DNSSEC.jp プロトコル理解SWG
RFC 4035

DNSSEC.jp

■ RFC 4035の構成

1. Introduction
2. Zone Signing
3. Serving
4. Resolving
5. Authenticating DNS Responses
6. IANA Considerations
7. Security Considerations
8. Acknowledgements
9. References

1. Introduction

- 本文書はDNSSECプロトコル処理を定義

- 関連文書

  RFC4033/4034

  RFC1034/1035

  RFC2181/2308

2. Zone Signing

- 2.1. Including DNSKEY RRs in a Zone
- 2.2. Including RRSIG RRs in a Zone
- 2.3. Including NSEC RRs in a Zone
- 2.4. Including DS RRs in a Zone
- 2.5. Changes to the CNAME Resource Record
- 2.6. DNSSEC RR Types Appearing at Zone Cuts
- 2.7. Example of a Secure Zone

2. Zone Signing

- 署名付きZONE
  - －DNSKEY(DNS Public Key)
  - －RRSIG(Resource Record Signature)
  - －NSEC(Next Secure)
  - －DS(Delegation Signer)

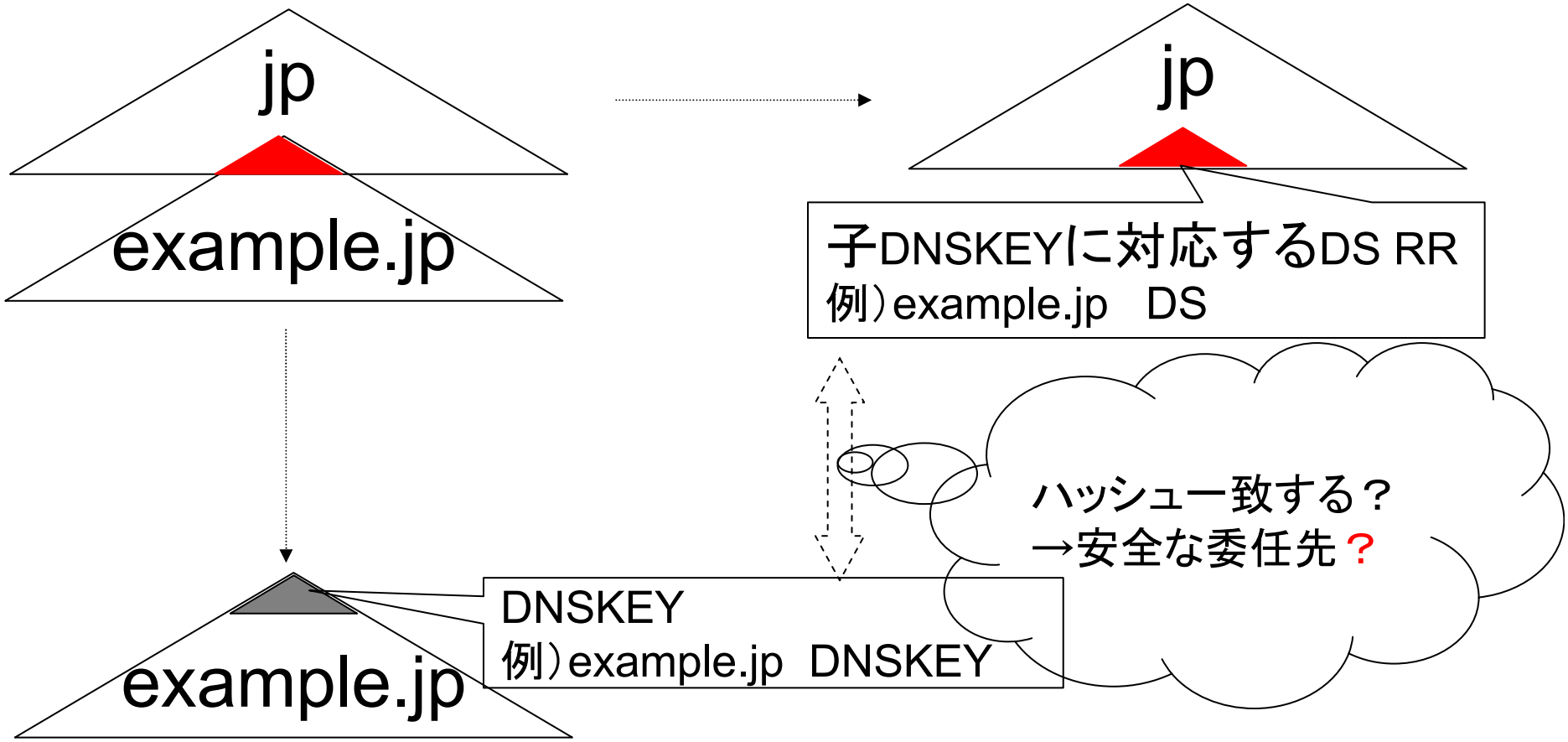
上記レコードを持たないものは署名なしZONEです

＊CNAME　RRの定義変更されます。

2.1 Including DNSKEY RRs in a Zone

- 1つ以上の公開/秘密鍵ペアがある
- RRSIG RR生成に使用した秘密鍵に対する公開鍵を保存したDNSKEY RRを持つべき
  - ―RFC4034で規定したキー鍵フラグを設定
  - ―他の公開鍵を規定以外のキー鍵フラグを設定しDNSKEY RRに保存してもよいが、RRSIG検証に使用してはいけない
- zone apexに最低1つのDNSKEY RRと親側のDS RRによりsecure entry pointと判断できる

2.1 Including DNSKEY RRs in a Zone



2.2 Including RRSIG RRs in a Zone

① RRSIG owner nameがRRset owner nameと等しい

② RRSIG classがRRset classと等しい

③ RRSIG typeがRRset typeと等しい

④ RRSIG Original TTL値がRRsetのTTL値と等しい

⑤ RRSIG RR's TTL値がRRsetのTTL値と等しい

⑥ RRSIG Labelsがowner nameのLabels数と等しい

ーnull root labelまたは最も左側のlabelがワイルドカードであった場合は、countしない

2.2 Including RRSIG RRs in a Zone

⑦ RRSIG Signer's NameがRRsetを持つZONE名と等しい

⑧ Algorithm & Signer's Name & Key Tagの内容により、zone内のDNSKEYが一つに特定できる

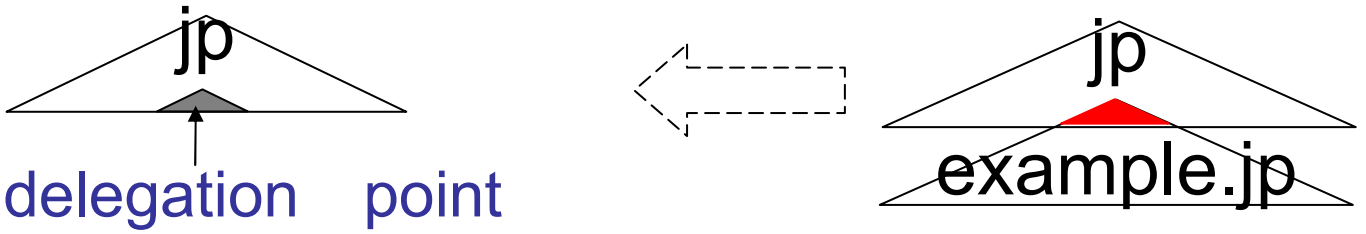
以上、8つの条件を満たしているRRSIGが1つ以上存在している

1つのRRsetが複数のRRSIG RRを持ってもよい

2.2 Including RRSIG RRs in a Zone

注意点

- RRSIG RR自身は署名されない
  ー無限ループ処理になる
- zone apexのNS RRsetは署名付きである
- 親ZONEの子に委任するためのNS RRsetは署名されない（delegation points）
  ー同様に、グルーアドレスRRsetは署名されない



2.2 Including RRSIG RRs in a Zone

①RRSIG owner nameがRRset owner nameと等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

②RRSIG classがRRset classと等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.

※class=IN

3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

③RRSIG typeがRRset typeと等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

④RRSIG Original TTL値がRRsetのTTL値と等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

⑤RRSIG RR's TTL値がRRsetのTTL値と等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

⑥RRSIG LabelsがOwner nameのLabels数と等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

⑦RRSIG Signer's NameがRRsetを持つZONE名と等しい

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.2 Including RRSIG RRs in a Zone

⑧Algorithm & Signer's Name & Key Tagの内容により、
zone内のDNSKEYが一つに特定できる

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=
3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

2.3 Including NSEC RRs in a Zone

- 署名付きZONEが権限を持つ全てのowner names または、子ZONEへ委譲するNS RRsetのowner namesに対して存在する

  ー親ZONEのグルーアドレスRRsetには

  NSEC RRとRRSIG RRも存在しない

- NSEC RRのTTL値は、ZONE SOA RRの最小値と同じであるべき

2.3 Including NSEC RRs in a Zone

注意点

- 特定のowner namesに対してNSEC RR（対応するRRSIG RR）だけしか存在しない状態があってはならない
- 親側のNSEC RRのビットマップに注意

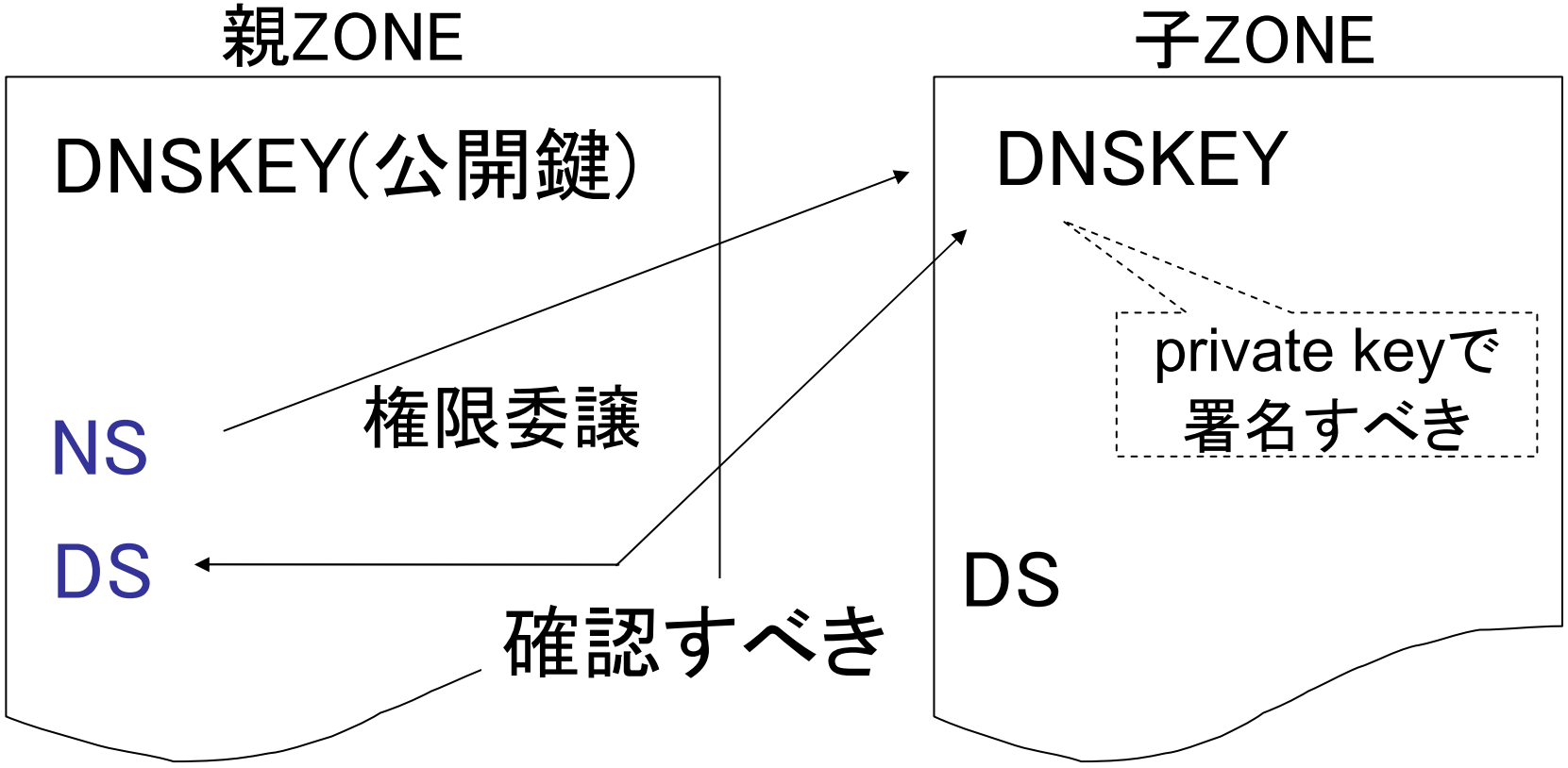
ー親ZONEが権限を持たないNS RRset以外のbitsは、クリアされなければならない

グルーアドレスRRset等のこと

2.4 Including DS RRs in a Zone

- 子ZONEが署名付きの場合、親ZONEにDS RRsetが存在すべき(認証連鎖を構築)
  - ーZONE内のDS RRsetは、すべて署名されるべき
  - ーDS RRsetは、zone ‘s apexに存在できない
  - ーDS RRsetは、複数設定可能
    対応する子ZONEのZSKを使用し、子ZONE内のRRSIGが検証できること

2.4 Including DS RRs in a Zone



※権限委譲NSのTTL値とDSのTTL値は一致すべき

2.4 Including DS RRs in a Zone

- 運用上の問題点
  ー子ZONEにある対応するDNSKEY RRの識別が必要

親ZONEと子ZONEのやり取りに関しては、
対象範囲外です。

2.5 Changes to the CNAME Resource Record

- 署名付きZONEにCNAME RRsetが存在する場合、適切なRRSIGとNSEC RRsetが必要
  - 動的な更新を行うKEY RRset使用可能

※ダイナミックDNS

- CNAME RRの定義
  - CNAME
  - NSEC（新規追加）
  - RRSIG（新規追加）

2.6 DNSSEC RR Types Appearing at Zone Cuts

- zone cutに存在可能なRR
  - NS RRset
  - NSEC RR（新規追加）
  - DS RR（新規追加）
- NSEC RRとDS RRは親が権限を持つ
  - RFC4033のAuthoritative RRset項目参照

2.7 Example of a Secure Zone

付録Aに署名付きのZONE一式の例があります

3. Serving

- 3.1 Authoritative Name Server
  - -3.1.1 including RRSIG RRs in a Response
  - -3.1.2 including DNSKEY RRs in a Response
  - -3.1.3 including NSEC RRs in a Response
  - -3.1.4 including DS RRs in a Response
  - -3.1.5 Responding to Queries for Type AXFR or IXFR
  - -3.1.6 The AD and CD Bits in an Authoritative Response
- 3.2 Recursive Name Servers
  - -3.2.1 The DO Bit
  - -3.2.2 The CD Bit
  - -3.2.3 The AD Bit
- 3.3 Example DNSSEC Response

3. Serving（サーバ側処理）の概要

■ネームサーバの要件

- EDNS0メッセージサイズ拡張をサポートすること
  ⇒推奨4000オクテット（最低でも1220オクテット）

- IPv6で転送されるUDPデータグラムをMTUに応じてフラグメント化すること

- クエリが『EDNSのOPTレコードを持たない』or『DOビットがクリアされていた』場合、RRSIG、DNSKEY、NSEC RRは、通常のRRsetと同様の応答を行うこと。
  ※DS RRについては3.1.4.1で規定する特別な処理が必要

3. Serving（サーバ側処理）の概要

- クエリに明示的なDNSSEC RRが含まれていて、クエリ内容とネームサーバ自身が管理する複数のゾーン内容とが一致する場合、応答に以下の1つを返しても良いこと。
  - 委任点の上に存在するRRset
  - 委任点の下に存在するRRset
  - 委任点の上、下に存在するRRset両方
  - Answer sectionを空（レコードなし）にして返す
  - それ以外の応答
  - エラー

3. Serving（サーバ側処理）の概要

- DNSSECはDNSメッセージヘッダ内に2つの新しいビットを割当
  - CD(Checking Disabled)ビット
  - AD(Authentic Data)ビット

  ※CDビットとADビットに関する詳細は、Section 3.1.6、3.2.2、3.2.3、4、4.9を参照

- CDビットはリゾルバ側で制御する。
  - ネームサーバはクエリのCDビットを対応する応答にコピーしなければならない。

- ADビットはネームサーバ側で制御する。
  - クエリに含まれるADビットを無視しなければならない。

3. Serving（サーバ側処理）の概要

- ネームサーバの規定（[RFC2672]）に従い、DNAMEからCNAME RRを合成した場合、合成したCNAME RRに対し署名を生成すべきではない

3.1 Authoritative Name Servers

- クエリが『EDNSのOPT RRを持っている』or『DOビットが設定されていた』場合、RRSIG、NSEC、DS RRを後述の規則に従って応答に付加しなければならない

- DNSSECはDNSのZONE転送プロトコルを変更しない

（詳細はSection3.1.5）

3.1.1 including RRSIG RRs in Response

- ネームサーバがDOビットが設定されたクエリに応答する場合、応答へのRRSIG RR付加は以下の規則に従う

※TC（Trancation）：1ビットで回答データが不完全であることを示し改めてTCPで問い合せることが求められる。

規則内容	要求レベル
Answer sectionに署名付きRRsetを付加する場合	
①・その署名付きRRsetのRRSIG RRもAnswer sectionに付加しなければならない。 ・他に付加されなければならないRRsetよりもRRSIG RRの方が優先度が高い。	MUST
②容量的にRRSIG RRを付加できない場合、TCビットを設定しなければならない。	MUST
Authority sectionに署名付きRRsetを付加する場合	
③・その署名付きRRsetのRRSIG RRもAuthority sectionに付加しなければならない。 ・他に付加されなければならないRRsetよりもRRSIG RRの方が優先度が高い。	MUST
④容量的にRRSIG RRを付加できない場合、TCビットを設定しなければならない。	MUST
Additional sectionに署名付きRRsetを付加する場合	
⑤その署名付きRRsetのRRSIG RRもAdditonal sectionに付加しなければならない。	MUST
⑥容量的にRRsetとそのRRsetのRRSIG RR両方が付加できない場合、RRsetだけを残してRRSIG RRを除外しても良い。	MAY
⑦この場合、RRSIG RRが入らなかったというだけの理由でTCビットを設定してはならない。	MUST NOT

3.1.2 including DNSKEY RRs in Response

- ネームサーバがDOビットが設定されたクエリに応答する場合、応答へのDNSKEY付加は以下の規則に従う

内容	要求レベル
①DOビットが設定された署名付きゾーンの頂点にあるSOA or NS RRのクエリに応答する場合 ⇒ZONE頂点にあるDNSKEY RRsetをAdditional sectionに入れて返しても良い。 ※この場合、DNSKEY RRsetと対応のRRSIG RRはAdditional sectionに入れられるべき他の情報よりも優先度は低くなる。	MAY
②DNSKEY RRsetとそのRRsetのRRSIG RRの付加に十分な容量が応答メッセージに無い場合 ⇒ネームサーバはDNSKEY RRsetを応答に付加すべきではない。	SHOULD NOT
③DNSKEYとそのDNSKEYのRRSIG RRの付加に十分な容量が無い場合 ⇒ネームサーバは両RRを除外しなければならない。	MUST
④これらのRRが入らなかったという理由だけでTCビットを設定してはならない（Section3.1.1参照）。	MUST NOT

3.1.3 including NSEC RRs in Response

- ネームサーバがDOビットが設定されたクエリに応答する場合、以下の応答毎の規則に従い、NSEC RR付加を行わなければならない。
  - No Data ※付録B.3参照
  - Name Error ※付録B.2参照
  - Wildcard Answer ※付録B.6参照
  - Wildcard No Data ※付録B.7参照

- 応答毎に異なった、NSEC RRを付加する理由※詳細は5.4.で説明
  - ネームサーバはZONE内に<SNAME,SCLASS,STYPE>に完全一致するものが存在しないことを証明するため
  - ネームサーバが返した応答がZONEデータが正しいことを証明するため

3.1.3.1 including NSEC RRs in Response “No Data”

No Data	ZONEは<SNAME, SCLASS>に完全一致するRRsetを持つが、<SNAME, SCLASS, STYPE>に完全一致するRRsetが存在しない

- ネームサーバがDOビットが設定されたクエリに応答する場合、且つ”No Data”応答へのNSEC RR付加は下記表の規則に従う
  - クエリ対象の名前は存在しているので、このクエリに対してWildcard展開は行われない
  - クエリのRRタイプが存在しないことを証明するには、単一の署名付きNSEC RRがあれば充分

内容	要求レベル
①ネームサーバはAuthority sectionに<SNAME, SCLASS>に関するNSEC RRとそのRRのRRSIG RRを共に付加しなければならない。	MUST
②容量的にNSEC RRまたはそのRRのRRSIG RRが付加できない場合 ⇒ネームサーバはTCビットを設定しなければならない。	MUST

3.1.3.2 including NSEC RRs in Response “Name Error”

Name Error	ZONEには<SNAME, SCLASS>に完全一致するRRsetも、Wildcard展開により一致するRRsetが存在しない

・ ネームサーバがDOビットが設定されたクエリに応答する場合、且つ”Name Error”応答へのNSEC RR付加は下記表の規則に従う

内容	要求レベル
①ネームサーバは下記のNSEC RRとそのRRのRRSIG RRをAuthority sectionに付加しなければならない。 ※owner名はWildcard展開されない - <SNAME、SCLASS>に完全一致するRRsetが存在しないことを証明するNSEC RR - Wildcard展開により<SNAME、SCLASS>に一致するRRsetが存在しないことを証明するNSEC RR	MUST
②単一のNSEC RRが上記2つのNSEC RRを証明する場合 ⇒ネームサーバはNSEC RRとそのRRのRRSIG RRをそれぞれ1つだけ Authority sectionに付加すべき	SHOULD
③容量的にNSEC RRとRRSIG RRが付加できない場合 ⇒ネームサーバはTCビットを設定しなければならない 。	MUST

3.1.3.3 including NSEC RRs in Response “Wildcard Anser”

Wildcard Answer	ZONEに<SNAME, SCLASS>に完全一致するRRsetは存在しないが、Wildcard展開により<SNAME,SCLASS,STYPE>に一致するRRsetが存在する

- ネームサーバがDOビットが設定されたクエリに応答する場合、且つ”Wildcard Answer”応答へのNSEC RR付加は下記表の規則に従う

内容	要求レベル
①ネームサーバはAnswer sectionにワイルドカード展開後の回答とRRSIG RRを付加しなければならない。	MUST
②Authority sectionにはゾーン内に<SNAME, SCLASS>に より良く一致するRRsetが無いことを証明するNSRC RRとそのRRのRRSIG RRを付加しなければならない。	MUST
③容量的にNSEC RRとそのRRのRRSIG RRが 付加できない場合 ⇒ネームサーバはTCビットを設定しなければならない（3.1.3参照）。	MUST

3.1.3.4 including NSEC RRs in Response “Wildcard No Data”

Wildcard No Data	・ZONEには<SNAME, SCLASS>に完全一致するRRsetは存在しない。 ・Wildcard展開により<SNAME,SCLASS>に一致する1つ以上のRRsetが存在するが、<SNAME,SCLASS,STYPE>に一致するRRsetは存在しない。

- ネームサーバがDOビットが設定されたクエリに応答する場合、且つ”Wildcard No Data”応答へのNSEC RR付加は下記表の規則に従う

内容	要求レベル
①Wildcard展開してもSTYPEに一致するRRsetは存在しないという場合 ⇒ネームサーバはAuthority sectionに以下の条件を満たすNSEC RRとそのRRのRRSIG RRを 共に付加しなければならない ※owner名はWildcard展開されない - Wildcardにより<SNAME, SCLASS>に一致するWildcardのowner名を持つRRsetの中でSTYPEに一致するものは無いことを証明するNSEC RR 。 - ZONE内に<SNAME, SCLASS>により良く一致するRRsetは存在しないことを 証明するNSEC RR。	MUST
②単一のNSEC RRが上記2つの内容を証明する場合 ⇒ネームサーバはNSECC RRとRRSIG RRをそれぞれ1つだけAuthority sectionに 含めるべきである	SOULD
③容量的にNSEC RRとRRSIG RRが付加できない場合 ⇒ネームサーバはTCビットを設定しなければならない（Section 3.1.1参照）	MUST

3.1.3.5 Finding the Right NSEC RRs ※詳細は5.4を参照

- 特定のSNAMEに一致するRRsetが存在しないことを証明するNSEC RRをネームサーバが付加しなければならない状況が存在する
- 権威を持つゾーン内でそのようなNSEC RRを発見する処理は概念上は容易である
- 以下のアルゴリズムで適切なNSEC RRが発見できる
  - 名前*N*に一致するRRsetがZONE*Z*に存在しないことを発見する
  - *Z*内の全RRsetのowner名から重複する名前を除外
  - owner名のリスト*S*を構築し、*S*を正規順序（[RFC4034]）に並べる
  - *S*の中に*N*が存在していたらその直前に順序づけられていたであろう名前*M*を発見する
  - *M*がowner名*N*を持つRRsetが存在しないことを証明するNSEC RRのowner名になる

3.1.4 including DS RRs in Response

- ネームサーバがDOビットが設定されたクエリに応答する場合、応答へのDS RR付加は以下の規則に従う
  - DS、NSEC、RRSIG RRを付加することにより、参照メッセージサイズが大きくなるため、幾つか、または全てのglue RRが省略されてしまう場合がある

内容	要求レベル
①委任点にDS RRsetが存在する場合 ⇒ネームサーバはNS RRsetに加えて、DS RRsetとそのRRsetのRRSIG RRを共にAuthority sectionに付加しなければならない	MUST
②委任点にDS RRsetが存在しない場合 ⇒ネームサーバはNS RRsetに加えて、DS RRsetが存在しないことを証明するNSEC RRとそのRRのRRSIG RRを共に 返さなければならない	MUST
③ネームサーバはまずNS RRsetを付加し、その後にNSEC RRsetとそのRRsetのRRSIG RRを付加しなければならない	MUST
④容量的にDS RRまたはNSEC RRとそのRRのRRSIG RRが付加できない場合 ⇒ネーム サーバはTCビットを設定しなければならない（Section3.1.1参照）。	MUST

3.1.4.1 including DS RRs in Response ※詳細は付録B.8を参照

- DS RRはZONE Cutの親ZONE側にしか存在しない点で特殊
  - 例）”foo.example”の委任に関するDS RRsetは、”foo.example”ZONEではなく、”example”ZONEに保存される

- ネームサーバとリゾルバ双方がDS RRを処理する際で、以下の条件（4つ）を全て満たす場合のみ特別な処理を必要とする
  - ネームサーバがZONE CutにあるDS RRsetの問い合わせを受信した
  - ネームサーバは子ZONEに対して権威を持つ
  - ネームサーバは親ZONEに対して権威を持たない
  - ネームサーバは再帰検索を行わない

内容	要求レベル
特別の処理規則に従った応答をする場合	
・ネームサーバはSNAMEに対して権威を持つが、問い合わされたRRsetを提供することができないため、ネームサーバは子ZONEの頂点にDS RRsetが存在しないことを示す権威を持つ”No Data”応答を返さなければならない。	MUST

3.1.4.1 including DS RRs in Response

- 特別の処理規則に従った応答を除いた全ての場合は、以下のいずれかのケースになり、ネームサーバはDS RRsetを通常の処理処理規則に従ったエラー応答を返すことができる
  - ネームサーバが別の手段でDS RRsetを得られる
  - DNSSEC処理規則以前の段階でDS RRsetが得られないことがわかっている

3.1.5 Responding to Queries for Type AXFR or IXFR

- DNSSECはDNSのZONE転送処理を変更しない。

- 署名付きZONEはRRSIG、DNSKEY、NSEC、DS RRsを持つが、これらのレコードはZONE転送処理の中では、特別な意味を持たない。
  - 権威ネームサーバは、ZONE転送送信前またはZONE転送受信後に、ZONEが適切に署名されているかを検証しなくてもよい。

全般	要求レベル
①Section2で規定した署名の要件をZONEが満たさない場合 ⇒権威ネームサーバはゾーン転送処理全てを 拒否してもよい 。	MAY
② ・ZONE転送の主要な目的は、全ての権威ネームサーバが 同一のZONEコピーを持つことを保証することである。 ・ZONE検証の実行を 選択した権威ネームサーバは、RRへの問合わせを選択的に拒否したり受容したりしてはならない 。	MUST NOT

3.1.5 Responding to Queries for Type AXFR or IXFR

- DS Rrsetsが権威を持つZONE転送
  - DS RRsetsを付加しなければならない。

- NSEC RRsが権威を持つZONE転送
  - NSEC RRsを付加しなければならない。
  - ZONE Cutの親側に存在するNSEC RRsは親ZONEの転送時に必ず付加されなければならない。
  - 子ZONEの頂点に存在するNSECは子ZONEの転送時に付加されなければならない。

- RRSIG RRsが権威を持つZONE転送
  - RRSIG RRsを付加しなければならない。

3.1.6 The AD and CD Bits in an Authoritative Response

- CDビットとADビットは、リゾルバと再帰ネームサーバ間のやりとりで使用するよう設計された。

- CDビットとADビットは ほとんどの場合、権威ネームサーバが行う問合わせ処理とは関係がない。

内容	要求レベル
①ネームサーバは権威を持つ応答を生成する際にCDビットをクリアすべきである。	SHOULD
②応答のAnswer section及びAuthority sectionに入れられたRRsetが全て信頼できない限り、ネームサーバは応答にADビットを設定してはならない。	MUST NOT
③ネームサーバ(セカンダリ)は、権威を持つZONEから得たデータを検証無しで信頼してもよい 。	MAY
④但し、ネームサーバ(セカンダリ)が権威を持つZONEを安全な手段(安全なZONE転送の仕組み等)で入手できない限り、検証無しの信頼をしてはならない。	MUST NOT
⑤明示的に設定されていない限り、このような 振る舞いをしてはならない	MUST NOT
⑥再帰ネームサーバは、再帰検索で得たデータを含む応答を生成する際に、Section3.2で規定するCDビットとADビットに関する規則に従わなければならない。	MUST

3.2 Recursive Name Servers

- 再帰ネームサーバとは、ネームサーバとリゾルバの両方の役割を担うものである。([RFC4033]規定)

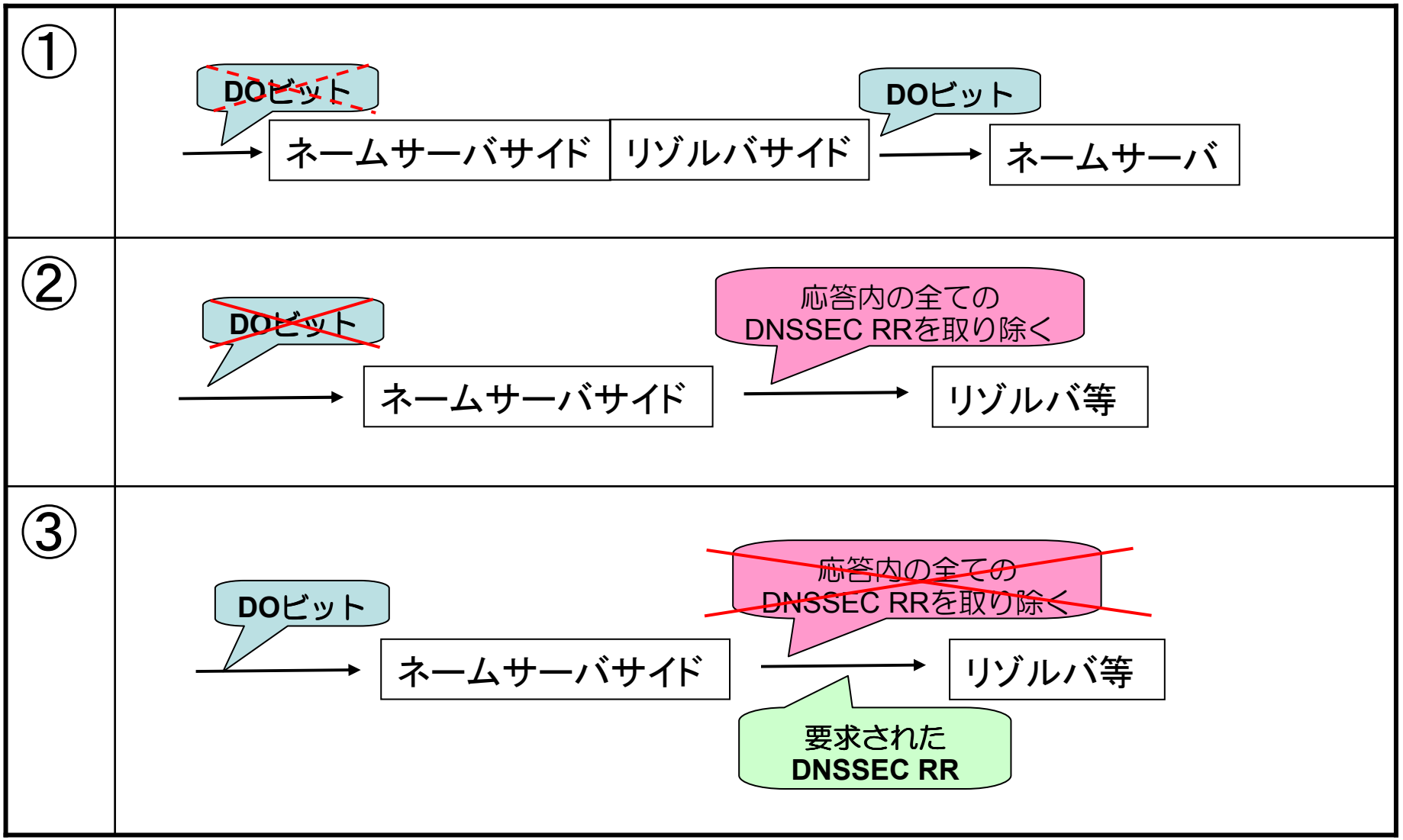
- このページ以降、説明を行う上での用語の定義
  - ネームサーバ機能実装部分　：　ネームサーバサイド
  - リゾルバ機能実装部分　：　リゾルバサイド

- リゾルバサイドは、通常のキャッシュ処理、ネガティブキャッシュ処理に関する規定に従う

3.2.1 The DO Bit

内容	要求レベル
①再帰ネームサーバのネームサーバサイドで受信した検索要求にDOビットが設定されているかにかかわらず、リゾルバサイドは、問い合わせ送信時にDOビットを設定しなければならない。	MUST
②DOビットを設定していない問い合わせを受けた場合、再帰ネームサーバのネームサーバサイドは、応答内に含まれる全てのDNSSEC RRを取り除かなくてはならない。	MUST
③問い合わせの中に明示的（DOビット設定有り）に要求されたDNSSEC RRタイプを取り除いてはいけない。 →要求されたDNSSEC RRを返す	MUST NOT

3.2.1 The DO Bit

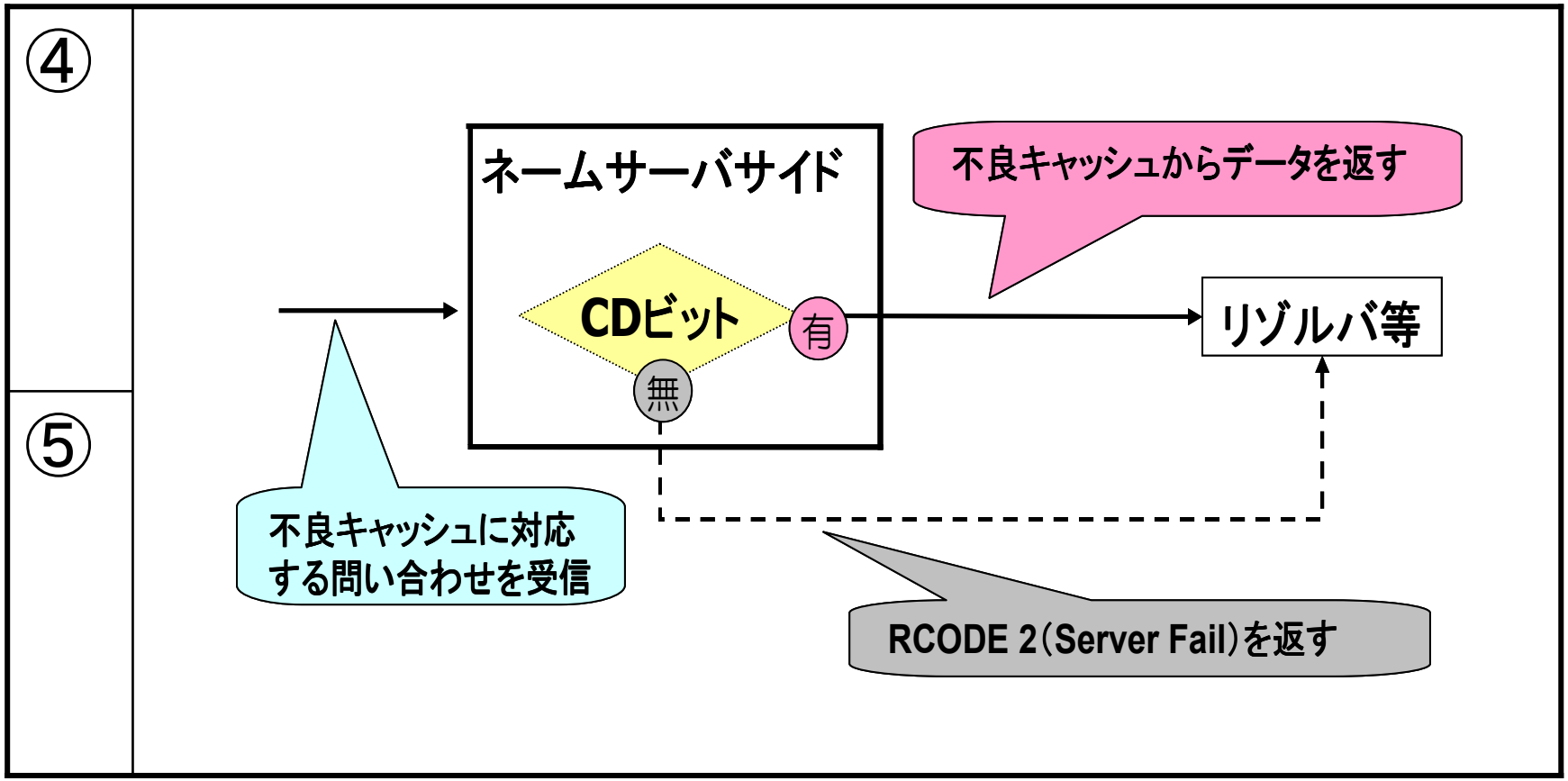


3.2.2 The CD Bit

・ リゾルバは、CDビットを設定してネームサーバに照会すると、署名検証を無効にすることができる。

内容	要求レベル
①CDビットが設定されているクエリに対し、ネームサーバサイドは、応答にCDビットの設定をコピーしなければならない。	MUST
②ネームサーバサイドは、クエリと共にCDビットの状態をリゾルバサイドに渡さなければいけない。	MUST
③クエリにCDビットが設定されている場合 →問い合わせの要求元であるリゾルバが自身で認証を実行する責任を負うため、再帰ネームサーバが干渉すべき事柄でないことを示している。	SHOULD
④不良キャッシュ（Section4.7参照）に対応するクエリを受信し、CDビットが設定されたいた場合 →ネームサーバは、不良キャッシュからデータを返すべきである。	SHOULD
⑤不良キャッシュに対応するクエリを受信し、CDビットが設定されていない場合 →ネームサーバは、RCODE 2（Server Fail）を返さなければならない。	MUST

3.2.2 The CD Bit



3.2.3 The AD Bit

内容	要求レベル
①応答内のAnswer section及びAuthority sectionの全RRsetが信頼できない場合 ⇒ネームサーバサイドは、ADビットを設定してはならない。	MUST NOT
②応答内のAnswer sectionの全RRset、及びAuthority sectionの任意のNegative RRsが信頼できる場合 ⇒ネームサーバサイドは、ADビットを設定すべきである。	SHOULD
③リゾルバサイドは、RRが信頼できるかの判断をする際にSection5の規定に従わなければならない。	MUST
④応答に含まれる署名無しCNAME RRsが信頼できるDNAME RRから生成されたのが明らかな場合 且つ、そのDNAME RRが[RFC2672]で規定された合意規則に従い応答に含まれていた場合 ⇒ネームサーバは、ADビットを設定しても良い。	MAY

3.3 Example DNSSEC Response

付録Bに応答パケットの例があります

4. Resolving

- 4.1 EDNS Support
- 4.2 Signature Verification Support
- 4.3 Determining Security Status of Data
- 4.4 configured Trust Anchors
- 4.5 Response Caching
- 4.6 Handing of the CD and AD Bits
- 4.7 Caching bad Data
- 4.8 Synthesized CNAMES
- 4.9 Stub Resolvers
  - 4.9.1 Handing of the DO Bit
  - 4.9.2 Handing of the CD Bit
  - 4.9.3 Handing of the AD Bit

4. Resolving

本セクションではセキュリティ対応リゾルバ機能を持つものの振る舞いを 規定する。

多くの場合、この機能はセキュリティ対応再帰ネームサーバの 一部であるが、単独で動作する(stand-alone)セキュリティ対応リゾルバも 多くの同じ要件を持つ。

セキュリティ対応再帰ネームサーバ固有の機能に ついてはセクション3.2で規定する。

＊本セクションで記載する要求レベルはRFC2119に記述されている通りに解釈される

4.1 EDNS Support

要求レベルMUSTの事項
・問い合わせ時にEDNS OPT pseudo-RR にDOビットを付加しなければならない。
・EDNS　OPT pseudo-RR内の『sender's UDP payload size』フィールドを使用して受信出来るメッセージサイズを広報しなければならない。

- サポートするオクテットサイズ

サイズ	要求レベル
1220	MUST
4000	SHOULD

4.2 Signature Verification Support

- 求められているサポート

内容	要求レベル
セクション5で規定する署名検証の仕組みをサポート	MUST
受信した応答全てに対してその仕組みを適用	SHOULD
ワイルドカード所有者名の検証	MUST

- 例外

リゾルバがキャッシュサーバ（再帰ネームサーバ）である時、問い合わせにCDビットが設定されていた場合。
何らかのアプリケーションにてセキュリティ検証を行わないように設定されている場合。
ローカルポリシーにより、問い合わせに対する検証が無効にされている場合

4.2 Signature Verification Support

- ワイルドカード所有者の検証を行う場合

内容	要求レベル
不足しているセキュリティRRの問い合わせを行ってもよい ※1	MAY
委任されたZONEの親側でNSEC RRが不足している場合、反復型リゾルバは親側のネームサーバに問い合わせる	MUST
委任されたZONEの親側でDSが不足している場合、反復型リゾルバは親側のネームサーバに問い合わせる ※2	MUST

※1
問い合わせで得られる情報は時間差がある為、検証を行うには不十分な場合がある。
例）オリジナルの問い合わせ後にゾーンの更新が行われた場合

※2
左から順番にラベルを取り除き問い合わせを行う。これをNS RRsetが見つかるか，ラベルが無くなるまで繰り返す。

4.3 Determining Security Stauts of Data

- リゾルバは以下の4つの状態を判断出来る必要がある。

Secure（安全）	指定したアンカーからそのRRsetまで署名付きDNSKEY RR及びDS RRの連鎖を構築出来る。
Insecure（安全でない）	指定したアンカーからそのRRsetまで署名付きDNSKEY RR及びDS RRの連鎖を構築出来ない。 ※1
Bogus（不適正）	RRsetまで信頼の連鎖を構築出来ると想定したが、署名検証に失敗した場合。もしくは存在すべき適切なDNSSEC RRが無いために信頼の連鎖が構築出来ない ※2
Indeterminate(不確定)	必要なDNSSEC RRを得られなかったため、そのRRsetが署名付きであるかを判断出来ない。 ※3

※1 特定のRRsetが署名無しゾーンに存在する場合や、署名無しゾーンの下位にRRsetが存在する場合等。

※2 攻撃の可能性がある。もしくは設定エラーか何らかのデータ改変を示す場合もある。

※3 適切なゾーンを持つネームサーバにアクセス出来ない場合に生じ得る

4.4 Configured Trust Anchors

- 求められている設定

内容	要求レベル
少なくとも一つの信頼出来る公開鍵またはDS RRを設定み組み込める機能	MUST
複数の信頼出来る公開鍵またはDS RRを設定に組み込める機能	SHOULD
リゾルバの起動時に鍵を組み込める適度に強固な仕組みを幾つかもつべき ※1	SHOULD

※1 ハードディスクやNAS等

またtrust anchorsは安全な方法で更新される鍵を扱う為注意が必要である。
安全な方法としては物理メディアを介するものや、鍵交換プロトコル、
他の帯域外（out of band）等の手段が考えられる。

4.5 Response Caching

- 求められている機能

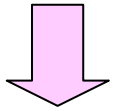
内容	要求レベル
最小単位のエントリーとして各応答をキャッシュする	SHOULD
エントリーが持つRRの中で、1つでも有効期限がきた時点で、関連するエントリーのキャッシュを破棄する	SHOULD

4.5 Response Caching

- 最小単位のエントリー構成

通常	QNAME QTYPE QCLASS
セクション3.1.3.2で規定した応答　※1	QNAME　QCLASS

※1　最初の問い合わせからデータをキャッシュし破棄するまでの間に、
権威を持つデータが動的更新等によって変更される可能性があるため。



1. RRSIGレコードを使用する事で、回答がワイルドカードから合成された事を推測出来る。
再帰ネームサーバは、最初に受信したオリジナル回答による**名前以外**の問い合わせに対して
応答を生成するために、このワイルドカードデータを保存し使用する。

2. 名前の不在を証明するNSEC RRを受信した場合、セキュリティ対応リゾルバは
その名前の範囲内にある名前全てが存在しないことを証明するために、
その **NSEC RRを再利用**することができる。

4.6 Handing of the CD and AD Bits

- 求められている機能

内容	要求レベル
応答内のRRsetに対してローカルポリシーが要求する何らかの認証処理を実行する為に、CDビットを付加する事が出来る	MAY
自分が理解できないヘッダービットを問合わせメッセージから応答メッセージに 無分別にコピーしてしまうような不適当な動作をするネームサーバからの影響を 避けるために、セキュリティ対応リゾルバは問合わせメッセージ生成時に ADビットをクリアしなければならない	MUST
応答メッセージが安全なチャネルから得られたか、安全なチャネルから応答を 得られなくてもメッセージヘッダーに注意するような設定が特にされていない 限り、リゾルバは応答に含まれるCDおよびADビットを無視しなければならない	MUST

4.7 Caching BAD Data

概念的にはネガティブキャッシュと同様。違いは有効な否定応答をキャッシュするのではなく、**特定の回答の検証に失敗したという事実**をキャッシュすること

- 求められている機能

内容	要求レベル
不要なDNSトラフィックを抑制する為に、署名が無効なデータを制限付きでキャッシュしてもよい	MAY
不良キャッシュを実装している場合は、そのキャッシュを利用したサービス不能攻撃を抑制する為に、幾つかの処理を行わなければならない	MUST
検証に失敗したRRsetは信頼出来るTTLを持たないので、TTLを自分で割り当てる必要がある	MUST
割り当てたTTLは攻撃の結果をキャッシュした影響を軽減する為に小さくすべきである	SHOULD
一時的な認証失敗をキャッシュしてしまわないように、認証が失敗した問い合わせを追跡記録すべきである	SHOULD
一定の閾値を超えて認証に失敗した場合は<QNAME,QTYPE,QCLASS>への問い合わせに対してのみ、 不良キャッシュを使用して応答すべきである	SHOULD
4.2で規定する規則に従った特定のRRsetの署名検証が求められない限り、RRsetへの問い合わせに対して不良キャッシュを使用してはならない	MUST NOT

4.8 Synthesized CNAMEs

- 求められている機能

内容	要求レベル
有効な署名付きDNAME RRから署名無しCNAME RRが生成された場合、DNAME RRの署名がCNAME RRも対象としている	MUST
署名が無いCNAME RRが存在するという理由で応答メッセージを拒否してはならない。また、リゾルバはこのようなCNAME RRをキャッシュに保存しても良い。	MAY

4.9 Stub Resolvers

- 求められている機能

内容	要求レベル
スタブリゾルバはDNSSEC RRをサポートしなければならない	MUST

※少なくともDNSSEC RRが含まれているだけの理由で誤った処理をしてはいけない

4.9.1 Handing of the DO Bits

4.9.1 DOビットの処理

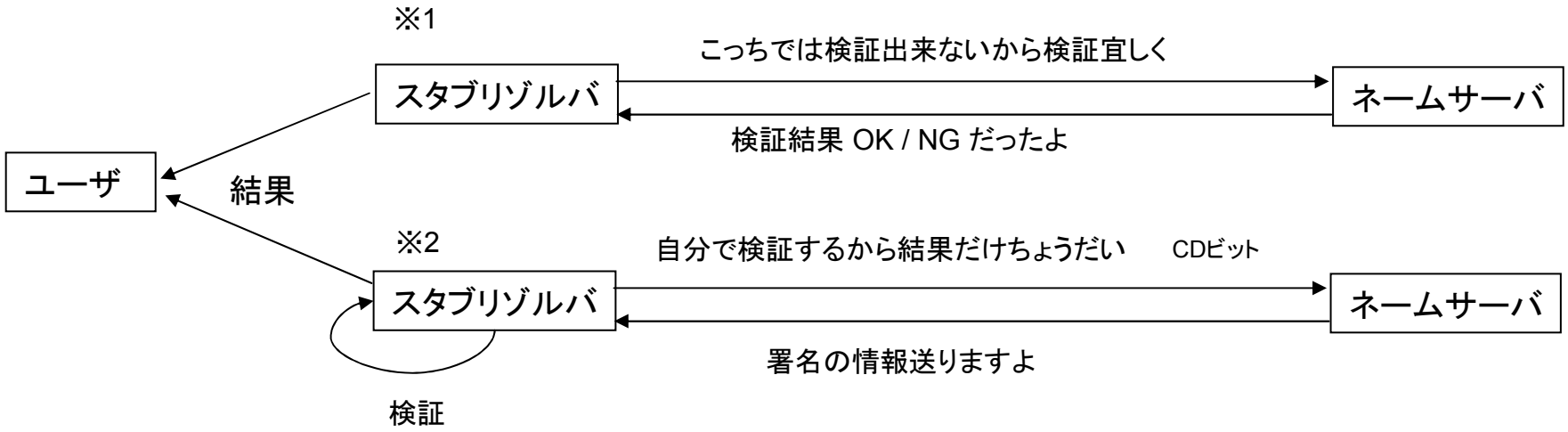
内容	要求レベル
検証機能無しセキュリティ対応スタブリゾルバはアプリケーションへの応答に再帰ネームサーバからのDNSSEC RRを含めてもよい	MAY
検証機能付きセキュリティ対応スタブリゾルバはDOビットを設定しなければならない	MUST

アプリケーションにDNSSEC RRを含めた応答を返そうとする検証機能無しスタブリゾルバは、再帰ネームサーバからDNSSEC RRを受信するためにDOビットを設定する必要がある。

4.9.2 Handing of the CD Bits

4.9.2 CDビットの処理

内容	要求レベル
検証機能無しセキュリティ対応スタブリゾルバはアプリケーション層から要求されない限り、送信時にCDビットを設定すべきではない ※1	SHOULD NOT
検証機能付きセキュリティ対応スタブリゾルバはCDビットを設定すべきである ※2	SHOULD



4.9.3 Handing of the AD Bits

4.9.3 ADビットの処理

内容	要求レベル
スタブリゾルバは再帰ネームサーバからの応答にADビットが設定されている事を調査しても良い。 (ADビットは暗号技術で検証された際に応答メッセージの回答部及び権威部付加される。)	MAY
セキュリティ対応スタブリゾルバが信頼出来るセキュリティ対応再帰ネームサーバから安全チャネルを通してデータを得た場合を除き、 如何なる場合でも自分の代わりに行ったとされる署名検証を信頼してはならない	MUST NOT
検証機能付きスタブリゾルバはADビットの有無に関わらず署名検証を行う為、応答メッセージにADビットが設定されているかを検査すべきではない	SHOULD NOT

5. Authenticating DNS Responses

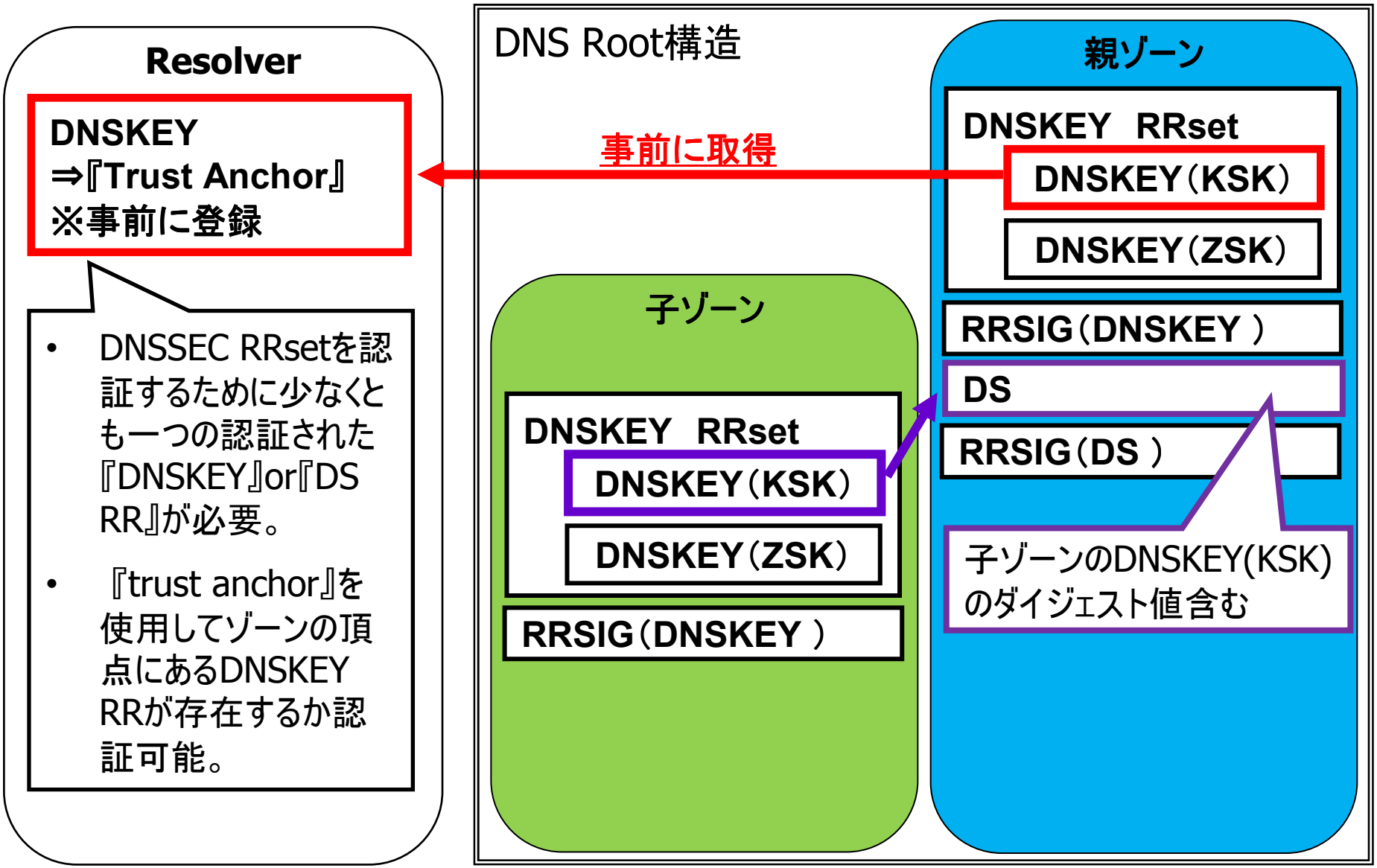
- 5.1 Special Consideratons for islands of Security
- 5.2 Authenticating Referrals
- 5.3 Authenticating an RRset with an RRSIG RR
  - -5.3.1 Checking the RRSIG RR Validity
  - -5.3.2 Reconstructing the Signed Data
  - -5.3.3 Checking the Signature
  - -5.3.4 Authenticating a Wildcard Expended RRset Positive Response
- 5.4 Authenticated Denial of Existence
- 5.5 Resolver Behavior When Signature Do Not Validate
- 5.6 Authentication Example

5. Authenticating DNS Responses

- DNSSEC RRsetを認証するために少なくとも一つの認証された『DNSKEY』or『DS RR』が必要。

- 『trust anchor』を使用してゾーンの頂点にあるDNSKEY RRが存在するか認証可能。

5. Authenticating DNS Responses

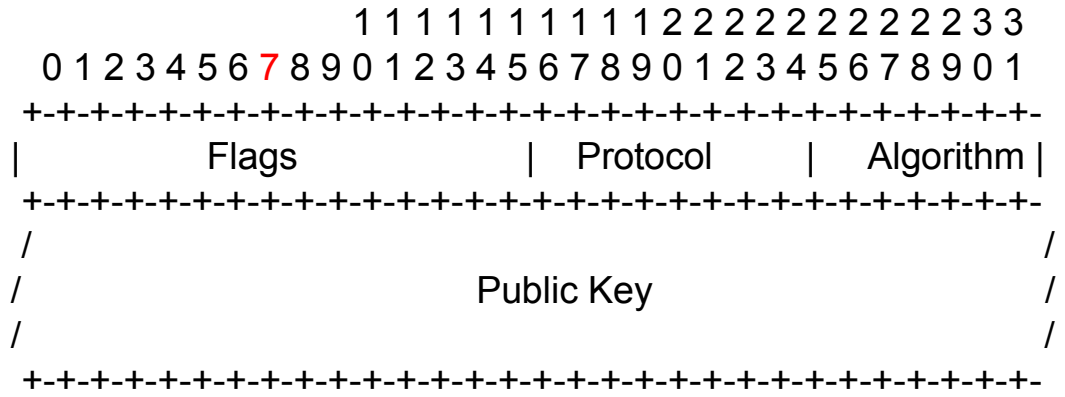


5. Authenticating DNS Responses

＜DNSKEY RR setを認証するためのResolverの処理＞

①ゾーン頂点にあるDNSKEY RRset内に起点となるDNSKEY RRが存在するか検証

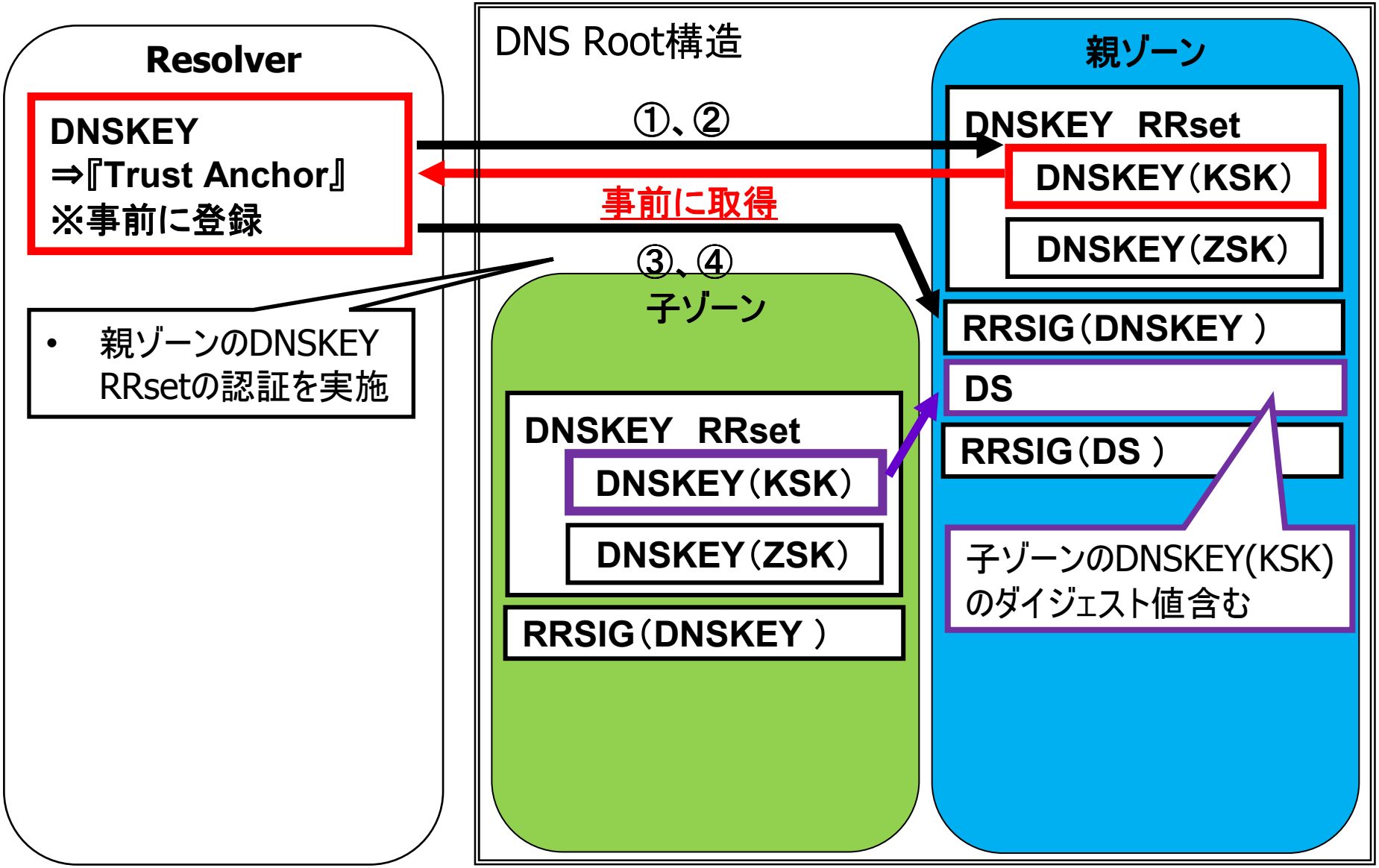
②そのDNSKEY RRのゾーン鍵フラグが設定されているか検証



③ゾーン頂点にあるDNSKEY RRsetの署名を持つRRSIG RRが存在するか検証

④そのRRSIG RRと起点となるDNSKEY RRの組み合わせによってDNSKEY RRsetを認証できるか検証（セクション5.3）

5. Authenticating DNS Responses

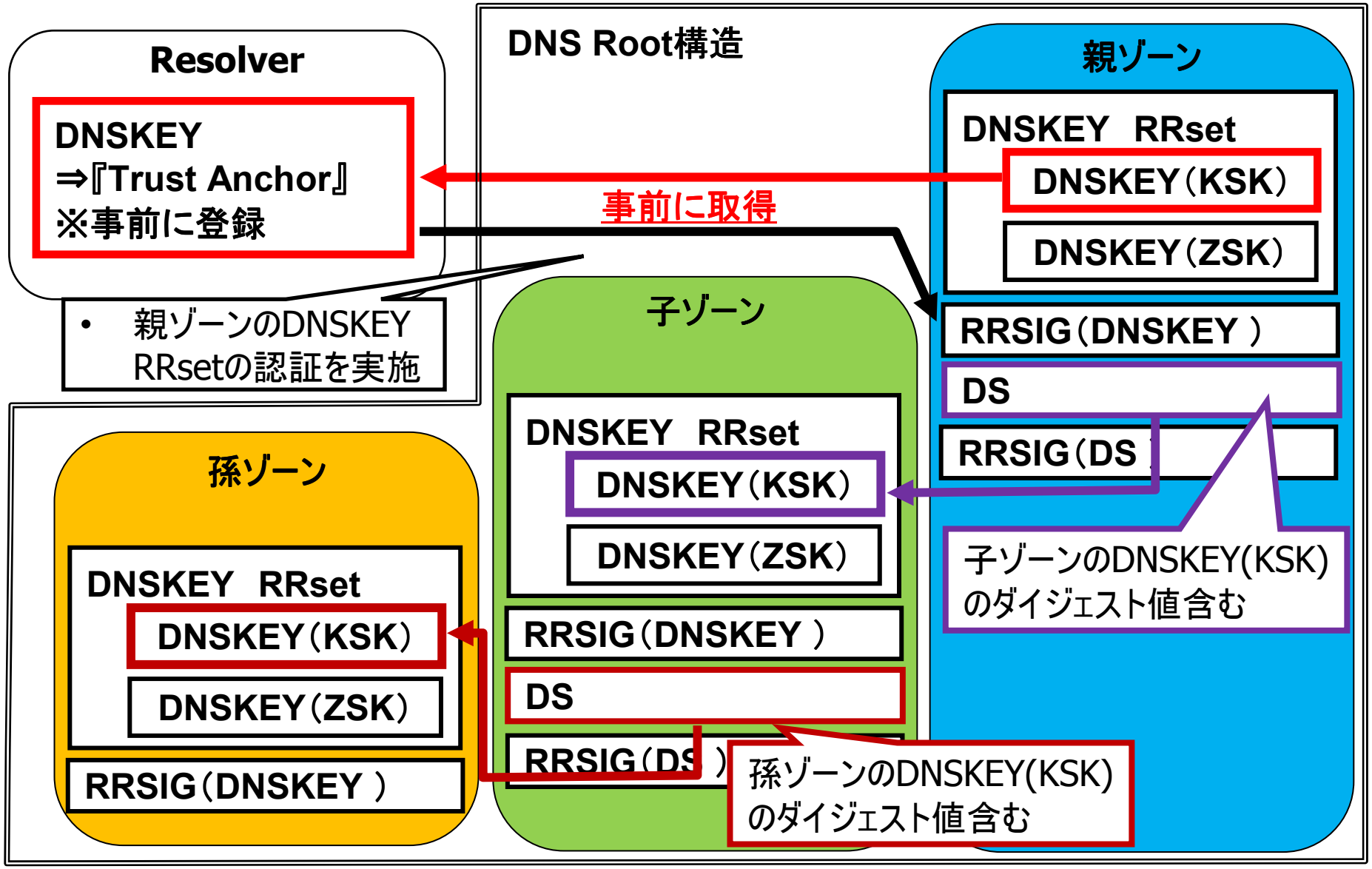


5. Authenticating DNS Responses

- リゾルバが起点となるDNSKEY RRを使用して、ゾーン頂点のDNSKEY RRsetを一度認証すれば、DS RRsetを使用してそのゾーンからの委任を認証をすることができる。

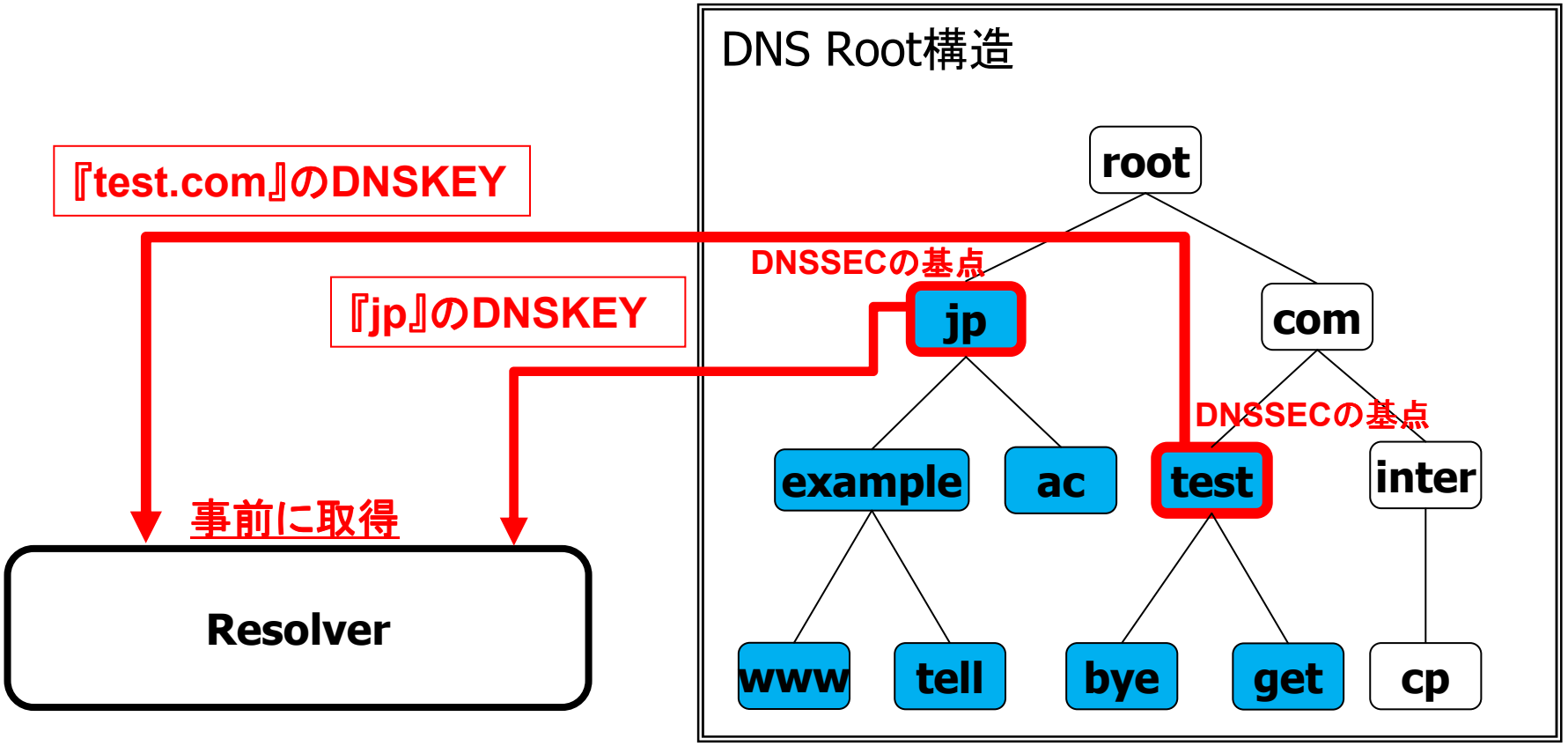
- リゾルバは起点となる鍵から認証を開始し、他のゾーン頂点にあるDNSKEY RRsetの取得とDS RRset の使用により、下方に向けて再帰的な認証を可能にする。

5. Authenticating DNS Responses



5.1 Special Considerations for Islands of Security

- Islands of Securityは親ゾーンからの認証の連鎖を構築できないため、基点となるゾーン鍵を取得する必要がある。⇒『Trust Anchor』
- もし検証者がゾーン鍵を取得できない場合は署名なしみなした運用に切り替える。



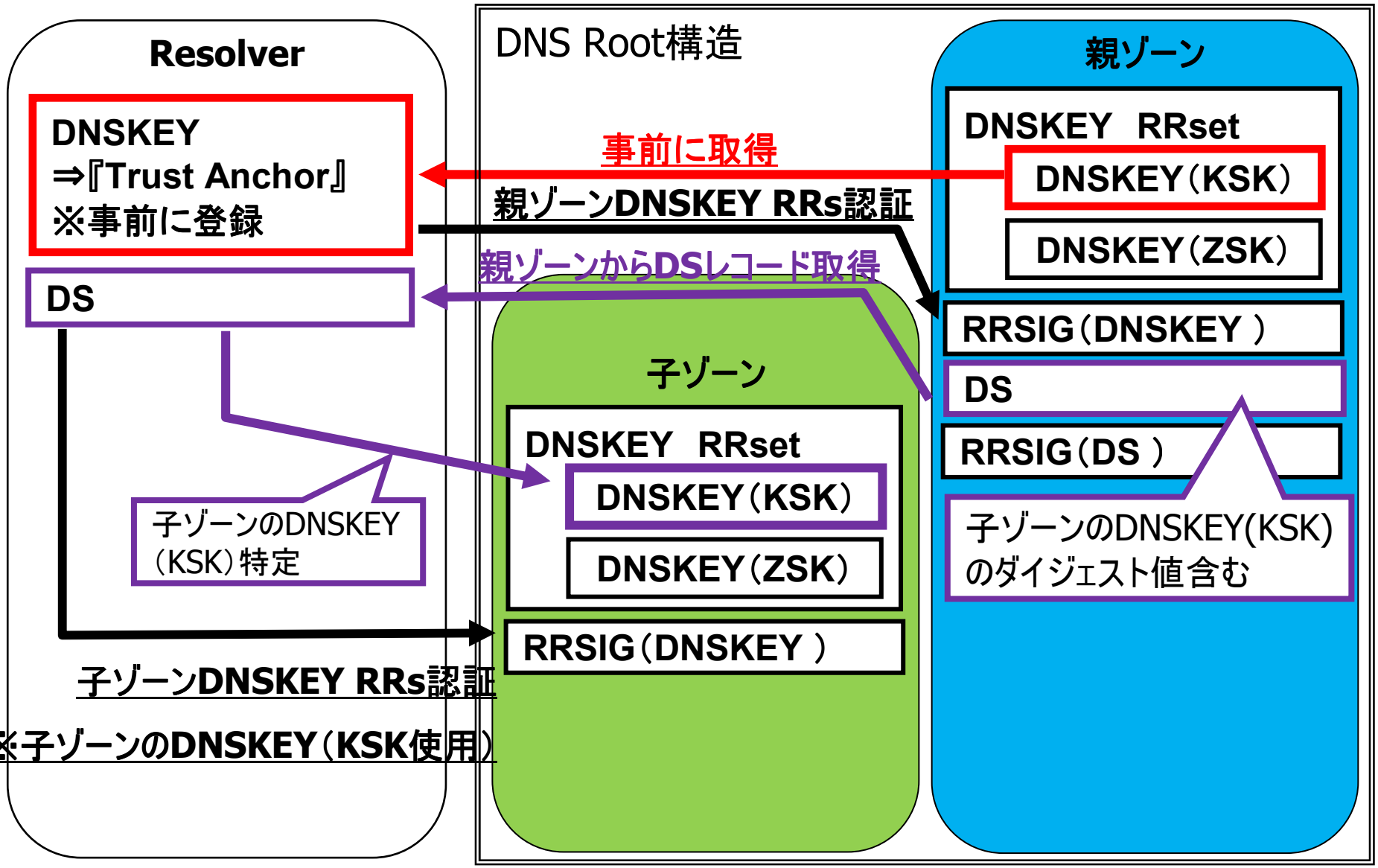
5.2 Authenticating Referrais

① 署名付きゾーンの頂点にあるDNSKEY RRsetが一度認証された場合、DS RRsetを使用して署名付き子ゾーンの認証をすることができる。

② DS RRは、子ゾーンの頂点にあるDNSKEY RRset内のDNSKEYを特定し、子ゾーンのDNSKEY RRの暗号ダイジェストを含む

③ リゾルバは子ゾーンのDNSKEY RRを使用して、子ゾーンの頂点にあるDNSKEY RRset全てを認証することができる。

5.2 Authenticating Referrais



5.2 Authenticating Referrais

DS RRが与えられた際、以下の条件が全て満たされれば子ゾーンの頂点にある

DNSKEY　RRsetを認証することができる。

① DS RRは、親ゾーンの頂点にあるDNSKEY　RRsetに含まれるDNSKEY RRによって認証済みである。

② DS RR内のアルゴリズムおよび鍵タグは、子ゾーンの頂点にあるDNSKEY RRset内のDNSKEY RRのアルゴリズムフィールドと鍵タグに一致。

③ DNSKEY RRの所有者名とRDATAがDS RRのダイジェストタイプフィールドで指定されるダイジェストフィールドでハッシュされている場合、得られたダイジェスト値はDS RRのダイジェストフィールドと一致

④ 子ゾーンにある、ダイジェストが一致したDNSKEY RRは、ゾーンフラグビットが設定されており、対応する秘密鍵によって子ゾーンの頂点にあるDNSKEY RRsetに署名している。

5.2 Authenticating Referrais

＜補足＞

① 親ゾーンから得られた参照がDS RRsetを含まない場合、委任された名前に関する DS RRsetが存在しないことを証明する署名付きNSEC RRsetを含めること。

② DNSSEC対応リゾルバは参照にDS RRsetもDS RRsetの不在証明をするNSEC RRsetも含まれない場合、その親のネームサーバに対してDS RRsetも関する問合せをしなければならない。

③ ゾーンにDS RRsetが存在しないことを証明するNSEC RRset を検証者が認証した場合、親から子に至る経路は存在しない。

④ リゾルバが子ゾーン内、又は、子ゾーンの下位ゾーンにある起点となるDNSKEY又は、DS RRを持っている場合、認証経路を再構築してもよい。

5.3 Authenticating an RRset with RRSIG RR

① 検証者は初めにRRSIG RRを精査する。

-RRset の署名を持っているか

-有効期限は過ぎていないか等

② RRSIG RDATA（署名フィールドを除く）を署名対象のRRsetの正規形式に付け加え、正規形式の署名付きデータを形成

③ 公開鍵と署名を使用して署名付きデータを認証する。

5.3.1 Checking the RRSIG RR validity

① 検証者は初めにRRSIG RRを精査する。

規則内容	要求レベル
①RRSIG RRとRRsetは同じ所有者を持ち、同じクラスで なければならない。	MUST
②RRSIG RRの署名者名フィールドはRRsetを含むゾーン の名前でなければならない	MUST
③RRSIG RRの署名対象フィールドはRRsetのタイプと同じでなければならない	MUST
④ RRsetの所有者名のラベル数は、RRSIG RRのラベルフィールドの値と同じかそれよりも大きくなければならない	MUST
⑤検証者の現在時刻を表現する値は、RRSIG RRの有効期間終了フィールドに記載された時間と同じかそれよりも小さくなければならない。	MUST
⑥検証者の現在時刻を表現する値は、RRSIG RRの有効期間開始フィールドに記載された時間と同じかそれよりも大きくければならない。	MUST
⑦RRSIG RRの署名者、アルゴリズム及び鍵タグフィールドは、ゾーン頂点にあるDNSKEY RRsetに含まれるDNSKEYの所有者名、アルゴリズム及び鍵タグフィールドに一致しなければならない	MUST
⑧ DNSKEY RRsetはゾーン頂点にあるDNSKEY RRsetに含まれていなければならず、また、ゾーンビットフラグビット（DNSKEY RDATAのフラグビット7）のフラグが設定されていなくてはならない。	MUST

5.3.1 Checking the RRSIG RR validity

① RRSIG RRとRRsetは同じ所有者を持ち、同じクラスで
なければならない。

※class=IN

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=

5.3.1 Checking the RRSIG RR validity

② RRSIG RRの署名者名フィールドはRRsetを含むゾーンの名前でなければならない

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=

5.3.1 Checking the RRSIG RR validity

③ RRSIG RRの署名対象フィールドはRRsetのタイプと同じでなければならない

```
example.  3600 NS ns1.example.
example.  3600 NS ns2.example.
          3600 RRSIG NS 5 1 3600
          20040509183619 20040409183619 38519 example.
          gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
          EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
          4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
          RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
          0HjMeRaZB/FRPGfJPajngcq6Kwg=
```

5.3.1 Checking the RRSIG RR validity

④ RRsetの所有者名のラベル数は、RRSIG RRのラベルフィールドの値と同じかそれよりも大きくなければならない

```
example.  3600 NS ns1.example.
example.  3600 NS ns2.example.
          3600 RRSIG NS 5 1 3600
          20040509183619 20040409183619 38519 example.
          gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
          EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
          4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
          RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
          0HjMeRaZB/FRPGfJPajngcq6Kwg=
```

5.3.1 Checking the RRSIG RR validity

⑤ 検証者の現在時刻を表現する値は、RRSIG RRの有効期間終了フィールドに記載された時間と同じかそれよりも小さくなければならない。

example. 3600 NS ns1.example.
example. 3600 NS ns2.example.
3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=

5.3.1 Checking the RRSIG RR validity

⑥ 検証者の現在時刻を表現する値は、RRSIG RRの有効期間開始フィールドに記載された時間と同じかそれよりも大きくければならない。

```
example.  3600 NS ns1.example.
example.  3600 NS ns2.example.
          3600 RRSIG NS 5 1 3600
          20040509183619 20040409183619 38519 example.
          gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
          EuivWc+wd1fmbNCyql0Tk7lHTX6UOxc8AgNf
          4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
          RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
          0HjMeRaZB/FRPGfJPajngcq6Kwg=
```

5.3.1 Checking the RRSIG RR validity

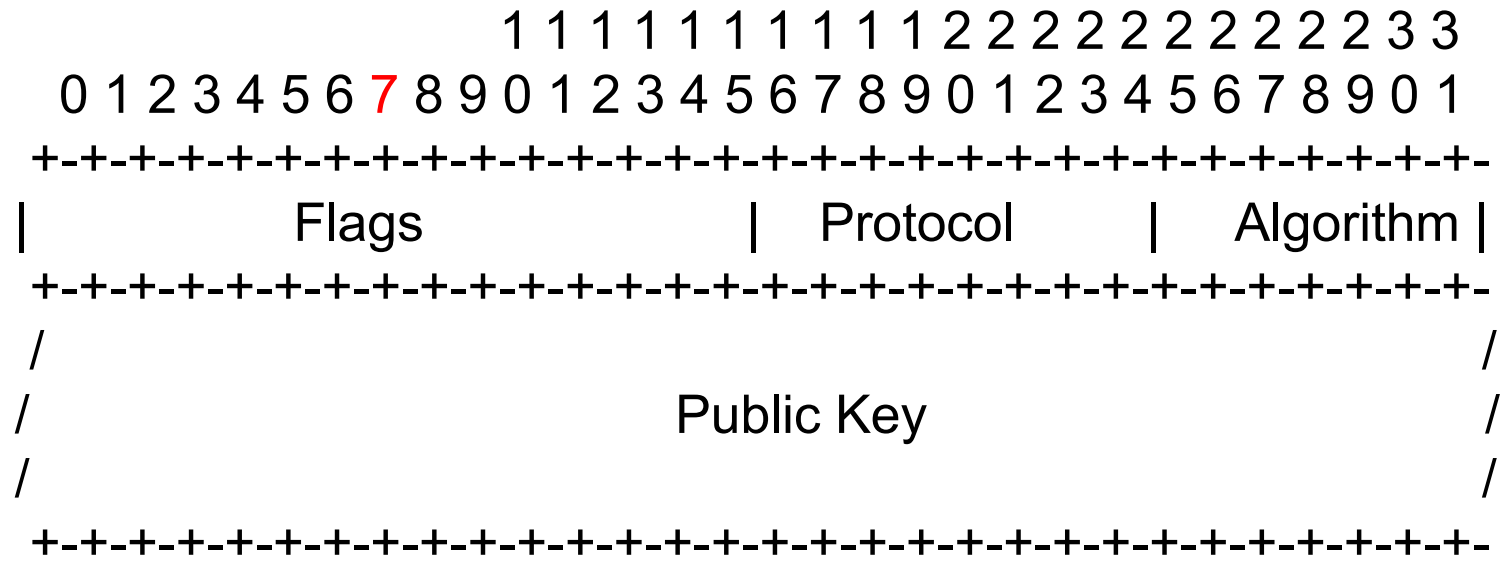
⑦ RRSIG RRの署名者、アルゴリズム及び鍵タグフィールドは、ゾーン頂点にあるDNSKEY RRsetに含まれるDNSKEYの所有者名、アルゴリズム及び鍵タグフィールドに一致しなければならない

3600 DNSKEY 256 3 5 AQOy1bZV･･･ ;key id = 38519

3600 RRSIG NS 5 1 3600
20040509183619 20040409183619 38519 example.
gl13F00f2U0R+SWiXXLHwsMY+qStYy5k6zfd
EuivWc+wd1fmbNCyql0Tk7IHTX6UOxc8AgNf
4ISFve8XqF4q+o9qlnqIzmppU3LiNeKT4FZ8
RO5urFOvoMRTbQxW3U0hXWuggE4g3ZpsHv48
0HjMeRaZB/FRPGfJPajngcq6Kwg=

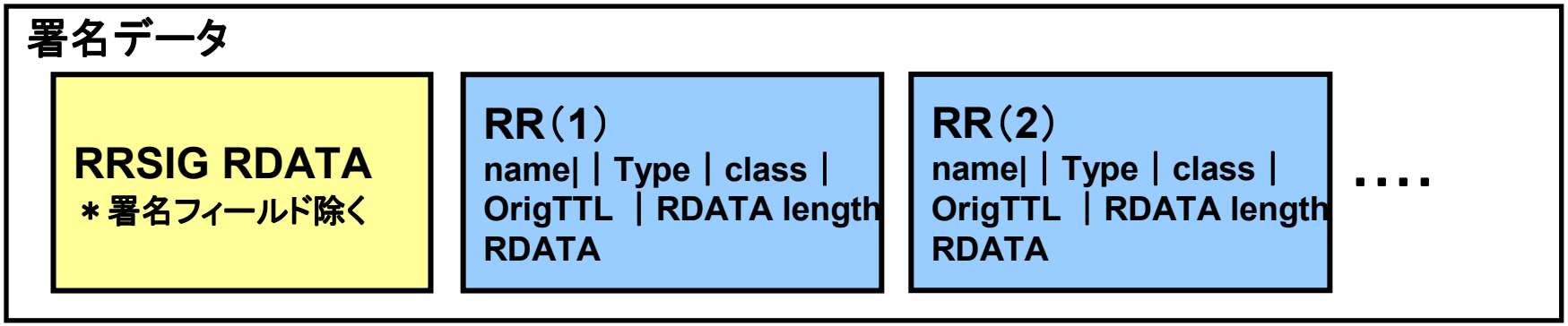
5.3.1 Checking the RRSIG RR validity

⑧DNSKEY RRsetはゾーン頂点にあるDNSKEY RRsetに含まれていなければならず、また、ゾーンビットフラグビット（DNSKEY RDATAのフラグビット7）のフラグが設定されていなくてはならない。



5.3.2 Reconstructing the Signed Data

② オリジナルの署名付きデータの再構成



①『class』はRRsetのクラスである

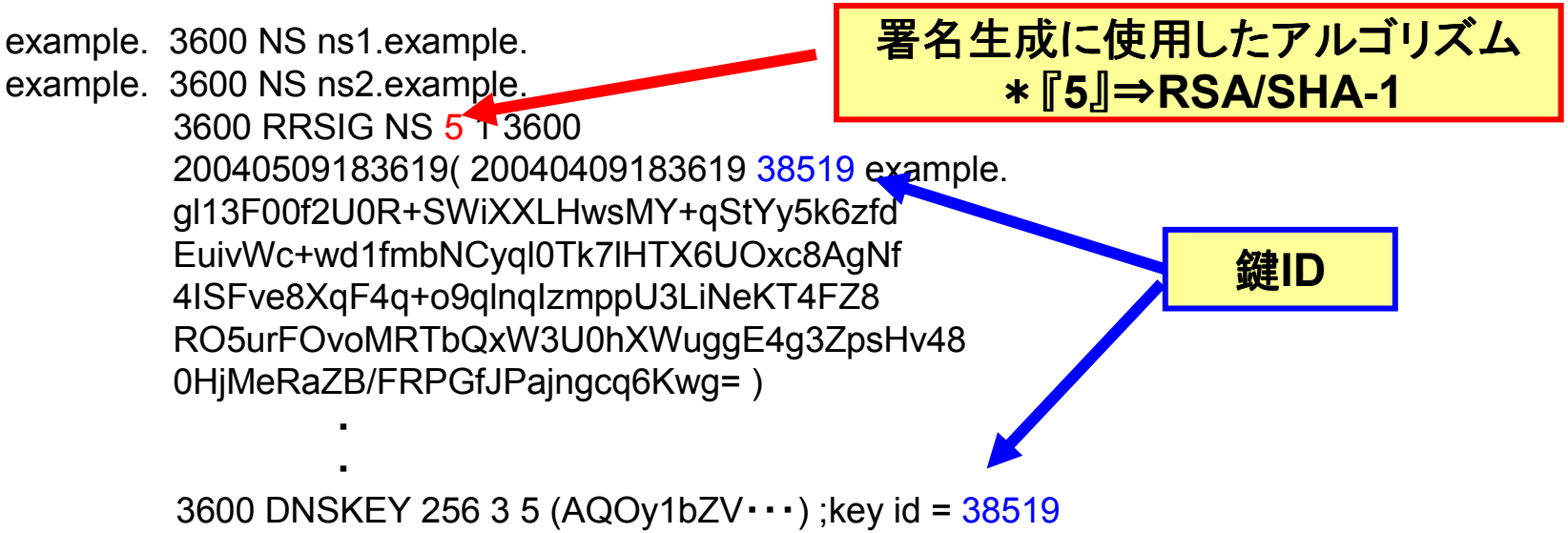
②『Type』はRRsetのタイプである。

③『OrigTTL』はRRSIGオリジナルTTLフィールドから得られた値

④『RDATA』フィールド内の名前はすべて正規形式（RFC4034参照）

5.3.3 Cheaking The Signature

③　公開鍵と署名を使用して署名付きデータを認証する



① RRSIG RRのアルゴリズムフィールドは署名生成に使用したアルゴリズムを特定する

②署名そのものはRRSIG RDATAの署名フィールドに保存

③署名の検証に使用する公開鍵は対応するDNSKEY RRの公開鍵フィールドに保存される。

5.3.3 Cheaking The Signature

★DNSKEY RRが2つ以上になる場合

条件に一致した公開鍵を1つずつ試していき、署名検証に成功するか、試すべき鍵が無くなるまで繰り返すことによってのみ、正式なDNSKEY RRを決定することができる。

★RRSIG RRのラベルフィールドがRRsetの完全修飾所有者名のラベル数と一致しない場合

RRsetが無効。もしくは、ワイルドカード展開によって得られたものかのどちらか。
リゾルバはRRsetを信頼できるとみなす前に、ワイルドカード展開が適切に行われたかを検証しなければならない。（MUST）

5.3.3 Cheaking The Signature

★リゾルバがRRsetを信頼できると判断した場合

検証者はRRSIG RRと認証されたRRsetに含まれる各RRのTTLを下記に示す値の最小値を超えないように設定しなければならない。（MUST）

- 応答を受信した際のRRsetのTTL
- 応答を受信した際のRRSIG RRのTTL
- RRSIG RRのオリジナルTTLフィールドに含まれる値
- RRSIG RRの署名有効期間終了時刻と現在時刻との差

5.3.4 Authenticating a Wildcard Expanded RRset Positive Response

★ **RRset**の所有者のラベル数が、署名を持つ**RRSIG RR**のラベルフィールドの値より大きい場合

⇒RRsetとその署名をもつRRSIG RRはワイルドカード展開の結果生成されたもの

★ リゾルバが受信した応答は、応答を認証するために必要な**NSEC RR**をすべて含むべきであること。（＊参照　**section 3.1.3**）

5.4 Authenticated Denial of Existence

Rules

Ⅰ）queried RR nameとNSEC RRのowner nameが同じ場合

NSEC RRのType bit mapを検証し判断する　例) ai.example.　MXの不在証明

example.	NSEC	a.example.	NS SOA MX RRSIG NSEC DNSKEY
a.example.	NSEC	ai.example.	NS DS RRSIG NSEC
ai.example.	NSEC	b.example.	A HINFO AAAA RRSIG NSEC
b.example.	NSEC	ns1.example.	NS RRSIG NSEC
ns1.example.	NSEC	ns2.example.	A RRSIG NSEC
ns2.example.	NSEC	*.w.example.	A RRSIG NSEC
*.w.example.	NSEC	x.w.example.	MX RRSIG NSEC
x.w.example.	NSEC	x.y.w.example.	MX RRSIG NSEC
x.y.w.example.	NSEC	xx.example.	MX RRSIG NSEC
xx.example.	NSEC	example.	A HINFO AAAA RRSIG NSEC

※最初に戻る

5.4 Authenticated Denial of Existence

Rules

Ⅱ）queried RR nameとNSEC RRのowner nameが違う場合

①②を実施した結果で、判断をする　例) az.example.の不在証明

example.	NSEC	a.example.
a.example.	NSEC	ai.example.
ai.example.	NSEC	b.example.
b.example.	NSEC	ns1.example.
ns1.example.	NSEC	ns2.example.
ns2.example.	NSEC	*.w.example.
*.w.example.	NSEC	x.w.example.
x.w.example.	NSEC	x.y.w.example.
x.y.w.example.	NSEC	xx.example.
xx.example.	NSEC	example.

※最初に戻る

①az.example.
を探しているんだけど・・・
ai.exampleの次は、b.example.
このZONEには存在しない

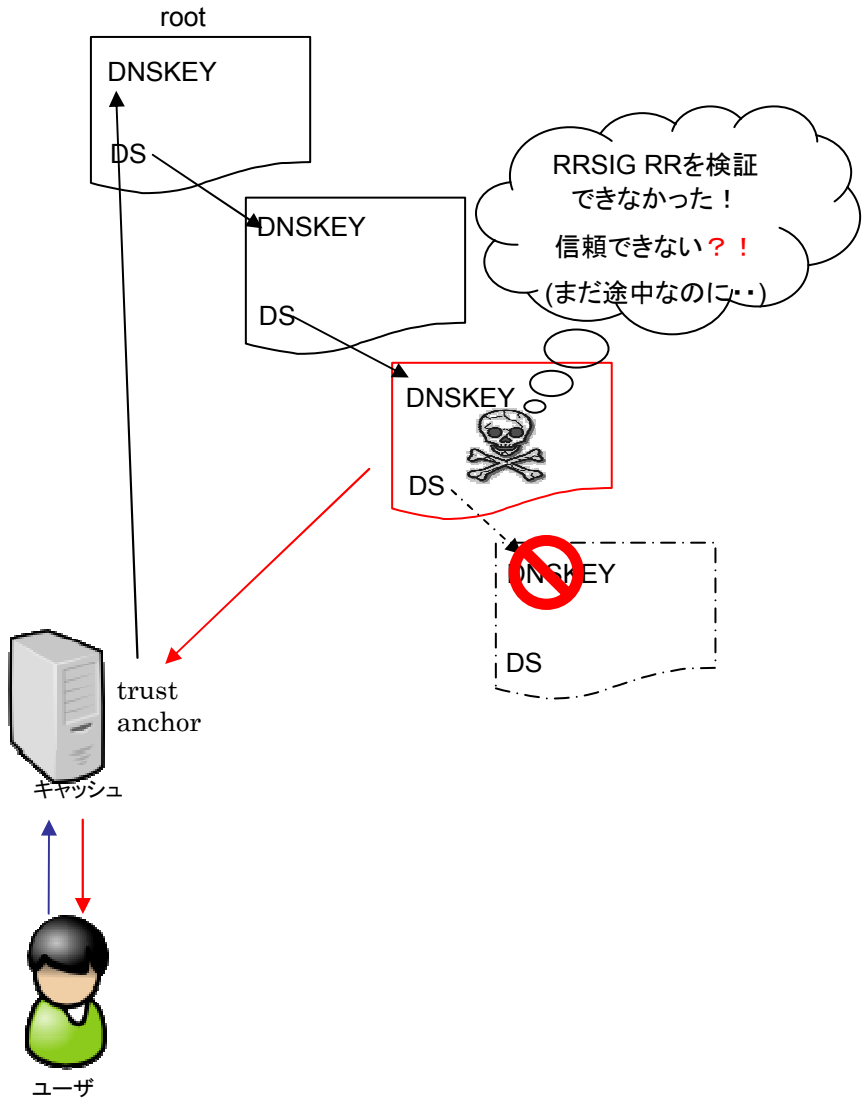
②ワイルドカード展開だ！！

5.4 Authenticated Denial of Existence

- Resolverは、署名済みNSEC RRを使用して、署名付きZONE内に特定のRRsetが存在しないことを証明する　　参照：Rules　Ⅰ
- queried RRsetの不在検証に必要なNSEC RRを全て取得するための処理を行う
  ーワイルドカードRRsetの不在も証明する必要ある
  ーしかし、特定の問い合わせへの回答だけに
  　労力を注がないように抑制する

参照：Rules　Ⅱ

5.5 Resolver Behavior When Signatures Do Not Validate



- CD bit なし
  - ー回答は、ServFail
  - ー検証失敗のRRsetは
    キャッシュできる

- CD bit あり
  - ー回答は、data from BAD cache

参照：Sections 4.7

※ data from BAD cache：認証に失敗した特定の
〈QNAME,QTYPE,QCLASS〉

5.6 Authentication Example

付録Cに認証処理の例があります

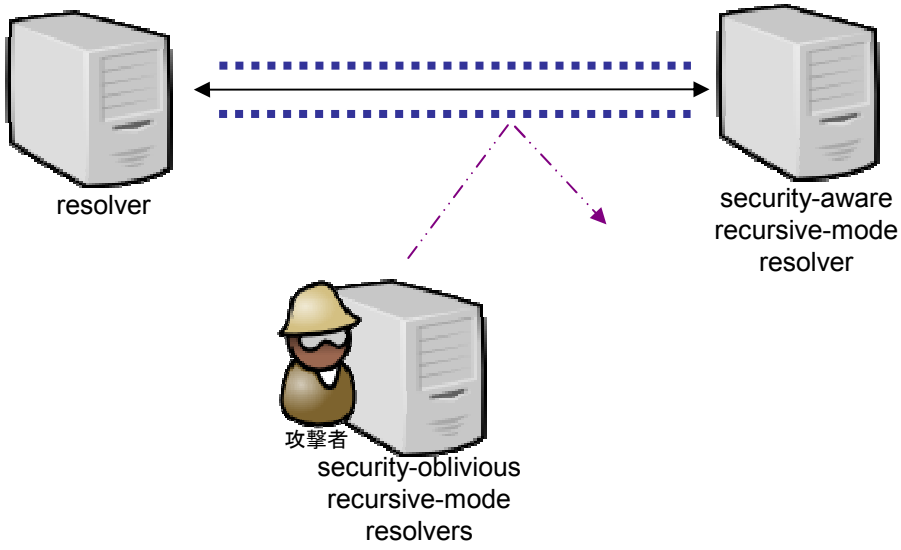
6. IANA Considerations

- RFC2535
  - CDおよびADビットを予約
- RFC3655
  - ADビットの意味を再定義
- RFC2671
  - EDNS導入
- RFC3225
  - DNSSEC OKビット定義

7. Security Considerations

- 本文書はRRsetに署名/認証する方法を規定
- security-aware recursive-mode resolverは、CD/ADビットを使用する場合、安全なchannelが必要になる

＊参照：Sections 3.2.2 & Sections 4.9



7. Security Considerations

- ローカルポリシーの策定
  - 認証失敗後の動作はアプリに依存
  - アプリの振る舞いに注意しましょう
  - 提供サービスに影響を与える可能性あり

8. Acknowledgements

9. References