

E2EE メッセージングのための透明性と頑健性を備えた コンテンツモデレーション

稲垣凜太郎^{1,a)} 望月 理来^{2,b)} 赤間 滉星^{c)} 阿部 涼介^{2,d)} 鈴木 茂哉^{2,e)}

概要：本稿では，End-to-End Encryption (E2EE) を導入したメッセージングサービスにおいて，コンテンツモデレーションの透明性と頑健性を両立する手法を提案する．E2EE 環境ではサービス運営者は平文を知り得ないので，メッセージがポリシー違反かの判定が困難である．先行研究は判定基準を秘匿することで攻撃者によるモデレーションの回避を困難にしたが，恣意的な検閲等がないかユーザーが検証できない不透明さが批判された．本稿では，(i) 判定基準の完全公開とその運用の検証可能性を保証する透明性，(ii) ポリシー違反メッセージへの検知能力低下を抑える頑健性を同時に満たす手法を提案する．提案手法では，送信者は送信前に対象のメッセージ送信に対するシードを受け取り，このシードとメッセージで評価される判定関数への準拠をゼロ知識証明で示す．シードの予測不可能性により攻撃戦略を探索攻撃に限定し，レート制御により単位時間あたりの攻撃成功回数を制限する．プロトタイプ実装と評価により実現可能性を実証し，E2EE 環境において透明性を犠牲にすることなく実効的なモデレーションを実現する新たなアプローチを示した．

キーワード：エンドツーエンド暗号化，コンテンツモデレーション，ゼロ知識証明，透明性，プライバシー

Content Moderation with Transparency and Robustness for E2EE Messaging

RINTARO INAGAKI^{1,a)} RIKU MOCHIZUKI^{2,b)} KOSEI AKAMA^{c)} RYOSUKE ABE^{2,d)} SHIGEYA SUZUKI^{2,e)}

Abstract: This study proposes a content moderation scheme that achieves both transparency and robustness for End-to-End Encryption (E2EE) messaging services. In E2EE environments, service operators cannot access plaintext messages, making it difficult to detect policy-violating content. Prior researches have maintained robustness by concealing detection criteria, but this lack of transparency has been criticized as it prevents users from verifying the absence of arbitrary censorship. This study proposes a scheme that simultaneously satisfies: (i) transparency that guarantees full disclosure of detection criteria and verifiability of their operation, and (ii) robustness that suppresses degradation in the ability to handle policy-violating messages. In our proposed scheme, senders receive an unpredictable seed generated by the server for each message transmission and demonstrate compliance with the detection function determined by this seed using zero-knowledge proofs. The unpredictability of the seed limits attack strategies to brute-force searches, while rate limiting mathematically bounds the number of successful attacks per unit time. Through prototype implementation and evaluation, we demonstrate the feasibility of our scheme and present a novel scheme for achieving effective moderation in E2EE environments without sacrificing transparency.

¹ 慶應義塾大学環境情報学部
Environment and Information Studies, Keio University
² 慶應義塾大学大学院 政策・メディア研究科
Graduate School of Media and Governance, Keio University
³ 情報通信研究機構

National Institute of Information and Communications
Technology
a) inaridiy@sfc.wide.ad.jp
b) riku-m@keio.jp
c) akama@nict.go.jp
d) chike@sfc.wide.ad.jp

1. はじめに

健全なメッセージングサービスの運営において、コンテンツモデレーションは重要である。メッセージングサービスとは、テキストなどのチャットメッセージを個人間あるいはグループ間でやり取りできるサービスの総称である。メッセージングサービスは、健全な状態を維持するため、児童性的虐待コンテンツ (Child Sexual Abuse Material; CSAM) の拡散やテロ行為といった犯罪への利用を阻止する必要がある。そのためにサービス運営者は、サービス上で流通するメッセージがポリシーに違反していないかを判定し、必要に応じて削除などを行う一連のプロセスを実施する。このプロセスをコンテンツモデレーションと呼ぶ。

一方で、プライバシー保護への関心の高まりから、メッセージングサービスにエンドツーエンド暗号化 (End-to-End Encryption; E2EE) を採用する事例が増えている。E2EE では、メッセージを送受信者間で暗号化・復号することで、サービス運営者を含む第三者からメッセージの機密性を保証する。E2EE はサービス運営者や外部の攻撃者によるデータ漏洩、内部不正、サービスによる監視などの脅威から、ユーザーのメッセージを保護する。WhatsApp, Signal, iMessage を含む主要プラットフォームは現在、デフォルトで E2EE を提供し、世界中で数十億人のユーザーにサービスを提供している [1], [5], [21]。

しかし、E2EE の機密性は、従来のモデレーション手法の適応を困難にするため、新たな手法が求められる。サービス運営者はメッセージの平文にアクセスできないため、メッセージをサービス運営者のサーバー上で平文を基に直接判定する従来の手法は適用できない。結果として、E2EE メッセージングはポリシーに準拠した正当な利用と同様に、CSAM の拡散などの犯罪行為にも機密性を提供しうる。公共機関からは、捜査目的で暗号化されたメッセージにアクセスできるバックドアの導入などが要請されるが、これは E2EE の目的と対立する。

この課題に対し、E2EE 環境におけるモデレーション手法は多く提案されてきたが、モデレーションの透明性が不十分という理由で強い批判を浴びている。本稿における透明性とは、サービスが公示するポリシーから逸脱した手法の採用・運用がなされていないことを、ユーザーや第三者が独立に検証可能であるという性質を指す。2021 年に Apple が CSAM の検知を目的として提案した手法は、内部のアルゴリズムや運用方法が不透明だった [3]。そのため、ユーザーが Apple による恣意的な検閲を検証できない点が強く非難され、最終的に撤回された [2], [13], [16]。

先行研究の多くで透明性が限定的であった原因は、透明性を担保するとポリシーの違反のメッセージに対応でき

る割合を低下させるという二者択一の関係にある。本稿においてモデレーションの頑健性とは、悪意あるユーザーによるモデレーション回避の試みに対して、ポリシー違反のメッセージを検出・遮断する能力の程度を指す。モデレーションの手法や運用の詳細がユーザーから検証可能になるほど、悪意あるユーザーはその情報を基にモデレーションを回避する方法を事前に探索することが容易となる。結果として、ポリシーに反するメッセージであっても判定を欺く可能性が高まり、モデレーションの頑健性が低下する。

本稿では、主に機械学習モデルを判定に用い、E2EE メッセージにおけるメッセージの機密性・モデレーションの透明性を満たしたうえで、ポリシー違反のメッセージに対応できる割合の低下を抑えるコンテンツモデレーション手法を提案する。本手法のキーアイデアは、メッセージがポリシーを満たすか判定するメッセージ判定関数の実装や内部のパラメータをすべて送信者に公開したうえで、次の 3 つの工夫を加えたことにある。1 つは、送信者がサーバーに対して、メッセージが判定関数を満たすことを、ゼロ知識証明プロトコルを用いて平文をサーバーに明かすことなく証明することである。2 つ目は、送信者がコミットメントスキームを用いて送信前にメッセージを固定した後サーバーからシードを受け取り、このメッセージとシードのペアで判定結果を確定させる。これにより、悪意ある送信者が事前に判定結果を予測して回避策を探索することを難しくする。3 つ目は、シード要求に対するレート制御により、単位時間あたりの探索攻撃試行回数を制限することである。

本研究の貢献は主に以下の 2 点である。

- メッセージの機密性・モデレーションの透明性・探索攻撃への頑健性を同時に満たすコンテンツモデレーション手法を提案する。
- テキストメッセージを判定するプロトタイプを実装・評価し、提案手法の実現可能性を実証するとともに、実用化に向けた性能上の課題を明らかにする。

2. 前提知識

本章では、本稿で用いる暗号プリミティブと機械学習モデルの敵対的攻撃、ゼロ知識証明プロトコルに関する背景知識を提供する。

2.1 共有鍵暗号プリミティブ

共有鍵暗号は、暗号化と復号に同一の鍵 k を用いる暗号方式である。本稿では、メッセージ m と鍵 k を入力として暗号文 C を出力する暗号化関数を $Enc(k, m) \rightarrow C$ と表記する。対応する復号関数は $Dec(k, C) \rightarrow m$ であり、暗号文 C と鍵 k から元のメッセージ m を復元する。鍵 k を共有する当事者間でのみメッセージの機密性を保証できる。

e) shigeya@wide.ad.jp

2.2 コミットメントスキーム

コミットメントスキームとは、ある値を事前に固定しながらその内容を秘匿し、後に必要に応じてその値を開示できる暗号プリミティブである。本稿では、値 m と乱数 r からコミットメント値 h を生成する関数を $\text{Commit}(m, r) \rightarrow h$ と表記する。このスキームは主に 2 つの性質を持つ。コミットメント値 h から元の値 m を知ることが困難である隠蔽性と、一度 h を公開した後に同じ h を生成する別の値 m' を見つけることが困難である拘束性である。

2.3 機械学習モデルの敵対的攻撃に対する頑健性と対策

機械学習モデルなどを用いた分類器には、敵対的攻撃と呼ばれる入力に微細な編集を加えて分類器に意図的な誤分類を誘発する攻撃手法が存在する [8]。攻撃の容易さは攻撃者の知識量に強く依存し、モデル内部のパラメータにアクセスできる場合、攻撃者はモデルの勾配情報などを利用して効果的な編集を求めることができる [18]。この攻撃は、ルールベースなどの任意の分類器に対しても、入力操作による誤判定の誘発という観点で共通の課題といえる [6]。

本稿では、分類器を分類対象 x を入力として分類結果 c を返す決定的な関数として定義する。

$$\text{Class}(x) \rightarrow c \quad (1)$$

分類器 Class の敵対的攻撃に対する頑健性を議論するため、2 つの定量的な指標を定義する。分類器の頑健性とは、攻撃者が入力を操作して分類結果を操作する試みに対し、分類器がどの程度の耐性を持つかの指標である。 $P_{\text{nat}}(\text{Class})$ は、自然な判定対象 x の分布における偽陰性率で、モデルの標準的な性質を示す。一方、 $P_{\text{adv}}(\text{Class})$ は、攻撃者が自身の知識の範囲で攻撃成功率を最大化した判定対象 x に対する偽陰性率であり、この値が低いほど分類器は敵対的攻撃に頑健といえる。

分類器の頑健性を向上させる方法として、分類器に確率的な振る舞いを導入し、攻撃者がその挙動を事前に予測困難にするアプローチが存在する。RanMASK [24] と呼ばれる手法では、文字列の分類において、攻撃者の知識によらず、一定の編集距離内での攻撃に対する頑健性を数学的に保証する。このように、特定の脅威に対して頑健性を数学的に保証する手法は認証付き防御と呼ばれ、広く研究されている [11]。

このような確率的な振る舞いを持つ分類器は、以下のような抽象的な関数としてモデル化できる。この関数は、分類対象 x と、分類器の確率的な振る舞いを一意に決定するためのシード s を入力として受け取る。

$$\text{SeedClass}(x, s) \rightarrow c \quad (2)$$

この関数は、同じ入力ペア (x, s) に対して決定的に振る舞う。しかし、攻撃者が分類時に使用されるシード s を事前

に知ることができなければ、攻撃者の視点では分類器の出力は x に対し予測不可能な振る舞いを示すことになる。

2.4 ゼロ知識証明プロトコル

ゼロ知識証明 (Zero-Knowledge Proof; ZKP) は、証明者が検証者に対して、ある命題が真であることを、その命題が真であること以外の情報を一切明かすことなく証明する暗号プロトコルである [22]。

ZKP プロトコルは、証明者と検証者の 2 者間で実行され、証明対象の命題を $R(x, w) \rightarrow \{\text{Accept}, \text{Reject}\}$ と表記する。ここで、 x は公開入力、 w は秘密入力 (witness) と呼ばれ、証明者は $R(x, w) = \text{Accept}$ となる w を知っていることを、 w を隠したまま検証者に証明する。実装上、この命題 R は R1CS などの算術回路で表現され、ゼロ知識回路 (ZK 回路) と呼ばれる。

ZKP は、完全性・健全性・ゼロ知識性の 3 つの性質を備える [22]。完全性は証明者が持つ命題 $R(x, w) = \text{Accept}$ が真のとき、検証者はそれを必ず真であるとわかる性質を指す。健全性は証明者の持つ命題 $R(x, w) = \text{Accept}$ が偽のとき、不正な証明者が検証者に誤った証明を受理させる確率は無視できるほど低いことを意味する。ゼロ知識性は $R(x, w) = \text{Accept}$ であること以外、検証者は w に関する情報を得られないことを保証する。

本稿では、ZKP に関して以下の 3 つの抽象的な関数を定義する。

- $\text{ZKSetup}(R) \rightarrow \text{crs}$: 命題 R に対する ZKP の公開パラメータ (Common Reference String; CRS) crs を生成する。
- $\text{ZKProve}(\text{crs}, x, w) \rightarrow \pi$: $R(x, w) = \text{Accept}$ となる w を知るところを主張する π を生成する。このとき、 π から w は導出できない。
- $\text{ZKVerify}(\text{crs}, x, \pi) \rightarrow \{\text{Accept}, \text{Reject}\}$: 証明 π を検証する。

多くの実用的な ZKP プロトコルは、証明 π のサイズと ZKVerify の計算コストが、命題 R の複雑さに対して対数オーダーまたは定数オーダーであるという性質を持つ [9]。

3. 関連研究における透明性の課題

E2EE 環境におけるコンテンツモデレーションは、E2EE の機密性と健全なサービス運営の両立を目指す上で重要な研究領域である。Scheffler らの SoK 論文 [20] で示したように、クライアントサイドスキャン、メタデータ解析、暗号プロトコルの応用など、多様なアプローチが提案されてきた。しかし、同研究でも強調されている通り、多くの手法はモデレーションの透明性に重大な課題を抱えている。モデレーションの仕組みや判定基準が不透明であると、サービス運営者による恣意的な検閲や、政府からの圧力による監視システムへの転用といったリスクが生じる。そのため、

ユーザーや第三者が判定アルゴリズムの実装やパラメータを検証し、公開されたポリシーに基づいて運用されていることを独立に確認できる透明性が重要とされる。

この課題を象徴するのが、Apple が提案した CSAM 検知のためのクライアントサイドスキャン手法である [3]。この手法は、既知の CSAM のハッシュ値からなる秘密集合とユーザーのデバイス上の画像を、Private Set Intersection (PSI) などの暗号技術を用いて照合する [4]。このアプローチは、サーバーに平文を送信しない点で一定の機密性を提供するものの、判定基準となるハッシュ集合が非公開であるため、何をモデレーション対象としているかを外部から検証できない。この不透明性は、「スリッパリー・スローブ問題」、すなわち将来的に政治的発言など CSAM 以外のコンテンツの検閲に悪用されかねないという懸念を呼び、多くの研究者や市民社会団体から厳しく批判された [2], [13]。結果として、判定基準を秘匿することで頑健性を保とうとする設計は、透明性を著しく損なうという根本的なトレードオフに直面してきた。

本稿と技術的なアプローチの類似性を持つ研究として、暗号化された通信内容をゼロ知識証明を用いて検証するクライアントサイドゼロ知識証明手法が存在する。この手法をネットワークミドルボックスで適用した ZK-Middlebox [10] では、暗号化されたパケットを機密性を保ちつつ中継プロキシ側で検証する点で本研究と共通のアプローチを持つ。しかし、その主目的はネットワークポリシーの適用などであり、メッセージングサービスにおけるモデレーションの透明性と、それに伴う探索攻撃への頑健性を両立させるという課題に直接取り組んだものではない。総括すると、モデレーションの透明性には、(1) 仕組みの公開、(2) 秘密集合や機械学習モデルのパラメータ等の判定基準の公開、(3) 公開情報に基づいた運用が行われているかの検証可能性、という複数の側面が含まれる。我々の知る限り、これら全ての側面で透明性を満たしつつ、モデレーションの頑健性を両立させた先行研究は存在しない。

4. 設計目標と脅威モデル

本章では、E2EE メッセージングにおけるモデレーション手法に求められる要件を整理する。まず E2EE メッセージングとモデレーションシステムの基本的なモデルと満たすべき性質を提示する。次に、送信者と運営者の両方が悪意を持つ可能性を考慮した脅威モデルを提示し、本稿で対処すべき攻撃シナリオを特定する。

4.1 システムモデル

本稿では、E2EE メッセージングとモデレーションシステムを送信者・サービス運営者のサーバー・受信者の三者モデルとして考える。送信者と受信者は共有鍵 k を事前に安全に共有しており、メッセージ m は暗号化関数

$Enc(k, m) \rightarrow C$ により暗号文 C に変換されてサーバー経由で送信される。受信者は復号関数 $Dec(k, C) \rightarrow m$ により平文を復元する。サーバーは暗号文 C を中継する過程で、その平文 m に直接アクセスせず m がポリシーを満たすか判定し、満たさない場合は中継しない。この判定には、メッセージ m を入力とし、判定結果を返す決定的な関数 $EvalMessage(m) \rightarrow \{\text{Accept}, \text{Reject}\}$ を用いる。

4.2 システム要件の目標

本稿では、E2EE メッセージングにおけるモデレーション手法が満たすべき要件を以下に定義する。

要件 R-1 (メッセージの機密性)。メッセージの平文 m は、判定関数による評価結果を除き、サービス運営者に対して一切の情報を漏洩しないこと。すなわち、サービス運営者は暗号文 C からメッセージの内容に関する情報を得ることができず、判定結果のみを取得可能であること。

要件 R-2 (モデレーションの透明性)。モデレーションシステムのアルゴリズムおよび判定関数を含むシステムコンポーネントの挙動が、送信者を含む第三者にとって既知であること。これは、プロトコルの仕様、実装、および機械学習モデルのパラメータを含む全ての構成要素が公開されることで実現される。さらに、実際のモデレーションが公開された仕様に従って実行されていることを、ユーザーおよび第三者が独立に検証可能であること。

要件 R-3 (モデレーションの頑健性)。モデレーションシステムが透明性を保証するとき、悪意ある送信者によるポリシー違反メッセージの送信成功率を制限する機構を備えること。提案手法における具体的な制限の程度については、第 8 章のセキュリティ解析で定量的に示す。

なお、本稿では以下を考慮しない。

- 受信者が送信者やサービス運営者と結託すること
- ポリシーに反するメッセージの平文を、運営者に開示すること
- 判定関数そのものの提案や改善

4.3 脅威モデル

本稿では、送信者と運営者の両方が、プロトコルから逸脱する場合も踏まえた、悪意ある参加者となりうることを想定する。

本稿で想定する悪意ある送信者の目的は、サーバーが本来中継してはいけないメッセージを送信することである。これには、2 つの攻撃ベクトルが考えられる。1 つは、クライアントアプリケーションの改造などによって判定関数そのものを回避する、プロトコル逸脱攻撃である。もう 1 つは探索攻撃である。これは、本来は判定関数を通過しないメッセージを様々に編集・改変することで、判定関数を欺くメッセージを見つけ出す攻撃である。

また、悪意ある運営者の目的は、メッセージの平文に関

する情報を不正に取得しメッセージの機密性を破ることと、判定関数などをユーザーが検証できない形で変更しモデレーションの透明性を損なうことである。

5. 透明性と頑健性を両立するモデレーション手法の設計

本章では、メッセージの機密性・モデレーションの透明性・モデレーションの頑健性を同時に満たすモデレーション手法の設計について述べる。本手法のキーアイデアは、クライアントサイドゼロ知識証明手法を基礎としつつ、モデレーションの透明性を確保した際に生じる探索攻撃への頑健性を向上させることである。

本手法では、サービス運営者は送信者に対しモデレーションシステムのプロトコルの仕様及びシステムコンポーネントの挙動を公開する。これには、実装コード、判定関数内部の機械学習モデルのパラメータ、後述する ZK 回路等、システムの動作にかかわるすべての要素が含まれる。送信者はメッセージ送信時に、ZKP を用いてメッセージが公開された判定関数を満たすことを、メッセージの平文を隠したままサーバーに証明する。これにより、悪意ある送信者によるプロトコル逸脱攻撃に対応できる。同時にサーバーは公開された判定関数を満たすこと以外の情報がわからないので、必然的に公開された情報に基づいたモデレーションしか行えない。

5.1 探索攻撃への対処

モデレーションシステムが完全に公開されると、悪意ある送信者は送信前に任意のメッセージについて判定結果を計算できるため、検知を回避する内容へと反復的に編集することが可能となる。本手法はこれに対処するため、送信者がメッセージの送信前に判定結果を確実な予測を困難にし、メッセージ送信そのものの試行回数にも制限を加える。

具体的には、判定関数 $EvalMessage(m)$ を、シード s を追加の入力とする関数 $SeedEvalMessage(m, s) \rightarrow \{Accept, Reject\}$ へと拡張する。この関数は、2.3 節で述べた確率的分類器 $SeedClass$ の考え方にに基づき、同一のペア (m, s) に対しては決定的に動作する。しかし、シード s が未知の場合、送信者にとってその出力は確率的に振る舞うため、判定結果の事前予測が困難となる。

さらに、送信者がメッセージ m を送信する際、送信前にコミットメントスキームを用いてコミットメント $Commit(m) = h$ を作成し、サーバーに h を送信する。サーバーはコミットメント h に対し、送信者が事前に予測できないランダムな値 s を生成し、送信者に返却する。送信者は、この (m, s) のペアに対して $SeedEvalMessage$ を評価したことをサーバーに証明する。コミットメントの拘束性により、送信者はシード s を観測した後に、コミットメント h に対応するメッセージ m を変更することが計算

量的に困難となる。この設計により、1 回の送信試行における攻撃成功確率は、判定関数自体の敵対的攻撃に対する頑健性 $P_{adv}(SeedEvalMessage)$ と同等になる。

しかし、この設計のみでは悪意ある送信者が判定を通過するまで繰り返し試行する探索攻撃を防ぐことはできない。そこで本手法では、送信者ごとにシード要求に対してメッセージが送信されなかった件数に単位時間当たりの上限 Q を設ける。これは、メッセージ m に対して発行されたシード s が、 $SeedEvalMessage$ の評価を満たさなかった件数に対応する。この制限によって、単位時間当たりの攻撃者の攻撃成功数は平均して、 $p := P_{adv}(SeedEvalMessage)$ とおくと $\frac{p}{1-p} Q$ に制限できる。

6. 提案手法のプロトコル

本章では、前章の設計に基づき、提案手法を構成するシステムコンポーネントと、サービス開始前に行うセットアップ手順、そして実際のメッセージ送信プロトコルについて詳述する。

6.1 サーバー運営者によるセットアップ

初めに本手法では、サービス運営者がメッセージ判定関数 $SeedEvalMessage$ と暗号プリミティブ $Enc, Dec, Commit$ を選定・実装する。次に、サービス運営者はこれらを用いて後述の ZK 回路 R を構築した後、 $crs = ZKSetup(R)$ を実行して公開パラメータ crs を生成する。最終的に、 $SeedEvalMessage, Enc, Dec, Commit, R, crs$ の実装をユーザーに公開する。

6.2 ゼロ知識証明回路の構成

上記の判定関数と暗号プリミティブを用いて、サービス運営者は以下の ZK 回路 R を構築する。この回路は、送信者がメッセージ m の平文を秘匿しながら、そのメッセージとシード s が判定関数 $SeedEvalMessage$ を満たすこと証明可能にする。

回路 R は公開入力 (C, h, s) と秘密入力 (k, m, r) を受け取り、以下の条件をすべて満たす場合に $Accept$ を出力する：

$$R : ((C, h, s), (k, m, r)) \rightarrow \{Accept, Reject\}$$

$$\text{where } \begin{cases} Enc(k, m) = C \\ Commit(m, r) = h \\ SeedEvalMessage(m, s) = Accept \end{cases} \quad (3)$$

ここで、公開入力 x は暗号化されたメッセージ C 、メッセージコミットメント h ($Commit(m, r)$ の結果)、およびサーバーが発行したシード s から構成される。秘密入力 w は暗号化鍵 k 、平文メッセージ m 、およびコミットメント用ランダム値 r から構成される。

6.3 メッセージ送信プロトコル

本節では、提案手法におけるメッセージ送信の具体的なプロトコルについて説明する．前提として、前節のセットアップが完了し、送信者と受信者間で暗号鍵 k が安全に共有されているものとする．プロトコルは以下の3つのステップから構成される．

Step 1: コミットとシード取得

送信者はメッセージ m の送信に先立ち、サーバーにメッセージのコミットメント h を送信し、シード s を取得する．まず、送信者は送信予定のメッセージ m と乱数 r を選択し、コミットメント $h = \text{Commit}(m, r)$ を計算する．次に、送信者はこの h をサーバーに送信してシードを要求する．サーバーは、この送信者の単位時間あたりのシード要求数からメッセージの送信数を引いた値が Q に達していないことを確認する．達していない場合 h に対し暗号学的に安全な乱数生成器を用いてシード s を生成し、送信者に返送する．上限に達している場合、サーバーは要求を拒否する．

Step 2: 証明生成

送信者はメッセージがポリシーに準拠することを証明する．送信者はメッセージ m とシード s を用いて $\text{SeedEvalMessage}(m, s)$ を計算し、判定結果を確認する． $\text{SeedEvalMessage}(m, s) = \text{Accept}$ の場合、送信者は暗号文 $C = \text{Enc}(k, m)$ を生成し、ゼロ知識証明 $\pi = \text{ZKProve}(crs, (C, h, s), (k, m, r))$ を生成する．一方、 $\text{SeedEvalMessage}(m, s) = \text{Reject}$ の場合は送信を中断する．

Step 3: メッセージ送信と検証

送信者は (C, h, s, π) をサーバーに送信する．サーバーは受信後、次の検証を順に行う．まず、 $\text{ZKVerify}(crs, (C, h, s), \pi) = \text{Accept}$ を実行し、証明 π を検証する．次に、シード s がコミットメント h に対して発行した未使用のシードであることを検証する．すべての検証に成功した場合、サーバーは暗号文 C を受信者に転送する．いずれかの検証に失敗した場合はメッセージを破棄する．

7. プロトタイプの実装と評価

本章では、提案手法の実現可能性と性能を評価するため、判定関数に Naive Bayes を用いたプロトタイプを実装し、ベンチマークを行った．実装は、メッセージ判定関数、暗号プリミティブ、そしてそれらを検証するゼロ知識証明回路から構成される．

7.1 判定関数: シード付き Naive Bayes

プロトタイプにおける判定関数 SeedEvalMessage には、簡単なトークナイザーとスパムフィルタ等で広く採用実績のある Naive Bayes 分類器 [14], [19] を改造して用い

表 1 評価環境

項目	スペック
CPU	Intel Core Ultra 7 265U
メモリ	32 GB DDR5
OS	Ubuntu 22.04 on WSL2
zkVM	OpenVM v1.2.1

た．この分類器は、メッセージに含まれる単語の出現頻度に基づき、そのメッセージがスパムか正常かなど、どのクラスに属するかのスコアを計算できる．

本プロトタイプでは、この Naive Bayes 分類器を 2.3 節で述べたようなシードにより決定される確率的な振る舞いをするよう改造し、 SeedEvalMessage として実装した．具体的には、シード s も受け取るよう拡張を施し、シード s から単語ごとに摂動 $\Delta_w(s)$ を疑似乱数的に生成し、モデルの重みに加算する．これにより、 SeedEvalMessage に求められる性質を満たす．

モデルの学習には、2 万件以上の SNS 投稿からなる HateXplain データセット [12] を用い、テストデータに対して約 88% の分類精度を達成した．

7.2 暗号プリミティブとゼロ知識証明回路

ZK 回路の実装には、Rust 言語等で回路を実装できる zkVM の実装の 1 つである OpenVM [17] を用いた．実装した回路は、メッセージに対して 6.2 節で定義した関係 R 、すなわち暗号文の正当性、コミットメントの正当性、メッセージと与えられたシードに対する SeedEvalMessage の判定を一括で証明できる．

暗号プリミティブについては、共有鍵暗号にはワンタイムパッド (one-time pad) を、ハッシュ関数には SHA256 を採用した、これらは ZK 回路内での実装が単純であり、プロトタイプの実現性を迅速に検証するために採用した．実運用環境では、共有鍵暗号には AES-GCM のような標準的な認証付き暗号の採用が望ましい [7] ．

7.3 ベンチマークと評価

本手法の性能特性を検証するため、プロトタイプに対し送信者側の ZKProve にかかる時間とサーバー側の ZKVerify にかかる時間を、メッセージ長を変化させながら計測した．評価環境の詳細は表 1 に、ベンチマーク結果は図 7.3 に示す．

ベンチマーク結果から、以下の特性が確認された．まず、 ZKProve に要する時間はメッセージ長にほぼ比例して線形に増加し、400 文字のメッセージで 8.9 秒を要した．これは、ZK 回路内で実行されるトークナイズ処理や Naive Bayes の計算量、暗号化処理などがメッセージ内の単語数に依存するためである．一方、サーバー側の検証時間は、ZKP の簡潔性によりメッセージ長に依存せず、約 0.6 秒とほぼ一定であった．この振る舞いは ZKP の理論的な性質

と整合しており、性能は想定範囲内であった。

しかし、証明生成に要する 8.9 秒という時間はユーザー体験を損なう可能性があり、検証時間の 0.6 秒もサーバー側で大量のリクエストを処理するには無視できないコストである。特に、計算資源の乏しいモバイル端末での実行や、将来的により表現力が高いがより複雑なモデルを判定関数に採用することを想定すると、証明時間は実用上の大きな障壁となりうる。

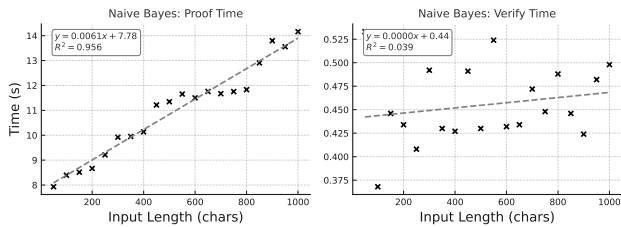


図 1 メッセージ長と証明・検証時間の関係

8. セキュリティ解析

本章では、提案手法が第 3 章で定めたセキュリティ性質を満たすことを簡潔に解析する。

8.1 メッセージの機密性

送信者がサーバーに送信するのは、公開入力 (C, h, s) と証明 π だけである。証明 π は ZKP のゼロ知識性により ZK 回路 R を満たすこと以外の情報を漏らさないため、 π から平文 m や鍵 k は一切漏洩しない。また、暗号文 C は共有鍵暗号で保護されており、鍵 k を持たないサーバーは復号できない。さらに、コミットメント $h = \text{Commit}(m, r)$ からは、 m の中身を導出できない。以上より、サーバーが知り得るのはシード要求、対応するシード s を発行した、 (C, h, s, π) が届き、検証が Accept/Reject だったという事実のみである。

8.2 モデレーションの透明性

本手法は、仕様と運用の両方を送信者や第三者が直接検証できる。具体的には、判定関数 SeedEvalMessage の実装や内部の機械学習モデルのパラメータなど、ZK 回路 R 、実装コード、および検証に必要な公開パラメータを一般に公開しモデレーションの仕組みと判定基準の透明性を確保できる。サーバーが裏で判定関数や内部のパラメータを変えても、公開された情報と整合しない限り π の検証は通過しないため、不正は表面化する。

8.3 モデレーションの頑健性

本手法は、モデレーションの透明性がある中でもポリシー違反メッセージへの対応能力の低下を抑える。まず、

送信者は先に $h = \text{Commit}(m, r)$ をサーバーに提出するため、シード s を見してから都合のいい m を差し替えることはできない。また、 $\text{SeedEvalMessage}(m, s) = \text{Accept}$ でなければ、ZKP の健全性により π を作れないため、判定関数そのものを回避できない。さらに、 s はコミット後にサーバーが選ぶので、送信者は事前に判定結果を確定できない。攻撃者が取り得る戦略は、判定関数を通過しうるメッセージ m を事前に作成し、サーバーから与えられるシード s で判定が通ることを期待する探索攻撃に実質的に限定される。

この探索攻撃において、探索の試行毎の攻撃成功確率は判定関数の敵対的攻撃に対する頑健性と一致する。シード s はメッセージのコミット後にサーバーで生成され、攻撃者には予測不可能であるため、各試行は独立した確率事象と見なせる。たとえ攻撃者が判定関数の内部ロジックやパラメータを完全に把握していたとしても、シードのランダム性により、あるメッセージ m が受理される確率は判定関数の敵対的攻撃に対する頑健性 $p := P_{\text{adv}}(\text{SeedEvalMessage})$ を超えることはない。

加えて、本手法はシード要求回数に制限を加えることで、探索攻撃の試行回数そのものを制限する。サーバーは、送信者が単位時間のシード要求回数からメッセージ送信回数を引いた値を送信失敗とみなし、この回数に上限 Q を設ける。単発試行の成功確率が p であるとき、1 回の成功を得るために期待される失敗回数は $(1-p)/p$ 回となる。そのため上限 Q の存在により、攻撃者が単位時間あたりに成功させられるメッセージ送信回数の期待値は、最大でも $\frac{p}{1-p}Q$ 回に制限される。

以上を総合すると、本手法は透明性を完全に確保した設定においても、判定関数自体の頑健性 p を継承し、さらにレート制御 Q によって時間あたりの攻撃成功回数を制限する。具体的な頑健性の数値見積もりと実用上の性能課題については次章の 9.1 節で議論する。

9. ディスカッション

9.1 頑健性の見積もりと限界

本手法の頑健性は、判定関数 SeedEvalMessage の単発試行に対する攻撃成功確率 $p_{\text{adv}} = P_{\text{adv}}(\text{SeedEvalMessage})$ に依存する。これに対し、認証付き防御 [24] のような理論的な保証を持つ手法から、RAPID ら [23] が報告する実用的な手法まで、判定関数を完全公開しても頑健性を維持する様々な方法が研究されている。特に後者は AGNEWS データセットで攻撃成功率を約 14% まで抑制した。

次に、これらの値を基に本手法の頑健性を試算する。通常のユーザーが 1 日 200 件送信し 10% が偽陽性で失敗する場合、送信失敗許容回数 $Q = 20$ と設定でき、 $p_{\text{adv}} = 0.14$ なら悪意ある攻撃者の 1 日あたりの攻撃成功回数期待値は $\frac{0.14}{1-0.14} \times 20 \approx 3.2$ 回となる。この結果は、従来手法が直面していた透明性と頑健性の二者択一を大幅に緩和したとい

える．

9.2 VRF の導入による透明性のさらなる向上

現在の設計では，サーバーがシード生成過程を恣意的に操作する可能性が残されている．この問題に対し，検証可能ランダム関数 (Verifiable Random Function; VRF) [15] の導入により，シード生成の透明性を更に向上させることができる．

VRF を用いた拡張では，サーバーは秘密鍵 sk_{vrf} を保持し，コミットメント h に対して VRF を計算することで，決定的かつ検証可能な方法でシード $s = VRF_{sk_{vrf}}(h)$ を生成する．同時に，各シードに対する証明 π_{vrf} も生成し，送信者に提供する．送信者は公開鍵 pk_{vrf} を用いて，受け取ったシードが正しく生成されたことを検証できる．

10. 結論と今後の課題

本稿では，E2EE 環境下でのコンテンツモデレーションを実現する新たな手法を提案した．従来の手法が直面していたモデレーションの透明性と頑健性の間のトレードオフに対し，本稿ではゼロ知識証明とサーバー発行シードを組み合わせた新しいアプローチを提示した．

今後の研究課題として，以下が挙げられる：

- (1) 性能の最適化：zkML 技術の活用や，より署名速度に特化した技術を用いて，証明生成時間の最適化を図る．
- (2) 敵対的環境での評価：実際の敵対的攻撃に対する頑健性を，実験によって評価する．
- (3) 形式的検証：プロトコルのセキュリティ性質に関する形式的な証明を与え，理論的な安全性保証を確立する．
- (4) 実システムへの統合：既存の E2EE メッセージングプロトコルとの互換性を考慮した実装と，大規模環境での実証実験を行う．

E2EE がデジタル時代のプライバシー保護の基盤として普及する中，E2EE の透明性と安全性の両立は社会的に重要な課題である．本手法は，この課題に対する技術的な解決策を提示し，プライバシーを守りながら安全なコミュニケーション環境を実現する理想的なモデレーションシステムの構築に向けた重要な一歩となる．

参考文献

- [1] Whatsapp security whitepaper. Technical whitepaper, August 2024.
- [2] Hal Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Vanessa Teague, and Carmela Troncoso. Bugs in our pockets: the risks of client-side scanning. *Journal of Cybersecurity*, 10(1), 2024.
- [3] Apple Inc. Csam detection - technical summary. Technical summary, 2021.
- [4] Apple Inc. The apple PSI system. Technical document,

2022.

- [5] Apple Inc. imessage security overview. Apple Platform Security, December 2024.
- [6] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *KDD*, pages 99–108, 2004.
- [7] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. Technical Report SP 800-38D, NIST, 2007.
- [8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv*, 2015.
- [9] Jens Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT*, pages 305–326, 2016.
- [10] Paul Grubbs, Arasu Arun, Ye Zhang, Joseph Bonneau, and Michael Walfish. Zero-knowledge middleboxes. In *USENIX Security*, 2022.
- [11] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *EMNLP*, 2019.
- [12] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI*, pages 14867–14875, 2021.
- [13] India McKinney and Erica Portnoy. Apple’s plan to “think different” about encryption opens a backdoor to your private life. EFF DeepLinks Blog, August 2021.
- [14] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes: Which naive bayes? In *CEAS*, 2006.
- [15] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *FOCS*, pages 120–130, 1999.
- [16] Lily Hay Newman. Apple quietly nixes its controversial CSAM detector for iCloud photos. *WIRED*, December 2022.
- [17] OpenVM Authors. *OpenVM The Operating System for Zero-Knowledge*. 2024.
- [18] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. *arXiv*, 2017.
- [19] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *AAAI Workshop on Learning for Text Categorization*, 1998.
- [20] Sarah Scheffler and Jonathan Mayer. SoK: Content moderation for end-to-end encryption. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2023(2), 2023.
- [21] Signal Messenger. Quantum resistance and the signal protocol. Signal Blog, September 2023.
- [22] Justin Thaler. *Proofs, Arguments, and Zero-Knowledge*. April 2022. Living textbook draft, actively updated.
- [23] Heng Yang and Ke Li. The best defense is attack: Re-pairing semantics in textual adversarial examples. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8439–8457, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [24] Jinghui Zeng, Jie Ren, Yisen Wang, and Quanquan Gu. Certified robustness to text adversarial attacks by randomized smoothing. *Computational Linguistics*, 49(2):395–436, 2023.