

JavaScript による **End-to-End** セキュリティ 標準規格とセキュリティエンジニアリング

栗原 淳

December 12, 2019

はじめに

はじめに

この資料は、「JavaScript による End-to-End セキュリティ」の補足資料である。

「標準規格」そのもの、および一連の勉強会にて話題に上げた鍵フォーマットや標準アルゴリズムについてを解説を与える。
また、鍵フォーマットなどは、JavaScript でハンドリングするためのサンプルコードも記述している。

セキュリティ関連の標準規格

PKCS (Public Key Cryptography Standards)

PKCS とは？

RSA Security 社¹の研究部門 RSA Labs が策定・公開している、公開鍵暗号を中心とする一連のセキュリティ技術標準のこと。比較的古い、枯れた標準になる。

- #1,...#15 の 15 本が存在²。
- 暗号化・署名生成の手続き・アルゴリズム、データフォーマットの規定など、いわゆる「ローレベル」の技術標準が中心。
- RSA 暗号標準を定める#1 を代表に、重要なものはメンテされ続けている。が、他の新標準規格で代替されるものなどは破棄・管理移譲されている。

¹ RSA 暗号を作った Rivest-Shamir-Adleman の会社。

² 破棄・廃盤も含む。

PKCS は、その名目上「私企業」が策定した技術標準。
しかし、採用された技術は、十分にセキュリティ評価されていると見做されるものが多く、また他の技術標準に採用・移植・継承されている。特に多くは IETF RFC の管理下へ継承されているようだ。

PKCS 文書一覧 (1/2)

	Ver.	名称	内容
PKCS #1	v2.2	RSA Cryptography Specifications ³	RSA 鍵ペアの構造、暗号化手法、署名手法を策定。
PKCS #3	v1.4	Diffie-Hellman Key Agreement Standard	Diffie-Hellman 鍵交換の仕様を策定。RFC では Internet Key Exchange (IKE) へ継承 (?)。
PKCS #5	v2.1	Password-Based Cryptography Specification ⁴	パスワードからの鍵導出手法、暗号化手法 (PBKDF1/2, PBES1/2) の策定。
PKCS #6	廃止	Extended-Certificate Syntax Standard	X.509v1 証明書の拡張。X.509v3 へ統合されて廃止。
PKCS #7	廃止 (?)	Cryptographic Message Syntax Standard ⁵	暗号メッセージ構文を策定。S/MIME に利用。より新しい仕様 (RFC5652) により廃止 (?)。
PKCS #8	廃止 (?)	Private-Key Information Syntax Specification ⁶	秘密鍵フォーマットを策定。より新しい仕様 (RFC5968) により v1.2 で廃止 (?)。
PKCS #9	v2.0	Selected Object Classes and Attribute Types ⁷	各種フォーマットにおける「属性」タイプを策定。

³<https://tools.ietf.org/html/rfc8017>

⁴<https://tools.ietf.org/html/rfc8018>

⁵<https://tools.ietf.org/html/rfc2315>

⁶<https://tools.ietf.org/html/rfc5208>

⁷<https://tools.ietf.org/html/rfc2985>

PKCS 文書一覧 (2/2)

	Ver.	名称	内容
PKCS #10	v1.7	Certification Request Syntax Specification ⁸	証明書リクエスト構文を策定。元は PKCS のみで策定されていたが、利用されるメディアタイプを RFC5967 で拡張。
PKCS #11	v2.40	Cryptographic Token Interface	Cryptoki としても知られる、暗号トークン (H/W セキュリティモジュール) インターフェースの仕様を策定。OASIS PKCS 11 Technical Committee へ継承。
PKCS #12	v1.1	Personal Information Exchange Syntax Standard ⁹	パスワード暗号化された秘密鍵、公開鍵証明書の構文を策定。IETF IESG 管理下へ継承。
PKCS #15	v1.1	Cryptographic Token Information Format Standard	暗号トークン向け、ユーザ特定標準仕様の策定。IC カード部分は ISO/IEC 7816-15 へ移譲。

策定中のまま立ち消えたものなどは削除。

IETF RFC などへ Republication、あるいは継承されて新しい標準になっている。

⁸<https://tools.ietf.org/html/rfc2986> + <https://tools.ietf.org/html/rfc5967>

⁹<https://tools.ietf.org/html/rfc7292>

IETF RFC (Request for Comments)

その他; 各国の推奨技術リストとしての標準規格

■ CRYPTOREC¹⁰

電子政府推奨暗号リストを作り、その実装や運用方法も含めて安全性を調査・評価・監視・検討するプロジェクト (2000 年～)

■ NESSIE¹¹

EU の制定した暗号標準リストを策定するプロジェクト (2000 年～)

基本的には、「評価検討した結果、既存の暗号アルゴリズム・プロトコルのどれそれを標準として採用する」という「**推奨技術リスト**」の策定プロジェクトだと思って差し支えない。

¹⁰ Cryptography Research and Evaluation Committee

¹¹ New European Schemes for Signature, Integrity, and Encryption

「各国独自」という意味

推奨技術リストへ採用されたアルゴリズム・プロトコルは、「その国において正しく評価された比較的安全なもの」というお墨付きを得る。
セキュリティ技術は国防上重要な意味を持つため、このお墨付きは、その技術を自国で利用して良いものかどうかを判定するもの、と言える。

仕様の詳細は NIST 公募 (NIST FIPS PUB), あるいは IETF (RFC) など国際的に比較的オープンな場でまず評価・採用・策定される¹²。

その後、各国が独自に調査検討して推奨技術リストとして採用する、というケースが多い。

¹²例外は存在する。元々 PKCS は RSA Labs. の独自標準を公開したものだったが、IETF の公開の場で Internet Draft の形で標準化場されてきている。

公開鍵・秘密鍵のフォーマット・エンコーディング

- PEM
- DER
- JWK
- [ECC 鍵のみ] RAW (Octet form)

JS の基本は JWK。OpenSSL や SSH などでは馴染みがあるのは PEM。

ECDH, ECDSA の鍵選択 (曲線の選択)

どの曲線を使えばいいのか？

- P-256,
- P-256K, (Bitcoin で利用)
- P-384,
- P-521

あたりが無難なところ。