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
   3.2. 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": "MKBCN6kKUSidi1i1y5s5256iDZBaiTo7Tu6KPAqv7D4",
      "y": "4Etl6SRWyiLUR5Nv5fVhUhp7x8Pxl1tmWWlbbM4FYm",
      "d": "870MB6gfU7HrUnVYMJpr5eUZNP4Bk43bVdj3eAE",
      "use": "enc",
      "kid": "1"
    },
    {
      "alg": "RSA",
      "n": "0x7ageoebGcQSuuPiLJXZptN9nndrQmBxEps2aiAFBWhM78Lhwx4cbbfAA9ATV86zwu1RK7aPFXuhDR16tSoc_BJECpebWKRXjBZCiFV4n3oknjHmStn64t_z2w_5JsG4Hc5n9yBXArw113ql77_RNn6Cf4q4Qv5v-65YG6QjR0_7FDW2Qvzq36bQMcIaTszs8KJZgYnB9c7d0zgDAHzHu6mQvRL5habrrn191CboqbiMSD0bQNYrdrkt-bFThAI4vMQFhh6mEuZu0fM41Fd2NcRwr3XPKsINH1Q-G_xBi1iQbw0Ls1jF44-csFCur-kEgU8awapJzKvnqDKgw",
      "e": "AQAB",
      "d": "X4CTeJY_gn4FYPsXB8rdXiz5vws1gF1N5En3Ea6GJoVH-HLLKD9M7dx5oo7GURKchnrReuUKc7hT5fJLM0WBbFAKNLWY2v7B6NqS2zxUvzT0_YSfqiwp3RT2l8CaXwp4odFK5n20g6y_nHNKroA0IkJ4n6PUohsXywbReAdYaMwFs9tv8d_cPVY3i07a3t8MN6TNWm0dfSawm94V7Ui1C13SK5i7GxoPLU4sbg1U2jx4IBTNBznbJszFKH66j78bgkqskg0gjskDJK19Zqjwbsnn4j2Wbi3RL-Us21GKV8yFZKmei2zHblKfzG6mquOytqc0X4jfcKoAC8Q",
      "p": "83i-71vMGX0mXCSkv7t3Kr8637Fi07Z27zv80j6pbWUQyLpQPQxtPVnwD29R-60eTDmD2ujmM5PoqMrmpRfmNhVWDtjMmCmj0pSXicFhj7X0uVIQyqvWlWHe6dN36GVZYk93N88c9vY41xy8B9RzzOGVQzXvNEv700nVbfs",
      "q": "3df0R9cuYq-QS-mkFLzgItgMEffZB2q3hWeHuM0G0Cuqn3b3voBlyumqjVQO3dIdwrgTnCdpYy2Cd0fW5r370A4RjxjWt_N6EiovonizhKp09VVS7TszFgXkiDrecRezS-1kYd_s1qDbx7kD3Q1vA1TAG9LUnA0nu4vC6b6yelxk",
      "dp": "4Gs PXkC6ya9y8o8W9j_Ij4xuppo01zi_-H7YTVKs8xjS5dX3c0EOoimYxxII2emTaue0U0a5dpqFyBj4c8tQ2VF4Q2XRugKDT9p80YhF05tAA77Qe_q_2m0tYZc3C3mI24G2GrV5ssDxXyAn2qZ8Lfn9Eums6Y30b8YeikKtIBj0",
      "dq": "s91AH9fggBsoFR80ac2R_E2gw282rT2kG0Ahv11LTE1efrA6huUUvMFBcmmp8q6w6vznY555QF7pMdc_agI3n8Ibp1Bu0b0JUiaraNq4UfLvhCp_d9GF4dh7e74WbRsoBnroujYTN1xCap6T061jWwrX-L18txw494Q_cgk",
      "qi": "GyM_p6JrXySiz1toFgKbWV-JdI3jQ4ypu9rbMWx3rQJGfmt0F0YmgUIEVEFecQwemMRN81oZoaA-BA-0kWNGDjJHZD0mFhW3AN7II-puxk_mHZQ1rxyR8055LSlE3SpMrfKwZl6yU24ZvXQKYFItlddUKGz061a6zTKhAVRU",
      "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.

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/