# ノード結託による通信路漏洩を防止する匿名通信路修復手法の提案

八田 望† 齋藤 彰一† 松尾 啓志†

†名古屋工業大学
466-8555 愛知県名古屋市昭和区御器所町

**あらまし** 複数ノードを経由する匿名通信手法として多重暗号方式が存在する．中継するノードの一つが離脱すると経路が切断され通信が行えなくなる．再び通信路を利用するために離脱したノードの箇所に代理のノードを充て経路を修復する．しかし代理ノードの選択を中継ノードに任せると結託したノードを選択する可能性が存在する．そこで代理ノードの選択を送信ノードが行うことにより悪意のあるノードによる意図的な結託を防止する手法を提案する．これによって，匿名通信路の匿名性の向上を期待できる．

# A Proposal for Repairing Anonymous Path with Preventing Path Infomation Leakage by Colluding Nodes

Nozomu Hatta† Shoichi Saito† Hiroshi Matsuo †

†Nagoya Institute of Technology, Gokiso-cho Showa-ku Nagoya Aichi 466-8555 Japan

**Abstract** Anonymous communication systems use many relay nodes and multiple encryptions. If a replay node leaves an anonymous path, a relaying anonymous message is lost. There are related researches that a backup node repairs the broken path. However the predecessor node of the leaving node on the anonymous path can choose the backup node in these researches. If the predecessor node is malicious, it can choose a collusion node as the backup node. We propose a node failure resilient method that the sender node only chooses the backup node. This method can prevent the malicious node from choosing the collusion node as the backup node. We think this method can improve anonymity of the anonymous communication system.

## 1 始めに

インターネットの普及に伴い投票や医療相談など匿名性を必要とするシステムが増加してきた．匿名通信を実現する代表的な手法として Onion Routing [1, 2] が採用している多重暗号を用いた多段中継手法が存在する．この手法は通信メッセージを複数の中継ノードを経由して宛先とするノードに送信することによって匿名性を確保する．送信ノードは，各中継ノードと受信ノードでそれぞれ異なる共有鍵を生成し，各共通鍵を使って多重に暗号化する．このメッセージを受け取った各中継ノードは自身が保持している送信ノードとの共有鍵を使いメッセージを復号し，次の中継ノードにメッセージを送信する．最後に受信ノードが共有鍵を用いて受け取ったメッセージを復号することにより多重暗号が全て復号され平文のメッセージを受け取ることが出来る．

多段中継方式の問題点として中継ノードの離脱がある．多重暗号方式はメッセージを正しい順番で復号しなければ平文に戻すことができない．中継ノードの離脱は，暗号を施した際に利

用した共有鍵の紛失を意味し，結果としてメッセージを復号できない．

この問題点を解決する既存手法 [3, 4] が存在する．これら手法は各中継ノードが自身の近隣ノードへ送信者との共有鍵を事前に配布することで，自身が離脱した場合に近隣ノードが自身の代理として動作できるようにする手法である．これらの手法の欠点として，1) 中継ノードの離脱に備えて送信ノードとの共有鍵を代理ノードと共有する必要があること，2) 匿名通信路において離脱したノードに隣接していたノードのうち，送信ノード側に位置するノード (以降，直前ノードという) が代理ノードを選択できることの 2 点が挙げられる．もしも直前ノードが悪意を持っており代理ノードの候補に結託したノードが存在した場合，意図的に結託ノードを代理ノードとして選択することが考えられる．この問題を解決するために代理ノードの選択を直前ノードが行うのではなく，ID を利用した導出規則を導入することによって通信路漏洩を防止する手法を提案する．

本稿では，2 章で基盤研究と既存研究，既存研究の問題点について述べ，3 章で提案方式について述べる．4 章で実装と評価を示し，5 章でまとめる．

# 2 関連研究

本章では基盤研究となる多重暗号方式 [1, 2] と IBE-Bifrost [5] について述べる．次に，既存研究と既存研究の問題点について述べる．

## 2.1 多重暗号方式

多重暗号方式とは送受信ノード間に複数の中継するノードを配置し，それらがメッセージを暗号化することにより実際の送信ノードと受信ノードを隠蔽する手法である．各中継ノードは自身の前後のノードのみしか知ることができず，匿名通信路の全体像は把握できない．送信ノードは平文のメッセージを生成した後，始めに受信ノードの公開鍵を用いて暗号化する．その後，匿名通信路上の受信ノードに近い中継ノードから順に当該ノードの公開鍵を用いて繰り返し暗号化することで多重暗号が施された匿名通信用のメッセージを生成する．各中継ノードは自身が持つ秘密鍵で受け取ったメッセージを復号し，次の中継ノードへ送信する．これを繰り返すことで徐々に暗号が剥がれる．最後に受信ノードが秘密鍵によって復号することで，送信ノードが生成した平文のメッセージを取得することが出来る．受信ノードから送信ノードへ返信を送る際は中継ノードが秘密鍵で暗号化し，送信ノードが公開鍵で繰り返し復号することによって平文を取得することが出来る．

公開鍵暗号方式は暗号化と復号処理に長時間要する欠点がある．よって初回の通信時に送信ノードは各中継ノードと受信ノードに対し，ノード毎に異なる共有鍵を配布し，2 度目以降は配布した共有鍵を用いることで暗号化に必要な時間を短縮する．

## 2.2 IBE-Bifrost

提案手法を実現する上で基盤となるシステムに IBE-Bifrost [5] がある．IBE-Bifrsot は，分散ハッシュテーブル (DHT) の Chord [6] と ID ベース暗号 [7](以下 IBE という) を使用した匿名通信方式である．本節では，この手法の特徴について述べる．

### 2.2.1 DHT

多重暗号方式で匿名通信路を構築するためには，参加ノード群から通信路を決定 (中継ノードを選択) し，各中継ノードの公開鍵を入手する必要がある．IBE-Bifrost では DHT の Chord によるノード管理と IBE を用いることで，中継ノードの選択と公開鍵の入手を低コストかつ高い匿名性で行うことを可能にしている．これらを用いることにより，ディレクトリサーバが不必要となり高いスケーラビリティを実現している．

### 2.2.2 ID ベース暗号

IBE とは暗号化の鍵を ID に基づいて決定することができる暗号方式である．IBE を用いる

**for** $i = 1$ to 160 **do**
  **for** $j = 2^{i-1}$ to $2^i - 1$ **do**
    **repeat**
      $t = (random \bmod 2^{i-1}) + 2^{i-1}$
    **until** $t$ is not already assigned
    $NodeID = bit\_order\_reverse(t)$//assign ID
  **end for**
**end for**

図 1: ノード ID 割り当て方法

ことでノード ID から公開鍵を各ノード内で算出可能となり，ディレクトリサーバから公開鍵の取得は不必要となる．しかし，IBE では ID を公開鍵とみなすため複数ノードが同じ ID を持った場合，対応する秘密鍵も複数存在することになる．このような状況を防ぐために任意の ID は常に一つのみ存在する必要がある．IBE-Bifrost では，ノード ID を重複して割り当てないために，ノード ID を割り当てる専用機構である Node ID Allocator(以下 NIA という) を使用している．

#### 2.2.3 Node ID Allocator

NIA は，IBE-Bifrost に参加するノードに重複のないノード ID を割り当てる．各ノードは ID を NIA に対して要求し，NIA は ID の要求を受けるたび異なる ID を生成し返信する．これにより唯一の ID を保証する．ノード ID の割り当てアルゴリズムを図 1 に示す．$2^{i-1} \leq$ ノード $ID \leq 2^i - 1$ $(i \geq 1)$ に含まれるノード ID の集合を第 $i$ 群と呼び，第 1 群から順番に割り当てる．なお，各群内でのノード ID の割り当て順は, ランダムである．

### 2.3 ノード離脱対策手法

本節では匿名通信路修復の既存手法について述べる．

ノードの離脱対策手法として Junzhou Luo らが提案する手法 [3] がある．この手法は再暗号化用鍵 [8] を利用することで，元々は Alice が復号することを目的とした暗号文を Bob が復号できる暗号文に変換することを可能にする．各中継ノードは各々の直前ノードに対して，自身の隣接ノードの情報と送信ノードとの共有鍵を自身の公開鍵で暗号化した隣接情報を渡す．このノードが離脱した時，直前ノードは渡された隣接情報から代理ノードを一つ選択し，暗号化された送信ノードとの共有鍵に対して再暗号化を施す．隣接ノードは再暗号化によって自身の秘密鍵で復号することが可能となり，中継ノードと送信ノードが共有する共有鍵を入手する．以降は離脱したノードの代理として動作する．

次に Fengjun Li らが提案する手法 [4] がある．匿名通信路の構築時に各中継ノードは次の中継ノードを二つ選ぶ．選ばれた二つのノードは同一の共有鍵を入手する．最終的に匿名通信路の形状は送信ノードを根とした二分木の形となる．このような二分木の匿名通信路において，中継ノードの一つが離脱した場合，離脱したノードの直前ノードは離脱していない方の経路を利用するよう自身が持つ経路の情報を更新する．また二つある次ノードの両方が離脱した場合は，直前ノードに対して他方の経路を利用するように通達する．

### 2.4 既存手法の問題点

前節で述べた既存手法が持つ問題点として，共有鍵を複数ノードが共有していることが挙げられる．仮に匿名通信路中に悪意のあるノードが存在し，代理ノードの候補に結託したノードがいた場合，たとえ正規の中継ノードが生存していたとしても離脱したと偽り，結託した代理ノード候補を経由するように通信路を変更することが可能である．

## 3 提案手法

既存の匿名通信手法には意図的なノード結託の可能性が存在する欠点があった．この問題を解決するための手法を提案する．提案手法は，代理ノードの決定方法と匿名通信の暗号化に用いる共有鍵の配布方法から構成される．まず，代理ノードの決定方法は，離脱を検知した直前ノー

ドが自動的に送信ノードに離脱を通知し，送信ノードが離脱ノードの ID から代理ノードの ID を算出することで直前ノードへの代理ノードの選択権を廃止する．共有鍵の配布方法は，事前共有を廃止し，ノードの離脱後に送信ノードが共有鍵を配布する手法である．これらの提案を実現するために，中継ノードの離脱検知を匿名通信路と DHT がそれぞれ独立して行う手法を採用する．本章では，始めに代理ノードの選択規則について，続いてノードの離脱後の各ノードの動作について述べる．

### 3.1 代理ノード選択

既存手法では代理ノードの選択を直前ノードが行っていたため，悪意のあるノード間の結託が行われる可能性が存在した．そこで，代理ノードを直前ノードが選択するのではなく離脱ノードのノード ID から計算することを提案する．本手法の基盤方式である IBE-Bifrost では，2.2.3 項で述べた通りノード ID は小さな値の群から順番に選択される．よって群の値が 5 のノードが存在するならば，すでに第 4 群以下のノードは参加していることになる．そこで，例えばノード ID=$(011010)_2$ が離脱した場合，1 が立っている最下位ビットを 0 に変更したノード ID=$(011000)_2$ を代理ノードの ID とする．本算出手法は 2.2.3 項で示したノード ID 割り当て手法を用いた場合，必ず離脱ノードの群よりも小さい群 (すでに割り当て済みの群) を求めることができ，未割当のノード ID を代理ノードとすることはない．これにより，直前ノードに代理ノードの選択権を与える必要がなくなるため，意図的な結託ノードが選択される可能性はない．

### 3.2 匿名通信路修復手法

本節ではノードの離脱を検知した後の各ノードの動作について述べる．各ノードの動作をまとめると以下のように大別できる．

1. DHT における離脱検知と秘密鍵配布
2. 匿名通信路における離脱検知と修復

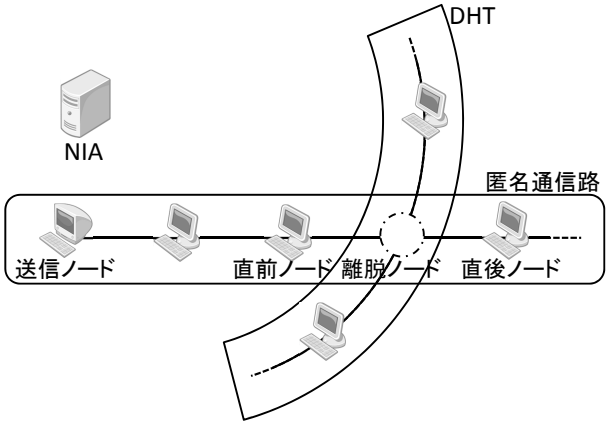


図 2: 提案手法を構成するノード群

2.2 節で述べた通り提案手法は IBE-Bifrost を基盤としていることから，ノード管理に DHT を利用している．このため，匿名通信路に含まれているノードは同時に DHT のノードとしても動作する．図 2 は，提案システムの DHT の一部と匿名通信路の一部を示している．全ノードが DHT のノードであるが，匿名通信路のノードを明確にするために DHT と匿名通信路のノードを分けて表した．また，離脱ノードを DHT と匿名通信路の両方に属すように図示した．以下，図 2 の接続に基づいて提案手法について述べる．

#### 3.2.1 DHT における離脱検知と秘密鍵配布

匿名通信路における修復手法で考えられる問題として，直前ノードが嘘の離脱通知を行う可能性が挙げられる．これに対する解決案の一つは，離脱の通知を受けた送信ノードが生存確認をする方法である．ノードの生存確認のためには離脱の通知があったノード ID に対しアクセスを試みる必要がある．しかし，この問い合わせにより，生存確認を行ったノードが送信ノードであることが第三者へ漏洩する問題がある．そこでノードの離脱の通知が正しいかを匿名通信路でなく DHT で確認する．DHT 上の各ノードは successor が離脱しているか否かについて定期的に検査を行う (図 3-(1))．提案手法では successor が離脱していることを検知した場合，NIA に対して離脱したノードの ID を通知する

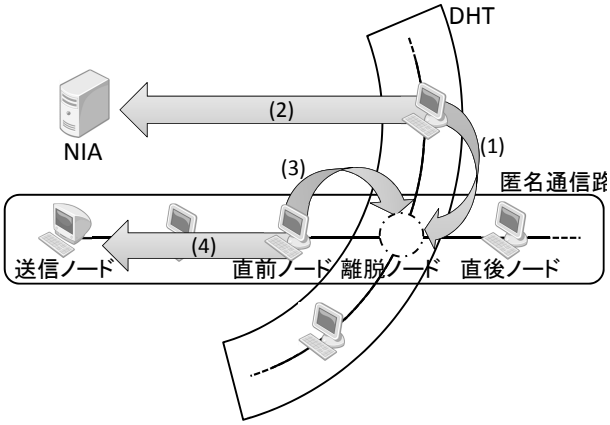


図 3: 離脱検知と通知

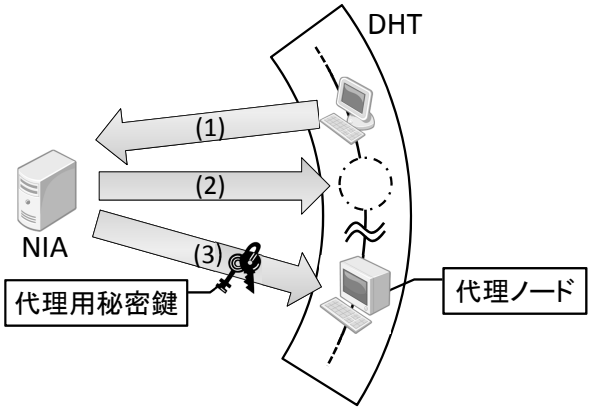


図 4: NIA の動作

(図 3-(2)).

図 4 に NIA の動作手順を示す．NIA は図 4-(1)(図 3-(2) と等しい) の通知が正しいか確認するため，通知された ID に対してアクセスを試みる (図 4-(2))．実際に離脱していることを検知した場合は 3.1 節の手法により代理ノード ID を算出し，対応するノードに対して代理用 ID に対応した IBE の秘密鍵を発行する (図 4-(3))．DHT で NIA が離脱の確認をする理由は，匿名通信路路に関して第三者である NIA が確認することによって，匿名通信路の漏洩を防止するためである．

### 3.2.2 匿名通信路における離脱検知と修復

既存手法では任意の中継ノードが離脱したとき，その直前ノードが離脱の検知後に自発的に代理ノードを選択した．しかし，この手法では

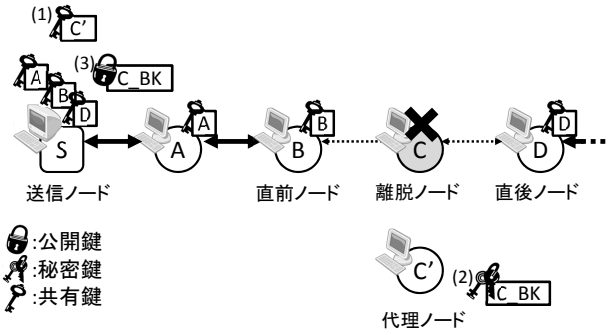


図 5: 離脱メッセージ受信時の経路状態

共謀したノードを選択する可能性があることが問題である．そこで直前ノードがノードの離脱を送信ノードに通知し，送信ノードが代理ノードを選択する手法を提案する．直前ノードは，匿名通信路の次ノード (すなわち離脱ノード) と通信できなくなった場合に離脱を検知 (図 3-(3)) し，離脱メッセージを作成して，匿名通信路を通して送信ノードに送る (図 3-(4))．このメッセージは，匿名通信路の復路 (受信ノードから送信ノードに向かう通信路を表す) を使用することから，各中継ノードでは送信ノードとの共有鍵で暗号化が行われる．このため，直前ノードは自身と送信ノードの間に位置する中継ノードとの共有鍵は不要である．また，直前ノードと送信ノードの間に位置する中継ノードから見た場合，離脱メッセージは通常の受信ノードからのメッセージと区別がつかない．メッセージを受け取った送信ノードは当該匿名通信路の共有鍵を順番に使って復号する．このとき，復号毎に離脱メッセージの有無を確認する．送信ノードが離脱メッセージを受け取った時の状況を図 5 に表す．どのノードが離脱したか把握した送信ノードは 3.1 節の手法を用いて代理ノードの ID を算出し，代理ノードとの共通鍵 (図 5-(1)) を生成する．また代理ノードは 3.2.1 項の手続きにより代理用秘密鍵 (図 5-(2)) を保持していることが分かるため，送信ノードは対応する公開鍵を IBE を用いて作成する (図 5-(3))．以上で修復用メッセージの生成に必要な情報が揃う．

次に，送信ノードは経路を再び利用できるようにするため修復用メッセージを生成する．作成する修復用メッセージの多重暗号の構成を図

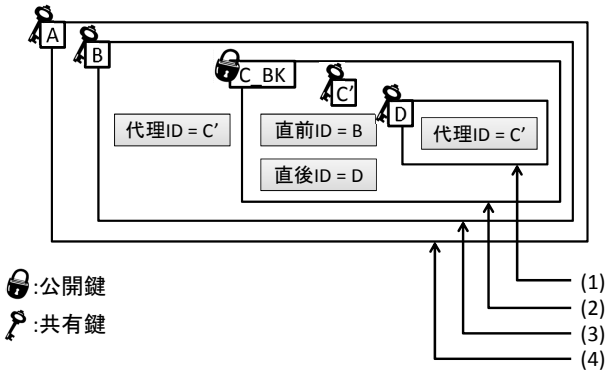


図 6: 匿名通信路修復用メッセージ

6 に示す．修復用メッセージは直後ノード，代理ノード，直前ノードに対して以下の情報を伝える必要がある

**直後ノード**
　代理ノード ID
**代理ノード**
　新しい共有鍵，直前ノード ID，直後ノード ID
**直前ノード**
　代理ノード ID

修復用メッセージの暗号化手順について述べる．始めに直後ノード (D) との共有鍵を使って代理ノード ID を暗号化する (図 6-(1))．続いて，共有鍵 (C') と直後ノード ID(D) と直前ノード ID(B)，図 6-(1) で作成したメッセージの組みを代理ノード用公開鍵 (公開鍵 C_BK) を使って暗号化する (図 6-(2))．その後，代理ノード ID(C') と図 6-(2) で作成したメッセージの一対を直前ノードとの共有鍵 (B) を使って暗号化する (図 6-(3))．もし直前ノードと送信ノードの間に中継ノードが存在するならば，直前ノードに近い該当中継ノードとの共有鍵から順に用いて暗号化を施す (図 6-(4))．

送信ノードが作成した修復用メッセージは，送信ノードと直前ノードの間に位置する中継ノードには匿名通信メッセージと区別できないため，送信ノードとの共有鍵を使い復号し次のノードへ送信する．直前ノードは送信ノードとの共通鍵で復号した時点で修復用メッセージだと認識する．修復用メッセージ (図 6-(3)) の内部に代理ノードの ID と代理ノードへの修復用メッセージ (図 6-(2)) が格納されているので，代理ノードへメッセージ (図 6-(2)) を送信する．代理ノードは 3.2.1 項の処理によって NIA から発行された代理用の IBE の秘密鍵 (秘密鍵 C_BK) を用いて，修復用メッセージ (図 6-(2)) を復号する．この復号により直前ノードの ID と直後ノードの ID，送信ノードとの共有鍵，そして直後ノードに送るメッセージ (図 6-(1)) を入手する．その後，直後ノードに対して直後ノード用の修復メッセージを送る．直後ノードは受け取った修復メッセージ (図 6-(1)) を復号することにより直前に位置するノードが代理ノードに変更されたことを把握し，以降は代理ノードに対してメッセージを送信する．

# 4　実装と評価

本章では実装に用いた環境，また評価に用いた環境について述べる．実装は言語は Java を用い，Overlay Weaver [9] 上で開発した IBE-Bifrost を改良することで実現した．

## 4.1　環境

評価環境は Core2Duo 3GHz，ネットワーク 1000Base-T の LAN 接続の PC を NIA 用に 1 台，その他通常ノード用に 32 台，計 33 台を使用した．提案手法において「直前ノードによる離脱検知時」を時間計測の開始地点,「直後ノードの修復用メッセージの受信時」を時間計測の終了時点とした．離脱ノードの位置を送信ノードから 2 つ先のノード，5 つ先のノードと変更して値を計測した．留意しておく点に，評価を取るにあたって動作手順を正規の順序から変更したことがある．DHT 上での離脱検知のタイミングによっては代理ノードが NIA から代理用秘密鍵を渡されるより先に送信ノードから修復メッセージを受け取る可能性がある．正規の動作ならばメッセージをバッファに蓄えておき，対応する代理用秘密鍵を受信してから復号する．しかし，時間計測にあたって，DHT 上での離脱検知のタイミングの影響により NIA からの

代理用秘密鍵を受け取るまでの時間にばらつきが発生することは好ましくない．そこで今回の評価では予め代理ノードに代理用秘密鍵を渡した後，ノードを離脱させて時間を計測した．

### 4.2 評価と考察

計測の結果，離脱ノードの位置が 2 ノード先と 5 ノード先，どちらの場合も匿名通信路を修復するのに要した処理時間は約 120 ミリ秒である．この結果は，処理時間の大半が送信ノードの公開鍵による暗号化と代理ノードの代理用秘密鍵による復号，代理ノードによる次ノード (直後ノード) の検索に要し，送信ノードから直前ノードまでの数に比例する通信時間と共有鍵を用いた暗号化と復号時間の影響が殆ど見られなかったためである．

なお，実際には NIA からの秘密鍵配布の時間も加わるため合計で数百ミリ秒程度は時間がかかると思われる．また今回は LAN 環境かつ比較的高性能な CPU の利用したため許容できる時間であったが，インターネット環境，また様々な CPU が混在した状況を考えるとこの数倍の時間がかかると推測される．

## 5 まとめ

本稿では多重暗号を用いた匿名通信手法に対して利用できる代理ノードの選択規則を用いた匿名通信路修復手法を提案し，IBE-Bifrost に組み込む形で実装した．提案手法は中継ノードに対して選択権を与えないためノードの結託が行われる確率は低くなる．しかし，欠点として送信ノードに中継ノードの離脱を通知する必要があり修復にかかる時間は増大する．これはトレードオフの関係であるため利用者の目的によって使い分ける必要がある．今後の課題として小さい ID 群のノードに対してバックアップノードとしての負荷が集中することが挙げられる．対応策として，定期的な IBE の秘密鍵の更新による離脱したノードの ID の再利用が考えられる．

## 参考文献


[1] Reed, M., Syverson, P. and Goldschlag, D.: Anonymous connections and onion routing, *Selected Areas in Communications, IEEE Journal on*, Vol. 16, No. 4, pp. 482–494 (1998).

[2] Syverson, P., Goldschlag, D. and Reed, M.: Onion routing for anonymous and private internet connections (1999).

[3] Luo, J., Wang, X. and Yang, M.: A resilient P2P anonymous routing approach employing collaboration scheme, *Journal of Universal Computer Science*, Vol. 15, No. 9, pp. 1797–1811 (2009).

[4] Li, F., Luo, B., Liu, P. and Chu, C.: A node-failure-resilient anonymous communication protocol through commutative path hopping, *INFOCOM, 2010 Proceedings IEEE*, IEEE, pp. 1–9 (2010).

[5] 田中寛之，齋藤彰一，松尾啓志:匿名通信におけるディレクトリサーバを用いないノード管理方式，情報処理学会論文誌， Vol. 53, No. 5, pp. 1558–1569 (2012).

[6] Stoica, I., Morris, R., Karger, D., Kaashoek, M. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *ACM SIGCOMM Computer Communication Review*, Vol. 31, No. 4, pp. 149–160 (2001).

[7] Boneh, D. and Franklin, M.: Identity-based encryption from the Weil pairing, Advances in Cryptology—CRYPTO 2001, Springer, pp. 213–229 (2001).

[8] Ateniese, G., Fu, K., Green, M. and Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage, *Proceedings of the 12th Annual Network and Distributed System Security Symposium*, pp. 29–44 (2005).

[9] Shudo, K., Tanaka, Y. and Sekiguchi, S.: Overlay weaver: An overlay construction toolkit, *Computer Communications*, Vol. 31, No. 2, pp. 402–412 (2008).