Poster: Flexible Remote Attestation System for Trusted Execution Environment using Delegating Verification

Takashi Yagawa
University of Tsukuba
yagawa@osss.cs.tsukuba.ac.jp

Tadanori Teruya
AIST
tadanori.teruya@aist.go.jp

Kuniyasu Suzaki
Institute of Information Security
suzaki@iisec.ac.jp

Hirotake Abe
University of Tsukuba
habe@cs.tsukuba.ac.jp

***Abstract*—The status of the Trusted Execution Environment (TEE) on the cloud platform can be verified remotely using Remote Attestation (RA), which uses dedicated services and tools to verify the status based on the evidence provided by the cloud platform. RA uses a dedicated service or tool to verify the status based on the evidence presented by the cloud platform. However, it is difficult for the current verification environment to satisfy the increase in RA and high responsiveness due to the rise in IoT devices and the expansion of microservices that will come shortly. To solve this problem, we proposed a method for safely deploying verification services called Delegating Verification in a previous paper. In this study, we implement a more practical verification service using this method and show that the measuring overhead and load balance results. The evaluation shows the effectiveness of our proposal, with only a small additional overhead and load balancing effectiveness.**

I. Introduction

In recent years, Trusted Execution Environment (TEE) has gained attention as a means to process data on cloud services confidentially. While users do not have physical access to the cloud platform, the status of TEE and the software running on it can be verified through Remote Attestation (RA), which remotely verifies the authenticity and integrity of the platform and programs. RA is a method of remotely verifying the authenticity and integrity of a platform or program. It is completed by verifying the evidence generated using the vendor's or others' information.

As the application of TEE progresses, the number of RA requests increases. For example, in edge computing, many IoTs request RA to TEE on a cloud platform. Also, in Function as a Service, each small program in a simple design requires attestation. Furthermore, real-time performance is also necessary in these cases.

However, RA verification for TEE does not currently consider such issues. Online verification services must be operated by a limited number of trusted organizations, but the concentration of requests on such a small number of verification servers can lead to response delays or even non-acceptance [1], [2]. Users can also build their own verification services, which is very costly for most users who prefer cloud services [3].

We previously proposed a solution to this problem: Delegating Verification that uses TEE to securely deploy verification services on servers managed by a third party[4]. This would allow anyone to build verification services and use them on whatever anyone else builds, thus enabling low-latency and scalable RA through load balancing. However, in the previous prototype, the delegation process is a simulation. This study evaluated a real-world delegated verifier that implements the delegation process. The evaluation showed a small enough overhead to be practical and the effectiveness of geographic scaling.

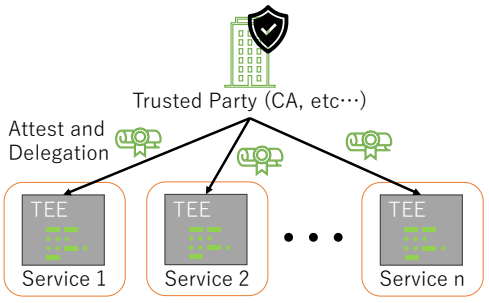


Fig. 1. In delegating verification, a trusted party attests to the verification service operated by a third party, and if there are no problems, issue the delegating certificate as proof of delegation.

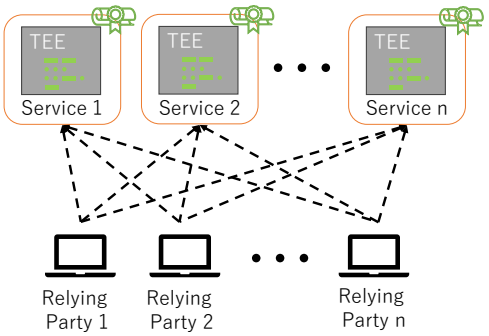


Fig. 2. The dashed line indicates a request for verification of evidence. The relying party can have the evidence verified securely by any delegated verification service.

II. Design

Figure 1 is a schematic diagram of the delegating verification. The verification program runs in the TEE on the

server, and its validity is guaranteed. Moreover, delegating verification, our proposed method, guarantees that a third-party-managed server performs trustworthy verification. First, a trusted party, such as a certification authority, performs RA to the TEE on the third-party server and identifies the verification program. The trusted party issues a delegating certificate within the RA-targeted TEE if it is a legitimate verification program. Because TEE can guarantee the authenticity and integrity of the verification program and certificate, verification services can expand without increasing the number of trusted parties.

Figure 2 shows that relying parties use delegated verification services, which are third-party servers with a delegating certificate. For verification requests from relying parties, delegated verification services can verify evidence received from other TEE platforms. A delegating certificate allows relying parties to confirm legitimate delegation. Specifically, a relying party receives a signed response and verifies the signature using the delegating certificate. Relying parties can improve RA response times by selecting delegated verification services that are available or close by.

III. Experiment

We modified Quote Verification Service (QVS) [5], an open-source verification service for SGX provided by Intel, and implemented a prototype by protecting it with gramine [6]. QVS is a REST API server, and verification is performed by sending evidence using the POST method. Gramine is a tool that allows you to apply SGX without modifying the code.

Delegating verification is implemented based on gramine's RA, which incorporates the RA flow into mbedTLS. First, the trusted party service executes RA against the third-party server. If it succeeds, the third-party service sends a certificate signing request (CSR) based on the internally generated asymmetric key. Finally, the trusted party service signs the CSR to generate a delegating certificate and send it to the third-party server.

The evaluation platform is Standard_DC1ds_v3 with Ubuntu 22.04 OS in Microsoft Azure, which has one core, 8GB RAM, 4GiB RAM for TEE.

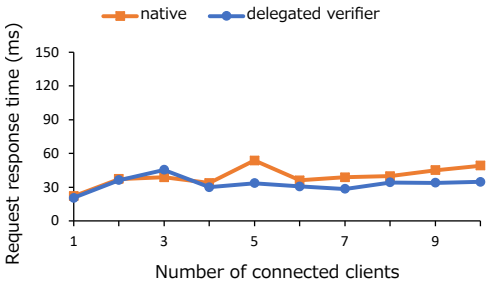


Fig. 3. Compares a native verification service with a Delegated Verifier in response time.

Figure 3 compares the response times of a verification request. This measurement was carried out by executing k6 within the same platform. The additional overhead in our proposal is the part where the verification result is signed with the key corresponding to the delegated certificate. However, the delegated verifier is mainly faster than native verifiers. Native is faster in bare metal environments, so we think this is due to optimization by Azure.

TABLE I
Comparison of response times with load balancing

Region	Response Time(ms) [No Load Balance]	Response Time(ms) [Load Balance]
East US	48	38
West Europe	310	39
Japan East	550	31
Brazil South	410	400
Australia East	660	350

Table I compares the case where Delegated Verifiers are load balanced across a geographically dispersed location with the case where there is only one Delegated Verifier in one area like Intel Attestation Service [1]. In the former case, we used Azure Traffic Manager as the geographically distributing load balancer and prepared the same VM located in the East US, West Europe, and Japan East. In the latter case, the verification service was deployed only in the East US. We used Azure Load Testing to evaluate the system, sending requests from five geographically separate locations. The result shows that the response is up to over 500ms faster, and load balancing for delegated verifier is effective.

IV. Conclusion

By evaluating the implementation using gramine, we confirm that our proposed delegation verifier is practical. Our proposal realizes remote attestation corresponding to more advanced confidential computing use cases than currently available and contributes to developing the relevant field.

In future work, we plan to perform formal verification of the delegation protocol based on the implemented API.

Acknowledgment

This work was supported by JST, PRESTO Grant Number JPMJPR21P6, JST CREST Grant Number JPMJCR21M3, JSPS KAKENHI Grant Number JP23H03373, and JST SPRING Grant Number JPMJSP2124, Japan.

References

[1] S. Johnson and V. Scarlata, "Intel Software Guard Extensions: EPID Provisioning and Attestation Services," Mar. 2016.
[2] "Intel® Trust Authority," Sep. 2023. [Online]. Available: https://www.intel.com/content/www/us/en/security/trust-authority.html
[3] "Veraison," https://github.com/veraison.
[4] T. Yagawa, T. Teruya, K. Suzaki, and H. Abe, "Delegating Verification for Remote Attestation Using TEE," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2024, pp. 186–192.
[5] "GitHub - intel/SGX-TDX-DCAP-QuoteVerificationService," https://github.com/intel/SGX-TDX-DCAP-QuoteVerificationService.
[6] C.-C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017, pp. 645–658.

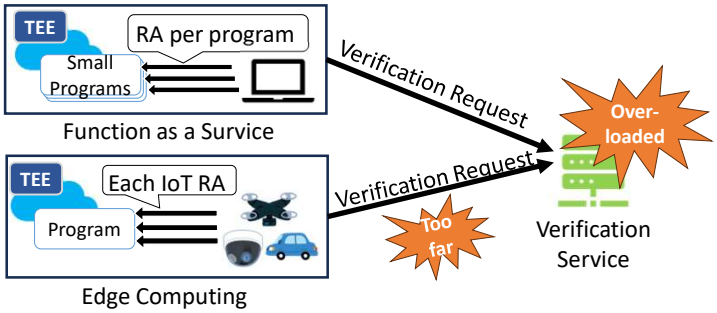
Poster: Flexible Remote Attestation System for Trusted Execution Environment using Delegating Verification

Takashi Yagawa[1,2], Tadanori Teruya[2], Kuniyasu Suzaki[3], Hirotake Abe[1]
[1]University of Tsukuba, [2]AIST, [3]Institute of Information Security

Motivation

Remote Attestation (RA) can remotely verify the status of the Trusted Execution Environment (TEE) on the cloud platform.
However, the current online verification service for RA does not satisfy the applied use case.

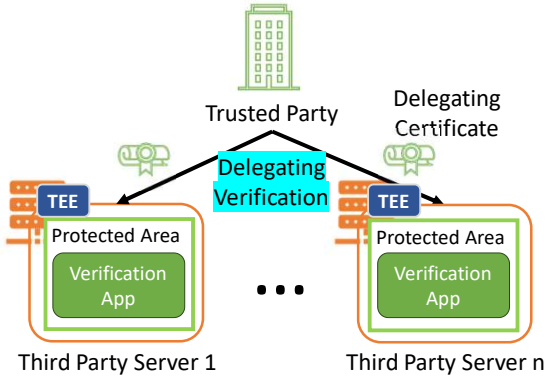
- FaaS: Increase verification request
- IoT: Require high responsiveness



Limitations of Existing Approach

- General-purpose online verification service (Intel Attestation Service (IAS) / Intel Trust Authority / Microsoft Azure Attestation Service)
  - Difficult to scale due to only restricted trusted operators
  - Privacy concerns to the operator if sensitive
- Quote Verification Library (QVL) / VERAISON
  - ✓ Anyone can build a verification service
  - Legitimacy depends on the operator
  - Costly to manage

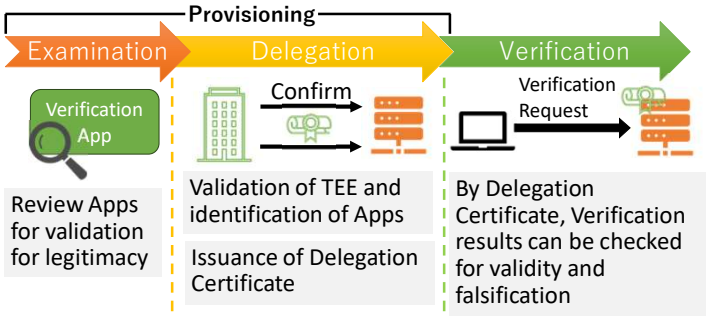
Approach



Delegating Verification[1]

- ✓ Trustable, independent of the operator
- ✓ Scalable, including geographic
- ✓ Confidential, verification within TEE

Overall Flow



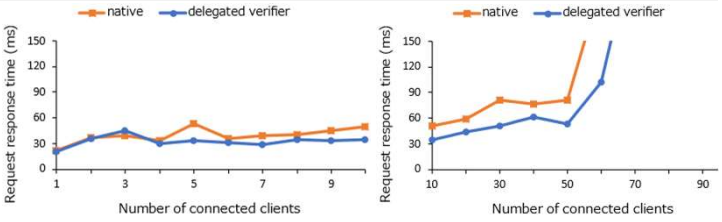
Implementation

TEE: Intel SGX + gramine (dev tool)
Third Party Server: Quote Verification Library
Trusted Party Server: gramine Sample

Evaluation

Compares a native verification service with a Delegated Verifier, which runs within TEE and signs the verification results corresponding to the Delegating Certificate.

➢ Communication overhead is practical enough



Using Microsoft Azure, Delegated Verifier was built in multiple regions to evaluate the effectiveness of load balancing.

➢ Successfully reduced response time due to geographic distributed

Region from	Response Time (ms)
East US	48
West Europe	310
Japan East	550
Brazil South	410
Australia East	660

No Load Balance (only East US)

Region from	Response Time (ms)
East US	38
West Europe	39
Japan East	31
Brazil South	400
Australia East	350

Load Balance (East US, West Europe, Japan East)

Conclusion

Delegated Verifier removes the reliability-dependent problems of previous studies and is suggested to be confidential and scalable.

[1] Takashi Yagawa, et al. “Delegating Verification for Remote Attestation Using TEE.” In 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 186–92. IEEE, 2024.