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 24, 2019

はじめに

はじめに

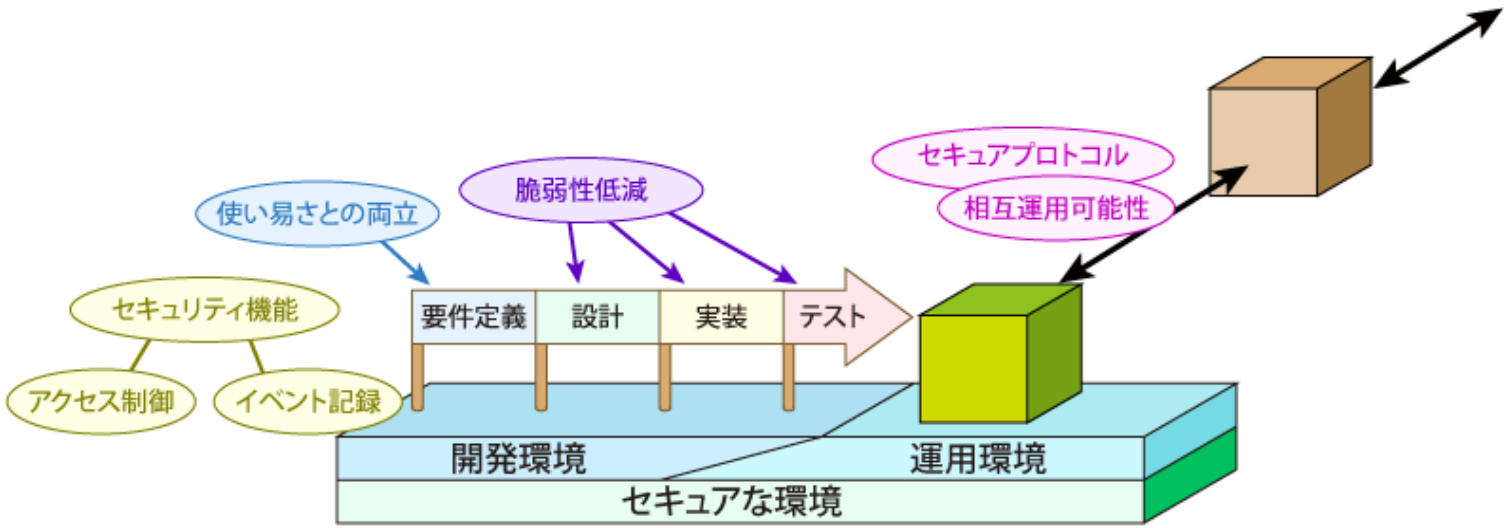
この資料は、「JavaScript による End-to-End セキュリティ」の補足資料である。
「標準規格」そのもの、および一連の勉強会にて話題に上げた鍵フォーマットや標準アルゴリズムについてを解説を与える。
また、鍵フォーマットなどは、JavaScript でハンドリングするためのサンプルコードも記述している。

セキュリティ関連の標準規格

はじめに： セキュリティエンジニアリングと標準規格

セキュリティエンジニアリング[1]

ソフトウェア・アプリケーション開発において、セキュリティを考慮したエンジニアリング、あるいはセキュリティエンジニアリングを行うためには、要件定義・設計・実装・テストの段階ごとに種々のセキュリティ関連事項を検討する必要がある。



[1] https://www.ipa.go.jp/security/awareness/vendor/software.html

一方で、標準規格は:

標準規格となるアルゴリズム・プロトコル

国家や団体の標準文書に載せるために、安全性・相互接続性・効率性を分析し、安全性や将来性などがある程度確保されたもの、と言える。

⇓

すなわち、標準規格を要件に応じて適切に選択し、設計・実装を行うことで脆弱性低減、使いやすさとの両立や相互運用可能性の担保が容易になると言える。

つまり、最新の標準規格とその推奨される利用方法[2]を把握しておくことで、効率的なセキュリティエンジニアリングが行える。

[2]なぜそのように利用するのか・載っているのか、も正しく知っておく必要がある。悪い例は、脆弱性はあるが互換性のために残っている RFC8017 の RSAES-PKCS1-v1_5。

次ページからは、実際のセキュリティ関連標準について紹介する。

PKCS (Public Key Cryptography Standards)

PKCS とは？

RSA Security 社[3] 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[4]。
- 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。

[3] RSA 暗号を作った Rivest-Shamir-Adleman の会社。

[4] 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[5]	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[6]	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[7]	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[8]	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[9]	各種フォーマットにおける「属性」タイプを策定。

[5] https://tools.ietf.org/html/rfc8017
[6] https://tools.ietf.org/html/rfc8018
[7] https://tools.ietf.org/html/rfc2315
[8] https://tools.ietf.org/html/rfc5208
[9] 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[10]	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[11]	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。

[10] https://tools.ietf.org/html/rfc2986 + https://tools.ietf.org/html/rfc5967
[11] https://tools.ietf.org/html/rfc7292

NIST FIPS および SP800[12]

NIST FIPS/SP800 とは？

米国国立標準技術研究所 (NIST; National Institute of Standards and Technology) の発行する文書のこと。

- FIPS; Federal Information Processing Standards: 米国商務長官の承認の下、NIST が公布した情報セキュリティ関連の米国の標準規格文書。詳細な基準や要求事項、ガイドラインが記載されている。
- SP800; Special Publication: 米国政府がセキュリティ対策を実施する際に参考とすることを前提とした、コンピュータセキュリティ関係のレポート。

[12] 参考: https://www.ipa.go.jp/security/publications/nist/

すなわち NIST FIPS は、米国ローカルの標準規格と言える。

- 多くは、他の国別標準規格同様に ANSI/ISO/IEEE 等で広く使われていた既存規格を引き継ぐ。
- 一部は NIST FIPS 独自に公募・評価・策定した独自規格。代表的なものは、公募されてきた ‘Rijndael’ という新暗号アルゴリズムを採用した FIPS 197; Advanced Encryption Standard (AES)。

IETF RFC

RFC (Request for Comments) とは？

「インターネット技術」全般の国際標準を議論策定するグループ IETF (Internet Engineering Task Force) で議論策定された、「インターネット技術標準」および「その他」[13] の広範な内容を扱う文書 (群) のこと。

詳細仕様を策定する ITU-T や ISO と異なり、「まずは動作させる」ことを目的として実験的な「Rough」な仕様をまず策定することが特徴。

[13] https://www.nic.ad.jp/ja/rfc-jp/RFC-Category.html

RFC は 5 つのカテゴリに分類される:

- Standards Track: Proposed Standard → Internet Standard という策定過程を経る「インターネット標準技術仕様」の文書。
- Informational: すでにデファクト標準であったり、インターネット標準の議論・策定において有益として公開されたもの。例えば、RSA セキュリティ社の PKCS#1 v2.1 = RFC8017。
- Experimental: デファクト標準を狙うような、研究等の目的で公開される技術仕様文書。
- Historical: 過去の記録として残す情報としての文書。
- Best Current Practice: 現状のベストプラクティスをまとめた仕様文書。

特に Standard Track, Informational, Experimental に関して、PKCS 等の他標準を引き継いだり、新たな技術標準を定めた文書が策定される。

セキュリティ関係の RFC 化の事例：

- 事例 1: OpenID Connect によって利用される鍵や署名、暗号化の仕様: JWS[14], JWE[15], JWK[16], JWT[17] について、OpenID Foundation のメンバにより、RFC Standards Track として国際標準化。
- 事例 2: PKCS#1, #5, #9 等の RSA セキュリティ社の独自標準は、Informational として RFC 化。
- 事例 3: HTTPS を支える TLS v1.3[18] は、Standards Track として RFC 化。

[14] JSON Web Signature https://tools.ietf.org/html/rfc7515
[15] JSON Web Encryption https://tools.ietf.org/html/rfc7516
[16] JSON Web Key https://tools.ietf.org/html/rfc7517
[17] JSON Web Token https://tools.ietf.org/html/rfc7519
[18] https://tools.ietf.org/html/rfc8446

ISO (International Organization for Standardization)

W3C (World Wide Web Consortium)

W3C とは？

W3C のセキュリティ関連 WG (Working Group) で有名な活動として、以下のような国際標準化策定が上げられる。

- WebCrypto WG[19]: WebCrypto API を策定、勧告として国際標準化。
- WebAuthn WG[20]: FIDO アライアンスの技術仕様を勧告として国際標準化 [21]

[19] https://www.w3.org/2012/webcrypto/ 現状は Close。

[20] https://www.w3.org/blog/webauthn/

[21] 対象はブラウザ・端末・認証サーバの連携プロトコルである FIDO2 WebAuthn https://www.w3.org/2019/03/pressrelease-webauthn-rec.html.ja。デバイス連携プロトコルである FIDO2 CTAP は ITU-T で国際標準化。

ITU-T SG17

ITU-T (International Telecommunication Union Telecommunication Standardization Sector) SG17 (Study Group 17) とは？[22]

- ITU-T: ITU (International Telecom. Union; 国際電気通信連合) における通信分野の標準技術を策定する「電気通信標準化部門」。策定された標準は「勧告」として発行される。
- Study Group 17: ITU-T においてセキュリティ関連勧告作成の中心となるグループ。

[22] https://www.ituaj.jp/wp-content/uploads/2016/07/2016_08-06-spotITU-T.pdf

ITU-T SG17 で取り扱う技術は、SDN・IoT・ITS・クラウドのセキュリティ技術、SPAM 対策、ID 管理技術、認証技術、テレバイオメトリクスなど、ソフトウェア実装のためのアルゴリズムというより「通信事業者」や「通信端末」を対象とした分野の技術。

昨今だと、「FIDO アライアンスの技術仕様を勧告として国際標準化[23]」している。

[23] 対象はデバイス連携プロトコルである FIDO UAF 1.1 および CTAP https://fidoalliance.org/fido-alliance-specifications-now-adopted-as-itu-international-standards/。Web 関連プロトコルである FIDO2 WebAuthn は W3C で国際標準化。

その他; 各国の推奨技術リストとしての標準規格

- CRYPTOREC[24]
  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〜)
- NESSIE[25]
  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。

[24] Cryptography Research and Evaluation Committee
[25] 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 IETF (RFC), ISO, NIST 公募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[26]。

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。

[26]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 の形で標準化されてきている。

その他; 諸々

- FIDO Alliance: フォーラム標準を定める業界団体。国際標準ではない。
- OpenID Foundation

公開鍵・秘密鍵・証明書のフォーマット・エンコード

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

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