Vol.2022-IOT-56 No.7
2022/3/7

複数フルサービスリゾルバと DNS over TLS 拡張による軽量なレコード検証手法

阿久津賢宏[1]　大久保拓哉[1]　島袋健大[1]　山井成良[1]　金勇[2]　中川令[1]

概要：インターネットを運用するうえで DNS(Domain Name System)は重要な部分を担っており，悪意ある攻撃の対象とされる場合がある．悪意ある攻撃の一種である DNS キャッシュポイズニングは成功すると，クライアントが意図せず攻撃者の悪意あるサイトに接続してしまい，フィッシング詐欺などの実害が発生する可能性がある．この攻撃への対応策として DNSSEC(Domain Name System Security Extensions)という電子署名を用いた DNS 応答の検証システムが存在する．この DNSSEC は完全展開することにより DNS キャッシュポイズニングを防ぐことができる．一方，DNSSEC の仕様が原因の様々な問題が挙げられており，いまだに完全展開が実現されていない．そこで，本研究ではサーバ証明書を権威サーバへ適用し，DNS over TLS の拡張を行うことで DNSSEC を用いずに権威サーバの実在性を検証する手法を提案する．本手法では権威サーバのサーバ証明書をクライアントが検証することを可能とし証明書を一度取得することで検証を行うことが可能であるため，通信回数や負荷の低下を実現できる．本手法のシステムを構築し，DNSSEC と本手法の特徴と問題点を比較したうえで，DNSSEC に代わる新たな手法の実現を目指す．

A Lightweight DNS Resource Record Validation Method with Multiple Full-service Resolvers and DNS over TLS Extension

TAKAHIRO AKUTSU[1]　OKUBO TAKUYA[1]　SHIMABUKURO KENTA[1]
NARIYOSHI YAMAI[1]　YONG JIN[2]　NAKAGAWA REI[1]

1. はじめに

現在，広く普及するインターネットを正しく運用するうえで DNS (Domain Name System) という技術が重要な役割を担っている．DNS とは，ネットワーク上の通信で必要な数字である IP アドレスと人が扱いやすいドメインを対応付けさせる技術のことを指す．DNS はドメインの委任という仕組みでドメイン名と IP アドレスのマッピングを階層構造で分散化させている．これはドメインの追加などの更新作業を容易にすると共に負荷分散の効果を持つ．こういった工夫により DNS は急速に成長するインターネットに対応し続けている．

しかし，DNS の重要性が増すにつれて DNS が悪意ある攻撃の対象にされる場合も増えており，実際に DNS キャッシュポイズニングといった攻撃が存在する．これは DNS の名前解決を高速化するために用いるキャッシュに偽のドメイン名と IP アドレスのマッピングを登録させる攻撃である．DNS キャッシュポイズニングが成功するとキャッシュの利用者を悪意あるサービスに誘導することが可能となり，例えばフィッシング詐欺などの実害が発生する可能性がある．このように DNS の正確な名前解決が行われないと目的のコンピュータとの通信が不能となり，目的のサービスを利用できなくなるためインターネットの運用が停止してしまう．

DNS キャッシュポイズニングのような偽の情報を登録させる攻撃に有効な対策として DNSSEC (Domain Name System Security Extensions) という技術が開発された．この DNSSEC は電子署名を用いて DNS 応答の完全性を検証することを可能する DNS の拡張機能であり，DNS キャッシュポイズニングを完全に防ぐことを可能にする技術として注目された．

DNSSEC を導入することで DNS キャッシュポイズニングを防ぐことが可能となる一方，DNSSEC にはいくつかの問題点が存在する．DNS は名前解決を行う際，各ドメインを管理する権威サーバに再帰的に問い合わせを行う．このとき，すべての DNS 応答に対し DNSSEC による検証が行われなければ最終的にユーザーが受け取る DNS 応答の完全性を保証できない．しかし，DNSSEC の検証をクライアントに提供するためには各階層のドメインを管理する権威サーバが DNSSEC を採用する必要がある一方, DNSSEC の採用率がいまだ低いという問題点があげられる[1]．これは電子署名の検証に伴う遅延や鍵の付加によるパケットサイズの増加などの DNSSEC の仕様上の問題が存在するためである[1].

この問題を解決するため，複数のフルサービスリゾルバへ同時に問い合わせを行い，それぞれの DNS 応答を用いてレコードの検証をクライアント側で行うシステムが提案された．このシステムでは問い合わせた複数フルサービスリゾルバからの各応答に共通して存在するレコードを信頼できる応答として利用する．このシステムにより，DNS キ

1 東京農工大学
Tokyo University of Agriculture and Technology

2 東京工業大学
Tokyo Institute of Technology

ャッシュポイズニングによる偽の応答をクライアントに利用させる困難さを高め，DNS 応答の信頼性の向上と検証の高速化を行っていた．しかしこの手法では DNS キャッシュポイズニングが行われていないにも関わらず，複数の応答に共通したレコードが存在しないという擬陽性の問題を抱えていた．

そこで本研究ではスタブリゾルバとフルサービスリゾルバ間の DNS 通信内容の秘匿化を目的とする DNS over TLS(DoT)という技術に着目し，DoT の拡張を行うことで関連研究の擬陽性に対応する共に DNSSEC より軽量なレコード検証手法を提案する．

2. 背景

2.1 DNS キャッシュポイズニング

DNS キャッシュポイズニングとはフルサービスリゾルバに偽のリソースレコードを登録させる攻撃である．DNS キャッシュポイズニングされたフルサービスリゾルバにクライアントが問い合わせを行うと登録された偽のリソースレコードがクライアントに送信される．その結果クライアントは偽のリソースレコードに記述された IP アドレスと通信を開始してしまう．つまりクライアントは意図せず悪意あるサービスへ誘導されてしまう．以下では DNS キャッシュポイズニングを行う手法の中からカミンスキー型攻撃手法について述べる．

2.1.1 カミンスキー型攻撃

カミンスキー型攻撃とは 2008 年 7 月に Dan Kaminsky 氏によって提案された効率の良い DNS キャッシュポイズニング手法である[2][3]．この手法は偽装対象のドメインに関するリソースレコードがフルサービスリゾルバに保存されていたとしても繰り返し攻撃を行うことができる．つまり従来の TTL を長く設定するという対策が有効でないと判明し，新たな対策手法が考えられるきっかけとなった．以下の図 1 にカミンスキー型攻撃の流れを示す．ここでは攻撃者が偽装するドメイン名を”www.example.jp.”，偽装する IP アドレスを”203.0.113.1”とする．

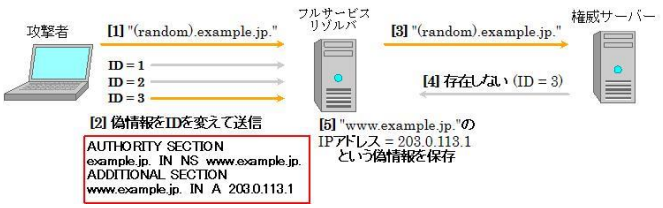

図 1 カミンスキー型攻撃の流れ

この攻撃は偽装したいリソースレコードを DNS 応答の付加情報としているため，フルサービスリゾルバに攻撃対象のドメインに関するリソースレコードがキャッシュとして存在しても行うことができる．つまり図 1 [1]で問い合わせるドメインのホスト名を変えることでキャッシュの TTL に関係なく攻撃を繰り返し行うことができる．

2.2 DNSSEC の動作

DNSSEC とは電子署名を用いてリソースレコードが改ざんされていないことを検証する DNS の拡張機能であり，DNS キャッシュポイズニングの根本的な解決策として注目された．

各ゾーンはリソースレコード検証用と子ゾーンの鍵検証用の 2 種類の鍵を用意し，親ゾーンから順に鍵とリソースレコードの検証を繰り返すことでリソースレコードの正当性を確保している．すべての権威サーバが信頼できるサーバであるとは限らないため，DNSSEC では子ゾーンの鍵の署名を親ゾーンの権威サーバが行うことで，信頼できる権威サーバから順に子ゾーンの信頼性検証を可能にしている．このような親ゾーンが子ゾーンの信頼性を担保する仕組みのことを信頼の連鎖と呼ぶ．またあらかじめ信頼できる権威サーバであるとされ，信頼の連鎖の起点となる権威サーバのことをトラストアンカーと呼ぶ．DNSSEC では名前解決対象のドメインを管理する権威サーバからトラストアンカーにたどり着くまですべての権威サーバの協力が必要である．

2.2.1 DNSSEC の問題点

DNSSEC には様々な問題が存在し[1]，以下でこれらの問題について述べる．

DNSSEC の普及率

DNSSEC は名前解決において問い合わせる全ての権威サーバが DNSSEC に対応し正常に検証が行える必要がある．2022 年 1 月 30 日時点では 1488 個の TLD の内ルートサーバに署名を登録しているものは 1364 個あり，TLD 全体の約 91.7%が署名を親ゾーンに登録している[4]．一方 TLD の子ゾーンの DNSSEC 対応率は依然として低く，例えば jp ドメインの子ゾーンの DNSSEC 対応率は 2022 年 1 月 30 日時点で 20.11%にとどまっている[5]．この結果から TLD の権威サーバによる DNSSEC 検証は行われている一方，信頼の連鎖が名前解決終了まで継続していない可能性が高いことがわかる．

パケットサイズの増加

一般に DNS は 512Byte までの応答サイズをサポートしている．しかし DNSSEC の検証に用いられる鍵は 128Byte 以上あるものが多く，DNSSEC を採用すると DNS 応答が 512Byte を超える可能性がある．実際 DNSSEC に対応した DNS 応答は 1500Byte を超えることが多い[1]．UDP を用いて DNS 通信を行う際に応答サイズが 512Byte を超える場合 EDNS0 (Extension Mechanisms for DNS) という DNS の拡張機能を導入する必要がある．EDNS0 に対応していない場

合 TCP 通信に切り替える必要があるが, TCP では 3 ウェイハンドシェイクといったオーバヘッドが発生しサーバへの負荷や通信の遅延が発生してしまう可能性がある.

相互運用性問題

1 フレームで送れるデータの最大値である PathMTU より大きいパケットは経路上でフラグメント化を起こす. DNSSEC を導入し DNS 応答サイズが増加すると DNS 応答がフラグメント化を起こす場合がある. しかしフラグメント化した DNS 応答はセキュリティのためファイアウォールによってブロックされてしまう場合がある. 実際に 24 種類のファイアウォールのうち 512Byte 以上の DNS 応答に対応したものは 4 つのみであった[1].

中間者攻撃問題

DNSSEC の導入により DNS 応答のフラグメント化が発生すると, 構造上ポート番号や DNS ヘッダ ID などのヘッダ情報が最初のフラグメントに含まれる. これを悪用し 2 つ目以降のフラグメントを偽装する第 2 フラグメント便乗攻撃により DNS キャッシュポイズニングを行うことができる[7][8]. つまり DNSSEC の導入により新たな攻撃が可能となる場合がある.

リフレクション攻撃

リフレクション攻撃とは攻撃対象の IP アドレスを用いて DNS 要求を送ることで攻撃対象に大量の応答を効率的に送り込む DoS 攻撃である. DNSSEC の導入による DNS 応答サイズの増加は効率的なリフレクション攻撃を可能にしてしまう. つまり DNSSEC の普及によりリフレクション攻撃のリスクが増加する.

DNSSEC の完全な普及は DNS キャッシュポイズニングに対する根本的な解決法である一方, 上記のような問題点も多く存在する. そこで本論文では DNSSEC を利用せず DNS 応答の検証を行うシステムを提案する.

2.3 DNS over TLS

DNS over TLS (DoT)とは DNS の通信内容の秘匿化を目的としたプロトコルのことを指す. TLS (Transport Layer Security)を用いて暗号化された DNS の通信を行うことで, DNS 通信内容の盗聴や改竄を防ぎ, サーバ証明書を用いることで通信相手の正当性を検証することができる. 現在 DoT はスタブリゾルバ（クライアント）とフルサービスリゾルバ間の通信の秘匿化のために RFC7858 として標準化されている.

DoT は Google Public DNS, Cloudflare, Quad9 などの主要パブリック DNS キャッシュサーバで実装済みであり, BIND や Unbound などの主要 DNS サーバも DoT に対応している[8].

3. 関連研究

3.1 複数フルサービスリゾルバへの並行名前解決

DNS キャッシュポイズニングの脅威が拡大する一方, DNS キャッシュポイズニングの根本的な解決法である DNSSEC には様々な問題が存在する. これらの問題の多くは DNSSEC の仕様により発生しているため, DNSSEC を導入した状態での解決が難しい.

そこでこの問題を解決するために, 先行研究では複数のフルサービスリゾルバへ並行に DNS 問い合わせを送信し各応答から信頼性の高いリソースレコードを抽出する手法を提案している[9]. 一般に名前解決を行う際にクライアントは対応する IP アドレスを一つ得ることができれば対象ドメインの目的のサービスにアクセスすることができる. DNS の応答は DNS ラウンドロビンにより複数の A レコードを含む場合がある. そこでこの手法ではクライアントが同時に異なるフルサービスリゾルバへ同一の問い合わせを行い, 全応答に共通して存在する A レコードを信頼できるリソースレコードとして抽出する. また全応答に共通して存在する A レコードが存在しない場合はポップアップ形式でクライアントに警告を行う. この手法に対して偽装したリソースレコードを利用させる場合, 攻撃者は複数のフルサービスリゾルバへ同時に DNS キャッシュポイズニングを行う必要がある. つまりこの手法を用いることで偽装したリソースレコードを利用してしまう確率の低下を期待できる. またこの手法では DNSSEC を用いないため遅延や負荷を抑えつつ DNS 応答の検証を行うことができる.

3.2 システムの動作設計

本システムでは平行に問い合わせを行うフルサービスリゾルバを Google Public DNS(8.8.8.8)と Cloudflare のパブリック DNS(1.1.1.1)を用いる. 本システムの動作の概要図を以下の図 2 に示す.

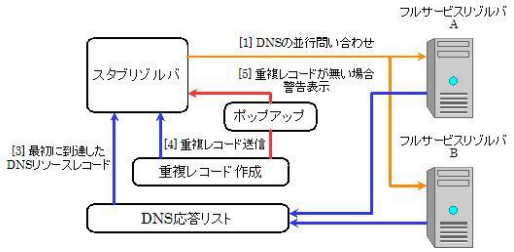

図 2 提案手法の動作の概要図

Step.1 同内容の名前解決要求を 8.8.8.8, 1.1.1.1 の 2 サーバへ送信する. この時名前解決要求をマルチスレッド化することで並行に問い合わせを行う. (図 2 [1])

Step.2 各フルサービスリゾルバからの DNS 応答を DNS 応答リストに追加する. (図 2 [2])

Step.3 名前解決におけるクライアントから見た応答速度を維持するため，Step.2 で最初に到達した応答のリソースレコードをクライアントへ送信する．(図 2 [3])

Step.4 各フルサービスリゾルバからの全応答が到着した後，全応答に共通して含まれるリソースレコードを抽出する．

Step.5 全応答に共通して含まれるリソースレコードが存在する場合，得られたリソースレコードを信頼できるものであるとして再度クライアントへ送信する．(図 2 [4])

Step.6 全応答に共通して含まれるリソースレコードが存在しない場合，クライアントが利用しているリソースレコードが危険なリソースレコードであるという内容の警告をポップアップ形式で表示する．

このような流れで信頼できる DNS 応答の抽出を行う．本システムでは 1 つのフルサービスリゾルバに偽装リソースレコードが登録されている場合必ず警告を表示することができる．一方，DNS ラウンドロビンや権威サーバの設定によって正規のリソースレコードに対しても警告を表示してしまう場合がある．この性質から本システムでは重複するリソースレコードが存在しない場合，リソースレコードをブロックするのではなくクライアントへ警告を表示する．また警告表示にとどめ，リソースレコードを利用するかどうかの判断はクライアントに委ねることで，全フルサービスリゾルバへの問い合わせに対して応答が得られる前にリソースレコードをクライアントに送ることができる (Step.3)．これによってクライアントから見た名前解決の速度を落とすことなくリソースレコードの信頼性を検証することができる．また複数フルサービスリゾルバへ並行に問い合わせた後，最も早く得られた応答をクライアントは利用することができるため，単一のフルサービスリゾルバへの問い合わせた場合の名前解決より早い名前解決が期待できる．

3.3 システムの可用性

本章の関連研究の中でシステムを用いて複数ドメインの名前解決を行った結果を示す．名前解決を行っているドメインは“Alexa – Top sites”[10]のランキングトップ 500 ドメインと，“Cisco Popularity List”[11]で提供される 100 万ドメインの計 1000500 ドメインである．また名前解決に用いるフルサービスリゾルバは 8.8.8.8 (Google Public DNS) と 1.1.1.1 (Cloudflare) ただしこのドメインリストは CNAME による重複が含まれる可能性があるため，まず 1000500 ドメインに対し NS レコードの問い合わせを行い権威サーバの名前を収集し重複なしでリスト化を行っている．ここで得られた権威サーバの全ドメインに対しホスト名を”www”に変更したものに対して，システムを用いて名前解決を行う．

権威サーバのドメインはすべてで 337793 個得られた．このドメインに対し名前解決を行った結果を分類した表を以下の表 1 に示す．

表 1 収集したドメインの名前解決に対する動作結果

	エラーなし	エラーあり
信頼できるレコードが存在する	272010 (80.5%)	0 (0.0%)
信頼できるレコードが存在しない	18339 (5.4%)	47444 (14.0%)

表 1 から 80%以上のドメインで信頼できる IP アドレスを抽出できていることがわかる．次にエラーが発生したものについてエラー内容をフルサービスリゾルバ毎に分類した表を以下の表 2 に示す．

表 2 エラー内容の割合

	“1.1.1.1”	“8.8.8.8”
NXDOMAIN	42545 (89.7%)	42580 (89.7%)
NoAnswer	3585 (7.6%)	3586 (7.6%)
Timeout	525 (1.1%)	259 (0.5%)
NoNameservers	713 (1.5%)	371 (0.8%)

表 2 から，エラーが出力されたドメインの 9 割以上が，ドメインが存在しないことを示す NXDOMAIN や，応答が存在しないことを示す NoAnswer によるものであるとわかる．これは DNS 応答率向上のため権威サーバ名に対しホスト名”www”付与したことにより，存在しないドメインの名前解決要求を送信していることが原因として考えられる．また存在するドメインの名前空間は常に変化しているため，ドメインリストを取得してから問い合わせを行うまでの間にドメインが消失した可能性も考えられる．

一方，タイムアウトの原因に関してはドメインが存在しないこと以外に通信経路上の問題や，権威サーバをポップする際の遅延などが考えられる．名前解決が遅延しており応答を返す時間が必要となっていることによるタイムアウトの場合，フルサービスリゾルバへの名前解決要求を再度送信することが問題緩和策として考えられる．

一般に名前解決対象のドメインに関するレコードが存在し権威サーバが正常に動作している場合，DNS 通信においてエラーが表示されることは少ない．つまり表 1 においてエラーが発生しておらず信頼できるリソースレコードが存在しない場合，正常な応答に対し警告を表示してしまっていると考えられる．エラーが発生していないという条件下で信頼できるリソースレコードが存在しないドメインは 18339 個存在し，エラーが発生していないドメインの約 6.3%を占める．関連研究ではこの約 6.3%をシステムの擬陽性率として課題としていた．

4. 提案手法

4.1 DNS over TLS の拡張

2.3 節で述べた通り，DoT はクライアントとフルサービスリゾルバ間の通信の秘匿化を目的として標準化されている．そこで本研究ではまず権威サーバにサーバ証明書を適用し，権威サーバで TLS 通信が行えるように DoT を拡張する. クライアントと権威サーバ間で DoT を介した通信を行い，クライアント上で DNS 権威サーバの正当性を検証することで, DNS 権威サーバから得られるリソースレコードの完全性を保証するシステムを構築する．

第 3 章の関連研究では抽出したリソースレコードの擬陽性率の高さが課題として挙げられていた．これは複数のフルサービスリゾルバから得られるリソースレコードに重複して A レコードが存在するドメイン数に依存する．この問題に対して，権威サーバのサーバ証明書の検証を行うことで重複したレコードが存在しない場合でもレコードの正当性を保証する．本研究の提案手法ではまず 3.1 節のシステムを用いて NS レコードを問い合わせ，再度 3.1 節のシステムを用いて得られた NS レコードの A レコードを問い合わせる. 得られた権威サーバに対し DoT を用いてクライアントが目的のドメインの A レコードを問い合わせることで，権威サーバのサーバ証明書の検証を介してレコードの完全性を保証する．以上の手法を用いて関連研究で課題となっていた擬陽性率の低下を行う．また，DNSSEC を用いずに信頼できるレコードの抽出を行う軽量な手法の提案を行う．

今回の研究では DNS 権威サーバに DoT の機能を組み込んだ DNS 名前解決の環境と DNSSEC 検証ができる DNS 名前解決環境をローカルなネットワーク上に作成する．用意した各環境で名前解決を行うことで提案手法と DNSSEC に関して名前解決における速度や負荷について比較を行う．

4.2 システムの動作設計

ここで本システムの全体の流れを示す．ここでは名前解決を行うドメインをd，用意する 2 つのフルサービスリゾルバをA, B，ドメインdをゾーンに持つ DNS 権威サーバをDとする．またクライアントは事前にフルサービスリゾルバA, Bと DoT で用いる TLS のコネクションを確立しているものとする．

Step.1 フルサービスリゾルバA, Bに対し，ドメインdの NS レコードを問い合わせる DNS 要求を送る．この時事前に確立している TLS のコネクションを用いて DoT を用いた DNS 通信を行う．またマルチスレッドを用いて同時にクエリを送信する．

Step.2 フルサービスリゾルバA, Bから得られた DNS 応答に重複する NS レコードを抽出する．この時重複する NS レコードが存在しない場合，先に得られた DNS 応答に含まれる NS レコードを参照する．

Step.3 得られた NS レコードに対する A レコードを問い合わせる DNS 要求をフルサービスリゾルバAに送信する. この時 Step.1 と同様にあらかじめ用意した TLS コネクションを用いて DoT による通信を行う．

Step.4 フルサービスリゾルバAの DNS 応答から得られた A レコードの IP アドレス(DNS 権威サーバD)に対してドメインdの A レコードを問い合わせる DNS 要求を送信する．この時新たに TLS のコネクションを確立し DoT を用いて DNS 通信を行う．

Step.5 Step.4 の TLS ネゴシエーションの際に DNS 権威サーバDのサーバ証明書を検証し DNS 権威サーバDの正当性が保証されれば得られたドメインdに対応する A レコードを信頼する．

以上の手順で信頼できるリソースレコードの抽出を行う．

5. 動作確認

5.1 提案手法の処理時間比較

DNSSEC と提案手法を比較するにあたり, DNS の名前環境(フルサービスリゾルバ・DNS 権威サーバ)はクラウド(Amazon Web Services)上に構築したローカルネットワーク内に用意する．処理時間を比較するうえでフルサービスリゾルバにレコードをキャッシュさせないため，ドメインごとのラベルが異なるものになるよう 10000 ドメインを用意する．用意したドメイン www.c0000.b0000.a0000 ~ www.c9999.b9999.a9999 の計 10000 ドメインの名前解決を行い，DNSSEC と提案手法の名前解決までの速度比較を行う．権威サーバは 4 台用意し，それぞれルートサーバから 3rd レベルドメインまでの各階層のドメインを管理する．この時各権威サーバには自身が管理するゾーンとは異なるドメイン名を割り当て，用意したドメインは全て外部ドメインとする．こうすることでドメインの NS レコードを問い合わせた際の応答に得られた NS レコードに対応する A レコードを付与させないことができる．ドメイン www.c0000.b0000.a0000（以下，対象 FQDN とする）の名前解決を行う際，処理の中で速度を計測する区間について以下に示す．ドメイン対象 FQDN を管理する権威サーバの FQDN を ns とする. またプロトコルを明示していない通信に関しては全て UDP を用いるものとする．

DNSSEC の処理時間計測区間

1. スタブリゾルバがフルサービスリゾルバへ対象 FQDN の A レコードを問い合わせる．
2. フルサービスリゾルバは各 DNS 権威サーバへ再帰的に対象 FQDN の A レコードを問い合わせる．
3. Step.2 と並行して DNSSEC を用いて得られたレコードの検証を行う．
4. DNSSEC の検証に成功し得られた対象 FQDN の A レコ

ードをフルサービスリゾルバからスタブリゾルバへ送信する．

5. スタブリゾルバが対象 FQDN の A レコードを得られれば計測終了．

提案手法の処理時間計測区間

1. スタブリゾルバがフルサービスリゾルバへ対象 FQDN の NS レコードを問い合わせる．この時あらかじめ用意しておいた TLS コネクションを用いて DoT による通信を行う．
2. フルサービスリゾルバは各 DNS 権威サーバへ再帰的に対象 FQDN の NS レコードを問い合わせる．
3. フルサービスリゾルバは得られた対象 FQDN の NS レコードをスタブリゾルバへ送信する．この時 DoT を用いて通信を行う．
4. スタブリゾルバが NS レコードとして得られたドメイン ns の A レコードをフルサービスリゾルバへ問い合わせる．この時 Step.1 で用いた TLS コネクションを用いて DoT による通信を行う．
5. フルサービスリゾルバは各 DNS 権威サーバへ再帰的にドメイン ns の A レコードを問い合わせる．
6. 得られたドメイン ns の A レコードをフルサービスリゾルバがスタブリゾルバへ送信する．この時 DoT を用いて通信を行う．
7. ns に対してスタブリゾルバが TLS のコネクションを確立する．
8. スタブリゾルバが対象 FQDN の A レコードを ns に問い合わせる．この時 Step.7 で確立した TLS コネクションを用いて DoT により通信を行う．
9. ns が対象 FQDN の A レコードをスタブリゾルバへ送信する．この時 DoT を用いて通信を行う．
10. スタブリゾルバが対象 FQDN の A レコードを得られれば計測終了．

DNSSEC と提案手法に関して 10000 ドメインの名前解決を順に行い，上記の計測区間について処理時間を計測する．10000 ドメインについて各ゾーンの権威サーバに登録するレコード数は A レコード・NS レコード共に 4 つずつとする．また本実験ではフルサービスリゾルバと DNS 権威サーバ間の物理的な距離やネットワーク遅延の影響を調査するため，各 DNS サーバ上でパケット送出時に遅延を持たせる．遅延時間は 10~100ms まで 10ms ずつ変化させて計測を繰り返す．DNSSEC と提案手法について処理時間を比較したグラフを図 3 に示す．

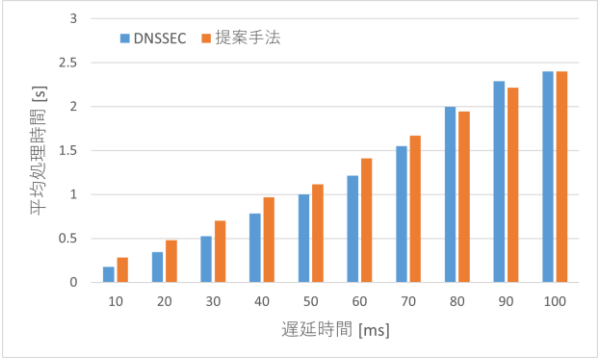

図 3　DNSSEC と提案手法の遅延時間別平均処理時間

図 3 を見ると遅延時間が小さい場合，提案手法より DNSSEC の方が処理時間は短く，80, 90ms など遅延が大きい場合に逆転する結果が得られた．

図 3 の結果について DNSSEC と提案手法の平均処理時間の有意差を比較するため t 検定を行う．有意差は t 検定における p 値を用いて判定し，有意水準は 0.05 とする．平均処理時間に関して提案手法が DNSSEC に対して有意に大きい場合”×”，小さい場合は”○”として示す．また平均処理時間に関して有意差が認められない場合は”-“とする．t 検定を行う際の有意差の有無についてまとめた表を以下の表 3 に示す．

表 3　DNSSEC と提案手法の処理時間の有意差

遅延時間[ms]	有意差
10	×
20	×
30	×
40	×
50	×
60	×
70	×
80	○
90	○
100	-

表 3 を見ると遅延時間が 10~70ms の場合，レコード数に関係なく DNSSEC の方が提案手法より処理時間が短いことがわかる．一方遅延時間を大きくすると処理時間において提案手法が優れる場合がある．つまりフルサービスリゾルバと DNS 権威サーバの物理的な距離が離れている際に処理速度の面において提案手法が優位に立つ場合がある．

5.2 処理時間の統計的ばらつき

DNS では一般に UDP を用いて通信が行われるためパケット損失の発生を検知することができない．そのため OS

によっては設定時間以内に DNS 応答が得られない場合 DNS 要求の再送を行う．このタイムアウトの時間設定を正確に行うことで素早い DNS 要求の再送を行うことができ，DNS の処理時間の高速化につながる．したがって名前解決時間の予測の容易さが DNS の品質に影響するため，ここでは DNSSEC と提案手法の処理時間に関してばらつきを調査した．パケットロスにより DNS 応答が得られなかったクエリを除いたものに関して標準偏差を比較したグラフを以下の図 4 に示す．

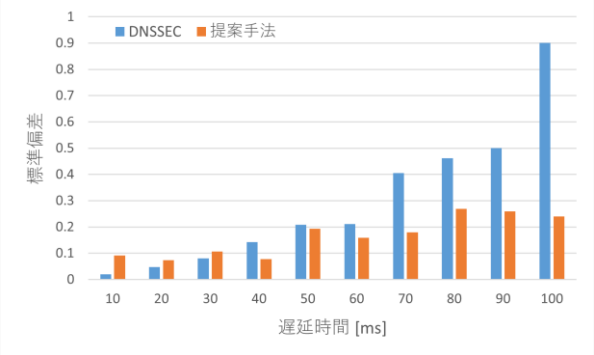

図 4　DNSSEC と提案手法の処理時間の標準偏差

図 4 を見ると遅延時間が 10~30ms では DNSSEC の方が提案手法よりもばらつきが小さいことがわかる．一方遅延時間が 40~100ms では提案手法の方が DNSSEC よりばらつきが小さい．また DNSSEC は遅延時間を大きくするにつれて標準偏差も大きくなる一方，提案手法では遅延時間を大きくしても標準偏差は安定していることが確認できる．

5.3 フルサービスリゾルバへの負荷検証

DNS は仕組み上フルサービスリゾルバへアクセスが集中する．そのためフルサービスリゾルバにはスケーラビリティが要求されており，DNSSEC ではフルサービスリゾルバへの負荷が問題となっていた．ここでは DNSSEC と提案手法のフルサービスリゾルバへの負荷を検証する．

負荷の検証には dnsperf というパフォーマンス検証ツールを用いる．負荷実験ではフルサービスリゾルバへ用意した 10000 ドメインの A レコードの名前解決要求を送信し，1 秒当たりに完了するクエリ数(QPS)とタイムアウトの割合を調査する．ここの実験ではネットワークに遅延はかけず，タイムアウトの時間は 2 秒とし，パケット送信時のクライアント数は 5 とする．また負荷の程度による QPS の比較を行うため未応答のクエリ数を変化させて実験を行う．未応答クエリ数は 10~100 の範囲で 10 ずつ変化させ，それぞれについて 10000 クエリの送出を 10 回行い，QPS とタイムアウト率の平均を求める．QPS について比較したグラフを以下の図 5，タイムアウトの割合について比較したグラフを図 6 に示す．

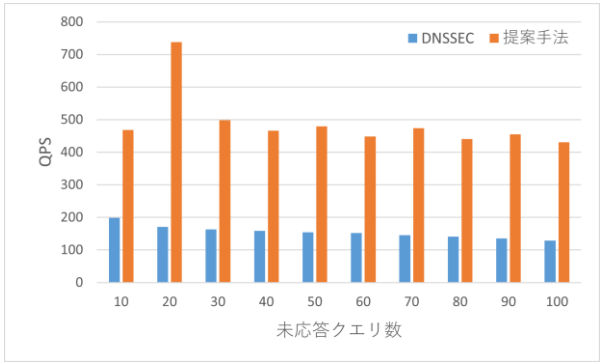

図 5　DNSSEC と提案手法の Query per Second

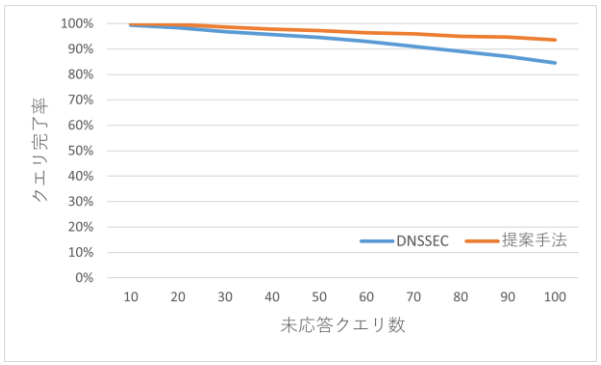

図 6　DNSSEC と提案手法のクエリ完了数の割合

図 5 を見ると負荷に関係なく提案手法の方が DNSSEC よりも高い QPS を維持していることが確認できる．つまり大量のクエリがフルサービスリゾルバへ到達する環境において，DNSSEC を利用するよりも提案手法を利用する方が軽量に動作することがわかる．次に図 6 を見ると負荷を変化させてもタイムアウトの割合は提案手法の方が DNSSEC よりも小さいことがわかる．また負荷を増やすとクエリの完了率は下がるが，提案手法の方が DNSSEC と比較して下がり方が緩やかであることがわかる．

5.4 考察

本研究では DNSSEC の問題点を補う新たな DNS 応答の検証手法を提案した． DNSSEC と比較した本手法の特徴について以下に示す．

動作の軽量さ

本実験では DNSSEC と比較して提案手法は 1 クエリの名前解決に関して時間を必要としていた．しかしフルサービスリゾルバに対しより実環境に近い負荷をかけた状態では 2~4 倍軽量に動作する様子が伺えた．

名前解決時間の安定性

提案手法では 1 クエリの名前解決時間に関してばらつきが小さく，DNSSEC と比較して処理時間が安定していることが判明した．DNSSEC と比較して提案手法では速度のば

らつきによる DNS 要求の再送や DNS タイムアウトの頻度を減らすことができる．また本手法では TCP 通信を用いているためパケットロスによる DNS タイムアウトも発生しにくい．

普及の容易さ

DNSSEC では完全展開を行うためにすべての権威サーバの協力が必要であったが，提案手法ではコンテンツサーバのドメインを管理する権威サーバのみの協力を得られれば良く，DNSSEC と比較して普及が容易である．

セキュリティの向上

提案手法に対して偽陰性を発生させる場合，まず攻撃者は複数フルサービスリゾルバへの同時 DNS キャッシュポイズニングを成功させる必要がある．また，攻撃対象のドメインを管理する全ての権威サーバの A レコードを偽装する必要がある．さらに偽装した権威サーバに適用するサーバ証明書の取得も必要であり，悪意あるサービスへ誘導する難度の上昇が期待される．

提案手法には以上の利点が存在し，現状の DoT に拡張を行うことで，本システムによりクライアントの手元で信頼性の高いリソースレコードの抽出を実現していることがわかる．

今回の実験では用意した 10000 ドメインはすべてが外部ドメインであるとして設計を行ったが，実環境においては内部ドメインも存在する．名前解決を行うドメインが内部ドメインである場合，ドメインの NS レコードを問い合わせると DNS 応答の ADDITIONAL SECTION に NS レコードに対応する A レコードが付加される．つまり内部ドメインである場合，提案手法で必要である 2 往復の DNS 通信が 1 往復で完了させることができる．そこで Cisco の Top 100 万ドメイン[11]に対して内部ドメインの割合を調査したところ, 101049 ドメイン(10.1%)が内部ドメインであることがわかった．したがって提案手法に関して名前解決の約 10%においては本実験より高速に動作することが期待できる．

6. おわりに

6.1 今後の課題

今回の実験では信頼の連鎖を持つ DNSSEC 環境をローカルに構築する必要があり，DoT を権威サーバに適用する際に自己署名証明書を用いた．実験ではサーバ証明書の検証を行っておらず，クライアント上の負荷について調査することができなかった．つまり今後の課題としてサーバ証明書検証によるクライアントの負荷調査と，サーバ証明書の検証手法の検討が挙げられる．

またその権威サーバの DoT 対応状況をクライアントに通知する方法も確立されていない．DoT に対応していない場合通常の UDP 通信にロールバックするような仕組みでは中間者攻撃を受ける可能性が存在するため適切な手法を確立する必要がある．

参考文献

[1] Amir Herzberg, Haya Shulman. “Towards Adoption of DNSSEC : Availability and Security Challenges”. IACR Cryptology e Print Archive 2013.
[2] Dan Kaminsky. “It’s The End Of The Cache As We Know It. Black Hat conference”. 入手先 <https://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf> (参照 2022-01-30).
[3] “新たなる DNS キャッシュポイズニングの脅威～カミンスキー・アタックの出現～”. 入手先 <https://jprs.jp/related-info/guide/topics-column/no9.html#a4> (参照 2022-01-30).
[4] “ICANN Research - TLD DNSSEC Report”. 入手先 <http://stats.research.icann.org/dns/tld_report/index.html> (参照 2022-01-30).
[5] “DNSSEC Validation Rate by country (%)”. 入手先 https://stats.labs.apnic.net/dnssec, (参照 2022-01-30).
[6] A. Herzberg and H. Shulman. "Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org," 2013 IEEE Conference on Communications and Network Security (CNS), National Harbor, MD, 2013, p. 224-232, DOI: 10.1109/CNS.2013.6682711.
[7] 太田 健也, 鈴木 恒彦. “DNS 第一フラグメント便乗攻撃の追検証と対策の検討”. 情報処理学会 第 81 回全国大会講演論文集 2019(1) p. 443-444
[8] “DNS Privacy Project – Public Resolvers”, 入手先<https://dnsprivacy.org/public_resolvers/> (参照 2022-01-30)
[9] 阿久津 賢宏, 山井 成良, 金 勇, “複数 DNS キャッシュサーバへの並行名前解決による DNS キャッシュポイズニング対策手法”, 情報処理学会 マルチメディア，分散，協調とモバイル(DICOMO2020)シンポジウム，6 月 2020, pp.625-632
[10] “Alexa – Top sites”. 入手先 <https://www.alexa.com/topsites> (参照 2020-01-20)
[11] “Cisco Popularity List”. 入手先 <http://s3-us-west-1.amazonaws.com/umbrella-static/index.html> (参照 2021-10-30).