情報処理学会論文誌 Vol.53 No.5 1558–1569 (May 2012)

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

田中 寛之[1,†1,a)] 齋藤 彰一[1] 松尾 啓志[1]

受付日 2011年8月26日, 採録日 2012年2月3日

概要：インターネットにおけるプライバシ保護の重要性が増しており，様々な匿名通信方式が提案されている．多くの既存方式では，ディレクトリサーバによる公開鍵配布と参加ノード管理が行われている．しかし，ディレクトリサーバは，スケーラビリティを低下させるという問題がある．我々は，ID ベース暗号と，ノードの参加状況を各ノードが有する情報のみで把握可能なノード ID 割当て方式を導入することでスケーラビリティを向上させる．また，ディレクトリサーバを用いない宛先ノード検索方式として，Tor の Introduction Point における情報漏洩問題を解決する方式を提案する．本提案方式のプロトタイプを作成した結果，ディレクトリサーバを用いずに匿名通信路を構築できることを確認した．本稿では，提案方式の構成方法を述べ，基本的な性能評価と考察について述べる．

キーワード：匿名通信，オーバーレイネットワーク，ID ベース暗号，分散ハッシュテーブル

Node Management without Directory Servers for Anonymous Communication System

Hiroyuki Tanaka[1,†1,a)] Shoichi Saito[1] Hiroshi Matsuo[1]

Received: August 26, 2011, Accepted: February 3, 2012

Abstract: Severity of revealing personal information is increased. Many anonymous communication systems have been studied. The systems usually use directory servers to manage participant nodes and those public keys. However, this way reduces scalability. We improve scalability by introducing ID-based encryption and a new method of Node ID allocation. The Node ID allocation method can grasp already assigned Node IDs only with their DHT routing table. Furthermore we propose a new method to solve information leaks of Introduction Point of Tor. Our prototype system can construct anonymous communication routes without any directory servers and can communicate over them. This paper describes structure of a proposed system, basically performance analysis and considerations.

Keywords: anonymous communication, overlay network, ID-based encryption, distributed hash table

1. はじめに

多種多様なサービスがインターネットで利用可能である．その中には，医療相談や人権相談等相談者の匿名性が重要なサービスもある．また，第三者に通信相手を特定されたくない状況もある．そのため，通信における匿名性を実現するために匿名通信の研究が行われている．匿名通信は，送信ノード以外が，送信ノードを特定できないこと，宛先ノードを特定できないこと，送信ノードと宛先ノード間の通信を追跡できないことの 3 つの要件を満たす必要がある [1]．以下，これらをまとめて**匿名性**といい，匿名性を備えた通信を**匿名通信**，匿名通信が使用する通信路を**匿名通信路**という．なお，本稿では，匿名通信システムに参加するコンピュータをノードという．

匿名通信を実現する代表的な方式は，Onion Rout-

[1] 名古屋工業大学
Nagoya Institute of Technology, Nagoya, Aichi 466–8555, Japan
[†1] 現在，日本電気株式会社
Presently with NEC Corporation
[a)] shoichi@nitech.ac.jp

ing [2], [3] 等が採用している多重暗号を用いた多段中継方式である．この方式は，通信メッセージを複数の中継ノードを介して宛先ノード（匿名通信システム内の Web サーバ等や，匿名通信システム外にある Web サーバ等に接続する出口ノード）まで送る方式である．そのとき，各中継ノードに送信ノードと宛先ノードが分からないようにするために，送信ノードは，メッセージを各中継ノードと宛先ノードの公開鍵で多重に暗号化する．ここで，この方式の匿名通信路に使用される中継ノードと宛先ノードは，送信ノードが指定する．送信ノードは，中継ノードの候補として匿名通信システムに参加しているノード群の IP アドレスやノード ID と公開鍵を取得する必要がある．そのため，ノード管理と公開鍵管理が必要である．

ノード管理の主な方式として，ディレクトリサーバを使用する方式と分散ハッシュテーブル（Distributed Hash Table: DHT）を使用する方式がある．公開運用されている匿名通信システムの Tor [4] では，分散ディレクトリサーバ方式が用いられている．しかし，ディレクトリサーバ方式には一般に問合せ量に対するスケーラビリティが欠ける問題がある．このため，DHT を使用した匿名通信システムが各種提案されている [5], [6], [7], [8]．DHT を用いることで，ノードを集中管理する必要がなくなりスケーラビリティを向上させることができるという利点がある．

次に，公開鍵管理の主な方式として，ディレクトリサーバや Certificate Authority（CA）による管理方式と，**ID ベース暗号**[9]（ID-Based Encryption: IBE）を用いた方式がある．ディレクトリサーバや CA を使用することによるスケーラビリティの低下は，ノード管理の場合と同様に発生する．一方，IBE を使用した方式 [10] が提案されている．IBE を使用することで，通信先のノードに割り当てられたノード ID から公開鍵を算出することが可能となり，ディレクトリサーバ等の外部システムの参照が不要となる．このため，スケーラビリティは低下しない．しかし，IBE で匿名通信を成立させるためには，送信ノードは通信路の各ノード（中継ノードと宛先ノード）の正確なノード ID を知る必要がある．ディレクトリサーバを使用できれば正確なノード ID の入手は容易であるが，スケーラビリティの向上のためにはディレクトリサーバを使用できない．したがって，IBE を有効活用するために，スケーラビリティを有するノード ID 管理方式の実現が，公開鍵管理の課題であるといえる．

以上より，スケーラビリティのある匿名通信方式には，DHT によるノード管理方式と，ディレクトリサーバを用いない IBE による公開鍵管理方式の組合せが適切である．この方式を実現するためには，次の課題がある．IBE の使用においてノード ID から公開鍵を算出するためには，1) 重複のないノード ID 割当てと，2) 割当て済みのノード ID の判断ができなければならない．また，DHT の使用においては，一般的に，ノード ID に乱数性があることから，3) 宛先ノードのノード ID を特定できなければならない．我々は，ディレクトリサーバを用いずにこれらの課題を解決するノード ID 割当て方式と宛先ノードのノード ID 取得方式を提案する．割当て方式は，外部参照を行うことなく各ノードが持つ情報のみで匿名通信システムに参加している全ノード（以下，参加ノードという）の ID 割当て状況を把握可能にし，IBE に必要なノード ID を特定できる方式である．ノード ID 取得方式は，Introduction Point 方式 [4] を拡張して IBE と組み合わせることで匿名性の向上とディレクトリサーバを不要とした方式である．

以下，2 章で匿名通信の概要と問題点について述べる．次に，3 章で IBE を匿名通信方式に取り入れるための課題について述べ，4 章で IBE 導入のための課題に対する方式を提案する．5 章で，宛先ノードのノード ID 取得に対する提案方式について述べる．6 章で性能評価を示し，7 章で提案方式について考察を行う．最後に 8 章でまとめる．

2. 匿名通信の概要と問題点

本章では多重暗号を用いた匿名通信方式の概要と，この方式の 2 つの問題点について述べる．

2.1 多重暗号を用いた多段中継方式

多重暗号は，暗号処理を入れ子で行う方式であり，暗号化の順番と逆順に復号することで平文メッセージを取得できる方式である．多重暗号による多段中継方式では，平文メッセージを，送信ノードが各中継ノードと宛先ノードの公開鍵で受信する順番の逆順に暗号化する．これを中継する各中継ノードは，自身の秘密鍵で復号し，復号したメッセージを次ノードに中継する．最後に，宛先ノードは自身の秘密鍵で復号することで平文メッセージを取得する．また，復路は，宛先ノードは自身の秘密鍵で暗号化して中継ノードに送る．各中継ノードは往路とは逆順に暗号化と中継を行う．送信ノードは，宛先ノードと各中継ノードで暗号化されたメッセージを受け取る．これを，復路の中継順とは逆順に復号することで復路の平文メッセージを取得する．なお，共有鍵でも多重暗号は可能である．

多段に中継しながら匿名性を保つには，各中継ノードが送信ノードと宛先ノードを特定できてはならない．つまり，各中継ノードは，自身が直接通信する前後の 2 ノード以外を知ることができず，かつ，匿名通信路全体を知ることができない必要がある．このために，匿名通信路構築時には，多重暗号メッセージを復号した中継ノードが，経路情報（次ノードの IP アドレスやノード ID）を平文で入手できるようにメッセージを構築する．さらに，指定された中継ノードのみが復号できるようにするために，初回メッセージ（匿名通信路構築時）は公開鍵暗号を使用する．このときに共有鍵を交換し，2 回目以降のメッセージは共有

鍵暗号を使用する．

共有鍵の代表的な交換方式として，メッセージ中に共有鍵を埋め込むOnion方式と，Diffie-Hellman鍵交換方式（以下，DH法）を各中継ノードと順番に行うTelescoping方式[4]がある．Onion方式では，送信ノードが共有鍵を生成して，各中継ノードに経路情報と同時に配布する．一方，Telescoping方式では，匿名通信路をn番目の中継ノードへと伸ばすためには，$n-1$番目までの中継ノードへのメッセージを共有鍵で多重暗号し，n番目の中継ノードへのメッセージのみ公開鍵で暗号化する．送信ノードは，この暗号化メッセージにより各中継ノードとDH法を順番に行い，1台ずつ共有鍵を生成して匿名通信路を構築する．これらの方式は多数の匿名通信方式[2], [3], [4], [5], [6], [7], [11], [12]で使用されている．

2.2 ノード管理と公開鍵管理の問題点

多重暗号による匿名通信路を構築するためには，送信ノードは，参加ノードから中継ノードを選択し，選択した各中継ノードの公開鍵を入手する必要がある．参加ノードの管理方法と公開鍵の入手方法として，1つ目に定期的に多数の参加ノード情報を取得する方式（Tor[4]），2つ目にDHTによるノード管理とディレクトリサーバやCAによる公開鍵取得を組み合わせる方式（Cashmere[5]，Bifrost[6]，Bluemoon[8]），3つ目にIBEとディレクトリサーバを組み合わせる方式[10]がある．

1つ目の参加ノード情報とその公開鍵を定期的に多数取得する方式では，送信ノードは取得した参加ノード群から任意のノードを中継ノードとして選ぶ．中継ノード選択ごとの通信が必要ないという利点があるが，定期的に参加ノード情報を取得する必要があるためスケーラビリティに欠けるという問題がある．

2つ目のDHTを用いる方式では，DHT空間に含まれる任意の値を指定して，その値を管理するノードを中継ノードとする．この方法では，各中継ノードが指定された値を用いて次ノードを検索して中継するため，送信ノードが中継ノードのIPアドレスを調べる必要がない．しかし，公開鍵の入手にはディレクトリサーバを検索する必要があるため，スケーラビリティの問題がある．

3つ目のIBEとディレクトリサーバを組み合わせる方式では，IBEを用いることで公開鍵を各ノード内で算出が可能である．しかし，送信ノードは，ディレクトリサーバからノード情報を取得して，そこに含まれるノード群から中継ノードを選択する必要がある．つまり，この方式においてもディレクトリサーバによるスケーラビリティに欠けるという問題がある．

以上より，1つ目の方式は，ノード管理と公開鍵管理の双方でスケーラビリティがない．2つ目の方式は，ノード管理にはスケーラビリティがあるが公開鍵管理にはない．3つ目の方式は，ノード管理にはスケーラビリティはないが，公開鍵はIBE方式によりスケーラビリティがある．これらの利点を活用する方法として，2つ目の方式のDHTによるノード管理方式と3つ目の方式のIBEによる公開鍵管理方式を組み合わせる方式が必要である．

2.3 宛先ノードID取得の問題点

DHTをノード管理に使用する場合，送信ノードは何らかの公開情報から受信ノードのノードIDを特定しなければならない．しかし，ノードIDは乱数性があるために静的には定めることができないことから，何らかの検索が必要である．本節では，ノードIDの乱数性の必要性と，ディレクトリサーバを使用しない検索方法について問題点を述べる．

2.3.1 ノードIDの乱数性

DHTを用いる既存匿名通信方式のノードID割当ては，主にIPアドレスとポート番号をハッシュにかけた値か乱数値が使用されている．ノードIDに乱数性を持たせる理由には以下のものがある．なお，CashmereとBluemoonにおいても乱数によるノードIDが使用されているが，各文献[5], [8]において理由は述べられていない．

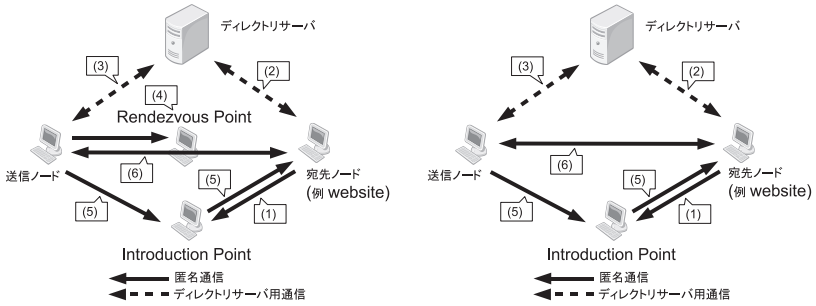
- ノードIDが単純な昇順で割り当てられた場合，DHTのID空間においてノード配置に偏りが生じて検索効率が著しく低下するため[7]．
- 攻撃者が通信を監視しやすい位置（ノードID）に参加するのを防ぐため[6], [13]．

以上から，DHTをノード管理に用いる匿名通信システムでは，ノードIDの割当てに乱数性を持たせる必要がある．このため，宛先ノードのノードIDは動的に変化することから，メッセージを送るためには匿名通信路構築前に宛先ノードのIDを調べる方法が必要である．しかし，公開鍵管理と同様にディレクトリサーバは使用できない．

2.3.2 ディレクトリサーバを用いない宛先ノードの検索

ディレクトリサーバを用いない検索方法には次の3つの方法が考えられる．1つ目の方法は，宛先ノードとなりうるサービスを提供する全ノードのノードIDをフラッディングによりシステム全体に知らせる方法である．この方法は，宛先ノードのノードIDが変化するたびに再送信する必要があるため，スケーラビリティに欠けるという問題がある．2つ目の方法は，既存のSecure Service Discovery方式[14]である．この方法は，検索キーワードを平文で流すため，第三者がどのようなサービスが提供されているのか，さらに，検索されているのかを容易に知ることができるという問題がある．最後に，Torで提案されているIntroduction Point方式である．

Introduction Point方式について簡単に述べる．Introductin Point方式は，宛先ノードの匿名性を確保する非公開サーバのために，接続要求用と通信路用に2つの一時



a) Tor/Telescoping 型
a) Tor/Telescoping style

b) Bifrost/Onion 型
b) Bifrost/Onion style

図 1 Introduction Point 方式
Fig. 1 Introduction Point method.

的な接続ノードを設ける方式である．Introduction Point 方式の接続の流れを**図 1** に示す．元々は Tor による Telescoping 方式用の実現方式（図 1-a 参照）があり，それを Bifrost が Onion 方式に適用した方式（図 1-b 参照）がある．以下，Tor による接続の流れを述べる．自身の IP アドレスを公開しない宛先ノードは，1) 複数の公開接続ノード（Introduction Point ノード）を定めて，それらノードと自ノードとの間に匿名通信路を構築し，2) ディレクトリサーバに Introduction Point ノードを登録する．その宛先ノードに接続を希望する送信ノードは，3) ディレクトリサーバでその宛先ノードの Introduction Point ノードを検索する．次に，4) 通信路用の接続ノードである Rendezvous Point ノードを作成し，5) Introduction Point ノード経由で宛先ノードに接続要求メッセージを送る．この接続要求メッセージにより Rendezvous Point ノードを伝達する．この後，6) 宛先ノードと送信ノードは，Rendezvous Point ノードを経由して匿名通信路を構築するという方式である．また，Bifrost による方式は，匿名通信路構築が Onion 方式であることを活かし，5) で接続要求メッセージとともに通信路構築情報を宛先ノードに送り，6) で宛先ノードから通信路を構築する方式である．このため 4) の Rendezvous Point ノードが不要となっている．Introduction Point 方式の問題は，Introduction Point ノードの接続機能が 1 つの宛先ノード専用であるために，Introduction Point ノードは当該宛先ノードへのアクセス状況を知ることができる点である．これにより，宛先ノードの利用状態の一部が第三者（Introduction Point ノード）に漏れるという問題が発生し，通信トラフィックを解析して通信路を特定する攻撃が容易になる可能性がある．

以上より，ディレクトリサーバを使用しない宛先ノードの検索には，Introduction Point ノードを 1 つの宛先ノードの接点とする問題を解決した方式が必要である．

3. IBE の導入とその課題

本章では，IBE の概要と，IBE 導入のための 2 つの課題について述べる．

3.1 ID ベース暗号

ID ベース暗号とは，暗号化を ID（任意の文字列，以下ノード ID と区別するために IBE-ID という）に基づいて行う暗号方式である．IBE の利用法の 1 つに，公開鍵暗号として利用するものがある．宛先ノードや中継ノードのノード ID を IBE-ID として，共通ハッシュ関数を用いて宛先ノードの公開鍵を計算して暗号化する．共通ハッシュ関数とは，すべての参加ノードが共有するハッシュ関数である．IBE では，信頼できる第三者が運用する秘密鍵生成局（Private Key Generator: PKG）と呼ばれる機関が存在する．この PKG がマスタ秘密鍵と共通ハッシュ関数の生成を行い，共通ハッシュ関数を全参加ノードに周知させる．また，PKG がマスタ秘密鍵と各利用者の IBE-ID を用いてそれぞれの秘密鍵を生成する．IBE では IBE-ID と共通ハッシュ関数の信頼性に基づいて安全性を確保するため，宛先ノードの公開鍵とその証明書を検証する必要がない．

3.2 IBE の導入

匿名通信路構築時における公開鍵での暗号化に IBE を導入することで公開鍵取得のための検索をなくすことができる．つまり，宛先ノード ID を取得し，それを IBE における公開鍵の基になる IBE-ID とすることで別途公開鍵を入手する必要がない．これにより，ディレクトリサーバが不要な匿名通信方式が実現できる．なお，文献 [10] においても IBE を利用した匿名通信方式が提案されている．この方法は，IBE を共通鍵生成に使用しており，ノード情報の管理にはディレクトリサーバ等が必要である．IBE により公開鍵取得を不要としている点では同じであるが，本方式とは利用の目的が異なる．

3.3 IBE 導入の課題

IBE を匿名通信方式に取り入れてディレクトリサーバを除外するためには，以下の解決すべき課題がある．本節では，これらの課題を定義する．これらの課題の解決策は次章で詳しく述べる．

3.3.1 重複の無いノード ID 割当ての課題

PKG がノード ID を IBE-ID と見なして，その ID に対応する秘密鍵を発行するためには，各ノードにノード ID が重複して割り当てられてはならない．もし，ノード ID が重複して割り当てられた場合には，そのノード ID で暗号化されたメッセージは当該ノード ID を持つすべてのノードが復号できるという問題が発生する．このため，ノード ID を重複することなく割り当てる仕組みが必要である．

3.3.2 割当て済み ID の判断の課題

IBE では，復号できる者がいない IBE-ID を用いても公開鍵を生成して暗号化が可能である．しかし，未割当てのノード ID を IBE-ID として匿名通信に用いた場合，復号できるノードが存在しないため，メッセージが宛先ノードに届かない．特に，DHT をノード管理に用いた場合には，DHT の検索により得られる ID の担当ノードが受信する場合がある．この場合，担当ノードのノード ID と IBE-ID が異なることから，受信した担当ノードは復号できない．したがって，IBE を使用する場合には，復号する担当ノード ID と IBE-ID を正確に一致させる必要がある．そのため，割当て済みのノード ID を識別して，そのノード ID を有するノードのみで通信を行う必要がある．このとき，あるノード ID が割当て済みか調べるためにディレクトリサーバを用いるとスケーラビリティが低下する．したがって，ディレクトリサーバを用いないでノード ID の割当て状況を確認する方法が必要である．

4. IBE の導入の課題に対する提案方式

本章では，3.3 節で述べた IBE 導入の課題を解決する方式を提案する．以下，各課題ごとに提案方式を述べる．

4.1 重複のないノード ID 割当て方式

IBE 導入によりノード ID を IBE-ID と見なして公開鍵を計算することから，ノード ID が重複して割り当てられてはならない．そのため，**ノード ID 割当て局**（NodeID Allocator: NIA）を用意して，ノードの参加時に重複のないノード ID の割当てを行う．NIA は信頼できる第三者が運用する機関が運用し，PKG との間で信頼性を確立すると仮定する．PKG が NIA の役割を担ってもよい．

NIA と PKG を用いたノードの新規参加手順を次に示す．NIA は，ノード ID を割り当てるメッセージに RSA を用いたデジタル署名を行う．PKG は，NIA のデジタル署名を確認することでノード ID の正当性を確認し，ノード ID に関連する秘密鍵を生成して当該参加ノードに配布する．

(1) 参加ノードが NIA にノード ID 割当て要求を送る．
(2) NIA がノード ID を決定し，RSA を用いたデジタル署名を施して通知する[*1]．NIA のノード ID 割当ては 4.2 節の方式とする．
(3) 参加ノードが，割り当てられたノード ID を PKG に通知する．
(4) PKG は，NIA のデジタル署名を確認し，ノード ID に対応する秘密鍵を生成して共通ハッシュ関数とともに参加ノードに返信する．

NIA のようにノード ID を集中的に発行する方式は，ノード ID の偽造も防ぐことが可能となり，DHT に対する Sybil 攻撃 [15] を防ぐ手法として用いられている．なお，各参加ノードと NIA，PKG との間の通信は SSL によって行う．

[*1] 参加ノードは，PKG と NIA の RSA を用いた公開鍵を取得済みとする．

4.2 割当て済みノード ID の判断方式

スケーラビリティを低下させずに割当て済み ID のみで暗号化するには，ノード ID を検索することなく割当て済みノード ID を取得できる必要がある．しかし，検索を行うことなしに，最新の割当て済みノード ID を取得することは困難である．そこで，割当て済みノード ID 取得の目的が最新のノード ID を得ることではなく，メッセージを復号するために確実に割当てが完了しているノード ID を得ることであることに注目し，グループ単位でノード ID の割当て状況を得る方法を提案する．

グループ単位でノード ID を割り当てることで，最新グループに属するノードを 1 つでも認知した既存ノードは，それ以前のグループにおけるノード ID 割当てが完了したことを知ることができる．これは，ノード管理に DHT を使用していることから，DHT の経路表を使用することで，匿名性を損なうことなしに多数のノード情報を収集することができる点からも有効である．本方式により，匿名性を低下させずに，またディレクトリサーバを用いることなく，割当てが完了したノード ID を特定できる．本節では，グループ分割の方法とノード ID 割当て方法について詳細に述べる．

4.2.1 グループ分割手法

グループに必要な要件として，次の 2 点がある．1) DHT が効率良く動作するために ID 空間の全体に分散して配置すること．2) ある新規グループのノードの割当てが完了したときに，既存グループのすべてのノードの DHT の経路表に，最低 1 つは新規グループのノードが含まれていることがある．これら 2 点を実現するグループ分割手法について述べる．

グループの分割方法として，まず，各ノードには **JoinNumber** と呼ばれる値を参加順に割り当てる．この JoinNumber を 2 進数表記しビット単位で上位と下位を入れ替えて逆順とする（6 ビットの例：000001 → 100000, 000010 → 010000）．この逆順の値をノード ID とする．次に，JoinNumber を基にして，複数のノードをまとめた **JoinGroup** というノードグループを導入する．第 i JoinGroup には，JoinNumber が 2^{i-1} から $2^i - 1$（$i \geq 1$）

表 1 JoinNumber とノード ID と JoinGroup の関係（ノード ID が 6 bit，ノード数が 16 台の場合）

Table 1 Relation of JoinNumer, NodeID and JoinGroup when NodeID consists of 6 bits and Number of Node is 16.

JoinNumber		ノード ID		JoinGroup
000000	(0)	000000	(0)	0
000001	(1)	100000	(32)	1
000010	(2)	010000	(16)	2
000011	(3)	110000	(48)	2
...		...		...
001100	(12)	001100	(12)	4
001101	(13)	101100	(44)	4
001110	(14)	011100	(28)	4
001111	(15)	111100	(60)	4

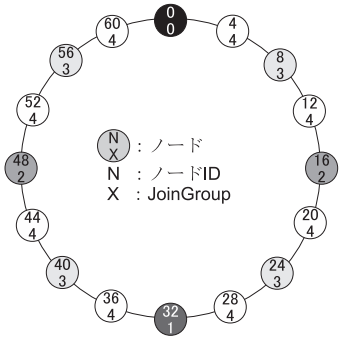


図 2 ノード ID と JoinGroup（ノード ID が 6 bit の場合）

Fig. 2 NodeID and JoinGroup when NodeID consists of 6 bits.

までのノードを含ませる[*2]．つまり，JoinNumber でビットが 1 となる最上位の桁数が JoinGroup の番号である．JoinNumber とノード ID と JoinGroup の関係について，ノード ID が 6 bit で第 4 JoinGroup まで割り当てられた場合を**表 1** に示す．さらに，DHT の ID 空間に各ノードを配置した場合のノード ID と JoinGroup の関係を**図 2** に示す．ここで，JoinNumber が昇順に割り当てられていることから，JoinGroup も昇順となる．よって，第 N JoinGroup が割り当てられているときには，第 $N-1$ JoinGroup 以下の JoinGroup の割当てが完了していることが分かる．さらに，図 2 から，ノード ID は ID 空間全体に均等に分散していることが分かる．これにより，DHT による検索効率を下げることはない．

図 2 より，この方法による新規割当てノード ID は，すでに割り当てられたノード ID の中間に位置する．つまり，各ノード自身もしくは隣接ノードは，新規割当て中の JoinGroup かその 1 つ前の JoinGroup に属する[*3]（以下，これら各ノードが知りうる最新の JoinGroup を第 R JoinGroup とする）．これにより，各ノードは，自身と隣接ノードの JoinGroup を調べるだけで，第 R JoinGroup を得ることができる．また，各ノードの経路表にはより多くのノード ID が含まれていることから，経路表に含まれるノード ID の JoinGroup を調べることで，より正確に第 R JoinGroup を知ることができる．このようにして得た第 R JoinGroup はまだノード ID の割当てが完了していないが，1 つ前の第 $R-1$ JoinGroup では割当ては完了している．これにより，割当てが完了したノード ID を，第 $R-1$ 以前の JoinGroup に属するノード ID として特定することができる．以上より，本ノード ID 割当て方式では，ノード ID 検索を行うことなしに，JoinGroup 単位で割当て済みノード ID を特定することが可能である．

[*2] ただし，第 0 JoinGroup にはノード ID が 0 のノードだけが含まれるとする．

[*3] 新しい JoinGroup に割当てを開始した直後は，最新の JoinGroup に属するノードが少ないため，全ノードが最新の JoinGroup を知ることはできないためである．

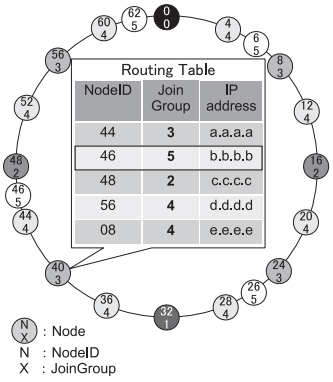


図 3 割当て済みノード ID の確認

Fig. 3 Verification of assigned NodeID.

4.2.2 ノード ID 割当ての乱数性

2.3 節で述べたように匿名通信のノード ID 割当てには乱数性が必要だが，4.2.1 項の方式には乱数性がない．そこで，部分的な乱数性を本割当て方式に追加する．本方式では，JoinGroup 単位にノード ID 割当て状況を得るために，JoinNumber が昇順に割り当てられていることを仮定している．この仮定は，JoinGroup に対する 2 つの仮定へと読み変えることができる．仮定 1 として，JoinGroup は昇順に割り当てる．仮定 2 として，新規 JoinGroup の割当ては 1 つ前の JoinGroup に属する全 JoinNumber（ノード ID）の割当てが完了してから行う．これらの仮定では，同一 JoinGroup 内におけるノード ID の割当て順は制約を受けないことが分かる．この制約がないことを利用して，同一 JoinGroup 内のノード ID の割当てに乱数性を持たせることができる．たとえば，**図 3** は，第 4 JoinGroup までが割当て済みで，第 5 JoinGroup が割当て途中の状況において，ノード 40 が割当て済み ID の確認をする様子を示している．なお，第 5 JoinGroup 内の割当て順はランダムである．図 3 のノード 40 は，経路表に 5 つのノードを有しており，各ノードの JoinGroup を求めた結果からノード 46 の第 5 JoinGroup が最新であることが分かる．このように，グループ内の割当て順に乱数性を持たせたとしても，グループ単位での割当て状況把握には問題はない．以上よ

り，検索なしで割当て済みノードIDを知りうる乱数性を保持したノードID割当て方式が実現できる．

4.3 第 R JoinGroup のノードの通信方法

本ノードID割当て手法では，第 $R-1$ 以前のJoinGroupへの割当てが完了している．そのため，送信ノードは第 $R-1$ 以前のJoinGroupに属するノードIDを中継ノードとして選択すれば，IBEによる復号が可能となる．本節では現在割当て中の第 R JoinGroupについて考える．

第 R JoinGroupは，割当てが完了していないことから，中継ノードに選択しても当該ノードが存在しない場合がある．したがって，第 R JoinGroupを中継ノードに選択することはできない．しかし，第 R JoinGroupに配置されたノードは，中継ノードに選択されないことで，他のノードのメッセージを中継する機会がない．このため，第 R JoinGroupのノードが通信を行う場合，自身が発する通信と中継による通信を混在することができず，当該ノードの通信はすべて自身が開始した通信であることが容易に判明する．これにより，第 R JoinGroupに属するノードは，第 $R+1$ JoinGroupの割当てが開始するまで自発通信ができないことになる．しかし，このような状況は許容できないことから，第 R JoinGroupのノードによる自発通信を可能にする方法を以下に述べる．

第 R JoinGroupのノードの自発通信を可能にするには，他のノードからのメッセージを中継することで，自発通信と中継通信の2種類を混在させる必要がある．ここで，第 R JoinGroupのノードを中継ノードに選択できない理由を考える．理由は，中継ノードに選択されたノードが存在しない場合にメッセージが復号されず，メッセージが宛先ノードに到達しない可能性があり中継ノードとして機能しないためである．したがって，中継ノードとして機能させるためには，復号できない場合に中継するノードをあらかじめ定めておき，確実に宛先ノードに到達することを保証すればよい．以下，この第 R JoinGroupのノードを含む，メッセージを復号できなかったノードが選択するノードをdefault routeという．default routeを定めることで，復号の可否に依存することなしにあらゆるノードを匿名通信路に組み入れることが可能となる．つまり，default routeの決定方法を全ノードが共有することで，送信ノードはdefault routeによる中継を含めた匿名通信路の経路設計が可能になる．

ここで，default routeは，全ノード共通の特定ノードではなく，各ノードからの相対的なID距離にあるノードとする．これにより，各ノードごとに異なるノードをdefault routeとして利用でき，また全ノードがdefault routeの決定方法を容易に共有できる．たとえば，DHT上のsuccessorをdefault routeにすることが考えられる．

次に，default routeを利用した場合の匿名通信路構築方法について述べる．第 R JoinGroupのノードを経由する匿名通信路を構築するには，中継ノードに第 R JoinGroupのノードを指定することが容易である．しかし，第 R JoinGroupのノードは復号できないノードであることから，暗号化に使用するノードID（復号可能な第 $R-1$ 以下のJoinGroupに属するノードID）と次中継ノードIDに違うIDを使用して，匿名通信路を構築するメッセージを作成する．つまり，メッセージはいったん次中継ノードに送られ，1つ以上のdefaulte routeを経由して復号可能な第 $R-1$ 以下のJoinGroupに属するノードに到達する．

以上により，第 R JoinGroupがメッセージを中継することが可能となる．さらに，このようなdefault routeを使用した匿名通信路をすべての匿名通信路に含ませることで，default routeの使用を例外でなくす．これにより，default routeしか利用できない第 R JoinGroupのノードによる自発通信とその他の通信を混在させることが可能となる．以上により，第 R JoinGroupのノードによる匿名通信が可能となる．

5. 宛先ノードID取得に対する提案方式

本章では2.3節で述べた宛先ノードIDの取得に対する提案方式について述べる．メッセージを送信するためには，ランダムに割り当てられた宛先ノードのノードIDが分からなければならない．そのためにスケーラビリティと匿名性を低下させずに，送信ノードが宛先ノードのIDを取得可能である必要がある．本章では，IBEを用いた宛先ノードのノードIDの取得方法を提案する．なお，宛先ノードは，宛先ノード指定用のキーとして自身のノードIDとは異なる**サービス名**（Service Name）を利用者に対して直接もしくはホームページ等を介して公開するものと仮定する．

5.1 Multi-connected Introduction Point 方式

我々の提案方式は，Torで提案されているIntroduction Point方式にフラッディングの特性を加える拡張を行い，IBEを応用した方式である．本取得方式では，2.3.2項で述べた，Introduction Point方式における利用状況が漏洩するという問題を解決する方式として**Multi-connected Introduction Point**（MIP）方式を提案する．MIP方式は，Introduction Point方式に，同時に複数の宛先ノードへの接続を維持する拡張を加え，かつ，接続要求メッセージを中継するときには接続しているすべての宛先ノードに中継する方式である．MIPとなるノード（以下，MIPノードという）の決定方法は，サービス名を全ノードが共有するハッシュ関数にかけて得たハッシュ値を，DHTのID空間で担当するノードとする．サービス名は公開情報であることから，宛先ノードのノードIDが変更された場合でも，サービス名に基づくMIPノードを送信ノードと宛先ノードの双方が共有できる．次に，1つのMIPノー

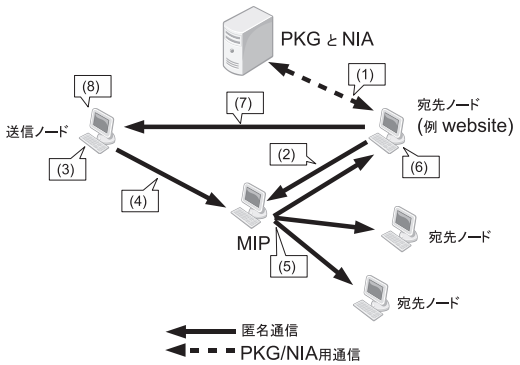


図 4 MIP を用いた宛先ノード ID の取得通信
Fig. 4 Retrieval of destination' NodeID usingg MIP.

ドを共有するすべての宛先ノードは同じ接続要求メッセージを受信することになるが，サービス名を IBE-ID とする公開鍵で接続要求メッセージを暗号化することで，受信すべき宛先ノードのみが復号可能となる．これらにより，Introduction Point ノードが特定の宛先ノードを示す関係を取り除き，利用状況の把握を困難とする．なお，IBE-ID にサービス名を用いるのは宛先ノードに接続要求メッセージを送るときのみであり，通常の匿名通信路構築にはノード ID を IBE-ID として使用する．このため，受信ノードは接続要求メッセージを受け取ったときだけ，サービス名に基づく復号を行えばよい．

サービス名は，1 台の宛先ノードが複数を公開することも可能である．この場合，宛先ノードが接続要求メッセージを受信したときには，自身が公開するすべてのサービス名に対応する秘密鍵で接続要求メッセージを復号して，復号できるか否かを確認する必要がある．

5.2 MIP 方式の処理手順

図 4 に，Onion 方式の多重暗号を行う Bifrost に MIP 方式を適用した場合の処理手順を示す．以下，処理の各手順について詳しく述べる．図中の番号は以下の手順番号を示す．なお，メッセージ内容の表記として，ID_N はノード N のノード ID を表す．P_N は ID_N に基づく IBE による公開鍵を表し，$P_N()$ は公開鍵 P_N による暗号化を表す．S_N は，ノード N と送信ノードとの共有鍵を表し，$S_N()$ は，共有鍵 S_N による暗号化を表す．M_{XtoY} は，ノード X からノード Y へのメッセージを示す．また，"|" は要素の結合を表す．

(1) 匿名通信路構築の準備としてサービス提供開始時に，宛先ノードはサービス名を PKG に登録して，サービス名を IBE-ID とした場合に対応する秘密鍵を取得する．

(2) 宛先ノードは，DHT の ID 空間で自身のサービス名のハッシュ値を担当する MIP ノードに対して匿名通信路を構築する．以下，この経路を第 2 通信路という．

(3) 送信ノード（I）が宛先ノード（D）に，宛先ノードのノード ID（ID_D）を要求する接続要求メッセージ（REQ）を作成する．この接続要求メッセージは，接続要求コマンド（COM）と，宛先ノードから送信ノードへの復路の通信路構築情報（H_{DtoI}）と送信ノードと宛先ノードの共有鍵（S_D）を含み，宛先ノードのサービス名の公開鍵（$P_{Service}$）によって暗号化される．

$$REQ = P_{Service}(COM|H_{DtoI}|S_D)$$

また，この接続要求メッセージは，送信ノードから MIP ノードへの L 台の中継ノード（E_1〜E_L：送信ノード側が E_1）と MIP ノードの公開鍵によって多重暗号化されて M_{ItoMIP} となる．

$$M_{ItoMIP} = P_{E_1}(ID_{E_2}|P_{E_2}(...P_{E_L}(ID_{MIP}|P_{MIP}(REQ))...))$$

このときに構築される匿名通信路を，以下第 1 通信路という．

(4) 第 1 通信路の各中継ノードは，メッセージ M_{ItoMIP} を復号し第 1 通信路を構築しながら接続要求メッセージを MIP ノードに送る．

(5) MIP ノードは，受信した接続要求メッセージを，自身を MIP ノードとして構築された全第 2 通信路に転送する．接続要求メッセージは，第 2 通信路の M 台の中継ノード（F_1〜F_M：MIP ノード側が F_1）で暗号化され，各宛先ノードに届いた時点で次のようになる．なお，ここで使用される共有鍵は第 2 通信路を構築した各宛先ノードと各中継ノードとの共有鍵である．

$$M_{MIPtoD} = S_{F_M}(S_{F_{M-1}}(...S_{F_2}(S_{F_1}(S_{MIP}(REQ)))...))$$

(6) 正しい宛先ノードは当該メッセージを受信し，IBE によるサービス名に対応する秘密鍵で復号する．これにより，宛先ノードの接続要求メッセージ（REQ）受信が完了する．

(7) 正しい宛先ノードは，送信ノードが作成した復路構築情報（H_{DtoI}）を用いて自身のノード ID（ID_D）を，N 台の中継ノード（G_1〜G_N：宛先ノード側が G_1）によって第 3 通信路を構築しながら，送信ノードに通知する．送信ノードが作成する復路構築情報を次に示す．なお，ここに含まれる共有鍵は，送信ノードが第 3 通信路の各中継ノードに配布する共有鍵である．

$$H_{DtoI} = ID_{G_1}|P_{G_1}(S_{G_1}|ID_{G_2}|P_{G_2}(...P_{G_N}(S_{G_N}|ID_I|dummy)...))$$

また，宛先ノードのノード ID は，各中継ノードが暗号化して送信ノードへ送る．その暗号化ための鍵は，復路構築情報を復号して得られる送信ノードとの共有

鍵を使用する．この暗号化された宛先ノードのノードIDをB_{DtoI}とすると，宛先ノードから送信ノードへと送られるメッセージM_{DtoI}の構成は次のようになる．

$$M_{DtoI} = H_{DtoI}|B_{DtoI}$$

送信ノードが受信するB_{DtoI}は次のように多重暗号化されており，送信ノードはこのB_{DtoI}を復号し，宛先ノードのノードID（ID_D）を得る．

$$B_{DtoI} = S_{G_N}(S_{G_{N-1}}(...S_{G_2}(S_{G_1}(S_D(ID_D)))...))$$

(8) 最後に，送信ノードは，宛先ノードとの間に匿名通信路を構築する．なお，このときに第1通信路と第3通信路は破棄する．

送信ノードとMIPノードと宛先ノードの3点を結ぶ通信路は，すべて匿名通信路であるので，MIPノードと宛先ノードは送信ノードを特定することはできない．また，MIPノードは複数の第2通進路を保持しており，接続要求メッセージがどの宛先ノード宛かを知ることはできない．以上から，匿名性を低下させることなく，送信ノードは宛先ノードのノードIDを取得することができる．

なお，MIPノードが複数の第2通信路を保つためには，MIPノードの数が宛先ノードの数よりも少なくなければならない．このため，番号の小さなJoinGroupに属するノードだけがMIPノードになることができるとする．このMIPノードになることができるJoinGroup番号の上限は，宛先ノードの総数によって定めるものとする．

6. 性能評価

提案方式を実装し評価を行った．評価事項はディレクトリサーバなしによる匿名通信の動作確認と，IBE導入の課題であるMIPノードを介した宛先ノードID取得に要する時間の2つである．実装は，OverlayWeaver [16] 上に開発したBifrostにIBEライブラリ [17] を適用して行った．Bifrostは受信エリアというdefault route機能と同等の機能があることから，提案方式のdefault route機能には受信エリアを使用した．また，NIAとPKGとMIP機能は追加実装した．

初めの評価として，ディレクトリサーバによる検索を行うことなしに任意の中継ノードを経由した匿名通信が可能であることを確認した．これは，本方式がディレクトリサーバの機能を実装していないことからも明らかである．

6.1 評価方法および評価環境

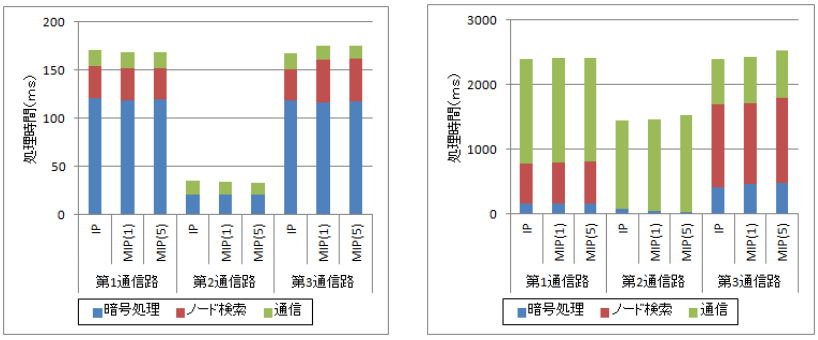
MIP方式の宛先ノードID取得時間評価として，既存方式であるIntroduction Point（IP）方式と比較する．また，複数ある第2通信路にMIPノードが中継する順番がノードID取得時間に与える負荷について評価する．この評価は，5台の受信ノードが第2通信路をMIPノードとの間に構築し，正しい受信ノードに向けてMIPノードが中継する順番が1番目の場合と5番目の場合についてノードID取得時間を計測することで行う．

評価環境は，LAN環境とインターネット環境（Planet Lab [18]）の2つを用いた．LAN環境は，CPU Core2Duo 3 GHz, OS CentOS 5.4, ネットワーク1000 Base-TのLAN接続のPCを32台を使用した．また，インターネット環境でも同様に32台のPCを使用した．評価ネットワークの構成は，第1から第3通信路におけるBifrostの受信エリア数を各3つとして合計9つである．また，中継ノード数を各6台として合計で18台である．

6.2 MIPを用いたノードID取得時間

ノードID取得時間における3つの通信路ごとの処理時間について，**図 5**-a) にLAN環境での結果を，図5-b) にインターネット環境における結果を示す．それぞれ10回ずつ測定した平均である．提案方式とその基となったIP方式における宛先ノードID取得には，「送信ノード（I）→MIPノード/IPノード」（第1通信路）と「MIPノード/IPノード→宛先ノード（D）」（第2通信路）と「宛先ノード（D）→送信ノード（I）」（第3通信路）の3つの匿名通信路を経由する．各グラフは，各通信路ごとにIP方式（図中のIP）の結果，MIP方式においてMIPノードが最初に中継する第2通信路に正しい受信ノードがいる場合（図中のMIP(1)）の結果と，最後に中継する第2通信路に正しい受信ノードがいる場合（図中のMIP(5)）の結果を示す．さらに，各グラフは，暗号処理時間，通信路構築のためのノード検索時間と通信時間を示す．

まず，受信ノードのノードIDの取得時間全体は，LAN環境で373 ms（IP）から377 ms（MIP(5)）であり，インターネット環境で6,222 ms（IP）から6,470 ms（MIP(5)）である．IP方式と2つのMIP方式の取得時間全体の差はLAN環境で4 ms（約1%）であり，インターネット環境で248 ms（約4%）である．インターネット環境では他の通信や処理の影響を受けることを考慮すると，取得時間全体の差は小さいといえる．以降，評価以外の通信と処理の影響が少ないLAN環境について，通信路ごとの処理時間の比較について述べる．第1通信路は両方式に差がないことから，すべての処理についてほぼ同じ処理時間である．第1通信路は第3通信路とともに，公開鍵暗号を用いて匿名通信路を構築する．このため，第2通信路に比べて暗号処理時間やノード検索時間が長い．第2通信路は，MIPノードによる処理負荷増加と複数の匿名通信路による通信回数が増加しているが，すべての処理と方式についてほぼ同じ処理時間である．低負荷である理由は，MIPノードが送信する第2通信路が復路であるためである．復路は，各中継ノードが共有鍵による暗号化を行い宛先ノードが多重暗号化されたメッセージを復号する構成となることから，MIP



a) LAN 環境
a) LAN

b) インターネット環境
b) Internet

図 5 MIP を用いたノード ID の取得時間
Fig. 5 NodeID retrieval time using MIP.

ノードは第 2 通信路ごとに 1 回ずつ共有鍵による暗号化を行えばよい．このため，MIP ノードの暗号処理負荷が少なく，第 2 通信路数の増加が大きな負荷とはならない．第 3 通信路では，MIP(1) と MIP(5) はほぼ同じ処理時間であるが，ノード検索時間が IP 方式と比較して MIP 方式では中継順に関係なく増加している．これは取得時間の増加の原因となっている．この原因は，複数ある第 2 通信路の通信と第 3 通信路のノード検索が同時に行われることにより，各ノードにかかる負荷が IP 方式よりも高いためと考えられる．本評価では 5 本の第 2 通信路を構築するために，全 32 ノードの内の 30 ノードを使用しており，ほぼすべてのノードに負荷がかかっている．しかし，参加ノード数がより多い一般的な環境では，第 2 通信路による負荷を受けるノードの割合は本評価よりも低くなると想定できることから，第 3 通信路に対する影響は少なくなり，MIP 方式による影響は本評価結果よりも軽微になると考えられる．これらの結果は，MIP 方式による処理負荷の増加量は少ないことを示しており，MIP 方式の有効性を示す結果であるといえる．

7. 考察

本提案手法に対する，ディレクトリサーバの削除によるスケーラビリティと，ノード ID の割当て手法の匿名性への影響と，ノード離脱対策について考察を述べる．

7.1 スケーラビリティの考察

ディレクトリサーバを使用した場合と本提案手法について，スケーラビリティに関する負荷を考察する．なお，ノード総数を N とする．

ディレクトリサーバでノード管理を行う場合，ディレクトリサーバのノード管理情報の通信コストは，ノード総数とノード管理情報の量の積に比例する．つまり，ディレクトリサーバを用いるシステムの通信コストは $O(N^2)$ である．加えて，ディレクトリサーバは，全 N ノードについて，定期的にノード情報を収集して更新しなければならない．一方，提案方式はディレクトリサーバが不要である．各ノードは，システムへの参加時に NIA と PKG に 1 回ずつアクセスするだけでよい．なお，PKG によるマスタ秘密鍵の更新による定期的な通信は必要であるが，各ノードからの情報収集は必要なく，その更新間隔は長くできる．また，NIA が各ノードに配布するデータはハッシュ関数だけである．このサイズは小さく，参加ノード数に比例しない．このため，提案手法の通信コストは $O(N)$ である．この結果より，提案手法は，ディレクトリサーバを用いる既存手法よりもスケーラビリティがあるといえる．

7.2 匿名性とノード ID 割当ての規則化

提案手法では，ノード ID の割当てをグループ単位に規則化しており，この規則を利用した攻撃が考えられる．新規参加ノードは既存ノードの中間に位置する（図 2 参照）．このため，同一時期に多数のノードを参加させることで，1 つの JoinGroup の全ノードを攻撃ノードで占めることができる可能性がある（この場合の攻撃ノードの占有率は 50%となる）．これにより，全既存ノードの predecessor と successor を得ることが可能となる．これは，通常の割当て方式では困難な攻撃方法である．

すべてのノードの predecessor と successor を攻撃ノードが占めることによる問題について考察する．まず，中継ノードを参加ノードから任意に選択する場合には，DHT の ID に基づくノード間の位置関係が通信路に影響を与えることはない．しかし，本提案手法における default route は DHT の ID に基づく中継ノード選択を行うことから，default route の決定方法によっては影響を受ける．たとえば，default route を successor にした場合，default route に基づく通信区間では 1 台おきに攻撃ノードを通過することになる．特に，default route 区間では暗号化処理が行われないことから，メッセージ文面が変化することなく中継されるため通信解析攻撃を容易にすると考えられる．しか

し，この解析攻撃により攻撃者に漏えいする情報は default route 区間の始点と終点だけであり，前後の default route 区間を含む匿名通信路の全体が漏えいすることはない．なお，匿名性への影響の定量的評価は，匿名性の評価方法を含めて今後の課題である．

7.3 ノード離脱に対する対策

ノード離脱により，割当て済み ID のノードが不在となると，当該 ID を利用した多重暗号を復号できなくなる．このため，ノード離脱が発生した場合には，速やかな対応が必要である．しかし，現時点において，ノード離脱対策処理は今後の課題である．既存のノード離脱対策方式としては，Cashmere や Bluemoon による，近隣 ID のノードによる鍵等のバックアップ方式がある．これは DHT のストレージ機能によるバックアップを利用して，事前に鍵情報等を他のノードに保存する．もし，ノードが離脱した場合にはバックアップを有するノードが保存された鍵やノード情報を利用して，当該ノードの代理を務める方式である．この方式はバックアップを有するノードの検索と離脱検知にも DHT の機能を利用することで匿名性を保持しており有効であると考えられる．しかし，事前に鍵を他のノードに保存することによる情報漏洩や攻撃への対策が必要という課題がある．

8. まとめ

本稿では，多重暗号による匿名通信方式に対して，スケーラビリティを確保する方法として DHT に基づくノード管理と IBE を適用する方式を提案した．さらに，IBE を適用するための 2 つの課題を明確にし，その解決策を示した．この解決策により，ディレクトリサーバが不要な匿名通信方式を実現し，ディレクトリサーバによってスケーラビリティが低下するという問題を解決した．

さらに，ノード ID がランダムに割り当てられる場合の接続方式について，既存の Introduction Point 方式を改良した MIP 方式を提案した．MIP 方式により，Introduction Point 方式による宛先サービスの利用状況が漏洩するという問題を解決した．また，MIP ノードの負荷を評価し，LAN 環境における負荷の増加は約 1%であることを示し，低負荷であることを示した．

今後の課題は，匿名通信路の構築速度の高速化である．また，ノード離脱への対策も今後の課題である．

謝辞 本研究の一部は文部科学省科学技術研究補助金基盤研究 C（課題番号：20500064，23500085）によるものである．

参考文献

[1] Pfitzmann, A. and Waidner, M.: Networks without user observability, *Computers & Security*, Vol.6, No.2, pp.158–166 (1987).
[2] Goldschang, D., Reed, M. and Syverson, P.: Onion routing for anonymous and private internet connections, *ACM SIGCOMM Computer Communication Review*, Vol.42, No.2, pp.39–41 (1999).
[3] Syverson, P.F., Goldschlag, D.M. and Reed, M.G.: Anonymous connections and Onion routing, *IEEE Journal on Specific Areas in Communications*, Vol.16, No.4, pp.482–494 (1998).
[4] Dingledine, R., Mathewson, N. and Syverson, P.: Tor: The Second-Generation Onion Router, *Proc. 13th USENIX Security Symposium*, pp.303–320 (2004).
[5] Zhuang, L., Zhou, F., Zhao, B.Y. and Rowstron, A.: Cashmere: Resilient anonymous routing, *Proc. 2nd Conference on Symposium on Networked Systems Design and Implementation*, pp.301–314 (2005).
[6] Kondo, M., Saito, S., Ishiguro, K., Tanaka, H. and Matsuo, H.: Bifrost: A Novel Anonymous Communication System with DHT, *2nd International Workshop on Reliability, Availability and Security*, pp.324–329 (2009).
[7] McLachlan, J., Tran, A., Hopper, N. and Kim, Y.: Scalable Onion Routing with Torsk, *Proc. 16th ACM Conference on Computer and Communications Security*, pp.590–599 (2009).
[8] Puttaswamy, K.P.N., Sala, A., Wilson, C. and Zhao, B.Y.: Protecting anonymity in dynamic peer-to-peer networks, *IEEE International Conference on Network Protocols*, pp.104–113 (2008).
[9] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, *SIAM J. Comput.*, Vol.32, No.3, pp.586–615 (2003).
[10] Kate, A., Zaverucha, G.M. and Goldberg, I.: Pairing-Based Onion Routing, *Proc. 7th Privacy Enhancing Technologies*, pp.95–112 (2007).
[11] Nambiar, A. and Wright, M.: Salsa: A structured approach to large-scale anonymity, *Proc. 13th ACM Conference on Computer and Communications Security*, pp.17–26 (2006).
[12] Zhu, Y. and Hu, Y.: TAP: A novel tunneling approach for anonymity in structured P2P systems, *International Conference on Parallel Processing*, pp.21–28 (2004).
[13] Mittal, P. and Borisov, N.: ShadowWalker: peer-to-peer anonymous communication using redundant structured topologies, *Proc. 16th ACM Conference on Computer and Communications Security*, pp.161–172 (2009).
[14] Trabelsi, S. and Roudier, Y.: Secure Service Discovery with Distributed Registries, *2nd IEEE Workshop on Service Discovery and Composition in Ubiquitous and Pervasive Environments*, pp.1–6 (2008).
[15] Douceur, J.R.: The Sybil Attack, *Proc. 1st International Workshop on Peer-to-Peer Systems*, pp.251–260 (2002).
[16] 首藤一幸，田中良夫，関口智嗣：オーバ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.SIG12(ACS15), pp.358–367 (2006).
[17] Yiyang, B.: ibe-javapairing, available from ⟨http://en.sourceforge.jp/projects/sfnet_ibe-javapairing/⟩ (accessed 2011-07-29).
[18] PlanetLab, available from ⟨http://www.planet-lab.org/⟩ (accessed 2011-07-29).

田中 寛之

2009 年名古屋工業大学工学部情報工学科卒業．2011 年同大学大学院情報工学専攻博士前期課程修了．同年日本電気株式会社入社，現在に至る．

齋藤 彰一 （正会員）

1993 年立命館大学理工学部情報工学科卒業．1995 年同大学大学院博士前期課程修了．1998 年同大学院博士後期課程単位習得中退．同年和歌山大学システム工学部情報通信システム学科助手．2003 年同講師，2005 年同助教授．2006 年名古屋工業大学大学院助教授，2007 年同准教授．現在に至る．オペレーティングシステム，インターネット，セキュリティ等の研究に従事．博士（工学），ACM，IEEE-CS 各会員．

松尾 啓志 （正会員）

1983 年名古屋工業大学工学部情報工学科卒業．1985 年同大学大学院修士課程修了．1989 年同大学院博士課程修了．同年名古屋工業大学電気情報工学科助手．講師，助教授を経て，2003 年同大学大学院教授，2006 年同大学情報基盤センターセンター長（併任），2011 年同大学附属図書館長（併任），現在に至る．分散システム，分散協調処理に関する研究に従事．工学博士．電子情報処理学会，人工知能学会，IEEE 各会員．