平成 30 年度

修士学位論文

動的グループにおけるクラウドデータ共有を実現するためのデータ所持証明方式の拡張

Extention Provable Data Possession method for Cloud Data Sharing in Dynamic Groups

1215090　　多田 菜南

指導教員　　清水 明宏

2019 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報学コース

要 旨

動的グループにおけるクラウドデータ共有を実現するためのデータ所持証明方式の拡張

多田 菜南

コールドデータと呼ばれる頻繁なアクセスは無いが，長期的な保存が必要なデータに注目が集まっている．コールドデータの具体例としてはアーカイブと呼ばれる長期的に記録しておく必要のある情報が挙げられる．また，記録を管理する際はは記録管理者だけでなく組織内の複数人に責任を割り当てることが望ましいとされている．アーカイブ等を電子的に保存する流れは広まっており，2020 年までにパブリッククラウドにはコンシューマデバイスよりも多くのデータが保存されることが予想されている．クラウドストレージは安価なストレージを提供するが，利用者はデータを完全に管理することができなくなり，そのデータのセキュリティはクラウドストレージに依存する．大量のコールドデータをクラウドストレージに委託する場合，これらのデータを完全にクラウドストレージが所持している証明が必要になる．

S-PDP はクラウドストレージ上のデータに破損がないかデータサイズに関係なく規定回数内であれば効率的に検証できるデータ所持証明方式である．この方式は対称鍵暗号を用いており，許可のないデータ変更等の不正行為を検知できるがデータ検証を行えるのはそのデータの所有者 1 人である．そこで，S-PDP を拡張し，クラウドストレージ上のデータをグループで検証可能な方式を提案する．動的なグループを想定し，グループメンバーの変更があった場合，データ所有者はグループ鍵の更新を実行するだけでよい．

キーワード　クラウドストレージ，データ所持証明，動的グループ

Abstract

Extention Provable Data Possession method for Cloud Data Sharing in Dynamic Groups

Nana TADA

Data that requires long-term preservation like archives and does not have frequent access is called cold data. To record important information, it is necessary to assign responsibility not only to the record manager but also to multiple people in the organization. Also, electronically storing cold data is widespread, and it is expected that more data will be stored in the public cloud than consumer devices by 2020. Cloud storage provides inexpensive storage, but the user can not fully manage the data and the security of that data depends on the cloud storage provider.

When entrusting a large amount of cold data to cloud storage, it is necessary to prove that these data are completely possessed by cloud storage. S-PDP is a provable data possession scheme that efficiently verify corruption of data on cloud storage as long as it is within the specified number of times regardless of data size. This scheme is used Symmetric key cryptosystem, and it is only the owner of the data that can detect fraudulent activity such as unauthorized data change. Therefore, we extend S-PDP and propose a method that can verify data in cloud storage by group. Assuming a dynamic group, if there is a change in group members, the data owner need only perform the group key update.

key words Cloud Storage, Provable Data Possession, Dynamic Group

目次

図目次

表目次

第 1 章

はじめに

1.1　目的

クラウドベースの大量データ向けストレージサービスが登場している．クラウドトレージは，クラウドコンピューティングによって提供されるサービスの 1 つで，データを維持，管理，リモートでバックアップしてネットワークを介してクライアントが利用できる．クラウドストレージは安価なストレージサービスを提供するが，企業や個人がデータをクラウドストレージに保存する場合，クライアント自身はデータを完全に管理することができなくなる．そのため，クライアントのデータはクラウドストレージプロバイダの情報セキュリティに依存することになる．

しかし，クラウドストレージなどの外部に保存されたデータは，人的エラー，ソフトウェアのバグ，ハードウェアの誤動作，悪意のある攻撃など，さまざまな理由で破損する可能性がある．特にクラウドが直面する脅威は，外部からだけでなく，クラウドの脆弱性を利用して内部者が危害を加えることもできる．そのため，悪意のある内部攻撃者や信頼性の低いクラウドストレージなどを想定し，外部に保存されたデータの情報セキュリティを考えることは重要である．

データが全て存在し，破損や不整合がないことを保証することをデータの完全性といい，情報セキュリティの基本的な要素の一つである．クラウドストレージなどの外部に保存されたデータも同様にデータの完全性を維持する必要がある．信頼できないクラウドストレージの中には，データ侵害をクライアントに隠蔽したり，使用頻度の低いデータやアクセスの少ないデータを削除してスペースを空けることがある．これらの問題が発生した時，クラウド

ストレージプロバイダが意図的にデータが破損しているという事実の隠蔽を防ぐため，クラウドに保存されたデータが正しく維持されているか定期的にチェックする必要がある [1].

近年，コールドデータと呼ばれるあまり頻繁なアクセスは無いが，長期的な保存が必要なデータに注目が集まっている．コールドデータの具体例としてはアーカイブと呼ばれる長期的に記録しておく必要がある重要な記録が挙げられる．法律によって記録が義務付けられているものもあり，株主総会議事録や決算書，請求書，建築図面などがある．加えて，企業にとって大きな問題（個人情報漏洩や労働争議）が発生した際に，業務内容についての説明責任を果たすために個人情報管理帳や研究データ，電子メール，システムログ，勤怠管理帳なども長期保存されるべきデータに該当する．これらのアーカイブなどを電子的に保存する流れは広まっており，2020 年のコールドデータ市場は 2000 エクサバイトと，2015 年と比較し約 4 倍になると予想されている [2,3]． コールドデータが世界的にも増えてくることが予想されている中，上記で挙げたような大量のデータ保存を安価なクラウドストレージに委託する場合，ストレージがこれらのデータを所持している保証が必要になる．また，コールドデータは頻繁なアクセスはないことが想定されるため，データの利用時にデータが削除されていたり，変更されていることが検知できるだけでは損失したデータを回復させるためには遅すぎる可能性がある [4]．そこで，クライアント側で大量のデータがクラウドストレージに保存されていることを効率的に検証する必要がある．

また，記録管理の標準規格 ISO/IEC15489-1 では，記録の責任は記録管理者だけでなく，経営者や部門ごとの責任者，システム管理者，業務の一環として記録を作成する人など，組織内の複数人の関係者に割り当てることが望ましいとしている．つまり，全員に管理する責任と権利がある．また，これらの記録情報の保存場所は要求されたときに保存場所が突き止められ，追跡できることにも言及されている．同時に，コールドデータは重要な記録であることが多いため，そのデータの所在が漏えいしてしまうことは，悪意のある第三者による攻撃を誘引する要因になり得る．そこで，データをストレージが所有していることを検証できるのはデータ所有者が信頼できる人物である必要がある．

1.2　本論文の構成

本論文では，コールドデータのような大量の静的データをクラウドストレージが所持していることを効率的に検証可能な方式を提案する．また，データ所有者が信頼できる人物にも検証権限を付与するために，信頼できる人物には検証用のグループ鍵を配布する．また，グループ鍵はユーザの退出を考慮し，グループ鍵の更新を考える必要がある．そのため，提案方式では動的なグループでのデータ検証を想定する．第 2 章では，データ検証とグループにおける鍵管理方式の関連研究と S-PDP に適用するために必要なグループ鍵管理要件についてまとめる．第 3 章では，提案方式で用いるいくつかの技術について説明を行う．第 4 章では，S-PDP に適用するグループ鍵管理方式を説明し，S-PDP への具体的な適用について詳細に示す．第 5 章では，提案方式が S-PDP に適したグループ鍵管理要件を満たしていることを示す．そのためにクラウドストレージが不正にグループ鍵を取得できないことや，失効ユーザが不正に検証を実行できないことを示す．また，S-PDP に提案方式を適用した場合のストレージと通信コストについて評価する．最後に，第 6 章では本論文の成果をまとめる．

第 2 章

関連研究

2.1 データの完全性検証についての関連研究

クラウドストレージに大量のデータを保存する場合，データはクラウドストレージのセキュリティに依存することになり，様々な脅威が考えられる．情報セキュリティの要素の一つであるデータ完全性については，クラウドストレージ管理者，マルウェア，不正なクラウドサービスプロバイダー，または別の場所へのデータの物理的な移動などにより，データの完全性が損なわれる可能性がある．したがって、リソースが限られているクライアントがクラウドストレージ上の大量のデータを完全性を保ったまま保存されているか効率的に検証できる必要がある [5].

これまでに信頼できないクラウドストレージ内のデータの完全性を検証する方式がいくつか提案されている．データの完全性を検証する方式には大きくプライベート監査と公開監査の 2 つの方式に大きく分かれている．プライベート監査はクラウドストレージ上のデータの所有者自身が検証タスクを実行する方式である．公開監査は信頼できる第三者 (TPA) を前提とし，TPA がデータ所有者に代わって検証タスクを実行する．

まず，プライベート監査によるデータの検証方式を紹介する．Ateniese ら [4] が提案した PDP（Provable Data Possession scheme）ではクライアントは信頼できないストレージにデータを保存する．この方式では，クライアントはストレージに格納されているデータの改ざんや破損がない正しいデータかどうかをデータサイズに関係なく一定のブロックの検証で確認できる．このシステムは，ストレージからブロックのランダムなセットをサンプリングすることによって所持証明を生成する．そのため，PDP は許可されていないデータの変

更などの不正行為に対する強力な抑止力として働く．また，クライアントは証明を検証するために検証メタデータを保持する必要があり，この方式はデータ変更のない静的なデータにのみ対応している．また，Ateniese ら [6] はその後，対称鍵暗号を使用した S-PDP 方式も提案している．この方式では一部の動的なデータ操作（削除・追加）に対応している．また，PDP と違いクライアントはマスター鍵のみを所持しておけばよい．この方式は対称鍵暗号に基づいているため，データ検証を行えるのはマスター鍵を持つデータの所有者 1 人である．また，S-PDP は検証回数をクラウドストレージへデータを登録する際に予め決めておく必要があり，検証タスクを実行する毎に検証可能回数が減っていく．S-PDP は静的なデータを対象としている．Chen ら [7] は MAC に基づくデータ検証アルゴリズムを提案している．この方式では S-PDP と同様にデータの検証に第三者を介さず，クライアントが保持する少量のローカルデータから，予め決められた回数だけデータの完全性を検証できる．この方式ではクラウドストレージ上に保存されたデータ内に検証用データブロックを埋め込む．そのため，検証に使われたデータブロックは不要なデータとしてクラウドストレージ上に残るため，不必要なデータがクラウドストレージの容量を浪費する問題として依然残っている．

次に，公開監査によるデータ完全性の検証方式を紹介する．Xu ら [8] はクラウドストレージの完全性検証を実現するために CBF(Count Bloom Filter) を使用することを提案し，メッセージ認証コードに基づく検証方式を実装した．しかしこの方式では誤判定があり，信頼できる第三者 (TPA) 上に保存しておく情報量が増加する．よって，クライアントのデータ検証タスクを TPA へ委譲することを考慮すると，第三者の参加もストレージコストを増大させ，同時に検証者へのデータ漏えいの可能性もある．公開監査では公開検証者が検証タスクを実行する際に，クライアントのデータ漏えいが無いことも求められる．Wang ら [9] はクラウドストレージ上の共有データの完全性公開検証のための方式 Panda を提案した．この方式は各データブロック毎に共有ユーザが署名をつけて保存するため，動的なデータに対応している．データ共有ユーザを効率的に失効するために準同型認証プロキシ再署名と呼ばれる方式を用いている．この方式は失効したユーザとクラウドストレージとの結託が無い

ことを前提としている．

2.2　動的なグループにおける鍵管理についての関連研究

既存のクラウドコンピューティング上で実現されている動的グループのデータ共有を実現する鍵管理方式について述べる．

Zhu ら [10] は動的グループにおけるクラウドへのデータ漏えいがないデータ共有方式を提案している．エンティティはクラウドストレージ，グループ管理者，グループメンバーである．クラウドストレージは信頼できないエンティティとして，動的グループでデータ共有が可能である．この方式は Access Contol Polynomial (ACP) scheme [11] で提案されている多項式関数（Polynomial Function）を応用し，ユーザの参加や失効時に効率的なグループ鍵配布を実現している．ユーザの参加や失効時，グループの他のユーザは秘密鍵を再計算する必要がない．この方式は効率的な鍵管理を実現しているが，古い鍵で暗号化された暗号文はグループ管理者が新しい鍵で暗号化し直してクラウドストレージへ再登録する必要がある．このため，この鍵管理方式はグループ管理者の負担が大きく，常にオンラインで待機する必要がある．

Song ら [12] は暗号化されたクラウドストレージ上のデータの動的なグループにおける共有のためのグループ鍵管理方式を提案している．この方式ではクラウドストレージは半信頼サーバとしており，信頼できる第三者やセキュアチャネルでの通信が不要である．グループ管理者はグループのユーザに対し，グループ鍵を各ユーザの秘密鍵で暗号化して配布する．グループ鍵の配布のためにグループ管理者は各ユーザのための再暗号化鍵を生成し，暗号化されたグループ鍵と各ユーザの再暗号化鍵を用いてクラウドストレージ経由でグループ鍵を配布する．

クラウドストレージサービスを用いたデータ共有における

また，共有データのアクセス制御に適した暗号技術として属性ベース暗号 (Attribute-Based Encryption: ABE) [13] がある．CP-ABE [14]，[15] は ABE の一種であり暗号化す

るときにアクセス構造を定めることで，暗号文ごとに復号できる属性に基づいたグループで指定することができる．そのため CP-ABE は，共有データの細かいアクセス制御に適している．よって，CP-ABE はクラウドに格納されたデータへの細かいアクセス制御ができるため，動的グループでのデータ共有に適している．既存の ABE の研究のほとんどはいかなる不正も行わない信頼できる第三者機関により属性や鍵発行を委託する必要がある．これを ABE の鍵エスクロー問題と呼ぶ．鍵エスクロー問題を軽減するため，階層型や複数の第三者機関を想定するもの [16] もあるが，環境構築コストが高くなる．

2.3　提案方式の設計目標

提案方式ではクラウドストレージに保存するデータとして静的なコールドデータを想定し，大量データが正しく保存されているかグループで効率的に検証可能な方式へ S-PDP を拡張する．動的グループで実現することを目指し，グループユーザの退出などを考慮し，失効について検討する．そこで，S-PDP の特徴とセキュリティ要件に合ったグループ鍵管理方式を提案する．グループで S-PDP を実現するための要件は以下の通りである．

信頼エンティティの最小限化

S-PDP はクラウドストレージを信頼できないエンティティとしており，グループ鍵管理においても信頼できるエンティティはデータ所有者とデータ所有者に認められたグループユーザだけとする．

動的グループでの効率的な鍵更新

ユーザの入退出がある動的グループでは，前方・後方秘匿性を実現するためにユーザの入退出の度にグループ鍵を新しくする必要がある．そのため効率的に鍵を更新する必要がある．この時，グループ鍵の更新は行うが，暗号文の暗号鍵更新は必要ないように実現する．信頼エンティティを最小限としているため，第三者の関与なしで，データ所有者やグループユーザの鍵更新による負担をできるだけ少なくすることが目的である．

失効ユーザの不正防止

2.3 提案方式の設計目標

S-PDP は検証できる回数が予め決められている．失効されたユーザは信頼できるエンティティではないため，失効ユーザが検証できると不正に検証回数を消費される可能性がある．そのため，提案方式ではグループから失効されたユーザは検証できないように鍵更新を行う．

第 3 章

前提知識

3.1 S-PDP（Scalable and Efficient Provable Data Possession）

S-PDP について説明する．S-PDP はデータ所有者がデータをクラウドストレージに登録する際，設定したい検証回数分だけ検証データを作成する．この検証回数を t とする．データ所有者がクラウドストレージに登録するデータファイルを D とし，ファイル D は d 個のブロックに分割されている．

以下では，データ所有者がクラウドストレージにデータと検証データ t 個を登録するデータ登録フェーズと i 個目の検証データをデータ所有者が検証する検証フェーズについて説明する．

3.1.1 データ登録フェーズ

データ登録フェーズではデータ所有者がクラウドストレージにデータと共に検証データを t 個登録する，i 回目 の検証データ作成について図 3.1 を用いて説明する．

1. ランダムな $kbit$ のマスター鍵 $W, Z, K \in \{0,1\}^k$ を選び，それぞれのマスター鍵を用いて疑似乱数生成器 $f_{Key}(\cdot)$ から 2 つの乱数 $f_W(i) = k_i$，$f_Z(i) = c_i$ を生成する．k_i はブロックの置換鍵，c_i はクラウドストレージが事前に検証データを生成できないようにするためのランダムノンスである．
2. 疑似乱数生成器 $g_{Key}(\cdot)$ を用いてランダムなインデックス番号 I_j を以下より求める．

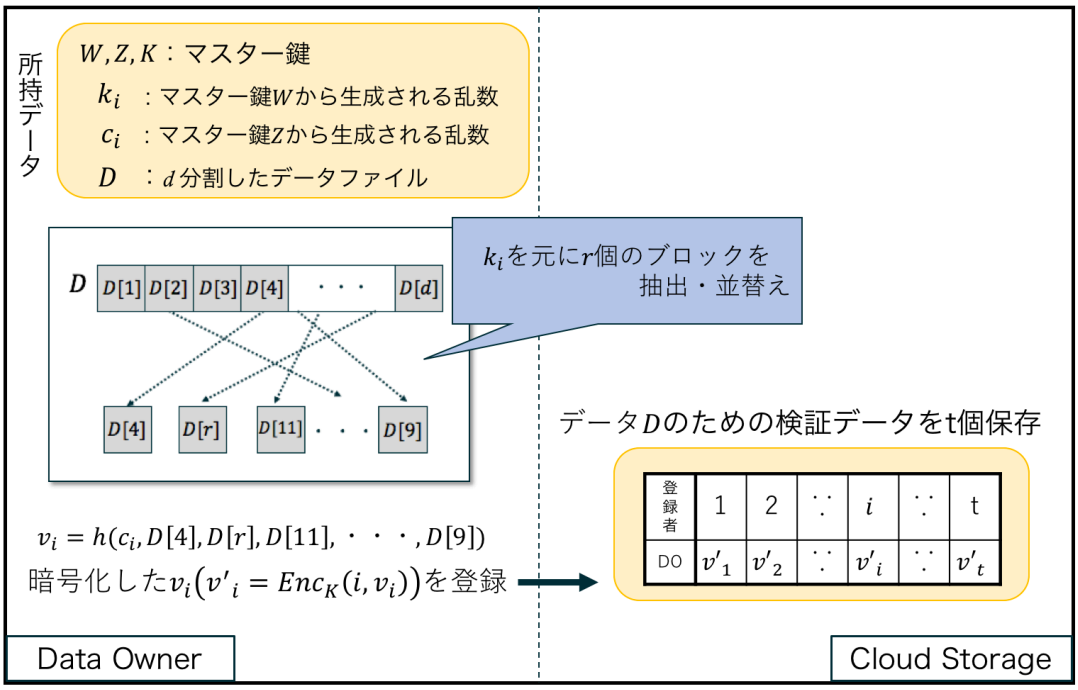

図 3.1　検証データ登録 (i 回目)

これにより，d 分割したデータファイル D から，ランダムに r 個のブロックをサンプリングし，並び替えることができる．

$$\{I_j \in [1, ..., d] | 1 \leq j \leq r\} \quad g_{k_i}(j) = I_j$$

3. ランダムノンス c_i とランダムに並べ替えられた r 個のデータブロック $\{D[I_1], D[I_2], ..., D[I_r]\} = \{D[4], D[r], ..., D[9]\}$ をハッシュ関数 h に入力し，ハッシュ値 v_i を得る．

$$v_i = h(c_i, D[I_1], D[I_2], ..., D[I_r])$$

4. マスター鍵 K を用いて (i, v_i) を暗号化 $v_i' = Enc_K(i, v_i)$ する．
5. ステップ 1-4 を t 回繰り返し，D と共に t 個の v_i' をクラウドストレージに登録する．

3.1.2　検証フェーズ

検証フェーズではデータ所有者がクラウドストレージに検証リクエストを送り，クラウドストレージから返ってきた検証データをデータ所有者が検証する．1 回目 のときの検証につ

3.1 S-PDP（Scalable and Efficient Provable Data Possession）

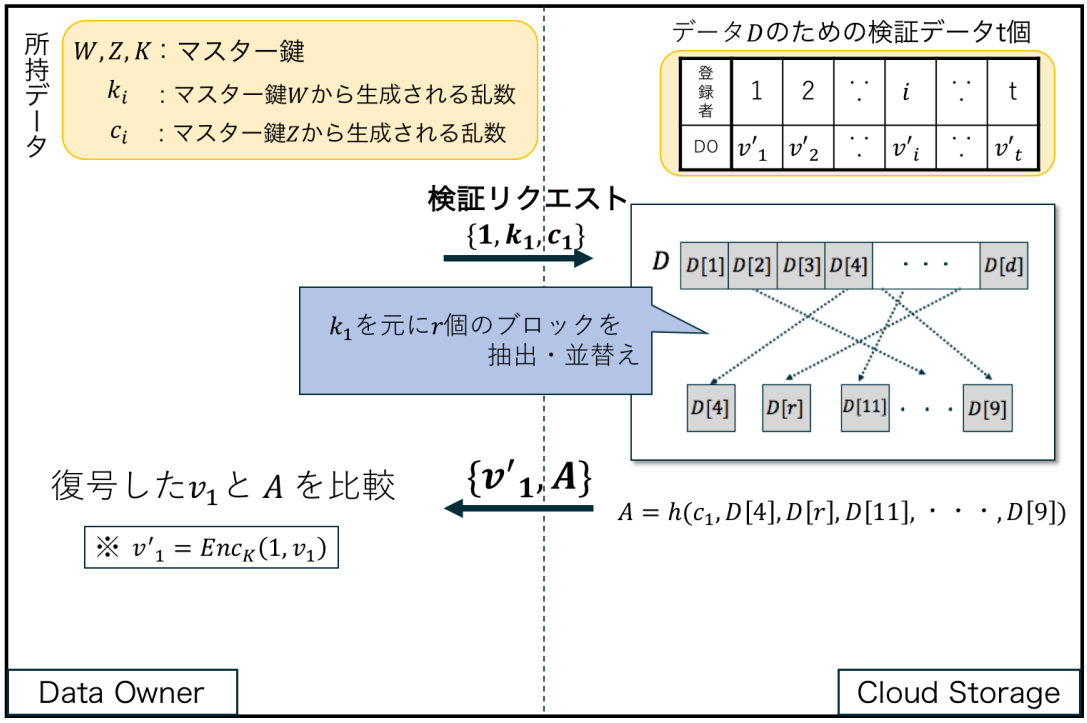

図 3.2　データ検証 ($i = 1$)

いて図 3.2 を用いて説明する．

1. データ所有者はマスター鍵 W, Z と検証回数 1 回目より，$f_W(1) = k_1, f_Z(1) = c_1$ を求める．$\{1, k_1, c_1\}$ をクラウドストレージへリクエストとして送信する．
2. クラウドストレージはデータ所有者から送られてきた k_1 から $g_{k_1}(j) = I_j$ を計算し，データファイル D からランダムなブロック $\{D[I_1], D[I_2], ..., D[I_r]\} = \{D[4], D[r], ..., D[9]\}$ を抽出する．

$$\{I_j \in [1, ..., d] | 1 \leq j \leq r\} \quad g_{k_1}(j) = I_j$$

3. クラウドストレージはデータ所有者から送られてきた c_1 と 2 で抽出した r 個のランダムブロックのハッシュ値 A を計算する．

$$A = h(c_i, D[I_1], D[I_2], \cdots, D[I_r])$$

4. クラウドストレージは保存しておいた検証データ $v_1{}'$ と Z をデータ所有者へ返す．
5. データ所有者はマスター鍵 K を用いて $v_1{}' = Enc_K(1, v_1)$ を復号し，v_1 を得る．

6. v_1 と A を比較し，等しければ検証が成功する．

3.1.3　S-PDP の評価

S-PDP の計算量・通信コスト・検知率についての評価を表 3.1 に示す．表中の r は 1 回の検証に利用するブロックの個数，b はデータファイルブロック d のサイズ，ρ はクラウドストレージ上のデータブロック d の破損率である．表から分かる通り，S-PDP は計算量とストレージコストがデータサイズ D に関係なく固定である例として，$D = 128GB, \rho = 1\%, b = 4KB, r = 512$ であるとき，$br = 2MB$ の通信コストで $128GB$ の内 1% のデータの破損を 99% 検知することができる．

表 3.1　S-PDP の評価

データ所有者計算量	クラウドストレージ計算量	通信コスト	検知率
$O(r)$	$O(r)$	$O(br)$	$1-(1-\rho)^{br}$

3.2　双線形写像

素数 q, 素数位数 q の 2 つの乗法巡回群 $\mathbb{G}_1, \mathbb{G}_2$，生成元 $g \in \mathbb{G}_1$ とする．双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ は以下の性質を満たす．

1. 計算可能性 ： 双線形写像 e を効率的なアルゴリズムで計算できる
2. $\forall u, v \in \mathbb{G}_1, \forall a, b \in Z_q$ では $e(u^a, v^b) = e(u, v)^{ab}$
3. 非退化性 ： $e(g, g) \neq 1$

3.3　プロキシ再暗号化 (PRE:Proxy Re-Encryption)

プロキシ再暗号化方式は，図 3.3 のように鍵 K_A を持つ人しか復号できない暗号化データ C_A を第三者によって復号することなく鍵 K_B を持つ人が復号できるような暗号文 C_B に変

3.3　プロキシ再暗号化 (PRE:Proxy Re-Encryption)

換することを可能にする暗号技術である．この変換処理を行う人 (代理人) は平文 m の内容を知ることはできない [17]．Blaze ら [18] は BBS を提案した．この方式は素数位数 q の 2 つの巡回群における双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ を用いた方式である．BBS について簡単に説明する．システムパラメータを $g \in \mathbb{G}_1$，$\mathbb{Z} = e(g, g) \in \mathbb{G}_2$ とする．

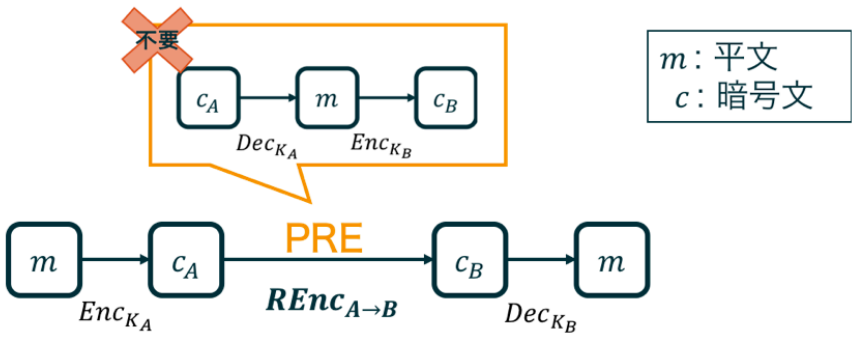

図 3.3　プロキシ再暗号化

鍵生成

ユーザ Alice は乱数 $\alpha \in Z_q$ を選び，$sk_A = \alpha$，$pk_A = g^\alpha$ をそれぞれ Alice の秘密鍵，公開鍵とする．ユーザ Bob は乱数 $\beta \in Z_q$ を選び，$sk_A = \beta$，$pk_A = g^\beta$ をそれぞれ Bob の秘密鍵，公開鍵とする．

PRE 鍵生成

ユーザ Alice は Bob の公開鍵 $pk_B = g^\beta$ を用いて Alice 宛の暗号文から Bob 宛の暗号文に変換する PRE 鍵 $rk_{A\to B} = g^{\frac{\beta}{\alpha}}$ を生成する．

第 1 レベル暗号化

Alice が平文 $m \in \mathbb{G}_2$ を秘密鍵 $sk_A = \alpha$ で暗号化する．Alice は乱数 $k \in Z_q$ を選び，第一レベル暗号文 $C_1 = (\mathbb{Z}^{\alpha k}, m\mathbb{Z}^k)$ を生成する．

第 2 レベル暗号化

平文 $m \in \mathbb{G}_2$ の第二レベル暗号文は $C_2 = (g^{\alpha k}, m\mathbb{Z}^k)$ である．

再暗号化

第 2 レベル暗号文 $C_2 = (g^{\alpha k}, m\mathbb{Z}^k)$ を PRE 鍵 $rk_{A\to B} = g^{\frac{\beta}{\alpha}}$ を用いて Alice から Bob の第 1 レベル暗号文 $C_1 = (\mathbb{Z}^{\beta k}, m\mathbb{Z}^k)$ に変換する.

$$\begin{aligned}(e(rk_{A\to B}, g^{\alpha k}), m\mathbb{Z}^k) &= (e(g^{\frac{\beta}{\alpha}}, g^{\alpha k}), m\mathbb{Z}^k) \\ &= (e(g,g)^{\beta k}, m\mathbb{Z}^k) \\ &= (\mathbb{Z}^{\beta k}, m\mathbb{Z}^k)\end{aligned}$$

復号

第 1 レベル暗号文 $C_1 = (\mathbb{Z}^{\beta k}, m\mathbb{Z}^k)$ は秘密鍵 $sk_B = \beta$ を持っている Bob のみ復号できる.

$$m = m\mathbb{Z}^k / \mathbb{Z}^{\beta k \frac{1}{\beta}}$$

第 2 レベル暗号文 $C_2 = (g^{\alpha k}, m\mathbb{Z}^k)$ を与えられた時, 秘密鍵 $sk_A = \alpha$ を持 Alice のみ復号できる.

$$m = m\mathbb{Z}^k / e(g^{\alpha k}, g)^{\frac{1}{\alpha}}$$

3.4 仮定

定義 Basic Diffie-Hellman Problem（BDHP）仮定 [10]

代数曲線上の点 P とし，$r \in Z_q$ に対して $r \cdot P$ を計算することは容易であるが，P，$r \cdot P$ を与えられたとき，離散対数問題より r を計算することは困難な仮定を BDHP 仮定と呼ぶ.

定義 Descrete Logarthmic Problem（DLP）仮定

$x \in Z_q$ であり，g, $g^x \in \mathbb{G}_1$ が与えられたとき，x を求めることが困難な仮定を DLP 仮定と呼ぶ.

第 4 章

提案方式

4.1 提案方式の概要

4.1.1 提案方式の構成

提案方式はクラウドストレージ (CS)，データ所有者 (DO), グループユーザの 3 者で構成される．構成図を図 4.1 に示す．CS は DO のデータを保存する信頼できないサーバである．DO は CS 上にデータを保存する際，検証データも生成する．また，グループユーザの入退出も管理する．グループユーザ u_A は DO にデータの検証が認められたグループ G のユーザとする．

4.1.2 前提条件

クラウドストレージはデータの削除などの不正行為の可能性のあるサーバとするが，提案方式のプロトコルには従う．データ所有者はクラウドストレージにデータを保存するユーザであるため，信頼できる．また，データ所有者に認証されているグループユーザは信頼できるため，クラウドストレージ上のデータ検証が認められている信頼できるユーザである．また，データ所有者にグループから失効されたユーザは信頼できないとする．

また，提案方式で用いられる表記を表 4.1 以下に示す．

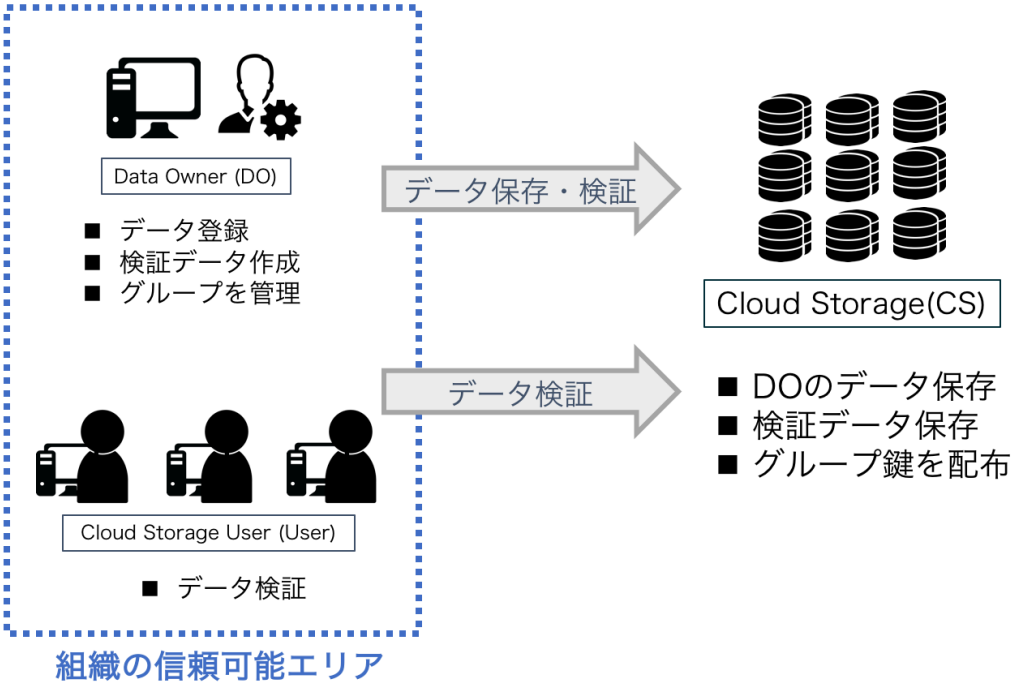

図 4.1　提案方式の構成図

4.2　S-PDP に適した鍵管理方式

S-PDP は対称鍵暗号の鍵を用いて検証データを作成・検証するため検証できるユーザは鍵を所持するデータ所有者 DO のみであった．提案方式では，S-PDP の検証に使用する検証鍵を S とした時，検証鍵 S は PRE により暗号化され，プロキシ再暗号の第二レベル暗号文として，CS へ保存する．CS は検証リクエストが来た際，PRE 鍵によりグループユーザが復号可能な第一レベル暗号文へ再暗号化する．この暗号文をグループユーザへ送ることでユーザは検証鍵 S を取得でき，グループでの検証を実現する．提案方式では，グループ鍵の配布に多項式関数，検証鍵の配布にプロキシ再暗号化を使用することでグループ鍵管理を実現する．本稿で提案するグループ鍵管理方式のグループ鍵生成，グループ鍵配布，検証鍵配送，グループユーザの失効について以下より説明する．

表 4.1 提案方式の表記

CS	クラウドストレージ
DO	クラウドストレージにデータを保存するデータ所有者
u_A	グループのユーザ $A(u_0 = DO)$
ID_A	u_A の識別子
ID_G	グループ G の識別子
x_A	u_A の秘密値
$h(\cdot)$	ハッシュ関数
$Enc_{Key}(\cdot)$	鍵 Key を用いた対称鍵暗号化アルゴリズム
$g_{Key}(j)$	置換関数 $I_j \quad \{I_j \in [1, ..., d] \mid 1 \leq j \leq r\}$
K_G	グループ G のグループ鍵
G_{add}	素数位相 q の加法巡回群
$rk_{A \to B}$	復号できる鍵を A から B の公開鍵に変更するための PRE 鍵
UL	グループユーザリスト
GL	グループリスト
VL	検証データリスト

表 4.2 CS が保持する UL

ユーザ ID	グループ ID
ID_A	ID_G
...	...

4.2.1 グループ鍵生成

図 4.2 で示すように，この操作はデータ所有者 DO と CS によって実行される．この操作は DO がグループ鍵 K_G を生成し，CS へグループ鍵 K_G と PRE 鍵 $rk_{DO \to G} = g^{K_G/\pi_0}$ を登録する．以下にグループ鍵生成の手順を示す．

1. DO は $U_j = h(ID_j, x_j)$， グループメンバー数 m とし，グループのための多項式関数

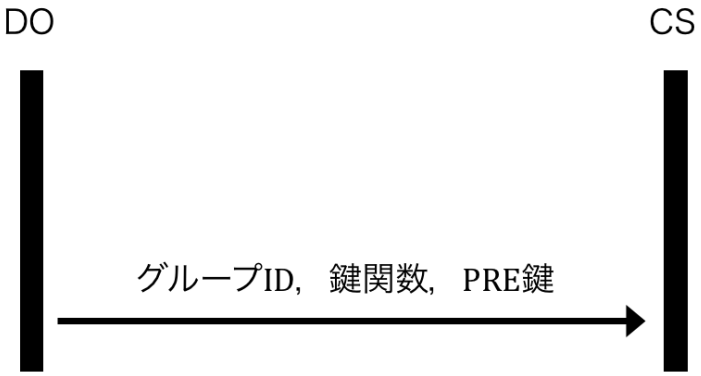

図 4.2　グループ鍵生成

表 4.3　グループリスト GL

グループ ID	鍵関数	PRE 鍵
ID_G	EK_G	$rk_{DO\to G}$
…	…	…

$f_p(x) = \prod_{j=1}^{m}(x - U_j)$ を構成する．

$$f_p(x) = \prod_{j=1}^{m}(x - U_j)$$
$$= \sum_{i=0}^{m} a_i x^j \quad (mod \quad q)$$

加法巡回群の生成元 $P \in \mathbb{G}_{add}$ と多項式関数 f_p より，$\{Y_0, ..., Y_m\} = \{P^{a_0}, ..., P^{a_m}\}$ を求める．

2. DO はグループ鍵 $K_G \in Z_q^*$ を選び，$EK_G = \{K_G \cdot Y_0, Y_1 \cdots, Y_m\}$ とする．また，DO は秘密鍵 $sk_{DO} = \pi_0$ を用いて，PRE 鍵 $rk_{DO\to G} = g^{K_G/\pi_0}$ を生成する．
3. 生成した EK と $rk_{DO\to G}$ を ID_G と表 4.2 の CS 用のユーザリスト UL を CS へ送る．
4. CS は DO から届いた UL を保存し，EK と $rk_{DO\to G}$ を表 4.3 で示すグループリストへ GL(グループ ID, 鍵関数, PRE 鍵) = $(ID_G, EK_G, rk_{DO\to G})$ として保存する．

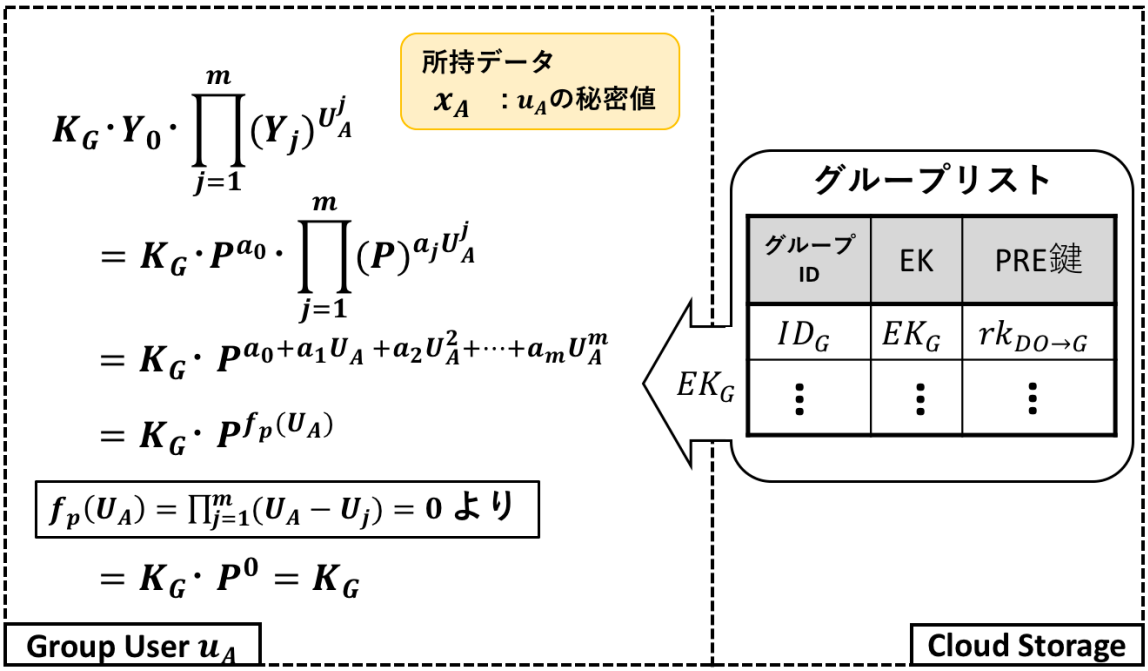

図 4.3　グループ鍵の取得

4.2.2　グループ鍵配布

この操作は，図 4.3 に示すように，グループユーザとクラウドストレージ CS によって実行される．グループ G のユーザ u_A は CS から鍵関数 EK_G を受け取り，グループ鍵 K_G を取得する．以下にグループ鍵配布の手順を示す．

1. u_A と CS は相互認証を確立し，CS はグループリスト GL を参照して u_A のグループ ID に該当する鍵関数 EK_G を u_A へ返す．
2. u_A は，自身の ID_A と秘密値 x_A より $U_A = h(ID_A, x_A)$ を計算し，以下の式より K_G を得る．秘密値 x_A は DO により配布されるユーザ毎の秘密の値である．

$$
\begin{aligned}
K_G \cdot Y_0 \cdot \prod_{j=1}^{m} (Y_j)^{{U_A}^j} &= K_G \cdot P^{a_0} \cdot \prod_{j=1}^{m} (P)^{a_j {U_A}^j} \\
&= K_G \cdot P^{a_0 + a_1 U_A + a_2 {U_A}^2 + \cdots + a_m {U_A}^m} \\
&= K_G \cdot P^{f_p(U_A)}
\end{aligned}
$$

ここで,

$$\begin{aligned} a_0 + a_1 U_A + a_2 {U_A}^2 + \cdots + a_m {U_A}^m &= \sum_{i=0}^{m} a_i {U_A}^j \\ &= f_p(U_A) \\ &= \prod_{j=1}^{m} (U_A - U_j) \\ &= 0 \end{aligned}$$

であるから,

$$K_G \cdot P^0 = K_G$$

4.2.3 検証鍵配送

提案方式では S-PDP の検証に使用する検証鍵を S とした時，S をプロキシ再暗号化により生成した第二レベル暗号文を CS 上で保存する．CS は検証のリクエストがあった場合グループ鍵 K_G を持つ人だけ復号できる第一レベル暗号文へ再暗号化される．図 4.4 は DO が検証鍵 S を CS へ登録し，グループ G のユーザ u_A が検証鍵 S を取得する流れである．以下に検証鍵配送の手順を示す.

1. DO はランダムな $k \in Z_q$ を選び，検証鍵 S を DO の公開鍵 g^{π_0} を用いて $C^2 = (g^{\pi_0 k}, S\mathbb{Z}^k)$ を求め，C^2 を CS へ送る.
2. CS はグループユーザが検証者 u_A から検証リクエストを受けると，C^2 と PRE 鍵 $rk_{DO \to G} = g^{\frac{K_G}{\pi_0}}$ を用いて C^1 を求め，u_A へ送る.

$$\begin{aligned} C^1 &= (e(rk_{DO \to K_G}, g^{\pi_0 k}), S\mathbb{Z}^k) \\ &= (e(g,g)^{K_G k}, S\mathbb{Z}^k) \\ &= (\mathbb{Z}^{K_G k}, S\mathbb{Z}^k) \end{aligned}$$

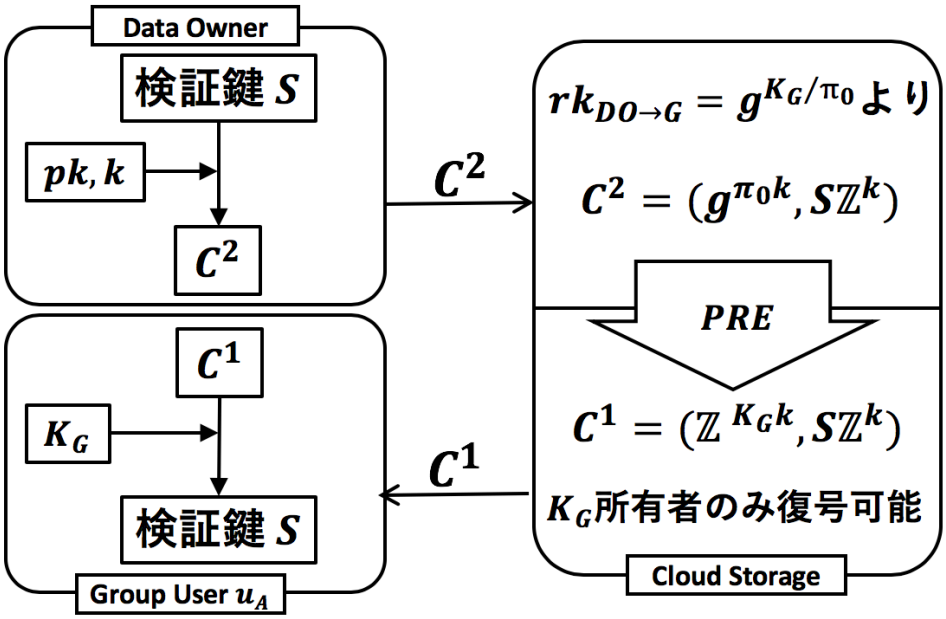

図 4.4　検証鍵 S の取得

3. u_A は C^2 から，自身の持つグループ鍵 K_G で S を取り出す．

$$\begin{aligned} S &= S\mathbb{Z}^k / \mathbb{Z}^{\frac{K_G k}{K_G}} \\ &= S\mathbb{Z}^k / \mathbb{Z}^k \\ &= S \end{aligned}$$

4.2.4　グループユーザの失効

この操作はグループ G からユーザ u_r が退出するとき，DO がグループ内のユーザ u_r を失効させるために実行する．DO が新しいグループ鍵 K'_G を作り，グループ鍵を CS に再登録するまでの手順を以下に示す．

1. DO は $U_j = h(ID_j, x_j)$，グループメンバー数 $m-1$ とし，u_r が退出した新しいグ

ループ G のための多項式関数 $f'_p(x) = \prod_{j=1}^{m-1}(x - U_j)$ を構成する．

$$f'_p(x) = \prod_{j \neq r \; j=1}^{m-1} (x - U_j)$$
$$= \sum_{i=0}^{m-1} a_i x^j \quad (mod \quad q)$$

加法巡回群の生成元 $P \in \mathbb{G}_{add}$ と多項式関数 f'_p より，$\{Y_0, ..., Y_{m-1}\} = \{P^{a_0}, ..., P^{a_{m-1}}\}$ を求める．

2. DO は新しいグループ鍵 K'_G を選び，新しい鍵関数を $EK'_G = \{K'_G \cdot Y_0, Y_1, \cdots, Y_{m-1}\}$ とする．また，DO は秘密鍵 $sk_{DO} = \pi_0$ を用いて，PRE 鍵 $rk_{DO \to G'} = g^{K_G{}'/\pi_0}$ を新しく生成する．
3. 生成した EK'_G と $rk'_{DO \to G}$ を ID_G と表 4.2 の CS 用ユーザリスト UL から u_r を取り消したものを CS へ送る．
4. CS は DO から届いた UL を保存し，EK'_G と $rk'_{DO \to G}$ を表 4.3 で示すグループリストへ GL(グループ ID, 鍵関数, PRE 鍵) = $(ID_G, EK'_G, rk_{DO \to G'})$ として更新する．

新しいグループ鍵のユーザへの配布は，4.2.2 節 グループ鍵配布フェーズにより行われる．鍵更新のあったグループのユーザがデータ検証をリクエストするために CS と認証を確立した時に鍵関数に更新があった場合，新たな鍵関数を配布することで完了する．

4.3 S-PDP の拡張

S-PDP を拡張した提案方式は以下の 7 つのアルゴリズムによって構成される．各アルゴリズムごとに，提案方式を説明する．

1. セットアップ
2. ユーザ登録
3. グループ鍵生成

4. グループ鍵配布
5. データ保存
6. 検証
7. グループユーザの失効

4.3.1 セットアップ

データ所有者 DO は，この操作を実行する．まず，双線形写像システム $Sys = (q, \mathbb{G}_1, \mathbb{G}_2, e(\cdot,\cdot))$ を設定する．DO は $g \in \mathbb{G}_1$ と乱数 $\pi_0 \in Z_q^*$ を選び，DO の公開鍵 $pk_{DO} = g^{\pi_0}$, 秘密鍵 $sk_{DO} = \pi_0$ とする．

4.3.2 ユーザ登録

この操作はユーザ u_A とデータ所有者 DO によって実行される．ユーザ u_A が DO へグループユーザ登録を依頼し，ユーザ固有の秘密値 x_A を配布してもらう手順を以下に示す．

表 4.4 DO が保持する UL

ユーザ ID	秘密値	グループ ID
ID_A	x_A	ID_G
…	…	…

1. u_A は乱数 $t_1 \in Z_q$ を選び，自身の (ID_A, pk, t_1) を DO へ送る．
2. DO は $r \in Z_q$ を選び，$R = e(g,g)^r, U = g^{r+\pi_0 \cdot t_1 \cdot h(pk||ID_A)}$ を計算し，R と U を u_A へ送る．
3. u_A は DO から受け取った R, U より，$R \cdot e(g^{r+\pi_0 \cdot t_1 \cdot h(pk||ID_A)}, g^{\pi_0}) == e(U, g)$ を比較する．等しければ，乱数 $t_2 \in Z_q{}^*$ を選び，$AEnc_{sk}(ID_A, t_1)$ と ID_A, t_2 を DO へ送る．
4. DO は $AEnc_{sk}(ID_A, t_1)$ を復号し，1 で送られてきた t_1 と同じであれば，$x_A \in Z_q$ を

選び，表 4.4 のユーザリスト UL(ユーザ ID, 秘密値, グループ ID) = (ID_A, x_A, ID_G) に登録する．$AEnc_{pk}(ID_A, x_A, t_2)$ を u_A へ送る．

5. u_A は $AEnc_{pk}(ID_A, x_A, t_2)$ を復号し，3 で自身が送信した t_2 であるか確認する．正しければ x_A をローカルに保存する．

4.3.3 グループ鍵生成

このフェーズは 4.2.1 節 のグループ鍵生成と同じである．以下に手順を示す．

1. DO は $U_j = h(ID_j, x_j)$， グループメンバー数 m とし，グループのための多項式関数 $f_p(x) = \prod_{j=1}^{m}(x - U_j)$ を構成する．

$$f_p(x) = \prod_{j=1}^{m}(x - U_j)$$
$$= \sum_{i=0}^{m} a_i x^j \quad (mod \quad q)$$

加法巡回群の生成元 $P \in \mathbb{G}_{add}$ と多項式関数 f_p より，$\{Y_0, ..., Y_m\} = \{P^{a_0}, ..., P^{a_m}\}$ を求める．

2. DO はグループ鍵 $K_G \in Z_q^*$ を選び，$EK_G = \{K_G \cdot Y_0, Y_1 \cdots, Y_m\}$ とする．また，DO は秘密鍵 $sk_{DO} = \pi_0$ を用いて，PRE 鍵 $rk_{DO \to G} = g^{K_G/\pi_0}$ を生成する．
3. 生成した EK と $rk_{DO \to G}$ を ID_G と表 4.2 の CS 用のユーザリスト UL を CS へ送る．
4. CS は DO から届いた UL を保存し，EK と $rk_{DO \to G}$ を表 4.3 で示すグループリストへ GL(グループ ID, 鍵関数, PRE 鍵) = $(ID_G, EK_G, rk_{DO \to G})$ として保存する．

4.3.4 グループ鍵配布

このフェーズは 4.2.2 節 のグループ鍵配布と同じである．以下に手順を示す．

1. u_A と CS は相互認証を確立し，CS はグループリスト GL を参照して u_A のグループ ID に該当する鍵関数 EK_G を u_A へ返す．

2. u_A は，自身の ID_A と秘密値 x_A より $U_A = h(ID_A, x_A)$ を計算し，以下の式より K_G を得る．秘密値 x_A は DO により配布されるユーザ毎の秘密の値である．

$$\begin{aligned} K_G \cdot Y_0 \cdot \prod_{j=1}^{m} (Y_j)^{{U_A}^j} &= K_G \cdot P^{a_0} \cdot \prod_{j=1}^{m} (P)^{a_j {U_A}^j} \\ &= K_G \cdot P^{a_0 + a_1 U_A + a_2 {U_A}^2 + \cdots + a_m {U_A}^m} \\ &= K_G \cdot P^{f_p(U_A)} \end{aligned}$$

ここで，

$$\begin{aligned} a_0 + a_1 U_A + a_2 {U_A}^2 + \cdots + a_m {U_A}^m &= \sum_{i=0}^{m} a_i {U_A}^j \\ &= f_p(U_A) \\ &= \prod_{j=1}^{m} (U_A - U_j) \\ &= 0 \end{aligned}$$

であるから，

$$K_G \cdot P^0 = K_G$$

4.3.5 データ保存

図 4.5 で示すように，データ所有者 DO とクラウドストレージ CS によって実行される．この操作では，データ所有者 DO がデータファイル D をクラウドストレージ CS に保存する際，一緒に検証データも t 個作成し，登録する．i 個目の検証データを作成する手順を以下に示す．

1. ランダムな置換鍵 $s_0 \in Z_q$ を選ぶ．
2. 置換鍵 s_0 と置換関数 $g_{Key}(\cdot)$ を用いてランダムなインデックス番号 I_j を求める．これにより，d 分割したデータファイル D からランダムに r 個のブロックをサンプリング

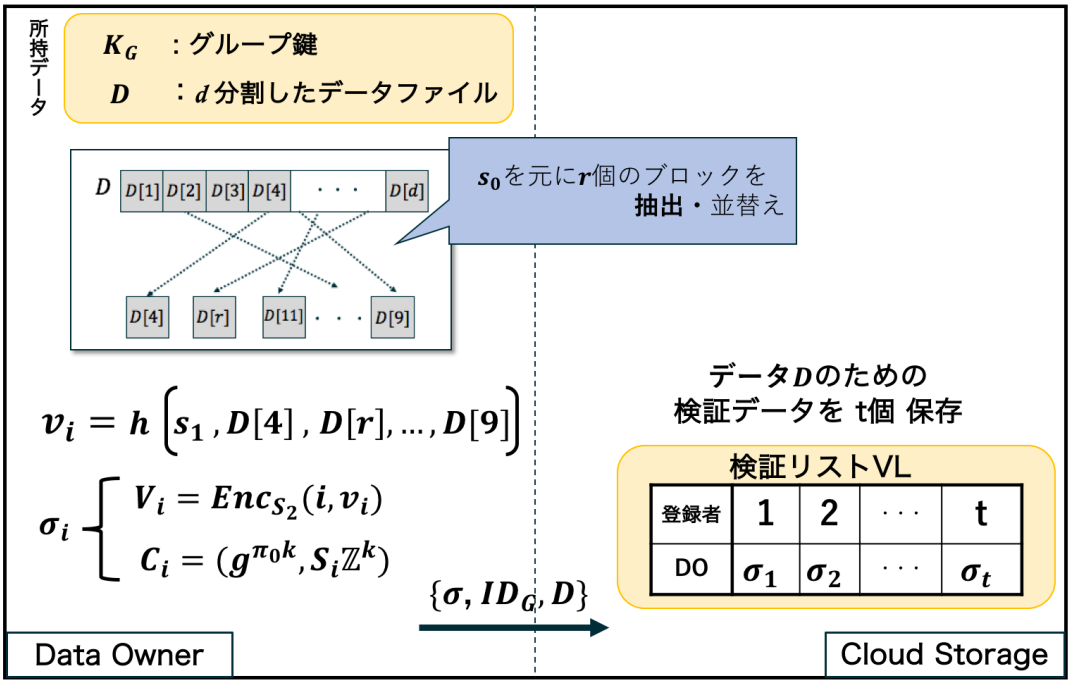

図 4.5　データ保存 (i 回目)

し，ランダムに並びかえることができる．

$$\{I_j \in [1, ..., d] | 1 \leq j \leq r\} \quad g_{s_0}(j) = I_j$$

3. ランダムなノンス $s_1 \in Z_q$ を選び，r 個のランダムブロックと共にハッシュ値を計算する

$$v_i = h(s_1, D[4], D[r], \cdots, D[9])$$

4. ランダムな暗号鍵 $s_2 \in Z_q$ を選び，v_i を暗号鍵 s_2 で暗号化する．暗号化したデータは V_i とする．

$$V_i = Enc_{s_2}(i, v_i)$$

5. 乱数 $k \in Z_q$ を選び，$S_i = (s_0, s_1, s_2)$ を DO の公開鍵 g^{π_0} を用いて暗号化し，第二レベル暗号文 ${C_i}^2 = (g^{\pi_0 k}, S_i \mathbb{Z}^k)$ を求める．
6. 1-5 を t 回繰り返す．

表 4.5 検証リスト VL

登録者	1	2	⋯	t
DO	σ_1	σ_2	⋯	σ_t

7. σ とデータファイル D を検証できるグループのグループ ID ID_G を D と共に CS へ送る．

$$\begin{aligned}\sigma &= \{\sigma_1, \sigma_2, \cdots, \sigma_t\} \\ &= \{(V_1, C_2), (V_2, C_2), \cdots, (V_t, C_t)\}\end{aligned}$$

8. CS は受け取った σ と ID_G を表 4.5 で示す，ID_G のデータファイル D のためのデータ D の検証リスト VL へ登録する．

4.3.6 検証

図 4.6 で示すように，検証したいユーザ u_A とクラウドストレージ CS によって実行される．この操作では，まずユーザ u_A とクラウドストレージ CS が相互認証を確立し，その後検証フェーズに入る．ユーザ u_A がクラウドストレージ CS のデータを検証する手順を以下に示す．

1. u_A は CS と認証を確立させ，必要であればグループ鍵配布フェーズを実行する．
2. u_A はリクエスト $\{i, ID_G\}$ を CS へ送る．
3. CS は表 4.5 を参照し，$\sigma_i = \{V_i, {C_i}^2\}$ を選ぶ．${C_i}^2 = (g^{\pi_0 k}, S_i\mathbb{Z}^k)$ と PRE 鍵 $rk_{DO\to G} = g^{K_G/\pi_0}$ を用いて，以下の式より ${C_i}^1 = (\mathbb{Z}^{K_G k}, S_i\mathbb{Z}^k)$ を計算する．

$$\begin{aligned}{C_i}^1 &= (e(rk_{DO\to K_G}, g^{\pi_0 k}), S_i\mathbb{Z}^k) \\ &= (e(g,g)^{K_G k}, S_i\mathbb{Z}^k) \\ &= (\mathbb{Z}^{K_G k}, S_i\mathbb{Z}^k)\end{aligned}$$

4. CS は $\{V_i, {C_i}^1\}$ を u_A へ返す．

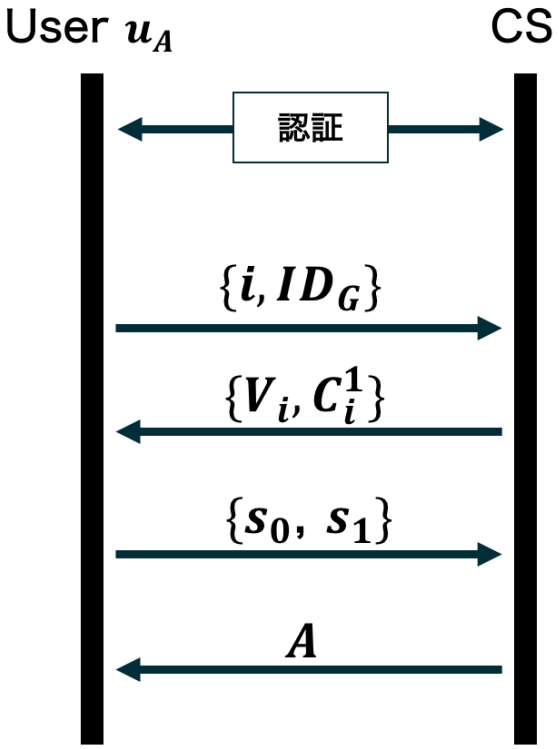

図 4.6 検証 (i 回目)

5. u_A は $C_i{}^1$ から，自身の持つグループ鍵 K_G で以下の式より $S_i = (s_0, s_1, s_2)$ を取り出す.

$$\begin{aligned} S_i &= S_i \mathbb{Z}^k / \mathbb{Z}^{\frac{K_G{}^k}{K_G}} \\ &= S_i \mathbb{Z}^k / \mathbb{Z}^k \end{aligned}$$

6. u_A は取り出した S_i から $\{s_0, s_1\}$ を CS へ送る.
7. CS は s_0 と置換関数 $g_{Key}(\cdot)$ を用いてインデックス番号 I_j を求め，ランダムな r 個のブロックをサンプリングする.

$$\{I_j \in [1, ..., d] | 1 \leq j \leq r\} \quad g_{s_0}(j) = I_j$$

ノンス $s_1 \in Z_q$ と r 個のランダムブロックからハッシュ値 A を計算し，u_A へ返す.

$$A = h(s_1, D[4], D[r], \cdots, D[9])$$

8. u_A は $V_i = Enc_{s_2}(i, v_i)$ を復号し，CS から送られた A のと v_i を比較し，等しければ

検証に成功する．

4.3.7 グループユーザの失効

このフェーズは 4.2.4 節のグループユーザの失効と同じである．グループ G からユーザ u_r が退出するとき，DO がグループ内のユーザ u_r を失効させるために実行する．DO が新しいグループ鍵 K'_G を作り，グループ鍵を CS に再登録するまでの以下に手順を示す．

1. DO は $U_j = h(ID_j, x_j)$， グループメンバー数 $m-1$ とし，u_r が退出した新しいグループ G のための多項式関数 $f'_p(x) = \prod_{j=1}^{m-1}(x - U_j)$ を構成する．

$$\begin{aligned} f'_p(x) &= \prod_{j\neq r\ j=1}^{m-1} (x - U_j) \\ &= \sum_{i=0}^{m-1} a_i x^j \quad (mod \quad q) \end{aligned}$$

   加法巡回群の生成元 $P \in \mathbb{G}_{add}$ と多項式関数 f'_p より，$\{Y_0, ..., Y_{m-1}\} = \{P^{a_0}, ..., P^{a_{m-1}}\}$ を求める．
2. DO は新しいグループ鍵 K'_G を選び，新しい鍵関数を $EK'_G = \{K'_G \cdot Y_0, Y_1 \cdots, Y_{m-1}\}$ とする．また，DO は秘密鍵 $sk_{DO} = \pi_0$ を用いて，PRE 鍵 $rk_{DO\to G'} = g^{K_G{}'/\pi_0}$ を新しく生成する．
3. 生成した EK'_G と $rk'_{DO\to G}$ を ID_G と表 4.2 の CS 用ユーザリスト UL から u_r を取り消したものを CS へ送る．
4. CS は DO から届いた UL を保存し，EK'_G と $rk'_{DO\to G}$ を表 4.3 で示すグループリストへ GL(グループ ID, 鍵関数, PRE 鍵) = $(ID_G, EK'_G, rk_{DO\to G'})$ として更新する．

第 5 章

評価

5.1 グループ鍵の安全性

5.1.1 CS へのグループ鍵の漏えい

グループ鍵が CS に漏えいする可能性のあるものとして，CS 上で保存している鍵関数，PRE 鍵，検証鍵が暗号化された S の暗号文 C^1, C^2 が考えられる．これらの情報から CS がグループ鍵を取得できないことを示す.

鍵関数

提案方式は鍵関数 $EK = K_G \cdot Y_0, \cdots, Y_m$ を保存している CS が K_G を知ることはできないことを示す.

提案方式では DO が $EK = K_G \cdot Y_0, \cdots, Y_m$ を CS へ保存し，DO に認証されたグループユーザ u_i は自身の ID と秘密値 x のハッシュ値 $U_A = h(ID_A, x_A)$ を入力することでが K_G を得ることができる.

$$
\begin{aligned}
K_G \cdot Y_0 \cdot \prod_{j=1}^{m} (Y_j)^{{U_A}^j} &= K_G \cdot Y_0 \cdot \prod_{j=1}^{m} (P)^{a_j {U_A}^j} \\
&= K_G \cdot P^{a_0 + a_1 U_A + a_2 {U_A}^2 + \cdots + a_m {U_A}^m} \\
&= K_G \cdot P^{f_p(U_A)}
\end{aligned}
\tag{5.1}
$$

上記の式 5.1 より CS が K_G を得るためには,$K_G \cdot Y_0$ から K_G を計算する必要があるが,これはに BDHP 仮定に矛盾している．よって，CS はグループ鍵を入手することができない.

PRE 鍵

PRE 鍵からグループ鍵を取得するためには，式 5.2 の通り，g と g^{π_0}，g^{K_G/π_0} を与えられると，g^{K_G} を求めることができる．そして，g と g^{K_G} から K_G を求める必要がある．しかし，これは DLP 仮定に矛盾している．よって，PRE 鍵から CS はグループ鍵を入手することができない.

$$rk_{DO \to G} = g^{K_G/\pi_0} \tag{5.2}$$

第一レベル暗号文

CS は検証鍵 S を暗号化した第二レベル暗号文 C^2 から PRE 鍵を用いて再暗号化処理することで，第一レベル暗号文 C^1 を取得できる．このグループ鍵を取得するためには，式 5.3 の通り，$\mathbb{Z} \in \mathbb{G}_2$ より，$\mathbb{Z}$ と $\mathbb{Z}^{K_G k}$ をあたえられた時，$\mathbb{Z}^{K_G}$ を求める必要がある．しかし，これは DLP 仮定に矛盾している．よって第一レベル暗号文から CS はグループ鍵を入手することができない.

$$\begin{aligned} C^2 &= (g^{\pi_0 k}, S\mathbb{Z}^k) \\ C^1 &= (\mathbb{Z}^{K_G k}, S\mathbb{Z}^k) \end{aligned} \tag{5.3}$$

5.1.2　失効ユーザへのグループ鍵の漏えい

提案方式では，グループ鍵管理のために多項式関数 f_p とプロキシ再暗号化の PRE 鍵を再生成する．提案方式では動的なグループを想定しているため，ユーザがグループから失効する場合のグループ鍵の更新と配布を考慮する必要がある．グループからユーザ u_r が退出する場合を考える．このとき，グループユーザの失効が実行される．この時新しく生成され

た多項式関数を f'_p，新しいグループ鍵を K'_G とする．

$$f'_p(x) = \prod_{j \neq r}^{m-1} \prod_{j=1} (x - U_j)$$
$$= \sum_{i=0}^{m-1} a_i x^j \quad (mod \quad q)$$

加法巡回群の生成元 $P \in \mathbb{G}_{add}$ と多項式関数 f'_p より，$\{Y_0, ..., Ym-1\} = \{P^{a_0}, ..., P^{a_{m-1}}\}$ として EK'_G を求める．

$$EK'_G = \{K'_G \cdot Y_0, Y_1, \cdots, Y_{m-1}\}$$

失効されたユーザ u_r は EK'_G が与えられた時，$U_r = h(ID_r, x_r)$ とすると．

$$K'_G \cdot Y_0 \cdot \prod_{j=1}^{m-1} (Y_j)^{{U_r}^j} = K'_G \cdot P^{a_0} \cdot \prod_{j=1}^{m-1} (P)^{a_j {U_r}^j}$$
$$= K'_G \cdot P^{a_0 + a_1 U_r + a_2 {U_r}^2 + \cdots + a_{m-1} {U_r}^{m-1}}$$
$$= K'_G \cdot P^{f'_p(U_r)}$$

しかし，

$$f'_p(U_r) = \prod_{j \neq r}^{m} \prod_{j=1} (U_r - U_j) \neq 0$$

より，失効されたユーザ u_r は $K'_G \cdot P^{f_p(U_r)} \neq K'_G$ であるためグループ鍵を得ることができない．

5.2 CS と失効ユーザの結託

5.2.1 結託による脅威

S-PDP の脅威としては CS が事前に検証データを作成できることが挙げられる．CS が検証データを作成するためには検証鍵を取得する必要があるが，検証鍵はグループ鍵を所持している者しか取得できない．また，5.1.1 節より，CS がグループ鍵を取得することができないことを示したため，CS は不正に検証データを事前計算することができない．

そこで，失効ユーザと結託した場合を考える．5.1.1 節と 5.1.2 節で示した通り，結託した場合でも CS も失効ユーザも BDHP 仮定に矛盾するためグループ鍵を入手することはできない．

あるグループ G の新しいグループ鍵が K_{G2}，古いグループ鍵が K_{G1} の時，失効ユーザ u_r が所持している古いグループ鍵 K_{G1} とそのグループ鍵に対応する PRE 鍵 $rk_{DO\to G1}$ を CS が不正に所持している場合を考える．この時，式 5.4 で示す通り，CS は検証鍵 S の第二レベル暗号文 $C^2 = (g^{\pi_0 k}, S\mathbb{Z}^k)$ を古い PRE 鍵で再暗号化し，得られた第一レベル暗号文 $C^1 = (\mathbb{Z}^{K_{G1}k}, S\mathbb{Z}^k)$ を失効ユーザへ送ることで失効ユーザは検証鍵 S を取得できる．

$$\begin{aligned} C^1 &= (e(rk_{DO\to G1}, g^{\pi_0 k}), S\mathbb{Z}^k) \\ &= (e(g,g)^{K_{G1}k}, S\mathbb{Z}^k) \\ &= (\mathbb{Z}^{K_{G1}k}, S\mathbb{Z}^k) \end{aligned} \tag{5.4}$$

そのため，提案方式では失効ユーザと CS の結託がないことを前提とする必要がある．

5.2.2 安全な運用のための要件

5.2.1 節で示したとおり，提案方式では CS と失効ユーザ u_r が結託した場合，検証データが漏えいする脅威が想定される．そこで，提案方式を安全に運用するための要件を定め，要件を満たした運用をすることでこの脅威を防ぐ．

情報セキュリティの対策は以下の 2 つに大きく 2 分類される [19, 20]．

物理的セキュリティ

建物や設備などを対象とした物理的なセキュリティ対策のことである．具体的には認証システムを導入した入退室管理，建物の耐震化，防火設備，電源設備などの物理的な保護のことを指す．建物への不正な侵入を防ぐことや，災害や障害などの被害を最小限に留めることで重要な情報資産を保護する．

論理的セキュリティ

物理的セキュリティ以外のセキュリティ対策を指す．論理的セキュリティはシステム的対策，管理的対策，人的対策に分けられる．

1. 【システム的対策】アクセス制御，認証，暗号化などの情報システムやネットワークなど技術的な対策
2. 【管理的対策】セキュリティポリシの策定，運用，監査，ソフトウェアライセンス管理など組織や情報システムの運用・管理における対策
3. 【人的対策】契約におけるセキュリティ対策，教育，セキュリティ事件・事故，ポリシ違反時の罰則などの対策

提案方式の結託における脅威は，図 5.1 に示すように失効ユーザが所持しているグループ鍵 K_{G1} とクラウドストレージが所持している $rk_{DO \to G1}$ が紐づくことで起きる．そこで，結託を防ぐためには 2 つの方法が考えられる．

① 失効ユーザが所持しているグループ鍵 K_{G1} とクラウドストレージが所持している $rk_{DO \to G1}$ が紐づかないようにする．

② 使用するクライアント端末を業務用 PC に限定し，グループ鍵情報へのアクセスをシステム管理者（DO）などに限定させる．

①について考える．グループ鍵と PRE 鍵が紐づく理由として，表 4.2 と 4.3 のユーザリストとグループリストをクラウドストレージが所持しているためである．そこで，図 5.2 に示すようにユーザとクラウドストレージの間に認証用のプロキシサーバーを設置する．プロキシサーバーには表 4.2 のユーザリストと鍵関数を所持する．これにより，クラウドストレー

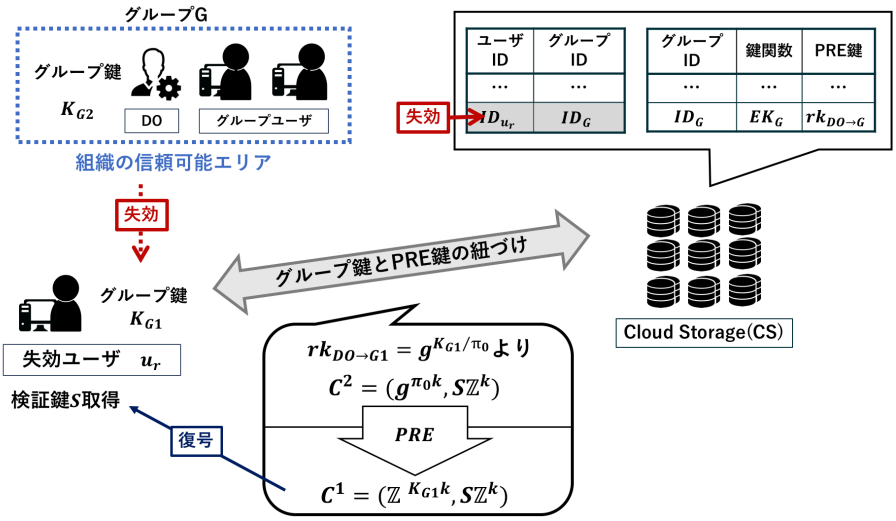

図 5.1　結託における脅威

ジが知り得る情報がグループ ID と PRE 鍵のみである．エンドユーザであるグループユーザらはプロキシサーバーで認証し，プロキシサーバーがユーザらの代わりにグループ ID で認証する．これにより，グループ鍵と PRE 鍵が紐づくことを防ぐことができる．またプロキシサーバーは 5.1.1 節で CS がグループ鍵を得られないことと同様に，グループ鍵を知ることはできない．

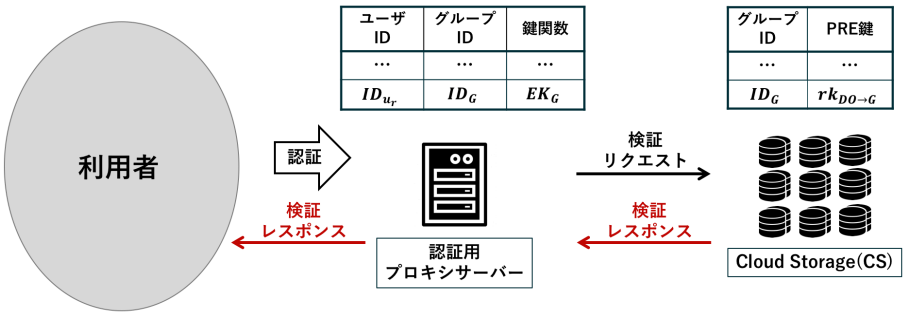

図 5.2　安全な運用のための構成

また，失効ユーザが自ら古いグループ鍵をクラウドストレージ管理者に教えた場合はこの不正を防ぐことはできない．そこで，②のように物理的セキュリティとシステム的対策を施す．また，各クライアント端末と認証用プロキシサーバー，クラウドストレージのアクセス

ログを取り，定期的な監査を行う管理的対策を施すことで不正を抑止する．

5.3　効率的なグループ鍵更新

提案方式では信頼できるエンティティを DO と DO に認められたユーザのみであり，この限られたエンティティだけで安全に効率よくグループ鍵の更新を実現した．通常グループ鍵の更新をすると，新しいグループ鍵を全ユーザへ配布し，古いグループ鍵を使用して暗号化された暗号文は新しいグループ鍵を用いて暗号化し直す必要がある．

しかし，提案方式では CS に秘匿するために暗号化された検証鍵は DO の秘密鍵 $sk_{DO} = \pi_0$ で暗号化されており，グループ鍵は PRE 鍵に使用されているだけである．そのため，グループ鍵の更新がある場合，古いグループ鍵で暗号化された暗号文を新しいグループ鍵で暗号化し直す必要がなく，新しい PRE 鍵の更新のみで実現できる．

また，グループ鍵の更新があったグループに所属するユーザへの鍵配布も各ユーザが鍵関数から各自で取得できるため，DO がユーザへグループ鍵を配布する必要がない．

CS はグループ鍵の更新がある場合，DO から送られてきた鍵関数と PRE 鍵を再登録するだけで良い．

よって，提案方式では限られた信頼できるエンティティだけで安全に効率よくグループ鍵の更新と配布をすることができる．

5.4　検証データのリプレイ攻撃

CS は検証者がリクエストした検証データを作るためには，検証者がリクエストした i と検証鍵 $= \{s_0, s_1, s_2\}$ が必要になる．i に該当する正しい S を得るためにはグループ鍵 K_G が必要であるが，CS がグループ鍵を取得することが困難なことはこれまでに示した．よって，i 番目の正しい検証データ V_i を作成のためには，CS は検証者のリクエストを待つ必要がある．

しかし，CS への検証リクエストで送られる i が既に他のユーザが検証している場合，CS

は既に使用した検証データ V_i を返すことができ，リプレイ攻撃が可能になる．よって，グループユーザが重複した検証リクエスト i を送らないように，DO が予め各ユーザ毎に検証する番号を指定することで提案方式のこの問題を回避できる．

5.5　検知率

G. Ateniese ら [4] が提案した PDP（Provable Data Possession scheme）方式ではサーバ上のデータ 1% の破損をランダムな 460 ブロックをサーバへ要求することで 99% の確率で検知できることを証明している．サーバが n ブロックに分割されたファイル D から t ブロック削除すると仮定する．クライアントがサーバに所持証明のために要求するブロック数を c とする．サーバによって削除されたブロックと一致するクライアントによって選択されたブロックの数を示す離散確率変数を X とする．クライアントにより選択されたランダムなブロックの内少なくとも 1 つがサーバによって削除されたブロックの一つと一致する確率 P_X を以下の式より求める．

$$\begin{aligned} P_X &= P\{X \geq 1\} \\ &= 1 - P\{X = 0\} \\ &= 1 - \frac{n-t}{n} \cdot \frac{n-1-t}{n-1} \cdot \frac{n-2-t}{n-2} \cdot \cdots \cdot \frac{n-c+1-t}{n-c+1} \end{aligned}$$

$\frac{n-i-t}{n-i} \geq \frac{n-i-1-t}{n-i-1}$ であるから，

$$1 - (\frac{n-t}{n})^c \leq P_X \leq 1 - (\frac{n-c+1-t}{n-c+1})^c \tag{5.5}$$

以上より，P_X はサーバがファイル D の内 t ブロック削除した場合にクライアントが c ブロック数サーバに要求することでサーバの不正行為を検出できる確率を示している．PDP ではファイルのブロック総数 n とは無関係に一定量のブロック c を要求することでサーバの不正行為を一定確率で検知できることを証明している. 例として式 5.5 より，$t = 1\%, c = 460$ としたとき，代入するとブロック総数 n の値にかかわらず $P_X = 0.99\cdots$ となり CS 上のデータの破損検知率が 99% となる．

5.6 通信・ストレージコスト

5.6.1 検証データのストレージコスト

表 5.1 検証データのストレージコスト比較

	ストレージコスト
S-PDP	$t\|V\|$
提案方式	$t(\|V\| + \|\mathbb{G}_1\| + \|S\|\times\|\mathbb{G}_2\|)$

表 5.2 1GB あたりの料金比較

	USD	円
AWS Glacier	0.004	0.432
Google Nearline	0.010	1.088
Google Clodline	0.007	0.756

S-PDP と提案方式のストレージコストの比較を表 5.1 に示す．$|V|$ は 1 回の検証データサイズを示しており，S-PDP と提案方式は共に検証番号とハッシュ値を暗号化した検証データサイズである．

S-PDP で CS に保存されるの 1 回の検証データは $v_i{}' = Enc_K(i, v_i)$ である．よって，S-PDP の一つあたりの検証データサイズはおよそ $(i + v_i)bit$ になる．i は検証回数を表現できる bit 長であるため，$i = 32 \times\ 365 = 11680 < 2^{11}$（32 年間毎日検証できる回数），$v_i$ を $32(2^5)Byte$ とする．このときの一つあたりの検証データサイズ $|V|$ はあたりおよそ $2KB$ である．つまり S-PDP では，検証データの保存のために $2KB \times\ t$ 分のストレージが必要になる．

提案方式では CS に保存される 1 回の検証データ σ_i は $V_i = Enc_{s_2}(i, v_i)$ と $C_i = (g^{\pi_0 k}, S_i\mathbb{Z}^k)$ から構成され，検証データ V_i のサイズは S-PDP と同じサイズである．$|\mathbb{G}_1| + |S| \times |\mathbb{G}_2|$ では，検証用の鍵データが含まれる C_i のサイズを示している．提案方式では

検証データに加えて検証用の鍵データ C_i も保存する必要がある．そのため，提案方式では S-PDP よりもストレージコストがかかる．S は 3 つの乱数が格納される．S に格納される乱数が一つあたり $128bit$，巡回群のサイズを $160bit$ と考えると，C_i のデータサイズ $|\mathbb{G}_1| + |S| \times |\mathbb{G}_2|$ は一つあたりおよそ $7.7KB$ である．つまり提案方式では，検証データの保存のために $(2KB + 7.7KB) \times t$ 分のストレージが必要になる．以上より，提案方式では S-PDP よりもストレージコストがかかる．これは検証用の鍵データも保存する必要があるためである．また，今回の提案方式ではコールドデータ向けの安価なストレージの利用を想定している．表 5.2 に，既存の安価なストレージサービス [21, 22] の $1GB$ あたりの単価を示す（2 月 1 日 15:00 時点での為替レート 1USD = 108.8 円）．提案方式では S-PDP よりも $7.7KB \times t$ 分のストレージが必要になるが，この増分を料金に換算すると，表 5.2 の最も高い Nearline の場合 1.088×10^{-6}円$/1KB$ である．よって，提案方式のストレージオーバーヘッドによる料金の増分はおよそ $7.7 \times 1.088 \times 10^{-6} \simeq 8.377 \times 10^{-6}$ 円であり，検証回数 t が 100,000 回であっても 1 円にも満たない．

5.6.2　通信コスト

S-PDP と提案方式の通信コストの比較を表 5.3 に示す．データ登録の $|D|$ は CS に保存するデータファイル D のサイズを表している．データ登録では，6.2 節 ストレージコストでも述べている通り，提案方式では鍵データも保存する必要があるため，S-PDP よりも通信コストが増える．

検証では，提案方式ではまずユーザから検証リクエストを送る必要があり，その際の通信コストが $|S|$，CS からのレスポンスの通信コストが $|V|$ になる．$|S|$ と $|V|$ は固定サイズであるが，提案方式では S-PDP よりも通信回数が増える．

表 5.3　通信コスト比較

	S-PDP	提案方式
データ登録	$\|D\| + t\|V\|$	$\|D\| + t(\|V\| + \|\mathbb{G}_1\| + \|S\| \times \|\mathbb{G}_2\|)$
検証	$\|V\|$	$\|S\|$ （$User \rightarrow CS$），$\|V\|$ （$CS \rightarrow User$）

5.7　比較評価

5.7.1　既存のグループ鍵管理方式との比較

提案方式では，信頼可能な鍵生成サーバが必要のないグループ鍵管理方式を提案した．そこで，既存の同様の鍵管理方式と比較を表 5.4 に示す．鍵生成を行う者がグループ管理者で

表 5.4　既存研究との比較評価

	Zhu et al. [10]	Song et al. [12]	提案方式
鍵生成者	グループ管理者		
鍵配布者	グループ管理者	クラウドストレージ	
鍵更新時の暗号文作成	必要	-	不要

ある点は同様であるが，Zhu らの方式はグループ管理者が鍵配布を行うため，常にオンラインで待機する必要がある．また，Song らは鍵配布者がクラウドストレージではあるが，鍵配布方式の提案であるため，暗号文の管理はない．Zhu らの方式は鍵更新時に暗号文へのアクセス制御を実現するために暗号文を作り直す必要があるが，提案方式では不要である．よって，グループ管理者の負担が少ないという点では提案方式が良いと言える．しかし，長期間の運用を考え，暗号鍵の危殆化を考慮すると，暗号文の作り直しを行う Zhu らの方式の方が安全である．よって，提案方式では検証データの保存期間を考慮した適切な鍵長を選択する必要がある．

5.7.2　セキュリティ強度

5.7.1 節で述べたように提案方式ではグループ管理者の負担の削減を実現する場合，検証データの保存期間を考慮した適切な鍵長を選択する必要がある．提案方式では長期的な保存が必要なデータへの適用を想定しているため，安全に運用できるアルゴリズムを選択する必要がある．

法律により保存義務が定められている書類は永久保存・30 年・10 年・7 年など，種類により保存期間が違う．総務・経理などの書類を想定する場合 10 年の保存が必要であるため，これを想定して適切なセキュリティ強度のあるものを選択する．また，NIST による暗号安全性評価 [23] では 2030 年以降は最低 256bit セキュリティが必要とされているため，提案方式でも 256bit セキュリティの達成を基準とする．提案方式でセキュリティ強度を考慮する必要のある要素技術と 256bit セキュリティを達成するための条件は表 5.5 の通りである．共通鍵暗号方式では AES などの暗号アルゴリズムを用いる際に，鍵長が 256bit 以上であることを示している．プロキシ再暗号化では 3357bit のペアリング暗号を用いることで 2030 年は安全であると見積もられている．また，グループ鍵は鍵関数としてクラウドストレージ上で管理される．鍵関数では代数曲線上の点による巡回群を生成し，この時の位数を 512bit 以上を設定することで 256bit 相当の離散対数問題と等価な安全性を有することができる [24]．

表 5.5　256bit セキュリティ達成条件

要素技術	条件
共通鍵暗号方式	鍵長 256bit
プロキシ再暗号化	ペアリング 3357bit
グループ鍵	鍵長 512bit

5.8　グループで S-PDP を実現するための要件と評価

2 章で定義した S-PDP を実現するための要件について評価する．

信頼エンティティの最小限化

提案方式では S-PDP と同様に，CS を信頼できないエンティティとしており，グループ鍵管理においても信頼できるエンティティは DO と DO に認められたグループユーザだけとする．また，CS の不正によるグループ鍵の取得や検証データの事前計算ができないことを示した．

動的グループでの効率的な鍵更新

提案方式では効率的に鍵更新を実現した．グループ鍵を更新する際，DO は新しいグループ鍵の鍵関数，PRE 鍵を生成し，CS へ登録するだけでグループ鍵の更新とグループユーザへの鍵配布ができる．グループ鍵の更新に伴う古いグループ鍵を暗号鍵として生成された暗号文の更新は必要ない．また，グループユーザは各自で鍵関数から新しいグループ鍵を取得でき，CS は DO から送られてくる新しい鍵関数と PRE 鍵へ更新するだけでグループ鍵の更新が完了する．よって，提案方式では限られた信頼できるエンティティだけで安全に効率よくグループ鍵の更新と配布を実現できた．

失効ユーザの不正

S-PDP は検証できる回数が予め決められた回数である．失効されたユーザは信頼できるエンティティではないため，失効ユーザが検証できると不正に検証データを消費される可能性があったが，失効ユーザはグループ鍵を得ることができないことを示したため，不正に検証はできない．しかし，CS が PRE 鍵の更新をせず，失効ユーザの古いグループ鍵に対応する PRE 鍵を所持している場合，失効ユーザと結託することで失効ユーザは検証を実施できるため，提案方式では CS と失効ユーザの結託がないことを前提とする．そのため，結託が起きない要件を示した．この要件を満たす対策を施すことで提案方式は安全に運用ができる．

5.8　グループで S-PDP を実現するための要件と評価

以上より，提案方式では S-PDP をグループで安全に運用するための要件を示し，その要件を満たす場合に安全な S-PDP をグループで実現できることを示した．

第 6 章

まとめ

クラウドストレージは安価なストレージサービスを提供するが，企業や個人がデータをクラウドストレージに保存する場合，クライアント自身はデータを完全に管理することができなくなる．そのため，クラウドストレージに保存されるデータはクラウドストレージプロバイダの情報セキュリティに依存することになる．悪意のある内部攻撃者や信頼性の低いクラウドストレージなどを想定し，外部に保存されたデータの所持証明を動的グループで実現する方式を提案した．

提案方式は，コールドデータのような大量の静的データをクラウドストレージが所持していることを効率的に検証可能な方式である．データ所有者は信頼できるグループユーザに検証権限を付与するためにグループ鍵を配布する．グループ内でグループ鍵を使用する際，ユーザの退出を考慮し，グループ鍵の更新を考える必要がある．そのため，提案方式では動的なグループでのデータ検証を想定した．

クラウドストレージや失効されたユーザがグループ鍵を不正に取得できないこと，クラウドストレージが検証データを事前計算できないことを示した．しかし，提案方式ではクラウドラウドストレージと失効ユーザの結託耐性がなく，結託した場合失効ユーザは検証鍵を取得できるため，提案方式ではクラウドストレージと失効ユーザの結託がない前提とする．また，動的なグループで所持証明を実現するために，検証データと共に検証鍵データもクラウドストレージ上で保存するため，ストレージと通信のオーバヘッドを示した．今後の課題として，システム全体を実装することが挙げられる．システム全体を評価し，CS の検証リクエスト数への負荷を調査する必要があると考えられる．

謝辞

本研究と論文の作成にあたり，御助言をいただきました高知工科大学情報学群 清水明宏教授に感謝いたします．また，本研究の副査を担当していただいた高知工科大学情報学群敷田幹文教授，福本昌弘教授に深く御礼申し上げます．敷田先生には研究へのアドバイスだけでなく，これから社会へ出ていく私のために，親身にお話をしてくださり，大変お世話になりました．福本先生にはセミナーなどを通じて研究への姿勢を教えていただき，大変勉強になりました．

同期の清水研究室 修士 2 年生の合田氏，藤田氏，安光氏には 4 年間という研究室生活を通して研究の相談だけでなく研究室外のこともたくさん支えていただきました．この 4 人で共に研究室生活を過ごせて本当によかったです．お世話になりました．これからも仲良くしてください．また，高知工科大学清水研究室 高橋氏をはじめとする研究室生のみなさんにはより良い研究室環境・雰囲気を作っていただき感謝しております．みなさんと研究室で過ごす中での他愛のない会話に元気がでました．すてきな研究室を作ってくれてありがとうございます．とても楽しかったです．

最後に，大学・大学院生活の計 6 年間を支えていただいた両親に心より感謝いたします．修士課程まで進学させていただいた分の恩返しをこれからしていきたいと思います．お世話になりました．

参考文献

[1] N. K. Yang and X. Jia, “Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities,” World Wide Web, Volume 15, Issue 4, pp.409–428, 2012.

[2] 日本経済新聞「コールドデータとは」, 2015 年 7 月 28 日付朝刊, https://www.nikkei.com/article/DGKKASDZ27HZ2_X20C15A7TI1000/, (2019/1/4 アクセス).

[3] D. Reinsel, J. Gantz, J. Rydning, ”Data Age 2025”, IDC White Paper, November 2018.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS ’07), pp.598-610, 2007.

[5] B. Wang, B. Li, and H. Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,” Proc. IEEE 5th Int’l Conf. Cloud Computing (CLOUD 12), vol. 2, no.1, pp.295–302, 2012.

[6] G. Ateniese , RD. Pietro , LV. Mancini, T. Sudik .”Scalable and efficient Provable data possession”. In: Proceedings of the 4th international Conference on security and privacy in communication Networks ACM, pp.1-10, 2008.

[7] Y. Chen, L. Li, Z. Chen, ”An Approach to Verifying Data Integrity for Cloud Storage”, 13th International Conference on Computational Intelligence and Security (CIS), pp.582–585, Dec 2017.

[8] XU Guang-wei, SUN Zhi-feng, CHEN Chun-lin1, et al. , ”Verification of Dynamic Data Integrity Using Counting Bloom Filter in Cloud Storage[J],” Journal of Chinese Computer Systems, 35(10): pp.2778-2283, 2014.

[9] B. Wang, L. Baochun, H. Li, ”Panda: Public auditing for shared data with efficient

user revocation in the cloud," IEEE Transactions on services computing, 8(1), pp.92-106, 2015.

[10] Z. Zhu and R. Jiang, "A secure anti-collusion data sharing scheme for dynamic groups in the cloud," IEEE Transactions on parallel and distributed systems, vol. 27, pp.40–50, 2016.

[11] X. Zou, Y.S. Dai, E. Bertino, "A practical and flexible key management mechanism for trusted collaborative computing", INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, 2008.

[12] W. Song, H. Zou, H. Liu, J. Chen, "A practical group key management algorithm for cloud data sharing with dynamic group," In: China Communications, 13(6), IEEE Journals and Magazines, 2016.

[13] A. Samit, B. Waters, "Fuzzy Identitiy-Based Encryption," Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp.457–473 2005.

[14] L. Cheung, C. Newport, "Provably Secure Ciphertext Policy ABE", Proc. 14th ACM Conference on Computer and Communications Security, pp.456–465 2007.

[15] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proc. 2007 IEEE Symposium on Security and Privacy, pp.321–334, 2007.

[16] G. Zhang, L. Liu, and Y. Liu, "An attribute-based encryption scheme secure against malicious KGC," in Proc. TRUSTCOM, Jun. pp.1376–1380, 2012.

[17] G. Ateniese, K. Fu, M. Green, S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," ACM Transaction on Information and System Security, vol. 9, no. 1, pp.1-30, January, 2006.

[18] M. Blaze, G. Bleumer, M. Strauss, "Divertible Protocols and Atom Proxy Cryptography," Proceedings of International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT), pp.127-144, May, 1998.

[19] 独立行政法人 情報処理推進機構, ”守るべき情報資産・情報リスクの考え方,” https://www.ipa.go.jp/files/000013297.pdf, (2019/2/1 アクセス).

[20] 上原孝之, ”情報処理教科書 情報セキュリティスペシャリスト”, 翔泳社, 2014.

[21] Amazon S3 Glacier, Amazon Web Services, https://aws.amazon.com/jp/glacier/, (2019/2/1 アクセス).

[22] Google ストレージプロダクト「Cloud Storagen の料金」, Google Cloud, https://cloud.google.com/storage/pricing?hl=ja, (2019/2/1 アクセス).

[23] National Institute of Standards and Technology, ”Recommendation for Key Management Part 1: General”, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf, (2019/2/18 アクセス)

[24] 森川郁也, 下山武司, ”暗号等価安全性”, 電子情報通信学会誌 The journal of the Institute of Electronics, Information and Communication Engineers 94.11, pp. 987-992, 2011.