

Independent Submission  
Request for Comments: 8351  
Category: Informational  
ISSN: 2070-1721

S. Leonard  
Penango, Inc.  
June 2018

## The PKCS #8 EncryptedPrivateKeyInfo Media Type

### Abstract

This document registers the application/pkcs8-encrypted media type for the EncryptedPrivateKeyInfo type of PKCS #8. An instance of this media type carries a single encrypted private key, BER-encoded as a single EncryptedPrivateKeyInfo value.

### Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see 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/rfc8351>.

### Copyright Notice

Copyright (c) 2018 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.

## Table of Contents

|                                       |   |
|---------------------------------------|---|
| 1. Introduction . . . . .             | 2 |
| 2. Registration Application . . . . . | 2 |
| 3. IANA Considerations . . . . .      | 6 |
| 4. Security Considerations . . . . .  | 6 |
| 5. Normative References . . . . .     | 6 |
| Author's Address . . . . .            | 7 |

## 1. Introduction

The private key is encrypted with an encryption algorithm, which could be a password-based encryption scheme as that term is used in PKCS #5: Password-Based Cryptography Specification Version 2.1 as published in [RFC2898] and updated by [RFC8018]. This document registers the application/pkcs8-encrypted media type for the EncryptedPrivateKeyInfo type of PKCS #8 (as originally described in [RFC5208], which was obsoleted by [RFC5958]). An instance of this media type carries a single encrypted private key [RFC5958] BER-encoded as a single EncryptedPrivateKeyInfo value.

## 2. Registration Application

Type name: application

Subtype name: pkcs8-encrypted

Required parameters: None.

Optional parameters:

password-mapping: The private key is encrypted with an encryption algorithm, which could be a password-based encryption scheme as that term is used in PKCS #5 ([RFC2898] and [RFC8018]). Such algorithms take a password as input. A "password" is a secret text value (see Section 3 of [RFC2898] and [RFC8018]), but for algorithmic purposes the term "password" refers to an octet string (see Section 2 of [RFC2898] and [RFC8018]). Therefore, there must be some mapping between the text value (which might be user input) and the octet string. Section 3 of [RFC2898] (which was replaced by [RFC8018]) recommends "that applications follow some common text encoding rules"; it then offers, but does not recommend, ASCII and UTF-8.

While many modern applications support Unicode and Unicode-based encodings such as UTF-8 and UTF-16, interchange is still needed with private key artifacts that are encrypted with passwords in other encodings. Therefore, this parameter specifies the charset (see Section 1.3 of [RFC2978]) that a recipient should attempt first, in "reverse", when mapping from a sequence of characters to an octet string. This parameter is not cryptographically protected, so recipients cannot rely on it as the exclusive mapping possibility.

This parameter has similar semantics to the charset parameter from text/plain, except that it only applies to the user's input (text value) of a password. There is no default value.

The following special values, which all begin with "\*" to distinguish them from registered charsets, are defined:

- \*pkcs12 UTF-16LE with U+0000 NULL terminator: PKCS #12 style, see [RFC7292].
- \*precis Preparation, Enforcement, and Comparison of Internationalized Strings (PRECIS) password profile, i.e., OpaqueString from Section 4 of [RFC7613], which was obsoleted by [RFC8265]: always UTF-8 in Normalization Form C (NFC).
- \*precis-XXX Any profile from the IANA "PRECIS Profiles" registry where "XXX" is replaced by the profile name as shown in the registry.
- \*hex hexadecimal input: the input is mapped to 0-9, A-F, and then converted directly to octets. If there are an odd number of hex digits, either the final digit 0 is appended or an error condition is raised. Compare with Annex M.4 of [IEEE.802.11-2012].
- \*dtmf The characters "0"- "9", "A"- "D", "\*", and "#", which map to their corresponding ASCII codes. "A"- "D" map to the uppercase range 0x41 - 0x44. (This is to support restricted-input devices, i.e., telephones and telephone-like equipment.) User input outside of these values is either ignored or an error condition is raised.

Otherwise, the value of this parameter is a charset, from the IANA "Character Sets" registry [CHARREG].

This parameter is case insensitive.

Encoding considerations: Binary.

Security considerations:

Carries a cryptographic private key. See Section 6 of [RFC5958].

EncryptedPrivateKeyInfo PKCS #8 data contains exactly one private key. Poor password choices, weak algorithms, or improper parameter selections (e.g., insufficient salting rounds) will make the confidential payloads much easier to compromise.

Interoperability considerations:

PKCS #8 is a widely recognized format for private key information on all modern cryptographic stacks. The contents are exactly one private key (with optional key attributes), so there is no possibility for hidden "Easter eggs" in the payload such as unexpected certificates or miscellaneous secrets.

The encrypted variation in this registration, EncryptedPrivateKeyInfo (Section 3, "Encrypted Private Key Info", of [RFC5958], and Section 6 of PKCS #8 as originally described in [RFC5208], which was obsoleted by [RFC5958]), is less widely used for exchange than PKCS #12, but it is much simpler to implement. Actually, PKCS #12 incorporates the PKCS #8 types, so a PKCS #12 processor ought to be able to process PKCS #8 data by embedding the PKCS #8 data in PKCS #12 "scaffolding".

The password-mapping parameter aids in interoperability when the creator (who encrypted the keying material) and the user (who is attempting to decrypt the keying material) are not operating in the same character-encoding environment. An anticipated scenario is that the creator may have created the keying material with a password in a Shift-JIS environment a long time ago, while the user is in a UTF-8 environment. There are potentially many Unicode sequences that code for the same abstract character, such as precomposed and decomposed forms; yet, such an abstract character (however coded in Unicode) will tend to map to one coding in the legacy charset, if it can be represented at all. Therefore, the password-mapping parameter will almost never be ambiguous when mapping to legacy encodings. When mapping from one Unicode form to another (such as an internal Unicode representation to \*pkcs12), code sequences are either preserved or

folded deterministically to common Unicode code points or sequences, producing the same holistic result as mapping to legacy encodings.

It is possible that an abstract character might map to multiple legacy encodings under the same charset. However, the possibility is sufficiently remote as to be ignored in this media type registration. One possible workaround is to set the user's (decrypting party's) local operating environment to the password-mapping legacy encoding parameter for the purpose of generating the password octet string from user input. Another possibility is to generate all possible legacy encoding combinations from the abstract text (i.e., Unicode text), attempting decryption with them. Customized behavior can be defined by updating this media type registration with a new password-mapping special value, prefixed with \*.

Published specification:

RSA Laboratories PKCS #8 v1.2 RSA Encryption Standard, November 1993 (republished as [RFC5208], May 2008, and updated as [RFC5958], August 2010); RFC 5958, August 2010

Applications that use this media type:

Machines, applications, browsers, Internet kiosks, and so on, that support this standard allow a user to import, export, and exercise a single private key.

Fragment identifier considerations: None.

Additional information:

Deprecated alias names for this type: N/A  
Magic number(s): None.  
File extension(s): .p8e  
Macintosh file type code(s): None. A uniform type identifier (UTI) of "com.rsa.pkcs-8-encrypted" is recommended.

Object Identifiers: 1.2.840.113549.1.12.10.1.2 (when in PKCS #12)

Person & email address to contact for further information:

Sean Leonard <dev+ietf@seantek.com>

Intended usage: COMMON

Restrictions on usage: None.

Author/Change controller: Sean Leonard <dev+ietf@seantek.com>

Provisional registration? No

### 3. IANA Considerations

IANA has registered the media type application/pkcs8-encrypted in the Standards tree using the information provided in Section 2 of this document.

### 4. Security Considerations

See the registration template.

### 5. Normative References

[CHARREG] IANA, "Character Sets", December 2013,  
<<http://www.iana.org/assignments/character-sets>>.

[IEEE.802.11-2012]  
IEEE, "IEEE Standard for Information technology--  
Telecommunications and information exchange between  
systems Local and metropolitan area networks--Specific  
requirements Part 11: Wireless LAN Medium Access Control  
(MAC) and Physical Layer (PHY) Specifications",  
IEEE 802.11-2012, DOI 10.1109/ieeestd.2012.6178212,  
<<http://ieeexplore.ieee.org/document/6178212/>>.

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography  
Specification Version 2.0", RFC 2898,  
DOI 10.17487/RFC2898, September 2000,  
<<https://www.rfc-editor.org/info/rfc2898>>.

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration  
Procedures", BCP 19, RFC 2978, DOI 10.17487/RFC2978,  
October 2000, <<https://www.rfc-editor.org/info/rfc2978>>.

[RFC5208] Kaliski, B., "Public-Key Cryptography Standards (PKCS) #8:  
Private-Key Information Syntax Specification Version 1.2",  
RFC 5208, DOI 10.17487/RFC5208, May 2008,  
<<https://www.rfc-editor.org/info/rfc5208>>.

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958,  
DOI 10.17487/RFC5958, August 2010,  
<<https://www.rfc-editor.org/info/rfc5958>>.

- [RFC7292] Moriarty, K., Ed., Nystrom, M., Parkinson, S., Rusch, A., and M. Scott, "PKCS #12: Personal Information Exchange Syntax v1.1", RFC 7292, DOI 10.17487/RFC7292, July 2014, <<https://www.rfc-editor.org/info/rfc7292>>.
- [RFC7613] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 7613, DOI 10.17487/RFC7613, August 2015, <<https://www.rfc-editor.org/info/rfc7613>>.
- [RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <<https://www.rfc-editor.org/info/rfc8018>>.
- [RFC8265] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 8265, DOI 10.17487/RFC8265, October 2017, <<https://www.rfc-editor.org/info/rfc8265>>.

#### Author's Address

Sean Leonard  
Penango, Inc.  
5900 Wilshire Blvd  
Ste 2600  
Los Angeles, CA 90036  
United States of America

Email: [dev+ietf@seantek.com](mailto:dev+ietf@seantek.com)  
URI: <http://www.penango.com/>

