

Disclaimer: The goal of this post is not to give an overview of cloud attestations or discussing their usage in traditional industry sectors. Rather its focus is solely on using cloud attestations while engineering TEEs for blockchain applications - where trust by third parties is required. Another blog post in this series will dive deeper into the 3rd party trust problem.

Required Reading: [DCAP](#), TPM ([1](#), [2](#)), [MAA](#)

Non-scope: Other cloud providers, SEV-SNP

TLDR

Cloud attestations provide stronger assurances against physical attacks compared to traditional remote attestation. However, they rely on trusting the cloud provider's proprietary attestation services, which are opaque "black boxes" that cannot be audited.

There are approaches to combine cloud and remote attestation for defense-in-depth, but ultimately, even with cloud attestations, we still have to trust the cloud provider to some extent in not tampering with the hardware itself.

Trusting the Cloud Service Provider (CSP) or anyone else to not tamper with the hardware remains an open problem that this approach does not solve. We need continued research and development to strengthen hardware trust models for TEEs.

Remote Attestation Overview

Remote attestation allows a third party to verify that a particular software is running on an untampered platform. At a high level, the CPU measures the Trusted Computing Base (i.e. boot firmware, kernel, and application binaries) into secure registers. It signs this measurement log using a private key which is embedded in the CPU chip, producing an "attestation report" that can be verified by a remote party.

Cloud vs Remote Attestation


Microsoft's Managed Attestation (MAA) service is different from traditional remote attestation. With remote attestation, you verify the SGX enclave directly. With MAA, you verify a signature from the cloud provider stating that the target enclave is running legitimately within their cloud.

Advantages of Cloud Attestation

The key advantage of cloud attestation is "Data Center Execution Assurance". Which provides a statement that the TEE is running in the legitimate cloud, rather than a side-channel attack lab using hardware exploits to extract secrets.

Traditional TEE solutions like SGX and TDX do not consider physical attacks in their threat model. Cloud attestations mitigate the risk of such attacks by ensuring the TEE is within the cloud provider's physically protected environment.

[

Screenshot 2024-06-06 at 12-17-46 henry  on X it's underappreciated that the guy with a glock (gender neutral) is actually a key part of the sgx security model <https://t.co/jXT7j5Rxqa> _ X

1196x803 150 KB

](<https://collective.flashbots.net/uploads/default/original/2X/1/121fc6876c5bf4293ec6a9d7e70622dca9ee49b1.jpeg>)

[Source - X](#)

Drawbacks of Cloud Attestation

Lack of Direct Attestation

With Microsoft's MAA cloud attestation, we are not directly attesting the TEE itself any more. We don't have full visibility into what is happening under the hood.

We are trusting a signature from Microsoft's attestation service that presumably attests the target TEE. However, this attestation service is proprietary, so we are effectively trusting Microsoft's integrity without being able to verify the internals.

Proprietary Nature

Most cloud attestation solutions today are proprietary "black boxes" provided by the respective cloud vendors. We have to blindly trust that their implementation is secure and follows best practices, without being able to audit the source code.

This increases the attack surface and could potentially introduce new vulnerabilities that are not present in the standard remote attestation flow.

Combining Cloud and Remote Attestation

Some projects like Gramine allow combining cloud attestation with direct remote attestation of the TEE.

With [Gramine and SGX](#), you can first verify the Microsoft cloud attestation, and then also independently verify the SGX remote attestation report directly from the enclave. This provides defense-in-depth by leveraging both attestation mechanisms.

MAA Cloud Attestation for TDX

A note before we dig into this: TDX on Azure is in preview, this means that it was difficult to gather official and factual information about Azures attestation approach. It also means the information gathered here is likely going to change in the long run. Aside from tinkering with azures TDX VMs and the best resources I could find is [a\) Azure describing its approach to confidential VMs](#) and [b\) a library implementing the attestation process in rust](#)

For TDX VMs on Azure, Microsoft uses [nested virtualization](#) to run a proprietary vTPM paravisor within the Trust Domain (TD). This paravisor boots the customer TDX VM and provides a TPM2 interface to measure the boot process.

The TDX attestation report references the paravisor's MRTD (Measurement Root Trust Domain). However, since Microsoft does not provide the source code, the paravisor's integrity cannot be externally verified.

The attestation report also contains the vTPM's signing key. If we trust the attestation report, we can then verify the signed PCR measurements from the vTPM.

Unlike the SGX approach, this TDX design requires fully trusting Azure, as there is currently no way to independently verify the boot measurements.

What's missing for Azure TDX?

While the current Azure TDX attestation model provides data center execution assurance by leveraging the proprietary paravisor, this same paravisor implementation makes it impossible to run TDX VMs on Azure without full trust in this proprietary implementation. Thus it would make sense to provide a Vanilla TDX implementation on Azure, where the plain TDX attestation report can be verified by MAA

in the same way as it currently works with SGX. If this is not possible, these are the missing pieces that limit the ability to fully verify trust:

Lack of Open Source Implementation

The paravisor code that measures the TDX VM boot process and provides the vTPM interface is closed-source and proprietary to Microsoft. This means there is no way for external parties to audit the implementation and ensure it follows security best practices. An open source paravisor implementation would allow the security community to inspect and harden it.

No Independent Boot Measurements

The attestation report includes measurements from the paravisor's root of trust. However, there is currently no way to independently verify these boot measurements from outside the Microsoft cloud. Providing a mechanism to allow verification of the boot measurements by a trusted third party would increase transparency.

Limited Auditability

Since the full attestation process is a "black box" controlled by Microsoft, there is limited auditability into what is actually being measured and attested. Increased openness in the attestation service's internals is needed to build more confidence.

Single Cloud Provider Dependency

The current approach ties attestation directly to Microsoft's Azure cloud. For multi-cloud or hybrid deployments, a more cloud-agnostic "Data Center Execution Assurance" primitive would be beneficial to avoid single provider lock-in.

Future work

- Check alternative cloud attestation implementations from other cloud or hosting providers, in particular Google Cloud.

- Check whether the MAA service can be tricked to sign SGX remote attestations not originating from Azure cloud
- Request cloud providers, in particular Azure, to make their attestation process fully auditable with reproducible builds
- Research and develop open source, auditable “Data Center Execution Assurance” toolchains that any hosting provider can setup.
- Proof of Cloud: even if a cloud provider doesn’t provide an attestation, we could still provide a way to determine that a given enclave is run by a cloud operator, even if they don’t support it. For example, suppose an SGX enclave running on CloudA completes a round trip TLS challenge to <https://CloudA.com/> and we use the instruction cycle counter to prove it completed so quickly that it had to have come from within the same data center that has a CloudA TLS private key. This would provide some TEE-based evidence associating the enclave with a cloud location, even if they don’t support it.