ホスト間直接通信型ファイル配送システムのための インスタントな宛先 ID とホストだけが管理する秘密を用いた ペアリングによる暗号通信

疇地 悠 † 毛利 公美 † 白石 善明 ‡ 土井 洋 ††
岐阜大学 † 名古屋工業大学 ‡ 情報セキュリティ大学院大学 ††

1 はじめに

複数の組織間でファイルを送受信する場合，組織外部のファイル配送サービスを利用することがある．しかし，そのようなサービスを利用する場合組織外のファイルサーバに情報が蓄積されることになるため，そこからの情報漏洩が懸念される．その解決策の一つとして，ファイル配送サーバを介さずに直接ホスト間でファイル配送を行う方法が考えられる．その場合，直接通信のための宛先情報の取得や通信路を暗号化するための秘密情報の生成／保管，暗号鍵の配送方法等，いくつかの検討すべき課題がある．

本研究では，ホスト間直接通信型ファイル配送システムを安全に運用するためのモデルを与え，そのモデルに適合する暗号化方式を提案する．

2 安全なホスト間直接通信型ファイル配送システムのモデル (提案モデル)

2.1 安全なファイル配送システムのコンセプト

想定するホスト間直接通信のモデルでは，ファイルの要求者 A がファイルの保持者 B に対して送信要求を出す場合，この要求情報（要求元：A の ID，要求先：B の ID）のみを接続仲介サーバが保持する．この情報を参照した B は，接続仲介サーバから，要求元となる A の情報を入手し，直接 A に対してファイルを送信する．

提案するモデルでは，宛先 ID を用いた暗号化通信によって安全なファル配送を実現する．一般に，宛先として利用できる ID は，ユーザ ID とホスト ID が考えられるが，ホスト ID を用いて暗号化する場合は，どのホスト間で暗号通信が行われたかが暗号文から追跡可能になる．このことは，社外へのファイルの持ち出し等，どのホストにファイルが保存されたのか，ファイルがどのように散らばったのかを管理者が把握できること意味しており，機密情報管理という観点からユーザ ID を用いるよりも有用である．

しかし，その反面，図 1 に示すように接続仲介サーバがファイル要求元／要求先の両方の ID を保持するため，もし，攻撃者にこのリストを入手されるとファイルの所在が漏洩することになる．これを避けるためには，ホスト ID をシステム起動ごとに変更し（ID 発行サーバの役割），接続仲介サーバから情報が漏洩した場合でも，ID 発行サーバと結託しない限りファイルの所在が漏洩しないようにすればよい．これにより，ホスト ID が漏洩しても攻撃者は実際に通信したホストを特定することができず，ファイルを保持しているホストへの直接的な攻撃を回避できる．一方，すべてのサーバ・ユーザ・ホストから情報を取得可能な権限を有する管理者は，ID 発行サーバの情報，接続仲介サーバの情報，ユーザの情報，暗号文を用いることで，ファイルの所在を追跡することができる．

以上のことから，提案する安全なファイル配送システムのモデルに対するコンセプトを次の 3 つにまとめる．

[コンセプト 1] ホスト ID を用いた暗号化
[コンセプト 2] ホスト ID をシステム起動ごとに変更する
[コンセプト 3] ホスト情報とユーザ ID を個別にホスト ID と紐付ける

2.2 コンセプトを満たすホスト間直接通信型ファイル配送システムのモデルに対する要件

2.1 節で示したコンセプトを満たすファイル配送システムをモデル化する際の暗号化に関する課題とそれを解決するための要件を以下に示す．

(課題 1) 暗号化に関する秘密情報を仲介サーバに保持させた場合，以下のような不都合が生じる．
1) その秘密情報を使ってサーバ側でファイルの内容が復元できてしまい，情報漏洩につながる．
2) ホストの ID が変更されるたびにサーバが全ての秘密情報を生成しなおす必要がある．

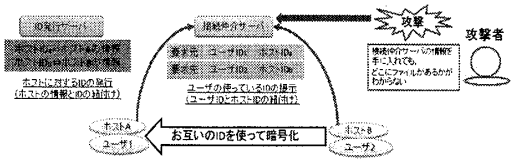

図 1: 安全なファイル配送システムのコンセプト

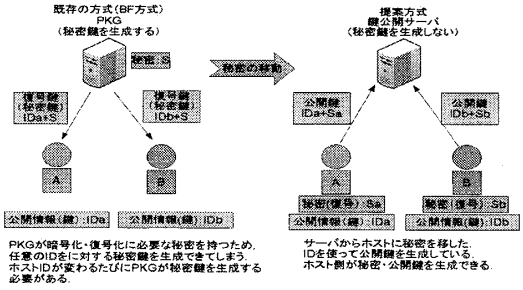

図 2: 提案暗号化方式のコンセプト

[要件 1] 暗号化に関する秘密情報を全てホスト側に移し，ホストだけで秘密情報を生成・管理できること．

(課題 2) 通信の宛先を表すホスト ID を用いて簡便に暗号化でき（コンセプト 1）かつ，管理者によってファイルの配送先が追跡できる暗号化方式が必要になる．

[要件 2-1] ホストが自身のホスト ID に対応した公開鍵（ホスト ID とユーザの秘密情報を用いた鍵）を生成できること．

[要件 2-2] ホストが自身の秘密鍵を生成できること．

(課題 3) システム起動ごとにホスト ID が変更されても暗号化できる方式が必要になる（コンセプト 2）．

[要件 3] ホスト ID が変更されるたびにホストが自身の公開鍵を生成できること．

2.3 提案するシステムモデル

2.2 節に示した要件を満たすシステムモデルを図 3 に示す．また，提案モデルにおける各エンティティの能力を以下のように定義する．

[ユーザ管理サーバ (UserManagementServer)]
正しいユーザのアカウントの登録と，登録されたユーザを認証してシステムへのアクセス許可証を発行する処理を行う．このときユーザアカウントは不正に改変されない．

[ID 発行サーバ (IdentityIssueServer)]
正しいユーザの使用するホストに ID を発行する．この ID はホストを一意に特定できるものであり，不正に改変されない．

[接続仲介サーバ (ConnectedMediationServer)]
ホストを認証し，接続要求を受け付けて公開する．このとき接続用情報は不正に改変されない

[長期鍵公開サーバ (LongTermKeyServer)]
アカウント登録時にユーザの長期鍵を登録し，登録された長期鍵の公開を行う．このとき長期鍵は不正に改変されない．

[短期鍵公開サーバ (ShortTermKeyServer)]
ユーザの短期鍵の登録と公開を行う．

[ユーザ（要求側/所持側）]
ユーザ管理サーバに登録されたユーザであり，秘密情報は漏洩しない．正しく認証が行われたユーザを正しいユーザとする．

[ホスト（要求側/所持側）]
システムが正しくインストールされた端末であり，正しいユーザに使われている．

[サーバとの通信]
ユーザ管理サーバとの通信は安全に行われ，通信情報の盗聴も入れ替えも行われない．また，IIS・LTKS・CMS との通信内容は入れ替えられない．

An Encryption Method Using Pairing for File Delivery System by End-to- End Communication
† Haruka Azechi and Masami Mohri·Gifu University ‡ Yoshiaki Shiraishi·Nagoya Institute of Technology †† Hiroshi Doi ·Institute of Information Security

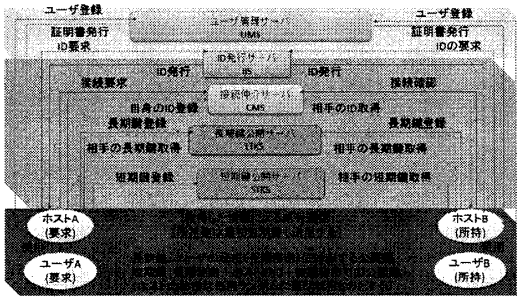

図 3: 提案するシステムモデル

3 宛先 ID を用いたホスト間直接通信の暗号化方式

本節では，2.3 節に示した宛先 ID を用いたホスト間直接通信型ファイル配送システムのモデルに適した暗号化方式を提案する．提案する暗号化方式では，ペアリングと呼ばれる写像関数を用いる．

3.1 ペアリング

$\mathbb{G}_1, \mathbb{G}_2$ を位数 q の群とし，その生成元をそれぞれ P,Q とする．このとき，ペアリング $\hat{e}(\mathbb{G}_1, \mathbb{G}_1) \rightarrow \mathbb{G}_2$ は以下の性質を満たすことが知られている．

1. 双線形性: 任意の $A, B \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_q$ に対して，$\hat{e}(A^a, B^b) = \hat{e}(A, B)^{ab}$ が成り立つ
2. 非縮退性: $A \in \mathbb{G}_1 (A \neq O)$ のとき，$\hat{e}(A, A) \neq 1$
3. 計算可能性: どの $A, B \in \mathbb{G}_1$ に対しても $\hat{e}(A, B)$ を効率的に計算するアルゴリズムがある

3.2 提案方式

以下に 2.3 節で示したモデルに適応する提案方式を与える．

[事前登録]

1. ユーザ登録

各ユーザは UMS にアクセスし，本人証明を行ってユーザ ID ($ID_{user} \in \{0,1\}^*$) とパスワード ($PW_{user} \in \{0,1\}^*$) を登録する．なお $\{0,1\}^*$ は任意の長さの文字列である．

2. 長期鍵登録

ユーザは秘密情報 ($PW_{LT} \in \{0,1\}^*$) を用いて秘密鍵 ($S_{LT} = H_1(PW_{LT})$) を生成し，それに対応した長期鍵 ($K_{LT} = P^{S_{LT}}$) を公開鍵として LTKS に登録する．

なお H_1 は $\{0,1\}^*$ を，H_2 は $\{0,1\}^n$ を $\mathbb{Z}_q$ 上の元へ移すハッシュ関数を表し，公開パラメータ $P, Q, q, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2, \hat{e}$ はツールとともに配布される．

[通信準備]

3. ユーザ認証

ユーザはシステム起動時に UMS にアクセスし，UMS は登録された正しい ID_{user} と PW_{user} を持つユーザに対してのみアクセス許可証を発行する．

4. ID 発行

アクセス許可証を持つ正しいユーザが使用しているホストに対して IIS がホスト ID ($ID_{host} \in \{0,1\}^n$) を発行する．

[ホスト間直接通信の暗号化処理]

接続確認

システムを起動しているホストは，一定時間ごとに CMS に対して接続要求を確認する．使用しているユーザへの接続要求があった場合，CMS に登録されている要求元の ID_{user}と ID_{host} を取得し，ID_{user} に対応した K_{LT}と K_{ST} をそれぞれ LTKS と STKS から取得する．

接続要求

接続したいユーザ A が自身の ID_{userA} と使用しているホスト A の ID_{hostA}，要求先 (ユーザ B) の ID_{userB} を CMS に書き込む．

短期鍵登録

ホスト A は短期秘密鍵 ($R_{ST_a} \in_{rand} \mathbb{Z}_q$) を生成し，

短期鍵 $\left(K_{ST_a} = P^{\frac{R_{ST_a}}{H_2(ID_{hostA})+S_{LT_a}}} \right)$ を STKS に登録する．

セッション鍵生成

ホスト B を用いるユーザ B がホスト A を用いるユーザ A からの接続要求を受けた場合，まず自身の短期秘密鍵 ($R_{ST_b} \in_{rand} \mathbb{Z}_q$) を生成し，相手の $K_{LT_a}, K_{ST_a}, ID_{hostA}$ を用いてセッション鍵

$$\begin{aligned} K_{SS} &= \hat{e}(K_{ST_a}, K_{LT_a}^{R_{ST_a}} P^{H_2(ID_{hostA})})^{R_{ST_b}} \\ &= \hat{e}\left(P^{\frac{R_{ST_a}}{S_{LT_a}+H_2(ID_{hostA})}}, P^{S_{LT_a}+H_2(ID_{hostA})} \right)^{R_{ST_b}} \\ &= \hat{e}(P,P)^{R_{ST_a} R_{ST_b}} \end{aligned}$$

を生成する．その後，短期鍵 $\left(K_{ST_b} = P^{\frac{R_{ST_b}}{H_2(ID_{hostB})+S_{LT_b}}} \right)$ を生成し，要求元のホスト A に送信する．ホスト A も同様にしてセッション鍵を生成することで，二者間に共通のセッション鍵が得られる．

暗号化通信

生成したセッション鍵を共通鍵とし，共通鍵暗号を行う．

3.3 提案方式の安全性

3.3.1 数論仮定

提案方式の安全性は次の数論仮定に基づく [1].

[CBDH 仮定]

$a, b, c \in \mathbb{Z}_q$ として，$P, P^a, P^b, P^c \in \mathbb{G}_1$ が与えられたとき，

$\hat{e}(P,P)^{abc} \in \mathbb{G}_2$ を計算する問題は困難である．

[ℓ - SDH 仮定]

$P, P^a, P^{a^2}, P^{a^3}, \ldots, P^{a^l} \in \mathbb{G}_1$ が与えられたとき，

w と $\hat{e}(P,P)^{\frac{1}{a+w}}$ を求めるのが困難である．

3.3.2 セッション鍵の復元

攻撃者が各サーバで公開されている情報からセッション鍵 K_{SS} を復元することを考える．

公開情報 : $P, K_{ST_a}, K_{LT_a}, K_{ST_b}, K_{LT_b}, ID_{hostB}$

出力 : $K_{SS} = \left(\hat{e}(P,P)^{R_a R_b}\right)$

これは，P^w, P^x, P^y, P^zから $e(P,P)^{wxyz}$ を求める問題となる．これを解くアルゴリズムを $Alg1(P^w, P^x, P^y, P^z) = \hat{e}(P,P)^{wxyz}$ とすると，CBDH 問題は $Alg1$ を用いて解くことができる．よってこの問題は困難であり．K_{SS} の復元は行えない．

3.3.3 短期鍵偽造によるなりすまし

攻撃者が各サーバで公開されている情報から短期鍵 $K_{ST}(= P^{\frac{R}{H_2(ID_{hostA})+S_a}})$ を偽造し，なりすましを試みる場合を考える．

公開情報 : $P, K_{LT_a}\left(= P^{S_a}\right), ID_{hostA}$

出力 : $R, P^{\frac{R}{S_a+H_2(ID_{hostA})}}$

これは，P, P^x から $R, P^{\frac{R}{x+w}}$ を求める問題である．これを解くアルゴリズムを $Alg2(P, P^x)$=$(R, P^{\frac{R}{x+w}})$ とすると，$P^{\frac{R}{x+w}\frac{1}{R}}$より $P^{\frac{1}{x+w}}$ を計算できる．しかし，$P^{\frac{1}{x+w}}$ を得ることは ℓ-SDH 仮定より困難であるため，攻撃者は正しい K_{ST} を作ることができない．よって，ファイル所持者と同じ K_{SS} を生成できず，相手が攻撃者であることが発覚する．

3.3.4 ID のすり替えによるホストのなりすまし

攻撃者がホストの ID をすり替えた場合を考える．ファイルが要求者に届かないことや攻撃者の手にわたる可能性があるが，ファイルはユーザの長期秘密鍵も使用して暗号化されているため情報を得ることはできない．

3.3.5 長期鍵の入れ替えによるユーザのなりすまし

攻撃者が長期鍵をすり替える場合を考える．2.3 節のモデルより LTKS の情報は不正に改変されず，サーバとの通信内容も入れ替えられないため，安全である．

3.3.6 ユーザの秘密鍵を利用したユーザのなりすまし

攻撃者がユーザの秘密情報を使いユーザになりすますことを考える．モデルよりユーザから秘密情報は漏洩しないので，総当たりしなくてはならない．

3.3.7 短期鍵の入れ替えによるユーザのなりすまし

攻撃者が短期鍵を入れ替え，ユーザになりすますことを考える．モデルより長期鍵と ID の入れ替えがされていないので，入れ替えられた短期鍵によってセッション鍵を生成すると演算結果が異なり，正しい値が得られないため安全である．

4 提案システムの評価とまとめ

本稿では，安全なホスト間直接通信型ファイル配送システムのモデルと，システム内でのホスト間直接通信を暗号化する方式を提案した．提案システムは，暗号化に必要な秘密をホスト側に移したことでサーバ LTKS,STLS は復号に必要な秘密情報を保持しないモデルになっている [要件 1]．また，短期鍵登録時にホスト自身がホスト ID と長期秘密鍵・短期秘密鍵による短期鍵の生成を行うことで，要件 2-1 および要件 2-2 を満たす．さらに，このモデルではシステム起動時に毎回 ID が変更されるが，短期鍵の生成はその後行われるため，要件 3 も満たしている．

以上より，2.2 節で挙げた全ての要件を満たすことを確認した．

参考文献

[1] CRYPTREC ID ベース暗号調査 WG,ID ベース暗号に関する調査報告書, 平成 21 年 3 月.