プライバシー保護異常検知フレームワーク

荒井 ひろみ[1,†1]　江村 恵太[1]　林 卓也[2,1]

概要：個人情報を含むデータの収集およびその分析を行う上で，如何にして異常データの提供者のみを特定し，それ以外のデータ提供者のプライバシーを保護するかは重要な課題である．統計処理や機械学習の分野において様々な異常検知手法が提案されているものの，そのそれぞれに対してアドホックにプライバシー保護手法を考案することは極めて非効率である．また何をもって異常データとするのかという規則をデータ収集前に決定することは困難であることから，データ形式や異常値判定規則によらず汎用的に利用可能なプライバシー保護手法であることが望ましい．そこで本論文では，任意の異常検知手法に適用可能なプライバシー保護フレームワークを提案する．具体的に，メッセージ依存開示可能グループ署名 (Group signatures with message-dependent opening, GS-MDO) 及び非対話開示可能公開鍵暗号 (Public key encryption with non-interactive opening, PKENO) からのプライバシー保護異常検知フレームワークの一般的構成を与える．提案フレームワークにおいて，データ提供者は異常データを提供しない限り匿名性が担保される．またデータと提供者との紐付けを行う権限を適切に分離することで，データ提供者を単独で特定可能な，いわゆるビッグブラザーが存在しない．さらに大原らの GS-MDO 方式 (ASIACCS 2013) 及び Galindo らの PKENO 方式 (Africacrypt 2010) を用いて実装を行った．その際効率性の観点から，大原らの GS-MDO 方式に阿部-星野-大久保変換 (CRYPTO 2016) を適用した．実装の結果，提案フレームワークのオーバーヘッドが高々数 10 ミリ秒程に収まることを確認した．

キーワード：プライバシー保護異常検知，メッセージ依存開示可能グループ署名，非対話開示可能公開鍵暗号

A Privacy Preserving Anomaly Detection Framework

Hiromi Arai[1,†1]　Keita Emura[1]　Takuya Hayashi[2,1]

Abstract: This paper proposes a privacy preserving anomaly detection framework that allows an authority to detect adversarial users while other honest users are kept anonymous. In our framework, no big brother exists, meaning that no single entity can identify users, while bad behaviors are always traceable. By using cryptographic techniques, group signatures with message-dependent opening (GS-MDO) and public key encryption with non-interactive opening (PKENO), we provide a correspondence table that links a user and data in a secure way, and we can employ any anonymization technique and any anomaly detection method. We also show implementation results of our framework. Briefly, the overhead of our framework is on the order of dozens of milliseconds.

Keywords: Privacy preserving anomaly detection, Group signatures with message-dependent opening, Public key encryption with non-interactive opening

1. はじめに

個人情報を利用した様々なサービスの提供が盛んに行われている一方で，個人情報を含むデータの収集およびその分析により個人情報の漏洩リスクが高まる懸念がある．そのためプライバシーの保護，とりわけ匿名性の担保は個人情報

[1] 情報通信研究機構
National Institute of Information and Communications Technology (NICT)
[2] 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
[†1] 現在，理化学研究所 革新知能統合研究センター
Presently with RIKEN Center for Advanced Intelligence Project, RIKEN

利活用時において重要視されている．例えば EU における General Data Protection Regulation (GDPR) では匿名化すれば個人情報とはみなされず，データ保護の対象から外れることが明記されている．すなわち，匿名化することで自由にデータを流通させることができるようになるため，個人情報の利活用にはそのような匿名化が必須であるといえる．他の例として，医療分野では Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule [10] が知られている．これは非特定化基準 (de-identification standard) を定めたものであり，個人を特定可能な部分情報を削除することを許可している．またデータ分析を委託する際には，適切なプライバシー保護を施してからデータを提供することが望ましいとされている．

上記より，識別子や疑似識別子を削除するなどの手続き，例えば k 匿名化 [29] などにより匿名化を図ることは個人情報の利活用に向けて重要であるといえる．その一方で，そのような匿名化データを元データに一意的に復元することはできないため，自身が特定されないことをいいことにユーザが悪意を持って振る舞う場合に問題となる．現実問題として，スマートシティにおけるセンサや，医療データ分析による医療過誤の特定，不正送金による銀行口座凍結など，様々な個人情報利活用の用途に応じて異常データの提供者を特定することは非常に重要である．その一方で正当なユーザのプライバシーは侵害されるべきではない．すなわち，如何にして異常データの提供者のみを特定し，それ以外のデータ提供者のプライバシーを保護するかは重要な課題であるといえる．

1.1 プライバシー保護異常検知フレームワーク

統計処理や機械学習の分野において様々な異常検知手法が提案されており，データ形式や異常の特徴により適切な検知手法を選択する必要がある．しかしながら，一般的に何をもって異常データとするのかという規則をデータ収集前に決定することは困難であるため，適切な異常検知手法の選択にはデータの様々な側面を観測しながらの試行錯誤が欠かせない．さらに異常判定規則が更新されることも十分に考えられる．そのため，様々な異常検知手法それぞれに対してアドホックにプライバシー保護手法を考案することは極めて非効率であるといえる．すなわち，匿名化データ形式や異常値判定規則によらず汎用的に利用可能なプライバシー保護フレームワークを構成することが望ましい．そのようなフレームワークに様々な匿名化方式 (差分プライバシー [14] など) や異常検知手法 (k 平均法など) を適用することで，用途に応じたプライバシー保護異常検知プロトコルを実現することができる．

以下，我々が考える異常検知におけるプライバシー保護とは何かについて説明する．まず最初に，正当なユーザのプライバシーを侵害してはならないことが挙げられる．一つの解決策として個人の識別が可能な対応表を用いることが考えられるが，一つの表に個人を特定可能な情報を集約することは安全上望ましくない．例えば，対応表の管理者はデータ提供者を単独で特定可能な，いわゆるビッグブラザーとなってしまう．そこで我々のモデルではデータ提供者に加え，データの収集と分析及び異常判定規則を管理する分析者，対応表を管理してユーザの特定を行う開示者を定義する．図 1 を参照されたい．

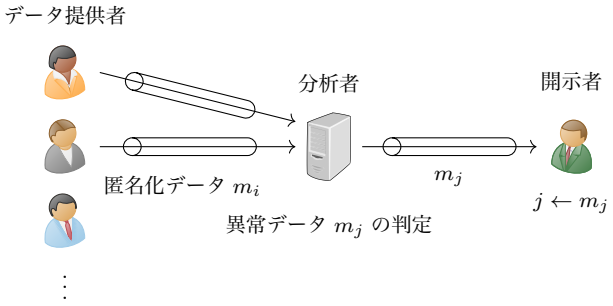

図 1 プライバシー保護異常検知モデル

データ提供者 $i \in [1, n]$ はデータ m_i を所持しているとする．ここで適切な匿名化方式により，m_i から i が漏れないと仮定する．ここでの匿名化手法には制限は無く，任意の手法を適用可能である．分析者はデータ提供者を特定することなしに異常データを抽出する．分析者が扱う異常検知手法もまた，任意の手法を適用可能である．開示者は分析者から得られた異常データにより，これらのデータの提供者を特定する．以下，各管理者がどのような情報を得てもよいのかを定義する．

データ提供者: 自身の識別子 i とデータ m_i のみ．
分析者: 全てのデータ $\{m_i\}_{i\in[1,n]}$ と異常判定規則のみ．
開示者: 異常データとその提供者の識別子のみ．

ここで単独でデータ提供者を特定可能な管理者，すなわちビッグブラザーが存在しないことに注意されたい．

1.2 安易な構成とその限界

ハッシュ関数を用いることで上記方式を簡単に構成できるように思われる．すなわち，m_i を持つデータ提供者が分析者の公開鍵で $m_i||\mathsf{Hash}(i)$ を暗号化する．このとき，分析者は m_i を得られるものの，ハッシュ関数の一方向性より識別子 i を得ることは困難であると想定できる．異常データ m_i を検知後，分析者は開示者の公開鍵で $m_i||\mathsf{Hash}(i)$ を暗号化する．ここで開示者は i と $\mathsf{Hash}(i)$ との対応表を管理しているとすると，この対応表を用いることで異常データとその提供者 $\{(i, m_i)\}$ を出力することができる．このとき暗号化により，外部観測者にデータの情報が漏れることはなく，また開示者は異常データとその提供者の識別子のみを得ることができ，異常データ提供者以外の識別子を知ることはない．

一見，上記方式で十分のように思えるが，以下に挙げるように様々な問題がある．まず第一に $m_i||\mathsf{Hash}(i')$ のようにデータ提供者が自身と異なる識別子をハッシュ関数に入

力する可能性がある．より単純には，ハッシュ関数の値域よりランダムな値を選んでもよい．このときデータ提供者が異常データを送付したとしても決して特定されることはない．つまり上記方式は追跡可能性の観点から問題があるといえる．第二に，分析者が自身で異常データ m を選び，$m||\mathsf{Hash}(i)$ を開示者に送る可能性もある．たとえ分析者が識別子 i を $\mathsf{Hash}(i)$ から特定できないとしても，i が異常データを提供したことにできてしまう．より単純には $m_i||\mathsf{Hash}(i)$ と $m_j||\mathsf{Hash}(j)$ からデータとハッシュ値の置き換えを行い $m_j||\mathsf{Hash}(i)$ を作成することも可能である．もちろん分析者がそのように振る舞う利点はないと考えられるが，ここで問題なのはデータ提供者のデータと識別子を分析者が変更できる余地があることである．すなわち，開示者が得た $\{(i, m_i)\}$ に対し，どのようにして m_i がデータ提供者 i によるものであるのかを保証するかは自明ではない．

次にグループ署名 [8], [11] を用いた方式を考える．グループ署名とは匿名性と追跡可能性を保証する署名方式である．ここで匿名性とは，署名者がグループの一員であることのみ検証可能であることを指す．なお Opener と呼ばれる管理者のみが署名者を特定する権限を有する．また追跡可能性とは，検証に通るものの署名者が追跡できないような署名を作成できないことを指す．グループ署名を利用することで，上記ハッシュ関数を単純に用いた場合の問題が解決するように思われる．すなわち第一の問題に対し，データ提供者がデータ m_i に対しグループ署名を作成することとすれば，グループ署名の追跡可能性によりデータ提供者の識別子 i を変更することはできない．また第二の問題に対しては，m_i が署名されている，すなわちグループ署名がデータに対する証明書の役割を果たしていることから，データと署名との置き換えができない．ここで開示者がグループ署名における Opener の権限を持つことで，異常データの提供者を特定できるとともに，グループ署名の匿名性により分析者がデータ提供者の識別子を得ることはない．

上記のように，グループ署名を利用することで問題が解決するように思える．しかしながら残る大きな問題として，開示者が任意のデータ提供者を特定できてしまう，すなわち匿名性に関して開示者がビッグブラザーになってしまうことが挙げられる．特に非異常データ提供者を特定可能であることはプライバシー上問題である．そのため，単純にグループ署名を使用するだけでは問題の解決にはならない．

1.3 本論文の貢献

本論文では，任意の匿名化方式，異常検知手法に適用可能なプライバシー保護フレームワークを提案する．安全性の観点から，提案フレームワークは以下をみたす．

- 健全性: 開示者の出力には，非異常データが含まれない．
- 一貫性: もし開示者が (i, m_i) を得た場合，データ m_i はデータ提供者 i が提出したものである．
- 追跡可能性: もし開示者が異常データを得た場合，そのデータ提供者を特定できる．
- 秘匿性: 盗聴者はデータに関する情報を得ることはできない．
- 分析者匿名性: 分析者はデータ提供者を特定することはできない．
- 開示者機密性: 開示者は非異常データに関する情報を得ることはできない．
- 開示者匿名性: 開示者は非異常データ提供者を特定することはできない．

我々はメッセージ依存開示可能グループ署名 (Group signatures with message-dependent opening, GS-MDO) [19], [20], [21], [24], [30]，非対話開示可能公開鍵暗号 (Public key encryption with non-interactive opening, PKENO)[12], [13], [15], [16], [18], [25]，及び公開鍵暗号方式からなるプライバシー保護異常検知フレームワークの一般的構成を与える．なお任意長のデータを暗号化するため，公開鍵暗号方式は ECIES [28] と AES-GCM [17] とのハイブリッド構成を採用した．

提案フレームワークは証明可能性を持つ，すなわち構成要素の暗号方式が安全であれば提案フレームワークは上記 7 つの安全性をみたす．仮にプライバシー侵害が発生した，例えば予期せずデータ提供者の識別子が漏えいした場合，問題箇所を暗号方式と匿名化方式とに切り分けることができる．このとき暗号方式が帰着される計算量問題 (例えば楕円曲線上の離散対数問題) が困難である限り，識別子の漏えいは匿名化方式を通して発生したことが保証される．

最後に大原らの GS-MDO 方式 [21] 及び Galindo らの PKENO 方式 [16] を用いて実装を行い，提案フレームワークのオーバーヘッドが高々数 10 ミリ秒程に収まることを確認した．大原らの GS-MDO 方式はランダムオラクルを仮定する必要があるものの，これまで提案された GS-MDO 方式の中で最も効率的な方式である．なお大原らの GS-MDO 方式は対称ペアリング (タイプ 1 ペアリング) で構成されているが，効率の観点からは非対称ペアリング (タイプ 3 ペアリング) での構成が望ましい．そこで阿部，星野，大久保によるタイプ 1 からタイプ 3 への変換手法 [1], [2] を適用し，タイプ 3 ペアリングで大原らの GS-MDO 方式を構成した (付録 A.1 を参照のこと)．また Galindo らの PKENO 方式 [16] として，Shoup-Gennaro しきい値暗号 [27] から構成される方式を採用した．この方式はランダムオラクルを仮定する必要があるものの，ペアリングを使用しないという利点がある．

2. 準備

2.1 メッセージ依存開示可能グループ署名 (GS-MDO)

GS-MDO [19], [20], [21], [24], [30] はグループ署名の一種であり，グループ署名における Opener がビッグブラザーに

なることを解消する方式である．GS-MDO では Opener とは別に Admitter と呼ばれる管理者を導入する．Admitter は署名するメッセージに依存したトークンを発行，Opener は自身の秘密鍵とこのトークンを用いることで初めて署名者の特定が可能となる．すなわち，Admitter 単体，Opener 単体では署名者の特定ができない．提案フレームワークでは分析者が Admitter の鍵を，開示者が Opener の鍵を管理する．

GS-MDO 方式は以下の 6 つのアルゴリズム (GKg, AKg, GSig, Td, GVf, Open) から構成される．なお元々の定義では Admitter の公開鍵と秘密鍵の生成アルゴリズム AKg は与えられていない．提案フレームワークでは，Admitter の秘密鍵 (トークン生成鍵) ak と Opener の秘密鍵 (開示鍵) ok がそれぞれ分析者と開示者に別々に管理されるため，本論文で AKg アルゴリズムを追加で定義する．なお大原らの GS-MDO 方式では ak と ok は独立に生成することが可能であり，AKg アルゴリズムの導入は安全性証明の観点からも影響を与えない．

定義 2.1 (GS-MDO のシンタックス)

- GKg: グループ鍵生成アルゴリズムはセキュリティパラメータ λ とグループメンバ数 n を入力とし，グループ公開鍵 gpk，Opener の秘密鍵 ok，n 個の署名鍵 $\{gsk_i\}_{i\in[n]}$ を出力する．
- AKg: Admitter 鍵生成アルゴリズムは gpk を入力とし，Admitter 公開鍵 apk と秘密鍵 ak を出力する．
- GSig: 署名アルゴリズムは gpk, apk, gsk_i, メッセージ M を入力とし，グループ署名 σ を出力する．
- Td: メッセージ依存トークン生成アルゴリズムは gpk, ak, M を入力とし，トークン t_M を出力する．
- GVf: 検証アルゴリズムは gpk, apk, M, σ を入力とし，1 (accept) または 0 (reject) を出力する．
- Open: 追跡アルゴリズムは gpk, apk, ok, M, σ, t_M を入力とし，$i \in \mathbb{N}$ または $\perp$ を出力する．

GS-MDO の正当性を以下で定義する．全ての λ, n, 全ての $(gpk, ok, \{gsk_i\}_{i\in[n]}) \leftarrow \mathsf{GKg}(1^\lambda, 1^n)$, $(apk, ak) \leftarrow \mathsf{AKg}(gpk)$，全ての $M \in \{0,1\}^*$ と $i \in [n]$ に対し，

$$\mathsf{GVf}(gpk, apk, M, \mathsf{GSig}(gpk, gsk_i, M)) = 1$$

及び

$$\mathsf{Open}(gpk, apk, ok, M, \sigma, \mathsf{Td}(gpk, ak, M)) = i$$

が成り立つ．ここで $\sigma \leftarrow \mathsf{GSig}(gpk, apk, gsk_i, M)$ である．

GS-MDO の安全性として，Opener 匿名性，Admitter 匿名性，追跡可能性が定義される．ここで Opener 匿名性とは，ok と全ての $\{gsk_i\}_{i\in[n]}$ を所持していたとしても署名者を特定できないことを保証し，Admitter 匿名性とは ak と全ての $\{gsk_i\}_{i\in[n]}$ を所持していたとしても署名者を特定できないことを保証する．追跡可能性は Opener，Admitter，グループメンバが結託したとしても署名者が特定できない，かつ検証アルゴリズムに通る署名を作成できないことを保証する．詳細な定義は論文 [24] を参照されたい．

2.2 非対話開示可能公開鍵暗号 (PKENO)

PKENO [12], [13], [15], [16], [18], [25] とは公開鍵暗号の一種であり，復号鍵を使用して復号結果が正しいことを証明可能な方式である．すなわち平文 m の暗号文 C に対し，C の復号結果が m である証明 π を生成する．ここで π は公開検証可能であり，復号鍵を開示せずに C の復号結果を証明可能であることに注意されたい．なお PKENO の安全性として証明健全性が要求される．すなわち平文 m の暗号文 C に対し，$m \neq m'$ かつ (C, m', π') が検証を通るような (m', π') を復号鍵を持っていたとしても生成できないことを保証する．提案フレームワークではこの証明健全性を利用することで，データ m_i が提供者 i によるものであることを保証する．

PKENO 方式は以下の 5 つのアルゴリズム (PKg, PEnc, PDec, PProve, PVerify) から構成される．

定義 2.2 (PKENO のシンタックス)

- PKg: 鍵生成アルゴリズムはセキュリティパラメータ λ を入力とし，公開鍵 pk と復号鍵 dk を出力する．ここで平文空間 $\mathcal{M}$ は pk に含まれていると仮定する．
- PEnc: 暗号化アルゴリズムは pk と平文 $m \in \mathcal{M}$ を入力とし，暗号文 C を出力する．
- PDec: 復号アルゴリズムは dk, C を入力とし，m または $\perp$ を出力する．
- PProve: 証明生成アルゴリズムは dk, C を入力とし，証明 π を出力する．
- PVerify: 証明検証アルゴリズムは pk, C, m, π を入力とし，1 (accept) または 0 (reject) を出力する．

PKENO の正当性を以下で定義する．全ての λ と $(pk, dk) \leftarrow \mathsf{PKg}(1^\lambda)$ について，

- 全ての平文 $m \in \mathcal{M}$ に対し，

$$\mathsf{PDec}(dk, \mathsf{PEnc}(pk, m)) = m$$

- (正当ではないものも含む) 全ての暗号文 C に対し，

$$\mathsf{PVerify}(pk, C, \mathsf{PDec}(dk, C), \mathsf{PProve}(dk, C)) = 1$$

PKENO の安全性として，証明健全性に加えて IND-CCPA 安全性 (indistinguishability against chosen ciphertext and proof attacks) が定義される．IND-CCPA 安全性は攻撃者が復号オラクルと証明オラクルにアクセスできる状況で，平文の情報が暗号文から漏えいしないことを保証する．詳細な安全性定義は論文 [16] を参照されたい．

なお通常の公開鍵暗号は PKENO から PProve と PVerify を省いた形で定義される．PKENO との混同を避けるため，本論文では公開鍵暗号方式を (Kg, Enc, Dec) で表記する．

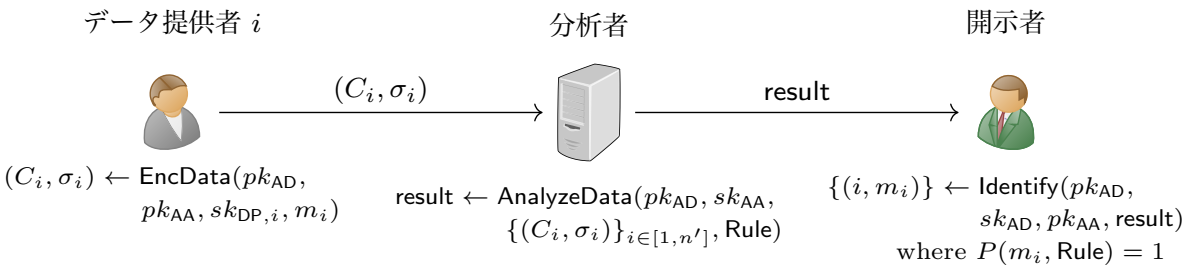

図 2 プライバシー保護異常検知フレームワーク

3. 提案フレームワークの定義

本章では提案フレームワークの定義を与える．まず異常検知をモデル化する．

定義 3.1 (異常検知) 規則空間 $\mathcal{ADR}$ に対し $\mathsf{Rule} \in \mathcal{ADR}$ を異常検知規則とする．またデータ空間 $\mathcal{M}$ に対し，$m \in \mathcal{M}$ をデータとする．述語 $P : \mathcal{M} \times \mathcal{ADR} \to \{0, 1\}$ を以下で定義する．

$$P(m, \mathsf{Rule}) = \begin{cases} 1 & (\text{if } m \text{ が規則 Rule に合致する場合}) \\ 0 & (\text{それ以外}) \end{cases}$$

次に提案フレームワークのシンタックスを定義する．

定義 3.2 (フレームワークのシンタックス) プライバシー保護異常検知フレームワークは 5 つのアルゴリズム $(\mathsf{KeyGen}_{\mathsf{AD}}, \mathsf{KeyGen}_{\mathsf{AA}}, \mathsf{EncData}, \mathsf{AnalyzeData}, \mathsf{Identify})$ から構成される．図 2 も参照されたい．以下，AD は開示者 (Anomaly Detector) を，AA は分析者 (Anomaly Analyzer) を指す．

- $\mathsf{KeyGen}_{\mathsf{AD}}$: 開示者の鍵生成アルゴリズムはセキュリティパラメータ λ とデータ提供者数 n を入力とし，開示者の公開鍵 pk_{AD}，秘密鍵 sk_{AD}，およびデータ提供者の秘密鍵 $\{sk_{\mathsf{DP},i}\}_{i\in[1,n]}$ を出力する．ここでデータ空間 $\mathcal{M}$ は pk_{AD} に含まれていると仮定する．
- $\mathsf{KeyGen}_{\mathsf{AA}}$: 分析者の鍵生成アルゴリズムは pk_{AD} を入力とし，分析者の公開鍵 pk_{AA} と秘密鍵 sk_{AA} を出力する．ここで異常検知規則空間 $\mathcal{ADR}$ は pk_{AA} に含まれていると仮定する．
- $\mathsf{EncData}$: データ暗号化アルゴリズムは pk_{AD}, pk_{AA}, $sk_{\mathsf{DP},i}$, データ $m_i \in \mathcal{M}$ を入力とし，暗号文 C_i とタグ σ_i を出力する．
- $\mathsf{AnalyzeData}$: データ分析アルゴリズムは pk_{AD}, sk_{AA}, $\{(C_i, \sigma_i)\}_{i\in[1,n']}$ [*1]，異常検知規則 $\mathsf{Rule} \in \mathcal{ADR}$ を入力とし，分析結果 result を出力する．
- $\mathsf{Identify}$: 識別子特定アルゴリズムは pk_{AD}, sk_{AD}, pk_{AA}, result を入力とし，識別子とデータ組の集合 $\{(i, m_i)\}$ を出力する．

[*1] 本論文では，各データ提供者は 1 つの暗号文とタグのペアを送付すると仮定する．その場合，$n' \leq n$ である．なお各データ提供者が複数の暗号文とタグのペアを送付する場合も単純な拡張で考慮可能である．

正当性は以下で定義される．全ての $(pk_{\mathsf{AD}}, sk_{\mathsf{AD}}, \{sk_{\mathsf{DP},i}\}_{i\in[1,n]}) \leftarrow \mathsf{KeyGen}_{\mathsf{AD}}(1^\lambda, 1^n)$, $(pk_{\mathsf{AA}}, sk_{\mathsf{AA}}) \leftarrow \mathsf{KeyGen}_{\mathsf{AA}}(pk_{\mathsf{AD}})$, $m_i \in \mathcal{M}$, $n' \leq n$, $\mathsf{Rule} \in \mathcal{ADR}$ に対し，もし $P(m_i, \mathsf{Rule}) = 1$ である (つまり m_i は異常データである) 場合，(i, m_i) は $\mathsf{Identify}(pk_{\mathsf{AD}}, sk_{\mathsf{AD}}, pk_{\mathsf{AA}}, \mathsf{result})$ の出力に確率 1 で含まれる．ここで $\mathsf{result} \leftarrow \mathsf{AnalyzeData}(pk_{\mathsf{AD}}, sk_{\mathsf{AA}}, \{(C_i, \sigma_i)\}_{i\in[1,n']}, \mathsf{Rule})$ 及び $(C_i, \sigma_i) \leftarrow \mathsf{EncData}(pk_{\mathsf{AD}}, pk_{\mathsf{AA}}, sk_{\mathsf{DP},i}, m_i)$ である．

プライバシー保護異常検知フレームワークの安全性として，健全性，一貫性，追跡可能性，秘匿性，分析者匿名性，開示者機密性，開示者匿名性が定義される．ページ数の都合上，詳細な定義は割愛する．それぞれの直観的な意味は 1.3 章を参照されたい．

4. プライバシー保護異常検知フレームワークの構成

本章では，GS-MDO 方式 (GKg, AKg, GSig, Td, GVf, Open), PKENO 方式 (PKg, PEnc, PDec, PProve, PVerify), 公開鍵暗号方式 (Kg, Enc, Dec) からのプライバシー保護異常検知フレームワークの一般的構成を与える．提案構成では，開示者は公開鍵暗号の鍵ペア $(\tilde{pk}, \tilde{dk})$ と GS-MDO 方式の開示鍵 ok を持ち，分析者は PKENO 方式の鍵ペア (pk, dk) と GS-MDO 方式のトークン生成鍵 ak を持つ．直観的には，提案構成法は以下のように説明される．

EncData. データ m_i を持つデータ提供者 i はまず m_i を分析者の公開鍵 pk で暗号化，その暗号文を C_i とする．次に i はこの C_i に対しグループ署名 σ_i を計算する．ここで m_i と i は C_i と σ_i から，それぞれ PKENO 方式の IND-CCPA 安全性と GS-MDO 方式の匿名性より漏えいしない．特に開示鍵 ok を持つ開示者であったとしても，GS-MDO 方式の Opener 匿名性により (C_i, σ_i) を観測したとしても i を特定することはできないことに注意されたい．

AnalyzeData. 分析者は C_i に対するグループ署名 σ_i の正当性を検証した後，C_i を復号し m_i を得る．ここで m_i は適切な匿名化手法により匿名化されたものであると仮定しているため，m_i からは i の情報が漏えいしないことに注意されたい．さらに GS-MDO の Admitter 匿名性により，分析者は (C_i, σ_i) から i を特定することはできない．もし m_i が異常データである場合，つまり $P(m_i, \mathsf{Rule}) = 1$ の場合，

分析者はトークン t_{C_i} 及び C_i の復号結果が m_i であることの証明 π_i を作成する．ここで PKENO 方式の証明健全性により，C_i の復号結果が m_i ではないことの証明は作成できない．なお分析者が自身で暗号文を作成することも考えられるが，分析者は GS-MDO 方式の署名鍵を所持していない．これらのことより，分析者は m_i がデータ提供者によるものであり，分析者が提供したものではないことを保証できる．最後に分析者は異常データとその証明 (m_i, π_i) 及び (C_i, σ_i, t_{C_i}) を開示者の公開鍵 $\tilde{pk}$ で暗号化する．

Identify. 開示者は C_i の復号結果が m_i であることを検証した後．GS-MDO の開示鍵 ok とトークン t_{C_i} とを用いて m_i の提供者を特定する．なお GS-MDO の追跡可能性より，データ提供者は自身の識別子を偽ることができない．より正確には，グループ署名が正当である場合，GS-MDO の Open アルゴリズムは署名者を正確に追跡することが保証される．

以下に提案フレームワークの構成を与える．

$\mathsf{KeyGen}_{\mathsf{AD}}(1^\lambda, 1^n)$: $(gpk, ok, \{gsk_i\}_{i\in[n]}) \leftarrow \mathsf{GKg}(1^\lambda, 1^n)$ 及び $(\tilde{pk}, \tilde{dk}) \leftarrow \mathsf{Kg}(1^\lambda)$ を実行し，$pk_{\mathsf{AD}} = (gpk, \tilde{pk})$, $sk_{\mathsf{AD}} = (ok, \tilde{dk})$，及び $\{sk_{\mathsf{DP},i} = gsk_i\}_{i\in[1,n]}$ を出力する．

$\mathsf{KeyGen}_{\mathsf{AA}}(pk_{\mathsf{AD}})$: $pk_{\mathsf{AD}} = (gpk, \tilde{pk})$ とする．$(apk, ak) \leftarrow \mathsf{AKg}(gpk)$ 及び $(pk, dk) \leftarrow \mathsf{PKg}(1^\lambda)$ を実行し，$pk_{\mathsf{AA}} = (apk, pk)$ 及び $sk_{\mathsf{AA}} = (ak, dk)$ を出力する．

$\mathsf{EncData}(pk_{\mathsf{AD}}, pk_{\mathsf{AA}}, sk_{\mathsf{DP},i}, m_i)$: $pk_{\mathsf{AD}} = (gpk, \tilde{pk})$, $pk_{\mathsf{AA}} = (apk, pk)$, $sk_{\mathsf{DP},i} = gsk_i$ とする．$C_i \leftarrow \mathsf{PEnc}(pk, m_i)$ 及び $\sigma_i \leftarrow \mathsf{GSig}(gpk, apk, gsk_i, C)$ を実行し，C_i 及び σ_i を出力する．

$\mathsf{AnalyzeData}(pk_{\mathsf{AD}}, sk_{\mathsf{AA}}, \{(C_i, \sigma_i)\}_{i\in[1,n']}, \mathsf{Rule})$: $pk_{\mathsf{AD}} = (gpk, \tilde{pk})$, $sk_{\mathsf{AA}} = (ak, dk)$ とする．全ての $i \in [1, n']$ に対し，$\mathsf{GVf}(gpk, apk, C_i, \sigma_i) = 1$ が成り立つときにのみ $m_i \leftarrow \mathsf{PDec}(dk, C_i)$ を実行する．ここで $n_a \leq n'$ を $P(m_i, \mathsf{Rule}) = 1$ をみたす異常データ数とし，$\{m_j\}_{j\in[1,n_a]}$ を異常データの集合とする．全ての $j \in [1, n_a]$ に対し，$t_{C_j} \leftarrow \mathsf{Td}(gpk, ak, C_j)$ 及び $\pi_j \leftarrow \mathsf{PProve}(dk, C_j)$ を計算する[*2]．$C \leftarrow \mathsf{Enc}(\tilde{pk}, \{(C_j, \sigma_j, m_j, \pi_j, t_{C_j})\}_{j\in[1,n_a]})$ を計算し，$\mathsf{result} = C$ を出力する．

$\mathsf{Identify}(pk_{\mathsf{AD}}, sk_{\mathsf{AD}}, pk_{\mathsf{AA}}, \mathsf{result})$: $pk_{\mathsf{AD}} = (gpk, \tilde{pk})$, $sk_{\mathsf{AD}} = (ok, \tilde{dk})$, $pk_{\mathsf{AA}} = (apk, pk)$, $\mathsf{result} = C$ とする．$\{(C_j, \sigma_j, m_j, \pi_j, t_{C_j})\}_{j\in[1,n_a]} \leftarrow \mathsf{Dec}(\tilde{dk}, C)$ を計算する．全ての $j \in [1, n_a]$ に対し，もし $\mathsf{GVf}(gpk, apk, C_j, \sigma_j) = 1$ であれば $\mathsf{PVerify}(pk, C_j, m_j, \pi_j)$ を実行する．ここで $n'_a \leq n_a$ を GVf アルゴリズム，PVerify アルゴリズムによる検証に通った $(C_j, \sigma_j, m_j, \pi_j)$ の数とし，$\{(C_k, \sigma_k, m_k, \pi_k)\}_{k\in[1,n'_a]}$ をそのような $(C_j, \sigma_j, m_j, \pi_j)$ の組とする．$\{(\mathsf{Open}(gpk, apk, ok, C_k, \sigma_k, t_{C_k}), m_k)\}_{k\in[1,n'_a]}$ を出力する．

基本的に上記構成は構成要素である GS-MDO 方式，PKENO 方式，公開鍵暗号方式が安全であれば要求された安全性をみたす．しかしながら暗号以外の仮定も用いているため，ここではこれら仮定について説明する．

まず前述のとおり，m_i は匿名化済みのデータであり，m_i から i に関する情報が漏れないと仮定する．

次にもし分析者が $P(m_i, \mathsf{Rule}) = 0$，つまり m_i が異常データではないにも関わらず result に m_i の暗号文を含めた場合，健全性，開示者秘匿性，開示者匿名性をみたさないことになる．しかし開示者も $P(m_i, \mathsf{Rule}) = 1$ かどうかを確認することができるため，そのような分析者の振る舞いは常に開示者により検知される．そのため妥当な仮定として，分析者は $P(m_i, \mathsf{Rule}) = 1$ であるときのみ result に m_i の暗号文を含めるとする．

最後に，開示者が $\{sk_{\mathsf{DP},i} = gsk_i\}_{i\in[1,n]}$ を所持していることによる影響を分析する．Opener 匿名性により ok と署名鍵を所持していたとしても匿名性に影響を与えることはないものの，データ提供者 i の代わりに開示者が (C_i, σ_i) を作成できてしまうという問題がある．もし開示者がそのような自身で作成した暗号文とタグを最終結果に含めると，一貫性や追跡可能性をみたさないことになる．そこで本論文では，開示者は暗号文とタグを作成しないと仮定する．[*3]

提案フレームワークの安全性は以下の定理で示される．証明はページ数の都合上割愛する．

定理 4.1 分析者は $P(m_i, \mathsf{Rule}) = 1$ であるときのみ result に m_i の暗号文を含めると仮定する．このとき，公開鍵暗号方式が正当性をみたすならば，提案フレームワークは健全性をみたす．

定理 4.2 開示者は自身で暗号文とタグを作成しないと仮定する．このとき，PKENO 方式が証明健全性をみたし GS-MDO 方式が追跡可能性をみたすならば，提案フレームワークは一貫性をみたす．

定理 4.3 開示者は自身で暗号文とタグを作成しないと仮定する．このとき，GS-MDO 方式が追跡可能性をみたすならば，提案フレームワークは追跡可能性をみたす．

定理 4.4 PKENO 方式が IND-CCPA 安全，公開鍵暗号方式が IND-CCA 安全なとき，提案フレームワークは秘匿性をみたす．

[*2] もし C_i が正当な暗号文ではない場合，PDec アルゴリズムは $m_i = \perp$ を出力する．もし異常検知規則 Rule が $\perp$ を異常値とするならば，このような不正な暗号文に対しても証明を作成する必要があるが，PProve アルゴリズムはこのような不正な暗号文に対しても動作するため問題とはならない．

[*3] なお暗号理論的な解決策として，対話型の署名鍵発行プロトコル [6], [9] を利用することが考えられる．このとき，GS-MDO は動的 (セットアップ後に署名者の追加が可能) となり，Non-frameability をみたす．ここで Non-frameability とは，1 人の正直な署名者を除く全ての署名者，管理者が結託したとしても，開示結果がこの正直な署名者となるような署名を作成できないことを保証する．さらに GS-MDO 方式が Opening Soundness [26] をみたす場合，開示者は (C_i, σ_i) の開示結果が i であることを示す証明を提示可能となる．なお Non-frameability や Opening Soundness をみたす動的 GS-MDO 方式がこれまで知られていないため，そのような GS-MDO 方式の構成は本論文における将来の課題とする．

定理 4.5 m_i から i に関する情報が漏れないと仮定する．このとき，GS-MDO 方式が Admitter 匿名性をみたすならば提案フレームワークは分析者匿名性をみたす．

定理 4.6 分析者は $P(m_i, \mathsf{Rule}) = 1$ であるときのみ result に m_i の暗号文を含めると仮定する．このとき，PKENO 方式が IND-CCPA 安全であるならば，提案フレームワークは開示者機密性をみたす．

定理 4.7 分析者は $P(m_i, \mathsf{Rule}) = 1$ であるときのみ result に m_i の暗号文を含めると仮定する．このとき，GS-MDO 方式が Opener 匿名性をみたすならば提案フレームワークは開示者匿名性をみたす．

5. 実装結果

本章では，提案フレームワークの実装結果を与える．本実装では楕円曲線上での演算とペアリング計算に RELIC toolkit library (ver.0.4.1) [3] を使用，ハッシュ関数や共通鍵暗号の計算に OpenSSL (ver.1.0.2k) [22] を使用した．128 ビット安全性をみたすため，Barbulescu と Duquesne の評価 [4] を鑑み，Pairing-friendly 楕円曲線として埋め込み次数 12，455 ビット素体上の Barreto-Lynn-Scott (BLS) 曲線 [5] を用いた．また，PKENO 方式と ECIES では楕円曲線パラメータに Curve25519 [7] を用いた．

まず通信コストの見積もりを表 1 に与える．たとえば PKENO 方式の平文空間が $\ell = 1{,}024$ バイトである場合，データ提供者から分析者へは 2,288 バイト，分析者から開示者へは 3,571 バイトの通信が必要である．

我々は MiniBooNE データセット [23] を元に各アルゴリズムの実行時間を見積もりを行った．このデータセットには属性数 50 のデータが 130,065 レコード含まれており，各レコードサイズは 700 バイトである．このデータセットに 10%の異常データレコードが含まれていると仮定し，実験ではランダムに異常データを選択した．また各データ提供者は 1 データレコードを分析者に送信するとした．ベンチマークは表 2 を参照されたい．実験環境は Core i7-7700K (4.20 GHz)，計算はすべてシングルスレッドで実行した．プログラムは C++で記述し，gcc 6.3.0，最適化オプションとして “`-O3 -march=native`” を用いてコンパイルを行った．結果，データ提供者は 1 レコードに対し 13.5 ミリ秒程度，分析者は 1 レコードに対し 14.3 ミリ秒程度，1 異常レコードに対し 1.6 ミリ秒程度，開示者は 1 異常レコードに対し 17.7 ミリ秒程度の計算で処理できることを確認した．

6. 結論

本論文では，任意の匿名化方式，異常検知手法に適用可能なプライバシー保護フレームワークを提案した．異常データの提供者のみを特定し，それ以外のデータ提供者の匿名性が担保される．本提案により，様々な匿名化方式や異常検知手法それぞれに対してアドホックにプライバシー保護手法を考慮する必要なく，データ提供者のプライバシーを考慮した異常検知プロトコルが構成可能となる．また大原らの GS-MDO 方式，Galindo らの PKENO 方式を適用した場合の実装を与え，1 レコードあたり数 10 ミリ秒程度の追加コストで動作することを確認した．

表 1 提案フレームワークにおける通信コスト

データ提供者 → 分析者	$(\ell + 1264)$ バイト/レコード
分析者 → 開示者	$(2\ell + 1523)$ バイト/異常レコード

ℓ: メッセージ長 (バイト).

表 2 提案フレームワークのベンチマーク

Algorithms	Benchmarks	Entity
$\mathsf{KeyGen}_{\mathsf{AD}}$	2.027 ミリ秒 1.731 ミリ秒/データ提供者	開示者
$\mathsf{KeyGen}_{\mathsf{AA}}$	0.433 ミリ秒	分析者
EncData	13.538 ミリ秒/レコード	データ提供者
AnalyzeData	14.252 ミリ秒/レコード 1.628 ミリ秒/異常レコード	分析者
Identify	17.704 ミリ秒/異常レコード	開示者

謝辞: 本研究は JST CREST JPMJCR168A の助成を受け実施されたものです．また大原らの GS-MDO 方式に対して阿部-星野-大久保変換を適用するにあたり，大久保美也子氏 (NICT) にご協力いただきました．ここに深く感謝いたします．

参考文献

[1] Abe, M., Groth, J., Ohkubo, M. and Tango, T.: Converting Cryptographic Schemes from Symmetric to Asymmetric Bilinear Groups, *CRYPTO*, pp. 241–260 (2014).
[2] Abe, M., Hoshino, F. and Ohkubo, M.: Design in Type-I, Run in Type-III: Fast and Scalable Bilinear-Type Conversion Using Integer Programming, *CRYPTO*, pp. 387–415 (2016).
[3] Aranha, D. F. and Gouvêa, C. P. L.: RELIC is an Efficient LIbrary for Cryptography, `https://github.com/relic-toolkit/relic`.
[4] Barbulescu, R. and Duquesne, S.: Updating key size estimations for pairings, *IACR Cryptology ePrint Archive*, Vol. 2017, p. 334 (2017).
[5] Barreto, P. S. L. M., Lynn, B. and Scott, M.: Constructing Elliptic Curves with Prescribed Embedding Degrees, *SCN*, pp. 257–267 (2002).
[6] Bellare, M., Shi, H. and Zhang, C.: Foundations of Group Signatures: The Case of Dynamic Groups, *CT-RSA*, pp. 136–153 (2005).
[7] Bernstein, D. J.: Curve25519: New Diffie-Hellman Speed Records, *Public Key Cryptography*, pp. 207–228 (2006).
[8] Boneh, D., Boyen, X. and Shacham, H.: Short Group Signatures, *CRYPTO*, pp. 41–55 (2004).
[9] Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E. and Groth, J.: Foundations of Fully Dynamic Group Signatures, *Applied Cryptography and Network Security*, pp. 117–136 (2016).
[10] Centers for Disease Control and Prevention and others: HIPAA privacy rule and public health. Guidance from CDC and the US Department of Health and Human Services, *MMWR: Morbidity and mortality weekly report*, Vol. 52, No. Suppl. 1, pp. 1–17 (2003).
[11] Chaum, D. and van Heyst, E.: Group Signatures, *EUROCRYPT*, pp. 257–265 (1991).

[12] Dachman-Soled, D., Fuchsbauer, G., Mohassel, P. and O'Neill, A.: Enhanced Chosen-Ciphertext Security and Applications, *Public-Key Cryptography*, pp. 329–344 (2014).
[13] Damgård, I., Hofheinz, D., Kiltz, E. and Thorbek, R.: Public-Key Encryption with Non-interactive Opening, *CT-RSA*, pp. 239–255 (2008).
[14] Dwork, C.: Differential Privacy, *ICALP (2)*, pp. 1–12 (2006).
[15] Galindo, D.: Breaking and Repairing Damgård et al. Public Key Encryption Scheme with Non-interactive Opening, *CT-RSA*, pp. 389–398 (2009).
[16] Galindo, D., Libert, B., Fischlin, M., Fuchsbauer, G., Lehmann, A., Manulis, M. and Schröder, D.: Public-Key Encryption with Non-Interactive Opening: New Constructions and Stronger Definitions, *AFRICACRYPT*, pp. 333–350 (2010).
[17] Iwata, T., Ohashi, K. and Minematsu, K.: Breaking and Repairing GCM Security Proofs, *CRYPTO*, pp. 31–49 (2012).
[18] Lai, J., Deng, R. H., Liu, S. and Kou, W.: Efficient CCA-Secure PKE from Identity-Based Techniques, *CT-RSA*, pp. 132–147 (2010).
[19] Libert, B. and Joye, M.: Group Signatures with Message-Dependent Opening in the Standard Model, *CT-RSA*, pp. 286–306 (2014).
[20] Libert, B., Mouhartem, F. and Nguyen, K.: A Lattice-Based Group Signature Scheme with Message-Dependent Opening, *Applied Cryptography and Network Security*, pp. 137–155 (2016).
[21] Ohara, K., Sakai, Y., Emura, K. and Hanaoka, G.: A group signature scheme with unbounded message-dependent opening, *ASIA CCS*, pp. 517–522 (2013).
[22] OpenSSL Project: OpenSSL: Cryptography and SSL/TLS Toolkit, `https://www.openssl.org`.
[23] Roe, B. P.: MiniBooNE particle identification Data Set, `https://archive.ics.uci.edu/ml/datasets/MiniBooNE+particle+identification`.
[24] Sakai, Y., Emura, K., Hanaoka, G., Kawai, Y., Matsuda, T. and Omote, K.: Group Signatures with Message-Dependent Opening, *Pairing-Based Cryptography*, pp. 270–294 (2012).
[25] Sakai, Y., Matsuda, T. and Hanaoka, G.: Tag-KEM/DEM framework for public-key encryption with non-interactive opening, *ISITA*, pp. 231–235 (2016).
[26] Sakai, Y., Schuldt, J. C. N., Emura, K., Hanaoka, G. and Ohta, K.: On the Security of Dynamic Group Signatures: Preventing Signature Hijacking, *Public Key Cryptography*, pp. 715–732 (2012).
[27] Shoup, V. and Gennaro, R.: Securing Threshold Cryptosystems against Chosen Ciphertext Attack, *J. Cryptology*, Vol. 15, No. 2, pp. 75–96 (2002).
[28] Smart, N. P.: The Exact Security of ECIES in the Generic Group Model, *Cryptography and Coding*, pp. 73–84 (2001).
[29] Sweeney, L.: *k*-Anonymity: A Model for Protecting Privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557–570 (2002).
[30] 江村 恵太, 花岡 悟一郎, 川合 豊, 松田 隆宏, 面 和成, 坂井 祐介：メッセージ依存開示可能グループ署名と匿名掲示板への応用, 第 28 回暗号と情報セキュリティシンポジウム (SCIS) (2011, 3A1-4).

付　　録

A.1 大原らの GS-MDO 方式（タイプ 3）

- $\mathsf{GKg}(1^\lambda, 1^n)$: $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ をタイプ 3 の双線形群とする．ここで，$\langle g_1 \rangle = \mathbb{G}_1$, $\langle g_2 \rangle = \mathbb{G}_2$．ハッシュ関数として $H_1 : \{0,1\}^* \to \mathbb{G}_2$, $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ を選ぶ．$u, v, h \xleftarrow{\$} \mathbb{G}_1$, $\xi_1, \xi_2, \xi_3, \gamma \xleftarrow{\$} \mathbb{Z}_p$ を選び，$\bar{g}_1 = u^{\xi_1} v^{\xi_3}$, $\bar{g}_2 = v^{\xi_2} h^{\xi_3}$, $\omega = g_2^\gamma$ を計算する．$i \in [1,n]$ について，$x_i \xleftarrow{\$} \mathbb{Z}_p$ を選び，$A_i = g_1^{1/(\gamma + x_i)}$ を計算，$gsk_i = (A_i, x_i)$ とする．$gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$, $ok = (\xi_1, \xi_2, \xi_3, \{e(A_i, g_2)\}_{i \in [1,n]})$, $\{gsk_i\}_{i \in [n]}$ を出力する．
- $\mathsf{AKg}(gpk)$: $gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$ とする．$\xi \xleftarrow{\$} \mathbb{Z}_p$ を選び，$y = g_1^\xi$ を計算，$apk = y$ および $ak = \xi$ を出力する．
- $\mathsf{GSig}(gpk, apk, gsk_i, M)$: $gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$, $apk = y$, $gsk_i = (A_i, x_i)$ とする．$\alpha, \beta, \rho, \eta \xleftarrow{\$} \mathbb{Z}_p$ を選び，$T_1 = u^\alpha$, $T_2 = v^\beta$, $T_3 = h^{\alpha+\beta}$, $T_4 = \bar{g}_1^\alpha \bar{g}_2^\beta A_i g_1^\eta$, $T_5 = g_1^\rho$, $T_6 = e(y, H_1(M))^\rho e(g_1, g_2)^{-\eta}$ を計算する．$r_\alpha, r_\beta, r_\rho, r_\eta, r_x, r_{\alpha x}, r_{\beta x}, r_{\rho x}, r_{\eta x} \xleftarrow{\$} \mathbb{Z}_p$ を選び，$R_1 = u^{r_\alpha}$, $R_2 = v^{r_\beta}$, $R_3 = h^{r_\alpha + r_\beta}$, $R_4 = e(T_4, g_2)^{r_x} e(g_1, \omega)^{-r_\alpha} e(\bar{g}_1, g_2)^{-r_{\alpha x}} e(\bar{g}_2, \omega)^{-r_\beta} e(\bar{g}_2, g_2)^{-r_{\beta x}} e(g_1, \omega)^{-r_\eta} e(g_1, g_2)^{-r_{\eta x}}$, $R_5 = g_1^{r_\rho}$, $R_6 = e(y, H_1(M))^{r_\rho} e(g_1, g_2)^{-r_\eta}$, $R_7 = T_1^{r_x} u^{-r_{\alpha x}}$, $R_8 = T_2^{r_x} v^{-r_\beta}$, $R_9 = T_5^{r_x} g_1^{-r_\rho}$, $R_{10} = T_6^{r_x} e(y, H_1(M))^{-r_\rho} e(g_1, g_2)^{r_{\eta x}}$, $c = H_2(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$, $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, $s_\rho = r_\rho + c\rho$, $s_\eta = r_\eta + c\eta$, $s_x = r_x + cx_i$, $s_{\alpha x} = r_{\alpha x} + c\alpha x_i$, $s_{\beta x} = r_{\beta x} + c\beta x_i$, $s_{\rho x} = r_{\rho x} + c\rho x_i$, $s_{\eta x} = r_{\eta x} + c\eta x_i$ を計算する．$\sigma = (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ を出力する．
- $\mathsf{Td}(gpk, ak, M)$: $gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$, $ak = \xi$ とする．$t_M = H_1(M)^\xi$ を出力する．
- $\mathsf{GVf}(gpk, apk, M, \sigma)$: $gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$, $apk = y$, $\sigma = (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ とする．$R'_1 = u^{s_\alpha} T_1^{-c}$, $R'_2 = v^{s_\beta} T_2^{-c}$, $R'_3 = h^{s_\alpha + s_\beta} T_3^{-c}$, $R'_4 = e(T_4, g_2)^{s_x} e(\bar{g}_1, \omega)^{-s_\alpha} e(\bar{g}_1, g_2)^{-s_{\alpha x}} e(\bar{g}_2, \omega)^{-s_\beta} e(\bar{g}_2, g_2)^{-s_{\beta x}} e(g_1, \omega)^{-s_\eta} e(g_1, g_2)^{-s_{\eta x}} (e(g_1, g_2)/e(T_4, \omega))^{-c}$, $R'_5 = g_1^{s_\rho} T_5^{-c}$, $R'_6 = e(y, H_1(M))^{s_\rho} e(g_1, g_2)^{-s_\eta} T_6^{-c}$, $R'_7 = T_1^{s_x} u^{-s_{\alpha x}}$, $R'_8 = T_2^{s_x} v^{-s_{\beta x}}$, $R'_9 = T_5^{s_x} g_1^{-s_{\rho x}}$, $R'_{10} = T_6^{s_x} e(y, H_1(M))^{-s_{\rho x}} e(g_1, g_2)^{s_{\eta x}}$ を計算する．$c = H_2(M, T_1, \ldots, T_6, R'_1, \ldots, R'_{10})$ であれば 1 を，それ以外であれば 0 を出力する．
- $\mathsf{Open}(gpk, apk, ok, M, \sigma, t_M)$: $gpk = ((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2), u, v, h, \bar{g}_1, \bar{g}_2, \omega, H_1, H_2)$, $apk = y$, $ok = (\xi_1, \xi_2, \xi_3, \{e(A_i, g_2)\}_{i \in [1,n]})$, $\sigma = (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ とする．$\mathsf{GVf}(gpk, apk, M, \sigma) = 0$ であれば $\perp$ を出力，それ以外では

$$e(T_4 / T_1^{\xi_1} T_2^{\xi_2} T_3^{\xi_3}, g_2) \cdot T_6 / e(T_5, t_M) = e(A_i, g_2)$$

をみたす $\exists i \in [1,n]$ について i を出力，そのような i が無い場合は $\perp$ を出力する．