Certifier Nvidia H100 GPU Confidential Computing Support Design

Thursday, September 14, 2023

Nvidia H100 GPUs extends existing CPU TEEs (e.g., SEV SNP VMs or Intel TDX VMs) to include code and data on the GPU. The GPU attestation report does not include measurement of the intended GPU user workload (e.g., ML model). It only contains the measurement of the GPU firmware. As a result of this design, it makes little sense for the Certifier to treat the GPU as a normal backend in addition to SEV-SNP and SGX, etc. Instead, we should extend the existing platform evidence to include GPU attestation report when H100 GPUs are present.

However, in the case of standalone GPU hardware/firmware attestation, the Certifier can be engineered as a drop-in replacement of the Nvidia Remote Attestation Service. This helps with specific use cases where the Nvidia service is not trusted. We will briefly describe this scenario at the end of this document.

Overview of CPU TEE Attestation with GPU Extension

A high-level architecture of CPU and GPU attestation with the Certifier is shown in Figure 1:

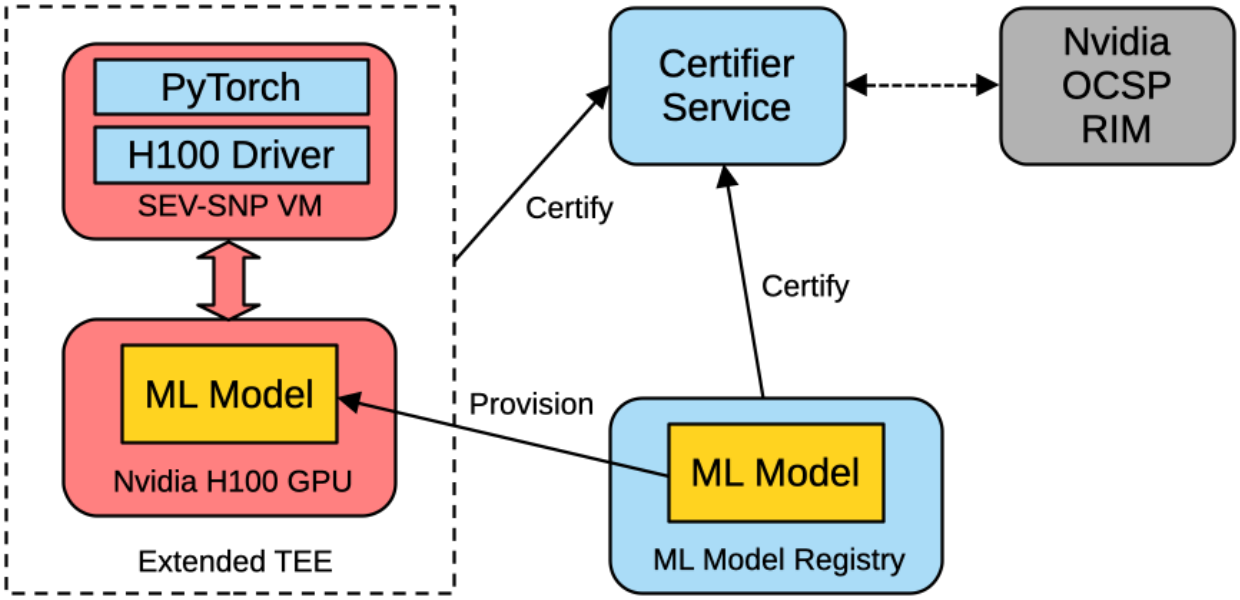

Figure 1: H100 GPU attestation with Certifier

The complete "Extended TEE" including both the CPU and GPU should be certified by the Certifier Service in its policy domain. Currently, H100 only supports SEV-SNP as the CPU TEE. Hence, the Certifier Service will check both the SEV-SNP attestation report and the H100 GPU attestation report to verify the extended TEE integrity and authenticity. The PyTorch application in the given example will be measured as part of the SEV-SNP VM attestation. However, the ML

model to be deployed onto the GPU is not included in the GPU attestation. We instead verify that the Nvidia H100 GPU attached to the SEV-SNP VM is authentic with the correct firmware and trust the Nvidia driver in the guest (measured when we carry out the SNP attestation) to establish the secure communication channel with the GPU hardware. Later the confidential ML model will be transferred securely from an ML model registry to the H100 GPU indirectly through the CPU TEE (SEV-SNP VM). Notice that in practice, the ML model registry does not have to provision the whole ML model, which can be very large for LLM. An encrypted model can be embedded in the SEV-SNP VM image and the decryption key be provisioned later by the registry instead.

Nvidia OCSP and RIM provide utilities for retrieving trusted measurements of firmware, CRL, and Nvidia Certificate chain verification. The Certifier can contact them to carry out the attestation verification. We are still investigating how much can be done locally and reasonably by the Certifier Service without contacting any Nvidia services.

Potential Security Issues

A major issue today is that the secure communication channel between the CPU TEE and the GPU is established by the Nvidia driver in the guest. So far, we are not able to confirm whether this secure channel can be established by third party instead. Based on the Nvidia guest software stack architecture, this seems less likely to be true. However, we do measure and trust the Nvidia guest driver software stack through the CPU TEE attestation. Once the GPU attestation is verified, we can simply treat it as part of the CPU TEE. The following scenarios can create some complications but should be resolvable:

1. What if there are multiple GPUs assigned to a single SEV-SNP VM and only one of them is Nvidia H100?
    a. We measure and trust the CPU TEE application (e.g., PyTorch). It should be feasible to test and identify the GPU device before we deploy the model to the GPU. In the extreme case, the application can bail if there are non-H100 GPUs present.
2. What if the VM reboots and an H100 with older/compromised firmware is swapped in instead?
    a. The Nvidia driver is supposed to carry out the attestation during secure channel establishment. Additionally, the secure channel establishment should be per-boot. It is not a persistent channel. So, the driver should be able to catch this case through its communication with the OCSP and RIM.
    b. Even if the above is not the case, we can force a re-certification against the Certifier Service when the platform evidence includes GPU attestation report.

An additional question is that if 2.a is true, why do we even need to perform the GPU attestation in the Certifier framework? Firstly, we need to make sure the Nvidia driver is carrying out the attestation. Does it happen for each boot? Does it contact OCSP and RIM every time? Secondly, even if this is the case, Certifier can still serve as a third-party alternative where GPU

hardware properties can be independently verified along with its authenticity. More on this later in the document.

H100 Attestation Process

This is how H100 attestation is performed by the local GPU verifier which is part of the NVIDIA Attestation SDK. We can use this as a reference when adding support for H100 in the Certifier Framework.

1. Generate a random 32 byte nonce
2. Obtain the driver version with an NVML call
3. Obtain the VBIOS version with an NVML call
4. Obtain the attestation certificate chain with an NVML call. The certificate chain always contains 5 certificates, ending with a self-signed cert.
5. Verify the attestation certificate chain using a device root certificate.
6. Validate the attestation certificate chain with OCSP
7. Obtain attestation report with an NVML call using the nonce from step1
8. Verify the attestation report
    a. it must be signed with the leaf cert from the attestation certificate chain
    b. it must contain the same nonce as in step1
    c. it must contain the same driver and VBIOS versions as in step2 and step3
9. Obtain driver RIM
10. Verify driver RIM cert chain using a trusted root certificate and then with OCSP
11. Obtain VBIOS RIM
12. Verify VBIOS RIM cert chain using a trusted root certificate and then with OCSP
13. Verify the measurements from the attestation report with the "golden" RIM measurements

Certifier Evidence Package and Policy for H100 GPU VM

We should come up with a new “submitted_evidence_type” for the “trust_request_message” and a new “enclave_type”. Since we are essentially dealing with a “new” extended TEE, this enclave can have “h100-sev-enclave” type. The corresponding evidence type should be “h100-sev-evidence”. In addition to the existing SEV SNP evidence, we will add another set of H100 evidence into the “h100-sev-evidence” evidence package (We can add another layer of indirection in the evidence package structure such as an “evidence group” to make this more flexible in the future when “h100-tdx-evidence” becomes available.).

The added H100 evidence would include:

1. Nvidia Certificate Chain for attestation verification.
2. Nvidia Hopper H100 GPU Hardware Attestation report.
    a. Nonce (user data) should be “public authentication key” (The same as the SEV-SNP user data).

Figure 4-1. The Certificate Chain

The Certifier Service Policy should also be augmented with:

1. Trusted Nvidia Hopper H100 measurements (from RIM).
2. Nvidia root certificate.
3. Nvidia H100 platform properties.

The trusted H100 firmware measurements can be retrieved from the Nvidia RIM service. We could allow Certifier users to specify individual trusted measurements orthogonal to the RIM. However, we might consider querying the RIM only if the user simply wants to trust whatever RIM deems as legitimate measurements.

H100 platform properties should contain at least (both are in the attestation report):

1. Driver version
2. VBIOS version

The Nvidia attestation SDK also supports attestation verification policies. Some of these can also be introduced into Certifier policies in the future:

```
{
    "version":"1.0",
    "authorization-rules":{
      "x-nv-gpu-available":true,
      "x-nv-gpu-attestation-report-available":true,
      "x-nv-gpu-info-fetched":true,
      "x-nv-gpu-arch-check":true,
      "x-nv-gpu-root-cert-available":true,
      "x-nv-gpu-cert-chain-verified":true,
      "x-nv-gpu-ocsp-cert-chain-verified":true,
      "x-nv-gpu-ocsp-signature-verified":true,
      "x-nv-gpu-cert-ocsp-nonce-match":true,
      "x-nv-gpu-cert-check-complete":true,
```

```
    "x-nv-gpu-measurement-available":true,
    "x-nv-gpu-attestation-report-parsed":true,
    "x-nv-gpu-nonce-match":true,
    "x-nv-gpu-attestation-report-driver-version-match":true,
    "x-nv-gpu-attestation-report-vbios-version-match":true,
    "x-nv-gpu-attestation-report-verified":true,
    "x-nv-gpu-driver-rim-schema-fetched":true,
    "x-nv-gpu-driver-rim-schema-validated":true,
    "x-nv-gpu-driver-rim-cert-extracted":true,
    "x-nv-gpu-driver-rim-signature-verified":true,
    "x-nv-gpu-driver-rim-driver-measurements-available":true,
    "x-nv-gpu-driver-vbios-rim-fetched":true,
    "x-nv-gpu-vbios-rim-schema-validated":true,
    "x-nv-gpu-vbios-rim-cert-extracted":true,
    "x-nv-gpu-vbios-rim-signature-verified":true,
    "x-nv-gpu-vbios-rim-driver-measurements-available":true,
    "x-nv-gpu-vbios-index-conflict":true,
    "x-nv-gpu-measurements-match":true
  }
}
```


To respond to a certification request of type “h100-sev-evidence”, the Certifier Service should verify both the SEV-SNP and the H100 evidence against the policies. The certification request can only be deemed successful if both passes. Otherwise, the request should fail.

The Certifier Service as an Alternative to the Nvidia Remote Attestation Service

In addition to the above architecture, an alternative (or parallel) role the Certifier Service can play is to act as a drop-in replacement for the Nvidia Remote Attestation Service as shown in Figure 2. In this specific case, we can consider verifying just the GPU attestation report and platform properties without the CPU TEE. This is useful when some use cases specifically choose to not trust the Nvidia Remote Attestation Service and would like a third-party open-source on-prem solution instead.

However, in most cases, we imagine that such a use case would simply leverage more features from the Certifier and directly certify an extended TEE including the GPU as described earlier. This mode seems to only apply to existing applications that specifically depend on Nvidia attestation SDK but would like to have an alternative remote or on-prem service for verification.

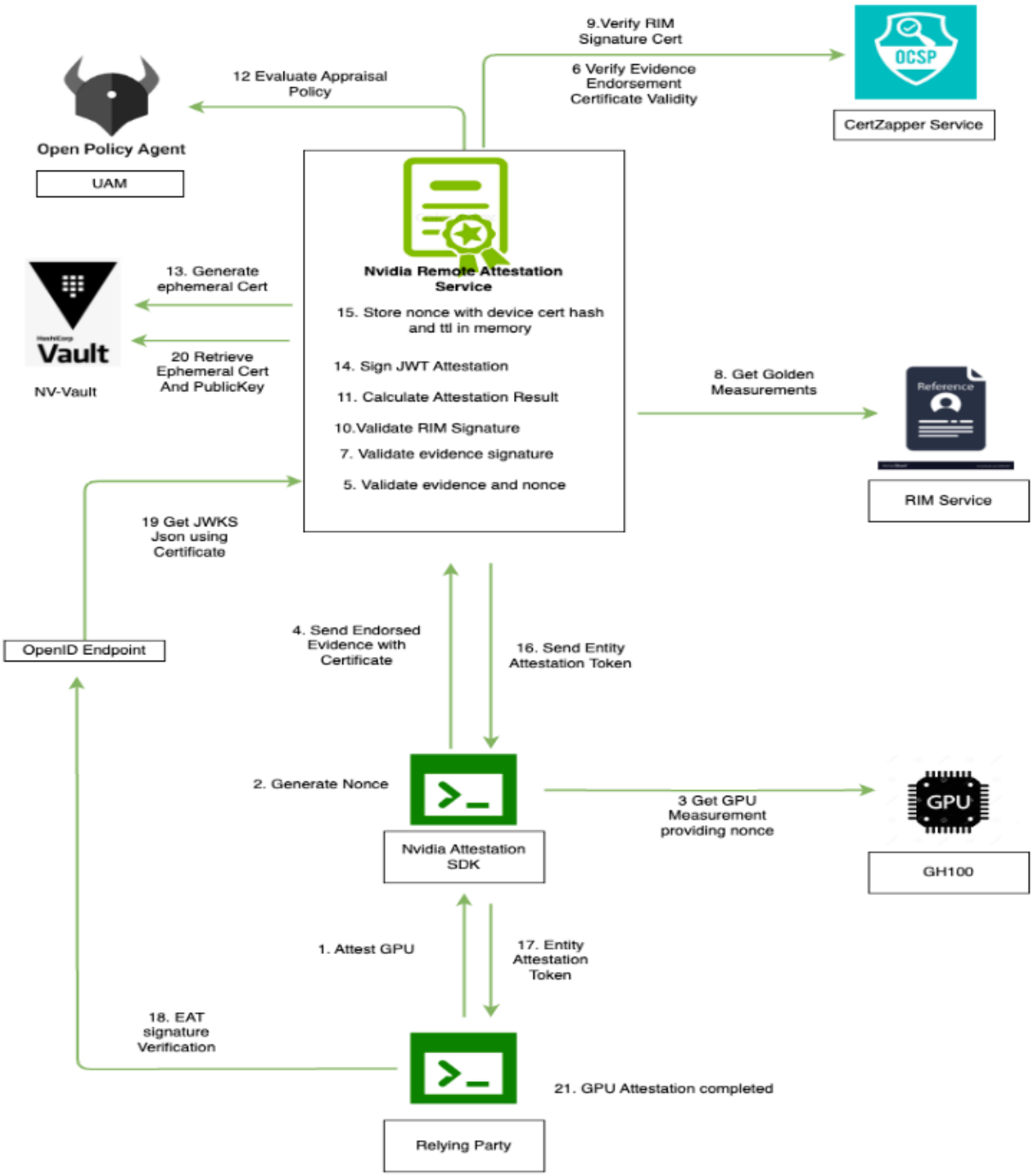

Figure 2: Nvidia Remote Attestation Service