TEEを用いた低遅延でスケーラブルなRemote Attestationのための検証委任機構

矢川 嵩[1]　照屋 唯紀[2]　須崎 有康[3]　阿部 洋丈[1]

概要：クラウドプラットフォーム上の Trusted Execution Environment (TEE) については、Remote Attestation (RA) を利用する事で、その状態を遠隔地から確認できる。RA ではクラウドプラットフォームから提示された証拠情報を基に、専用サービスやツールを使用して状態を検証する。しかし、RA の検証について、IoT デバイスの増加やマイクロサービスの拡大への対応は考慮されていない。そこで本研究では、TEE を利用した、検証サービスを安全かつ容易に展開可能な検証委任システムを提案する。これにより、エッジコンピューティングや FaaS といったサービスにも対応した低遅延でスケーラブルな RA を実現できる。このシステムのプロトタイプを Intel Software Guard Extensions (SGX) [2] を用いて実装及び評価し、検証時の追加オーバーヘッドは小さく実用的であることを示す。

1. はじめに

近年、クラウドサービス上で機密にデータ処理ができる手段として、Trusted Execution Environment (TEE) が注目されている。クラウドプラットフォームにはユーザーは物理的にアクセスできないが、TEE やそこで動くソフトウェアの状態はの Remote Attestation（RA）によって確認できる。RA は、プラットフォームやプログラムの真正性と完全性を遠隔から確認する方式であり、生成された証拠をベンダー等が配布する情報を使って検証する事で完了する。

TEE の応用が進むにつれて、RA のリクエスト数は増加し、要求される条件も増えている。例えば、エッジコンピューティングでは、多くの IoT がクラウド上の TEE に対して RA を実行する。また、Function as a Service では、単純な設計では各小規模プログラムが RA を必要とする。さらに、これらのケースではリアルタイム性も求められる。

しかし、現状 TEE に対する RA の検証について、そのような問題は考慮されていない。オンラインの検証サービスは限られた数の信頼できる機関によって運営される必要があるが、そのような少数の検証サーバにリクエストが集中すると、応答の遅延や受付不可につながる可能性がある。検証サービスをユーザー自ら用意する事も可能であるが、これはクラウドサービスを利用したいと考えている大半のユーザーにとっては非常にコストがかかる。RA を実行するクライアント上で検証する事もできるが、デバイスの性能に処理速度が依存するため、IoT デバイス等の場合、レスポンスタイムが大きくなってしまう可能性がある。

本研究ではこの課題の解決策として、TEE を利用する事で、第三者が管理するサーバ上で検証サービスを安全に展開できる検証委任システムを提案する。展開された検証サービスは誰でも利用でき、低遅延でスケーラブルな RA を実現できる。そのプロトタイプを、Intel Software Guard Extensions (SGX) を用いて実装した。性能評価では、検証時の追加オーバーヘッドは小さく実用的であることが示された。

2. 設計

図 1 は検証委任システムの概略図である。サーバ上の TEE の中では検証用プログラムが動作しており、この検証プログラムの内容の正当性は保証されているものとする。検証資格の委任では、サードパーティが管理するサーバ上で正当な検証が実行されることを保証する。まず、認証局などの信頼できる第三者機関は、サーバ上の TEE 内に対し RA を実行し、検証プログラムを識別する。想定される正規の検証プログラムであれば、RA 対象の TEE 内に証明書を発行する。TEE によって検証プログラムと証明書の真正性と完全性を保証できるため、信頼する第三者の数を増やすことなく検証サービスの数を増やすことができる。

[1] 筑波大学
University of Tsukuba
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[3] 情報セキュリティ大学院大学
Institute of Information Security

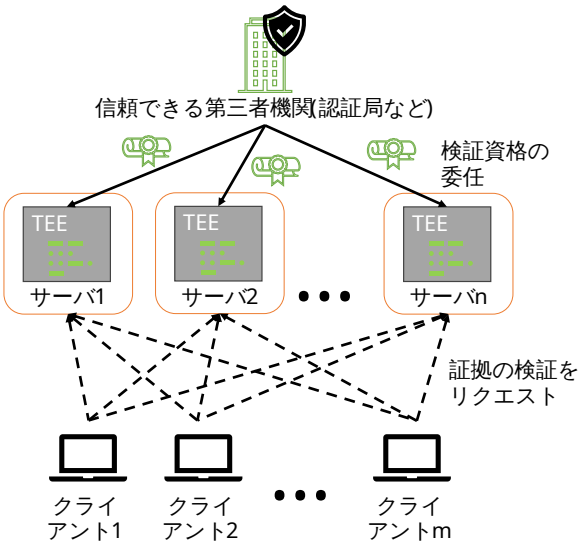

図 1 検証委任システムの概略図

クライアントによる検証リクエストでは、他 TEE プラットフォームから受け取った証拠を検証するために、検証資格を委任されたサーバを検証サービスとして利用できる。委任の確認には、信頼できる第三者機関からの証明書を利用する。また、クライアントは空いていたり距離的に近かったりする検証サービスを選択することで、RA の応答時間を改善できる。

3. 実装と評価

Intel が公開している Quote Verification Service (QVS) [1] を改変し、それを gramine[3] で保護することでプロトタイプを実装した。QVS は REST API サーバーであり、証拠を POST メソッドで送信することでその検証が行われる。gramine は、enclave に libOS を含めることでコードを変更せずに SGX を適用できるツールである。QVS を実行するコンテナに対して Gramine Shielded Containers を用いることで、SGX の保護領域である Enclave 内で QVS を動作させた。

評価プラットフォームは、Intel(R) Xeon(R) Silver 4314、64GB RAM、Ubuntu22.04 OS、6.2.0-36-generic kernel である。SGX SDK のバージョンは 2.22.100.3、SGX PSW のバージョンは 1.19.100.3-jammy1 を用いた。

元の RA と比較して追加のオーバーヘッドとなるのは、委任された証明書に対応した鍵で検証結果に署名する部分である。検証資格の委任では検証プログラムの確認と証明書の発行に時間がかかるが、これは事前に行われるため、実行時のオーバーヘッドにはならない。図 2 は、証拠を送信してから応答を受信するまでの時間を示している。SGX の有効・無効、委任による署名の有無でそれぞれ 4 つのケースを比較した。SGX によるオーバーヘッドは約 4 ミリ秒、署名によるオーバーヘッドは約 5 ミリ秒である。また、署名によるオーバーヘッドは、Enclave 内外で差はなかった。オーバーヘッドの合計は約 10 ミリ秒であり、これは実用的なほど小さい値である。

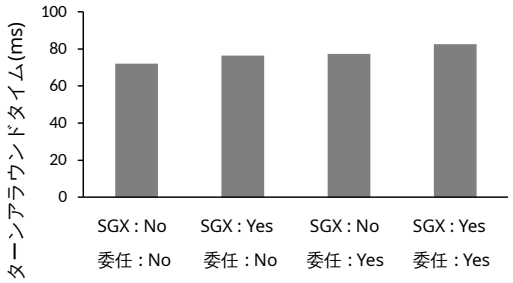

図 2 SGX と委任の有無によるオーバーヘッド評価

4. まとめ

gramine を利用したプロトタイプ評価により、提案のオーバーヘッドは約 10 ミリ秒であり、実用的であることが示された。今後は、検証資格の委任を API 経由で実行できるよう設計し、その性能を評価する予定である。

謝辞 本研究は、JST さきがけ（JPMJPR21P6）、JST CREST（JPMJCR21M3）、JSPS 科研費（JP23H03373）、および JST SPRING（JPMJSP2124）の支援を受けている。

参考文献

[1] : GitHub - intel/SGX-TDX-DCAP-QuoteVerificationService, `https://github.com/intel/SGX-TDX-DCAP-QuoteVerificationService`.
[2] Costan, V. and Devadas, S.: Intel SGX Explained., *IACR Cryptol. ePrint Arch.*, Vol. 2016, No. 86, pp. 1–118 (2016).
[3] Tsai, C.-C., Porter, D. E. and Vij, M.: {Graphene-SGX}: A practical library {OS} for unmodified applications on {SGX}, *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pp. 645–658 (2017).