マルチメディア通信と分散処理 111－31
コンピュータセキュリティ 20－31
（２００３．２．２８）

DNS におけるアクセス制御プロトコルの検討

馬場 達也　日下 貴義　山岡 正輝　松田 栄之
株式会社 NTT データ 技術開発本部
〒104-0033 東京都中央区新川 1-21-2 茅場町タワー
E-mail: {babatt, kusakat, yamaokam, matsudasg}@nttdata.co.jp

あらまし DNS（Domain Name System）は、インターネットにおいて、ホスト名と IP アドレスを変換するという重要な役割を担っている。しかし、インターネットは研究者間の通信基盤として発展してきたという経緯もあり、DNS におけるセキュリティ対策は十分であるとはいえない。これまでに、DNS にセキュリティ機能を加えるための拡張が議論されてきたが、DNS のデータに対するアクセス制御の仕組みはまだ十分に実現されていない。そこで、本稿では、DNS の問い合わせ元を、DNS のデータであるリソースレコードを保持するネームサーバ側で認証し、その認証結果に基づいてリソースレコードの内容を返却するか否かを決定するアクセス制御の仕組みと、その仕組みを実現するためのプロトコルについて提案する。
キーワード DNS，アクセス制御，ネームサーバ，セキュリティ

Proposal of Access Control Mechanism and Protocol for Domain Name Systems

Tatsuya BABA, Takayoshi KUSAKA, Masaki YAMAOKA, and Shigeyuki MATSUDA
Research and Development Headquarters, NTT Data Corporation
Kayabacho Tower, 1-21-2, Shinkawa, Chuo-ku, Tokyo, 104-0033 Japan
E-mail: {babatt, kusakat, yamaokam, matsudasg}@nttdata.co.jp

Abstract DNS (Domain Name System) plays an important role in the Internet. It provides the mechanism for translating internet domain names for network hosts into IP addresses, for email routing, and for other information. As the Internet has grown to become a business infrastructure, security extensions to the DNS have been discussed and developed. However, any sort of access control lists or other means to differentiate inquirers are not provided in these extensions. In this paper, we propose a mechanism and a protocol for access control in DNSs.
Keyword DNS, Access Control, Name Server, Security

1. はじめに

DNS（Domain Name System）[1, 2]は、インターネットにおいて、ホスト名を IP アドレスに変換する名前解決と呼ばれる重要な機能を提供しているだけでなく、メール配送やサービス検索などにも利用されており、インターネットのインフラともいうべき重要なシステムとなっている。しかし、インターネットは、元来、研究者間の通信基盤として発展してきたという経緯もあり、DNS におけるセキュリティ対策は十分であるとはいえない。

DNS におけるセキュリティ対策としては、これまでに、DNSSEC（DNS Security Extensions）[3]や TSIG（Transaction Signature）[4]、SIG(0)[5]などが提案されてきた。DNSSEC では、DNS のデータであるリソースレコードに対して、公開鍵暗号技術を使用して署名を施すことにより、データの改ざんを防ぐことを可能としている。また、TSIG および SIG(0)では、ネットワーク上を流れる DNS メッセージに MAC（Message Authentication Code）やデジタル署名を付加することによって、ネットワーク上での DNS メッセージの改ざんや、送信者のなりすましを防ぐことを可能としている。

しかし、DNS への問い合わせに対するアクセス制御の機能は、これらの技術では提供されていない。これは、DNS のデータは公開されるべきものという考えの下で設計されてきたため、アクセス制御に対するユーザの要求はあるものの、議論の対象外とされてきたためである。しかし、ホストにアクセスする前の DNS の名前解決の段階でアクセス制御を行うことができれば、従来のホストやファイアウォールでのアクセス制御に加えて、さらにセキュリティを高めることができると考えられる。また、年々増加しているサービス妨害攻撃（DoS 攻撃）では、送信元を偽造した大量のパ

ケットを送りつけることにより、ターゲットのネットワークやホストに負荷をかけることを目的としており、このような攻撃はホストやファイアウォールでのアクセス制御では防ぐことができない。しかし、DNS の名前解決の段階で、アクセスを許可するユーザのみに IP アドレスを開示するようなアクセス制御を行うことができれば、これらの攻撃をある程度防ぐことができると考えられる。

そこで、著者らは、DNS におけるアクセス制御を実現する仕組みとして、公開鍵暗号を使用したチャレンジレスポンス方式によって問い合わせ元を認証し、アクセス制御を行う方式を提案している[6]。この方式では、問い合わせ元をユーザ単位で認証し、認証結果に基づいてリソースレコード毎にアクセス制御を実現することが可能である。しかし、認証方式として採用したチャレンジレスポンス方式は、クライアントからの認証要求に応じてチャレンジを送信した後に、クライアントからのレスポンスを待つ必要があるため、不正な要求パケットを大量に送り付けるような攻撃に弱いという問題がある。そこで本稿では、新たなアクセス制御のための仕組みとプロトコルを提案する。

2. アクセス制御を実現する従来手法の問題点

DNS データへのアクセスを制限するには、BIND[7]に実装されているアクセス制御機能を用いる方法がある。BIND では、問い合わせパケットの送信元 IP アドレスによって問い合わせ元ホストを識別し、識別結果に基づいてゾーン単位のアクセス制御を実現している。また、BIND に実装されている TSIG の機能を利用することにより、あらかじめ配布した TSIG 用の認証鍵の有無によってアクセス制御を行うことも可能となる。

しかしながら、IP アドレスによって問い合わせ元ホストを識別する方式を使用した場合、DHCP 等により動的に IP アドレスを割り当てている環境では、個々のホスト毎にアクセス制御を行うことはできない。また、IP アドレスを偽造されてアクセスされた場合には対応できないという問題もある。もう一方の TSIG を利用する方法は、アドレスの偽造にも対応できるという利点を持っているが、あらかじめネームサーバとクライアントとの間で認証用の鍵を共有しておく必要があるため、スケーラビリティの面で劣るという問題がある。さらに、BIND では、ゾーン単位でアクセス制御を行っており、同一のゾーンに属する WWW サーバやメールサーバなどのサービスごとにアクセス制御を実現することができないという問題もある。サービスごとにアクセス制御を行うためには、ゾーン単位ではなく、リソースレコード単位で設定できる必要がある。

以上の DNS でのアクセス制御の現状を表１にまとめる。

表 1　DNS のアクセス制御の現状

アクセス制御方式	なりすましへの対応	アクセス制御の単位	欠点
IP アドレスによる識別（BIND）	×	ゾーン単位	DHCP 環境では識別子が変化するため利用が難しい
TSIG による認証（BIND）	○	ゾーン単位	認証用の秘密鍵をあらかじめ共有しておく必要がある

3. アクセス制御に求められる機能および条件

アクセス制御を実現するために必要な機能および条件としては、以下のものがある。

①アクセス元の識別および認証

ネームサーバがアクセス制御を行うためには、ネームサーバが、アクセス元のクライアントを何らかの識別子を用いて識別する機能が必要となる。また、識別するだけでは、なりすましに対応できないため、確かにそのクライアントがアクセスしてきていることを認証する機能が必要となる。

②データの機密性保護

アクセスを許可している相手に返答したデータを、アクセスが許可されていない第三者にネットワーク上で盗聴される可能性がある。このため、返答するデータを暗号化する機能が必要となる。

③アクセス制御リストの管理

アクセス制御を行うネームサーバは、アクセス制御リスト（ACL：Access Control List）を保持し、管理する必要がある。この ACL の内容は、プライマリネームサーバとセカンダリネームサーバの間で一貫している必要がある。

④リソースレコード単位のアクセス制御

ゾーン単位ではなく、リソースレコード単位でアクセス制御できる必要がある。

⑤既存システムとの相互運用性の確保

DNS はインターネットのインフラとして既に広く普及しているため、アクセス制御機能を導入することによって、既存の DNS の仕組みが大きく変更されてはならない。

⑥スケーラビリティが確保されること

クライアントが増加しても認証情報などの管理が複雑にならないように、スケーラビリティを考慮した方式である必要がある。

4. アクセス制御方式の提案

ここでは、前章のアクセス制御に求められる機能および条件に基づいて、アクセス制御方式を提案する。

4.1. アクセス元の識別および認証

アクセス元を識別および認証する方式として、DNS メッセージにデジタル署名を付加することにより、なりすましやメッセージの改ざんを検知する技術である SIG(0)を利用する方式を提案する。SIG(0)が DNS メッセージに付加する SIG(0)リソースレコード（SIG(0) RR）には、デジタル署名とドメイン名形式の署名者 ID が含まれているため、ネームサーバは、署名者 ID を基にアクセス元を識別し、署名を検証することでアクセス元を認証することができる。署名者 ID としては、通常は FQDN（Fully Qualified Domain Name）で記述されたホスト名が使用されるが、ユーザの ID を「baba._user.example.com」のようにドメイン名形式で表記することで、ホスト単位だけではなく、ユーザ単位でもアクセス制御を行うことができるようにする。

アクセス元を識別および認証する技術としては、SIG(0)の他に TSIG を利用することもできるが、TSIG では、事前に共有された認証用の秘密鍵を使用して MAC を生成するため、個々のクライアントを識別するためには、異なる秘密鍵をあらかじめネームサーバとクライアントとの間で共有しておかなければならないという欠点がある。これに対して、SIG(0)では、署名を検証するために必要な公開鍵は、現在標準化が進められている DNSSEC により、DNS を利用して安全に入手することが可能となるため、あらかじめネームサーバにクライアントの公開鍵を登録しておく必要はない。このため、SIG(0)は、処理速度の面で TSIG より劣るものの、鍵管理の面で優れているため、アクセス元の識別および認証には、SIG(0)を利用することとした。以上に述べた SIG(0)と TSIG の違いを表 2 にまとめる。

ただし、SIG(0)や TSIG は、現在、ダイナミックアップデートやゾーン転送で相手の認証のために使用されているものの、ローカルネームサーバを介す通常の名前解決では使用されていない。これは、ローカルネームサーバを介すと、ローカルネームサーバ上でメッセージの内容が変更されてしまい、署名が無効となってしまうためである。このため、SIG(0)を認証方式として使用するために、クライアントはローカルネームサーバを介さずに、ネームサーバと直接通信する必要がある。

表 2　認証方式の比較

認証方式	識別子	特徴
TSIG	鍵 ID	現在の BIND で実装されているが、鍵の管理のためのコストがかかる
SIG(0)	署名者 ID	TSIG より処理コストがかかるが、鍵の管理は容易

4.2. データの機密性保護

DNS の仕様では、DNS メッセージの機密性を保護するための仕組みは規定されていない。データの機密性を保護するための仕組みとしては、SSL/TLS や SSH 、IPsec などが存在するが、DNS は通常 UDP を使用するため、SSL/TLS や SSH を使用することはできない。また、IPsec は DNS パケットを暗号化することはできるが、基本的に DNS の仕組みとは独立しているため、アクセスが制限されているリソースレコードを含む DNS パケットは暗号化し、その他の DNS パケットは暗号化しないという機能は、IPsec を用いて実現することは難しい。

そこで、DNS メッセージを暗号化するための仕組みとして、TENC リソースレコード（TENC RR：Transaction Encryption Resource Record）を新たに提案する。TENC では、オリジナルの DNS メッセージ全体を、共通鍵暗号を使用して暗号化し、暗号化したメッセージを新たに作成した DNS メッセージ中の TENC RR のデータ部に挿入して送信する（図 1）。DNS メッセージに含まれるリソースレコードだけでなく、DNS ヘッダも暗号化することにより、返却したエラーの種類やリソースレコードの数なども第三者に知られないようにすることが可能となる。

ただし、TENC を利用するためには、暗号化用の秘密鍵をあらかじめクライアントとネームサーバとの間で共有しておく必要がある。これは、TSIG で使用する認証鍵をセットアップするために使用される TKEY（Transaction Key）[8]を拡張することで実現する。TKEY では、鍵のセットアップの際に、生成する鍵の種類を相手に通知するために、TSIG で使用する MAC アルゴリズムの ID を指定する仕様となっている。そこで、TENC で TKEY を利用できるようにするために、このアルゴリズム ID に、TENC で使用する共通鍵暗号アルゴリズムの ID を追加する。現在、TKEY では、TSIG で使用する MAC アルゴリズムである HMAC-MD5 用に、”HMAC-MD5.SIG-ALG.REG.INT”という ID を定義しているが[9]、この他に、表 3 のような共通鍵暗号アルゴリズム用の ID を追加する。

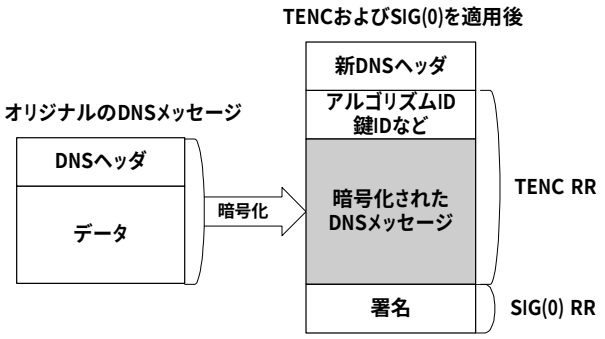

図 1　TENC による DNS メッセージの暗号化

表 3　追加する TKEY のアルゴリズム ID の例

アルゴリズム	生成する鍵の長さ	アルゴリズム ID
DES-CBC	64 ビット（パリティ含む）	des-cbc
3DES-CBC	192 ビット（パリティ含む）	3des-cbc
Blowfish-CBC	128 ビット	128.blowfish-cbc
CAST-128-CBC	128 ビット	cast-128-cbc
AES-CBC	128 ビット	128.aes-cbc
	192 ビット	192.aes-cbc
	256 ビット	256.aes-cbc

4.3. アクセス制御リストの管理およびリソースレコード単位のアクセス制御

ACL の内容は、同じゾーンを管理するネームサーバ間で異なっていてはならない。このため、ゾーン転送時に同時に ACL も転送されるように、ゾーンデータ内に TXT レコードを使用して ACL を記述する方式を提案する。この TXT レコードでは、以下のように、「ACL」という文字に続いて、対象となるリソースレコードのタイプ、識別子（SIG(0) RR の署名者 ID で使用する FQDN）を記述することで、アクセスを許可するホストまたはユーザをリソースレコード毎に記述する。

```
www    IN  A     192.168.0.30
       IN  TXT   "ACL A baba._user.example.com."
       IN  TXT   "ACL A kusaka._user.example.com."
```

ただし、この ACL は、DNS メッセージの回答部に含まれるリソースレコードのみに適用することとし、権威部および付加情報部にリソースレコードを挿入する場合は、そのリソースレコードに対する ACL は無視する。例えば、メール配送プログラムが MX レコードを問い合わせた場合は、通常は、回答部にメールサーバのホスト名を含む MX レコードが入り、さらに、付加情報部にメールサーバの IP アドレスを含む A レコードが入る。ここで、ACL で MX レコードに対してはアクセスを許可しているが、メールサーバの A レコードに対してはアクセスを許可していなかった場合に、クライアントに MX レコードのみが返されるのは問題であるため、付加情報部に入るリソースレコードは、ACL の内容に依らずに無条件に挿入することとする。

また、DNS のツリー構造をたどるために必要な SOA、NS、SIG、NXT、KEY、DS の各リソースレコードに対してはアクセス制御を行わず、すべての問い合わせに対して回答することとする。

4.4. 既存システムとの相互運用性の確保

DNS では、ローカルネームサーバがクライアントに代わって外部ネームサーバに対して問い合わせを行い、名前解決を行うというモデルになっている。しかし、既存のローカルネームサーバは、クライアントが付加した TSIG RR や SIG(0) RR を外部ネームサーバに転送することができない。既存のローカルネームサーバにこれらの認証用のリソースレコードを転送する機能を加えることも可能であるが、その場合は、リモートアクセスで使用する ISP のローカルネームサーバや、出張先で使用するローカルネームサーバにも新たに機能を加えなければならないという問題がある。

そこで、クライアントがアクセス制御されているゾーン（アクセス制御対象ゾーン）のリソースレコードにアクセスする場合は、ローカルネームサーバを介さずに、アクセス先ネームサーバに直接 SIG(0) RR を付加した問い合わせを発行するようにする。クライアントは、アクセス先ゾーンを管理するネームサーバを発見するために、最初に、名前解決をしようとするゾーンの NS レコードをローカルネームサーバ経由で問い合わせ、さらに、そこで得られたネームサーバのホスト名に対して A レコードを問い合わせることでアクセス先ネームサーバの IP アドレスを取得する。クライアントとネームサーバが直接通信することにより、ローカルネームサーバ上でアクセス制御対象のリソースレコードがキャッシュされてしまうという問題を回避することも可能となる。

4.5. スケーラビリティの確保

SIG(0)署名を検証するために必要なホストまたはユーザの公開鍵は KEY レコードに登録しておき、DNS から自動的に取得できるようにしておく。これにより、クライアントが増加した場合でも、スケーラビリティを確保できるようにする。

5. 提案するアクセス制御プロトコルの動作

ここでは、DNS アクセス制御プロトコルの動作について説明する。5.1 節では、アクセス制御機能付クライアント（以後、AC クライアントと呼ぶ）がアクセ

ス制御対象ゾーンの名前解決を行う場合に、アクセス先となるアクセス制御機能付ネームサーバ（以後、ACネームサーバと呼ぶ）を探索してアドレスを取得するまでの動作を説明する。そして、5.2 節では、取得したアドレスを使用して、AC クライアントから AC ネームサーバに対して、直接、署名付きの問い合わせを発行し、回答を得るまでの動作を説明する。また、AC ネームサーバの保持するアクセス制御対象ゾーンに対して、通常の問い合わせが行われた場合の動作を 5.3 節で説明する。

5.1. アクセス先 AC ネームサーバの探索

AC クライアントには、あらかじめアクセス制御対象ゾーンの一覧を設定しておく。AC クライアントが名前解決を行う場合には、この内容を確認し、解決しようとしている名前がアクセス制御対象ゾーンに含まれているのかどうかを確認する。解決しようとしている名前がアクセス制御対象ゾーンに含まれていない場合は、ローカルネームサーバ経由で通常の名前解決を行う。解決しようとしている名前がアクセス制御対象ゾーンに含まれている場合は、図 2 に示すように、そのゾーンの NS レコードをローカルネームサーバ経由で問い合わせ、ゾーンを管理する AC ネームサーバのホスト名を取得する。この際に、AC ネームサーバの IP アドレスも同時に取得することができればそれを使用し、取得することができなければ、さらに、その AC ネームサーバの A レコードを問い合わせることにより、解決しようとしている名前を管理している AC ネームサーバの IP アドレスを取得する。

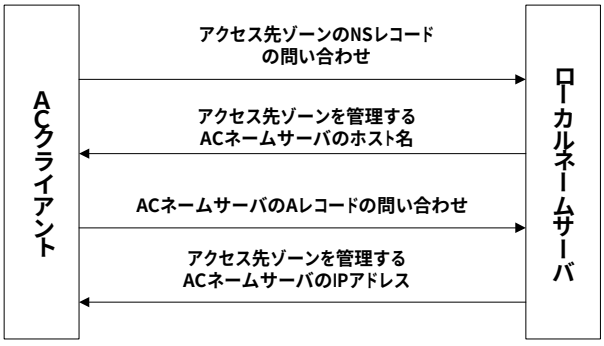

図 2 AC ネームサーバの探索シーケンス

5.2. AC ネームサーバへの問い合わせ

アクセス先ゾーンを管理する AC ネームサーバの IP アドレスを取得したら、AC クライアントは、AC ネームサーバに対して、ローカルネームサーバを介さずに直接問い合わせを行う。この際に、DNS メッセージを暗号化するかどうかを選択することができる。

（1）DNS メッセージを暗号化しない場合

DNS メッセージを暗号化しない場合は、図 3 のように、問い合わせメッセージに、クライアントユーザまたはホストの秘密鍵で署名を施した SIG(0) RR を付加したものを、AC ネームサーバに直接送信する。AC ネームサーバでは、SIG(0) RR の署名者 ID に対応した公開鍵を DNS から取得し、受信した DNS メッセージに付加されている署名を検証する。署名の検証に成功した場合は、該当するリソースレコードの ACL を参照し、リソースレコードの内容を返却して良いかどうかを判断する。アクセスが許可されている場合には、回答にサーバの秘密鍵で署名を施した SIG(0) RR を付加した DNS メッセージをクライアントに送信する。AC クライアントでは、受信した DNS メッセージに付加されている署名をサーバの公開鍵を使用して検証する。

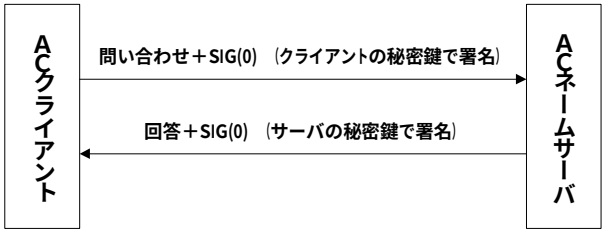

図 3 暗号化を行わない場合のプロトコルシーケンス

（2）DNS メッセージを暗号化する場合

DNS メッセージを暗号化する場合は、図 4 のように、最初に AC クライアントと AC ネームサーバとの間で TKEY による暗号化用の秘密鍵のセットアップを行う。そして、TKEY によって生成された秘密鍵を使用して、DNS メッセージを暗号化し、その内容を TENC RR に挿入して送信する。ここで交換されるメッセージには、すべて SIG(0) RR を付加して認証を行う。

図 4 暗号化を行う場合のプロトコルシーケンス

5.3. 署名のない問い合わせを受信した場合

AC ネームサーバが、署名が付加されていない問い合わせを受信した場合は、最初に、問い合わせ対象リソースレコードが、「すべてに対してアクセスを許可

しているか」どうかを判断する。もし、すべてに対してアクセスを許可しているのであれば、そのリソースレコードの内容を返却する。しかし、アクセス制御対象となっている場合にはエラーを返却する。

また、SOA、NS、SIG、NXT、KEY、DS の各リソースレコードはアクセス制御対象としないため、これらのリソースレコードに対して署名が付加されていない問い合わせがあった場合は､無条件に回答を返却する。

6. 考察

提案した方式を、セキュリティ面および運用面から考察する。

6.1. セキュリティ面

提案したプロトコルは、ネームサーバがクライアントからの要求に答えた後に、クライアントからの返答を待つような仕様になっていない。このため、不正な要求パケットを大量に送信されても、ネームサーバ側で待ち状態が多く発生してダウンしてしまうようなことはなく､DoS 攻撃に対して耐性があると考えられる。

ただし、署名による認証では、悪意のある第三者がネットワーク上に送出された署名付きの DNS パケットを取得し、それを再度利用するというリプレイ攻撃が可能になってしまうという問題がある。しかし、SIG(0)では、署名の有効期限を設定できるため、この署名の有効期間を短くすることにより、リプレイ攻撃が可能な時間を短くすることができる。

6.2. 運用面

ユーザは、署名に使用する鍵のペアを生成し、その公開鍵を DNS の KEY レコードとして登録さえしておけば、アクセス制御対象ゾーンを指定するだけでアクセス制御機能を利用することができる。このため、ユーザにとっての運用上の不便さは少ないと考えられる。

ネームサーバ側では、署名に使用する鍵ペアを生成し、公開鍵を DNS の KEY レコードとして登録さえしておけばよい。また、ACL はゾーン転送によって自動的に転送されるため、プライマリネームサーバのみで記述しておけばよい。ただし、SIG(0) RR では、署名の有効期限を設定するため、クライアントとネームサーバの間では、NTP（Network Time Protocol）を使用して時刻を同期させておく必要がある。

また、本方式はローカルネームサーバに手を加える必要がないため、導入は比較的容易であると考えられる。しかし、クライアントから外部のネームサーバに直接アクセスするため、ローカルネームサーバ以外からの DNS パケットの通過をファイアウォールで許可していないサイトでは、ローカルネームサーバに DNS パケットを転送するプロキシ機能を追加して対処する必要がある。プロキシで DNS パケットを転送させるには、最終宛先となるネームサーバのアドレスをクライアントからプロキシに伝える必要があるが、これは、最終宛先アドレスを格納するためのリソースレコードを新たに定義し、DNS プロキシが、そのリソースレコードを含むパケットを受け取った場合に、その最終宛先に対して転送することで実現できる。

7. まとめ

DNS の問い合わせ元をホストまたはユーザ単位で認証し、認証結果に基づいてリソースレコードへのアクセス制御を行う方式と、方式を実現するために必要なプロトコルについて提案した。今後は、提案した方式およびプロトコルをプロトタイプとして実装し、性能面などの評価を行う予定である。

謝辞

本研究は、通信・放送機構（TAO）の委託研究テーマ「次世代 DNS に関する研究開発」の一環として行われているものである。

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] D. Eastlake 3rd, "Domain Name System Security Extensions", RFC 2535, March 1999.

[4] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[5] D. Eastlake 3rd, “DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[6] 山岡，田中，松田，“ DNS におけるアクセス制御の一検討 ”，情報処理学会第 64 回全国大会講演論文集（分冊 3），pp.395-396，March 2002.

[7] Internet Software Consortium, "BIND (Berkeley Internet Name Domain) ", http://www.isc.org/products/BIND/

[8] D. Eastlake 3rd, "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[9] The Internet Assigned Numbers Authority, “TSIG Algorithm Names“, http://www.iana.org/assignments/tsig-algorithm-names