7N-01

サプライチェーン連携基盤における What-If 分析フレームワークを用いた改ざんデータ復元手法の提案

堀 遥 †　　Le Hieu Hanh†　　小口　正人 †
† お茶の水女子大学

1　はじめに

脱炭素社会実現に向け，サプライチェーン全体での CFP(Carbon Footprint of Products) 管理基盤の実現が期待されている．このようなシステムは，ホットスポットの特定などによる CFP 削減への貢献が挙げられる一方で，環境規制への対応圧力などによるデータ改ざんのリスクや複数企業が参加するデータ連携基盤での正確な追跡と信頼性の担保への対応が求められる．

我々はこれまでに，製品間に階層関係を持つ仕組みの CFP データの改ざんを検知し，具体的な発生箇所が特定可能であるようなデータ連携基盤をブロックチェーンベースで提案してきた [1]．本稿ではこれに加え，What-If 分析フレームワークを導入することで，特定した改ざんが本来の正しいデータに復元可能となるようなシステムを検討する．ここで，提案システムは自動車部品製造業での運用を想定した設計，実証実験を行った．

2　関連研究

2.1　先行研究

先行研究 [1] では，CFP データに発生した改ざんを特定可能にすべく，“ハッシュ部品木” を提案した．各部品の CFP データをハッシュ化し，部品の構造関係に従って複数のハッシュ値を XOR で組み合わせることで，部品ごとのハッシュ値を決定する．具体的なハッシュ値の定義は以下の通りである．ここで，$\boldsymbol{Hash}(x)$ は x にハッシュ関数 SHA256 を適用した結果を出力し，$\boldsymbol{Path}(x,y)$ は部品 x から部品 y までのパスを出力する．

- 任意の単品部品 P_s : $H_s = Hash(CFP_s)$
- n 個の子部品 $\{P_{c-1}, P_{c-2}, \cdots, P_{c-n}\}$ $(2 \leq i \leq n)$ を持つ任意の構成部品 P_c :
  $H_c = Hash(CFP_c) \oplus H_{c-1} \oplus H_{c-2} \oplus \cdots \oplus H_{c-n}$
  P_{c-i} が重複部品の場合，H_{c-i} を以下に置換する．
  $Hash(H_{c-i}||Path(P_c, root))$

ハッシュ部品木における XOR が持つ性質の貢献は二点である．第一に，交換法則 $(a \oplus b = b \oplus a)$ によって，ハッシュ値の統合順序に非依存となる．第二に，自己反転性 $(a \oplus a = 0)$ によって，改ざん特定が実現可能となる．詳細は 3.2 節で述べる．一方で，ハッシュ部品木に重複部品が含まれる場合，上位ノードにて重複部品同士で自己反転性が働いてしまうため，ハッシュ値の置換が必要となる．

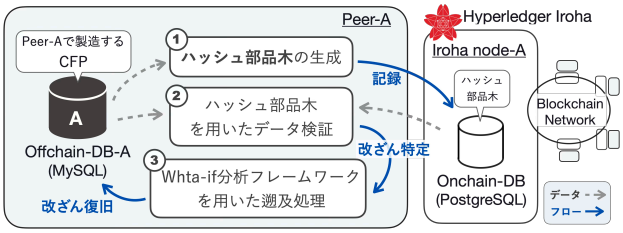

図 1: 提案システム

2.2　What-If 分析フレームワークに関する研究

What-If 分析では，データ記録や取引履歴における仮想的な変更をシミュレートし，変更された条件下での結果を予測する．従来の What-If 分析フレームワークは，ソフトウェアの Application 層または Database 層のいずれかのみを対象とした設計であったが，本稿で紹介する “What-If 分析フレームワーク Ultraverse” はこれらの層をシームレスに統合する手法を実現している [2]．Ultraverse は準備段階にて，抽象構文木と動的シンボリック実行によって，アプリケーションコードを同等の SQL プロシージャに変換する．これにより，意味論において正確性の高い遡及処理を高速に実行する．さらに，列・行単位のクエリの依存解析や Hash jamper らによって効率的な遡及処理が実現される．

3　提案手法

提案システムの全体像を図 1 に示す．各 Peer は Hyperledger Iroha で接続する．CFP データの改ざんを特定・復元する手法を以降の節に述べる．

3.1　改ざん特定の事前準備: ハッシュ部品木の生成

改ざんを特定可能にするために，あらかじめハッシュ部品木を以下の手順で生成しておく必要がある．

1) 全 Peer の Offchain-DB から CO_2 を集約．
2) 定義に従い部品ごとのハッシュ値を決定．
3) Hyperledger Iroha のスマートコントラクトで 2) を Onchain-DB に格納．

3.2　データ検証による改ざん特定

ハッシュ部品木を Onchain-DB に格納することで，Offchain-DB の CFP データに改ざんが発生しても，“

A Method for Recovering Falsified Data using a What-If Analysis Framework on a Supply Chain Integration Platform
†Haruka Hori　†Hieu Hanh Le　†Masato Oguchi
†Ochanomizu University

Copyright ©2026 Information Processing Society of Japan.
All Rights Reserved.

どの部品に改ざんが発生したのか” という具体的な特定が可能となる．以下に改ざんの検証手順を示す．

1) 検証部品のハッシュ値を現在のデータから再算出．
2) OnchainDB に保管したハッシュ値と比較．
3) 2) が不一致なら改ざん特定処理を実行．
   i. 検証部品からハッシュの不一致を探索．
   ii. 親 (H'_P) から改ざんされた子の影響 (H'_C) を除去したものに，Onchain-DB から取得した値 (H'_C) を XOR した $Check$ を算出．

$$Check = H'_P \oplus (H'_C \oplus H_C)$$

   iii. $Check$ と正しいハッシュ値 (H_C) を比較し改ざんの再判定を行う．一致なら改ざんは子部品のみ，不一致ならみ特定の改ざんあり．

3.3 改ざん復旧

Ultraverse により，意味論的に矛盾のない遡及処理で「もし過去のあの時点で改ざんが発生していなかったら，現在のデータはどうなっているか？」という What-If 分析を実現する．以下に手順を示す．

1) 特定した改ざんのトランザクション ID を取得．
2) Ultraverse に 1) を与え What-If 分析によるデータ復旧を実行．

4 評価

本稿では，3.3 節で紹介した改ざん復旧の手順において，2) の Ultraverse による遡及処理の時間的コストを計測した．ここで本評価は，先行研究 [1] で提案した「CFP データの改ざんが特定可能であるシステム」上に改ざん修復機能を追加する前提で実施される．

実装の構築環境とマシンの性能は図 2 に示す．2 種の DBMS が混在する理由としては，Hyper-ledger Iroha は PostgreSQL または RocksDB を Onchain-DB として利用できる仕様であり，Ultraverse は MySQL をメイン DBMS として採用しているためである．

30,000 件の初期データを反映させたの Offchain-DB に，UPDATE トランザクションを 10,000/100,000/1,000,000 個ランダム発生させて実行した．トランザクションのうち，一件を改ざんトランザクションとし，トランザクション ID を Ultraverse に与えて遡及処理を行なった．各トランザクション数ごとに，3 回ずつ実行時間を計測し，平均値を表 1 に示した．

自動車一台分の部品数に相当するデータ数 30,000 点の場合，ハッシュ部品木の生成には 12 秒程度であった．また，30,000 点のうち 10%の部品に改ざんが発生した場合，16 秒程度で全ての改ざんが特定可能であることが確認されている．また，表 1 より，トランザクション数 10,000 の準備時間は 0.90 秒程度であった．また，その 100 倍規模においても 2.8 秒程度という結果が得

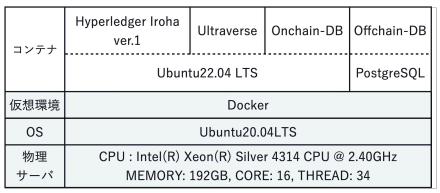

図 2: 構築環境

表 1: 実験結果

トランザクション数	準備時間	遡及処理時間
10,000	0.901 秒	0.194 秒
100,000	1.782 秒	0.212 秒
1,000,000	2.804 秒	0.230 秒

られた．ここで，本シナリオにおいて最も重視されるべき要件は，通信の低遅延性ではなく，正確性および信頼性である．多少の計算時間を費やしてでも正確な状態へ修復することの方がシステムとしての価値が高い．以上より，提案システムは現実的な運用時間であると考えられる．

5 結論

本稿では，先行研究で開発した改ざんが特定可能な自動車部品 CFP データ連携システムに，What-If 分析フレームワークの Ultraverse を導入することで改ざん復旧可能となるような手法を検討した．評価実験では，トランザクション数 1,00,000 において，What-If 分析実行のための準備時間は 2.80 秒程度，実際の遡及処理時間は 0.23 秒程度であることが確認された．

今後は，第一に，改ざんが特定可能な自動車部品 CFP データ連携システムと Ultraverse のインターフェイスを作成していく．その上で，改ざんを特定してから復旧するまでの時間，リソースコストを調査する．第二に，二つの汎用 DBMS が混在する現提案手法において，よりシームレスな挙動を目指す．そのために，Hyperledger Iroha の API であり，Ultraverse のホストアプリケーション言語でもある Javascript の使用を検討していく．

謝辞

本研究は一部，JST CREST JPMJCR22M2 の支援を受けたものである．

参考文献

[1] 堀遥, Le Hieu Hanh, 小口正人. 改ざんが特定可能なデータ連携基盤の構築と自動車部品製造業への応用における評価. 電子情報通信学会 ネットワークシステム研究会（NS）信学技報, 第 125 巻, pp. 1–6, 2025.

[2] R. Ko, C. Xiao, M. Onizuka, Z. Lin, and Y. Huang. Ultraverse: An efficient what-if analysis framework for software applications interacting with database systems. *PACMMOD*, Vol. 3, No. 1, 2025.

Copyright ©2026 Information Processing Society of Japan.
All Rights Reserved.