コンピュータセキュリティ 9－7
（2000．5．10）

匿名通信フレームワークとその評価

北澤 繁樹* 双紙 正和* 宮地 充子*

*北陸先端科学技術大学院大学 情報科学研究科,
〒 923-1292 石川県能美郡辰口町旭台 1-1

あらまし
本論文では匿名通信プロトコルを記述するフレームワークについて議論する．最初にその匿名通信フレームワークを定義し，次に，そのフレームワークにおいて，いくつかの匿名通信プロトコルに共通する基本操作を考える．さらに，それらを用いて複数の匿名通信プロトコルを記述することによって，提案フレームワークが，それらのプロトコルにおける匿名性を実現するための機能を明確に表現できることを示す．これにより，様々な既存の匿名通信プロトコルをフレームワーク上で評価，分類することが可能となる．さらに，フレームワークの妥当性に関しても議論する．

The Anonymous Communication Framework and its Evaluation

Shigeki KITAZAWA* Masakazu SOSHI* Atsuko MIYAJI*

*School of Information Science, Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Tatsunokuchi, Nomi, Ishikawa 923-1292, JAPAN

Abstract
In this paper, we discuss the framework of anonymous communication protocols. First, we define the framework and then consider the basic operations commonly found in some anonymous protocols. Furthermore, by describing such protocols with the basic operations, we show that our framework can specify how the protocols provide anonymity. This also implies that in our framework we can evaluate and classify such protocols. Finally we discuss the appropriateness of the framework.

1 はじめに

ネットワーク通信上のユーザの匿名性や位置情報プライバシを保護するために，数々の研究が行われてきた [1, 2, 3, 4, 6, 7]．1981 年に D. Chaum によって複数の MIX と呼ばれる中継ホストを経由する匿名通信方式が提案された [1]．この方式では，送信者は通信内容とその宛先を入れ子にしていくつかの中継 MIX の公開鍵を用いて暗号化しておく．中継 MIX は，メッセージを受信した相手とメッセージを送る相手のアドレスしか分からない．この MIX の方式を応用したものもいくつか提案されている [5, 7]．

また，ブロードキャストやマルチキャストを利用して匿名性を得る方式が提案されている [2, 3]．ブロードキャストを用いる方式 [3] では，グループ全員にダイレクトで通信するので通信遅延がない．また，マルチキャストを用いた方式 [2] は，中継ノードにおいてデータを保存しておく必要がないといった利点がある．

一方，暗号化やブロードキャストなどのネットワークアーキテクチャに頼らずに匿名性を実現する方式 (Crowds) を 1998 年に M. K. Reiter らが提案した [6]．この方式では，メッセージの送信時に確率的に経路を決定することで，最終的なメッセージの宛先に対して crowd メンバの匿名性を得ることができる．

これらの提案方式は，様々な手法を用いて匿名通信プロトコルを実現している．本論文では，複数の既存の匿名通信プロトコルを記述することによって，それらのプロトコルに共通となる基本操作を明確にし，

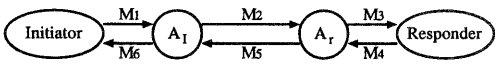

図 1: 一般的なメッセージ通信

どのような機能で匿名性を実現しているかを表現することができることを示す．これにより，匿名通信プロトコルを構成する際に利用できる性質やその実現方法をフレームワークを用いて表現する．さらに，プロトコルを記述して得られた結果から得られた結果からフレームワークの妥当性に関しての評価を行う．

本論文の構成は次の通りである．2章では，匿名通信プロトコルを記述するフレームワークについての定義と説明を行う．3章では機能別の基本関数を定義し，それを用いてプロトコルを記述する．4章では3章での結果を基に記述した匿名通信プロトコルと解析と，提案フレームワークの妥当性について議論する．

2 匿名通信フレームワーク

2.1 エンティティ

提案フレームワーク上のエンティティは以下のように定義される．

イニシエータ I: 送信内容 V を生成し送信を起動するユーザ．

レスポンダ r: イニシエータが出す送信内容 V の最終的な宛先ユーザ．

エージェント A: ユーザの代わりに実際の通信を行うプロセス．本研究では各ユーザにつきひとつのエージェントが存在するものとする．

送信者 S，**受信者** R: ある通信リンク上において，直接通信を送信，受信するユーザあるいはエージェントを表す．ただし，R がグループ ID のときは，R はそのグループ全てのメンバを意味し，メッセージもそのグループメンバ全員にブロードキャストされるものとする．

以上の定義をもとにすると，一般的な通信方式のメッセージ形式は $m = (S, R, I, r, V)$ と定義することができる．一般的な通信のモデルは，図 1で与えられる．ここで，A_I および A_r はそれぞれイニシエータ，レスポンダに対するエージェントである．

2.2 メッセージ形式

従来の匿名通信方式におけるメッセージ M は次の形式で表すことができる．

$$m = (S, R, PI, Pr, PV)$$

ここで，S, R はこのメッセージを直接送信，受信するユーザあるいはエージェントである．また，PI, Pr, PV はそれぞれ、

PI: 返信メッセージの宛先 I に関する情報
返信経路，宛先 I，メッセージ ID など

Pr: 往信メッセージの宛先 r に関する情報
往信経路，宛先 r，メッセージ ID など

PV: 通信内容 V あるいは V を加工した値
V，暗号化した V，分割した V など

と定義する．本文中ではこれらを総称して仮想的なパラメータと呼ぶ．これらは，それぞれ I, r, V における本来の情報を隠しつつ最終的なメッセージの到達可能性や返信可能性を確保するための要素である．

2.3 匿名通信関数

この節では，既存の匿名通信方式を記述するために，フレームワークにおけるエージェントの挙動について説明する．このために，受信メッセージのタイプにより次の送信先を決定し，送信先へのメッセージを生成する匿名通信関数 F を導入する．匿名通信関数 $F(m)$ は，通信路アーキテクチャを決定するメッセージ評価関数 g_x と送信メッセージを生成するメッセージ生成関数 f_x で構成される．これらについて以下で順に説明していく．

2.3.1 メッセージタイプ

各エンティティにおける受信メッセージのタイプは

Type1
イニシエータからエージェントへのメッセージ

Type2
エージェントからエージェントへのメッセージ

Type3
エージェントからレスポンダへのメッセージ

Type4
レスポンダからエージェントへのメッセージ

Type5
エージェントからイニシエータへのメッセージ

の5つである．よって，あるエージェントが受信する可能性があるのは Type1，Type2，Type4 のメッセージタイプである．またそれとは逆にエージェントが送信する可能性があるメッセージタイプは，Type2，Type3，Type5 となる．

2.3.2 エージェントの状態

匿名通信プロトコル上のエージェントは受信メッセージをどのように処理をするのかを，各エージェントがメッセージを受信した時点で持っている状態によって判断する．提案フレームワークでは，エージェントが持つ状態を

Key:匿名通信プロトコルで使う暗号システムの鍵のデータベース
DB_r:エージェントが受信メッセージのデータベース
DB_s:エージェントが送信メッセージのデータベース

に保存されている情報で表現する．これらのデータベースの要素はメッセージのタイプで分けられたデータベースである．

2.3.3 メッセージ評価関数

$g_x(m, \mathrm{Key}, \mathrm{DB}_s, \mathrm{DB}_r)$ (x=2,3,5,{2,3},{2,5}) は，受信メッセージとエージェントの持つ状態から，次のメッセージ送信先のエンティティを判定する．戻り値は true または false である．関数の添字 x はエージェントが送信するメッセージのタイプを表している．メッセージの往信時に g_3 (次の送信先がレスポンダであると判断する関数) または $g_{\{2,3\}}$ (次の送信先がレスポンダとその他のエージェントであると判断する関数) が真になるステップが少なくとも 1 回成立することは，その匿名通信方式におけるメッセージ到達可能性を意味する．また同時に，その状態はレスポンダを特定できる条件を表しているといえる．同様に，返信時に g_5 (次の送信先がイニシエータであると判断する関数) または $g_{\{2,5\}}$ (次の送信先がイニシエータとその他のエージェントであると判断する関数) が少なくとも 1 回成立するか否かで，あるエンティティに対してイニシエータの匿名性を持たせた場合のメッセージ返信可能性を評価することができる．

2.3.4 メッセージ生成関数

メッセージ生成関数 $f_x(m, \mathrm{Key}, \mathrm{DB}_s, \mathrm{DB}_r)$ (x = 2, 3, 5, {2, 3}, {2, 5}) は m のタイプと受信したときのエージェントの状態から次のエンティティに送信するためのメッセージを生成する関数である．よって，受信メッセージに含まれる仮想的なパラメータから，送信メッセージに含める仮想的なパラメータを生成することができる．メッセージの生成法は個々のプロトコルの仕様により決定される．

f_3 および $f_{\{2,3\}}$ はメッセージの仮想的なパラメータとエージェントの持つ状態を用いて実際のレスポンダを特定する機能を持っている．同様に，f_5 および $f_{\{2,5\}}$ にはイニシエータを特定する機能がある．

2.3.5 匿名通信関数

以上の議論に基づき，匿名通信関数 F は以下のように与えられる．

$\mathrm{DB}_{s1}, \mathrm{DB}_{s2}, \mathrm{DB}_{s2}, \mathrm{DB}_{s4}, \mathrm{DB}_{s5}, \mathrm{DB}_{s\{2,3\}}, \mathrm{DB}_{s\{2,5\}} \in \mathrm{DB}_s$
$\mathrm{DB}_{r1}, \mathrm{DB}_{r2}, \mathrm{DB}_{r3}, \mathrm{DB}_{r4}, \mathrm{DB}_{r5} \in \mathrm{DB}_r$
m: エージェントが受信したメッセージ
$type(m)$: 受信したメッセージのタイプを 1, 2, 3, 4, 5 で出力する関数
$\mathcal{M}$: エージェントが出力するメッセージの集合

```
 1  function F(m);
 2   begin
 3    x ← type(m)
 4    M ← ∅
 5    DB_rx ← DB_rx + {m}
 6    if (x = 1) then
 7     begin
 8      M ← f_2^(1)(m, Key, DB_s, DB_r)
 9      DB_s2 ← DB_s2 + {M}
10     end
11    else if (x = 4) then
12     begin
13      M ← f_2^(4)(m, Key, DB_s, DB_r)
14      DB_s2 ← DB_s2 + {M}
15     end
16    else if (x = 2) then
17     begin
18      if g_{2,3}(m, Key, DB_s, DB_r) then
19       begin
20        M ← f_{2,3}^(2)(m, Key, DB_s, DB_r)
21        DB_s{2,3} ← DB_s{2,3} + {M}
22       end
23      else if g_{2,5}(m, Key, DB_s, DB_r) then
24       begin
25        M ← f_{2,5}^(2)(m, Key, DB_s, DB_r)
26        DB_s{2,5} ← DB_s{2,5} + {M}
27       end
28      else if g_2(m, Key, DB_s, DB_r) then
29       begin
30        M ← f_2^(2)(m, Key, DB_s, DB_r)
31        DB_s2 ← DB_s2 + {M}
32       end
33      else if g_3(m, Key, DB_s, DB_r) then
34       begin
35        M ← f_3^(2)(m, Key, DB_s, DB_r)
36        DB_s3 ← DB_s3 + {M}
37       end
38      else if g_5(m, Key, DB_s, DB_r) then
39       begin
40        M ← f_5^(2)(m, Key, DB_s, DB_r)
41        DB_s5 ← DB_s5 + {M}
42       end
43      endif /* line 18, 23, 28, 33, 38 */
```

```
44      end /* line 17 */
45    endif /* line 6, 10, 16 */
46    return(M)
47  end;
```

ここで，18行目の$g_{\{2,3\}}$はType2とType3のメッセージを両方出力する時に真となる．20行目の$f_{\{2,3\}}$はType2とType3のメッセージを両方生成する処理を意味している．また，21行目の表記は，生成したType2とType3のメッセージを両方とも$\mathrm{DB}_{\{2,3\}}$へ加える処理を表している．23行目，25行目，26行目もそれぞれ同様である．

関数g_xおよび関数f_xの引数であるKeyはそのエージェントが管理する暗号化鍵と復号鍵の集合を表している．これらの鍵はメッセージ自体の暗号化や復号にも用いられるが，Iやrを匿名にした場合の匿名通信方式においてデータを復号できるか否かで自分宛であるかを確かめるときにも用いられる．なお，本研究ではこれらの鍵の配送方式などについては言及しない．

また，関数g_xおよび関数f_xはエージェントの状態（Key，DB_s，DB_r）に対する副作用をおよぼさないとする．すなわち，メッセージの生成順序による出力結果の差異は生じない．

7行目から10行目の処理と12行目から15行目の処理はそれぞれエージェントがI，rからメッセージを受信したときの処理である．ただし，フレームワークではType1，Type4のメッセージを受信した場合，エージェント以外のエンティティへ次のメッセージ送信することは想定していない．

17行目から44行目の処理は，メッセージの送信元がエージェントであるメッセージを受信したときの処理である．このとき，エージェントが生成し得るメッセージタイプはType2，Type3，Type5の3タイプである．

46行目では，1回の処理で$F(m)$が出力する全てメッセージを関数の戻り値として返す．ただし，$\mathcal{M}=\emptyset$であった場合には，エージェントの状態は遷移するが，どのエンティティにもメッセージを送信しない．

3 匿名通信プロトコルの記述

ここでは，いくつかの既存の匿名通信プロトコルを提案フレームワークを用いて記述した際のメッセージ生成関数f_xを，機能別のメッセージ生成基本関数を用いて構築する．その匿名通信プロトコルが匿名性がどのような機能によって実現されているかを考察する．

3.1 基本関数の定義

ここで，メッセージ生成関数を構成する基本関数についてあらかじめ定義する．

Encrypt(Key, V)：
入力:暗号化に用いる鍵Key，データV
出力:Keyを用いてを暗号化したV

Decrypt(Key, V)：
入力:復号に用いる鍵Key，データV
出力:Keyを用いて復号されたV

Random()：
入力:なし
出力:必要な長さの真性乱数

Retrieval(x, DB)：
入力:変数x，データベースDB
出力:xをキーワードとして，DBを検索した結果または$null$

Shuffle(n, V)：
入力:変数n，データV 出力:Vをn個の要素に分割し並べ変えたリスト$\mathbf{V}$

Separate(k, V)：
入力:変数k，ベクトルV
出力:Vを無作為にk個に分解した部分ベクトル$\mathrm{V}_1, \cdots, \mathrm{V}_k$

Pickup(t,L)：
入力:変数t，集合L
出力:Lの要素からランダムに選択されたt個の要素

Pselect(P)：
入力:確率P
出力:確率Pで1，0は確率$1-P$

e_a(M)：
入力:メッセージM
出力:Mの要素a (S, R, PI, Pr, PVのいずれか)

ここで，Encrypt関数およびDecrypt関数は対称鍵暗号システムと公開鍵暗号システムのどちらにも適用できるものとする．

3.2 表記

ここで，プロトコルを記述するために必要ないくつかの表記の定義を行う．

OwnAddress: メッセージを生成中のエージェントの識別子
MIX_n: MIXNETにおけるn番目の中継ノード
null: 無効な値（値が無いこと表す）
K_n: MIX_nが使用する暗号システムの鍵（公開鍵と対称鍵は便宜上区別しない）
R_n : MIX_n宛の通信内容をパディングする乱数（$=$*Random*()）
G_A: グループAのグループID
G_A: グループAのグループメンバリスト
crowd: Crowdsにおけるcrowdのメンバリスト
mid: あるメッセージに対して割り当てられる識別子
PathID: ある匿名通信路に対して割り当てられる識別子

3.3 既存プロトコルの記述

ここでは，提案フレームワークを用いて一般に良く知られている匿名通信プロトコルである，MIX [1]，井上-松本方式 [2]，Crowds [6] の記述を行い，4章で，匿名性を評価する．ただし，紙面の都合上，各プロトコルに置ける匿名性に関して重要な記述のみ抜粋して示す．また，これらのプロトコルについてはメッセージ評価関数 g_x は比較的単純であるのでここでは言及しない．

3.3.1 MIX

本節では，MIX におけるメッセージ生成関数に関して記述を行う．MIX は一方向通信のプロトコルであるので，ここでは往信に関してのみ議論する．

メッセージ生成

$f_2^{(1)}$:

$$\begin{cases} S = OwnAddress \\ R = \mathrm{MIX}_n \\ PI = null \\ Pr = PV = Encrypt(K_n, (R_n, \\ \quad Encrypt(K_{n-1}, (R_{n-1}, Encrypt(\cdots, \\ \quad Encrypt(R_1, Encrypt(K_r, V), A_r)))))) \end{cases}$$

$f_2^{(2)}$:

$$\begin{cases} S = Ownaddress\ (= \mathrm{MIX}_i) \\ R = \mathrm{MIX}_{i-1} \\ PI = null \\ Pr = PV = Decrypt(K_i, (Encrypt(K_i, (R_i, \\ \quad Encrypt(K_{i-1}, (R_{i-1}, Encrypt(\cdots, \\ \quad Encrypt(R_1, Encrypt(K_r, M), A_r))))))) \end{cases}$$

3.3.2 井上-松本方式

つぎに，井上-松本方式における基本プロトコルについての記述を行う．基本プロトコルでは，メッセージを受信したグループメンバは直接レスポンダと通信する必要があるので，レスポンダの匿名性については考慮していない*．

メッセージ生成

$f_2^{(1)}$:

$\mathcal{M}_1 = (S, R_1, PI, Pr, PV_1)$

$\mathcal{M}_2 = (S, R_2, PI, Pr, PV_2)$

$$\begin{cases} S = OwnAddress\ (= A_I) \\ (R_1, R_2) = Pickup(2, \mathbf{G_A}) \\ PI = (G_A, mid\ (= Random())) \\ Pr = Ar \\ (PV_1, PV_2) = Separate(2, Shuffle(n, V))) \end{cases}$$

$f_2^{(4)}$:

$$\begin{cases} S = OwnAddress(= A_r) \\ R = G_A\ (= e_{PI}(Retrieval(mid, \mathtt{DB_r}))) \\ PI = (G_A, mid) \\ Pr = Ar \\ PV = V' \end{cases}$$

$f_2^{(2)}$:

$\mathcal{M}_1 = (S, R_1, PI, Pr, PV_1)$

$\mathcal{M}_2 = (S, R_2, PI, Pr, PV_2)$

$\mathcal{M}_3 = (S, R_3, PI, Pr, PV_3)$

$$\begin{cases} S = Ownarddress \\ (R_1, (R_2, R_3)) = (A_r, Pickup(2, \mathtt{G}_A)) \\ PI = (G_A, mid) \\ Pr = A_r \\ (PV_1, (PV_2, PV_3)) = (V_h\ (= Pickup(1, \mathtt{V})), \\ \quad Separate(2, \mathtt{V} - \{V_h\})) \end{cases}$$

3.3.3 Crowds

ここでは，Crowds についての記述を行う．ただし，簡単のため Crowds の匿名通信プロトコルの初めの送信に関して記述を行う．

メッセージ生成

$f_2^{(1)}$:

$$\begin{cases} S = OwnAddress \\ R = Pickup(1, \mathtt{crowd}) \\ PI = PathID(= Random()) \\ Pr = Ar \\ PV = V \end{cases}$$

$f_2^{(2)}$:

$$\begin{cases} S = OwnAddress \\ R = \begin{cases} Pickup(1, \mathtt{crowd}) & \cdots Pselect(p_f) = 1 \\ A_r & \cdots Pselect(p_f) = 0 \end{cases} \\ PI = PathID, OwnAddress \\ Pr = Ar \\ PV = V \end{cases}$$

*ただし，基本プロトコルを基に，レスポンダの匿名性を考慮した拡張プロトコルの提案もされている [2].

	Iの匿名性		rの匿名性
	中継エージェントに対して	rに対して	中継エージェントに対して
MIX	メッセージの中継		$Encrypt(K_n, (R_n, \cdots, Encrypt(K_r, V), A_r))$
Crowds	$Pickup(1, \mathbf{crowd})$	crowd	なし
井上-松本方式	$Pickup(2, \mathbf{G}_A)$	G_A	なし

表 1: 既存プロトコルの匿名性の実現

4 考察

4.1 既存プロトコルの分類と比較

表 1に，3.3節で記述した匿名通信プロトコルにおいて匿名性がどのように実現されているのかをまとめたものを示す．表 1から，匿名性を実現する手法は以下の 2 種類あることが分かる．

1. 暗号システムを用いて，I や r を秘匿
2. A_I や A_r と中継エージェントの挙動を，他のエンティティが区別できないようにする

暗号システムを用いて匿名性を得る場合には，匿名性は使用する暗号システムの強度に依存する．したがって，その暗号システムが解読された場合には匿名性は失われる．表 1において MIX の r のメッセージ中継エージェントに対する匿名性はこの手法を利用している．

また，中継エージェントの挙動を区別できなくする手法は，メッセージを受け取った相手とメッセージを送った相手しか分からないという前提を基にメッセージはグループ内のあるエージェントが I または r であるところまでは分かるが，その中から I や r を特定することは確率的にしかできない．よって，エージェントの数が多くなればなるほど匿名性が増す．3.3節で記述した，MIX，井上-松本方式，Crowds ともに，中継エージェントに対する I の匿名性をこの手法を用いて実現している．ただし，MIX に関しては，MIX の中継エージェントメンバには A_I が存在しないので，r の受信メッセージの I となる可能性があるのは r を除いた全てのユーザとなる．

4.2 フレームワークの妥当性

提案フレームワークでは，通信においてメッセージを送受信するエンティティとしてエージェントを定義し，その挙動を匿名通信関数 F とその内部関数であるメッセージ評価関数 g とメッセージ生成関数 f を用いて記述することで，匿名通信プロトコルの評価，分類をすることが可能である．例えば，表 1で示したように，Crowds と井上-松本方式は，共通しているメッセージ生成基本関数が多く，Crowds と井上-松本方式は同じ手法で匿名性を実現しているといえる．

5 むすび

本論文では匿名通信プロトコルを記述するフレームワークについて議論するため，匿名通信フレームワークを定義し，フレームワークにおける，いくつかの匿名通信プロトコルに共通する基本操作となる関数を定義した．さらに，それらを用いて複数の匿名通信プロトコルを記述することによって，提案フレームワークが，それらのプロトコルにおける匿名性を実現するための機能を明確に表現できることを示した．これにより，フレームワーク上の MIX，井上-松本方式，Crowds のプロトコルの匿名性について評価，分類することができた．その結果，提案フレームワークは匿名通信プロトコルを記述するフレームワークとして妥当であることが分かった．

参考文献

[1] Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM*, Vol. 24, No. 2, pp. 84–88 (1981).

[2] Inoue, D. and Matsumoto, T.: Anonymity Achieved by Group Communication, *Proc. IEICE JW-ISC2000* (ISEC-99-100).

[3] Kikuchi, H.: Sender and Recipient Anonymous Communication without Public Key Cryptography, 情報処理学会研究報告 (98-CSEC-1-8) (1998).

[4] 長野悟, 北澤繁樹, 双紙正和, 宮地充子: 環状経路を用いた匿名性と位置情報プライバシの保護, コンピュータセキュリティシンポジウム (CSS99), pp. 37–42 (1999).

[5] Pfitzmann, A., Pfitzmann, B. and Waidner, M.: ISDN-MIXes: Untraceable Communication with Very Small Bandwidth Overhead, *Proc. IFIP/Sec'91*, pp. 245–258 (1991).

[6] Reiter, M. K. and Rubin, A. D.: Crowds: Anonymity for Web Transactions, *ACM Trans. Info. Syst. Security*, Vol. 1, No. 1, pp. 66–92 (1998).

[7] Syberson, P. F., Coldschlag, D. M. and Reed, M. G.: Anonymous Connections and Onion Routing, *IEEE Symposium on Security and Privacy*, pp. 44–54 (1997).