Designing SGX Applications for Recoverability

Stanford University, Google* Paul Crews ptcrews@cs.stanford.edu

Background and Motivation

Background: Most SGX threat models assume that SGX and the enclave application are secure

Problem: SGX vulnerabilities and application software bugs make this an unrealistic assumption

Proposal: Define a new threat model that allows an attacker to compromise an enclave at time t_C

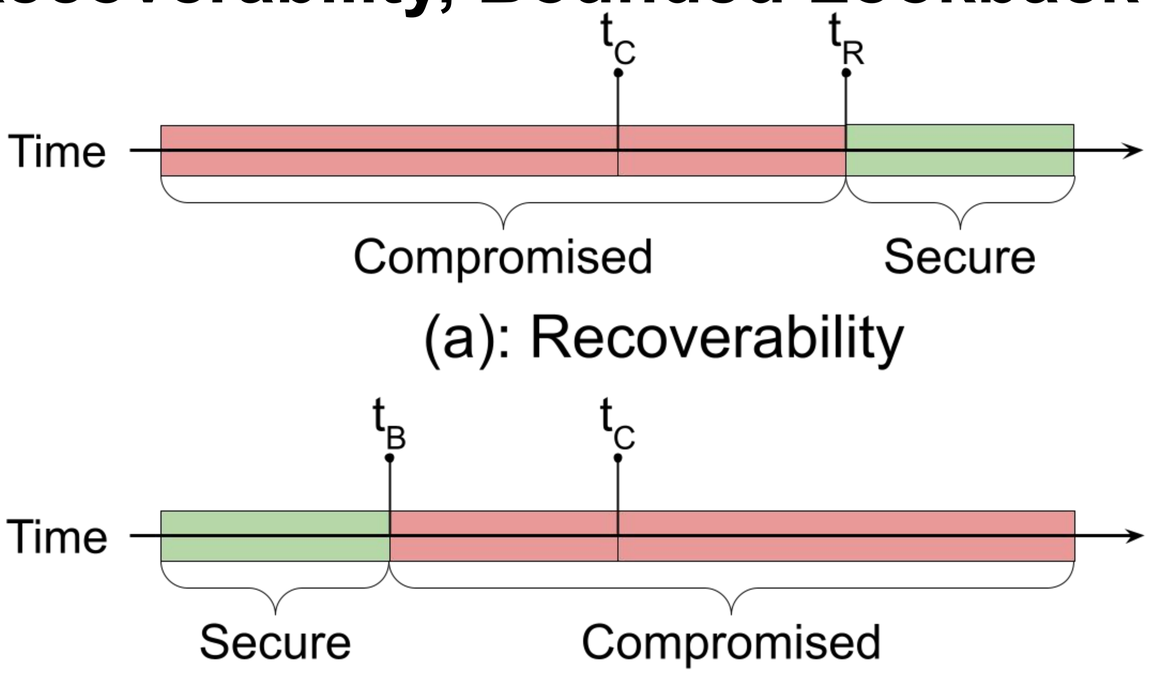
Motivating Questions:

- What security guarantees can we provide?
- What applications are suitable to run in an enclave?

Enclave Security Property Compromised	Examples
Confidentiality	Page Fault Attacks[1,2], Plundervolt[3], Foreshadow[4], SGAxe[5]
Integrity	Plundervolt[3]
Attestation	Plundervolt[3], Foreshadow[4], SGAxe[5]

1. Shweta Shinde, Zheng Leong Chua, Viswesh Narayanan, and Prateek Saxena. 2016. Preventing page faults from telling your secrets.
2. Jo Van Bulck, Nico Weichbrodt, Rüdiger Kapitza, Frank Piessens, and Raoul Strackx. 2017. Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution.
3. Kit Murdock, David Oswald, Flavio D Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. 2020. Plundervolt: Software-based fault injection attacks against Intel SGX.
4. Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, and Raoul Strackx. 2018. Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution.
5. Stephan van Schaik, Andrew Kwong, Daniel Genkin, and Yuval Yarom. 2020. SGAxe: How SGX fails in practice.

Recoverability, Bounded Lookback



Threat Model: An attacker compromises an enclave at time t_C

Recoverability: Recovery actions at time t_R prevent attackers from persisting after t_R if $t_C < t_R$

Bounded Lookback: Bounding actions at time t_B prevent attackers from compromising security properties before time t_B if $t_B < t_C$

Analyzing SGX Applications

Analysis: Examined how to add recoverability, bounded lookback to EnclaveDB[6], SecureKeeper[7], and Signal's Private Contact Discovery[8]

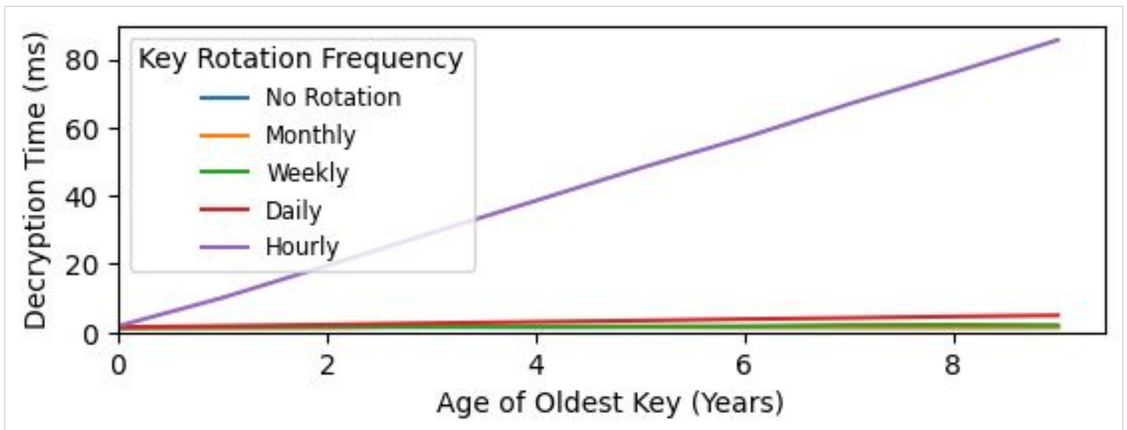
Implementation: Added recoverability to a mock EnclaveDB by implementing key rotations (~1k LoC)

- Measured performance overhead for decryption

Key Takeaways:

- Difficulty is based on the specific security property
  - E.g. Confidentiality + bounded lookback is hard
- Persistent state inhibits recoverability/lookback

SGX Application	How to Add Recoverability	Bounded Lookback
EnclaveDB[6]	Key Rotation	Request Validation
SecureKeeper[7]	Not Possible?	Not Possible?
Signal's PCD[8]	Supported	Supported



6. Christian Priebe, Kapil Vaswani, and Manuel Costa. 2018. EnclaveDB: A Secure Database Using SGX.
7. Stefan Brenner, Colin Wulf, David Goltzsche, Nico Weichbrodt, Matthias Lorenz, Christof Fetzer, Peter Pietzuch, and Rüdiger Kapitza. 2016. SecureKeeper: Confidential ZooKeeper Using Intel SGX.
8. Moxie Marlinspike. 2017. https://signal.org/blog/private-contact-discovery/.

Future Work

- Implement a recoverability library for SGX
- Examine how application design influences recoverability, bounded lookback
- Generalize to other trusted hardware (AMD SEV-SNP, Intel TDX, TPMs)

*This work was performed in the context of a course at Stanford University. Current affiliation: Google LLC

Disclaimer: This work was performed in the personal capacity of the author and does not represent the views of Google or Alphabet.