社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DPS－126（43）
2006－CSEC－32（43）
2006／3／17

権限証明書を用いた名前解決のアクセス制御方式とその評価

神山　剛　　中山　雅哉

東京大学大学院　新領域創成科学研究科　基盤情報学専攻

DNS（Domain Name System）の名前解決という機能を利用することで、誰でもサービス提供元ホストを発見することができる。近年のインターネットには、誰でもが利用可能ではなく、個人や組織内などクローズドな利用目的でのサービスも多く存在する。このようなサービスは、無断利用だけでなく、発見されること自体も望ましくない。DNS の利用自体が元々オープンなものであるから、扱われる個別のリソース自体を守ろうとする仕組みはあまり議論されていない。本稿では、DNS で扱われる特定のリソースに対しての名前解決の際、許可されたユーザが権限証明書を提示することで、ユーザとそのアクセス権限を識別し、アクセス制御を行う仕組みを提案する。

Access Control Mechanism for Name Resolution using Authorization Certificate

Takeshi Kamiyama and Masaya Nakayama

Dept. of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo

Using DNS Name Resolution, everyone can look up the host with desirable services. In modern Internet, there are many services not intended for everyone but only to limited users. It is undesirable for these services to be not only used but also looked up by unknown users. However, because DNS usage is unrestricted, protecting each resource explicitly is not well discussed. In this paper, we propose an access control mechanism for name resolution in DNS that identify user and access right by requiring Authorization Certificate to be shown.

1.　はじめに

近年、計算機資源やネットワーク関連技術の著しい発展により、個人レベルで PC が普及し、広帯域なネットワーク接続環境が整備されてきた。特にインターネットは、我々の生活・仕事を支える重要な社会基盤のひとつとして、IP ネットワーク上において多種多様なサービスが提供されることが期待されている。

従来のインターネットにおいて、Web とメールなどは提供される主要なサービスであった。これらはインターネット上のユーザに広く公開されるサービスであり、ユーザはこれらを利用する際には、サービスを提供するホストを発見するために DNS（Domain Name System）[1][2]を利用している。

DNS は古くからインターネットを支えるシステムであり、ホスト名と IP アドレスの対応付けを管理し、名前の問い合わせに IP アドレスを回答する、名前解決という機能を提供する。サービスを利用するという観点から、DNS はサービス発見のためのシステムであるということができる。

一方、近年では広く公開されることが前提のインターネットとは異なる利用形態のサービスが多く提案されている。個人の宅内における情報家電・機器を相互接続するホームネットワークや、企業などの組織内におけるグループウェアにおけるサービスが例に挙げられる。これらの環境におけるサービスは、個人から組織レベルと比較的閉じた環境におけるユーザを対象にしていることから、ここでのサービス発見は Jini[3]や UPnP[4]などのミドルウェア、もしくはアプリケーション固有の発見方法が用いられている。

しかし、IPv6 化を含め、これらのローカルな環境におけるサービスを外部から利用するという利用形態においては、両方式とも OS などの環境に依存してしまうため適さない。そこで、OS 環境に依存せず、異なるサービスも統一的に扱うことができる DNS を用いる方式について本稿では取り扱うことにする。

しかし、DNS で取り扱われるリソースは公開されることが前提となっており、誰でも自由に問合わせを行い、名前解決結果を得ることができる。しかし、例えばホームネットワーク内のように、プライベートな利用目的のリソースが取り扱われることになると、名前解決による発見自体が、第三者による「のぞき行為」になってしまい、そこから不正利用など被害が拡大することになると予想される。つまり、DNS には、ホームネットワーク内のリソースの存在を第三者からの問い合わせに対して答えない様なアクセス制御機構を持ち合わせていない。

これまで、特定のリソースに対する名前解決を、ユーザ単位で識別し、認証を行う方式として、PKI の仕組みを用いたアクセス制御方式が提案されている[5]が、DNS のアクセスの特徴であるキャッシュをうまく

活用できない問題点があった。そこで、本稿では、権限証明書を用いた機構を提案し、その有効性について示す。

2. 関連研究

2.1. 概要

DNSにおける名前解決をユーザ単位で認証する方式として、ユーザが名前解決を行う際、問い合わせメッセージにユーザの秘密鍵による署名を付与し、メッセージを受け取ったネームサーバがその署名を検証することでユーザを識別する方式が提案されている。[5]この際の DNS メッセージの署名には、DNS ダイナミックアップデートなどで送信元認証のために定義されている SIG(0)[6]が用いられている。

また、特定のリソースに対し、どのユーザを許可するか、つまりアクセス制御リスト(ACL)が必要である。これは、図1 に示すように、リソースレコードを管理するネームサーバにおける Zone データ内に、対象となるリソースのレコードに続いて、名前解決許可を与えるユーザ分だけ制御レコードを記載している。この ACL は、各ユーザの署名者 ID を含むものであり、各ユーザの公開鍵に対応付けられた ID でもある。署名付き問い合わせメッセージを受け取ったネームサーバは、この ID を基にユーザの公開鍵を取得し、メッセージの SIG(0)署名を検証することになる。

図1 Zone データ内の ACL の記述

```
www     IN  A      192.168.1.20
        IN  TXT    "ACL A kami._user.test.com."
        IN  TXT    "ACL A alice._user.test.com."
        IN  TXT    "ACL A bob._user.test.com."
```

2.2. 考察

前項で述べた既存方式について、考察を述べる。

まず、この方式における問題点は、DNS キャッシュを有効に活用できないことであると考える。

通常の DNS の動作においては、ユーザが利用しているリゾルバ(キャッシュサーバ)が、ユーザからの問い合わせに対する DNS サーバからの回答から回答リソースレコードをキャッシュし、次回以降同じリソースレコードへの問い合わせにはキャッシュ参照による高速な回答を行うことができる利点を有するが、この方式でリゾルバがキャッシュによるアクセス制御を処理するためには([5]では、アクセス制御付き名前解決の際はキャッシュサーバを利用することなく、権威ネームサーバが直接問い合わせ処理している)、対象リソースレコードに加え、ACL となるアクセス制御レコードもキャッシュしなければならない。

しかし、ACL は許可するユーザ毎に制御レコードが必要になるため、仮にリゾルバが 1 ユーザ分の制御レコードをキャッシュしていても、他のユーザからの問い合わせを処理することができないため、該当するユーザの制御レコードがなければ、権威ネームサーバへの問い合わせが発生することになってしまう。

多くのユーザからの問い合わせをキャッシュでまかなうために、あらかじめ制御レコードをまとめてキャッシュする手段をとることができるが、「誰がいつ」というアクセス頻度次第で、大きなキャッシュサイズを消費した割には参照回数が少なく非効率な場合も考えられる。

加え、アクセス制御を処理するためには、ユーザからの問い合わせメッセージに付加された署名を検証することになるが、これに必要なユーザ公開鍵の取得すること、さらに取得した公開鍵自体が正当なものであるか確認しなければならない。通常、公開鍵基盤における公開鍵の正当性の確認には、広く信用できる主体による鍵への署名を検証するなどの方法が用いられる。例えば、DNSSEC (DNS Security Extension)[7]における Zone 公開鍵のように、上位 Zone からの信用の連鎖を確認するステップが必要になる。いずれにせよ、ユーザ公開鍵の正当性は、それ単体では保証されるものではなく、外部への問い合わせが発生するため、DNS 名前解決のように頻度が高い問い合わせの度にこのステップを踏むことになる。信用を確認した上で公開鍵をキャッシュすることも可能であるが、先に述べた ACL の制御レコードと同様、これによる有効性はアクセス頻度に依存してしまうと考える。

一般に、ネットワークを介した問い合わせから回答までの一連のステップにおいて、計算機上における処理時間よりもメッセージ送受信における伝送時間のほうが大きい。そのため、この方式におけるアクセス制御処理にキャッシュが活用できないことによって、通常の DNS 名前解決と比べるとレスポンスタイムに大きな差が生じてしまうと考えられる。

3. 権限証明書を用いたアクセス制御方式の提案

前章では、DNS における名前解決をユーザ単位でアクセス制御を行う既存方式を取り上げ、問題点としてキャッシュ利用効率が低く、DNS が本来持つような高いスケーラビリティと早い応答が望めない仕組みであることを述べた。本章では、本稿における提案である権限証明書を導入することで、ユーザからの問い合わせを受けたリゾルバがアクセス制御情報をキャッシュし、そのキャッシュを参照するだけでアクセス制御処理を行う方式を提案する。

3.1. 提案方式の概要

まず、問い合わせメッセージがどのユーザによるものかを識別する仕組みは、既存方式と同様、ユーザが SIG(0)メッセージ署名を行い、それを受け取ったネームサーバまたはリゾルバが署名の検証を行う方法を提案する。

既存方式と異なるのは、そのユーザがアクセス制御対象リソースに対しての名前解決を許可されているかを確認

する仕組みである。既存手法が名前解決を許可するユーザごとに、その公開鍵を識別し、公開鍵 ID を含む ACL を Zone データ内に記述するため、キャッシュをうまく活用できなかった。そこで、提案方式ではリソースの管理者、つまり Zone 管理者が許可を与える各ユーザに対し、後述する権限証明書をあらかじめ作成し、配布する。そして、ユーザは問い合わせを行う際、この権限証明書を問い合わせメッセージ中に含め、問い合わせが改ざんされることがないよう、メッセージ全体にSIG(0)署名を付加する。問い合わせを受けたネームサーバもしくはリゾルバは、権限証明書の検証と、問い合わせメッセージのSIG(0)署名を検証することでアクセス制御を処理する。

3.2. 権限証明書

図 2 に示すように、権限証明書は、Zone 管理者の公開鍵、許可を与えるユーザの公開鍵、アクセス制御対象リソースの FQDN、証明書の有効期限を含み、発行者である Zone 管理者の秘密鍵で署名を付加したものになる。

この権限証明書の作成は、対象リソースに対する名前解決をユーザに許可することであるが、証明書に含めるユーザ公開鍵が本当にそのユーザのものであるか、ユーザごとに正当性を確認する必要があるが、既存方式における ACL 内の署名者 ID を記述する際にも、同様に本人証明が必要になるため特別な手間が発生するというものではない。

権限証明書の発行の段階で、Zone 管理者がユーザ公開鍵の正当性を確認してから、証明書を作成し Zone 鍵による署名を行うため、名前解決の際にユーザから提示された権限証明書を検証するだけで、ユーザに与えられたアクセス権限の確認だけでなく、SIG(0)メッセージ署名に必要なユーザ公開鍵を、正当性の検証で他者に問い合わせることなく取得することができる。

さらに、この権限証明書は、ユーザごとに作成・配布されるためそれぞれ異なるが、発行者である Zone 管理者の公開鍵だけあれば、許可ユーザ数が増えてもアクセス制御処理を行う上では影響はない。

図 2　権限証明書のフォーマット

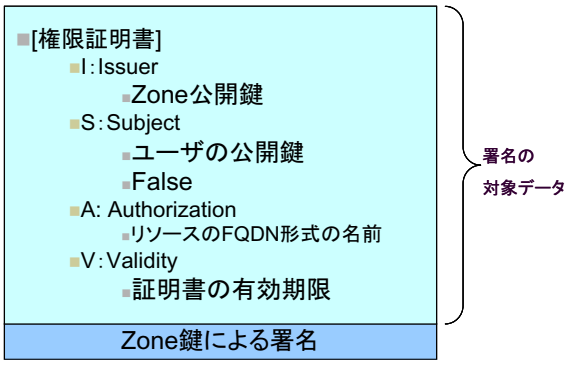

図 3　提案方式における Zone データの記述

```
.......
test.com.   IN NS    test.com.
            IN A     192.168.1.10
            IN KEY   256 3 3 Ci71eZg7X6JLEVv5rK····
.......
www         IN A     192.168.1.20
            IN ACRR  "AC=www.test.com. TYPE=AKEY=test.com. "
.....
```

3.3. Zone データの記述

図 3 は、本提案方式におけるネームサーバが管理する Zone データの例(SOA レコードなど一部省略)である。

この例ではまず、www という名前の A リソースレコードをアクセス制御対象とするため、同じ名前の ACRR というタイプのリソースレコードを追加することを提案している。ACRR レコードが示す情報は、アクセス制御対象レコードの FQDN、タイプ、そしてユーザに与えた権限証明書 を検証するための公開鍵の FQDN を指定している。また、この公開鍵は、同一 Zone 内に KEY レコードとして管理されている。

3.4. キャッシュ参照によるアクセス制御処理

3.2.で述べたように、ユーザからの問い合わせメッセージに対するアクセス制御処理を行うには、Zone 管理者の公開鍵だけがあればよいとした。

よって、リゾルバは前項でも述べた以下の 3 つのレコードをキャッシュすれば、ネームサーバに代わりアクセス制御処理とキャッシュ回答を返すことができる。

① アクセス制御対象リソースレコード(A レコード等)
② ACRR レコード
③ 権限証明書を検証するための Zone 公開鍵レコード

これら 3 つのレコードは、通常の DNS キャッシュと同様、権威ネームサーバからの回答メッセージから取得可能である。また、Zone 公開鍵レコードなど、リゾルバがネームサーバからの回答を信用できるかという点については、DNSSEC など既存の仕組みを用いることで解決できると考える。

3.5. 提案方式のまとめ

以上に示したように、本稿で提案する権限証明書を導入し、3.4.の 3 つのレコードをリゾルバがキャッシュすることで、まずユーザのアクセス権限を問い合わせメッセージから即座に確認できる。また、この方式は通常の DNS キャッシュに比べると 2 つのキャッシュレコードが増えることになるが、許可ユーザ数に影響しないため、キャッシュサイズを最低限に抑えることができる。

さらに、問い合わせメッセージ署名を検証するために必要なユーザ公開鍵自体の正当性の確認は、権限証明書を作成する際に 1 回だけ行えばよいから、名前解決という高頻度な問い合わせの際に即座にユーザ公開鍵が信用できることは、キャッシュによる効果と併せて、高速な応答

を返す上で非常に有効であると考える。

4. 評価実験

ここまで、アクセス制御が伴う名前解決処理にも、リゾルバにおけるキャッシュ機能を活用することで、DNS 本来のスケーラブルで高速な応答を返すことができると考えたが、2.では既存方式はこれに適さないことを問題点と指摘した。これを踏まえ、提案方式として権限証明書を導入することで、既存方式に比べ消費するキャッシュサイズが少なく、かつ、キャッシュによる効果が得られる仕組みであることを述べた。

本章では、評価として、2 つの方式におけるアクセス制御付き名前解決処理やキャッシュ回答処理などの必要なルーチンを実装し、ユーザ数や対象リソース数など同一のシチュエーションをパラメータとして与え、このとき 1 つのリゾルバが消費するキャッシュサイズと得られるキャッシュのヒット率を定量的に比較するための評価計算を行う。

4.1. 実験方法

4.1.1. シチュエーション

ユーザ数を 100、アクセス制御対象リソース数を 100 とし、全ユーザが 1 つのリゾルバに問い合わせを行うものとする。このとき、各アクセス制御対象リソースに割り当てられる許可ユーザ数は 20～100 人と、少ないものから多いものまでを連続的に割り当てる。これは、リソース毎にアクセス頻度が異なるよう、重み付けをした。

4.1.2. 問い合わせ発生パターン

まず、今回の評価では、十分に多い回数の問い合わせを発生させ、それらを全てリゾルバが処理し終えるまで時間経過とともに計測することになるが、この時間軸は、実時間上のものではなく、今回の評価実験上のもので、経過時間単位を 1 とする。キャッシュ時間となるリソースレコードの TTL も同一の時間軸上に従うものとする。

問い合わせ発生の手順は、まず問い合わせを行うユーザをランダムに選択し、そのユーザが名前解決を行う対象リソースを、自らに許可されたリソースの中からランダムに決定するという流れである。総計測時間を 100,000 単位時間とし、各問い合わせがリゾルバに到着する時間間隔を 0~3 のランダム値とした。

4.1.3. 計測

以上の条件に基づき発生する問い合わせをリゾルバが処理する際、消費するキャッシュサイズと得られるキャッシュヒット率を 1,000 単位時間毎に計測する。

なお、計測によって 1,000 単位時間あたりに発生する問い合わせ回数は約 400 回前後であった。

4.2. 消費キャッシュサイズに関する評価

図 4 は計測時間毎に消費するキャッシュサイズの平均を、

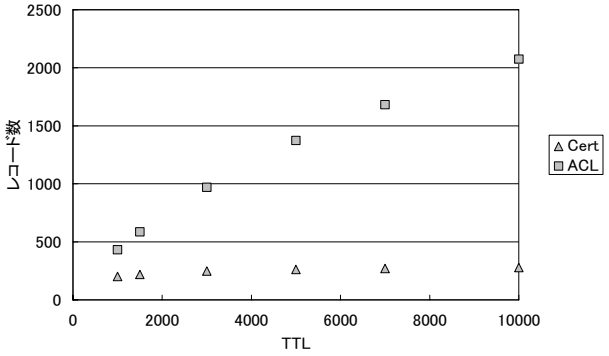

図 4　TTL 別・平均消費キャッシュサイズ

TTL 別に示したものである。なお、このときの TTL 値は、全リソースレコードに同じ値を設定し、1000、3000、5000、7000、10000 の 6 パターンを計測した。

まず、図を見れば明らかなように、全ての TTL の場合において、提案方式に比べ、既存方式のほうがキャッシュ消費が大きいことがわかる。また、既存方式・提案方式共に、TTL 値に比例してキャッシュ消費が増加している。これは通常の DNS キャッシュと同様、単に TTL が大きいためにキャッシュレコードが時間経過と共に累積しているためである。

また、TTL が 1000 のときは約 2. 1 倍、TTL が 10000 のときは約 7. 1 倍と、TTL 値が大きくなるにつれこの差が大きくなっている。これは、既存方式においてアクセス制御を処理するためには、ユーザ毎に ACL となるレコードを定義しなければならず、このレコードがキャッシュされることと、TTL 値が大きいため蓄積による増加率がより大きくなってしまう。

4.3. キャッシュヒット率に関する評価

図 5 と図 6 は、それぞれ既存方式と提案方式の、時間経過によるキャッシュヒット率の、1000 単位時間毎の変化を示したものである。

共に、TTL 値が大きいほうが、キャッシュヒット率が高くなることは両図から明らかである。

今回の実験におけるシチュエーションでは、提案方式は、全ての TTL 値において、既存方式より高いキャッシュヒット率を出すことができ、常に約 80%以上のキャッシュヒット率を維持している。

既存方式においても、TTL が 10000 のときに最大で約 64%のヒット率を記録しているが、図と照らし合わせるとわかるように、キャッシュヒット率は消費キャッシュサイズに比例して、高くなる傾向があるため、これ以上のヒット率を出すためにはさらにキャッシュサイズが必要になることがわかる。

また、既存方式と提案方式のヒット率の時間経過で異なる変化傾向をみせている。提案方式のほうは、ある一定周期でヒット率が急激に下がり、急激に上昇しており、TTL 値

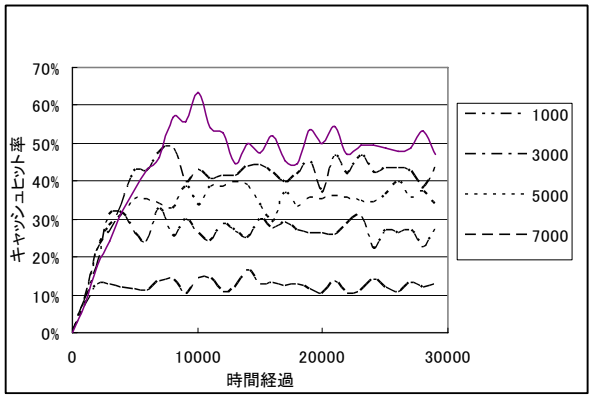

図 5　キャッシュヒット率の時間経過（既存方式）

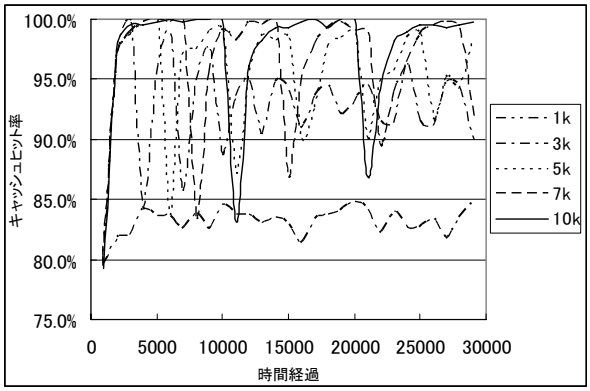

図 6　キャッシュヒット率の時間経過（提案方式）

が高い場合には特に顕著である。これは、4.1.1.で述べたようにリソースごとに重み付けをしてアクセス頻度に偏りを持たせたことにより、アクセス頻度の高いリソースのキャッシュ TTL が切れたことによる影響だと考えられる。

5.　まとめと今後の課題

本稿では、DNS における名前解決を、ユーザを識別してアクセス制御を行う仕組みに、提案する権限証明書を導入すること、リゾルバにおけるキャッシュの利用効率を高めることで、アクセス制御処理を行う場合でも、より高速な回答を得られることを述べ、実験では提案方式が既存方式に比べ、消費キャッシュサイズを最小限に抑え、高いキャッシュヒット率を維持できることを示した。

今後の課題としては、実際の環境では、アクセス制御に関係しない DNS キャッシュが混在するなど、今回用意したシチュエーションとはまた異なる要因が性能に影響を与えると考えられるため、より多面的に検討していくとともに、提案方式に基づくプロトタイプを実装し、実環境における性能評価を行いたいと考えている。

参考文献

[1] P. Mockapetris, “Domain Names – Concepts and Facilities”, RFC 1034, November 1987.

[2] P. Mockapetris, “Domain Names – Implementation and Specification”, RFC 1035, November 1987.

[3] Sun Microsystems, Jini Network Technology 2.0., http://wwws.sun.com/software/jini/

[4] UPnP Forum, Universal Plug and Play., http://www.upnp.org/

[5] 馬場達也、日下貴義、山岡正輝、松田栄之、“アクセス制御機能付 DNS の実装と評価”、情報処理学会研究報告、2004-CSEC-24, Vol.2004, No.17, pp.99-104, March 2004.

[6] D. Eastlake, “DNS Request and Transaction Signature (SIG(0))”, RFC 2931, September 2000.

[7] D. Eastlake, “Domain Name System Security Extensions”, RFC 2535, March 1999.