

Network Working Group  
Internet-Draft  
Obsoletes: [4307](#) (if approved)  
Updates: [7296](#) (if approved)  
Intended status: Standards Track  
Expires: September 30, 2017

Y. Nir  
Check Point  
T. Kivinen  
INSIDE Secure  
P. Wouters  
Red Hat  
D. Migault  
Ericsson  
March 29, 2017

**Algorithm Implementation Requirements and Usage Guidance for IKEv2**  
**draft-ietf-ipsecme-rfc4307bis-18**

Abstract

The IPsec series of protocols makes use of various cryptographic algorithms in order to provide security services. The Internet Key Exchange (IKE) protocol is used to negotiate the IPsec Security Association (IPsec SA) parameters, such as which algorithms should be used. To ensure interoperability between different implementations, it is necessary to specify a set of algorithm implementation requirements and usage guidance to ensure that there is at least one algorithm that all implementations support. This document updates [RFC 7296](#) and obsoletes [RFC 4307](#) in defining the current algorithm implementation requirements and usage guidance for IKEv2, and does minor cleaning up of the IKEv2 IANA registry. This document does not update the algorithms used for packet encryption using IPsec Encapsulated Security Payload (ESP).

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 September 30, 2017.

## Copyright Notice

Copyright (c) 2017 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

|                        |   |                    |
|------------------------|---|--------------------|
| <a href="#">1.</a>     | <a href="#">Introduction</a>  | <a href="#">2</a>  |
| <a href="#">1.1.</a>   | <a href="#">Conventions Used in This Document</a>                                 | <a href="#">3</a>  |
| <a href="#">1.2.</a>   | <a href="#">Updating Algorithm Implementation Requirements and Usage Guidance</a> | <a href="#">3</a>  |
| <a href="#">1.3.</a>   | <a href="#">Updating Algorithm Requirement Levels</a>                             | <a href="#">4</a>  |
| <a href="#">1.4.</a>   | <a href="#">Document Audience</a>   | <a href="#">5</a>  |
| <a href="#">2.</a>     | <a href="#">Algorithm Selection</a>   | <a href="#">5</a>  |
| <a href="#">2.1.</a>   | <a href="#">Type 1 - IKEv2 Encryption Algorithm Transforms</a>                    | <a href="#">5</a>  |
| <a href="#">2.2.</a>   | <a href="#">Type 2 - IKEv2 Pseudo-random Function Transforms</a>                  | <a href="#">7</a>  |
| <a href="#">2.3.</a>   | <a href="#">Type 3 - IKEv2 Integrity Algorithm Transforms</a>                     | <a href="#">8</a>  |
| <a href="#">2.4.</a>   | <a href="#">Type 4 - IKEv2 Diffie-Hellman Group Transforms</a>                    | <a href="#">9</a>  |
| <a href="#">2.5.</a>   | <a href="#">Summary of Changes from <a href="#">RFC 4307</a></a>                  | <a href="#">10</a> |
| <a href="#">3.</a>     | <a href="#">IKEv2 Authentication</a>  | <a href="#">11</a> |
| <a href="#">3.1.</a>   | <a href="#">IKEv2 Authentication Method</a>                                       | <a href="#">11</a> |
| <a href="#">3.1.1.</a> | <a href="#">Recommendations for RSA key length</a>                                | <a href="#">12</a> |
| <a href="#">3.2.</a>   | <a href="#">Digital Signature Recommendations</a>                                 | <a href="#">12</a> |
| <a href="#">4.</a>     | <a href="#">Algorithms for Internet of Things</a>                                 | <a href="#">13</a> |
| <a href="#">5.</a>     | <a href="#">Security Considerations</a>   | <a href="#">14</a> |
| <a href="#">6.</a>     | <a href="#">IANA Considerations</a>   | <a href="#">15</a> |
| <a href="#">7.</a>     | <a href="#">Acknowledgements</a>  | <a href="#">15</a> |
| <a href="#">8.</a>     | <a href="#">References</a>  | <a href="#">16</a> |
| <a href="#">8.1.</a>   | <a href="#">Normative References</a>  | <a href="#">16</a> |
| <a href="#">8.2.</a>   | <a href="#">Informative References</a>  | <a href="#">16</a> |
|                        | <a href="#">Authors' Addresses</a>  | <a href="#">17</a> |

## [1. Introduction](#)

The Internet Key Exchange (IKE) protocol [[RFC7296](#)] is used to negotiate the parameters of the IPsec SA, such as the encryption and authentication algorithms and the keys for the protected communications between the two endpoints. The IKE protocol itself is



also protected by cryptographic algorithms which are negotiated between the two endpoints using IKE. Different implementations of IKE may negotiate different algorithms based on their individual local policy. To ensure interoperability, a set of "mandatory-to-implement" IKE cryptographic algorithms is defined.

This document describes the parameters of the IKE protocol and updates the IKEv2 specification. It changes the mandatory to implement authentication algorithms of [Section 4 of \[RFC7296\]](#) by saying RSA key lengths of less than 2048 SHOULD NOT be used. It does not describe the cryptographic parameters of the AH or ESP protocols.

### **[1.1.](#) Conventions Used in This Document**

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 [\[RFC2119\]](#).

When used in the tables in this document, these terms indicate that the listed algorithm MUST, MUST NOT, SHOULD, SHOULD NOT or MAY be implemented as part of an IKEv2 implementation. Additional terms used in this document are:

- |         |  |
|---------|--|
| SHOULD+ | This term means the same as SHOULD. However, it is likely that an algorithm marked as SHOULD+ will be promoted at some future time to be a MUST.   |
| SHOULD- | This term means the same as SHOULD. However, an algorithm marked as SHOULD- may be deprecated to a MAY in a future version of this document.   |
| MUST-   | This term means the same as MUST. However, it is expected at some point that this algorithm will no longer be a MUST in a future document. Although its status will be determined at a later time, it is reasonable to expect that if a future revision of a document alters the status of a MUST- algorithm, it will remain at least a SHOULD or a SHOULD- level. |
| IoT     | stands for Internet of Things.   |

### **[1.2.](#) Updating Algorithm Implementation Requirements and Usage Guidance**

The field of cryptography evolves continuously. New stronger algorithms appear and existing algorithms are found to be less secure than originally thought. Therefore, algorithm implementation requirements and usage guidance need to be updated from time to time to reflect the new reality. The choices for algorithms must be conservative to minimize the risk of algorithm compromise. Algorithms need to be suitable for a wide variety of CPU



architectures and device deployments ranging from high end bulk encryption devices to small low-power IoT devices.

The algorithm implementation requirements and usage guidance may need to change over time to adapt to the changing world. For this reason, the selection of mandatory-to-implement algorithms was removed from the main IKEv2 specification and placed in this separate document.

### **1.3. Updating Algorithm Requirement Levels**

The mandatory-to-implement algorithm of tomorrow should already be available in most implementations of IKE by the time it is made mandatory. This document attempts to identify and introduce those algorithms for future mandatory-to-implement status. There is no guarantee that the algorithms in use today may become mandatory in the future. Published algorithms are continuously subjected to cryptographic attack and may become too weak or could become completely broken before this document is updated.

This document only provides recommendations for the mandatory-to-implement algorithms or algorithms too weak that are recommended not to be implemented. As a result, any algorithm listed at the IKEv2 IANA registry not mentioned in this document MAY be implemented. For clarification and consistency with [\[RFC4307\]](#) an algorithm will be denoted here as MAY only when it has been downgraded.

Although this document updates the algorithms to keep the IKEv2 communication secure over time, it also aims at providing recommendations so that IKEv2 implementations remain interoperable. IKEv2 interoperability is addressed by an incremental introduction or deprecation of algorithms. In addition, this document also considers the new use cases for IKEv2 deployment, such as Internet of Things (IoT).

It is expected that deprecation of an algorithm is performed gradually. This provides time for various implementations to update their implemented algorithms while remaining interoperable. Unless there are strong security reasons, an algorithm is expected to be downgraded from MUST to MUST- or SHOULD, instead of MUST NOT. Similarly, an algorithm that has not been mentioned as mandatory-to-implement is expected to be introduced with a SHOULD instead of a MUST.

The current trend toward Internet of Things and its adoption of IKEv2 requires this specific use case to be taken into account as well. IoT devices are resource constrained devices and their choice of algorithms are motivated by minimizing the footprint of the code, the computation effort and the size of the messages to send. This



document indicates "(IoT)" when a specified algorithm is specifically listed for IoT devices. Requirement levels that are marked as "IoT" apply to IoT devices and to server-side implementations that might presumably need to interoperate with them, including any general-purpose VPN gateways.

#### **1.4. Document Audience**

The recommendations of this document mostly target IKEv2 implementers who need to create implementations that meet both high security expectations as well as high interoperability between various vendors and with different versions. Interoperability requires a smooth move to more secure cipher suites. This may differ from a user point of view that may deploy and configure IKEv2 with only the safest cipher suite.

This document does not give any recommendations for the use of algorithms, it only gives implementation recommendations regarding implementations. The use of algorithms by users is dictated by the security policy requirements for that specific user, and are outside the scope of this document.

IKEv1 is out of scope of this document. IKEv1 is deprecated and the recommendations of this document must not be considered for IKEv1, as most IKEv1 implementations have been "frozen" and will not be able to update the list of mandatory-to-implement algorithms.

## **2. Algorithm Selection**

### **2.1. Type 1 - IKEv2 Encryption Algorithm Transforms**

The algorithms in the below table are negotiated in the SA payload and used for the Encrypted Payload. References to the specification defining these algorithms and the ones in the following subsections are in the IANA registry [[IKEV2-IANA](#)]. Some of these algorithms are Authenticated Encryption with Associated Data (AEAD - [[RFC5282](#)]). Algorithms that are not AEAD MUST be used in conjunction with one of the integrity algorithms in [Section 2.3](#).



| Name                   | Status   | AEAD? | Comment |
|------------------------|----------|-------|---------|
| ENCR_AES_CBC           | MUST     | No    | (1)     |
| ENCR_CHACHA20_POLY1305 | SHOULD   | Yes   |         |
| ENCR_AES_GCM_16        | SHOULD   | Yes   | (1)     |
| ENCR_AES_CCM_8         | SHOULD   | Yes   | (IoT)   |
| ENCR_3DES              | MAY      | No    |         |
| ENCR_DES               | MUST NOT | No    |         |

(1) - This requirement level is for 128-bit and 256-bit keys. 192-bit keys remain at MAY level. (IoT) - This requirement is for interoperability with IoT. Only 128-bit keys are at SHOULD level. 192-bit and 256-bit remain at the MAY level.

ENCR\_AES\_CBC is raised from SHOULD+ for 128-bit keys and MAY for 256-bit keys in [RFC4307](#) to MUST. 192-bit keys remain at the MAY level. ENCR\_AES\_CBC is the only shared mandatory-to-implement algorithm with [RFC4307](#) and as a result it is necessary for interoperability with IKEv2 implementation compatible with [RFC4307](#).

ENCR\_CHACHA20\_POLY1305 was not ready to be considered at the time of [RFC4307](#). It has been recommended by the Crypto Forum Research Group (CFRG) of the IRTF as an alternative to AES-CBC and AES-GCM. It is also being standardized for IPsec for the same reasons. At the time of writing, there were not enough IKEv2 implementations supporting ENCR\_CHACHA20\_POLY1305 to be able to introduce it at the SHOULD+ level.

ENCR\_AES\_GCM\_16 was not considered in [RFC4307](#). At the time [RFC4307](#) was written, AES-GCM was not defined in an IETF document. AES-GCM was defined for ESP in [RFC4106](#) and later for IKEv2 in [RFC5282](#). The main motivation for adopting AES-GCM for ESP is encryption performance compared to AES-CBC. This resulted in AES-GCM being widely implemented for ESP. As the computation load of IKEv2 is relatively small compared to ESP, many IKEv2 implementations have not implemented AES-GCM. For this reason, AES-GCM is not promoted to a greater status than SHOULD. The reason for promotion from MAY to SHOULD is to promote the slightly more secure AEAD method over the traditional encrypt+auth method. Its status is expected to be raised once widely implemented. As the advantage of the shorter (and weaker) ICVs is minimal, the 8 and 12 octet ICV's remain at the MAY level.

ENCR\_AES\_CCM\_8 was not considered in [RFC4307](#). This document considers it as SHOULD be implemented in order to be able to interact with Internet of Things devices. As this case is not a general use



case for non-IoT VPNs, its status is expected to remain as SHOULD. The 8 octet size of the ICV is expected to be sufficient for most use cases of IKEv2, as far less packets are exchanged in those cases, and IoT devices want to make packets as small as possible. The SHOULD level is for 128-bit keys, 256-bit keys remains at MAY level.

ENCR\_3DES has been downgraded from [RFC4307](#) MUST- to MAY. All IKEv2 implementations already implement ENCR\_AES\_CBC, so there is no need to keep support for the much slower ENCR\_3DES. In addition, ENCR\_CHACHA20\_POLY1305 provides a more modern alternative to AES.

ENCR\_DES can be brute-forced using off-the-shelf hardware. It provides no meaningful security whatsoever and therefore MUST NOT be implemented.

## 2.2. Type 2 - IKEv2 Pseudo-random Function Transforms

Transform Type 2 algorithms are pseudo-random functions used to generate pseudo-random values when needed.

| Name              | Status   | Comment |
|-------------------|----------|---------|
| PRF_HMAC_SHA2_256 | MUST     |         |
| PRF_HMAC_SHA2_512 | SHOULD+  |         |
| PRF_HMAC_SHA1     | MUST-    |         |
| PRF_AES128_XCBC   | SHOULD   | (IoT)   |
| PRF_HMAC_MD5      | MUST NOT |         |

(IoT) - This requirement is for interoperability with IoT

As no SHA2 based transforms were referenced in [RFC4307](#), PRF\_HMAC\_SHA2\_256 was not mentioned in [RFC4307](#). PRF\_HMAC\_SHA2\_256 MUST be implemented in order to replace SHA1 and PRF\_HMAC\_SHA1.

PRF\_HMAC\_SHA2\_512 SHOULD be implemented as a future replacement for PRF\_HMAC\_SHA2\_256 or when stronger security is required. PRF\_HMAC\_SHA2\_512 is preferred over PRF\_HMAC\_SHA2\_384, as the additional overhead of PRF\_HMAC\_SHA2\_512 is negligible.

PRF\_HMAC\_SHA1 has been downgraded from MUST in [RFC4307](#) to MUST- as cryptographic attacks against SHA1 are increasing, resulting in an industry-wide trend to deprecate its usage

PRF\_AES128\_XCBC is only recommended in the scope of IoT, as Internet of Things deployments tend to prefer AES based pseudo-random functions in order to avoid implementing SHA2. For the non-IoT VPN



deployment it has been downgraded from SHOULD in [RFC4307](#) to MAY as it has not seen wide adoption.

PRF\_HMAC\_MD5 has been downgraded from MAY in [RFC4307](#) to MUST NOT. Cryptographic attacks against MD5, such as collision attacks mentioned in [[TRANSCRIPTION](#)], are resulting in an industry-wide trend to deprecate and remove MD5 (and thus HMAC-MD5) from cryptographic libraries.

### 2.3. Type 3 - IKEv2 Integrity Algorithm Transforms

The algorithms in the below table are negotiated in the SA payload and used for the Encrypted Payload. References to the specification defining these algorithms are in the IANA registry. When an AEAD algorithm (see [Section 2.1](#)) is proposed, this algorithm transform type is not in use.

| Name                   | Status   | Comment |
|------------------------|----------|---------|
| AUTH_HMAC_SHA2_256_128 | MUST     |         |
| AUTH_HMAC_SHA2_512_256 | SHOULD   |         |
| AUTH_HMAC_SHA1_96      | MUST-    |         |
| AUTH_AES_XCBC_96       | SHOULD   | (IoT)   |
| AUTH_HMAC_MD5_96       | MUST NOT |         |
| AUTH_DES_MAC           | MUST NOT |         |
| AUTH_KPDK_MD5          | MUST NOT |         |

(IoT) - This requirement is for interoperability with IoT

AUTH\_HMAC\_SHA2\_256\_128 was not mentioned in [RFC4307](#), as no SHA2 based transforms were mentioned. AUTH\_HMAC\_SHA2\_256\_128 MUST be implemented in order to replace AUTH\_HMAC\_SHA1\_96.

AUTH\_HMAC\_SHA2\_512\_256 SHOULD be implemented as a future replacement of AUTH\_HMAC\_SHA2\_256\_128 or when stronger security is required. This value has been preferred over AUTH\_HMAC\_SHA2\_384, as the additional overhead of AUTH\_HMAC\_SHA2\_512 is negligible.

AUTH\_HMAC\_SHA1\_96 has been downgraded from MUST in [RFC4307](#) to MUST- as cryptographic attacks against SHA1 are increasing, resulting in an industry-wide trend to deprecate its usage

AUTH\_AES\_XCBC\_96 is only recommended in the scope of IoT, as Internet of Things deployments tend to prefer AES based pseudo-random functions in order to avoid implementing SHA2. For the non-IoT VPN



deployment, it has been downgraded from SHOULD in [RFC4307](#) to MAY as it has not been widely adopted.

AUTH\_DES\_MAC, AUTH\_HMAC\_MD5\_96, and AUTH\_KPDK\_MD5 were not mentioned in [RFC4307](#) so their default statuses were MAY. They have been downgraded to MUST NOT. There is an industry-wide trend to deprecate DES and MD5. MD5 support is being removed from cryptographic libraries in general because its non-HMAC use is known to be subject to collision attacks, for example as mentioned in [\[TRANSCRIPTION\]](#).

#### **[2.4.](#) Type 4 - IKEv2 Diffie-Hellman Group Transforms**

There are several Modular Exponential (MODP) groups and several Elliptic Curve groups (ECC) that are defined for use in IKEv2. These groups are defined in both the [\[RFC7296\]](#) base document and in extensions documents and are identified by group number. Note that it is critical to enforce a secure Diffie-Hellman exchange as this exchange provides keys for the session. If an attacker can retrieve one of the private numbers (a or b) and the complementary public value ( $g^{**b}$  or  $g^{**a}$ ), then the attacker can compute the secret and the keys used and decrypt the exchange and IPsec SA created inside the IKEv2 SA. Such an attack can be performed off-line on a previously recorded communication, years after the communication happened. This differs from attacks that need to be executed during the authentication which must be performed online and in near real-time.

| Number | Description   | Status     |
|--------|---|------------|
| 14     | 2048-bit MODP Group                                   | MUST       |
| 19     | 256-bit random ECP group                              | SHOULD     |
| 5      | 1536-bit MODP Group                                   | SHOULD NOT |
| 2      | 1024-bit MODP Group                                   | SHOULD NOT |
| 1      | 768-bit MODP Group                                    | MUST NOT   |
| 22     | 1024-bit MODP Group with 160-bit Prime Order Subgroup | MUST NOT   |
| 23     | 2048-bit MODP Group with 224-bit Prime Order Subgroup | SHOULD NOT |
| 24     | 2048-bit MODP Group with 256-bit Prime Order Subgroup | SHOULD NOT |

Group 14 or 2048-bit MODP Group is raised from SHOULD+ in [RFC4307](#) to MUST as a replacement for 1024-bit MODP Group. Group 14 is widely implemented and considered secure.



Group 19 or 256-bit random ECP group was not specified in [RFC4307](#), as this group was not defined at that time. Group 19 is widely implemented and considered secure and therefore has been promoted to the SHOULD level.

Group 5 or 1536-bit MODP Group has been downgraded from MAY in [RFC4307](#) to SHOULD NOT. It was specified earlier, but is now considered to be vulnerable to being broken within the next few years by a nation state level attack, so its security margin is considered too narrow.

Group 2 or 1024-bit MODP Group has been downgraded from MUST- in [RFC4307](#) to SHOULD NOT. It is known to be weak against sufficiently funded attackers using commercially available mass-computing resources, so its security margin is considered too narrow. It is expected in the near future to be downgraded to MUST NOT.

Group 1 or 768-bit MODP Group was not mentioned in [RFC4307](#) and so its status was MAY. It can be broken within hours using cheap off-the-shelves hardware. It provides no security whatsoever. It has therefore been downgraded to MUST NOT.

Group 22, 23 and 24 are MODP Groups with Prime Order Subgroups that are not safe-primes. The seeds for these groups have not been publicly released, resulting in reduced trust in these groups. These groups were proposed as alternatives for group 2 and 14 but never saw wide deployment. It has been shown that Group 22 with 1024-bit MODP is too weak and academia have the resources to generate malicious values at this size. This has resulted in Group 22 to be demoted to MUST NOT. Group 23 and 24 have been demoted to SHOULD NOT and are expected to be further downgraded in the near future to MUST NOT. Since Group 23 and 24 have small subgroups, the checks specified in "Additional Diffie-Hellman Test for the IKEv2" [\[RFC6989\] section 2.2](#) first bullet point MUST be done when these groups are used.

## **2.5. Summary of Changes from [RFC 4307](#)**

The following table summarizes the changes from [RFC 4307](#).

RFC EDITOR: PLEASE REMOVE THIS PARAGRAPH AND REPLACE XXXX IN THE TABLE BELOW WITH THE NUMBER OF THIS RFC



| Algorithm           | <a href="#">RFC 4307</a> | RFC XXXX   |
|---------------------|--------------------------|------------|
| ENCR_3DES           | MUST-                    | MAY        |
| ENCR_NULL           | MUST NOT[errata]         | MUST NOT   |
| ENCR_AES_CBC        | SHOULD+                  | MUST       |
| ENCR_AES_CTR        | SHOULD                   | (*)        |
| PRF_HMAC_MD5        | MAY                      | MUST NOT   |
| PRF_HMAC_SHA1       | MUST                     | MUST-      |
| PRF_AES128_XCBC     | SHOULD+                  | SHOULD     |
| AUTH_HMAC_MD5_96    | MAY                      | MUST NOT   |
| AUTH_HMAC_SHA1_96   | MUST                     | MUST-      |
| AUTH_AES_XCBC_96    | SHOULD+                  | SHOULD     |
| Group 2 (1024-bit)  | MUST-                    | SHOULD NOT |
| Group 14 (2048-bit) | SHOULD+                  | MUST       |

(\*) This algorithm is not mentioned in the above sections, so it defaults to MAY.

### 3. IKEv2 Authentication

IKEv2 authentication may involve a signatures verification. Signatures may be used to validate a certificate or to check the signature of the AUTH value. Cryptographic recommendations regarding certificate validation are out of scope of this document. What is mandatory to implement is provided by the PKIX Community. This document is mostly concerned with signature verification and generation for the authentication.

#### 3.1. IKEv2 Authentication Method

| Number | Description                           | Status     |
|--------|---------------------------------------|------------|
| 1      | RSA Digital Signature                 | MUST       |
| 2      | Shared Key Message Integrity Code     | MUST       |
| 3      | DSS Digital Signature                 | SHOULD NOT |
| 9      | ECDSA with SHA-256 on the P-256 curve | SHOULD     |
| 10     | ECDSA with SHA-384 on the P-384 curve | SHOULD     |
| 11     | ECDSA with SHA-512 on the P-521 curve | SHOULD     |
| 14     | Digital Signature                     | SHOULD     |

RSA Digital Signature is widely deployed and therefore kept for interoperability. It is expected to be downgraded in the future as its signatures are based on the older RSASSA-PKCS1-v1.5 which is no longer recommended. RSA authentication, as well as other specific



Authentication Methods, are expected to be replaced with the generic Digital Signature method of [[RFC7427](#)].

Shared Key Message Integrity Code is widely deployed and mandatory to implement in the IKEv2 in the [RFC7296](#). The status remains MUST.

ECDSA based Authentication Methods are also expected to be downgraded as these do not provide hash function agility. Instead, ECDSA (like RSA) is expected to be performed using the generic Digital Signature method. It's status is SHOULD.

DSS Digital Signature is bound to SHA-1 and has the same level of security as 1024-bit RSA. It is currently at SHOULD NOT and is expected to be downgraded to MUST NOT in the future.

Digital Signature [[RFC7427](#)] is expected to be promoted as it provides hash function, signature format and algorithm agility. Its current status is SHOULD.

### **[3.1.1](#). Recommendations for RSA key length**

| +-----+-----+                             |            |
|---|------------|
| Description                               | Status     |
| +-----+-----+                             |            |
| RSA with key length 2048                  | MUST       |
| RSA with key length 3072 and 4096         | SHOULD     |
| RSA with key length between 2049 and 4095 | MAY        |
| RSA with key length smaller than 2048     | SHOULD NOT |
| +-----+-----+                             |            |

The IKEv2 [RFC7296](#) mandates support for the RSA keys of size 1024 or 2048 bits, but key sizes less than 2048 are updated to SHOULD NOT as there is industry-wide trend to deprecate key lengths less than 2048 bits. Since these signatures only have value in real-time, and need no future protection, smaller keys were kept at SHOULD NOT instead of MUST NOT.

### **[3.2](#). Digital Signature Recommendations**

When a Digital Signature authentication method is implemented, the following recommendations are applied for hash functions:



| Number | Description | Status   | Comment |
|--------|-------------|----------|---------|
| 1      | SHA1        | MUST NOT |         |
| 2      | SHA2-256    | MUST     |         |
| 3      | SHA2-384    | MAY      |         |
| 4      | SHA2-512    | SHOULD   |         |

When the Digital Signature authentication method is used with RSA signature algorithm, RSASSA-PSS MUST be supported and RSASSA-PKCS1-v1.5 MAY be supported.

The following table lists recommendations for authentication methods in [RFC7427](#) [RFC7427] notation. These recommendations are applied only if Digital Signature authentication method is implemented.

| Description                        | Status   | Comment |
|------------------------------------|----------|---------|
| RSASSA-PSS with SHA-256            | MUST     |         |
| ecdsa-with-sha256                  | SHOULD   |         |
| sha1WithRSAEncryption              | MUST NOT |         |
| dsa-with-sha1                      | MUST NOT |         |
| ecdsa-with-sha1                    | MUST NOT |         |
| RSASSA-PSS with Empty Parameters   | MUST NOT | (*)     |
| RSASSA-PSS with Default Parameters | MUST NOT | (*)     |

(\*) Empty or Default parameters means it is using SHA1, which is at level MUST NOT.

#### 4. Algorithms for Internet of Things

Some algorithms in this document are marked for use with the Internet of Things (IoT). There are several reasons why IoT devices prefer a different set of algorithms from regular IKEv2 clients. IoT devices are usually very constrained, meaning the memory size and CPU power is so limited, that these clients only have resources to implement and run one set of algorithms. For example, instead of implementing AES and SHA, these devices typically use AES\_XCBC as integrity algorithm so SHA does not need to be implemented.

For example, IEEE Std 802.15.4 [IEEE-802-15-4] devices have a mandatory to implement link level security using AES-CCM with 128 bit keys. The IEEE Recommended Practice for Transport of Key Management Protocol (KMP) Datagrams [IEEE-802-15-9] already provide a way to use



Minimal IKEv2 [[RFC7815](#)] over 802.15.4 to provide link keys for the 802.15.4 layer.

These devices might want to use AES-CCM as their IKEv2 algorithm, so they can reuse the hardware implementing it. They cannot use the AES-CBC algorithm, as the hardware quite often do not include support for AES decryption needed to support the CBC mode. So despite the AES-CCM algorithm requiring AEAD [[RFC5282](#)] support, the benefit of reusing the crypto hardware makes AES-CCM the preferred algorithm.

Another important aspect of IoT devices is that their transfer rates are usually quite low (in order of tens of kbits/s), and each bit they transmit has an energy consumption cost associated with it and shortens their battery life. Therefore, shorter packets are preferred. This is the reason for recommending the 8 octet ICV over the 16 octet ICV.

Because different IoT devices will have different constraints, this document cannot specify the one mandatory profile for IoT. Instead, this document points out commonly used algorithms with IoT devices.

## **5. Security Considerations**

The security of cryptographic-based systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

The Diffie-Hellman Group parameter is the most important one to choose conservatively. Any party capturing all IKE and ESP traffic that (even years later) can break the selected DH group in IKE, can gain access to the symmetric keys used to encrypt all the ESP traffic. Therefore, these groups must be chosen very conservatively. However, specifying an extremely large DH group also puts a considerable load on the device, especially when this is a large VPN gateway or an IoT constrained device.

This document concerns itself with the selection of cryptographic algorithms for the use of IKEv2, specifically with the selection of "mandatory-to-implement" algorithms. The algorithms identified in this document as "MUST implement" or "SHOULD implement" are not known to be broken at the current time, and cryptographic research so far leads us to believe that they will likely remain secure into the foreseeable future. However, this isn't necessarily forever and it is expected that new revisions of this document will be issued from time to time to reflect the current best practice in this area.



## 6. IANA Considerations

This document renames some of the names in the "Transform Type 1 - Encryption Algorithm Transform IDs" registry of the "Internet Key Exchange Version 2 (IKEv2) Parameters". All the other names have ENCR\_ prefix except 3, and all other entries use names in format of uppercase words separated with underscores except 6. This document changes those names to match others.

This document requests IANA to rename following entries for the AES-GCM cipher [[RFC4106](#)] and the Camellia cipher [[RFC5529](#)]:

| +-----+-----+                         |                      |
|---------------------------------------|----------------------|
| Old name                              | New name             |
| +-----+-----+                         |                      |
| AES-GCM with a 8 octet ICV            | ENCR_AES_GCM_8       |
| AES-GCM with a 12 octet ICV           | ENCR_AES_GCM_12      |
| AES-GCM with a 16 octet ICV           | ENCR_AES_GCM_16      |
| ENCR_CAMELLIA_CCM with an 8-octet ICV | ENCR_CAMELLIA_CCM_8  |
| ENCR_CAMELLIA_CCM with a 12-octet ICV | ENCR_CAMELLIA_CCM_12 |
| ENCR_CAMELLIA_CCM with a 16-octet ICV | ENCR_CAMELLIA_CCM_16 |
| +-----+-----+                         |                      |

In addition to add this RFC as reference to both ESP Reference and IKEv2 Reference columns for ENCR\_AES\_GCM entries, keeping the current references there also, and also add this RFC as reference to the ESP Reference column for ENCR\_CAMELLIA\_CCM entries, keeping the current reference there also.

The final registry entries should be:

| Number | Name                 | ESP Reference                         | IKEv2 Reference                       |
|--------|----------------------|---------------------------------------|---------------------------------------|
| ...    |                      |                                       |                                       |
| 18     | ENCR_AES_GCM_8       | [ <a href="#">RFC4106</a> ] [RFCXXXX] | [ <a href="#">RFC5282</a> ] [RFCXXXX] |
| 19     | ENCR_AES_GCM_12      | [ <a href="#">RFC4106</a> ] [RFCXXXX] | [ <a href="#">RFC5282</a> ] [RFCXXXX] |
| 20     | ENCR_AES_GCM_16      | [ <a href="#">RFC4106</a> ] [RFCXXXX] | [ <a href="#">RFC5282</a> ] [RFCXXXX] |
| ...    |                      |                                       |                                       |
| 25     | ENCR_CAMELLIA_CCM_8  | [ <a href="#">RFC5529</a> ] [RFCXXXX] | -                                     |
| 26     | ENCR_CAMELLIA_CCM_12 | [ <a href="#">RFC5529</a> ] [RFCXXXX] | -                                     |
| 27     | ENCR_CAMELLIA_CCM_16 | [ <a href="#">RFC5529</a> ] [RFCXXXX] | -                                     |

## 7. Acknowledgements

The first version of this document was [RFC 4307](#) by Jeffrey I. Schiller of the Massachusetts Institute of Technology (MIT). Much of the original text has been copied verbatim.



We would like to thank Paul Hoffman, Yaron Sheffer, John Mattsson, Tommy Pauly, Eric Rescorla and Pete Resnick for their valuable feedback and reviews.

## **8. References**

### **8.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", [RFC 4106](#), DOI 10.17487/RFC4106, June 2005, <<http://www.rfc-editor.org/info/rfc4106>>.
- [RFC4307] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", [RFC 4307](#), DOI 10.17487/RFC4307, December 2005, <<http://www.rfc-editor.org/info/rfc4307>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, [RFC 7296](#), DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.
- [RFC5282] Black, D. and D. McGrew, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", [RFC 5282](#), DOI 10.17487/RFC5282, August 2008, <<http://www.rfc-editor.org/info/rfc5282>>.

### **8.2. Informative References**

- [RFC7427] Kivinen, T. and J. Snyder, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)", [RFC 7427](#), DOI 10.17487/RFC7427, January 2015, <<http://www.rfc-editor.org/info/rfc7427>>.
- [RFC6989] Sheffer, Y. and S. Fluhrer, "Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2)", [RFC 6989](#), DOI 10.17487/RFC6989, July 2013, <<http://www.rfc-editor.org/info/rfc6989>>.



- [RFC7815] Kivinen, T., "Minimal Internet Key Exchange Version 2 (IKEv2) Initiator Implementation", [RFC 7815](https://www.rfc-editor.org/info/rfc7815), DOI 10.17487/RFC7815, March 2016, <<http://www.rfc-editor.org/info/rfc7815>>.
- [RFC5529] Kato, A., Kanda, M., and S. Kanno, "Modes of Operation for Camellia for Use with IPsec", [RFC 5529](https://www.rfc-editor.org/info/rfc5529), DOI 10.17487/RFC5529, April 2009, <<http://www.rfc-editor.org/info/rfc5529>>.
- [IKEV2-IANA] "Internet Key Exchange Version 2 (IKEv2) Parameters", <<http://www.iana.org/assignments/ikev2-parameters>>.
- [TRANSCRIPTION] Bhargavan, K. and G. Leurent, "Transcript Collision Attacks: Breaking Authentication in TLS, IKE, and SSH", NDSS , feb 2016.
- [IEEE-802-15-4] "IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Standard 802.15.4, 2015.
- [IEEE-802-15-9] "IEEE Recommended Practice for Transport of Key Management Protocol (KMP) Datagrams", IEEE Standard 802.15.9, 2016.

#### Authors' Addresses

Yoav Nir  
Check Point Software Technologies Ltd.  
5 Hasolelim st.  
Tel Aviv 6789735  
Israel

EMail: [ynir.ietf@gmail.com](mailto:ynir.ietf@gmail.com)

Tero Kivinen  
INSIDE Secure  
Eerikinkatu 28  
HELSINKI FI-00180  
FI

EMail: [kivinen@iki.fi](mailto:kivinen@iki.fi)



Paul Wouters  
Red Hat

E-Mail: [pwouters@redhat.com](mailto:pwouters@redhat.com)

Daniel Migault  
Ericsson  
8400 boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Phone: +1 514-452-2160  
E-Mail: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

