Stream:	Internet Engineering Task Force (IETF)
RFC:	9963
Category:	Standards Track
Published:	April 2026
ISSN:	2070-1721
Authors:	D. Benjamin	A. Popov
---	---	---
	Google LLC	*Microsoft Corp.*

RFC 9963
Legacy RSASSA-PKCS1-v1_5 Code Points for TLS 1.3

Abstract

This document allocates code points for the use of RSASSA-PKCS1-v1_5 with client certificates in TLS 1.3. This removes an obstacle for some deployments to migrate to TLS 1.3.

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/rfc9963.

Copyright Notice

Table of Contents

1. Introduction

TLS 1.3 [RFC8446] removed support for RSASSA-PKCS1-v1_5 [RFC8017] in CertificateVerify messages in favor of RSASSA-PSS. While RSASSA-PSS is a long-established signature algorithm, some legacy hardware cryptographic devices lack support for it. While uncommon in TLS servers, these devices are sometimes used by TLS clients for client certificates.

For example, Trusted Platform Modules (TPMs) are ubiquitous hardware cryptographic devices that are often used to protect TLS client certificate private keys. However, a large number of TPMs are unable to produce RSASSA-PSS signatures compatible with TLS 1.3. TPM specifications prior to 2.0 did not define RSASSA-PSS support (see Section 5.8.1 of [TPM12]). TPM 2.0 includes RSASSA-PSS, but only those TPM 2.0 devices compatible with US FIPS 186-4 can be relied upon to use the salt length matching the digest length, as required for compatibility with TLS 1.3 (see Appendix B.7 of [TPM2]).

TLS connections that rely on such devices cannot migrate to TLS 1.3. Staying on TLS 1.2 leaks the client certificate to network attackers [PRIVACY] and additionally prevents such deployments from protecting traffic against retroactive decryption by an attacker with a quantum computer [RFC9954].

Additionally, TLS negotiates protocol versions before client certificates. Clients send ClientHellos without knowing whether the server will request to authenticate with legacy keys. Conversely, servers respond with a TLS version and CertificateRequest without knowing if the client will then respond with a legacy key. If the client and server, respectively, offer and negotiate TLS 1.3, the connection will fail due to the legacy key, when it previously succeeded at TLS 1.2.

To recover from this failure, one side must globally disable TLS 1.3 or the client must implement an external fallback. Disabling TLS 1.3 impacts connections that would otherwise be unaffected by this issue, while external fallbacks break TLS's security analysis and may introduce vulnerabilities [POODLE].

This document allocates code points to use these legacy keys with client certificates in TLS 1.3. This reduces the pressure on implementations to select one of these problematic mitigations and unblocks TLS 1.3 deployment.

2. Conventions and Definitions

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. PKCS #1 v1.5 SignatureScheme Types

The following SignatureScheme values are defined for use with TLS 1.3.

```
enum {
    rsa_pkcs1_sha256_legacy(0x0420),
    rsa_pkcs1_sha384_legacy(0x0520),
    rsa_pkcs1_sha512_legacy(0x0620),
} SignatureScheme;
```

The above code points indicate a signature algorithm using RSASSA-PKCS1-v1_5 [RFC8017] with the corresponding hash algorithm as defined in [SHS]. They are only defined for signatures in the client CertificateVerify message and are not defined for use in other contexts. In particular, servers that intend to advertise support for RSASSA-PKCS1-v1_5 signatures in the certificates themselves should use the `rsa_pkcs1_*` constants defined in [RFC8446].

Clients **MUST NOT** advertise these values in the `signature_algorithms` extension of the ClientHello. They **MUST NOT** accept these values in the server CertificateVerify message.

Servers that wish to support clients authenticating with legacy RSASSA-PKCS1-v1_5-only keys **MAY** send these values in the `signature_algorithms` extension of the CertificateRequest message and accept them in the client CertificateVerify message. Servers **MUST NOT** accept these code points if not offered in the CertificateRequest message.

Clients with such legacy keys **MAY** negotiate the use of these signature algorithms if offered by the server. Clients **SHOULD NOT** negotiate the use of these signature algorithms with keys that support RSASSA-PSS, though this may not be practical to determine in all applications. For example, attempting to test a key for support might result in a message to the user or have other side effects.

TLS implementations **SHOULD** disable these code points by default. See Section 4.

4. Security Considerations

The considerations in Section 1 do not apply to server keys, so these new code points are forbidden for use with server certificates. RSASSA-PSS continues to be required for TLS 1.3 servers using RSA keys. This minimizes the impact to only those cases in which it is necessary to unblock deployment of TLS 1.3.

When implemented incorrectly, RSASSA-PKCS1-v1_5 admits signature forgeries [MFSA201473]. Implementations producing or verifying signatures with these algorithms **MUST** implement RSASSA-PKCS1-v1_5 as specified in Section 8.2 of [RFC8017]. In particular, clients **MUST** include the mandatory NULL parameter in the DigestInfo structure and produce a valid DER [X690] encoding. Servers **MUST** reject signatures which do not meet these requirements.

5. IANA Considerations

IANA has created the following entries in the "TLS SignatureScheme" registry. The "Recommended" column has been set to "N", and the "Reference" column refers to this document.

Value	Description
0x0420	`rsa_pkcs1_sha256_legacy`
0x0520	`rsa_pkcs1_sha384_legacy`
0x0620	`rsa_pkcs1_sha512_legacy`

Table 1

6. References

6.1. 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>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[SHS] NIST, "Secure Hash Standard", NIST FIPS 180-4, DOI 10.6028/NIST.FIPS.180-4, August 2015, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

[TPM12] Trusted Computing Group, "TPM Main, Part 2 - Structures of the TPM", Level 2, Version 1.2, Revision 116, 1 March 2011, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf>.

[TPM2] Trusted Computing Group, "Trusted Platform Module Library, Part 1: Architecture", Family 2.0, Level 00, Revision 01.59, 8 November 2019, <https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM2_r1p59_Part1_Architecture_pub.pdf>.

[X690] ITU-T, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.690>.

6.2. Informative References

[MFSA201473] Delignat-Lavaud, A., "Mozilla Foundation Security Advisory 2014-73: RSA Signature Forgery in NSS", 24 September 2014, <https://www.mozilla.org/en-US/security/advisories/mfsa2014-73/>.

[POODLE] Moeller, B., "This POODLE bites: exploiting the SSL 3.0 fallback", Google Security Blog, 14 October 2014, <https://security.googleblog.com/2014/10/this-poodle-bites-exploiting-ssl-30.html>.

[PRIVACY] Wachs, M., Scheitle, Q., and G. Carle, "Push away your privacy: Precise user tracking based on TLS client certificate authentication", 2017 Network Traffic Measurement and Analysis Conference (TMA). pp. 1-9, DOI 10.23919/tma.2017.8002897, June 2017, <https://doi.org/10.23919/tma.2017.8002897>.

[RFC9954] Stebila, D., Fluhrer, S., and S. Gueron, "Hybrid Key Exchange in TLS 1.3", RFC 9954, DOI 10.17487/RFC9954, April 2026, <https://www.rfc-editor.org/info/rfc9954>.

Acknowledgements

Thanks to Rifaat Shekh-Yusef, Martin Thomson, and Paul Wouters for providing feedback on this document.

Authors' Addresses

David Benjamin
Google LLC
Email: davidben@google.com

Andrei Popov
Microsoft Corp.
Email: andreipo@microsoft.com