Computation Capabilities of Server-Side Trusted Execution Environments

A Comparison of TEEs to Privacy-Preserving Technologies

Research Questions

What are the computational limitations and capabilities of server-side Trusted Execution Environments concerning functionality, efficiency, security, and usability?

How do Trusted Execution Environments compare to Fully Homomorphic Encryption (FHE), Oblivious RAM (ORAM), Structured Encryption (STE), and Secure Multi-Party Computation (MPC)?

1. Introduction

- Despite running vulnerable software that can leak sensitive data-in-use, cloud servers are complex and widely adopted.
- TEEs provide hardware-based isolation for secure computation, with embedded cryptographic primitives. Many implementations exist, with varying trade-offs.
- Some protocols exist (FHE, MPC, ORAM, StE), but are known to be inflexible or inefficient. No prior work has systematically compared TEEs to these cryptographic techniques in a unified framework.

2. Methodology

- Literature review by following the **Backward Snowballing** method, starting from a System of Knowledge work [7].
- Selected relevant technologies and extracted their whitepaper work for a detailed description.
- Enriched literature by querying papers on Scopus regarding the four characteristics, leading to 69 materials.
- Compared common features of technologies with peers.

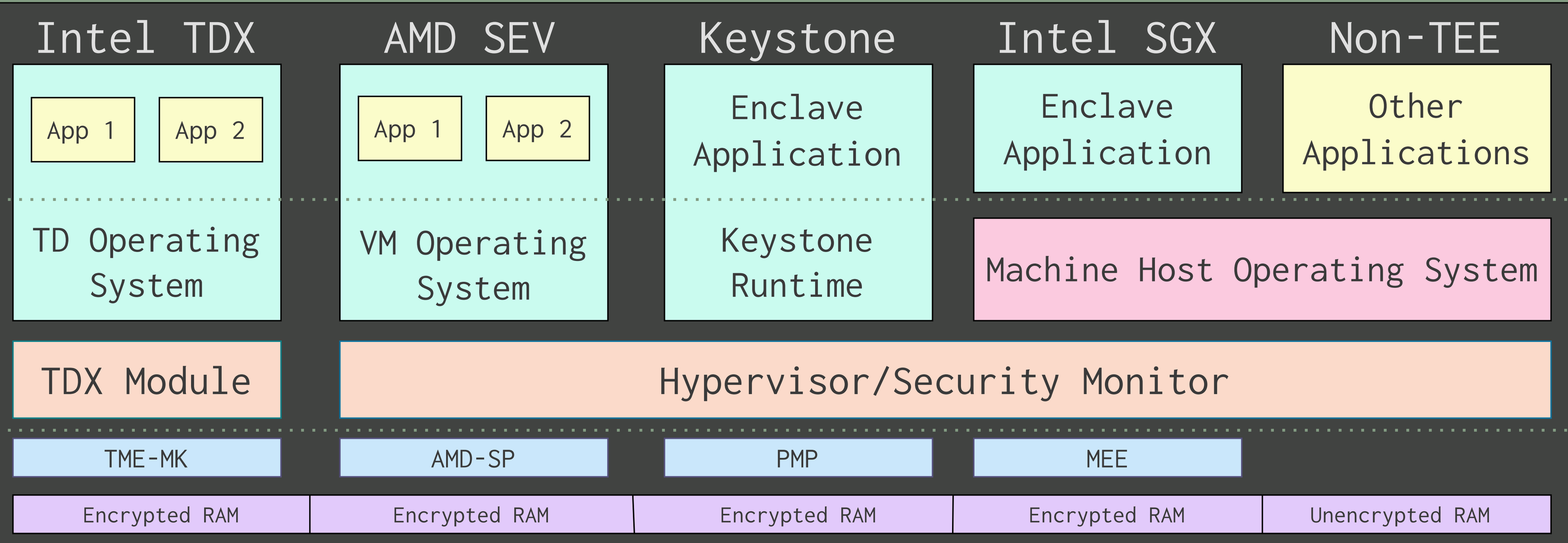
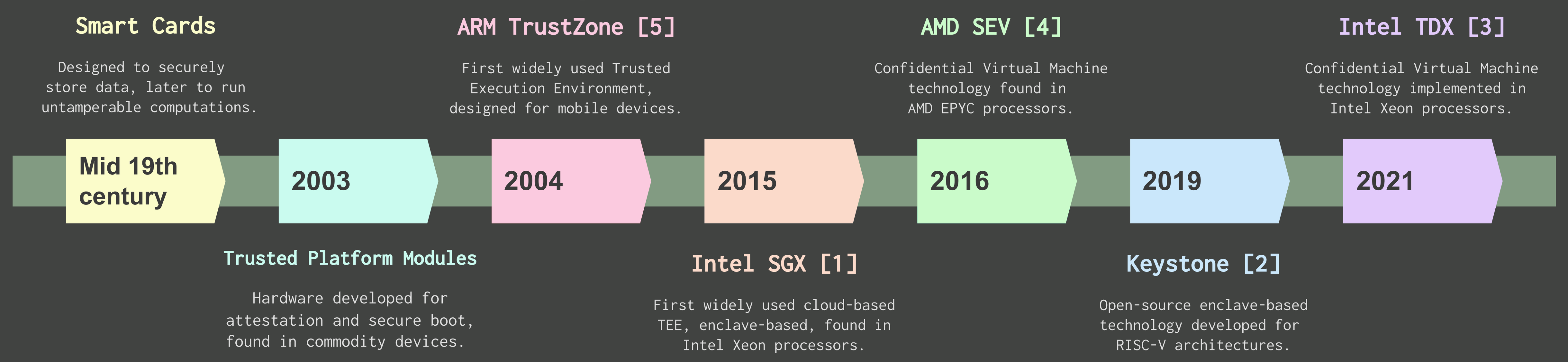


Figure 1: Discussed Trusted Execution Environments Designs

3. Technologies: Past to Present



4. Characteristics

Functionality - How restrictive is the technology, and can it perform these features?
Efficiency - What is the performance overhead compared to the non-isolated option?
Security - How large is the trusted computing base, and how many vulnerabilities were there in the past?
Usability - How practical is the system from the developer's perspective?

	Functionality		Efficiency		Security		Usability	
	Attestation	Sealing	Speed	Integrity Overhead	TCB Size	CVEs	Deployment	Debugging
Intel TDX	✓	✗	Near-native	MACs	Largest	10	Minimal OS mods	On-TD and Off-TD
AMD SEV	✓	✗	Near-native	None, added RMT in SNP	Large	49	No mods	API-based
Intel SGX	✓	✓	Near-native with SGXv2	EPC extras MACs	Small	48	SDK or Gramine	Release and Debug Mode
Key stone	✓	✓	Near-native	Optional in SW or HW	Smallest (flexible)	0*	SDK or RISC binaries	✗

Table 1: Key properties of selected Trusted Execution Environments (TEEs)

	Computation	Parties	Performance	Threat Model	Leakage	Found in	Use Cases
TEE	Any (Code)	Client-Server Attestation	Near-native, except I/O	Malicious, full software	Plaintext CPU RAM patterns	Cloud servers optional	Data analysis Trusted AI
FHE	Any (Circuit)	Client-Server	Expensive KeyGen	IND-CCA2 threat	None	Open-source libraries	Medical data ML training
MPC	Any (Circuit)	Client-Server Distributed	Slow, const** scaling	Semi-honest Malicious	Ideally none	Distributed protocols	Auctions DNA analysis
ORAM	Any query	Client-Server	Slow, log scaling	Semi-honest Malicious	Side-channels	Secure processors	Signal ObliDB
StE	Some queries	Client-Server	Fast, log access	Semi-honest	Volume size RAM patterns	Encrypted databases	MongoDB

Table 2: Comparison criteria of privacy-preserving computation technologies and TEEs

*Issues or the GitHub repository can be found
**At least constant, can be linear

5. Comparison

FHE - Strongest in security, not practical yet for general use cases.
MPC - Best in distributed environments and flexible, with offline setup overhead.
ORAM - Great at hiding access patterns, limited to querying, and can be slow.
STE - Efficient, good in I/O access, but only for querying data.
TEE - Easy to integrate, efficient, but exposed to powerful attacks.

6. Conclusions

- There is no "silver bullet" - properties must be balanced based on use cases.
- CVMs have a better performance and ease of use at the cost of a larger TCB than enclave-based counterparts.
- TEEs outperform the other techniques, but are less secure, shifting the trust to the vendor.
- MPC and FHE ”will inevitably become attractive” [6, p. 138] with stronger hardware.

7. Future Work

- Discuss more properties, such as live migration or physical attacks.
- Benchmarking framework among the five techniques to measure equal uses.
- Joining techniques with TEEs and test efficiency, security, functionality.

[1] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative Technology for CPU Based Attestation and Sealing. In Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy, volume 13. ACM New York, NY, USA, 2013
[2] Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanovic, and Dawn Song. Keystone: An Open Framework for Architecting Trusted Execution Environments.In Proceedings of the Fifteenth European Conference on Computer Systems, EuroSys’20, 2020.
[3] Intel. Intel Trust Domain Extensions Whitepaper. Technical report, Intel Corporation, 2020. Accessed: 2025-05-22.
[4] David Kaplan, Jeremy Powell, and Tom Woller. AMD Memory Encryption. White paper, 13:12, 2016.
[5] Sandro Pinto and Nuno Santos. Demystifying ARM Trustzone: A Comprehensive Survey. ACM computing surveys (CSUR), 51(6):1-36, 2019.
[6] Dominic P Mulligan, Gustavo Petri, Nick Spinale, Gareth Stockwell, and Hugo JM Vincent. Confidential Computing—a Brave New World. In 2021 international symposium on secure and private execution environment design (SEED), pages 132-138. IEEE, 2021.
[7] Mengyuan Li, Yuheng Yang, Guoxing Chen, Mengjia Yan, and Yinqian Zhang. Sok: Understanding Design Choices and Pitfalls of Trusted Execution Environments. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pages 1600-1616, 2024.

Author: Vlad-Stefan Popescu <popescu-9@student.tudelft.nl> **Responsible Professor**: Lilika Markatou