A survey of Trusted Execution Environments (TEE) use cases

Harsh Agrawal
Computer Science Dept.
Indian Institute of Technology, Delhi
India
cs1190431@cse.iitd.ac.in

***Abstract*—Nowadays, there is a trend to design complex, yet secure systems. In this context, the Trusted Execution Environment (TEE) was designed to enrich the previously defined trusted platforms. TEE is commonly known as an isolated processing environment in which applications can be securely executed irrespective of the rest of the system. Although this technology has remained relatively underground since its inception, over the past years, numerous initiatives have significantly advanced the state of the art involving TEE(s). Motivated by this revival of interest, this paper presents an in-depth study of the existing use cases of the Trusted Execution Environments covering domains ranging from industry to academia. Furthermore, we analyze the most relevant weaknesses of existing systems and try to draw a comparison between them.**

***Index Terms*—TEE, Isloated execution, TrustZone.**

I. Introduction

User requirements for security are increasingly becoming demanding. New challenges arise, since modern systems are becoming more and more complex, open and connected. Traditional security technologies can no longer meet the security requirements of such architectures. This explains the recent trend to integrate trusted computing concepts into different systems. The academia and industry are giving in their best efforts to make the world a more secure place by introducing security features in such systems.

Trusted Execution Environments (TEE) [17] is one such technology introduced for enhancing the security of systems to allow greater data protection and reduce dependability on the surrounding environment. Typically, a TEE is divided into two segments a normal world and a secure world. The normal world is unprotected and not secure, whereas the design of TEE guarantees that the applications (called enclaves) running inside the secure world are affected by the intentions of the normal world. The protection mechanisms placed inside the secure world ensure that enclaves are not only isolated from the outside world but are also isolated from each other inside the secure world. This is what makes TEE(s) so popular.

Although the technology is remarkable in providing secure execution environments, it was not put in much use in real life projects due to its closed nature. As a result, for nearly ten years, TEE(s) were mostly used by device manufacturers for monetizing proprietary secure services, the security of which was difficult to examine due to their closed nature. Yet, over the past few years, we have witnessed a growing interest in TEE(s) from both academia and industry. TEE(s) have been leveraged in many academic research projects and commercial products alike, providing the security foundations for systems.

TEE(s) have been utilized in several domains to improve the security of systems. Several implementations have been proposed for each domain that utilizes different technologies and serves a different purpose. Given that there are no conflicting tradeoffs between several implementations, they can be integrated together to serve a greater purpose and provide more secure systems than ever before. This can be possible only when we have a comprehensive list of all the implementations in a domain with associated technologies used and purpose served along with certain limitations that can be used for tradeoff analysis. We believe this approach can enable any system developer to analyse the various approaches before choosing a single one of them to allow an efficient and secure system design.

In this paper we survey [68], [53], [2], [48], [3] the use cases of TEE(s) that have been in literature and present a comparative analysis for each domain of use. We have identified the following domains in which TEE(s) were implemented (which by no means is exhaustive) and utilized for solving a problem in the domain.

- Cloud and Edge Computing
- Hypervisor and virtualization
- Mobile security
- OS and kernel security
- IoT/Embedded system security
- Trusted services
- Miscellanous

In each of the section below we list the researches that have utilised TEE in the domain and a brief description along with their advantages and disadvantages.

II. Cloud and Edge Computing

The recent edge computing infrastructure introduces a new computing model that works as a complement of the traditional cloud computing. The edge nodes in the infrastructure reduce the network latency and increase data privacy. Application of

TEEs on the edge nodes would be a natural choice to secure the computation and sensitive data on these nodes.

A. Preliminary Study of Trusted Execution Environments on Heterogeneous Edge Platforms

[38] It investigates the typical hardware assisted TEEs and evaluates the performance of these TEEs to help analyze the feasibility of deploying them on the edge platforms.

- **Advantages:** Deployment of TEEs can improve the security of computations on edge platforms. Although there is a performance overhead involved but with recent hardware assisted TEEs it would not be substantial.
- **Limitations:** The research claims that there is a certain performance overhead in deploying TEEs to edge computing platforms. This performance depends on the hardware used.

B. VC3: Trustworthy Data Analytics in the Cloud using SGX

[55] The research proposes a system that allows users to run distributed MapReduce computations in the cloud while keeping their code and data secret, and ensuring the correctness and completeness of their results.

- **Advantages:** VC3 relies on SGX processors to isolate memory regions on individual computers, and to deploy new protocols that secure distributed MapReduce computations. VC3 optionally enforces region self-integrity invariants for all MapReduce code running within isolated regions, to prevent attacks due to unsafe memory reads and writes.
- **Limitations:** VC3's average runtime overhead: 4.5% with write integrity and 8% with read/write integrity.

C. Shielding Applications from an Untrusted Cloud with Haven

[6] Haven, is the first system to achieve shielded execution of unmodified legacy applications, including SQL Server and Apache, on a commodity OS (Windows) and commodity hardware.

- **Advantages:** It introduces the notion of shielded execution, which protects the confidentiality and integrity of a program and its data from the platform on which it runs. Haven leverages the hardware protection of Intel SGX to defend against privileged code and physical attacks such as memory probes.
- **Limitations:** The trusted computing base (TCB) of Haven is substantial, because the LibOS includes a large subset of Windows. Haven does not currently prevent rollback of filesystem state beyond the enclave's lifetime. Our prototype relies on the host for system time and timeouts. However, a malicious host may lie about the time or signal timeouts early.

D. Running ZooKeeper Coordination Services in Untrusted Clouds

[9] This research adds a transparent encryption layer to ZooKeeper by means of a privacy proxy supposed to run inside a TEE.

- **Advantages:** ZooKeeper can be deployed at untrusted cloud providers, establishing confidential coordination for distributed applications. With our privacy proxy, all ZooKeeper functionality is retained while there is little degradation of throughput.
- **Limitations:** The current solution still allows the inference of usage details based on client access patterns and the znode hierarchy.

E. A Cloud based Dual-Root Trust Model for Secure Mobile Online Transactions

[29] This paper proposes a a dual-root trust online transaction model that provides a dual-root trust model including both the user's mobile device and a delegation mobile cloud using a modified CP-ABE cryptography

- **Advantages:** In this approach, most cryptographic functions are delegated from users to the mobile cloud, making it a lightweight scheme for mobile devices

F. Practical Privacy Preserving Cloud Resource-Payment for Constrained Clients

[51] This paper aims to tackle the issue of privacy in Cloud Computing by providing a method which allows clients to anonymously consume resources of a Cloud Provider (CP) such that the CP is not able to track users' activity patterns

- **Advantages:** The proposed method used prepad tokens that work as credits that are transfarable between trusted parties. Thus privacy is maintained even while buying the credits.
- **Limitations:** The test code, support libraries and JavaVM are way too large to fit into a TrustZone environment

G. DFCloud : A TPM-based Secure Data Access Control Method of Cloud Storage in Mobile Devices

[57] This research proposes DFCloud, a secure data access control method of cloud storage services based on Trusted Platform Module (TPM) to manage all the encryption keys and define a key sharing protocol among legal users.

- **Advantages:** DFCloud is secure by design as it uses client-based encryption method to guarantee that server-side data leakage cannot occur. It also performs remote attestation on each client to prevent data or credential leakages caused by malicious programs in client-side, before a data encryption key is used.
- **Limitations:** The performance overhead for transfering files with secure world encryption is quite high as compared to alternative platforms (about twice for 4 kB files).

H. ARM TrustZone for Secure Image Processing on the Cloud

[10] The research implementes Darkroom, a secure image processing service for the cloud leveraging ARM TrustZone technology. It stores image data in encrypted form and uses TrustZone-enabled ARM processors to securely process such images in isolation from the operating system

- **Advantages:** The system enables users to securely process image data in a secure environment that prevents exposure of sensitive data to the operating system
- **Limitations:** The solution adds a small overhead to image processing when compared to computer platforms that require the entire operating system to be trusted.

I. TrApps: Secure Compartments in the Evil Cloud

[8] The research proposes TrApps, a secure platform for general purpose trusted execution in an untrusted cloud with multiple isolated tenants based on the ARM TrustZone technology

- **Advantages:** The system targets the parallel execution of partitioned applications of distinct tenants with lean security-sensitive components, and is based on a minimal trusted code base in the secure world of ARM TrustZone when compared to similar systems
- **Limitations:** On average, usage of TrApps imposes an overhead of 36.9% on Secure Memcached. Equally, the network traffic of Secure Memcached is lower, as less requests are transmitted per second.

III. Hypervisor and Virtualization

A. Towards a TrustZone-assisted Hypervisor for Real Time Embedded Systems

[46] The research presents the implementation of a TrustZone-based hypervisor for real time embedded systems, which allows multiple RTOS partitions on the same hardware platform.

- **Advantages:** The approach surpasses related work by implementing a TrustZone-assisted solution that allows the execution of an arbitrary number of guest OSes while providing the foundation to drive next generation of secure virtualization solutions for resource-constrained embedded devices.
- **Limitations:** It was observed that the virtualization overhead has a slight impact on execution performance.

B. T-KVM: A Trusted architecture for KVM ARM v7 and v8 Virtual Machines Securing Virtual Machines by means of KVM, TrustZone, TEE and SELinux

[41] Trusted Kernel-based Virtual Machine (T-KVM), is a novel security architecture for the KVM-on-ARM hypervisor, is proposed to satisfy the current market trend The proposed architecture combines four isolation layers: ARM Virtualization and Security Extensions (also known as ARM VE and TrustZone), GlobalPlatform Trusted Execution Environment (TEE) APIs and SELinux Mandatory Access Control (MAC) security policy.

- **Advantages:** T-KVM integrates software and hardware components to secure guest Operating Systems (OSes) and enable Trusted Computing in ARM virtual machines. The T-KVM architecture can be implemented on platforms based on ARM v7 and v8 architectures, without requiring additional custom hardware extensions.
- **Limitations:** The different isolation layers which compose T-KVM provide high security, but at a cost of additional overhead.

C. µRTZVisor: A Secure and Safe Real-Time Hypervisor

[37] µRTZVisor is a new TrustZone-assisted hypervisor that distinguishes itself from state-of-the-art TrustZone solutions by implementing a microkernel-like architecture while following an object-oriented approach.

- **Advantages:** µRTZVisor is able to run nearly unmodified guest OSes, while, contrarily to existing TrustZone-assisted solutions, it provides a high degree of functionality and configurability, placing strong emphasis on the real-time support.
- **Limitations:** The more partitions in a system, tend to increase the latency for the response which is one of the limitation of the approach.

D. LTZVisor: TrustZone is the Key

[47] The reserach proposes a Lightweight TrustZone assisted Hypervisor (LTZVisor) as a tool to understand, evaluate and discuss the benefits and limitations of using TrustZone hardware to assist virtualization

- **Advantages:** It was demonstrated how TrustZone can be adequately exploited for meeting the real-time needs, while presenting a low performance cost on running unmodified rich operating systems.
- **Limitations:** One of the main identified limitations is related to the number of supported virtual machines. LTZVisor supports the coexistence of two VMs, one running in the secure world and one running in the non-secure world.

E. Reconciling Security with Virtualization: A Dual-Hypervisor Design for ARM TrustZone

[13] This research proposes a novel design to enable the virtualization of both secure and non-secure worlds offered by ARM platforms with TrustZone technology.

- **Advantages:** The design is based on a dual-hypervisor scheme that allows executing multiple twoworld domains in isolation, where each of them can comprise both a standard (i.e., non-secure) execution environment, and a trusted execution environment (TEE).
- **Limitations:** The major latencies introduced by the realized implementation are due to the bootstrap phase of the X Monitor, the world switches, and the crossing of the virtualization layers of the two hypervisors

F. Asymmetric Virtualisation for Real-Time Systems

[11] This research describes how an asymmetric virtualisation layer has been realised on top of the ARM TrustZone security extension, in order to support the concurrent execution of both a real-time and a general purpose operating system on the same processor.

- **Advantages:** It introduces a very small overhead because it does not require privileged instruction emulation. The

practical feasibility of this approach has been further remarked by the implementation of a VMM prototype and the overhead introduced by virtualisation has been measured with encouraging results
- **Limitations:** The only limitation posed on the operating system hosted on the non-secure side is that it can no longer use the TrustZone features by itself.

G. vTZ: Virtualizing ARM TrustZone

[20] The research conducts a study on variable approaches to virtualizing TrustZone in virtualized environments and then presents vTZ, a solution that securely provides each guest VM with a virtualized guest TEE using existing hardware.

- **Advantages:** vTZ leverages the idea of separating functionality from protection by maintaining a secure co-running VM to serve as a guest TEE, while using the hardware TrustZone to enforce strong isolation among guest TEEs and the untrusted hypervisor. vTZ further leverages Constrained Isolated Execution Environments (CIEEs) in the normal world to virtualize the functionality of TrustZone.

H. Lightweight Multicore Virtualization Architecture exploiting ARM TrustZone

[42] This work presents the extension of a TrustZone-assisted hypervisor to an asymmetric multi-processing configuration. We describe and demonstrate how to run a general-purpose operating system side-by-side with an real-time operating system

- **Advantages:** The achieved results demonstrated that the implemented multicore approach not only completely eliminates starvation, but also increases the general-purpose operating system's performance, especially when the real-time workload is demanding
- **Limitations:** The implemented solution is limited to the one-to-one mapping between cores, guests and worlds which is one of the limitations.

I. Towards a lightweight embedded virtualization architecture exploiting ARM TrustZone.

[44] This research presents the implementation of an embedded virtualization architecture through commodity hardware.

- **Advantages:** ARM TrustZone technology is exploited to implement a lightweight virtualization solution with low overhead and high determinism, corroborated by promising preliminary results.

J. VOSYSmonitor, a Low Latency Monitor Layer for Mixed-Criticality Systems on ARMv8-A

[35] The research proposes VOSYSmonitor, a multi-core software layer, which allows the co-execution of a safety-critical Real-Time Operating System (RTOS) and a noncritical General Purpose Operating System (GPOS) on the same hardware ARMv8-A platform

- **Advantages:** VOSYSmonitor main differentiation factors with the known solutions is the possibility for a processor to switch between secure and non-secure code execution at runtime. The partitioning is ensured by the ARM TrustZone technology, thus allowing to preserve the usage of virtualization features for the GPOS.
- **Limitations:** However, only one safety critical RTOS can be executed with the current implementation. VOSYSmonitor lacks of a way to take back the control if the Secure world OS does not release the core resource

K. Dual Operating System Architecture for Real-Time Embedded Systems

[27] This research presents a dual operating system virtualization architecture that supports integrated scheduling to enhance the responsiveness of the GPOS while preserving the determinism of the RTOS.

- **Advantages:** The proposed approach takes advantage of common embedded security hardware (ARM TrustZoneR) to improve the reliability and isolation of the RTOS with low overhead and no modifications to the GPOS core.
- **Limitations:** The current implementation focuses on data caches (but could be extended to apply to instruction caches) and cannot detect if samples were disturbed by the processing of non-maskable interrupts

L. Dual-OS Infrastructure for Mixed-Criticality Systems on ARMv8 Platforms

[58] In this research, a Dual-OS environment for Mixed-Criticality systems using ARM devices, by exploiting latest architecture changes and software advancements is proposed.

- **Advantages:** Through architecture technologies such as TrustZone and a proper firmware layer like ATF, two completely independent and isolated OS instances can be run simultaneously on the same hardware resources.

M. Affordable Separation on Embedded Platforms

[56] The research investigates an approach in which a few minor hardware additions together with virtualization offer protected execution in embedded systems while still allowing non-virtualized execution when secure services are not needed

- **Advantages:**Integrity and privacy of trusted guests are maintained at all times: while virtualization is active (in protected mode), while it is not (in normal mode), and while the machine is powered off.
- **Limitations:** The main costs for enabling isolated services consists of their decryption and the integrity check of those services and of the lightweight hypervisor.

N. Space and time partitioning with hardware support for space applications

[49] This research describes a real time hypervisor for space applications assisted by commercial off-the-shell (COTS) hardware.

- **Advantages:**The secure hypervisor is flexible enough to run unmodified guest OSes at higher performance, while guaranteeing strong isolation between guests, with

more than 99% performance for a 10 milliseconds guest-switching rate

- **Limitations:** In spite of using a multicore hardware architecture, the current implementation only supports a single-core configuration

IV. OS and Kernel security

A. Hypervision Across Worlds: Real-time Kernel Protection from the ARM TrustZone Secure World

[4] The research proposes TrApps, a secure platform for generalpurpose trusted execution in an untrusted cloud with multiple isolated tenants based on the ARM TrustZone technology

- **Advantages:** In this approach, normal world functions are forced to route through the secure world for inspection and approval before being executed. TZ-RKP's control of the normal world is non-bypassable. It can effectively stop attacks that aim at modifying or injecting kernel binaries. It can also stop attacks that involve modifying the system memory layout.
- **Limitations:** A better performance is achievable when the number of monitored events is reduced.

B. SPROBES: Enforcing Kernel Code Integrity on the TrustZone Architecture

[16] SPROBES, a novel primitive that enables introspection of operating systems running on ARM TrustZone hardware.

- **Advantages:** Using SPROBES, an introspection mechanism protected by TrustZone can instrument individual operating system instructions of its choice, receiving an unforgeable trap whenever any SPROBE is executed. They identify a set of five invariants whose enforcement is sufficient to restrict rootkits to execute only approved, SPROBE-injected kernel code.
- **Limitations:** Since smartphone boot-times can be performance-critical, SPROBES overhead may be an issue.

C. On-board Credentials with Open Provisioning

[28] The research describes how general-purpose secure hardware can be used to develop an architecture for credentials which is called On-board Credentials (ObCs).

- **Advantages:** ObCs combine the flexibility of virtual credentials with the higher levels of protection due to the use of secure hardware. A distinguishing feature of the ObC architecture is that it is open: it allows anyone to design and deploy new credential algorithms to ObC-capable devices without approval from the device manufacturer or any other third party.
- **Limitations:** Techniques for determining and describing the level of security in the secure environment on the target device are needed. the provisioning server needs ways to specify policies on how the provisioned credentials are to be accessed and used locally on the target device. the security and the usability of ObC system need to be more rigorously validated.

D. The ANDIX Research OS - ARM TrustZone Meets Industrial Control Systems Security

[15] The research introduces ANDIX OS, a security Operating System using the ARM TrustZone architecture to create a Trusted Execution Environment.

- **Advantages:** ANDIX OS can assign peripherals solely to the secure world, thus isolating these peripherals from the normal world. ANDIX OS's architecture minimizes the components responsible for creating the TEE thus reducing the attack surface.
- **Limitations:** In this approach, sharing device drivers between secure and normal worlds has significant overhead as their states need to be stored and transmitted too.

V. Mobile system security

A. TrustICE: Hardware-assisted Isolated Computing Environments on Mobile Devices

[61] The research proposes a TrustZone based isolation framework named TrustICE to create isolated computing environments (ICEs) in the normal domain. TrustICE securely isolates the secure code in an ICE from an untrusted Rich OS in the normal domain

- **Advantages:** The trusted computing base (TCB) of TrustICE remains small and unchanged regardless of the amount of secure code being protected. the switching time between an ICE and the Rich OS is less than 12 ms.
- **Limitations:** For the CPU and I/O devices, it have to rely on the careful design of TDC to clean up the CPU states and control the interrupts for the ICEs. However, when one ICE is running in the normal domain, it cannot access other ICEs' memory spaces that are protected by the watermark region, which can only be configured by TDC in the secure domain.

B. Enhancing the Security of Mobile Applications by using TEE and (U)SIM

[1] This research presents a security architecture and a novel mobile payment and multimedia content playback solution leveraging on the existing post-paid billing method

- **Advantages:** It addresses some of the security problems and lack of trust in Android ecosystem due to rooting and re-flashing with custom software. It provides an additional layer of security binding such that the trust worthiness of a mobile phone is removed once it is rooted or re-flashed.

C. Building Trusted Path on Untrusted Device Drivers for Mobile Devices

[31] This research proposes TrustUI, a new trusted path design for mobile devices that enables secure interaction between end users and services based on ARM's TrustZone technology

- **Advantages:** TrustUI takes a step further by excluding drivers for user-interacting devices like touch screen from

its trusted computing base (TCB). So, it has a much smaller TCB, requires no access to device driver code, and may easily adapt to many devices.
- **Limitations:** Doesn't support multiple secure applications and introduce a tiny sandbox mechanism to isolate them.

D. Trustworthy Execution on Mobile Devices: What security properties can my mobile platform give me?

[64] This research analyzes why widely-deployed hardware security primitives on mobile device platforms are inaccessible to application developers and end-users and show how modifications to existing proposals can strengthen the security properties available.

- **Advantages:** It strengthens the security properties available to applications and users without reducing the properties currently enjoyed by OEMs and network carriers

E. The Untapped Potential of Trusted Execution Environments on Mobile Devices

[14] This article discusses why TEEs are so widely deployed in mobile devices, and what kind of capabilities they support

- **Advantages:** The hardware-security APIs have been modeled after usage paradigms of hardware security modules(HSMs), cryptographic tokens and smart cards, and allow creation of hardware-protected keys, and common cryptographic operations, such as encryption and signatures, on them.
- **Limitations:** Its not clear how secrets collected and amassed in a trusted application are migrated to a new device when the user updates his device

F. Using ARM TrustZone to Build a Trusted Language Runtime for Mobile Applications

[54] This reserach presents the design, implementation, and evaluation of the Trusted Language Runtime (TLR), a system that protects the confidentiality and integrity of .NET mobile applications from OS security breaches.

- **Advantages:** The main benefit of the TLR is to bring the developer benefits of managed code to trusted computing. With the TLR, developers can build their trusted components with the productivity benefits of modern high-level languages, such as strong typing and garbage collection.
- **Limitations:** The TLR is statically allocated a portion of the device RAM by the bootloader itself. This might lead to wastage of precious resources as the memory cannot be reclaimed when not in use by the TrustZone.

G. Inspecting data from the safety of your trusted execution environment.

[65] This research presents a proof of concept that uses ARM TrustZone to perform introspection of a Linux kernel running in the normal world from within a secure-world system

- **Advantages:** It enumerates the steps necessary for taking advantage of existing volatile memory analysis techniques to be employed in the creation of a custom application that implements validation of the normal world system call table
- **Limitations:** Writing to the normal world from the secure world is implemented but untested.

H. Regulating ARM TrustZone Devices in Restricted Spaces

[7] This reserach develops mechanisms that allow hosts to analyze and regulate ARM TrustZone-based guest devices using remote memory operations. In our approach, restricted space hosts use remote memory operations to analyze and regulate guest devices within the restricted space

- **Advantages:** THey show that the ARM TrustZone allows our approach to obtain strong security guarantees while only requiring a small trusted computing base to execute on guest devices.
- **Limitations:**There are questions about its adoptability in real-world settings that remain to be answered

I. Retrofitting the Partially Privileged Mode for TEE Communication Channel Protection

[23] This research proposes the TEE defense (TFence) framework that enables the creation of a partially privileged (par-priv) process, which benefits from the coordination of the system mode and virtualization extension. More specifically, on ARM architecture, direct invocation of hypercall and SMC is not allowed in the user process

- **Advantages:** This approach removes the kernel dependency when the process communicates with the TEE, which also reduces the attack surface to the critical part of the application involved in the communication. Besides, direct communication with the hypervisor facilitates the adoption of application-shielding approaches to protect the critical part and to restrict arbitrary access to the TEE.
- **Limitations:** TFence consistently imposes overhead on the overall system owing to the hypervisor activation, which is around 6% with LMBench

VI. IoT/Embedded System Security

A. Software Abstractions for Trusted Sensors

[33] This research proposes two software abstractions for offering trusted sensors to mobile applications and the implementation of these abstractions on both x86 and ARM platforms.

- **Advantages:** It tries to overcome the privacy challenges posed by trusted sensors and demonstrates how differential privacy can be used with a GPS sensor to improve privacy.

B. Trusted Computing Building Blocks for Embedded Linux-based ARM TrustZone Platforms

[66] This research outlines an approach to merge TCG-style Trusted Computing concepts with ARM TrustZone technology

in order to build an open Linux-based embedded trusted computing platform

- **Advantages:** Debugging of the user-space supervisor process for the non-secure world VM is relatively straightforward in comparison to purely kernel-base virtualisation approaches.
- **Limitations:** The secure-world Linux prototype implementation supports two different in-kernel handling mechanisms for secure monitor calls originating in non-secure world. However these lowlevel in-kernel monitor call handlers are restricted as they must not perform any actions which might sleep or cause a secureworld task switch.

C. Secure device access for automotive software

[25] The research proposes a secure automotive software platform that has secure device access method with TrustZone

- **Advantages:** This approach restricts a direct access of the extension software and also supports multicore processors. In addition, the measured overhead is reduced by less than 1% degradation, and the maximum bandwidth of device access is achieved up to 5MB/s.
- **Limitations:** Inter Processor Interrupt (IPI) is restricted in TrustZone. Secure world can send IPI to any world in any core, but normal world can send IPI to only normal world, not secure world.

D. PROTC: PROTeCting drone's peripherals through ARM TrustZone

[34] This research proposes a new mechanism PROTC to protect the essential peripherals of a drone from being maliciously accessed. The protection is abstracted through the feature of ARM TrustZone.

- **Advantages:** PROTC ensures the drone's safety and important data's integrity even when the drone's OS is compromised and also allows the installation of third-party applications easily
- **Limitations:** Some malicious critical applications could have malicious proxy that compromises drone normal world OS by leveraging rootkits

E. FreeTEE: When real-time and security meet

[43] This research presents FreeTEE, an embedded architecture that emphasizes and preserves the real-time properties of the system but still guarantees security from the outset

- **Advantages:** This approach preserves the realTime properties of the system and still guarantees security from the outset and only a small overhead on latency is induced.
- **Limitations:** As FreeTEE is a single-core implemenation, the GPOS only runs when there is no RT ready-to-run task in the system.

F. Experimenting with ARM TrustZone Or: How I met friendly piece of trusted hardware

[67] This research discusses experiences made by the authors with a rather inexpensive development board and shows how system-level development on TrustZone-enabled hardware is possible in class-room settings

- **Advantages:** It integrates with the standard ARM Linux boot process and can be directly invoked from a boot loader like u-boot, also uses less than half of the available I-TCM and D-TCM space leaving 8K of the available D-TCM and 8K of the available I-TCM space for secure-world user applications
- **Limitations:** This approach configures all interrupts as FIQ-type interrupts and redirects them to secure monitor mode. This is problematic as Linux usually does not expect FIQ-type interrupts which requires additional code in secure monitor mode to simulate delivery of an IRQ-type interrupt

G. A Private User Data Protection Mechanism in TrustZone Architecture Based on Identity Authentication

[69] This research proposes a private user data protection mechanism in TrustZone to avoid user data leakage risk from Client Applications

- **Advantages:**This uses identity verification module to a multi-user TrustZone environment, and establish the correspondence between a data object and its owner to verify the identity of each data access requester. This mechanism guarantees the security of data access, and enhances the protection of users' private data.
- **Limitations:** The communication between the normal world and the secure world employs the default SMC command which is time consuming

H. IIoTEED: An Enhanced, Trusted Execution Environment for Industrial IoT Edge Devices

[45] The research demonstrates why TrustZone is becoming a reference technology for securing IoT edge devices, and how enhanced Tees can help meet industrial IoT applications' real-time requirements

- **Advantages:** IIoTEED has fully or partially achieved the three fundamental elements of CIA: confidentiality, integrity, and availability. it also reinforces edge device encryption and transport layer security.
- **Limitations:** IIoTEED only partially addresses the industrial security puzzle at edge level. To help with end-to-end security while meeting the right tradeoff between security and real-time requirements, it must be extended with other tailored, hardware-assisted security and acceleration strategies

I. MIPE: a practical memory integrity protection method in a trusted execution environment

[12] The research proposes a practical memory integrity protection method on an ARM-based platform, called MIPE, to defend against security threats including kernel data attacks and direct memory access attacks

- **Advantages:** MIPE utilizes TrustZone technique to create a isolated execution environment, which can protect the sensitive code and data against attacks. To present the

integrity protection strategies, we provide the design of MIPE using B method, which is a practical formal methodtion and transport layer security.
- **Limitations:** Doesn't provide the memory integrity verification by formalization

J. CFI CaRE: Hardware-Supported Call and Return Enforcement for Commercial Microcontrollers

[40] The research proposes CaRE, the first interrupt-aware CFI scheme for low-end MCUs. CaRE uses a novel way of protecting the CFI metadata by leveraging TrustZone-M security extensions introduced in the ARMv8-M architecture.

- **Advantages:** Its binary instrumentation approach preserves the memory layout of the target MCU software, allowing pre-built bare-metal binary code to be protected by CaRE. Leveraging hardware assisted security is an important enabler in CaRE, but it also meets other requirements important for practical deployment on small devices

K. Providing Root of Trust for ARM TrustZone using On-Chip SRAM

[70] The research presents the design, implementation and evaluation of the root of trust for the Trusted Execution Environment (TEE) provided by ARM TrustZone based on the on-chip SRAM Physical Unclonable Functions

- **Advantages:** The root of trust resists software attackers capable of compromising the entire rich OS. we leverage the on-chip SRAM, commonly available on mobile devices, to achieve a low-cost, secure, and efficient design of the root of trust

VII. Trusted services

A. TrustOTP: Transforming Smartphones into Secure One-Time Password Tokens

[59] TrustOTP, is a secure onetime password solution that can achieve both the flexibility of software tokens and the security of hardware tokens by using ARM TrustZone technology.

- **Advantages:** TrustOTP can not only protect the confidentiality of the OTPs against a malicious mobile OS, but also guarantee reliable OTP generation and trusted OTP display when the mobile OS is compromised or even crashes. It is flexible to integrate multiple OTP algorithms and instances for different application scenarios on the same smartphone platform without modifying the mobile OS.
- **Limitations:** TrustOTP depends on the battery of the smartphone, which needs to be recharged every one or two days. When the smartphone is out of battery, TrustOTP cannot work until being charged by a computer or a power outlet. TrustOTP may suffer from man-in-the-middle attacks when used alone.

B. TrustDump: Reliable Memory Acquisition on Smartphones

[60] The research proposes a TrustZone-based memory acquisition mechanism called TrustDump that is capable of reliably obtaining the RAM memory and CPU registers of the mobile OS even if the OS has crashed or has been compromised.

- **Advantages:** The mobile OS is running in the TrustZone's normal domain, and the memory acquisition tool is running in the TrustZone's secure domain, which has the access privilege to the memory in the normal domain. we rely on ARM TrustZone to achieve a hardware-assisted isolation with a small trusted computing base (TCB) of about 450 lines of code.

C. E-Pass Using DRM in Symbian v8 OS and TrustZone : Securing Vital Data on Mobile Devices

[21] This research addresses the issue of protecting the content of digital media in Symbian OS mobile phones using a secured E-Pass stored on Symbian v8 OS mobile devices

- **Advantages:** The integration of Symbian DRM and Trustzone not only adds another layer of security but also remains transparent to the end user because the data is stored in a secure domain within the operating system. As this security protection happens in the background, user never sees the add-on security and gives no different whatsoever from user's point of view
- **Limitations:** This would restrict the user's ability to freely install applications.

D. A Path Towards Ubiquitous Protection of Media

[63] This research outlines a possible approach to enable a Secure Media Path on mobile devices to support the needs of the different stakeholders, with respect to openness, content protection and client privacy.

- **Advantages:** Media processing is isolated from the rich OS, so the interests of the content provider to protect their content from piracy are preserved. At the same time, such software is not able to subvert the security and privacy of the user because it can access the relevant parts of the media pipeline only.

E. Using Innovative Instructions to Create Trustworthy Software Solutions

1) One-time Password (OTP): [19] A simple application that can be created with SGX is a generator of one-time passwords.

- **Advantages:** The OTP prototype described here prevents malicious software from gaining access to the OTP pre-shared key, including targeted attacks from malicious software with higher privilege.
- **Limitations:** Instead of sending the challenge directly to the OTP enclave, the remote service could generate a bitmap containing a visual representation of the challenge code, and encrypt this before sending to the OTP enclave. To harden the solution even further, a trusted input path

could be established between an input device and Intel's Manageability Engine

2) Enterprise Rights Management: [19] The research describes an SGX technology-based ERM architecture, focused on document distribution, access control and viewing, that addresses the vulnerabilities found in today's ERM systems.

- **Advantages:** With SGX technology enabled the application to withstand these attacks and satisfy all its security objectives. The protected document is encrypted within the enclave with a randomly generated key which is stored on the server, and later distributed to authorized Intel® SGX protected document viewers. The encrypted documents themselves need not be stored in the server database. All communication between the server and authenticated clients is encrypted and also offers integrity and replay protection to provide end-to-end security for various use cases.

3) Video Conferencing: [19] The research examines how the security of a video conferencing application can be hardened using SGX technology

- **Advantages:** SGX allows a video conferencing application to protect its assets on the platform and enables strong participant authentication, thus mitigating a broad range of threats. we protected the audio and video output as well, through integration of SGX and PAVP technologies, which protects against attacks such as video buffer scraping.

F. Trusted Execution Environment-Based Authentication Gauge (TEEBAG)

[5] The approach uses existing features and capabilities of TEEs toenhance the security of user authentication.

- **Advantages:** They reverse the way traditional authentication schemes work: instead of the user presenting their authentication data to a remote device, they request the remote device to send the stored authentication template(s) to the local device
- **Limitations:** The main challenge is to enable the user to identify or authenticate the trusted application within the device they are using to differentiate it from an impostor

G. Secure Block Device - Secure, Flexible, and Efficient Data Storage for ARM TrustZone Systems

[18] The research introduces the Secure Block Device, a secure, easy to use, flexible, efficient, and widely applicable minimal Trusted Computing Base solution to provide data confidentiality and integrity for Data at Rest. They evaluate the Secure Block Device by using it as the core component in a secure storage Trusted Application that uses the ARM TrustZone Trusted Execution Environment to provide a confidential and integrity protected block device to the normal world OS.

- **Advantages:** The approach uses a Merkle-Tree in conjunction with a selectable Authenticated Encryption scheme to provide an easy to integrate solution for applications that require fast and secure random access to data, but not a fully fledged file system

H. Secure Block Device - Secure, Flexible, and Efficient Data Storage for ARM TrustZone Systems

[32] This research designs and builds DroidVault— the first realization of a trusted data vault on the Android platform

- **Advantages:** DroidVault establishes a secure channel between data owners and data users while allowing data owners to enforce strong control over the sensitive data with a minimal trusted computing base (TCB). DroidVault can be adopted by legacy cloud storage services and support popular operations on sensitive data.

VIII. Miscellanous

A. Ninja: Towards Transparent Tracing and Debugging on ARM

[39] NINJA, is a transparent malware analysis framework on ARM platform with low artifacts

- **Advantages:** NINJA leverages a hardware-assisted isolated execution environment TrustZone to transparently trace and debug a target application with the help of Performance Monitor Unit and Embedded Trace Macrocell. NINJA does not modify system software and is OS-agnostic on ARM platform.

B. A First Step Towards Leveraging Commodity Trusted Execution Environments for Network Applications

[26] The research explores the possibility of using Intel SGX to provide security and privacy in a wide range of network applications.

- **Advantages:** Leveraging hardware protection of TEEs opens up new possibilities, often at the benefit of a much simplified application/protocol design. Their design shows that SGX can not only improve the security and privacy of existing applications, but also enable new services
- **Limitations:** However, denialof-service is out of scope; a malicious privilege software could easily crash or halt the system. TEE's usage has been rather limited due to performance issues.

C. Mobile Ticketing System Employing TrustZone Technology

[22] The research proposes a mobile receipt system that offers security for the customer and verifiability for the merchant based on TrustZone Architecture for ARM devices.

- **Advantages:** TrustZone protects the data stored in the device from being altered while providing transparency to the user and reliability to the vendor

D. Identity Verification Schemes for Public Transport Ticketing with NFC Phones

[62] The research examines the feasibility of using mobile phone with a TEE for identity verification of transport ticketing with a perspective focusing on security and performance. It uses the On-board Credentials (ObC) architecture as the TEE.

- **Advantages:** Confidentiality and integrity of the data exchange is guaranteed by the ObC provisioning protocol.

It saves on the time budget considerably as compared to its peers.

- **Limitations:** This method aims to reduce the efficacy of MITM attacks, but the threat of a MITM attack is still present. Also this does not provide a trusted receipt of the transaction. Has privacy issues - CA and TA can build a route information profile of a user based on the user's traveling behaviour.

E. A Framework for Privacy-Preserving Mobile Payment on Security Enhanced ARM TrustZone Platforms

[50] This research demonstrates how privacy friendly payment can be realized using a recent payment mechanisms in combination with an ARM processor platform with TrustZone enhancements

- **Advantages:** The proposed framework is, however, not limited to this application and can be used for any application requiring a privacy preserving online remote prepaid payment system suitable for micro as well as macro payments.

F. Using Trusted Execution Environments in Two-factor Authentication: comparing approaches

[52] The research examines two TEE technologies, Intel's IPT and ARM TrustZone, revealing that, although it is possible to get close to classic two-factor authentication in terms of ser interaction security, both technologies have distinct drawbacks.

- **Advantages:** The model also clearly shows an open problem shared by many TEEs: how to prove to the user that they are dealing with a trusted application when trusted and untrusted applications share the same display
- **Limitations:** Intel IPT has a serious issue where there is no trusted input path for the user to enter data. ARM TrustZone requires careful selection of the right components by the system-on-a-chip designer that puts the parts of the TEE together to guarantee that it can be trusted. Both technologies share a common issue, which is how to prove to the user that they are dealing with a trusted application.

G. Smartphones as Practical and Secure Location Verification Tokens for Payments

[36] The research proposes a location-based second-factor authentication solution for modern smartphones making use of TEEs commonly available on modern smartphones.

- **Advantages:** The proposed method can be effectively used for the detection of fraudulent transactions caused by card theft or counterfeiting and it also does not require any changes in the user behavior at the point of sale or to the deployed terminals.
- **Limitations:** The card issuer can ask the user's device for location statements and track the user over time. The card issuer may also abuse the system and query the user's device for a location statement when the card is not involved in a transaction.

H. AdAttester: Secure online mobile advertisement attestation using TrustZone.

[30] This research proposes a verifiable mobile ad framework called AdAttester, based on ARM's TrustZone technology

- **Advantages:** AdAttester provides two novel security primitives, namely unforgeable clicks and verifiable display. The two primitives attest that ad-related operations (e.g., user clicks) are initiated by the end user (instead of a bot) and that the ad is displayed intact and timely.

I. SeCReT: Secure Channel between Rich Execution Environment and Trusted Execution Environment

[24] This research proposes SeCReT to ameliorate the problem of secure communication between REE and TEE. SeCReT is a framework that builds a secure channel between the REE and TEE by enabling REE processes to use session keys in the REE that is regarded as unsafe region

- **Advantages:** To protect the key, SeCReT interposes with every switch between user mode and kernel mode, verifying the code's integrity and the coarse-grained control flow of the process. To minimize the performance overhead, SeCReT's key-protection mechanism is activated only during the runtime of the process that has permission to access TrustZone

REFERENCES

[1] Zaheer Ahmad, Lishoy Francis, Tansir Ahmed, Christopher Lobodzinski, Dev Audsin, and Peng Jiang. Enhancing the security of mobile applications by using TEE and (u)SIM. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*. IEEE, December 2013. doi:10.1109/uic-atc.2013.76.

[2] N. Asokan, Jan-Erik Ekberg, Kari Kostiainen, Anand Rajan, Carlos Rozas, Ahmad-Reza Sadeghi, Steffen Schulz, and Christian Wachsmann. Mobile trusted computing. *Proceedings of the IEEE*, 102(8):1189–1206, August 2014. doi:10.1109/jproc.2014.2332007.

[3] N. Asokan, Jan-Erik Ekberg, Kari Kostiainen, Anand Rajan, Carlos Rozas, Ahmad-Reza Sadeghi, Steffen Schulz, and Christian Wachsmann. Mobile trusted computing. *Proceedings of the IEEE*, 102(8):1189–1206, 2014. doi:10.1109/JPROC.2014.2332007.

[4] Ahmed M. Azab, Peng Ning, Jitesh Shah, Quan Chen, Rohan Bhutkar, Guruprasad Ganesh, Jia Ma, and Wenbo Shen. Hypervision across worlds: Real-time kernel protection from the arm trustzone secure world. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 90–102, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2660267.2660350.

[5] Ranjbar A. Balisane and Andrew Martin. Trusted execution environment-based authentication gauge (TEEBAG). In *Proceedings of the 2016 New Security Paradigms Workshop*. ACM, September 2016. doi:10.1145/3011883.3011892.

[6] Andrew Baumann, Marcus Peinado, and Galen Hunt. Shielding applications from an untrusted cloud with haven. *ACM Transactions on Computer Systems*, 33(3):1–26, September 2015. doi:10.1145/2799647.

[7] Ferdinand Brasser, Daeyoung Kim, Christopher Liebchen, Vinod Ganapathy, Liviu Iftode, and Ahmad-Reza Sadeghi. Regulating ARM TrustZone devices in restricted spaces. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, June 2016. doi:10.1145/2906388.2906390.

[8] Stefan Brenner, David Goltzsche, and Rüdiger Kapitza. Trapps: Secure compartments in the evil cloud. In *Proceedings of the 1st International Workshop on Security and Dependability of Multi-Domain Infrastructures*, XDOMO'17, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3071064.3071069.

[9] Stefan Brenner, Colin Wulf, and Rüdiger Kapitza. Running zookeeper coordination services in untrusted clouds. In *Proceedings of the 10th USENIX Conference on Hot Topics in System Dependability*, HotDep'14, page 2, USA, 2014. USENIX Association.

[10] Tiago Brito, Nuno O. Duarte, and Nuno Santos. Arm trustzone for secure image processing on the cloud. In *2016 IEEE 35th Symposium on Reliable Distributed Systems Workshops (SRDSW)*, pages 37–42, 2016. doi:10.1109/SRDSW.2016.17.

[11] Marco Cereia and Ivan Cibrario Bertolotti. Asymmetric virtualisation for real-time systems. In *2008 IEEE International Symposium on Industrial Electronics*. IEEE, June 2008. doi:10.1109/isie.2008.4677005.

[12] Rui Chang, Liehui Jiang, Wenzhi Chen, Yang Xiang, Yuxia Cheng, and Abdulhameed Alelaiwi. MIPE: a practical memory integrity protection method in a trusted execution environment. *Cluster Computing*, 20(2):1075–1087, March 2017. doi:10.1007/s10586-017-0833-4.

[13] Giorgiomaria Cicero, Alessandro Biondi, Giorgio Buttazzo, and Anup Patel. Reconciling security with virtualization: A dual-hypervisor design for ARM TrustZone. In *2018 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, February 2018. doi:10.1109/icit.2018.8352425.

[14] Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. The untapped potential of trusted execution environments on mobile devices. *IEEE Security Privacy*, 12(4):29–37, 2014. doi:10.1109/MSP.2014.38.

[15] Andreas Fitzek, Florian Achleitner, Johannes Winter, and Daniel Hein. The ANDIX research OS - ARM TrustZone meets industrial control systems security. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. IEEE, July 2015. doi:10.1109/indin.2015.7281715.

[16] Xinyang Ge, Hayawardh Vijayakumar, and Trent Jaeger. Sprobes: Enforcing kernel code integrity on the trustzone architecture, 2014. arXiv:1410.7747.

[17] GlobalPlatform. *TEE system architecture*. [Online], 2011. URL: http://globalplatform.org/specificationsdevice.asp.

[18] Daniel Hein, Johannes Winter, and Andreas Fitzek. Secure block device – secure, flexible, and efficient data storage for ARM TrustZone systems. In *2015 IEEE Trustcom/BigDataSE/ISPA*. IEEE, August 2015. doi:10.1109/trustcom.2015.378.

[19] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, Vinay Phegade, and Juan Del Cuvillo. Using innovative instructions to create trustworthy software solutions. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2487726.2488370.

[20] Zhichao Hua, Jinyu Gu, Yubin Xia, Haibo Chen, Binyu Zang, and Haibing Guan. Vtz: Virtualizing arm trustzone. In *Proceedings of the 26th USENIX Conference on Security Symposium*, SEC'17, page 541–556, USA, 2017. USENIX Association.

[21] Wan Huzaini Wan Hussin. E-pass using DRM in symbian v8 OS and TrustZone : Securing vital data on mobile devices. In *2006 International Conference on Mobile Business*. IEEE, December 2006. doi:10.1109/icmb.2006.14.

[22] Wan Huzaini, Wan Hussin, P. Coulton, and R. Edwards. Mobile ticketing system employing TrustZone technology. In *International Conference on Mobile Business (ICMB'05)*. IEEE. doi:10.1109/icmb.2005.71.

[23] Jinsoo Jang and Brent Byunghoon Kang. Retrofitting the partially privileged mode for TEE communication channel protection. *IEEE Transactions on Dependable and Secure Computing*, 17(5):1000–1014, September 2020. doi:10.1109/tdsc.2018.2840709.

[24] Jinsoo Jang, Sunjune Kong, Minsu Kim, Daegyeong Kim, and Brent Kang. Secret: Secure channel between rich execution environment and trusted execution environment. 01 2015. doi:10.14722/ndss.2015.23189.

[25] Se Won Kim, Chiyoung Lee, MooWoong Jeon, Hae Young Kwon, Hyun Woo Lee, and Chuck Yoo. Secure device access for automotive software. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, December 2013. doi:10.1109/iccve.2013.6799789.

[26] Seongmin Kim, Youjung Shin, Jaehyung Ha, Taesoo Kim, and Dongsu Han. A first step towards leveraging commodity trusted execution environments for network applications. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, HotNets-XIV, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2834050.2834100.

[27] Florian Kluge, Jörg Mische, Sascha Uhrig, and Theo Ungerer. An operating system architecture for organic computing in embedded real-time systems. In *Lecture Notes in Computer Science*, pages 343–357. Springer Berlin Heidelberg. doi:10.1007/978-3-540-69295-9_28.

[28] Kari Kostiainen, Jan-Erik Ekberg, N. Asokan, and Aarne Rantala. On-board credentials with open provisioning. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09*. ACM Press, 2009. doi:10.1145/1533057.1533074.

[29] Li Li, Dijiang Huang, Zhidong Shen, and Samia Bouzefrane. A cloud based dual-root trust model for secure mobile online transactions. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 4404–4409, 2013. doi:10.1109/WCNC.2013.6555287.

[30] Wenhao Li, Haibo Li, Haibo Chen, and Yubin Xia. AdAttester. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, May 2015. doi:10.1145/2742647.2742676.

[31] Wenhao Li, Mingyang Ma, Jinchen Han, Yubin Xia, Binyu Zang, Cheng-Kang Chu, and Tieyan Li. Building trusted path on untrusted device drivers for mobile devices. In *Proceedings of 5th Asia-Pacific Workshop on Systems*, APSys '14, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2637166.2637225.

[32] Xiaolei Li, Hong Hu, Guangdong Bai, Yaoqi Jia, Zhenkai Liang, and Prateek Saxena. DroidVault: A trusted data vault for android devices. In *2014 19th International Conference on Engineering of Complex Computer Systems*. IEEE, August 2014. doi:10.1109/iceccs.2014.13.

[33] He Liu, Stefan Saroiu, Alec Wolman, and Himanshu Raj. Software abstractions for trusted sensors. In *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*. ACM Press, 2012. doi:10.1145/2307636.2307670.

[34] Renju Liu and Mani Srivastava. PROTC. In *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications - DroNet '17*. ACM Press, 2017. doi:10.1145/3086439.3086443.

[35] Pierre Lucas, Kevin Chappuis, M. Paolino, Nicolas Dagieu, and D. Raho. Vosysmonitor, a low latency monitor layer for mixed-criticality systems on armv8-a. In *ECRTS*, 2017.

[36] Claudio Marforio, Nikolaos Karapanos, Claudio Soriente, Kari Kostiainen, and Srdjan Čapkun. Smartphones as practical and secure location verification tokens for payments. In *Proceedings 2014 Network and Distributed System Security Symposium*. Internet Society, 2014. doi:10.14722/ndss.2014.23165.

[37] José Martins, João Alves, Jorge Cabral, Adriano Tavares, and Sandro Pinto. μrtzvisor: A secure and safe real-time hypervisor. volume 6, 2017. URL: https://www.mdpi.com/2079-9292/6/4/93, doi:10.3390/electronics6040093.

[38] Zhenyu Ning, Jinghui Liao, Fengwei Zhang, and Weisong Shi. Preliminary study of trusted execution environments on heterogeneous edge platforms. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, October 2018. doi:10.1109/sec.2018.00057.

[39] Zhenyu Ning and Fengwei Zhang. Ninja: Towards transparent tracing and debugging on arm. In *Proceedings of the 26th USENIX Conference on Security Symposium*, SEC'17, page 33–49, USA, 2017. USENIX Association.

[40] Thomas Nyman, Jan-Erik Ekberg, Lucas Davi, and N. Asokan. CFI CaRE: Hardware-supported call and return enforcement for commercial microcontrollers. In *Research in Attacks, Intrusions, and Defenses*, pages 259–284. Springer International Publishing, 2017. doi:10.1007/978-3-319-66332-6_12.

[41] M. Paolino, A. Rigo, A. Spyridakis, Jeremy Fanguede, Petar Lalov, and D. Raho. T-kvm : A trusted architecture for kvm arm v 7 and v 8 virtual machines securing virtual machines by means of kvm , trustzone , tee and selinux. 2015.

[42] S. Pinto, A. Oliveira, J. Pereira, J. Cabral, J. Monteiro, and A. Tavares. Lightweight multicore virtualization architecture exploiting ARM TrustZone. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, October 2017. doi:10.1109/iecon.2017.8216603.

[43] S. Pinto, D. Oliveira, J. Pereira, J. Cabral, and A. Tavares. FreeTEE: When real-time and security meet. In *2015 IEEE 20th Conference on*

Emerging Technologies & Factory Automation (ETFA). IEEE, September 2015. doi:10.1109/etfa.2015.7301571.

[44] S. Pinto, D. Oliveira, J. Pereira, N. Cardoso, M. Ekpanyapong, J. Cabral, and A. Tavares. Towards a lightweight embedded virtualization architecture exploiting ARM TrustZone. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, September 2014. doi:10.1109/etfa.2014.7005255.

[45] Sandro Pinto, Tiago Gomes, Jorge Pereira, Jorge Cabral, and Adriano Tavares. IIoTEED: An enhanced, trusted execution environment for industrial IoT edge devices. *IEEE Internet Computing*, 21(1):40–47, January 2017. doi:10.1109/mic.2017.17.

[46] Sandro Pinto, Jorge Pereira, Tiago Gomes, Mongkol Ekpanyapong, and Adriano Tavares. Towards a TrustZone-assisted hypervisor for real-time embedded systems. *IEEE Computer Architecture Letters*, 16(2):158–161, July 2017. doi:10.1109/lca.2016.2617308.

[47] Sandro Pinto, Jorge Pereira, Tiago Gomes, Adriano Tavares, and Jorge Cabral. LTZVisor: TrustZone is the Key. In Marko Bertogna, editor, *29th Euromicro Conference on Real-Time Systems (ECRTS 2017)*, volume 76 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:22, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7153, doi:10.4230/LIPIcs.ECRTS.2017.4.

[48] Sandro Pinto and Nuno Santos. Demystifying arm TrustZone. *ACM Computing Surveys*, 51(6):1–36, February 2019. doi:10.1145/3291047.

[49] Sandro Pinto, Adriano Tavares, and Sergio Montenegro. Space and time partitioning with hardware support for space applications. 05 2016.

[50] Martin Pirker and Daniel Slamanig. A framework for privacy-preserving mobile payment on security enhanced ARM TrustZone platforms. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, June 2012. doi:10.1109/trustcom.2012.28.

[51] Martin Pirker, Daniel Slamanig, and Johannes Winter. Practical privacy preserving cloud resource-payment for constrained clients. In *Privacy Enhancing Technologies*, pages 201–220. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-31680-7_11.

[52] Roland Rijswijk-Deij and Erik Poll. Using trusted execution environments in two-factor authentication: comparing approaches. 09 2013.

[53] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64, 2015. doi:10.1109/Trustcom.2015.357.

[54] Nuno Santos, Himanshu Raj, Stefan Saroiu, and Alec Wolman. Using ARM trustzone to build a trusted language runtime for mobile applications. *ACM SIGARCH Computer Architecture News*, 42(1):67–80, April 2014. doi:10.1145/2654822.2541949.

[55] Felix Schuster, Manuel Costa, Cedric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, and Mark Russinovich. *VC3: Trustworthy Data Analytics in the Cloud Using SGX*. IEEE, May 2015. doi:10.1109/sp.2015.10.

[56] Oliver Schwarz, Christian Gehrmann, and Viktor Do. Affordable separation on embedded platforms. In *Trust and Trustworthy Computing*, pages 37–54. Springer International Publishing, 2014. doi:10.1007/978-3-319-08593-7_3.

[57] Jaebok Shin, Yungu Kim, Wooram Park, and Chanik Park. DFCloud: A TPM-based secure data access control method of cloud storage in mobile devices. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, December 2012. doi:10.1109/cloudcom.2012.6427606.

[58] Alexander Spyridakis, Petar Lalov, and Daniel Raho. Dual-os infrastructure for mixed-criticality systems on armv8 platforms. *Virtual Open Systems*, 2015.

[59] He Sun, Kun Sun, Yuewu Wang, and Jiwu Jing. Trustotp: Transforming smartphones into secure one-time password tokens. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 976–988, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2810103.2813692.

[60] He Sun, Kun Sun, Yuewu Wang, Jiwu Jing, and Sushil Jajodia. TrustDump: Reliable memory acquisition on smartphones. In *Computer Security - ESORICS 2014*, pages 202–218. Springer International Publishing, 2014. doi:10.1007/978-3-319-11203-9_12.

[61] He Sun, Kun Sun, Yuewu Wang, Jiwu Jing, and Haining Wang. TrustICE: Hardware-assisted isolated computing environments on mobile devices. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, June 2015. doi:10.1109/dsn.2015.11.

[62] Sandeep Tamrakar, Jan-Erik Ekberg, and N. Asokan. Identity verification schemes for public transport ticketing with NFC phones. In *Proceedings of the sixth ACM workshop on Scalable trusted computing - STC '11*. ACM Press, 2011. doi:10.1145/2046582.2046591.

[63] Ronald Tögl, Johannes Winter, and Martin Pirker. A path towards ubiquitous protection of media. In *Proc.Workshop on Web Applications and Secure Hardware*, volume 1011 of *CEUR Workshop Proceedings*, pages 32–38. Technical University of Aachen, 2013. urn:nbn:de:0074-1011-6; Workshop on Web Applications and Secure Hardware ; Conference date: 20-06-2013.

[64] Amit Vasudevan, Emmanuel Owusu, Zongwei Zhou, James Newsome, and Jonathan McCune. Trustworthy execution on mobile devices: What security properties can my mobile platform give me? volume 7344, pages 159–178, 06 2012. doi:10.1007/978-3-642-30921-2_10.

[65] John Williams. Inspecting data from the safety of your trusted execution environment. *Proceedings of the Black Hat Conference*, 2015.

[66] Johannes Winter. Trusted computing building blocks for embedded linux-based ARM trustzone platforms. In *Proceedings of the 3rd ACM workshop on Scalable trusted computing - STC '08*. ACM Press, 2008. doi:10.1145/1456455.1456460.

[67] Johannes Winter. Experimenting with ARM TrustZone – or: How i met friendly piece of trusted hardware. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, June 2012. doi:10.1109/trustcom.2012.157.

[68] Fengwei Zhang and Hongwei Zhang. Sok: A study of using hardware-assisted isolated execution environments for security. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, HASP 2016, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2948618.2948621.

[69] Bo Zhao, Yu Xiao, Yuqing Huang, and Xiaoyu Cui. A private user data protection mechanism in trustzone architecture based on identity authentication. *Tsinghua Science and Technology*, 22(2):218–225, April 2017. doi:10.23919/tst.2017.7889643.

[70] Shijun Zhao, Qianying Zhang, Guangyao Hu, Yu Qin, and Dengguo Feng. Providing root of trust for ARM TrustZone using on-chip SRAM. In *Proceedings of the 4th International Workshop on Trustworthy Embedded Devices - TrustED '14*. ACM Press, 2014. doi:10.1145/2666141.2666145.