

JavaScript による **End-to-End** セキュリティ 03

公開鍵暗号はどうやって使えばいいのか？ 編

栗原 淳

2019 年 10 月 3 日

はじめに

はじめに

前回・前々回 (第 1,2 回) では

- End-to-End (E2E) セキュリティの原則と必要性
- Web サイトでの E2E セキュリティ実践のため、JavaScript での暗号 (AES) の正しく・安全に利用する方法

を勉強した。

ところで、AES(共通鍵暗号) とは別に、「公開鍵暗号」というのが存在する。

今回は正しく・安全に公開鍵暗号を使っていくためのお話。

この講義で最終的に学びたいこと

- 公開鍵暗号はどのようなものか。AES と比べた pros/cons。
- RSA 暗号と楕円曲線暗号¹の違い。
- AES と公開鍵暗号を組み合わせデータ暗号化するために。

細かい所の話もしますが、なるべく数式を使わないで「イメージ」をつかめるようにする。

¹楕円曲線 Diffie-Hellman

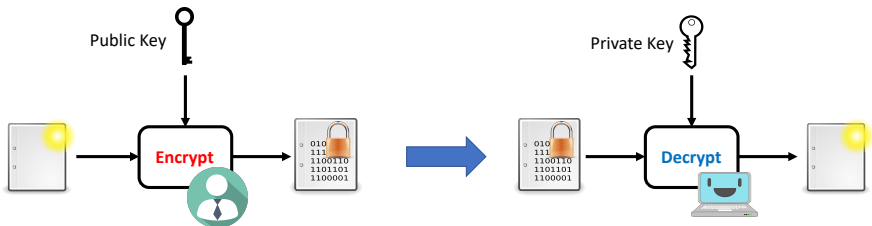
公開鍵暗号って？

既知だと思うが、まずざっと定義しておく。

定義: 公開鍵暗号

以下のステップで暗号化・復号が行われる暗号方式のこと

- 1 特殊な数学的条件を満たす鍵ペア「公開鍵 PK と 秘密鍵 SK 」を生成
- 2 PK は公開、 SK は秘匿
- 3 データ D を PK によって暗号化して、暗号化データ X を生成
- 4 X は SK によってデータ D に復号される。



Anyone can encrypt information only with public key without knowing private key!

Only the private key owner can decrypt the information encrypted under its paired public key.



暗号化・復号の鍵を分けて、暗号化の鍵を公開してしまうことでパスワードなどの共有が不要になる。

この講義の対象と事前準備

対象:

- 暗号・セキュリティ技術に興味がある初学者
- Web に暗号技術を導入したい Web 系のエンジニア

必須ではないが触って楽しむのには必要な事前準備:

- Git が使えるようになっていること
- Node.js が使えるようになっていること
- Google Chrome 系ブラウザ and/or Firefox が利用可能なこと

公開鍵暗号の使い方 事始め

公開鍵暗号の種類

公開鍵暗号の定義「特殊な数学的条件を満たす鍵ペアを生成」



この「数学的条件」に複数の種類が存在。

JavaScriptに限らず、各種環境で利用可能な代表的な公開鍵暗号：

- 素因数分解に関する条件
→ **RSA 暗号**
- 楕円曲線上の離散対数に関する条件
→ **楕円曲線暗号 (Elliptic Curve Diffie-Hellman; ECDH)**

この2つの使い方、注意ポイントを今回は取り上げる。

RSA 暗号のさわり

楕円曲線暗号 (ECDH) のさわり

AES と比べた公開鍵暗号の Pros/Cons

RSA 暗号を使ってみよう

RSA 暗号って何？

RSA 暗号を使うためのお作法

Padding:

RSA OAEP <https://tools.ietf.org/html/rfc8017>

PKCS1-v1.5 padding はやばい。Cryptrec から落ちた。

<https://www.cryptrec.go.jp/method.html>

<https://tools.ietf.org/html/rfc8017> 既知の攻撃が知られており、

RSASSA, RSAES 共に PKCF1-v1.5 は非推奨@RFC

でも Node.js では OAEP 非サポートなので作った

RSA-OAEP によるデータ暗号化を試みよう

楕円曲線暗号 (ECDH) を使ってみよう

ECDHって何？

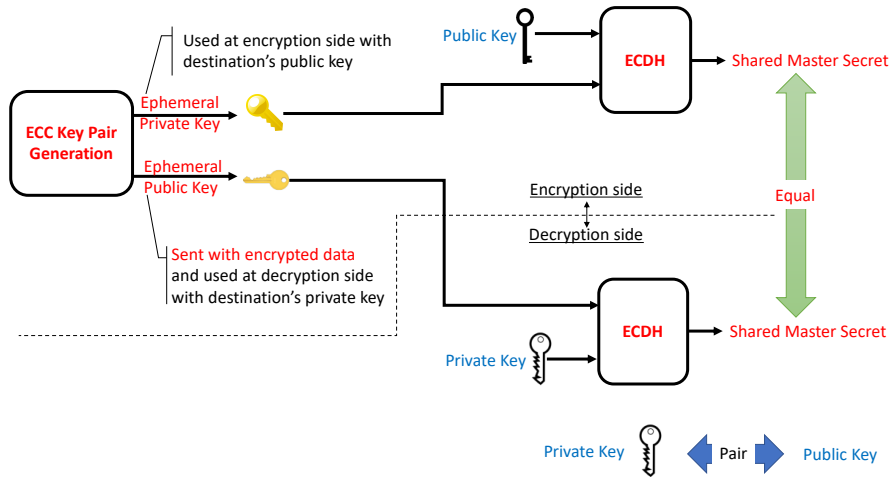
TLS <https://tools.ietf.org/html/rfc8422>

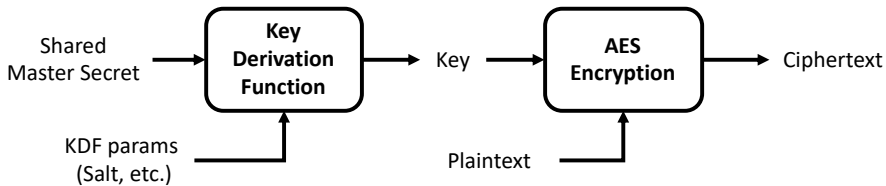
JOSE だと Concat KDF を使うだけ。

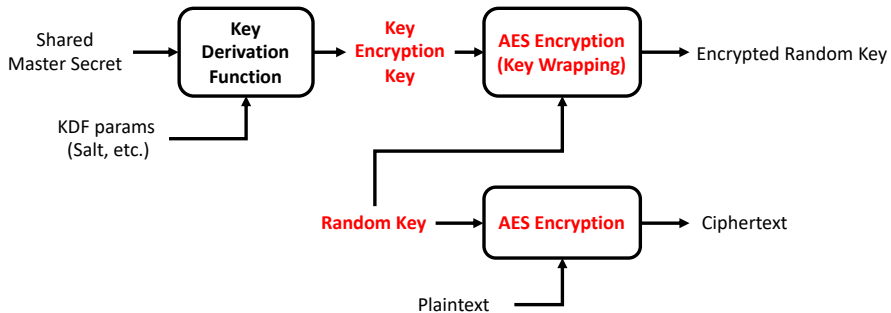
<https://tools.ietf.org/html/rfc8037>

直接 Concat KDF を暗号化の鍵にするか、あるいは Concat KDF の結果を AESKW の鍵として Content Encryption Key を暗号化するのに使う。

ECDH Ephemeral (ECDHE)







ECDH によるデータ暗号化を試みよう

今回は HKDF で暗号化してみる。

AES と公開鍵暗号のいいとこ取り

公開鍵暗号と **AES** の比較

ハイブリッド暗号化

AES の暗号化データをガンガン使い回せる！
msgpack-light を使ったコードを提供

まとめ

まとめ

お疲れ様でした。

- 公開鍵暗号を利用する際のお作法を学んだ。

次回以降…リクエスト次第ですが、

- 「情報が改ざんされてない」ことを保証するために（電子署名と MAC）
- RFC とアルゴリズム・フォーマット

などを予定。