Tomás
Carvalho Bogalho

Plataforma de computação confidencial com SGX

Confidential computing platform with SGX

DOCUMENTO PROVISÓRIO

Tomás
Carvalho Bogalho

Plataforma de computação confidencial com SGX

Confidential computing platform with SGX

o júri / the jury

presidente / president

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

Prof. Doutor João Antunes da Silva
professor associado da Universidade de Aveiro

**agradecimentos /
acknowledgements**

palavras-chave

texto livro, arquitetura, história, construção, materiais de construção, saber tradicional.

resumo

Um resumo é um pequeno apanhado de um trabalho mais longo (como uma tese, dissertação ou trabalho de pesquisa). O resumo relata de forma concisa os objetivos e resultados da sua pesquisa, para que os leitores saibam exatamente o que se aborda no seu documento.

Embora a estrutura possa variar um pouco dependendo da sua área de estudo, o seu resumo deve descrever o propósito do seu trabalho, os métodos que você usou e as conclusões a que chegou.

Uma maneira comum de estruturar um resumo é usar a estrutura IMRaD. Isso significa:

- Introdução

- Métodos

- Resultados

- Discussão

Veja mais pormenores aqui:
`https://www.scribbr.com/dissertation/abstract/`

keywords

textbook, architecture, history, construction, construction materials, traditional knowledge.

abstract

An abstract is a short summary of a longer work (such as a thesis, dissertation or research paper).

The abstract concisely reports the aims and outcomes of your research, so that readers know exactly what your paper is about.

Although the structure may vary slightly depending on your discipline, your abstract should describe the purpose of your work, the methods you've used, and the conclusions you've drawn.

One common way to structure your abstract is to use the IMRaD structure. This stands for:

- Introduction

- Methods

- Results

- Discussion

Check for more details here:

https://www.scribbr.com/dissertation/abstract/

acknowledgement of use of AI tools

Recognition of the use of generative Artificial Intelligence technologies and tools, software and other support tools.

I acknowledge the use of [insert AI system(s) and link] to [specific use of generative artificial intelligence or other tasks]. I acknowledge the use of [software, codes or platforms] to [specific use software, codes or platforms or to other tasks].

Example 1: I acknowledge the use of ChatGPT 3.5 (Open AI, https://chat.openai.com) to summarise the initial notes and to proofread the final draft and the use of Office365 (Microsoft, https://www.office.com) for text writing and productivity.

Example 2: No content generated by AI technologies has been used in this Thesis.

Contents

List of Figures

List of Tables

Glossary

RD	Research and Development	**ASP**	AMD Secure Processor
ML	Machine Learning	**SMI**	System Management Interrupt
VM	Virtual Machine	**SMM**	System Management Mode
SQL	Structured Query Language	**VMPL**	Virtual Machine Protection Level
POC	Proof of Concept	**SVSM**	Secure Virtual Machine Service Module
SDK	Software Development Kit	**OS**	Operating System
UU	Universal User	**MAC**	Media Access Control
AP	Access Point	**DRAM**	Dynamic Random-Access Memory
RSSI	Received Signal Strength Indicator	**ROM**	Read-Only Memory
CSI	Channel State Information	**SRAM**	Static Random-Access Memory
IEEE	Institute of Electrical and Electronics Engineers	**MMU**	Memory Management Unit
PWR	Passive WiFi Radar	**CPU**	Central Processing Unit
TEE	Trusted Execution Environments	**I/O**	Input/Output
TEE	Trusted Execution Environment	**DMA**	Direct Memory Access
TCB	Trusted Computing Base	**SMM**	System Management Mode
VC3	Verifiable Confidential Cloud Computing	**EPC**	Enclave Page Cache
TTP	Trusted Third Party	**EPCM**	Enclave Page Cache Metadata
TC	Trusted Computing	**SGX**	Software Guard eXtensions
TPM	Trusted Platform Module	**PRM**	Processor Reserved Memory
TDC	Trust Domain Extensions	**SDM**	Software Development Manual
CVM	Confidential Virtual Machine	**SECS**	SGX Enclave Control Structure
TDX	Trust Domain Extensions	**ELRANGE**	Enclave Linear Address Range
SEAM	Secure-Arbitration Mode	**XFRM**	Extended Features Request Mask
VT	Virtualization Technology	**EDL**	Enclave Definition Languagem
MKTME	Multi-key Total Memory Encryption	**SoC**	System-on-Chip
DCAP	Data Center Attestation Primitives	**NS**	Non Secure
TD	Trusted Domain	**SCR**	Secure Configuration Register
VMX	Virtual Machine Extensions	**SMC**	Secure Monitor Call
SEV	Secure Encrypted Virtualization	**TZASC**	TrustZone Address Space Controller
SEV-ES	Secure Encrypted Virtualization Encrypted State	**TZMA**	TrustZone Memory Adapter
SEV-SNP	Secure Encrypted Virtualization Secure Nested Paging	**TZPC**	TrustZone Protection Controller
		REE	Rich Execution Environment

Introduction

The presence of wireless networks in every day infrastructures has evolved beyond simple connectivity, due to its large presence and usage, it has become a reliable source of data. In campus environments, the metadata [1] generated by user communications to the WiFi network, like the eduroam network, allows the somewhat precise localization of individuals, from here several indicators can be extracted such as room occupancy rates, the verification of class attendance, and the understanding of movement flows within the campus.

However, the extraction of statistical information from this raw data creates a significant difficulty between utility and privacy, namely the processing of raw connection logs exposes sensitive information to system administrators and potential attackers. Usual security models protect the data at rest and during transit with encryption and hashing, but often leave it vulnerable during the actual computation, where data needs to be in cleartext, exposed in memory and to the Operating System (OS), in order to be processed.

To resolve this, a paradigm known as confidential computing occurs, unlike commom privacy-preserving techniques that rely on obfuscation or anonymization, confidential computing uses hardware-based TEEs to protect data, this ensures that only code which is authorized and authenticated can process data without exposing the raw information to the underlying infrastructure or privileged users.

1.1 MOTIVATIONS

Maybe o que está no documento de descrição da tese

1.2 OBJECTIVES

The main goal of this dissertation is to deliver a Proof of Concept (POC) of a Intel SGX implementation that executes authorized applications to produce aggregated data from

---

[1]Refers to data that describes the content, formats, or attributes of a data record or information resource, description by de Keyser [9]

raw WiFi logs. This process aims to generate statistical data for the university without compromising the anonymity of WiFi users or revealing sensitive information.

However, extracting and processing these indicators in a trusted environment presents technical challenges, like the limits of the secure memory available to an enclave and the lack of native operating system swapping. The execution environment should then develop ways to provide essential primitives for various programs, such as memory usage monitors, the secure ingestion of encrypted data from external sources, and the production of results for various destinations, these mechanisms should include an equivalent to a swap area to allow the processing of large datasets despite the internal enclave memory, or a snapshot capability to enable discontinuous processing.

In order to explore a solution to this problem, this work focused on the below steps:

1. Elaboration of the state-of-the-art regarding WiFi-based indoor positioning, the application of SGX in this domain, and techniques for large-scale data processing.

2. Definition of a generic architecture capable of processing large volumes data, supporting both the simultaneous processing and over extended periods.

3. Development and documentation of a POC application to demonstrate the proposed architecture.

4. Experimental validation of the system using raw WiFi logs to extract useful aggregated statistical indicators, verifying the accuracy of the results.

1.3 OUTLINE

To be done

CHAPTER 2

Context

The present chapter provides a comprehensive understanding of the technical underpinnings of SGX and the protection of sensitive data. It begins with the introduction of the concept of sensitive data, followed by TEEs, giving an overview of this type of mechanism. Then the chapter compares technologies, analyzing the distinctions between several TEEs implementations. After this, the discussion focuses on Intel SGX, detailing its architecture, workflow, and security mechanisms, followed by the technical challenges of data processing in an enclave and concluding with potential attacks on SGX.

2.1 SENSITIVE DATA

Driven by technological advances and institutional needs, nearly all buildings are now equipped with internet access provided through Access Point (AP). These allow for the determination of user location with a reasonable degree of precision, due to the need of the user to connect to a specific AP in order to use the network.

Although often categorized as technical metadata, it presents privacy challenges when analyzed by definition of sensitive data, while raw location logs may not possess intrinsic sensitiveness, like a direct medical diagnosis might, its possible to acquire sensitive information through their processing context. As Quinn [10] argue, the critical factor is the computational distance required to infer sensitive information from apparently neutral data.

In the case of indoor positioning, this computational distance is low, the aggregation of location traces creates a high risk of linkability, where disparate datasets, such as anonymous WiFi logs and public class schedules, can be easily cross referenced. This reduces the effort necessary to de-anonymize users, transforming technical data into identifiable information.

Example of some sensitive information used for indoor location mentioned throughout this document:

- **Media Access Control (MAC) Address:** A persistent hardware identifier assigned to the network interface controller, this serves as a static fingerprint that uniquely identifies a user's device across different sessions and locations.
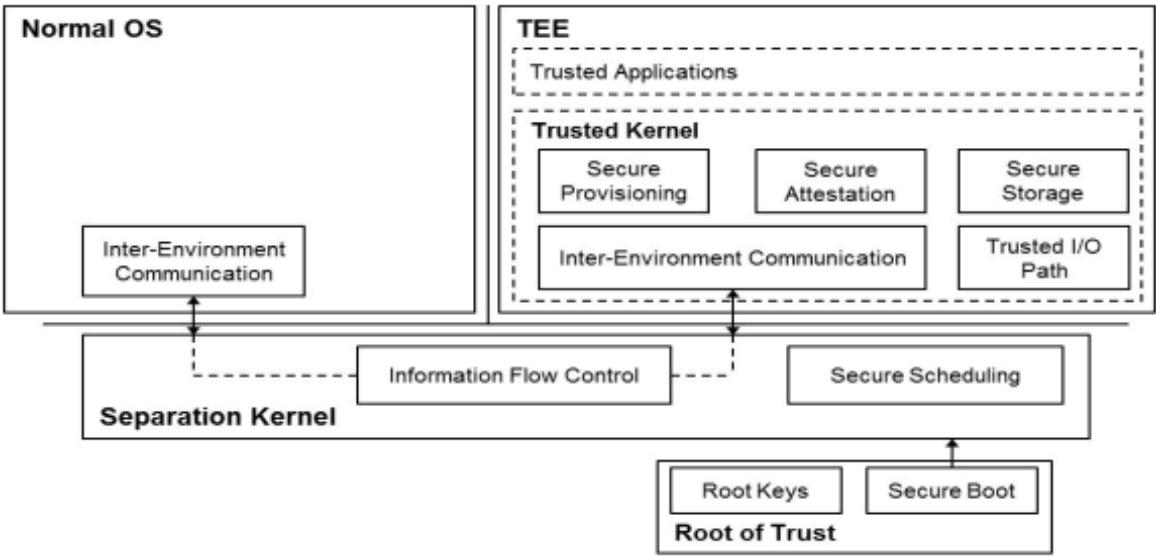
- **Universal User (UU):** The unique user identifier (such as a student ID, employee username, or email) required for network authentication.
- **Connection Logs:** Time records containing precise association and disassociation timestamps, these allow for the reconstruction of a user's daily routine, duration of stay, and punctuality patterns.
- **Spatial Hierarchy:** Topological data classifying location from macro to micro levels, like the location of the AP the user connected to.
- **Visual and Biometric Data:** In systems utilizing optical sensors, raw data includes high-resolution images or video capable of revealing biometric features and behavioral signatures.

2.2 Trusted Execution Environments (TEEs)

Before defining TEEs, it is essential to understand the underlying concept of Trusted Computing (TC), as defined by Sabt *et al.* [1], a TC aims to enable systems to achieve secure computation, privacy, and data protection. Historically, TC relied on a dedicated hardware component to provide a functional interface for security operations, one common example is the Trusted Platform Module (TPM), which allows a system to prove its integrity and securely store cryptographic keys, however, a limitation of the TPM is its inability to perform isolated execution of arbitrary code, since it is mainly designed for passive storage and reporting, rather than active processing. But more recent approaches to TC, shift the execution of arbitrary code to a trusted environment that allows tamper resistant execution of its applications.

While various terminologies exist in the literature, this dissertation adopts the term TEE, this is defined by Sabt *et al.* [1] as a secure, integrity protected computational environment, typically with isolated memory and specific storage capabilities. Its primary objectives are to guarantee the authenticity of the executed code, the integrity of the runtime state and the confidentiality of the data, like Central Processing Unit (CPU) registers, while effectively protecting code, memory, and sensitive Input/Output (I/O) from unauthorized access, in addition a TEE should have the ability to support remote attestation, allowing third parties to verify the trustworthiness of the environment, although specific implementation details may vary across different types of TEEs.

The implementation of these can differ, leading to a categorization of TEEs based on their enforcement mechanism, hardware-based TEEs rely on specific hardware extensions, such as Intel SGX, to enforce isolation at the lower level, effectively anchoring trust in the physical processor, in contrast, software-based TEEs depend on a privileged software layer typically a hypervisor or microkernel to protect the execution environment. Sabt *et al.* [1] proposed some definitions represented in the following figure.
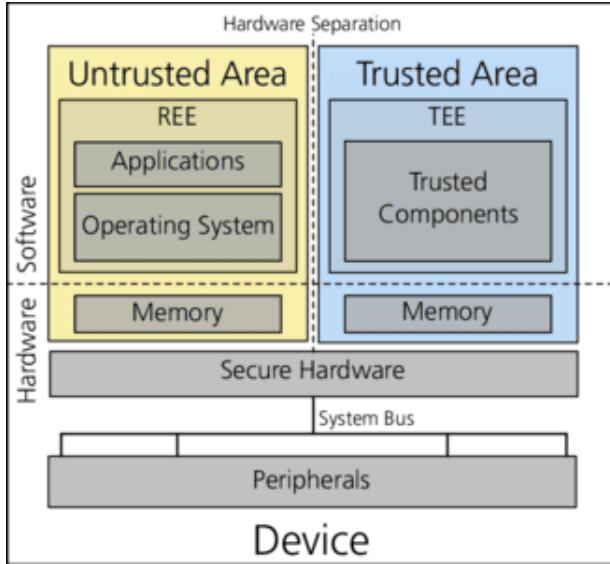
Figure 2.1: Overview of TEE building blocks, image by Sabt *et al.* [1].

As seen in the previous picture, a TEE in composed by building blocks such as:

- Secure Boot: guarantees that only code of a certain property can be loaded during boot.
- Secure Scheduling: provides a balanced coordination between the TEE and the rest of the system.
- Inter-Environment Communication: Interface that allows the TEE to communicate with the rest of the system.
- Secure Storage: Storage where confidentiality, integrity and authorized access of stored data is guaranteed.
- Trusted I/O Path: Communication between I/O peripherals with authenticity and sometimes confidentiality.

In contrast to the isolation of the TEE, the remaining portion of the system is often referred to as the REE or the untrusted environment, this includes the traditional operating system, the hypervisor, and the standard application stack (yellow component of the Figure2.2). While the REE is optimized for performance, versatility, and accessing peripherals, it looses the security guarantees required for sensitive data processing, as it makes it vulnerable software attacks. The architectural principle of trusted computing is the strict separation between these two worlds, the TEE protects its assets even when the surrounding REE is fully compromised. This separation can be observed in the Figure 2.2.
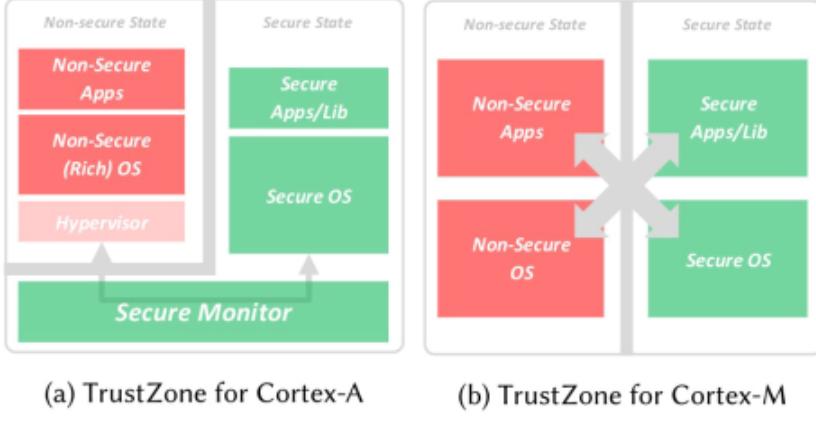
Figure 2.2: Overview of REE and of TEE, image by González [2].

2.3   TEE TECHNOLOGIES

Several TEE implementations exist, such as ARM TrustZone, AMD SEV, and Intel most recent TDX, this chapter provides a comparative overview of these technologies, however, Intel SGX, the focus of this dissertation, is reviewed in greater detail in the subsequent chapter.

2.3.1   ARM TrustZone

Initially introduced in 2004, ARM TrustZone is a set of hardware security extensions available on ARM processors, while first implemented on the Cortex-A series, it was later adapted for the more recent Cortex-M series, with some architectural differences suited for microcontrollers. Contrary to a standalone component, TrustZone is an approach designed to be integrated into a System-on-Chip (SoC), securing not just the processor but the entire electronic device, as described by Pinto and Santos [3], in this model the architecture is centered around the concept of distinct domains, the Secure World, the trusted environment, and the Normal World, the untrusted environment, as seen in Figure 2.3.
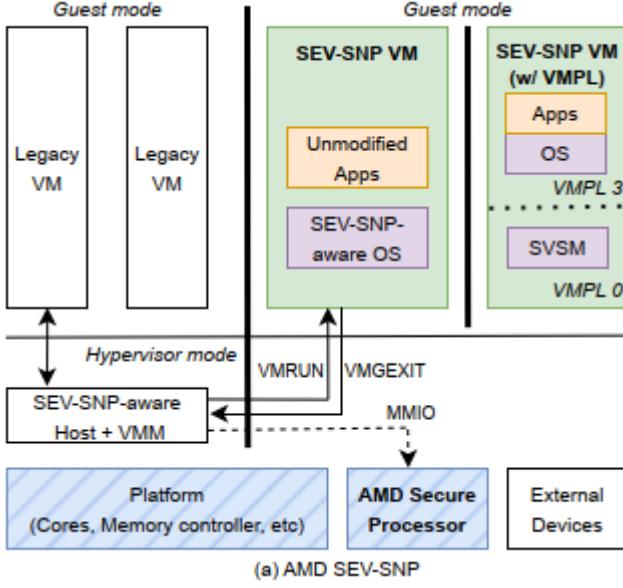
Figure 2.3: ARM TrustZone technology, image by Pinto and Santos [3].

This isolation allows software to execute with different privilege levels, preventing untrusted applications in the Normal World from directly accessing resources allocated to the Secure World. The state of the processor is determined by the Non Secure (NS) bit, typically located in the Secure Configuration Register (SCR), which effectively partitions the system's hardware resources.

To manage the transition between worlds, TrustZone employs a monitor mode, which executes in a secure state independent of the NS bit, this mode bridges the two software stacks and is entered via the privileged Secure Monitor Call (SMC) instruction or through specifically configured hardware exceptions.

Hardware isolation is reinforced by banked registers, which guarantees that security critical system registers are either hidden from the untrusted environment or are strictly accessed from the trusted environment. Regarding memory infrastructure, TrustZone introduces components such as the TrustZone Address Space Controller (TZASC) for Dynamic Random-Access Memory (DRAM) and the TrustZone Memory Adapter (TZMA) for off-chip Read-Only Memory (ROM)/Static Random-Access Memory (SRAM), these controllers partition memory into trusted and untrusted regions, preventing the Normal World from accessing secure memory while allowing the Secure World full access. TrustZone Memory Management Unit (MMU) maintains separate virtual to physical translation tables for each domain, and cache isolation is enforced via an additional tag bit on cache lines. Finally, the TrustZone Protection Controller (TZPC) extends this isolation to system peripherals, restricting access to specific devices based on the security domain.

In recent years, academic and industrial interest in this technology has surged, driven by the availability of technical documentation and accessible development platforms from manufacturers like Xilinx, this transparency has facilitated adoption within the community, particularly in the context of Research and Development (RD).

Figure 2.4: AMD SEV Architecture, image by Misono *et al.* [4].

2.3.2 AMD SEV

Introduced in 2016, SEV is stated to be the first x86 technology that was design to isolate Virtual Machine (VM)s from the hypervisor, since historically these have been considered trusted components, this allows for example, clients in the cloud to secure their VM workload from the cloud administrator, keeping their data confidential and minimizing exposure. AMD did this using main memory encryption in SEV, with this, individuals VMs were assigned an individual key to encrypt all data, preventing the hypervisor from reading clear text information, as cited by Kaplan [11]. Later in 2017, a new version came out, the AMD Secure Encrypted Virtualization Encrypted State (SEV-ES) that added the additional feature of CPU register state protection, in order to do this, in SEV-ES the VM register state is encrypted at each hypervisor transition, preventing the hypervisor to see the data that is being used, adding confidentiality to in-memory data.

The most recent version is the Secure Encrypted Virtualization Secure Nested Paging (SEV-SNP), built on top of the previous versions while adds hardware based protections like memory integrity protection to prevent attacks like data replay, memory re-mapping, it also implements protection in interrup behavior.

This is, in the reviewed literature, according to Misono *et al.* [4], the first Confidential Virtual Machine (CVM), which are VMs made to to alleviate the programmability and usability challenges of the previously proposed trusted computing technologies, by providing abstraction at the level of the VM. This technology uses a dedicated processor, the AMD Secure Processor (ASP) to manage the security features, this is an ARM-based processor that is separated from the main 32-bit core, but directly integrated in the CPU, which also allows for remote attestation.

The Figure 2.4 shows the overview of the AMD SEV-SNP, where green zones ilustrate CVMs with unmodified software (orange) and with modified software (purple) and the blue areas are trusted components, the vertical black line indicates that each CVM is isolated from the host and other encrypted VMs. This VM executes in an isolated environment where its memory and state registers are encrypted via a unique key managed by the ASP, this safeguards the integrity of guest address translations and strictly isolates the execution by blocking System Management Interrupt (SMI) events and preventing access from the System Management Mode (SMM). This architecture also introduces Virtual Machine Protection Level (VMPL), a four-tiered privilege hierarchy to traditional x86 rings that enables a Secure Virtual Machine Service Module (SVSM) running at the highest level VMPL0 to securely offer services like virtual TPM and live migration.

2.3.3 Intel TDX
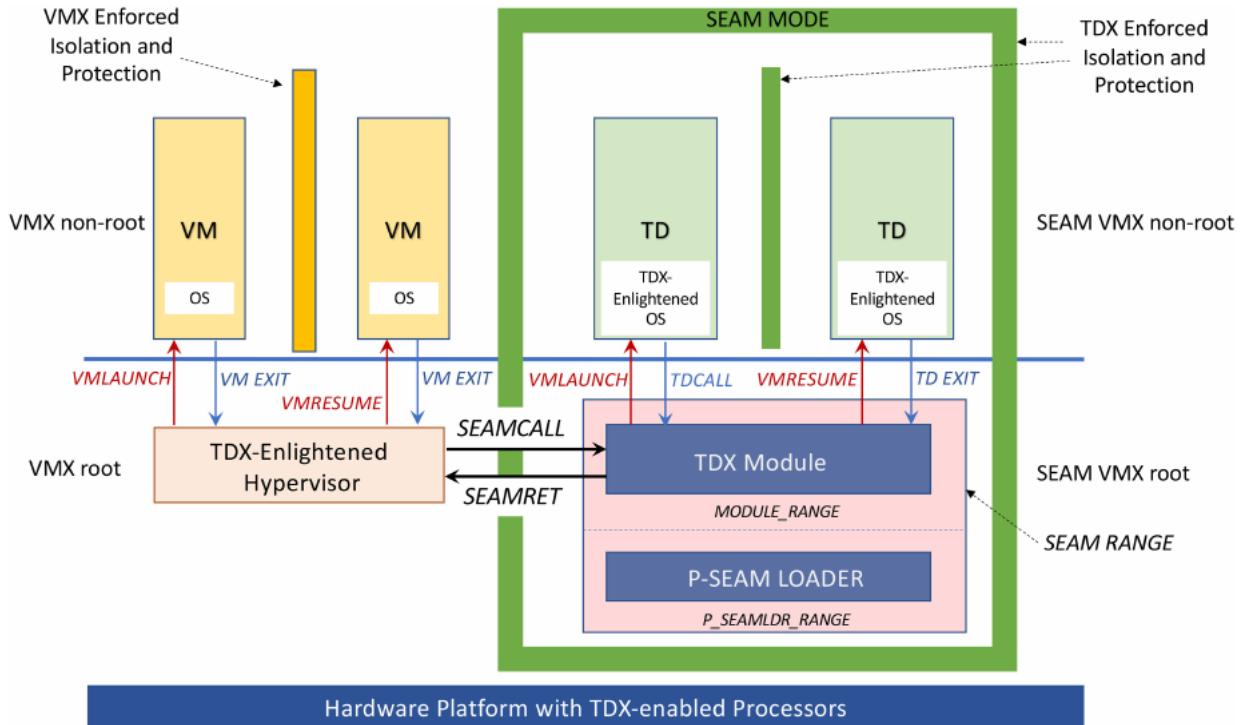
With the same goals as AMD SEV, Intel TDX is an extension introduced with the 4th Generation Intel Xeon Scalable processors to enable confidential computing. This technology facilitates the deployment of VMs, referred to as Trusted Domain (TD)s, under the supervision of the Secure-Arbitration Mode (SEAM), this provides encrypted CPU state and memory, with integrity protection and remote attestation, addressing scenarios where the cloud host platform is considered untrustworthy within the threat model [1].

Technically, TDX builds upon other Intel technologies such as Virtualization Technology (VT) and Multi-key Total Memory Encryption (MKTME), together, these components enforce the confidentiality and integrity of TD memory and CPU states, ensuring that they can neither be accessed nor tampered with by other entities in the environment, such as the hypervisor or neighboring VMs in the same cloud tenant. Although it also uses Data Center Attestation Primitives (DCAP) and SGX for remote attestation.

According to Cheng *et al.* [5] it achieves the previous using a combination of three components, namely, memory access control, runtime memory encryption and an Intel-signed TDX module that does the handling of security-sensitive TD management operations.

---

[1]A threat model is a structured approach to identifyingpotential security risks, defining attacks from the perspective of an adversary to strengthen system defenses [12].

Figure 2.5: Intel TDX Architecture, image by Cheng *et al.* [5].

The Figure 2.5 illustrates the runtime architecture of the TDX technology. The two main components are the TDX compatible processors, that provide functional capabilities like hardware-assisted virtualization, memory encryption, integrity protection, and the TDX Module, that facilitates the manipulation of TD while applying security guarantees, besides this, the module provides two interface functions, a host-side interface and a guest-side interface. The previous module is loaded in to the SEAM range, a portion of memory that is reserved during boot. Another component present in this range is the SEAM loader, which is responsible for installing and updating the module. SEAM provides two execution modes, the SEAM Virtual Machine Extensions (VMX) root mode and non-root mode, a TDX compatible hypervisor operates in the VMX root mode and uses the SEAMCALL introduction to call host-side functions, functions that start with TDH, of the modulea.

This is possible since the the logical processor changes VMX root mode into SEAM VMX root mode and starts executing code within the TDX Module, upon completing it returns in VMX root mode with the instruction SEAMRET. The TDs run in SEAM VMX non-root mode, since TDX supports the execution of unmodified user-level applications within a TD, like a normal VM, but the guest OS kernel (TDX-enlightened OS in Figure 2.5), needs to be altered in order to align with the TDX plataform, namely managing TDX exceptions via an internal handler, implementation of a mechanism for communication between the TD and the TDX module, transitioning memory pages from private to shared I/O operations, although implementation details may vary according to the OS in use.

In similarity with previous TEE implementations, SGX is a set of extensions for the Intel architecture with the goal to allow for integrity and confidentiality during security-sensitive computation in a computer where the all the privileged components might be compromised, as defined by Costan and Devadas [6].

This emerged as a solution to a recurring problem known as secure remote computation, which refers to the difficulty of executing software on a remote computer owned and maintained by an untrusted party. In this context, SGX is one of many attempts to address this challenge by leveraging trusted hardware. To achieve this, the hardware establishes a secure container and a service that enables secure computation for the user.

In simpler terms, trust in this model relies on software attestation, a capability that allows the hardware to prove to a remote user that they are communicating with the intended application on non-compromised hardware, proving that the code has not been altered.

This is achieved through a cryptographic signature that certifies the hash of the enclave's content, the user verifies this signature against a root of trust in the manufacturer's hardware, confirming the validity of the attestation key.

2.4.1 Enclave

An important concept in SGX technology is an enclave, as described by Cheng *et al.* [5], is a protected region within an application's address space that allows the isolation sensitive code and data from the rest of the system.
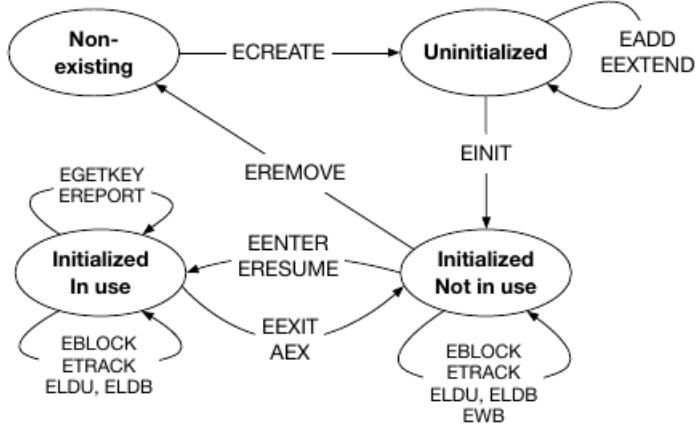
To maintain the security properties and persistent identity of each enclave instance, SGX employs a control structure known as the SGX Enclave Control Structure (SECS). This structure contains the stored metadata for each individual enclave and is kept in a dedicated EPC page with the type `PT_SECS`. During the enclave's lifecycle, the SECS is the first page allocated and the last to be deallocated, serving as the persistent identifier, consequently, the system software utilizes the virtual address of the SECS to reference the specific enclave when issuing SGX instructions. Despite being mapped for identification purposes, the SECS page remains inaccessible to the trusted enclave code and the untrusted system software, due to the present of sensitive information regarding the enclave.

With the enclave's identity and attributes anchored in the SECS, its execution context is defined by a window in the virtual address space, known as the Enclave Linear Address Range (ELRANGE). This range is used to map all code and sensitive data to the corresponding EPC pages and any memory access attempted by the enclave outside this defined area is automatically mapped to access non-EPC memory via the same virtual addresses, allowing for interaction with the rest of the application.

While the ELRANGE defines the memory boundaries of the enclave, the nature of its execution environment is defined by the ATTRIBUTES field within the SECS, via flags such as `DEBUG`, Extended Features Request Mask (XFRM), and `MODE64BIT`. The `DEBUG` is intended to be used during development, since when set, it enables debugging features that allow the reading and modification of the enclave's memory. The XFRM enforces the enclave code to

run with the architectural extensions enabled by the compiler, done by defining the value for the `XCR0` register during execution. Finally, the `MODE64BIT` flag determines whether the enclave operates using the 64-bit architecture, likely for backward compatibility.

Once these operational parameters are defined in the metadata, the enclave must pass through a specific sequence of states to become executable. Consequently, the enclave lifecycle is related with resource management, specifically the allocation of EPC pages. This process begins with creation, where the `ECREATE` instruction transforms a free EPC page into the enclave's SECS, initializing it with architectural parameters and setting the INIT attribute to false, still in this state, the lifecycle enters the loading phase, where the system software uses `EADD` to allocate pages for code and data, while `EEXTEND` updates the enclave's cryptographic measurement for future attestation. After loading, initialization is started by the `EINIT` instruction, that validates a token and sets the `INIT` attribute to true, sealing the enclave from modification and enabling execution in Ring 3. Finally, the destruction of the enclave occurs with the EREMOVE instruction, which deallocates pages by clearing their `VALID` bit, ensuring that the SECS page is only destroyed once all other pages associated with the enclave have been successfully removed, as shown in Figure 2.6.

Figure 2.6: An enclave lifecycle, by Costan and Devadas [6].

2.4.2 Architecture and Workflow

To secure its memory from the untrusted environment, SGX reserves a region in DRAM known as the Processor Reserved Memory (PRM). This region is isolated by the CPU, blocking all non-enclave memory accesses from softwares such as the kernel or hypervisor, as well as protecting against SMM and Direct Memory Access (DMA) attacks from peripherals.
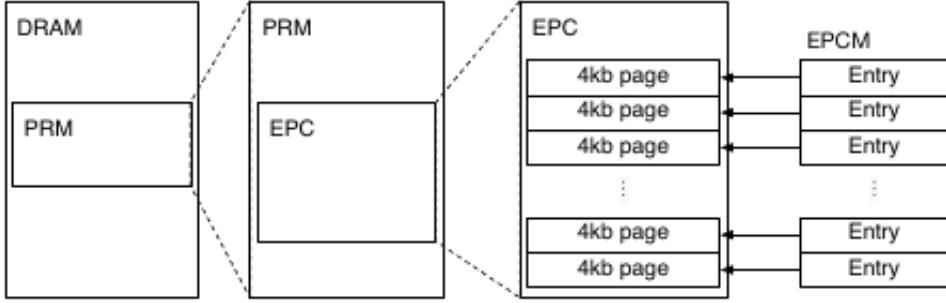
In the PRM alo is the EPC, which consists of 4KB memory pages that store the enclave's code and data. While the allocation of these EPC pages is managed by the untrusted system software, the security is enforced by hardware. The CPU tracks the state and permissions of every EPC page in the Enclave Page Cache Metadata (EPCM) to guarantee that each page belongs to a specific enclave and cannot be mapped by unauthorized entities.

The workflow begins with the loading of code and data into the enclave, during this operation the untrusted system software asks the CPU to copy data from unprotected memory into the EPC pages and assigns them to the just created enclave. After the loading is completed, the system software indicates to the CPU to mark the enclave as initialized, at this point the loaded code becomes executable in the enclave and the loading method is dissabled to prevent modification. Throughout this process, the CPU computes a cryptographic hash of the enclave's content, which becomes the enclave's measurement hash. This allows a remote party to execute a software attestation process, to prove the communication is done with an enclave with a specific measurement hash, and is running in a safe environment.

The execution flow of an enclave is governed by specific CPU instructions, these are similar to context switching between user mode and kernel mode, but the enclave execution remains within Ring 3.

2.4.3 Enclave Page Cache

As mentioned previously, the content of the enclave and necessary data are data structures in the EPC as shown in the Figure 2.7.

Figure 2.7: EPC Memory overview, by Costan and Devadas [6].

The SGX design allows for multiple enclaves running at the same time, which translates to the necessity of a multi-process environment. This is possible due to the capability of the EPC to assign different pages to different enclaves. Non-trusted software cannot directly access this, since it is contained in the PRM.

This mechanism is managed by the same system software that manages the rest of the computers physical memory, which could be a hypervisor or the OS kernel. Both accomplish this with the use of SGX introductions to allocate and free pages in the enclaves. But as mentioned previously, this is not a trusted component, so the SGX processors verifies every allocation decision, and has the possibility to refuse them if they compromise any security guarantee.

To perform these verifications, SGX keeps some information about previous allocation decisions for each EPC page in the EPCM, an array with one entry per page. The three main fields in each entry that determin the ownership of the page are `VALID`, a bit indicating whether an EPC page is currently active and allocated to an enclave, if set to one it signifies the page is in use, preventing reallocation or overwriting by other instructions, `PT`, eight bits that define the type of page and `ENCLAVESECS` which identifies the enclave that owns the page.

The contents of this map can only be read by the SGX as stated in Intel Software Development Manual (SDM).

2.4.4 ECalls and OCalls

In Intel SGX, `ECALL` and `OCALL` are mechanisms for communication between the untrusted application and the trusted enclave.

`ECALL`s allow the untrusted application to invoke a pre-defined function inside the enclave, passing input parameters and pointers to shared memory within the application.

On the other hand, `OCALL`s enable the enclave to invoke a pre-defined function in the untrusted application, but unlike `ECALL`s, cannot share enclave memory directly and must copy parameters into the application's memory before the call.

Both `ECALL`s and `OCALL`s are defined using the Enclave Definition Languagem (EDL), where `ECALL`s are declared in the trusted section and OCALLs in the untrusted section of the

file. During the build process, the Edger8r [2] tool parses the EDL file and generates functions wrappers around the actual functions.
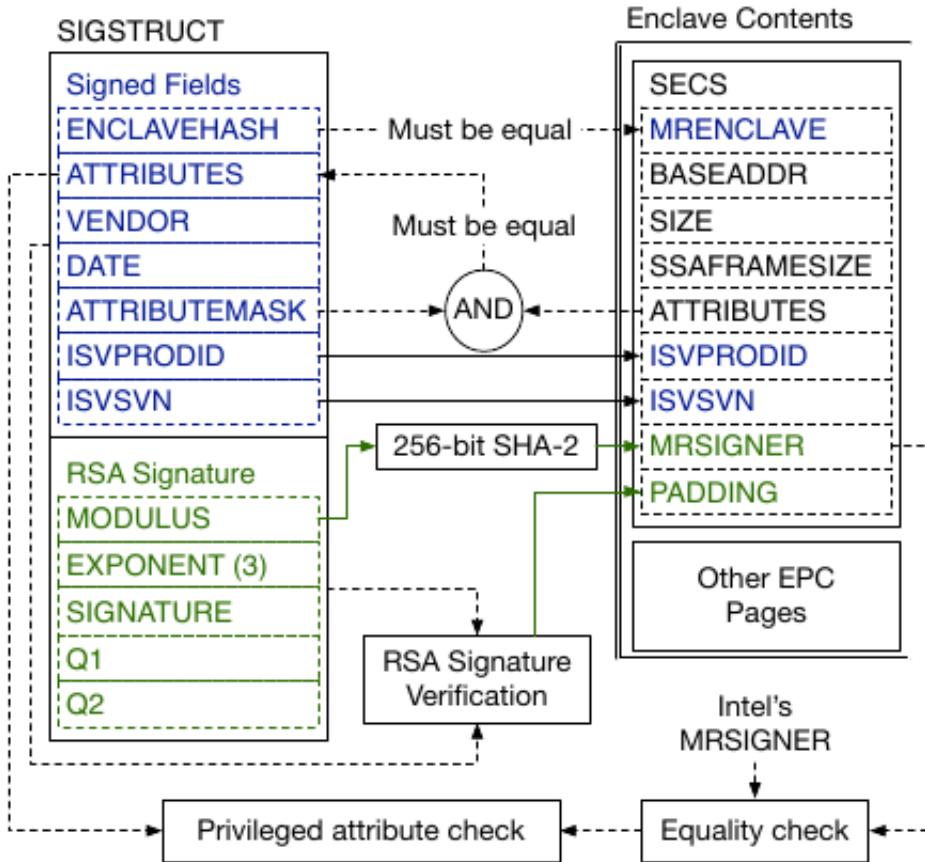
2.4.5   Attestation, Sealing and Unsealing

SGX relies on software attestation to prove to a remote party that it is communicating with a specific piece of software running in a secure container hosted by trusted hardware. Central to this process is the enclave's identity, which is cryptographically established during its initialization and loading phases, as mentioned previously.

The identity of an enclave is primarily defined by two measurement registers located in the enclave's SECS. The first, `MRENCLAVE`, contains a SHA-256 hash that uniquely identifies the enclave's code and data contents, acting as a cryptographic log of the creation process, it incorporates every instruction used to build the enclave, consequently, any modification to the loaded code, data, or their order results in a completely different MRENCLAVE value, ensuring that the running software corresponds exactly to the developer's intent. Complementing this is MRSIGNER, which identifies the entity that signed the enclave rather than the code itself. This register stores the SHA-256 hash of the public RSA key modulus used to sign the SIGSTRUCT certificate, allowing remote parties to trust an enclave based on the author's reputation.

---

[2]Tool provided by the Intel SGX Software Development Kit (SDK) that automatically generates the code that connects the untrusted application and the trusted enclave

Figure 2.8: Establishing an Enclave's Identity, by Costan and Devadas [6].

The attestation process leverages these identities through a mechanism known as local attestation, where an enclave proves its identity to another enclave on the same platform using the EREPORT instruction. To extend this trust remotely, SGX uses a privileged enclave known as the Quoting Enclave, that receives the local report, verifies it, and replaces the local MAC with a cryptographic signature produced by the processor's unique attestation key. This signature, which includes the MRENCLAVE and MRSIGNER, serves as the final proof to the remote service provider that the enclave is authentic and running on genuine hardware.

The initialization process is finalized by the `EINIT` instruction, which validates the enclave's `SIGSTRUCT`, this checks that the measurement in the certificate matches the calculated `MRENCLAVE` and populates the `MRSIGNER` field in the SECS, defining the enclave's identity before it can begin execution.

2.5 Memory and Performance Limitations in SGX

Eventhough SGX provides confidentiality and integrity, it introduces performance overheads and resource constraints. These limitations are derived from the restricted size of the secure memory region and the high cost of transitioning between trusted and untrusted execution modes.

One constraint is the limited capacity of the EPC, since it resides within the PRM, its size is usually 128 MB. When an application's working set exceeds the available EPC capacity, the system software must perform paging to evict EPC pages to unprotected DRAM. Unlike normal paging, SGX paging requires cryptographic operations, where the hardware encrypts the page before writing it to untrusted memory, and verify and decrypt it upon restoration. This causes performance degradation, often referred to as the EPC Bottleneck, making large-memory workloads impractical without careful optimization. Various approaches in related work, such as data partitioning and demand paging optimization, attempt to mitigate this by minimizing the frequency of page swaps.

In addition to memory capacity, the cost of entering (`ECALL`) and exiting (`OCALL`) an enclave is evident. A domain transition involves hardware operations, including flushes, security checks, and context saving, which are necessary to prevent information leakage.
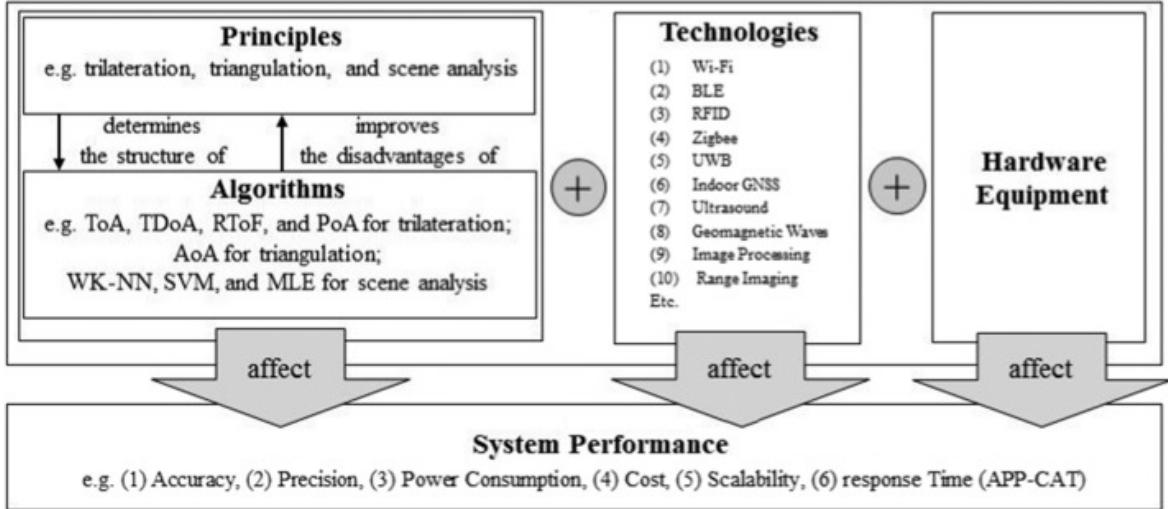
CHAPTER 3

Related Work

This chapter establishes the context for the proposed work by reviewing the related work in WiFi based indoor positioning and occupancy detection. Besides this, it examines the application of SGX in location based systems and analyzes architectural strategies for secure large data processing within enclaves While nearly identical solutions, combining already existing infrastructure monitoring with hardware enforced privacy, are scarce in the literature, distinct bodies of research provide the necessary basis for this dissertation. The review begins by analyzing technical methodologies for indoor positioning using standard WiFi technologies, it then transitions to the critical challenge of user privacy in location analytics. The following chapter analyzes SGX as a possibility for secure indoor positioning, evaluating its current status and performance in privacy preserving location based applications. Finishing with the specific analysis of data processing within SGX, reviewing architectural strategies for handling large scale sensitive data under enclave constraints.

3.1 Indoor Positioning System

As defined by Li *et al.* [7] an Indoor Positioning System is viewed as the grouping of three components: positioning principles and corresponding algorithms, technologies and hardware equipment. These are considered to have a meaningful impact on the performance of the positioning system (see Figure 3.1).

Figure 3.1: General structure of the model of an indoor positioning system by Li *et al.* [7].

3.1.1 Technologies used for Indoor Positioning System

There are several technologies used for an Indoor Positioning System, Li *et al.* [7] divided them in three categories, this overview has been expanded to include Optical and Environmental Sensors, reflecting the methodologies found in research literature, as shown by Table 3.1.

Table 3.1: Indoor Positioning Categories

Category	Technologies
Radio Frequency (RF)	WiFi
	Bluetooth Low Energy (BLE)
	Zigbee
	Ultra-wide band (UWB)
	Radio frequency identification (RFID)
	Indoor Global Navigation Satellite System (GNSS)
	Frequency modulation radio (FM-radio)
Non-Electromagnetic Waves	Ultrasound
	Geomagnetic Waves
Full-spectrum light	Range imaging
	Visible light positioning
	Laser
Optical	Cameras / Computer Vision
	Infrared
Environmental Sensors	CO2 Sensors

In the researche literature, several terms related to indoor positioning appear interchangeably, such as positioning, localization, detection, counting, and tracking. While authors like Zou *et al.* [13] often distinguish these terms based on technical granularity, where localization implies pinpointing specific (x, y) coordinates and tracking involves temporal notion, this thesis adopts a broader operational perspective.

Instead of focusing on the precise coordinate level tracking of targets, the current work prioritizes occupancy detection and crowd estimation in a certain area, in this context, the specific terminology is less about the geometric precision of a single user's location and more about the aggregate presence of devices within a defined zone, treating location as a categorical state.

To achieve these various levels of granularity, ranging from simple presence detection to precise tracking, researchers leverage different characteristics of the WiFi signal. Consequently, the literature presents several distinct methodologies for indoor positioning, each utilizing a specific data metric. Prominent examples include the analysis of WiFi probe requests as demonstrated by Wang *et al.* [14], the measurement of Received Signal Strength Indicator (RSSI) performed by Zou *et al.* [13], and the more complex Channel State Information (CSI) techniques researched by Sobron *et al.* [15]. These approaches, along with others, form the technical basis for the studies reviewed throughout this chapter.

The following chapters will be divided according to the main goal of the Wifi for Indoor Positioning.

3.2.1 Energy Savings and Efficiency

Estimating building occupancy is a necessary step for optimizing building operations, in this scenario WiFi connection counts are presented as a cost efficient proxy for occupancy when comparing to most commom methods, even showing a strong correlation with actual people counts as shown by Mohottige *et al.* [16].

In 2021, Alishahi *et al.* [17] proposed a framework that uses Machine Learning (ML) alongside with statistical analysis to extract occupancy indicators from WiFi connection counts, the authors argued that traditional sensors often suffer from latency, high maintenance costs, and limited scalability. In contrast, stated that associated WiFi device counts, which are devices attempting to authenticate with the AP not necessarily being able to, as a cost effective proxy for human presence. The framework showed positive results, however, Alishahi *et al.* [17] also highlighted limitations that impact the design of this thesis's methodology. It was noted that raw connection counts can possesse a degree of noise and require calibration with ground-truth data to account for two sources of error, the stationary devices like printers, desktops, and projectors that permanently connect to the network must be filtered out and the device to occupant ratio, which is the variance in the number of devices carried by each user. The study emphasizes that these errors become more pronounced at smaller spaces, where attributing a connection to a specific zone is difficult without the precise localization data provided by metrics like RSSI.

In the 2020 Institute of Electrical and Electronics Engineers (IEEE) Radar Conference, Tang *et al.* [18] presented a methodology for people counting based on Passive WiFi Radar (PWR). The authors presented this approach to overcome the limitations of standard WiFi sensing, since RSSI methods are prone to unpredictable fluctuations and false positives due to multipath effects, and CSI techniques typically require specific hardware modifications or high-rate transmissions that degrade network throughput. The overall results presented a high accuracy of 99.54% for tasks of occupancy detection and 98.80% for people counting. Taking into account the positive results, the authors defend that the PWR system is applicable as it leverages existing commercial WiFi APs without requiring any modifications to the WiFi infrastructure or additional devices on the network. However, this advantage comes with added computational complexity, as the system relies completely on signal processing.

Wang *et al.* [19] approached the challenge of discontinuous communication in mobile devices, where smartphones suspend WiFi transmission to conserve battery, causing users to disappear temporarily from network scans. To maintain accuracy during these periods, the authors developed an event triggered framework that estimates occupancy based on entry and exit events. The framework also integrates a location filter using RSSI thresholds to discard devices located outside the target zone by applying a study of the mean values of the measured data in inside and outside positions which could be helpful for the current work, and a non-human filter to exclude stationary equipment, such as printers, via MAC address analysis. The authors also acknowledge limitations such as the reliance on RSSI thresholds restricts the system to zone level accuracy, the behavior of smartphones suspension of data transmission and the detection errors from user behavior, like occupants carrying multiple devices or separating from their smartphones.

In 2017, Ouf *et al.* [20] presented a case study to demonstrate the effectiveness of using WiFi to detect occupancy as opposed to the more common CO_2 sensors. To allow this comparison, the authors monitored WiFi connection counts and CO_2 at the same time, observing concentration levels in a university classroom over one week, using manual occupant counts as ground truth. The analysis revealed that WiFi counts served as a more suited predictor of occupancy, showing a statistical correlation of $r = 0.839$ when CO_2 levels show $r = 0.728$. Furthermore, the authors highlighted that while CO_2 sensors suffered from a detection lag of approximately 20 minutes and susceptibility to non-occupancy related fluctuations, WiFi data provided a more accurate, real time proxy for occupancy with the added benefit of utilizing existing infrastructure without additional cost.

In a Master's thesis, Verma [21] developed a framework for optimizing human resource allocation, directed for cleaning and maintenance, this was done by comparing three occupancy detection models: static university course schedules, thermal occupancy sensors, and WiFi location tracking. The study highlighted that while schedule based models are often inaccurate due to student absenteeism, thermal sensors provided the highest accuracy of approximately 98% by counting heat signatures at doorways. However, the author emphasized the negative considerations on the scalability of thermal sensors, due to a large installation cost. In contrast, the WiFi based model, utilizing the existing network achieved a comparable accuracy

of roughly 90% with zero additional infrastructure costs. A supporting survey within the study validated the viability of this approach, revealing that 95.2% of campus users carried smartphones, with 90.5% actively logged into the university network, confirming that WiFi connection counts serve as a reliable, cost-effective proxy for real time occupancy.

In 2019, Nunna *et al.* [22] proposed a model for an occupancy monitoring system based on the RSSI detected by low-cost microcontrollers. To address the instability of wireless signals, the system employs a mean algorithm, which triggers an occupancy state only when the average RSSI over seven iterations dips below a calibrated threshold. Experimental trials demonstrated a high detection accuracy of 95%, approximately one error every 20 minutes, but only within a limited 3 meter line of sight range. The authors noted that beyond this 3 meter radius, the accuracy degrades due to multipath propagation. [1] This observation is relevant to the current thesis, as it highlights the physical limitations of RSSI based ranging in indoor environments.

3.2.2 Behavioral Monitoring

A case study performed by Christl [24] in 2024 analyzed the privacy implications of existing technologies for behavioral monitoring and profiling. The report examined solutions from major vendors such as Cisco, Juniper, Spacewell, and Locatee, identifying a trend where workplace infrastructure is used for tracking user behavior. Specifically in Section 3.2, the study focuses on Cisco, the manufacturer of the devices generating the logs utilized in this dissertation. The report shows a product called Cisco Spaces, a cloud platform that processes large amounts of data to profile user behavior. For user classification, the report notes that the system categorizes persons into groups like employee, student or guest based on the SSID category, this points that the system identifies the type of user based of the WiFi network the user connects to, allowing for distinct tracking of different user groups and their behaviors.

In 2016, Zhou *et al.* [25] proposed the EDUM (EDUcation Measurement) system to characterize educational behaviors using data collected from large-scale WiFi infrastructures, analyzing students punctuality based on longitudinal WiFi connection traces and to assess lecture attractiveness and student distraction. The system utilizes the mobile phone's interactive state, the screen on/off status, at a per minute basis. Deployed at Tsinghua University with approximately 700 student volunteers and 2,800 APs, the study provided results that revealed a negative correlation between high mobile phone usage during class and academic performance (GPA), and confirmed that students seated in the back rows exhibit significantly higher distraction levels than those in the front.

Later in 2025, Álvarez-Merino *et al.* [26] realized a study to investigate indoor location technologies for future integration in Smart Education (SE) environments, the authors reviewed several technologies, as mentioned earlier on this chapter, but for the sake of the thesis the focus will be toward WiFi. The authors identified WiFi as a accessible technology for SE, citing it's availability in educational institutions and the connectivity provided by networks

---

[1]Phenomenon in wireless communication where a transmitted signal reaches the receiver antenna through multiple paths due to reflections and scattering from environmental objects, resulting in different signal copies with varying attenuation, phase shifts, and time delays as defined by Kaluuba *et al.* [23]

like eduroam, the one used on the present thesis. Regarding its capabilities, the study showed the potential of the IEEE 802.11mc standard, which allows for ranging, within approximately 1 meter, and preserves user privacy with calculations on the client device side. However, in their experimental POC for automatic attendance, the WiFi approach achieved an accuracy of 93.77% using a regression model, it was outperformed by 5G (97.21%), leading the authors to conclude that while WiFi is a useful tool due to it's low cost and presence, a fusion of technologies provides the most robust solution for critical attendance monitoring.

3.2.3 Occupancy Monitoring

In the 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Mohottige *et al.* [16] proposed a study to assess the feasibility of using WiFi AP infrastructure for room level occupancy monitoring across the University of New South Wales campus, a scenario similar to this thesis. The study had a duration of four weeks across rooms with varying numbers of APs, comparing WiFi data against beam counter sensors and ground truth enrollment numbers. To account for transient users, passing by, connecting to the APs, connections lasting less than five minutes were filtered out. The results indicated that the WiFi method achieved a stronger correlation with actual occupancy ($R = 0.85$) compared to the beam counters ($R = 0.68$). it also demonstrated a lower error rate.

Later in 2022, Mohottige *et al.* [27] published in the IEEE Sensors Journal a ML framework to infer classroom occupancy, sharing the same goal as the current work but employing standard statistical modeling. The authors utilized daily WiFi logs from the university's IT department, a scenario identical to this dissertation, comprising data from 70 APs including UU, MAC address, event timestamps, and AP names. Besides the similar data source, Mohottige *et al.* [27] identified critical limitations in using raw logs, such as the overlapping coverage of APs (where a student inside a room connects to a hallway AP), the presence of "bystanders", and the variability of multiple device connections per user. Ultimately, the framework achieved a symmetric Mean Absolute Percentage Error of 13.1%, a result comparable to dedicated beam counter sensors, this once more demonstrated that existing WiFi infrastructure can yield accurate occupancy estimation with no additional deployment costs only computing costs.

In 2022, for a Master's thesis, Carrol-Woolery [28] proposed the "Building Floor Zone" technique to improve WiFi counting precision without requiring new hardware. The methodology involved mapping APs to specific zones based on room numbering and assigning hallway APs to the center of adjacent rooms, in similarity with the present thesis. Experimental analysis confirmed that this approach resulted in a higher statistical correlation with schedule based ground truth compared to standard floor level aggregation. An interesting mention is that the study found that counting all unique connections provided better accuracy than filtering for long duration sessions, challenging the assumption that transient users are merely noise.

Earlier in 2016, Anand *et al.* [29] proposed a attendance system that combines facial recognition for user authentication with WiFi network analysis to verify the student's location.

While the authors acknowledged methods like trilateration [2], due to signal interference adopted a fingerprinting approach using the ML algorithm. Regarding the results, the system demonstrated positive precision, achieving a positioning error between 1 and 2.5 meters. In practical testing, this methodology identified whether a student was inside the target classroom 94% of the time.

Khan *et al.* [30] presented a smartphone based attendance system utilizing WiFi signal strength for indoor localization. As noted in previous works, distinguishing user presence within confined boundaries, like a specific room, is challenging due to signal fluctuations. To address this, the authors employed a zone based fingerprinting approach, dividing the space into virtual grids and using machine learning to classify them. A valuable insight for this thesis is the authors explicit classification of signal quality, which categorizes RSSI values into ranges according to signal strength, something that could be necessary in the current work. The authors system achieved 100% accuracy in one test room and 92% in another. Furthermore, the study demonstrated significant robustness, the accuracy remained stable even when individual APs were removed or when different smartphone models were used.

3.2.4 Fingerprinting

In a study focusing on probabilistic localization, Le Dortz *et al.* [31] developed a WiFi fingerprinting system that utilizes probability distribution comparisons rather than simple RSSI averaging. Crucially for the context of this thesis, the methodology also used a offline phase to construct a radio map, a database where fingerprints were manually collected at several known rooms throughout the building. During the offline phase, the authors collected 100 RSSI samples at each reference point to estimate the signal strength probability distributions for every visible access point. This data collection process demonstrates the significant calibration effort, necessary in fingerprinting approaches. While their method achieved a median positioning error of 2.4 meters, the requirement to build and maintain such a detailed radio map highlights the scalability challenges that the passive, infrastructure based approach proposed in this current work seeks to eliminate.

Similarly, in 2020, Ninh *et al.* [32] proposed a random statistical method for indoor positioning that relies on the same two phase architecture common to fingerprinting solutions. The system explicitly distinguishes between an offline handling process, where a large number of WiFi signals are collected at specific reference points to create a database, and an online positioning process that uses the Mahalanobis distance to match live user data against this stored radio map.

3.3 Approaches to Privacy in Literature

The ubiquity of WiFi networks as a sensing infrastructure presents a fundamental problem: the same granular data that enables precise occupancy monitoring also introduces significant privacy risks for the individuals being tracked. As noted in the previous sections, transforming

---

[2]Trilateration is a geometric method of determining location by measuring distances from at least three known reference points using signal strength.

a standard wireless network into a soft sensor involves analyzing device footprints, specifically MAC addresses and signal characteristics, which can serve as strong identifiers of specific individuals.

Consequently, the academic literature demonstrates a variety of methodologies designed to connect the trade off between service utility and user privacy. These approaches range from data minimization strategies, that avoid collecting unique identifiers, to more complex anonymization schemes like hashing and k-anonymity. However, as the demand for room level precision increases, so does the inadequacy of traditional methods.

This section reviews these privacy preserving strategies, categorizing them into distinct paradigms found in the literature: omission of privacy mechanisms, where technical feasibility takes precedence; data aggregation and minimization, where sensitive data is aggregated at the source; anonymization and pseudonymization, where identifiers are masked; institutional compliance and ethical clearance, relying on governance frameworks; and volunteer based or opt-in models.

3.3.1 Omission/Inexistence of Privacy Mechanisms

Wang *et al.* [19] focused on the technical application of detecting inactive smartphones to improve detection accuracy. While the system utilizes MAC address analysis to filter non-human devices, the study does not detail any mechanisms for anonymizing these addresses or protecting the identity of the smartphone owners.

Furthermore, Christl [24] provided an analysis of commercial indoor positioning systems, identifying a lack of privacy by design in several applications. The report shows that these platforms often prioritize profiling and categorizing users into groups like employees or students, over data protection, allowing for the tracking of individual movements without obfuscation or user optional mechanisms.

Mohottige *et al.* [16] conducted a validation of WiFi occupancy sensing by benchmarking it against hardware counters across a university campus. The study focused on the utility of the metadata, like user's MAC address which can endanger students privacy, since the authors did not detail privacy preserving architectures, it leads to the assumption of inexistence of privacy preserving mechanisms.

3.3.2 Data Aggregation and Minimization

Alishahi *et al.* [17] adopted a data minimization strategy by relying only on aggregated connection counts per AP. By converting raw network logs into numerical totals before analysis, the framework inherently discards individual identifiers, ensuring that no unique user data is retained.

Tang *et al.* [18] took the concept of data minimization to the physical layer by employing a device free approach known as Passive WiFi Radar. Instead of decoding data packets or logging MAC addresses, this method analyzes signal reflections caused by moving bodies. Consequently, the system avoids collecting any digital identifiers, making the data inherently anonymous and decoupling the occupancy detection from the users' personal devices.

Similarly, Ouf *et al.* [20] employed an aggregation strategy to validate WiFi sensing against environmental benchmarks. By utilizing the total number of connections as a bulk metric to correlate with CO_2 levels, the authors treated the crowd as a single entity. This approach avoids the privacy difficulties of tracking, as the system monitors the state of the room and not the behavior of the individuals within it.

Verma [21] demonstrated a resource optimization framework that relies on aggregated occupancy density. By only using bulk connection counts directly from the university's central IT department, rather than logging individual devices, the system calculates the "usage intensity" of classrooms.

3.3.3 Anonymization and Pseudonymization

Carrol-Woolery [28] implemented a form of pseudonymization to mitigate the risks of long-term profiling. In this study, the unique identifiers were masked using a hashing algorithm that was reset every 24 hours, something similar to what will be done during the development process of this dissertation.

3.3.4 Institutional Compliance and Ethical Clearance

Mohottige *et al.* [27] addressed the privacy implications of tracking students by operating under a governance framework, since the study obtained formal ethical clearance from the university.

3.3.5 Volunteer Based Model or Opt-in Model

Anand *et al.* [29] implemented a privacy model based on user interaction. In their system, attendance is not monitored passively, instead, students must actively engage with a smartphone application to capture a facial scan and submit their WiFi fingerprint, which falls on the category of an opt-in, ensuring that location data is only transmitted with the user's direct consent and knowledge.

Zhou *et al.* [25] employed a volunteer based model to justify the collection of highly granular behavioral data. In their "EDUM" system, the researchers recruited approximately 700 students who consented to having their WiFi traces and mobile application usage monitored.

Álvarez-Merino *et al.* [26] highlighted the privacy advantages of the IEEE 802.11mc standard, which shifts the location calculation to the client device. To demonstrate this architecture, the authors developed a mobile application, by requiring students to actively install and engage with this software to register their presence, the system uses a opt-in model.

Likewise, Khan *et al.* [30] developed a smartphone based attendance system that leverages RSSI zoning to verify student presence. Since the primary utility is the validation of attendance records using the students smartphone, the architecture operates on an opt-in model where students consent to the tracking of their devices.

3.4 PREVIOUS WORKS WITH SGX IN INDOOR POSITIONING

While the literature regarding the specific application of SGX to indoor positioning remains scarce, a select number of studies have established a base foundation for this approach.

To address the privacy concerns associated with collecting user location data, Yan *et al.* [33] proposed a trusted computing framework based on SGX. The authors stated that traditional techniques like k-anonymity or encryption, may include computational overheads or require dependencies on a Trusted Third Party (TTP). In contrast, their work demonstrates that SGX allows a privacy scheme, by processing sensitive location data within a hardware protected enclave, the system ensures that raw identity information remains inaccessible to the operating system or the service provider, allowing secure analysis without exposing individual user data. This approach validates the architectural choice in this thesis to utilize SGX enclaves for processing sensitive WiFi logs, ensuring compliance with privacy standards while maintaining system performance.
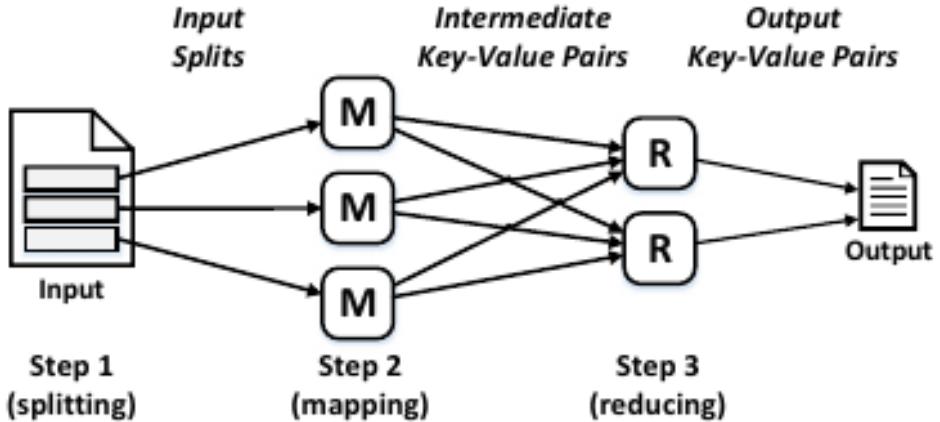
Complementing the architectural advantages mentioned in the previous work, Kulkarni *et al.* [34] provided an evaluation of SGX for location based services, explicitly comparing it against traditional k-anonymity techniques. The study highlighted a trade off in privacy handling, that traditional obfuscation methods degrades service accuracy to achieve privacy. In contrast, their experiments demonstrated that an SGX based approach provides better result accuracy because the computation occurs on exact data within the secure enclave, protected from the host system. Furthermore, the authors quantified the performance cost of this security, finding that SGX introduces only a small compared to bare metal implementation, due to the one time events such as the enclave creation, copying data, sealing and unsealing. This finding is important for the current thesis, as it validates that shifting the occupancy estimation logic into an SGX enclave is a viable strategy that secures student data without compromising the real time performance or the accuracy of the occupancy counts.

3.5 DATA PROCESSING IN SGX

Processing large scale data within SGX introduces memory challenges, primarily due to the limited size of the EPC and the high cost of transitions between trusted and untrusted memory. Schuster *et al.* [8] presented Verifiable Confidential Cloud Computing (VC3), a distributed MapReduce model [3] designed to address these limitations by strictly minimizing the Trusted Computing Base (TCB). Unlike systems that attempt to execute entire legacy applications or libraries from the OS inside an enclave, VC3 adopts a partitioned architecture where the heavy lifting of job scheduling and data storage remains in the untrusted Hadoop (open source framework for distributed storage and processing of large datasets). Only the specific map and reduce functions, along with a minimal cryptographic shim, are loaded into the protected enclave. To cope with the memory constraints of the enclave, VC3 relies on a streaming data model. Encrypted data is ingested in input splits and processed within the enclaves heap, and the results are encrypted and then streamed out. To mitigate the performance overhead of SGX context switches, between ECalls and OCalls, VC3 utilizes a shared memory region for communication and implements aggressive batching. By processing key value pairs in batches, rather than individually, the system reduces the frequency of expensive enclave transitions.

---

[3]Programming model for processing large data sets, users write map and reduce functions, and the execution of both functions is automatically parallelized and distributed, as defined by Dean and Ghemawat [35]

Figure 3.2: Steps of MapReduce with mappers (M) and reducers (R), image by Schuster *et al.* [8].

Furthermore, to avoid the memory footprint of standard libraries, VC3 utilizes a custom runtime library and heap allocator, ensuring that the limited secure memory is dedicated primarily to the user's data and logic.

In contrast to the streaming model employed by MapReduce frameworks, Priebe *et al.* [36] proposed EnclaveDB, a database engine designed to minimize the overhead of data processing by keeping all sensitive state resident within the enclave memory. Leveraging the Hekaton, a Structured Query Language (SQL) engine, EnclaveDB avoids the cost of disk I/O by ensuring that encryption and decryption occur only at the transaction boundaries rather than at the page level during execution. To reduce the TCB and optimize memory usage, the system adopts a pre-compiled execution model, complex query parsing and optimization are offloaded to a trusted client, and only the resulting native machine code is deployed to the enclave. This architecture reduces the frequency of context switches, a known performance bottleneck in SGX, by executing entire transactions within the protected region without passing control back to the untrusted host until the transaction commits. Furthermore, to address the integrity of persistent data without the high memory cost of maintaining, a hash tree over the transaction log, EnclaveDB implements a lightweight protocol using hardware based counters and serialization points to guarantee the data updates and prevent rollback attacks.

In the domain of data analytics, Shaon *et al.* [37] introduced SGX-BigMatrix, a framework for performing secure matrix operations on encrypted data. Recognizing that large scale matrices often exceed the limited EPC capacity, the authors developed a BigMatrix abstraction that partitions data into fixed size, encrypted blocks. These blocks are dynamically loaded into the enclave for processing and evicted to untrusted memory via a software managed swapping mechanism, effectively bypassing the performance overhead of the operating system's paging. Crucially, to address the vulnerability of side-channel attacks, SGX-BigMatrix enforces data oblivious execution, by utilizing vectorized oblivious primitives and a specialized compiler to manage block transitions, the framework ensures that the sequence of memory accesses remains independent of the sensitive data, thereby preventing adversaries from inferring

28

information through memory access traces.

The literature confirms that while WiFi connection counts are a viable proxy for occupancy, current implementations either lack rigorous privacy protections or rely on opt-in models that limit scalability.

While SGX provides the necessary hardware enforced privacy, standard big data frameworks like VC3 or EnclaveDB introduce development complexity.

References

[1] M. Sabt, M. Achemlal, and A. Bouabdallah, «Trusted execution environment: What it is, and what it is not», vol. 1, pp. 57–64, 2015. DOI: 10.1109/Trustcom.2015.357.

[2] J. González, «Operating system support for run-time security with a trusted execution environment», Mar. 2015. DOI: 10.13140/RG.2.1.4827.8161.

[3] S. Pinto and N. Santos, «Demystifying arm trustzone: A comprehensive survey», *ACM Comput. Surv.*, vol. 51, no. 6, Jan. 2019, ISSN: 0360-0300. DOI: 10.1145/3291047. [Online]. Available: https://doi.org/10.1145/3291047.

[4] M. Misono, D. Stavrakakis, N. Santos, and P. Bhatotia, «Confidential vms explained: An empirical analysis of amd sev-snp and intel tdx», *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 8, no. 3, Dec. 2024. DOI: 10.1145/3700418. [Online]. Available: https://doi.org/10.1145/3700418.

[5] P.-C. Cheng, W. Ozga, E. Valdez, *et al.*, «Intel tdx demystified: A top-down approach», *ACM Comput. Surv.*, vol. 56, no. 9, Apr. 2024, ISSN: 0360-0300. DOI: 10.1145/3652597. [Online]. Available: https://doi.org/10.1145/3652597.

[6] V. Costan and S. Devadas, «Intel sgx explained»,

[7] C. T. Li, J. C. Cheng, and K. Chen, «Top 10 technologies for indoor positioning on construction sites», *Automation in Construction*, vol. 118, p. 103 309, 2020, ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2020.103309. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519306016.

[8] F. Schuster, M. Costa, C. Fournet, *et al.*, «Vc3: Trustworthy data analytics in the cloud using sgx», pp. 38–54, 2015. DOI: 10.1109/SP.2015.10.

[9] P. de Keyser, «8 - metadata formats and indexing», in *Indexing*, ser. Chandos Information Professional Series, P. de Keyser, Ed., Chandos Publishing, 2012, pp. 143–166, ISBN: 978-1-84334-292-2. DOI: https://doi.org/10.1016/B978-1-84334-292-2.50008-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781843342922500088.

[10] P. Quinn, «The difficulty of defining sensitive data – the concept of sensitive data in the eu data protection framework», *SSRN Electronic Journal*, Jan. 2020. DOI: 10.2139/ssrn.3713134.

[11] D. Kaplan, «Hardware vm isolation in the cloud», *Commun. ACM*, vol. 67, no. 1, pp. 54–59, Dec. 2023, ISSN: 0001-0782. DOI: 10.1145/3624576. [Online]. Available: https://doi.org/10.1145/3624576.

[12] Q. Do, B. Martini, and K.-K. R. Choo, «The role of the adversary model in applied security research», *Computers and Security*, vol. 81, pp. 156–181, 2019, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2018.12.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818306369.

[13] H. Zou, H. Jiang, J. Yang, L. Xie, and C. Spanos, «Non-intrusive occupancy sensing in commercial buildings», *Energy and Buildings*, vol. 154, pp. 633–643, 2017, ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.08.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778816311987.

[14] W. Wang, J. Wang, J. Chen, G. Huang, and X. Guo, «Multi-zone outdoor air coordination through wi-fi probe-based occupancy sensing», *Energy and Buildings*, vol. 159, pp. 495–507, 2018, ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.11.041. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778817321217.

[15] I. Sobron, J. Del Ser, I. Eizmendi, and M. Vélez, «Device-free people counting in iot environments: New insights, results, and open challenges», *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4396–4408, 2018. DOI: `10.1109/JIOT.2018.2806990`.

[16] I. P. Mohottige, T. Sutjarittham, N. Raju, H. H. Gharakheili, and V. Sivaraman, «Role of campus wifi infrastructure for occupancy monitoring in a large university», pp. 1–5, 2018. DOI: `10.1109/ICIAFS.2018.8913341`.

[17] N. Alishahi, M. Nik-Bakht, and M. M. Ouf, «A framework to identify key occupancy indicators for optimizing building operation using wifi connection count data», *Building and Environment*, vol. 200, p. 107 936, 2021, ISSN: 0360-1323. DOI: `https://doi.org/10.1016/j.buildenv.2021.107936`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0360132321003401`.

[18] C. Tang, W. Li, S. Vishwakarma, K. Chetty, S. Julier, and K. Woodbridge, «Occupancy detection and people counting using wifi passive radar», pp. 1–6, 2020. DOI: `10.1109/RadarConf2043947.2020.9266493`.

[19] J. Wang, N. C. F. Tse, and J. Y. C. Chan, «Wi-fi based occupancy detection in a complex indoor space under discontinuous wireless communication: A robust filtering based on event-triggered updating», *Building and Environment*, vol. 151, pp. 228–239, 2019, ISSN: 0360-1323. DOI: `https://doi.org/10.1016/j.buildenv.2019.01.043`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0360132319300794`.

[20] J. M. Ouf, M. H. Issa, A. Azzouz, and A.-M. Sadick, «Effectiveness of using wifi technologies to detect and predict building occupancy», *Research Gate*, vol. 2, 2017. DOI: `https://doi.org/10.1051/sbuild/2017005`.

[21] P. Verma, «Classroom occupancy-based human resource optimization using sensor- and wifi-based location tracking», 2017. DOI: `10.7939/R3HT2GQ36`. [Online]. Available: `https://doi.org/10.7939/R3HT2GQ36`.

[22] A. Nunna, A. Varma, R. Kumar N., S. Tarun, and M. Sangeetha, «Classroom automation using rssi», pp. 1–6, 2019. DOI: `10.1109/ICESIP46348.2019.8938387`.

[23] L. Kaluuba, G. Taban-Wani, and D. Waigumbulizi, «The fading channel problem and its impact on wireless communication systems in uganda», J. Mwakali and G. Taban-Wani, Eds., pp. 621–634, 2006. DOI: `https://doi.org/10.1016/B978-008045312-5/50068-6`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780080453125500686`.

[24] W. Christl, «Tracking indoor location, movement and desk occupancy in the workplace», *Research Gate*, 2019. DOI: `https://doi.org/10.1109/ISECon.2019.8882085`.

[25] M. Zhou, M. Ma, Y. Zhang, K. SuiA, D. Pei, and T. Moscibroda, «Edum: Classroom education measurements via large-scale wifi networks», UbiComp '16, pp. 316–327, 2016. DOI: `10.1145/2971648.2971657`. [Online]. Available: `https://doi.org/10.1145/2971648.2971657`.

[26] C. S. Álvarez-Merino, E. J. Khatib, A. T. Muñoz, and R. B. Moreno, «Integrating indoor localization technologies for enhanced smart education: Challenges, innovations, and applications», *IEEE Access*, vol. 13, pp. 105 317–105 333, 2025. DOI: `10.1109/ACCESS.2025.3578718`.

[27] I. P. Mohottige, H. H. Gharakheili, T. Moors, and V. Sivaraman, «Modeling classroom occupancy using data of wifi infrastructure in a university campus», *IEEE Sensors Journal*, vol. 22, no. 10, pp. 9981–9996, 2022. DOI: `10.1109/JSEN.2022.3165138`.

[28] L. Carrol-Woolery, «Localization and counting of indoor populations on a university campus using wi-fi connection logs and floor plans», 2022. [Online]. Available: `http://hdl.handle.net/10012/19281`.

[29] S. Anand, K. Bijlani, S. Suresh, and P. Praphul, «Attendance monitoring in classroom using smartphone and wi-fi fingerprinting», *2016 IEEE Eighth International Conference on Technology for Education*, pp. 62–67, 2016. DOI: `10.1109/T4E.2016.021`.

[30] S. M. Khan, M. T. Maliha, M. S. Haque, and A. Rahman, «Wifi received signal strength (rss) based automated attendance system for educational institutions», pp. 172–180, 2025. DOI: `10.1145/3704522.3704523`. [Online]. Available: `https://doi.org/10.1145/3704522.3704523`.

[31] N. Le Dortz, F. Gain, and P. Zetterberg, «Wifi fingerprint indoor positioning system using probability distribution comparison», in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2301–2304. DOI: `10.1109/ICASSP.2012.6288374`.

[32] D. B. Ninh, J. He, V. T. Trung, and D. P. Huy, «An effective random statistical method for indoor positioning system using wifi fingerprinting», *Future Generation Computer Systems*, vol. 109, pp. 238–248, 2020, ISSN: 0167-739X. DOI: `https://doi.org/10.1016/j.future.2020.03.043`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0167739X19324835`.

[33] Z. Yan, X. Qian, S. Liu, and R. Deng, «Privacy protection in 5g positioning and location-based services based on sgx», *ACM Trans. Sen. Netw.*, vol. 18, no. 3, Aug. 2022, ISSN: 1550-4859. DOI: `10.1145/3512892`. [Online]. Available: `https://doi.org/10.1145/3512892`.

[34] V. Kulkarni, B. Chapuis, and B. Garbinato, «Privacy-preserving location-based services by using intel sgx», *Proceedings of the First International Workshop on Human-centered Sensing, Networking, and Systems (HumanSys'17)*, no. CONFERENCE, 6 p. CHAPUIS, Bertil est chercheur à la HES-SO, HEIG-VD, depuis 2021. DOI: `https://doi.org/10.1145/3144730.3144739`. [Online]. Available: `http://arodes.hes-so.ch/record/15319`.

[35] J. Dean and S. Ghemawat, «Mapreduce: Simplified data processing on large clusters», *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008, ISSN: 0001-0782. DOI: `10.1145/1327452.1327492`. [Online]. Available: `https://doi.org/10.1145/1327452.1327492`.

[36] C. Priebe, K. Vaswani, and M. Costa, «Enclavedb: A secure database using sgx», pp. 264–278, 2018. DOI: `10.1109/SP.2018.00025`.

[37] F. Shaon, M. Kantarcioglu, Z. Lin, and L. Khan, «Sgx-bigmatrix: A practical encrypted data analytic framework with trusted processors», CCS '17, pp. 1211–1228, 2017. DOI: `10.1145/3133956.3134095`. [Online]. Available: `https://doi.org/10.1145/3133956.3134095`.

Additional content