HTTPS RRの現状と今後

山口崇徳@IIJ

DNS Summer Day 2024

HTTPS RRの現状

HTTPS RRってなに?

- 基本的な話は省略します

- このへん見てください → https://eng-blog.iij.ad.jp/archives/23963

RFCになりました

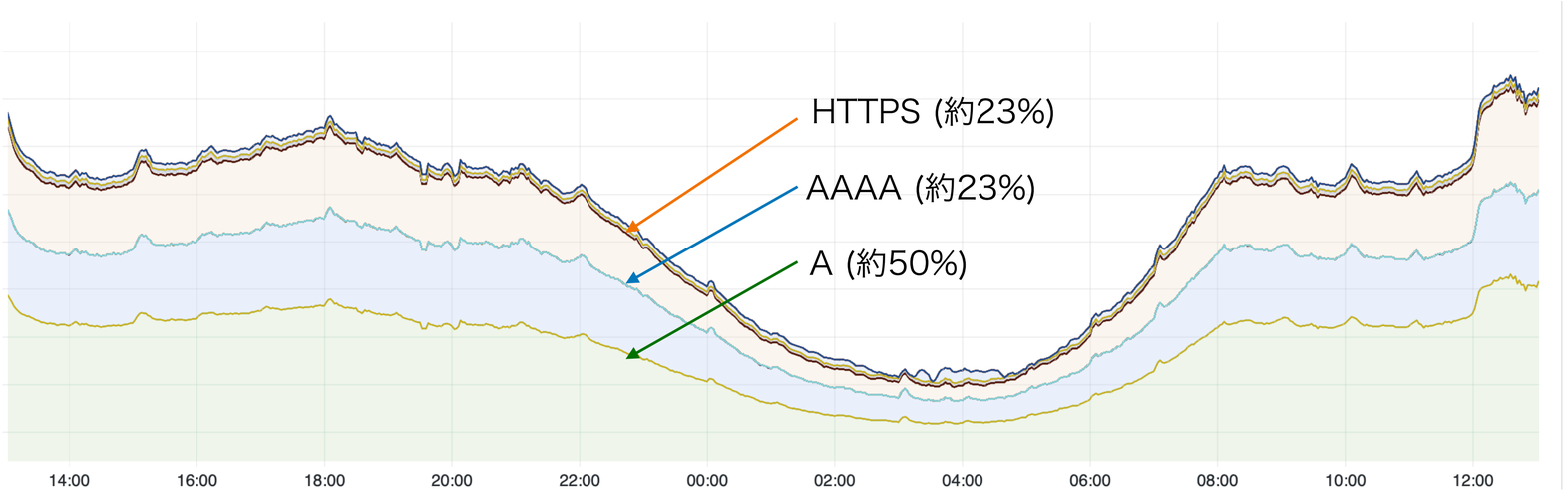
- 2023/11 draft-ietf-dnsop-svcb-httpsがRFC9460として標準化
  - https://www.rfc-editor.org/rfc/rfc9460.html
- もともとのドラフトにはTLS 1.3 Encrypted Client Hello拡張(draft-ietf-tls-esni)に関するサービスパラメータを含んでいたが、RFC9460では削除された
  - ECHの議論が長引いて終わりが見えない状態だったため
  - 遅れて3月末にWG Last Callが終わったので、ECHも近いうちにRFCになりそう

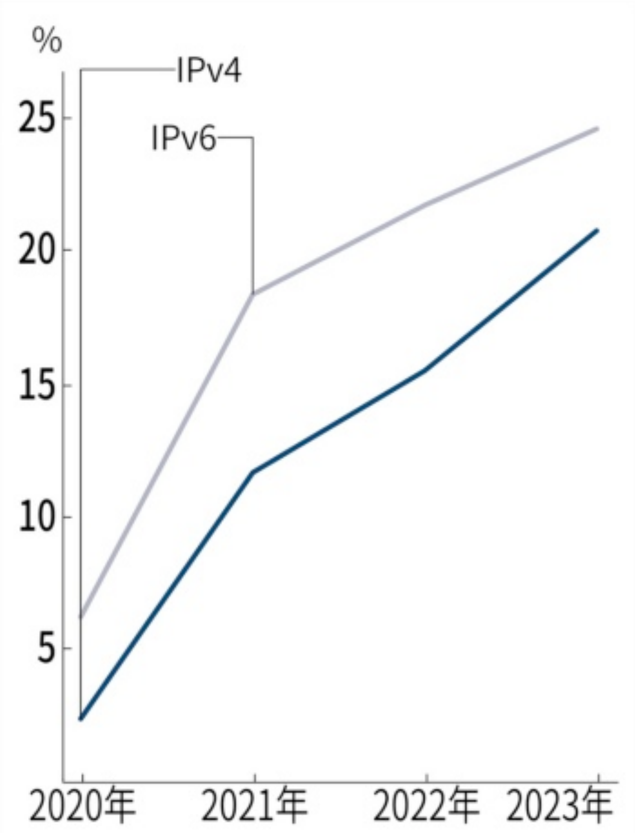
サービスパラメータの追加

- HTTPS/SVCB RRはサービスパラメータの追加が容易な構造
- すでにRFC9460にないパラメータがいくつか追加されている
  - dohpath (RFC9461): DoH URLのパス部分
  - ohttp (RFC9540): Oblivious HTTP (RFC9458)
    - 組み合わせるとOblivious DoHになる(が、cloudflareのODoHサーバは ohttp 未指定)
  - その他internet-draftがいくつか提案されている(RFCになるかどうかは不明)
- これらを含む全パラメーター覧 (IANAで管理している割り当て表)
  - https://www.iana.org/assignments/dns-svcb/dns-svcb.xhtml

利用状況

- クエリ全体に占めるHTTPS RRの割合
  - 2024/06の現状(左)と年ごとの推移(右)





IIJ提供のデータを元に日経xTECHが作成
https://xtech.nikkei.com/atcl/nxt/column/18/00001/09076/

  - SVCBは0.35%程度(iOS/macOSからのDDR)

ブラウザの対応状況

- いっぱいクエリが出ていることはわかった
- で、返ってきた応答をブラウザはどの程度ちゃんと解釈してるの?
- 調べてみました
  - Chrome125(Win11)、Safari17.4(macOS Sonoma)、Firefox126(Win11)

HTTPS RRのクエリを出しているか(1)

- Safari
  - 2020年ごろからデフォルト有効(iOS/macOS)
- Chrome
  - Chromeの内蔵リゾルバ(AsyncDNS)でHTTPS RRに対応
  - OS標準のリゾルバからAsyncDNSへの切り替え時期はOSによって異なる
    - Mac/Androidはけっこう前からAsyncDNSに切り替え済み
    - WindowsでOSのリゾルバからAsyncDNSへの切り替えで起きたのが、昨年のSummer Dayで発表のあったTCPクエリ増加問題

HTTPS RRのクエリを出しているか(2)

- Firefox
  - DoHを使うよう設定変更することで有効に
    - 念のため、DoHとHTTPS RRは本来まったく無関係です
    - 一部の国ではDoHがデフォルト有効になっているため、HTTPS RRもデフォルトで使われていると思われる(未確認)
  - network.dns.native_https_query を true にすればDoHでなくても使える
    - ただし、OSによっては動作が不安定とのこと
      - https://groups.google.com/a/mozilla.org/g/dev-platform/c/oh_Tk0iLT9A
    - Firefox127でデフォルト有効にする予定だったようだが、入らなかった

エイリアスモードへの対応

- RFC9460では多段エイリアスを許容
  - 最低1段以上(MUST)〜最大何段までにするかは実装側の裁量
- Safari: エイリアスに対応はしているが多段にはできない
- Chrome、Firefox: エイリアス非対応(RFC違反)
  - Chromeのチケット → https://issues.chromium.org/issues/40257146
  - Firefox → https://bugzilla.mozilla.org/show_bug.cgi?id=1869075
- HTTPS RRの目玉機能(ゾーン頂点でCNAMEが使えない問題の解決策)のはずなのに…

サービスモードへの対応

- 以下のようにゾーンに書くと、RFCに従うなら192.0.2.2にアクセスするはず

```
www     IN HTTPS 1 target
www     IN A     192.0.2.1
target  IN A     192.0.2.2
```

- Chromeはターゲット指定を無視する(RFC違反)
  - 192.0.2.1にアクセスする(そもそもtargetの名前解決をしない)
  - ipv4hint=192.0.2.2 を追加しても変わらず
- SafariとFirefoxはまとも
- 優先度に従うかどうかの検証はせず(Chromeがこの惨状では調べる意味が…)

サービスパラメータの解釈

- alpn
  - Chrome、Safari、Firefoxとも見てるっぽい
- ipv4hint、ipv6hint
  - Safari、Firefoxは見ているようだが、Chromeは無視
  - ipv4hintやipv6hintはあくまでヒントであって、したがわなくても差し支えない
- port、ech等の挙動は未確認
  - portは使うべきではない(ポート番号を変更するとファイアウォールやプロクシ等を越えられない環境が多いので)

Best Current Practice

- 現状はブラウザ側の対応状況に大きな差がある
  - ChromeはHTTPS RRをまともに実装できているとは言い難い
  - FirefoxはデフォルトではHTTPS RRのクエリを出さない
  - Safariは2段以上のエイリアスが使えないがいちばん無難
- どんなブラウザでも同じように動くことを求めるなら、以下の記述以外はダメ

```
www IN HTTPS 1 . alpn=xxx ipv4hint=xxx ipv6hint=xxx
```

- あくまでcurrent(2024/06時点)のbest practice
  - 将来的には用途に応じたいろんな記述ができるようになるはず……

(こういう調査はDNS屋ではなくWeb屋さんのコミュニティでやったほうがいいと思うんだ…)

権威サーバの対応状況

- オープンソースの主要な実装はどれも対応済み
- 大手権威DNSサービスの対応状況
  - 対応済み: Cloudflare DNS、Google Cloud DNS、さくらインターネット、IIJなど
  - 未対応: Amazon Route 53、Microsoft Azure DNS、その他多数
- CloudflareのCDNとDNSをどちらも利用している場合、自動的にHTTPS RRが登録される模様
  - Tranco top 1Mにリストされているwebサイトを対象にした調査では、HTTPS RRが設定されていたサイトの99.9%がCloudflareをNSに設定していたとのこと
    - https://arxiv.org/pdf/2403.15672

public DNSサービスの対応状況

- 念のため、フルリゾルバはHTTPS RRのために特別な対応は必要ありません
  - が、よけいな対応をしているところがありまして…
- 1.1.1.1 for Families
  - マルウェアやアダルトサイトをブロッキングするpublic DNSサービス(1.1.1.1とは別)
  - ブロッキング対象のドメインのHTTPS RRを聞くと、解釈できない壊れた応答を返す
- OpenDNS
  - 当初(すくなくとも2022年まで)はHTTPS RRのクエリにREFUSEDを返していた
  - 現在は拒否せずちゃんと応答するようになっている

Additional Sectionの取り扱い

- RFC9460曰く「ターゲット名の情報をAdditional Sectionに追加すべし」
  - ipv[46]hintは参照先をadditionalに追加できる実装が普及するまでの一時しのぎ
- 参照先をAdditional Sectionに追加する
  - BIND 9.19.24(権威かつminimal-responses noのとき)、Knot 3.3.5
- 参照先をAdditional Sectionに追加しない
  - BIND(上記以外のとき)、NSD 4.9.1、Unbound 1.20.0、Knot Resolver 5.7.3
  - 1.1.1.1、8.8.8.8、9.9.9.9
- 受け取る側(フルリゾルバ、ブラウザ)が追加情報を利用しているかは未確認

対応状況まとめ

- ブラウザの対応状況はまだまだ
  - 対応していないならまだしも、明らかに挙動のおかしな実装も
- 権威もまだまだ
  - 自前運用なら問題ないが、権威DNSサービスで対応しているところは少ない

- 現時点ではそれほどメリットを生かせる状況ではない

サービスパラメータ

HTTPS RRの今後

プロトコルの拡張

- SMTPはEHLOではじめるとサーバが対応している拡張機能のリストを取得できる
  - IMAP4ならCAPABILITY、POP3ならCAPA
- サーバが対応している拡張機能に応じて、クライアントは挙動を変えることができる
  - 暗号化(STARTTLS)
  - 複数コマンドの一括送信/一括応答(PIPELINING)
  - メッセージサイズの事前通知(SIZE) など

```
S: 220 mx.example.com ESMTP Postfix
C: EHLO foo.example.jp
S: 250-mx.example.com
S: 250-PIPELINING
S: 250-SIZE 10240000
S: 250-STARTTLS
S: 250-ENHANCEDSTATUSCODES
S: 250-8BITMIME
S: 250-DSN
S: 250 CHUNKING
C: STARTTLS
S: 220 2.0.0 Ready to start TLS
(TLS handshake)
C: MAIL FROM:<foo@example.jp> SIZE=1234
C: RCPT TO:<bar@example.com>
C: RCPT TO:<baz@example.com>
S: 250 2.1.0 OK
S: 250 2.1.5 Ok
S: 250 2.1.5 Ok
…
```

平文

TLS

HTTPの拡張

- SMTPはクライアントとサーバが交互に何往復もやりとりを繰り返す
- が、HTTPは1往復のやりとりだけで終わるプロトコル
  - サーバからの応答によって、クライアントがその後の動作を変えることはできない
  - あえてやるなら、初回接続は最低限の機能だけ、2回目以降の接続で拡張機能を使う
- Webサーバに接続する前に、対応している拡張機能を知ることができれば、初回接続からその機能を使うことができる
  - 権威DNSにその情報を登録しておけばよい → HTTPS RR (サービスパラメータ)
- 今後どんな拡張がされるのかは、Web屋さんの御心のままに…

HTTPの拡張

- SMTPはクライアントとサーバが交互に何往復もやりとりを繰り返す
- が、HTTPは1往復のやりとりだけで終わるプロトコル
  - サーバからの応答によって、クライアントがその後の動作を変えることはできない
  - あえてやるなら、初回接続は最低限の機能だけ、2回目以降の接続で拡張機能を使う
- Webサーバに接続する前に、対応している拡張機能を知ることができれば、初回接続からその機能を使うことができる
  - 権威DNSにその情報を登録しておけばよい → HTTPS RR (サービスパラメータ)
- 今後どんな拡張がされるのかは、Web屋さんの御心のままに…

HTTP以外にもあるよね

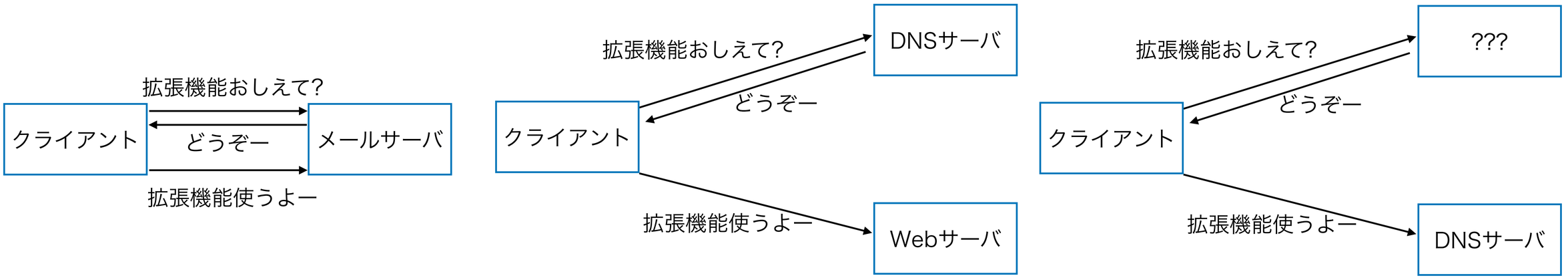
そう、DNS

DNSの拡張

- すでにEDNSがある
  - DNSメッセージのAdditional Sectionに疑似RRを付加
  - クライアント/サーバそれぞれが対応している拡張機能のリストを相手に通知
- クライアントからのEDNSの情報に応じてサーバが動作を変えることは可能
  - 「このクライアントにはUDPで512バイト超の応答を返しちゃっても大丈夫だな」
- が、DNSは1往復で終わるプロトコルなので、サーバからのEDNS情報に応じてクライアントが動作を変えることはできない
  - あえてやるなら2回目のクエリから → 初回からやるには?

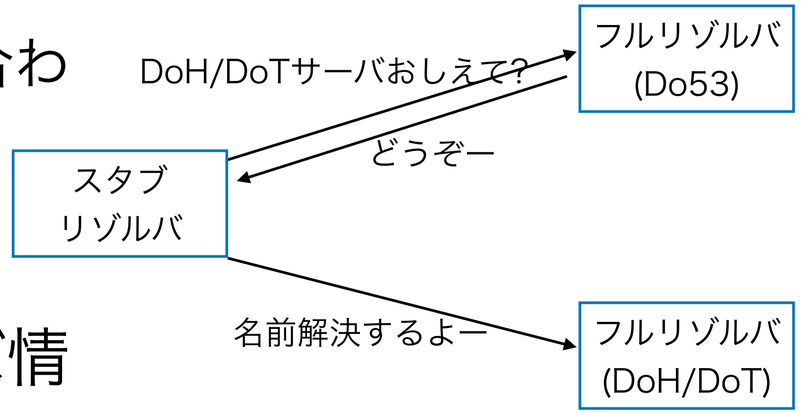
拡張機能の情報はどこに?

- メールサーバの拡張機能はメールサーバ自身に聞けばよい
- Webサーバの拡張機能はDNSサーバに聞けばよい
- DNSサーバの拡張機能はどこに聞く?



DoH/DoTサーバの自動検出

- RFC9462 Discovery of Designated Resolvers (DDR)
  - Do53なフルリゾルバに、DoH/DoTサーバの場所を問い合わせる特殊なクエリを送出
  - 応答に含まれるDoH/DoTフルリゾルバを利用
- DHCP、IPv6 RA、IKEv2にもDoH/DoTフルリゾルバ情報を通知するオプションが追加 (RFC9463、9464)
- どうせなら暗号化に必要な情報だけでなく、それ以外の情報もいっしょに送ってしまえばいいのでは?



フルリゾルバの機能拡張

- DoH/DoTフルリゾルバの場所を通知するために作られたプロトコルは、いずれもサービスパラメータを格納できる設計になっている
  - DNSを使うRFC9462はSVCB RR
  - DNSを使わないRFC9463、9464は拡張オプション
- DoH/DoTフルリゾルバ自動検出のための仕組みだが、将来的には暗号化以外にもさまざまな拡張機能を通知する手段として使われるようになると予想

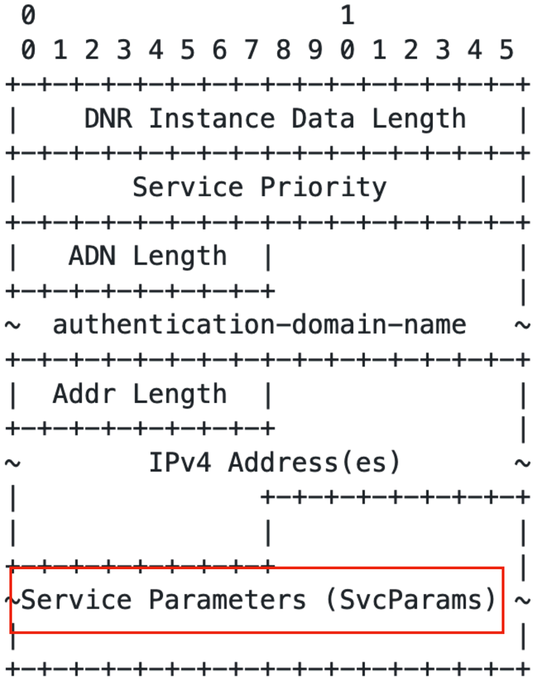


Figure 5: DNR Instance Data Format

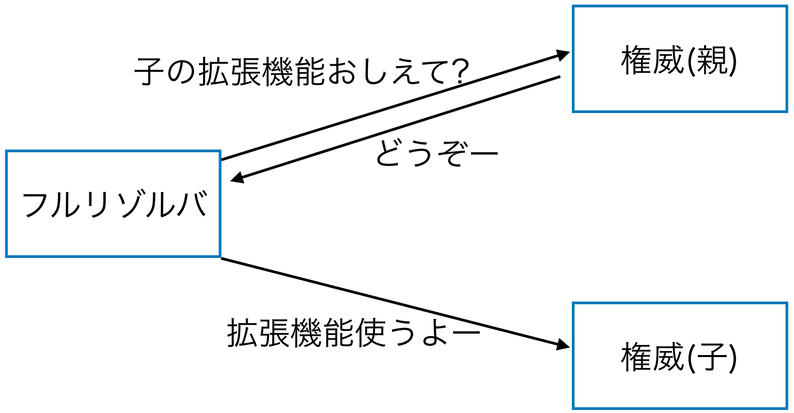
DHCPに格納されるサービスパラメータ(RFC9463)

権威サーバの暗号化

- ある権威サーバがDoTに対応済みだとして、そのことをフルリゾルバはどうやって知ればいいのか?
  - 案1: 予備知識なしでとりあえずDoTで聞いてみて、ダメならDo53でやりなおす
    - DoTのアクセスをMITMで妨害することでDo53を強制できてしまう
    - この方式がRFC9539になっちゃったけど(experimental)
  - 案2: DoTに対応しているという情報をどこかに置く
    - その情報が改竄されたらやはりDo53になってしまうので、何らかの方法で守る必要がある
    - 「何らかの方法」はDNSSECでいいとして、「どこかに置く」ってどこ?

権威サーバの機能拡張

- 権威サーバは階層構造
  - 子ゾーンの権威サーバに問い合わせる前に親ゾーンの権威に問い合わせる
- ならば、拡張機能の情報を親権威サーバに登録してやればよい



- でも親ゾーンにそんな情報を登録する場所ないよ?

DELEG RR

- IETFでDELEGという新しいリソースレコードが提案されて議論中
  - https://datatracker.ietf.org/doc/draft-dnsop-deleg/
- NSで定義されていたDNSの階層構造を、DELEG RRで置き替え

```
example.com.     IN NS    ns.example.com.                          ; これまでの委譲
example.com.     IN DELEG 1 ns.example.com. ipv4hint=192.0.2.1 ; 提案されている委譲
ns.example.com. IN A     192.0.2.1                                ; glue
```

  - NSは委譲元・委譲先に同じものを登録するが、DELEGは委譲元だけに登録
  - サービスパラメータを記述できる = 権威サーバが対応している拡張機能を通知できる

DNSの拡張機能って具体的には?

- DELEG RRの提案者曰く、
  - たとえば、alpn=dot とすることで権威サーバがDoTに対応していることを通知できる
    - さらに、tlsa=“鍵情報” も追加してサーバ証明書の情報を格納してもいい
  - たとえば、ds=“鍵ハッシュ” とすることでDS RRを代替できる
  - たとえば、dnsproto=2 とすることで既存のDNSを捨ててまったく新しいDNS ver. 2 に移行できるかも?
  - https://ripe87.ripe.net/wp-content/uploads/presentations/61-DELEGations.pdf

DELEG RRの例

```
; この権威サーバはDoTもDo53も喋れるよ
example.jp. IN DELEG 1 ns1.example.jp. (
        alpn=dot
        ipv4hint=192.0.2.1 )

; こっちはDoQ専用だよ(Do53は喋れないよ)
; 証明書の情報も入れておくよ(DANE)
example.jp. IN DELEG 1 ns2.example.jp. (
        mandatory=alpn
        alpn=doq
        ipv4hint=192.0.2.2
        tlsa="3 1 1 0123456789ABCDEF…" )
```

```
; エイリアスも使えるよ
example.com. IN DELEG 0 ns.example.org.


; エイリアスの先はDELEGじゃなくてSVCB
; DNSSEC公開鍵をエイリアス先に書けば
; レジストリに登録申請する手間が不要に
ns.example.org. IN SVCB 1 . (
        ipv4hint=198.51.100.1
        ds="12345 13 2 01234567…" )
```

- 注: こんなことをやりたいねーという議論がされているだけで、こうなると決まったわけではない

DELEG RRのインパクト

- HTTPS、SVCBは他のプロトコルが使うための入れ物を提供するだけ
  - DNS自体にはさほど影響はない
- DELEGはDNS自身が使うためのもの
  - 「サービスパラメータを書けるNS」ではなく、エイリアスなど他にもさまざまな機能
  - 複数の権威がすべて同等に扱われるNSと違い、機能の異なる権威サーバが共存できる
  - DNSの仕組みそのものに超特大の影響
    - メリットだけでなくデメリットも (格納される情報が増える→ パケット肥大化 → ampの踏み台?)
- 標準化されるとしても当分先なので、いまは横目で眺めてるだけでよい

まとめ

- SVCB/HTTPS RRはRFC9460になりました
  - Web屋さんが欲しいといってできたRRなのに、Web屋さんの実装はまだ不十分
  - HTTPS RRのいろんな機能をちゃんと使えるようになるのはまだ先になりそう
- HTTPS RRで発明されたサービスパラメータという概念を、DNS自体にも取り入れようという動きがある
  - DoH/DoTの自動検出の仕組みを使うことで、暗号化に必要な情報以外も通知できる
  - 権威サーバの対応している機能を通知できるようにする仕組みの議論もはじまった
  - 10年後のDNSはいまとはだいぶ違うものになってるかも