大規模分散システムにおける階層的リモートアテステーション手法の提案と評価

千田 拓矢[1,a)]　長谷川 慶太[2]　永田 智大[1]　中川 智尋[1]　佐々木 一也[1]　寺田 雅之[1]

概要：近年，クラウドや分散システムの発展に伴い，異なる組織間で安全に通信・連携を行うためのリモートアテステーションの重要性が増している．しかし，大規模なシステムにおいて各組織の多数のサーバ間で個別に相互アテステーションする手法は，処理時間や運用コストの面で非効率である．そこで本研究では，各システム内においてリーダー／フォロワー構成を導入し，階層的にリモートアテステーションを行う手法を提案する．各フォロワーサーバは TEE を用いて自身の構成証明をリーダーに提出し，リーダーがこれを集中的に検証する．リーダー同士が相互にアテステーションを実施し，クレデンシャル情報を配布することで，システム全体が信頼できることを効率的に検証できる．評価の結果，処理時間はサーバ台数に対して線形にスケールし，数百台規模のシステムにおいても現実的な時間で完了することが確認できた．本手法により，アテステーションのスケーラビリティと実用性の向上が可能となる．

キーワード：クラウド, Confidential Computing, リモートアテステーション, TEE

Design and Evaluation of a Hierarchical Remote Attestation Framework for Scalable Distributed Systems

Takuya Chida[1,a)]　Keita Hasegawa[2]　Tomohiro Nagata[1]　Tomohiro Nakagawa[1]
Kazuya Sasaki[1]　Masayuki Terada[1]

Abstract: In recent years, the growth of cloud and distributed systems has increased the need for secure remote attestation across organizations. Conventional methods, which require pairwise attestation among many servers, are inefficient at scale. We propose a hierarchical leader–follower approach: followers submit TEE-based attestations to a leader, leaders attest to each other, and credentials are shared. This allows efficient verification that the entire system can be trusted. Evaluation shows processing time scales linearly with server count, completing within practical time even for hundreds of servers, thus improving scalability and practicality.

Keywords: Cloud Computing, Confidential Computing, Remote Attestation, TEE

1. はじめに

1.1 背景と目的

近年，クラウドコンピューティングの発展に伴い，大規模分散システム上でのデータ処理や機械学習の利用が拡大している．これらの環境では，数十台から数百台規模のサーバ群が協調して処理を行うことが一般的であり，特に個人情報などの機密データを扱う場合には，高い信頼性と安全性の確保が不可欠である．Trusted Execution Environment (TEE) を用いたセキュアな計算環境は，ハードウェアベースで機密性と完全性を保証する仕組みとして注目を集めている．しかし，複数の組織やクラウド環境にまたがる大規模分散システムにおいて，各サーバが TEE 上で正しく動作していることを効率的に保証する仕組みは十分に確立されておらず，依然として重要な研究課題である．

[1] NTT DOCOMO モバイルイノベーションテック部
[2] NTT 社会情報研究所
[a)] takuya.chida.mn@nttdocomo.com

1.2 データ連携における課題

リモートアテステーションは，TEE 内で稼働するソフトウェアや環境の正当性を外部から検証可能とする技術であり，セキュアなデータ連携の基盤を支えている．しかし，大規模分散システムにおいては以下の課題が存在する．

- **スケーラビリティ**：各サーバが直接相互にアテステーションを行う場合，接続数はサーバ台数の二乗に比例して増加し，数百台規模では通信および検証処理のオーバーヘッドが深刻となる．
- **運用効率**：多数のサーバに対して個別にアテステーション結果を検証・管理することは，運用コストや処理時間の点で非効率である．
- **セキュリティリスク**：アテステーションの不備や不正ノードの混入により，システム全体の信頼性が損なわれる可能性がある．

したがって，大規模分散システムに適した効率的かつ安全なリモートアテステーション手法が求められている．

1.3 本研究の提案

本研究では，大規模分散システムにおける効率的な信頼確立を可能とする階層的リモートアテステーション手法を提案する．本手法では，各組織にリーダーとなるサーバを配置し，リーダーが自身の配下のフォロワーサーバのアテステーションを一括して実行した上で，リーダー間で相互アテステーションを行う．これにより，サーバ台数の増加に伴う処理負荷を抑制し，システム全体の信頼性を効率的に確保することを目指す．本研究では，クラウド環境 (AWS Nitro Enclaves) に実装し，性能評価を通じて実用性を検証する．

1.4 本稿の貢献

本研究の主な貢献は以下の通りである．

- 階層的リモートアテステーション手法の提案：リーダー・フォロワー構造を導入することで，セキュリティリスクを軽減し，スケーラブルかつ効率的なアテステーション方式を実現した．
- 実装と評価：AWS Nitro Enclaves 上にシステムを実装し，サーバ台数を増加させた実験により，スケーラビリティと効率性を検証した．
- 実用性の検討：実験を通じて，大規模分散システムにおいても現実的な時間で信頼確立が可能であることを示し，マルチクラウド運用などの拡張性についても提案手法が有効であることを議論した．

1.5 論文構成

本論文の構成は以下の通りである．第 2 章では分散システムにおけるリモートアテステーションの関連研究を整理し，第 3 章で提案手法を説明する．第 4 章ではクラウド環境における実装を示し，第 5 章で評価結果を述べる．第 6 章ではセキュリティ強度やスケーラビリティについて考察し，最後に第 7 章で本研究の結論と今後の課題を述べる．

2. 関連研究

リモートアテステーションは，機密性の高いデータを扱うサーバやデバイスにおけるセキュリティ確保の基盤技術として，多くの研究が蓄積されてきた．その中で，Birkholz らによる RFC9334 [2] はリモートアテステーションの基本的な概念とアーキテクチャを標準化し，Attester，Verifier，Relying Party といった役割を整理した点で大きな貢献を果たしている．また，Asokan ら [1] は IoT など多数のデバイスを対象に，スパニングツリー構造を利用した効率的なアテステーション手法を提案しており，大規模環境における検証効率化に重点を置いている．

Intel SGX を対象とした研究としては，Chen らによる OPERA [3] が挙げられる．これは複数の論理コアを利用した並列的なアテステーションの性能評価を行い，ハードウェア特性に基づく実用的な手法を提示している．さらに Chen らは後に MAGE [4] を提案し，様々な種類の TEE 間でアテステーションを行う仕組みを導入することで，信頼の成立方法に新たな可能性を示した．

その他の分散環境を前提とする研究としては，Li らの DMA [5] があり，EPIID グループと呼ばれる合意アルゴリズムを利用して，第三者に依存しない相互アテステーションを実現している．Kwon ら [7] は集約署名を利用して，複数デバイスから得られるアテステーション結果を効率的に統合する方式を提案し，検証負荷の軽減に貢献している．また，Petzi ら [6] はブロックチェーン上のスマートコントラクトを利用することで，スケーラブルかつ不正な検証者への耐性を備えた方式を提案しており，IoT ネットワークの利用を想定した設計となっている．

このように，リモートアテステーションの研究は標準化の試みから特定ハードウェアに特化した実装，さらには分散環境を考慮した新たなプロトコル設計まで多岐にわたって展開されている．本研究はこれら既存の知見を基盤としつつ，異なる組織間の大規模分散システムにおいて効率的なアテステーション手法の実現を目指すものである．

3. 提案手法

3.1 システムモデルと前提条件

本論文では，異なる組織が独自にクラウド上へ構築した大規模分散システム間でのリモートアテステーションを前提とする．各組織は独自のセキュリティポリシーを維持しつつ，相互に信頼関係を確立しながらデータ連携を行う必要がある．このような環境下では，全てのデータ処理サーバが信頼できることが前提となる．

さらに，本研究では 各組織は自組織のシステムを信頼

できるものと仮定するが，相手組織のシステムは信頼しない という前提を置く．したがって，相互にリモートアテステーションを行い，相手側の環境の完全性を検証することが不可欠である．

また，本論文ではクラウド環境を跨いでリモートアテステーションを行う方式をグローバルアテステーション (Global Attestation) と定義し，同一クラウド環境内でのアテステーションをインターナルアテステーション (Internal Attestation) と定義する．前者は異なる組織間における信頼確立を目的とし，後者は自組織のサーバ群が相手組織から見て信頼できることを技術的に示すことを目的とする．

3.2 構成証明の生成と検証

本手法では，各組織のクラウド環境内で動作するサーバー群が，TEE を利用して構成証明を生成し，相互に検証することを前提とする．具体的には，AWS Nitro Enclaves や AMD SEV-SNP などの TEE 環境を利用することで，サーバの実行環境の正当性を保証する．

構成証明ドキュメント (Attestation Document) は，TEE 内で実行されるアプリケーションや環境の状態を外部に証明するためのドキュメントである．具体例として AWS Nitro Enclaves における構成証明ドキュメントの例を以下に示す．

```
AttestationDocument = {
    module_id: text,
    timestamp: uint .size 8,
    digest: digest,
    pcrs: { + index => pcr },
    certificate: cert,
    cabundle: [* cert],
    public_key: user_data,
    user_data: user_data,
    nonce: user_data,
}
```

本研究では，PCR（Platform Configuration Register）値を用いてアプリケーションおよびその実行環境に改ざんがないかを検証する．さらに，証明書（certificate）および CA バンドル（cabundle）を利用して構成証明署名の正当性を確認する．これにより，各サーバの実行環境の完全性を検証でき，ユーザのプログラムが信頼できる環境で実行されていることを技術的に保証する．

3.3 Leader/Follower による階層的構成証明手法

本研究におけるシステムは，リーダー (Leader) とフォロワー (Follower) から構成される．また，各組織が独立したクラウド環境を持ち，相互に信頼関係を構築することを前提とする．具体的なシステムモデルは以下の通りである．

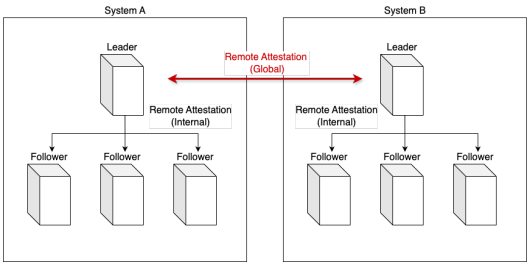

図 1 提案手法のリモートアテステーション

- **リーダー (Leader)**: 自組織内の演算サーバ群を統括する制御サーバである．フォロワーから構成証明ドキュメントを収集・検証し，同一組織内でのアテステーション（インターナルアテステーション）を実施する．さらに，他組織のリーダーとの相互アテステーション（グローバルアテステーション）を行い，信頼を確立した上でクレデンシャル情報を配布する．各組織のフォロワーは，自組織のリーダーが責任を持って信頼できることを確認する．
- **フォロワー (Follower)**: アプリケーション処理を担当するサーバ群であり，TEE を用いて自身の構成証明を生成しリーダーに提出する．信頼が確立した後，リーダーからクレデンシャル情報を受領し，アプリケーション処理に利用する．

このように，本手法により各組織は自組織内のサーバの信頼性を確保しつつ，効率的に他組織との信頼関係を構築できる．

3.4 提案手法と従来手法の比較

提案手法と従来手法の概念図をそれぞれ図 1，図 2 に示す．提案手法では，フォロワーの構成証明をリーダーが事前に一括で検証するため，従来手法のように相手組織の全サーバに対して直接相互アテステーションを行う必要がなく，通信および検証の負荷を大幅に軽減できる．従来手法では，サーバ台数を N とすると接続数が二乗に比例して増加し，計算量は $\mathcal{O}(n^2)$ となるため，サーバ台数の増加に伴い検証処理の負荷が急増する．一方，提案手法ではリーダーがフォロワーの構成証明を一括で検証するため，計算量は $\mathcal{O}(n)$ に抑えられ，スケーラビリティの向上が期待できる．

さらに，提案手法では通信の大部分が同一クラウド環境内で行われるインターナルアテステーションであり，クラウド環境を跨ぐグローバルアテステーションの実行回数を削減できる．これにより，対向環境とのネットワーク構築や運用にかかるコストも低減される．また，リーダーが自組織の全サーバを責任を持って検証するため，接続数を削減したとしてもシステム全体の信頼性が損なわれることはない．

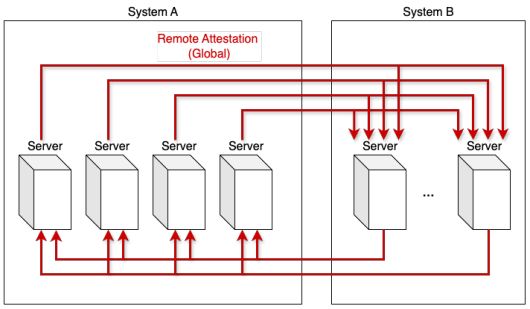

図 2　従来手法のリモートアテステーション

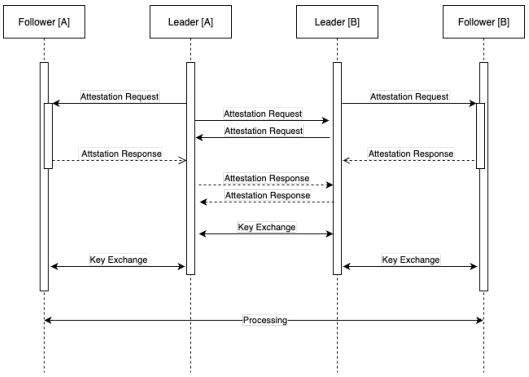

図 3　Leader/Follower によるリモートアテステーション

3.5　リモートアテステーション詳細

本研究におけるリモートアテステーションの手順を図 3 に示す．具体的には以下の手順で構成される．

(1) **フォロワーによる構成証明生成**: リーダーは各フォロワーに対して構成証明ドキュメントを生成するようにリクエスト (Attestation Request) を送信する．フォロワーは自身の TEE 内で構成証明ドキュメントを生成し，リーダーにレスポンス（Attestation Response）を送信する．

(2) **リーダーによる構成証明生成**: リーダーは対向環境のリーダーに対して構成証明ドキュメントを生成するようにリクエスト (Attestation Request) を送信する．対向環境のリーダーは自身の TEE 内で構成証明ドキュメントを生成する．

(3) **リーダーによる検証処理**: リーダーは受け取った自身のフォロワーの構成証明ドキュメントを検証し，構成証明ドキュメントの正当性と PCR 値の整合性を確認する．

(4) **リーダー間の相互アテステーション**: 各組織のリーダー同士が相互にアテステーションを行い，信頼を確立する．

(5) **クレデンシャル情報の配布**: 相互アテステーション完了後，リーダーは検証済みのフォロワーにクレデンシャル情報（その後の処理で利用する共通鍵やトークンなど）を配布する．

4.　実装と評価

4.1　実装環境

本研究の提案手法を検証するため，AWS クラウド上に実験環境を構築した．実装したシステムの構成図を図 4 に示す．

4.1.1　ハードウェア環境とソフトウェア環境

ハードウェア環境として，インスタンスタイプは `m5.2xlarge`（8 vCPU, 32 GiB メモリ）を採用し，TEE として AWS Nitro Enclaves を採用した．CPU には Intel Xeon® Platinum 8175M を利用し，最大 10 Gbps のネットワーク帯域を利用可能とした．Enclave に対しては親インスタンスとなる EC2 のリソースの内, 2vCPU/1GiB を割り当てる．ネットワークには東京リージョン（ap-northeast-1a）のプライベートサブネットを利用した．

ソフトウェア環境としては以下の通りである．

項目	バージョン
OS（オペレーティングシステム）	Amazon Linux 2023
プログラミング言語	Python 3.9.23
コンテナ基盤	Docker 25.0.8
Enclave 制御ツール	Nitro CLI 1.4.2

表 1　ソフトウェア環境

4.1.2　AWS の各機能の概要

本実装では，AWS の各種サービスを利用してシステムを構築した．以下に使用した機能の概要を示す．

- **AWS EC2**: 仮想サーバを提供するサービスであり，リーダーおよびフォロワーのサーバを構築するために利用した．各 EC2 サーバ上では，後述の AWS Nitro Enclaves を用いて隔離された実行環境を構築する．
- **AWS Nitro Enclaves**: EC2 の拡張機能として，信頼された隔離実行環境（TEE）を提供する．Enclave 内では，アプリケーションの実行やデータ処理をセキュアに行うことが可能である．
- **AWS S3**: クラウドストレージサービスであり，構成証明ドキュメントやクレデンシャル情報の保存・共有に利用した．S3 バケットは，同一環境内のデータ保存用（Internal）と対向環境との共有用（Share）の 2 種類を用意した．特に Share バケットは外部アクセス権限の付与が必要となるため，バケットポリシーを適切に設定した．
- **AWS Auto Scaling**: サーバの自動スケーリングを行うサービスであり，フォロワーサーバを指定台数まとめて起動・終了する際に利用した．本機能を用いて，サーバ台数を 8 台から 128 台まで段階的に増加させ，スケーラビリティ評価を実施した．
- **AWS SSM (Systems Manager)**: サーバのリモー

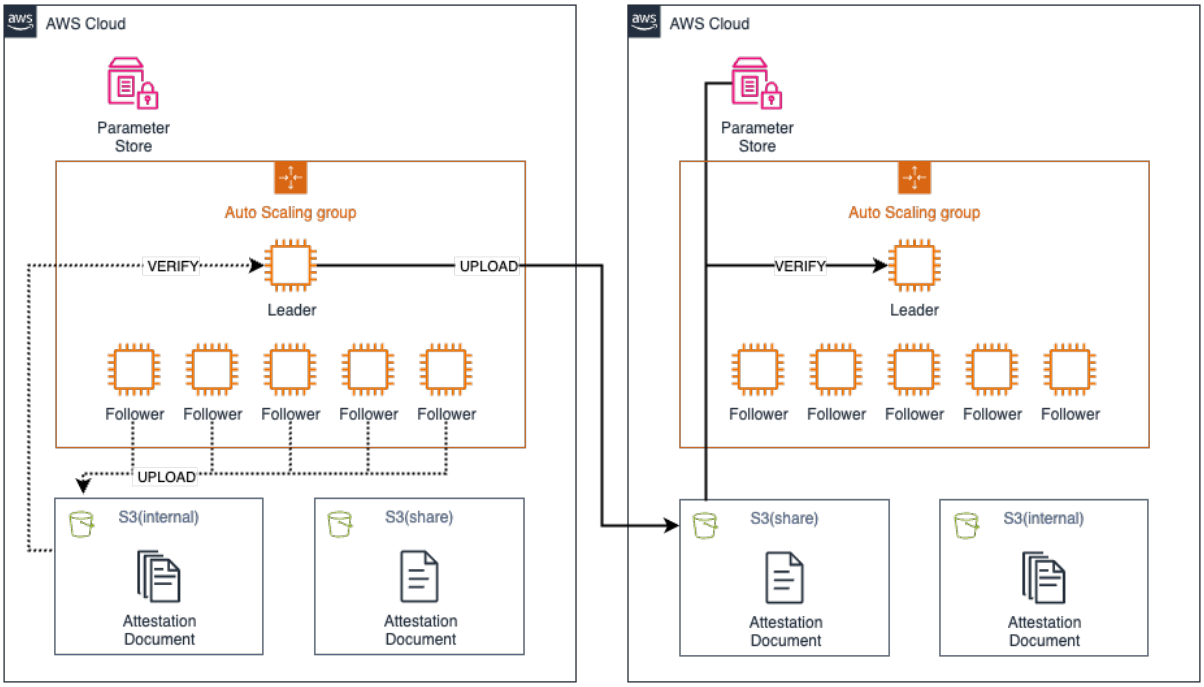

図 4　システム構成図

ト管理やコマンド実行を可能にするサービスである．本実装では，各サーバへの処理開始コマンドの送信（Run Command）や，PCR 値の検証に用いるパラメータの外部保管（Parameter Store）に利用した．

- **AWS CloudWatch**: Enclave 内で稼働するアプリケーションのログ収集に利用した．通常，Enclave 内のログはデバッグモードでなければ外部から取得できないが，本実装ではアプリケーションから CloudWatch Logs へ直接出力することで，外部からのログ参照を可能にした．

4.2　処理手順

具体的な処理手順は以下の通りである．

4.2.1　親インスタンスの起動と構成

まず，Enclave を利用するための親インスタンスを起動し，必要なソフトウェア（Docker，Nitro CLI，boto3 など）をインストールする．親インスタンスと Enclave 環境は vsock[*1] を介して接続され，構成証明ドキュメントや中間データの送受信が可能である．親インスタンスには，Enclave への構成証明ドキュメント生成リクエストや，生成されたドキュメントを S3 に保存する処理（UPLOAD）を行う Python スクリプトを用意する．

4.2.2　Enclave の起動と構成

各サーバは AWS Nitro Enclaves を利用して隔離環境を構築する．具体的には，EC2 インスタンス上で `nitro-cli` を用いて Enclave 用の Docker イメージ（EIF: Enclave Image File）を作成し，インスタンス起動時に自動的に実行されるように設定する．EIF 作成時には複数の PCR 値が出力されるが，本研究では PCR0 を利用する．PCR0 は EIF およびその実行環境を含むハッシュ値であり，Enclave 内で実行されるアプリケーションの完全性検証に用いる．Enclave 内では，親インスタンスからのリクエストに応じて構成証明ドキュメントの生成・検証や外部へのログ送信を行うスクリプトを配置する．

4.2.3　サーバの起動と構成証明の生成

Auto Scaling を利用して，指定台数のサーバを起動する．起動したサーバのうち InstanceID が最小のサーバをリーダー，その他をフォロワーとする．AWS SSM の RunCommand 機能を用いて，各サーバに構成証明生成リクエストを送信する．Enclave 内で生成された構成証明ドキュメントは，親インスタンスを経由して S3 バケットにアップロードされる．

4.2.4　同一環境内のアテステーション

まず同一環境内のアテステーション（Internal Attestation）を行う．リーダーは，複数フォロワーからアップロードされた構成証明ドキュメントを S3 経由で取得し，それぞれを検証（VERIFY）する．検証には Nitro CLI および Nitro Secure Module library API を用い，署名の正当性と PCR 値の整合性を確認する．これにより，フォロワーが正当な TEE 環境で動作していることを一括で保証できる．

4.2.5　リーダー間の相互アテステーション

次に，各組織のリーダー同士が相互アテステーション（Global Attestation）を行う．手順はリーダー／フォロワー間の検証と同様であり，CLI および API を用いて構

[*1] vsock (Virtual Socket) は，親インスタンスと Enclave 間の通信に特化したソケットインターフェース．

成証明を交換し，信頼関係を確立する．生成された構成証明ドキュメントは，バケットポリシーを設定した共有用 S3 バケットを介して交換される．また，対向環境の PCR 値はパラメータストアに保存されており，リーダーはこれを参照して検証を行う．

4.2.6 共通鍵の生成と配布

相互アテステーション完了後，リーダーはクレデンシャル情報として共通鍵を生成し鍵交換を行う．この共通鍵は既に検証済みのフォロワーに配布する．この共通鍵を用いることで，以降のデータ連携や暗号処理を安全に実行できる．

4.3 評価方法

本研究では以下の二種類の評価を行った．

- **評価 1: 従来手法と提案手法の処理時間の比較**
  提案手法では，グローバルアテステーションはリーダー間でのみ行われるため回数は一定に収まり，大部分の処理はインターナルアテステーションで構成される．一方，従来手法では，ほぼ全ての処理がグローバルアテステーションで占められる．サーバ台数を N とした場合，従来手法ではグローバルアテステーションの回数が $\mathcal{O}(n^2)$，提案手法ではインターナルアテステーションの回数が $\mathcal{O}(n)$ で増加する．
  本評価では，アテステーションにかかる時間を構成証明ドキュメント生成（Generate），アップロード（Upload），ダウンロード（Download），検証（Verify）の 4 つの処理に分けて計測し，数百台規模のシステムでも現実的な時間でアテステーションが完了できるかを試算する．これにより，提案手法と従来手法の処理時間の違いを明らかにする．
- **評価 2: サーバ台数のスケーラビリティ評価**
  サーバ台数を 8 台，16 台，32 台，64 台，128 台と変化させ，起動から共通鍵配布までの全体処理時間を計測する．本評価により，提案手法における処理時間が $\mathcal{O}(n)$ で増加することを実際に確認し，スケーラビリティの性能を定量的に評価する．

4.4 評価結果

- **評価 1 の結果**　まず，アテステーション処理の所要時間の内訳を図 5 に示す．全体の処理時間としては，Internal が約 2.5 秒，Global が約 3.2 秒であった．そのうち構成証明ドキュメントの生成が約 2 秒で支配的となっており，その他の処理は数十ミリ秒である．Global の方がドキュメント生成にやや時間がかかるのは，Enclave 環境の違いによる誤差と考えられる．また，図 6 にはドキュメント生成を除いた所要時間を積み上げグラフで示す．Global では構成証明書のダウンロードに時間がかかっており，Internal と比べて通信

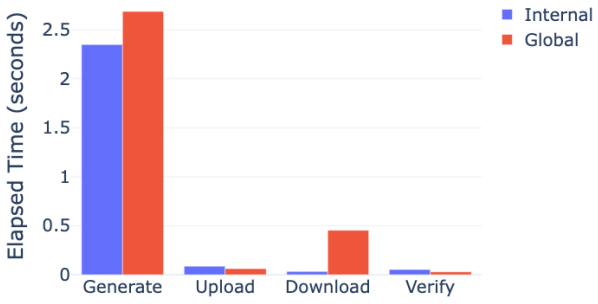

図 5　リモートアテステーションの所要時間内訳

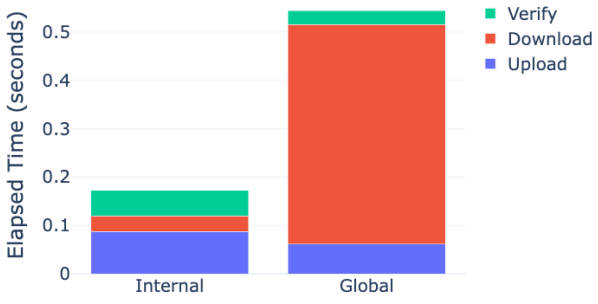

図 6　構成証明ドキュメント生成後の所要時間比較

コストが高いことがわかる．
さらに，提案手法は $\mathcal{O}(n)$，従来手法は $\mathcal{O}(n^2)$ の計算量であるため，アテステーションの回数だけでも提案手法の方が効率的である．加えて，Global と Internal で処理時間に差があることから，全体の処理時間においても提案手法では従来手法より大幅な短縮が可能であると言える．

- **評価 2 の結果**
  実験結果を図 7 に示す．サーバ台数を 8 台から 128 台に増加させた場合，処理時間はサーバ台数にほぼ線形に比例して増加しており，提案手法が $\mathcal{O}(n)$ の計算量で済むことが確認できた．また，128 台の場合でも全体の処理時間は約 70 秒であり，数百台規模の分散システムにおいてもおおむね実用的な性能を有すると言える．

5. 考察

本研究で提案したリーダー／フォロワー型のリモートアテステーション方式について，セキュリティ強度，スケーラビリティ，運用コストおよび拡張性の観点から考察を行う．

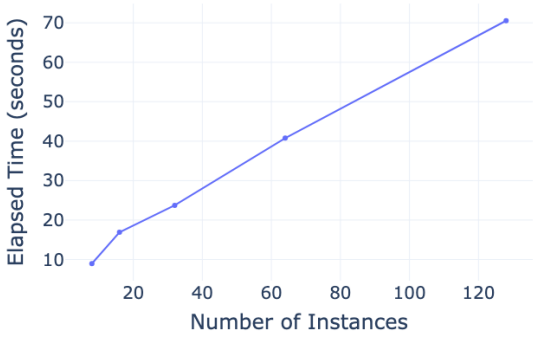

図 7 サーバ台数ごとの処理時間の変化

5.1 セキュリティ強度

従来手法では，全てのサーバが相手組織のサーバを個別に検証するのに対し，提案手法ではリーダー同士の検証のみを行い，組織間でのフォロワー同士の直接検証は行わない．そのため，取得できる情報には偏りが生じる可能性がある．それに対して，提案手法では構成証明の対象としてアプリケーション自体も含めており，リーダーがフォロワーの完全性検証を行う役割そのものをアテステーションの対象としている．これにより，各組織のリーダーがアプリケーションレベルで自組織のフォロワーを検証していることが外部から技術的に保証されるため，セキュリティ強度は損なわれないと考えられる．ただし，アプリケーションにバグや脆弱性が存在するリスクは依然として残るため，脆弱性スキャンやソフトウェアの適切なアップデートなどで対策する必要がある．

5.2 スケーラビリティ

大規模分散システムの課題であるスケーラビリティに対して，本研究の提案手法は $\mathcal{O}(n)$ の計算量でアテステーションの接続数を抑えることができることを示した．従来手法では組織間をまたぐグローバルアテステーションの回数が $\mathcal{O}(n^2)$ に従って増加するため，サーバ台数が増えると数百台規模のシステムでは 1 時間以上を要することが想定される．それに対して，提案方式は 128 台のフォロワーインスタンスを対象に性能評価を行い，十分な処理性能を確認できた．しかし，台数がさらに増加するとリーダーインスタンスへの負荷が集中し，ボトルネックとなる可能性がある．これに対しては，図 8 のようにリーダー配下に中間リーダーを配置した階層的構成や，同一フォロワー内で複数の Enclave を並列動作させ，親インスタンスで構成証明ドキュメントを集約するといった対応策が考えられる．

また，障害耐性については，フォロワー障害時には個別にアテステーションを再実行すればよいが，リーダー障害時には全体の再実行が必要となる．この点は今後の課題として，冗長化やリーダー切替の仕組みを導入する必要がある．

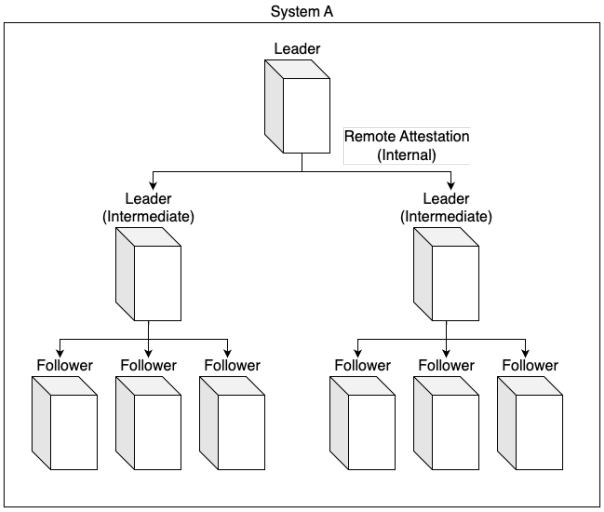

図 8 中間リーダーを配置した階層構造

5.3 運用コスト

本提案手法では各組織のクラウドをまたぐ通信を最小限に抑えることが出来るため，ネットワークや権限設定を簡略化することが可能であり，運用上大きなメリットがある．また，自組織に限って言えば，相手の環境構築を待つことなく非同期にリモートアテステーションを実行することが出来るため，全体の処理時間削減にもつながる．

5.4 拡張性

本研究では AWS Nitro Enclaves を利用して検証を行ったが，アテステーション方式に関しては RFC9334[2] で定義されるアーキテクチャに則しているため，異なる種類の TEE（Intel TDX や AMD SEV-SNP など）に対しても適用可能である．また，KMS や IAM といったクラウドベンダ固有のセキュリティ機能に依存していないため，AWS に限定されず他のクラウド環境（Azure やオンプレクラウド等）へも容易に展開可能である．このことから，本方式はマルチクラウド環境においても有効な枠組みとなりうる．

6. 結論と今後の課題

本研究では，大規模分散システムにおける信頼確立手法として，階層的なリモートアテステーション方式を提案し，その有効性を示した．リーダーサーバを導入することで，組織を跨ぐグローバルアテステーションの数を削減し，ネットワークコストや運用コストを低減できることを明らかにした．また，数百台規模の分散システムにおいても，現実的な時間内でアテステーションを完了させ，全体の信頼性を検証できることを確認した．これにより，従来の全てのサーバ同士が相互にアテステーションを行う場合に比べ，スケーラビリティおよび効率性の向上が可能であるこ

とを示した．

今後の課題としては，より大規模かつ柔軟な環境に対応するために，中間層にリーダーを配置したマルチリーダー構成の導入や，異なる TEE が混在する環境での相互運用性の検証が挙げられる．また，本手法は特定の TEE やクラウドベンダーに依存しない設計であるため，マルチクラウド環境への適用拡大が期待できる．これらの課題に取り組むことで，提案手法をさらに発展させ，異種混在かつ異なるクラウド間に展開される大規模分散システムにおいても実用的な信頼基盤を提供できると考える．

参考文献

[1] Asokan, Nadarajah, et al. ”Seda: Scalable embedded device attestation.” Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015.

[2] Birkholz, Henk, et al. ”RFC 9334: Remote ATtestation procedureS (RATS) Architecture.” (2023).

[3] Chen, Guoxing, Yinqian Zhang, and Ten-Hwang Lai. ”Opera: Open remote attestation for intel’s secure enclaves.” Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019.

[4] Chen, Guoxing, and Yinqian Zhang. ”MAGE: Mutual attestation for a group of enclaves without trusted third parties.” 31st USENIX Security Symposium (USENIX Security 22). 2022.

[5] Li, Peixi, Xiang Li, and Liming Fang. ”DMA: Mutual Attestation Framework for Distributed Enclaves.” International Conference on Information and Communications Security. Singapore: Springer Nature Singapore, 2024.

[6] Petzi, Lukas, et al. ”SCRAPS: Scalable collective remote attestation for Pub-SubIoT networks with untrusted proxy verifier.” 31st USENIX Security Symposium (USENIX Security 22). 2022.

[7] Kwon, Hyunsoo. ”Secure and Scalable Device Attestation Protocol with Aggregate Signature.” Symmetry 17.5 (2025): 698.