分散台帳技術を用いたスナップショット配信による分散ファイルシステムの実装

芳賀 郁弥[1]　新城 靖[1]　周 毅[1]　宝田 一希[1]　中村 公洋[1]　佐藤 聡[1]　中井 央[1]

1. はじめに

現在 Google Drive 等のストレージサービスが広く使われている．そのようなサービスの提供者はデータの内容を監視することができ，ユーザのアカウントを停止することが可能である．これを避けるために，非中央集権的なファイル共有の仕組みが注目を集めている．例えば，P2P (peer-to-peer) ネットワークに基づいた IPFS (Interplanetary File System)[1] が開発されている．IPFS はローカルファイルシステムや中央集権的なストレージサービスで提供されているようなディレクトリや ACL (Access Control List) に基づくアクセス制御の機能を持たない．

我々は，分散台帳技術を利用して中央サーバに依存しない分散ファイルシステムを実装している．最初の実装 [2] では，ファイルのデータを暗号化してコストが低い既存の分散ストレージに保存する．そして，ファイルのメタデータ (inode) を分散台帳ネットワーク上で動作するコードであるスマートコントラクトで管理する．この実装ではアクセス制御の設定を含むメタデータ全てを分散台帳に記録しているためプライバシの問題があった．また，階層的なディレクトリに対応していなかった．

この論文では，階層的なディレクトリを持ち，プライバシを保護するような分散台帳技術を用いた分散ファイルシステムについて述べる．本研究ではアクセス制御をスマートコントラクトではなく，ローカルコンピュータ上で暗号により実装する．また，ディレクトリの木構造をローカルファイルシステムが持つスナップショット機能を利用して実装する．ノード間でのスナップショットの転送に本研究室で実装したファイル転送機能 [3] を用いる．分散台帳に書き込まれたデータは誰でもアクセス可能であり，またトランザクションの手数料を支払うために送信者を秘匿することはできない．このファイル転送機能は，暗号化されたメッセージを分散台帳ネットワークにブロードキャストするため，ファイルの受信者は秘匿される．

[1] 筑波大学

2. 分散台帳技術を用いたスナップショット配信による分散ファイルシステム

図 1 に，本分散ファイルシステムの全体構成を示す．本ファイルシステムでは，1 人の所有者が保持する平文ディレクトリを読み出しが許可された複数のユーザのコンピュータに複製する．所有者もユーザもそれぞれ分散台帳のウォレットの公開鍵と秘密鍵およびそのアドレスを持つ．所有者は，ユーザの代理ユーザの ID，ユーザネーム，ユーザの分散台帳上のウォレットのアドレス，ユーザの公開鍵の 4 つのカラムを持つユーザ識別子管理表を管理する．また，所有者は平文ディレクトリ以下でファイルの作成，修正，削除，および読取をすることができる．そして，`setfacl` コマンド等を利用して，代理ユーザ ID を用いてアクセス制御リストを設定する．

本分散ファイルシステムの暗号化機能は所有者のコンピュータにおいて以下の手順で，平文ディレクトリと同一構造を持つ暗号文ディレクトリを作成する．

(1) 各平文ファイルごとにその内容の暗号化のための共通鍵を生成する
(2) (1) の各共通鍵で各平文ファイルの内容を暗号化する
(3) アクセス制御リストを参照して読み出しが許可されたユーザの代理ユーザの ID のリストを取得する
(4) ユーザ識別子管理表を参照して，リスト中の各ユーザ ID に紐づいたユーザの公開鍵を取得し，その公開鍵を用いて (1) の各共通鍵を暗号化する
(5) 平文ディレクトリにおける各平文ファイルと同階層に位置するように，暗号文ディレクトリ以下に各暗号文ファイルを作成し，(2) の暗号文および (4) の結果得られた暗号化された鍵のリストを保存する

上の (4) に示したように，共通鍵を各ユーザの公開鍵で暗号化することで，(2) の暗号化を 1 ファイルにつき 1 回で済ますことができる．

所有者のコンピュータではローカルファイルシステムが持つ機能で，暗号文ディレクトリのスナップショットを取り，単一ファイルへマーシャリングする．本システムでは，ローカルファイルシステムとして Sun MicroSystems 社の

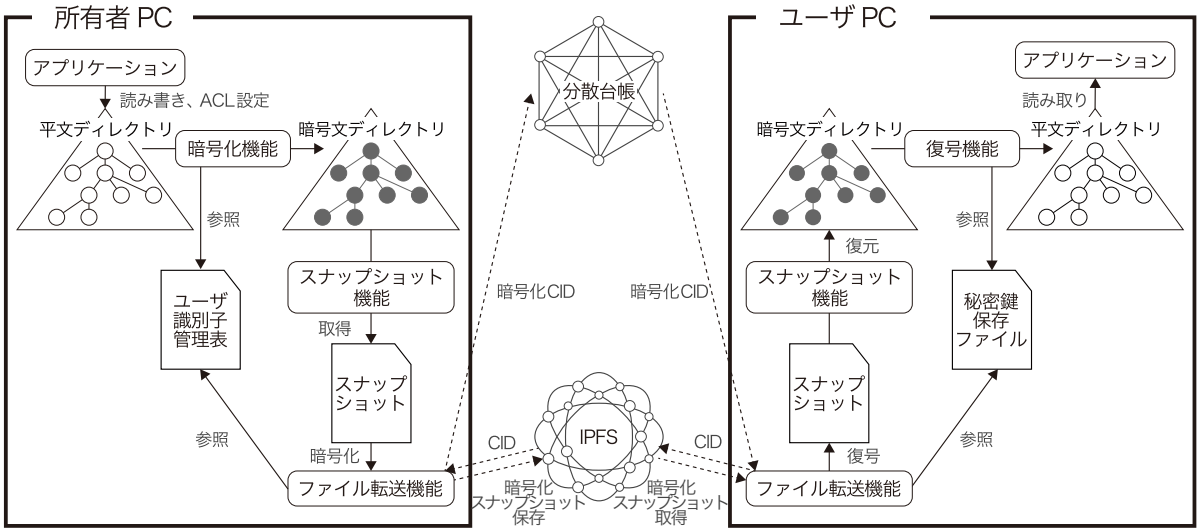

図 1 分散ファイルシステム全体構成

ZFS[4] を利用する．ZFS はスナップショットが高速に取ることができ，かつインクリメンタルにスナップショットを取ることが可能である．

所有者のコンピュータで得られたスナップショットをファイル転送機能 [3] で，ユーザのコンピュータへ転送する．このファイル転送機能は，まずそのスナップショットのデータを暗号化して IPFS に保存する．すると，IPFS はその内容のハッシュ値から生成されるユニークなコンテント識別子 (CID) を返す．次に，その CID をユーザの公開鍵で暗号化し，スマートコントラクトで分散台帳ネットワーク上にイベントとしてブロードキャストする．ユーザはイベントを監視し，ブロードキャストされたデータ，すなわち暗号化された CID を自身が持つ秘密鍵で復号できれば，その CID を受け取る．そして，CID を使って IPFS からスナップショットを取得する．

スナップショットを受け取ったユーザのコンピュータでは所有者と同様の暗号文ディレクトリが復元される．本分散ファイルシステムの復号機能は，上で述べた暗号化機能の逆の手順で，暗号文ディレクトリ以下の全てのファイルを復号することを試みる．具体的には自身の秘密鍵を用いて，暗号文ファイルに含まれる公開鍵によって暗号化された共通鍵のリストの各要素を順に復号する．復号に成功すると，ユーザは暗号文ファイルに含まれる暗号化された平文ファイルの内容を先に復号した共通鍵を用いて復号し，平文ディレクトリ以下の同じ位置に保存する．

3. 関連研究

文献 [5] でも本研究と同様に，IPFS と分散台帳技術を用いた分散ファイルシステムについて述べている．このファイルシステムでは IPFS プロキシと呼ばれる，単一ノードでアクセス制御が実装されている．本研究ではそのような集中的なノードは存在しない．

4. まとめ

本論文では，階層的なディレクトリを持ち，プライバシが保護される分散台帳技術を用いた分散ファイルシステムについて述べた．階層的なディレクトリを複数のユーザと共有するためにローカルファイルシステムが持つスナップショット機能を利用した．また，プライバシを保護するためにアクセス制御をスマートコントラクトではなくローカルコンピュータ上で暗号により実装した．

今後の課題は，ユーザによるファイルの更新を許すことである．そのために，オーバレイファイルシステムを利用することを検討している．

参考文献

[1] Juan Benet: “IPFS - Content Addressed, Versioned, P2P File System (DRAFT3)”, arXiv:1407.3561, 11 pages, 2014.
[2] 中村公洋，新城靖，佐藤聡，中井央．“分散台帳技術を用いた分散ファイルシステムの構想”，情報処理学会第 31 回コンピュータシステム・シンポジウム (ComSys2019) ポスターセッション，2019.
[3] 渡部太揮，新城靖，周毅，宝田一希，中村公洋．“分散台帳技術 Ethereum を用いた監視や検閲に強い信頼できるメールの実装”，情報処理学会グループウェアとネットワークサービス 30 周年記念シンポジウム＆ワークショップ 2022，8 ページ，2022.
[4] J. Bonwick, M. Ahrens, V. Henson, M. Maybee, and M. Shellenbaum. “The Zettabyte File System”, Technical report, Sun Microsystems, 2003.
[5] Hsiao-Shan Huang and Tian-Sheuan Chang and Jhih-Yi Wu. “A Secure File Sharing System Based on IPFS and Blockchain”, Proceedings of the 2nd International Electronics Communication Conference, Pages 96–100, 2020.