Journal of Technological Information, Management & Engineering Sciences (2026),7(1), 16-29

DOI : https://doi.org/10.5281/zenodo.23994341

Hardware-Based Security Mechanisms for Protecting Computer Systems: A Structured Narrative Review of Trusted Execution, Attestation, and Memory-Safety Hardware

[1] **Danish Iqbal**
[2] **Areeba Khan**

How to cite this article:

Iqbal, D., & Khan, A. (2026). Hardware-Based Security Mechanisms for Protecting Computer Systems: A Structured Narrative Review of Trusted Execution, Attestation, and Memory-Safety Hardware. *Journal of Technological Information, Management & Engineering Sciences, 7(1), 16–29.*

Received	Accepted	Published
8 February 2026	10 April 2026	10 June 2026

Abstract

Security mechanisms that are embedded in processors and platforms are driven by the fact that software-only defenses have been undermined by compromises in the operating system, hypervisor, or application. This paper is a structured narrative review of the hardware based security mechanisms for computer system security, from eleven verifiable sources published from 2018 to 2026. Sources consist of surveys of trusted execution environments (TEEs) and remote attestation, Arm TrustZone, Intel SGX research and attack papers. The purpose of the review was to categorize the principal mechanisms, review the protection and reported limitations, and analyze documented attacks. The level of granularity, support for attestation, protection from physical adversaries, and other mechanisms vary widely. Several designs have been based on the assumption that is now been proven to be incorrect due to microarchitectural and memory-interposition attacks, and developer practice has added to this risk. CHERI provides excellent memory-safety guarantees with workload-dependent overheads that range from nearly nonexistent to 1.65×. The synthesis enables threat modeling for selecting mechanisms, defense in depth, and integrity and freshness protection for memory in physically exposed deployments.

Keywords: *hardware security, trusted execution environment, remote attestation, confidential computing, memory safety and side-channel attacks.*

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan.
Corresponding Email : daniiiqbal432@gmail.com
[2] Department of Computer Science, COMSATS University Islamabad, Pakistan.
Email : areebakhan@yahoo.com

Copyright (c) 2026 Author(s)

Introduction

Background

Traditional security mechanisms (access control, scanning for malware, hardening of operating systems) run on the same processor and memory that are being secured. Such controls provide limited assurance when the operating system or hypervisor is compromised, or when a cloud host cannot be trusted. To solve this issue there are hardware-based security measures which embed security in the processor, the memory subsystem, or dedicated security devices. The introduction of trusted execution environments (TEEs) that allow software programs to execute with stronger guarantees of privacy and integrity without a trusted operating system is said to be a major decrease in the attack surface for powerful attackers, such as Intel SGX, AMD SEV, Arm TrustZone, and RISC-V designs (Ménétrey et al., 2022).

A hardware-based security includes several different families: attestation and roots of trust to enable remote parties to verify a platform's code and data integrity (Maene et al., 2018), isolation mechanisms to protect code and data from privileged software (Pinto & Santos, 2019; Zheng et al., 2021), and memory-safety extensions to ensure that code and data is kept safe from pointer and bounds checks in silicon (Sun et al., 2025). But it's not just the software that can be attacked. In 2019, Kocher et al. demonstrated that speculative execution, a chip optimization employed by Intel, AMD and ARM processors, can be exploited to leak out confidential memory, thus compromising the security postulates around which numerous software protections rely.

The problem that is presented and the significance of the problem.

The information about hardware protection is distributed throughout technology-specific surveys, attack papers and performance studies, and it's often not obvious what each mechanism provides and what it doesn't. Many organisations are lured into believing that hardware isolation is the ultimate protection, when in fact it may not be what the threat model of a mechanism presumes. This paper

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

brings together the most important recent evidence to enable system designers, researchers and students to compare mechanisms on a common basis. The synthesis is important because the sensitive data and code required by confidential computing, mobile payments, and embedded devices are increasingly being protected by these mechanisms.

Literature Review

Roots of Trust and Attestation

The Trusted Computing project provides assurances to the user and remote parties to software and hardware that it relies on. Maene et al. (2018) survey the field and present twelve attestation and isolation architectures from academia and industry that are implemented in hardware. In Remote Attestation, a relying party is able to verify the integrity and authenticity of software prior to trusting it. The key idea of hardware-based attestation is a root of trust, which can be tamper-resistant hardware including trusted platform module (TPM), physical unclonable function (PUF) using unique manufacturing variations in hardware, or a secret embedded in the processor die (Ménétrey et al., 2022).

The use of the Trusted Execution Environments in Commodity Processors

A TEE protects critical resources of a system, for instance regions of memory, from access by more privileged, but less trusted, software, such as the kernel (Ménétrey et al., 2022). Intel SGX enclaves are built within user space processes; Zheng et al. investigated 128 papers related to Intel SGX, in terms of deployment, attacks, and enhanced security. In this paper, Arm TrustZone divides a processor into secure and normal worlds; Pinto and Santos (2019) survey academic and industrial TrustZone-based TEEs and hardware-assisted virtualizations. AMD SEV is designed to protect virtual machines from untrusted hypervisors, reducing the developers' burden when creating virtualized systems, but increasing the size of the trusted computing base due to the inclusion of the guest OS (Ménétrey et al., 2022). There are other open RISC-V proposals including Keystone, Sanctum, TIMBER-V, and LIRA-V offering additional design points (Ménétrey et al., 2022).

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

Memory-Safety Hardware

Another line of hardware security addresses memory-safety issues. CHERI (Capability Hardware Enhanced RISC Instructions) is an emerging hardware solution designed to provide strong spatial and temporal memory protections, and the overheads of the CHERI solution have been evaluated on real applications using the CHERI-enabled Arm Morello platform (Sun et al., 2025).

Trading in hardware protections and attacks against them.Trade in hardware protections and attacks thereof.

Muñoz et al. (2023) give a comprehensive classification of vulnerabilities in TEEs based on the literature, public exploits, and vulnerability databases and attribute them to design flaws. More recent work focuses on the memory bus. De Meulemeester et al. (2025) demonstrated that an attacker with a short term physical exposure to a memory module's SPD chip, can cause the creation of aliases in the physical address space and exploit access control mitigations implemented by the CPU. Next, De Meulemeester et al. (2026) constructed a low-cost DDR4 interposer for aliasing attacks at runtime, which tricks the vendor into executing its aliasing checks.

Research Gap

Three gaps emerge. The first is that major surveys are restricted to a specific piece of technology or function (e.g., SGX, Zheng et al., 2021; TrustZone, Pinto & Santos, 2019; attestation, Ménétrey et al., 2022), thus limiting cross-family comparison of protections and limitations. Second, most surveys precede the most recent physical memory-interposition attacks (De Meulemeester et al., 2025, 2026) and their impact on threat models has not been combined with classic side-channel results. Third, there are multiple venues where evidence of developers' use of TEEs, and of the performance cost of hardware memory safety, is made available. It is therefore appropriate to consolidate a reading which compares the mechanisms to a shared set of criteria.

Objectives

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

The aim of this paper is to gather recent evidence of hardware based security mechanisms for protecting computer systems. Specific objectives are:

To describe the main hardware-based security mechanisms and the security that they offer.

2. To evaluate mechanisms of threat-model coverage and reported limitations.

To study reported attacks/misuse, such as recent physical and microarchitectural attacks.

4. To draw design and deployment conclusions, implications and research directions.

Methodology

Research Design

A secondary-research design, structured narrative review, identified and integrated findings from published sources was used in this study. The design is appropriate for a field where the technologies are different, and there is no shared experimental benchmark. Not a systematic review and not based on PRISMA procedures. No new experiments were performed.

Identify search strategies and sources.Identify search strategies and sources.

Web search was carried out in October 2026 with a generic web search engine, which displays pages of publishers, proceedings websites, institutional repositories and preprint servers. There were search strings that included terms like trusted execution environment, Intel SGX, Arm TrustZone, remote attestation, CHERI, memory aliasing, and confidential computing attacks. For each source retained, bibliographic details (authors, venue, year and identifiers) were verified from a publisher, conference, repository or university record.

Eligibility Criteria

Sources were selected based on the following criteria: (a) they were on the topic of hardware-based security for general-purpose or embedded computer systems, (b) a survey, peer-reviewed technical study, or attack paper, (c) published during the 2018-2026 time frame, and (d) accessible abstract or full text. The trade-press

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

report was only kept because it was a response to a related attack. The set is not intended to represent all the sources that were available; there are 11 sources.

Interpretation and summarization of data.

The mechanism studied, the protection provided, the threat model and the reported limits for each source were extracted and coded. The sources were then divided into six mechanism families and compared qualitatively (Table 1). The use of quantitative pooling was not appropriate as evaluations were conducted using various workloads, platforms, and threat models. Figures are reported as published and have not been computed.

Ethical Considerations

The review did not require human participants, nor did it involve any published secondary sources, thus ethical approval was not necessary. No procedures are reproduced in attack papers, which are discussed at a level of published findings.

Results

The characteristics of the Reviewed Sources are described.The Reviewed Sources are characterized.

The eleven sources include five surveys/reviews (Maene et al., 2018; Pinto & Santos, 2019; Zheng et al., 2021; Ménétrey et al., 2022; Muñoz et al., 2023); one empirical study of developer practice (Niu et al., 2025); one performance evaluation (Sun et al., 2025); three attack papers (Kocher et al., 2019; De Meulemeester et al., 2025, 2026); and one trade-press report (The Hacker News, 2025). Information was collected from abstracts for the majority of sources, except for Ménétrey et al. (2022) which was read.

Family of mechanisms: Protection and limitation.

Table 1 gives a summary of the six mechanism families. First, the SGX enclaves are isolated inside an application process, while the TrustZone partitions the entire system into two worlds and the SEV protects a whole virtual machine (Ménétrey et al., 2022). They are second in attestation. According to Ménétrey et al. (2022), SGX is a full-fledged built-in remote attestation protocol for different independent trusted applications, while TrustZone and native RISC-V do not offer any way to

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

attest software in the trusted environment and require community-developed alternatives, but TrustZone-M supports the root of trust that assists these alternatives. The extensions that are available for RISC-V are very different.

Table 1

A comparison of different hardware-based security mechanisms.A comparison of different hardware-based security mechanisms.

Family of mechanism: Core protection: Principal limitation reported: Sources:

Anchors identity and measurements in hardware to allow remote parties to verify software – hardware root of trust and attestation (TPM, PUF, fused keys) – attestation support is uneven: there is no built-in support for attestation in the trusted environment for native RISC-V and TrustZone-A.Anchors identity and measurements in hardware to allow remote parties to verify software – hardware root of trust and attestation (TPM, PUF, fused keys) – there is no built-in support for attestation in the trusted environment in native RISC-V and TrustZone-A.

Process-level enclaves (Intel SGX) Isolates enclave memory from the operating system and hypervisor; built-in remote attestation; Scalable SGX has no replay protection or integrity against hardware attacks; applications need to be partitioned into trusted and untrusted parts Zheng et al. (2021); Ménétrey et al. (2022)

World based isolation (Arm TrustZone) Divides the processor into secure and normal world TrustZone-A has no built-in attestation, limited memory (few MB) is available, vulnerabilities are enumerated Pinto & Santos (2019); Muñoz et al. (2023); Ménétrey et al. (2022)

VM-level isolation (AMD SEV family): Protected all virtual machines from untrusted hypervisors Increased trusted computing base; physical memory-aliasing attacks demonstrated against AMD SEV-SNP Ménétrey et al. (2022); De Meulemeester et al. (2025, 2026)

Open RISC-V TEEs (Keystone, Sanctum, TIMBER-V, LIRA-V) PMP- or memory-tag-based isolation via open designs Differing designs in terms of attestation and support for root of trust Ménétrey et al. (2022)

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

Capable-based spatial and temporal memory protection Overheads from negligible to 1.65× across 20 applications, highest for pointer-intensive workloads Sun et al. (2025)

Note. PUF = physical unclonable function, PMP = physical memory protection, TEE = trusted execution environment, TPM = trusted platform module.

If you're looking for memory protection against physical adversaries, this is the ideal option.

A recurring result is protection against an attacker having physical access to the memory bus. If server processors are the target, Ménétrey et al. (2022) point out that scalable SGX can provide much more memory than client SGX, but it does not provide integrity and replay protection against hardware attacks. This design trade-off is reflected in the empirical attack papers. De Meulemeester et al. (2025) compromised the integrity guarantees of AMD SEV-SNP using physical address aliasing and De Meulemeester et al. (2026) circumvented Intel Scalable SGX and AMD SEV-SNP by using a low-cost interposer that avoids boot-time checks. The authors of the latter study conclude that their results cast doubt on assumptions in the field of encrypted-memory security, and highlight vulnerabilities in the performance–security trade-offs of existing confidential computing.

Exploits based on microarchitectural features and developer negligence.Microarchitectural attacks and developer negligence.

The following threat classes have been documented in Table 2. Isolation in software can be overcome by hardware behavior, as demonstrated by Kocher et al. (2019) who were able to read arbitrary memory from the victim's process in the processors used in billions of devices, including Intel, AMD, and ARM. Another factor to consider is misuse: Niu et al. (2025) analyzed 241 open-source projects that use SGX or TrustZone, and found that 35.6% of SGX projects and 27.1% of TrustZone projects re-implement standard cryptographic operations instead of using the official SDK APIs.

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

Table 2

This document contains information on the Threat Classes against which documented Hardware-based protections are offered.

Threat Class Example (source) Affected Hardware Reported effect

Speculative-execution side channel Spectre (Kocher et al., 2019) Intel, AMD, and ARM processors Leaks victim memory; violates process assumptions (process separation, containerization, and JIT compilation)

Physical memory aliasing BadRAM (De Meulemeester et al., 2025) AMD SEV-SNP; DDR4 and DDR5 modules Brief physical access to SPD chip creates address aliases, which bypass access control of the CPU and violate integrity guarantees.

Battering RAM (De Meulemeester et al., 2026) - Low-cost interposer tamper with address lines during runtime, prior to boot-time firmware mitigations.Battering RAM (De Meulemeester et al., 2026) - Low-cost interposer, tamper with address lines at runtime before boot-time firmware mitigations.

35.6% of SGX and 27.1% of TrustZone projects reimplement standard crypto instead of official SDK APIs, due to developer misuse

Note. Descriptions are provided in the context of abstracts or full text of the cited source(s).

I would love to hear your thoughts on hardware memory safety, and what you think the practical limitations would be.

Sun et al. (2025) tested 20 C/C++ applications on the CHERI-enabled Arm Morello platform, such as SPEC CPU2017, a SQL database engine, a JavaScript engine, and a large language model inference framework. The overhead of CHERI was negligible in some workloads, and 1.65x in others, with the most significant overheads being observed in pointer-intensive workloads and those where memory is a large part of the workload. The authors estimate that moderate improvements in the microarchitecture can bring down these costs significantly. Some practical limitations also impact TEEs: most SGX implementations restrict enclave page cache to 93.5 MB before paging negatively affects the latency;

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

legacy code can be difficult to port due to the fact that the reference TrustZone-A trusted OS does not have a POSIX API and only provides a few MB of trusted memory (Ménétrey et al., 2022).

Discussion

The Principal findings are presented below.The Principal findings are presented, as follows.

Three patterns emerge. Trust is not eliminated by hardware protection, it's simply moved. Enclave and VM isolation are intended to minimize the dependence on the hypervisor and operating system, which is transferred to the processor vendors, the processor firmware and to closed components, such as the SGX quoting enclave, microcode and XuCode, which are closed source (Ménétrey et al., 2022). Secondly, the threat model is more important than the name. A countermeasure against a malicious hypervisor might not be sufficient to protect against a physical adversary; indeed, the interposer results demonstrate this (De Meulemeester et al., 2025, 2026). Reporting on the related TEE in trade press.Physical attacks are beyond the scope of SEV-SNP, according to Fail attack, and Intel responded similarly, The Hacker News (2025). The scope limitation is therefore not considered an edge case, but a design constraint, for designers of cloud, edge and embedded deployments where physical access to the hardware is possible.

Third, as well as being correct in design, the use must be correct as well. There is a wide variation in the support for attestation, which is demonstrated by Ménétrey et al. (2022) and a high frequency of developers reimplementing cryptography rather than using an API that has been vetted, as shown by Niu et al. (2025). Such practices may be encouraged by the development constraints mentioned in TrustZone-A and SGX, but the sources reviewed do not prove that these constraints cause such practices. Finally, the CHERI evidence demonstrates that hardware-enforced memory safety is achievable at workload dependent cost (Sun et al., 2025), as opposed to hardware isolation of software from a hostile host.

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

What are the implications for System Design and Deployment?What are the implications for System Design and Deployment?

The evidence reviewed provides evidence for five practical directions. First, choose the explicit threat model with which you want to use your hypervisor: hypervisor-level, physical-level, and/or application-level memory bugs require different hardware. Second, if physical access is possible, it is preferable to have designs with memory integrity and freshness, and when these are absent, to consider them inappropriate to that threat. Third, attach something to the hardware root of trust, and then verify it before provisioning secrets (Ménétrey et al., 2022). Fourth, employ vendor-supported cryptographic APIs instead of custom implementations (Niu et al., 2025). Fifth, stack mechanisms on top of each other, e.g. memory-safety hardware for application code and TEEs for isolation; attacks of the Spectre variety demonstrate that a single mechanism is not enough (Kocher et al., 2019).

Limitations

This review is a limited review. It is narrative, non-exhaustive and the search was conducted with open web sources, not through a comprehensive selection of subscription databases. For most sources, only abstracts were examined. The other source is trade press and used only for vendor statements. The attacks are dynamic and some may have already been partially mitigated by other sources and not be in this field in 2025 or 2026. Last but not least, the diversity of platforms and threat models made quantitative comparison impossible, and the review does not feature any experimental work conducted by the authors.

Guidance for future research

Further research is needed on memory encryption that offers integrity and freshness at server scale while maintaining an acceptable overhead; current scalable designs compromise on these properties while maintaining capacity. Vendors could provide a standardized threat model documentation to enable comparisons of claims by adopters. Empirical research on TEE usability and developer tooling to facilitate its use and usefulness might minimize misuse. It

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

would be nice to have some quantitative evidence from the literature that does not exist: controlled comparisons of TEEs and memory-safety hardware with common benchmarks and threat models.

Conclusion

The aim of this structured narrative review was to compare the hardware-based security mechanisms of the families themselves, based on eleven recent sources. Root of trust, enclaves, world- and VM-level isolation, open RISC-V designs, and capability-based memory safety each provide protection to a different asset from a different adversary and have different properties when it comes to attestation support, isolation granularity and resistance to physical attacks. Speculative-execution and memory-interposition attacks demonstrate that hardware guarantees are based on explicit threat model assumptions, and developer practice can erode them further. Hardware memory safety is very protective, with a workload-dependent cost. Therefore, system designers need to select mechanisms based on the threat they are facing, require integrity and freshness when there is a possibility of physical access, anchor attestation in hardware, and layer defenses. This paper is a secondary synthesis of a quickly moving field and as such the conclusions are tentative and ought to be updated as new attacks and mitigations become available.

Conflict of Interest

The authors declare that they have no financial, professional, technological, commercial, institutional, or personal conflicts of interest that could have influenced the content, interpretation, or conclusions presented in this article.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

Data Availability

No new clinical, experimental, or patient-level datasets were generated or analysed during this study. All information discussed in the review was obtained from the published literature and publicly available sources cited in the manuscript. Additional information relating to the review may be obtained from the corresponding author upon reasonable request.

Informed Consent

Not applicable. No patients or other human participants were recruited, and no identifiable personal or clinical information was collected or analysed for this review.

Use of Generative AI

The authors declare that no generative artificial intelligence tools were used to conduct the literature review, generate the scholarly content, interpret the evidence, or formulate the conclusions presented in this article. The discussion of generative AI, artificial intelligence, and machine-learning technologies in the manuscript relates solely to their applications in healthcare resource optimization and does not constitute their use in preparing the article.

References

De Meulemeester, J. et al. (2026). Dynamic memory aliasing is a low-cost method of attacking confidential computing via interposers called battering RAM. In IEEE Proceedings of the 47th IEEE Symposium on Security and Privacy (S&P 2018). 4391–4407). IEEE.

De Meulemeester, J.; Wilke, L.; Oswald, D.; Eisenbarth, T.; Verbauwhede, I.; Van Bulck, J. (2025). BadRAM: Practical Memory Aliasing Attacks on Trusted Execution Environments. In 2025 IEEE Symposium on Security and Privacy (SP). IEEE. https://doi.org/10.1109/SP61157.2025.00104

Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., & Yarom, Y. (2019). The Spectre attacks: An attack exploiting speculative execution. This is a part of the

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

proceedings of IEEE Symposium on Security and Privacy (SP) 2019 (pp. 1–19). IEEE. https://doi.org/10.1109/SP.2019.00002

Maene, P., Götzfried, J., de Clercq, R., Müller, T., Freiling, F., & Verbauwhede, I. (2018). Isolation and attestation using trusted computing hardware architectures. This research is published in IEEE Transactions on Computers (IEEE 67(3): 361–374). https://ieeexplore.ieee.org/document/7807249

Ménétrey, J., Göttel, C., Khurshid, A., Pasin, M., Felber, P., Schiavoni, V. & Raza, S. (2022). Trusted execution environment attestation explained. In 22nd International Conference on Distributed Applications and Interoperable Systems (DAIS 2022). Springer. https://doi.org/10.1007/978-3-031-16092-9_7

Muñoz, A., Rios, R., Roman, R., & Lopez, J. (2023). An analysis of the security level of the trusted execution environment (TEE) ecosystem. Computers & Security, Article 103180. https://doi.org/10.1016/j.cose.2023.103180

Niu, Y., Shi, J., Han, R., Liu, Y., Ma, C., Lyu, Y., & Lo, D. (2025). How developers misuse trusted execution environments—and it's not as complicated as it seems [Preprint]. arXiv. https://arxiv.org/abs/2512.17363

Pinto, S., & Santos, N. (2019). A comprehensive survey of Arm Trustzone. ACM Computing Surveys, 51(6), Article 130. https://doi.org/10.1145/3291047

Sun, X., Singer, J., & Wang, Z. (2025). Sweet or sour CHERI: On the Arm Morello platform. In 2025 IEEE International Symposium on Workload Characterization (IISWC), 423–438). IEEE. https://doi.org/10.1109/IISWC66894.2025.00042

The Hacker News. (2025, October 28). New TEE.Thehackernews.com publishes the following article:

Zheng, W., Wu, Y., Wu, X., Feng, C., Sui, Y., Luo, X., & Zhou, Y. (2021). Introduction to Intel SGX and its uses. Frontiers of Computer Science, 15(3), Article 153808. https://doi.org/10.1007/s11704-019-9096-y

[1] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan

[2] Department of Computer Science, COMSATS University Islamabad, Pakistan

Copyright (c) 2026 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.