3ZA-6

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

伴　拓也†　　毛利　公美‡　　土井　洋††　　白石　善明†

名古屋工業大学†　　岐阜大学‡　　情報セキュリティ大学院大学††

1. はじめに

ID ベース暗号(Identity-Based Encryption, IBE)は，任意の ID を公開鍵として利用できることで注目されている．IBE では，PKG(Private Key Generator)と呼ばれる信頼のおける機関を利用する．PKG は次の 2 つの役割を持つ．一つは，公開パラメータの発行，もう一つは，ユーザの ID に対応する復号鍵の発行である．復号鍵の発行では，ユーザに復号鍵を正しく渡すためにユーザ認証を行う．RFC5408[1]では，復号鍵発行時の認証の方法として，Basic 認証，Digest 認証，その他の認証が挙げられている．Basic 認証，Digest 認証を利用する場合，ユーザによるユーザ登録，PKG によるユーザ情報の管理が必要になる．

本稿では，復号鍵発行時の認証を，メールサービスを利用して行う手法を提案する．PKG とメールサービスの認証を連携するには，復号鍵受信ユーザへのなりすましとメールサーバ(MTA)間での盗聴を防止する必要がある．提案手法では，メールの送信者認証と DH 鍵共有によるセッション鍵を使い復号鍵を暗号化する．Boneh，Franklin の IBE を実装して動作を確認し，認証の処理に要する時間を測定した．

2. ID ベース暗号と復号鍵の発行

IBE では，個人と 1 対 1 で結びつく任意の ID を公開鍵として利用できる．電子メールアドレス等の ID は持ち主が自明であるので，公開鍵証明書が不要となる．

IBE は 1984 年に Shamir による概念の発表以来，多くの方式が発表されている．ペアリングを用いた Boneh-Franklin らの方式(BF 方式)[2]や，Boneh-Boyen らの方式(BB1 方式)[3]などがある．

送信者から IBE により暗号化したデータを受信者に送り，復号する場合，図 1 の(1)～(5)のような流れになる．送信者は PKG から公開パラメータを取得し，それらと受信者の ID から暗号鍵を作成する．平文を暗号化して暗号文を受信者に渡す．受信者は復号鍵で暗号文を復号する．

受信者が復号鍵を取得する場合，図 1 の(i)～(iv)のような流れになる．受信者は自身の ID を PKG に渡し，PKG は ID に対応する復号鍵を作成して受信者に渡す． このとき PKG は ID に対応する復号鍵を発行する際に，ユーザ認証を行い，ID の持ち主による復号鍵発行要求であることを確認する必要がある．

3. 復号鍵発行時の認証

3.1 復号鍵発行時に用いられる認証

RFC5408 では，ID ベース暗号のセキュリティアーキテクチャが記されている．ここでは復号鍵発行時の認証方法として，Basic 認証，Digest 認証，その他の認証の利用について示されている．Basic 認証，Digest 認証を PKG の復号鍵発行時に利用する場合，PKG に認証情報を渡す際の通信路の暗号化，ユーザによる認証情報の登録，PKG による認証情報の管理が必要となる．

本稿では，運用コストやユーザの利便性を考慮し，認証情報を PKG に登録することなく認証を行う手法を提案する．

A Private Key Issue Method Using Authentication Function of Email Service for Identity-based Encryption

† Takuya BAN and Yoshiaki SHIRAISHI ・ Nagoya Institute of Technology

‡ Masami MOHRI ・ Gifu University

†† Hiroshi DOI ・ Institute of Information Security

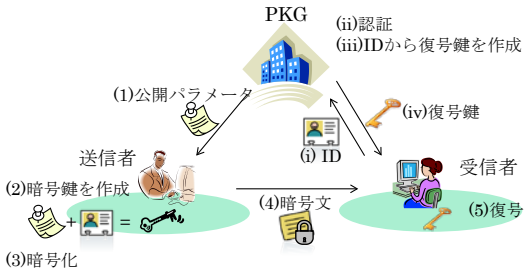

図 1　IBE によるデータの暗号化／復号

3.2 PKG への認証情報の登録・送信を不要とする認証

PKG への認証情報の登録および送信を不要にする認証方式として，Email Answerback 認証がある．Email Answerback 認証は，メールサービスにアクセスできるのは本人のみであることを利用した認証方式である．

Email Answerback 認証の手順を次に示す．(1)証明者は検証者にメールアドレスを送る．(2)検証者は，メール本文にセッション ID 付きの URL を載せて証明者に送ると同時に，直接，証明者に認証子を送信する．(3)証明者は，受信したメールの URL にアクセスし，セッション ID と認証子を検証者に送る．(4)検証者は，セッション ID と認証子を照合することで認証する．

Email Answerback 認証で，セッション ID 付きの URL が書かれたメールを検証者から証明者に送る際に，メールは，検証者，検証者の利用する MTA，証明者の利用する MTA，証明者の順に受け渡される．これらのうち，検証者から検証者の利用する MTA への通信路と，証明者の利用する MTA から証明者への通信路は暗号化が可能であるが，検証者の利用する MTA から証明者の利用する MTA への通信路は暗号化されないので盗聴が懸念される．

以下では，Email Answerback 認証と同様に，メールサービスのログイン認証を利用することで，ユーザによるユーザ登録と，PKG によるユーザ情報の管理を不要にし，かつ安全に復号鍵発行時の認証を行う手法を提案する．

4. メールサービスの認証を利用した復号鍵発行手法

4.1 復号鍵の暗号化

PKG が作成したユーザの復号鍵をセッション鍵により暗号化し，暗号化された復号鍵を受け取ったユーザは，同一のセッション鍵で暗号文を復号し，復号鍵を得る．このようにすることで，暗号化されない通信路で第三者に復号鍵が渡らないようになる．具体的には，PKG とユーザ間で 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ユーザに渡す．

4.2 メールの送信者認証

PKG は，復号鍵発行要求を受け取ったとき，ID が本人のものかを確認するために認証を行う．ユーザが復号鍵発行要求時にメールを利用して自身の ID を送るようにすれば，メールサービスのログイン認証を復号鍵発行時の認証として利用できる．

メールサービスでは，メールヘッダを書き換えることによる送信者の偽装や，ユーザが送ったメールの内容を改ざんすることが技術的に可能である．これらの対策に 2 つのメール送信者認証技術がある．一つは，SPF(Sender Policy Framework)[4]である．SPF は，メールの送信ドメインが送信元と同一かどうかを受信サーバが送信サーバの情報を取得して確認することで，送信ドメインの詐称を検知可能である．もう一つは，

DKIM(Domain Keys Identified Mail)[5]である．DKIMでは送信者によりメールヘッダに署名を追加し，受信者がその署名を基に認証する．これによりメール本文の改ざんを検知可能である．

送信者認証の結果はメールのヘッダに追記される．このヘッダ情報を解析することで，メールサービスを復号鍵発行時の認証として利用する．

4.3 復号鍵発行の手順

復号鍵発行の手順を図 2 に示す．以降のべき乗剰余演算は，PKGの公開情報 p を法とする．

(1) ユーザは，PKGの公開情報 g，ユーザの秘密情報 a を用いて，DH鍵共有での配送鍵 g^a を作成する
(2) ユーザは，g^a をメールでPKGに送信する
(3) ユーザは，PKGにIDを送信し，復号鍵の発行を要求する
(4) PKG はメールを取得し，メールヘッダを調べ，Received-SPF:が pass であることと，Authentication-Results:にdkim=passが含まれることの確認を行う
(5) 送信者が認証されたら，IDを基に復号鍵を生成する
(6) PKG の秘密情報 b と g^a からセッション鍵 g^{ab} を作成し，g^{ab} で復号鍵を暗号化し，暗号文 C を作成する
(7) PKGは，C と配送鍵 g^b をユーザに渡す
(8) ユーザは g^b から g^{ba} を作成し，g^{ba} により C を復号し，復号鍵を得る

5. 安全性

提案手法では，以下の2通りの攻撃が想定される．

【送信元詐称なりすまし攻撃】
[手段] メールヘッダの書き換え
[手順]
(1) 攻撃者は，偽の秘密情報 a' から配送鍵 $g^{a'}$ を作成し，$g^{a'}$ を本文に書いたメールを作成する
(2) 攻撃者は，メールのヘッダの From:と Return-Path:を復号鍵受信者のIDに書き換えてPKGに送信する
(3) 攻撃者は，暗号化した復号鍵を盗聴し，セッション鍵で復号する

【配送鍵改ざん攻撃】
[手段] メール本文の改ざん，メールの盗聴
[手順]
(1) 攻撃者は，偽の秘密情報 a' から配送鍵 $g^{a'}$ を作成し，復号鍵受信者が PKG に復号鍵リクエストのメールを出したときに，本文の配送鍵を $g^{a'}$ に置き換える
(2) 攻撃者は，暗号化した復号鍵を盗聴し，セッション鍵で復号する

一つ目の送信元詐称なりすまし攻撃は，送信元ドメインの詐称を検知可能な SPF で防ぐことができる．二つ目の配送鍵改ざん攻撃は，メール本文の改ざん検知が可能な DKIM で防ぐことができる．

6. 動作確認：BF 方式での利用

6.1 動作確認に用いた環境

サンプルプログラムの実行環境を表 1 に示す．PKG とクライアントは同じマシン上で実行した． PKG を Java のサーブレットで実装し，Apache Tomcat 上で動作させた．クライアントは Adobe ActionScript3.0 で実装し，ランタイムとして，外部の MTA に Socket でアクセス可能な Adobe AIR を利用した．SPF と DKIM が両方利用可能なメールサービスの一つに Gmail がある．復号鍵受信者と PKG が利用するメールサービスとして Gmail を利用した．

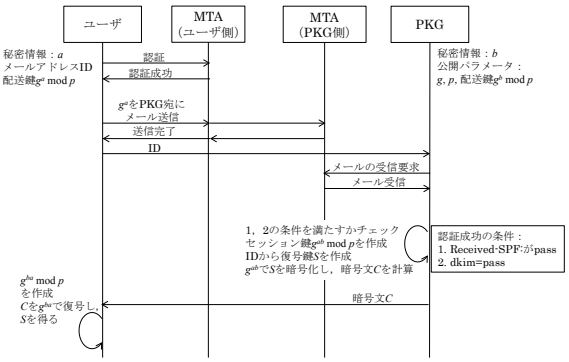

図 2　メールサービスを利用した復号鍵発行手順

表 1　サンプルプログラムの実行環境

PKG，クライアント共通	
CPU	Intel(R) Core i5 M560 2.67GHz
RAM	4.0 GB
OS	Windows 7 Professional 64bit
PKG	
ランタイム	JRE 1.6.1_29
サーバ	Apache Tomcat 7
クライアントアプリケーション	
ランタイム	Adobe AIR 3.1

6.2 サンプルプログラムの構成

送信者が，文字列を IBE により暗号化し，暗号文を受信者に送り，受信者は提案手法により PKG から復号鍵を取得し，暗号文を復号するサンプルプログラムを実装した．

IBE の方式は BF 方式の最も簡単な BasicIdent[2]を利用した．クライアントから PKG へのメール送信通知と復号鍵の受け取りは，HTTP の GET メソッドを利用し，ユーザから直接リクエストを送信するようにした．復号鍵はセッション鍵を利用して AES 暗号で暗号化した．BF 方式のクライアント側で必要となるペアリング演算は ActionScript3.0 のライブラリである As3Pairing[6]を利用し，暗号化ライブラリとして As3Crypto[7]を利用した．

6.3 提案手法の動作確認の結果

提案手法の DH 鍵共有によるセッション鍵の共有，AES による復号鍵の暗号化／復号が動作することを確認した．復号鍵の発行要求から暗号化した復号鍵を復号し，復号鍵を取得するまでの所要時間は 10 回の試行の平均で約 10 秒であった．この内，ユーザ側の Gmail へのメール送信処理に要した時間は約 6 秒，PKG でのメール受信処理に要した時間は約 2 秒であった．

7. おわりに

本稿では，ユーザのメールサービスへの認証を利用した復号鍵発行手法を提案した．提案手法では，DH 鍵共有の利用と，メールのドメイン詐称が検知可能な SPF とメール本文の改ざんが検知可能な DKIM の利用により，IBE での安全な復号鍵発行を実現した．動作確認として，IBE による文字列の暗号化／復号を行うサンプルプログラムを実装し，表 1 の実行環境において 10 秒程度で取得できることを確認した．

参考文献

[1] G. Appenzeller, L. Martin, M. Schertler, “Identity-Based Encryption Architecture and Supporting Data Structures,” RFC5408, 2009-1.
[2] D. Boneh, M. Franklin, “Identity-Based Encryption from the Weil Pairing,” SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003.
[3] D. Boneh, X. Boyen, “Efficient selective-ID secure identity based encryption without random oracles,” EUROCRYPT 2004, LNCS 3027, Springer-Verlag, pp. 223-238, 2004.
[4] M. Wong, W. Schlitt, “Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1,” RFC4408, 2006-4.
[5] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, “DomainKeys Identified Mail (DKIM) Signatures,” RFC4871, 2007-5.
[6] 伴拓也，毛利公美，白石善明，野口亮司，“ActionScript による η_T ペアリング演算ライブラリ，” DICOMO2011，pp.1285-1295, 2011.
[7] H. Torgemane, “as3crypto -Project Hosting on GoogleCode-(online) , “ http://code.google.com/p/as3crypto/ (accessed 2011-12-7).