分散したコンピュータによる URI とコンテンツ発信証明手法の検討

永井 俊行　　坪川 宏

東京工科大学 バイオ・情報メディア研究科 コンピュータサイエンス専攻

1 背景と問題

インターネットが普及したことにより，誰でも世界規模の情報発信を行うことが可能になった．インターネットの特徴として，ある時にある情報が公開されていたという事実が残らない点がある．ディジタルデータは改竄や捏造が容易であるため，情報の取得者は，その出典とオリジナリティを主張することができない．既存のセキュリティ技術によりディジタル情報の存在性と完全性は証明が可能であるが，インターネットで公開していたという事実を証明する機能は持たない．

インターネットでは，情報の発信はいつでも内容の変更・中止を行うことができる．インターネットの Web 上で後を絶たない誹謗中傷や著作権侵害といった行為は，この発信した事実が曖昧という性質が助長している側面もある．また，インターネット上で公開された技術や意匠が，その公知日時が曖昧なことにより，後に特許制度などにより第三者の独占利用が認められてしまう可能性も考えられる．

これらの問題から，インターネットでコンテンツが公開されている事実を記録する手段が求められる．

2 関連手法

こうした問題を解決する方法として，国内ではウェブ魚拓[1]サービスがよく用いられている．利用者がサービスに対象の URL を入力すると，サーバが対象のコンテンツを取得し新たな URL を付与し，複製に誰でもアクセスできるようになる．この複製は恒久的であるため，発信された事実を記録することができる．Web ページの存在証明サービス[2]は，同様にサーバがコンテンツを取得するが，それにディジタルタイムスタンプを付与し，サーバに複製を置かず利用者に送信する点で異なる．

Study of method of publication proof of URI and content by distributed computer.
Toshiyuki NAGAI, Hiroshi TSUBOKAWA, Tokyo University of Technology

両者は，サービス側でコンテンツを取得することにより，サービスがそのコンテンツが公開されていたことを証明する．しかしこうした単一的な取得元に対して，情報の発信元は特別な情報発信を行うことができ，情報発信事実の秘匿や偽造が可能である．

3 分散したコンピュータによる発信証明手法

前述したように，単一的な取得元による情報発信証明は行えない．そこで，分散したコンピュータによる，多視点な情報発信証明手法が必要不可欠である．

以降，発信証明を行う利用者をクライアント，分散したコンピュータの各々を公証ノード，発信証明を行う対象のコンテンツを公開するサーバを対象ホストと呼ぶ．

図1に本研究で提案する分散したコンピュータによる発信証明手法の概要を示す．

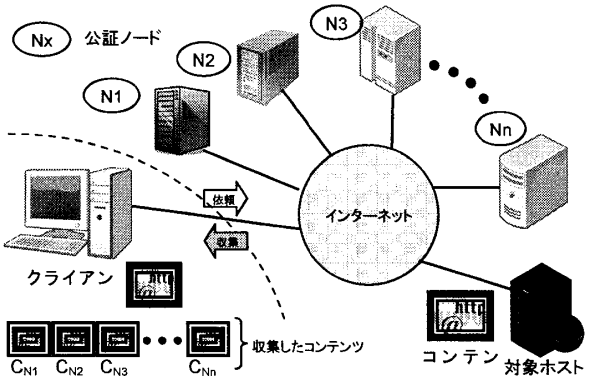

図1　情報発信証明手法の概要

発信証明のため，クライアントは公証ノード $N1 \cdots Nn$ に依頼を行う．公証ノードは対象ホストからコンテンツを取得し，クライアントはそのコンテンツ $C_{N1} \cdots C_{Nn}$ を得る．それらの比較を行うことで情報発信を証明する．この手法には，次の機能が要求される．

- $C_{N1} \cdots C_{Nn}$ が公証ノード $N1 \cdots Nn$ によって取得されたことを示せること
- 収集したコンテンツは比較可能な形であること

・ クライアントのアクセス権限を越えないようコンテンツを収集できること
・ 公証ノードの同時アクセスによる対象ホストへの過大負荷を制限できること

上述の要件を満たすための手法を次に述べる．

4 実現の検討

4．1 各公証ノードによる付加情報

分散したノード発信証明手法では，公証ノードの識別子，時間，URI の名前解決を行った様子，対象ホストへ行ったリクエストの様子などの情報が必要である．公証ノードが取得したコンテンツと分け，これらの情報を付加する必要がある．

また，コンテンツの取得や付加情報の作成に，そのノードが責任を持っていることを示すため，各公証ノードはディジタル署名を付与する必要である．

つまり，クライアントが収集する情報として，([コンテンツ，付加情報]，署名)が必要となる．

そのため，各ノードは秘密鍵 K_i を予め生成し保存している必要がある．

4．2 ブロックハッシュ化

公証ノードが取得したコンテンツは，そのままの状態で比較可能という要件を満たすが，クライアントにアクセス権がないコンテンツの送信を行ってしまう可能性がある．

そこで，クライアントは自身でもコンテンツを取得しておくこととすれば，公証ノードは配信するコンテンツを，それから算出されるハッシュ値で置換することが考えられる．クライアントは自身で取得したコンテンツからハッシュ値を算出し，公証ノードから収集したハッシュ値と比較することで一致を示せる．またハッシュの性質により元のコンテンツの復元はできず，二つの要件を満たす．

しかし，Web では広告領域など，コンテンツの一部がアクセス毎に変化することも珍しくない．そのため，次の方法が考えられる．

・ メディア(HTML・XML 等)に適した形でブロックに分割しNブロックの列とする
・ N ブロック列のそれぞれのハッシュ値を算出し置換する

この処理を行うと，コンテンツの部分ごとに比較が可能で，全体の一致度が算出できる．また，コンテンツを完全に復元することはできないため，両要件を満たすことができる．

4．3 アクセス過大の阻止

対象ホストへの過大な負荷を防ぐ方法として，各公証ノードが対象ホスト応答時間などで個別に判断する方法が考えられる．しかし，応答時間の低下を感知してから対策を行っていては，未然に負荷を防いでいるとはいえない．また応答時間などはネットワーク上の距離など状況により異なるため，判断に用いる閾値の設定が困難である．

次に，対象ホストごとに同時アクセス数を閾値により一律に制限する方法が考えられる．だが，公証ノードは分散しているため，ある対象ホストへの同時アクセス数を一元的に管理する役割を担うサーバが必要となる．

そこでこの役割を公証ノード同士で分散させることが考えられる．それには，公証ノードでオーバレイネットワーク(ON)を構成し，DHT のような機構を利用する方法がある．例えば，h(IP アドレス)で計算される値で管理ノードを決定し，そのノードが対象の IP アドレスへの同時アクセス上限を管理することとする．

各公証ノードは対象ホストにアクセスする前，管理ノードに申し出て許可を得ることとする．対象のホストにアクセスを行ったあとは，完了を管理ノードに伝える．

ON のトポロジーと実現・維持には様々な方法が提案されているが，頻繁なノードの出入りが無いことを想定すると，各ノードが全体の知識を有する One-hop DHT[3]のような形態が適していると思われる．

ON を構築することにより，公証ノード間の分散したメッセージング機構としても利用できるだろう．

5 まとめ

まず，現在のインターネットの特徴を示し，そこに潜在している問題を述べた．次に，既存の手法では，不十分な点があることを述べた．そして，分散した多数のコンピュータによる解決方法と，その実現方法について検討した．今回検討した方法は，公証ノードの追加や除去に柔軟であり，公証ノードは大学やコミュニティによって設置されることが想定される．

今後は，Web に対してのブロックハッシュ化の有効性の検証行った後，システムの実装を行い，負荷の計測などを行っていきたい．

参考文献

[1] ウェブ魚拓
http://megalodon.jp/

[2] Web ページの存在証明サービス
http://www.existingproof.jp/

[3] A. Gupta and et al. One hop lookups for peer-to-peer overlays. In Proc.HotOS 2003.