Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 21, 2021

D. Atkins
Veridify Security

November 17, 2020

Use of the Walnut Digital Signature Algorithm with CBOR Object Signing and Encryption (COSE) draft-atkins-suit-cose-walnutdsa-06

Abstract

This document specifies the conventions for using the Walnut Digital Signature Algorithm (WalnutDSA) for digital signatures with the CBOR Object Signing and Encryption (COSE) syntax. WalnutDSA is a lightweight, quantum-resistant signature scheme based on Group Theoretic Cryptography with implementation and computational efficiency of signature verification in constrained environments, even on 8- and 16-bit platforms.

The goal of this publication is to document a way to use the lightweight, quantum-resistant WalnutDSA signature algorithm in COSE in a way that would allow multiple developers to build compatible implementations. As of this publication, WalnutDSA has not been endorsed by the IETF.

WalnutDSA(TM) and Walnut Digital Signature Algorithm(TM) are trademarks of Veridify Security Inc..

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 21, 2021.

Table of Contents

1. Introduction

This document specifies the conventions for using the Walnut Digital Signature Algorithm (WalnutDSA) [WALNUTDSA] for digital signatures with the CBOR Object Signing and Encryption (COSE) [RFC8152] syntax. WalnutDSA is a Group-Theoretic [GTC] signature scheme where signature

validation is both computationally- and space-efficient, even on very small processors. Unlike many hash-based signatures, there is no state required and no limit on the number of signatures that can be made. WalnutDSA private and public keys are relatively small; however, the signatures are larger than RSA and ECC, but still smaller than most all other quantum-resistant schemes (including all hash-based schemes).

COSE provides a lightweight method to encode structured data. WalnutDSA is a lightweight, quantum-resistant WalnutDSA signature algorithm. The goal of this specification is to document a method to leverage WalnutDSA in COSE in a way that would allow multiple developers to build compatible implementations.

As with all cryptosystems, the initial versions of WalnutDSA underwent significant cryptanalysis, and in some cases, identified potential issues. For more discussion on this topic, a summary of all published cryptanalysis can be found in Section 5.2. Validated issues were addressed by reparameterization in updated versions of WalnutDSA. Although the IETF has not endorsed WalnutDSA as of this publication, this document provides a method to use WalnutDSA in conjunction with IETF protocols. As always, users of any security algorithm are advised to research the security properties of the algorithm and make their own judgment about the risks involved.

1.1. Motivation

Recent advances in cryptanalysis [BH2013] and progress in the development of quantum computers [NAS2019] pose a threat to widely deployed digital signature algorithms. As a result, there is a need to prepare for a day that cryptosystems such as RSA and DSA that depend on discrete logarithm and factoring cannot be depended upon.

If large-scale quantum computers are ever built, these computers will be able to break many of the public-key cryptosystems currently in use. A post-quantum cryptosystem [PQC] is a system that is secure against quantum computers that have more than a trivial number of quantum bits (qubits). It is open to conjecture when it will be feasible to build such computers; however, RSA, DSA, ECDSA, and EdDSA are all vulnerable if large-scale quantum computers come to pass.

WalnutDSA does not depend on the difficulty of discrete logarithm or factoring. As a result this algorithm is considered to be resistant to post-quantum attacks.

Today, RSA and ECDSA are often used to digitally sign software updates. Unfortunately, implementations of RSA and ECDSA can be relatively large, and verification can take a significant amount of

time on some very small processors. Therefore, we desire a digital signature scheme that verifies faster with less code. Moreover, in preparation for a day when RSA, DSA, and ECDSA cannot be depended upon, a digital signature algorithm is needed that will remain secure even if there are significant cryptoanalytic advances or a large-scale quantum computer is invented. WalnutDSA, specified in [WALNUTSPEC], is one such algorithm.

1.2. Trademark Notice

WalnutDSA(TM) and Walnut Digital Signature Algorithm(TM) are trademarks of Veridify Security Inc..

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. WalnutDSA Algorithm Overview

This specification makes use of WalnutDSA signatures as described in [WALNUTDSA] and more concretely specified in [WALNUTSPEC]. WalnutDSA is a Group-Theoretic cryptographic signature scheme that leverages infinite group theory as the basis of its security and maps that to a one-way evaluation of a series of matrices over small finite fields with permuted multiplicants based on the group input. WalnutDSA leverages the SHA2-256 and SHA2-512 one-way hash algorithms [SHA2] in a hash-then-sign process.

WalnutDSA is based on a one-way function, E-Multiplication, which is an action on the infinite group. A single E-Multiplication step takes as input a matrix and permutation, a generator in the group, and a set of T-values (entries in the finite field) and outputs a new matrix and permutation. To process a long string of generators (like a WalnutDSA signature), E-Multiplication is iterated over each generator. Due to its structure, E-Multiplication is extremely easy to implement.

In addition to being quantum-resistant, the two main benefits of using WalnutDSA are that the verification implementation is very small and WalnutDSA signature verification is extremely fast, even on very small processors (including 16- and even 8-bit MCUs). This lends it well to use in constrained and/or time-sensitive environments.

WalnutDSA has several parameters required to process a signature. The main parameters are N and q. The parameter N defines the size of the group by defining the number of strands in use, and implies working in an NxN matrix. The parameter q defines the number of elements in the finite field. Signature verification also requires a set of T-values, which is an ordered list of N entries in the finite field F_q.

A WalnutDSA signature is just a string of generators in the infinite group, packed into a byte string.

4. WalnutDSA Algorithm Identifiers

The CBOR Object Signing and Encryption (COSE) [RFC8152] supports two signature algorithm schemes. This specification makes use of the signature with appendix scheme for WalnutDSA signatures.

The signature value is a large byte string. The byte string is designed for easy parsing, and it includes a length (number of generators) and type codes that indirectly provide all of the information that is needed to parse the byte string during signature validation.

When using a COSE key for this algorithm, the following checks are made:

- The 'kty' field MUST be present, and it MUST be 'WalnutDSA'.
- If the 'alg' field is present, and it MUST be 'WalnutDSA'.
- If the 'key_ops' field is present, it MUST include 'sign' when creating a WalnutDSA signature.
- If the 'key_ops' field is present, it MUST include 'verify' when verifying a WalnutDSA signature.
- If the 'kid' field is present, it MAY be used to identify the WalnutDSA Key.

5. Security Considerations

5.1. Implementation Security Considerations

Implementations MUST protect the private keys. Use of a hardware security module (HSM) is one way to protect the private keys. Compromise of the private keys may result in the ability to forge signatures. As a result, when a private key is stored on non-

volatile media or stored in a virtual machine environment, care must be taken to preserve confidentiality and integrity.

The generation of private keys relies on random numbers. The use of inadequate pseudo-random number generators (PRNGs) to generate these values can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys, searching the resulting small set of possibilities, rather than brute force searching the whole key space. The generation of quality random numbers is difficult, and [RFC4086] offers important guidance in this area.

The generation of WalnutDSA signatures also depends on random numbers. While the consequences of an inadequate pseudo-random number generator (PRNG) to generate these values is much less severe than the generation of private keys, the guidance in [RFC4086] remains important.

5.2. Method Security Considerations

The Walnut Digital Signature Algorithm has undergone significant cryptanalysis since it was first introduced, and several weaknesses were found in early versions of the method, resulting in the description of several attacks with exponential computational complexity. A full writeup of all the analysis can be found in [WalnutDSAAnalysis]. In summary, the original suggested parameters (N=8, q=32) were too small, leading to many of these exponential-growth attacks being practical. However, current parameters render these attacks impractical. The following paragraphs summarize the analysis and how the current parameters defeat all the previous attacks.

First, the team of Hart et al found a universal forgery attack based on a group factoring problem that runs in O(q^((N-1)/2)) with a memory complexity of log_2(q) N^2 q^((N-1)/2). With parameters N=10 and q=M31 (the Mersenne prime 2^31 - 1), the runtime is 2^139 and memory complexity is 2^151. W. Beullens found a modification of this attack but its runtime is even longer.

Next, Beullens and Blackburn found several issues with the original method and parameters. First they used a Pollard-Rho attack and discovered the original public key space was too small. Specifically they require that q^(N(N-1)-1) > 2^(2*Security Level). One can clearly see that N=10, q=M31 provides 128-bit security and N=10, q=M61 provides 256-bit security.

Beullens and Blackburn also found two issues with the original message encoder of WalnutDSA. First, the original encoder was non-

injective, which reduced the available signature space. This was repaired in an update. Second, they pointed out that the dimension of the vector space generated by the encoder was too small. Specifically, they require that q^dimension > 2^(2*Security Level). With N=10, the current encoder produces a dimension of 66 which clearly provides sufficient security with q=M31 or q=M61.

The final issue discovered by Beullens and Blackburn was a process to theoretically "reverse" E-Multiplication. First, their process requires knowing the initial matrix and permutation (which is known for WalnutDSA). But more importantly, their process runs at O(q^((N-1)/2)) which, for N=10, q=M31 is greater than 2^128.

A team at Steven's Institute leveraged a length-shortening attack that enabled them to remove the cloaking elements and then solve a conjugacy search problem to derive the private keys. Their attack requires both knowledge of the permutation being cloaked and also that the cloaking elements themselves are conjugates. By adding additional concealed cloaking elements the attack requires an N! search for each cloaking element. By inserting k concealed cloaking elements, this requires the attacker to perform (N!)^k work. This allows k to be set to meet the desired security level.

Finally, Merz and Petit discovered that using a Garside Normal Form of a WalnutDSA signature enabled them to find commonalities with the Garside Normal Form of the encoded message. Using those commonalities they were able to splice into a signature and create forgeries. Increasing the number of cloaking elements, specifically within the encoded message, sufficiently obscures the commonalities and blocks this attack.

In summary, most of these attacks are exponential in run time and can be shown that current parameters put the runtime beyond the desired security level. The final two attacks are also sufficiently blocked to the desired security level.

6. IANA Considerations

IANA is requested to add entries for WalnutDSA signatures in the "COSE Algorithms" registry and WalnutDSA public keys in the "COSE Key Types" and "COSE Key Type Parameters" registries.

6.1. COSE Algorithms Registry Entry

The new entry in the "COSE Algorithms" registry has the following columns:

Name: WalnutDSA

Value: TBD1 (Value between -65536 to -257 or 256-65535 to be assigned by IANA)

Description: WalnutDSA signature

Reference: This document (Number to be assigned by RFC Editor)

Recommended: No

6.2. COSE Key Types Registry Entry

The new entry in the "COSE Key Types" registry has the following columns:

Name: WalnutDSA

Value: TBD2 (Value to be assigned by IANA)

Description: WalnutDSA public key

Reference: This document (Number to be assigned by RFC Editor)

6.3. COSE Key Type Parameter Registry Entries

The following sections detail the additions to the "COSE Key Type Parameters" registry.

6.3.1. WalnutDSA Parameter: N

The new entry N in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: N

Label: TBD (Value to be assigned by IANA)

CBOR Type: uint

Description: Group and Matrix (NxN) size

Reference: This document (Number to be assigned by RFC Editor)

6.3.2. WalnutDSA Parameter: q

The new entry q in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: q

Label: TBD (Value to be assigned by IANA)

CBOR Type: uint

Description: Finite field F_q

Reference: This document (Number to be assigned by RFC Editor)

6.3.3. WalnutDSA Parameter: t-values

The new entry t-values in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: t-values

Label: TBD (Value to be assigned by IANA)

CBOR Type: array (of uint)

Description: List of T-values, enties in F_q

Reference: This document (Number to be assigned by RFC Editor)

6.3.4. WalnutDSA Parameter: matrix 1

The new entry matrix 1 in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: matrix 1

Label: TBD (Value to be assigned by IANA)

CBOR Type: array (of array of uint)

Description: NxN Matrix of enties in F_q in column-major form

Reference: This document (Number to be assigned by RFC Editor)

6.3.5. WalnutDSA Parameter: permutation 1

The new entry permutation 1 in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: permutation 1

Label: TBD (Value to be assigned by IANA)

CBOR Type: array (of uint)

Description: Permutation associated with matrix 1

Reference: This document (Number to be assigned by RFC Editor)

6.3.6. WalnutDSA Parameter: matrix 2

The new entry matrix 2 in the "COSE Key Type Parameters" registry has the following columns:

Key Type: TBD2 (Value assigned by IANA above)

Name: matrix 2

Label: TBD (Value to be assigned by IANA)

CBOR Type: array (of array of uint)

Description: NxN Matrix of enties in F_q in column-major form

Reference: This document (Number to be assigned by RFC Editor)

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[SHA2] National Institute of Standards and Technology (NIST), "FIPS Publication 180-3: Secure Hash Standard", October 2008.

[WALNUTDSA] Anshel, I., Atkins, D., Goldfeld, D., and P. Gunnells, "WalnutDSA(TM): A group-theoretic digital signature algorithm", November 2020, <https://doi.org/10.1080/23799927.2020.1831613>.

7.2. Informative References

[BH2013] Ptacek, T., Ritter, J., Samuel, J., and A. Stamos, "The Factoring Dead: Preparing for the Cryptopocalypse", August 2013, <https://media.blackhat.com/us-13/us-13-Stamos-The-Factoring-Dead.pdf>.

[GTC] Vasco, M. and R. Steinwandt, "Group Theoretic Cryptography", April 2015, <https://www.crcpress.com/Group-Theoretic-Cryptography/Vasco-Steinwandt/p/book/9781584888369>.

[NAS2019] National Academies of Sciences, Engineering, and Medicine, "Quantum Computing: Progress and Prospects", 2019, <http://dx.doi.org/10.17226/25196>.

[PQC] Bernstein, D., "Introduction to post-quantum cryptography", 2009, <http://www.pqcrypto.org/www.springer.com/cda/content/document/cda_downloaddocument/9783540887010-c1.pdf>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[WalnutDSAAnalysis] Anshel, I., Atkins, D., Goldfeld, D., and P. Gunnells, "Defeating the Hart et al, Beullens-Blackburn, Kotov-Menshov-Ushakov, and Merz-Petit Attacks on WalnutDSA(TM)", May 2019, <https://eprint.iacr.org/2019/472>.

[WALNUTSPEC]
Anshel, I., Atkins, D., Goldfeld, D., and P. Gunnells, "The Walnut Digital Signature Algorithm Specification", November 2018, <https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions>.

Appendix A. Acknowledgments

A big thank you to Russ Housley for his input on the concepts and text of this document.

Author's Address

Derek Atkins
Veridify Security
100 Beard Sawmill Rd, Suite 350
Shelton, CT 06484
US

Phone: +1 617 623 3745
Email: datkins@veridify.com