INTERNET-DRAFT
OBSOLETES: RFC 2539

Expires: April 2007

Diffie-Hellman Information in the DNS
Donald E. Eastlake 3rd
Motorola Laboratories
October 2006

Storage of Diffie-Hellman Keying Information in the DNS

<draft-ietf-dnsext-rfc2539bis-dhk-08.txt>

Status of This Document

Abstract

The standard method for encoding Diffie-Hellman keys in the Domain Name System is specified.

Acknowledgements

Part of the format for Diffie-Hellman keys and the description thereof was taken from a work in progress by Ashar Aziz, Tom Markson, and Hemma Prafullchandra. In addition, the following persons provided useful comments that were incorporated into the predecessor of this document: Ran Atkinson, Thomas Narten.

Table of Contents

1. Introduction

The Domain Name System (DNS) is the global hierarchical replicated distributed database system for Internet addressing, mail proxy, and other information [RFC1034], [RFC1035]. The DNS has been extended to include digital signatures and cryptographic keys as described in [RFC4033], [RFC4034, [RFC4035] and there is additional work which would use keying information in the DNS such as TKEY [RFC2930] and GSS-TSIG [RFC3645]. This document does not change the wire format of KEY RR's but extends the use of Diffie-Hellman DNS keys to cover the DNSKEY RR.

1.1 About This Document

This document describes how to store Diffie-Hellman keys in the DNS. Familiarity with the Diffie-Hellman key exchange algorithm is assumed [Schneier], [RFC2631].

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].

1.2 About Diffie-Hellman

Diffie-Hellman requires two parties to interact to derive keying information which can then be used for authentication. Thus Diffie-Hellman is inherently a key agreement algorithm. As a result, no format is defined for Diffie-Hellman "signature information". For example, assume that two parties have local secrets "i" and "j". Assume they each respectively calculate X and Y as follows:

```
X = g**i ( mod p )

Y = g**j ( mod p )
```

They exchange these quantities and then each calculates a Z as follows:

```
Zi = Y**i ( mod p )

Zj = X**j ( mod p )
```

Zi and Zj will both be equal to g**(i*j)(mod p) and will be a shared secret between the two parties that an adversary who does not know i or j will not be able to learn from the exchanged messages (unless

the adversary can derive i or j by performing a discrete logarithm

mod p which is hard for strong p and g).

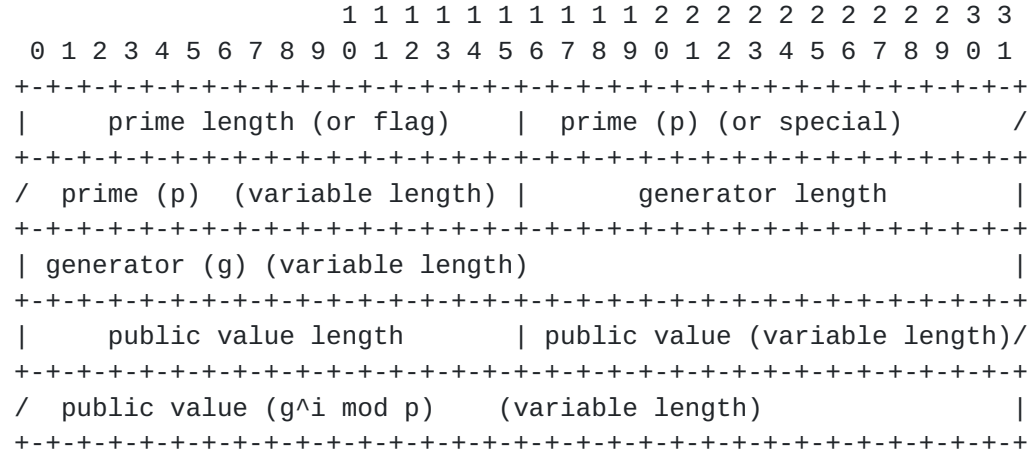
The private key for each party is their secret i (or j). The public key is the pair p and g, which is the same for both parties, and their individual X (or Y).

For further information about Diffie-Hellman and precautions to take in deciding on a p and g, see [RFC2631].

2. Encoding Diffie-Hellman Keying Information

When Diffie-Hellman keys appear within the RDATA portion of a RR, they are encoded as shown below.

The period of key validity is not included in this data but is indicated separately, for example by an RR such as RRSIG which signs and authenticates the RR containing the keying information.



Prime length is the length of the Diffie-Hellman prime (p) in bytes if it is 16 or greater. Prime contains the binary representation of the Diffie-Hellman prime with most significant byte first (i.e., in network order). If "prime length" field is 1 or 2, then the "prime" field is actually an unsigned index into a table of 65,536 prime/generator pairs and the generator length SHOULD be zero. See Appendix A for defined table entries and Section 4 for information on allocating additional table entries. The meaning of a zero or 3 through 15 value for "prime length" is reserved.

Generator length is the length of the generator (g) in bytes. Generator is the binary representation of generator with most significant byte first. PublicValueLen is the Length of the Public Value (g**i (mod p)) in bytes. PublicValue is the binary representation of the DH public value with most significant byte

first.

3. Performance Considerations

Current DNS implementations are optimized for small transfers, typically less than 512 bytes including DNS overhead. Larger transfers will perform correctly and extensions have been standardized [RFC2671] to make larger transfers more efficient. But it is still advisable at this time to make reasonable efforts to minimize the size of RR sets containing keying information consistent with adequate security.

4. IANA Considerations

Assignment of meaning to Prime Lengths of 0 and 3 through 15 requires an IETF consensus as defined in [RFC2434].

Well known prime/generator pairs number 0x0000 through 0x07FF can only be assigned by an IETF Standards Action. [RFC2539], the Proposed Standard predecessor of this document, assigned 0x0001 through 0x0002. This document additionally assigns 0x0003 through 0x0008. Pairs number 0s0800 through 0xBFFF can be assigned based on Specification Required as specified in [RFC2434]. Pairs number 0xC000 through 0xFFFF are available for private use and are not centrally coordinated. Use of such private pairs outside of a closed environment may result in conflicts and/or security failures.

5. Security Considerations

Keying information retrieved from the DNS should not be trusted unless (1) it has been securely obtained from a secure resolver or independently verified by the user and (2) this secure resolver and secure obtainment or independent verification conform to security policies acceptable to the user. As with all cryptographic algorithms, evaluating the necessary strength of the key is important and dependent on security policy.

In addition, the usual Diffie-Hellman key strength considerations apply. (p-1)/2 SHOULD also be prime, g SHOULD be primitive mod p, p SHOULD be "large", etc. See [RFC2631], [Schneier].

Normative References

[RFC2119] - Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2434] - "Guidelines for Writing an IANA Considerations Section in RFCs", T. Narten, H. Alvestrand, October 1998.

[RFC2631] - "Diffie-Hellman Key Agreement Method", E. Rescorla, June 1999.

[RFC3526] - Kivinen, T., and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", May 2003.

[RFC4034] - Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

Informative References

[RFC1034] - "Domain names - concepts and facilities", P. Mockapetris, November 1987.

[RFC1035] - "Domain names - implementation and specification", P. Mockapetris, November 1987.

[RFC2930] - Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", September 2000.

[RFC2539] - "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", D. Eastlake, March 1999, obsoleted by this RFC.

[RFC2671] - "Extension Mechanisms for DNS (EDNS0)", P. Vixie, August 1999.

[RFC3645] - Kwan, S., et al "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)", October 2000.

[RFC3755] - Weiler, S., "Legacy Resolver Compatibility for Delegation Signer (DS)", May 2004.

[RFC4033] - Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4035] - Arends, R., Austein, R., Larson, M., Massey, D., and S.

Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[Schneier] - Bruce Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C" (Second Edition), 1996, John Wiley and Sons.

Appendix A: Well known prime/generator pairs

These numbers are copied from the IPSEC effort where the derivation of these values is more fully explained and additional information is available. Richard Schroeppel performed all the mathematical and computational work for this appendix.

A.1. Well-Known Group 1: A 768 bit prime

The prime is 2^768 - 2^704 - 1 + 2^64 * { [2^638 pi] + 149686 }. Its decimal value is

```
155251809230070893513091813125848175563133404943451431320235
119490296623994910210725866945387659164244291000768028886422
915080371891804634263272761303128298374438082089019628850917
069131659317536746955176311984337163722100721057791 9
```

Prime modulus: Length (32 bit words): 24, Data (hex):

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF
```

Generator: Length (32 bit words): 1, Data (hex): 2

A.2. Well-Known Group 2: A 1024 bit prime

The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }. Its decimal value is

```
179769313486231590770839156793787453197860296048756011706444
423684197180216158519368947833795864925541502180565485980503
646440548199239100050792877003355816639229553136239076508735
759914822574862575007425302077447712589550957937778424442426
617334727629299387668709205606050270810842907692932019128194
467627007
```

Prime modulus: Length (32 bit words): 32, Data (hex):

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF
```

Generator: Length (32 bit words): 1, Data (hex): 2

A.3. Well-Known Group 3: A 1536 bit prime

The prime is 2^1536 - 2^1472 - 1 + 2^64 * { [2^1406 pi] + 741804 }.
Its decimal value is

```
241031242692103258855207602219756607485695054850245994265411
694195810883168261222889009385826134161467322714147790401219
650364895705058263194273070680500922306273474534107340669624
601458936165977404102716924945320037872943417032584377865919
814376319377685986952408894019557734611984354530154704374720
774996976375008430892633929555996888245787241299381012913029
459299994792636526405928464720973038494721168143446471443848
852094012745984428885933652689632091963391 9
```

Prime modulus Length (32 bit words): 48, Data (hex):

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFF FFFFFFFF
```

Generator: Length (32 bit words): 1, Data (hex): 2

A.4 Well known Groups 4 through 8

The additional Diffie-Hellman Groups specified in [RFC3526] are also adopted and assigned well known group numbers as follows:

Group Number	Size
4	2048-bit
5	3072-bit
6	4096-bit
7	6144-bit
8	8192-bit

Appendix B: Changes from RFC 2539

When [RFC2539] was published, keys in the DNS appeared only in KEY resource records. As described in [RFC3755], due to a revision in DNS data origin authentication security, the recommended RR was changed to DNSKEY which is described in [RFC4034]; however KEY continues to be used in connection with TKEY [RFC2930].

Thus the primary change from [RFC2539] in this document is to

eliminate the tie to the KEY RRs. In addition, more well known Diffie-Hellman Groups are listed and assigned identification numbers and many references have been updated.

Author's Address

Donald E. Eastlake 3rd
Motorola Laboratories
111 Locke Drive
Marlborough, MA 01752 USA

Telephone: +1-508-786-7554
EMail: Donald.Eastlake@motorola.com

Expiration and File Name

This draft expires in April 2007.

Its file name is draft-ietf-dnsext-rfc2539bis-dhk-08.txt.