

JavaScript による **End-to-End** セキュリティ

AES はどうやって使えばいいのか？ (共通鍵暗号の使い方) 編

栗原 淳

August 22, 2019

はじめに

はじめに

前回 (第 1 回) 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) の利用のさわり

を勉強した。

E2E セキュリティの重要性はわかった。
AES を使ってみることもできた。

でも、実際の App で **正しく・安全に AES を使うにはどうすべきなのか？**

今回は正しく・安全にAES を試してみる方法、についてのお話。

この講義で最終的に学びたいこと

- パスワードを使って AES を暗号化するのってどうすればいいか？¹
- バイナリ鍵を使って AES を暗号するにはどうすればいいか？²

「たったこれだけ」のことで、気をつけなければならない「**重要なお作法**」がある。

¹ RFC8018 PBES2 <https://tools.ietf.org/html/rfc8018> による AES 暗号化

² RFC5869 HKDF <https://tools.ietf.org/html/rfc5869> による鍵導出と AES 暗号化

お作法を守る・守らないで安全性は大違いなので、注意しなければならない。³

³世の中のソフトウェアのコード見ると、全くお作法を守ってないのがあって…本当危険…

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

AES の使い方 事始め

AES を使う際に気をつけるお作法は、ざっと 3 点。

- 1 AES で使う鍵のランダム具合
- 2 AES で使う鍵を総当りする際の大変さ⁴
- 3 AES の利用モードの安全性

つまりどういうこと？

⁴ 1 点目と 2 点目は似ているようで異なる。

1: AES で使う鍵のランダム具合？

⇒ 過去の利用履歴も含めたランダムさのこと

つまり…

- 過去に使った鍵は二度と使わない
- 暗号化の鍵は毎回ランダム変更し、可能ならば使い捨てということ。

…なぜか？

- 鍵が1つ漏れてしまうと、過去の暗号化データまで一網打尽…。
- なので、万一暗号鍵1つ漏れちゃったとしても、他の暗号化データにまで影響が出ないことを保証しなきゃならない。⁵

⁵これを保証することを (Perfect) Forward Secrecy とか呼ぶ。

ただ、現実的にはパスワードなどを毎回変更することは難しい。



固定のパスワードなどから毎回ランダムな鍵を生成する方法を使う⁶。

⁶PBKDF2 (RFC8018), HKDF (RFC5869)

2: AES で使う鍵を総当たりする際の大変さ？

⇒ 総当たり攻撃やるには膨大な計算が必要になるようにすること

3: AES の利用モードの安全性？

⇒ AES の API で設定できる利用モード ('AES256-CBC' とか) と、そのパラメタ設定の適切な設定が必要。

AES の「利用モード」

AES の処理 1 回で暗号化できるのはたった 16bytes にすぎない。長いデータを連続で暗号化するために、**暗号化処理を連続して組み合わせる方法**が利用モード。

AESの使い方: とりあえず暗号化してみよう

パスワードで暗号化してみる

pbkdf 使え

バイナリ鍵で暗号化してみる

hkdf 使え

危ない暗号化モードで暗号化してみる

ecb とか論外だから ctr とか使え

AES の使い方: 細かめの解説

PBKDF2 の使い方 in JavaScript

jscu なら動くよ

HKDF の使い方 in JavaScript

暗号化モードの設定

ctr 云々。

js でのサポート具合を列挙

まとめ

まとめ

お疲れ様でした。



次回以降…リクエスト次第ですが、

- 公開鍵暗号とその使い方
- 「情報が改ざんされていない」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。

宣伝 1

E2E 暗号化ファイル転送サービス「iTransfy」を提供しています。

宣伝 2

Zettant ではイケイケの仲間を募集しています。

- 1 今回は共通鍵暗号
- 2 公開鍵暗号& Hybrid Encryption
- 3 ハッシュ・署名と HMAC
- 4 超マニアック講座：RFC とアルゴリズム・フォーマット

Appendix

This page is not counted.