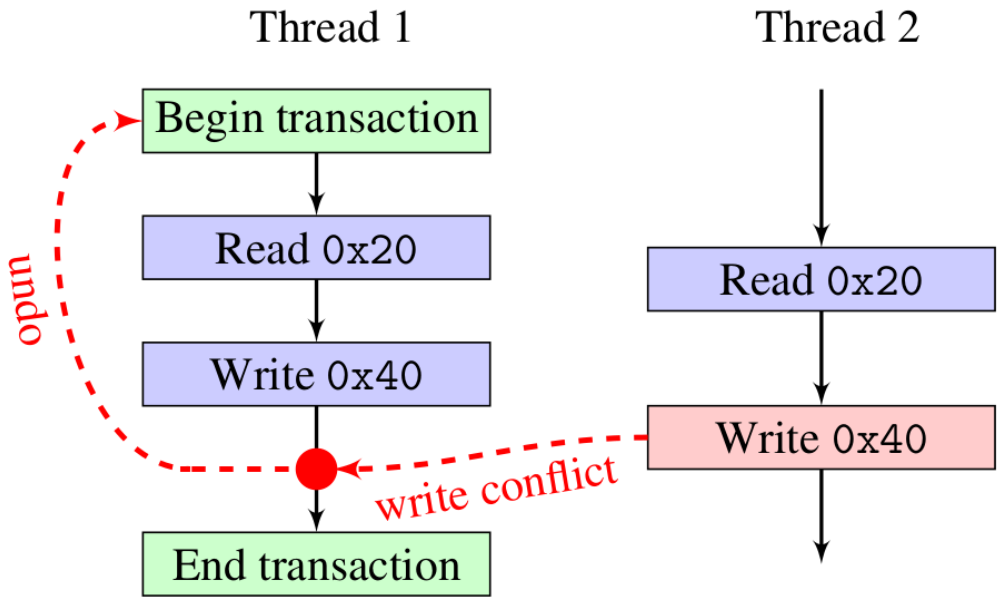
Intel SGX



Introduction

Intel TSX

- Introduced with Haswell
- Critical sections management
- Monitoring serialization
- Restricted Transactional Memory
  - New Instruction set interface



Attacks

Attack Model

- Enclave protected by SGX
  - No stranger has the key
- Operating System can manage task queues, interrupts, and exceptions
- Viewing data transfer between memories and caches
- Program pinned to one core, no excessive interrupts, isolated

Attacks

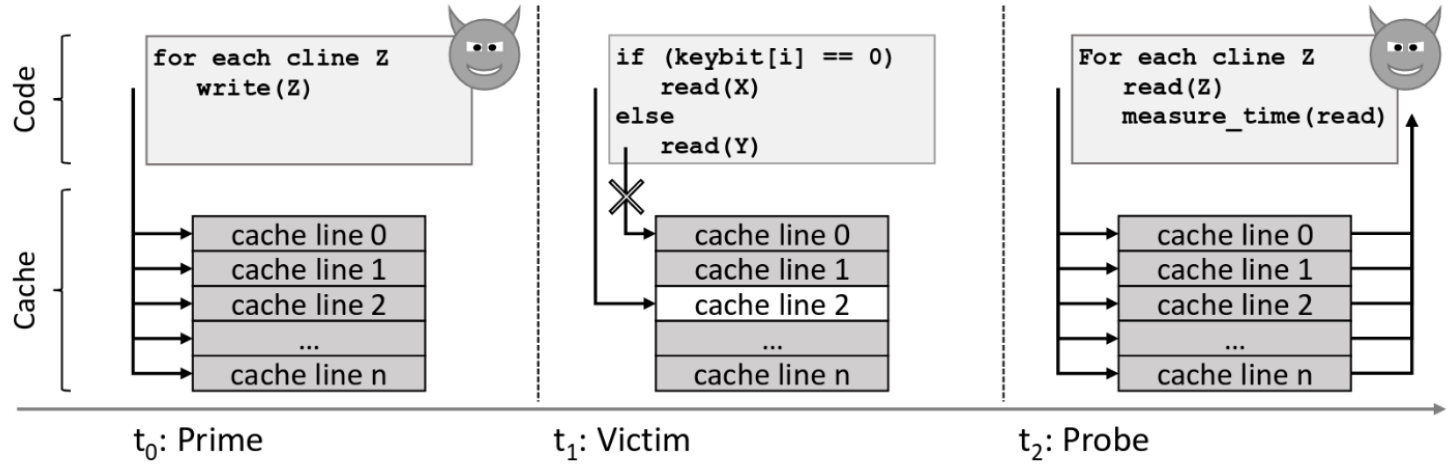
Points of attack

- TLB: shared between enclave and non-enclave programs ⇨ with hyperthreading enabled it creates side-channels
- Page faults: visible to OS
- TLB flushes: on context switch
- Page table entry flags
- Enclave memory address beginning and offset are known
- Enclave exits (AEX)

Attacks

Prime+Probe

- Prime: attacker executes conflicting cache lines ⇨ cache miss
- Probe: executing all the cache lines and measuring execution time



Attacks

Flush+Reload

- Flush target memory line
- Wait for the victim to access

```
if (secret) {
    ;
} else {
    ;
}
```

- Request to access target memory line ⇨ access time
- Based on the access time, access by the victim will be revealed

Attacks

Stealthy page table based attacks

- Introduced by Van Bulck et al.
- Basic page-fault attack:
  - OS controls page tables
  - Sets trap by making pages inaccessible
  - Observe page-fault patterns
  - Or simply monitor certain pages and cause page-faults
- Page table entry flags
  - Accessed and dirty flags

Attacks

Sneaky page monitoring attacks

- Introduced by Wang et al.
- Accessed flags monitoring
  - Flags in TLB are not updated
  - Flush the TLB (by IPI)
- Timing enhancement
  - Only one interrupt ⇨ better performance
  - Measure the time between two repeatedly accessed entries
- TLB flushing through hyperthreading
  - TLB is shared between hyperthreads
  - Invalidate TLB entries without IPI

Attacks

DRAMA

- Disable caching
  - Prevent cache based side channel attacks
- Attacker allocates two memory lines inside one bank
- Regularly access one of the memory lines
- Victim accesses the other memory line ⇨ conflict ⇨ attacker's next fetch will take more time
- Cache-DRAM attack: Prime+Probe and DRAMA

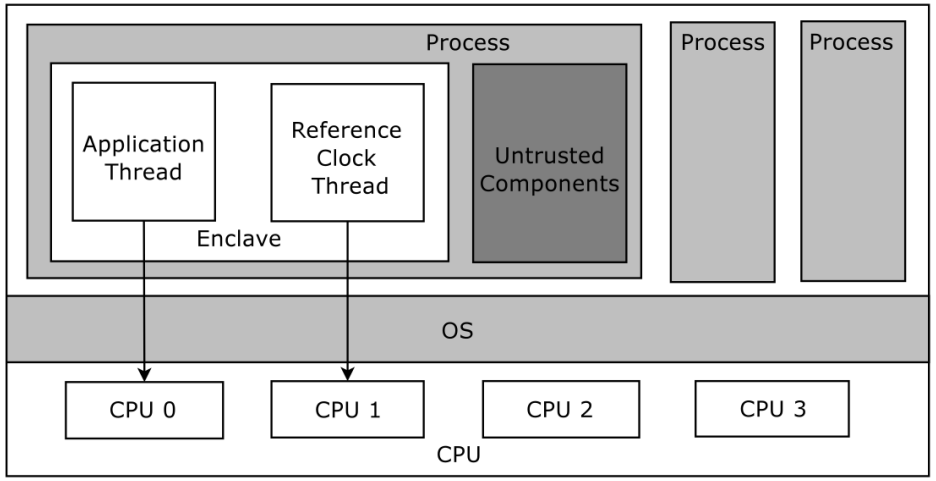
Attacks

Flush+Flush

- *clflush* instruction: flush a cache line, if empty ⇨ abort
- Abort takes less time than flush
- Use the delay ⇨ which memory address is accessed

Defenses

- Déjà vu
  - Embed a clock inside enclave
  - Protect the clock inside TSX transaction
  - Record regular execution time
  - Time runs out ⇨ AEX instability ⇨ attack detected
- Shinde et al.
  - Deterministic page access profile
  - Fake accesses



Defenses

- T-SGX
  - CPU does not deliver page-fault to the OS
  - Abort transaction and run the fallback code
  - Lots of aborts
  - Break into execution blocks
- Cloak
  - Pin data in cache: Not supported by hardware
  - Preload code/data in transaction and run the algorithm
- SGX-Shield
  - Address space layout randomization (ASLR)
    - Secure in enclave loading
    - Limited memory ⇨ small randomization entropy ⇨ brute force attacks

Analysis

- Cloak
  - Makes cache-pinning possible to some extent
  - Execution time can leak
  - Aborts do not cancel concurrent memory accesses
  - Execution behavior and branch prediction uncertainties
- T-SGX
  - Attacks based on monitoring flags are possible

Analysis

- Déjà vu
  - Sneaky page monitoring attacks still effective
  - Only works on AEX based attacks
- Bottom line
  - Transaction based defenses usually come with high overload and low utilization

Analysis

- SGX-Shield
  - No live randomization ⇨ observe and monitor random patterns
- Shinde et al.
  - Cache and TLB based attacks possible

Analysis

- Other methods
  - Attack detection methods ⇨ Unreliable
  - Shuffling memory ⇨ expensive
  - ORAM (make addresses input-independent) ⇨ expensive
- The perfect solution?
  - Remove all branches and conditional code
  - Pin data/code to cache and TLB

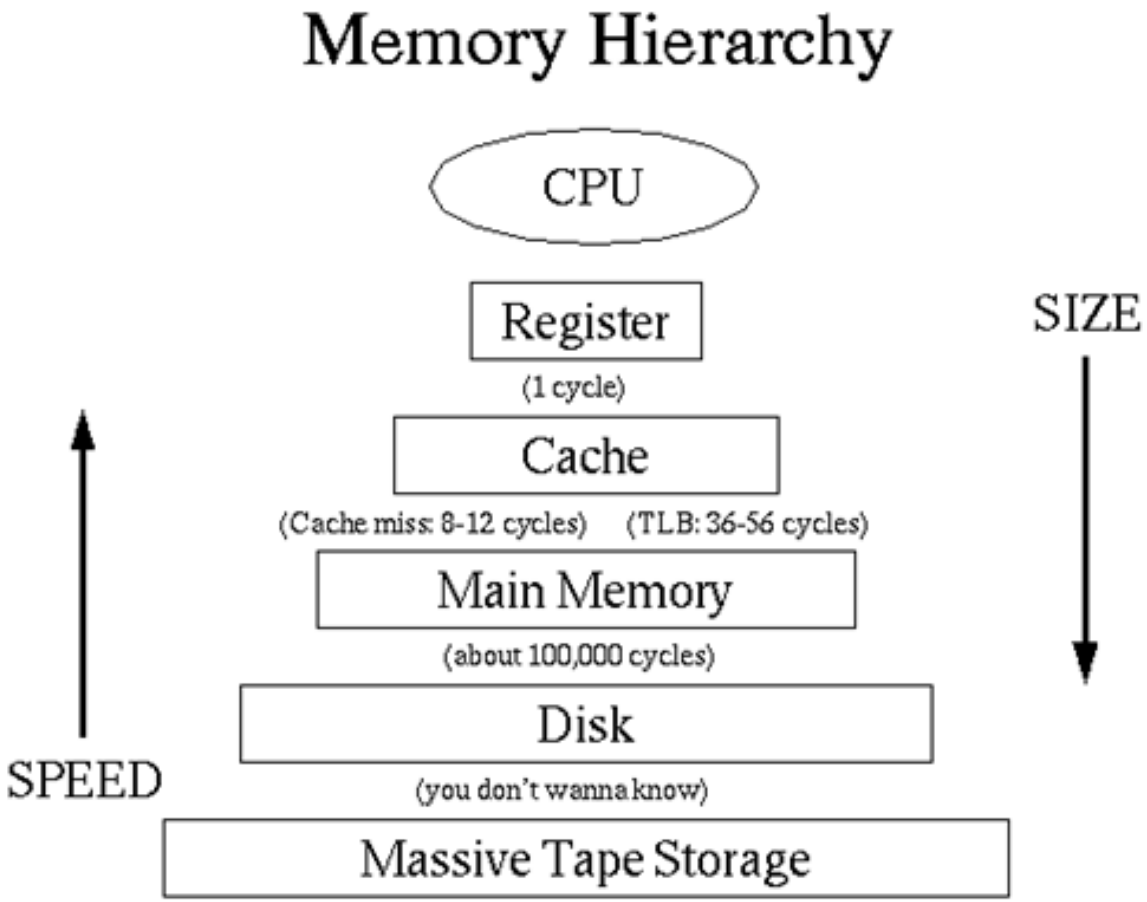
Conclusion

- We studied multiple attacks and defenses
- Still no robust defense
- Attack are emerging
- Open to research

References

[1] Ferdinand Brasser, Srdjan Capkun, Alexandra Dmitrienko, Tommaso Frassetto, Kari Kostiainen, Urs Mu¨ller, and Ahmad-Reza Sadeghi. 2017. DR. SGX: Hardening SGX Enclaves against Cache Attacks with Data Location Randomization. arXiv preprint arXiv:1709.09917 (2017).
[2] Ferdinand Brasser, Urs Mu¨ller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and AhmadReza Sadeghi. 2017. Software grand exposure: SGX cache attacks are practical. arXiv preprint arXiv:1702.07521 (2017), 33.
[3] Sanchuan Chen, Xiaokuan Zhang, Michael K Reiter, and Yinqian Zhang. 2017. Detecting privileged side-channel attacks in shielded execution with D´ej´a Vu. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 7–18.
[4] Oded Goldreich. 1987. Towards a theory of software protection and simulation by oblivious RAMs. In Proceedings of the nineteenth annual ACM symposium on Theory of computing. ACM, 182–194.
[5] Oded Goldreich and Rafail Ostrovsky. 1996. Software protection and simulation on oblivious RAMs. Journal of the ACM (JACM) 43, 3 (1996), 431–473.
[6] Daniel Gruss, Julian Lettner, Felix Schuster, Olya Ohrimenko, Istvan Haller, and Manuel Costa. 2017. Strong and efficient cache side-channel protection using hardware transactional memory. In USENIX Security Symposium.
[7] Daniel Gruss, Cl´ementine Maurice, Klaus Wagner, and Stefan Mangard. 2016. Flush+ Flush: a fast and stealthy cache attack. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 279–299.
[8] Mehmet Kayaalp, Nael Abu-Ghazaleh, Dmitry Ponomarev, and Aamer Jaleel. 2016. A high-resolution side-channel attack on last-level cache. In Proceedings of the 53rd Annual Design Automation Conference. ACM, 72.
[9] Rafail Ostrovsky. 1990. Efficient computation on oblivious RAMs. In Proceedings of the twenty-second annual ACM symposium on Theory of computing. ACM, 514–523.
[10] Dag Arne Osvik, Adi Shamir, and Eran Tromer. 2006. Cache attacks and countermeasures: the case of AES. In Cryptographers' Track at the RSA Conference. Springer, 1–20.
[11] Peter Pessl, Daniel Gruss, Cl´ementine Maurice, Michael Schwarz, and Stefan Mangard. 2016. DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks.. In USENIX Security Symposium. 565–581.
[12] Sajin Sasy, Sergey Gorbunov, and Christopher Fletcher. 2017. ZeroTrace: Oblivious memory primitives from Intel SGX. In Symposium on Network and Distributed System Security (NDSS).
[13] Jaebaek Seo, Byounyoung Lee, Seongmin Kim, Ming-Wei Shih, Insik Shin, Dongsu Han, and Taesoo Kim. 2017. SGX-Shield: Enabling address space layout randomization for SGX programs. In Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA.
[14] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. 2017. T-SGX: Eradicating controlledchannel attacks against enclave programs. In Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA.
[15] Shweta Shinde, Zheng Leong Chua, Viswesh Narayanan, and Prateek Saxena. 2015. Preventing your faults from telling your secrets: Defenses against pigeonhole attacks. arXiv preprint arXiv:1506.04832 (2015).
[16] Eran Tromer, Dag Arne Osvik, and Adi Shamir. 2010. Efficient cache attacks on AES, and countermeasures. Journal of Cryptology 23, 1 (2010), 37–71.
[17] Jo Van Bulck, Nico Weichbrodt, Ru¨diger Kapitza, Frank Piessens, and Raoul Strackx. 2017. Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution. In Proceedings of the 26th USENIX Security Symposium. USENIX Association.
[18] Wenhao Wang, Guoxing Chen, Xiaorui Pan, Yinqian Zhang, Xiaofeng Wang, Vincent Bindschaedler, Haixu Tang, and Carl A. Gunter. 2017. Leaky Cauldron on the Dark Land: Understanding Memory Side-Channel Hazards in SGX. 2421–2434.
[19] Yuval Yarom and Katrina Falkner. 2014. FLUSH+ RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack.. In USENIX Security Symposium

Introduction



Defenses

- SGX-Shield
  - Address space layout randomization
- ORAM+SGX
- Data shuffling