Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Defence, Civil Protection and Sport DDPS

National Cyber Security Centre NCSC
Analysis and Prevention

5 May 2025 (revised)

Technology brief

Confidential computing

1 Introduction

Ensuring the security of information processing operations in IT systems is a major challenge, particularly given the increasing use of cloud computing and related services. While data storage and transmission can be effectively secured using cryptographic methods, processing data securely remains a core challenge – particularly when the software (code) controlling the process and the execution environment are managed by a cloud provider and lie outside the user's control.[1] Various approaches to address this issue have been developed – and, in some cases, implemented.

This technology brief introduces and discusses one such approach: confidential computing. It describes how it works, outlines its advantages and disadvantages, presents current market solutions, and explores future prospects.

2 The challenge

As mentioned in the introduction, processing data securely within an IT system remains a major challenge. The integrity of the execution environment (i.e. the environment in which the application runs) and the confidentiality of the data during processing must both be guaranteed. As shown in Figure 1, the execution environment includes not only the application itself (including both data and code), which usually runs inside a virtual machine (VM), but also the VM's guest operating system, hypervisor, host operating system, and underlying hardware.[2] All of these components must be protected against compromise and data exfiltration; otherwise, the security of the application cannot be ensured. The components highlighted in red in

[1] In this context, the term *data in use* is also used (as opposed to *data at rest* for stored data and *data in transit* for transmitted data).

[2] If virtualisation is largely avoided and no VM is used, the situation becomes simpler. In that case, the application can run directly on the hardware and the host operating system.

Figure 1 are therefore critical to secure data processing and must be assumed to be trustworthy.[3]

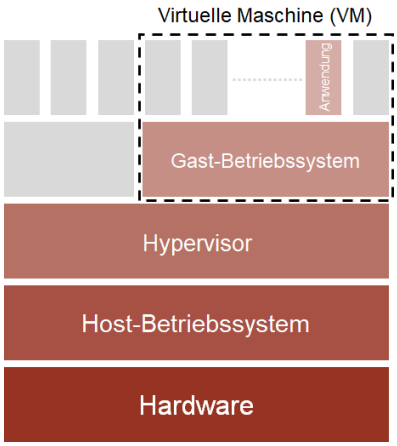

Figure 1: Processing of data in an IT system

If an application uses the services of a cloud provider instead of being run locally (*on premises*), the data processor must be able to rely not only on the trustworthiness of their own IT systems, but also on that of the cloud provider and its infrastructure. This is illustrated in Figure 2. In this setup, the data processor sends data and code to the cloud provider. The cloud provider then runs the code on the data within an execution environment that it manages, returning the result to the data processor in a secure manner. However, in most cases, the data processor has no way of directly verifying the security of the execution environment. Instead, they must rely on the cloud provider's assurances, as well as supporting certifications and attestations.

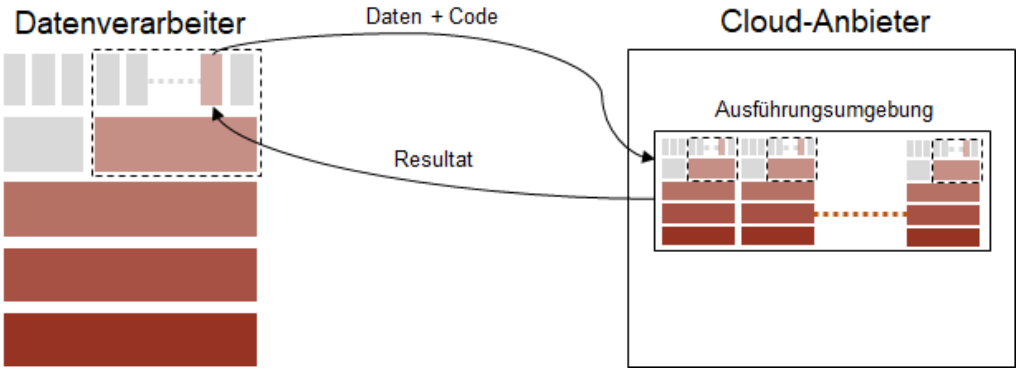

Figure 2: The execution of an application in the context of cloud computing

This challenge is commonly described in the literature as *secure remote computation* – the problem of providing a trustworthy execution environment on third-party infrastructure that cannot be directly controlled, while preventing data exfiltration and ensuring secure data processing overall. This question essentially reverses a more familiar problem: how to protect

[3] In IT security, (system) components whose security can neither be proven nor verified must be assumed to be trustworthy. In their entirety, they represent the Trusted Computing Base (TCB), which must be kept as small as possible.

trusted IT infrastructure from potentially malicious code (such as Java applets). This issue can be partially addressed through sandboxing, a method that minimises the execution environment and isolates it from the rest of the infrastructure to prevent malicious code from interfering with or compromising the system. However, sandboxing is not a viable solution to the secure remote computation problem because, in this case, the goal is to protect the data and code rather than the infrastructure. Other, often more complex, approaches are required.

3 Solutions

In the long term, fully homomorphic encryption (FHE) is likely to be the solution to the problem of secure remote computation. FHE enables the data processor to provide their data to the cloud provider in an encrypted format. This allows the cloud provider to process the data without ever decrypting them; in other words, the cloud provider never sees the data in plain text. Once processing is complete, the result is returned encrypted, allowing the data processor to decrypt it. This offers a software-based solution to the secure remote computation problem. Unfortunately, the currently available FHE methods are still too inefficient for general data processing tasks.

Until practical FHE becomes available, the secure remote computation problem can be solved using trusted hardware components. These components can ensure the integrity of the execution environment to a certain extent. Trusted hardware components come in various forms, including smart cards, security elements, hardware security modules and trusted platform modules (TPMs). TPMs, in particular, have been widely deployed as part of the Trusted Computing Initiative by the Trusted Computing Group (TCG).[4] For instance, in standard IT systems, a TPM can guarantee that the system boots into a defined state and has not been compromised by tampered software. This secures the integrity of the execution environment – at least until unknown software is loaded and executed. However, since this will inevitably happen, the effort involved in using TPMs may outweigh their benefits.

An industry consortium has proposed a simpler and more widely applicable form of hardware support under the name *confidential computing*.[5] It involves using hardware to create a trusted execution environment (TEE), which is a secure enclave inside the processor where data can be processed in a protected mode. In this configuration, all processing occurs within the processor itself and no software other than the application operating the TEE can access or read data from it. In addition, before any data is loaded, the application can validate both the TEE and the code it intends to run inside it – this process is called *remote attestation*. The TEE also supports sealing to allow state variables or results to be temporarily stored. Sealing refers to the cryptographic protection of data stored in designated memory areas. The cryptographic keys required for remote attestation and sealing only exist in the relevant TEEs or the hardware and cannot be extracted from them without considerable effort.

Figure 3 shows what it looks like to run an application using a TEE. Note that in this scenario, the only trusted components are the application itself and the TEE (shown in red in the diagram). Crucially, the IT system operator has no access to the application's data, provided that the hardware fully meets the confidentiality requirements of confidential computing. In practice, this means that trust must be placed in the hardware supplier.

[4] https://trustedcomputinggroup.org

[5] https://confidentialcomputing.io

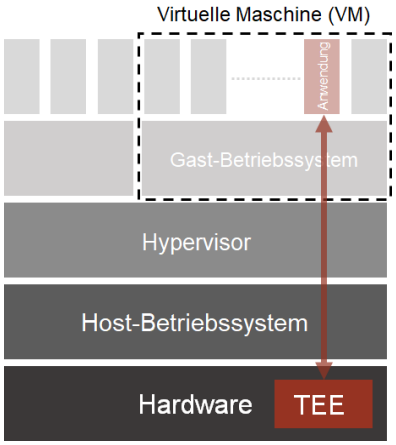

Figure 3: Executing an application with a TEE

Confidential computing shows particular promise in a cloud environment. Figure 4 shows how the scenario previously described in Figure 2 can be extended to include confidential computing. In this setup, the data processor sends data and code to the cloud provider's TEE in a way that keeps everything end-to-end encrypted, only decrypting it within the TEE itself. Since execution also occurs entirely within the TEE, the cloud provider – as the operator of the execution environment – cannot access the data, tamper with the code, or interfere with the processing. The TEE acts as a secure execution environment whose integrity can be verified by the application itself. Using remote attestation, the application can confirm that both the TEE and the code running inside it are genuine, ensuring that processing is carried out as intended by the data processor. The final result is then returned directly to the application from the TEE that performed the processing. Of course, data security on the side of the customer (the data processor in Figure 4) still depends on all components being secure; however, the number of components that need to be trusted is significantly reduced. This also shifts responsibility: since the cloud provider fundamentally has no way of accessing the data or compromising the code, they are not responsible for any potential compromise.

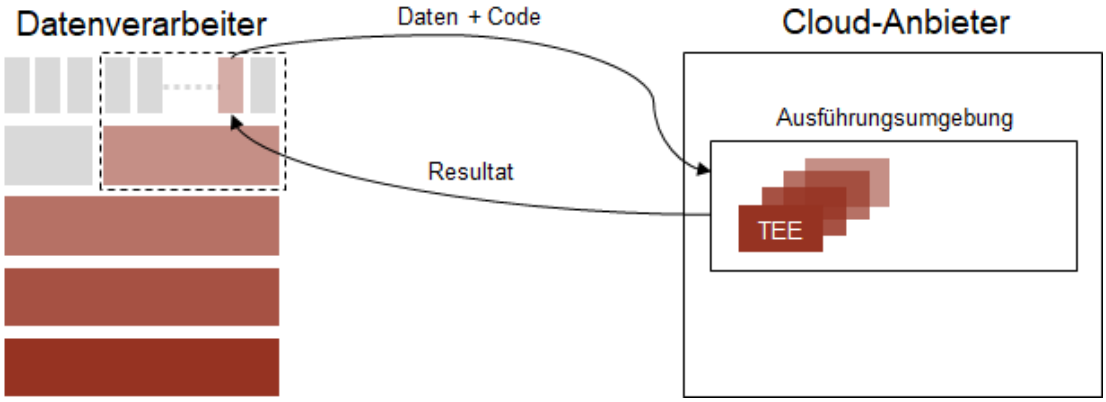

Figure 4: Executing an application using a TEE in a cloud environment

4 Market overview

Confidential computing refers to the technological approach described above, implemented in various ways by different processor manufacturers. Broadly speaking, there are two main approaches:

- A manufacturer may implement confidential computing so that individual applications, or parts of them that are critical for security (e.g. within a container), are executed within a TEE, while other applications, or parts of the same application, run outside it. The most well-known examples of this approach are Intel's Software Guard Extensions (SGX) [1] and ARM's TrustZone.
- Or, confidential computing may be implemented so that entire VMs (including all applications running on them) are executed within a TEE. Examples include Intel's TDX [2], AMD's Secure Encrypted Virtualization (SEV) and confidential computing implementations on specialised NVIDIA processors.[6, 7]

Both approaches have their advantages and disadvantages. The first approach is more selective, as only critical applications or specific parts need to run inside a TEE. The second approach is more comprehensive, as entire VMs are executed within a TEE. This makes it easier to implement in practice since applications only need to be moved rather than rewritten to run inside a TEE.

Based on today's processors that support confidential computing in one form or another, major international hyperscalers now offer a range of related services. The examples below are illustrative only – other cloud providers, such as Alibaba with its Enclave VM, also offer similar services. The market is highly dynamic and, due to inconsistent terminology, often difficult to navigate.

- Microsoft has a broad portfolio and offers a wide range of confidential computing services via its Azure Cloud which are tailored to the strengths and weaknesses of different processors.[8] A dedicated confidential computing service is also available for the Azure Kubernetes Service (AKS).
- Google also offers a diverse range of services, with *confidential VMs* playing a central role. Similar to Microsoft's approach with AKS, as part of its Google Kubernetes Engine (GKE), Google provides *Confidential GKE Nodes* and supports confidential computing on NVIDIA's specialised processors for high-performance AI/ML applications.
- Amazon Web Services (AWS) has developed Nitro Enclaves based on its proprietary Nitro System for the virtualisation and automated operation of Elastic Compute Cloud (EC2) instances. Nitro Enclaves enable security-critical components of an EC2 instance, such as a key management system, to be isolated and run in a secure 'enclave' in line with confidential computing principles.

A growing number of specialised providers and service companies are also emerging in the confidential computing space, such as Decentriq[9] and CYSEC[10] in Switzerland, and enclaive[11] in Germany, which can support specific projects and use cases.

---

[6] According to https://www.amd.com/en/developer/sev.html, SEV is available in various forms, such as SEV-ES (SEV Encrypted State) and SEV-SNP (SEV Secure Nested Paging).

[7] https://www.nvidia.com/en-us/data-center/solutions/confidential-computing/

[8] An overview is available at https://learn.microsoft.com/en-us/azure/confidential-computing/overview-azure-products

[9] https://www.decentriq.com

[10] https://www.cysec.com

[11] https://www.enclaive.io

5 Outlook

Confidential computing and TEEs offer a way to make data processing more secure, reducing reliance on the trustworthiness of cloud providers. This approach is a promising step towards solving the problem of secure remote computation and makes a valuable contribution to strengthening digital sovereignty in an increasingly cloud-based world. Alongside existing implementations (e.g. Intel SDX and TDX, ARM TrustZone, AMD SEV, and NVIDIA solutions), we can expect further technologies to enter the market and be integrated into hyperscalers' and other cloud providers' data centres. This will primarily affect Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) and, to a lesser extent, Software-as-a-Service (SaaS) solutions. When used appropriately, confidential computing can protect applications where data needs to be shielded from the cloud provider (e.g. when managing cryptographic keys or handling sensitive data; a notable example is the contact-matching process in Signal, the end-to-end encrypted messaging service).

Despite its advantages, confidential computing is not a universal solution to IT security issues in IaaS and PaaS solutions. While confidential computing reduces dependence on the trustworthiness of the cloud provider, it introduces a new dependency: the trustworthiness of the TEE vendor must also be ensured. In theory, a cloud provider could collude with a hardware supplier to compromise data processing within a TEE. However, such collusion would be extremely difficult to carry out in practice, let alone keep secret. Like any technology, confidential computing and its underlying TEEs can be vulnerable to attack. Currently, the most relevant threats involve (timing-based) side-channel attacks.[12] Many of these attacks are still theoretical and mainly serve to demonstrate what could be possible in principle. Nevertheless, the familiar cat-and-mouse dynamic of IT security is likely to play out here as well. It will be crucial to carefully analyse new types of attacks in context and assess their potential impact on real-world cloud environments. As is often the case in cybersecurity, such assessments are complex and require a high level of expertise. From a technical security perspective, however, there is no reason not to use confidential computing and TEEs. On the contrary, such solutions should be adopted, wherever possible and economically feasible, for security-critical applications.

Abbreviations

AI	Artificial Intelligence
GKE	Google Kubernetes Engine
IaaS	Infrastructure-as-a-Service
IT	Information Technology
ML	Machine Learning
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
SEV	Secure Encrypted Virtualisation
SEV-ES	SEV Encrypted State
SEV-SNP	SEV Secure Nested Paging
SGX	Software Guard Extensions
TCB	Trusted Computing Base

[12] Many known attacks against various TEEs are summarised in [3] (and in [4] exclusively attacks against Intel SGX).

TCG	Trusted Computing Group
TEE	Trusted Execution Environment
TPM	Trusted Platform Module
VM	Virtual Machine

References

[1] V. Costan and S. Devadas, *Intel SGX Explained*, IACR Cryptology ePrint Archive, Vol. 2016, No. 086, https://eprint.iacr.org/2016/086.pdf

[2] Intel, *Intel Trust Domain Extensions*, White Paper, updated in February 2023

[3] A. Muñoz, R. Ríos, R. Román and J. López, *A Survey on the (In)security of Trusted Execution Environments*, Computers & Security, Vol. 129, June 2023, 103180, https://www.sciencedirect.com/science/article/pii/S0167404823000901

[4] A. Nilsson, P. Nikbakht Bideh and J. Brorsson, *A Survey of Published Attacks on Intel SGX*, arXiv: 2006.13598, updated in June 2020, https://arxiv.org/abs/2006.13598