オフラインでも利用可能な機密実行環境のアテステーション方式の提案

宮崎 想也[1]　新城 靖[1]

1. 序論

近年、インターネット上では電子決済など多様なサービスが提供され、その中には個人情報や著作物などの保護すべきデータをやり取りするものがある。そのようなデータは、機密実行環境（Trusted Execution Environment for confidential computing, TEE）と呼ばれる隔離されたプログラムの実行環境を用いることで保護することができる。信頼できないホスト上に構築された機密実行環境で動作するプログラムと通信を行う場合、そのホストやプログラムの真正性を検証する必要がある。そのために、リモートアテステーションと呼ばれる技術が使用される。

リモートアテステーションを行うためには、CPU ベンダなどが提供する情報にアクセスする必要がある。災害時や山間部などのインターネットから遮断されたオフライン状況では、従来の手法でリモートアテステーションを行えないという課題が存在する。また、リモートアテステーションだけでは通信路を保護するための機能を提供しないため、安全な通信路を確立するためには Transport Layer Security (TLS) のような暗号プロトコルと連携させる必要がある。この連携の実装は極めて難解である。

そこで本研究では、次の 2 つの目的を設定する。第 1 にオフライン状況においても、機密実行環境のアテステーションを可能にすることである。第 2 に、TLS を利用した通信路の確立とリモートアテステーションを簡単に行えるようにすることである。それを実現するためにこの論文ではリモートアテステーションと同時に証明書を発行するサービスを実装し、その証明書を用いて TLS 通信路を確立することを提案する。

2. 脅威モデル

本研究では以下の要素を信頼する。

- CPU とそのベンダのサーバ、認証局
- Web でも使われている X.509 に基づく認証局

本研究では以下の攻撃を想定する。

- ネットワーク上のメッセージの盗聴、改ざん

[1] 筑波大学
University of Tsukuba

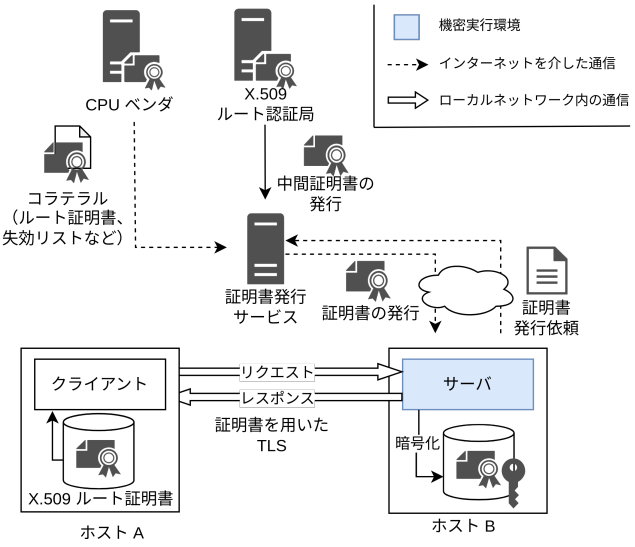

図 1　証明書を用いたアテステーションの実装と安全な通信路の確立

- プログラムや主記憶の内容の改ざん
- 永続記憶に記録されたデータの改ざん
- 信頼できるサーバへのなりすまし

3. 提案手法

3.1 概要

本研究では、オフライン状況でも利用可能なアテステーションを図 1 に示すように証明書発行サービスで実装する。機密実行環境で動作するサーバは、証明書発行サービスが発行する証明書を TLS におけるサーバ証明書として利用することで、自身のホストやプログラムが真正なものであることをクライアントに示す。

3.2 証明書発行サービス

証明書発行サービスは機密実行環境で動作するプログラムがオンラインの時にリモートアテステーションを行い、証明書を発行する。このサービスは中間認証局として位置し、そのサーバ証明書は Web ブラウザなどで利用されるルート認証局のルート証明書によって署名される。これにより、悪意のあるサイトが作成した証明書発行サービスによる不正な証明書の発行を抑制する。

証明書発行サービスが発行する証明書は、一般的に TLS

で利用される X.509 形式の証明書である。証明書の発行依頼を行った機密実行環境の構成情報（Quote）は、先行研究 [1][2] と同様に TLS 1.3 で利用可能な証明書の拡張領域に格納する。この構成情報は、機密実行環境のコードとその署名者、機密実行環境のビルド環境などから生成される識別子を含む。

3.3 証明書の発行

最初に、機密実行環境で動作するサーバはオンライン状況で、ハードコードされた X.509 ルート証明書で検証可能な証明書発行サービスに接続し、リモートアテステーションを受ける。次に、以下の情報を含む Certificate Signing Request (CSR) を送る。

- 機密実行環境内で生成した公開鍵
- リモートアテステーションで得られた機密実行環境の構成情報

機密実行環境のリモートアテステーションに成功した場合、証明書発行サービスはこれらの情報を含む署名付きの証明書を発行する。一方で、機密実行環境の検証に失敗した場合は、サーバ証明書の発行を行わない。機密実行環境で動作しているプログラムは発行された証明書と秘密鍵を、機密実行環境が固有に導出できる鍵を用いて暗号化し、ホスト上の信頼できない永続記憶上に保存する。

3.4 証明書を用いた TLS 通信路の確立

クライアントは機密実行環境で動作しているサーバと TLS による通信路を確立する時に、一般的な TLS と同様にサーバが提示した証明書を用いてハンドシェイクを行う。このとき、サーバは永続記憶に暗号化して保存した証明書を復号し、サーバ証明書としてクライアントに送信する。クライアントはこのサーバ証明書を Web ブラウザと同じ手法で検証する。次に、証明書に格納された構成情報を確認し、想定された通信相手かどうかを判定する。リモートアテステーションにより構成情報の完全性は保証されているため、クライアントはその内容を信頼して判定を行う。問題がなければ、クライアントは TLS 通信路を確立する。クライアントもサーバと同様に証明書を提示することで、相互 TLS 認証の通信路を確立できる。

4. セキュリティに関する議論

攻撃者は永続記憶に保存された証明書を改ざんできるが、復号して得られる証明書は不正なものとなるため、攻撃を検知できる。また、証明書の発行後にプログラムを改変した場合にも、機密実行環境の機能により、固有に導出できる鍵が変化するため復号に失敗するようになり、同様に攻撃を検知できる。

攻撃者が不正に証明書発行サービスを構築したとしても、サーバは Web ブラウザと同様に X.509 ルート証明書からのチェーンをたどることで検出できる。また、クライアントは自分が意図しているサーバと接続していることを、セーフブラウジングリストなどの既存の手法で確認できる。

5. 実装

本研究では、機密実行環境の実装として、Intel Software Guard Extensions (SGX) を利用する。リモートアテステーション方式には、Data Center Attestation Primitives (DCAP) を用いる。証明書発行サービスおよび機密実行環境で動作するアプリケーションは Rust 言語で実装する。また、Rust から Intel SGX SDK を利用するために Automata SGX SDK を使用する。

6. 関連研究

RA-TLS[1] は、まず機密実行環境のリモートアテステーションを行い構成情報を含む一時的な自己署名証明書を作成し、その証明書を用いて TLS による安全な通信路の確立を行う手法である。TLS+RA[2] は、TLS のハンドシェイク時に機密実行環境の構成情報を送信し、検証することで TLS の通信路確立と同時にリモートアテステーションを行う手法である。

これらの研究は、リモートアテステーションと TLS を統合するという点で本研究と共通している。本研究は、オフライン状況でも証明書による認証が行えるよう、事前にリモートアテステーションを行い、自己署名ではなく認証局により署名された証明書を利用する点で異なる。

7. まとめ

本研究では、機密実行環境で動作するプログラムが事前にリモートアテステーションを行いサーバ証明書を取得する方法を提案した。クライアントはこのサーバ証明書を用いることで、通常の TLS と同じ手順で安全な通信路を確立しながら、通信相手の機密実行環境とプログラムの真正性を確認できる。その特徴は、オフライン状況においても、安全な通信と機密実行環境のアテステーションを行うことができることである。今後は、証明書発行サービスの実装を進め、機能評価および性能評価を行う。

参考文献

[1] Thomas Knauth, Michael Steiner, Somnath Chakrabarti, Li Lei, Cedric Xing, and Mona Vij. Integrating remote attestation with transport layer security. `https://arxiv.org/abs/1801.05863`, 2019.

[2] Carsten Weinhold, Muhammad Usama Sardar, Ionuț Mihalcea, Yogesh Deshpande, Hannes Tschofenig, Yaron Sheffer, Thomas Fossati, and Michael Roitzsch. Separate but together: integrating remote attestation into tls. In *Proceedings of the 2025 USENIX Conference on Usenix Annual Technical Conference*, pp. 1319–1326, 2025.