RFC 5702　プロトコル理解 SWG

2010年07月22日

株式会社NTTPCコミュニケーションズ
加藤 優佐

Abstract

このドキュメントでは、DNSSEC（RFC 4033, RFC 4034, RFC 4035 ）において、RSA/SHA-256, RSA/SHA-512を利用したDNSKEYとRRSIGリソースレコードを作成する方法を説明する。

1.Introduction

- RFC4034では、DNSKEYとRRSIGリソースレコードを格納する方法と、利用する暗号化アルゴリズムの指定の仕様が記載されている。
- このドキュメントではRSA/SHA-256,RSA/SHA-512のアルゴリズムについて拡張し、そのハッシュアルゴリズムでのDNSKEY格納と、RRSIG作成の仕様を説明する。
- このドキュメントはDNSSEC、RSA、SHA-2[FIPS. 180-3.2008]関連のアルゴリズムの理解を前提としている。

1.Introduction

- このドキュメントでは、読みやすさのため、SHA-256,SHA-512をまとめてSHA-2と表記する。
- 区別が必要な場合はそれぞれの名前を用いる。
- 同様にRSA/SHA-256,RSA/SHA-512をRSA/SHA-2と表記する。
- 一般的にはSHA-224,SHA-384についてもSHA-2と呼ぶこともあるが、DNSSECではSHA-256,SHA-512しか利用しないため、それのみを指すこととする。
- “MUST”“SHALL”等の言葉の指す意味についてはRFC2119に記載されている通り。

2.DNSKEY Resource Records

- DNSSECの署名にRSA/SHA-1を用いた、DNSKEY RRのフォーマットについてはRFC4034、RFC3110に記載されています。

2-1.RSA/SHA-256 DNSKEY Resource Records

- RSA/SHA-256を使用した公開鍵はアルゴリズム番号「８」でDNSKEY RRに格納される。

- RFC3110にもあるが、RSA/SHA-256の鍵長は512bit以下でも、4096bit以上でもいけない。
  （512bit～4096bitで鍵を作成する）

2-2.RSA/SHA-512 DNSKEY Resource Records

- RSA/SHA-512を使用した公開鍵はアルゴリズム番号「10」でDNSKEY RRに格納される。

- RFC3110にもあるが、RSA/SHA-512の鍵長は1024bit以下でも、4096bit以上でもいけない。
  （1024bit～4096bitで鍵を作成する）

3.RRSIG Resource Records

- RRSIG RRの署名フィールドの値は、RSASSA-PKCS1-v1_5の署名方式にて、後述の通り計算される。
- signatureデータに先立つ、RDATAフィールドの値はRFC4034で定められている。

hash = SHA-XXX（data）

- XXXは利用するアルゴリズムに応じて、256か512になる。（FIPS PUB 180-3）
- dataはRFC4034で定義された、署名されたRR-SETのデータフォーマットになる。

3.RRSIG Resource Records

signature =（00 | 01 | FF* | 00 | prefix | hash）** e（mod n）

- “00” “01” “FF” “00” はそれぞれのオクテットに対応した、固定の16進の値。
- “e” は署名したRSA鍵の指数。
- “n” は署名鍵の公開鍵 。
- FFのオクテットは、署名者の公開鍵の長さ（”n”）と等しい、括弧で連結された項の全長と同じ回数だけ、繰り返されなくてはならない。

3.RRSIG Resource Records

***signature =（00 | 01 | FF* | 00 | prefix | hash）** e（mod n）*

- “prefix” は標準の暗号化ライブラリの使用を、より容易にするために意図されている。

- これらの仕様はPKCS #1 v2.1のRSASSA-PKCS1-v1_5（RFC3447のSection 8.2）の仕様および、EMSA-PKCS1-v1_5 encoding（RFC3447のSection 9.2）から抜粋されている。

3-1.RSA/SHA-256 RRSIG Resource Records

- RSA/SHA-256 の署名は、DNSではアルゴリズム番号 ”8” のRRSIGリソースレコードとして配置される。

- prefixはASN.1 DER SHA-256 アルゴリズムで呼び出される prefixであり、PKCS #1 v2.1（RFC3447）で以下の様に表されている。

hex 30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20

3-2.RSA/SHA-512 RRSIG Resource Records

- RSA/SHA-512 の署名は、DNSではアルゴリズム番号 ”10” のRRSIGリソースレコードとして配置される。

- prefixはASN.1 DER SHA-512 アルゴリズムで呼び出される prefixであり、PKCS #1 v2.1（RFC3447）で以下の様に表されている。

hex 30 51 30 0d 06 09 60 86 48 01 65 03 04 02 03 05 00 04 40

4.Deployment Considerations

4-1.Key Sizes

- 2章での制限は別にして、このドキュメントではどのような鍵のサイズを使用したら良いかについての言及はしない。
- それは運用上の問題であり、主に環境と使用用途に依存する。
- 詳しい情報が知りたければNIST SP800-57を参照すること。

4-2.Signature Sizes

- これらの署名アルゴリズムでは、署名のサイズは署名の際に使用されるハッシュアルゴリズムではなく、鍵のサイズに依存する。

- ゆえに、鍵長が同じであればRSA/SHA-256かRSA/SHA-512で作成されたRRSIG RRは、RSA/SHA-1で作成されたものと同じサイズになる。

5.Implementation Considerations

5-1.Support for SHA-2 Signatures

- DNSSECを意識した実装では、このドキュメントで定義されている通りに、RSA/SHA-2アルゴリズムで作成されたRRSIGとDNSKEY RRをサポートする必要がある。

5-2.Support for NSEC3 Denial of Existence

- RFC5155は、署名されたゾーンが、不在証明を示すNSECレコードと同様にNSEC3を使用できることを示すために、その署名アルゴリズムを表す、新しいアルゴリズム識別子を定義している。

  ex.) アルゴリズム番号:7　NSEC3RSA/SHA1

- RFC5155が定義される以前、本来知りうることが出来なかったレコードを取得してしまうことが起きたため、NSEC3はそれを保護するために選ばれた。

- このドキュメントはこのアルゴリズム（NSEC3）を定義しない。

5-2.Support for NSEC3 Denial of Existence

- RFC5155で定義されている通り、RSA/SHA-2が実装されたDNSSECのバリデータは、ハッシュアルゴリズム1を使用したNSECとNSEC3の両方で、不在応答を検証できなければならない。

- NSEC3を実装していない権威サーバは、まだRSA/SHA-2を使用したゾーンにNSEC不在証明を返しているかもしれない。

6.Examples

6-1.RSA/SHA-256 Key and Signature

- 秘密鍵

```
Private-key-format: v1.2
Algorithm:        8 (RSASHA256)
Modulus:          wVwaxrHF2CK64aYKRUibLiH30KpPuPBjel7E8ZydQW1HYWHfoGm
                  idzC2RnhwCC293hCzw+TFR2nqn8OVSY5t2Q==
PublicExponent:   AQAB
PrivateExponent:  UR44xX6zB3eaeyvTRzmskHADrPCmPWnr8dxsNwiDGHzrMKLN+i/
                  HAam+97HxIKVWNDH2ba9Mf1SA8xu9dcHZAQ==
Prime1:           4c8IvFu1AVXGWeFLLFh5vs7fbdzdC6U82fduE6KkSWk=
Prime2:           2zZpBE8ZXVnL74QjG4zINIDfH+EOEtjJJ3RtaYDugvE=
Exponent1:        G2xAPFfK0KGxGANDVNxd1K1c9wOmmJ51mGbzKFFNMFk=
Exponent2:        GYxP1Pa7CAwtHm8SAGX594qZVofOMhgd6YFCNyeVpKE=
Coefficient:      icQdNRjlZGPmuJm2TladubcO8X7V4y07aVhX464tx8Q=
```

6-1.RSA/SHA-256 Key and Signature

- DNSKEYレコード

ZSK　　RSA/SHA-256

```
example.net.        3600    IN  DNSKEY   ( 256  3  8  AwEAAcFcGsaxxdgiuuGmCkVl
                              my4h99CqT7jwY3pexPGcnUFtR2Fh36BponcwtkZ4cAgtvd4Qs8P
                              kxUdp6p/DlUmObdk= )

{id = 9033 (zsk), size = 512b}
```

- RRSIGレコード

RSA/SHA-256

```
www.example.net. 3600   IN  A   192.0.2.91
www.example.net. 3600   IN  RRSIG   (A 8 3 3600 20300101000000 20000101000000 9033
                              example.net. kRCOH6u7l0QGy9qpC9
                              l1sLncJcOKFLJ7GhiUOibu4teYp5VE9RncriShZNz85mwlMgNEa
                              cFYK/lPtPiVYP4bwg==)
```

6-2.RSA/SHA-512 Key and Signature

Private-key-format: v1.2	
Algorithm:	*10 (RSASHA512)*
Modulus:	*0eg1M5b563zoq4k5ZEOnWmd2/BvpjzedJVdflsDcMuuhE5SQ3pfQ7qmdaeMlC6Nf8DK GoUPGPXe06cP27/WRODtxXquSUytkO0kJDk8KX8PtA0+yBWwy7UnZDyCkynO00Uu k8HPVtZeMO1pHtlAGVnc8VjXZlNKdyit99waaE4s=*
PublicExponent:	***AQAB***
PrivateExponent:	*rFS1lPbJllFFgFc33B5DDlC1egO8e81P4fFadODbp56V7sphKa6AZQCx8NYAew6VXFF PAKTw41QdHnK5klYOwxvfFDjDcUGza88qbjyrDPSJenkeZblSMUSSqy7AMFzEolkk6 WSn6k3thUVRgSlqDoOV3SElAsrB043XzGrKlVE=*
Prime1:	*8mbtsu9Tl9v7tKSHdCleprLlQXQLzxlSZun5T1n/OjvXSUtvD7x nZJ+LHqaBj1dlgMbCq 2U8O04QVcK3TS9GiQ==*
Prime2:	*3a6gkfs74d0Jb7yL4j4adAif4fcp7ZrGt7G5NRVDDY/Mv4TERAK Ma0TKN3okKE0A7X+ Rv2K84mhT4QLDlllEcw==*
Exponent1:	*v3D5A9uuCn5rgVR7wgV8ba0/KSpsdSiLgsoA42GxiB1gvvs7gJM MmVTDu/ZG1p1Zn pLbhh/S/Qd/MSwyNlxC+Q==*
Exponent2:	*m+ezf9dsDvYQK+gzjOLWYeKq5xWYBEYFGa3BLocMiF4oxkzOZ3J PZSWU/h1Fjp5R V7aPP0Vmx+hNjYMPlQ8Y5w==*
Coefficient:	*Je5YhYpUron/WdOXjxNAxDubAp3i5X7UOUfhJcylggqwY86lE0Q /Bk0Dw4SC9zxnsi mmdBXW2lzd8Lwuk8FQcQ==*

6-2.RSA/SHA-512 Key and Signature

- DNSKEYレコード

ZSK　RSA/SHA-512

```
example.net.    3600  IN  DNSKEY  ( 256  3  10  AwEAAdHoNTOW+et86KuJOWRD
                p1pndvwb6Y83nSVXXyLA3DLroROUkN6X0O6pnWnjJQujX/AyhqFD
                xj13tOnD9u/1kTg7cV6rklMrZDtJCQ5PCl/D7QNPsgVsMu1J2Q8g
                pMpztNFLpPBz1bWXjDtaR7ZQBlZ3PFY12ZTSncorffcGmhOL )
{id = 3740 (zsk), size = 1024b }
```

- RRSIGレコード

RSA/SHA-512

```
www.example.net. 3600  IN  A  192.0.2.91
www.example.net. 3600  IN  RRSIG  (A 10 3 3600 20300101000000 20000101000000 3740
                example.net. tsb4wnjRUDnB1BUi+t6TMTXThjVnG+eCkWqjvvjhzQL1
                d0YRoOe0CbxrVDYd0xDtsuJRaeUw1ep94PzEWzr0iGYgZBWm/zpq+
                9fOuagYJRfDqfReKBzMweOLDiNa8iP5g9vMhpuv6OPlvpXwm9Sa9Z
                XlbNl1MBGk0fthPgxdDLw =)
```

7.IANA Considerations

- このドキュメントはIANAの「DNS SECURITY ALGORITHM NUMBERS（RFC4035）」をアップデートする。
- 以下のエントリーが追加される。

Value	*Description*	*Mnemonic*	*Zone Signing*	*Trans.Sec.*	*Reference*
8	*RSA/SHA-256*	*RSASHA256*	*Y*	*	*RFC 5702*
10	*RSA/SHA-512*	*RSASHA512*	*Y*	*	*RFC 5702*

*** Transaction Security について、このアルゴリズム利用の標準化では定められていない。**

8.Security considerations

8-1.SHA-1 versus SHA-2 Considerations for RRSIG RR

- DNSSECのユーザは、ソフトウェアが対応したら、即座にSHA-2を利用することを推奨する。
- 一般的にSHA-2はSHA-1より攻撃に耐性がある。
- SHA-1は最近の攻撃手法により、信頼性が下がっている。
- SHA-2は十分な強度を持っていると考えられているが、今後の暗号化技術の開発予測については、このドキュメントの範囲外とする。

8-2.Signature Type Downgrade Attacks

- 各RRSetは公開されている各アルゴリズムで署名されなければならないため、両者が存在している場合、悪意ある集団はRSA/SHA-2のRRSIGを除いて、validatorにRSA/SHA-1の署名を強制的に使用させることができない。

- validatorがRSA/SHA-2をサポートする場合は、Downgrade Attackへの耐性が提供されるだろう。