Abstract

The JSON Private Key specification extends the JSON Web Key (JWK) and JSON Web Algorithms (JWA) specifications to define a JavaScript Object Notation (JSON) representation of private keys.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on May 10, 2013.

Copyright Notice

Copyright (c) 2012 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Table of Contents

1. Introduction .................................................. 3
   1.1. Notational Conventions ................................... 3
2. Terminology .................................................. 3
3. JWK Parameters for Private Keys .............................. 3
   3.1. JWK Parameters for Elliptic Curve Private Keys .......... 3
   3.1.1. "d" (ECC Private Key) Parameter ........................ 3
   3.2. JWK Parameters for RSA Private Keys ..................... 3
   3.2.1. "d" (Private Exponent) Parameter ...................... 4
   3.2.2. "p" (First Prime Factor) Parameter .................... 4
   3.2.3. "q" (Second Prime Factor) Parameter ................... 4
   3.2.4. "dp" (First Factor CRT Exponent) Parameter .......... 4
   3.2.5. "dq" (Second Factor CRT Exponent) Parameter .......... 4
   3.2.6. "qi" (First CRT Coefficient) Parameter ............... 4
   3.2.7. "oth" (Other Primes Info) Parameter .................. 4
   3.2.7.1. "r" (Prime Factor) .................................. 5
   3.2.7.2. "d" (Factor CRT Exponent) .......................... 5
   3.2.7.3. "t" (Factor CRT Coefficient) ....................... 5
4. Example Private Keys .......................................... 5
5. IANA Considerations ........................................... 6
   5.1. JSON Web Key Parameters Registration .................... 7
   5.1.1. Registry Contents ..................................... 7
6. Security Considerations ....................................... 7
7. Normative References ......................................... 8
Appendix A. Document History .................................... 8
Author's Address ................................................ 8
1. Introduction

The JSON Private Key specification extends the JSON Web Key (JWK) [JWK] and JSON Web Algorithms (JWA) [JWA] specifications to define a JavaScript Object Notation (JSON) [RFC4627] representation of private keys.

1.1. Notational Conventions

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 Key words for use in RFCs to Indicate Requirement Levels [RFC2119].

2. Terminology

This specification uses the same terminology as the JSON Web Key (JWK) [JWK] and JSON Web Algorithms (JWA) [JWA] specifications.

3. JWK Parameters for Private Keys

This section defines additional JSON Web Key parameters that enable JWKs to represent private keys.

3.1. JWK Parameters for Elliptic Curve Private Keys

When the JWK "alg" member value is "EC", the following member MAY be used to represent an Elliptic Curve private key:

3.1.1. "d" (ECC Private Key) Parameter

The "d" (ECC private key) member contains the Elliptic Curve private key value. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array. The array representation MUST not be shortened to omit any leading zero bytes. For instance, when representing 521 bit integers, the byte array to be base64url encoded MUST contain 66 bytes, including any leading zero bytes.

3.2. JWK Parameters for RSA Private Keys

When the JWK "alg" member value is "RSA", the following member MAY be used to represent an RSA private key:
3.2.1. "d" (Private Exponent) Parameter

The "d" (private exponent) member contains the private exponent value for the RSA private key. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array. The array representation MUST not be shortened to omit any leading zero bytes. For instance, when representing 2048 bit integers, the byte array to be base64url encoded MUST contain 256 bytes, including any leading zero bytes.

3.2.2. "p" (First Prime Factor) Parameter

The "p" (first prime factor) member contains the first prime factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

3.2.3. "q" (Second Prime Factor) Parameter

The "q" (second prime factor) member contains the second prime factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

3.2.4. "dp" (First Factor CRT Exponent) Parameter

The "dp" (first factor CRT exponent) member contains the Chinese Remainder Theorem (CRT) exponent of the first factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

3.2.5. "dq" (Second Factor CRT Exponent) Parameter

The "dq" (second factor CRT exponent) member contains the Chinese Remainder Theorem (CRT) exponent of the second factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

3.2.6. "qi" (First CRT Coefficient) Parameter

The "dp" (first CRT coefficient) member contains the Chinese Remainder Theorem (CRT) coefficient of the second factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

3.2.7. "oth" (Other Primes Info) Parameter

The "oth" (other primes info) member contains an array of information about any third and subsequent primes, should they exist. When only
two primes have been used (the normal case), this parameter MUST be omitted. When three or more primes have been used, the number of array elements MUST be the number of primes used minus two. Each array element MUST be an object with the following members:

### 3.2.7.1. "r" (Prime Factor)

The "r" (prime factor) parameter within an "oth" array member represents the value of a subsequent prime factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

### 3.2.7.2. "d" (Factor CRT Exponent)

The "d" (Factor CRT Exponent) parameter within an "oth" array member represents the CRT exponent of the corresponding prime factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

### 3.2.7.3. "t" (Factor CRT Coefficient)

The "t" (factor CRT coefficient) parameter within an "oth" array member represents the CRT coefficient of the corresponding prime factor, a positive integer. It is represented as the base64url encoding of the value's unsigned big endian representation as a byte array.

### 4. Example Private Keys

The following example JWK Set contains two keys represented as JWKs containing both public and private key values: one using an Elliptic Curve algorithm and a second one using an RSA algorithm. This example extends the example in Section 3 of [JWK], adding private key values. (Line breaks are for display purposes only.)
"keys": [
  {
    "alg": "EC",
    "crv": "P-256",
    "x": "MKBTCNlCKUSDii11ySs35261DZ8AiTo7Tu6KPAqv7D4",
    "y": "4Et6SR2YilUrL5Nv5fVhuhp7x8Px1tmWlwbbM4IfyM",
    "d": "870M6gfuTJ4HtUnVYWMyJpr5eUZNPal4k43bVdj3eAE",
    "use": "enc",
    "kid": "1"
  },
  {
    "alg": "RSA",
    "n": "0x7aogebGcSuuPiLJXZptN9nndrQmbXEp5aiAFBWhM78LhwX4
          cbbfAAvtT68z6wuiR7aPFFxuhDR1l6tSoc_BJECPeWKRXjBZCiFV4n3oKnfhjMst
          n64tZ_tW-5JsG4HCs91YBAw1l931q7w_h5w66F0h4QyQ5v-65Y6YjQRo_FD6WQ
          vzqY368QMlCAtaSgq5KJZgYb9c7d0zgdAHzu6MQvRL5hajrnn191IC0pbIS
          D0bqNYrlrkt-bFTHwAI4vMQFh6WeZu8fM41Ff2NcRwr3XPksINHqQ-G_xBniIqbw
          0Ls1jF44-csFCur-kEgU8awapJzKq6DKgw",
    "e": "AQAB",
    "d": "X4tCtTJ_yg4FYpSXB8rdXix5wvsw1FGLN5E3Ea6GRJoVH-HLKDL2
          M7dx936U6KhnhrRweUK7T5fjLMWbFAKNLYw2v76NqXzUVxZ_T0_Ysfqiij
          wp3RTz1BaCw4p4dofK5N2o8Gy_y-hHKnKroADIk46PrUohsXywbrAdYaMwFs9tv8d
          _cPVY310jA378MN6Tnw0dSawm947Ui13K52ziG7xoPl4sbg1U2jxi41BTBZ
          nbJSzFHk66jT8bkgXsGjkxGksZDK1j9Z4jwbsnn4j2Bi3RL-UslGVyK8fFk
          meiz0HBidfz6YmeqNYkjcQ04jJfckAoAC8Q",
    "q": "83i-7IvMGXoXMCskv73TKr8637Fi07Z27zv8o6pbWUQyLPQ8xtPV
          nwD20R-60eTDm2ujnM5P0qMrm8RfmNhWVdtjJmMmCmjopXxicFHJ7Xo0V1Qyqv
          WlWEn6dN36GVYyk93N8Bc9vY41ixy8B9BzzOGVqXvNEvnl700nVbfs",
    "q": "3df0R9cuYq-0S-mkFfLzgITgMEfFzB3q3weHuMuG00Cuqnb3vblolYum
          jqvQ01dIrwdgTnCdpYzBcofW5r370A8jiw5t_NGEiovonIchzKpo9VVS7TzFg
          kIDrecRezsZ-1kYd_iq1DBxtkdEGfAITAG9LUnADun4rbc6yelxk",
    "dp": "4sPxmC6y9a9ly8y0Wj9_IJx4upp0u0liH7yV7ks8xj5sd3eo000m
          YxwiI2eMaAue00a5dpgFybi4c8tQ2VF4Q2XRugKDTp8kYh05tAA77Qe-Nmtu
          Yzc3C3m1242G2ivR5sSDxuynAN2zq8Lfn9Eums6rY30b8Ye1kKtIBj0",
    "d": "s91Ah9fggBsoFR80ac2R_E2gw28Z2rT2KG0AhV1LLEt1efrA6huUU
          vMFbCmBnle8ww6vznYY555OF7pMdc_agI3nG8Ib18U6b0UIuraRqUfLhcQb_d9
          GFr4dh7e74WbfvbsobnRouJyNY1xCaP6T061jVWrX-L18txXw494Q_cgk",
    "qi": "GyM_p6JrXySiz1toFgKbWV-Jdl3jQ4ypu9rbMwX3rQ3J8fmt50FoYzg
          UIizVFCGqOwemRN810oDaAA-Bk0KWNyDjJHZD3dmFw3AN71P-pukx_mHZGCJI1r
          yrR8055XlSe3SPmRfKwZr6yU24ZxvQKFYItldUKGz061a62zTKhAVRU",
    "kid": "2011-04-29"
  }
]

5. IANA Considerations
5.1. JSON Web Key Parameters Registration

This specification registers the parameter names defined in Section 3.1 and Section 3.2 in the IANA JSON Web Key Parameters registry [JWK].

5.1.1. Registry Contents

- Parameter Name: "d"
  - Change Controller: IETF
  - Specification Document(s): Section 3.1.1 of [this document]

- Parameter Name: "d"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.1 of [this document]

- Parameter Name: "p"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.2 of [this document]

- Parameter Name: "q"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.3 of [this document]

- Parameter Name: "dp"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.4 of [this document]

- Parameter Name: "dq"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.5 of [this document]

- Parameter Name: "qi"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.6 of [this document]

- Parameter Name: "oth"
  - Change Controller: IETF
  - Specification Document(s): Section 3.2.7 of [this document]

6. Security Considerations

The security considerations for this specification are the same as those for the JSON Web Key (JWK) [JWK] specification and the portion of the JSON Web Algorithms (JWA) [JWA] specification that pertains to key representations.
7. Normative References


Appendix A. Document History

[[ to be removed by the RFC editor before publication as an RFC ]]

-01

o Changed the names of the RSA key parameters so that the identifiers are the same as those used in RFC 3447.

-00

o Added the RSA private key fields enabling Chinese Remainder Theorem (CRT) calculations, based upon their use in RFC 3447.

-00

o Created draft-jones-jose-json-private-key to facilitate discussion of the question from the W3C WebCrypto WG to the IETF JOSE WG of whether JOSE plans to support a format for representing private keys.

Author's Address

Michael B. Jones
Microsoft

Email: mbj@microsoft.com
URI: http://self-issued.info/