Use of GOST signature algorithms in DNSKEY and RRSIG Resource Records for DNSSEC

draft-ietf-dnsext-dnssec-gost-02

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 10 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info).

Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes how to produce GOST signature and hash algorithms DNSKEY and RRSIG resource records for use in the Domain Name System Security Extensions (DNSSEC, RFC 4033, RFC 4034, and RFC 4035).
Table of Contents

1. Introduction ............................................. 2
2. DNSKEY Resource Records ............................... 3
   2.1. Using a public key with existing cryptographic libraries... 3
   2.2. GOST DNSKEY RR Example ............................ 3
3. RRSIG Resource Records ................................ 4
   3.1 RRSIG RR Example .................................. 4
4. DS Resource Records ..................................... 4
   4.1 DS RR Example ....................................... 5
5. Deployment Considerations ............................... 5
   5.1. Key Sizes ......................................... 5
   5.2. Signature Sizes ................................... 5
   5.3. Digest Sizes ....................................... 5
6. Implementation Considerations .......................... 5
   6.1. Support for GOST signatures ......................... 5
   6.2. Support for NSEC3 Denial of Existence ............... 5
   6.3. Byte order ....................................... 5
7. Security consideration .................................... 5
8. IANA Considerations ..................................... 6
9. Acknowledgments ......................................... 6
10. References .............................................. 6
    10.1. Normative References ............................... 6
    10.2. Informative References ............................. 7
Authors’ Addresses ........................................... 8

1. Introduction

The Domain Name System (DNS) is the global hierarchical distributed database for Internet Naming. The DNS has been extended to use cryptographic keys and digital signatures for the verification of the authenticity and integrity of its data. RFC 4033 [RFC4033], RFC 4034 [RFC4034], and RFC 4035 [RFC4035] describe these DNS Security Extensions, called DNSSEC.

RFC 4034 describes how to store DNSKEY and RRSIG resource records, and specifies a list of cryptographic algorithms to use. This document extends that list with the signature and hash algorithms GOST [GOST3410, GOST3411], and specifies how to store DNSKEY data and how to produce RRSIG resource records with these hash algorithms.

Familiarity with DNSSEC and GOST signature and hash algorithms is assumed in this document.

The term "GOST" is not officially defined, but is usually used to refer to the collection of the Russian cryptographic algorithms GOST R 34.10–2001, GOST R 34.11–94, GOST 28147–89. Since GOST 28147–89 is not used in DNSSEC, "GOST" will only refer to the GOST R 34.10–2001 and GOST R 34.11–94 in this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
2. DNSKEY Resource Records

The format of the DNSKEY RR can be found in RFC 4034 [RFC4034].

GOST R 34.10-2001 public keys are stored with the algorithm number {TBA1}.

The wire format of the public key is compatible with RFC 4491 [RFC4491]:

According to [GOSTR341001], a public key is a point on the elliptic curve \( Q = (x,y) \).

The wire representation of a public key MUST contain 66 octets, where the first octet designates public key parameters, the second octet designates digest parameters next 32 octets contain the little-endian representation of \( x \) and the second 32 octets contain the little-endian representation of \( y \).

This corresponds to the binary representation of \(<y>256|<x>256\) from [GOSTR341001], ch. 5.3.

The only valid value for both parameters octets is 0. Other parameters octets values are reserved for future use.

Corresponding public key parameters are those identified by id-GostR3410-2001-CryptoPro-A-ParamSet (1.2.643.2.2.35.1) [RFC4357], and the digest parameters are those identified by id-GostR3411-94-CryptoProParamSet (1.2.643.2.2.30.1) [RFC4357].

2.1. Using a public key with existing cryptographic libraries

Existing GOST-aware cryptographic libraries at the time of this document writing are capable to read GOST public keys via a generic X509 API if the key is encoded according to RFC 4491 [RFC4491], section 2.3.2.

To make this encoding from the wire format of a GOST public key with the parameters used in this document, prepend the last 64 octets of key data (in other words, substitute first two parameter octets) with the following 37-byte sequence:

```
0x30 0x63 0x30 0x1c 0x06 0x06 0x03 0x02 0x02 0x13 0x30 0x12 0x06 0x07 0x2a 0x85 0x03 0x02 0x02 0x23 0x01 0x06 0x07 0x2a 0x85 0x03 0x02 0x02 0x1e 0x01 0x03 0x43 0x00 0x04 0x40
```

2.2. GOST DNSKEY RR Example

Given a private key with the following value (the value of GostAsn1 field is split here into two lines to simplify reading; in the private key file it must be in one line):

```
Private-key-format: v1.2
Algorithm: {TBA1} (GOST)
GostAsn1: MEUCAQAwHAYGk0UDAgITMBIGByqFAwICiWEGByqFAwICgEERIgQV/S 2FxdkM3KJBeH2vJF41VSSx666TwqS/MFwKSH/3E=
```

V.Dolmatov Expires May 10, 2010
The following DNSKEY RR stores a DNS zone key for example.net

```
example.net. 86400 IN DNSKEY 256 3 {TBA1} (AADMrb12vAs4hklTmmzGE3JWhtJ8D110u0jq
tGRbNJkeJguZQjy9EpGwmmQRh9hekPlLzh2Ph6
yB7i836EfzmJ05LP) ; key id = 15820
```

3. RRSIG Resource Records

The value of the signature field in the RRSIG RR follows RFC 4490 [RFC4490] and is calculated as follows. The values for the RDATA fields that precede the signature data are specified in RFC 4034 [RFC4034].

```plaintext
hash = GOSTR3411(data)
```

where "data" is the wire format data of the resource record set that is signed, as specified in RFC 4034 [RFC4034].

Hash MUST be calculated with GOST R 34.11-94 parameters identified by id-GostR3411-94-CryptoProParamSet [RFC4357].

Signature is calculated from the hash according to the GOST R 34.10-2001 standard and its wire format is compatible with RFC 4490 [RFC4490].

Quoting RFC 4490:

"The signature algorithm GOST R 34.10-2001 generates a digital signature in the form of two 256-bit numbers, r and s. Its octet string representation consists of 64 octets, where the first 32 octets contain the big-endian representation of s and the second 32 octets contain the big-endian representation of r."

3.1. RRSIG RR Example

With the private key from section 2.2 sign the following RRSet, consisting of one A record:

```
www.example.net. 3600 IN A 192.0.32.10
```

Setting the inception date to 2000-01-01 00:00:00 UTC and the expiration date to 2030-01-01 00:00:00 UTC, the following signature should be created (assuming {TBA1}==249 until proper code is assigned by IANA)

```
www.example.net. 3600 IN RRSIG A {TBA1} 3 3600 20300101000000 (
20000101000000 15820 example.net.
K4sw+T0Jz47xqF6685ItDfkhktvgyXrLdX
aqLX01mMzbJUp6tzetBYGpdHciAW5RLvHLvB
P8RtFK8Qv5DrsA== )
```

Note: Several GOST signatures calculated for the same message text will differ because of using of random element in signature generation process.

4. DS Resource Records

GOST R 34.11-94 digest algorithm is denoted in DS RRs by the digest
type {TBA2}. The wire format of a digest value is compatible with RFC 4490 [RFC4490].
Quoting RFC 4490:

"A 32-byte digest in little-endian representation."

The digest MUST always be calculated with GOST R 34.11-94 parameters identified by id-GostR3411-94-CryptoProParamSet [RFC4357].

4.1. DS RR Example

For key signing key (assuming {TBA1}==249 until proper code is assigned by IANA)

```
example.net. 86400 DNSKEY 257 3 {TBA1} (AAADr5vmKVdXo780hSRU1YZYWuMZUbEe9R7C RRLc7Wj2osDXv2XbCnIpTUx8dVLnLkmDBquu 9tCz5oSsZ10cL0R2 ) ; key id = 21649
```

DS RR will be

```
example.net. 3600 IN DS 21649 {TBA1} {TBA2} (A8146F448569F30B91255BA8E98DE14B18569A524C49593ADCA4103A A44649C6 )
```

5. Deployment Considerations

5.1. Key Sizes

According to RFC4357 [RFC4357], the key size of GOST public keys MUST be 512 bits.

5.2. Signature Sizes

According to the GOST signature algorithm specification [GOST3410], the size of a GOST signature is 512 bits.

5.3. Digest Sizes

According to the GOST R 34.11-94 [GOST3411], the size of a GOST digest is 256 bits.

6. Implementation Considerations

6.1. Support for GOST signatures

DNSSEC aware implementations SHOULD be able to support RRSIG and DNSKEY resource records created with the GOST algorithms as defined in this document.

6.2. Support for NSEC3 Denial of Existence

Any DNSSEC-GOST implementation is required to have either NSEC or NSEC3 support.

6.3 Byte order

Due to the fact that all existing industry implementations of GOST cryptographic libraries are returning GOST blobs in little-endian format and in order to avoid the necessity for DNSSEC developers
to handle different cryptographic algorithms differently, it was chosen to send these blobs on the wire "as is" without transformation of endianness.

7. Security considerations

Currently, the cryptographic resistance of the GOST 34.10-2001 digital signature algorithm is estimated as $2^{128}$ operations of multiple elliptic curve point computations on prime modulus $2^{256}$. 

V.Dolmatov  Expires May 10, 2010  [Page 5]
Currently, the cryptographic resistance of GOST 34.11-94 hash algorithm is estimated as $2^{128}$ operations of computations of a step hash function. (There is known method to reduce this estimate to $2^{105}$ operations, but it demands padding the colliding message with 1024 random bit blocks each of 256 bit length, thus it cannot be used in any practical implementation).

8. IANA Considerations

This document updates the IANA registry "DNS SECURITY ALGORITHM NUMBERS -- per [RFC4035] " (http://www.iana.org/assignments/dns-sec-alg-numbers). The following entries are added to the registry:

<table>
<thead>
<tr>
<th>Zone</th>
<th>Trans.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{TBA1}</td>
<td>GOST R 34.10-2001 GOST</td>
</tr>
</tbody>
</table>

This document updates the RFC 4034 [RFC4034] Digest Types assignment (RFC 4034, section A.2):

<table>
<thead>
<tr>
<th>Value</th>
<th>Algorithm</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>{TBA2}</td>
<td>GOST R 34.11-94</td>
<td>OPTIONAL</td>
</tr>
</tbody>
</table>

9. Acknowledgments

This document is a minor extension to RFC 4034 [RFC4034]. Also, we tried to follow the documents RFC 3110 [RFC3110], RFC 4509 [RFC4509], and RFC 4357 [RFC4357] for consistency. The authors of and contributors to these documents are gratefully acknowledged for their hard work.

The following people provided additional feedback and text: Dmitry Burkov, Jaap Akkerhuis, Olafur Gundmundsson, Jelte Jansen and Wouter Wijngaards.

10. References

10.1. Normative References


10.2. Informative References


V.Dolmatov          Expires May 10, 2010
Authors’ Addresses

Vasily Dolmatov, Ed.
Cryptocom Ltd.
Bolotnikovskaya, 23
Moscow, 117303, Russian Federation
EMail: dol@cryptocom.ru

Artem Chuprina
Cryptocom Ltd.
Bolotnikovskaya, 23
Moscow, 117303, Russian Federation
EMail: ran@cryptocom.ru

Igor Ustinov
Cryptocom Ltd.
Bolotnikovskaya, 23
Moscow, 117303, Russian Federation
EMail: igus@cryptocom.ru