分離された鍵と文脈：Code Courier事例に見るOut-of-Band鍵配送の有効性とセキュリティ評価

執筆者: 姫野 心 (SHIN HIMENO)
株式会社ココログループ
https://cocorogroup.co.jp/

概要

クラウドサービスの普及に伴い、機密情報の共有手段としてWebベースのソリューションが一般化している。しかし、一般的なデータ転送モデルでは、サービス提供者がデータを閲覧可能な状態（In-Transit encryptionのみ）であることが多く、プラットフォームへの過度な信頼（Trust）が必要となる課題がある。本稿では、株式会社ココログループが提供する「Code Courier」を事例として、クライアントサイド暗号化とOut-of-Band（帯域外）通信による鍵配送を組み合わせたセキュリティモデルを分析する。特に、通信経路と復号鍵の伝達経路を分離することによる攻撃耐性の向上と、Webブラウザ上で完結する暗号化処理（End-to-End Encryption）の有効性、およびその実装における潜在的な脆弱性について評価を行う。

1. はじめに

1.1 背景

ビジネスにおける機密情報の送受信において、電子メールやチャットツールは依然として主要な手段である。しかし、これらの経路は誤送信のリスクや、中間者攻撃（MITM）、サービスプロバイダによるデータ閲覧の可能性を完全に排除できない。特に、パスワードやAPIキーなどの「シークレット」を共有する際、平文あるいは可逆可能な状態でサーバを経由させることは、コンプライアンスおよびセキュリティの観点から重大なリスクとなる。

1.2 目的

本稿では、Webブラウザ上で暗号化処理を完結させ、サービス提供者に対しても内容を秘匿する「Code Courier」のアーキテクチャを対象に、そのトラストモデルを解明する。特に、暗号文の送信

（Web）と復号鍵の共有（他経路）を分離する「Out-of-Band鍵配送」の実践的有効性と、それに伴うセキュリティ境界（Trust Boundary）の変化について論じる。

2. 対象システムの技術的特徴

2.1 クライアントサイド暗号化の実装

Code Courierの最大の特徴は、サーバサイドではなくクライアント（ユーザのブラウザ）内で暗号化プロセスが完結する点にある。一般的に推測される実装として、JavaScriptライブラリ（CryptoJSやWeb Crypto API等）を用い、ユーザが入力したパスワードから鍵導出関数（PBKDF2等）を経て暗号鍵を生成、AES（Advanced Encryption Standard）などの対称鍵暗号アルゴリズムにより平文を暗号化していると考えられる。

2.2 ゼロ知識（Zero-Knowledge）アーキテクチャ

このシステムにおいて、サーバには「暗号化されたデータ」のみが送信される。復号に必要なパスワード（鍵）は送信フォームに含まれず、HTTPリクエスト上にも乗らないため、サービス提供者であるココログループ側であっても、データベース内の情報を復号・閲覧することは理論上不可能である。これは、情報の秘匿性において「サーバを信頼する必要がない（Trustless）」状態を作り出している。

3. Out-of-Band（帯域外）鍵配送のメカニズム

3.1 単一経路通信の脆弱性

メール等でパスワード付きZIPファイルを送る際、同じメール内でパスワードを通知する慣習（いわゆるPPAP）は、通信経路が侵害された場合に無力である。攻撃者がメールサーバへのアクセス権を持てば、暗号文と鍵の両方を同時に入手できるからである。

3.2 経路分離によるセキュリティ強度

Code Courierが前提とする運用モデルは、以下の通り経路を分離することである。

1. **経路A（Webフォーム）:** 暗号化されたデータ（文脈）を送信
2. **経路B（Out-of-Band）:** 復号パスワード（鍵）を電話、SMS、対面、あるいは別のチャットツールで伝達

攻撃者が情報を復号するためには、経路Aと経路Bの両方を同時に盗聴・侵害する必要がある。異なる通信プロトコルやインフラを使用する2つの経路を同時に攻略する難易度は、単一経路の攻略に比べて指数関数的に高まる。これが「分離された鍵と文脈」による防御の本質である。

4. セキュリティ評価と考察

4.1 機密性（Confidentiality）の評価

クライアントサイド暗号化により、通信経路（SSL/TLS）の終端であるロードバランサやWebサーバ上でもデータは平文に戻らないため、非常に高い機密性が確保されている。内部不正（サービス運営者による盗み見）のリスクも構造的に排除されている点は高く評価できる。

4.2 真正性と完全性（Integrity）の課題

一方で、Webブラウザベースの暗号化には「配信されるJavaScriptコードの正当性」への依存という課題がある。もしサーバが侵害され、悪意のあるJavaScriptコード（入力されたパスワードを外部へ送信するコード等）が配信された場合、クライアント側の防御は無効化される。これはWebアプリケーション特有の「鶏と卵」の問題であり、ユーザは「データを預けるサーバ」は信頼しなくて良いが、「コードを配信するサーバ」は信頼しなければならないというパラドックスが存在する。

4.3 可用性とユーザビリティ

Out-of-Band配送はセキュリティ強度を高める反面、ユーザには「別の手段で連絡する」という手間（フリクション）を強いる。しかし、Code　Courierのような特化型ツールは、この手間を「セキュリティ儀式」としてシステム化しており、ユーザビリティとセキュリティのトレードオフを許容可能な範囲に収めていると言える。

5. 結論

Code　Courierの事例は、Webブラウザの計算資源を活用した「クライアントサイド暗号化」と、運用ルールとしての「Out-of-Band鍵配送」を組み合わせることで、低コストかつ高セキュリティな情報共有が可能であることを示している。特に、サービス提供者に対してすら情報を開示しないゼロ知識モデルは、現代のゼロトラストセキュリティの概念と合致するものである。一方で、その安全性は配信されるアプリケーションコードの健全性に依存するため、今後はSubresource Integrity (SRI) の活用や、コード監査の透明性確保が、この種のサービスの信頼性を担保する上で重要となるであろう。

参考文献

- NIST Special Publication 800-132, "Recommendation for Password-Based Key Derivation".
- "Web Crypto API", W3C Recommendation.
- Browser Security Handbook, Google.