RL-006

HTTP 通信におけるプロクシ認証を利用した NAT ルータ配下の PC 識別手法

An Identification Method of PCs under NAT router with Proxy Authentication on HTTP Communication

石川 義基 †1 Yoshiki Ishikawa　岡山 聖彦 †2 Kiyohiko Okayama　山井 成良 †2 Nariyoshi Yamai　中村 素典 †3 Motonori Nakamura

1 まえがき

近年のインターネットの普及に伴って，IPv4 アドレスの枯渇が問題となっている．根本的な解決として，より大きなアドレス空間を持つ IPv6 アドレスへの移行が求められているが，既存の IPv4 機器の置き換えを要するためさほど進んでいないのが現状である．この問題の一時的な解決策の一つに，NAT(Network Address Translation)[1] がある．NAT は IP アドレスの変換技術であり，NAT 機能を有するルータ (以下，NAT ルータ) は，プライベートネットワーク内の PC から外部に向けて発信されたパケットの送信元アドレスを特定のアドレス (多くの場合は NAT ルータのグローバル IP アドレス) に変換して中継する．プライベートネットワーク内の PC は NAT ルータのグローバル IP アドレスを共用できるため，グローバル IP アドレスを節減できる．さらに，NAT ルータの上位ネットワークからはプライベートネットワーク内の PC に直接アクセスできないため，プライベートネットワーク内の PC を外部から護る手法としても広く用いられている．

しかし，組織の基幹ネットワークなどの上位ネットワークで LAN アクセス制御サーバを導入し，送信元 IP アドレスあるいは MAC アドレスに基づくアクセス制御を行っている場合，下位ネットワークに NAT ルータがあると問題が生じる．NAT ルータはプライベートネットワーク内の PC から受け取ったパケットを外部へ転送する際，送信元 IP アドレスおよび MAC アドレスを NAT ルータのものに変換するため，プライベートネットワーク内のある PC が認証に成功すると，LAN アクセス制御サーバは NAT ルータの IP アドレス，または MAC アドレスをアクセス許可リストに登録する．そのため，プライベートネットワーク内の他の PC が外部にアクセスする際，LAN アクセス制御サーバの認証を経ずにアクセスすることが可能になってしまう．

これに対し，我々の研究グループでは，MAC アドレス中継型 NAT ルータ [2] を提案している．MAC アドレス中継型 NAT ルータは，プライベートネットワーク内の PC から送信されたパケットに含まれる送信元 MAC アドレスをそのまま外部へ中継する．このため，LAN アクセス制御サーバは，MAC アドレスに基づいたアクセス制御により，MAC アドレス中継型 NAT ルータ配下の PC を識別可能である．しかし，既設の NAT ルータをすべて MAC アドレス中継型 NAT ルータに置き換える必要があるので，その台数に比例して置き換えのためのコストが大きくなるという問題がある．

そこで本研究では，インターネットの主要なサービスである Web に注目し，プロクシ認証を利用した NAT ルータ配下の PC 識別手法を提案する．従来のプロクシ認証では HTTP ヘッダ内の Proxy-Authentication(以下，認証ヘッダ) の有無によって認証済みであるかの確認が行われ，認証処理を NAT ルータよりも上位にあるプロクシサーバに集約することができるため，上述した問題が生じることなくアクセスを識別することができる．しかし，従来のプロクシ認証の対象はユーザであるため，そのままでは同一ユーザが複数の PC を同時に利用する際に PC を識別することができない．そこで本研究では，認証ヘッダ内の realm と呼ばれる認証の適用範囲を示すフィールドを利用し，これを PC に対して一意になるように設定することで PC の識別を可能にした．この機能を実装した試作システムを作成して，実験ネットワークを構築し，複数の PC を接続して動作確認実験を行うことにより，NAT ルータの配下の複数の PC を正しく識別できることを確認した．さらに，PC 識別を実現可能なもう一つの手法であるクッキー認証を実装して性能評価実験を行うことにより，本研究で提案するプロクシ認証がクッキー認証よりも優れていることも確認している．

2 NAT とその問題点

2.1 NAT の概要

NAT とは，プライベートネットワーク内から外部に接続する際に，プライベート IP アドレスをグローバル IP アドレスに変換する機能のことである．NAT の中でも IP アドレスとポート番号の変換を行うものを NAPT(Network Address Port Translation)，または IP マスカレードと呼ぶ．NAPT は，1 つの IP アドレスを同時に共用することができグローバル IP アドレスを節減できるため，一般的によく用いられている．

一方，変換前アドレスと変換後アドレスの対応情報を持つアドレス変換表にあらかじめ変換アドレスの組を登録することにより変換を行う NAT は静的 NAT と呼ばれる．静的 NAT を用いることでプライベート IP アドレスとグローバル IP アドレスの対応情報が固定されるため，外部からプライベートネットワーク内の PC にアクセスすることが可能となる．これに対し，外部へのアクセス時に，動的にアドレス変換の登録を行う NAT は動的 NAT と呼ばれ，セッション終了と同時にその登録の解除を行う．そのため，アドレスが固定されず，外部からプライベートネットワーク内へアクセスすることができないが，セキュリティ対策や IP アドレスの節減に役立つ．以下，本論文では NAT/NAPT を総称して NAT と呼ぶことにし，動的 NAT を対象と

†1 岡山大学大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University
†2 岡山大学情報統括センター, Center for Information Technology and Management, Okayama University
†3 国立情報学研究所, National Institute of Informatics

する．

プライベートネットワーク内の PC が NAT ルータを介して外部へアクセスし，外部の PC から応答を受け取るまでの流れを図 1 に示す．まず，PC1 からパケットが送信され，NAT ルータがこのパケットを受信すると，NAT ルータは送信元 IP アドレス (192.168.0.2) とポート番号 (1024) をアドレス変換表に従って，割り当てられた IP アドレス (10.0.0.3) とポート番号 (5000) に書き換える．この際，アドレス変換表に対象 PC の情報が登録されていなければ，重複の無いようにアドレスおよびポート番号の登録を行う．その後，外部の PC2 へパケットを送信する．逆方向，すなわち，PC2 からの応答パケットを NAT ルータが受け取ったとき，NAT ルータは宛先ポート番号 (5000) を基にアドレス変換表を参照し，割り当てられた宛先 IP アドレス (192.168.0.2) とポート番号 (1024) に書き換え，パケットをプライベートネットワーク内の PC1 へ送信する．このようにアドレス変換を行うことにより，プライベートネットワーク内の PC から NAT ルータを介して外部との通信が可能となる．

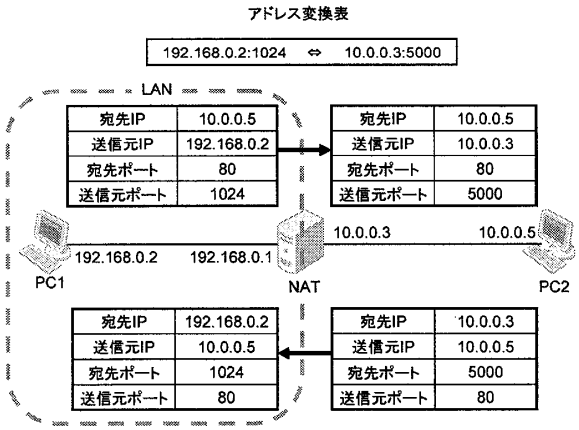

図 1: NAT ルータの動作

2.2 NAT の問題点

NAT ルータがプライベートネットワーク内の PC から外部へ向かうパケットを中継する際の，ヘッダ内の MAC アドレスと IP アドレスの変更の流れを図 2 に示す．

NAT ルータはプライベートネットワーク内の PC から受け取ったパケットを外部へ送信する際，自身のネットワークインタフェースを使用するため，送信されるパケットに付与されるイーサネットヘッダの送信元 MAC アドレスは NAT ルータの外向きインタフェースの MAC アドレス（MAC_{NATOUT}）になる．PC1 と PC2 のアドレス変換前のヘッダ情報には，それぞれ自身に関する情報があるが，アドレス変換後に NAT ルータが送信するものには，PC1 や PC2 に関する情報が一切なく，NAT ルータ自身が送信するパケットとして全く同じものとなってしまう．結果的に，NAT ルータは LAN 外に対してプライベートネットワーク内の PC に関する情報を隠すことになる．

このため，組織の基幹ネットワークなどの上位ネットワークで LAN アクセス制御サーバを導入し，IP アドレスあるいは MAC アドレスに基づくアクセス制御を行っている場合，以下のような問題が生じる．ここで述べる LAN アクセス制御サーバとは PC のネットワーク接続時に利用者認証を行い，認証に成功した PC のみに対して外部とのアクセスを許可するサーバのことであり，ファイアウォール機能を動的に利用してアクセス制御を行う．

例えば，図 2 の PC3 が LAN アクセス制御サーバであったとする．プライベートネットワーク内の PC1 が外部にアクセスしようとすると，アクセス制御サーバで認証が行われ，認証に成功すれば通常は PC1 の IP アドレスあるいは MAC アドレスが LAN アクセス制御サーバのアクセス許可リストに登録されるが，PC1 と LAN アクセス制御サーバの間に NAT ルータが介在しているため，実際に登録されるのは NAT ルータのグローバル IP アドレスあるいは MAC アドレスである．さらに，PC2 が外部へのアクセスを試みると，PC2 から送信されるパケットの送信元 IP アドレスと MAC アドレスは共に NAT ルータのものに変換されるため，LAN アクセス制御サーバは許可済であると見做して認証することなくパケットを通過させてしまう．この問題は，LAN アクセス制御機能を持つ NAT ルータなどを利用し，NAT ルータの部分で LAN アクセス制御を行うことで回避できると考えられる．しかし，上位ネットワークにおいて組織内にある全ての PC のアクセスを集中的に管理するような場合，大規模なネットワークでは全ての NAT ルータに対して，認証設定やアクセスログなどの管理を分散することになり好ましくない．また，NAT ルータが基幹ネットワークの管理下に無い，すなわち，NAT ルータを利用者が独自に設置した場合には，基幹ネットワーク側で NAT ルータの存在を知ることは非常に困難であり，認証の抜け穴ができることによって組織ネットワークのセキュリティを低下させる恐れがある．

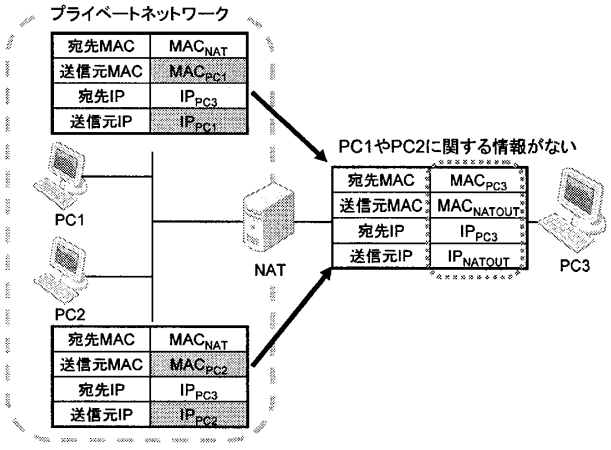

図 2: NAT ルータの問題点

3 プロクシ認証を利用した PC 識別手法

3.1 実現方針

前章で述べた問題を解決するため，我々の研究グループでは MAC アドレス中継型 NAT ルータを提案している．MAC アドレス中継型 NAT ルータは，配下の PC から送出されたパケットの送信元 MAC アドレスを，自身が上位ネットワークに送出するパケットの送信元 MAC アドレスにそのまま埋め込む．このため，上位ネットワークの LAN アクセス制御サーバは，MAC アドレスに基づいたアクセス制御を行うことにより，MAC アドレス中継型 NAT ルータ配下の PC を識別することができる．しかし，この方法ではすでに設置されている NAT ルータをすべて MAC アドレス中継型 NAT ルータに置き換える必要があるので，その台数に比例して置き換えのためのコストが大きくなるという問題がある．そこで本研究では，IP アドレスや MAC アドレスを用いず，アプリケーション層で PC を識別する方法を提案する．これを実現する方法として，プロクシ認証やプロクシサーバでクッキー認証を利用する方法などがある．これらの方法は Web サービス (HTTP 通信) にしか適用することができないが，現在のインターネットでは，Web サービスは電子メールとともに必要不可欠なサービスであるため，これを利用した認証はユーザに対して十分な強制力を持つと考えられる．以下にプロクシ認証と，クッキー認証について簡単に述べる．

プロクシサーバとは，内部ネットワークとインターネットとの間に設置し，内部ネットワークの PC に代わり，インターネットとの接続を行うサーバである．プロクシサーバで行われる認証はプロクシ認証と呼ばれ，PC はプロクシ認証に成功しなければインターネットにアクセスすることができない．内部ネットワークの PC がインターネットに接続する際には必ずプロクシサーバを経由するため，外部への HTTP アクセスを一括して監視・制御することができる．そのため，学校や会社のネットワークなど広く使われている．プロクシ認証の流れを以下に示す．

1. クライアント PC がプロクシサーバにアクセスする．
2. プロクシサーバは認証ヘッダの有無を調べ，無い場合は認証要求を返す．
3. ユーザは，ユーザ名とパスワードを入力し，プロクシサーバに送信する．
4. プロクシサーバは入力されたユーザ名とパスワードを確認し，正しいものであればインターネットへとアクセスする．

認証済みであるかの確認は認証ヘッダの有無によって決定される．そのため，IP アドレスや MAC アドレスに関係なくクライアント PC を識別することができる．

一方，Web サーバが，Web ブラウザを通じてクライアント PC に一時的にデータを書き込み，保存させる仕組みをクッキー [3] という．クッキー認証はこのクッキーを利用する認証方式である．プロクシサーバでクッキー認証を利用する方法とは，クッキー認証をプロクシサーバがオリジンサーバになりすまして行うものである．以下に通常のクッキー認証の流れを以下に示す．

1. クライアント PC が任意の Web サーバ (以下，オリジンサーバ) にアクセスする．
2. 何らかの認証を行い，完了した際に，オリジンサーバは認証済みであることを示すクッキー (以下，認証クッキー) をクライアント PC に保存させる．
3. PC は，このオリジンサーバにアクセスする際にページ取得のリクエストと共に認証クッキーを送信する．
4. オリジンサーバは，認証クッキーの有無の確認を行い，手順 2 で発行したクッキーが存在する場合はページを返す．

クッキーはクライアント PC の Web ブラウザごとに保存されるため，NAT 配下の PC に対しても識別が可能である．しかし，クライアント PC はクッキーを発行したオリジンサーバと通信する際にしかクッキーを送信しない．そのため，プロクシサーバがオリジンサーバになりすましてクッキー認証を行う場合，別のオリジンサーバと通信する度に認証を行い，新たなクッキーを保存させる必要がある．また，SSL 通信では，プロクシサーバは通信をトンネリングするのみなので，先ほど述べたオリジンサーバになりすましてクッキーを保存させることが困難である．

以上のことから，本研究では実装が容易なプロクシ認証をベースにした PC 識別手法を提案する．以下，プロクシサーバにおける PC 識別機能ついて述べた後，システムの動作手順について述べる．

3.2 プロクシにおける PC 識別機能

前節で述べたように，プロクシ認証では認証ヘッダの有無によって認証済みであるかの確認が行われる．しかし，認証完了後，PC は認証ヘッダを付けて通信を行うため，識別方法がユーザ名しかない．そのため，従来のプロクシ認証をそのまま適用すると，同一ユーザが複数の PC を同時に利用する際に，これらの PC を区別できないという問題が生じる．そこで提案手法では，realm と呼ばれる認証の適用範囲を示すためのフィールドを利用する．realm はプロクシサーバが送信する認証要求に含まれるもので，認証ヘッダの要素の一つとして使用される．この realm に，Web ブラウザがプロクシサーバに最初にアクセスした時間とランダムな文字列を埋め込むようにする．これにより，認証ヘッダ内の realm は PC に対して一意なものとなり，認証完了後でも PC の識別が可能となる．

プロクシ認証において，PC は認証ヘッダの情報により識別される．しかし，認証ヘッダには IP アドレスや MAC アドレスといった PC 固有の情報が無いため，PC を識別することはできても，識別した PC を特定することができない．そこで提案手法では，認証完了後に MAC アドレスを収集し realm と関連付けることで

PC を特定することにする．MAC アドレスを収集する方法としては，Java アプレットで作成したエージェントを使用することにする．Java アプレットを実行するためには，PC に Java ランタイム環境 (JRE) が必要である．しかし，現在では多くの Web サイトで Java の利用が一般的となっているため，JRE のインストールは運用上大きな問題とはならないと考えられる．

上述した方法は，Web ブラウザが常に認証ヘッダを付加してアクセスすることを想定している．これに対し，Internet Explorer 系の Web ブラウザでは，認証完了後に新規ウィンドウやタブを開くと，プロクシサーバが認証要求を行うまでは認証ヘッダを付加しないことが判明した．プロクシサーバは認証要求を返す度に異なる realm の値を生成するので，先の認証完了時に Web ブラウザが保存した値との食い違いが生じて，再度認証を行わなければならなくなる．この問題を回避するため，提案手法では realm の値をクッキーに保存することとした．Web ブラウザはプロクシサーバにアクセスする際には常にクッキーを付加するので，Web ブラウザのアクセス時に認証ヘッダがなくても，プロクシサーバはクッキーに含まれる realm の値を用いて認証要求を送信すれば，認証を繰り返すことはなくなる．なお，クッキーを保存させるのはオリジンサーバではなくプロクシサーバであるため，前節で述べた欠点は生じない．

3.3 動作手順

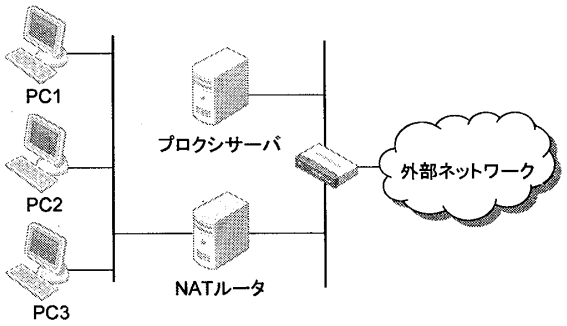

図 3: システム構成

図 3 のようなネットワーク構成において，未認証 PC がネットワークに接続する場合の動作手順を図 4 に示す．

プライベートネットワーク内の未認証 PC が外部の Web サイトにアクセスする際，アクセスが許可されるまでのプロクシサーバにおける処理手順は以下のようになる．なお，括弧内は図 4 のアルファベットに対応している．

1. クライアント PC がオリジンサーバにアクセスする．(a)
2. プロクシサーバは認証ヘッダの有無を確認し，認証ヘッダがないため認証要求を返す．(b)
3. ユーザはユーザ名とパスワードを入力し，プロクシサーバに送信する．(c)

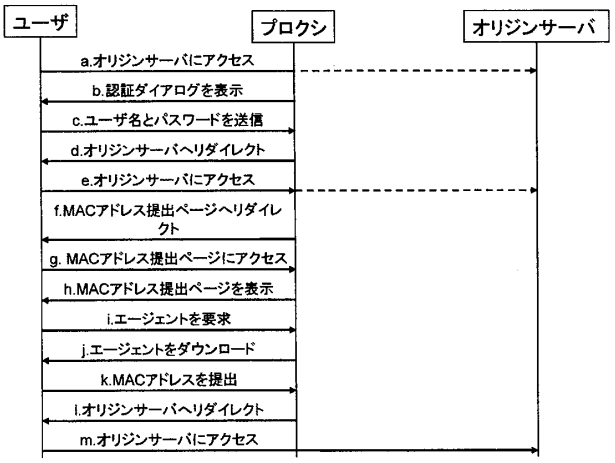

図 4: 動作手順

4. プロクシサーバは，入力されたユーザ情報が正しければ，オリジンサーバにリダイレクトする．(d)
5. クライアント PC は，プロクシサーバからリダイレクト要求を受け取り，オリジンサーバにアクセスする．(e)
6. プロクシサーバは realm と MAC アドレスとの関連付けを行っているかの確認を行い，Java アプレットへのリンクを含む MAC アドレス提出ページへリダイレクトする．(f)
7. クライアント PC はリダイレクト要求を受け取り，MAC アドレス提出ページにアクセスする．(g)
8. クライアント PC は MAC アドレス提出ページを受信し，ブラウザはリンクにしたがってプロクシサーバ内の Java アプレットを自動的にダウンロードし，実行する．(h，i，j)
9. クライアント PC が Java アプレットに含まれる MAC アドレス提出ボタンをクリックすると，抽出された MAC アドレスがプロクシサーバに送信される．(k)
10. プロクシサーバは MAC アドレスを受信後，オリジンサーバにリダイレクトする．(l)
11. クライアント PC はオリジンサーバにアクセスし，ページを取得する．(m)

認証を終えた Web ブラウザを立ち上げている状態であれば，新しいウィンドウやタブを開いたとしても認証は必要ないが，一度 Web ブラウザを閉じ再接続する場合は，再度認証を行う必要がある．

4 試作システムの実装と評価

4.1 試作システムの実装

本実装では，代表的な Web サーバである Apache2[6] の標準的なモジュールである mod_proxy[7] を使用した．実装した試作システムの内部構造を図 5 に示す．本実装は以下の 2 つの機能によって構成されている．

- realm 管理機能

  realm の値を決定したり現在状態 (詳細は後述) を更新したりするなど，realm の状態を制御する．また，realm と MAC アドレスの関連付けや，3.2 節で述べた新しいウィンドウやタブの作成への対策であるクッキーの有無の確認も行う．本機能は CGI として PHP 言語により作成した．

- アクセス先 URL 書き換え機能

  Apache の mod_rewrite[8] を使用してアクセス先 URL の書き換えを行う．mod_rewrite とは，クライアント PC が送信してきた URL をサーバ側で変更して処理するためのモジュールである．ルールを定義することができ，このルールに一致した際に URL の書き換えが行われる．また，mod_rewrite の構成要素でもある RewriteMap を使用することで，プロクシサーバの処理途中に外部のプログラムを使用することができる．この RewriteMap を利用することで，realm の現在状態に基づいてアクセス先 URL を書き換えることができる．外部のプログラムは PHP を使用しており，mod_rewrite にライブラリとして使用される．

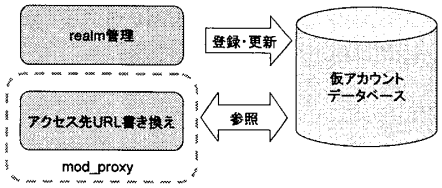

図 5: プロクシサーバの内部構成

本実装では，realm を使用しているブラウザが現在どこまで処理を終えているかを，realm の現在の状態として登録している．realm の状態としては，プロクシ認証前，プロクシ認証完了後，MAC アドレス提出完了後の 3 つを設けている．アクセス先 URL の書き換えは、これらの状態に基づいて行う．

4.2 動作確認実験

試作システムの動作を確認するため，図 3 と同様の実験環境を構築し，以下の実験を行った．

- PC 識別実験

  プライベートネットワーク内の PC を NAT ルータの外部から識別できるか確認するため，複数の PC を Web アクセスさせた．

- Web ブラウザ対応実験

  試作システムを介して外部のネットワークに接続した際の動作に不具合がないを確認するため，表 1 に示す Web ブラウザで動作確認を行った．

上記の実験を行った結果，PC 識別実験では，プライベートネットワーク内のある PC が認証を完了した後に，別の PC を外部にアクセスさせようとすると，認証が行われた．これは，外部にあるプロクシサーバが NAT ルータ配下の複数の PC を正しく識別していることを意味する．

表 1: 動作確認を行った Web ブラウザ

Web ブラウザ名	バージョン
Internet Explorer	7.0，8.0
FireFox	3.5
GoogleChrome	3.0
Opera	10.10
Netscape Navigator	9.0
Lunascape6	6.0

次に，Web ブラウザ対応実験では，実験に使用した各 Web ブラウザで不具合が無いことを確認し，同一ブラウザで新規ウィンドウやタブを作成した場合に再度認証が行われないことを確認した．

4.3 性能評価実験

試作システムの性能評価を行うために，3.1 節で述べたプロクシサーバにおけるクッキー認証との比較を行った．この比較を行うにあたり，プロクシサーバにおけるクッキー認証を実装した．これは以下の 2 つの機能によって構成されている．

- アクセス先 URL 書き換え機能

  mod_rewrite の構成要素である RewriteMap を使用して認証クッキーの有無を確認する．認証クッキーが無い場合は認証クッキーを発行するページへにリダイレクトさせ，有る場合はオリジンサーバへアクセスする．

- 認証クッキーの発行機能

  プロクシサーバがオリジンサーバになりすまして認証クッキーの発行を行う．

3.1 節で述べたとおり，プロクシサーバはオリジンサーバにりすまして認証クッキーを発行し，クライアント PC が認証クッキーを発行したオリジンサーバとは別のオリジンサーバにアクセスした場合，新たに認証クッキーを発行するために認証が必要となる．そのため，試作システムよりもアクセス時間が増大すると予想される．本実験を行うため，図 6 のような実験環境を用意した．ホスト 1 台ではアクセス時間が短すぎ，時間の計測がしにくいため，本実装においてのリダイレクト回数の限界である 7 つのホストを準備し，これらをオリジンサーバとして機能させる．そして，認証完了後にプライベートネットワーク内の PC からホスト 1 にアクセスする．ホスト 1 からホスト 2，ホスト 3 と順にリダイレクトさせていき，最後のホスト 7 のページの読み込みが完了するまでの時間を測定した．この測定法で 100 回アクセスしたときの平均アクセス時間を求めた．なお，アクセス時間とは，クライアント PC がホスト 1 にリクエストを送信してからホスト 7 のページの読み込みが完了するまでの時間を意味する．

実験結果を表 2 に示す．プロクシサーバでクッキー認証を行った場合，新たなホスト (オリジンサーバ) に

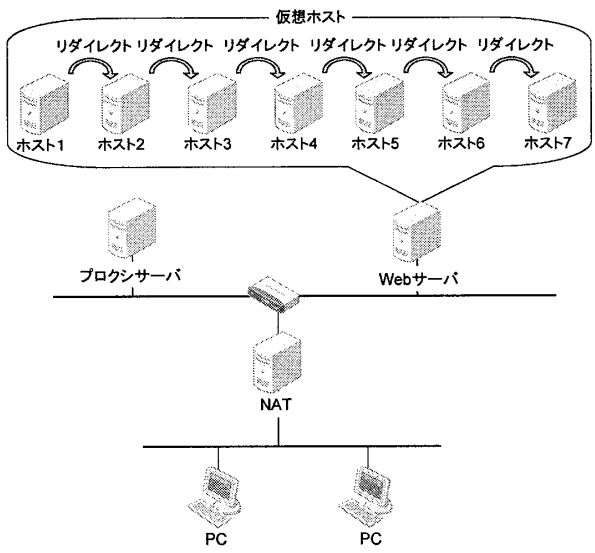

図 6: 性能評価実験環境

アクセスするたびに認証を行うため，提案手法のアクセス時間はクッキー認証をベースにした手法の約半分の時間となった．今回の実験では，アクセス元となるクライアント PC は 1 台であったが，複数のクライアント PC が同時にアクセスするような場合，クッキー認証を行うプロクシサーバの負荷は試作システムのプロクシサーバよりも大きくなることが予想されるため，平均アクセス時間の差も大きくなると考えられる．

表 2: 性能評価実験結果

PC 識別手法	平均アクセス時間 (ms)
試作システム	172.73
クッキー認証	343.86

5 あとがき

本論文では，プライベートネットワーク内の PC を外部ネットワークから識別するために，プロクシ認証を利用した NAT ルータ配下の PC 識別手法を提案した．そして，試作システムを実装して動作確認を行い，NAT ルータ配下の PC を識別できることを確認するとともに，クッキー認証と比較した性能評価実験も行い，提案手法のアクセス時間がクッキー認証をベースにした手法よりも小さいことを確認した．

今後の課題として，今回の実験の対象外となった Web ブラウザや他 OS での動作確認実験を行う予定である．

謝辞

本研究の一部は平成 21〜23 年度科学研究費補助金 (基盤研究 (C)，課題番号 21500075) の補助を受けている．ここに記して感謝の意を表する．

参考文献

[1] P. Srisuresh, K. Egevang: “Traditional IP Network Address Translator (Traditional NAT),” RFC3022, 2001.

[2] 村上亮, 岡山聖彦, 山井成良: “LAN 内 PC を外部から識別するための MAC アドレス中継型 NAT ルータ,” インターネットと運用技術シンポジウム 2009 論文集, pp.95-100(2009-12).

[3] D. Kristol, L. Montulli: “HTTP State Management Mechanism,” RFC2965, 2000.

[4] Internet2 Intellectual Property Framework: “Shibboleth,” http://shibboleth.internet2.edu/.

[5] 国立情報学研究所 UPKI: “学術認証フェデレーション,” https://upki-portal.nii.ac.jp/docs/fed.

[6] Apache Software Foundation: “The apache web server,” http://www.apache.org.

[7] Apache Software Foundation: “Apache Module mod_proxy,” http://httpd.apache.org/docs/2.2/en/mod/mod_proxy.html.

[8] Apache Software Foundation: “Apache Module mod_rewrite,” http://httpd.apache.org/docs/2.2/ja/mod/mod_rewrite.html.

[9] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart: “HTTP Authentication: Basic and Digest Access Authentication,” RFC2617, 1999.