

Network Working Group  
Request for Comments: 2539  
Category: Standards Track

D. Eastlake  
IBM  
March 1999

## Storage of Diffie-Hellman Keys in the Domain Name System (DNS)

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

### Abstract

A standard method for storing Diffie-Hellman keys in the Domain Name System is described which utilizes DNS KEY resource records.

### Acknowledgements

Part of the format for Diffie-Hellman keys and the description thereof was taken from a work in progress by:

Ashar Aziz <ashar.aziz@eng.sun.com>  
Tom Markson <markson@incog.com>  
Hemma Prafullchandra <hemma@eng.sun.com>

In addition, the following person provided useful comments that have been incorporated:

Ran Atkinson <rja@inet.org>  
Thomas Narten <narten@raleigh.ibm.com>

## Table of Contents

Abstract.....	1
Acknowledgements.....	1
1. Introduction.....	2
1.1 About This Document.....	2
1.2 About Diffie-Hellman.....	2
2. Diffie-Hellman KEY Resource Records.....	3
3. Performance Considerations.....	4
4. IANA Considerations.....	4
5. Security Considerations.....	4
References.....	5
Author's Address.....	5
Appendix A: Well known prime/generator pairs.....	6
A.1. Well-Known Group 1: A 768 bit prime.....	6
A.2. Well-Known Group 2: A 1024 bit prime.....	6
Full Copyright Notice.....	7

## 1. Introduction

The Domain Name System (DNS) is the current global hierarchical replicated distributed database system for Internet addressing, mail proxy, and similar information. The DNS has been extended to include digital signatures and cryptographic keys as described in [RFC 2535]. Thus the DNS can now be used for secure key distribution.

## 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].

## 1.2 About Diffie-Hellman

Diffie-Hellman requires two parties to interact to derive keying information which can then be used for authentication. Since DNS SIG RRs are primarily used as stored authenticators of zone information for many different resolvers, no Diffie-Hellman algorithm SIG RR is defined. For example, assume that two parties have local secrets "i" and "j". Assume they each respectively calculate X and Y as follows:

$$X = g^{**i} \text{ ( mod } p \text{ ) } \quad Y = g^{**j} \text{ ( mod } p \text{ ) }$$

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

$$Z_i = Y^{**i} \text{ ( mod } p \text{ ) } \quad Z_j = X^{**j} \text{ ( mod } p \text{ ) }$$

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

Diffie-Hellman keys are stored in the DNS as KEY RRs using algorithm number 2. The structure of the RDATA portion of this RR is as shown below. The first 4 octets, including the flags, protocol, and algorithm fields are common to all KEY RRs as described in [RFC 2535]. The remainder, from prime length through public value is the "public key" part of the KEY RR. The period of key validity is not in the KEY RR but is indicated by the SIG RR(s) which signs and authenticates the KEY RR(s) at that domain name.

[illegible]

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 ( $g^i \pmod p$ ) in bytes. PublicValue is the binary representation of the DH public value with most significant byte first.

The corresponding algorithm=2 SIG resource record is not used so no format for it is defined.

### 3. Performance Considerations

Current DNS implementations are optimized for small transfers, typically less than 512 bytes including overhead. While larger transfers will perform correctly and work is underway to make larger transfers more efficient, it is still advisable to make reasonable efforts to minimize the size of KEY RR sets stored within the DNS consistent with adequate security. Keep in mind that in a secure zone, an authenticating SIG RR will also be returned.

### 4. IANA Considerations

Assignment of meaning to Prime Lengths of 0 and 3 through 15 requires an IETF consensus.

Well known prime/generator pairs number 0x0000 through 0x07FF can only be assigned by an IETF standards action and this Proposed Standard assigns 0x0001 through 0x0002. Pairs number 0x0800 through 0xBFFF can be assigned based on RFC documentation. 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.

### 5. Security Considerations

Many of the general security consideration in [RFC 2535] apply. Keys retrieved from the DNS should not be trusted unless (1) they have 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 local 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. [Schneier]

## References

- [RFC 1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.
- [RFC 1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987.
- [RFC 2535] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.
- [Schneier] Bruce Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", 1996, John Wiley and Sons

## Author's Address

Donald E. Eastlake 3rd  
IBM  
65 Shindegan Hill Road, RR #1  
Carmel, NY 10512

Phone: +1-914-276-2668(h)  
+1-914-784-7913(w)  
Fax: +1-914-784-3833(w)  
EMail: dee3@us.ibm.com

## 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 Schroepel performed all the mathematical and computational work for this appendix.

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

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

```
155251809230070893513091813125848175563133404943451431320235
119490296623994910210725866945387659164244291000768028886422
915080371891804634263272761303128298374438082089019628850917
0691316593175367469551763119843371637221007210577919
```

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  $2^{1024} - 2^{960} - 1 + 2^{64} * \{ [2^{894} \text{ pi}] + 129093 \}$ . 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

## Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

