\Orchestrating a brighter world NEC

No Forking Way: Detecting Cloning Attacks on Intel SGX Applications

Samira Briongos, Ghassan Karame, Claudio Soriente, Annika Wilde

Annual Computer Security Applications Conference (ACSAC) 2023

SPATIAL

Artifacts Evaluated acm Functional V1.1

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

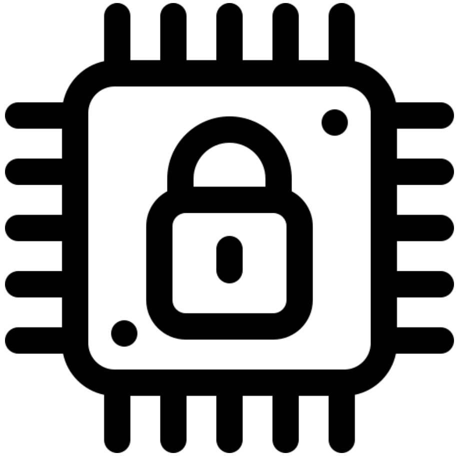
RUHR UNIVERSITÄT BOCHUM RUB

Contents

1. Motivation
2. Research question
3. CloneBuster
4. Conclusions

Motivation: Intel SGX

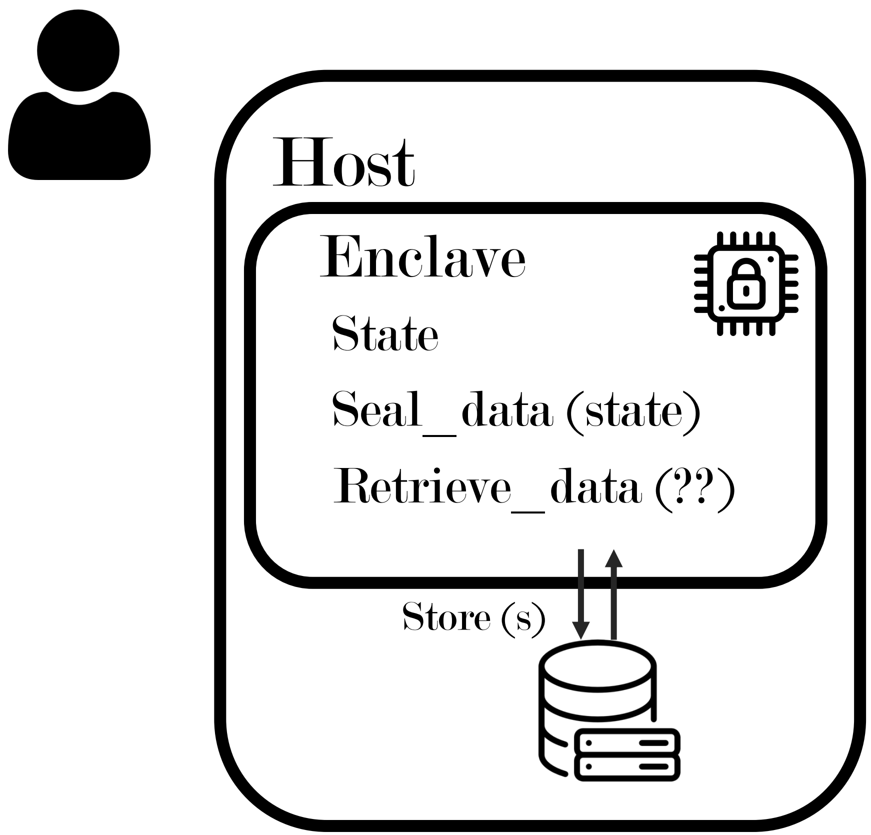
- Intel SGX is a set of extensions that provide runtime hardware protection to both code and data even if other code components are malicious

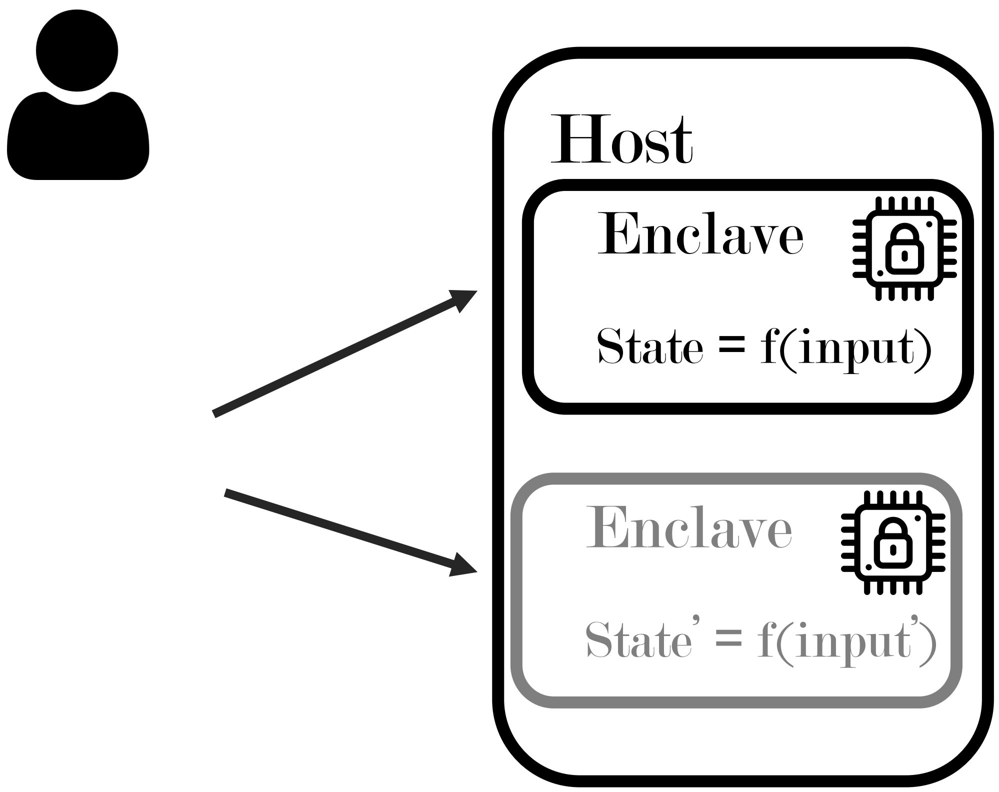


- Vulnerable to different attacks: transient execution attacks, microarchitectural attacks, rollback attacks, forking attacks…

Motivation: Rollback and Forking attacks

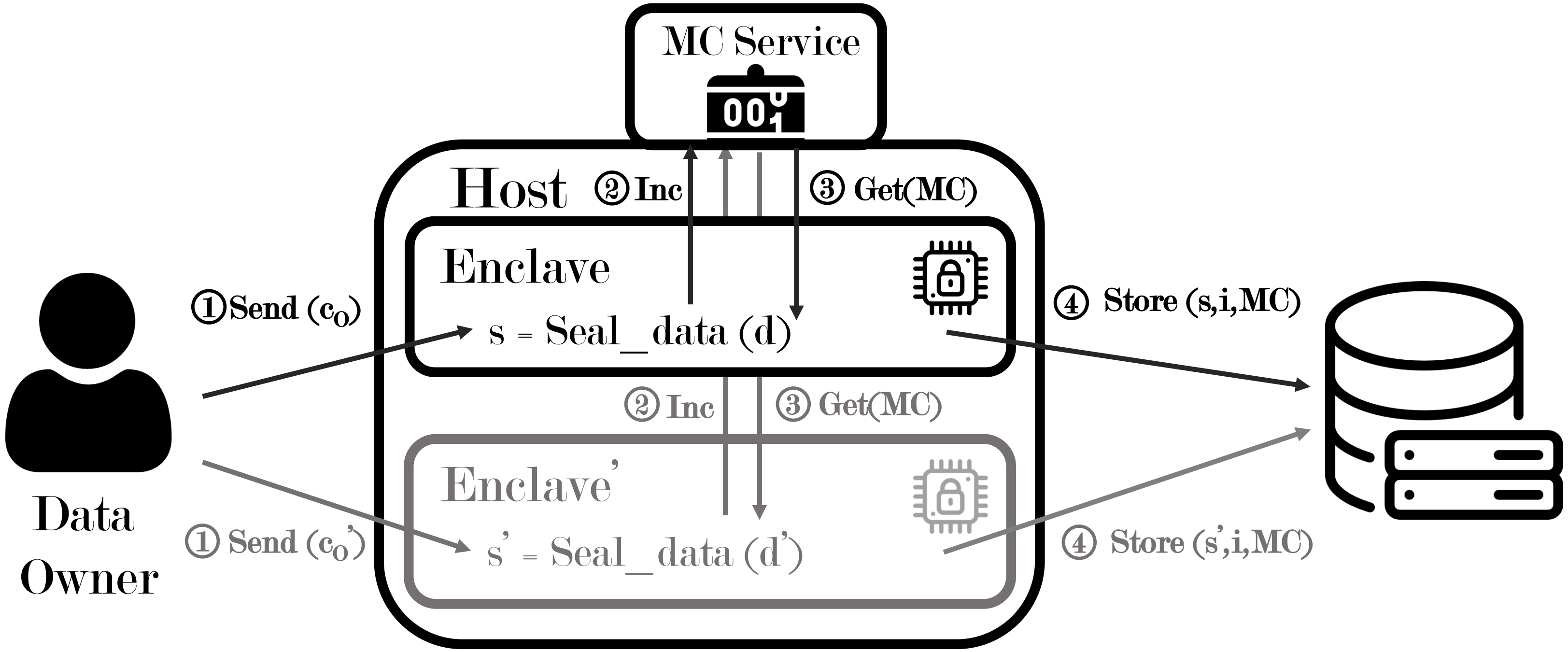
- Rollback attacks: the enclave state can be reverted to a previous one
- Forking attacks: multiple clones of an enclave lead to an inconsistent state





Motivation: Analysis of SGX Applications

- We analyzed 72 SGX-based applications and 14 of them were vulnerable to forking attacks (3 of them included monotonic counters)



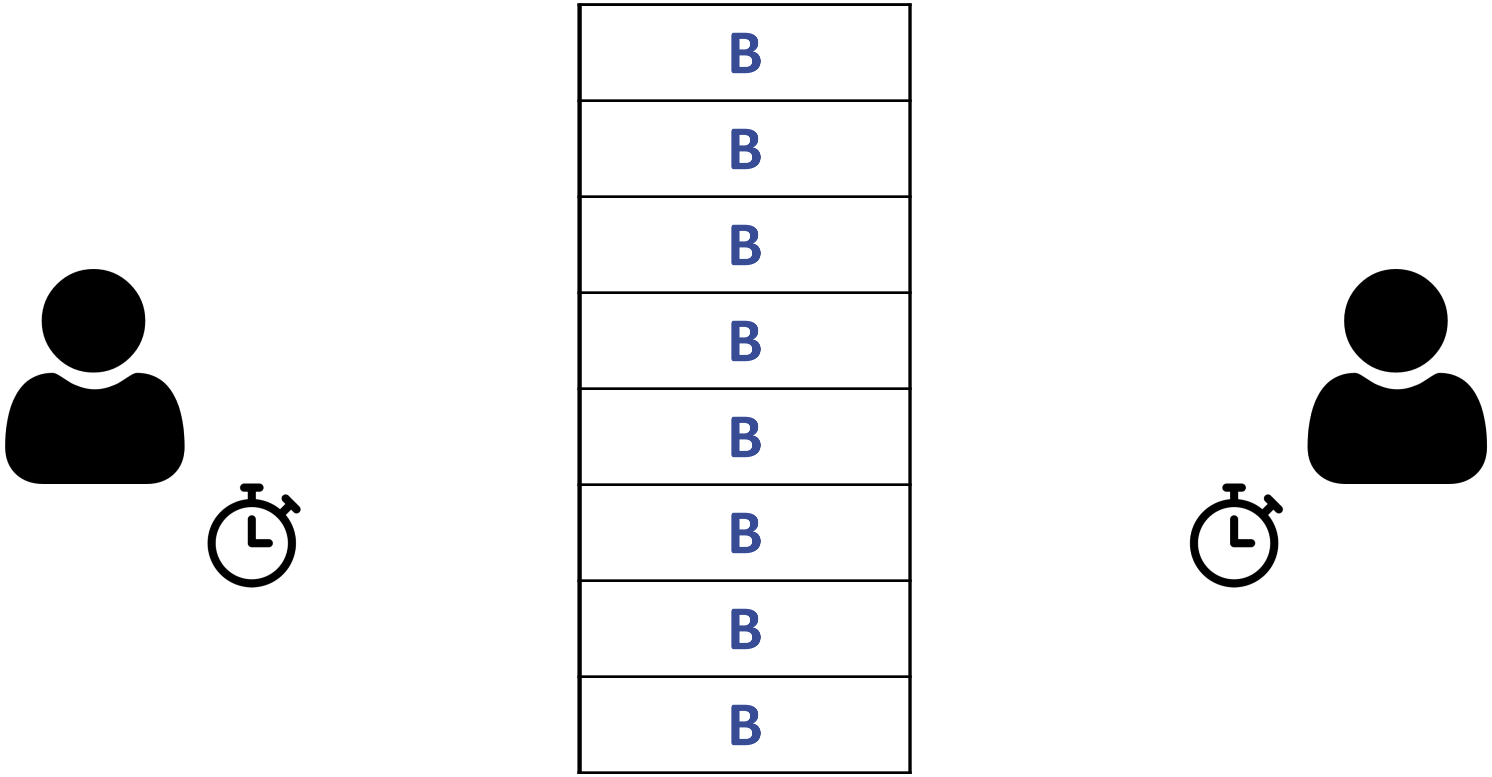
Research questions

- Can we design an anti-cloning solution that is:
  - practical,
  - efficient,
  - and does not require a TTP?

- Recall that clones share the same binaries and the same **hardware**

CloneBuster

- Idea: it is possible to establish a covert channel between to processes running on the same machine
  - Cache memories.

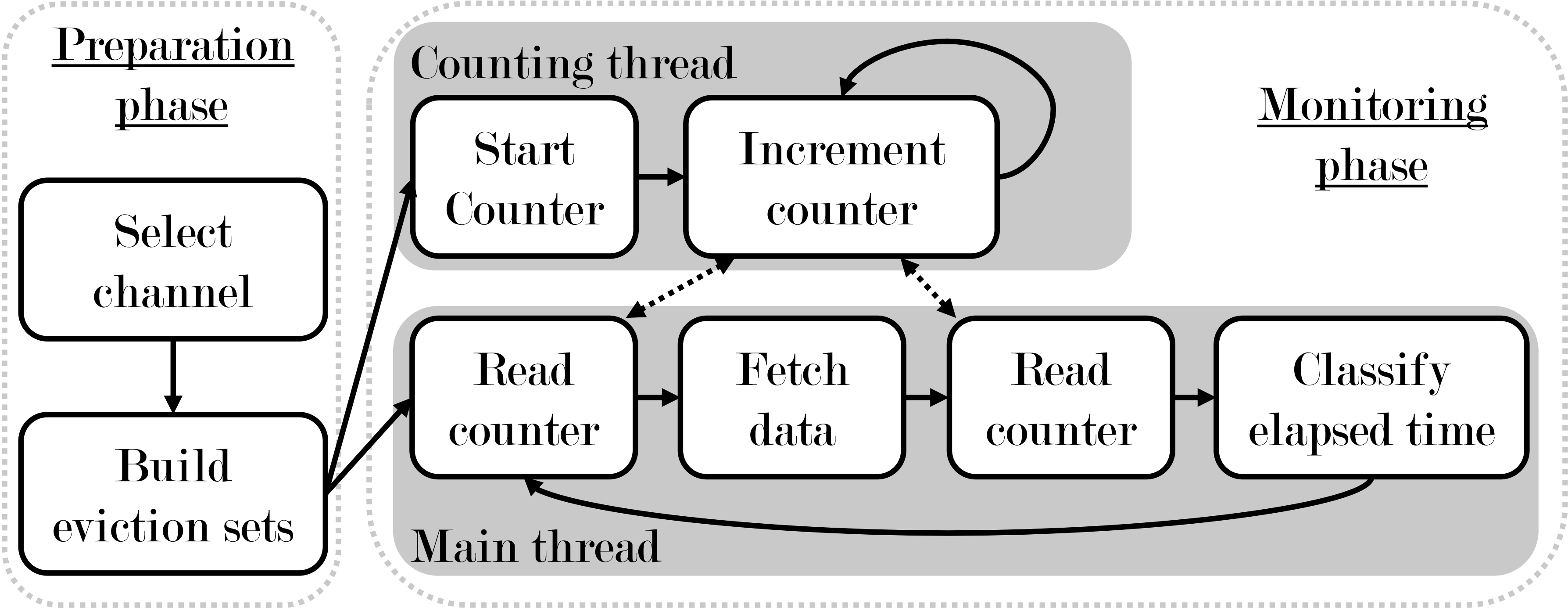


CloneBuster

- Considerations:
- Sgx does not provide high accuracy timers (e.g. rdtsc)
  - Previous work suggest a counting thread
- Enclaves are not aware of physical addresses of their data
  - Still they can gain some information if the mapping functions of the cache or DRAM are known in advance
- The enclave needs to know some details about the HW in advance
- The OS might be malicious and try to break the communication

CloneBuster

- Proposal



CloneBuster

- We have implemented a prototype for its evaluation:
  - Access pattern that minimizes clone detection time
  - Defines up to 64 channels for monitoring the cache.
- Runs several tests to ensure all the sets in the channel have been built
- Does not allow applications to run until all the eviction sets are created
- Data might be prefetched
- Needs to be running during the whole execution time of the protected application

CloneBuster: Evaluation

- We have evaluated the impact on performance of
  - Observation window size
  - Number or monitored ways per set
  - Classification algorithm
  - Noise (other applications running on the same machine)
  - Overhead (WolfSSL benchmark)
- Less than 5% overhead introduced in protected applications
- F1 score of 0.99 even in the presence of noise

- Further experiments in an extended version of the paper

Conclusions

- Providing protection against forking attacks is tricky and SGX applications are still vulnerable to them.
- Clones share the same hardware, which can be leveraged to detect the presence of clones.
- We have designed CloneBuster
  - Does not require a TTP
  - Low overhead
  - Robust in the presence of noise
- Source code is available

Artifacts Evaluated
acm
Functional V1.1

Thank you very much for your Attention

Artifacts: https://github.com/nec-research/CloneBuster

Artifacts Evaluated Functional V1.1

Contact: Samira.Briongos@neclab.eu

𝕏 @sambriongos