平成21年度　　卒業研究論文

匿名通信方式Bifrostの受信エリアを利用した不正者追跡処理の暗号化コスト削減

指導教官
齋藤 彰一 准教授

名古屋工業大学　情報工学科
平成18年度入学　18115007番

秋山　晋

目 次

第1章

はじめに

今の時代，インターネット技術は社会に深く浸透し，ネットワーク利用者の数は増え続けている．これにより，接続する端末数が多い場合，クライアント・サーバーモデルを用いたネットワーク形態では端末からの要求をサーバーがすべて処理しきれず，パンクすることがある．そこで，近年 P2P（Peer To Peer）方式と呼ばれるネットワーク形態が注目を浴びている．P2P 方式は対等なもの同士（Peer）が通信を行うことを特徴としており，サーバーが存在しないため，パンクするという事態を避けることが可能である．

近年，この P2P 方式を用いた通信において，個人情報の漏洩や通信内容の改竄などの事件が後を絶えない．最近では Winny 事件が記憶に新しい．ファイル共有ソフト Winny[1] の利用者の個人情報が，本人の知らぬ間に流出していた，という事件である．この個人情報の中には，会社の顧客情報や機密情報も含まれており，社会に与えた影響は大きなものであった．

このような情報漏洩事件が発生するにつれて，ネットワーク利用者は通信の安全性に対する意識を高め，通信に際しての匿名性が関心を集めるようになった．ただ，これには通信の匿名性を高めることで，ネットワーク利用者の情報を保護することが出来るが，逆に，不正者の匿名性も確保してしまうという問題がある．通信において不正が見つかった場合，どの利用者が不正を行ったかを特定するまでに時間がかかったり，最悪，特定不可能となる可能性が考えられるのである．しかし，不正者の特定を前提とすると匿名性が確保できなくなり，逆に匿名性を意識しすぎると不正者の特定

が難しくなる．よって，匿名性と不正者の特定可能性はトレードオフの関係にあるといえる．匿名性と不正者特定可能性はトレードオフの関係にあるものの，前者はネットワーク利用者の立場から考えて，後者はネットワーク管理者の立場から考えて，どちらも満たされるべきである．ネットワーク利用者としては，プライバシー保護は欠くことはできないが，ネットワーク管理者としては，不正者を割り出し，警察に突き出すなり，然るべき処置を施す必要があるからである．

以上の解決手段として，匿名性と不正者追跡可能性を閾値制御可能な方式，RANS (Revocable Anonymous Network System)[2] が提案されている．このシステムでは，事前に指定された第三者機関のうちの一定数以上がネットワーク利用者の匿名性失効を必要と判断したときのみ，その不正利用者を特定可能な技術を提案している．また，OnionRouting[3] や MixNet[4] といった匿名通信技術を用いることで，通信に匿名性を持たせている．

この不正者特定技術であるが，これは通信を行う際に通信経路となるすべての端末に対して，その端末の受信情報をメッセージに付加させることで，不正者追跡の際，付加させた情報とメッセージを比較させて改竄がなかったかを検証する手法をとっている．この手法では，各端末がメッセージを受け取る度に受信情報を付加させているので，中継端末を増やすほど暗号化コストが高くなる．

そこで本稿では，RANS における各端末の受信情報を付加する単位を，端末単位ではなく，端末の集合単位で行うことで暗号化コストを低下させる手法を提案する．2 章では本研究の関連研究について説明する。3 章では提案手法について述べる。4 章で実装方法について述べる。5 章では提案手法の評価と考察を行う。6 章で結果をまとめる。

第2章

関連研究

2.1 匿名通信路技術

本章では本研究を行う際の関連研究について説明する．

2.1.1 Onion Routing

匿名通信方式のひとつで，送受信者を特定されないようにすることで通信路に匿名性を持たせる．送信者はメッセージを送信前に，各中継ノードの送信するべき宛先をメッセージに含め，各ノードの公開鍵で多重暗号してから送信を行う．暗号化されたメッセージを受け取ったノードは自身の持っている秘密鍵で復号し，送信者がメッセージに含めた宛先を得て，次のノードへとメッセージを中継する．この動作を繰り返すことで最終的には受信者へとメッセージが届く，という仕組みである．

図 2.1 に OnionRouting によるメッセージの送受信の流れを示す．送信者 S はフラッディングにより受信者 R までの経路を得る．図 2.1 では送信者 S から A,B を経由して受信者 R へと通信路が張られている．送信者 S は始めに A へメッセージを転送するわけだが，転送前にメッセージに対して各ノードの公開鍵により暗号化を行う．まず受信者 R との共通鍵によって暗号化したメッセージ M を，共通鍵と一緒に受信者 R の公開鍵 K_R で暗号化する．この処理により，メッセージを受け取った受信者 R は自らの秘密鍵で暗号文 M と共通鍵を得ることができる．次に，このメッセージを，ノード B の送信先である受信者 R のアドレスと一緒に公開鍵 K_B で暗号化する．同様に，K_B

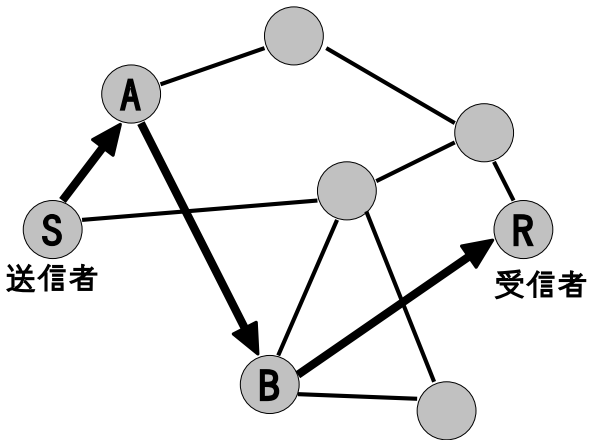

図 2.1: OnionRouting

で暗号化したメッセージを，ノード A の送信先であるノード B のアドレスと一緒に公開鍵 K_A で暗号化する．この一連の処理により，メッセージを受け取った中継ノードは自らの秘密鍵で，暗号文と自身の送るべき宛先のアドレスを得ることができる．以上が送信者 S が転送前にメッセージに施す暗号化の手順である．各中継ノードは，メッセージを受信した場合先ず自身の持つ秘密鍵で復号し，そこから得られたメッセージを，同様に得られたアドレスへ転送する．この動作を繰り返すことでメッセージは最終的に受信者に至る．

以上のようにメッセージを送ったとき，中継ノードはそのメッセージの差出人と受取人を知ることはないので，自身からみて直前直後 1 ノードが，送信者と受信者であることまでしか知りえない．これは，メッセージを受け取った中継ノードは，メッセージを復号するまでは自身の送るべき宛先がわからないことによる．つまり，各ノードがメッセージを受け取った場合，自らの秘密鍵で復号して得た宛先へと送ることで通信路に匿名性を持たせている．ただし，この通信路匿名技術は，中継するノードの共通鍵など，送信者がメッセージを多重暗号するための情報をすべて管理する必要がある．これは，ネットワーク内の各ノードが，全ノードの情報を管理する必要性があるということであり，ネットワーク規模が拡大するつれて各ノードが負うネットワーク管理コストが高くなる恐れがある．

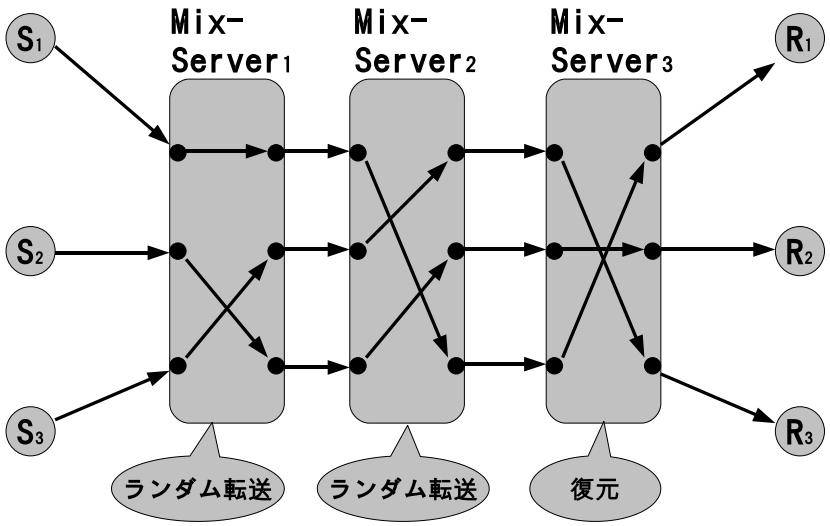

図 2.2: Mix-net

2.1.2 Mix-net

Mixnet は送信者らの送ったメッセージをある程度集まるまで蓄積してからシャッフルし，どのメッセージを誰が送信したかの対応関係を隠蔽することで匿名性を持たせている．ここに，複数の送受信者，中継サーバがあったとする．中継サーバは送信者からメッセージを受け取った場合，別の中継サーバへメッセージを中継させ，最終的に受信者へ送り届ける．ただし，このメッセージ通信には３つの制約がある．

- 中継サーバは，ある程度のメッセージ数が蓄積されるまで送信を行えず，蓄積されたメッセージは詰め物をすることでサイズを均一にする．
- 送信の際，送信者は，各中継サーバが復号できるよう，公開鍵を使って多重暗号化する．
- また，送信するある一定量のメッセージ群は，中継サーバによってミックス（シャッフル）することでメッセージの流れを隠蔽する．

図2.2にMix-netによるメッセージ送受信の流れを示す．送信者S_1,S_2,S_3はそれぞれ受信者R_1,R_2,R_3へメッセージを送信したいとする．このとき，送信者は$Server_3$,$Server_2$,$Server_1$の公開鍵で順にメッセージを暗号化する．$Server_1$は暗号文を受け取るとき，制約どおりに，一定数メッセージが蓄積されるまで送信を行わない．暗号文が充分蓄積された$Server_1$は，各暗号文のサイズを揃えるために詰め物をする．サイズの揃った暗号文は，受信順と関係なくランダム（ミックス）に$Server_2$へ送出される．以降Serverでは同様の処理を行い，受信者の接続先である$Server_3$が各受信者Rへ送出する．

以上の制約どおりにメッセージを送った場合，各中継サーバが通信を盗聴されたとしても，誰のどのメッセージがいつ中継サーバから送出されたかを隠蔽できる．なぜなら，メッセージ群はサイズが均一であるから，どれが誰の送ったメッセージなのか見分ける手がかりを消しているからである．加えて，中継サーバによってミックスしていくことで中継サーバ自身も送信者がどのメッセージを受信者に宛てて送信したのか対応関係を知り得ない．この手法は電子投票に用いられるほど強力な匿名通信方式である．しかし，サーバにメッセージをある程度蓄積するまでに時間がかかるという欠点を持つ．

2.2 既存の不正者特定可能な匿名通信方式RANS (Revocable Anonymous Network System)

2.2.1 不正者追跡用データ

RANSでは，送信者や受信者を含め，メッセージの通信路上にいるノードすべては不正者追跡用データ[2]を生成する．生成した追跡用データは信用できる第3者機関との共通鍵で暗号化する．不正者追跡用データ$\Gamma_i(0 < \cdots < i < \cdots m : m$は受信者$)$はノード$N_i$が送信元ノード$Ni-1$から受信した情報によって構成される．具体的には，メッセージC_{i-1}，送信元ノードの生成した不正者追跡用データΓ_{i-1}，署名S_{i-1}である．追跡を行う者は，信用できる第3者機関から許可を得ることで不正者追跡用データが復号可能となり,，受信者の不正者追跡用データΓ_mから再帰的に復号していく．復号して得たメッセージC_{i-1}と，検証対象のノードに届けられたメッセージC_{i-1}を比

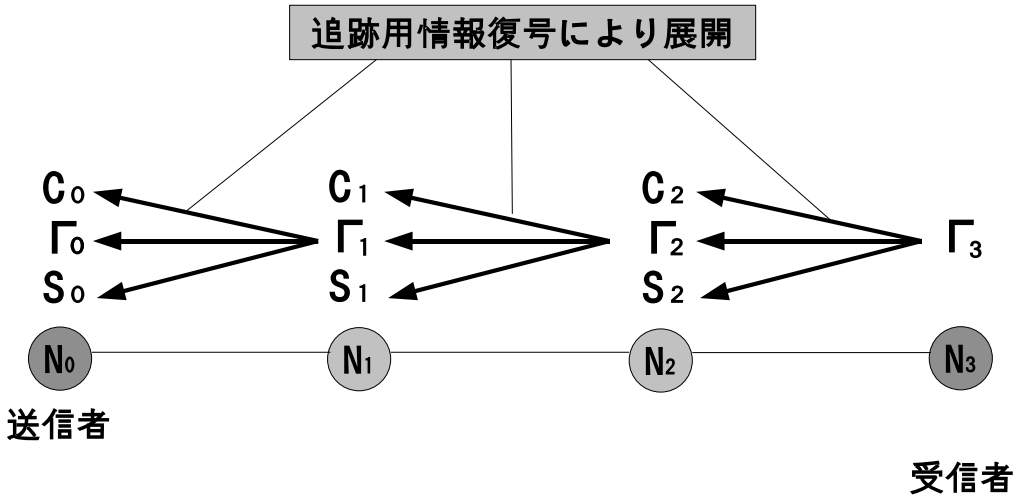

図 2.3: 不正者追跡から特定までの流れ

較することで，メッセージに改竄がなかったかを検証することが可能となる．検証の結果，あるノード N_i が改竄なしとされた場合，今度はそのノードからみて送信元ノードの検証を行う．同様に Γ_{i-1} を送信元ノード N_{i-1} との共通鍵で復号し，メッセージ C_{i-2}，Γ_{i-2}，署名 S_{i-2} を得ることで N_{i-1} の検証を行う．

図 2.3 に RANS の不正者追跡の流れを示す．送信者 N_0 は N_1,N_2 を経由して N_3 へ，メッセージ C_3' を多重暗号化して生成した C_0' を送信する．よって，改竄なく受信者に届いた場合，メッセージは C_3' を得るはずである．

図 2.3 より，受信者 N_3 は N_2 から受信した C_2, Γ_2,S_2 を使って Γ_3 を生成する．不正者追跡を行う場合，まず受信者のもつ不正者追跡用データ Γ_3 を得ることから始まる．受信者 N_3 が不正者追跡データとは別に受信して得たメッセージ C_3 を，送信者 N_0 が多重暗号化に使った各ノードの公開鍵 K_3 を使うことで，N_0 が生成したメッセージ C_2'

を再現する．C_2' と，Γ_3 から得られた C_2 の内容が一致するか検証し，一致するのなら N_3 は不正者でないことが証明できる．同じ要領で，公開鍵 K_2 を用いて C_1' を生成し，Γ_2 から得られた C_1 と比較することで N_2 の検証が可能である．検証結果が不一致であれば，その検証対象ノードが不正者であるし，送信者 N_0 まで追跡できたのであれば，N_0 が不正を働いていたということになる．

2.2.2 検証可閾値暗号

閾値暗号 [5] とは，秘密鍵を分散して扱い，それをそれぞれ保持する複数の機関が一定数以上の協力を得られたときのみ，復号可能となる暗号系のことである．この検証可閾値暗号の長所として挙げられるのが，分散鍵をそれぞれ保持している任意の機関が，秘密鍵を明かすことなく，自身の正当性を証明できる点である．つまり，秘密鍵を明かさないことで特定の暗号文のみを復号する，といった処理を可能とするのである．

RANS では分散鍵を，信用できる複数の第三者機関に保持させてあり，追跡を行う者は，信用できる複数の第三者機関へ不正者追跡の許可を申請する．複数の第三者機関から一定数以上の同意を得られた場合に限り，第三者の保持する分散鍵から秘密鍵を得る．

2.2.3 基盤方式

RANS における通信時の処理概要を図 2.4 に示す．U：ユーザ端末，M_i(i=1,⋯,m): 中継端末,R:受信端末　とする．ε_i,P を閾値暗号関数とする．また，U,M_i はそれぞれ署名生成関数 S_0,S_i を保持し，対応する署名検証関数をそれぞれ V_0,V_i とする．不正者追跡時の処理概要を図 2.5 に示す．

1. U は以下を行う．

   (a) R のアドレス $Addr_{m+1}$, 中継させるノードのアドレス $Addr_i$ (i=1,⋯,m) を取得する．

   (b) メッセージ $C_m = msg$ に対し，$i = m-1,\cdots,0$ の順に $C'_i = (Addr_{i+2}||C_{i+1})$,$C_i = \varepsilon_{i+1}(C'_i)$ を計算する．

   (c) $\sigma_0 = S_0(C_0)$ を計算し，(C_0, σ_0) を $Addr_1$ に従って M_1 に送信する．

2. $i = 1,\cdots,m$ の順に M_i は以下を行う．

   (a) $\mathrm{V}_{i-1}(C_{i-1}||\ \Gamma_{i-1}, \sigma_{i-1} = 1)$ が成り立つ事を確認する．(Γ_0 は空の値)．

   (b) C_{i-1} を復号し，$C'_{i-1} = (Addr_{i+1}||C_i)$ を得る．

   (c) $\Gamma_i = \mathrm{P}\ (C_{i-1}||\ \Gamma_{i-1}||\sigma_{i-1})$,$\sigma_i = S_i(C_i||\ \Gamma_i)$ を計算し，$(C_i,\ \Gamma_i, \sigma_i)$ を $Addr_{i+1}$ に従って $M_{i+1}(i = m$ であれば $R)$ に送信する．

3. R は以下を行う．

   (a) $\mathrm{V}_m(C_m||\ \Gamma_m, \sigma_m) = 1$ が成り立つことを確認する．

   (b) 入力組 $(C_m,\ \Gamma_m, \sigma_m)$ を保管する．

図 2.4: 基盤方式 [6] の通信時の処理フロー

- 信用できる第 3 者機関から許可を得た追跡者は以下を行う．
  1. R の保管する C_m, Γ_m,σ_m を R から入手する．
  2. $\mathrm{V}_m(C_m||\ \Gamma_m, \sigma_m = 1)$ が成り立つ事を確認する．成り立たなければ R が不正者であると判断．
  3. $i = m,\cdots,1$ について以下を行う．
     (a) Γ_i を復号する．(復号結果を $(C^*_{i-1},\ \Gamma^*_{i-1}, S^*_{i-1})$ とする)．
     (b) ε_i に対応する分散鍵を用いて C^*_{i-1} を復号する（復号結果を C^*_i とする．すなわち $C^*_{i-1} = \varepsilon_i(C_i$ が成り立つ）．
     (c) $C^*_i = C_i$および$\mathrm{V}_{i-1}(C^*_{i-1}||\ \Gamma^*_{i-1}, S^*_{i-1}) = 1$ が成り立つか確認する．成り立たなければ M_i の不正と判断
     (d) $(C_{i-1},\ \Gamma_{i-1}, S_{i-1}) \leftarrow (C^*_{i-1},\ \Gamma^*_{i-1}, S^*_{i-1})$ とする．
  4. $(C_0,\ \Gamma_0, S_0)$ から不正利用者が U と特定する．

図 2.5: 不正者追跡時の処理フロー

2.3 分散ハッシュテーブル (DHT)

ハッシュテーブルを複数のノードで管理することにより，管理コストを抑える技術のことで，本章では分散ハッシュテーブル [7] について説明する．

2.3.1 ハッシュテーブル

ハッシュテーブルとはキー (key) と値 (value) のペアを格納し，ひとつ以上のハッシュ関数によってアクセスできるデータ構造のこと．key をもとに生成されたハッシュ値を配列の添え字とすることで，key に対応する value をすばやく参照することが出来る．ハッシュテーブルを用いてノードを管理する場合，key にノードにユニークに割り振られた ID を，value にアドレスを設定することで，ノードの ID からすばやくアドレスを参照することが出来る．以降，ID と IP アドレスのペアを **ID アドレスペア**と呼ぶ．

ネットワーク全体を網羅したハッシュテーブルを，複数のノードで管理することを分散ハッシュテーブルとよぶ．分散して管理するので，自身の持つハッシュテーブル内にアクセスしたいノードの ID が含まれていない場合がある．このとき，別のノードに問い合わせることで ID アドレスペアを入手する．問い合わせ先のノードのハッシュテーブルにも含まれていなかったとき，そのハッシュテーブルから，アクセスしたいノードに一番近いノードへ問い合わせていき，最終的にアドレスを得る．

chord

chord[8] は分散ハッシュテーブルを実装するアルゴリズムのひとつである．ノードを円状に管理するのが特徴で，各ノードがハッシュテーブルで管理するノードの範囲はある計算式によって導かれる．図 2.6 にノード ID とペアで格納されているアドレスの参照方法を示す．ノード ID1 はノード ID15 にアクセスするために，ノード ID15 のアドレスが得たいとする．このとき ID1 は自身のハッシュテーブルから ID15 の ID アドレスペアが格納されているかを調べる．ID15 が格納されていないので，保持しているペアの中で一番近い ID5 に問い合わせる．ID5 のハッシュテーブルにも ID15 の ID アドレスペアが保持されていないので，同様に ID15 に一番近い ID11 へ問い合わせる．

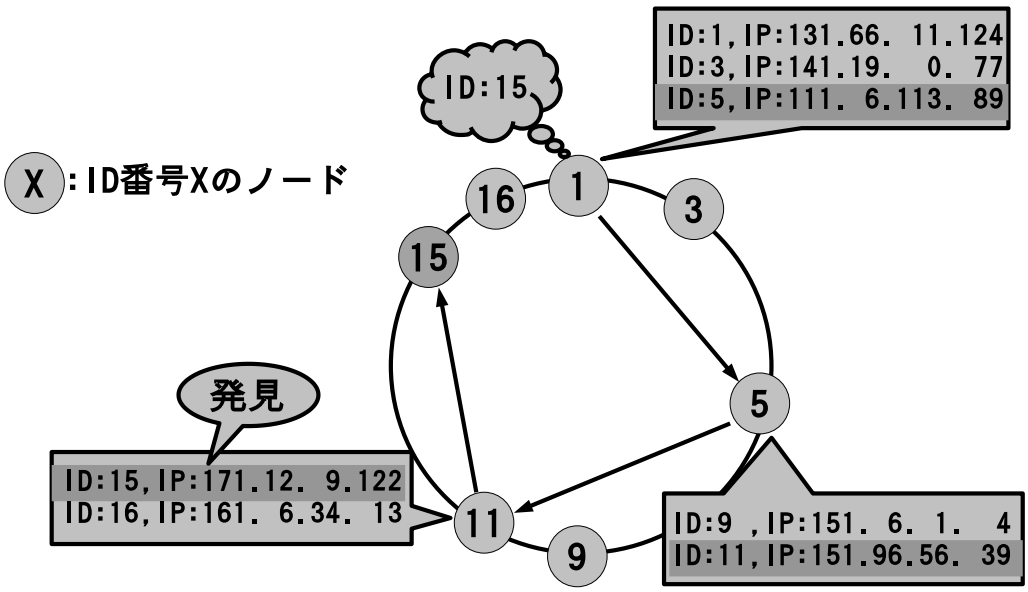

図 2.6: chord

ID11 のハッシュテーブルの中に ID15 のペアが含まれているので，問い合わせを終了し，いままで問い合わせてきたノードを中継することで ID15 にアクセス可能となる．

2.4 Bifrost

Bifrost[9] とは，オーバーレイネットワーク上のノードを，分散ハッシュテーブルの実装アルゴリズムである chord を用いて管理するネットワークシステムである．Bifrost では**受信エリア**を設定し，エリア内ではメッセージを中継し，エリア末端まで着いた場合，別の受信エリアの先頭へメッセージを転送する．以降，ノード ID 上隣接している直前直後のノードをそれぞれ **Successor**,**Predecessor** と呼ぶ．

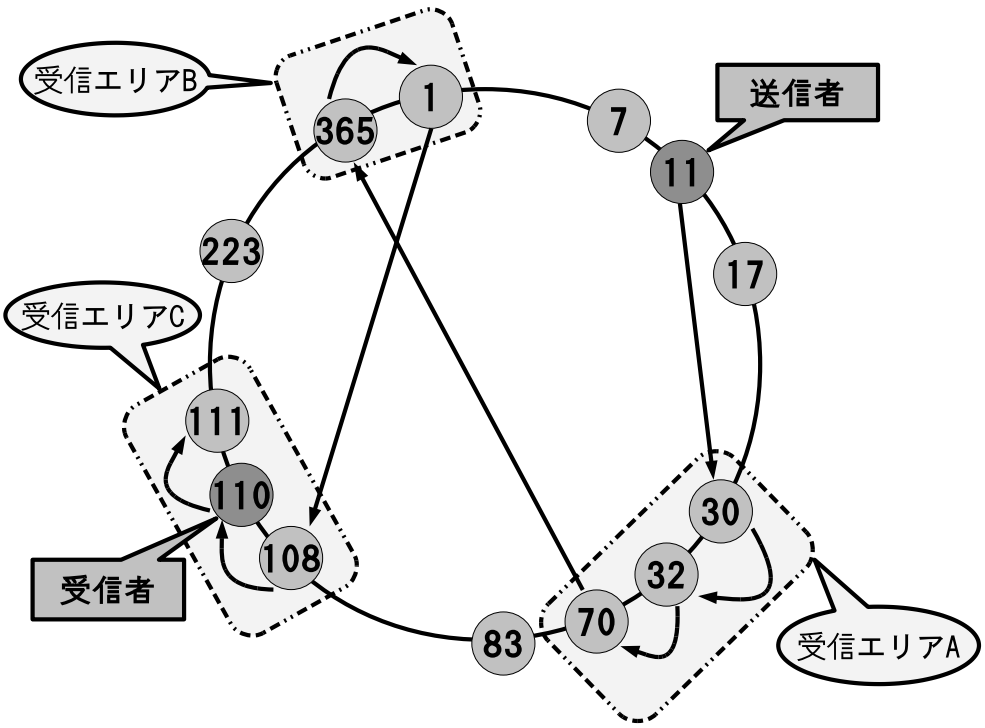

図 2.7: 受信エリア

2.4.1 受信エリア

受信エリアとは，Bifrost におけるメッセージを受信するノードの集合単位である．受信エリアは ID が連続した複数のノードによって構成され，送信者がエリアを設定する．受信エリア内のノードはメッセージを受け取ると，そのメッセージを Successor へ転送する．受信エリア末端のノードは，自身の送るべき受信エリアがあるか確認し，存在する場合はその受信エリアの先頭ノードへメッセージを転送する．

図 2.7 に受信エリアの設定方法について示す．受信エリアは送信者が送信前に始点，終点を指定することで設定され，始点と終点の間に存在するすべてのノードもひとつの受信エリアとして設定されたことになる．受信エリア A のように，始点を ID30，終点を ID70 と定めた場合，ID30 から ID70 の間に存在する ID32 を含めた 3 ノードが受信エリアに設定されたことになる．ただし，設定する受信エリアのうちひとつに必ず

受信者ノードを含めなければならない．また，メッセージは受信エリアを設定した順に中継される．送信者の設定した順が受信エリア A,B,C とする．この場合，まず送信者は受信エリア A の先頭ノードである ID30 へメッセージを送信する．受信エリア内ではメッセージは Successor へ転送されるので，ID30 から ID32 を経て，受信エリア A の末端 ID70 に転送される．受信エリア末端のノードはまだ送信するべき受信エリア (受信エリア B) が存在するので，ID70 は受信エリア B の先頭ノード ID365 へメッセージを転送する．同様に受信エリア B 内でもメッセージは Successor へ転送される．受信エリア B の末端である ID1 は送信するべき受信エリア（受信エリア C）の存在を確認し，受信エリア C の先頭ノード ID108 へメッセージを転送する．受信エリア C の末端である ID111 は，送信するべき受信エリアの存在が確認できないので，メッセージの転送は終了となる．

2.4.2 受信エリア内ノードの処理

各ノードはメッセージを受け取った際，メッセージのヘッダ部が自身の保持する秘密鍵で復号できるかを試みる．ヘッダ部は送信者により，受信エリアの末端ノードの公開鍵でヘッダ部を暗号化してある．よって，ヘッダ部を復号できるのならばそのノードは受信エリアの末端であることがわかる．受信エリアの末端はヘッダ部を復号することで得た宛先（次の受信エリアの先頭ノード）へメッセージを転送する．復号できなければ，そのノードは受信エリアの末端ではないので，メッセージを Successor へ転送する．

メッセージを転送し終えたノードは，今度はメッセージのボディ部を自身の保持する秘密鍵で復号できるか試みる．復号できるなら，そのノードは受信者である．復号して得た暗号文を，同じく復号して得た共通鍵で復号することで，平文を得ることができる．

2.5 問題点

RANS では経路となるノードすべてが不正者追跡情報を生成している．不正者追跡情報を生成することで暗号化コストが高くなるが，実際に追跡が行われるのは誰かが不正行為をし，かつ信用できる第 3 者から許可が下りた場合に限る．よって，膨大に行われる不正のない通常の通信にとって，暗号化コストが増大することで通信の性能低下に繋がる．故に，不正者追跡情報を生成する際に生じる暗号化コストをできる限り抑える必要がある．

第3章

提案方式

追跡用情報の付加をノード単位ではなく，ノード集合単位で行うことで暗号化コストを押さえ，通信コストを下げる手法を提案する．

3.1 全体構成

前章で述べた問題点を解決する手法を提案する．提案方式では暗号化を行うノードを減らすことで，既存方式よりも暗号化コストを削減する．既存方式では通信に参加する全ノードが暗号化を行っているが，本提案では暗号化する単位をノード単位からノードの集合単位へ変更する．ノードの集合単位であるが，これは前章で説明した既存研究である Bifrost で定義されている受信エリアを用いる．RANS での逐次的に追跡情報を生成する手法を比べ，各受信エリアの先頭ノードのみが追跡情報を生成するので，生成回数を減らすことが可能である．受信エリアの先頭ノードはメッセージを受け取った場合, 受信エリア内の Predecessor へとメッセージを転送する．以降，不正者追跡用の情報を**追跡署名**とよぶ.

3.2 追跡署名の付加

提案手法では RANS と異なり，追跡署名を行う単位をノードの集合である受信エリアとした．そこで，受信エリアを追跡署名を行う単位とした場合の追跡署名方法について説明する．

3.2.1 メッセージ受信時の各ノードの動作

各ノードは始めに送られた署名を検証する．次に，自ノードが受信エリアの始点であったときに限り，追跡署名を行う．受信エリアの始点以外のノードは追跡署名を行わない．生成した追跡署名，受信したメッセージ，自身の生成した署名の 3 点を Successor へ転送する．受信エリアの始点以外のノードは追跡署名を行わないので，受信して受け取った追跡署名をそのまま Succssor へ転送する．

3.3 不正者追跡

追跡署名を用いて不正者を追跡する方法を説明する．

3.3.1 追跡方法

追跡を行う者は信用できる第 3 者機関から閾値以上の同意を得ることで追跡可能となる．RANS 同様，各ノードが生成した追跡署名を復号することで得られるノードの受信情報に改竄がなされていないかを検証する．このとき受信エリアの先頭ノード以外は追跡署名を行っていないので，追跡者は受信エリア内の Predecessor より遡ることで受信エリアの先頭ノードを特定する．受信エリアの先頭ノードの生成した追跡署名を検証し，改竄無しと判断された場合，そのノードにメッセージを転送したノードを追跡署名から特定する．改竄ありと判断された場合，その先頭ノードを含む受信エリアの全ノードは不正者ではないと保証されていないので，不正者の候補と判断される．

3.3.2 提案手法における追跡可能性と精度

RANS では逐次的に追跡署名を行っている．暗号化コストを削減するために追跡署名を行うノード数を減らした場合，追跡署名がなされていないノード間の情報が得られないので，どのノードがどこからメッセージを受け取ったか不明となる．つまり，追跡署名を逐次的に行わなければ，送信者まで追跡不可能になる．

提案手法では，受信エリアを追跡署名の単位とすることで，追跡可能性を実現して

いる．受信エリアの先頭ノードのみが追跡署名を行う場合，追跡者は受信エリアの先頭ノードへメッセージを送信したノードしか辿ることが出来ない．しかし，Bifrost ではノード ID やアドレスがディレクトリサーバによって管理されているので，ディレクトリサーバに問い合わせることでノードの Predecessor を特定することが可能である．受信エリア内のノードは Successor へメッセージを転送するので，逆に Predecessor を辿ることで受信エリア内の先頭ノードを見つけることができる．よって受信エリアを追跡署名の単位とした場合，受信エリアの先頭ノードさえ追跡署名を行っていれば送信者まで追跡が可能となる．

この提案手法では，受信エリアの先頭ノードのみが追跡署名を行うため，不正者を 1 つのノードに特定することはしない．しかし，受信エリア内のノードのどれかが不正を働いたと，容疑者を絞り込むことが可能である．容疑者の候補を挙げることで，警察やシステム管理者などが不正者特定する手がかりになる．また，通信の高速化と，システム利用者少数の不正の追跡可能性，両方を考えると妥当といえる．

3.3.3 不正者追跡の動作例

図 3.1 に不正者追跡の動作例を示す．図 3.1 はノード ID1 が送信者で，ノード ID7 が受信者である場合の通信処理である．送信者は受信エリア A，受信エリア B，受信エリア C をそれぞれ ID4〜5，ID2〜3，ID6〜8 と設定している．不正者追跡を受信者であるノード ID7 から始めたとする．このとき追跡者は，受信者の検証を行わず，受信エリアの先頭を目指すためにディレクトリサーバに問い合わせて Predecessor を辿っていく．受信者が保持している追跡署名は受信エリアの先頭ノードであるノード ID6 が生成しただからである．受信エリアの先頭ノードである ID6 を特定した追跡者は，このときになって初めてノードの不正検証を行う．検証で不正が発覚した場合，受信エリアのノード ID7 と ID6 どちらかが改竄を行ったと判断される．検証で不正無しと判断された場合，ID6 へメッセージを送信した ID3 を追跡署名より特定する．ID3 は受信エリアの先頭ノードではないので，追跡者は同様にディレクトリサーバに問い合わせて Predecessor を辿る．受信エリアの先頭ノードである ID2 を特定した追跡者は検証を行い，不正が発覚すれば受信エリア B 内のノードが不正者であると判断される．同

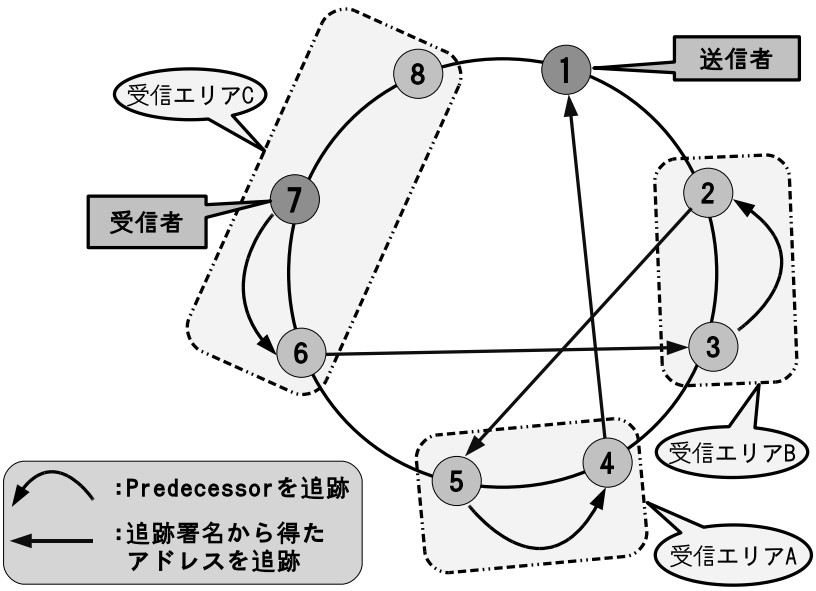

図 3.1: 不正者追跡の流れ

じようにして追跡者は受信エリア A の先頭ノードを特定し，検証を行う．最後に受信エリア A の先頭ノード ID4 の生成した追跡署名から送信者を割りだす．送信者までたどり着いた場合，今まで検証したノードが不正者でないことから，送信者である ID1 が不正者であると判断される．

第4章

実装

4.1 実装方法

本章では 3 章で提案した方式の実装方法を説明する．

4.1.1 使用ツール

オーバーレイ構築ツールキットである OverlayWeaver を使用した．これは Chord などの複数のルーティングアルゴリズムの実装を提供している．また，OverlayWeaver は Java 言語で実装されている．

4.1.2 OverlayWeaver

Overlay Weaver[10] はオーバレイ構築ツールキットの１つであり，Chord など複数のルーティングアルゴリズムの実装を提供している．P2P などのインターネット上で分散して動作するようなアルゴリズムを設計し評価する場合，可能であれば数百万ノードといった規模を想定した方が望ましい．しかし，実際に大規模な実験環境を用意することは困難であるため，通常はシミュレーションでアルゴリズムの評価を行う．OverlayWeaver はエミュレータを用いて数十万ノードの実験を行えるような環境を提供しており，新規，既存アルゴリズム間の比較を大規模に行う場合に適している．また，実装したアルゴリズムをそのまま実ネットワーク上で動作することが出来るため，アルゴリズムが直接アプリケーションに結び付く．

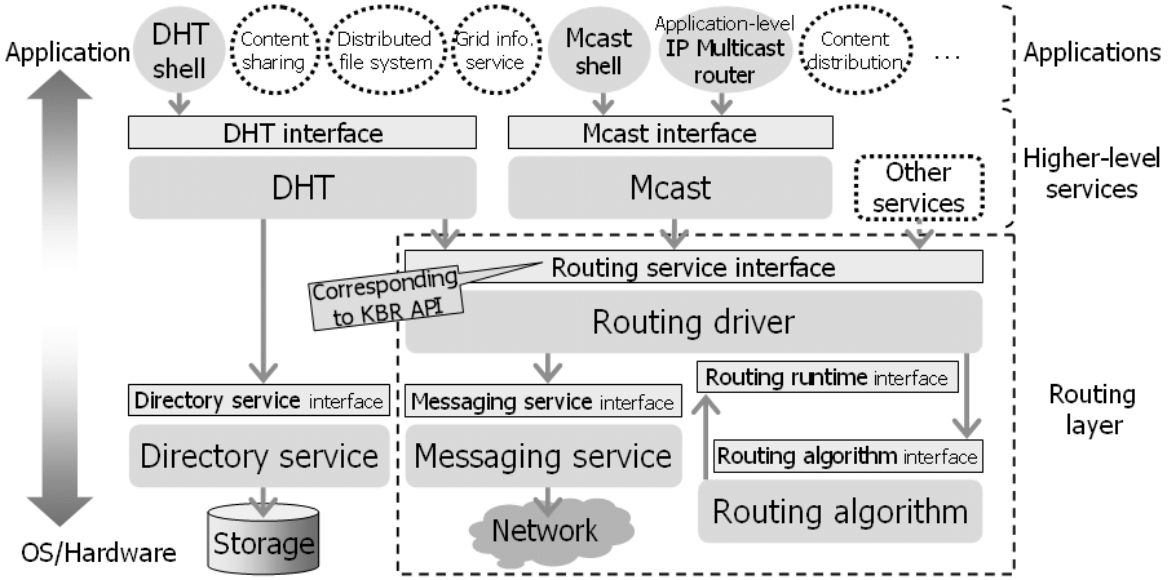

図 4.1: OverlayWeaver

4.1.3

4.1.4 公開鍵暗号による署名

本提案手法では，ノード間のデジタル署名として公開鍵暗号方式を用いている．その中でも今回はRSA 暗号を使用している．素因数分解問題が困難であることを安全性の根拠とした公開鍵暗号の一つである.

ここでデジタル署名について説明する．通常，暗号化を目的とした場合, 送信者側は受信者の公開鍵で平文を暗号化し, それを受け取った受信者側は自身の持つ秘密鍵を使って，暗号文を復号することで平文を得る．一方，公開鍵暗号方式を用いてのデジタル署名を行う場合では異なる．送信者側が自身の持つ秘密鍵で平文を暗号化し，それを受信者側が送信者の公開鍵で平文を得る．なぜなら，送信者が自身の持つ唯一の秘密鍵で暗号化することが，それによって生成された暗号文は送信者のみが生成できる，ということを意味するからである.

以上の手順でデジタル署名を行うことで，受信者は自分の意図した送信者からメッセージを受け取っていることの証明が行える．また，送受信中にメッセージが改竄されていないかも検証することが出来る.

4.1.5　共通鍵による暗号

本提案手法では共通鍵による暗号に，アメリカ合衆国の新暗号規格として規格化された共通鍵暗号方式である AES 暗号を使用している．この共通鍵は各ノードが所持しており，追跡情報を付加してから暗号化するときに用いられる．また，この共有鍵は信用できる第三者機関によって分散して管理され，閾値以上の同意を得た場合に限り，追跡情報を復号することが可能となる．

第5章

評価と考察

5.1 計測による評価と考察

本章では提案方式の暗号化時間の評価を，既存研究である RANS と比較することで評価を行う．評価はローカル環境で 8 台用いた。評価環境は以下のとおりである．

CPUCore2	Duo 2.4GHz
ネットワーク速度	100Mbps
使用台数	8 台

表 1:評価環境

5.1.1 追跡署名に要する時間の評価

提案手法と，既存研究である RANS の暗号化時間を計測する。まず，送信者が要求を転送してから返信を受け取るまでの時間 (RTT:Round Trip Time) を計測する。次に，追跡署名にかかる時間を計測する。図 5.1 に計測時のノードの配置，および受信エリアの設定を示す．参加するノードは 8 ノードで，3 ノードで構成する受信エリア A と 4 ノードで構成する受信エリア B の 2 つを設定する．受信者は受信エリア B に含まれ，先頭ノードから数えて 3 番目に位置する．つまり，RANS では受信者がメッセージを受信するまで (往路) に 7 回，返信を送信者が受け取るまで (復路) に 7 回の計 14 回追跡署名を行う．一方，提案手法では往路に 3 回，復路に 3 回の計 6 回追跡署名を行う．送信するデータは 1kB，4kB，16kB，64kB，128kB，256kB，512kB，1024kB の，

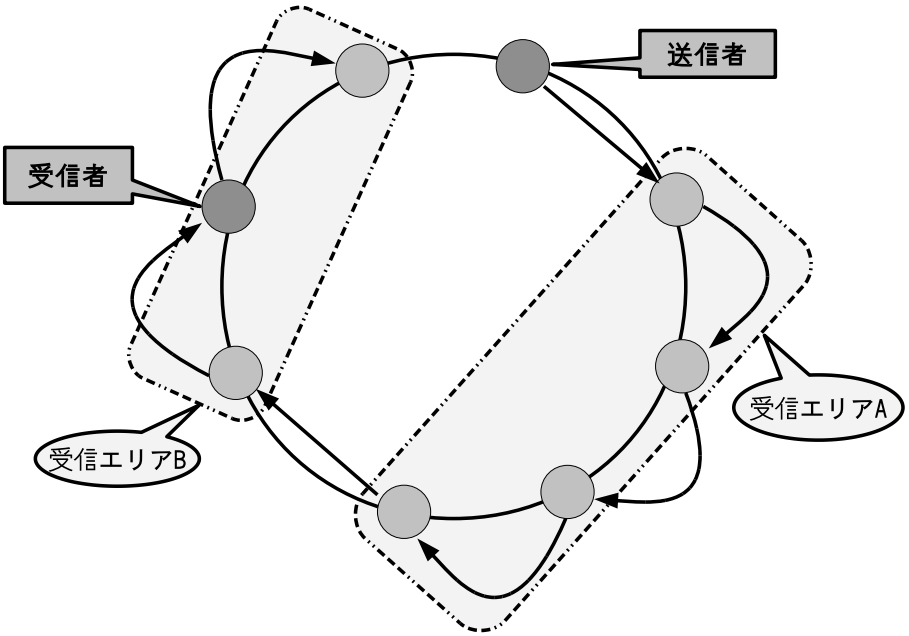

図 5.1: ノードの配置および往路での受信エリアの設定

サイズが異なるデータ 8 種類を扱う．計測はミリ秒 (ms) 単位で行う．各データサイズ毎に，10 回試行し平均した値を使って評価する．

RTT(Round Trip Time) の計測

提案手法と，既存研究である RANS における RTT を計測する．RTT とはデータを送って相手から応答があるまでの時間のことを意味し，今回は送信者がメッセージをしてから受信の旨を伝える応答を受け取るまでの時間とする．測定した結果を図 5.2 に示す．データサイズを増やすほど，追跡署名を逐次的に行う場合より，受信エリア単位で行う方が RTT は短かくなっていることがわかる．これは不正者追跡の基盤方式が追跡署名毎にメッセージを付加することでサイズが増大し，中継する程データサイズが雪だるま式に増えていくことが大きく影響している．ただ，送信データサイズを増やすほど RTT は長くなることはわかるが，図 5.2 からは追跡署名の回数が暗号化コストに影響していることについてはわからない．

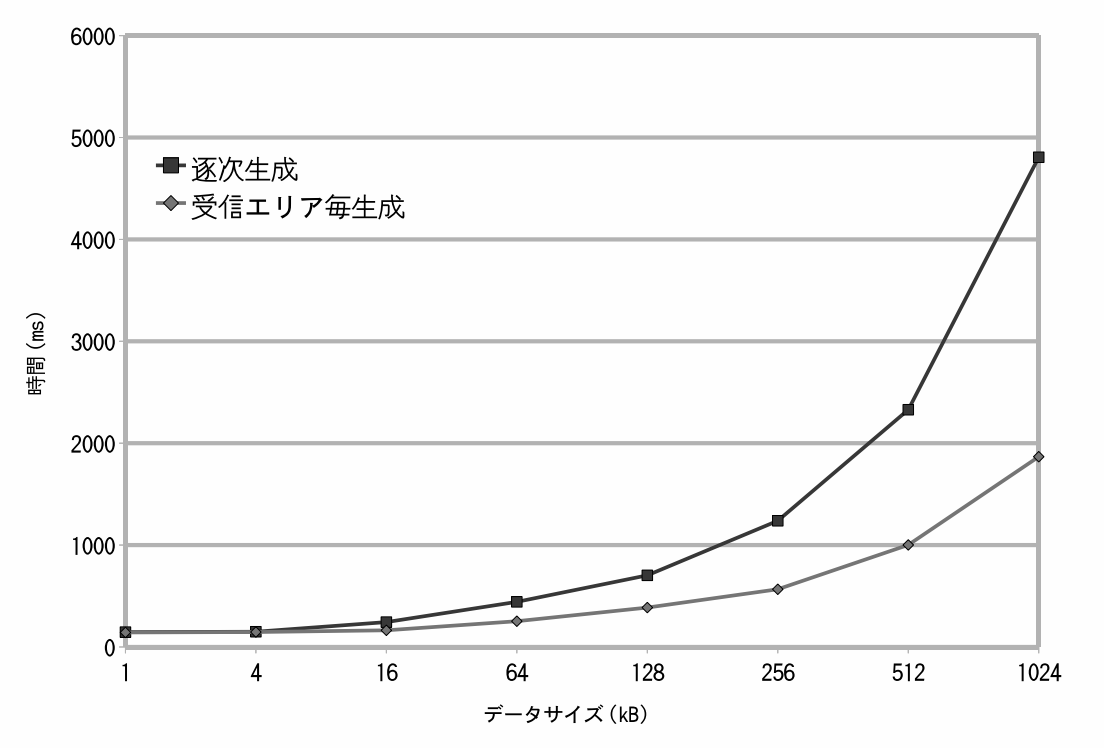

図 5.2: RTT

追跡署名時間の計測

提案手法と RANS における，全体の追跡署名にかかる時間を計測する．RTT では通信全体の時間を比較したが，ここでは各ノードがメッセージを受け取った際，追跡署名を行うにあたってかかる時間を合計した値を比較する．測定した結果を図 5.3 に示す．グラフは追跡署名時間，署名生成時間，署名検証時間の 3 項目を合計した値となっており，各データサイズに対して左の棒グラフが RANS の，右が提案方式の計測値である．追跡署名生成時間は RANS より短く，送信するデータサイズを大きくするほどその差は顕著になっている．

以上の結果から，提案方式は既存方式である RANS より暗号化コストを抑制しているといえる．

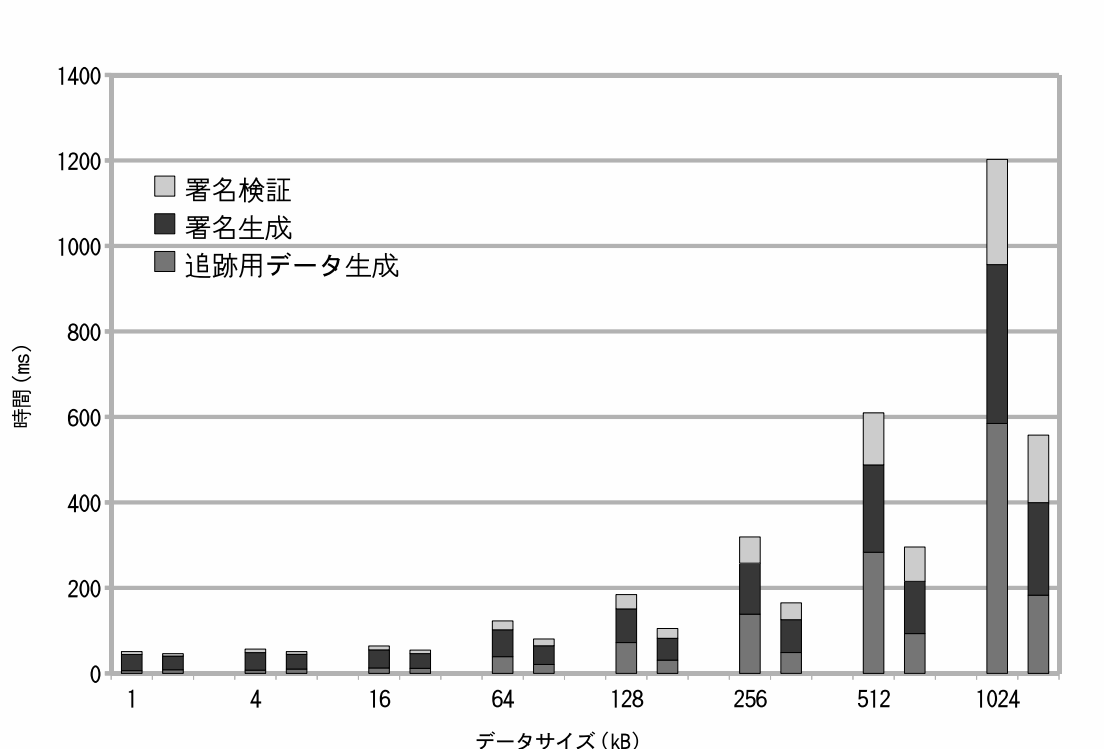

図 5.3: 追跡署名にかかる時間

第6章

まとめ

既存研究である不正者追跡可能な匿名通信路に対して，暗号化コストの削減を提案した．提案手法では，既存研究である Bifrost で定義されている受信エリアを用いている．既存研究では，メッセージが通過するノードすべてが逐次的に追跡署名を行っている．そこで，受信エリアを用いて，その先頭ノードがのみが追跡情報を生成することで，1 回の送受信に対して行われる追跡情報の生成回数を抑えた．

その代わりとして，不正者特定の精度を落とした．不正者を唯一に特定することは出来なくなったが，不正者が受信エリア内のノードにいる，と絞りこむことが可能である．膨大な通信回数に比べ，不正は極わずかであると言えるので，不正者特定の精度を落としてまで通信速度を向上させる本提案は充分妥当であるといえる．

ただ，受信エリア内に不正の容疑者を絞りこんだとはいえ，受信エリア内のノード数が多いとき，特定に時間がかかるおそれがある．よって，今回の通信速度を保ちつつも，不正者特定の精度を向上させることが今後の課題である．

謝辞

まず，本研究を進めるにあたって尽力していただいた名古屋工業大学　齋藤彰一准教授に感謝いたします．また，本研究に限らず多くの助言をしていただいた松尾啓志教授，津邑公暁准教授，松井俊浩助教，および齋藤研究室ならびに松尾・津邑研究室の皆様に深く感謝いたします．

参考文献

[1] Winny 著作権法違反幇助事件 地裁判決
(http://www.venus.dti.ne.jp/~inoue-m/hn_061213WinnyHoujyoTisai.html)

[2] 千田浩司, 小宮輝之, 林徹. 匿名性確保と不正者追跡の両立が可能な通信方式. 情報処理学会論文誌:Aug. 2004 Vol.45 No.8

[3] Goldschang, D. , Reed, M. and Syverson, P. : Onion routing for anonymous and private internet connections, Comm. ACM, Vol. 42, No. 2, pp. 3941 (1999).

[4] Douglas Wikstrom:An Efficient Mix-Net.Swedish Institute of Computer Sience ,T2002:21,December 3 ,2002

[5] Gennero, R., Jarecki, S., Krawczyk, H. and Rabin, T.: Robust and efficient sharing of RSA functions, Advances in Cryptology-CRYPTO'96,LNCS 1109,Koblitz,N.(Ed.),pp.157-172,Springer-Verlag(1996)

[6] 千田浩司, 小宮輝之, 塩野入理, 金井敦. 不正者追跡可能な匿名通信方式の実装と評価. 情報処理学会論文誌:研究報告 2005-CSEC-29(5)

[7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan (2001). " Chord: A scalable peer-to-peer lookup service for internet applications ". ACM SIGCOMM Computer Communication Review 31 (4): 149 - 160. New York, NY, USA: ACM Press. DOI: 10.1145/964723.383071

[8] Stoica, I., Morris, R., Karger, D., Kaashoek, F. and Bal akrishnan, H. : Chord: A Scalable PeerToPeer Lookup Ser vice for Internet Applications, Proc. 2001 ACM SIGCOMM Conference, pp. 149160 (2001).

[9] 近藤正基, 齋藤彰一, 石黒聖久, 田中寛之, 松尾啓志:Bifrost: A Novel Anonymous Communication System with DHT(draft), Second International Workshop on Reliability, Availability, and Security (WRAS 2009)

[10] 首藤一幸, 田中良夫, 関口智嗣:オーバーレイ構築ツールキット Overlay Weaver. 情報処理学会論文誌:コンピューティングシステム,Vol.47,No.ACS15,(2006)