TLS2VC: Web通信の真正性をVerifiable Credentialとして証明可能にする分散WebProof方式

姜 皓程[1,a)]　張 一凡[2,1]　松浦 幹太[1]

概要：Web3 や Self-Sovereign Identity（SSI）の普及に伴い，Verifiable Credential（VC）によるユーザ属性証明への関心が高まっている．VC を用いた証明は VC 発行者に起因するため，普及には多くの VC 発行者が必要であるが，既存発行者はまだ少ない．本研究では，ユーザに提示される Web 情報を証明として利用する WebProof と VC を連携させることを目指す．特に，既存の WebProof は単一 Notary による証明を行っており，結託による偽装リスクがあるが，本研究では，複数の Notary による署名を用いて TLS 通信の真正性と Notary 自身の正当性を確保できる VC 発行方式を提案する．また，安全性を満たしつつ必要最小の Notary 参加数の設計についても示す．これにより，既存の Web サービスから取得できる情報を Web3 や SSI での証明情報として活用する基盤の構築を後押しする．

キーワード：WebProof，Web 通信による VC 発行，TLSNotary，Web3

TLS2VC: A Decentralized WebProof Framework Enabling Verifiable Credential for TLS Sessions

Kotei Jiang[1,a)]　Iifan Tyou[2,1]　Kanta Matsuura[1]

Abstract: With the growing adoption of Web3 and Self-Sovereign Identity (SSI), interest is increasing in methods that allow users to present their attributes as Verifiable Credentials (VCs) to third parties. However, issuing VCs often requires multiple trusted issuers, while existing services are still limited. In this context, WebProof, which leverages web information shown exclusively to the user for VC issuance, is gaining attention. Conventional WebProof systems typically rely on a single Notary, creating a vulnerability under collusion. This research proposes a WebProof mechanism that, by employing signatures from multiple Notaries, ensures both the authenticity of TLS communication and the integrity of the Notaries themselves. We further design a method to compute the minimum number of Notaries required under a target security level, improving practicality. This enables the reuse of existing web information as trusted VCs in Web3 and SSI environments.

Keywords: WebProof，VC，TLSNotary，Web3

1. はじめに

1.1 背景

近年，Web 上の情報流通量の増加に伴い，取得情報を第三者に提示する二次流通の重要性が高まっている．例えばクレジットカード明細を提示して立て替え払いを申請することが発生するが，その際に金額改ざんの有無を検証できることが重要となる．米国 NIST もオンライン取引の信頼確立を標準化課題に挙げており [8]，金融残高提示は DECO プロトコルの代表的応用例とされる [23]．一般的な Web サイトでは運営者が Signed HTTP Exchange などにより送信情報を保証しない限り，利用者がページの真正性

[1] 東京大学生産技術研究所
Institute of Industrial Science, The University of Tokyo
[2] NTT 株式会社 社会情報研究所
Social Informatics Laboratories, NTT
[a)] jiangkt@iis.u-tokyo.ac.jp

を示す手段は限定的である．実際，米国金融機関でも正式にオープン API を導入している銀行は一部に過ぎず，多くのサイトでは外部利用者によるデータ取得や検証手段がない [13]．利用者が使えるスクリーンショットや HTML 保存に依存し，改ざん耐性や検証可能性に乏しい．

この課題に対し注目されているのが WebProof である．TLSNotary [3] に代表される WebProof は，ユーザーが取得した Web ページが改ざんされていないことを，Web サイトの協力なしに第三者へ証明する技術である．典型的には，サーバとクライアント間の通信内容を公証人（Notary）が保証する構成を取り，検証者は Notary の信頼に基づいて一次流通情報の検証が可能になる．

時を同じくして，分散型ウェブ（Web3） [6] の概念に基づき，分散型識別子（DID）や検証可能な資格情報（VC） [9] による情報証明が普及しつつある．VC は情報にデジタル署名を付与し，出所や改ざんの有無を第三者が検証可能にする形式であり，自己主権型アイデンティティ（SSI）の基盤技術である．ただし，VC 発行者が流通の起点となる SSI サービスはまだ少なく，多くの Web サイトは VC 化に必要な情報を提供しているにもかかわらず，VC 発行機能やインセンティブを持たないのが現状である．

1.2 目的

本研究では WebProof と VC を組み合わせ，Web から得た情報を個人属性や事象の証明に活用することを目的とした．これにより，DID/VC 手法において任意の Web サーバを証明者として利用でき，従来の証明者や証明情報の不足を補え，WebProof に対して標準形式を提示できる．従来，WebProof 技術では，ユーザーがある Web サイトから受け取った情報を改ざんせずに提示していることを第三者に証明するため，信頼できる単一の Notary の存在を仮定する構成が一般的であった．しかしこのような構成では，Notary がすべての検証者から信頼されていることを前提とせざるを得ず，検証者ごとに信頼モデルが異なる自己主権型アイデンティティ（SSI）や，非中央集権的設計思想である Web3 の考え方とは整合しにくいという課題があった．この問題を回避する手法として zkTLS のように，検証者自身が Notary の役割を果たす方式も提案されているが，これはリアルタイム検証に限定され，VC のような事後検証には適用できない．

そこで，本研究では WebProof によって取得した TLS 通信の正当性を VC 形式で発行・提示できる TLS2VC を提案する．ユーザーが Web ページから得た情報で自身の属性を証明する VC を作り，第三者に提示・検証できるようにしたい．そこで，単独 Notary の信頼性を前提としない構成を実現するために，複数 Notary の選出アルゴリズムを導入し，攻撃 Notary だけでは証明が成立しないことを確率的に保証する安全性を証明する．これにより，事後検証可能な VC を Web3 型の利用モデルにも適合する形で提示可能にする．

1.3 貢献

本研究の貢献は以下である．

- **TLS2VC の定式化**：サーバ非改修という前提の下で，Server，Notary，Holder および Verifier の役割とデータフローを定義し，既存の VC 利用のワークフロー（発行・提示・検証）に自然に接続できる設計方針を示した．
- **Notary 選出手法の提案と安全性証明**：VRF に基づく公開検証可能な選出と t-of-k 閾値型マルチ署名を組み合わせ，Trusted な単一 Notary への依存を排した Trustless Notary 選出構成を提示した．提示手法における攻撃成功事象を定式化し，Notary 選出手法で安全性を確保として，攻撃成功確率 ε を無視できるようにするために必要なノード数 k を計算可能にし，その数値例を示した．
- **実現可能性の検証**：端末負荷が大きい Proxy 通信と署名に着目し，プロトタイプ実装により性能負荷を検証し，バックグラウンド処理として動作できる負荷の低さと実現可能性を示した．

ここで Trustless とは，単一の中央集権的な Notary への依存を排除することを意味する．ただし，本研究における安全性は，全体の過半数の Notary が正直であるという前提に依存している点に注意する必要がある．

これらの貢献により，サーバ協力が得られない状況においても，TLS 通信の正当性を再提示可能な VC として提示でき，一部の Notary に故障や悪意が存在する場合でも t-of-k により健全性を保持し得る道筋を与える．

2. ビルディングブロック

2.1 TLS

TLS（Transport Layer Security） [18] は，インターネット上の通信を暗号化し，通信相手の正当性とデータの完全性を保証するために広く用いられているプロトコルである．TLS サーバアクセス時には，クライアントとサーバとの間で TLS ハンドシェイクによりサーバ証明書の提示や共通鍵の共有が行われる．この過程でクライアントは「サーバが提示する証明書は信頼された認証局 CA（Certificate Authority）により承認（署名）されていること」，「サーバが証明書に対応した秘密鍵を有していること」を検証することで，偽装サーバではないことを確認できる．ただし，鍵交換後の通信に対して改ざん検知を可能にする通信完了時のサーバ署名は付与されない．そこで，WebProof では Notary の保証によって，第三者は提示された通信文に改ざんがないことを検証できる．

2.2 デジタル署名

デジタル署名は，署名者が署名鍵（秘密鍵）でメッセージに署名し，検証者が対応する検証鍵（公開鍵）でその正当性を検証する暗号技術である．これにより，署名鍵を知らない者が有効な署名を生成出来ないことを保証する EUF-CMA 安全性や署名付きメッセージの改ざんを防ぐ integrity が保証される．

2.3 マルチ署名

マルチ署名（Multi-Signature）[2] [16] は，同一メッセージに対して複数の署名者が協力し，1 つの署名を生成する暗号技術である．最も基本的な形式では，n 人の署名者がそれぞれの秘密鍵で個別に署名を行い，それらをまとめて提出することで検証を行う「列挙型」がある．この方式は実装が容易である一方，署名サイズや検証コストが署名者数に比例するという課題がある．この問題を解決するために，MuSig2 のような各署名者の署名を 1 つに統合する効率的な手法が提案されている．さらに近年では，すべての署名者の参加を必要としない k-of-n の閾値型マルチ署名への関心が高まっており，Threshold BLS[1] などの手法によって，一部の署名者が不在でも署名生成が可能な耐故障性と柔軟性が実現されている．本研究では，このような閾値型マルチ署名を活用する．

2.4 DID/VC

自己主権型アイデンティティ（SSI） [20] は，ユーザーが自らの識別情報を完全に制御することを目指す概念であり，その実現手段として第三者が発行管理する ID ではなく非集権的な ID である DID（Decentralized Identifiers）とそれに基づく証明書である VC（Verifiable Credentials）を用いて身分や実績の証明を行う [9], [10].

DID は，中央集権的な機関に依存することなく個人や組織が自己管理の元でその ID（個人・組織の識別子）を生成・管理するための仕組みである．VC は，発行者が特定の主体に対してデジタル形式で発行する証明書であり，その改ざん検出と出所の証明にデジタル署名を用いる．証明者，証明対象の DID を指定することで関係性を記載できる．基本的な構成要素は以下の通りである．

- 発行者（Issuer）：VC を発行するユーザー.
- 所有者（Holder）：VC を保持・提示するユーザー.
- 検証者（Verifier）：VC の正当性を検証するユーザー.
- レジストリ（Verifiable Data Registry）：検証に必要な公開鍵を保持するデータベースや分散型台帳.

VC 発行では Issuer がデジタル署名を含めることで発行者の偽造や内容の改ざんを防ぎ，Holder が Verifier へ提示するときに再度 Verifiable Presentation（VP）として Holder のデジタル署名を含めることで ID のなりすましを防ぐ．Verifier は VDR に Issuer と Holder の公開鍵及び VC の失効情報を問い合わせて VP の検証を行う．

2.5 VRF（Verifiable Random Function）

VRF（Verifiable Random Function）[14] は，秘密鍵 sk と入力 x に対し，決定論的に $(r, \pi) \leftarrow \mathrm{VRF.Eval}_{sk}(x)$ を生成する．ここで r は乱数値，π はその正当性の証明である．次に対応する検証関数 $\mathrm{VRF.Ver}_{pk}(x, \pi, r)$ によって r の正当性を第三者が検証できる．VRF は，出力値が使用者により操作できず，かつ検証可能であるため，攻撃者の集中を回避でき，分散システムにおけるノード選出や公平な抽選に利用される．

3. 関連研究

3.1 WebProof の先行研究

TLS 通信の証明を行う WebProof プロトコルの代表的な手法として TLS Notary[3] やゼロ知識証明を用いた手法が提案されている．TLS は暗号化・認証・通信中の改ざん検知は可能だが，取得後データへの電子署名は付与されない．そこで TLSNotary は，クライアントと Notary が秘密計算（MPC）で TLS ハンドシェイクを共同実行し，受信した HTTPS ページの正当性を第三者に証明可能にする．このとき Notary はセッション鍵の一部シェアのみを保持し，通信内容を漏洩させずに「特定のページを受信した」事実の改ざん不可能な証跡を生成する．MPC を用いたプロトコル以外にも，ゼロ知識証明を用いることで，検証者に示したいクライアント受信データの一部データだけを選択的に開示し，他の情報を秘匿したまま開示データの正当性を検証者が検証可能にする手法も提案されている．WebProof プロトコルには主に CPU の信頼性に基づく Trusted Excution Environment(TEE) [4] モデル，秘密計算を行う MPC [22] モデル，中継者による証明を行う Proxy モデルの 3 種類のモデルがある．これらはどれも検証可能にする効果は同じであるが，どのように TLS の証明を作るかにおいて差が生じる．

WebProof プロトコルと提案手法に関連する既存手法の機能・安全性の比較を Table 1 にまとめた．既存の手法はいずれも Notary の信頼性を仮定するか，複数 Notary への対応を記載するにとどまり，その安全性の検討や証明がなされていない．

3.2 Trustless ノードによる合意とその安全性証明

特定のサービス・サーバへの信頼を前提としない Decentralized なコンセンサス参加者選出方法として Algorand [7] は，VRF を用いて，コンセンサス参加者（委員会）を秘密裏に選抜しつつ当選後には公開検証可能とすることで，攻撃パーティによる乗っ取りを防ぎつつ「過半数正直」である確率を高める．参加者数 k を十分大きく設定することで，攻撃者が過半数以上を占める確率を指数関数的に抑制でき

表 1　zkTLS プロトコルの比較
Table 1　Comparison of zkTLS Protocols

Technology	VC Format Compatible	Security Proofs	Trustless Notary
DECO (MPC+ZK) [23]	△ (VC 未考察)	✓ (UC ベースの安全性証明)	△ (分散 Notary の安全性証明なし)
TLSNotary [3]	△ (VC 未考察)	✗ (理論的な安全性証明なし)	△ (分散 Notary の安全性証明なし)
Reclaim Protocol [17]	✓	✗ (理論的な安全性証明なし)	△ (分散 Notary の安全性証明なし)
提案手法 TLS2VC	✓	△ (確率計算による安全性証明)	✓ (分散 Notary の安全性証明)

ることを Chernoff 型の集中不等式による証明によって示されている．一方，Ouroboros [5] は，台帳情報の半同期モデル下で適応的結託（fully-adaptive corruption）に対して安全保証を与える．VRF を使ったスロットリーダー選出と鍵が進化する前方秘匿署名（forward secure signatures）により，攻撃者がリアルタイムで切り替わっても過去の VRF 出力や署名が破られない安全性を確立している．これにより，通常の遅延や悪意による遅延環境下でもフォークの発生確率を極小に抑え，ブロックチェーンの安全性を確保している．これらの先行研究を参考に VRF による攻撃ノードの集中を防ぎ，確率的な安全性解析を本研究でも利用する．

4. 提案手法 TLS2VC の設計

提案方式では，以下の 4 者を想定する．

- TLS サーバ (Server): 通常の TLS 対応サーバ（署名付与などの機能追加不可）
- 状態証明者 (Holder)=TLS Client: TLS サーバにアクセスし，その内容を証明したい当事者
- 証明生成ノード (Notary)=VC Issuer: MPC や Proxy に代わり，TLS 接続の真正性検証に協力する複数の第三者ノード群（攻撃ノードが混在可能）
- 検証者（Verifier）: 提示された VC にある署名を検証し，VC の正しさを評価する利用者

Notary は通信内容ではなく，通信先の正当性（接続先が正しいこと）のみを検証する．

なお，信頼スコア s_i は各 Notary の過去の挙動に基づき定期的に更新されることを想定している．新規ノードには中立的な初期値を与え，過度な相互評価による偏りを抑制することで，信頼値に基づく選出の正当性をより高めることができる．

4.1 提案プロトコル

(1) **Notary 選出**：Holder は，TLS 通信の証明を依頼するため，まず Notary 群と先に接続し，接続情報を共有する．各 Notary は選出に応じて Proxy として Holder と Server の通信を中継し，その通信先と通信文に対して自身の秘密鍵で署名を生成し，Holder に返送する．図 1 は Notary との接続の概要を示す．

(2) **TLS 接続の確立**：Notary 選出後，Holder は各 Notary を経由して Server に対して，通常の TLS ハンドシェイクを行い，共通鍵を確立する．この過程で，Server は自身の証明書を提示し，Holder はその妥当性を検証する．本ステップは，従来の TLS プロトコルに従って実行され，Server 側の改修は不要である．

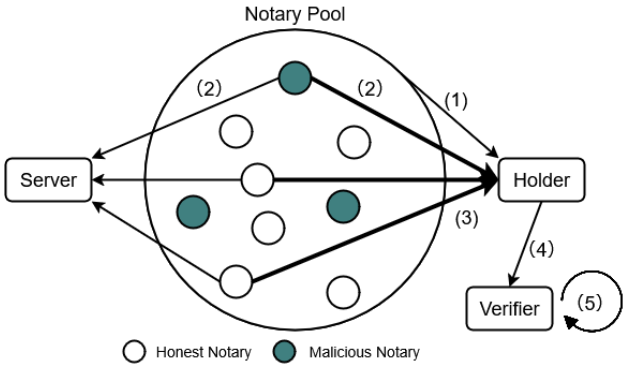

図 1　Notary 接続の概要
Fig. 1　Overview of Notary Connection

(3) **Notary 署名による通信先の正当性証明**：各 Notary は通信内容に対して Holder の DID 情報と自身の署名を付けて Holder に返送する．Holder は返送された署名を k-of-n のマルチ署名方式[*1]として集約する．このとき，一部に攻撃者が混在していても，閾値 k を満たす限り，通信先の正当性を十分に保証する．

(4) **Holder による通信内容の正当性提示**：Holder は，Server から受信した Web ページ（HTML 等）の通信文，TLS ハンドシェイクと得られる復号鍵をメッセージとして，Notary による署名をつけて 1 つの VC として構成する．Verifier 提示する際には自身の署名を付加して Verifiable Presentation（VP）を生成し，Verifier に提示する．

(5) **Verifier による検証**：Verifier は VP の署名により，提示者が Holder であることを確認し，VC 発行者である Notary の署名とそこに示される通信情報に基づいて通信内容と通信先の改ざん有無を検証する．さらに，通信情報に記録される TLS 鍵交換時のサーバ公開鍵情報に基づいて，Server との正当な通信であったかどうかを判定する．

[*1] 実際の TLS 鍵乱数が通信の度に異なるので厳密には同じ電文に対する署名にはならないが，一定数の証明が集まれば成立する署名方式として記載

4.2 提案手法の特徴と優位性

本研究で提案する WebProof 手法は，署名のみを用いて TLS 通信の真正性を証明し，その結果を Verifiable Credential（VC）として直接発行可能な軽量な証明方式である．本手法は，既存の TLS 通信環境に変更を加えることなく適用でき，オーバーヘッドが大きいゼロ知識証明（ZKP）を省略することで計算コストを抑えつつ，Malicious な Holder と Notary の結託が想定される状況でも安全な証明を実現する．これにより，以下のような特徴と優位性を有する．

- **ゼロ知識証明を使用しない軽量設計**：ZKP を用いずに通信内容を証明することで，Web ページの一部秘匿などができない代わりに ZKP で必要な計算負荷を大幅に削減し，Holder・Notary の処理負担を軽減できる．
- **TLS サーバの変更不要**：既存の TLS サーバを変更することなく適用可能であり，任意の TLS 通信に対する証明として TLSNotary などの先行手法と同等の利便性を持つ．
- **Malicious な証明協力者の許容**：複数の証明協力者（MPC ノードやプロキシ）の一部が Malicious であっても，マルチ署名により安全な証明が可能であり，TLSNotary などの先行手法よりも Notary の攻撃に対して高い安全性を確保できる．

これらの特徴により，提案手法は既存手法と比較して，安全性と実用性のバランスを保ちながら，TLS 通信の真正性を証明できる新たな WebProof 方式であることを示した．

5. Notary 選出手法

本章は，§4.1(2) における Notary の選出方法を形式化し，§4.1(3)(4) を踏まえて §4.1(5) における検証者視点での安全性を評価するものである．まず，後半で導入する重み付き方式との比較と解析を明瞭にするため，VRF のみを用いる単純方式をベースラインとして定式化する．次に，VRF に EigenTrust 等から得る動的信頼スコアとローテーションペナルティによる有界重みを併用する重み付き方式を提案し，攻撃ノードのみが選ばれる確率の低減を図る．さらに両方式について，t-of-k 閾値署名が成立するのに十分な正直ノード数を確率的に確保できることを示す安全性解析を行う．

5.1 VRF のみを用いた Notary 選出アルゴリズム

本節では重み付けを行わない単純方式（ベースライン）を提示する．まず，Holder は TLS 通信証明が必要なタイミングで時刻と通信先 Hash などから乱数入力 x を生成し，全 Notary に公開する．各 Notary i は自分の VRF 秘密鍵で

$$r_i = \mathrm{VRF.Eval}_{sk_i}(x)$$

を計算し，乱数 $r_i \in (0,1)$ を得る．

この VRF 出力に基づき上位 k 台を選出する．

5.2 VRF のみを用いた t-of-k 型選出の安全性証明

本節では，§5.1 の Notary 選出アルゴリズムで（3）の w_i による制御をせずに $f_i = r_i$ とした選出手法の安全性証明を行う．全体 N 台のノード中，攻撃ノード数を f とし攻撃率を

$$q = \frac{f}{N}\ (< \tfrac{1}{3})$$

とおく．本手法では，t-of-k 型の閾値署名により，少なくとも t 台の正直ノードが選出されることを要件とする．ここで求める確率

$$\Pr[\text{正直ノード数} \geq t] \geq 1 - \varepsilon$$

は，攻撃者が過半数を超えない前提で t-of-k 署名の安全性を保証するために必要な「十分な数の正直ノードを確保する」意義を示す．防止すべき攻撃事象は主に 2 つあり，

(1) 攻撃者ノードが t 台以上集まること
(2) 正直ノードが t 台集まらないこと

である．本証明では後者，つまり「正直ノード数 $< t$」の事象に特化して証明を行う．なお，ρ を選ばれた Notary k のうちの正直なノードの割合とし，$\rho \in (\frac{1}{2}, 1]$ として $t = \lceil \rho k \rceil > k/2$ を前提とすると，前者の事象（攻撃者ノード数 $\geq t$）は後者の事象（正直ノード数 $\leq k - t$）に含まれるため，後者を抑えれば両者を同時に防げる．

本節では，Algorand の証明に倣って Chernoff 型不等式により失敗確率の上界を与えるとともに，提案手法の運用では Holder が重複なく k 台を選出して検証を依頼する前提が成り立つため，この場合に適用できる超幾何分布によって攻撃成功確率を厳密に与える．

5.2.1 上界評価（Chernoff）

安全性要件として，「t-of-k 型閾値署名を成立させるために必要な正直ノード数 $t = \lceil \rho k \rceil$ を確保できない」という事象を失敗条件と定義する．これは正直ノード数 $< t$，すなわち攻撃ノード数 $X > k - t$ の場合に相当し，$\rho > 1/2$ のもとでは $X \geq (1-\rho)k$ とみなせる．

VRF により乱数入力が固定されると，各ノードは等しい確率で選ばれる．このとき k 台中に含まれる攻撃者の平均台数は kq（$q = f/N$）である．以下では，この平均を上回る偏りが生じる場合，すなわち $(1-\rho)k$ 個以上の攻撃者が選ばれてしまう場合の確率を上から抑える．この上界を適用するには，評価対象の閾値 $(1-\rho)k$ が平均 kq を上回ること，すなわち $\rho < 1 - q$ を仮定する．このとき，拡張 Chernoff [15] より，

$$\Pr\big[X \geq (1-\rho)k\big] \ \leq\ \exp\Big(-k\,D\big(1-\rho \,\|\, q\big)\Big)$$

ここで

$$D(a\|b) = a \ln \frac{a}{b} + (1-a) \ln \frac{1-a}{1-b}$$

は KL ダイバージェンスである．よって

$$k \;\geq\; \frac{\ln(1/\varepsilon)}{D(1-\rho\,\|\,q)}$$

を満たせば $\Pr[X \geq (1-\rho)k] \leq \varepsilon$ となり，正直ノード数 $\geq t$ の確率が $1-\varepsilon$ を超えて本手法の安全性が成立する．

5.2.2 厳密評価（超幾何分布）

本手法におけるノード選出は，全 N 台のうち攻撃ノード f 台を含めて重複なく k 台を無作為抽出する操作であるため，超幾何分布を使うことができ，このときの攻撃ノード数 X は

$$X \sim \mathrm{Hypergeometric}(N,\, f,\, k)$$

に従う．ここで，無作為・重複なしとは，Holder が VRF で固定された乱数に基づき Notary の属性を参照せずに母集団から一括で k 台を選出し，選出後に結果を見て入れ替える再抽選を行わないことを指す．この前提の下では，選出集合に含まれる攻撃者台数の分布が一意に定まり，攻撃成立の閾値に対する確率を以下のように評価できる．$t = \lceil \rho k \rceil$ とおくと，正直ノード数 $\geq t$ は $X \leq k - t$ と同値であり，

$$\begin{aligned}\Pr[\text{正直ノード数} \geq t] &= \sum_{x=0}^{k-t} \frac{\binom{f}{x}\binom{N-f}{k-x}}{\binom{N}{k}} \\ &= 1 - \sum_{x=k-t+1}^{\min\{k,\,f\}} \frac{\binom{f}{x}\binom{N-f}{k-x}}{\binom{N}{k}}\end{aligned}$$

よって，所望の ε に対して

$$\sum_{x=k-t+1}^{\min\{k,\,f\}} \frac{\binom{f}{x}\binom{N-f}{k-x}}{\binom{N}{k}} \;\leq\; \varepsilon$$

を満たすように k を選べば，正直ノード数 $\geq t$ の確率が $1-\varepsilon$ を超えて本手法の安全性が成立する．

5.2.3 数値例

数値例として，$q = \frac{1}{3}$, $\rho = \frac{1}{2}$, $\varepsilon = 2^{-30}$, $N = 1000$（$\rho < 1 - q$）とする．このとき，上界評価（Chernoff）では $k = 354$，厳密評価（超幾何分布）では $k = 226$ となる．

本手法は悪意ノード率 q が小さいことを前提としており，そのためには DID 等による身元保証や EigenTrust に基づく信頼スコア付けを併用することが想定される．これにより，Sybil 攻撃やノードの水増しを抑制し，提案手法の前提条件を現実的に支えることができる．

5.3 重み付き Notary 選出アルゴリズム

本節では新たに重み付きの Notary 選出アルゴリズムについて提案する．このアルゴリズムは以下のステップに基づいて Notary を選出する．

(1) Holder は TLS 通信証明が必要なタイミングで時刻と通信先 Hash などから乱数入力 x を生成し，全 Notary に公開する．各 Notary i は自分の VRF 秘密鍵で

$$r_i = \mathrm{VRF.Eval}_{sk_i}(x)$$

を計算し，乱数 $r_i \in (0,1)$ を得る．

(2) 各 Notary は，EigenTrust により与えられる信頼値 $s_i \in [0,1]$ を持つ．

(3) ローテーション指標は指数移動平均で更新する．

$$w_i^{(T+1)} = (1-\gamma)\, w_i^{(T)} + \gamma \cdot \mathbf{1}_{\{i \text{ が選出された}\}}, \qquad 0 < \gamma \leq 1$$

選出されなかった回では w_i は自然減衰し，一度も選ばれていないノードにも機会が与えられる．

(4) ローテーションを織り込んだ信頼指標を

$$\theta_i \;=\; s_i\,(1 - \lambda\, w_i), \qquad 0 \leq \lambda \leq 1$$

と定め，基本重みを

$$\alpha_i \;=\; 1 + a\,\theta_i, \qquad a > 0$$

とする．評価値は

$$f_i \;=\; \frac{-\ln r_i}{\alpha_i}$$

で計算する．s_i が大きく w_i が小さいほど α_i は大きくなり f_i は小さくなって，当選しやすくなる．連続選出で w_i が増えると θ_i が抑制され，同一ノードへの偏りが緩和される．

(5) すべての f_i を公開し，値の小さい順に上位 k 個を Notary として採用する．計算過程は第三者が再計算でき，不正があれば検知可能である．VC 発行時，Holder は次の情報を含む選出証明を添付する．

- 乱数入力 x
- 各 Notary の識別子，信頼値 s_i とローテーション指標 w_i，および θ_i，$\alpha_i = 1 + a\theta_i$
- 各 Notary の VRF 出力 r_i とその検証可能な証明
- 評価値 $f_i = \dfrac{-\ln r_i}{\alpha_i}$ に基づく，上位 k 個の Notary 一覧

Verifier はこれらを用いて選出過程を再計算し，正当性を確認できる．

本手法の意義

- 改ざん不能かつ検証可能な乱数生成: 入力 x は TLS 由来で Notary が操作できず，各 Notary の VRF 出力は公開検証可能である．
- 信頼とローテーションの考慮: s_i を用い，$\theta_i = s_i(1 - \lambda w_i)$ により過去の良好な実績を反映しつつ，選ばれ過ぎたノードの優先度を抑制する．

5.4 重み付き t-of-k 型選出の安全性証明

本節は，§5.3 の重み付き選出（$\theta_i = s_i(1 - \lambda w_i)$, $\alpha_i = 1 + a\,\theta_i$, $f_i = (-\ln r_i)/\alpha_i$）を前提とし，§ 5.2（VRF のみ）と同じ失敗事象（正直ノード数 $< t = \lceil \rho k \rceil$）に対する上界を与える．以下，記号や前提は §5.2 に従う．

5.4.1 上界評価（Chernoff）

重み付きの場合についても Chernoff 境界により攻撃者が過半数に達する確率を簡潔に上から抑える．重みの最悪ケースから当選確率の上限を置き，多数決の基準を当て込むことで，目標とする誤り上限に対して必要な選出台数を即座に見積もることができる．復元なし抽出は独立抽選よりばらつきが小さいため，独立抽選への置換で上界を求める [11]．

ここで，1 回の選出で攻撃ノードが選ばれる確率の上限を $\beta_{\max}$ とおく．例として，任意の悪意／正直の重み比が高々 $\omega_{\max}$ のときは

$$\beta_{\max} \;=\; \frac{\omega_{\max}\, q}{\omega_{\max}\, q + (1-q)}, \qquad q = \frac{f}{N}$$

と置く．また，元の仮定 $\alpha_i = 1 + a\,\theta_i$ $(0 \le \theta_i \le 1)$ より $\alpha_i \in [1,\, 1+a]$ の場合には $\omega_{\max} = 1 + a$ としてよい．

このとき拡張 Chernoff より

$$\Pr[X \ge u] \;\le\; \exp\!\Big(-k\, D\big(\tfrac{u}{k} \,\|\, \beta_{\max}\big)\Big), \qquad u \;=\; k - \lceil \rho k \rceil + 1$$

が成り立つ．適用条件は $\frac{u}{k} > \beta_{\max}$ であり，特に $\rho < 1 - \beta_{\max}$ を必要とする．よって

$$k \;\ge\; \frac{\ln(1/\varepsilon)}{D\big(1-\rho \,\|\, \beta_{\max}\big)}$$

を満たせば $\Pr[X \ge (1-\rho)k] \le \varepsilon$ となり，正直ノード数 $\ge t$ の確率が $1-\varepsilon$ を超えて本手法の安全性が成立する．

5.4.2 厳密評価（Wallenius の非中心超幾何分布）

重み付き選出は，各ノードに与えた重みの比で表せる非復元抽出として記述でき，Wallenius の非中心超幾何分布はこの過程に一致するモデルである．攻撃者側の重みを α_m，正直側の重みを α_h とし，重み比 $\omega = \alpha_m / \alpha_h$ を置けば，k 台選出に紛れる攻撃者台数とその発生確率を計算できる．これにより，攻撃者が多数決の閾値を超える確率を直接評価でき，Chernoff による上界よりタイトな k の設計が可能となる．攻撃ノード数 X は Wallenius の非心超幾何分布 [21]

$$X \sim \mathrm{WNCHypergeo}(N,\, f,\, k,\, \omega), \qquad \omega = \frac{\alpha_m}{\alpha_h}$$

に従う．

本分布の確率質量関数は

$$\Pr[X = x] = \binom{f}{x}\binom{N-f}{k-x} \int_0^1 \big(1 - t^{\omega/D}\big)^x \big(1 - t^{1/D}\big)^{k-x}\, dt$$

で与えられる．ここで，

$$D = \omega(f-x) + \big((N-f) - (k-x)\big)$$

$$\max\{0,\, k - (N-f)\} \le x \le \min\{f,\, k\}$$

$t = \lceil \rho k \rceil$ とおくと，正直ノード数 $\ge t$ は $X \le k - t$ と同値であり，

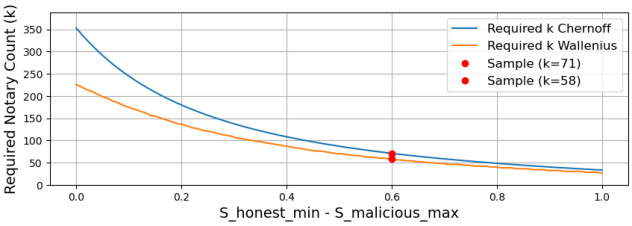

図 2　必要選出台数 k とスコア分離度の関係

Fig. 2　Required Committee Size k vs. Trust Score Separation

$$\begin{aligned}\Pr[\text{正直ノード数} \ge t] &= \sum_{x=0}^{k-t} \Pr[X = x] \\ &= 1 - \sum_{x=k-t+1}^{\min\{f,\,k\}} \Pr[X = x]\end{aligned}$$

したがって，所望の ε に対して

$$\sum_{x=k-t+1}^{\min\{f,\,k\}} \Pr[X = x] \;\le\; \varepsilon$$

を満たすように k を選べば，正直ノード数 $\ge t$ の確率が $1-\varepsilon$ を超えて本手法の安全性が成立する．

5.4.3 数値例

数値例として，$q = \frac{1}{3}$，$\rho = \frac{1}{2}$，$\varepsilon = 2^{-30}$，$N = 1000$，$a = 5.0$（$\rho < 1 - q$）とする．

攻撃能力を最大化するため，攻撃者が想定の上限 1/3 存在するときに，EigenTrust スコアの平均を正直な参加者＝`S_HONEST_MIN`，攻撃者＝`S_MALICIOUS_MAX` とすると，攻撃者が正直な参加者に対して持ちうる最大の重み比が定まり，これを用いて単一の Notary 選出における攻撃ノードの選出確率の上限値 $\beta_{\max}$ を算出できる．この上限値を Chernoff 境界の式に適用することで，Separation に応じた k を導出できる．ここで，`S_HONEST_MIN` − `S_MALICIOUS_MAX` は EigenTrust の有効性に相当し，0 の場合は効果がなく，1 の場合は確実に識別できることを意味する．

図 2 は，分離度 S に対する必要選出台数 k を示す．S が大きいほど必要 k は小さい．図中の「71」「58」は，マーカー位置における例である．

6. 実装検証

提案するシステムの実用性と運用上の制約を調べるため，システムの信頼性評価などを除く Notary の Proxy 処理の負荷測定を行った．実装内容は Notary 選出や情報共有を非集権的に行う場を LibP2P[12] ベースで実装し，そのうえで，通信の Proxy 処理とキャプチャを Go 言語と tcpdump によって実装した．リソース使用量の測定では gopsutil[19] を利用した．下記検証システム構成を示す．端末：Macbook Air M2,OS：MacOS 15.6,Mem：24GB，言語：Go1.23.11 darwin/arm64, ライブラリ：tcpdump 4.99.1-Apple 148.

Notary 負荷測定のために Notary 選出をスキップし，6

プロセスを立ち上げた状態での負荷測定を行った．クライアントからは検索サイトへのアクセスをリクエストし，その通信の転送を Notary に実施させた結果の処理負荷は表 6 の通りであり，リクエストを受ける前の CPU 負荷 0.05 % や Mem 負荷 7.25MB はともに現在の計算機やスマートフォン性能から見ても十分に小さい値であった．ただし，通信量については閲覧サイトに依存して増加するため，大容量のファイル送受信については中断機能が求められる．

表 2 Notary1 プロセスあたりの性能負荷

	測定値
CPU 負荷	0.05∼0.08%
Mem 使用量	7.25∼10.73MB
通信量	24KB

以上から提案システムで証明に協力する立場の Notary での負荷は証明相手が少なければ小さく，実用には影響が少ないことが確認できた．このことは Notary はビジネスのためのノード運用だけではなく，一般ノードも VC を Notary から取得する見返りとして，自分が Notary として参加するような利用も現実的であることを示している．よってより多くのノードが Notary として提案方式の選出に参加できることが期待される．

7. まとめ

本研究では，通信の正当性に着目し，サーバ非改修の前提で TLS セッションを Verifiable Credential（VC）として提示可能とする構成を提示した．TLS2VC の接続関係を定式化し，分散 Notary による構成と，所望の誤り確率に対する k の見積もりに基づく Notary 選出アルゴリズムの設計指針を示した．さらに，軽量プロトタイプにより実現可能性を確認した．

今後の展望としては，提案手法の実装を進めること，および安全性や設計指針等に関する理論研究をより発展させることを挙げる．

謝辞．本研究の一部は，JST 戦略的創造研究推進事業（CREST）JPMJCR22M1 の支援を受けたものである．

参考文献

[1] Bacho, R. and Loss, J.: On the Adaptive Security of the Threshold BLS Signature Scheme, Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22, New York, NY, USA, Association for Computing Machinery, p. 193–207 (online), DOI: 10.1145/3548606.3560656 (2022).

[2] Bellare, M. and Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma, Proceedings of the 13th ACM conference on Computer and communications security, pp. 390–399 (2006).

[3] Chase, M., Meiklejohn, S. and Zaverucha, G.: TLSNotary: Client-Server TLS Auditability without Server Modification, Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 841–852 (online), DOI: 10.1145/2660267.2660370 (2014).

[4] Costan, V. and Devadas, S.: Intel SGX explained, Cryptology ePrint Archive (2016).

[5] David, B., Gaži, P., Kiayias, A. and Russell, A.: Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, pp. 66–98 (2018).

[6] Gan, W., Ye, Z., Wan, S. and Yu, P. S.: Web 3.0: The future of internet, Companion Proceedings of the ACM Web Conference 2023, pp. 1266–1275 (2023).

[7] Gilad, Y., Hemo, R., Micali, S., Vlachos, G. and Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies, Proceedings of the 26th symposium on operating systems principles, pp. 51–68 (2017).

[8] Grassi, P. A., Garcia, M. E. and Fenton, J. L.: NIST Special Publication 800-63-3 digital identity guidelines, National Institute of Standards and Technology, Los Altos, CA (2023).

[9] Group, W. C. C.: Verifiable Credentials Data Model 1.0, `https://www.w3.org/TR/vc-data-model/` (2019).

[10] Group, W. D. I. W.: Decentralized Identifiers (DIDs) v1.0, `https://www.w3.org/TR/did-core/` (2022).

[11] Hoeffding, W.: Probability inequalities for sums of bounded random variables, Journal of the American statistical association, Vol. 58, No. 301, pp. 13–30 (1963).

[12] libp2p Foundation, T.: libp2p.

[13] Lin, X., Zhang, S. S. and Zachariadis, M.: Open data and API adoption of US banks, Journal of Financial Intermediation, Vol. 63, p. 101162 (2025).

[14] Micali, S., Rabin, M. and Vadhan, S.: Verifiable random functions, 40th annual symposium on foundations of computer science (cat. No. 99CB37039), IEEE, pp. 120–130 (1999).

[15] Mulzer, W.: Five proofs of chernoff's bound with applications, arXiv preprint arXiv:1801.03365 (2018).

[16] Nick, J., Ruffing, T. and Seurin, Y.: MuSig2: Simple two-round Schnorr multi-signatures, Annual International Cryptology Conference, Springer, pp. 189–221 (2021).

[17] Protocol, R.: Reclaim Protocol Docs — docs.reclaimprotocol.org, `https://docs.reclaimprotocol.org/`.

[18] Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446 (2018).

[19] shirou: gopsutil: psutil for golang.

[20] Tobin, A. and Reed, D.: The inevitable rise of self-sovereign identity, The Sovrin Foundation, Vol. 29, No. 2016, p. 18 (2016).

[21] Wallenius, K. T.: Biased sampling; the noncentral hypergeometric probability distribution, Technical report (1963).

[22] Yao, A. C.: Protocols for secure computations, 23rd annual symposium on foundations of computer science (sfcs 1982), IEEE, pp. 160–164 (1982).

[23] Zhang, F., Maram, D., Malvai, H., Goldfeder, S. and Juels, A.: Deco: Liberating web data using decentralized oracles for tls, Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1919–1938 (2020).