Vol.2019-IOT-44 No.28
2019/3/8

IDベース暗号を用いた実用的電子メールシステムの設計と実装

奥村 泰久[1,a)]　上原 哲太郎[1,b)]

概要：電子メールの暗号化規格にはS/MIMEなどがあるが、広く使われるには至っていない。それは、これらの暗号化がユーザ毎の鍵管理を必要としているが、このコストが大きすぎるためであると考えられる。そこで我々はIDベース暗号を用いて、ドメインの電子メール管理者のシステム側が鍵管理をすることで、鍵管理のコストを抑えた実用的な電子メールシステムを提案する。本発表ではその設計と実装を述べる。

Design and Implementation of a Practical ID-based Secure E-mailing System

1. はじめに

電子メールは今日においてもインターネット上の主要なメッセージングツールであり続けている．しかし，標準的には暗号化がされておらず，他人からも読み取りが可能なものとして扱われてきた．

電子メールの暗号化の手法としてS/MIME[12]やPGP[9]が提案されているが，今日においてもこれらの手法は一般に広く普及していると言い難い．日本ネットワークセキュリティ協会の調査によると，S/MIMEやPGPを使用している企業は2003年時点で34.0%となっている[11]．この理由として，上記の手法では公開鍵暗号が用いられているが，その暗号化のプロセスが複雑なことや，鍵や証明書の管理コストが大きいことが挙げられる．

ここで，上記の欠点を解決できることが期待される手法としてIDベース暗号（IBE）が提案されている．IBEは，メールアドレスや電話番号などといったユーザ固有の情報を公開鍵として用いる公開鍵暗号方式である．これにより，受信者がわかると同時に公開鍵もわかるようになるため，鍵の所有者を認証する必要がなくなり，証明書を用いた認証システムを用いる必要が無くなる．しかし，IBEはメールシステムを想定して提案された暗号であるにも関わらず，IBEを用いたメールシステムはほとんど普及していない．これはIBEにはユーザ秘密鍵が漏洩した際のリスクが非常に大きいことや，IBEにおいては鍵管理を行うサーバが全てのIDについて秘密鍵を生成できるので結果的に全ての暗号文を復号可能という問題があることが原因であると考えられる．そこで，本研究ではIBEを用いたドメインの電子メール管理者が鍵管理を行うことで，鍵管理コストを抑えた実用的な電子メールシステムを提案し，その実装を行った．

[1] 立命館大学 情報理工学部
College of Information Science and Engineering, Ritsumeikan Univerisy
[a)] yasuhisa@cysec.cs.ritsumei.ac.jp
[b)] t-uehara@fc.ritsumei.ac.jp

2. 背景

2.1 メールの暗号化

電子メールの暗号化には，伝送路の暗号化とメール本文の暗号化がある．このうち伝送路の暗号化は，主にMail Transfer Agent (MTA) 間のSMTPのTLSによる暗号化，およびMTAとMail User Agent (MUA) 間のIMAPやPOPのTLSによる暗号化がある．MUAからMTAには認証にかかるIDとパスワードが送られる例が多いため，暗号化の必要性が高い．MTA間についても，例えばGoogle社の透明性レポートによると，2019年1月時点ではGmailとの間では送受信とも90%以上の通信がTLSにより暗号化されている[5]．

これに対し，メール本文の暗号化は広がっているとは言い難い．メール本文のエンドツーエンドの暗号化を実現する規格にはS/MIMEとPGPがあるが，いずれも多くの人が利用できる環境は整っていない．

2.2 S/MIME

S/MIME は，エンドツーエンドの電子メール暗号化および署名の標準規格である．元は RSA Security 社が提唱したものであったが IETF に委ねられ，現在では S/MIME version 3.2 が RFC5750〜RFC5751 として標準化されている．公開鍵暗号を用いた規格であり，公開鍵の認証は公開鍵基盤に基づいて行われる．

S/MIME では以下のようなフローで暗号化が行われる．

- 証明書の作成
  受信者は事前に証明書を作成する必要がある．証明書は公開鍵に所有者情報などを追加したものに認証局と呼ばれる信頼できる機関からの署名を受けたものである．証明書を認証局から受け取った後，その証明書を送信者に送信する．
- 証明書の検証
  受信者の証明書が正しいものであるかどうかを検証する．受け取った証明書が認証局から正しく署名されていればその証明書を信用する．
- メール本文の暗号化
  本文の暗号化用の鍵を生成し，その鍵を用いて共通鍵暗号でメール本文の暗号化を行う．その後，共通鍵を受信者と送信者の公開鍵で暗号化する．そして暗号文と暗号化鍵をメールに記載して送信する．
- メール本文の復号
  受信者は受信したメールに記載された暗号化鍵を自身の秘密鍵で復号する．その後，復号された共通鍵を用いてメール本文を復号する．

S/MIME は今日では一般には普及しているとはいいがたい．特に，暗号化された電子メールの送信手段としてはほとんど使われていない．この原因としては，電子証明書の管理が煩雑であること，送信者が事前に受信者の証明書を入手しなくては暗号化されたメールを送ることができないことなどが挙げられる．

2.3 PGP

PGP は，1991 年に Philip Zimmermann が公開した暗号ソフトウェアであり，1998 年に OpenPGP として標準化されている．PGP によるメール送信の流れは S/MIME と似ているが，公開鍵の検証に公開鍵基盤ではなく信頼の輪と呼ばれるシステムを用いる．信頼の輪では各ユーザがそれぞれ信頼できるユーザの公開鍵に対し署名をするとともに，交換の結果受け取った公開鍵に，他の信頼できるユーザからの署名があればその公開鍵を信用する．

PGP も S/MIME と同様に今日では一般には普及していない．この原因としては，送信者が事前に受信者の公開鍵を入手することが常に出来るとは限らないこと，鍵や証明書の管理をユーザが行わなければならず煩雑であること，証明書の失効情報の共有が困難なことなどが考えられる．また，S/MIME と比較すると PGP に対応している MUA が少ないことも原因として挙げられる．

2.4 暗号化メールの運用上の課題

S/MIME や PGP によるメールの暗号化は，運用上の課題も多い．例えば，特に企業や組織にとってはメールの SPAM 対策やマルウェア対策は大きな課題であり，現在は多くの場合 MTA で行われているが，エンドツーエンドの暗号化メールは MTA ではその内容がわからないため，SPAM 対策，マルウェア対策とも MUA で行うことになる．各端末の MUA における SPAM 対策，マルウェア対策が正しく動作するようにシステム管理者は管理を徹底する必要があるが，これは大きな管理負荷となる．

さらに，鍵の世代管理も大きな課題である．S/MIME と PGP ではメールが MUA 上で復号されることを想定しているが，IMAP では MTA 上のメールボックスでメールを保存することを想定している．そこで，メールボックスにはメールが暗号化された状態で保存され，閲覧する際に MUA 上で復号するなどの方式がとられる．このとき，証明書の有効期限が切れるなどして鍵の更新が行われた場合に，更新前の鍵で暗号化されたメールが復号できなくなることを防ぐため，MUA 側では過去に受信した暗号化メールを復号するための電子証明書は有効期限経過後も全て保存しておく必要がある．このような管理は煩雑であり，電子メールの暗号化を阻む要因になりかねない．

2.5 DKIM

電子メールにおける認証技術として DKIM がある．S/MIME や PGP はエンドツーエンドの暗号化，認証技術であるが，DKIM はドメインレベルでの認証技術である．

DKIM は S/MIME と同じく公開鍵基盤を用いた方式であるが，鍵の管理はメールサーバが行う．署名は以下のようなフローで行われる．

- 公開鍵の DNS への登録
  メールサーバのドメインの (サービスタイプ)._domainkey サブドメインに公開鍵を DNS の TXT レコードとして登録する．サービスタイプとはその公開鍵が何に使われるかを示す値であり，デフォルトでは*，電子メールでは email となっている．
- メールへの署名
  メールサーバはメールを送信する際にメールの署名を生成する．署名はメールの DKIM-Signature ヘッダとしてメールに記載される．
- 署名の検証
  メールを受信したメールサーバは DKIM-Signature ヘッダの d タグに記載されているドメインの DNS へアクセスし，公開鍵を取得する．その後，取得した公開鍵を元に署名を検証する．

DKIM には作成者署名と第三者署名が存在する．作成者署名では送信者のドメインのサーバが署名をするが，第三者署名ではメールは一度署名用サーバに送信されて署名された後に受信者へ送信される．第三者署名の場合には DKIM-Signature ヘッダの d タグの値に署名用サーバのドメインを設定する．第三者署名は他者が運用している署名サービスを用いるため鍵の管理が不要となり，簡単に利用することができる．しかし，作成者署名ではメールが送信されたドメインを認証できるが，第三者署名ではそのドメインで署名されたことしかわからない．

総務省の調査によると 2018 年 9 月に受信されたメールのうち，DKIM が設定されており認証をパスしたメールの割合は 58.20%となっている [15]．これより，DKIM は普及が進んできていると考えられる．

DKIM は鍵の管理をメールサーバが行うため，メールサービスの管理者が準備をすればユーザは気にせず利用することができる．しかし，どのドメインから送信されたメールかは検証できるが，本当にその送信者が送信したメールであるかは検証できない．また，第三者署名を用いた場合には署名されたドメインの認証しかできず，送信ドメインの認証ができなくなる．さらに，暗号化に関してはサポートされておらず，メールの内容を保護することはできない．

2.6 IBE

IBE は Adi Shamir が 1985 年に提案した公開鍵暗号方式である [13]．IBE の特徴は公開鍵を任意の値に設定することが可能な点である．これにより公開鍵をメールアドレスや電話番号といった，個人に固有な情報 (ID 情報) に設定することで公開鍵の所有者の認証が不要になる．ユーザの秘密鍵は鍵生成局と呼ばれる第三者がユーザの ID 情報から生成するが，送信者は受信者が秘密鍵を生成しているかに関わらず暗号文を送信することができる．

IBE による暗号化は以下のようなフローで行われる．

- セットアップ
  鍵生成局はマスター鍵を生成し，それをもとに公開パラメータを生成する．
- 鍵生成
  鍵生成局はユーザから秘密鍵リクエストを受けると，ユーザの ID 情報とマスター鍵からユーザ秘密鍵を生成する．
- 暗号化
  送信者は鍵生成局の公開パラメータと受信者の ID 情報を用いてメッセージを暗号化する．
- 復号
  受信者は受け取った暗号文を自身のユーザ秘密鍵を用いて復号する．

IBE にはいくつかの問題点が存在する．

ひとつは，鍵の失効が困難なことである．IBE において公開鍵は自身の ID 情報であるため，鍵を失効することは ID 情報を失効することとなり困難となる．鍵を失効したい場合にとれる方法としては，ID を変更することやシステムが用いるパラメータを変更することが挙げられる．しかし，前者は変更後に他のユーザへの再認知が必要であり，後者はシステム全体での対応が要求される．そのため，ユーザ秘密鍵の漏洩は公開鍵暗号以上に大きな問題となる．

もうひとつは，鍵生成局が全てのユーザ秘密鍵を入手可能なことである．IBE では鍵生成局がユーザ秘密鍵を生成するため，全てのユーザ秘密鍵を自分で生成することができる．そのため，鍵生成局は全てのユーザへの暗号文を復号することが可能である．

2.7 関連研究

Kihidis らは Registration Service(RS) を設置することによる IBE を用いたメールシステムの実装を行った [6]．RS はユーザの登録や管理を行い，鍵生成局は RS と共有されたデータベースを用いてユーザ認証を行う．このシステムでは各ユーザが秘密鍵を管理し，復号も自身で行う必要がある．

Chen らはドメインごとにプロキシを設置することによる IBE を用いたメールシステムの実装を行った [4]．プロキシがユーザのメール送信を代行し，暗号化や復号などもプロキシが行う．ただし，プロキシはユーザの秘密情報を扱うため，そのセキュリティを強化するべきとしている．このシステムではユーザは秘密鍵の管理や復号を行う必要はないが，プロキシサーバを準備しそれを通してメールの送受信を行う必要がある．

ID ベース暗号を実用化したサービスとして Voltage SecureMail が存在する [10]．有償サービスになっており，専用アプリや既存のメールクライアントのアドオンなども存在する．メールの送信はクライアントで行えるが，暗号化メールの復号は Voltage の Web ページで行う．このシステムは有償のため，誰もが利用できるものではない．また，ユーザは秘密鍵の管理を行う必要はないものの，メールを読むのに復号を依頼する必要がある．

本研究では，ユーザが秘密鍵の管理や復号などを行う必要がなく，各メールサーバ上で運用可能な IBE メールシステムを提案する．

3. 提案手法

3.1 概要

本研究では IBE を用いた実用的な電子メールシステムを提案する．本システムではドメインごとに IBE システムを構築する．各 IBE システムにはユーザ，鍵生成局，メールサーバが存在する．

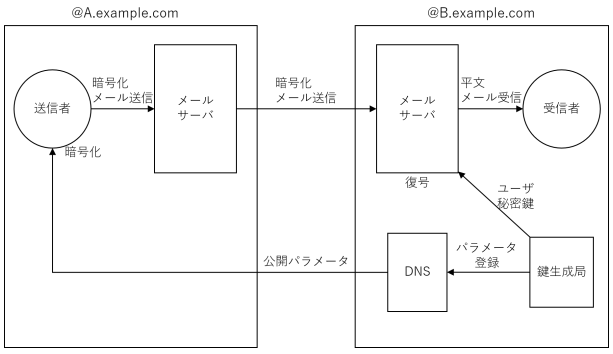

図 1 提案手法による暗号化
Fig. 1 Encryption by proposed method.

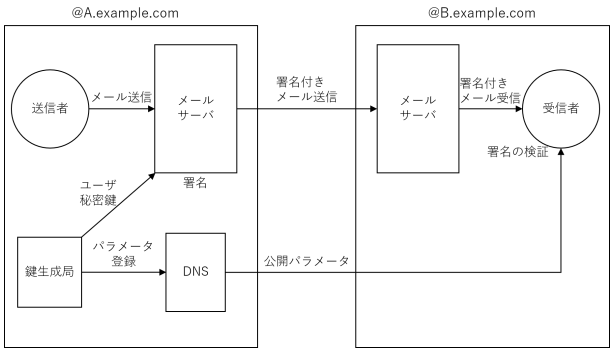

図 2 提案手法による署名
Fig. 2 Signature by proposed method.

3.1.1 暗号化

暗号化は以下ようなフローで行われる．暗号化メールを送信する様子を**図 1** に示す．

- セットアップ
  鍵生成局はマスタ鍵を生成し，それを用いて公開パラメータを生成する．公開パラメータは自身のドメインの DNS レコードに記載することで公開する．
- 鍵生成
  鍵生成局はマスタ鍵とユーザのメールアドレスを用いてユーザ秘密鍵を生成する．鍵生成リクエストはメールサーバが行い，ユーザ秘密鍵もメールサーバが管理する．
- 暗号化
  送信者は受信者のメールアドレスのドメインの DNS サーバから公開パラメータを取得する．その後，送信者が取得したパラメータと受信者のメールアドレスを用いてメールを暗号化し，送信する．
- 復号
  受信者側のメールサーバは受信した暗号化メールを受信者の秘密鍵を用いて復号する．復号したメールはユーザのメールボックスに保存する．

暗号化において受信側のメールサーバと受信者の間では平文のメールが送信される．そのため，この通信には IMAP4s や POP3s を使用する．

3.1.2 署名

署名の場合は，セットアップ，鍵生成を暗号化と同様に行い，以下のように署名と検証を行う．ただし，生成するパラメータは署名アルゴリズムに用いるもののため暗号化で生成するパラメータとは異なる可能性がある．署名付きメールを送信する様子を**図 2** に示す．

- 署名
  送信者側のメールサーバは送信するメールに対し，送信者の秘密鍵を用いて署名を行ってから送信する．
- 検証
  受信者はメールを受信すると送信者のメールアドレスのドメインの DNS から公開パラメータを取得する．取得したパラメータと送信者のメールアドレスを用いて署名を検証する．

署名ではメールサーバがユーザ秘密鍵を用いて署名をするが，ユーザ秘密鍵を使用するには送信者の認証が必要になる．また，通信路での改竄などの危険性も存在する．そのため，送信者からメールサーバへの通信は SMTP Auth と SMTPs を用いる．

3.2 議論

3.2.1 運用

このシステムでは，鍵生成局とメールサーバは同一のサーバ上で運用することを想定している．これはユーザ秘密鍵を送信する通信路を確保できることや，鍵生成におけるユーザ認証を SMTP Auth や POP，IMAP の認証機能を用いてメールと同時に行えることが期待できるためである．鍵生成局の運用コストは DKIM と比較すると，ユーザ管理はメールと同時に行えることから，守る対象がマスター鍵とユーザ秘密鍵に増えた分増大すると考えられる．しかし，ユーザ秘密鍵はマスター鍵と同じ場所で保管できることや，保管せずに必要に応じて生成することもできることから，鍵生成局の運用コストは DKIM の運用コストと大きく変わらないと言える．そのため，この想定は妥当なものであると思われる．

マスター鍵やユーザ秘密鍵は公開鍵暗号における秘密鍵と同様に，時間経過により危殆化する．そのため，危殆化が心配される場合には新しくパラメータを生成し直す．その際に DNS に設定している公開パラメータを変更する必要があるが，変更がインターネット全体に反映されるには多少の時間がかかる．また，パラメータ更新からしばらくの間は前のパラメータを使用しているメールが多く存在しているため，それらに対応する必要がある．そのため，パラメータ更新期間を設けその期間中は新旧の公開パラメータを並行して公開する．また，旧パラメータのユーザ秘密鍵も更新期間は破棄せずに保持しておく必要がある．

3.2.2 安全性

このシステムでは IBE を用いているため，ユーザ秘密鍵

が漏洩した際の鍵失効が困難である．そのため，ユーザ秘密鍵は厳重に保管されるべきである．このシステムにおいてユーザ秘密鍵を扱うのは鍵生成局やメールサーバといったシステム管理者側のエンティティのみである．これにより，ユーザ秘密鍵の漏洩の危険性を低減させられることが期待できる．

IBE には鍵生成局が全ユーザの秘密鍵を入手可能であるという問題が存在する．このシステムではドメインごとに鍵生成局を設置することにより，秘密鍵を入手可能なユーザの範囲を自身のドメインのユーザのみに制限している．

このシステムではメールを受信した際にメールサーバが鍵生成局からユーザ秘密鍵を取得し，メールの復号を行う．そのため，完全なエンドツーエンドの暗号化にはなっていない．このような形式をとったのは鍵生成局とメールサーバの管理者が同一であるという想定でシステムを設計しているためである．鍵生成局とメールサーバの管理者が同一であるならメールサーバが復号を行っても良いと考え，メールサーバ上でメールを復号する形式をとった．この形式をとることによりユーザがメールの復号を行う必要がなくなることで負担が軽減されるとともに，メールサーバで SPAM やマルウェアチェックを行うことも可能となる．

署名に関しては，送信者側のメールサーバが署名し受信者側で検証するというモデルは実質的には DKIM と同じ機能しか提供できない．しかし，メールサーバの協力がなければ送信者の偽装やメール内容の改竄は難しいため，メールサーバの信用のもとでは送信者の認証ができると言える．

エンドツーエンドでの暗号化を重視する環境であれば，メールサーバと鍵生成局を分割して運用することも可能である．この場合，メールサーバはメールを暗号化したまま保存し，ユーザはメールを受信した後鍵生成局にメールを渡して復号してもらうという形式がとれる．しかし，鍵生成局でユーザ認証を行う必要があることや，IMAP を用いる場合には鍵生成局で失効後の鍵も管理しておく必要があるなどの点に注意する必要がある．

3.2.3 事故対応

システムを運用する上で発生するであろう事故と発生した際に考慮すべきことや対応を述べる．

- マスター鍵の漏洩
  マスター鍵が漏洩した際には，ドメイン内の全てのユーザの秘密鍵をマスター鍵から生成可能なため，あらゆるユーザへ向けた暗号メールが復号可能になる．この際にはパラメータを新しく生成し直し，DNS への登録とユーザ秘密鍵の再生成をする必要がある．また，全ユーザに現在のパラメータで暗号化されたメールが第三者に復号された可能性があることを伝えるべきである．
- ユーザ秘密鍵の漏洩
  ユーザ秘密鍵が漏洩した際には，そのユーザへ送信されたメールのみが復号可能になる．しかし，公開鍵の失効が困難であるという IBE の性質のため，漏洩した鍵のみを失効することが難しい．そのため，マスター鍵が漏洩した際と同様にパラメータを生成し直すことですべての鍵を失効する．また，秘密鍵が漏洩したユーザに現在のパラメータで暗号化されたメールが第三者に復号された可能性があることを伝えるべきである．
- ユーザパスワードの漏洩
  ユーザパスワードが漏洩するとユーザのメールボックスへのアクセスを許すことになり，メールは復号されて保存されているためそれらを読まれることになる．しかし，ユーザは自身の秘密鍵を知ることはできないため，秘密鍵が漏洩することはない．そのため，パラメータの更新などは必要なく，ユーザにパスワードの変更を依頼すればよい．

4. 実装

今回の実装では提案手法における暗号化の部分の実装を行った．ソースコードは付録に記載する．

4.1 使用技術

実装にあたり，まず使用する IBE のアルゴリズムを決定した．DNS レコードにパラメータを公開することから，公開パラメータは少ない方が好ましいことや，実装の容易性の観点からアルゴリズムがシンプルなことを条件に IBE アルゴリズムを探した結果，BFIBE[3] と BB2IBE[2] が候補として見つかった．BFIBE は 2001 年に Boneh と Franklin によって提案された IBE である．アルゴリズムやパラメータが非常にシンプルであり，ランダムオラクルモデルにおいて選択暗号文攻撃に耐性を持つ．BB2IBE は 2011 年に Boneh と Boyen によって提案された IBE である．こちらは BFIBE と比較するとアルゴリズムやパラメータは多少複雑になるが，スタンダードモデルにおいて選択平文攻撃や選択 ID 攻撃に耐性を持つ．どちらのアルゴリズムでも実装とパラメータサイズに問題はないと考え，選択 ID 攻撃に耐性を持つ BB2IBE を使用するアルゴリズムとして決定した．

BB2IBE の計算にはペアリングと呼ばれる演算が必要となってくるため，ペアリング計算用のライブラリを決定した．ペアリングライブラリには PBC[8]，TEPLA[7]，mcl[14] などがあったが，実行速度が高速なことや扱いやすさから mcl を用いることにした．ライブラリの決定に伴い，開発に使用するプログラミング言語は C++とした．

共通鍵暗号やハッシュ関数には OpenSSL ライブラリを用い，共通鍵暗号方式には認証付き暗号を用いるのが良いと考え，AES-256-GCM を用いた．

また，電子メールにおける MIME データ形式を扱うラ

```
ver=1.0; a=bb2; c=bn256; n=1;
g=1 BaQYXQLIUfpEMoTf7EUgLxPeT2W5GmtWt9vWssXAz6o=
JBf9Huo+s9/AE0hRDNuwDVeycX/eO/gnmZ2yqA/plxY=;
X=1 Asj4ANHV9sjyMZ1RGQ7wdZpM6s3fGASw87KSlUG8Fu0=
DgJWQju9BmTlJ5/ojr1SkF+H/blvoopC5QP/1s2lm1U=;
Y=1 FDYKGQcxSoX24MMjDu+taHGFcVt7KVhd5/JvUsGOkCs=
A3wcyC5gPnKeDm6uAv45jffxXcjvCuyOlMwJhVqUdRU=;
v=FG65FYLn15SCNV9YVkcpcFOgTy3PJJ3ojBy9WY/YPk4=
GJWAoM7FjxmJ3GoOipL0EUczwsZUAWih/pHycoBt8r4=
GHIEYZbKLhIswcXtdTgK9amm/wYNUruRgwvFpcVdlHY=
Alpt951FSXzh01vGt588QNuJDW80RveOB08UBQgCxpg=
B9Jk7iHoymgjRObZcPBm89hpo7MXFZay6PnBIJdmabA=
IwNcCClwYtxzTOVdxZpbq7e60yN+nB/pFuy+jl3b54o=
DKDEnKSeOgUpbLZm45BnWfwAfvPKJlFgUcFJuADLOL0=
F4mBW6Y99hfHw6p9qplxccVjecRMgDHAf1TYXfC2ycs=
HZuSMyHOi6xfFqYGmo3H9BezDl4haX6t5V2T/7WxXKk=
lqd9NmSs6w8831GXMHFabw7qCGdVXFBu+kITC5d9mXI=
GxDpkG29B9pgpgAFieaWwY/l2QBx0+5NB1dI01rwYX0=
HNquFaOpTq2Kb2k2hBleiRDA3KDGnaJw23LJrSN3GXg=;
```

図 3 公開パラメータの例
Fig. 3 Example of public parameters.

イブラリとして mimetic[1] を用いた．

4.2 仕様

実装したプログラムの仕様を述べる．

作成したプログラムは kgc，encrypter，decrypter の 3 つである．

kgc は鍵生成局の動作を行うプログラムである．オプションに-s を指定するとセットアップが実行され，ibemail.param ファイルに生成されたマスター鍵を base64 エンコードしたものと公開パラメータを DNS に記載する形式にしたものが保存される．BB2IBE に用いるパラメータの多くはいくつかの数の組で表されるデータであるため，その base64 エンコードは各数を base64 エンコードし，それらを空白で繋げた形式とした．DNS レコードに記載する公開パラメータの例を**図 3** に示す．公開パラメータは複数のタグによって構成され，タグは; で連結される．各タグの内容は**表 1** に示す．公開パラメータは自身のドメインの_ibemailkey サブドメインに TXT レコードとして記載する．この際，DNS の TXT レコードには最大 255 文字までしか連続で設定できないが，公開パラメータの文字列はそれよりも長いため複数に分割して登録した．また，オプションに-g を指定し，続けて ID を入力すると，入力された ID に対応したユーザ秘密鍵を生成する．この際，マスター鍵と公開パラメータは ibemail.param ファイルに保存されている値を用いる．生成されたユーザ秘密鍵は (入力された ID).userkey ファイルに base64 エンコードされて保存される．

encrypter は標準入力に与えられたメール全体を暗号化し，暗号化したメールを送信するプログラムである．この際，暗号化に使用する ID 情報には To ヘッダのアドレスの@以前を用い，公開パラメータは@以降のドメインの DNS から取得する．また，共通鍵，追加認証データはランダムに生成され，初期化ベクトルは UNIX 時間とランダムデータを結合することで作成される．生成される共通鍵，追加認証データは 256bit，初期化ベクトルは 92bit で前 64bit が UNIX 時間となっている．暗号文は**図 4** のような形式で表される．暗号化メールのヘッダは**図 5** のように表される．このうち，From，To ヘッダの内容は入力されたメールのものを使用する．このメールの本文に暗号文を base64 エンコードした値を記載したものを送信する．メールの送信には sendmail コマンドを使用している．

decrypter は標準入力に与えられた暗号化メールを復号するプログラムである．復号に用いるユーザ秘密鍵は ~/.ibemail/decryptkey に保存されているものを使用する．復号したメールは，受信した時間を yyyyMMddHHmmss 形式で表したファイル名で~/MailDir 下に保存する．また，復号された印として，Ibemail-Decrypted: true というヘッ

表 1 公開パラメータタグ
Table 1 Public parameter tag.

ver=	システムのバージョンを示す．論文執筆時点では 1.0 である．
a=	使用するアルゴリズムを示す．BB2IBE は bb2 で示す．
c=	使用する曲線を示す．今回は mcl ライブラリが BN254 として定義している曲線を bn256 として用いた．
n=	パラメータの番号を示す．パラメータを更新する際にはこの値を増やす．
g=	BB2IBE の公開パラメータのうちの g の値を base64 エンコードで示す．
X=	BB2IBE の公開パラメータのうちの X の値を base64 エンコードで示す．
Y=	BB2IBE の公開パラメータのうちの Y の値を base64 エンコードで示す．
v=	BB2IBE の公開パラメータのうちの v の値を base64 エンコードで示す．

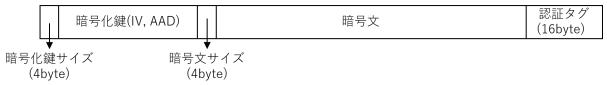

図 4 暗号文形式
Fig. 4 Ciphertext format.

```
From: sender@A.example.com
To: recipient@B.example.com
Subject: encrypted mail
Mime-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
      filename="ibemail.cipher"
Content-Type: application/ibemail-encrypted
```

図 5 暗号化メールのヘッダ
Fig. 5 Header of encrypted e-mail.

```
:0 Hc :
* ^Content-Type: application/ibemail-encrypted
| /home/okumura/.ibemail/decrypter
:0
```

図 6　使用した.procmailrc
Fig. 6　Used .procmailrc.

```
From: okumura@sakura.cysec-lab.org
To: okumura@sub1.cysec-lab.org
Date: Mon, 28 Jan 2019 10:47:11 +0900 (JST)
Subject: testMail

It is a test mail.
```

図 7　encrypter に渡したメール
Fig. 7　E-mail entered in encrypter.

```
Return-Path: <okumura@sakura.cysec-lab.org>
X-Original-To: okumura@sakura.cysec-lab.org
Delivered-To: okumura@sakura.cysec-lab.org
Received: by sakura.cysec-lab.org (Postfix, from userid 1006)
    id 6E7FBA238C; Mon, 28 Jan 2019 11:42:55 +0900 (JST)
From: okumura@sakura.cysec-lab.org
To: okumura@sub1.cysec-lab.org
Subject: encrypted mail
Mime-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
    filename="ibemail.cipher"
Content-Type: application/ibemail-encrypted
Message-Id: <20190128024255.6E7FBA238C@sakura.cysec-lab.org>
Date: Mon, 28 Jan 2019 11:42:55 +0900 (JST)

vgAAAK7IResqVbIU/dZaCpkqMLrL1s8rXR7S0DvtehB4VYzhijfu5CxrX2pB4/ykOXXYt4vDuLx9
PeP6ndwrvgENFDmAzRnPmmS81Ec2czcstwuVeC2jGPU7rMUEIV1nDB3SqrB1wr2yoYSpf4I0Sbr+
olHnB77745A1QaviJ5ogAQWEJAe4kQKAy8fZPWaTI1Uph9HgOnxE5/J22kKdqgEcElxx8ke2luaO
Xp+cn/5lM6VlVlxH5pDe1ow14Az0XKDRAAAA8jmyocSrWGYMYF3wKu+sWLgs6KP7ilKsc6XFU6S0
IHNWUVBtdPnBbFaTYRIw/iUgpPxGe2NHgQFrq3KhERUcQ1fQd9086HVAi3eQyWThUF8h2kqVwBgX
hzpPl8xzkmnzeiKiiRz3bngmr3fk5m9bWpWP2xl0TBZyLNYTPqB3MeCdjGHWVzo0pF/SuNahJcJq
6yYo3mF9dTRV5kSKTSTd4klZiRwudoUAi8aDYDg3y/3PaDXhcwQS6esRc+4z/PWqYaiPQ+PpfBq9
hKjGG0UxjzUa7FEI38UTj40YUSznJovH
```

図 8　受信した暗号化メール
Fig. 8　Received encrypted e-mail.

ダを付加する．このプログラムは.forward や procmail により呼び出されることを想定している．今回用いた.procmailrc を**図 6** に示す．ここでは decrypter プログラムを ~/.ibemail/decrypter に配置している．

実行は postfix が動作している sub1.cysec-lab.org ドメインのメールサーバにユーザ okumura を作成し，/home/okumura/.ibemail 下に decrypter プログラムを配置して行った．この際，公開パラメータは事前に DNS へ登録しておき，ユーザ秘密鍵も事前に生成し/home/okumura/.ibemail/decryptkey に配置しておいた．encrypter を用いて okmura@sakura.cysec-lab.org から okumura@sub1.cysec-lab.org 宛にメールを送信した結果，/home/okumura/Maildir 下に送信されたメールと同じメールが保存された．encrypter に渡したメールを**図 7**，サーバで受信した暗号化メールを**図 8**，/home/okumura/Maildir 下に保存されたメールを**図 9** に示す．メールサーバが付加したヘッダが失われてしまっているが，これは実装の簡単化のためヘッダを含むメール全体を暗号化し，それを復号した結果を保存しているためである．

今回の実装では提案手法による暗号化，送信，復号の流れが既存のメールシステム上で動作することが確認できた．しかし，パラメータ更新時の動作などは実装できておらず確認ができなかった．

```
Ibemail-Decrypted: true
From: okumura@sakura.cysec-lab.org
To: okumura@sub1.cysec-lab.org
Date: Mon, 28 Jan 2019 10:47:11 +0900 (JST)
Subject: testMail

It is a test mail.
```

図 9　保存されたメール
Fig. 9　Stored e-mail.

5. 評価

この章では提案手法の評価を述べる．

提案手法では, IBE を用いたことによって事前に公開鍵を送信する必要がなくなっている．事前の鍵送信は S/MIME や PGP において問題となっており，これをなくすことができるのは大きな利点であると考えられる．鍵送信が不要になっているかわりに公開パラメータの送信が必要となるが，これは DKIM と同様に DNS に公開パラメータを記載することで解決されている．

また，ユーザは送信時に IBE による暗号化を行うだけで暗号メールを利用することができる．S/MIME や PGP では，ユーザは事前に鍵と証明書を生成する必要があることや，鍵と証明書をユーザ自身で管理する必要があり，利用する上でユーザへの負担が大きかった．IBE による暗号化に対応したメールクライアントを用意するだけで利用可能となるため，既存手法に比べユーザが利用しやすい形式となっている．

本提案では IBE を用いているため，ユーザ鍵の失効が困難な点に注意を払う必要がある．S/MIME や PGP では対応する証明書を証明書失効リストに登録，公開することで鍵失効が可能だが，IBE では公開鍵を失効することが難しいため，ID（つまりメールアドレス）を変更するか，システムパラメータと各ユーザ秘密鍵を生成し直すなどといった対応が必要である．そのため，ユーザ秘密鍵の漏洩には注意が必要がある．本研究の提案では，ユーザ秘密鍵を各ユーザが管理する [6] の手法と比較し，システム管理者が集中管理することでユーザ秘密鍵が漏洩する危険性を低減している．

公開パラメータは DNS を用いて公開しているため，DNS スプーフィングなどにより内容が差し替えられる可能性がある．送信者が受信者の公開パラメータを取得する際にその公開パラメータを攻撃者のものに差し替えることで，中間者攻撃が可能となる．この攻撃のイメージを**図 10** に示す．この対策としては公開パラメータ取得に DNSSEC や

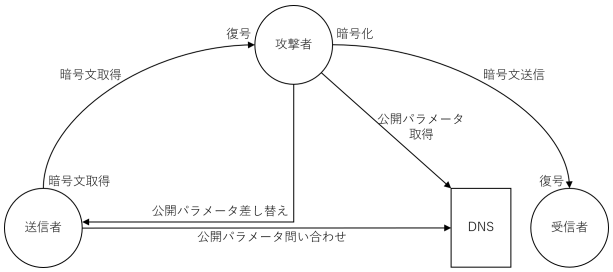

図 10 DNS 差し替えによる中間者攻撃
Fig. 10 Man in the middle attack by DNS spoofing.

DNS over HTTPS を用いることが挙げられる．しかしこれらの手法が利用できない場合でも，暗号化したメッセージに署名をすることでこの攻撃を困難にすることが可能である．この場合には送信者側と受信者側の公開パラメータの両方を攻撃者のものに差し替えなければ検証と復号が正常に通らなくなるためである．

ドメインごとに鍵生成局を設置することで，鍵生成局がユーザ秘密鍵を取得可能なユーザの範囲を制限している．これにより，鍵生成局を全体に 1 つ設置する [4], [6], [10] と比較して鍵生成局の権限は小さなものとなっている．また鍵生成局が管理するユーザはドメイン下のユーザのみのため，鍵生成局のユーザ管理コストも縮小できていると言える．さらに，各ユーザの秘密鍵は鍵生成局が全て管理しているが，このことを利用してそのドメインに届くメールの SPAM 対策およびマルウェア対策をシステム側で集中して行うこともできる．

6. おわりに

本研究では，IBE を用いたドメインの電子メール管理者が鍵管理を行うことで，鍵管理コストを抑えた電子メールシステムを提案し，その暗号化部分の実装を行った．ユーザ秘密鍵の管理を管理者が行うことでユーザは鍵の管理を行う必要がなくなり，IBE において大きな問題となるユーザ秘密鍵の漏洩の危険性を低減させることができる．また，鍵生成局をドメインごとに設置することにより，鍵生成局がユーザ秘密鍵を取得できるユーザの範囲に制限がかかり，ユーザ管理のコストも低減している．暗号化や署名は完全にはエンドツーエンドになっていないがメールサーバによるスパムチェックなどの需要に対応することができる．実装では提案手法の暗号化部分の実装を行い，現在利用されているメールシステムの上で暗号化が実行可能であることが確認できた．しかし，パラメータ更新などの動作は確認できなかったため，今後実装して確認したい．また，署名部分の実装はできなかったため，今後署名部分の実装も行えたらよいと考えられる．

参考文献

[1] Barbato, S.: mimetic, C++ MIME Library, (online), available from ⟨http://www.codesink.org/mimetic_mime_library.html⟩ (accessed 2019-1-24).
[2] Boneh, D. and Boyen, X.: Efficient Selective Identity-Based Encryption WithoutRandom Oracles, *Journal of Cryptology*, Vol. 24, No. 4, pp. 659–693 (online), DOI: 10.1007/s00145-010-9078-6 (2011).
[3] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, *Advances in Cryptology — CRYPTO 2001* (Kilian, J., ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 213–229 (2001).
[4] Chen, T. and Ma, S.: A Secure Email Encryption Proxy Based on Identity-Based Cryptography, *2008 International Conference on MultiMedia and Information Technology*, pp. 284–286 (online), DOI: 10.1109/MMIT.2008.96 (2008).
[5] Google: 透明性レポート配信中のメールの暗号化，Google（オンライン），入手先 ⟨https://transparencyreport.google.com/safer-email/overview⟩（参照 2019-2-4）.
[6] Kihidis, A., Chalkias, K. and Stephanides, G.: Practical Implementation of Identity Based Encryption for Secure E-mail Communication, *2010 14th Panhellenic Conference on Informatics*, pp. 101–106 (online), DOI: 10.1109/PCI.2010.48 (2010).
[7] Laboratory of Cryptography and Information Security: TEPLA - LCIS, University of Tsukuba, University of Tsukuba (online), available from ⟨http://www.cipher.risk.tsukuba.ac.jp/tepla/license.html⟩ (accessed 2019-1-24).
[8] Lynn, B.: PBC Library - Pairing Based Cryptography - About, Stanford University (online), available from ⟨https://crypto.stanford.edu/pbc/⟩ (accessed 2019-1-24).
[9] M. Elkins, D. Del Torto, R. L. and Roessler, T.: RFC 3156 - MIME Security with OpenPGP, IETF (online), available from ⟨https://tools.ietf.org/html/rfc3156⟩ (accessed 2019-1-18).
[10] Micro Focus: Email Encryption Security Solution, Secure Email, Micro Focus (online), available from ⟨https://www.microfocus.com/en-us/products/email-encryption-security/overview⟩ (accessed 2019-1-18).
[11] NPO 日本ネットワークセキュリティ協会：情報セキュリティ被害調査ワーキンググループ活動発表，NPO 日本ネットワークセキュリティ協会（オンライン），入手先 ⟨https://www.jnsa.org/houkoku2003/20040518/303_6.pdf⟩（参照 2019-1-26）.
[12] Ramsdell, B. and Turner, S.: RFC 5750 - Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling, IETF (online), available from ⟨https://tools.ietf.org/html/rfc5750⟩ (accessed 2019-1-18).
[13] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes, *Advances in Cryptology* (Blakley, G. R. and Chaum, D., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 47–53 (1985).
[14] 光成滋生：herumi/mcl: a portable and fast pairing-based cryptography library, (online), available from ⟨https://github.com/herumi/mcl⟩ (accessed 2019-1-24).
[15] 総務省：電気通信消費者情報コーナー — 迷惑メール対策，総務省（オンライン），入手先 ⟨http://www.soumu.go.jp/main_sosiki/joho_tsusin/d_syohi/m_mail.html#toukei⟩（参照 2019-1-20）.