一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2022-219(2023-03)

IOTA を用いた ICN の名前管理方式

岡田　鉄平†　　上山　憲昭†

† 立命館大学 情報理工学部 情報理工学科
〒525-0058 滋賀県草津市野路東 1-1-1
E-mail: †is0498ex@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

あらまし　従来のインターネットのように，IP アドレスをもとにコンテンツを転送するのではなく，要求されたコンテンツの名称をもとにルータにキャッシュしながらコンテンツ要求者 (consumer) に配信を行う情報指向ネットワーク (ICN: information-centric networking) が次世代のネットワークとして注目を集めている．この ICN において，誰でも publisher としてコンテンツをアップロードすることが可能だが，正当な publisher を騙る攻撃者が，実在するコンテンツ名で fake コンテンツをアップロードすることでキャッシュの機能を低下させる CPA (content poisoning attack) の問題が指摘されている．既存研究では，CPA の対策として，consumer が受信したコンテンツの正当性を，公開鍵を用いたディジタル署名によって判断し，不当なコンテンツをルータに通知する方式が提案されている．しかし，コンテンツに紐づいた公開鍵により生成されたディジタル署名と一致する fake 型 CPA では，先の方式では検知は困難となる．中でも，公開鍵を管理している認証局 (CA: certification authority) の職員が攻撃者と結託し，正当な publisher の公開鍵を攻撃者のものに書き換えることにより，実在する高人気コンテンツを騙る fake コンテンツをキャッシュに注入する詐称 fake 型 CPA は対策が困難である．そこで本稿では，publisher によるコンテンツのアップロードに際し，登録データの改ざんが困難な分散型台帳技術の一つである IOTA でコンテンツ名を管理し，詐称 fake 型 CPA を未然に防ぐ方式を提案する．提案方式では，台帳内でコンテンツ名を検索する四つの探索手法の検索時間と，必要メモリ量の比較を行う．その結果，検索時間と必要メモリ量のトレードオフの関係を確認した．
キーワード　ICN，詐称 fake 型 CPA，IOTA

Name Management Using IOTA in ICN

Teppei OKADA† and Noriaki KAMIYAMA†

† College of Information Science and Engineering, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-0058
E-mail: †is0498ex@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

Abstract　Information-centric networking (ICN), which doesn't forward content based on IP addresses as in the conventional Internet, but cache content in routers based on the name of the requested content and deliver it to the content requester (consumer), has got attention as a next-generation network. In ICN, although anyone can upload content as a publisher, the problem of content poisoning attack (CPA), which an attacker pretending to be a legitimate publisher uploads fake content under a real content name and degrades the cache functionality, has been pointed out. As a countermeasure against CPA, existing research has proposed a method that determines the legitimacy of content received by consumers using digital signatures with public keys, and alerts routers of unjustified content. However, it is difficult for the previous method to detect fake-CPA that matches a digital signature generated by a public key associated with the content. Especially, it is also difficult to take measures against spoofed fake-CPA, in which the staff of the certification authority (CA) that manages the public key colludes with the attacker to rewrite the legitimate publisher's public key to the attacker's and inject fake contents that pretends to be real and popular contents into the cache. In this paper, we propose a method to prevent spoofed fake-CPA by managing content names with IOTA, a distributed ledger technology that is difficult to tamper with registration data, when the content is uploaded by publishers. We also compare the search time and memory requirements of four search methods that search for content names managed in the ledger in the proposed method. As a result, we confirmed the trade-off between the search time and memory requirements.
Key words　ICN, spoofed fake-CPA, IOTA

1. はじめに

従来のインターネットでは，ユーザがドメインネームの名前解決を DNS (domain name system) サーバに依頼し，そこで回答された IP アドレスに基づいて動画像やウェブサイトなどのコンテンツを配信する．そのため通信開始時にコンテンツの配信元を定める必要があるが，動的に変化するネットワークの負荷状態やネットワークトポロジを考慮した効率的な配信元の決定が難しい．そこで通信開始時に名前解決を行わず，要求されたコンテンツの名称をもとに要求パケット (interest) を転送し，配信サーバ (publisher) からコンテンツ要求者 (consumer) へ経由するルータにキャッシュしながらコンテンツを配信する情報指向ネットワーク (ICN: information-centric networking) が，次世代のコンテンツ配信方式として注目を集めている．

この ICN において，consumer は取得したいコンテンツの名称を interest として送信する．その interest を受信したルータのキャッシュに要求コンテンツが存在する場合は，interest を棄却し，そこからコンテンツの配信を行い，存在しない場合は，要求コンテンツが存在するルータ，もしくは publisher に interest を転送し，同様に配信を行う．配信元から経由するルータにキャッシュしながらコンテンツを転送するため，次回 consumer から同じ要求があった場合，そのコンテンツがキャッシュされている近くのルータから配信を行うことが可能であり，配信コストの抑制が期待される．

ICN では誰もが publisher としてコンテンツをアップロード可能だが，正当な publisher を騙る攻撃者が，実在するコンテンツ名で fake コンテンツをアップロードすることでキャッシュの機能を低下させる CPA (content poisoning attack) の問題が指摘されている [1]．CPA の対策として，consumer が公開鍵暗号を用いたディジタル署名によりコンテンツの正当性を判断し，不当なコンテンツをルータに通知する方式が提案されている [2]．CPA は，コンテンツに紐づいた公開鍵により生成されたディジタル署名と一致する fake 型と，一致しない corrupted 型があるが [3]，前者に対しては [2] の方式では検知は困難となる．中でも，公開鍵を管理する認証局 (CA: certification authority) の職員が攻撃者と結託し，正当な publisher の公開鍵を攻撃者の公開鍵に書き換えることにより，実在する高人気コンテンツを騙る fake コンテンツをキャッシュに注入する詐称 fake 型 CPA は，対策が困難である．本攻撃は，アクセス数が多い人気コンテンツが詐称されると攻撃の影響が大きいことが推測される [4]．また，正当な publisher の公開鍵を乗っ取って書き換えているため，攻撃者のコンテンツの方が正当化される．

詐称 fake 型 CPA は，公開鍵を CA のような一つの機関のみで管理することにより発生する．そこで本稿では，分散型台帳技術の一つである IOTA [5] を用いてコンテンツ名を管理することで詐称 fake 型 CPA を未然に防ぐ方式を提案する．分散型台帳としては blockchain が代表的であるが，blockchain はスケール性に課題があるのに対し，IOTA はスケール性が高い．提案方式では，publisher のコンテンツ登録時，ならびに consumer の配信要求時に台帳内で該当する transaction を探索する必要がある．そこで transaction の探索手法として，ハッシュチェイン法，二分探索木，幅優先探索，深さ優先探索を各々用いた場合の検索時間と必要メモリ量を比較する．

2. 節では CPA について述べ，3. 節では IOTA について述べる．そして 4. 節で提案方式を述べ，5. 節で性能評価について述べる．最後に 6. 節でまとめを述べる．

2. CPA

corrupted 型 CPA に対しては署名を検査することにより容易に検知できるが，fake 型 CPA では検知が困難である．我々は以前，fake 型 CPA を独自 fake 型 CPA と詐称 fake 型 CPA に分類した [4]．図 1 に fake 型 CPA の動作の概略を示す．

独自 fake 型では，攻撃者が独自に作成した架空のコンテンツを結託ユーザ (colluder) から要求することで，ルータに注入する．図 1 に示した例では，攻撃者 (A_a) は独自に作成した fake コンテンツ y をネットワーク上に公開している．独自 fake 型 CPA では，正常ユーザ (legitimate user) からの fake コンテンツに対する要求はなく，少数の結託ユーザからの要求のみであるため，fake コンテンツがネットワークを流れる量は少ない．そのため fake コンテンツがルータにキャッシュされるのは一時的であるため，攻撃の影響は小さいと推測される．一方，詐称 fake 型では，実在するコンテンツを騙る偽のコンテンツを正常ユーザから要求することで，ルータに注入する．図 1 に示した例では，攻撃者 (A_b) は実在するコンテンツ x を騙った偽のコンテンツ x_{fake} をネットワーク上に公開し，正当なコンテンツの名前を乗っ取って偽のコンテンツを配信する．詐称 fake 型 CPA では多数の正常ユーザから fake コンテンツに対し要求があるため，長時間，fake コンテンツがキャッシュに残る可能性が高く，攻撃の影響は高いと推測される．本稿では攻撃の影響が大きい詐称 fake 型 CPA を防ぐことに焦点を当てる．

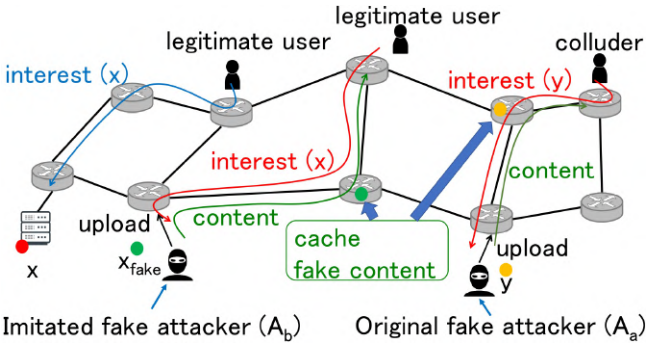

図 1　fake 型 CPA

3. IOTA

IOTA では，新しい transaction が未承認の transaction である tip の中から二つ選択し，tangle と呼ばれる有向非巡回グラフ (DAG: directed acyclic graph) を形成することで分散型台帳が実装される [5]．また tangle 構造では，全ての transaction から直接的，間接的に承認を受ける最初の transaction である，ジェネシス transaction が存在する．図 2 に DAG を例示する．A, B, C, D, E, F は transaction を表しており，それぞれ既存の二つの transaction を参照する．また，右下の数字はその transaction の重みを，左上の数字は累積重みを表している．累積重みは，直接的，間接的にその transaction を参照している他の transaction の重みと，自身の重みの総和と定義される [6]．例えば，F の累積重みは 12 で，これは A, B, C, D, E の重みと自身の重みを足し合わせたものとなっている．また，選択されていない A や B のような transaction が tip である．その tip 選択アルゴリズムとしては，以下に示す三つが代表的である．

- **uniform random selection (URS)**: 存在している tip の中からランダムに二つ選択する．
- **unweighted random walk (URW)**: ジェネシス transaction から参照されている transaction を等確率に選択していき，tip を選択する．これを二回行うことで，二つの tip を選択する．
- **weighted random walk (WRW)**: URW と同様にジェネシス transaction から選択していくが，累積重みを考慮して選択する．WRW では，重みの大きな transaction が優先して選択される．

WRW における transaction y から x への遷移確率 P_{xy} は次式で得られる．

$$P_{xy} = \frac{e^{-\alpha(H_x - H_y)}}{\sum_{z:\ z \to x} e^{-\alpha(H_x - H_z)}} \qquad (1)$$

ここで，H_x, H_y は，transaction x および y の累積重みを表している．また，$\alpha(\geq 0)$ は累積重みのパラメタである．$\alpha = 0$ のときは重みのないランダムウォーク，つまり URW となる．また，α の値が大きくなるにつれ，累積重みの影響が大きくなるため，選択される tip に偏りが生じる．

IOTA では，同じ暗号資産を複数回使用することにより攻撃者が不正に資金を取得する二重支払い攻撃が問題視されている．その例として，攻撃者が tangle を二手に分岐させて二重支払い transaction の累積重みを増やすことで，それを承認させる分裂攻撃 [7] や，メイン tangle とは別の tangle に二重支払い transaction を追加し，短時間でリンクの数を増やすことで承認させやすくするパラサイトチェーン攻撃 [8] が挙げられる．これらの攻撃は，tip 選択の際に累積重みを考慮しない URS や URW で発生する可能性が高いため，対策として tangle に分岐や分裂が生じても累積重みの大きい方のみに遷移されるように，式 (1) の α の値を大きく設定した WRW を採用することが有効である．

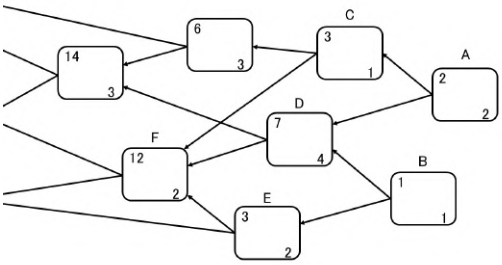

図 2　IOTA における DAG

IOTA を用いた関連研究として，エネルギー市場において任意の家庭同士が電力会社などを介さず，IOTA 台帳を通して手数料なしで直接電力を売買する P2P 電力型トレード方式が提案されている [6]．また [9] では，IOTA を用いたマイクロペイメントによって，低遅延や帯域幅など，保証されたネットワーク資源に対してユーザがネットワークプロバイダに料金を支払う方式が提案されており，ユーザがネットワークの状況に関わらず固定の料金を支払うことなく，サービスを利用できることが保証される．また [10] では，協調高度道路交通システムにおいて，車両が送信するメッセージに付属する証明書を IOTA で管理する方式が提案されている．これにより，証明書発行の透明性が保証される．

4. 提 案 方 式

提案方式では，transaction にコンテンツの prefix，ID，公開鍵およびそれに対応するディジタル署名，そして prefix，公開鍵，ディジタル署名により構成されたコンテンツ名を管理する．分散型台帳の性質上，登録データの改ざんは困難であるため，正当性が担保される．

4.1 概　　要

コンテンツの prefix とコンテンツ名の定義を例を用いて示す．コンテンツの prefix は，「www.ritsumei.ac.jp」のようなドメイン名である．一方コンテンツ名は，「www.ritsumei.ac.jp/514720/1307f51b14b5」のように，コンテンツの prefix，公開鍵，ディジタル署名で構成されており，「/」でそれらを区別する．また，提案方式の流れを以下に示す．

（1） publisher がコンテンツのアップロードに際して，コンテンツの prefix，ID，公開鍵，ディジタル署名，コンテンツ名を transaction に登録

（2） 重複するコンテンツ名の管理を防ぐため，既に同じコンテンツ名が管理されているかどうかを，コンテンツの prefix で IOTA 台帳内を検索し，既に存在していれば登録を拒否し，そうでなければ登録

（3） publisher によるアップロードが終了した後, consumer がコンテンツの prefix を要求

（4） 要求された prefix で IOTA 台帳内を検索し，ヒットしたコンテンツ名を consumer に回答

（5） consumer がそのコンテンツ名で interest を送信し，コンテンツを要求

このように提案方式では，DNS の名前解決のような流れで consumer がコンテンツ名を取得する．提案方式を用いることで，ある prefix に対して最初に IOTA 上に名前を登録した publisher のみが正答な publisher として保証されるため，実在する Predix に対して攻撃者が偽りのコンテンツをネットワーク上に公開する詐称 fake 型 CPA を防ぐことが可能である．

4.2 DAG におけるコンテンツ名探索手法

提案方式では，publisher によるコンテンツ名登録時と，consumer による名前解決時の二つのタイミングで，DAG 上の該当 transaction を探索する必要がある．transaction の探索法により，探索に要する遅延時間や所要メモリ量が大きく影響する．本稿では，以下の四つの探索手法を各々，想定した場合の評価を行う．

- **ハッシュチェイン法 (hash: hash-chain)**: データの検索キーをハッシュ関数にかけて得られたハッシュ値に該当するアドレスのハッシュテーブルの要素にデータを格納する．異なるデータでも同じハッシュ値となる衝突が発生しうるが，ハッシュチェイン法ではそれらを連結リストで繋ぐことで，同じバケットに複数のデータを管理することが可能である．提案方式では，prefix を数値に変換したものを検索キーとし，ハッシュテーブルで該当するコンテンツの名前を格納する DAG の transaction のアドレスを管理する．すなわちハッシュチェインで得られたアドレスをもとに DAG 上にアクセスすることでコンテンツ名を取得する．
- **二分探索木 (bst: binary search tree)**: 木構造の一種で，各ノードのキーは「左の子のキー < 親のキー < 右の子のキー」という性質を有する．最上位のノードである根から探索を開始し，探索キーがノードの値より小さければ左に，大きければ右に移動する．この流れを発見するまで繰り返す．二分探索木でもハッシュチェイン法と同様に，コンテンツの prefix を探索キーとし，bst の各ノードには DAG の transaction のアドレスを管理する．
- **幅優先探索 (bfs: breadth-first search)**: DAG 上のノード (transaction) を直接，幅優先で探索する．DAG 上のジェネシス transaction からのホップ数の小さい transaction から順に探索する．同じホップ数の transaction を全て探索しても未発見の場合，一つ大きなホップ数である transaction の探索を開始し，発見するか，全探索するまで繰り返す．
- **深さ優先探索 (dfs: depth-first search)**: DAG 上のノード (transaction) を直接，深さ優先で探索する．幅優先探索と同様に，ジェネシス transaction から探索を始めるが，子を持たない transaction，つまり tip に行き着くまで，深く探索する．tip に到達しても未発見の場合，最後に分岐した箇所まで戻り，未探索の transaction を探索する．

ハッシュチェイン法，二分探索木では該当 transaction の DAG 上のアドレスをハッシュテーブルや二分木で管理しているため，アドレスを取得後に DAG にアクセスしてコンテンツ名を取得する必要がある．一方，幅優先探索，深さ優先探索では，DAG の transaction にコンテンツ名が記録されているた

め，該当 transaction を発見した時点で探索が終了する．

5. 性能評価

提案方式における，前節で述べた四つの検索手法間で，検索時間と必要メモリ量を計算機シミュレーションにより比較する．計算機シミュレーションは，IOTA シミュレータである DAGsim [11] に変更を加えたもので実施した．検索時間は，検索時間の平均値，中央値，そして最大の検索時間の指標として上位 95% 値で評価する．また，必要メモリ量では，DAG 上で直接データを管理する幅優先探索および深さ優先探索を DAG としてまとめ，hash と bst と比較する．

5.1 評価条件

2017 年 11 月に Alexa の web ページで公開されているアクセス数上位の 8000 の web ページ [12] を閲覧した際，エラーとならずに web ページが表示された 7,131 個のドメイン数をコンテンツの名称のセットとして評価に用いる．DAG 生成時では，URS，URW，WRW の三つの tip 選択アルゴリズムに対し，URS と URW については transaction 数 N_t を 100, 1,000, 7,131 の三通りで，WRW については 100, 1,000 の二通りで DAG を生成する(注1)．

新規コンテンツ名登録による transaction 生成はレート $\lambda_1 = 50$(/秒) の指数分布に従い発生させる．また遷移確率の式 (1) における α を 0.1 とする．consumer による要求は，総要求数を 5,000 とし，発生レートを $\lambda_2 = 50$(/秒) の指数分布に従い発生させる．またハッシュチェイン法におけるテーブルのバケット数 B を 100 とする．

5.2 コンテンツ名登録時の検索時間

図 3 に，$N_t = 100$ における (a)URS，(b)URW，(c)WRW でのハッシュチェイン法，二分探索木，幅優先探索，深さ優先探索のコンテンツ名登録時の検索時間の実測値の平均値，中央値，95% 値を示す．いずれの場合も検索時間は，ハッシュチェイン法 < 二分探索木 < 深さ優先探索 < 幅優先探索となる．ハッシュチェイン法では，ハッシュ値を算出してバケットにアクセスし，その中で管理されているリストを探索するが，コンテンツ名登録時において，最初の方はバケットで管理されているデータ数も少ないため，他の手法と比べて短い時間で全探索が可能である．二分探索木においても，全てのデータではなく，必要な部分を探索するため，検索時間も比較的短い．一方，深さ優先探索，幅優先探索では，しらみ潰しに DAG を全探索するため時間がかかる．tip 選択アルゴリズムごとに比較すると，どの手法も検索時間の差はほとんどないことが確認できる．ハッシュチェイン法，二分探索木では DAG とは別のテーブルを用いて探索を行うため，DAG の形状に影響を受けない．また，幅優先探索，深さ優先探索においても，少ない transaction 数では DAG の形状に差があまりないため，結果として検索時間に差が生じにくい．

図 4 に，$N_t = 1,000$ におけるコンテンツ名登録時の検索時間の実測値の平均値，中央値，95% 値を示す．$N_t = 100$ と比べると DAG の規模が大きくなるため，いずれの tip 選択アルゴリズムでも，DAG とは別にデータを管理しているハッシュチェイン法および二分探索木と，DAG で直接管理している幅優先探索，深さ優先探索との検索時間の差が顕著になっている．WRW の幅優先探索に着目すると，URS，URW に比べて検索時間が大きいことが確認できる．これは，WRW では tip の重みが大きいものが優先して選択され，ジェネシス transaction からのホップ数が大きな transaction が多くなるため，幅優先探索では全探索に時間を費やす．深さ優先探索では異なる tip 選択アルゴリズムでも差は見られないが，この手法では，ホップ数の大きな transaction から優先して探索されるためだと推測される．なお，ハッシュチェイン法，二分探索木は検索時間が DAG の形状に依存しないため，同様にあまり差は生じない．

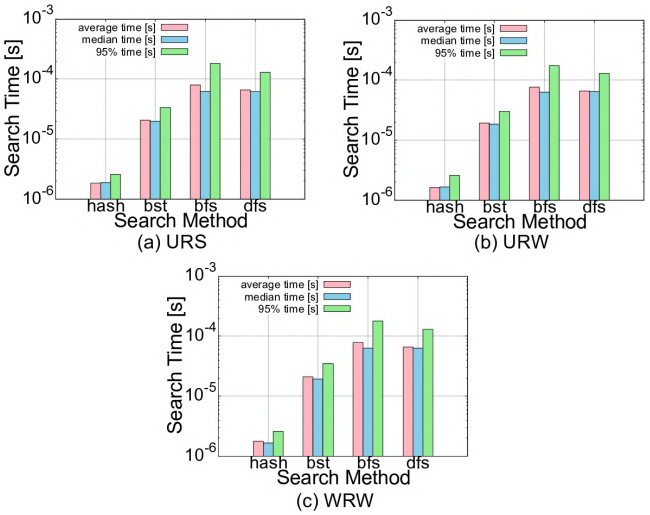

図 3　$N_t = 100$ におけるコンテンツ名登録時の検索時間

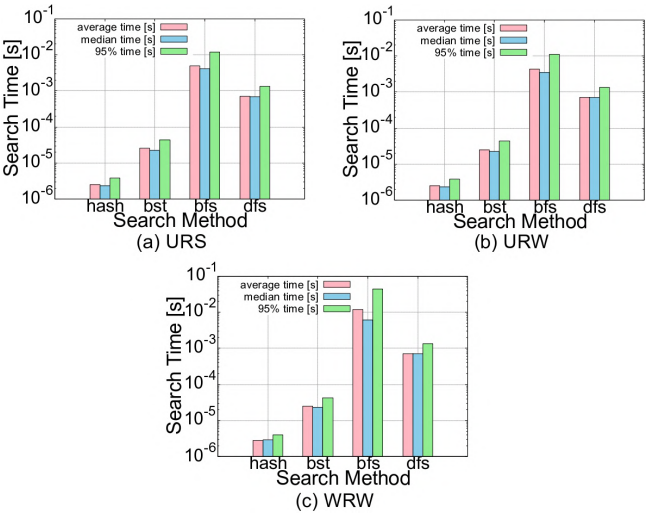

図 4　$N_t = 1,000$ におけるコンテンツ名登録時の検索時間

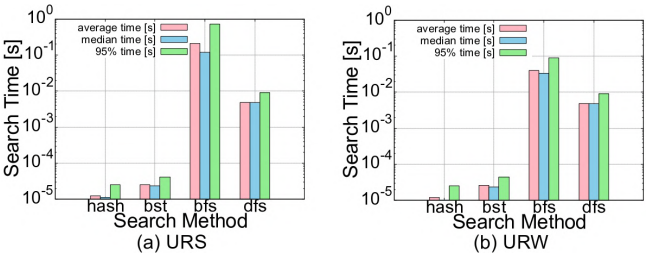

図 5　$N_t = 7,131$ におけるコンテンツ名登録時の検索時間

図 5 に，$N_t = 7,131$ におけるコンテンツ名登録時の検索時間の実測値の平均値，中央値，95% 値を示す．transaction 数が増加するとハッシュチェイン法，二分探索木と幅優先探索，深さ優先探索との差はさらに大きくなることが分かる．ここで幅優先探索において，URS の方が URW よりも探索に時間を費やしていることが確認できる．これは，URS では存在している tip の中からランダムに二つ選択するため，ホップ数の大きな tip でも選択されるが，URW ではジェネシス transaction

(注1)：WRW においては $N_t = 7,131$ のとき処理が終了しなくなったため，評価対象外とする．

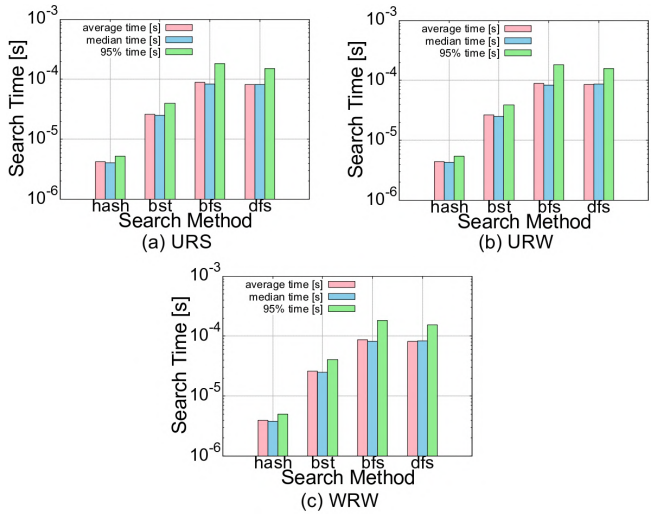

図 6　$N_t = 100$ における名前解決時の検索時間

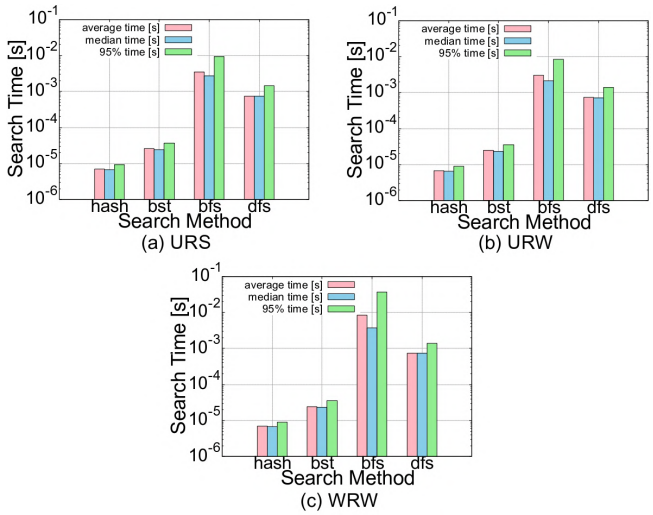

図 7　$N_t = 1,000$ における名前解決時の検索時間

から等確率に transaction を選択するため，ホップ数の小さな transaction が DAG 上に多く存在するためだと考えられる．

5.3　名前解決時の検索時間

図 6 に，$N_t = 100$ における名前解決時の検索時間の実測値の平均値，中央値，95% 値を示す．比較の結果，ハッシュチェイン法 < 二分探索木 < 深さ優先探索 < 幅優先探索となっている．こちらでもやはり transaction 数が少ないため，tip 選択アルゴリズムの違いによる検索時間の差がほとんど見られない．コンテンツ名登録時の図 3 と比較しても，検索時間にほとんど差がないことが確認できる．

図 7 に，$N_t = 1,000$ における名前解決時の検索時間の実測値の平均値，中央値，95% 値を示す．transaction 数が増加するため，図 6 と比較して，DAG で管理する幅優先探索，深さ優先探索と，テーブルなどで管理するハッシュチェイン法，二分探索木との差が大きい．また，ジェネシス transaction からのホップ数が多い transaction が多く存在するため，WRW の幅優先探索では URS，URW と比較して探索に時間を必要とする．コンテンツ名登録時の図 4 と比較すると，同様にあまり差は見られない．

図 8 に，$N_t = 7,131$ における名前解決時の検索時間の実測値の平均値，中央値，95% 値を示す．URS，URW いずれの場合も，ハッシュチェイン法と二分探索木を比較すると検索

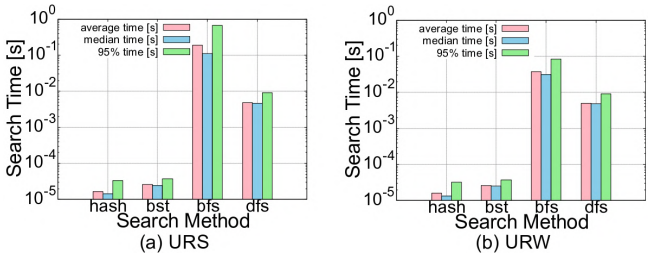

図 8　$N_t = 7,131$ における名前解決時の検索時間

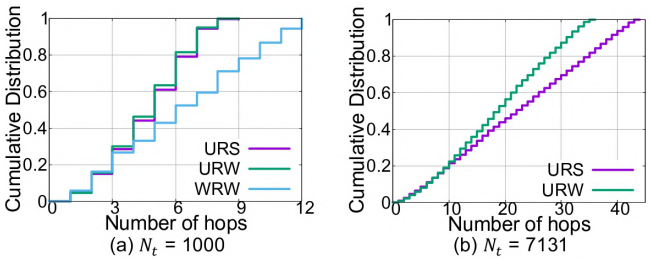

図 9　DAG 上の transaction のジェネシス transaction からのホップ数の累積分布

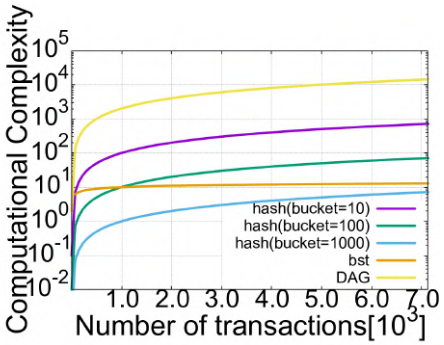

図 10　DAG の transaction 数に対する平均計算量

時間にほとんど差がないことが確認できる．ハッシュチェイン法の平均計算量は $O(N_t/B)$ であるのに対し，二分探索木は $O(\log_2 N_t)$ と表される．つまり，transaction 数が少ないときはハッシュチェイン法の方が，多いときは二分探索木の方が計算量が少ないため，結果として検索時間の差が縮まるためだと推測される．

図 9 に，DAG 上の各 transaction のジェネシス transaction からのホップ数の累積分布をプロットする．$N_t = 1,000$ では，URS と URW は同じような分布となっており，似たような DAG の形状となっていると推測される．また，WRW ではホップ数の大きな transaction が他の二つと比べて多く，図 4，図 7 に示すように，幅優先探索ではいずれも WRW が最も時間がかかっていることが確認できる．また，$N_t = 7,131$ では，URS の方が，URW よりもホップ数の多い transaction が多いことが分かる．したがって，図 5，図 8 において，いずれも URS の方が URW よりも幅優先探索に時間を費やすことが裏付けられる．また，$N_t = 100$ では，ホップ数が 1 および 2 の transaction のみ存在し，異なる tip 選択アルゴリズムでも近似的に分布しているため，DAG の形状に違いはほとんどないと推測される．ゆえに図 3 で示すように，URS，URW および WRW に対し，どの探索手法でも検索時間に差が見られない．

それぞれの探索手法の平均計算量は，前述したようにハッシュチェイン法は $O(N_t/B)$，二分探索木は $O(\log_2 N_t)$ であり，DAG は任意の transaction が既存の二つの transaction を選択するため，エッジ数 $E = 2N_t$ とすると，$O(E)$ と表される．こ

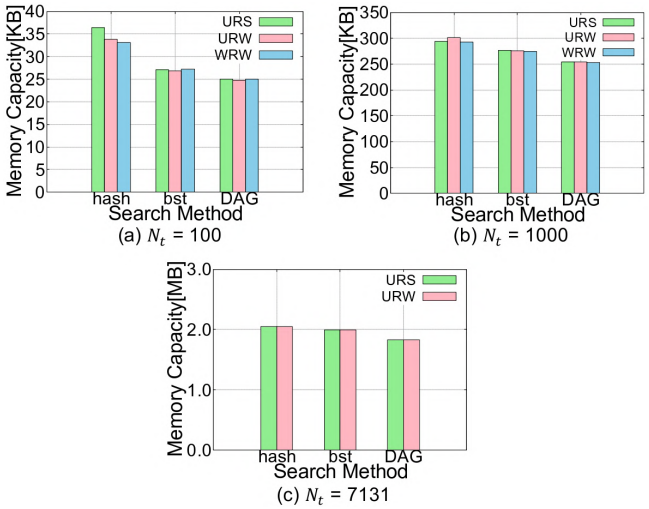

図 11　必要メモリ量

れらの計算量はコンテンツ名登録時，名前解決時ともに同じである．図 10 に，DAG の transaction 数 N_t に対するこれら平均計算量を示す．ここで，ハッシュチェイン法は $B = 10, 100, 1000$ の三通りで，幅優先探索および深さ優先探索を DAG としてプロットする． ハッシュチェイン法では，B が増加するほど一つのバケットで管理されるデータ数が減少するため，計算量が少なくなる．本稿で行った $B = 100$ におけるハッシュチェイン法と二分探索木を比較すると，$N_t < 1,000$ ではハッシュチェイン法の方が計算量が少なく，$N_t \geq 1,000$ では二分探索木の方が少なくなる．実際に，$N_t = 100$ の図 3 や図 6 において，検索時間はハッシュチェイン法の方が二分探索木よりも小さく，N_t が増加するに従って，二つの検索時間の差が縮まることが確認できる．また，DAG では，transaction 数に依存せず最も計算量が多く，シミュレーションでも同様の結果となった．

5.4　必要メモリ量

図 11 に，探索手法ごとの必要メモリ量をプロットする．(a), (b), (c) のいずれの場合でも必要メモリ量は，ハッシュチェイン法 > 二分探索木 >DAG となる．ハッシュチェイン法ではハッシュテーブルで最大の容量を持つバケットの容量が全てのバケットで確保されるため，大量のメモリを必要とする．また，未使用メモリが存在するバケットもあるため，メモリ効率は悪い．二分探索木では，transaction 数分のノードの容量を必要とする．そのため，余分なメモリがなく，ハッシュチェイン法よりも必要メモリ量は抑えられる．DAG では，他のテーブルなどで管理する必要がないため，最も必要メモリ量が少ない．また，transaction 数が増加するに従ってハッシュチェイン法と二分探索木の差が縮まることが確認できる．これはハッシュチェイン法では生成データ数にかかわらず，ハッシュテーブルで固定的に大量のメモリが必要となるためである．

以上の結果から，検索時間と必要メモリ量のトレードオフの関係を確認した．

6. ま と め

要求されたコンテンツの名称をもとにパケットを転送し，ルータにてコンテンツをキャッシュしコンテンツ要求者 (consumer) に配信する ICN が，次世代のネットワークとして注目を集めている． ICN では誰もが publisher としてコンテンツをアップロード可能だが，正当な publisher を騙る攻撃者が，実在するコンテンツ名で fake コンテンツをアップロードすることでキャッシュの機能を低下させる CPA の問題が指摘されている．CPA の対策として，consumer が公開鍵暗号を用いたディジタル署名によりコンテンツの正当性を判断可能である．しかし，公開鍵を管理する認証局の職員が攻撃者と結託し，攻撃者の公開鍵と書き換えることにより，実在する高人気コンテンツを騙る fake コンテンツをキャッシュに注入する詐称 fake 型 CPA は対策が困難である．そこで本稿では，publisher によるコンテンツのアップロードに際し，登録データの改ざんが困難な分散型台帳技術の一つである IOTA でコンテンツ名を管理し，詐称 fake 型 CPA を未然に防ぐ方式を提案した．提案方式では，台帳内でコンテンツ名を検索する四つの探索手法の検索時間と，必要メモリ量の比較を行い，シミュレーション評価により以下のことを確認した．

- 提案方式でコンテンツ名を探索するタイミングは，publisher によるコンテンツ名登録時と，consumer による名前解決時の二つである．いずれの場合においても，検索時間はハッシュチェイン法が最も短く，二分探索木，深さ優先探索，幅優先探索の順に長くなる．また，transaction 数が増加するほどそれぞれの探索手法の検索時間は長くなるが，DAG の形状に影響を受ける深さ優先探索，幅優先探索では特に長くなる．
- 必要メモリ量は，ハッシュチェイン法が最も多く，二分探索木，DAG の順に少なくなる．DAG では，他のテーブルなどで管理する必要がないため，最も必要メモリ量が少ない．一方，ハッシュチェイン法と二分探索木では DAG のメモリ量に加え，ハッシュテーブルや二分木でデータを管理するため，多くのメモリを必要とする．また，ハッシュチェイン法では，ハッシュテーブルで最大の容量を持つバケットの容量が全てのバケットで確保されるため，未使用メモリが多い．二分探索木ではデータ数分のメモリを確保するため，ハッシュチェイン法よりも必要メモリ量は抑えられる．

以上の結果から，検索時間を抑えることを重視する場合は検索時間の短いハッシュチェイン法や二分探索木を用い，メモリ量を抑えることを重視する場合は所要メモリ量の少ない深さ優先探索や幅優先探索を用いることが望ましいことが明らかとなった．

謝辞本研究成果は JSPS 科研費 18K11283 と 21H03437 の助成を受けたものである．ここに記して謝意を表す．

文　献

[1] T. Nguyen, et al., ”Content Poisoning in Named Data Networking: Comprehensive Characterization of real Deployment.”, IFIP/IEEE IM 2017.
[2] W. Cui, et al., “Feedback-Based Content Poisoning Mitigation in Named Data Networking”, IEEE ISCC 2018.
[3] P. Gasti, et al., ”DoS and DDoS in Named Data Networking”, IEEE ICCCN 2013.
[4] 工藤 多空飛, 上山 憲昭, ”ICN における Fake 型コンテンツポイズニング攻撃の影響分析”, 信学会 CQ 研究会, CQ2022-23, 2022 年 7 月.
[5] S. Popov, et al., Equilibria in the Tangle, Computers & Industrial Engineering, 136, pp.160-172, Oct. 2019.
[6] J. Park, et al., ”A Block-Free Distributed Ledger for P2P Energy Trading: Case with IOTA?”, CAiSE 2019, LNCS 11483, pp. 111-125, 2019.
[7] G. Bu, et al., ”G-IOTA: Fair and confidence aware tangle”, IEEE INFOCOM WKSHPS 2019.
[8] S. Ghaffaripour and A. Miri, ”Parasite Chain Attack Detection in the IOTA Network”, IEEE IWCMC 2022.
[9] 政木 秀也, グエン キエン, 関屋 大雄, ”マイクロペイメントによるネットワーク資源保証の提供”, 信学会 NS 研究会, NS2021-32, 2021 年 7 月.
[10] A. Tesei, et al., ”IOTA-VPKI: a DLT-based and Resource Efficient Vehicular Public Key Infrastructure”, IEEE VTC 2018.
[11] M. Zander. 2018. Python IOTA Tangle simulation. https://github.com/manuelzander/iota_simulation. (2022).
[12] Alexa webpage, https://www.alexa.com/siteinfo.