6D-01

ブロックチェーン上の構造化データに対する権限分散を考慮した開示制御手法

渡邊 大喜† 大橋 盛徳† 藤村 滋† 中平 篤†

日本電信電話株式会社 NTT サービスエボリューション研究所†

1 はじめに

仮想通貨ビットコインの中核技術であるブロックチェーンの応用が始まっている[1]．ブロックチェーンを利用したシステムでは，自律分散化したノード間において情報を全て公開し，お互いの取引を検証する．しかしながら用途によっては，トランザクションの一部の領域を秘匿化し，開示先を制限して共有したい場合がある．例えば，登録するデータの中に，企業間の機密情報や利用者のパーソナルデータなど，ある関係者にとって必要ではあるが積極的に開示すべきではない情報が含まれる場合に，開示制御機能が必要となる．

このような場合，従来の解決策として，開示制御用のサーバを設け，非開示の情報はブロックチェーン外（off-blockchain）で必要な関係者間のみで共有することが行われている．しかしながら，ブロックチェーンシステムのような分散化したネットワークでは，開示制御サーバの管理が中央集権化し，ブロックチェーンの持つ自律分散性や改ざん耐性といった特徴を損ねる可能性がある．

そこで著者らは，アクセス管理や秘匿化の機能をブロックチェーンシステムの系に分散配置し，ブロックチェーンシステムと連携動作するアーキテクチャにより，サーバで集中管理せずとも開示制御可能な方式を提案する．また本技術方式では，鍵交換の処理など開示制御に関わる処理が全てトランザクション化され，証跡として記録されるため，系全体で処理の証拠性が高まる効果が期待される．

2 スマートコントラクト型ブロックチェーン

仮想通貨の取引だけでなく，チューリング完全なプログラムの実行結果に関しても，ブロックチェーン上で分散的に合意を得る仕組みを「スマートコントラクト」と呼ぶ．代表的なスマートコントラクト型のブロックチェーンとして，Ethereum[2]やHyperledger Fabic[3]が知られている．スマートコントラクト型のブロックチェーンでは，個々のトランザクションは，ブロックチェーン上に登録されたプログラム（コントラクト）の入力として機能し，その出力値はステート DB と呼ばれる Key-Value Store（KVS）に構造化データとして記録される．ブロックチェーンの特徴として，トランザクションが承認されると，全ての分散ノード上でコントラクトが実行され，個々のノードが持つステート DB は同一の値に更新される．すなわちブロックチェーンのネットワークに参加する全てのノードは，登録された全ての値について同一のコピーをそれぞれの端末上に保持することとなる．

例えば，A 社と B 社の間で，ある合意事項に関する契約書がブロックチェーンを介して記録される場合，その契約項目を Key とし，項目値を Value とした構造化データがステート DB に保持される．登録された構造化データは A 社と B 社以外に，ブロックチェーンネットワークを維持するノード（例えば C 社）群にも共有される．この時，契約書の項目に A 社と B 社間のみで共有する機密情報やパーソナルデータが含まれる場合，データの保護およびアクセス制限が必要になる．

3 従来の開示制御の問題

従来のブロックチェーンシステムの開示制御の方法として，例えば Zyskind[4]らの提案に見られるように，暗号化した機密情報のハッシュ値のみをブロックチェーン内に記録し，これを暗号化した機密情報へのポインタとする手法がある．

A 社・B 社間で個別に管理可能な範囲であれば有効だが，機密を共有する関係者が増えたり，機密の項目が細分化されたりした場合，非開示な情報の管理が煩雑化する．そのため，開示制御サーバの働きや権限が大きくなり，中央集権化されてしまう問題があった．

4 提案手法

4.1 機能の分散配置

本研究では，ブロックチェーンネットワーク内に，開示制御サーバのような集中管理のサーバを持たずとも，ノード間で分散して連携動作する開示制御の手法を提案する．ここでのノードとは，独立した権限を有する組織ごと（例えば，企業や団体など）に一つとして運用されているものとする．ノードには，ブロックチェーン基本ソフトおよびこれと連携するアプリケーションサーバが配置されている． 各組織に属するユーザはアプリケーションサーバにアクセスし，GUI 等を通じて処理要求を行う．アプリケーションサーバは，ユーザの操作を受け付け，ブロ

Decentralized Disclosure Control for Structured Data on Blockchain

† NTT Service Evolution Laboratories, Nippon Telegraph and Telephone Corp.

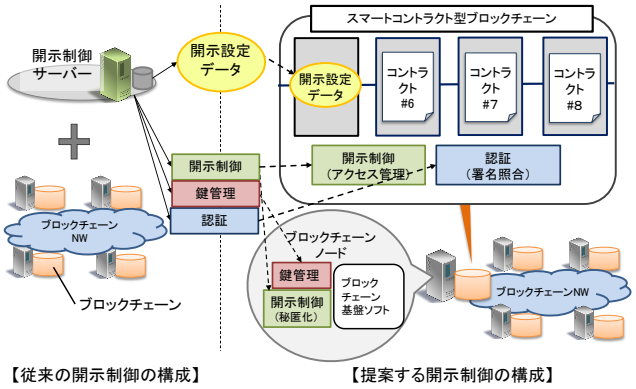

図 1 開示制御手法のシステム構成

ックチェーン基本ソフトに備わる API を利用し，ブロックチェーンに必要なデータを登録する．

著者らは，手法の検討にあたり，開示制御サーバの機能を次の通りに整理した．

(1) 開示設定データ：どのユーザにどの項目を提示するかを示す設定情報
(2) 開示制御（アクセス管理・秘匿化）：開示設定データを参照し，値の読み書き権限や暗号・復号化を制御する
(3) 鍵管理：値が暗号化されている際に秘密鍵を管理する
(4) 認証：アクセスしたユーザが許可を持つ正当なユーザか判別する

本提案手法の特徴は，図 1 に示す通り，従来開示制御サーバで集中管理していた上記の機能をスマートコントラクトと BC ノード上に分散配置することにある．具体的には(1)開示設定データや(2)アクセス管理に関わる開示制御，(4)署名検証を利用した認証の機能など，ロジックの共有が可能な範囲はスマートコントラクト上に配置し，(3)鍵管理や(2)秘匿化に関わる開示制御の機能など，保護すべき情報の処理についてはノード内に配置した．

4.2 開示制御プロトコル

さらに，構造化データの登録だけではなく，開示制御に必要なデータの登録も，トランザクションを介して行い，ブロックチェーンシステム外の経路を全て排除した開示制御プロトコルを開発した．

具体的には，図 2 に示すように，開示制御に必要な情報を生成する際に，個々の処理をまとめてカプセル化し，最終的に署名付きトランザクションに含めてブロックチェーンに登録し，情報を共有する．秘匿化の処理については，秘密を共有するノード間で共通鍵を配布する必要があるが，共通鍵を受け渡すための事前の鍵交換の処理も含めて，全てブロックチェーンを介して情報の交換を実施する．

このように全ての処理がブロックチェーン上（on-blockchain）で完結することにより，非中央集権性は必然的に保たれ，また署名付きのトランザクションとして処理の証跡がブロックチェーン上に

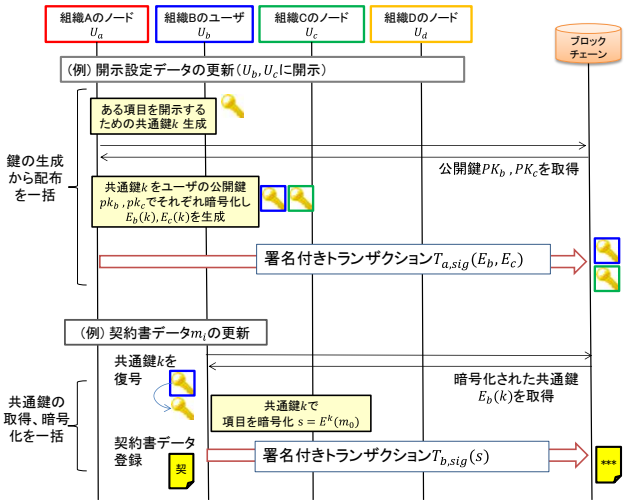

図 2 開示制御プロトコルの例

残るため，記録の証拠性も向上する．

5 実装

提案手法は，オープンソースのブロックチェーン基盤 Hyperledger Fabric(v1.0.0)を用いて実装できることを確認した．ノード内のアプリケーションサーバの構築には，Node.js 用 SDK（v1.0.0）を利用する．開示制御（秘匿化）や鍵管理の各機能はアプリケーションサーバのモジュールとして実装し組み込む．本モジュールは仮想コンテナ docker 上で動作する Hyperledger Fabric の基本ソフト(Peer や Orderer)と連携して，開示制御プロトコルが実現できることを確認した

6 おわりに

本稿では，ブロックチェーン上の構造化データに対して，集中管理された開示制御サーバを持たずとも開示制御可能な手法について提案した．本手法のフィージビリティについては Hyperledger Fabric にて実装し動作することを確認した．

本手法においては，開示制御に必要なデータの更新は全てトランザクションを使用し，on-blockchain で完結するプロトコルとなる．このためブロックチェーン上に全ての処理の証跡が残り，証拠性の向上が期待できる．一方で，ブロックチェーンに記録するデータの肥大化が懸念されるため，長期的な実用にあたっては性能評価を通じて影響を調査する必要がある．

参考文献

[1] S. Underwood, “Blockchain beyond bitcoin,” Communications of the ACM, vol. 59, Issue 11, 2016.
[2] Ethereum Project, online, https://www.ethereum.org/
[3] Welcome to Hyperledger Fabric, online, https://hyperledger-fabric.readthedocs.io/en/release/
[4] G. Zyskind and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data." *Security and Privacy Workshops (SPW)*, IEEE, 2015.