6D-02

複数データストアを持つブロックチェーンアーキテクチャの提案

大橋　盛徳†　　渡邊　大喜†　　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

ブロックチェーンをデータ管理に利用し、登録データの真正性を保証する使い方が提案されている[1][2]。真正性を保証したいデータをトランザクションに記載し、ブロックチェーンの台帳に登録する方法である。台帳は、一度登録したデータの書き換えができないため、肥大化を回避する目的からデータ本体ではなく、データのハッシュ値のみを台帳に登録する真正性の保証方式が主流となっている。

しかし、ハッシュ値だけでは、大きく 2 点の問題がある。1 点目は、正しいハッシュ値が登録されていることをブロックチェーンネットワークの参加者が検証できないことである。2 点目は、ハッシュ値だけでは、データの内容に基づいたスマートコントラクトを実現できないことである。

本稿では、1 点目の問題に対して、トランザクションに電子署名の範囲外となるデータフィールドを新設する方法を提案する。この方法により登録するデータをブロックチェーンネットワークの参加者で検証しつつ、データ本体を台帳に残さない制御が可能になる。2 点目の問題に対しては、データ本体やデータ本体に対する処理結果を一時的に格納するデータストアを新設する方法を提案する。この方法により、トランザクションに含まずともデータ本体やデータに対する処理結果を参加者間で同期し、必要なタイミングで削除できるようになる。

2. 課題

ブロックチェーンの台帳は、電子署名付きのトランザクションを含むデータブロックをハッシュでつないだ構造をしており、過去の履歴が残っている[3]。履歴が全てあり、電子署名により守られるため、一度登録されたデータは、整合性をすべての参加者で検証することができ、信頼性が高い。しかし、登録されるデータそのものの正誤は検証することができない弱点がある。そのため、データの真正性保証にブロックチェーンを利用する場合、ハッシュ値のみでは、誤ったデータの登録を防ぐ事ができない。著作権をブロックチェーンで管理する場合を例にすると、ユーザからのコンテンツ登録を仲介し、正しい権利情報が登録されることを保証する特定の信頼できる参加者を仮定する必要がある。

ブロックチェーンは相互検証の下に、処理を自動実行するスマートコントラクトのプラットフォームとしても期待されているが、ハッシュ値のみでは、処理の実行がブロックチェーン外となり、実行結果の信頼性もまた、実行を担う特定の参加者に依存することになる。

これら問題に対処するためには、以下 2 つの相反する要件を満たす必要がある。1 つ目は、データ本体に対する検証や特定処理の実行をブロックチェーンの相互検証の下で行えるようにすることであり、2 つ目は、台帳が肥大化を軽減できることである。

3. 提案方式

我々の方式は、トランザクション構造とデータ保存、共有の仕組みに工夫がある。

まず図 1 に示すトランザクションの構造から解説する。通常のブロックチェーンでは、トランザクション全体に対して電子署名がつけられ、改ざんを防止しているため、トランザクションの一部を削除することはできない。そこで、我々は、トランザクションに署名の範囲外の領域(除外データ領域)を新設した。署名の範囲外とすることで、ブロックにトランザクションを含める際に当該除外データ領域のデータを削除することができる。除外データ領域のデータのハッシュ値を署名の範囲内に入れることで、当該領域の改ざんを防ぐ事ができる。

ユーザから発行されたトランザクションは、各ノードに共有され、トランザクションの検証後にブロックにまとめられる。ブロックに入る際に、トランザクションの除外データ領域を削除することで台帳の肥大化を軽減することができる。

A new Blockchain architecture with additional state database

†Shigenori OHASHI, Hiroki WATANABE, Shigeru FUJIMURA, Atsushi NAKADAIRA
†NTT Service Evolution Laboratories, NTT Corporation

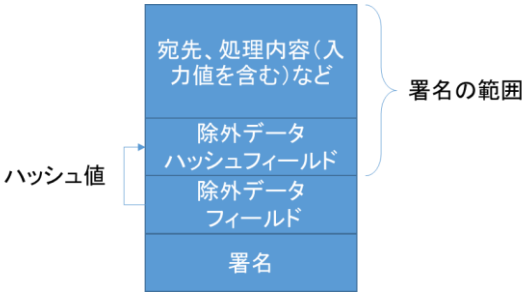

図 1: トランザクション構造

次にデータの保存と共有の仕組みについて解説する。ブロックに除外データを含めない場合には、データ本体やそのデータに対する処理結果の共有が課題となる。Ethereum[3]を始めとするスマートコントラクト型のブロックチェーンでは、台帳とともに処理結果を格納する状態データベースを持ち、ブロックの共有プロセスで、更新される。我々は、図 2 に示すようにブロックから除外したデータやそのデータに対する処理結果を格納する新たなデータストア（一時データベース）を設け、ブロックの共有プロセスを利用して更新できるようにした。ブロックの共有プロセスでは、図 3 に示すようにブロックとともに除外データを共有している。一時データベースは、通常の状態データベースとは独立しているため、データの保存ポリシーを分離できる。例えば、登録するハッシュ値とデータが正しいことを確認した後に、一時データベースから即座にデータ本体を削除し、ブロックチェーンネットワークに残さない制御も可能になる。

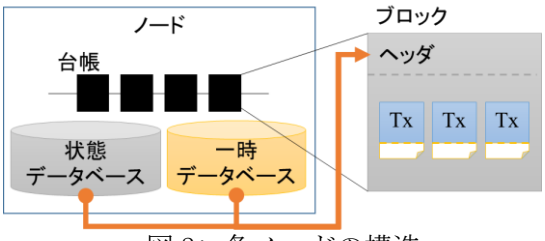

図 2: 各ノードの構造

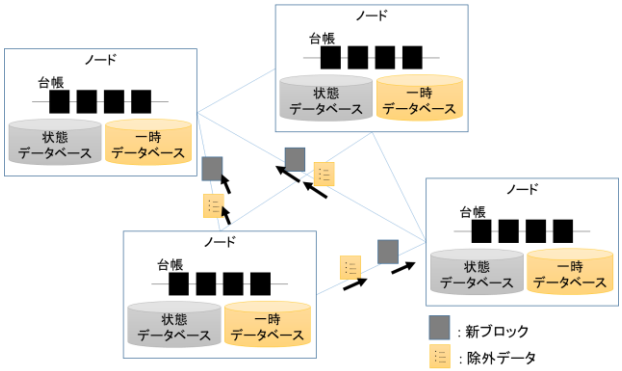

図 3: ブロック共有のイメージ

4. 評価

Python 版の Ethereum をベースに提案方式を実装した。一定サイズのランダムデータを一時データベースに登録し、検証後に削除する操作を一定期間実施し、台帳および 2 つのデータベースのデータサイズを計測する評価実験を実施した。提案方式は、ハッシュ値のみをブロックチェーンネットワークに登録する従来方式と同様に台帳の肥大化を軽減できることを確認した。また、データ本体をブロックチェーンネットワークに登録する場合に比べると、提案方式では、2048bytes のデータに対して約 80%のブロックチェーン上のデータ量削減を実現できた。

5. おわりに

ブロックチェーンで、データ本体を管理できる新しいブロックチェーンのアーキテクチャを提案した。評価実験で使用したデータサイズは、アイコン程度の画像ファイルにしか相当しないが、より大きなサイズのファイルにも同様の方式で対応できると考えられる。また、評価実験で使用したデータサイズでも楽譜などの設計図に相当するデータの管理などで利用できると考えられる。

提案方式は、Ethereum の仮想実行環境の外部で処理する方式のみであり、通常のトランザクションから一時データベースのデータを扱うことはできない。今後は通常のトランザクションから実行できるスマートコントラクトで一時データベースに含まれるデータを扱うことを可能にする必要がある。

参考文献

[1] “Proof of Existence”, [online]https://poex.io, 参照 Jan.10, 2018.
[2] “21st century notarization | Stampery”, stampery, [online]https://stampery.com/, 参照 Mar. 29, 2016.
[3] “Ethereum Project”, Ethereum Foundation, [online]https://www.ethereum.org/, 参照 Mar. 29, 2016.