Elliptic Curve KEYs in the DNS

Richard C. Schroeppel
Donald Eastlake 3rd

Status of This Document

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

This draft is intended to be become a Proposed Standard RFC. Distribution of this document is unlimited. Comments should be sent to the DNS mailing list <namedroppers@ops.ietf.org>.

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 a "work in progress."

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

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

Abstract

The standard method for storing elliptic curve cryptographic keys and signatures in the Domain Name System is specified.

Copyright Notice

Acknowledgement

The assistance of Hilarie K. Orman in the production of this document is greatfully acknowledged.

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. The DNS has been extended to include digital signatures and cryptographic keys as described in [RFC 4033, 4034, 4035].

This document describes how to store elliptic curve cryptographic (ECC) keys and signatures in the DNS so they can be used for a variety of security purposes. Familiarity with ECC cryptography is assumed [Menezes].

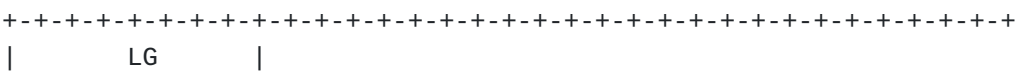
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

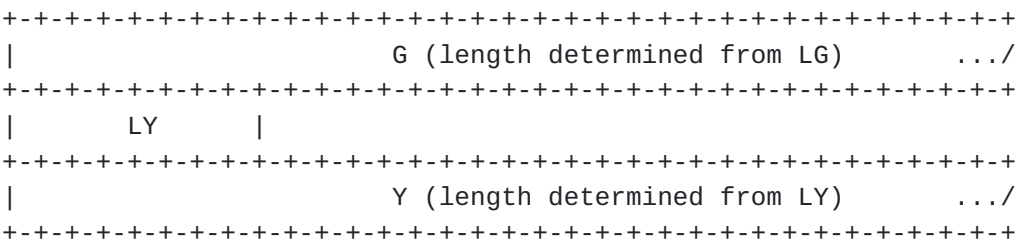
2. Elliptic Curve Data in Resource Records

Elliptic curve public keys are stored in the DNS within the RDATA portions of key RRs, such as RRKEY and KEY [RFC 4034] RRs, with the structure shown below.

The research world continues to work on the issue of which is the best elliptic curve system, which finite field to use, and how to best represent elements in the field. So, representations are defined for every type of finite field, and every type of elliptic curve. The reader should be aware that there is a unique finite field with a particular number of elements, but many possible representations of that field and its elements. If two different representations of a field are given, they are interconvertible with a tedious but practical precomputation, followed by a fast computation for each field element to be converted. It is perfectly reasonable for an algorithm to work internally with one field representation, and convert to and from a different external representation.


1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
S M -FMT- A B Z
LP
P (length determined from LP) .../
LF
F (length determined from LF) .../
DEG
DEGH
DEGI
DEGJ
TRDV
S
LH
H (length determined from LH) .../
S
LK
K (length determined from LK) .../
LQ
Q (length determined from LQ) .../
LA
A (length determined from LA) .../
ALTA
LB
B (length determined from LB) .../
LC
C (length determined from LC) .../


LG



SMFMTABZ is a flags octet as follows:

S = 1 indicates that the remaining 7 bits of the octet selects one of 128 predefined choices of finite field, element representation, elliptic curve, and signature parameters. MFMTABZ are omitted, as are all parameters from LP through G. LY and Y are retained.

If S = 0, the remaining parameters are as in the picture and described below.

M determines the type of field underlying the elliptic curve.

M = 0 if the field is a GF[2^N] field;

M = 1 if the field is a (mod P) or GF[P^D] field with P>2.

FMT is a three bit field describing the format of the field representation.

FMT = 0 for a (mod P) field.
> 0 for an extension field, either GF[2^D] or GF[P^D]. The degree D of the extension, and the field polynomial must be specified. The field polynomial is always monic (leading coefficient 1.)

FMT = 1 The field polynomial is given explicitly; D is implied.

If FMT >=2, the degree D is given explicitly.

= 2 The field polynomial is implicit.
= 3 The field polynomial is a binomial. P>2.
= 4 The field polynomial is a trinomial.
= 5 The field polynomial is the quotient of a trinomial by a short polynomial. P=2.
= 6 The field polynomial is a pentanomial. P=2.

Flags A and B apply to the elliptic curve parameters.

A = 1 When P>=5, the curve parameter A is negated. If P=2, then A=1 indicates that the A parameter is special. See the ALTA parameter below, following A. The combination A=1, P=3 is forbidden.

B = 1 When P>=5, the curve parameter B is negated. If P=2 or 3, then B=1 indicates an alternate elliptic curve equation is used. When P=2 and B=1, an additional curve parameter C is present.

The Z bit SHOULD be set to zero on creation of an RR and MUST be ignored when processing an RR (when S=0).

Most of the remaining parameters are present in some formats and absent in others. The presence or absence of a parameter is determined entirely by the flags. When a parameter occurs, it is in the order defined by the picture.

Of the remaining parameters, PFHKQABCGY are variable length. When present, each is preceded by a one-octet length field as shown in the diagram above. The length field does not include itself. The length field may have values from 0 through 110. The parameter length in octets is determined by a conditional formula: If LL<=64, the parameter length is LL. If LL>64, the parameter length is 16 times (LL-60). In some cases, a parameter value of 0 is sensible, and MAY be represented by an LL value of 0, with the data field omitted. A length value of 0 represents a parameter value of 0, not an absent parameter. (The data portion occupies 0 space.) There is no requirement that a parameter be represented in the minimum number of octets; high-order 0 octets are allowed at the front end. Parameters are always right adjusted, in a field of length defined by LL. The octet-order is always most-significant first, least-significant last. The parameters H and K may have an optional sign bit stored in the unused high-order bit of their length fields.

LP defines the length of the prime P. P must be an odd prime. The parameters LP,P are present if and only if the flag M=1. If M=0, the prime is 2.

LF,F define an explicit field polynomial. This parameter pair is present only when FMT = 1. The length of a polynomial coefficient is ceiling(log2 P) bits. Coefficients are in the numerical range [0,P-1]. The coefficients are packed into fixed-width fields, from higher order to lower order. All coefficients must be present, including any 0s and also the leading coefficient (which is required to be 1). The coefficients are right justified into the octet string of length specified by LF, with the low-order "constant" coefficient at the right end. As a concession to storage efficiency, the higher

order bits of the leading coefficient may be elided, discarding high-order 0 octets and reducing LF. The degree is calculated by

determining the bit position of the left most 1-bit in the F data (counting the right most bit as position 0), and dividing by ceiling(log2 P). The division must be exact, with no remainder. In this format, all of the other degree and field parameters are omitted. The next parameters will be LQ,Q.

If FMT>=2, the degree of the field extension is specified explicitly, usually along with other parameters to define the field polynomial.

DEG is a two octet field that defines the degree of the field extension. The finite field will have P^DEG elements. DEG is present when FMT>=2.

When FMT=2, the field polynomial is specified implicitly. No other parameters are required to define the field; the next parameters present will be the LQ,Q pair. The implicit field poynomial is the lexicographically smallest irreducible (mod P) polynomial of the correct degree. The ordering of polynomials is by highest-degree coefficients first -- the leading coefficient 1 is most important, and the constant term is least important. Coefficients are ordered by sign-magnitude: 0 < 1 < -1 < 2 < -2 < ... The first polynomial of degree D is X^D (which is not irreducible). The next is X^D+1, which is sometimes irreducible, followed by X^D-1, which isn't. Assuming odd P, this series continues to X^D - (P-1)/2, and then goes to X^D + X, X^D + X + 1, X^D + X - 1, etc.

When FMT=3, the field polynomial is a binomial, X^DEG + K. P must be odd. The polynomial is determined by the degree and the low order term K. Of all the field parameters, only the LK,K parameters are present. The high-order bit of the LK octet stores on optional sign for K; if the sign bit is present, the field polynomial is X^DEG - K.

When FMT=4, the field polynomial is a trinomial, X^DEG + H*X^DEGH + K. When P=2, the H and K parameters are implicitly 1, and are omitted from the representation. Only DEG and DEGH are present; the next parameters are LQ,Q. When P>2, then LH,H and LK,K are specified. Either or both of LH, LK may contain a sign bit for its parameter.

When FMT=5, then P=2 (only). The field polynomial is the exact quotient of a trinomial divided by a small polynomial, the trinomial divisor. The small polynomial is right-adjusted in the two octet field TRDV. DEG specifies the degree of the field. The degree of TRDV is calculated from the position of the high-order 1 bit. The trinomial to be divided is X^(DEG+degree(TRDV)) + X^DEGH + 1. If DEGH is 0, the middle term is omitted from the trinomial. The quotient must be exact, with no remainder.

When FMT=6, then P=2 (only). The field polynomial is a pentanomial, with the degrees of the middle terms given by the three 2-octet

values DEGH, DEGI, DEGJ. The polynomial is X^DEG + X^DEGH + X^DEGI + X^DEGJ + 1. The values must satisfy the inequality DEG > DEGH > DEGI > DEGJ > 0.

DEGH, DEGI, DEGJ are two-octet fields that define the degree of a term in a field polynomial. DEGH is present when FMT = 4, 5, or 6. DEGI and DEGJ are present only when FMT = 6.

TRDV is a two-octet right-adjusted binary polynomial of degree < 16. It is present only for FMT=5.

LH and H define the H parameter, present only when FMT=4 and P is odd. The high bit of LH is an optional sign bit for H.

LK and K define the K parameter, present when FMT = 3 or 4, and P is odd. The high bit of LK is an optional sign bit for K.

The remaining parameters are concerned with the elliptic curve and the signature algorithm.

LQ defines the length of the prime Q. Q is a prime > 2^159.

In all 5 of the parameter pairs LA+A,LB+B,LC+C,LG+G,LY+Y, the data member of the pair is an element from the finite field defined earlier. The length field defines a long octet string. Field elements are represented as (mod P) polynomials of degree < DEG, with DEG or fewer coefficients. The coefficients are stored from left to right, higher degree to lower, with the constant term last. The coefficients are represented as integers in the range [0,P-1]. Each coefficient is allocated an area of ceiling(log2 P) bits. The field representation is right-justified; the "constant term" of the field element ends at the right most bit. The coefficients are fitted adjacently without regard for octet boundaries. (Example: if P=5, three bits are used for each coefficient. If the field is GF[5^75], then 225 bits are required for the coefficients, and as many as 29 octets may be needed in the data area. Fewer octets may be used if some high-order coefficients are 0.) If a flag requires a field element to be negated, each non-zero coefficient K is replaced with P-K. To save space, 0 bits may be removed from the left end of the element representation, and the length field reduced appropriately. This would normally only happen with A,B,C, because the designer chose curve parameters with some high-order 0 coefficients or bits.

If the finite field is simply (mod P), then the field elements are simply numbers (mod P), in the usual right-justified notation. If the finite field is GF[2^D], the field elements are the usual right-justified polynomial basis representation.

LA,A is the first parameter of the elliptic curve equation. When P>=5, the flag A = 1 indicates A should be negated (mod P). When P=2 (indicated by the flag M=0), the flag A = 1 indicates that the parameter pair LA,A is replaced by the two octet parameter ALTA. In this case, the parameter A in the curve equation is x^ALTA, where x is the field generator. Parameter A often has the value 0, which may be indicated by LA=0 (with no A data field), and sometimes A is 1, which may be represented with LA=1 and a data field of 1, or by setting the A flag and using an ALTA value of 0.

LB,B is the second parameter of the elliptic curve equation. When P>=5, the flag B = 1 indicates B should be negated (mod P). When P=2 or 3, the flag B selects an alternate curve equation.

LC,C is the third parameter of the elliptic curve equation, present only when P=2 (indicated by flag M=0) and flag B=1.

LG,G defines a point on the curve, of order Q. The W-coordinate of the curve point is given explicitly; the Z-coordinate is implicit.

LY,Y is the user's public signing key, another curve point of order Q. The W-coordinate is given explicitly; the Z-coordinate is implicit. The LY,Y parameter pair is always present.

3. The Elliptic Curve Equation

(The coordinates of an elliptic curve point are named W,Z instead of the more usual X,Y to avoid confusion with the Y parameter of the signing key.)

The elliptic curve equation is determined by the flag octet, together with information about the prime P. The primes 2 and 3 are special; all other primes are treated identically.

If M=1, the (mod P) or GF[P^D] case, the curve equation is Z^2 = W^3 + A*W + B. Z,W,A,B are all numbers (mod P) or elements of GF[P^D]. If A and/or B is negative (i.e., in the range from P/2 to P), and P>=5, space may be saved by putting the sign bit(s) in the A and B bits of the flags octet, and the magnitude(s) in the parameter fields.

If M=1 and P=3, the B flag has a different meaning: it specifies an alternate curve equation, Z^2 = W^3 + A*W^2 + B. The middle term of

the right-hand-side is different. When P=3, this equation is more

commonly used.

If M=0, the GF[2^N] case, the curve equation is Z^2 + W*Z = W^3 + A*W^2 + B. Z,W,A,B are all elements of the field GF[2^N]. The A parameter can often be 0 or 1, or be chosen as a single-1-bit value. The flag B is used to select an alternate curve equation, Z^2 + C*Z = W^3 + A*W + B. This is the only time that the C parameter is used.

4. How do I Compute Q, G, and Y?

The number of points on the curve is the number of solutions to the curve equation, + 1 (for the "point at infinity"). The prime Q must divide the number of points. Usually the curve is chosen first, then the number of points is determined with Schoof's algorithm. This number is factored, and if it has a large prime divisor, that number is taken as Q.

G must be a point of order Q on the curve, satisfying the equation

Q * G = the point at infinity (on the elliptic curve)

G may be chosen by selecting a random [RFC 1750] curve point, and multiplying it by (number-of-points-on-curve/Q). G must not itself be the "point at infinity"; in this astronomically unlikely event, a new random curve point is recalculated.

G is specified by giving its W-coordinate. The Z-coordinate is calculated from the curve equation. In general, there will be two possible Z values. The rule is to choose the "positive" value.

In the (mod P) case, the two possible Z values sum to P. The smaller value is less than P/2; it is used in subsequent calculations. In GF[P^D] fields, the highest-degree non-zero coefficient of the field element Z is used; it is chosen to be less than P/2.

In the GF[2^N] case, the two possible Z values xor to W (or to the parameter C with the alternate curve equation). The numerically smaller Z value (the one which does not contain the highest-order 1 bit of W (or C)) is used in subsequent calculations.

Y is specified by giving the W-coordinate of the user's public signature key. The Z-coordinate value is determined from the curve equation. As with G, there are two possible Z values; the same rule is followed for choosing which Z to use.

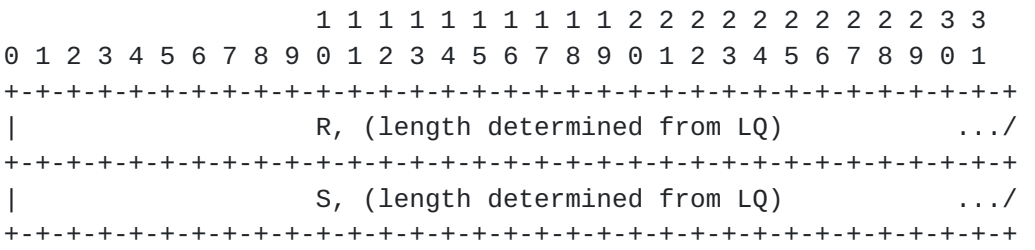
During the key generation process, a random [RFC 1750] number X must be generated such that 1 <= X <= Q-1. X is the private key and is used in the final step of public key generation where Y is computed as

Y = X * G (as points on the elliptic curve)

If the Z-coordinate of the computed point Y is wrong (i.e., Z > P/2 in the (mod P) case, or the high-order non-zero coefficient of Z > P/2 in the GF[P^D] case, or Z sharing a high bit with W(C) in the GF[2^N] case), then X must be replaced with Q-X. This will correspond to the correct Z-coordinate.

5. Elliptic Curve SIG Resource Records

The signature portion of an RR RDATA area when using the EC algorithm, for example in the RRSIG and SIG [RFC records] RRs is shown below.



R and S are integers (mod Q). Their length is specified by the LQ field of the corresponding KEY RR and can also be calculated from the SIG RR's RDLENGTH. They are right justified, high-order-octet first. The same conditional formula for calculating the length from LQ is used as for all the other length fields above.

The data signed is determined as specified in [RFC 2535]. Then the following steps are taken where Q, P, G, and Y are as specified in the public key [Schneier]:

hash = SHA-1 (data)

Generate random [RFC 4086] K such that 0 < K < Q. (Never sign two different messages with the same K. K should be chosen from a very large space: If an opponent learns a K value for a single signature, the user's signing key is compromised, and a forger can sign arbitrary messages. There is no harm in signing the same message multiple times with the same key or different keys.)

R = (the W-coordinate of (K*G on the elliptic curve)) interpreted

as an integer, and reduced (mod Q). (R must not be 0. In this astronomically unlikely event, generate a new random K and recalculate R.)

S = (K^(-1) * (hash + X*R)) mod Q.

S must not be 0. In this astronomically unlikely event, generate a new random K and recalculate R and S.

If S > Q/2, set S = Q - S.

The pair (R,S) is the signature.

Another party verifies the signature as follows:

Check that 0 < R < Q and 0 < S < Q/2. If not, it can not be a valid EC sigature.

hash = SHA-1 (data)

Sinv = S^(-1) mod Q.

U1 = (hash * Sinv) mod Q.

U2 = (R * Sinv) mod Q.

(U1 * G + U2 * Y) is computed on the elliptic curve.

V = (the W-coordinate of this point) interpreted as an integer and reduced (mod Q).

The signature is valid if V = R.

The reason for requiring S < Q/2 is that, otherwise, both (R,S) and (R,Q-S) would be valid signatures for the same data. Note that a signature that is valid for hash(data) is also valid for hash(data)+Q or hash(data)-Q, if these happen to fall in the range [0,2^160-1]. It's believed to be computationally infeasible to find data that hashes to an assigned value, so this is only a cosmetic blemish. The blemish can be eliminated by using Q > 2^160, at the cost of having slightly longer signatures, 42 octets instead of 40.

We must specify how a field-element E ("the W-coordinate") is to be interpreted as an integer. The field-element E is regarded as a radix-P integer, with the digits being the coefficients in the polynomial basis representation of E. The digits are in the ragne [0,P-1]. In the two most common cases, this reduces to "the obvious thing". In the (mod P) case, E is simply a residue mod P,

and is taken as an integer in the range [0,P-1]. In the GF[2^D]

case, E is in the D-bit polynomial basis representation, and is simply taken as an integer in the range [0,(2^D)-1]. For other fields GF[P^D], it's necessary to do some radix conversion arithmetic.

6. Performance Considerations

Elliptic curve signatures use smaller moduli or field sizes than RSA and DSA. Creation of a curve is slow, but not done very often. Key generation is faster than RSA or DSA.

DNS implementations have been optimized for small transfers, typically less than 512 octets including DNS overhead. Larger transfers will perform correctly and and extensions have been standardized to make larger transfers more efficient [RFC 2671]. However, it is still advisable at this time to make reasonable efforts to minimize the size of RR sets stored within the DNS consistent with adequate security.

7. Security Considerations

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 essential and dependent on local policy.

Some specific key generation considerations are given in the body of this document.

8. IANA Considerations

The key and signature data structures defined herein correspond to the value 4 in the Algorithm number field of the IANA registry

Assignment of meaning to the remaining ECC data flag bits or to values of ECC fields outside the ranges for which meaning in defined in this document requires an IETF consensus as defined in [RFC 2434].

Copyright and Disclaimer

Informational References

[RFC 1034] - P. Mockapetris, "Domain names - concepts and facilities", 11/01/1987.

[RFC 1035] - P. Mockapetris, "Domain names - implementation and specification", 11/01/1987.

[RFC 2671] - P. Vixie, "Extension Mechanisms for DNS (EDNS0)", August 1999.

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

[RFC 4035] - Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC 4086] - Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

[Menezes] - Alfred Menezes, "Elliptic Curve Public Key Cryptosystems", 1993 Kluwer.

[Silverman] - Joseph Silverman, "The Arithmetic of Elliptic Curves", 1986, Springer Graduate Texts in mathematics #106.

Normative Refrences

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

[RFC 2434] - T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", October 1998.

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

Author's Addresses

Rich Schroeppel
500 S. Maple Drive
Woodland Hills, UT 84653 USA

Telephone: +1-505-844-9079(w)
Email: rschroe@sandia.gov

Donald E. Eastlake 3rd
Motorola Laboratories
155 Beaver Street
Milford, MA 01757 USA

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

Expiration and File Name

This draft expires in January 2006.

Its file name is draft-ietf-dnsext-ecc-key-07.txt.