RFC 9045
Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)

Abstract
This document updates the cryptographic algorithm requirements for the Password-Based Message Authentication Code in the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF) specified in RFC 4211.

Status of This Memo
This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9045.

Copyright Notice
Copyright (c) 2021 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 (https://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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

**Table of Contents**

1. Introduction
2. Terminology
3. Signature Key POP
4. Password-Based Message Authentication Code
   4.1. Introduction Paragraph
   4.2. One-Way Function
   4.3. Iteration Count
   4.4. MAC Algorithm
5. IANA Considerations
6. Security Considerations
7. References
   7.1. Normative References
   7.2. Informative References

Acknowledgements
Author's Address
1. Introduction

This document updates the cryptographic algorithm requirements for the Password-Based Message Authentication Code (MAC) in the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF) [RFC4211]. The algorithms specified in [RFC4211] were appropriate in 2005; however, these algorithms are no longer considered the best choices:

- HMAC-SHA1 [HMAC] [SHS] is not broken yet, but there are much stronger alternatives [RFC6194].
- DES-MAC [PKCS11] provides 56 bits of security, which is no longer considered secure [WITHDRAW].
- Triple-DES-MAC [PKCS11] provides 112 bits of security, which is now deprecated [TRANSIT].

This update specifies algorithms that are more appropriate today.

CRMF is defined using Abstract Syntax Notation One (ASN.1) [X680].

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. Signature Key POP

Section 4.1 of [RFC4211] specifies the proof-of-possession (POP) processing. This section is updated to explicitly allow the use of the PBMAC1 algorithm presented in Section 7.1 of [RFC8018].

OLD:

```
algId identifies the algorithm used to compute the MAC value. All implementations MUST support id-PasswordBasedMAC. The details on this algorithm are presented in section 4.4.
```

NEW:

```
algId identifies the algorithm used to compute the MAC value. All implementations MUST support id-PasswordBasedMAC as presented in Section 4.4 of [RFC4211]. Implementations MAY also support PBMAC1 as presented in Section 7.1 of [RFC8018].
```
4. Password-Based Message Authentication Code

Section 4.4 of [RFC4211] specifies a Password-Based MAC that relies on a one-way function to compute a symmetric key from the password and a MAC algorithm. This section specifies algorithm requirements for the one-way function and the MAC algorithm.

4.1. Introduction Paragraph

Add guidance about limiting the use of the password as follows:

OLD:

This MAC algorithm was designed to take a shared secret (a password) and use it to compute a check value over a piece of information. The assumption is that, without the password, the correct check value cannot be computed. The algorithm computes the one-way function multiple times in order to slow down any dictionary attacks against the password value.

NEW:

This MAC algorithm was designed to take a shared secret (a password) and use it to compute a check value over a piece of information. The assumption is that, without the password, the correct check value cannot be computed. The algorithm computes the one-way function multiple times in order to slow down any dictionary attacks against the password value. The password used to compute this MAC SHOULD NOT be used for any other purpose.

4.2. One-Way Function

Change the paragraph describing the "owf" as follows:

OLD:

owf identifies the algorithm and associated parameters used to compute the key used in the MAC process. All implementations MUST support SHA-1.
4.3. Iteration Count

Update the guidance on appropriate iteration count values as follows:

OLD:

iterationCount identifies the number of times the hash is applied during the key computation process. The iterationCount must be a minimum of 100. Many people suggest using values as high as 1000 iterations as the minimum value. The trade off here is between protection of the password from attacks and the time spent by the server processing all of the different iterations in deriving passwords. Hashing is generally considered a cheap operation but this may not be true with all hash functions in the future.

NEW:

iterationCount identifies the number of times the hash is applied during the key computation process. The iterationCount must be a minimum of 100; however, the iterationCount should be as large as server performance will allow, typically at least 10,000 [DIGALM]. There is a trade-off between protection of the password from attacks and the time spent by the server processing the iterations. As part of that trade-off, an iteration count smaller than 10,000 can be used when automated generation produces shared secrets with high entropy.

4.4. MAC Algorithm

Change the paragraph describing the "mac" as follows:

OLD:

mac identifies the algorithm and associated parameters of the MAC function to be used. All implementations must support HMAC-SHA1 [HMAC]. All implementations should support DES-MAC and Triple-DES-MAC [PKCS11].

NEW:

owf identifies the algorithm and associated parameters used to compute the key used in the MAC process. All implementations must support SHA-256 [SHS].
NEW:

mac identifies the algorithm and associated parameters of the MAC function to be used. All implementations **MUST** support HMAC-SHA256 [HMAC]. All implementations **SHOULD** support AES-GMAC [AES] [GMAC] with a 128-bit key.

For convenience, the identifiers for these two algorithms are repeated here.

The ASN.1 algorithm identifier for HMAC-SHA256 is defined in [RFC4231]:

```
id-hmacWithSHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) digestAlgorithm(2) 9 }
```

When this object identifier is used in the ASN.1 algorithm identifier, the parameters **SHOULD** be present. When present, the parameters **MUST** contain a type of NULL as specified in [RFC4231].

The ASN.1 algorithm identifier for AES-GMAC [AES] [GMAC] with a 128-bit key is defined in [RFC9044]:

```
id-aes128-GMAC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) aes(1) 9 }
```

When this object identifier is used in the ASN.1 algorithm identifier, the parameters **MUST** be present, and the parameters **MUST** contain the GMACParameters structure as follows:

```
GMACParameters ::= SEQUENCE {
    nonce        OCTET STRING,
    length       MACLength DEFAULT 12 }
MACLength ::= INTEGER (12 | 13 | 14 | 15 | 16)
```

The GMACParameters nonce parameter is the GMAC initialization vector. The nonce may have any number of bits between 8 and (2^64)-1, but it **MUST** be a multiple of 8 bits. Within the scope of any GMAC key, the nonce value **MUST** be unique. A nonce value of 12 octets can be processed more efficiently, so that length for the nonce value is **RECOMMENDED**.

The GMACParameters length parameter field tells the size of the message authentication code in octets. GMAC supports lengths between 12 and 16 octets, inclusive. However, for use with CRMF, the maximum length of 16 octets **MUST** be used.

**5. IANA Considerations**

This document has no IANA actions.
6. Security Considerations

The security of the Password-Based MAC relies on the number of times the hash function is applied as well as the entropy of the shared secret (the password). Hardware support for hash calculation is available at very low cost [PHS], which reduces the protection provided by a high iterationCount value. Therefore, the entropy of the password is crucial for the security of the Password-Based MAC function. In 2010, researchers showed that about half of the real-world passwords in a leaked corpus can be broken with less than 150 million trials, indicating a median entropy of only 27 bits [DMR]. Higher entropy can be achieved by using randomly generated strings. For example, assuming an alphabet of 60 characters, a randomly chosen password with 10 characters offers 59 bits of entropy, and 20 characters offers 118 bits of entropy. Using a one-time password also increases the security of the MAC, assuming that the integrity-protected transaction will complete before the attacker is able to learn the password with an offline attack.

Please see [RFC8018] for security considerations related to PBMAC1.

Please see [HMAC] and [SHS] for security considerations related to HMAC-SHA256.

Please see [AES] and [GMAC] for security considerations related to AES-GMAC.

Cryptographic algorithms age; they become weaker with time. As new cryptanalysis techniques are developed and computing capabilities improve, the work required to break a particular cryptographic algorithm will reduce, making an attack on the algorithm more feasible for more attackers. While it is unknown how cryptanalytic attacks will evolve, it is certain that they will get better. It is unknown how much better they will become or when the advances will happen. For this reason, the algorithm requirements for CRMF are updated by this specification.

When a Password-Based MAC is used, implementations must protect the password and the MAC key. Compromise of either the password or the MAC key may result in the ability of an attacker to undermine authentication.

7. References

7.1. Normative References


7.2. Informative References


Acknowledgements

Many thanks to Hans Aschauer, Hendrik Brockhaus, Quynh Dang, Roman Danyliw, Lars Eggert, Tomas Gustavsson, Jonathan Hammell, Tim Hollebeek, Ben Kaduk, Erik Kline, Lijun Liao, Mike Ounsworth, Francesca Palombini, Tim Polk, Ines Robles, Mike StJohns, and Sean Turner for their careful review and improvements.

Author's Address

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA 20170
United States of America
Email: housley@vigilsec.com