

## 複数サーバに復号権限を分散した Web ベース ファイル送受信システム

佐藤誠<sup>†1</sup> 毛利公美<sup>†2</sup> 土井洋<sup>†3</sup> 白石善明<sup>†1</sup>

内容を秘匿すべきファイルを送受信したいとき、ファイルを共通鍵暗号方式で暗号化し、ファイルの暗号化に用いたセッション鍵を公開鍵暗号方式を使って送受信者間で共有するという手段が一般的である。セッション鍵の共有に公開鍵基盤 (PKI) ベースの公開鍵暗号化を使うと、受信者の秘密鍵を保存した端末に利用が限定される。本稿では、利用者がソフトウェアを新たにインストールすることなく Web ブラウザで利用できる暗号化ファイル送受信システムを提案する。ファイルの暗号化に用いたセッション鍵を、個人を特定できる情報 (ID) を公開鍵として利用することのできる ID ベース暗号 (IBE) の一方式を使って送受信者間で共有する。IBE では、秘密鍵 (復号鍵) を復号鍵発行サーバ (PKG) が生成する。復号鍵配布時の PKG による受信者の認証に既存のメール送信者認証を利用することで、認証システムの新規構築を不要とした。提案システムは PKI ベースの暗号化メールをファイル送受信に使う場合と比較して、利用者が端末にしばられず利便性が高いことを確認した。

### A Web-Based File Transfer System Cooperating with Multiple Servers in Decryption

MAKOTO SATO<sup>†1</sup> MASAMI MOHRI<sup>†2</sup> HIROSHI DOI<sup>†3</sup>  
YOSHIAKI SHIRAISHI<sup>†1</sup>

#### 1. はじめに

内容を秘匿すべきファイルを送受信したいとき、ファイルを共通鍵暗号方式で暗号化し、ファイルの暗号化に用いたセッション鍵を公開鍵暗号方式で送信者から受信者に配送するという手段が一般的である。この方法は例えば代表的な暗号化メール S/MIME[1]に見られる。

S/MIME でのセッション鍵の配送には、公開鍵基盤 (PKI) が構築されていることを前提として、その認証局により発行された受信者の公開鍵証明書が使われる。送信者は受信者の公開鍵証明書をメーラーにインポートしてメール本文およびその添付ファイルを公開鍵暗号で暗号化する。メールの本文に暗号化されたセッション鍵を含めて送信して鍵配送がされる。他方の受信者にも端末に暗号化メールに対応したメーラーがインストールされていることが求められる。受信者がメーラーにインポートして利用する秘密鍵は受信者自身とひもづけられるため、自身の秘密鍵がインポートされたメーラーがインストールされている端末に使用が限定されることになる。端末の限定は安全性の担保という面では望ましいこともある一方で、厳格な運用に起因する利便性の低下による適切な暗号化ファイルの送受信がなされなかったり、情報交換の頻度が落ちて知的生産性が低下するといったことも考えられる。本研究では、安全性を

担保することは必要条件として安全な暗号技術を基本的な構成要素として使い、そして、知的生産性の低下を避けるために利便性を重視するという立場で暗号化ファイル送受信システムを実現することを目的としている。

受信者の E-mail アドレスなどの個人を特定できる情報 (ID) を公開鍵として利用できる ID ベース暗号 (IBE) は公開鍵証明書の準備が不要であり、利便性という観点からは優れた暗号技術である。PKI ベースの公開鍵暗号では公開鍵と秘密鍵のペアを受信者自身が生成するのに対して、IBE では受信者が復号鍵発行サーバ (PKG) に送った ID に対応する秘密鍵 (復号鍵) を PKG から受け取るという違いがある。このことが、PKI ベースで受信者が鍵ペアを作成し、公開鍵証明書を発行して送信者がそれを受け取りインストールすることと受信者自身が秘密鍵をインポートすることといった事前準備が IBE では不要になることを意味している。安全な IBE の方式を利用することで、システムの安全性は確保しながら端末が限定されるという制約が取り除かれることになる。著者らは既に、端末を限定しないことを目的とした受動的攻撃に対して安全な Web ベースファイル送受信システムのための IBE 方式[2]を提案している。

文献[2]の方式には、受信者は PKG から復号鍵を安全に取得するという前提があった。セッション鍵を復元するための IBE の秘密鍵の受け取りには受信者の認証が必要になる。既存の利用者認証機能がついているサービスを用いることができれば、暗号化ファイル送受信システムのために認証システムを新規に構築する必要がなくなり、ひいては

<sup>†1</sup> 名古屋工業大学  
Nagoya Institute of Technology

<sup>†2</sup> 岐阜大学  
Gifu University

<sup>†3</sup> 情報セキュリティ大学院大学  
Institute of Information Security

利用者はアカウント発行の手続きをせずに暗号化ファイル送受信システムの利用が可能となる。著者らは既に、PKG への ID の送信をメールで行いメール送信者認証を復号鍵発行時の PKG による受信者認証に利用する手法[3]を提案している。

本論文では、ファイル暗号化に使うセッション鍵の配送に著者らの提案している IBE[2]を適用し、復号鍵発行は文献[3]の手法を使うことで、利用者がソフトウェアを新たにインストールすることなく Web ブラウザで利用できる暗号化ファイル送受信システムを提案する。以降、2 章で提案システムに利用する暗号方式を説明し、3 章で提案システムの構成、4 章で提案システムの実装について述べる。そして 5 章で提案システムを評価し、6 章で本稿をまとめる。

## 2. 提案システムに利用する暗号方式

### 2.1 モデル

提案システムに利用する文献[2]の暗号方式は以下の 7 つのアルゴリズムから構成される。図 1 にモデルを示す。

**PKG.Setup** : PKG が実行するアルゴリズム。セキュリティパラメータ  $k$  を入力として、公開パラメータ  $params$ , マスター秘密鍵  $msk$  を出力する。

**PKG.Ext** : PKG が実行するアルゴリズム。公開パラメータ  $params$ , マスター秘密鍵  $msk$ , ID を入力とし、復号鍵  $d_{ID}$  を出力する。

**MCD.KG** : メッセージ成分保管サーバ (MCD) が実行するアルゴリズム。公開パラメータ  $params$  を入力とし、公開鍵 MCD.PK, 秘密鍵 MCD.SK を出力する。

**Enc** : 送信者が実行するアルゴリズム。公開パラメータ  $params$ , 平文  $M$ , ID を入力とし、暗号文  $C = \langle C_R, C_M \rangle$  を出力する。以降、暗号文のうち、平文の情報を含む成分をメッセージ成分、含まない成分を乱数成分と表記する。

**MCD.Enc** : 送信者が実行するアルゴリズム。メッセージ成分  $C_M$ , MCD の公開鍵 MCD.PK を入力とし、 $C'_M$  を出力する。

**MCD.Dec** : MCD が実行するアルゴリズム。メッセージ成分  $C'_M$ , MCD の秘密鍵 MCD.SK を入力とし、 $C_M$  を出力する。

**Dec** : 受信者が実行するアルゴリズム。公開パラメータ  $params$ , ID に対応する復号鍵  $d_{ID}$ , 暗号文  $C = \langle C_R, C_M \rangle$  を入力とし、平文  $M$  を出力する。

### 2.2 方式の構成法

文献[2]の暗号方式は Boneh, Franklin によって 2001 年に提案された IBE (以下 BF 方式とする) の BasicIdent[4]をベースにした方式であり、MCD, PKG のいずれか一方が攻撃者と結託した場合にも受動的な攻撃に対して安全であることが証明されている。具体的な構成法を以下に示す。

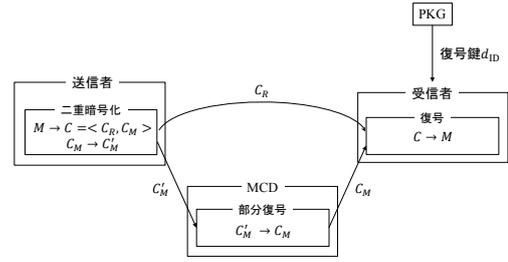


図 1 提案システムに利用する暗号方式のモデル

Figure 1 Model of the encryption scheme used in the proposed system.

**PKG.Setup** : セキュリティパラメータ  $k \in \mathbb{Z}^+$  を入力として、素数  $q$ , 素数位数  $q$  の群  $\mathbb{G}_1, \mathbb{G}_2$  と双線形写像  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  を出力する。ランダムに生成元  $P \in \mathbb{G}_1$ ,  $s \in \mathbb{Z}_q^*$  を選択し、 $P_{pub} = sP$  を計算する。ハッシュ関数  $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ ,  $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ ,  $H_M : \mathbb{G}_1 \rightarrow \{0,1\}^n$  を選択する。システムの公開パラメータ  $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_M \rangle$ , マスター秘密鍵  $msk = s$  を出力する。平文空間  $\mathcal{M} = \{0,1\}^n$ , 暗号文空間  $\mathcal{C} = \mathbb{G}_1^* \times \{0,1\}^n$  と定める。

**PKG.Ext** : 公開パラメータ  $params$ , 任意の  $ID \in \{0,1\}^*$ ,  $msk$  を入力とし、 $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$  を計算する。ID に対する秘密鍵  $d_{ID} = sQ_{ID}$  を計算し  $d_{ID}$  を出力する。

**MCD.KG** :  $params$  を入力とし、ランダムに  $b \in \mathbb{Z}_q^*$  を選び、 $(MCD.PK, MCD.SK) = (bP, b)$  を出力する。ここで、MCD.PK, MCD.SK はそれぞれ MCD の公開鍵と秘密鍵である。

**Enc** : 公開パラメータ  $params$ , 平文  $M \in \{0,1\}^n$ ,  $ID \in \{0,1\}^*$  を入力とし、 $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$  を計算する。 $r \in \mathbb{Z}_q^*$  をランダムに選び、暗号文  $\langle C_R, C_M \rangle = \langle rP, M \oplus H_2(\hat{e}(Q_{ID}, P_{pub})^r) \rangle$  を生成する。

**MCD.Enc** (メッセージ成分の暗号化) : **Enc** によって出力されたメッセージ成分  $C_M$ , MCD.PK =  $bP$  を入力とし、 $a \in \mathbb{Z}_q^*$  をランダムに選び、 $C'_{M1} = aP$  とする。 $H_M(abP)$  を計算し、 $C'_{M2} = C_M \oplus H_M(abP)$  とする。 $C'_M = \langle C'_{M1}, C'_{M2} \rangle$  を出力する。

**MCD.Dec** (メッセージ成分の部分復号) : **MCD.Enc** で生成した  $C'_M$  と MCD.SK =  $b$  を入力とし、 $C_M = C'_{M2} \oplus H_M(bC'_{M1})$  を計算し、 $C_M$  を出力する。

**Dec** : 公開パラメータ  $params$ , **Enc** によって生成された乱数成分  $C_R$ , **MCD.Dec** で部分復号されたメッセージ成分  $C_M$ , 復号鍵  $d_{ID}$  を入力とし、 $M = C_M \oplus H_2(\hat{e}(d_{ID}, C_R))$  を計算し、 $M$  を出力する。

## 3. 提案システム

### 3.1 システムの構成

共通鍵暗号方式でファイルを暗号化し、暗号化に使ったセッション鍵を文献[2]の暗号方式で送受信者間で共有するファイル送受信システムを図 2 のように構成する。シス

システムの構成要素は、送信者用クライアントアプリケーション、受信者用クライアントアプリケーション、MCD, PKG, 暗号化ファイル保管サーバ、送信者側メールサーバ、受信者側メールサーバ、PKG 側メールサーバの 8 つであり、セッション鍵配送サブシステム、暗号化ファイル配送サブシステム、復号鍵配送サブシステムの 3 つのサブシステムから構成される。

### 3.2 サブシステムの構成

ファイルの暗号化に使用するセッション鍵は図 3 に示すセッション鍵配送サブシステムによって共有する。暗号化したファイルの配送には図 4 に示す暗号化ファイル配送サブシステムを使い、暗号化されたセッション鍵の復号に必要な復号鍵の配送は図 5 に示す復号鍵配送サブシステムによって行う。以下でそれぞれのサブシステムで送信者用クライアントアプリケーション、受信者用クライアントアプリケーション、MCD, PKG, 暗号化ファイル保管サーバ、送信者側メールサーバ、受信者側メールサーバ、PKG 側メールサーバが行う処理を示す。

#### 3.2.1 セッション鍵配送サブシステム

##### 送信者用クライアントアプリケーション

- (1) アルゴリズム **Enc** に従って、受信者の ID と PKG の公開パラメータ  $params$  を使いセッション鍵を暗号化し暗号文  $C = \langle C_R, C_M \rangle$  を生成する
- (2) **MCD.Enc** に従って、MCD の公開鍵を使い  $C_M$  を暗号化し、 $C'_M$  を生成する
- (3)  $C_R, C'_M$  を MCD に送信する

##### MCD

###### [暗号文受信]

- (1) 送信者から  $C_R, C'_M$  を受信する
- (2) **MCD.Dec** に従って、MCD の秘密鍵を使い  $C'_M$  を部分復号し、 $C_M$  を生成する

###### [暗号文送信]

- (1) 受信者から鍵取得リクエストを受信する
- (2) 受信者の認証を行う
- (3)  $C = \langle C_R, C_M \rangle$  を受信者に送信する

##### 受信者用クライアントアプリケーション

- (1) MCD に鍵取得リクエストを送り、 $C = \langle C_R, C_M \rangle$  を受け取る
- (2) **Dec** に従って、PKG から受け取った復号鍵  $d_{ID}$  を使い、 $C$  を復号してセッション鍵を得る

#### 3.2.2 暗号化ファイル配送サブシステム

##### 送信者用クライアントアプリケーション

- (1) セッション鍵でファイルを暗号化し、暗号化ファイル保管サーバに送信する
- (2) ファイルの送信を送信者側メールサーバに通知する

##### 暗号化ファイル保管サーバ

###### [暗号化ファイル受信]

- (1) 暗号化されたファイルを受信する

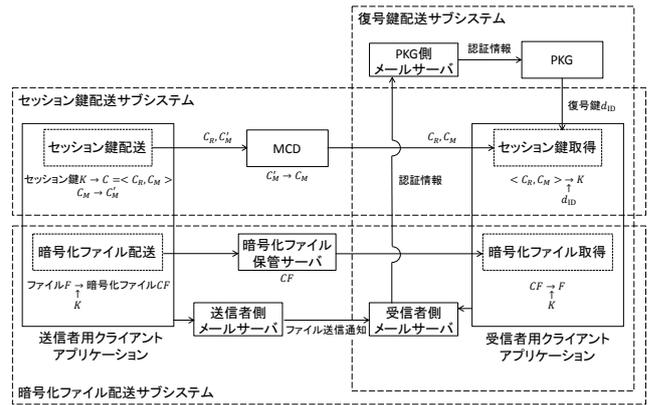


図 2 提案システムの構成

Figure 2 Structure of the proposed system.

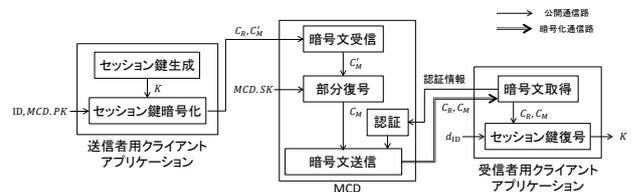


図 3 セッション鍵配送サブシステムの構成

Figure 3 Structure of the session key distribution subsystem.

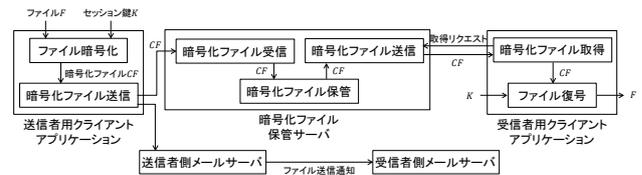


図 4 暗号化ファイル配送サブシステムの構成

Figure 4 Structure of the encrypted file distribution subsystem.

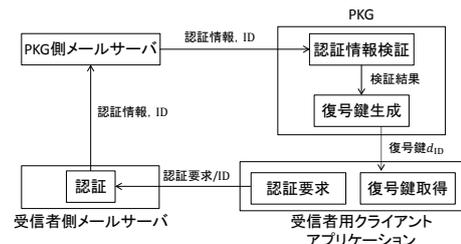


図 5 復号鍵配送サブシステムの構成

Figure 5 Structure of the decryption key distribution subsystem.

- (2) 受信者の宛先ごとに暗号化ファイルを保管する

###### [暗号化ファイル送信]

- (1) ファイル取得リクエストを受信する
- (2) 暗号化ファイルを受信者に送信する

##### 送信者側メールサーバ

ファイル送信通知にメールを利用し、受信者は能動的にファイルの受信を問い合わせる必要が無い構成とした。

- (1) 送信者から受け取ったファイル送信通知からファイル送信通知メールを作成する

(2) メールを受信者側メールサーバに送信する

#### 受信者側メールサーバ

送信者側メールサーバからファイル送信通知メールを受信する

#### 受信者用クライアントアプリケーション

- (1) 暗号化ファイル保管サーバにファイル取得リクエストを送信する
- (2) 暗号化ファイルを受信する
- (3) セッション鍵でファイルを復号する

### 3.2.3 復号鍵配送サブシステム

RFC5408[5]では、PKG の復号鍵発行時の認証方法に Basic 認証や Digest 認証が挙げられている。これらの認証には利用者によるユーザ登録と、PKG によるユーザ情報管理が必要になる。

受信者が復号鍵発行を要求した際にメールで自身のIDをPKGに送ることで、メール送信者認証を復号鍵発行時の認証として利用することができ、ユーザ登録・ユーザ情報管理が不要になる。

#### 受信者用クライアントアプリケーション

[認証要求]

- (1) 受信者側メールサーバに認証を要求する
- (2) 受信者側メールサーバから認証結果を受け取る

[復号鍵発行要求]

IDを受信者側メールサーバに送信する

[復号鍵取得]

PKG からIDに対応する復号鍵 $d_{ID}$ を受け取る

#### 受信者側メールサーバ

[認証]

- (1) 受信者を認証する
- (2) 認証結果と受信者のIDを PKG 側のメールサーバに送信する

#### PKG 側メールサーバ

- (1) PKG からメール取得要求を受ける
- (2) 受信者側のメールサーバから受け取った認証結果と受信者のIDを PKG に送信する

#### PKG

[認証結果検証]

PKG から受け取った認証結果を検証する

[復号鍵生成]

検証に成功した場合にアルゴリズム **PKG.Ext** に従って ID と PKG の公開パラメータ  $params$  から復号鍵 $d_{ID}$ を生成し、受信者用クライアントアプリケーションに送る

## 4. 提案システムの実装

[2]の暗号方式に利用する二つのサーバ (PKG, MCD), 暗号化ファイル保管サーバ, 送受信者および PKG 側メールサーバを Java のサーブレットで Web サーバとして実装し, 送受信者が利用するクライアントアプリケーションは

表 1 サーバ, クライアントの動作環境

Table 1 Operation environment of servers and clients.

サーバ・クライアント共通	
CPU	Core i5 3.20GHz
Memory	4.00GB
OS	Windows 7 Professional SP1
サーバ	
Web Server	Apache Tomcat 7
クライアント	
Browser	Firefox 21.0

JavaScript を用いてブラウザ上で動作する Web アプリケーションとして実装した。サーバ, クライアントの動作環境を表 1 に示す。

### 4.1 セッション鍵配送サブシステム

#### 送信者用クライアントアプリケーション

- (1) セッション鍵 $K \in \{0,1\}^n$ , 受信者のID  $\in \{0,1\}^*$ , 公開パラメータ  $params$  を入力とし,  $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$  を計算する
- (2)  $r \in \mathbb{Z}_q^*$  をランダムに選び, 暗号文  $C = \langle C_R, C_M \rangle = \langle rP, M \oplus H_2(\hat{e}(Q_{ID}, P_{pub})^r) \rangle$  を生成する
- (3)  $C_M$ , MCD の公開鍵  $MCD.PK = bP$  を入力として,  $a \in \mathbb{Z}_q^*$  をランダムに選び,  $C'_{M1} = aP$  とする
- (4) HTTP の POST メソッドで  $C_R, C'_M = \langle C'_{M1}, C'_{M2} \rangle$  を MCD に送信する

#### MCD

[暗号文受信]

- (1) 送信者から  $C_R, C'_M$  を受け取る
- (2)  $C'_M$  と MCD の秘密鍵  $MCD.SK = b$  を入力として,  $C_M = C'_{M2} \oplus H_M(bC'_{M1})$  を計算する

[暗号文送信]

- (1) 受信者から ID と認証結果を受信する
- (2) 受信者の認証を行う
- (3)  $C_R, C_M$  を HTTPS のレスポンスで受信者に送信

#### 受信者用クライアントアプリケーション

- (1) MCD に HTTPS の POST メソッドで ID と認証結果を送信する
- (2) MCD からレスポンスで  $C_R, C_M$  を受信する
- (3) 公開パラメータ  $params, C_R, C_M$  と PKG から受け取った復号鍵 $d_{ID}$ を入力として,  $K = C_M \oplus H_2(\hat{e}(d_{ID}, C_R))$  を計算し, セッション鍵 $K$ を出力する

### 4.2 暗号化ファイル配送サブシステム

#### 送信者用クライアントアプリケーション

- (1) セッション鍵 $K \in \{0,1\}^n$ を用いてファイル $F$ をAES暗号により暗号化し暗号化ファイル $CF$ を生成する
- (2) 暗号化ファイル $CF$ を HTTP の POST メソッドにより暗号化ファイル保管サーバに送信する
- (3) HTTP の POST メソッドでファイルの送信を送信者

側メールサーバに通知する

#### 暗号化ファイル保管サーバ

[暗号化ファイル受信]

- (1) 送信者から HTTP の POST メソッドで受信者のIDと暗号化ファイル $CF$ を受け取る
- (2) 受信者のIDごとに暗号化ファイルを保管する

[暗号化ファイル送信]

- (1) 受信者からIDを受け取る
- (2) IDに対応する暗号化ファイルをレスポンスとして受信者に送る

#### 送信者側メールサーバ

- (1) 送信者から受け取ったファイル送信通知からファイル送信通知メールを作成する
- (2) メールを受信者側メールサーバに送信する

#### 受信者側メールサーバ

送信者側メールサーバからファイル送信通知メールを受信する

#### 受信者用クライアントアプリケーション

- (1) HTTP の POST メソッドで暗号化ファイル保管サーバにIDを送信する
- (2) レスポンスとして暗号化ファイルを受け取る
- (3) セッション鍵でファイルを復号する

### 4.3 復号鍵配送サブシステム

メール送信者認証に SPF (Sender Policy Framework) [6] と DKIM (Domain Keys Identified Mail) [7]を利用した。SPF によって送信ドメインの詐称を検知可能であり、DKIM ではメール本文の改ざんが検知可能である。送信者認証の結果はメールのヘッダに追記され、ヘッダを解析することで復号鍵発行時の認証に利用する。

また、PKG と受信者間で DH 鍵共有を利用してセッション鍵を共有し、PKG はセッション鍵で復号鍵を暗号化して送信するように実装することで、暗号化通信路を使わなくても安全に復号鍵を授受できる。

#### 受信者用クライアントアプリケーション

[認証要求]

- (1) 受信側メールサーバに HTTP の POST メソッドを用いて認証要求を送る
- (2) 受信側メールサーバからレスポンスとして認証結果を受け取る

[復号鍵発行要求]

- (1) PKG の公開情報 $P$ , 受信者の秘密情報 $0 \leq t \leq q-2$ から $y_A = P^t$ を生成する
- (2)  $y_A$ , IDをメールで受信者側メールサーバに送信する

[復号鍵取得]

- (1) PKG から暗号化復号鍵,  $y_B$ を受信する
- (2)  $y_B$ と受信者の秘密情報 $t$ からセッション鍵 $P^{tu}$ を生成する
- (3) セッション鍵を使って復号鍵を生成する

#### 受信者側メールサーバ

[認証]

- (1) 受信者を認証する
- (2) 認証結果と受信者のIDをメールで PKG 側メールサーバに送信する

#### PKG 側メールサーバ

- (1) PKG からメール取得要求を受け取る
- (2) 受信者側メールサーバから受け取った認証結果と受信者のIDをメールで PKG に送信する

#### PKG

[認証結果検証]

- (1) PKG 側メールサーバからメールを受信する
- (2) 取得したメールのヘッダを調べ、Received-SPF:が pass であることと、Authentication-Results:に dkim = pass が含まれていることを確認すると認証成功となる

[復号鍵生成]

- (1) 公開パラメータ $params$ , 任意のID  $\in \{0,1\}^*$ ,  $msk$ を入力とし、IDに対する秘密鍵 $d_{ID} = sH_1(ID)$ を計算する
- (2) PKG の秘密情報 $u$ と $y_A$ からセッション鍵 $P^{tu}$ を生成し、これを用いて復号鍵を暗号化する
- (3) PKG の秘密情報 $0 \leq u \leq q-2$ と公開情報 $P$ から $y_B = P^u$ を生成する
- (4) HTTP の POST メソッドを使って暗号化した復号鍵と $y_B$ を受信者に送信する

### 4.4 システムの利用手順

送信者と受信者はそれぞれ以下の操作をすることでシステムを利用する。利用の手順を図6に示す。

#### i. システムにログイン

【送信者/受信者共通】

- (1) 図7のログイン画面から、(a)にIDを入力し、(b)のログインボタンを押下することで、図8の送受信選択画面へ遷移する。
- (2) 送受信選択画面において、ファイルを送る場合は(a)のフィールドを、ファイルを受け取りたい場合は(b)のフィールドを押下する。(a)のフィールドを押下した場合には図9のファイル送信画面へ、(b)のフィールドを押下した場合には図10のファイル受信画面へ遷移する。

#### ii. ファイル・セッション鍵暗号化

【送信者】

- (1) 図9のファイル送信画面において、(a)に受信者のIDを入力する。
- (2) (b)のファイル選択ボタンを押下することで、(c)のエクスプローラーから送信するファイルを選択するか、(d)のフィールドにファイルをドラッグ&ドロップすることでファイルを送信する。

### iii. ファイル・セッション鍵復号

#### 【受信者】

- (1) 図 10 のファイル受信者画面において, (a)のフィールドから受信したいファイルを選択することでファイルを受信する。

## 5. 提案システムの評価

内容を秘匿すべきファイルを送受信したいとき, ファイルを共通鍵で暗号化し, ファイルの暗号化に用いたセッション鍵を公開鍵暗号方式を使って送信者から受信者に配送するといった手段が一般的である。この方法の代表例としては PKI ベースの S/MIME を使い, ファイルの配送をメールで行う手法があることを 1 章で述べた。

提案システムの目的は以下の 2 つである。

1. 安全性を担保する
  2. 利便性を損なわない
    - (ア) 操作が単純である
    - (イ) 利用端末を限定しない
- 1.は提案システムに用いる暗号方式の安全性から達成されている。

本章では, 提案システムと S/MIME ライクな暗号化メールによるファイル送受信の手順を比較して違いを考察し,

2. (ア) および (イ) の各項目を評価する。

### 5.1 暗号化メールによるファイル送受信

受信者の公開鍵を含む公開鍵証明書をメーラーにインポートすることで, メールを暗号化できる。暗号化メールでは, メール本文だけでなく添付ファイルも暗号化される。

プライベート認証局を設置して, 送受信者間で暗号化メールを送受信する手順を図 11 に示す。プライベート認証局は OpenSSL[8]などを使って構築できる。

#### i. ルート証明書を生成

##### 【認証局】

- (1) 認証局の秘密鍵を生成する
- (2) 証明書署名要求 (CSR) を生成する
- (3) 証明書署名要求に認証局の秘密鍵で自己署名し, ルート証明書を生成する
- (4) ルート証明書を送信者と受信者に送信する

#### ii. 受信者の証明書を生成

##### 【受信者】

- (1) 受信者の秘密鍵を生成する
- (2) CSR を生成し認証局に送信する

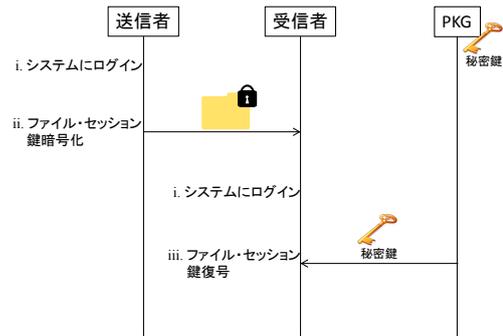


図 6 提案システムの利用手順

Figure 6 Procedures of using proposed system.



図 7 ログイン画面

Figure 7 Login screen.

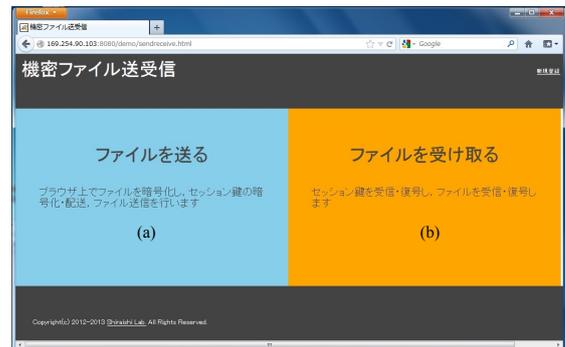


図 8 送受信選択画面

Figure 8 Send and receive selection screen.

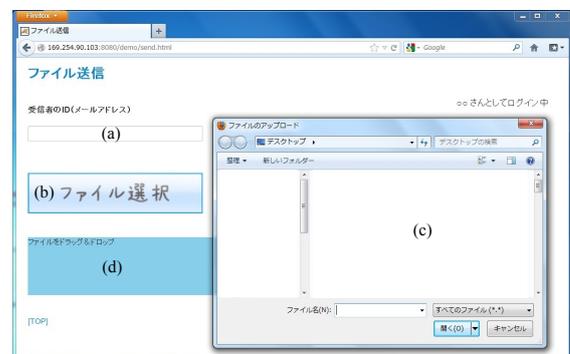


図 9 ファイル送信画面

Figure 9 File send screen.

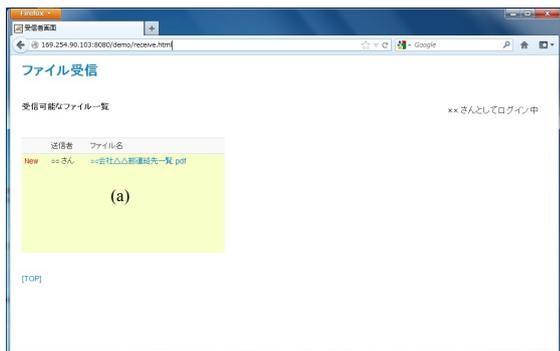


図 10 ファイル受信画面  
Figure 10 File receive screen.

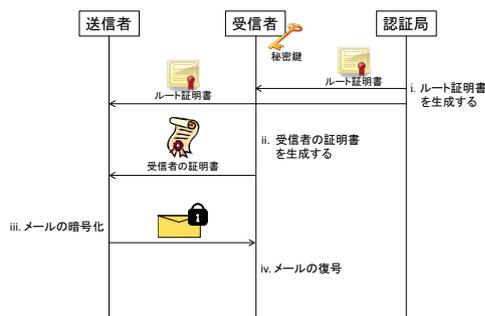


図 11 暗号化メールの送受信手順  
Figure 11 Procedures of sending and receiving encrypted E-mail.

(3) 認証局から証明書を受け取り、送信者に送る

**【認証局】**

- (1) CSR を検証する
- (2) CSR に認証局の秘密鍵で署名し、受信者の証明書を生成する
- (3) 証明書を受信者に送る

**iii. メールの暗号化**

**【送信者】**

- (1) ルート証明書と受信者の証明書をメーラーにインポートする
- (2) 受信者の証明書を使ってメールを暗号化し、受信者に送信する

**iv. メールの復号**

**【受信者】**

- (1) PKCS#12[9]形式のファイルを作成する。PKCS#12 形式とは秘密鍵と証明書を一つのファイルに格納する形式である
- (2) PKCS#12 形式ファイルをメーラーにインポートする
- (3) PKCS#12 形式のファイルを使って、送信者から受信した暗号化メールを復号する

**5.2 暗号化メールによるファイル送受信と提案システムの比較**

4.4 節に示した提案システムの利用手順と、5.1 節に示した暗号化メールによるファイル送受信手順について、送受

表 2 暗号化メール・提案システムの CUI・GUI 操作  
Table 2 Operations of encrypted E-mail and proposed system by CUI and GUI.

暗号化メール		提案システム	
CUI 操作	GUI 操作	CUI 操作	GUI 操作
ii.【受信者】(1)	ii.【受信者】(3)		i.【送信者】(1)
ii.【受信者】(2)	iii.【送信者】(1)		i.【送信者】(2)
iv.【受信者】(1)	iii.【送信者】(2)		ii.【送信者】(1)
	iv.【受信者】(2)		ii.【送信者】(2)
	iv.【受信者】(3)		i.【受信者】(1)
			i.【受信者】(2)
			iii.【受信者】(1)

信者の操作を CUI 操作と GUI 操作に分類したものを表 2 に示す。表 2 からわかるように、暗号化メールを利用する場合には、秘密鍵、CSR、PKCS#12 形式のファイル作成の際に CUI 操作をする必要があり、コマンドラインからの正確なコマンド入力求められる。一方で提案システムを使う場合は、送受信者は共に CUI 操作をする必要がなく、マウスクリックなどの GUI 操作だけでファイルの送受信ができる。

また、暗号化メールを利用する場合、受信者は ii.【受信者】(1)の操作で秘密鍵を作成し、iv.【受信者】(1)の操作で秘密鍵を含む PKCS#12 形式のファイルを作成してメーラーにインポートする。秘密鍵を持たない端末では暗号化されたメールを復号できないので、利用する端末が限定されることになる。一方、提案システムでは、暗号化ファイルの復号に使うセッション鍵の復号に必要な復号鍵の取得は iii.【受信者】(1)の操作で行うので利用する端末は限定されない。

**6. おわりに**

機密ファイルを安全に送受信する手段では、共通鍵によってファイルを暗号化し、暗号化に使ったセッション鍵を公開鍵暗号方式で共有するのが一般的である。

本稿では、利用者がソフトウェアを新たにインストールすることなく Web ブラウザから利用できる暗号化ファイル送受信システムを提案した。ファイルの暗号化に用いたセッション鍵を ID ベース暗号の一方式を利用して送受信者間で共有する。暗号化されたセッション鍵の復号には PKG から発行される復号鍵を使う。復号鍵発行時の PKG による受信者認証に既存のメール送信者認証を利用することで、新たな認証システムの構築を不要とした。この認証手法では利用者によるユーザ登録と PKG によるユーザ情報管理が不要である。

提案システムは PKI ベースの暗号化メールをファイル送受信に利用する場合と比較して、送受信者の CUI 操作が不要であることを確認した。今後の課題として、提案システ

ムへの認証連携の適用について検討することが挙げられる。

### 参考文献

- 1) Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification RFC 5751, available from <<http://datatracker.ietf.org/doc/rfc5751/>> (accessed 2013-05-17).
- 2) 佐藤誠, 毛利公美, 土井洋, 白石善明: 部分的に二重暗号化する ID ベース暗号方式とその評価, 電子情報通信学会技術研究報告, 情報通信システムセキュリティ (ICSS), Vol.112, No.499, pp.19-24 (2013).
- 3) 伴拓也, 毛利公美, 土井洋, 白石善明: メールサービスの認証を利用した ID ベース暗号の復号鍵発行手法, 情報処理学会全国大会講演論文集, Vol.74, No.3, pp.649-650 (2012).
- 4) Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, SIAM J. of Computing, Vol.32, No.3, pp.586-615 (2003).
- 5) Identity-Based Encryption Architecture and Supporting Data Structures RFC 5408, available from <<http://datatracker.ietf.org/doc/rfc5408/>> (accessed 2013-05-17)
- 6) Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1 RFC 4408, available from <<http://datatracker.ietf.org/doc/rfc4408/>> (accessed 2013-05-15).
- 7) DomainKeys Identified Mail (DKIM) Signatures RFC 6376, available from <<http://datatracker.ietf.org/doc/rfc6376/>> (accessed 2013-05-15).
- 8) OpenSSL Cryptography and SSL/TLS Toolkit, available from <<http://www.openssl.org/>> (accessed 2013-05-15).
- 9) RSA Laboratories, available from <<http://www.rsa.com/rsalabs/node.asp?id=2138>> (accessed 2013-05-15).