あなたの知らない
DNS権威サーバの世界

山本和彦
(株)インターネットイニティアティブ

おしながき

- 自己紹介
- DNSの仕組み
- DNS権威サーバの振る舞い
- DNSSECの仕組み
- 実装の体験談

自己紹介

山本和彦

- (株) インターネットイニシアティブ (IIJ)
  - 技術研究所 技術開発室 室長
- 九州大学工学部情報工学科修了
  - (論文)博士 (情報科学)
- 2022年より介護のため山口県へ移住
  - 契約社員
  - 完全なリモートワーク
- 4児の父
- e-mail: kazu@iij.ad.jp
- 個人のホームページ: https://mew.org/~kazu/

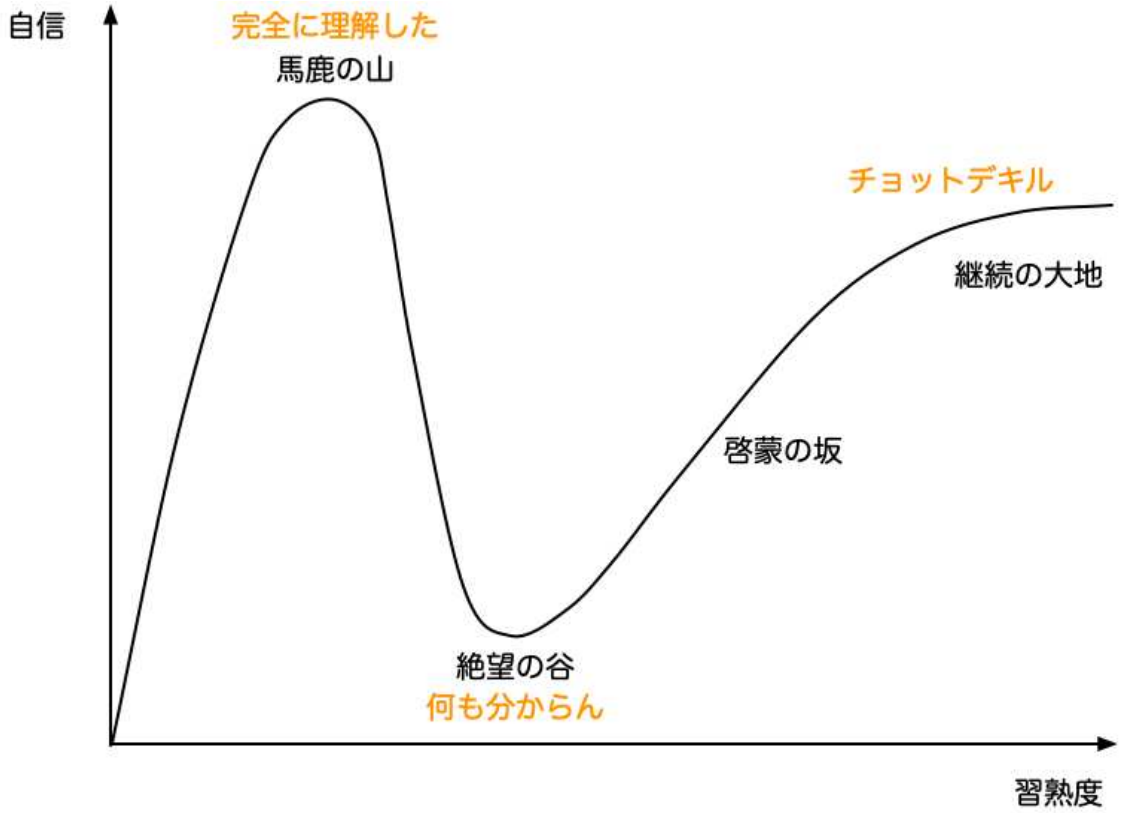
職務

- プロトコルの標準化へ貢献すること
  - IETF(Internet Engineering Task Force)で
    議論されているプロトコルを実装し
    他の実装と相互接続性を検証することで
    プロトコルの標準化へ貢献する
  - その活動を通じてIIJのプレゼンスを高める
- TLS 1.3: RFC 8846 の Contributors の節

  Kazu Yamamoto
  Internet Initiative Japan Inc.
  kazu@iij.ad.jp
- 最近はDNS関連

DNS 何も分からん

- ダニング・クルーガー曲線

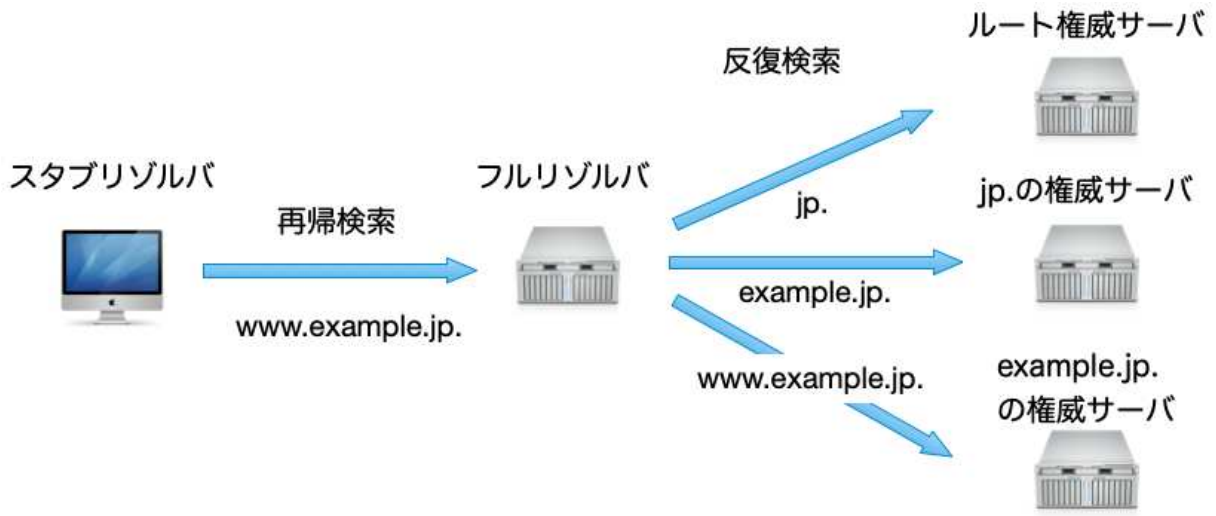


DNSの仕組み

Domain Name System

- 名前 → IPアドレス
- 広域分散データベース
- 権威サーバ(authoritative)
  - データを提供
- リゾルバ (resolver)
  - データを検索
  - 1) フルリゾルバ (キャッシュサーバ)
  - 2) スタブリゾルバ

DNSの構成



- フルリゾルバのキャッシュ
  - 応答速度の向上
  - 無駄なトラフィックの抑制
    - キャッシュ機能がないとフルリゾルバは増幅器になってしまう

反復検索

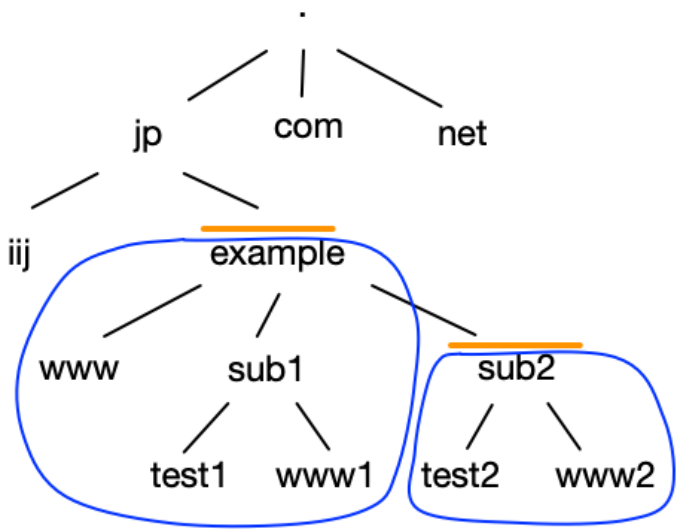
- 問い合わせ名 (QNAME)
  - 昔は常に最終的な問い合わせ名
    - www.example.jp.
  - 最近はQNAME minimisation
    - プライバシ強化のため
    - RFC 9156: "DNS Query Name Minimisation to Improve Privacy"
    - jp. → example.jp. → www.example.jp.
- ルートの権威サーバ
  - 委任情報：`jp.  NS ns.jp.`
  - グルー　：`ns.jp  A  192.0.2.1`
- jp. の権威サーバ
  - 委任情報：`example.jp.  NS ns.example.jp.`
  - グルー　：`ns.example.jp  A  198.51.100.1`
- example.jp. の権威サーバ
  - 答え：　`www.example.jp A  203.0.113.1`

ドメインとゾーン

- ドメイン
  - 木構造をなす名前、その内のある階層、部分木
- ゾーン
  - 管理権限(所有)の範囲
  - ゾーンカットで切り取られた木構造の一部
- RFC 9499
  - DNS Terminology
  - 上記の用語の説明はイマイチ

ゾーンカットとゾーン

- RFC 1034
  - DOMAIN NAMES - CONCEPTS AND FACILITIES



- フラットではないゾーンの身近な例
  - jp. と ac.jp. は同じゾーン

NS RR とゾーンカット

- NS RR(Resource Record)でゾーンをカットする
- ルートの権威サーバ

```
.               SOA ...
                NS  a.root-servers.net.
jp.             NS  ns.jp.          ;; 委任 (所有しない)
ns.jp.          A   192.0.2.1       ;; グルー(所有しない)
```

- jp. の権威サーバ

```
jp.             SOA ...
                NS  ns.jp.          ;; 所有
                A   192.0.2.1
example.jp.     NS  ns.example.jp.  ;; 委任
ns.example.jp.  A   198.51.100.1    ;; グルー
```

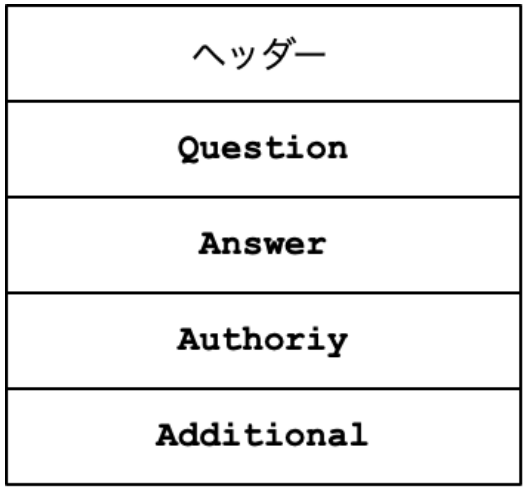
- ゾーンデータはSOAで開始
  - Start Of a zone of Authrority

今日のテーマ

- 今日は権威サーバについて説明
  - Haskell で clove という権威サーバを実装中
- フルリゾルバに関しては以下を参照
  - 「DNSフルリゾルバbowlineの設計と実装」，
    Internet Infrastructure Review（IIR）Vol.68
- スタブリゾルバに関しては以下を参照
  - 「DNS検索コマンド dugの紹介」，IIJ Engineers Blog

DNS権威サーバの振る舞い

DNS質問・応答の書式



問い合わせ

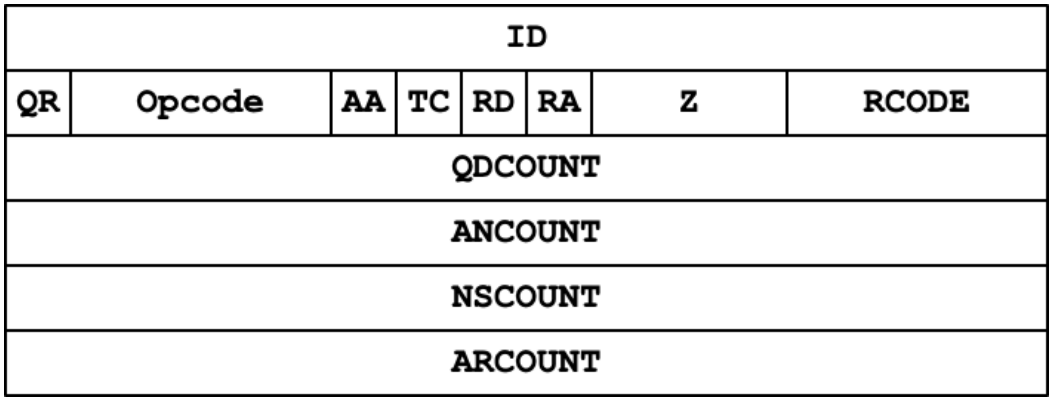
答え

委任情報(NS)・SOA

グルー(A, AAAA)

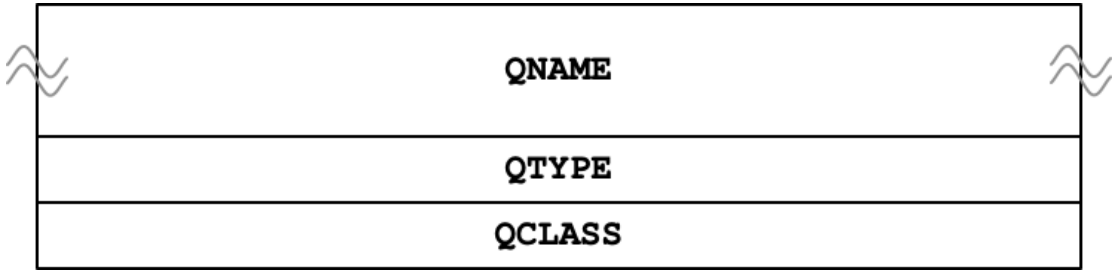
- RFC 1035
  - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

ヘッダー



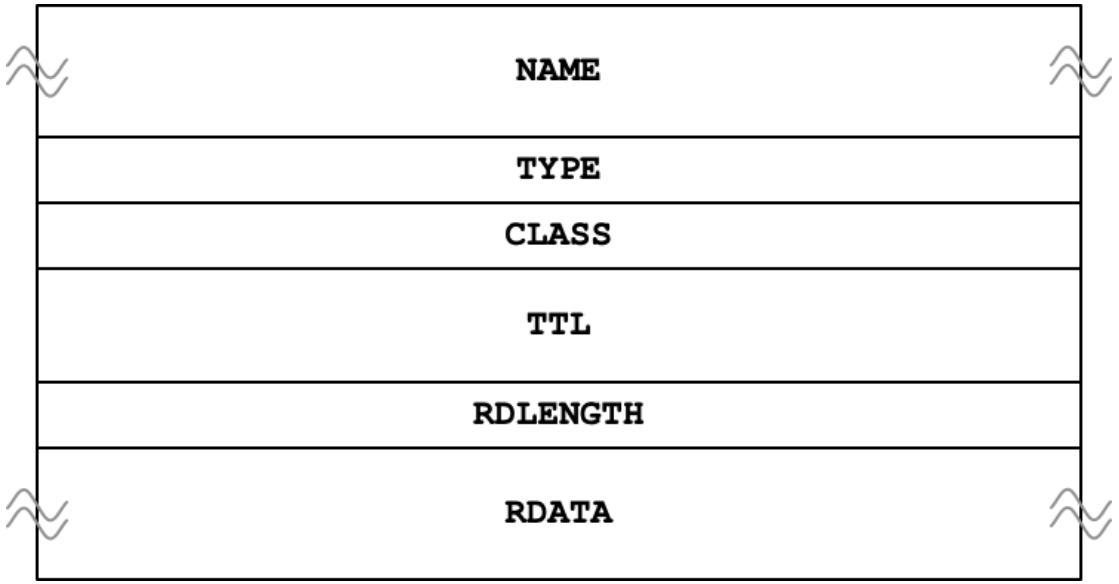
- ID
  - 質問と応答を付き合わせるためのタグ (UDPでの通信が基本)
- QR
  - 質問のとき 0、応答のとき 1
- RD (Recursion Desired)
  - 再帰検索のとき 1、反復検索のとき 0
- AA (Authoritative Answer)
  - 自分が所有している場合は1、(委任など)そうでなければ 0

Question セクション



- RFC 9619
  - In the DNS, QDCOUNT Is (Usually) One

その他解答用のセクション



- TTL (Time To Live)
  - フルリゾルバがキャッシュする時間 (秒)
- Type, Length, Value 形式

エラー応答

- 管理外の問い合わせを受けた場合
  - RCODE = REFUSED

- 例

```
% dig @dns0.iij.ad.jp. www.exmaple.com
;; ->>HEADER<<- opcode: QUERY, status: REFUSED
```

肯定応答

- 管理下で答えが存在する場合
  - RCODE = NOERROR
  - Answer セクションに答えを入れる
  - AAフラグはオン
- 例

```
% dig @dns0.iij.ad.jp. www.iij.ad.jp.
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; ANSWER SECTION:
www.iij.ad.jp.  300 IN A 202.232.2.180
```

委任応答

- 下位のゾーンへ委任している場合
  - RCODE = NOERROR
  - Authority セクションに NS RR を入れる
  - Additional セクションに in-domain の A RR / AAAA RR を入れる
  - AAフラグはオフ
- 例

```
% dig @a.dns.jp. www.iij.ad.jp.
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; AUTHORITY SECTION:
iij.ad.jp.      86400 IN NS   dns0.iij.ad.jp.
;; ADDITIONAL SECTION:
dns0.iij.ad.jp. 86400 IN A    210.130.0.5
dns0.iij.ad.jp. 86400 IN AAAA 2001:240::105
```

グルーと in-domain

- in-domain
  - 自分のサブドメインに相当する名前
- 例： org. から example.org. への委任
  - ns.example.org. は in-domain
  - ns.example.jp. は unrelated
- NS の値が in-domain の場合は、必ずグルーを返す
  - そうしないと名前は解決できない
  - 現在のグルーは A か AAAA
- RFC 9471
  - DNS Glue Requirements in Referral Responses

NODATA 否定応答

- 名前は存在するがタイプがない場合
  - NODATA と呼ばれる (疑似 RCODE)
  - RCODE = NOERROR
  - Answer セクションが空
  - Authority セクションに SOA RR
  - AAフラグはオン
- 例

```
% dig www.iij.ad.jp txt
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; AUTHORITY SECTION:
iij.ad.jp. 928 IN SOA bh0.iij.ad.jp. \
  postmaster.iij.ad.jp. 1783872604 3600 \
  1800 1209600 3600
```

- RFC 2308
  - Negative Caching of DNS Queries (DNS NCACHE)
  - SOA RR の minimum をネガティブキャッシュのTTLとして再定義

NXDOMAIN 否定応答

- 名前が存在しない場合
  - RCODE = NXDOMAIN (Non-eXistent)
    - 昔は Name Error と呼ばれていた
  - Answer セクションが空
  - Authority セクションに SOA RR
  - AAフラグはオン
- 例

```
% dig example.jp.
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN
;; AUTHORITY SECTION:
jp.   710 IN SOA z.dns.jp. root.dns.jp. ...
```

- RFC8020
  - NXDOMAIN: There Really Is Nothing Underneath

ANY 問い合わせタイプ

- その名前に対するすべての RR を返すと危険
  - 増幅攻撃
- 実装方法の一つ
  - どれか一つだけ RR を返す
- RFC 8482
  - Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY
- 例 (後述の Empty Non Terminal ではないことが分かる)

```
% dig @a.dns.jp. ac.jp any
;; ANSWER SECTION:
ac.jp.   86400 IN TXT "ac.jp."
```

CNAME

- CNAME: 正式名
  - Canonical name
  - 例：`www.example.jp. CNAME alpha.example.jp.`
  - www の正式名は alpha
- 他のタイプと共存できない
  - RFC 1912
    - Common DNS Operational and Configuration Errors
  - RFC 2181
    - Clarifications to the DNS Specification
  - ただし、後述の RRSIG を除く
- 間違ったゾーンデータの例

```
www.example.jp.   CNAME alpha.example.jp.
                  A     192.0.2.1
alpha.example.jp. A     198.51.100.1
```

CNAME 応答

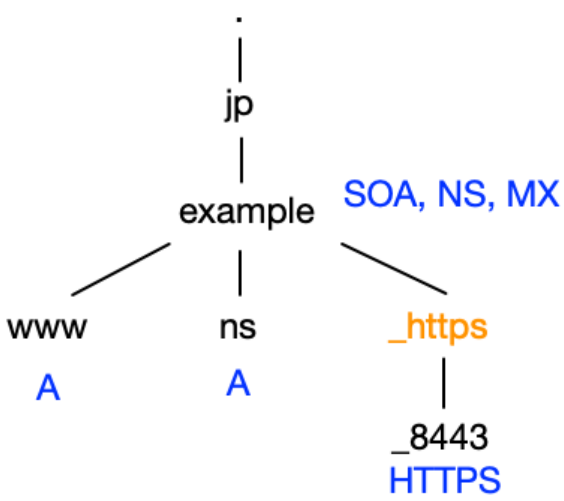
- 問い合わせタイプが CNAME の場合
  - Answer セクション：CNAME RR
- 問い合わせタイプが CNAME ではない場合
  - Answer セクション：CNAME RR
  - 正式名称が in-domain の場合、それを問い合わせ名にして再検索
    - 結果を合成
    - RCODE は再検索の結果
    - AAフラグは(参照先が委任でも)オン
- 例

```
% dig @sh3.iijlab.net www.iijlab.net. a

;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; ANSWER SECTION:
www.iijlab.net.  3600 IN CNAME sh3.iijlab.net.
sh3.iijlab.net.  3600 IN A     202.238.220.76
```

Empty Non Terminal

- 名前は存在するがタイプが付随しない
- 例
  - `_8443._https.example.jp. 7200 IN HTTPS 1 . alpn=h3`



Empty Non Terminalへの応答

- Empty Non Terminal に対する問い合わせへの答え
  - タイプがない場合の否定応答と同じ
  - NODATA
  - RCODE = NOERROR
  - Answer セクションが空
  - Authority セクションに SOA RR
- RFC 4592
  - The Role of Wildcards in the Domain Name System
  - 後述のワイルドカードを明確したRFC

ワイルドカード

- あるレベルで合致しなければ、"*" があるか調べる
  - 応答では "*" を問い合わせ名に置き換える
- ゾーンデータの例

```
example.           SOA   <SOA RDATA>
example.           NS    ns.example.com.
example.           NS    ns.example.net.
*.example.         TXT   "this is a wildcard"
*.example.         MX    10 host1.example.
sub.*.example.     TXT   "this is not a wildcard"
host1.example.     A     192.0.2.1
```

- 質問の例

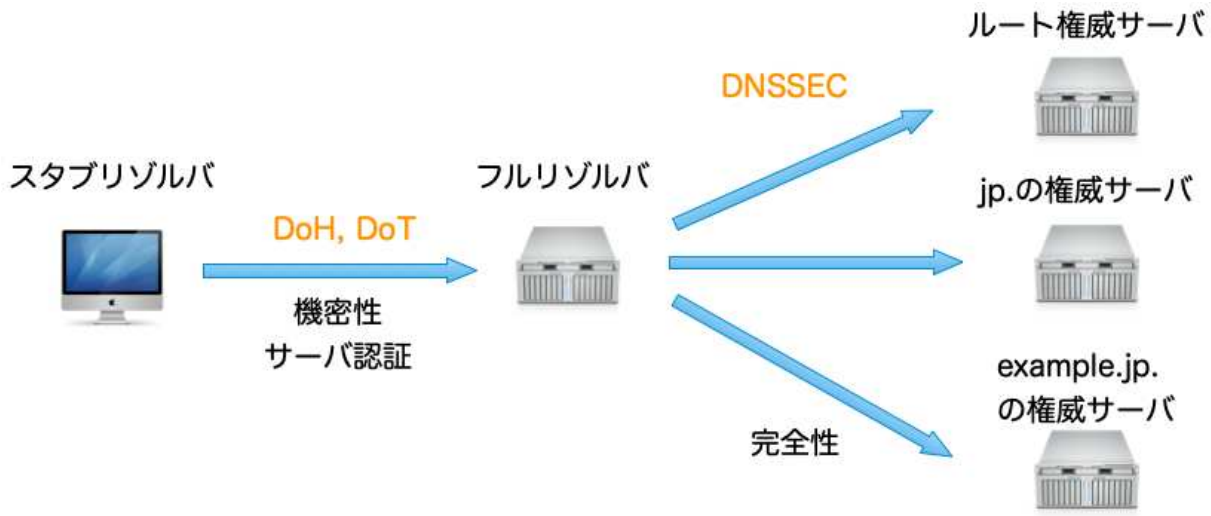
```
host1.example.   MX  → NODATA
host3.example.   MX  → MX    10 host1.example.
host3.example.   A   → NODATA
foo.bar.example. TXT → TXT   "this is a wildcard"
```

DNSSECの仕組み

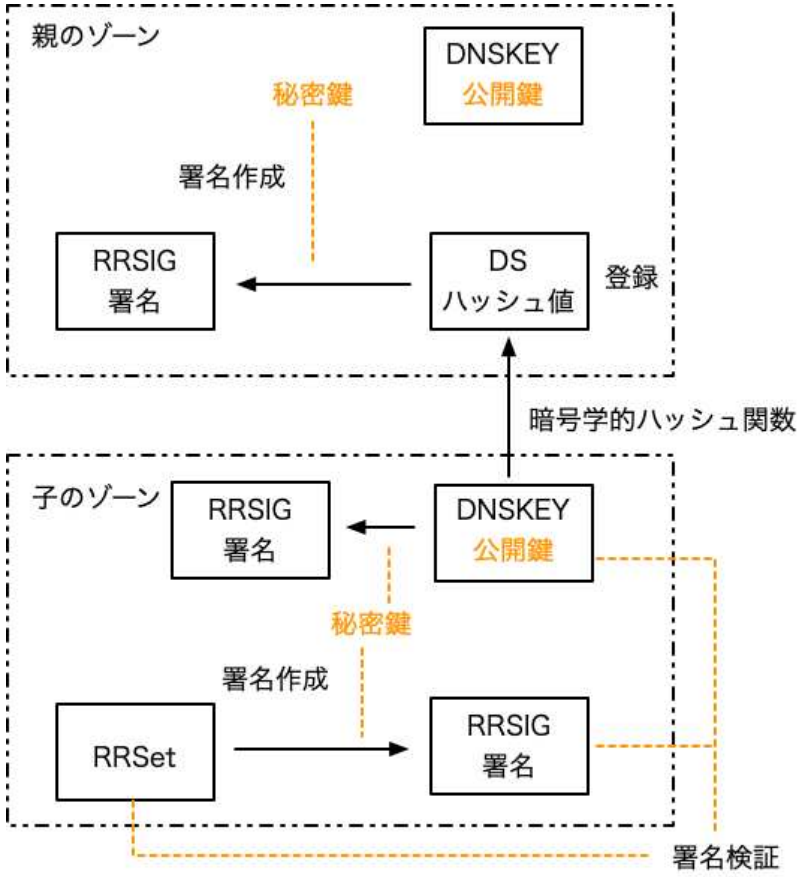
DNSSEC

- 目的
  - RRに署名を施して完全性を実現する
  - 改竄を検知する
- DNSSEC三兄弟
  - RFC4033
    - DNS Security Introduction and Requirements
  - RFC4034
    - Resource Records for the DNS Security Extensions
  - RFC4035
    - Protocol Modifications for the DNS Security Extensions

セキュリティが強化されたDNS



DNSSECのRR


親のゾーン
DNSKEY
公開鍵
秘密鍵
署名作成
RRSIG
署名
DS
ハッシュ値
登録
暗号学的ハッシュ関数
子のゾーン
RRSIG
署名
DNSKEY
公開鍵
秘密鍵
署名作成
RRSet
RRSIG
署名
署名検証

RRSIGの検証

- 親の権威サーバから委任情報を得る
  - フルリゾルバはDO(DNSSEC OK)フラグをオンにして問い合わせる
  - 「子のNS」と共に「子のDS」と「DSに対するRRSIG」を得る
  - 「DSに対するRRSIG」で検証し、子のDSを信頼する
- 子の権威サーバから
  - 「DNSKEY」と「DNSKEYに対するRRSIG」を得る
    - 自己署名
  - 「DS」および「DNSKEYに対するRRSIG」を検証し、DNSKEYを信頼する
  - A と「Aに対するRRSIG」を得る
  - 「Aに対するRRSIG」を検証し、A を信頼する
- 注意
  - DNSKEYをZSK(Zone Signing Key)とKSK(Key Signing Key)へ分けられることは、今日は説明の範囲外

DNSSECの肯定応答

- 管理下で答えが存在する場合
  - RCODE = NOERROR
  - Answer セクションに答え + RRSIG を入れる
- 例

```
% dig @dns0.iij.ad.jp. www.iij.ad.jp a +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; ANSWER SECTION:
www.iij.ad.jp. 300 IN A     202.232.2.180
www.iij.ad.jp. 300 IN RRSIG A 8 4 300 ...
```

DNSSECの委任応答

- 下位のゾーンへ委任している場合
  - RCODE = NOERROR
  - Authority セクションに NS RR を入れる
    - 下位のゾーンがDNSSEC対応の場合、DS RR + RRSIGを入れる
    - 下位のゾーンがDNSSEC未対応の場合、NSEC(後述)を入れてDSがないことを示す
  - Additional セクションに in-domain の A RR / AAAA RR を入れる
- 例

```
% dig @a.dns.jp iij.ad.jp ns +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; AUTHORITY SECTION:
iij.ad.jp.  86400 IN NS dns0.iij.ad.jp.
iij.ad.jp.  86400 IN NS dns1.iij.ad.jp.
iij.ad.jp.  7200  IN DS 45574 8 2 ...
iij.ad.jp.  7200  IN RRSIG DS 8 3 ...
;; ADDITIONAL SECTION:
dns0.iij.ad.jp.  86400 IN A    210.130.0.5
dns1.iij.ad.jp.  86400 IN A    210.130.1.5
dns0.iij.ad.jp.  86400 IN AAAA 2001:240::105
dns1.iij.ad.jp.  86400 IN AAAA 2001:240::115
```

DNSSECの否定応答

- 署名すべきRRSetがない！
- たとえば、SOAに署名することにする
  - SOAに対する RRSIG は静的
  - 悪者が一旦手に入れれば、偽の否認応答を生成できる
- 解決案：NSEC (Next Secure) RR
  - 名前の範囲を示す RR を作る
  - ドメイン名の正規順序を定義
    - bar.example.jp. の次は foo.example.jp. の場合
    - この範囲の NSEC RR は、bar の RR で、foo までの範囲は名前が存在しない
  - フィールド
    - 「存在する次の名前」
    - 「ある存在する名前」に対して存在するタイプ
    - ラベルの数 (ワイルドカード用)
- 解決案：NSEC3
  - 今日は説明の範囲外

DNSSEC NODATA

- 名前は存在するがタイプがない場合
  - RCODE = NOERROR
  - Answer セクションが空
  - Authority セクションに SOA RR + RRSIG
  - さらに、問い合わせ名に一致する NSEC ＋ RRSIG
    - NSEC で、そのタイプがないことを示す
- 例

```
% dig @a.root-servers.net. . txt +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NOERROR
;; AUTHORITY SECTION:
.   86400 IN SOA a.root-servers.net. ...
.   86400 IN RRSIG SOA ...
.   86400 IN NSEC aaa. NS SOA RRSIG ...
.   86400 IN RRSIG NSEC ...
```

DNSSEC NXDOMAIN

- 名前が存在しない場合
  - RCODE = NXDOMAIN
  - Answer セクションが空
  - Authority セクションに SOA RR + RRSIG
  - その名前をカバーする NSEC ＋ RRSIG
  - さらに、ワイルドカードをカバーする NSEC ＋ RRSIG
- 例

```
% dig @a.root-servers.net. c. a +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN
;; AUTHORITY SECTION:
.    86400 IN SOA   a.root-servers.net.  ...
.    86400 IN RRSIG SOA 8 0 ...
bzh. 86400 IN NSEC  ca. NS DS RRSIG NSEC
bzh. 86400 IN RRSIG NSEC 8 1 ...
;; 以下がワイルドカードをカバーする NSEC
.    86400 IN NSEC  aaa. NS SOA RRSIG NSEC DNSKEY ...
.    86400 IN RRSIG NSEC 8 0 ...
```

例外だらけのRR

- あるゾーンにおいて
  - 普通の RR
    - 所有しているのでRRSIGがある
    - NSECもある
  - Empty Non Terminal
    - 所有しているのにRRSIGはない
    - NSECもなし
  - 子への NS
    - 所有してないのでRRSIGはなし
    - NSECはある！
  - 子へのDS
    - 所有しているのでRRSIGがある
    - NSECもある
  - 子へのグルー (A,AAAA)
    - 所有してないのでRRSIGはなし
    - NSECはなし

体験談

Request For Comments

- クイズ：DNS関係で何個のRFC が出てきたか？
- 答え：14個
- DNSに関するRFCはカオス
  - 紹介してないRFCも多数
- RFCが読みきれないので、AIに質問して理解した
  - 「DNSのauthority sectionに入る可能性のあるRRは何ですか？」
  - 「○○○を定めているのは、どのRFCですか？」
- AIは、DNSに対しては正確に答える
- DNSSECに対しては間違うこともある
  - 同じ質問をしても真逆の答え

データ構造

- 最初は一段の key value store (辞書)として実装
  - DB -- (ドメイン, タイプ) -> RR
  - 問題：Empty Non Terminal が実現できない
- 二段の key value store へ変更
  - DB -- (ドメイン) -> 内部 DB -- (タイプ) -> RR
  - 問題：ワイルドカードを完全には表現できない
- ユニットテストの強化
  - CNAME
  - Emtpy Non Terminal
  - ワイルドカード
  - DNSSEC
- PATRICIA tree を採用
  - ドメインの木構造を素直に表現
  - テストのおかげで、ほぼ一日で実現した

フィールドテスト

- mew.org でフィールドテスト
  - 最初は DNSSEC なし
  - 権威サーバとしての振る舞い
  - セカンダリへのゾーン転送
- DNSSEC のフィールドテスト
  - org. に mew.org. の DS を登録してない状態でテスト
  - mew.org. の DS は、テストサイトに手で入力
  - https://dnscheck.jp/
  - https://zonemaster.net/