CONFIDENTIAL COMPUTING CONSORTIUM

Confidential Computing: Hardware-Based Trusted Execution for Applications and Data

A Publication of The Confidential Computing Consortium

November 2022, V1.3

confidentialcomputing.io

Introduction

Today, data is often encrypted at rest in storage and in transit across the network, but not while in use in memory. Additionality, the ability to protect data and code while it is in use is limited in conventional computing infrastructure. Organizations that handle sensitive data such as Personally Identifiable Information (PII), financial data, or health information need to mitigate threats that target the confidentiality and integrity of either the application or the data in system memory.

Confidential Computing protects data in use by performing computation in a hardware-based, attested Trusted Execution Environment. These secure and isolated environments prevent unauthorized access or modification of applications and data while they are in use, thereby increasing the security level of organizations that manage sensitive and regulated data.

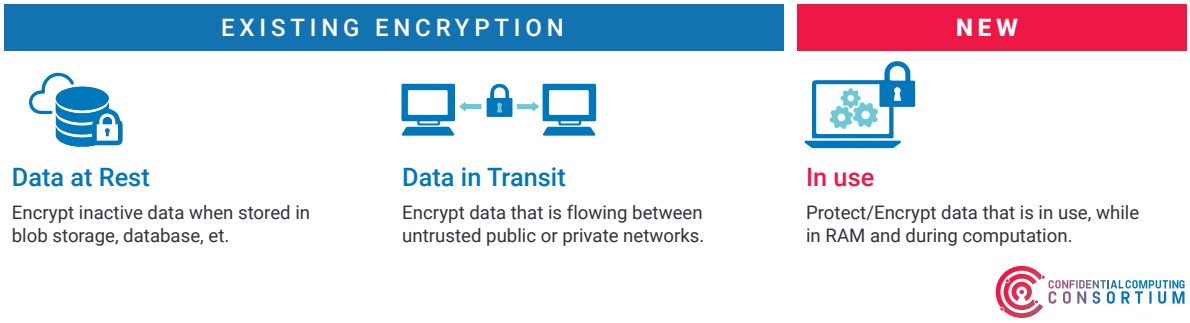
Why the need for confidential computing

States of data

In computing, data exists in three states: in transit, at rest, and in use. Data traversing the network is "in transit," data in storage is "at rest," and data being processed is "in use." In a world where we are constantly storing, consuming, and sharing sensitive data - from credit card data to medical records, from firewall configurations to our geolocation data - protecting sensitive data in all of its states is more critical than ever. Cryptography is now commonly deployed to provide both data confidentiality (stopping unauthorized viewing) and data integrity (preventing or detecting unauthorized changes). While techniques to protect data in transit and at rest are now commonly deployed, the third state - protecting data in use - is the new frontier.

Security risks for unprotected data "in use"

As threat vectors against network and storage devices are increasingly thwarted by the protections that apply to data in transit and at rest, attackers have shifted to targeting data-in-use. The industry witnessed several high-profile memory scraping, such as the **Target breach**, and CPU-side-channel attacks which have dramatically increased attention to this third state, as well as several high profile attacks involving malware injection, such as the **Triton attack** and the **Ukraine power grid attack**.



In addition, as more data is moved to the cloud, network and physical perimeter security are increasingly limited in their ability to protect against attacks. Widely studied attack patterns against cloud-based applications include hypervisor and container breakout, firmware compromise, and insider threats, each of which rely on different techniques to target code or data while it is in use. While previous safeguards (protecting data in transit and at rest) remain an essential part of a good defense-in-depth strategy, they are no longer sufficient for use cases in which sensitive data is being processed.

Increasing regulation of data processing, such as General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA), may make data custodians directly responsible when the data of users, customers, or clients is leaked due to a breach. As the cost of a data breach under the GDPR may be as high as 4% of gross annual revenue, data custodians are strongly incentivized to protect potential surface areas against attack, including data-in-use.

As more data is stored and processed on mobile, edge, and IoT devices - where processing takes place in remote and often-difficult to secure locations - the protection of data and applications during execution is increasingly important. Furthermore, due to the personal nature of information stored on mobile devices, manufacturers and mobile operating system providers need to prove that access to personal data is protected, is not observable by device vendors or third parties during sharing and processing, and that those protections meet regulatory requirements.

Even when you control all your infrastructure, protecting your most-sensitive data while in-use is part of a good defense-in-depth strategy.

How does Confidential Computing help?

Confidential Computing is the protection of data in use using hardware-based, attested Trusted Execution Environments. Through the use of Confidential Computing, we are now able to provide protections against many of the threats described in the previous section.

What are Trusted Execution Environments?

A Trusted Execution Environment (TEE) is commonly defined as an environment that provides a level of assurance of **data integrity, data confidentiality,** and **code integrity.** A hardware-based TEE uses hardware-backed techniques to provide increased security guarantees for the execution of code and protection of data within that environment.

In the context of confidential computing, unauthorized entities could include other applications on the host, the host operating system and hypervisor, system administrators, service providers, and the infrastructure owner—or anyone else with physical access to the hardware. Data confidentiality means that those unauthorized entities cannot view data while it is in use within the TEE. Data integrity is preventing unauthorized entities from altering data when data is being processed, by any entity outside the TEE. Code integrity means that the code in the TEE cannot be replaced or modified by unauthorized entities. Together, these attributes provide not only an assurance that the data is kept confidential, but also that the computations performed are actually the correct computations, allowing one to trust the results of the computation as well. This assurance is often missing in approaches that do not use a hardware-based TEE.

The following table compares a typical TEE implementation with two other emerging classes of solution that protect data in use, **Homomorphic Encryption** (HE) and **Trusted Platform Modules** (TPM).

	HW TEE	Homomorphic Encryption	Secure Element e.g., TPM
Data Integrity	Y	Y (subject to code integrity)	Keys only
Data confidentiality	Y	Y	Keys only
Code integrity	Y	No	Y
Code confidentiality	Y (may require work)	No	Y
Programmability	Y	Partial (“circuits”)	No
Unspoofability/Recoverability	Y	No	Y
Attestability	Y	No	Y

Table 1 - comparison of properties of Confidential Computing vs. HE and TPMs

In practice, the extent to which the properties are present can vary by vendor, model, and algorithm, but the first three properties highlight the key differences in security properties. For example, a typical TPM protects keys, but by itself cannot vouch for the validity of the data signed or encrypted by those keys, and it is not programmable with arbitrary code, whereas a TEE is programmable and protects that code and its data. A typical homomorphic encryption algorithm can protect arbitrary data, but by itself cannot ensure that the correct operations have been done and that the code has not been tampered with, whereas a TEE again protects both the data and the code. These techniques are often complementary and can even be used together in solutions for stronger security.

Depending on the particulars of the TEE, it may also provide:

- **Code Confidentiality:** In addition to protecting data, some TEEs may protect code while in use from being viewed by unauthorized entities. For example, this can protect an algorithm that is considered to be sensitive intellectual property.
- **Authenticated Launch:** Some TEEs may enforce authorization or authentication checks prior to launching a requested process, and may refuse to launch a process which is not authorized or authenticated.
- **Programmability:** Some TEEs may be programmed with arbitrary code, while some may only support a limited set of operations. A TEE might even include or be composed entirely of code fixed at the time of manufacture
- **Recoverability:** Some TEEs may provide a mechanism for recovery from a non-compliant or potentially-compromised state. For example, if it is determined that a firmware or software component no longer meets compliance requirements and the launch authentication mechanism fails, it may be possible to update that component and retry (recover) the launch. Recoverability generally requires that some component(s) of the TEE remain trusted, which can act as a “root” when other components are updated.

- **Attestability:** Often, a TEE can provide evidence or measurements of its origin and current state, so that the evidence can be verified by another party and—programmatically or manually—it can decide whether to trust code running in the TEE. It is typically important that such evidence is signed by hardware that can be vouched for by a manufacturer, so that the party checking the evidence has some assurance that it was not generated by malware or other unauthorized parties.

Availability attacks (such as DoS or DDoS) attacks are not addressed by current hardware-based TEEs as part of their threat model. Software projects and service providers may provide mitigations to such attacks.

Why is Hardware Necessary for Confidential Computing

Security is only as strong as the layers below it, since security in any layer of the compute stack could potentially be circumvented by a breach at an underlying layer. This drives the need for security solutions at the lowest layers possible, down to the silicon components of the hardware. By providing security though the lowest layers of hardware, with a minimum of dependencies, it is possible to remove the operating system and device driver vendors, platform and peripheral vendors, and service providers and their admins, from the list of required trusted parties, thereby reducing exposure to potential compromise.

With the goal of decreasing the reliance on proprietary software for confidential computing environments, the Confidential Computing Consortium has excluded from its scope TEEs that have only software roots of trust, and focused on hardware-based security guarantees for confidential computing environments.

Utilization Paradigms within Confidential Computing

There are multiple ways hardware-based TEEs are being used today to deliver the efficient defense-in-depth mechanisms and security boundaries sought by confidential computing. The approaches vary in the size of their Trusted Computing Base (TCB), which is a rough measurement of how many lines of code a user needs to trust, and how applications use the corresponding TEE.

Users should understand the differences between the two common approaches and how they meet the qualities of confidential computing, so they can make the corresponding selection that supports their use case.

Approach #1: Application SDKs

Here the developer becomes responsible for identifying and possibly partitioning their application's code into trusted and untrusted components. How they perform such an activity may be influenced by whether the application is written for one hardware-TEE, or whether TEE-specifics are abstracted into a common programming model by a software development kit which allows for a degree of portability across hardware-TEEs. This approach may allow for closer scrutiny of the code that is run in the TEE and the interfaces it offers, since there may be less code to review than in Paradigm #2, but may require applications to be designed (or changed) to use a TEE-aware SDK.

Approach #2: Run-time Deployment Systems

This approach involves minimizing the effort needed to convert a typical application workload into one that can run in a TEE. Parallel efforts exist to allow the development of cross-TEE portable applications, while other efforts are underway to deploy unmodified applications into TEEs. Deploying full applications has its advantages and draw-backs: on the one hand it reduces the cost to deploy applications that gain additional isolation, however the original applications were most likely not designed to take advantage of the other features of of TEEs such as attestation and secret protection. These advantages may be foregone in the interest of ease-of-use, or may be handled by the run-time deployment system or via another mechanism.

Use Cases for Confidential Computing?

Keys, Secrets, Credentials and Tokens Storage and Processing

Cryptographic keys, secrets, credentials and tokens are the "keys to the kingdom" for organizations responsible for protecting sensitive data. Historically, the storage and processing of these critical information assets required an on-premises hardware security module (HSM) that complied with local national security standards, such as the US Federal Information Processing Standard (FIPS 140-2, 140-3) security requirements for cryptographic modules. The proprietary nature of traditional HSM hardware increases their cost, limits their scalability, and presents cost and compatibility challenges for deployment in cloud and edge computing environments.

Confidential Computing is already in use, both by independent software vendors (ISVs) and within large organizations, to store and process cryptographic and secret information using standardized compute infrastructure available on-premises, in public/hybrid clouds, and even at the network edge for IoT use cases. Key management applications store and process cryptographic keys, secrets, and tokens inside a secure, hardware-based TEE, providing data confidentiality, data integrity, and code integrity to achieve security comparable to a traditional HSM.

Public Cloud Use Cases

In a traditional public cloud scenario, trust must be placed in many layers within the cloud provider: the hardware; the firmware for core and peripheral devices; the host operating system, the hypervisor, and the cloud provider's orchestration system itself. While public cloud providers obviously go to great lengths to secure all layers of this stack, Confidential Computing offers additional protection guarantees and significantly reduces the number of layers that must be trusted by the end user.

With a hardware-based TEE protecting applications and data in use, it becomes significantly more difficult for an unauthorized actor—even one with physical access to the hardware, root access to the host OS or the hypervisor, or privileged access to the orchestration system—to gain access to the protected application code and data. Confidential Computing aims to allow the removal of even

the cloud provider from the Trusted Computing Base, so that only the hardware and the protected application itself would be within the attack boundary.

This enables many workloads to move to the public cloud which previously could not due to security concerns or compliance requirements.

Multi-Party Computing

New compute paradigms are emerging to enable datasets and processing power to be shared across multiple parties. However, the data and computational models may be sensitive or regulated, such as in the financial services, healthcare, government and non-profit domains. So how can the confidentiality and integrity of the data be preserved even when shared across platforms that may not be trusted by the other transacting parties? Organizations want to be freed from the limitations of data silos, but still must ensure the data sources are not compromised when they are shared and the results can only be accessed by the parties approved for sharing.

For example, private multi-party analytics can be applied to any case in which multiple parties have private data that needs to be combined and analyzed without exposing the underlying data or machine learning models to any of the other parties. The technology could be applied to preventing fraud in financial services, detecting or developing cures for diseases in the healthcare industry, or gaining business insight. As an illustration, multiple hospitals could combine data to train a machine learning model to more accurately detect brain tumors using radiology information. But individual patient data would be kept confidential, even in the event of a breach.

Utilizing Confidential Computing, organizations can now ensure that data on remote systems is protected against tampering and compromise, including against insider threats within the partnering organizations, and can also validate the integrity of the code processing that data. The data can be combined and analyzed within the TEE, and the results can be sent in an encrypted format back to each party. Data remains protected throughout the entire process: while it is transferred, during computation, and while stored.

These capabilities will help drive the advancement of a global data-sharing playing field, such as the technologies mentioned above, that enables organizations to unlock previously unleveraged data sets for collaborative analysis and exchange with other organizations – with reduced risk of security, privacy, and regulatory impacts.

Blockchain

A blockchain is a shared, immutable ledger that records the exchange of data, digital assets, or currency between a network of participants. Blockchains provide the infrastructure to record

and validate the transactions without the need for a centralized 3rd party. Blockchains can provide transparency into supply chain activities, facilitate the exchange of digital assets, or underpin compliance processes such as Know Your Customer (KYC). A key characteristic of a blockchain is that it ensures that participants who should have a piece of data in common know for sure that they see the same thing, and that once entered on the blockchain data is immutable. It is up to the application developer to ensure that sensitive data such as PII is not stored on the immutable blockchain.

Confidential Computing can be used to enhance the implementation of a Blockchain-based system. By combining the power of Confidential Computing and Blockchain technologies, users may leverage a hardware-based TEE to provide attestation and verification services that optimize for scalability, privacy and security. The assurance of data consistency between users of a blockchain usually depends on each party having independently validated all historical data upon which the validity of any current data depends. This requires visibility of those historical data sets, a potential scalability or privacy concern. Users can execute smart contracts in a hardware-based TEE instead of independently accessing and validating historical data and associated smart contracts for themselves. Once the transaction is finalized, TEEs provide attestation services to prove the reliability of the transaction, meaning no subsequent participant needs to perform the verification again for themselves. TEE-based attestation services can also help address some of the computational and communication inefficiencies that arise from consensus protocols [5].

Mobile And Personal Computing Devices

Use cases on client devices mostly involve application developers or mobile device manufacturers who provide assurances that personal data is not observable during sharing or processing. From a compliance and best practice perspective, this removes equipment manufacturers from the liability loop as they can assert that they can not observe a client's personal data. Where a TEE is able to ensure the attestability and code integrity properties mentioned before, functional correctness of computation (and hence trustworthiness of the result of the computation) can be formally proven[1][2]. This provides an environment where application developers can prove to the user that their data has not left the device.

For example, in an implementation of continuous authentication (always logged on), the user account login application on the device uses signals that can be gathered from user interaction to identify the user. These may include sensitive data like biometrics or physical patterns in how the user interacts with the device that would benefit from operating within a TEE. The user behavior engine only needs to identify the user, without exposing the raw user behavioral data to other devices or to code outside the TEE.

Another example is the case of decentralized on-device model training, where the desire is to improve a model, and allow such improvements to be shared with other devices, without leaking the data used to train the model. Design of protocols using hardware-based confidential computing are easier to use than statistical models like Differential Privacy[3]. Furthermore, statistical methods still rely on the consumer to trust the application developer to inject the right amount of noise to adequately hide sensitive data. By contrast, an on-device HW TEE could enable users to set policy and constraints on their own devices for the usage and processing of their data through the property of mutual attestation.

Edge And IoT Use Cases

Cases such as local search and filtering in the context of (for example) DDoS detection taking place in a home router are examples where a confidential computing environment would fit well. In most cases, TCP/IP packet metadata needs to be kept confidential as sensitive user behaviour may be inferred. Other examples include Edge confidential Machine Learning processing, like video-metadata generation for reducing latency and/or bandwidth to a backend network; CCTV camera surveillance, where the provider needs to load templates of persons of interest that could be harmful if leaked; or on-device training models similar to those described above in the mobile context.

Some devices are, by their design, physically accessible to untrusted parties. Confidential computing technology can be used to help mitigate attacks which depend on physical access to the device.

Point of Sale devices / payment processing

Use of hardware-based, attested Trusted Execution Environments is already common in the payment processing industry today. Point-of-Sale devices that accept chip-and-PIN style credit or debit cards are used to protect secrets such as credit card numbers, PINs, expiration dates, CVVs, and key material used for protecting messages. Often, other devices such as cash registers or tablets are general purpose computing devices, and chip-reading hardware is used to keep the payment processing code protected from any potential malware on such general purpose computing devices.

To protect user-entered information such as a PIN, it is important to also protect the means of data entry, so that such data cannot be read or tampered with as the user enters it. In secure devices, the number pad is isolated such that the input is only readable by code within the hardware-based TEE. In this manner, the data can be safely operated on and used to generate encrypted messages to a payment processing service, all out of reach of any malware or unauthorized access by third parties.

Conclusion

The Confidential Computing landscape is rapidly evolving to provide new tools to business and end-users that protect sensitive data and code against a class of threats occurring during data execution which were previously difficult, if not impossible, to protect against.

Solution providers have developed different approaches for confidential computing by making trade-offs, for example around TCB size, ranging from partitioning the application's code into trusted and untrusted components, to enabling the migration of existing applications with few or no changes.

These different approaches support various use cases, but all with the end goal of helping ensure the confidentiality of the sensitive, business critical information and workloads. As confidential computing continues to evolve, more approaches may emerge, or evolutions of these approaches may occur. The Confidential Computing Consortium is optimistic about the innovation that lays before this field.

About the Confidential Computing Consortium

To learn more check out our in depth technical analysis »

The Confidential Computing Consortium (CCC) is a community focused on projects securing data in use using hardware-based TEEs, and accelerating the adoption of confidential computing through open collaboration. The CCC brings together hardware vendors, cloud providers, and software developers to accelerate the adoption of Trusted Execution Environment (TEE) technologies and standards.

It is not the intent of this whitepaper, or the CCC, to evaluate or compare specific vendor technologies. This whitepaper aims to set context and define consistent terminology that vendors can use to describe their products, allowing others to make a like-to-like comparison between the various offerings in this space.

This content represents the collaborative work ("Work") of the Confidential Computing Consortium ("CCC"). The CCC is solely responsible for the Work and Individual CCC members may not have contributed or participated, or may not endorse it.

Further Reading

- **Confidential Computing Consortium**
- **Draft NIST paper**
- **UN Handbook on privacy preserving techniques**

References

[1] "Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims" Report by Miles Brundage et al., April 2020.
https://arxiv.org/pdf/2004.07213.pdf

[2] "Remote Credential Management with Mutual Attestation for Trusted Execution Environments" Carlton Shepherd, Raja N. Akram, and Konstantinos Markantonakis. 12th IFIP WG 11.2 Intl. Conference, WISTP 2018, Brussels, Belgium, Dec. 10-11, 2018.
https://arxiv.org/pdf/1804.10707.pdf

[3] The Algorithmic Foundations of Differential Privacy" Cynthia Dwork, Aaron Roth. Foundations and Trends in Theoretical Computer Science Vol. 9, Nos. 3–4 (2014) 211–407.
https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf

[4] "Differential Privacy" Cynthia Dwork, ICALP 2006: Automata, Languages and Programming pp 1-12.
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dwork.pdf

[5] Veronese, Giuliana & Correia, Miguel & Bessani, Alysson & Lung, Lau & Veríssimo, Paulo. (2013). Efficient Byzantine Fault-Tolerance. Computers, IEEE Transactions on. 62. 16-30. 10.1109/TC.2011.221.
https://www.gsd.inesc-id.pt/~mpc/pubs/tc13-minimal.pdf

CONFIDENTIAL COMPUTING
CONSORTIUM

The Confidential Computing Consortium (CCC) is a community focused on projects securing data in use using hardware-based TEEs, and accelerating the adoption of confidential computing through open collaboration. The CCC brings together hardware vendors, cloud providers, and software developers to accelerate the adoption of Trusted Execution Environment (TEE) technologies and standards.

g.io