Lessons Learned from Blockchain Applications of Trusted Execution Environments and Implications for Future Research

Rabimba Karanjai
University Of Houston
TX, USA
rkaranjai@uh.edu

Lei Xu
University of Texas Rio Grande Valley
Texas, USA
xuleimath@gmail.com

Lin Chen
Texas Tech University
Texas, USA
Lin.Chen@ttu.edu

Fengwei Zhang
Southern University of Science and Technology (SUSTech)
Shenzhen, China
zhangfw@sustech.edu.cn

Zhimin Gao
Auburn University at Montgomery
Alabama, USA
mtion@msn.com

Weidong Shi
University Of Houston
TX, USA
wshi3@uh.edu

ABSTRACT

Modern computer systems tend to rely on large trusted computing bases (TCBs) for operations. To address the TCB bloating problem, hardware vendors have developed mechanisms to enable or facilitate the creation of a trusted execution environment (TEE) in which critical software applications can execute securely in an isolated environment. Even under the circumstance that a host OS is compromised by an adversary, key security properties such as confidentiality and integrity of the software inside the TEEs can be guaranteed. The promise of integrity and security has driven developers to adopt it for use cases involving access control, PKS, IoT among other things. Among these applications include blockchain-related use cases. The usage of the TEEs doesn't come without its own implementation challenges and potential pitfalls. In this paper, we examine the assumptions, security models, and operational environments of the proposed TEE use cases of blockchain-based applications. The exercise and analysis help the hardware TEE research community to identify some open challenges and opportunities for research and rethink the design of hardware TEEs in general.

CCS CONCEPTS

• **Software and its engineering → Distributed systems organizing principles**.

KEYWORDS

Blockchain Trusted Execution Environment Survey Crypto-currency.

ACM Reference Format:

Rabimba Karanjai, Lei Xu, Lin Chen, Fengwei Zhang, Zhimin Gao, and Weidong Shi. 2021. Lessons Learned from Blockchain Applications of Trusted Execution Environments and Implications for Future Research. In *Workshop on Hardware and Architectural Support for Security and Privacy (HASP '21), October 18, 2021, Virtual, CT, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3505253.3505259

HASP '21, October 18, 2021, Virtual, CT, USA

ACM ISBN 978-1-4503-9614-1/21/10…$15.00
https://doi.org/10.1145/3505253.3505259

1 INTRODUCTION

Modern computer systems tend to rely on large trusted computing bases (TCBs) for operations. Typically, a TCB includes complex software components such as the OS kernel, device drivers, privileged services, and libraries. Systems built on top of a bloated TCB tend to be more vulnerable to exploits and have large attack surface compared with the systems with small-sized TCBs. Such systems including the application software sitting on top of the platforms are more likely to be exposed to attacks, opening more doors for potential security deficiencies and effective exploitation.

To address such TCB bloating problems, hardware vendors have developed mechanisms to enable or facilitate the creation of a trusted execution environment (TEE) in which critical software applications can execute securely in an isolated environment. Even under the circumstance that a host OS is compromised by an adversary, key security properties such as confidentiality and integrity of the software inside the TEEs can be guaranteed.

The security properties of a TEE implemented as hardware features protect the confidentiality and integrity of computations that take place inside it. In addition, to enforce isolated execution, a TEE abstraction often defines support for basic security operations such as remote attestation for verifying that a piece of software is executed by a genuine CPU using a TEE, secure provisioning of code and data (including cryptographic keys) into the TEE, and trusted communication channels for retrieving computing results and outputs from the TEE. These security properties make the hardware TEEs suited for a wide range of applications.

Among these applications include blockchain-related use cases. Blockchain-based applications of the TEEs are somewhat unique because of the operational environments (often decentralized and open to the public as a computing infrastructure) and security requirements. There are clear financial motivations for adversaries to attack TEE based blockchain environments. Leveraging the security properties like isolation and attestation, one can construct trusted nodes for blockchain based applications. Recognizing the TEE's potential, researchers have proposed to apply the TEEs to support various blockchain components and functionalities including distributed consensus protocols, smart contract execution, exchange

services, payment network, oracle service provider, cryptocurrency wallet, etc. For instance, distributed consensus protocol design can leverage the TEEs to improve performance [18], and one can build reliable oracle services to introduce external information into a blockchain [47].

In this paper, we analyze the blockchain use cases of the TEEs and identify the security models of these use cases. The goal is to take the lessons learned from these use cases and systems, and rethink the TEE designs, security requirements for the TEEs, and targeted operational environments. Although we focus on blockchain applications in this paper, the same exercise can be applied to other blockchain applications. The contributions of this work include: (i) We summarized some of the TEE use cases in the construction of blockchain-based systems and applications, and analyzed their security models; (ii) We discussed the challenges of utilizing the TEEs in these applications; and (iii) We identified future TEE research directions and opportunities to enhance the hardware TEEs to meet the security and operational needs of applications.

2 COMMERCIAL HARDWARE TEE PLATFORMS

In this section, we briefly describe the popular commercial TEE platforms. Readers who are knowledgeable of these platforms can skip this section.

2.1 Intel SGX

Intel Software Guard Extensions (SGX) [17, 29] is the instruction set architecture that enables the creation of an enclave, a TEE. SGX enclaves are isolated memory regions of code and data residing in a protected memory space called EPC (enclave page cache). Only the code within the enclave can access the sensitive content stored inside the protected memory regions. All external accesses will be denied.

For managing the protected memory regions, EPC is split into pages of equal size. These pages can be assigned to different enclaves using the SGX instructions. SGX utilizes a hardware memory encryption engine to protect the EPC memory so that all the EPC pages stored in the main memory are encrypted. Memory access control is enforced at the hardware level where all the read and write requests to the EPC pages are checked for violation of access management and handled by the memory encryption engine. All non-enclave accesses are prohibited. The enclave entry point is predefined in a compilation phase. SGX can detect violations of memory integrity or replayed memory content. However, SGX does not hide the locations where memory updates and reads refer to, which could be exploited as a side-channel. SGX supports remote attestation [15] that allows establishing a secure communication channel between a secure enclave and a client. Remote attestation is achieved through a centrally managed Intel Attestation Service (IAS) that verifies the enclave and provides the client the assurance of enclaves' authenticity. This means to use SGX enclave, one must trust the Intel Attestation Service. To protect privacy, Intel uses a group signature-based scheme, called EPID [19] for authentication. SGX2 [45] extends the SGX instruction set to enable dynamically managed memory within an SGX enclave. It allows developers to use new SGX2 features for implementing functions such as dynamic heap management, stack expansion, and thread context creation.

SGX DCAP [39] is an extension to SGX remote attestation support. It allows third-party users of SGX to manage their own attestation infrastructure. This new feature is designed for data center operators and cloud service providers. It can potentially benefit application models working in a distributed environment. Currently, DCAP is only supported by the Intel Xeon E Processors.

2.2 AMD SEV

Recognizing the need for a secure cloud computing environment, AMD has developed its own TEE support, named Secure Encrypted Virtualization (SEV) [20, 21]. AMD SEV integrates hardware-enabled memory encryption capabilities with the existing AMD-V virtualization architecture to support encrypted virtual machines. Encrypting virtual machines can help protect them not only from certain physical attack threats (e.g., physical memory probing) but also from other virtual machines or even the hypervisor itself. It specifically targets customers of data centers and cloud computing environments.

SEV depends on AMD Secure Processor (AMD-SP) for key management and remote attestation. AMD-SP is an ARM-based processor (Cortex-A5) integrated with the AMD SoC as a dedicated security subsystem. Main memory encryption is performed via dedicated hardware in the on-die memory controllers called Secure Memory Encryption (SME). SME is a general-purpose main memory encryption mechanism integrated into the AMD CPU architecture, and it is designed to provide high performance for server workloads.

SME supports either a full memory encryption model where all of DRAM is encrypted or a partial memory encryption model where only selected regions are encrypted such as memory spaces corresponding to the guest virtual machines. To protect SEV enabled guests, AMD-SP firmware assists in the enforcement of security properties such as the authenticity of the platform, attestation of a launched guest, and confidentiality protection of the guest's data.

Platform authentication prevents adversaries from masquerading as a legitimate AMD SEV enabled platform. The authenticity of the platform is proven with its platform identity key signed by AMD. Similar to the SGX attestation service, AMD provides its own centralized service for remote attestation [2].

SEV-ES (encrypted state) offers an extended feature to protect CPU register states. During context switch (transition between a hypervisor and a guest virtual machine), the virtual machine register state will be encrypted to protect the confidentiality of the data in the guest memory. The recent extension, called SEV-SNP (securely nested paging) adds new hardware-based security protection features on top of SEV and SEV-ES [3]. SEV-SNP enables memory integrity protection to prevent attacks such as replay of encrypted data, data corruption, memory re-mapping, memory aliasing, TCB rollback, etc. Furthermore, SEV-SNP implements additional security features to protect interrupt behavior, and reduce vulnerability against certain side-channel attacks like the poison of BTB (branch target buffer) from hypervisor. A comparison between SGX and SEV can be found in [31].

2.3 ARM TrustZone

TrustZone is a security extension to the ARM architecture with modifications to the processor, memory, and I/O devices [4]. The most important architectural change at the processor level consists in the introduction of two protection domains designated by the name of: the secure world and the normal world. The memory system has been extended with TrustZone security features by adding hardware components called, TrustZone Address Space Controller (TZASC) and TrustZone Memory Adapter (TZMA). The TZASC can be used to configure specific memory regions as secure or non-secure. It partitions the main memory into different regions and manages its association with a specific world (normal or secure). For instance, applications running in the secure world can access memory regions associated with the normal world, but not the other way around. The TZMA provides a similar function but for off-chip SRAM or ROM. The TrustZone-aware Memory Management Unit (MMU) allows for each world to have its own virtual-to-physical memory address translation tables. Memory isolation is also supported at the cache level with each cache line tagged to indicate under which world that cache line access has been performed.

Due to the ARM's license model, the adoption of TrustZone depends on specific SoC implementation. Xilinx Zynq and NXP i.MX6 are examples of SoCs that provide full support for the TrustZone technology. TrustZone is an enabling technology or building block for creating ARM-based TEEs. TrustZone by itself does not implement TEE attestation. TEE implementation built on top of TrustZone can support attestation (e.g., [44]), and other security features like secure communication channel. Examples of ARM based TEEs are OP-TEE [26] and Open-TEE [28].

2.4 FPGA based TEE

FPGA (field programming gate array) provides an alternative general-purpose platform for building TEEs, which leverages the reconfigurability of the FPGA technology to support security guarantees such as confidentiality and integrity. In FPGA-based design, secure enclaves can be implemented in the programmable logic as physically isolated domains. Physical isolation is a major advantage in terms of security. Programmable logic can function as a TEE as long as security features such as attestation, memory encryption, and proper access control mechanism of memory accesses are implemented. All of them can be implemented as programmable logic modules. Typically, FPGA itself does not implement TEE attestation. PUF based attestation can be enabled by specific FPGA TEE design [22, 36]. FPGA allows general-purpose computing through the integration of a soft-core CPU.

2.5 Summary of commercial TEE platforms

In general, certain observations can be made regarding commercial TEEs.

Firstly, remote attestations often depend on centrally managed services. **Secondly**, each platform appears to target different application sectors, for instance, ARM for mobile devices (limited presence of ARM devices in the server market), AMD-SEV for data centers and cloud. Intel also has started to focus on the cloud market by introducing DCAP. **Thirdly**, the boundary between hardware support for the TEEs and software security features has not been finalized. The evolving security features of SEV, SEV-ES, and SEV-SNP provide an example. Often security primitives in the TEE hardware do not protect against side-channel attacks (e.g., memory access patterns), which the hardware vendors consider responsibility for software developers [16]. **Fourthly**, FPGA offers a unique solution for TEEs because it can physically isolate the TEEs from the untrusted components. **Fifthly**, from the portability aspect, AMD TEEs provide the best environment because existing software can be supported out-of-box. FPGA may be the least portal option because of its unique development environment.

3 USE CASE ANALYSIS

In this section, we provide a summary on some use cases of the TEEs in decentralized ledgers and blockchains. The goal is to understand how these applications apply the TEEs, the key security assumptions behind them, and the TEE functionality requirements crucial for the application use cases. It is by no means intended as a survey of TEE applications in blockchains.

Several research efforts aim to support efficient BFT protocols by leveraging the SGX enclaves (e.g., [6, 25]). The security primitives provided by the enclaves can reduce the number of replicas and/or communication phases for running BFT protocols. To minimize TCB size, in FastBFT, only cryptographic keys and secret-shares are protected by the enclaves using SGX's crypto library.

PoET [18] is a Nakamoto-style consensus algorithm [32]. It replaces the PoW by a wait time randomly generated by a SGX enclave. This elapsed time consensus assumes that the TEE has a trusted timer. To handle potential threats posed by compromised SGX enclaves, the PoET protocol uses a rate limit function based on z-test to restrict the number of blocks that a participant can publish in any particular larger set of blocks. Another SGX for distributed consensus example is Proof of Luck [30]. In Proof-of-Luck (PoL) design, participants use the SGX enclaves to perform routines to maintain a blockchain. PoL assumes an economic security model such that compromising many enclaves is prohibitively expensive. It contains a defense mechanism against scenarios that an adversary compromises a limited number of enclaves.

In [7], the authors describe private data objects and apply the SGX enclave to protect data confidentiality during smart contract execution. It enables mutually untrusted parties to run smart contracts over private data (PDOs). SGX enclaves are used to preserve data confidentiality, execution integrity and enforce data access policies. A distributed ledger verifies and records the transactions produced by the PDOs. This way, the scheme enables stateful functionalities for otherwise stateless SGX enclaves.

Obscuro [43] is a Bitcoin mixer that utilizes the SGX enclaves. Leveraging the TEE's confidentiality and integrity guarantees, Obscuro aims to ensure the correct mixing operations and the protection of sensitive information like private keys and mixing logs. For attestation, Obscuro assumes that the measurement report is distributed to a public bulletin board (can be provided by a distributed ledger). The security model assumes that an adversary can launch an attack that rewinds enclave state. To mitigate such an attack, Obscuro stores no permanent states. To defend against potential side-channel exploits, the authors mention three approaches, (i) performing address space layout randomization (ASLR) protection

inside the enclave using SGX-Shield; (ii) using non-vulnerable cryptography libraries resistant to cache side-channel attacks; and (iii) applying Zigzagger compiler to reduce control flow footprint.

TEEs are ideally suited for protecting cryptocurrency wallets. SBLWT [10] is a secure and lightweight wallet based on Trustzone for cryptocurrencies. It is more portable compared with the hardware wallet, and more secure than the software wallet. Leveraging TrustZone isolation, SBLWT protects the private key and the wallet's address from being compromised by adversaries no matter whether the host environment including the main OS is malicious or not. TEEOD enables FPGA based TEEs [37]. TEEOD implements secure enclaves in the programmable logic (PL) by instantiating a customized and dedicated security processor per application use case on demand. It can instantiate up to six simultaneous enclaves. Each enclave includes a softcore CPU and uses shared memory for communications. A Bitcoin wallet implementation is demonstrated using TEEOD.

These applications make a range of assumptions. Below, we provide a summary of common security assumptions and/or security requirements regarding the TEEs.

- It is assumed that the adversary can control everything outside the TEEs including an interface to the TEEs. Further, an adversary can manipulate input and world view to the TEEs. For the FPGA-based TEEs, an adversary can launch a bitstream re-programming attack. Almost all the TEE use cases have considered such attack scenarios.
- Certain use cases require that the TEE has a trusted real-time clock or equivalent functionality (e.g., [18, 47]). An additional requirement is that TEE is equipped with a monotonic counter which can never be decreased in value, even after the TEE is reset [18].
- Certain use cases assume that TEE can provide a means to prevent undetected replay attacks or rewinding attacks. In the case of a replay attack, an adversary saves the encrypted state of a trusted component and starts a new instance using the exact same state to reset the component. The SGX hybrid BFT design requires such protection [6]. Obscuro mixer design assumes that an adversary can launch an attack by rewinding enclave state [43]. A mitigation strategy to such attacks is to consider whether the design can be stateless, which is a strategy used by some use cases (e.g., [43]).
- A common assumption is that an adversary cannot mount a scalable attack to the TEEs (e.g., [7, 48]). Under this assumption, an adversary can only compromise a limited number of TEEs and gain full control of these machines including compromising the TEE's signing and attestation keys. This means that attackers can masquerade as the compromised TEEs and issue valid attestations using their identities. However, it is assumed that the attackers cannot arbitrarily forge enclave identities. A question is that *is it possible multiple adversaries compromise a large percentage of TEEs in a single blockchain system even individual adversary can only subvert a limited number of TEEs?*
- All existing use cases assume that the TEE hardware vendors are trusted, which includes the centrally managed remote attestation services.

4 TEE ATTACK SURFACE

TEEs are vulnerable to certain attacks. Often, it is assumed that attackers to the TEEs include both local attackers who control the hardware and software platform for hosting enclaves and remote attackers who successfully compromise a victim platform used for running enclaves and consequently gain full privileged control of the victim. In general, it is assumed that the TEE hardware vendors are trusted. At least, it may be safe to assume that the TEE manufacturers would not launch all-out indiscriminate attacks that target their TEEs. However, the risk from the TEE hardware vendors cannot be completely ignored which we will examine in the next section.

For both local and remote attackers, one can assume that the attackers can intercept the input to and the output from the TEEs. Attackers can manipulate, modify, insert, delete data communicated with the TEEs, or change the ordering of data. Furthermore, attackers can halt or reset the physical machines at any time. In case that a security model considers physical attacks, the attackers can intercept DDR bus transactions or manipulate DDR bus. Through I/O devices, attackers can issue rogue DMA transactions or hardware signals to tamper with the TEEs. Most application security models do not consider physical attacks because of the resources and skills required by such attacks.

We consider that TEE security goals include, confidentiality, integrity, attestation, protection against DoS, and privacy. Privacy is included as an independent goal to cover certain information leakage scenarios like fingerprinting the application executed by a TEE. In the literature, attacks are often reported with a focus on a specific TEE platform, for instance, SGX, TrustZone, or SEV. However, many side-channels exploits are inherent risks universal to modern high-performance micro-architecture designs thus affect almost all the TEE products (e.g., cache timing-based attacks [13, 27, 34, 46], out-of-order speculative execution and other micro-architectural features [9, 11, 14, 23]). These exploits can not only compromise both confidentiality and integrity protection of the TEEs but also attestation signing keys embedded in a TEE platform. Some recent surveys of TEE attacks and countermeasures can be found in [12, 33].

5 CHALLENGES FROM THE USE CASES

Analysis reveals certain shortcomings or uncharted areas in TEE designs and research from the use case aspects.

5.1 Reliance on centralized infrastructure and single trusted party

Most TEE solutions rely on the TEE hardware vendors' centralized services such as remote attestation. In decentralized systems and applications, these services are centralized building blocks that can make the entire system exposed to certain risks such as potential single point of failure and tampering from the hardware TEE vendors. One can assume that the hardware vendors will not engage in activities that undermine the security of their own products. However, in the unlikely event, vendors can be compelled to disclose TEE identities or perform DoS targeting specific TEEs or TEE applications. Such concern cannot be completely ignored. According

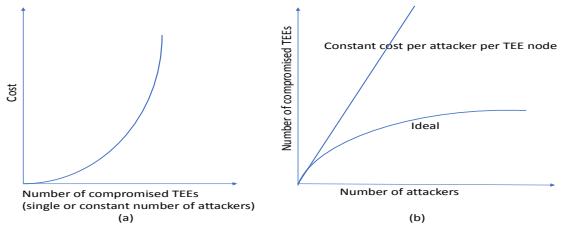

Figure 1: Economic security model of hardware TEEs: (a) single or constant number of adversaries; (b) multiple adversaries in a P2P or decentralized system.

to [42], weakness in the EPID implementation could make it possible to target specific TEEs or TEEs used in a specific application. If TEEs are hosted by a service provider like the cloud, the risk must be higher because it is not that difficult for a service provider to fingerprint a TEE application using side-channel information, for instance, network ports and traffic patterns, performance counters. Then DoS can be launched targeting specific TEE users or specific TEE based systems.

SGX DCAP has the potential to facilitate more flexible enclave attestation schemes, a step forward in reducing dependency on centralized attestation infrastructure. At this moment, it primarily focuses on the cloud and data centers. Whether it is well suited for P2P applications or decentralized systems remains to be demonstrated.

5.2 Missing economic based TEE security model

Many use cases assume that there exists an economic-based TEE security model where financial cost grows in such a pace that it prevents adversaries from achieving large-scale breaches of TEE security. Rarely, this has been a requirement for TEE design and development because such economy driven security model for the hardware TEEs has not been well understood and studied. One could assume that for a single attacker if an exploit requires significant investment in physical resources and sophisticated skills like an attack on the DDR bus, it is reasonable to assume that only a very limited number of TEEs could be compromised using such an attack. However, most documented attacks in the literature to the SGX enclaves and AMD SEV are software-based attacks, which can be launched by local attackers. So if the attackers are sufficiently motivated by financial gains (for instance, using TEEs in Proof-of-Work blockchain consensus to compute winners of block reward), a large scale attack scenario launched by many individual attackers is plausible in the real-world, even for unskilled attackers if they can download attack software and run it to compromise local TEEs that they have direct control in a P2P or decentralized system.

In summary, it is elusive to develop a quantitative economy based TEE security model. Very little research has been done. Incorporating such a model as requirements into hardware TEE design is equally challenging because of the vague assumptions and uncertainty in quantifying security properties using economic terms.

5.3 Unclear security boundary between hardware TEE and software

Similar to any computer architecture research on hardware-software interfaces, security research, and applications based on the commercial hardware TEE platforms once again illustrate the challenges in drawing a boundary in TEE security between the software and hardware. What security protections and which kinds of security primitives should be ideally implemented by a hardware TEE? What security features should primarily be software developers' responsibilities? A clear understanding of these questions is crucial not only for designing future hardware TEEs but also for developing a general hardware-software co-design framework for the TEEs. Arguments and counter-arguments often can be found from both sides involving hardware TEE designers and architects, security researchers, and application developers. If the history of computer architecture can be applied as a prediction of the future, there will never be a clearly delineated boundary. The security boundary will remain open and dynamic, sometimes heatedly debated. However, nothing will prevent researchers and developers to explore the hardware TEE design space and experiment with different options. As we have described earlier, blockchain-based applications are unique in contributing to this discussion because of their unique security requirements and decentralized operational environments.

5.4 Lack support of rapid detection and response to compromised TEEs

Many use cases and applications of the TEEs assume that compromised TEEs can be quickly detected and swiftly removed from the system. Despite a large body of published work, there is a lack of research in efficient detection of compromised TEEs and rapid response mechanisms. Likely this lack of research is partly due to the closed and centralized remote attestation services where only the vendors can remove the compromised TEEs from attestation. When the privacy of TEE identities is a requirement, the challenge will be more significant because the requirements on protection of TEE privacy and rapid detection/response to compromised TEEs may be conflicting goals in designing the TEE identity and attestation schemes. Linkable TEE signatures will facilitate rapid response to compromised TEEs. However, it may raise privacy concerns. A zero-knowledge proof based hardware TEE identity scheme can provide full privacy protection. On the other hand, it makes detection and removal of the compromised TEEs from a distributed system a much difficult task. In addition, how to prevent such detection and response mechanisms from being abused by malicious attackers, who can leverage the existence of such services for denial of service attacks. In case TEE attestation service is decentralized, one may face additional challenges such as reaching consensus on rogue TEE detection and agreements in responses. Overall, it is not entirely clear how a hardware TEE platform can balance these requirements like privacy, the operational environment, etc to realize a detection and response scheme for compromised or rogue TEEs that is rapid, privacy-preserving, flexible enough to meet the diverse operational needs based on the applications.

5.5 Lack of standards and inter-operability

A great challenge that hinders hardware TEEs from gaining broader adoption than what has been achieved so far is that there are no well-established and well-accepted standards that the vendors agree to follow. Commercial TEEs such as SGX, SEV, and TrustZone are all based on proprietary designs and implementations. This creates significant barriers to supporting interoperability across the TEEs. The research community has realized the advantages of a heterogeneous TEE environment to improve infrastructure resilience and security posture against attacks. However, many challenges remain for developing applications over a heterogeneous TEE environment. Developers may face issues like incompatible attestation schemes, heterogeneous root-of-trust, lack of formal semantic of security primitives offered by different hardware TEE vendors, etc. Moreover, micro-architecture designs of TEEs are closed from the security community as trade secrets and protected intellectual properties. This makes the task of assessing security risks and threat modeling of different hardware TEEs very difficult.

6 OPPORTUNITIES FOR ENHANCEMENT AND FUTURE RESEARCH

Despite the challenges, focused studies of application use cases, their requirements, security assumptions as well as hurdles behind them provide insights of potential topics for future research of TEE designs and implementations. These improvements if adopted could benefit a broad range of applications beyond what we have used for analysis.

6.1 New hardware TEE security primitives

There are opportunities to develop new security primitives and/or enhance the existing security primitives provided by a hardware TEE platform. Here we provide some examples based on the use case studies.

Trusted and accurate wall time clock. There is a clearly identified need for a secure TEE based wall time clock. Many factors could affect a computer's wall clock time. Variations of these factors can cause severe drift of clock time, sometimes even lead to the problem with NTP synchronization. Some of these factors depend on the motherboard chipset. In the past, control for variable speed of processors, front-side bus spread spectrum oscillators, mis-detection in TSC all could contribute to inaccurate clock time and large drift in normal situation even without an adversary. An accurate and tamper-resistant TEE wall time clock is desired by many applications (for instance, Google's noSQL database applies GPS time for data synchronization). To reduce reliance on NTP, which can be manipulated by attackers, even chip size atomic time source could be integrated with a hardware TEE platform to provide accurate and secure TEE time.

Extension of the hardware TEE to support continuous state. Another security primitive is to support non-rewindable and persistent TEE storage space for storing application states that can not be reverted. One of the uses cases of hardware backed secure storage and what it can store and utilize has been on the topic of persistent data integrity. However a lot of the security mechanisms hold up when the system is in an uninterrupted powered state. In real life however power crashes,system reboot are pretty common, making these assumptions unrealistic. State-protected modules must securely store their state. In the topic of state continuity, Intel SGX and other hardware-based TEE's are prone to state module replay attacks. This assumes that the attacker has complete control over the untrusted software stack and can replay the enclaves by manipulating execution control. Strackx et al in Ariadne [41] proposes a system that achieves state continuity and gives a solution to this problem by providing a minimal attack surface. This is done by ensuring the following three properties of rollback prevention, liveliness detection of the system and also ensuring an execution is not interrupted before it reaches end of cycle. However, this subject has not received much attention, showing mostly Ariadne [41] and Parno et al. [35] are one of the few who has proposed solutions. Because SGX enclave is stateless, many blockchain applications with such needs are forced to customize the designs to be stateless, which is less ideal and possibly may introduce new attack surface due to extra design complexity. Other applications may benefit as well from such security feature. As discussed above, alternative approaches are to, integrate other secure hardware component like a TPM together to achieve non-rewindable states for the TEEs or leverage the immutability feature of blockchains. However, these alternative approaches are far less than build-in support of non-rewindable states.

6.2 Decentralized TEE identity management and attestation

There has been an explosion of research in decentralized and self-governed identity management applying the features of blockchains. In contrast, not much has been done to extend the concept to the TEE identity. Identity management and attestation for hardware TEE platforms remain centralized and rely on the vendors' centrally managed attestation services. Existing research on remote attestation using commercial TEE platforms mainly focus developing API wrappers or proxy attestation service that can hide the heterogeneity of hardware TEE attestation. An example is ARM research's Veracruz project [5] that aims to support cloud-based IoT applications with heterogeneous TEEs. The project, under Confidential Computing Consortium [1], demonstrates some of the hurdles that one has to overcome in order to support an infrastructure of heterogeneous TEEs, for instance, how to manage diverse attestation schemes and compatibility issues in heterogeneous root-of-trust. To conclude, how to enable a truly decentralized infrastructure for hardware TEE identity management and attestation looks like a promising direction of research, specifically innovative approaches that can balance the needs of privacy, decentralization, detection and rapid response to compromised hardware TEEs.

6.3 Programmable TEE security

Various techniques such as address space randomization (both code and data), hiding of control flow patterns using compiler, software based ORAM, side-channel resistant cryptographic libraries are proposed as defense on attacks to the SGX enclaves (e.g., [8, 24, 38, 40]). Performance overhead to the TEE applications after applying some countermeasure techniques varies from less than 10% to potentially

hundred of times slow down (see [12] for a survey of countermeasures and performance overhead). In most cases, TEE applications have to be executed in single thread mode. To support efficient execution of diverse applications with different security requirements, future TEEs can make the hardware TEE platform more configurable or programmable.

A hardware TEE platform can provide a programmable security engine to the developers to implement advanced security features like address space randomization or ORAM (Oblivious RAM) in hardware. For instance, a programmable memory controller can be used to randomize memory operations, which can significantly increase the performance of the TEE applications that prefer to use address space randomization and ORAM as a defense. This can be achieved using a FPGA integrated memory controller for the TEEs. Based on the need, TEE application developers can supply a FPGA bitstream that implements certain enhanced security protection mechanism like a lightweight and customized ORAM layer. The measurement report will be extended to include configuration settings and digital signatures of the security programs applied to such memory controller. This framework can provide a more flexible security environment with high performance to meet the diverse security and performance needs of TEE applications.

6.4 Hardware TEE standards

Despite the challenges, developing TEE standards seems too important to be ignored considering the stake behind them. Both the developer and the TEE research communities perhaps would agree that the future of TEEs will not be ideal if they remain to be vendor locked and fragmented.

7 CONCLUSION

TEEs have been proved a powerful tool for constructing secure applications. Using blockchain-based TEE applications as targets, we examine and summarize the key security models and assumptions of these applications. From this perspective, the paper provides an overall picture of research challenges and opportunities in this area. The lessons learned from these use cases provide some potential directions for future research and areas for improvement, which allows the TEE researchers and designers to rethink some of the basic concepts in hardware TEE design.

REFERENCES

[1] [n.d.]. Confidential Computing Consortium. https://confidentialcomputing.io
[2] Advanced Micro Devices Inc. [n.d.]. AMD CEK Certificate Server. https://kdsintf.amd.com/cek/.
[3] Advanced Micro Devices Inc. [n.d.]. Isolation with Integrity Protection and More. https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vmisolation-with-integrity-protection-and-more.pdf.
[4] ARM Inc. 2020. TrustZone for Armv8-A. https://documentation-service.arm.com/static/602167b6873dd96c4deaf49b.
[5] ARM Research. [n.d.]. Veracruz. https://github.com/veracruz-project/veracruz.
[6] Johannes Behl, Tobias Distler, and Rüdiger Kapitza. 2017. Hybrids on Steroids: SGX-Based High Performance BFT. In *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade, Serbia) *(EuroSys '17)*. Association for Computing Machinery, New York, NY, USA, 222–237. https://doi.org/10.1145/3064176.3064213
[7] Mic Bowman, Andrea Miele, Michael Steiner, and Bruno Vavala. 2018. Private Data Objects: an Overview. arXiv:1807.05686 [cs.CR]
[8] Ferdinand Brasser, Srdjan Capkun, Alexandra Dmitrienko, Tommaso Frassetto, Kari Kostiainen, Urs Müller, and Ahmad-Reza Sadeghi. 2017. DR.SGX: Hardening SGX Enclaves against Cache Attacks with Data Location Randomization. *CoRR* abs/1709.09917 (2017). arXiv:1709.09917 http://arxiv.org/abs/1709.09917
[9] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, and Ten H. Lai. 2019. SgxPectre: Stealing Intel Secrets from SGX Enclaves Via Speculative Execution. In *2019 IEEE European Symposium on Security and Privacy (EuroS P)*. 142–157. https://doi.org/10.1109/EuroSP.2019.00020
[10] Weiqi Dai, Jun Deng, Qinyuan Wang, Changze Cui, Deqing Zou, and Hai Jin. 2018. SBLWT: A Secure Blockchain Lightweight Wallet Based on Trustzone. *IEEE Access* 6 (2018), 40638–40648. https://doi.org/10.1109/ACCESS.2018.2856864
[11] Dmitry Evtyushkin, Ryan D. Riley, Nael CSE, ECE Abu-Ghazaleh, and D. Ponomarev. 2018. BranchScope: A New Side-Channel Attack on Directional Branch Predictor. *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems* (2018).
[12] Shufan Fei, Zheng Yan, Wenxiu Ding, and Haomeng Xie. 2021. Security Vulnerabilities of SGX and Countermeasures: A Survey. *ACM Computing Survey* 54, 6, Article 126 (July 2021), 36 pages. https://doi.org/10.1145/3456631
[13] Daniel Gruss, Raphael Spreitzer, and Stefan Mangard. 2015. Cache Template Attacks: Automating Attacks on Inclusive Last-Level Caches. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 897–912. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/gruss
[14] Tianlin Huo, X. Meng, Wenhao Wang, Chunliang Hao, Pei Zhao, Jian Zhai, and Mingshu Li. 2020. Bluethunder: A 2-level Directional Predictor Based Side-Channel Attack against SGX. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2020 (2020), 321–347.
[15] Intel. [n.d.]. Attestation Service for Intel Software Guard Extensions (Intel SGX): API Documentation. https://api.trustedservices.intel.com/documents/sgxattestation-api-spec.pdf.
[16] Intel. [n.d.]. Intel SGX and Side-Channels. https://software.intel.com/en-us/articles/intel-sgx-and-side-channels.
[17] Intel. [n.d.]. Intel Software Guard Extensions. https://software.intel.com/sites/default/files/332680-001.pdf.
[18] Intel. 2017. Sawtooth Lake. https://github.com/hyperledger/sawtooth-core/tree/0-7/validator/sawtooth_validator/consensus/poet1.
[19] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. 2016. Intel software guard extensions: EPID provisioning and attestation services. *White Paper* 1, 1-10 (2016), 119.
[20] David Kaplan. 2016. AMD x86 Memory Encryption Technologies. USENIX Association, Austin, TX.
[21] David Kaplan, Jeremy Powell, and Tom Woller. 2016. AMD memory encryption. *White paper* (2016).
[22] Joonho Kong, Farinaz Koushanfar, Praveen K. Pendyala, Ahmad-Reza Sadeghi, and Christian Wachsmann. 2014. PUFatt: Embedded platform attestation based on novel processor-based PUFs. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1145/2593069.2593192
[23] Esmaeil Mohammadian Koruyeh, Khaled N. Khasawneh, Chengyu Song, and Nael Abu-Ghazaleh. 2018. Spectre Returns! Speculation Attacks using the Return Stack Buffer. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. USENIX Association, Baltimore, MD. https://www.usenix.org/conference/woot18/presentation/koruyeh
[24] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. 2017. Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 557–574. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/lee-sangho
[25] Bijun Li, Nico Weichbrodt, Johannes Behl, Pierre-Louis Aublin, Tobias Distler, and Rüdiger Kapitza. 2018. Troxy: Transparent Access to Byzantine Fault-Tolerant Systems. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 59–70. https://doi.org/10.1109/DSN.2018.00019
[26] Linaro. [n.d.]. OP-TEE. https://wiki.linaro.org/WorkingGroups/Security/OP-TEE.
[27] Fangfei Liu, Yuval Yarom, Qian Ge, Gernot Heiser, and Ruby B. Lee. 2015. Last-Level Cache Side-Channel Attacks Are Practical. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP '15)*. IEEE Computer Society, USA, 605–622. https://doi.org/10.1109/SP.2015.43
[28] Brian McGillion, Tanel Dettenborn, Thomas Nyman, and N. Asokan. 2015. Open-TEE – An Open Virtual Trusted Execution Environment. *2015 IEEE Trustcom/BigDataSE/ISPA* (Aug 2015). https://doi.org/10.1109/trustcom.2015.400
[29] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. 2013. Innovative Instructions and Software Model for Isolated Execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy* (Tel-Aviv, Israel) *(HASP '13)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1 pages. https://doi.org/10.1145/2487726.2488368
[30] Mitar Milutinovic, Warren He, Howard Wu, and Maxinder Kanwal. 2016. Proof of Luck: An Efficient Blockchain Consensus Protocol. In *SysTEX '16 Proceedings of the 1st Workshop on System Software for Trusted Execution* (Trento, Italy). ACM, 2:1–2:6. http://eprint.iacr.org/2017/249.pdf

[31] Saeid Mofrad, Fengwei Zhang, Shiyong Lu, and Weidong Shi. 2018. A Comparison Study of Intel SGX and AMD Memory Encryption Technology. In *Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy* (Los Angeles, California) *(HASP '18)*. Association for Computing Machinery, New York, NY, USA, Article 9, 8 pages. https://doi.org/10.1145/3214292.3214301

[32] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf. Accessed: 2015-07-01.

[33] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. 2020. A Survey of Published Attacks on Intel SGX. *ArXiv* abs/2006.13598 (2020).

[34] Dag Arne Osvik, Adi Shamir, and Eran Tromer. 2006. Cache Attacks and Countermeasures: The Case of AES. In *Topics in Cryptology – CT-RSA 2006*, David Pointcheval (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–20.

[35] Bryan Parno, Jacob R Lorch, John R Douceur, James Mickens, and Jonathan M McCune. 2011. Memoir: Practical state continuity for protected modules. In *2011 IEEE Symposium on Security and Privacy*. IEEE, 379–394.

[36] Jared Russell Patten. 2018. A Flexible FPGA-Assisted Framework for Remote Attestation of Internet Connected Embedded Devices.

[37] Sérgio Pereira, David Cerdeira, Cristiano Rodrigues, and Sandro Pinto. 2021. Towards a Trusted Execution Environment via Reconfigurable FPGA. arXiv:2107.03781 [cs.CR]

[38] Sajin Sasy, Sergey Gorbunov, and Christopher W. Fletcher. 2018. ZeroTrace : Oblivious Memory Primitives from Intel SGX. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_02B-4_Sasy_paper.pdf

[39] V. Scarlata, Simon Johnson, J. Beaney, and P. Żmijewski. 2018. Supporting Third Party Attestation for Intel® SGX with Intel® Data Center Attestation Primitives.

[40] Jaebaek Seo, Byoungyoung Lee, Seong-Min Kim, Ming-Wei Shih, Insik Shin, Dongsu Han, and Taesoo Kim. 2017. SGX-Shield: Enabling Address Space Layout Randomization for SGX Programs. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017*. The Internet Society. https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/sgx-shield-enabling-address-space-layout-randomization-sgx-programs/

[41] Raoul Strackx and Frank Piessens. 2016. Ariadne: A Minimal Approach to State Continuity. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 875–892. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/strackx

[42] Yogesh Swami. 2017. Intel SGX Remote Attestation is not sufficient.

[43] Muoi Tran, Loi Luu, Min Suk Kang, Iddo Bentov, and Prateek Saxena. 2017. Obscuro: A Bitcoin Mixer using Trusted Execution Environments. Cryptology ePrint Archive, Report 2017/974. https://ia.cr/2017/974.

[44] Ziwang Wang, Yi Zhuang, and Zujia Yan. 2020. TZ-MRAS: A Remote Attestation Scheme for the Mobile Terminal Based on ARM TrustZone. *Secur. Commun. Networks* 2020 (2020), 1756130:1–1756130:16. https://doi.org/10.1155/2020/1756130

[45] Bin Cedric Xing, Mark Shanahan, and Rebekah Leslie-Hurd. 2016. Intel Software Guard Extensions (Intel SGX) Software Support for Dynamic Memory Allocation inside an Enclave. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016* (Seoul, Republic of Korea) *(HASP 2016)*. ACM, New York, NY, USA, Article 11, 9 pages. https://doi.org/10.1145/2948618.2954330

[46] Yuval Yarom and Katrina Falkner. 2014. FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack. In *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, San Diego, CA, 719–732. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/yarom

[47] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 ACM sIGSAC conference on computer and communications security*. 270–282.

[48] Fan Zhang, Ittay Eyal, Robert Escriva, Ari Juels, and Robbert Van Renesse. 2017. REM: Resource-Efficient Mining for Blockchains. In *USENIX Security 17*. Vancouver, BC, 1427–1444. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/zhang