SoK: A Systematic Review of TEE Usage for Developing Trusted Applications

Arttu Paju*, Muhammad Owais Javed*, Juha Nurmi*, Juha Savimäki*†, Billy Bob Brumley*
*Tampere University, Tampere, Finland
†Unikie Oy, Tampere, Finland

***Abstract*—Trusted Execution Environments (TEEs) provide an additional layer of trust for applications. This means that modern central processing units (CPUs) can protect a process and its data from other processes and even the operating system (OS). The CPU acts as a gatekeeper when an application is running inside the TEE system. Other programs, including the hypervisor and the OS, see only encrypted memory data. The purpose of this article is to assist the reader in understanding TEEs, including when, why, and how they are used.**
To understand TEE utilisation, we compile academic and practical examples by collecting a total of 0 references for this review. We summarise the literature and provide a publication timeline. We discover that the academic and practical work on TEE fit well under the categories of review, framework, container, and application, all of which are growing trends according to our analysis. Similarly, we categorise TEE use cases into major groups to better understand their application domains.

***Index Terms*—Trusted Execution Environment; TEE; Trusted Computing; Confidential Computing; Privacy and Confidentiality; Remote Attestation; Usability; Application Security; Trust model**

I. Introduction

Often, sensitive data is processed on potentially compromised devices. Typically, when a device is compromised, the data and code are also compromised [1]. For instance, if an adversary takes control of an IoT device or a cloud instance, the adversary can also access the software processes [1, 2].

For the additional privacy, protection, and authenticity of data in process, Trusted Execution Environments (TEEs) satisfy all these requirements. TEEs protect processes from other processes running in the Rich Execution Environment (REE), including the high-level operating system (OS).

A TEE is a component of the main CPU that ensures the confidentiality and integrity of the code and data loaded inside. Remote attestation is accomplished through the use of trusted firmware. When an application is attested, the untrusted component loads the trusted component into memory: the trusted application is then protected against modification.

Different hardware security features separate the TEE and REE: the TEE is trusted, whereas the REE is not protected with the same rigour by the hardware [3].

TEE implementations are available from a variety of hardware vendors, including AMD Platform Security Processor (PSP), ARM TrustZone, Google Titan M, Intel Software Guard Extensions (SGX), Apple Secure Enclave Processor (SEP), and RISC-V Keystone [4].

Even with appropriate domain expertise, it is challenging to provide direction for TEE application development. To address this, we take the software developer's perspective and review TEE development tools and use cases for Trusted Applications (TAs).

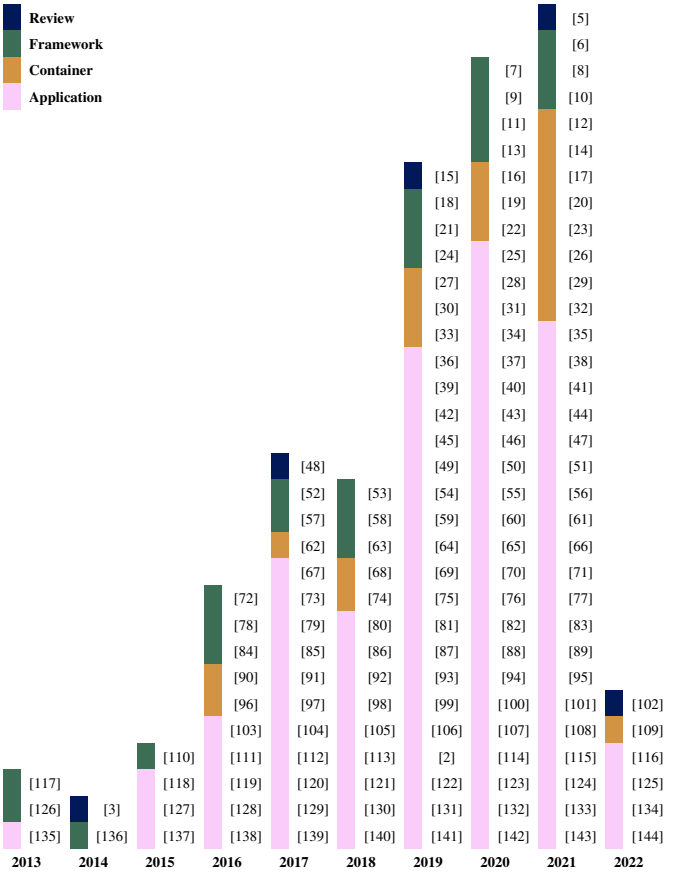

Fig. 1. Literature related to TEEs placed under the categories of review, framework, container, and application. Ordered by year.

Our research questions (RQs) are:

RQ1. **What are the applications of TEEs?**
RQ2. **Which TEE software tools are available for TA development?**
RQ3. **What types of TEE containers are available for TA packaging?**

The scope of this review is to understand practical TEE deployments. Figure 1 illustrates our references, categorised

and ordered by year. In case of multiple references for the same topic, we list the most peer-reviewed one.

We collected a total of 0 references between 2022-03-11 and 2022-06-07 for this review. Our TEE reference search methodology encompasses both academic and applied activity. We discover that the publications fit well under the categories of review, framework, container, and application. The selection of these categories is based on the references' natural fit within them.

We notice that publication velocity increases after 2015. Between 2013 and 2015, there are few publications, but the rate of publication began rising in 2016. The number of publications in 2018 remained almost unchanged from the previous year. In 2019, the number of publications nearly doubled, and it has continued to rise in 2020 and 2021. There are more publications in 2021 than ever before, and we speculate this trend will continue in 2022 and the foreseeable future.

Two-thirds of the publications demonstrate applications. Almost one-third of articles cover frameworks and containers. Few publications are reviews, including surveys and systematisations of knowledge.

This provides the motivation for our article: systematisation of knowledge is required because there are numerous articles but few reviews. These reviews have a limited perspective, typically cover only one piece of hardware, and do not seek real usage examples. Our knowledge systematisation assists software developers and encompasses heterogeneous hardware.

This preliminary categorisation of TEE papers shows how academic literature focuses on application demonstrations, and helps form a basis for our research questions. It is the initial beginning of our extensive systematisation of knowledge.

A. Related work

As there is prior work on systematising TEE knowledge, we began studying publications and resources that organise TEE utilisation. These data sources cover the following topics.

Development frameworks. Each CPU vendor has its own TEE. To assist TA development, there are numerous frameworks and containers that aid the software developer [5]. Intel Software Guard Extensions SDK, OP-TEE, etc. intend to make the development easier.

Trusted containers (tcons). To execute an application within a TEE, a developer must apply framework-specific modifications to the original application, which can be a time-consuming operation. Trusted containers (tcons) solve this usability issue by allowing direct execution of unmodified binary code within a TEE, or for performing automated transformations on source code prior to loading it into a TEE executable [96]. Certain tcons support multiple hardware backends, eliminating the need for the software developer to make hardware design selections at the code level [5]. We utilise the existing work on TEE containers by Liu et al. [5] in our categorisations in Table III and Table IV. Their work provides a comprehensive analysis of 15 existing tcon solutions' designs and implementations, highlighting the most common security pitfalls. We are not evaluating containers in terms of security, but rather analysing the software wrapper stack and hardware support of 20 containers. Additionally, we check which containers are open source.

Applications of TEEs. Tamrakar [48] covers several applications of TEEs, including attestation and access control. Our categorisation of TEE utilisation in Figure 2 is not based on said work, yet we included the applications presented therein. We also used the study of attestation mechanisms for TEEs by Ménétrey et al. [145] for systemising knowledge of TEE attestation applications.

Curated lists of TEE publications. Schiavoni [146] maintains a curated list of SGX papers while Novella [147] maintains a similar list for TrustZone publications. Whereas the former aims to list all peer-reviewed publications regarding SGX, the latter focuses on attacks against TrustZone-based TEEs and is primarily composed of technical reports, blog postings, and hacking conference presentations.

Attacks against TEEs. There are also other surveys on TEEs not directly relevant to our work, for example, how TEEs reduce attack surface, but do not eliminate it. Zhao et al. [15] systemise knowledge of hardware security support for TEEs. Numerous attacks have been launched against TEE protection mechanisms and TA implementations [148]. Researchers and practitioners target security flaws and propose solutions for real-world applications, for example, Cerdeira et al. [149] and Koutroumpouchos et al. [150] present a security analysis of popular TrustZone-assisted TEE systems.

TEE technology provides a variety of use situations in which code, data, or an application requires hardware protection. The following are the most prevalent usage scenarios [48].

Digital rights management. Copyright holders frequently use TEEs to prevent consumers from copying video or audio [125]. TEEs protect digitally encoded media on connected devices including smartphones, tablets, and high-definition televisions [151, 152]. Along with the fact that the TEE and the device's display are connected via a protected hardware channel, this prohibits the device's owner from reading stored secrets.

Online payments. Mobile wallets, peer-to-peer payments, cryptocurrency wallets, and the use of a mobile device as a point of sale terminal all have well-defined security requirements. Blockchain systems use lightweight clients, which outsource the computational and storage load over full blockchain nodes [49]. It is possible to use a TEE to protect privacy of the light clients without compromising the performance of the assisting full nodes [49]. TEEs can be used as trusted backend systems to provide the necessary security to facilitate financial transactions. This may necessitate the entry of a PIN, password, or biometric identifier by the user.

Authentication. TEEs are commonly used to implement biometric identity methods (facial recognition, fingerprint sensor, and voice authorisation). For instance, the Android OS saves fingerprint biometrics in the TEE because it is inaccessible and encrypted from the ordinary OS environment [153]. Often, bio-

metric identifications are convenient to use and more difficult to steal than PINs and passwords. TEEs are utilised to protect the biometric identification method. Similarly to biometric identification information, cryptographic private keys can also be stored in the TEE. Combining the biometric identification information and the private keys allows passwordless authentication standards such as Apple's passkeys [144].

Trusted cloud. Typically, when a cloud (the server or the backend) is compromised, the adversary gains access to the cloud's processes and data. TEEs provide protection against compromised infrastructure: the adversary is unable to access selective parts of the TA, which safeguards sensitive code and data.

Privacy-preserving data analysis. Machine learning has become an essential part of data processing in several application domains such as healthcare, stock prediction, and artificial intelligence. Sometimes these applications process sensitive data, and to protect said data a TEE-based solution can be use to maintain the integrity of the machine learning process and prevent attacks [154].

Runtime integrity. TEEs can be used for runtime integrity, such as real-time kernel protection. If an attacker targets kernel binaries, the security monitoring service can shut down if it is isolated in a secure environment [155].

Secure modular programming. As it decouples functionalities into small, self-contained modules, modular programming is an efficient way to build software architectures for software assets that encourages reuse. In this instance, each module contains everything necessary to perform its intended function, and the TEE permits execution of the module while protecting it from the vulnerabilities of other modules.

II. Methodology

A. Collecting references for the review

We begin our search for scientific literature with *Google Scholar*[1], *arXiv open-access archive*[2], *the DBLP computer science bibliography*[3], *ACM Digital Library*[4], and *IEEE Xplore*[5] using TEE-related search terms, such as "TEE", "Trusted Execution Environment", "OP-TEE", "TrustZone", "(Intel) SGX", "AMD SEV", "confidential computing", etc. While this paints an overall picture of TEE-related scholarly work, it does not cover more applied aspects such as toolkits and deployments.

To address this gap, we then mined real source code using the Sourcegraph[6] search engine, to find examples of practical TEE utilisation. Sourcegraph covers GitLab, GitHub, and BitBucket, as well as other public software source repositories. Table I details our search terms regarding Sourcegraph, with examples[7]. The most difficult aspect of the mining process was locating appropriate TEE applications, development frameworks, and container repositories. Typically, a keyword search yields thousands of repositories. These repositories contain OSs and kernels, as well as forks and projects with work-in-progress status. In addition to specifying the search type as code, the first strategy for searching repositories was to provide meaningful keywords. Then, sort or filter the results to identify the most relevant ones, and finally, manually examine the results.

We based our selection of important phrases on the constants, variables, and functions utilised in the source code of each TEE-based application. The alternative method for picking specific search phrases was to consult the documentation of various TEE-based frameworks and containers, such as the GlobalPlatform API [156]. It reveals applications and other frameworks, containers, and repositories. However, this required manually combing through each repository to obtain the desired results.

B. Dimensions for knowledge systematisation

Based on related work and our observations while gathering and reviewing the publications, we organise the TEE literature and practical work as follows.

In Section III we address RQ1. Our goal is to assist the reader in comprehending TEEs, how they are utilised, and when, how, and why they could be used. To accomplish this, we tag the applications with 91 distinct keywords, which we merge into 17 primary category keywords based on initial similarities. We discover that the primary 17 drivers for using TEE for data protection in application development are *privacy*, *machine learning*, *integrity*, *confidentiality*, *cryptography*, *content sharing*, *cloud computing*, *access control*, *finance*, *secure storage*, *blockchain*, *secure channels*, *attestation*, *network security*, *medical data*, *computer games*, and *smart contracts*. See Figure 2 for an approximate hierarchy of the primary categories. Applications fall under these categories, and frequently many of them overlap. Consequently, this is the classification we utilise while reviewing existing TA demonstrations and practical implementations in Table II.

In Section IV we address RQ2. We compare the software frameworks targeting developers. It is difficult to compare TEE software development tools due to lack of similar work and public information about their features. Hence, we compile Table III detailing the available tools, their software licences, and the hardware they support.

In Section V we address RQ3. We organise the TEE containers for the developers. Again, it is difficult to directly compare TEE container tools due to the absence of shared and unique characteristics. In addition, some containers are not actively developed, while others, such as the Enarx container, are updated every month with new features. In response, we compile Table IV, which details the available tools, interfaces, software licences, and hardware supported by each vendor.

C. Limitations and bias

Lack of documentation of closed source proprietary systems. Companies that own proprietary solutions utilising TEEs

[1] https://scholar.google.com/
[2] https://arxiv.org/
[3] https://dblp.org/
[4] https://dl.acm.org/
[5] https://ieeexplore.ieee.org/
[6] https://sourcegraph.com/
[7] https://sourcegraph.com/search?q=context:global+Op-TEE&patternType=literal

TABLE I
USING THE SOURCEGRAPH SEARCH ENGINE, WE COMPILE REAL-WORLD APPLICATIONS OF TEES WITH THE PROVIDED SEARCH TERMS.

Search terms	Applications	Containers	Frameworks
SGX_CREATE_ENCLAVE_EX_PCL_BIT_IDX	1[139]	1[157]	2[52, 58]
TEEC_InvokeCommand	5[31, 39, 42, 80, 158]	0	1[18]
SGX_CREATE_ENCLAVE_EX_SWITCHLESS	3[159–161]	0	2[7, 9]
TEEC_MEMREF_TEMP_OUTPUT	3[162–164]	0	0
sgx_enclave_id_t	2[25, 36]	1[165]	1[63]
TEEC_RegisterSharedMemory	0	0	1[166, 167]
enarx	2[35, 116]	0	0

typically withhold information about their systems from the public. Therefore, it is difficult to find detailed information regarding closed source proprietary solutions that employ TEEs. Because of this, the data we collected may be biased towards open source data and may not provide an all-encompassing viewpoint of reality. For instance, there may be many more proprietary closed source applications and development frameworks for ARM TrustZone than we presented in this paper.

Date of initial release. It is often difficult to discover when a specific application, framework, or container was first released. Due to this, the publication year information in Figure 1 may not be entirely accurate.

Manual keyword search. The likelihood of omitting relevant repositories is the most significant shortcoming of manual search. Although using Sourcegraph as a repository search engine simplifies the search process, it also generates a large number of irrelevant results, and there is a chance of missing other applications, development frameworks, and containers that employ different keywords not on our list.

III. APPLICATION SCENARIOS FOR TEE

RQ1: What are applications of TEE?

The TEE isolates and protects the TA code and data in terms of confidentiality and integrity. While we may be unaware, a large number of gadgets around us, most notably smartphones, set-top boxes, videogame consoles, and Smart TVs, utilise a TEE. The number of gadgets utilising a TEE and designed for many different purposes results in a wide range of use cases. These use cases vary from everyday user applications to backend services, such as mobile financial services and cloud services [48].

In Figure 2, we create an approximate hierarchy of the key use cases. To address RQ1, Table II combines TEE application scenarios based on our categorisation. We gathered a total of 95 application use cases in Table II.

According to Table II, the vast majority of TEE applications operate on Intel SGX, ARM TrustZone, or both. Only a minority of applications operate on other platforms such as AMD SEV, RISC-V, or GPU TEEs. All of the references we collected fit within the 17 categories outlined in Section II. The categories and the number of references corresponding to each category are the following:

1) *Privacy*: 33 references

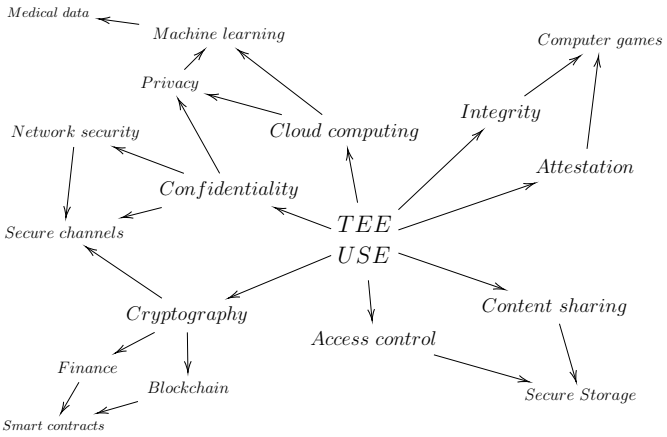

Fig. 2. We define which classification aggregates the usage examples in a meaningful way after reviewing the TEE example applications. This is an approximate hierarchy of the key use cases.

2) *Integrity*: 30 references
3) *Confidentiality*: 24 references
4) *Machine learning*: 19 references
5) *Attestation*: 14 references
6) *Cloud computing*: 14 references
7) *Access control*: 13 references
8) *Content sharing*: 11 references
9) *Blockchain*: 10 references
10) *Cryptography*: 10 references
11) *Finance*: 8 references
12) *Secure storage*: 7 references
13) *Secure channels*: 6 references
14) *Smart contracts*: 5 references
15) *Computer games*: 4 references
16) *Medical data*: 3 references
17) *Network security*: 2 references

On this basis, the majority of TEE applications aim to secure user privacy. The protection of data integrity and confidentiality is another important feature of TEEs. Use cases based on machine learning and attestation are also common. Cloud computing is frequently associated with machine learning applications, and is a quite prevalent category of TEE use cases. Application domains surrounding access control, content sharing, blockchain, cryptography, finance,

TABLE II
WE CLASSIFIED TEE APPLICATION SCENARIOS INTO 17 GROUPS.

			Privacy	Integrity	Confidentiality	Machine learning	Attestation	Cloud computing	Access control	Content sharing	Blockchain	Cryptography	Finance	Secure storage	Secure channels	Smart contracts	Computer games	Medical data	Network security	Intel SGX	ARM TrustZone	Open source	Real deployments
2015	AdAttester: Secure Online Mobile Advertisement Attestation Using TrustZone	[137]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	SeCReT: Secure Channel between Rich Execution Environment and TEE	[127]	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	○
	Using TEEs in Two-factor Authentication: comparing approaches	[135]	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	VC3: Trustworthy Data Analytics in the Cloud Using SGX	[118]	●	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
2016	A Case for Protecting Computer Games With SGX	[128]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	●	○	○	○
	Oblivious Multi-Party Machine Learning on Trusted Processors	[111]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Screen after Previous Screens: Spatial-Temporal Recreation of Android App Displays from Memory Images	[103, 163]	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●
	Secure Content-Based Routing Using Intel Software Guard Extensions	[119, 168]	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	●	●
	Town Crier: An Authenticated Data Feed for Smart Contracts	[138, 169]	○	○	○	○	○	○	●	○	●	○	○	○	○	●	○	○	○	●	○	●	●
2017	A Formal Foundation for Secure Remote Execution of Enclaves	[97]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Enhancing Security and Privacy of Tor's Ecosystem by Using TEEs	[79, 170]	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Establishing Mutually Trusted Channels for Remote Sensing Devices with TEEs	[129]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	IRON: Functional Encryption using Intel SGX	[120]	○	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○
	Komodo: Using verification to disentangle secure-enclave hardware from software	[73, 171]	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●
	MIPE: a practical memory integrity protection method in a TEE	[67]	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	Private Contact Discovery Service	[139]	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Securing Data Analytics on SGX with Randomization	[91, 172]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	SGX-BigMatrix: A Practical Encrypted Data Analytic Framework With Trusted Processors	[104]	●	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	SGX-Log: Securing System Logs With SGX	[112, 173]	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	TrustJS: Trusted Client-side Execution of JavaScript	[85]	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
2018	CYCLOSA: Decentralizing Private Web Search through SGX-Based Browser Extensions	[121]	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	DelegaTEE: Brokered Delegation Using TEEs	[92]	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Graviton: TEEs on GPUs	[105]	○	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	LibSEAL: revealing service integrity violations using trusted execution	[130, 174]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Obscuro: A Bitcoin Mixer using TEEs	[140, 175]	●	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	●	○	●	●
	PubSub-SGX: Exploiting TEEs for Privacy-Preserving Publish/Subscribe Systems	[98, 176]	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	●	●
	SafeBricks: Shielding Network Functions in the Cloud	[113, 177]	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	●	●
	SafeKeeper: Protecting Web Passwords using TEEs	[86, 178]	○	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	TizenFX	[80]	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	●	●
2019	BITE: Bitcoin Lightweight Client Privacy using Trusted Execution	[49]	●	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	●	○	○	○
	Clemmys: towards secure remote execution in FaaS	[59]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Forward and Backward Private Searchable Encryption with SGX	[99]	●	○	○	○	○	●	○	○	○	●	○	●	○	○	○	○	○	●	○	○	○
	Fuzzing OP-TEE with AFL	[45, 162]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●
	LightBox: Full-stack Protected Stateful Middlebox at Lightning Speed	[131, 179]	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	●	●
	NeXUS: Practical and Secure Access Control on Untrusted Storage Platforms using Client-Side SGX	[106, 180]	○	●	●	○	●	○	●	○	○	●	○	●	○	○	○	○	○	●	○	●	●
	OPERA: Open Remote Attestation for Intel's Secure Enclaves	[141]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	OP-TEE based keymaster and gatekeeper HIDL HAL	[42]	○	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	●	●
	PrivaTube: Privacy-Preserving Edge-Assisted Video Streaming	[75]	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○
	SDK for the Valve Steam Link	[39]	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	●	●	●
	SecTEE: A Software-based Approach to Secure Enclave Architecture Using TEE	[122]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	ShieldStore: Shielded In-memory Key-value Storage with SGX	[93, 159]	●	●	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware	[81, 181]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	StreamBox-TZ: Secure Stream Analytics at the Edge with TrustZone	[54]	●	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	Teechain: a secure payment network with asynchronous blockchain access	[69, 182, 183]	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	●	●
	Trust more, serverless	[64]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Using TEEs for Secure Stream Processing of Medical Data - (Case Study Paper)	[2]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○
	WebAssembly Micro Runtime (WAMR)	[36]	○	●	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	●	●
	ZLiTE: Lightweight Clients for Shielded Zcash Transactions Using Trusted Execution	[87]	○	○	○	○	○	○	●	●	○	●	○	○	○	○	○	○	○	●	○	○	○
2020	BDTF: A Blockchain-Based Data Trading Framework with TEE	[70, 184]	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	●	○
	BlackMirror: Preventing Wallhacks in 3D Online FPS Games	[132]	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	●	○	○	○
	Custos: Practical Tamper-Evident Auditing of Operating Systems Using Trusted Execution	[65]	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	CVShield: Guarding Sensor Data in Connected Vehicle with TEE	[107]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	DarkneTZ: towards model privacy at the edge using TEEs	[76, 158]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Design and Implementation of Hardware-Based Remote Attestation for a Secure Internet of Things	[34]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	Fine-Grained Access Control-Enabled Logging Method on ARM TrustZone	[46]	●	○	●	○	○	○	●	○	○	●	○	●	○	○	○	○	○	○	●	○	○
	GOAT: GPU Outsourcing of Deep Learning Training With Async. Probabilistic Integrity Verification	[40]	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Keybuster	[31, 102]	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	●	●
	MAGE: Mutual Attestation for a Group of Enclaves without Trusted Third Parties	[43, 161]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	MobileCoin: Private payments for mobile devices	[25, 185]	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	●	○	●	●
	Privacy-preserving Payment Channel Networks using TEE	[88]	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○
	ProximiTEE: Hardened SGX Attestation by Proximity Verification	[114]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Reboot-Oriented IoT: Life Cycle Management in TEE for Disposable IoT devices	[142]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
	SafeTrace: COVID-19 Self-reporting with Privacy	[28]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	●	○	●	●
	Secure Cloud Storage with Client-side Encryption using a TEE	[123]	○	●	●	○	○	●	○	○	○	●	○	●	○	○	○	○	○	●	○	○	○
	secureTF: A Secure TensorFlow Framework	[82]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	SeGShare: Secure Group File Sharing in the Cloud using Enclaves	[94]	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○
	SENG, the SGX-Enforcing Network Gateway: Authorizing Communication from Shielded Clients	[50, 160]	○	●	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Telekine: Secure Computing with Cloud GPUs	[60]	○	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Towards Formalization of Enhanced Privacy ID (EPID)-based Remote Attestation in Intel SGX	[100]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	TZ4Fabric: Executing Smart Contracts with ARM TrustZone: (Practical Experience Report)	[55, 186]	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	●	●	●
	TZ-MRAS: A Remote Attestation Scheme for the Mobile Terminal Based on ARM TrustZone	[37]	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○
2021	Atlas: Automated Scale-out of Trust-Oblivious Systems to TEEs	[124, 187]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Building Enclave-Native Storage Engines for Practical Encrypted Databases	[41]	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	○	○	●	○	○	○
	Distributed Learning in TEE: A Case Study of Federated Learning in SGX	[89]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Enarx Shim SGX	[35]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Formal Verification of a TEE-Based Architecture for IoT Applications	[47]	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	IceClave: A TEE for In-Storage Computing	[83]	○	●	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○
	IvyCross: A Trustworthy and Privacy-preserving Framework for Blockchain Interoperability	[51]	●	○	○	○	○	○	●	●	○	○	○	●	○	○	○	○	○	●	○	○	○
	MeetGo: A TEE for Remote Applications on FPGA	[66]	○	●	●	●	○	○	○	○	○	●	○	●	○	○	○	○	○	●	○	○	○
	Memory-Efficient Deep Learning Inference in TEEs	[95]	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Poster: FLATEE: Federated Learning Across TEEs	[101]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	PPFL: privacy-preserving federated learning with TEEs	[77, 164]	●	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	●	●
	Privacy-preserving genotype imputation in a TEE	[38, 188]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	●	○	○	●
	S2Dedup: SGX-enabled secure deduplication	[71, 189]	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	●	●
	ShuffleFL: gradient-preserving federated learning using TEE	[108]	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	TEEKAP: Self-Expiring Data Capsule using TEE	[6, 115]	●	○	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	●	○	○	○
	Tora: A Trusted Blockchain Oracle Based on a Decentralized TEE Network	[133, 190]	●	○	○	○	○	○	○	●	●	○	○	○	●	○	○	○	○	●	○	●	●
	Toward a Secure, Rich, and Fair Query Service for Light Clients on Public Blockchains	[143]	●	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	●	○	○	○
	Trustzone-based secure lightweight wallet for hyperledger fabric	[44]	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	●	○	○
	TZ-Container: protecting container from untrusted OS with ARM TrustZone	[61]	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○
	TZMon: Improving mobile game security with ARM trustzone	[56, 191]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	●	●	●
2022	Exploring Widevine for Fun and Profit	[125]	○	●	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	●	○	●
	MMLedger: A ledger for confidential computing shims for tracking memory management system calls	[116]	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●
	OLIVE: Oblivious and Differentially Private Federated Learning on TEE	[134]	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○
	Supporting Passkeys	[144]	○	●	●	○	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	●

and secure storage are also somewhat prominent, albeit they are noticeably less prevalent than the use cases previously stated. TEEs are also used to improve the security of smart contracts and harden secure channels. Anti-cheat safeguards for computer games, medical data protection, and network security hardening are examples of highly specific TEE uses with few existent applications.

The number of applications utilising TEEs has steadily increased since 2015. Approximately half of the references we collected are from 2020 or after. Only about half of the reference applications have been deployed to actual users, according to our study. The vast majority of reference applications deployed to actual users are licenced under an open source licence. Notably, despite this, a large number of proprietary applications with closed source licences comparable to the Widevine DRM component [125] utilise TEEs. Typically, these proprietary applications are not accompanied by any public documentation or scholarly studies, hence they are largely absent from our work. Therefore, Table II should not be misconstrued as providing an accurate representation of all applications that employ TEEs.

IV. Tools for developing TEE software

RQ2: Which TEE software tools are available for TA development?

TABLE III
TEE software development tools.

Framework		Intel SGX	AMD SEV	ARM TrustZone	RISC-V	Open source
Asylo	[63]	●	●	○	○	●
Edgeless RT	[13]	●	○	○	○	●
Intel SGX SDK	[72]	●	○	○	○	●
Keystone	[11, 192]	○	○	○	●	●
Occlum's fork of Intel SGX SDK	[9]	●	○	○	○	●
Open-TEE	[110]	●	●	●	○	●
OP-TEE	[136, 166]	○	○	●	○	●
Open Enclave SDK	[58]	●	○	●	○	●
QSEE SDK	[117, 193, 194]	○	○	●	○	○
Samsung Knox SDK	[53]	○	○	●	○	○
Samsung Knox Tizen SDK	[21]	○	○	●	○	○
Samsung mTower	[18]	○	○	●	○	●
Samsung TEEGRIS SDK	[57]	○	○	●	○	○
Sanctum	[84, 195, 196]	○	○	○	●	●
SecGear	[7]	●	○	○	○	●
Teaclave SGX SDK	[52]	●	○	○	○	●
Teaclave TrustZone SDK	[8]	○	○	●	○	●
TEEKAP	[6]	●	○	○	○	●
Confidential Consortium	[24]	●	○	○	○	●
Trustonic TEE SDKs	[10, 167, 197]	○	○	●	○	○
Trusty TEE	[78]	●	○	●	○	●
Webinos	[126, 198]	○	○	●	○	●

TAs entail establishing a distinction between the normal and secure worlds. Because *software cannot protect software*, hardware serves as a barrier between the two realms. This is a radically different approach to application development for software engineers. Numerous middleware frameworks are available to assist developers with TEE development, deployment, and maintenance. These frameworks permit the development of TEE-enabled applications without requiring the developer to master a specific low-level CPU TEE API. To address RQ2, Table III combines tools for developing TEE software.

Developers have a good variety of frameworks available for them in different categories, which are glanced as an overview in this paper. The referenced frameworks are available as open source or can be brand focused commercial solutions by different manufacturers, like the C and C++ application-focused Open Enclave SDK or Samsung Knox SDK for Samsung Android devices [53, 58]. However, the majority of the referred frameworks focus on Intel SGX or ARM TrustZone, as Table III shows.

The frameworks are made to address a variety of practical purposes to smoothen development efforts. Several frameworks focus on mobile devices and wearables, where the intent is to provide ready-made APIs to support application development [21, 53, 57]. The framework references are also focused on IoT devices or web applications, but due to the wide range of programming language support, the frameworks cover also many other areas [24, 126, 197, 198]. Some of the frameworks are focused on or support very niche areas, like Trustonic's Kinibi-520a SDK [10], where Symmetric-Multi-Processing enables development on biometric functions, like fingerprint scanning and face recognition.

The choice of development framework by the developer is usually severely constrained by the hardware architecture. For instance, mobile application developers are restricted to options that are compatible with ARM TrustZone. We find that open source development frameworks such as OP-TEE [136], Open Enclave SDK [58], Teaclave TrustZone SDK [8], and Trusty TEE [78] support TrustZone at least in some capacity and are still actively maintained. These frameworks may provide open source alternatives for mobile application developers, who have traditionally been limited to proprietary closed source frameworks such as the Samsung Knox SDK [53] or Trustonic's TEE SDKs [10, 167, 197]. Nevertheless, many open source frameworks only support specific platforms, so proprietary SDKs may remain the only option for developers on unsupported platforms. Some of the open source frameworks such as Open-TEE [110] and Webinos [198] are deprecated and/or no longer under active development.

V. Middleware TEE containers

RQ3: What types of TEE containers are available for TA packaging?

For an application to function on any TEE technology, the development process must adhere to required framework changes so that it may execute within TEE. This makes the procedure difficult and time-intensive for application developers. In addition, attestation must be implemented because it is the TEE's verification process prior to trusting the application. For addressing the usability issue on different TEE technologies, a set of Trusted containers (tcons) enables either direct execution of unmodified binary code inside a TEE or automatic transformation of source code prior to loading it

TABLE IV
TEE CONTAINERS.

Container		libc wrapper	LibOS	WASI	Intel SGX	AMD SEV	Open source
AccTEE	[33, 199]	○	○	●	●	○	●
Anjuna	[74]	○	●	○	●	●	○
Apache Teaclave	[16]	○	○	●	●	○	●
Chancel	[26]	●	○	○	●	○	○
Decentriq	[20]	○	●	○	●	○	○
Deflection	[32, 200]	●	○	○	●	○	●
EGo SDK	[17]	●	○	○	●	○	●
Enarx	[14]	○	○	●	●	●	●
Fortanix EDP	[90]	○	●	○	●	○	●
GOTEE	[30, 201]	○	○	○	●	○	●
Gramine	[62, 202]	○	●	○	●	○	●
MesaPy	[19, 203]	○	○	○	●	○	●
Mystikos	[12]	○	●	○	●	○	●
Occlum	[22, 165]	○	●	○	●	○	●
Ratel	[23, 204]	●	○	○	●	○	●
Ryoan	[68, 205]	●	○	○	●	○	●
SCONE	[96, 206]	●	○	○	●	○	●
SGX-LKL	[27, 207]	○	●	○	●	○	●
Twine	[29, 157]	○	○	●	●	○	●
vSGX	[109]	○	○	○	○	●	●

into a TEE executable [5]. In order to address RQ3, Table IV enumerates middleware TEE containers. We collect 20 distinct containers, in addition to identifying supporting hardware and the application middleware interface.

We discover that the bulk of tcons are developed for Intel SGX, yet still a few support AMD SEV. In addition, we find no tcons that support TrustZone TEE technology, confining mobile application developers to frameworks. There are just three tcons available for AMD SEV, limiting the number of possible tcon-based development approaches.

The majority of tcons utilise wrappers around the C standard library (libc) as an application middleware interface. Furthermore, a library OS (LibOS) removes the traditional boundary between kernel and user space, allowing applications to access system resources directly with much lower overhead. The LibOS concept predates TEE technologies by at least a decade, motivated by applications in the embedded space due to severe resource constraints [208]. A LibOS typically has extremely restricted functionality, which is in fact conducive to the TEE concept by reducing both the Trusted Computing Base (TCB) and attack surface.

To this end, most TEE containers provide a software shim between libc or the tcon's LibOS and unmodified applications: the goal is to intercept and control the function calls (including system calls) issued by binary code in order to arbitrate the application's interactions with the underlying OS. Moreover, the WebAssembly System Interface can also be used in a similar fashion [5]. Another compelling aspect is all the tcons included in our study are open source except three, thus enabling easy access for setup and usage. Comparing the year of debut for the majority of tcons, 2021 is the pinnacle. In addition to the mentions of review, application, and framework, Figure 1 depicts the reference publication of the containers across different years. The most important security aspect of TEE containers is to ensure execution in an isolated environment and to achieve complete attestation.

A recent trend seems to be containers that support multiple hardware architectures. The objective is to allow developers to adapt the same program to many platforms without having to alter the source code. Enarx [14] is a good example of such a tcon. Recently published vSGX [109] supports directly running SGX-enabled applications inside AMD SEV.

VI. CONCLUSION

This article organised the history and current state of TEE applications (use cases), frameworks, containers, and reviews. We collected a total of 0 references for this review between 2022-03-11 and 2022-06-07 and organised the knowledge. This includes both academic and applied work on TEE technologies.

The following are our key conclusions:

1. The rate of TEE publications is increasing. We discovered that the rate of publication accelerates after 2015. The number of publications has continued to rise in 2019, 2020, and 2021. Approximately half of the references we collected are from 2020 or after. This illustrates the recent surge in the academic interest in TEE technologies.

2. Intel SGX and ARM TrustZone are the most researched. Most of the publications are demonstrating application use cases, and Intel SGX is the most popular hardware. Indeed, the vast majority of TEE applications operate on Intel SGX, ARM TrustZone, or both. Only a minority of applications operate on other platforms such as AMD SEV, RISC-V, or GPU TEEs.

3. Privacy is the primary motivation for open source TAs. Additionally, we analysed the primary elements of the execution and data people attempt to secure with their TAs. We gathered a total of 95 application use cases in Table II. *Privacy* (33 references), *Integrity* (30 references), *Confidentiality* (24 references) and *Machine learning* (19 references) are the most common of the 17 primary drivers to use TEE for data protection in application development. Interestingly, only about half of the reference applications are available to actual users.

4. Open source TEE frameworks help TA creation. Typically, a developer must make laborious framework-specific modifications to the original application in order for it to run within a TEE. Many of the development frameworks are open source and available for free. We listed 22 middleware frameworks available to aid developers with TEE deployment, out of which 17 are open source.

5. Open source TEE tcons are gaining popularity. A trusted container (tcon) solves the usability issue raised in the previous paragraph by enabling either the direct execution of unmodified binary code within a TEE or the automatic transformation of source code prior to loading a TEE executable. Most of the tcons are free and open source. We provided a list of 20 tcons

that eliminate the need for software developers to write TEE-related code, out of which 17 are open source.

6. Current tcons support primarily Intel SGX or AMD SEV. The choice of development framework and tcon by the developer is severely constrained by the hardware architecture. For instance, mobile application developers are restricted to options that are compatible with ARM TrustZone, which means there are no tcons available and a limited number of frameworks to choose from, the majority of which are closed source proprietary frameworks. Some recent tcons, such as Enarx [14] and vSGX [109], enable the execution of the same application within TEEs based on multiple hardware architectures without requiring code modifications. Typically, though, tcons only support Intel SGX, AMD SEV, or both.

A. Future work

1. Real world security of TEE implementations. There are frequently discrepancies between ideal systems and real-world implementations, as real-world implementations are often susceptible to side-channel attacks (SCAs). As we mentioned in Section I-A, Fei et al. [148] previously systematised knowledge of real-world SCAs against Intel SGX, whereas Cerdeira et al. [149] and Koutroumpouchos et al. [150] provided similar systemisations for ARM TrustZone. To our knowledge, however, there have been no surveys that combine knowledge of SCAs against various hardware architectures into a single survey.

In addition, if we observe the security properties of tcons, they might fail to achieve all their security goals in ideal systems. The majority of tcons lack meaningful documentation concerning API protection mechanisms, including the arguments provided and values returned from the OS which might result in information leaks [5]. Another challenge for containers is maintaining isolated execution in the presence of SCA-enabled attackers. In order to comprehend the effectiveness of containers against these non-exhaustive attack varieties, we believe that further research is required. That is, research which unifies attacks and defences against different architectures under select threat models would benefit developers who wish to employ TEEs by providing a holistic view of the security of TEEs in the real world. Additionally, it would be useful to know whether current applications that utilise TEEs implement SCA mitigations or are susceptible to them.

2. TrustZone TEE technology and containers. In Section V, to our surprise we discovered no tcons supporting TrustZone TEE technology. We speculate this is mostly due to the drastically different application domains between the embedded space (e.g., ARM architectures) and the server space (e.g., Intel architectures). TEE deployments leveraging TrustZone are often highly vendor specific, and the scope of tasks performed by TrustZone-based TEEs usually encompasses not only trusted execution of applications, but also broader platform security. In summary, vendors of TrustZone-based TEEs seem scantly concerned with making it easier for third parties to execute unmodified or instrumented code leveraging TEEs.

3. RISC-V, Sanctum, and Keystone. According to academic references, RISC-V TEE technologies are *interesting*, but few publications are available about them. The scientific community is ideally suited to pursue the objective of open source hardware, which is undeniably a concrete development step. The objective is to create a secure and trustworthy hardware-backed enclave for RISC-V that is freely available under open source licences. Sanctum [84] and Keystone [11] are seminal steps in this direction, yet we are unaware of any deployments. This lack of mainstream hardware inhibits growth of the surrounding software ecosystem, somewhat analogous to TrustZone-based TEE technologies such as On-board Credentials (ObC) [209] in the 2000s: it is clear ObC predates unified TEE software architectures such as the GlobalPlatform API [156], yet such standardisation and unification efforts arguably emerged too late to prevent fragmentation of the software ecosystem. In summary, as a community we should steer TA software development in a consistent and narrow fashion, and to do that we need mainstream hardware available with TEE-relevent hardware-assisted security features that are open and accessible to developers.

4. Comparison between TEEs and other privacy-preserving computing techniques. TEE is one method to achieve private computing, but there are other options. For example, in 2014, Signal App developers wrestled with the practical question, "How do we determine which contacts are registered with a service, without revealing the contacts to the service?"[8]. In 2017, after evaluating potential solutions to the problem, they resolved contact discovery using the Intel SGX TEE service[9]. Prior to that, they considered a variety of alternatives, including hashing, bloom filters, encrypted bloom filters, sharded bloom filters, private information retrieval, and private set intersection.

There are additional technologies available to application developers, such as homomorphic encryption (a type of encryption that enables users to perform computations on its encrypted data without first decrypting it), secure multi-party computation (computation of a function over inputs while maintaining their confidentiality), and functional encryption (restricted secret keys that enable a key holder to learn a specific function of encrypted data while learning nothing else).

Considering how challenging it is to compare solutions, even for the relatively basic case of private contact discovery, It would be very beneficial and practical to provide guidelines for: How should a developer choose between these technologies? What are their trust and threat models? What are their security limitations? How can they be securely combined? Under what circumstances are they feasible choices?

Thus, research to resolve these practical questions and help developers in selecting TEE-related technologies and

[8]https://signal.org/blog/contact-discovery/

[9]https://signal.org/blog/private-contact-discovery/

developing their TAs (or not using it and selecting other technical methods!) would immediately generate more secure and privacy-centric technical solutions.

Acknowledgments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under project numbers 804476 (SCARE) and 952622 (SPIRS).

Supported in part by the Cybersecurity Research Award granted by the Technology Innovation Institute (TII) in UAE and Technology Innovation Institute's Secure Systems Research Center (SSRC) in UAE.

REFERENCES

[1] T. Fischer, C. Lesjak, D. Pirker, and C. Steger, "RPC based framework for partitioning IoT security software for trusted execution environments," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 430–435.

[2] C. Segarra, R. Delgado-Gonzalo, M. Lemay, P. Aublin, P. R. Pietzuch, and V. Schiavoni, "Using trusted execution environments for secure stream processing of medical data - (case study paper)," in *Distributed Applications and Interoperable Systems - 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17-21, 2019, Proceedings*, ser. Lecture Notes in Computer Science, J. Pereira and L. Ricci, Eds., vol. 11534. Springer, 2019, pp. 91–107. [Online]. Available: https://doi.org/10.1007/978-3-030-22496-7_6

[3] G. Arfaoui, S. Gharout, and J. Traoré, "Trusted execution environments: A look under the hood," in *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2014, Oxford, United Kingdom, April 8-11, 2014*. IEEE Computer Society, 2014, pp. 259–266. [Online]. Available: https://doi.org/10.1109/MobileCloud.2014.47

[4] "Introduction to Trusted Execution Environment and ARM's TrustZone," https://embeddedbits.org/introduction-to-trusted-execution-environment-tee-arm-trustzone/, accessed: 2022-06-02.

[5] W. Liu, H. Chen, X. Wang, Z. Li, D. Zhang, W. Wang, and H. Tang, "Understanding TEE containers, easy to use? hard to trust," *CoRR*, vol. abs/2109.01923, 2021. [Online]. Available: https://arxiv.org/abs/2109.01923

[6] M. Gao, "Teekap," 2021, Latest release in 2022. [Online]. Available: https://github.com/MingyuanGao/TEEKAP

[7] openEuler, "secGear," 2020, latest release in 2022. [Online]. Available: https://github.com/openeuler-mirror/secGear

[8] The Apache Software Foundation, "Teaclave TrustZone SDK," 2021, Latest release in 2022. [Online]. Available: https://github.com/apache/incubator-teaclave-trustzone-sdk

[9] Occlum team, "Intel(R) Software Guard Extensions for Linux," 2020, latest release in 2022. [Online]. Available: https://github.com/occlum/linux-sgx

[10] L. Hanel, "Kinibi-520a: The latest Trustonic Trusted Execution Environment (TEE)," https://www.trustonic.com/technical-articles/kinibi-520a-the-latest-trusted-execution-environment-tee/, 2021.

[11] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanovic, and D. Song, "Keystone: An open framework for architecting trusted execution environments," in *Proceedings of the Fifteenth European Conference on Computer Systems*, ser. EuroSys '20, 2020.

[12] deislabs, "Mystikos," 2021, latest release in 2022. [Online]. Available: https://github.com/deislabs/mystikos

[13] Edgeless Systems, "Edgeless RT," 2020, Latest release in 2022. [Online]. Available: https://github.com/edgelesssys/edgelessrt

[14] Enarx, "Enarx," 2021, latest release in 2022. [Online]. Available: https://github.com/enarx/enarx

[15] L. Zhao, H. Shuang, S. Xu, W. Huang, R. Cui, P. Bettadpur, and D. Lie, "Sok: Hardware security support for trustworthy execution," *CoRR*, vol. abs/1910.04957, 2019. [Online]. Available: http://arxiv.org/abs/1910.04957

[16] The Apache Software Foundation, "Teaclave: A Universal Secure Computing Platform," 2020, latest release in 2022. [Online]. Available: https://github.com/apache/incubator-teaclave

[17] Edgeless Systems, "Welcome to EGo," 2021, latest release in 2022. [Online]. Available: https://docs.edgeless.systems/ego/#/

[18] Samsung, "mTower," 2019, latest release in 2022. [Online]. Available: https://github.com/Samsung/mTower

[19] H. Wang, M. Sun, Q. Feng, P. Wang, T. Li, and Y. Ding, "Towards memory safe python enclave for security sensitive computation," *CoRR*, vol. abs/2005.05996, 2020. [Online]. Available: https://arxiv.org/abs/2005.05996

[20] Decentriq, "Decentriq," 2021, latest release in 2022. [Online]. Available: https://www.decentriq.com/

[21] Samsung, "Welcome to the Knox Tizen SDK for Wearables," https://docs.samsungknox.com/dev/knox-tizen-sdk/index.htm, 2019, Latest release in 2021.

[22] Y. Shen, H. Tian, Y. Chen, K. Chen, R. Wang, Y. Xu, Y. Xia, and S. Yan, "Occlum: Secure and efficient multitasking inside a single enclave of intel SGX," in *ASPLOS '20: Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, March 16-20, 2020*, J. R. Larus, L. Ceze, and K. Strauss, Eds. ACM, 2020, pp. 955–970. [Online]. Available: https://doi.org/10.1145/3373376.3378469

[23] J. Cui, S. Shinde, S. Sen, P. Saxena, and P. Yuan, "Dynamic binary translation for SGX enclaves," *CoRR*, vol. abs/2103.15289, 2021. [Online]. Available: https://arxiv.org/abs/2103.15289

[24] Microsoft, "The Confidential Consortium Framework," 2019, Latest release in 2022. [Online]. Available: https://github.com/microsoft/CCF

[25] MobileCoin Foundation, "MobileCoin: Private payments for mobile devices," 2020, latest release in 2022. [Online]. Available: https://github.com/mobilecoinfoundation/mobilecoin

[26] A. Ahmad, J. Kim, J. Seo, I. Shin, P. Fonseca, and B. Lee, "CHANCEL: efficient multi-client isolation under adversarial programs," in *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/chancel-efficient-multi-client-isolation-under-adversarial-programs/

[27] C. Priebe, D. Muthukumaran, J. Lind, H. Zhu, S. Cui, V. A. Sartakov, and P. R. Pietzuch, "SGX-LKL: securing the host OS interface for trusted execution," *CoRR*, vol. abs/1908.11143, 2019. [Online]. Available: http://arxiv.org/abs/1908.11143

[28] SCRT Labs, "SafeTrace: COVID-19 Self-reporting with Privacy," 2020. [Online]. Available: https://github.com/scrtlabs/SafeTrace

[29] J. Ménétrey, M. Pasin, P. Felber, and V. Schiavoni, "Twine: An embedded trusted runtime for webassembly," in *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE, 2021, pp. 205–216. [Online]. Available: https://doi.org/10.1109/ICDE51399.2021.00025

[30] A. Ghosn, J. R. Larus, and E. Bugnion, "Secured routines: Language-based construction of trusted execution environments," in *2019 USENIX Annual Technical Conference, USENIX ATC 2019, Renton, WA, USA, July 10-12, 2019*, D. Malkhi and D. Tsafrir, Eds. USENIX Association, 2019, pp. 571–586. [Online]. Available: https://www.usenix.org/conference/atc19/presentation/ghosn

[31] shakevsky, "Keybuster," 2020. [Online]. Available: https://github.com/shakevsky/keybuster

[32] W. Liu, W. Wang, H. Chen, X. Wang, Y. Lu, K. Chen, X. Wang, Q. Shen, Y. Chen, and H. Tang, "Practical and efficient in-enclave verification of privacy compliance," in *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2021, Taipei, Taiwan, June 21-24, 2021*. IEEE, 2021, pp. 413–425. [Online]. Available: https://doi.org/10.1109/DSN48987.2021.00052

[33] D. Goltzsche, M. Nieke, T. Knauth, and R. Kapitza, "Acctee: A webassembly-based two-way sandbox for trusted resource accounting," in *Proceedings of the 20th International Middleware Conference, Middleware 2019, Davis, CA, USA, December 9-13, 2019*. ACM, 2019, pp. 123–135. [Online]. Available: https://doi.org/10.1145/3361525.3361541

[34] J. Ahn, I. Lee, and M. Kim, "Design and implementation of hardware-based remote attestation for a secure internet of things," *Wirel. Pers. Commun.*, vol. 114, no. 1, pp. 295–327, 2020. [Online]. Available: https://doi.org/10.1007/s11277-020-07364-5

[35] Enarx, "Enarx Shim SGX," 2021. [Online]. Available: https://github.com/enarx/enarx-shim-sgx

[36] Bytecode Alliance, "WebAssembly Micro Runtime (WAMR)," 2019, latest release in 2022. [Online]. Available: https://github.com/bytecodealliance/wasm-micro-runtime

[37] Z. Wang, Y. Zhuang, and Z. Yan, "TZ-MRAS: A remote attestation scheme for the mobile terminal based on ARM trustzone," *Secur. Commun. Networks*, vol. 2020, pp. 1 756 130:1–1 756 130:16, 2020. [Online]. Available: https://doi.org/10.1155/2020/1756130

[38] N. Dokmai, C. Kockan, K. Zhu, X. Wang, S. C. Sahinalp, and H. Cho, "Privacy-preserving genotype imputation in a trusted execution environment," *Cell Systems*, vol. 12, no. 10, pp. 983–993.e7, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405471221002891

[39] Valve Software, "SDK for the Valve Steam Link," 2019, latest release in 2021. [Online]. Available: https://github.com/ValveSoftware/steamlink-sdk

[40] A. Asvadishirehjini, M. Kantarcioglu, and B. A. Malin, "GOAT: GPU outsourcing of deep learning training with asynchronous probabilistic integrity verification inside trusted execution environment," *CoRR*, vol. abs/2010.08855, 2020. [Online]. Available: https://arxiv.org/abs/2010.08855

[41] Y. Sun, S. Wang, H. Li, and F. Li, "Building enclave-native storage engines for practical encrypted databases," *Proc. VLDB Endow.*, vol. 14, no. 6, pp. 1019–1032, 2021. [Online]. Available: http://www.vldb.org/pvldb/vol14/p1019-sun.pdf

[42] Linaro Security Working Group, "OP-TEE based keymaster and gatekeeper HIDL HAL," 2019, latest release in 2021. [Online]. Available: https://github.com/linaro-swg/kmgk

[43] G. Chen and Y. Zhang, "MAGE: mutual attestation for a group of enclaves without trusted third parties," *CoRR*, vol. abs/2008.09501, 2020. [Online]. Available: https://arxiv.org/abs/2008.09501

[44] W. Dai, Q. Wang, Z. Wang, X. Lin, D. Zou, and H. Jin, "Trustzone-based secure lightweight wallet for hyperledger fabric," *J. Parallel Distributed Comput.*, vol. 149, pp. 66–75, 2021. [Online]. Available: https://doi.org/10.1016/j.jpdc.2020.11.001

[45] M. Bogaard, "Fuzzing OP-TEE with AFL," https://static.linaro.org/connect/san19/presentations/san19-225.pdf, 2019.

[46] S. Lee, H. J. Jo, W. Choi, H. Kim, J. H. Park, and D. H. Lee, "Fine-grained access control-enabled logging method on ARM trustzone," *IEEE Access*, vol. 8, pp. 81 348–81 364, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2991431

[47] D. C. G. Valadares, Á. A. de Carvalho César Sobrinho, A. Perkusich, and K. C. Gorgônio, "Formal verification of a trusted execution environment-based architecture for iot applications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17 199–17 210, 2021. [Online]. Available: https://doi.org/10.1109/JIOT.2021.3077850

[48] S. Tamrakar, "Applications of Trusted Execution Environments (TEEs)," Doctoral thesis, School of Science, 2017. [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-7463-4

[49] S. Matetic, K. Wüst, M. Schneider, K. Kostiainen, G. Karame, and S. Capkun, "BITE: bitcoin lightweight client privacy using trusted execution," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, 2019, pp. 783–800. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/matetic

[50] F. Schwarz and C. Rossow, "Seng, the sgx-enforcing network gateway: Authorizing communication from shielded clients," in *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, S. Capkun and F. Roesner, Eds. USENIX Association, 2020, pp. 753–770. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/schwarz

[51] M. Li, J. Weng, Y. Li, Y. Wu, J. Weng, D. Li, and R. H. Deng, "Ivycross: A trustworthy and privacy-preserving framework for blockchain interoperability," *IACR Cryptol. ePrint Arch.*, p. 1244, 2021. [Online]. Available: https://eprint.iacr.org/2021/1244

[52] The Apache Software Foundation, "Teaclave SGX SDK," 2017, Latest release in 2022. [Online]. Available: https://github.com/apache/incubator-teaclave-sgx-sdk

[53] Samsung, "Knox SDK," https://developer.samsungknox.com/knox-sdk, 2018, Latest release in 2022.

[54] H. Park, S. Zhai, L. Lu, and F. X. Lin, "Streambox-tz: Secure stream analytics at the edge with trustzone," in *2019 USENIX Annual Technical Conference, USENIX ATC 2019, Renton, WA, USA, July 10-12, 2019*, D. Malkhi and D. Tsafrir, Eds. USENIX Association, 2019, pp. 537–554. [Online]. Available: https://www.usenix.org/conference/atc19/presentation/park-heejin

[55] C. Müller, M. Brandenburger, C. Cachin, P. Felber, C. Göttel, and V. Schiavoni, "Tz4fabric: Executing smart contracts with ARM trustzone: (practical experience report)," in *International Symposium on Reliable Distributed Systems, SRDS 2020, Shanghai, China, September 21-24, 2020*. IEEE, 2020, pp. 31–40. [Online]. Available: https://doi.org/10.1109/SRDS51746.2020.00011

[56] S. Jeon and H. K. Kim, "Tzmon: Improving mobile game security with ARM trustzone," *Comput. Secur.*, vol. 109, p. 102391, 2021. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102391

[57] Samsung, "SAMSUNG TEEGRIS SDK," https://developer.samsung.com/teegris/overview.html#SAMSUNG-TEEGRIS-SDK, 2017.

[58] Open Enclave, "Open Enclave SDK," 2018, Latest release in 2022. [Online]. Available: https://github.com/openenclave/openenclave

[59] B. Trach, O. Oleksenko, F. Gregor, P. Bhatotia, and C. Fetzer, "Clemmys: towards secure remote execution in faas," in *Proceedings of the 12th ACM International Conference on Systems and Storage, SYSTOR 2019, Haifa, Israel, June 3-5, 2019*, M. Hershcovitch, A. Goel, and A. Morrison, Eds. ACM, 2019, pp. 44–54. [Online]. Available: https://doi.org/10.1145/3319647.3325835

[60] T. Hunt, Z. Jia, V. Miller, A. Szekely, Y. Hu, C. J. Rossbach, and E. Witchel, "Telekine: Secure computing with cloud gpus," in *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, R. Bhagwan and G. Porter, Eds. USENIX Association, 2020, pp. 817–833. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/hunt

[61] Z. Hua, Y. Yu, J. Gu, Y. Xia, H. Chen, and B. Zang, "Tz-container: protecting container from untrusted OS with ARM trustzone," *Sci. China Inf. Sci.*, vol. 64, no. 9, 2021. [Online]. Available: https://doi.org/10.1007/s11432-019-2707-6

[62] C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: A practical library OS for unmodified applications on SGX," in *2017 USENIX Annual Technical Conference, USENIX ATC 2017, Santa Clara, CA, USA, July 12-14, 2017*, D. D. Silva and B. Ford, Eds. USENIX Association, 2017, pp. 645–658. [Online]. Available: https://www.usenix.org/conference/atc17/technical-sessions/presentation/tsai

[63] Google, "Asylo," 2018, Latest release in 2022. [Online]. Available: https://github.com/google/asylo

[64] S. Brenner and R. Kapitza, "Trust more, serverless," in *Proceedings of the 12th ACM International Conference on Systems and Storage, SYSTOR 2019, Haifa, Israel, June 3-5, 2019*, M. Hershcovitch, A. Goel, and A. Morrison, Eds. ACM, 2019, pp. 33–43. [Online]. Available: https://doi.org/10.1145/3319647.3325825

[65] R. Paccagnella, P. Datta, W. U. Hassan, A. Bates, C. W. Fletcher, A. Miller, and D. Tian, "Custos: Practical tamper-evident auditing of operating systems using trusted execution," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/custos-practical-tamper-evident-auditing-of-operating-systems-using-trusted-execution/

[66] H. Oh, K. Nam, S. Jeon, Y. Cho, and Y. Paek, "Meetgo: A trusted execution environment for remote applications on FPGA," *IEEE Access*, vol. 9, pp. 51 313–51 324, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3069223

[67] R. Chang, L. Jiang, W. Chen, Y. Xiang, Y. Cheng, and A. Alelaiwi, "MIPE: a practical memory integrity protection method in a trusted execution environment," *Clust. Comput.*, vol. 20, no. 2, pp. 1075–1087, 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-0833-4

[68] T. Hunt, Z. Zhu, Y. Xu, S. Peter, and E. Witchel, "Ryoan: A distributed sandbox for untrusted computation on secret data," *ACM Trans. Comput. Syst.*, vol. 35, no. 4, pp. 13:1–13:32, 2018. [Online]. Available: https://doi.org/10.1145/3231594

[69] J. Lind, O. Naor, I. Eyal, F. Kelbert, E. G. Sirer, and P. R. Pietzuch, "Teechain: a secure payment network with asynchronous blockchain access," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP 2019, Huntsville, ON, Canada, October 27-30, 2019*, T. Brecht and C. Williamson, Eds. ACM, 2019, pp. 63–79. [Online]. Available: https://doi.org/10.1145/3341301.3359627

[70] G. Su, W. Yang, Z. Luo, Y. Zhang, Z. Bai, and Y. Zhu, "BDTF: A blockchain-based data trading framework with trusted execution environment," in *16th International Conference on Mobility, Sensing and Networking, MSN 2020, Tokyo, Japan, December 17-19, 2020*. IEEE, 2020, pp. 92–97. [Online]. Available: https://doi.org/10.1109/MSN50589.2020.00030

[71] M. Miranda, T. Esteves, B. Portela, and J. Paulo, "S2dedup: Sgx-enabled secure deduplication," in *SYSTOR '21: The 14th ACM International Systems and Storage Conference, Haifa, Israel, June 14-16, 2021*, B. Wassermann, M. Malka, V. Chidambaram, and D. Raz, Eds. ACM, 2021, pp. 14:1–14:12. [Online]. Available: https://doi.org/10.1145/3456727.3463773

[72] Intel Corporation, "Intel(R) Software Guard Extensions for Linux* OS," 2016, Latest release in 2022. [Online]. Available: https://github.com/intel/linux-sgx

[73] A. Ferraiuolo, A. Baumann, C. Hawblitzel, and B. Parno, "Komodo: Using verification to disentangle secure-enclave hardware from software," in *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. ACM, 2017, pp. 287–305. [Online]. Available: https://doi.org/10.1145/3132747.3132782

[74] Anjuna, "Anjuna Confidential Cloud Software," 2018, latest release in 2022. [Online]. Available: https://www.anjuna.io/

[75] S. D. Silva, S. B. Mokhtar, S. Contiu, D. Négru, L. Réveillère, and E. Rivière, "Privatube: Privacy-preserving edge-assisted video streaming," in *Proceedings of the 20th International Middleware Conference, Middleware 2019, Davis, CA, USA, December 9-13, 2019*. ACM, 2019, pp. 189–201. [Online]. Available: https://doi.org/10.1145/3361525.3361546

[76] F. Mo, A. S. Shamsabadi, K. Katevas, S. Demetriou, I. Leontiadis, A. Cavallaro, and H. Haddadi, "DarkneTZ: towards model privacy at the edge using trusted execution environments," in *MobiSys '20: The 18th Annual International Conference on Mobile Systems, Applications, and Services, Toronto, Ontario, Canada, June 15-19, 2020*, E. de Lara, I. Mohomed, J. Nieh, and E. M. Belding, Eds. ACM, 2020, pp. 161–174. [Online]. Available: https://doi.org/10.1145/3386901.3388946

[77] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "PPFL: privacy-preserving federated learning with trusted execution environments," in *MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual Event, Wisconsin, USA, 24 June - 2 July, 2021*, S. Banerjee, L. Mottola, and X. Zhou, Eds. ACM, 2021, pp. 94–108. [Online]. Available: https://doi.org/10.1145/3458864.3466628

[78] AndroidOpen Source Project, "Trusty TEE," https://source.android.com/security/trusty, 2016, Latest release in 2020.

[79] S. M. Kim, J. Han, J. Ha, T. Kim, and D. Han, "Enhancing security and privacy of tor's ecosystem by using trusted execution environments," in *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*, A. Akella and J. Howell, Eds. USENIX Association, 2017, pp. 145–161. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/kim-seongmin

[80] Samsung, "TizenFX," 2018, latest release in 2022. [Online]. Available: https://github.com/Samsung/TizenFX

[81] F. Tramèr and D. Boneh, "Slalom: Fast, verifiable and private execution of neural networks in trusted hardware," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=rJVorjCcKQ

[82] D. L. Quoc, F. Gregor, S. Arnautov, R. Kunkel, P. Bhatotia, and C. Fetzer, "secureTF: A secure TensorFlow framework," in *Middleware '20: 21st International Middleware Conference, Delft, The Netherlands, December 7-11, 2020*, D. D. Silva and R. Kapitza, Eds. ACM, 2020, pp. 44–59. [Online]. Available: https://doi.org/10.1145/3423211.3425687

[83] L. Kang, Y. Xue, W. Jia, X. Wang, J. Kim, C. Youn, M. J. Kang, H. J. Lim, B. L. Jacob, and J. Huang, "Iceclave: A trusted execution environment for in-storage computing," in *MICRO '21: 54th Annual IEEE/ACM International Symposium on Microarchitecture, Virtual Event, Greece, October 18-22, 2021*. ACM, 2021, pp. 199–211. [Online]. Available: https://doi.org/10.1145/3466752.3480109

[84] V. Costan, I. A. Lebedev, and S. Devadas, "Sanctum: Minimal hardware extensions for strong software isolation," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 857–874. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/costan

[85] D. Goltzsche, C. Wulf, D. Muthukumaran, K. Rieck, P. R. Pietzuch, and R. Kapitza, "TrustJS: Trusted client-side execution of JavaScript," in *Proceedings of the 10th European Workshop on Systems Security, EUROSEC 2017, Belgrade, Serbia, April 23, 2017*, C. Giuffrida and A. Stavrou, Eds. ACM, 2017, pp. 7:1–7:6. [Online]. Available: https://doi.org/10.1145/3065913.3065917

[86] K. Krawiecka, A. Kurnikov, A. Paverd, M. Mannan, and N. Asokan, "Safekeeper: Protecting web passwords using trusted execution environments," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds. ACM, 2018, pp. 349–358. [Online]. Available: https://doi.org/10.1145/3178876.3186101

[87] K. Wüst, S. Matetic, M. Schneider, I. Miers, K. Kostiainen, and S. Capkun, "ZLiTE: Lightweight clients for shielded Zcash transactions using trusted execution," in *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*, ser. Lecture Notes in Computer Science, I. Goldberg and T. Moore, Eds., vol. 11598. Springer, 2019, pp. 179–198. [Online]. Available: https://doi.org/10.1007/978-3-030-32101-7_12

[88] P. Li, X. Luo, T. Miyazaki, and S. Guo, "Privacy-preserving payment channel networks using trusted execution environment," in *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*. IEEE, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICC40277.2020.9149447

[89] T. Xu, K. Zhu, A. Andrzejak, and L. Zhang, "Distributed learning in trusted execution environment: A case study of federated learning in SGX," in *7th IEEE International Conference on Network Intelligence and Digital Content, IC-NIDC 2021, Beijing, China, November 17-19, 2021*. IEEE, 2021, pp. 450–454. [Online]. Available: https://doi.org/10.1109/IC-NIDC54101.2021.9660433

[90] Fortanix, "Fortanix Rust Enclave Development Platform," 2016, latest release in 2022. [Online]. Available: https://github.com/fortanix/rust-sgx

[91] S. Chandra, V. Karande, Z. Lin, L. Khan, M. Kantarcioglu, and B. M. Thuraisingham, "Securing data analytics on SGX with randomization," in *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I*, ser. Lecture Notes in Computer Science, S. N. Foley, D. Gollmann, and E. Snekkenes, Eds., vol. 10492. Springer, 2017, pp. 352–369. [Online]. Available: https://doi.org/10.1007/978-3-319-66402-6_21

[92] S. Matetic, M. Schneider, A. Miller, A. Juels, and S. Capkun, "DelegaTEE: Brokered delegation using trusted execution environments," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, W. Enck and A. P. Felt, Eds. USENIX Association, 2018, pp. 1387–1403. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/matetic

[93] T. Kim, J. Park, J. Woo, S. Jeon, and J. Huh, "Shieldstore: Shielded in-memory key-value storage with SGX," in *Proceedings of the Fourteenth EuroSys Conference 2019, Dresden, Germany, March 25-28, 2019*, G. Candea, R. van Renesse, and C. Fetzer, Eds. ACM, 2019, pp. 14:1–14:15. [Online]. Available: https://doi.org/10.1145/3302424.3303951

[94] B. Fuhry, L. Hirschoff, S. Koesnadi, and F. Kerschbaum, "Segshare: Secure group file sharing in the cloud using enclaves," in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 - July 2, 2020*. IEEE, 2020, pp. 476–488. [Online]. Available: https://doi.org/10.1109/DSN48063.2020.00061

[95] J. Truong, W. Gallagher, T. Guo, and R. J. Walls, "Memory-efficient deep learning inference in trusted execution environments," in *IEEE International Conference on Cloud Engineering, IC2E 2021, San Francisco, CA, USA, October 4-8, 2021*. IEEE, 2021, pp. 161–167. [Online]. Available: https://doi.org/10.1109/IC2E52221.2021.00031

[96] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, M. Stillwell, D. Goltzsche, D. M. Eyers, R. Kapitza, P. R. Pietzuch, and C. Fetzer, "SCONE: secure linux containers with intel SGX," in *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, K. Keeton and T. Roscoe, Eds. USENIX Association, 2016, pp. 689–703. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/arnautov

[97] P. Subramanyan, R. Sinha, I. A. Lebedev, S. Devadas, and S. A.

Seshia, "A formal foundation for secure remote execution of enclaves," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 2435–2450. [Online]. Available: https://doi.org/10.1145/3133956.3134098

[98] S. Arnautov, A. Brito, P. Felber, C. Fetzer, F. Gregor, R. Krahn, W. Ozga, A. Martin, V. Schiavoni, F. Silva, M. Tenorio, and N. Thummel, "PubSub-SGX: Exploiting trusted execution environments for privacy-preserving publish/subscribe systems," in *37th IEEE Symposium on Reliable Distributed Systems, SRDS 2018, Salvador, Brazil, October 2-5, 2018*. IEEE Computer Society, 2018, pp. 123–132. [Online]. Available: https://doi.org/10.1109/SRDS.2018.00023

[99] G. Amjad, S. Kamara, and T. Moataz, "Forward and backward private searchable encryption with SGX," in *Proceedings of the 12th European Workshop on Systems Security, EuroSec@EuroSys 2019, Dresden, Germany, March 25, 2019*. ACM, 2019, pp. 4:1–4:6. [Online]. Available: https://doi.org/10.1145/3301417.3312496

[100] M. U. Sardar, D. L. Quoc, and C. Fetzer, "Towards formalization of enhanced privacy ID (epid)-based remote attestation in intel SGX," in *23rd Euromicro Conference on Digital System Design, DSD 2020, Kranj, Slovenia, August 26-28, 2020*. IEEE, 2020, pp. 604–607. [Online]. Available: https://doi.org/10.1109/DSD51259.2020.00099

[101] A. Mondal, Y. More, R. H. Rooparaghunath, and D. Gupta, "Poster: FLATEE: Federated learning across trusted execution environments," in *IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*. IEEE, 2021, pp. 707–709. [Online]. Available: https://doi.org/10.1109/EuroSP51992.2021.00054

[102] A. Shakevsky, E. Ronen, and A. Wool, "Trust dies in darkness: Shedding light on samsung's trustzone keymaster design," *IACR Cryptol. ePrint Arch.*, p. 208, 2022. [Online]. Available: https://eprint.iacr.org/2022/208

[103] B. Saltaformaggio, R. Bhatia, X. Zhang, D. Xu, and G. G. R. III, "Screen after previous screens: Spatial-temporal recreation of android app displays from memory images," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 1137–1151. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/saltaformaggio

[104] F. Shaon, M. Kantarcioglu, Z. Lin, and L. Khan, "Sgx-bigmatrix: A practical encrypted data analytic framework with trusted processors," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 1211–1228. [Online]. Available: https://doi.org/10.1145/3133956.3134095

[105] S. Volos, K. Vaswani, and R. Bruno, "Graviton: Trusted execution environments on gpus," in *13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018*, A. C. Arpaci-Dusseau and G. Voelker, Eds. USENIX Association, 2018, pp. 681–696. [Online]. Available: https://www.usenix.org/conference/osdi18/presentation/volos

[106] J. B. Djoko, J. Lange, and A. J. Lee, "NeXUS: Practical and secure access control on untrusted storage platforms using client-side SGX," in *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019*. IEEE, 2019, pp. 401–413. [Online]. Available: https://doi.org/10.1109/DSN.2019.00049

[107] S. Hu, Q. A. Chen, J. Joung, C. Carlak, Y. Feng, Z. M. Mao, and H. X. Liu, "Cvshield: Guarding sensor data in connected vehicle with trusted execution environment," in *AutoSec@CODASPY '20: Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security, New Orleans, LA, USA, March 18, 2020*, Q. A. Chen, Z. Zhao, and G. Ahn, Eds. ACM, 2020, pp. 1–4. [Online]. Available: https://doi.org/10.1145/3375706.3380552

[108] Y. Zhang, Z. Wang, J. Cao, R. Hou, and D. Meng, "ShuffleFL: gradient-preserving federated learning using trusted execution environment," in *CF '21: Computing Frontiers Conference, Virtual Event, Italy, May 11-13, 2021*, M. Palesi, A. Tumeo, G. I. Goumas, and C. G. Almudéver, Eds. ACM, 2021, pp. 161–168. [Online]. Available: https://doi.org/10.1145/3457388.3458665

[109] S. Zhao, M. Li, Y. Zhang, and Z. Lin, "vSGX: Virtualizing SGX enclaves on AMD SEV," 2022, will be published in the 2022 IEEE Symposium on Security and Privacy (SP). [Online]. Available: https://www.computer.org/csdl/proceedings-article/sp/2022/131600a687/1A4Q3q3W28E

[110] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-TEE - an open virtual trusted execution environment," in *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*. IEEE, 2015, pp. 400–407. [Online]. Available: https://doi.org/10.1109/Trustcom.2015.400

[111] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, "Oblivious multi-party machine learning on trusted processors," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 619–636. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/ohrimenko

[112] V. Karande, E. Bauman, Z. Lin, and L. Khan, "Sgx-log: Securing system logs with SGX," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, R. Karri, O. Sinanoglu, A. Sadeghi, and X. Yi, Eds. ACM, 2017, pp. 19–30. [Online]. Available: https://doi.org/10.1145/3052973.3053034

[113] R. Poddar, C. Lan, R. A. Popa, and S. Ratnasamy, "Safebricks: Shielding network functions in the cloud," in *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, S. Banerjee and S. Seshan, Eds. USENIX Association, 2018, pp. 201–216. [Online]. Available: https://www.usenix.org/conference/nsdi18/presentation/poddar

[114] A. Dhar, I. Puddu, K. Kostiainen, and S. Capkun, "ProximiTEE: Hardened SGX attestation by proximity verification," in *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, March 16-18, 2020*, V. Roussev, B. M. Thuraisingham, B. Carminati, and M. Kantarcioglu, Eds. ACM, 2020, pp. 5–16. [Online]. Available: https://doi.org/10.1145/3374664.3375726

[115] M. Gao, H. Dang, and E. Chang, "TEEKAP: self-expiring data capsule using trusted execution environment," in *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*. ACM, 2021, pp. 235–247. [Online]. Available: https://doi.org/10.1145/3485832.3485919

[116] Enarx, "MMLedger: A ledger for confidential computing shims for tracking memory management system calls," 2022. [Online]. Available: https://github.com/enarx/mmledger

[117] Google Git, "qseecom: Add qseecom Driver," 2013. [Online]. Available: https://android.googlesource.com/kernel/msm/+/d316c3dc0464e9703234bc1631700d832b2695bc

[118] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "VC3: trustworthy data analytics in the cloud using SGX," in *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*. IEEE Computer Society, 2015, pp. 38–54. [Online]. Available: https://doi.org/10.1109/SP.2015.10

[119] R. Pires, M. Pasin, P. Felber, and C. Fetzer, "Secure content-based routing using intel software guard extensions," in *Proceedings of the 17th International Middleware Conference, Trento, Italy, December 12 - 16, 2016*. ACM, 2016, p. 10. [Online]. Available: http://dl.acm.org/citation.cfm?id=2988346

[120] B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov, "IRON: functional encryption using intel SGX," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 765–782. [Online]. Available: https://doi.org/10.1145/3133956.3134106

[121] R. Pires, D. Goltzsche, S. B. Mokhtar, S. Bouchenak, A. Boutet, P. Felber, R. Kapitza, M. Pasin, and V. Schiavoni, "CYCLOSA: decentralizing private web search through sgx-based browser extensions," in *38th IEEE International Conference on Distributed Computing Systems, ICDCS 2018, Vienna, Austria, July 2-6, 2018*. IEEE Computer Society, 2018, pp. 467–477. [Online]. Available: https://doi.org/10.1109/ICDCS.2018.00053

[122] S. Zhao, Q. Zhang, Y. Qin, W. Feng, and D. Feng, "SecTEE: A software-based approach to secure enclave architecture using TEE," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK,*

November 11-15, 2019, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 1723–1740. [Online]. Available: https://doi.org/10.1145/3319535.3363205

[123] M. da Rocha, D. C. G. Valadares, A. Perkusich, K. C. Gorgônio, R. T. Pagno, and N. C. Will, "Secure cloud storage with client-side encryption using a trusted execution environment," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science, CLOSER 2020, Prague, Czech Republic, May 7-9, 2020*, D. Ferguson, M. Helfert, and C. Pahl, Eds. SCITEPRESS, 2020, pp. 31–43. [Online]. Available: https://doi.org/10.5220/0009130600310043

[124] A. Georgios, "Atlas: Automated scale-out of trust-oblivious systems to trusted execution environments," Master's thesis, University of Crete, 2021. [Online]. Available: https://elocus.lib.uoc.gr/dlib/e/6/1/metadata-dlib-1637579552-223704-1365.tkl

[125] G. Patat, M. Sabt, and P. Fouque, "Exploring widevine for fun and profit," *CoRR*, vol. abs/2204.09298, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2204.09298

[126] C. Namiluko, A. J. Paverd, and T. de Souza, "Towards enhancing web application security using trusted execution," in *Proceedings of the Workshop on Web Applications and Secure Hardware (WASH'13), Co-located with the 6th International Conference on Trust and Trustworthy Computing (TRUST 2013), London, United Kingdom, June 20, 2013*, ser. CEUR Workshop Proceedings, J. Lyle, S. Faily, and M. Winandy, Eds., vol. 1011. CEUR-WS.org, 2013. [Online]. Available: http://ceur-ws.org/Vol-1011/4.pdf

[127] J. S. Jang, S. Kong, M. Kim, D. Kim, and B. B. Kang, "SeCReT: Secure channel between rich execution environment and trusted execution environment," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*. The Internet Society, 2015. [Online]. Available: https://www.ndss-symposium.org/ndss2015/secret-secure-channel-between-rich-execution-environment-and-trusted-execution-environment

[128] E. Bauman and Z. Lin, "A case for protecting computer games with SGX," in *Proceedings of the 1st Workshop on System Software for Trusted Execution, SysTEX@Middleware 2016, Trento, Italy, December 12, 2016*. ACM, 2016, pp. 4:1–4:6. [Online]. Available: https://doi.org/10.1145/3007788.3007792

[129] C. Shepherd, R. N. Akram, and K. Markantonakis, "Establishing mutually trusted channels for remote sensing devices with trusted execution environments," in *Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017*. ACM, 2017, pp. 7:1–7:10. [Online]. Available: https://doi.org/10.1145/3098954.3098971

[130] P. Aublin, F. Kelbert, D. O'Keeffe, D. Muthukumaran, C. Priebe, J. Lind, R. Krahn, C. Fetzer, D. M. Eyers, and P. R. Pietzuch, "LibSEAL: revealing service integrity violations using trusted execution," in *Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, April 23-26, 2018*, R. Oliveira, P. Felber, and Y. C. Hu, Eds. ACM, 2018, pp. 24:1–24:15. [Online]. Available: https://doi.org/10.1145/3190508.3190547

[131] H. Duan, C. Wang, X. Yuan, Y. Zhou, Q. Wang, and K. Ren, "Lightbox: Full-stack protected stateful middlebox at lightning speed," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 2351–2367. [Online]. Available: https://doi.org/10.1145/3319535.3339814

[132] S. Park, A. Ahmad, and B. Lee, "Blackmirror: Preventing wallhacks in 3d online FPS games," in *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, J. Ligatti, X. Ou, J. Katz, and G. Vigna, Eds. ACM, 2020, pp. 987–1000. [Online]. Available: https://doi.org/10.1145/3372297.3417890

[133] L. Chen, R. Yuan, and Y. Xia, "Tora: A trusted blockchain oracle based on a decentralized tee network," in *2021 IEEE International Conference on Joint Cloud Computing (JCC)*, 2021, pp. 28–33.

[134] F. Kato, Y. Cao, and M. Yoshikawa, "OLIVE: oblivious and differentially private federated learning on trusted execution environment," *CoRR*, vol. abs/2202.07165, 2022. [Online]. Available: https://arxiv.org/abs/2202.07165

[135] R. van Rijswijk-Deij and E. Poll, "Using trusted execution environments in two-factor authentication: comparing approaches," in *Open Identity Summit 2013, September 9th - 11th 2013, Kloster Banz, Germany*, ser. LNI, D. Hühnlein and H. Roßnagel, Eds., vol. P-223. GI, 2013, pp. 20–31. [Online]. Available: https://dl.gi.de/20.500.12116/17195

[136] TrustedFirmware.org, "OP-TEE Documentation," 2014, Latest release in 2022. [Online]. Available: https://optee.readthedocs.io/en/latest/

[137] W. Li, H. Li, H. Chen, and Y. Xia, "Adattester: Secure online mobile advertisement attestation using trustzone," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2015, Florence, Italy, May 19-22, 2015*, G. Borriello, G. Pau, M. Gruteser, and J. I. Hong, Eds. ACM, 2015, pp. 75–88. [Online]. Available: https://doi.org/10.1145/2742647.2742676

[138] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 270–282. [Online]. Available: https://doi.org/10.1145/2976749.2978326

[139] Signal, "Private Contact Discovery Service," 2017, latest release in 2022. [Online]. Available: https://github.com/signalapp/ContactDiscoveryService

[140] M. Tran, L. Luu, M. S. Kang, I. Bentov, and P. Saxena, "Obscuro: A bitcoin mixer using trusted execution environments," in *Proceedings of the 34th Annual Computer Security Applications Conference, ACSAC 2018, San Juan, PR, USA, December 03-07, 2018*. ACM, 2018, pp. 692–701. [Online]. Available: https://doi.org/10.1145/3274694.3274750

[141] G. Chen, Y. Zhang, and T. Lai, "OPERA: open remote attestation for intel's secure enclaves," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 2317–2331. [Online]. Available: https://doi.org/10.1145/3319535.3354220

[142] K. Suzaki, A. Tsukamoto, A. Green, and M. Mannan, "Reboot-oriented IoT: Life cycle management in trusted execution environment for disposable IoT devices," in *ACSAC '20: Annual Computer Security Applications Conference, Virtual Event / Austin, TX, USA, 7-11 December, 2020*. ACM, 2020, pp. 428–441. [Online]. Available: https://doi.org/10.1145/3427228.3427293

[143] C. Cai, L. Xu, A. Zhou, and C. Wang, "Toward a secure, rich, and fair query service for light clients on public blockchains," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[144] ApplePasskeys, "Supporting Passkeys," https://developer.apple.com/documentation/authenticationservices/public-private_key_authentication/supporting_passkeys, 2022.

[145] J. Ménétrey, C. Göttel, M. Pasin, P. Felber, and V. Schiavoni, "An exploratory study of attestation mechanisms for trusted execution environments," *CoRR*, vol. abs/2204.06790, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2204.06790

[146] V. Schiavoni, "sgx-papers," 2022. [Online]. Available: https://github.com/vschiavoni/sgx-papers

[147] E. Novella, "A curated list of public TEE resources for learning how to reverse-engineer and achieve trusted code execution on ARM devices," 2022. [Online]. Available: https://github.com/enovella/TEE-reversing

[148] S. Fei, Z. Yan, W. Ding, and H. Xie, "Security vulnerabilities of SGX and countermeasures: A survey," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 126:1–126:36, 2021. [Online]. Available: https://doi.org/10.1145/3456631

[149] D. Cerdeira, N. Santos, P. Fonseca, and S. Pinto, "Sok: Understanding the prevailing security vulnerabilities in trustzone-assisted TEE systems," in *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*. IEEE, 2020, pp. 1416–1432. [Online]. Available: https://doi.org/10.1109/SP40000.2020.00061

[150] N. Koutroumpouchos, C. Ntantogian, and C. Xenakis, "Building trust for smart connected devices: The challenges and pitfalls of trustzone," *Sensors*, vol. 21, no. 2, p. 520, 2021. [Online]. Available: https://doi.org/10.3390/s21020520

[151] N. Asokan, "Hardware-assisted trusted execution environments: Look back, look ahead," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1687–1687. [Online]. Available: https://doi-org.libproxy.tuni.fi/10.1145/3319535.3364969

[152] J.-E. Ekberg, K. Kostiainen, and N. Asokan, "The untapped potential

of trusted execution environments on mobile devices," *IEEE Security Privacy*, vol. 12, no. 4, pp. 29–37, 2014.
[153] Joel Snyder, "Using biometrics for authentication in Android," Apr 21, 2021. [Online]. Available: https://insights.samsung.com/2021/04/21/using-biometrics-for-authentication-in-android-2
[154] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Inf. Sci.*, vol. 522, pp. 69–79, 2020. [Online]. Available: https://doi.org/10.1016/j.ins.2020.02.037
[155] A. M. Azab, P. Ning, J. Shah, Q. Chen, R. Bhutkar, G. Ganesh, J. Ma, and W. 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, Scottsdale, AZ, USA, November 3-7, 2014*, G. Ahn, M. Yung, and N. Li, Eds. ACM, 2014, pp. 90–102. [Online]. Available: https://doi.org/10.1145/2660267.2660350
[156] GlobalPlatform Device Technology, "TEE Client API Specification Version 1.0," GlobalPlatform, Tech. Rep., 2010. [Online]. Available: https://globalplatform.org/wp-content/uploads/2010/07/TEE_Client_API_Specification-V1.0.pdf
[157] J. Ménétrey, "Twine: An Embedded Trusted Runtime for WebAssembly," 2022. [Online]. Available: https://github.com/JamesMenetrey/unine-twine
[158] F. V. Mo, "DarkneTZ: Towards Model Privacy at the Edge using Trusted Execution Environments," 2020. [Online]. Available: https://github.com/mofanv/darknetz
[159] T. Kim, "ShieldStore," 2020. [Online]. Available: https://github.com/cocoppang/ShieldStore
[160] sengsgx, "SENG, the SGX-Enforcing Network Gateway," 2020. [Online]. Available: https://github.com/sengsgx/sengsgx
[161] G. Chen, "MAGE: Mutual Attestation for a Group of Enclaves without Trusted Third Parties," 2019, latest release in 2021. [Online]. Available: https://github.com/donnod/linux-sgx-mage
[162] Riscure, "OP-TEE Fuzzer," 2019, latest release in 2021. [Online]. Available: https://github.com/Riscure/optee_fuzzer
[163] CyFI-Lab-Public, "RetroScope: Android memory forensics framework," 2016. [Online]. Available: https://github.com/CyFI-Lab-Public/RetroScope
[164] F. V. Mo, "Privacy-preserving Federated Learning with Trusted Execution Environments," 2021. [Online]. Available: https://github.com/mofanv/PPFL
[165] Occlum team, "Occlum," 2022. [Online]. Available: https://github.com/occlum/occlum
[166] OP-TEE, "OP-TEE Client API," 2015, latest release in 2022. [Online]. Available: https://github.com/OP-TEE/optee_client
[167] Trustonic, "Trustonic TEE User Space," 2015. [Online]. Available: https://github.com/Trustonic/trustonic-tee-user-space/
[168] R. Pires, "Secure content-based routing (SCBR)," 2019. [Online]. Available: https://github.com/rafaelppires/scbr
[169] F. Zhang, "Town Crier: An Authenticated Data Feed For Smart Contracts," 2021. [Online]. Available: https://github.com/bl4ck5un/Town-Crier
[170] kaist-ina, "SGX-Tor," 2019. [Online]. Available: https://github.com/kaist-ina/SGX-Tor
[171] Microsoft, "Project Komodo," 2017. [Online]. Available: https://github.com/Microsoft/Komodo
[172] S. Chandra, "Securing Data Analytics on SGX with Randomization," 2017. [Online]. Available: https://github.com/swarupchandra/secure-analytics-sgx
[173] utds3lab, "SGX-Log: Securing System Logs With SGX," 2017. [Online]. Available: https://github.com/utds3lab/sgx-log
[174] Large-Scale Data & Systems (LSDS) Group, "LibSEAL," 2021. [Online]. Available: https://github.com/lsds/LibSEAL
[175] BitObscuro, "Obscuro," 2020. [Online]. Available: https://github.com/BitObscuro/Obscuro
[176] SELIS Project, "The SELIS Publish/Subscribe system," 2019. [Online]. Available: https://github.com/selisproject/pubsub
[177] Y. Zhou, "SafeBricks," 2020. [Online]. Available: https://github.com/YangZhou1997/SafeBricks
[178] SafeKeeper, "SafeKeeper - Protecting Web passwords using Trusted Execution Environments," 2018. [Online]. Available: https://github.com/SafeKeeper/safekeeper-server
[179] P. Wang, "LightBox," 2019. [Online]. Available: https://github.com/patrickwang96/LightBox
[180] B. Djoko, "Secure cloud access/usage control using client-side SGX," 2020. [Online]. Available: https://github.com/sporgj/nexus-code
[181] F. Tramer, "SLALOM," 2021. [Online]. Available: https://github.com/ftramer/slalom
[182] J. Lind, I. Eyal, P. R. Pietzuch, and E. G. Sirer, "Teechan: Payment channels using trusted execution environments," *CoRR*, vol. abs/1612.07766, 2016. [Online]. Available: http://arxiv.org/abs/1612.07766
[183] Large-Scale Data & Systems (LSDS) Group, "Teechain: A Secure Payment Network with Asynchronous Blockchain Access," 2019. [Online]. Available: https://github.com/lsds/Teechain
[184] A. Elbüz, "Blockchain Based Data Trading Platform," 2022. [Online]. Available: https://github.com/aelbuz/BlockchainBasedDataTradingPlatform
[185] C. Ndolo, S. A. Henningsen, and M. Florian, "Crawling the mobilecoin quorum system," *CoRR*, vol. abs/2111.12364, 2021. [Online]. Available: https://arxiv.org/abs/2111.12364
[186] C. Müller, "Hyperledger Fabric chaincode execution with OP-TEE," 2019. [Online]. Available: https://github.com/piachristel/open-source-fabric-optee-chaincode
[187] Atlas Runtime, "Atlas: Automated Scale-out of Trust-Oblivious Systems to Trusted Execution Environments," 2022. [Online]. Available: https://github.com/atlas-runtime/applications/
[188] K. Dokmai, "SMac: Secure Genotype Imputation in Intel SGX," 2022. [Online]. Available: https://github.com/ndokmai/sgx-genotype-imputation
[189] M. Miranda, "S2Dedup," 2021. [Online]. Available: https://github.com/mmm97/S2Dedup
[190] Jseam, "Tora-Zilliqa," 2021. [Online]. Available: https://issueantenna.com/repo/JSeam2/Tora-Zilliqa#
[191] S. K. Jeon, "TZMon: Improving mobile game security with ARM trustzone," 2018. [Online]. Available: https://github.com/kppw99/TZMon
[192] Keystone Enclave, "Keystone: An Open-Source Secure Enclave Framework for RISC-V Processors," 2018, Latest release in 2022. [Online]. Available: https://github.com/keystone-enclave/keystone
[193] D. Dong, "Build TA images on different TEE," https://dqdongg.com/android/fingerprint/2021/02/03/Fingerprint-build-ta.html#2, 2021.
[194] F. Khalid and A. Masood, "Vulnerability analysis of qualcomm secure execution environment (QSEE)," *Comput. Secur.*, vol. 116, p. 102628, 2022. [Online]. Available: https://doi.org/10.1016/j.cose.2022.102628
[195] V. Costan, "Sanctum," 2015, Latest release in 2019. [Online]. Available: https://github.com/pwnall/sanctum
[196] I. Lebedev, "The MIT Sanctum processor system," 2019, Latest release in 2020. [Online]. Available: https://github.com/ilebedev/sanctum
[197] Trustonic, "Secure IoT Development with Kinibi-M," https://www.trustonic.com/technical-articles/kinibi-m/, 2018, Latest release in 2020.
[198] webinos, "Secure Web Operating System Application Delivery Environment." 2013. [Online]. Available: https://github.com/webinos/Webinos-Platform
[199] Distributed Systems group at IBR, TU Braunschweig, "AccTEE: A WebAssembly-based Two-way Sandbox for Trusted Resource Accounting," 2020. [Online]. Available: https://github.com/ibr-ds/AccTEE
[200] W. Liu, "Deflection (CAT-SGX)," 2021. [Online]. Available: https://github.com/StanPlatinum/Deflection
[201] USB armory, "GoTEE - example application," 2022. [Online]. Available: https://github.com/usbarmory/GoTEE-example
[202] The Gramine Project, "Gramine Library OS with Intel SGX Support," 2022. [Online]. Available: https://github.com/gramineproject/gramine
[203] MesaLock Linux, "MesaPy: A Memory-Safe Python Implementation based on PyPy," 2019. [Online]. Available: https://github.com/mesalock-linux/mesapy
[204] ratel-enclave, "Ratel - a new framework for instruction-level interposition on enclaved applications," 2022. [Online]. Available: https://github.com/ratel-enclave/ratel
[205] Operating Systems and Architecture, "Ryoan: A distributed sandbox for untrusted computation on secret data," 2022. [Online]. Available: https://github.com/ut-osa/ryoan
[206] Scontain, "SCONE Confidential Computing," 2022. [Online]. Available: https://sconedocs.github.io/
[207] Large-Scale Data & Systems (LSDS) Group, "SGX-LKL-OE (Open Enclave Edition)," 2022. [Online]. Available: https://github.com/lsds/sgx-lkl

[208] D. E. Porter, S. Boyd-Wickizer, J. Howell, R. Olinsky, and G. C. Hunt, "Rethinking the library OS from the top down," in *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2011, Newport Beach, CA, USA, March 5-11, 2011*, R. Gupta and T. C. Mowry, Eds. ACM, 2011, pp. 291–304. [Online]. Available: https://doi.org/10.1145/1950365.1950399

[209] J. Ekberg, N. Asokan, K. Kostiainen, and A. Rantala, "Scheduling execution of credentials in constrained secure environments," in *Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing, STC 2008, Alexandria, VA, USA, October 31, 2008*, S. Xu, C. Nita-Rotaru, and J. Seifert, Eds. ACM, 2008, pp. 61–70. [Online]. Available: https://doi.org/10.1145/1456455.1456465