

# UnboundとDNSSEC

日本Unboundユーザ会 滝澤隆史

2011-03-04

日本Unboundユーザ会 OSC 2011 Tokyo/Spring発表資料

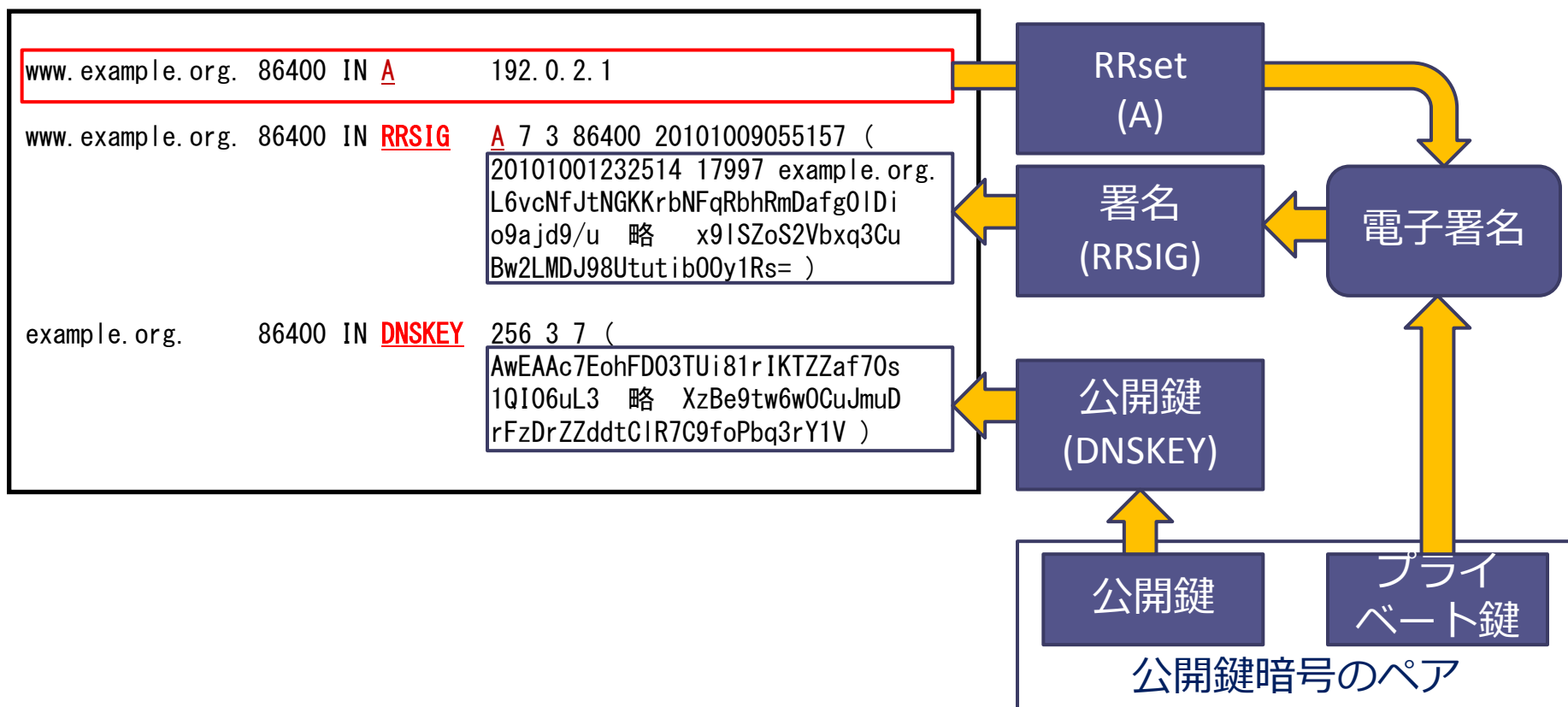
# DNSSECの簡単なおさらい

# DNSSEC

- DNSSECはDNSセキュリティ拡張
- キャッシュポイズニング対策
- クエリーに対する応答の詐称の検出
- 応答に含まれるDNS資源レコードの改ざんの検出
- 電子署名の技術を使う
  - 公開鍵暗号
  - メッセージダイジェスト

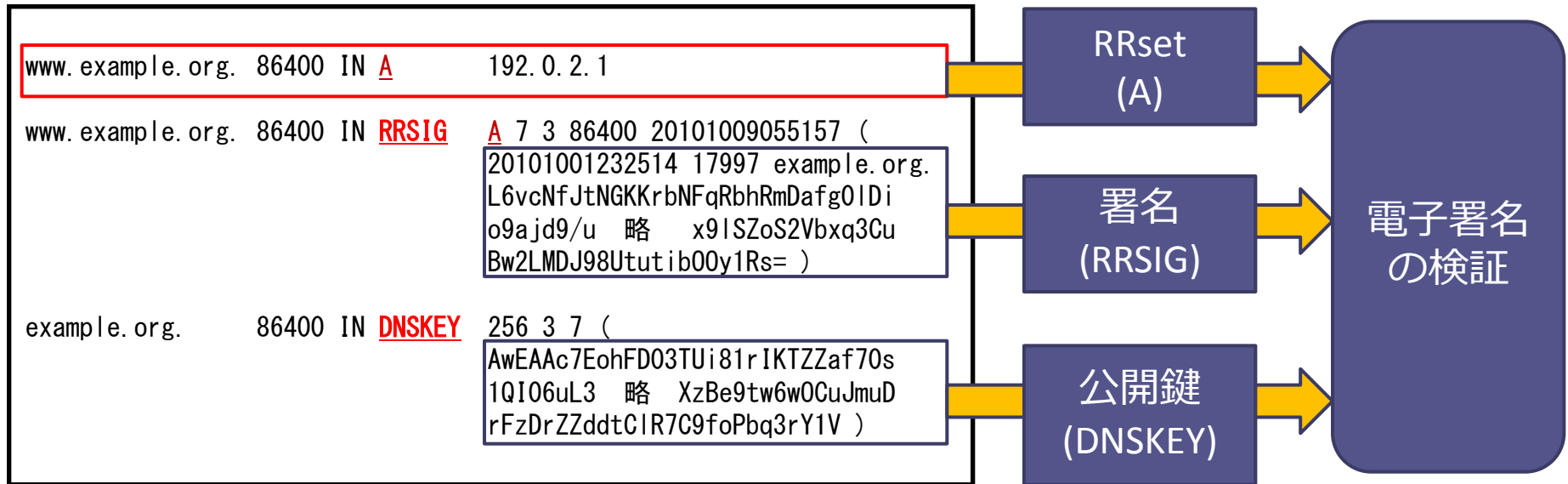
# 権威サーバ側

RRsetにプライベート鍵（ゾーン署名鍵(ZSK)) で署名



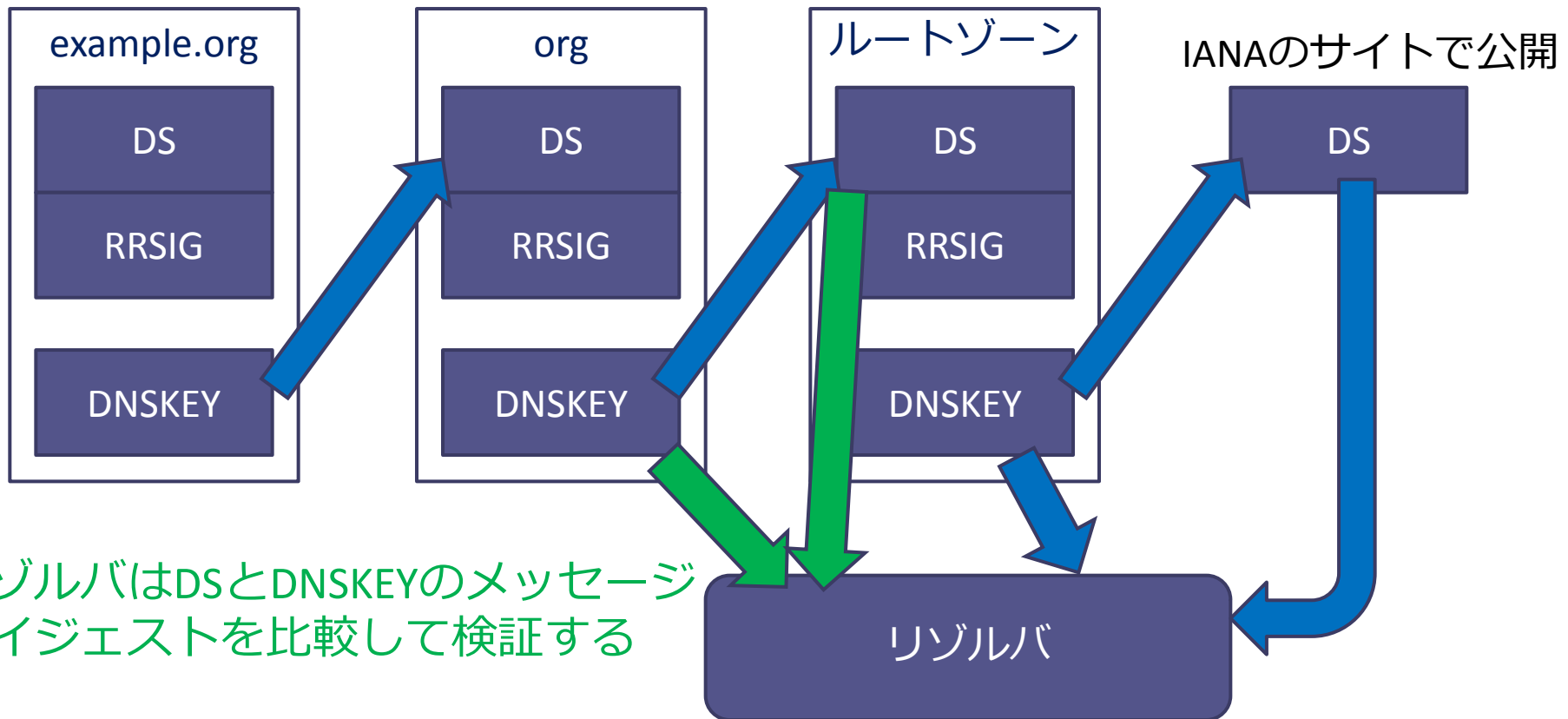
# リゾルバ側

DNSKEYレコードの公開鍵でRRSIGレコードの署名を検証



# 信頼の連鎖

DNSKEYのメッセージダイジェストを親ゾーンのDSとして登録



リゾルバはDSとDNSKEYのメッセージダイジェストを比較して検証する

予めルートゾーンのDSあるいはDNSKEYを  
トラスタンカーとして登録

# DNSSECを有効にする方法

# DNSSECを有効にする方法

- トラストアンカーを登録する



# トラストアンカーの登録

- IANAのサイトからルートゾーンのDSを取得
  - <https://data.iana.org/root-anchors/>
  - PGPなどで検証する

## Index of /root-anchors

Kjqmt7v.crt	15-Jul-2010 19:13 1.0K
Kjqmt7v.csr	15-Jul-2010 19:13 765
draft-icann-dnssec-trust-anchor.html	15-Jul-2010 20:44 32K
draft-icann-dnssec-trust-anchor.txt	15-Jul-2010 20:44 14K
icann.pgp	16-Jul-2010 14:19 2.0K
icannbundle.p12	15-Jul-2010 19:13 3.8K
icannbundle.pem	15-Jul-2010 19:13 17K
root-anchors.asc	15-Jul-2010 19:13 189
root-anchors.p7s	15-Jul-2010 19:13 4.9K
<b>root-anchors.xml</b>	15-Jul-2010 19:13 418

*Apache Server at data.iana.org Port 80*

# トラストアンカーの登録

- root-anchors.xmlファイルからDSレコードを作成

```
<?xml version="1.0" encoding="UTF-8"?>
<TrustAnchor id="AD42165F-3B1A-4778-8F42-D34A1D41FD93" source="http://data.iana.org/root-anchors/root-anchors.xml">
<Zone>.</Zone>
<KeyDigest id="Kjqmt7v" validFrom="2010-07-15T00:00:00+00:00">
<KeyTag>19036</KeyTag>
<Algorithm>8</Algorithm>
<DigestType>2</DigestType>
<Digest>49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5</Digest>
</KeyDigest>
</TrustAnchor>
```



```
. IN DS 19036 8 2 49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5
```

# トラストアンカーの登録

- ルートゾーンのDSをファイルとして保存
  - `/etc/unbound/root.key`
- unboundが読み書きできるように所有者を変更する
  - `chown unbound:unbound /etc/unbound/root.key`
- 設定ファイルに設定を記述
  - `auto-trust-anchor-file: "/etc/unbound/root.key"`

# トラストアンカーの登録

- unboundの起動時にルートゾーンのDNSKEYレコード (KSK) が取得される

```
; autotrust trust anchor file
;;id: . 1
;;last_queried: 1285978943 ;;Sat Oct 2 09:22:23 2010
;;last_success: 1285978943 ;;Sat Oct 2 09:22:23 2010
;;next_probe_time: 1286020224 ;;Sat Oct 2 20:50:24 2010
;;query_failed: 0
;;query_interval: 43200
;;retry_time: 8640
. 86400 IN DNSKEY 257 3 8
AwEAAagAikIVZrpC6la7gEzahOR+9W29euxhJhVVLOyQbSEW008gcCjFFVQUtf6v58fLjwBd0YI0EzrAcQqBGCzh/RSt
loO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXpoY68Ls
vPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGlcGOYI7OyQdXfZ5
7relSQageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=
;{id = 19036 (ksk), size = 2048b} ;;state=2 [ VALID ] ;;count=0
;;lastchange=1285978943 ;;Sat Oct 2 09:22:23 2010
```

# トラストアンカーの登録

- ロールオーバーに対応している
- DNSKEYレコードの自動更新ができる

# unbound-anchor

- トラストアンカーを取得・更新するツール
- バージョン1.4.7から
- 動作
  - 組み込みのルートゾーンのDSレコードを  
/etc/unbound/root.keyに書き込む
  - ルートゾーンのDNSKEYレコードを取得して  
root.keyファイルを上書きする

# unbound-anchor

- 動作（組み込みのDSで失敗した場合）
  - IANAのサイトからroot-anchors.xmlファイルを取得し、組み込みのCA証明書により検証する。
  - root-anchors.xmlファイルのXMLを解析してDSレコードを作成し、root.keyファイルに書き込む。
  - ルートゾーンのDNSKEYレコードを取得してroot.keyファイルを上書きする

# unbound-anchor

- root.keyファイルを取得したら、unboundが読み書きできるように所有者を変更する
  - `chown unbound:unbound /etc/unbound/root.key`
- 設定ファイルに設定を記述
  - `auto-trust-anchor-file: "/etc/unbound/root.key"`



# 試してみる (dig)

```
$ dig @127.0.0.1 +dnssec . SOA
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18108
```

```
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 14, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags: do; udp: 4096
```

```
;; QUESTION SECTION:
```

```
.; IN SOA
```

```
;; ANSWER SECTION:
```

```
. 86400 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2011022701
```

```
1800 900 604800 86400
```

```
. 86400 IN RRSIG SOA 8 0 86400 20110306000000 20110226230000
```

```
21639 . eQWDIgYBTzjhQ6lIEEr3iioS0Pt5z0EePuuzLCIQeXnO+pj5Vhqeg711
```

```
XoT9F9ZBG+sG2gbF6u5xmtcS9MBij7tnXt0A8r7Hf78zxPKsVte3ExqV
```

```
esjOif1ni/SYSa+KoxRirwdvOakowQaDf4dIYdDxolwzTgAD/Rxu5Ot2 4A4=
```

略

# 試してみる (drill)

```
$ drill @127.0.0.1 -D . SOA
```

```
;; ->HEADER<<- opcode: QUERY, rcode: NOERROR, id: 11726
```

```
;; flags: qr rd ra ad ; QUERY: 1, ANSWER: 2, AUTHORITY: 14, ADDITIONAL: 0
```

```
;; QUESTION SECTION:
```

```
;; . IN SOA
```

```
;; ANSWER SECTION:
```

```
. 86400 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2011022701 1800 900  
604800 86400
```

```
. 86400 IN RRSIG SOA 8 0 86400 20110306000000 20110226230000 21639 .
```

```
eQWDIgYBTzjhQ6lIEEr3iioS0Pt5z0EePuuzLCIQeXnO+pj5Vhqeg711XoT9F9ZBG+sG2gbF6u5xmt  
cS9MBij7tnXt0A8r7Hf78zxPKsVte3ExqVesjOif1ni/SYSa+KoxRirwdvOakowQaDf4dlYdDxolwzTg  
AD/Rxu5Ot24A4= ;{id = 21639}
```

略

```
;; EDNS: version 0; flags: do ; udp: 4096
```

```
;; SERVER: 127.0.0.1
```

```
;; WHEN: Mon Feb 28 16:05:30 2011
```

```
;; MSG SIZE rcvd: 612
```

# 検証失敗の記録方法

- val-log-level: 1
  - Mar 1 20:12:42 mercury unbound: [7275:0] info: validation failure <www.dnssec-failed.org. A IN>
- val-log-level: 2
  - Mar 1 20:14:19 mercury unbound: [7301:0] info: validation failure <www.dnssec-failed.org. A IN>: signature expired from 68.87.29.164 for key dnssec-failed.org. while building chain of trust

# 検証の失敗の確認

- <http://www.dnssec-failed.org/>
  - comcastによる確認サイト
  - DNSSECの検証に失敗するため、DNSSECの検証に対応しているとアクセスできない
- [www.dnssec-failed.org](http://www.dnssec-failed.org/)のAレコードを問い合わせて確認する。

# 検証の失敗例

```
$ dig @127.0.0.1 +dnssec www.dnssec-failed.org A
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 54175
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.dnssec-failed.org.      IN      A

;; Query time: 1346 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Feb 28 16:02:29 2011
;; MSG SIZE rcvd: 50
```

# DNSSECを無効にする方法

# DNSSECの運用障害

- これまでDNSSECの運用に関してTLDにおいていくつかの障害が発生している
  - 鍵の更新の失敗
  - 証明の有効期間の間違い
  - HSM (hardware security module) の故障
- 署名の検証に失敗するとそのドメインに関する回答を返さない

# DNSSECの障害を回避する

- 大規模な障害が起きたときに一時的に回避できる方法を知っておいた方がよい



# 署名の検証の部分停止

- 信頼の連鎖の停止
  - domain-insecure: "example.jp"

# 寛容 (permissive) モード

- 署名の検証は実施するが、検証に失敗してもSERVFAILを返さない
  - val-permissive-mode: yes
- 検証失敗の記録
  - val-log-level: 1
  - val-log-level: 2

# 署名の検証の停止

- トラストアンカーの設定の削除
  - 設定ファイルにおいてトラストアンカーの設定をコメントアウトする
- validatorモジュールの無効化
  - `module-config: "iterator"`

# 終わり