

# 受信者の匿名性を送信者に対しても確保可能な DHT を利用した匿名通信路構築手法の提案

## A Proposal of DHT-based Anonymous Routing for Preserving Pseudonymity of Message Receivers

及川 一樹\*      王家宏\*      児玉 英一郎\*      高田 豊雄\*  
Kazuki Oikawa      Jiahong Wang      Eiichiro Kodama      Toyoo Takata

あらまし P2P ネットワーク技術において、分散ハッシュテーブルを利用することにより、スケーラビリティに優れたキーベースのルーティングを行うことが可能になる。しかし、分散ハッシュテーブルには匿名性が無いため、宛先となるキーが知られてしまえば、それに対応する本人到達性の高い IP アドレスを取得されてしまう、即ち、匿名性を維持したまま、自分宛のメッセージを受信することができない。そこで、我々は分散ハッシュテーブルを用いたルーティングにオニオンルーティングを組み合わせることで、宛先となるキーが知られた場合でも、IP アドレスを秘匿し、匿名性 (Pseudonymity) を確保する、双方向匿名通信路の構築手法を提案する。

キーワード 分散ハッシュテーブル, オニオンルーティング, 双方向匿名通信

### 1 はじめに

現在、P2P ネットワーク技術は一般的なものとなり、BitTorrent といったファイルの転送を行うアプリケーションや、映像配信を行うアプリケーションなどに広く利用されている。

P2P ネットワーク技術には中央に管理用サーバを持つハイブリッド P2P 型や、すべてのノードが対等な関係にあるピア P2P 型などの種類がある。ピア P2P 型は管理用サーバが不要で単一故障点が存在しないという点などで優れており、問題点だったデータの探索効率が悪いという点も、分散ハッシュテーブルに代表される構造型ネットワークの登場により解決され、今後も様々な用途に活用されると考えられている。

分散ハッシュテーブルとは、ハッシュテーブルを複数のノードで管理する技術で、各ノードをハッシュ値空間に写像し、その空間を各ノードで分割管理することで、ピア P2P 型でありながら、スケーラビリティに優れた探索を可能にする。そのため、分散ハッシュテーブルでは、各ノードにハッシュ値が割り当てられ、その値をキーにした探索が行われる。

この分散ハッシュテーブルを利用することにより、キー

を宛先としたメッセージの送受信ができるようになるため、仮名 (Pseudonym) をキーとすることで、インスタントメッセージなどといったアプリケーションがサーバレスで実現できるようになる。しかし、分散ハッシュテーブルには匿名性が無いため、宛先となるキーを知ることができれば、そのキーと IP アドレスの組み合わせが漏洩してしまい、匿名性 (Pseudonymity) が確保できない。この問題は、インスタントメッセージのようなアプリケーションを想定した場合、宛先となるキー (仮名) を信頼できる知人以外には漏らしてはならないことを意味し、仮名を利用する意味が無くなってしまう。

そこで我々は、分散ハッシュテーブルにオニオンルーティングを組み合わせることで、匿名性を維持したまま自分宛のメッセージを受信することが可能な、双方向匿名通信路の構築手法を提案する。本提案手法では、送受信者の匿名性を自分以外の全ノードに対して確保するほか、管理用サーバを必要としないためスケーラビリティに優れる手法となっている。

本論文の構成は次のとおりである。まず、第 2 節で既存研究を紹介し問題点を述べた後、第 3 節で本提案手法を説明し、第 4 節で匿名性や効率について考察し、第 5 節でまとめる。

\* 岩手県立大学大学院, 〒 020-0193 岩手県岩手郡滝沢村滝沢字菓子 152-52, Iwate Prefectural University, 152-52, Sugo, Takizawa, Takizawa village, Iwate 020-0193

## 2 既存研究

分散ハッシュテーブルを利用した双方向匿名通信路に関する提案は幾つか提案されている [1][2].

これらの提案では、送信者は受信者がいずれかのグループに所属するように、いくつかのグループを作成、グループ間のルーティングは分散ハッシュテーブルを利用して効率よく行い、グループ間でやりとりされるメッセージを多重暗号化することで、前後のグループ以外の情報を漏らさないようにし、送信者の匿名性を確保している。そしてペイロードは受信者の公開鍵で暗号化し、各グループ内でペイロードをブロードキャストすることで、受信者を他のグループメンバに漏らすことなく、ペイロードの配送が可能になる。

しかし、これらの提案では以下に示す2つ問題点が挙げられる。

**受信者の匿名性が送信者に対して確保されない** これらの

提案では、内部告発など送信者の匿名性のみが重要視される用途向けに設計されているため、受信者の匿名性は考慮されていない。そのため、宛先として分散ハッシュテーブルでノードに割り当てられるキーをそのまま利用するので、宛先のキーよりそのノードのIPアドレスを特定することが可能になってしまう。

**ディレクトリサーバを必要とする** 中継ノードの公開鍵を取得するために、全ノードのキーと公開鍵のペアを保存するサーバが必要なほか、中継ノードの選出にもディレクトリサーバを利用するため、分散ハッシュテーブルのスケラビリティという特徴が生かせないほか、単一故障点となり得る。

## 3 提案手法

本節では以上の問題点を解決する手法を述べる。

まず、第3.1節で本提案手法が利用する、分散ハッシュテーブルとオニオンルーティングに関して簡単に説明し、第3.2節で前提となる条件を述べ、第3.3節で本提案手法が確保する匿名性に関して述べる。そして、第3.4節で本提案手法の概要を示した後、第3.5節～第3.8節で詳細を説明する。

### 3.1 分散ハッシュテーブルとオニオンルーティング

本提案手法では、分散ハッシュテーブルとオニオンルーティングを利用する。

分散ハッシュテーブル (DHT: Distributed Hash Table) とは、Chord[3] や Tapestry[4], Kademlia[5] といったアルゴリズムに代表される技術で、一般的には、ハッシュ値が近いノードに関する情報は密に、ハッシュ値が遠い

ノードに関する情報は疎に持つことで、 $O(\log N)$  程度のホップ数で探索が可能で、スケラビリティに優れているという特徴を持つ。

オニオンルーティング [6] は、通信路全体を盗聴することができないという仮定のもと、多重暗号化されたメッセージを中継ノード経由で送信することで、送信元を隠蔽する技術である。ここで、宛先を  $P_k$ 、中継ノードを  $P_i (1 \leq i \leq k)$  とし、各  $P_i (1 \leq i \leq k)$  の公開鍵を  $PK_i$ 、公開鍵  $PK_i$  で暗号化する記号を  $\langle \dots \rangle_{PK_i}$ 、送信元と中継ノード  $P_i$  とのメッセージ送受信用共通鍵を  $K_i$  とし、各メッセージ  $M_i$  は以下のように構築される。

$$M_i = \begin{cases} \langle P_{i+1}, K_i, M_{i+1} \rangle_{PK_i} & (i < k) \\ \langle K_k, \text{Payload}, \text{Padding} \rangle_{PK_k} & (i = k) \end{cases}$$

このような記号のもと、送信者は  $M_1 \sim M_k$  作成後、 $M_1$  を  $P_1$  に送信する。 $M_i$  を受け取った中継ノード  $P_i$  は、秘密鍵でメッセージを復号し、 $P_{i+1}$  と  $K_i$ 、 $M_{i+1}$  を取り出し、 $P_{i+1}$  に  $M_{i+1}$  を送信する。もし、復号したメッセージに  $P_{i+1}$  が含まれていない場合は自身が宛先なので、ペイロードを処理する。メッセージ返信時には各中継ノードに配布した共通鍵  $K_i$  を利用して、多重暗号化を施す。

このように、各中継ノードは前後のノードの情報しか得られないため、送信者を特定することが出来ない。また経路作成後にやりとりするメッセージも、各中継ノードにおいて共通鍵  $K_i$  によって暗号化されるため、中継ノード間でやりとりされるデータは全て異なったものとなり、連続していない中継ノード間の繋がりを特定することが出来ない。そのため、全中継ノードが結託しない限りは、送信者の匿名性は確保される。

### 3.2 前提条件

本提案手法では、以下を前提条件とする。

- 通信路全体を盗聴することは不可能 オニオンルーティングを用いるため、この仮定を継承する
- 善良なノードのルーティングテーブルに含まれる悪意を持つノードの割合は一律に分布する 本提案手法では、分散ハッシュテーブルのルーティングテーブルに対しての攻撃を考慮しないため、この仮定を前提とする
- 常にある一定数の匿名通信路において通信が行われている ネットワーク全体において、ただ1つの匿名通信路のみで通信が行われている場合、オニオンルーティングの中継ノードすべてが結託ノードでなくても、送受信者の匿名性が確保できなくなるため、この仮定を前提とする

### 3.3 確保する匿名性

本提案手法では、宛先として利用されるキーとそのキー宛てに送信されたメッセージを受信するノードのIPアドレスの組み合わせを有意な確率で特定することが困難な状態であることを、匿名性があると定義する。つまり、キーを仮名 (Pseudonym) としたときに、本人到達性の高いIPアドレスを特定困難にすることで、匿名性 (Pseudonymity) を確保する。

また、前提条件の下、上記の意味の匿名性を以下に対して確保する。

**送信者の匿名性** 送信者のIPアドレスを自分以外のノードから特定することが困難

**受信者の匿名性** 受信者のIPアドレスを自分以外のノードから特定することが困難

**送受信者の繋がり**の匿名性 通信を行っている送受信者のキーの組み合わせを送受信者以外のノードから特定することが困難

### 3.4 提案手法の概要

本提案では第2節で挙げた2つの問題点に対して、以下のような手法で解決を図る。

まず一つ目の問題点である、受信者の匿名性が送信者に対して確保されないという問題は、宛先として利用するキーが、匿名性のない分散ハッシュテーブルにおいてノードに割り当てられるキーであるという点に起因している。そこで我々は宛先として利用されるキーと、分散ハッシュテーブルにおいてノードに割り当てられるキーを、それぞれ別々なものとして扱い、2つの間に関係性を持たせないことで、分散ハッシュテーブルを利用する際の匿名性破綻を防ぐ。そして、2者間の通信においては、オニオンルーティングを経由することで、匿名性の確保をはかる。

二つ目の問題点である、公開鍵の管理や中継ノード選出を行うディレクトリサーバが必要という問題は、以下の手法により解決する。まず、公開鍵を管理するサーバが必要な理由は、キーと公開鍵が別々なことに起因するため、公開鍵そのものをキーとして利用する手法を提案する。また、中継ノード選出の際には、自分や他ノードの分散ハッシュテーブルのルーティングテーブルよりランダムにノードを選択し選出する手法を提案する。

本提案手法の概要を図1に示す。宛先として利用されるキーは分散ハッシュテーブルのキーとは異なるため、何らかの方法でキーと自分自身への経路の組み合わせを匿名性を維持しつつ、全体へ通知しなければならない。そこで、自分自身への経路をオニオンルーティングを利用して構築し、分散ハッシュテーブル上に、宛先となる

キーとオニオンルーティングの終点ノードの組み合わせを登録する(以降、この操作をパブリッシュと呼び、この操作を行ったノードをパブリッシャーと呼ぶ)。そして、メッセージを送信するノード(イニシエータ)は、オニオンルーティングを経由して分散ハッシュテーブルにアクセスし、宛先となるキーで探索を行い、そのキーに対応するパブリッシャー側オニオンルーティングの終点ノードの情報を取得し、イニシエータ側オニオンルーティングの終点ノードとパブリッシャー側オニオンルーティングの終点ノードを接続し、イニシエータとパブリッシャー間で通信路を構築する。

以上により、本提案手法では、送受信者の匿名性を自身以外のノードに対して確保可能な、匿名通信路を構築する。

次節より、本提案手法の詳細について説明する。

### 3.5 記号の定義

$N$ : 全ノード数

$KY$ : パブリッシュまたは宛先となるキー

$N_{key}$ : キー  $key$  を担当するノード

$L$ : オニオンルーティングの中継ノード数

$R_i (0 \leq i \leq L)$ : オニオンルーティングの  $i$  番目の中継ノード. 特に  $R_0$  は始点ノード

$K_i (1 \leq i \leq L)$ : オニオンルーティングの  $i$  番目の中継ノードと共有する共通鍵

$\langle \dots \rangle_{PK_{N_i}}$ : ノード  $N_i$  の公開鍵で暗号化

$\langle \dots \rangle_{K_i}$ : 共通鍵  $K_i$  で暗号化

$V_{N_i}$ : ノード  $N_i$  が生成した乱数

$M$ : 送信するメッセージ

$PL_i$ : 中継ノード  $R_i$  に送信するメッセージのペイロード

### 3.6 公開鍵の管理

関連研究では公開鍵の管理に管理用サーバを用いていたが、本提案では、公開鍵そのものを分散ハッシュテーブルや宛先として利用するキーとして利用する。

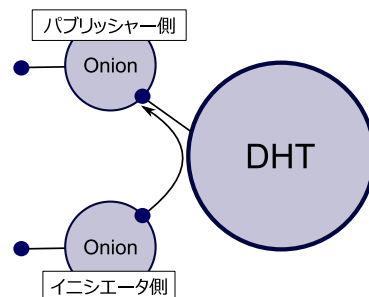


図 1: 本提案手法の概要

本提案では公開鍵暗号アルゴリズムとして、楕円曲線暗号を想定しているため、利用する公開鍵の鍵長は、192～256bit程度で十分な強度を得られる。この長さは、一般的な分散ハッシュテーブルのアルゴリズムが想定する長さである128～160bitよりは長いが、第3.1節で挙げた代表的なアルゴリズムには問題なく適用できる。

分散ハッシュテーブル用の公開鍵はネットワークに参加する度に変更可能で、各ノードは乱数 $d$ を発生させそれを秘密鍵とし、グローバルパラメータである楕円曲線ドメインパラメータの基点 $G$ を利用して、公開鍵を $dG$ とする。

宛先用の公開鍵は、仮名を表すものなので、同じ仮名を使う限りは変更できない。この計算方法も上記と同じで、乱数 $d'$ を秘密鍵とし、 $d'G$ を公開鍵とする。

### 3.7 経路の構築

経路の構築にはパブリッシャー側とイニシエータ側の2種類があり、パブリッシャー側が先に構築した経路にイニシエータ側が接続するという順番になる。また、2つとも経路の作成時には中継ノードの選出が必要であるため、先に第3.7.1節にて中継ノードの説明をした後、第3.7.2節でパブリッシャー側の経路構築に関して説明し、第3.7.3節でイニシエータ側の経路構築について説明する。

#### 3.7.1 中継ノードの選出

本提案手法ではディレクトリサーバを利用しないで適当なノードを中継ノードとして選出するために、ランダムなノードのルーティングテーブルからランダムに1エントリ取得し、中継ノード選出用に自ノードにプールしておき、選出時にはそこから中継ノードを選択する。エントリの取得方法としては以下のような方法が挙げられる。

- 分散ハッシュテーブルにおいて、メッセージをルーティングした際や、ルーティングテーブルメンテナンスの際に、他ノードのルーティングテーブルからランダムなエントリを1つ取得する
- 定期的にキーをランダムに生成し、そのキーに近いノードのルーティングテーブルからランダムなエントリを1つ取得する

#### 3.7.2 パブリッシャー側経路の作成

1. パブリッシャー $R_0$ は中継ノード $R_i(1 \leq i \leq L)$ を選出
2. 各中継ノードに配布する共通鍵 $K_i(1 \leq i \leq L)$ を作成

3. パブリッシャーは $M = KY$ として、ペイロード $PL_1$ を以下の式により構築

$$PL_i = \begin{cases} \langle R_{i+1}, K_i, PL_{i+1} \rangle_{PK_{R_i}} & (1 \leq i < L) \\ \langle NULL, K_i, M \rangle_{PK_{R_i}} & (i = L) \end{cases}$$

4. パブリッシャーは乱数 $V_{R_0}$ を生成し、 $(V_{R_0}, PL_1)$ を、 $R_1$ に送信。経路情報として $(V_{R_0}, R_1, K_1, K_2, \dots, K_L)$ を記憶
5.  $(V_{R_{i-1}}, PL_i)$ を受信した $R_i$ は、秘密鍵で $PL_i$ を復号し、 $(R_{i+1}, K_i, PL_{i+1})$ を取得
6.  $R_i$ は乱数 $V_{R_i}$ を生成し、 $(V_{R_i}, PL_{i+1})$ を、 $R_{i+1}$ に送信。経路情報として $(V_{R_{i-1}}, R_{i-1}, V_{R_i}, R_{i+1}, K_i)$ を記憶
7. 操作5,6を $i = L - 1$ となるまで続ける
8.  $i = L$ となるノードがペイロードを受信し操作5と同様に復号を行うと、次の送信先であるノードの情報がNULLになっており、ペイロードにはキーが平文で含まれている。ノード $R_L$ は、乱数 $V_{R_L}$ を作成した後、分散ハッシュテーブルを利用してそのキーを探索し、探索されたノード $N_{KY}$ に対して、 $(V_{R_L}, KY)$ を送信。ノード $R_L$ は経路情報として $(V_{R_{L-1}}, R_{L-1}, V_{R_L}, K_L)$ を記憶し、 $N_{KY}$ は $(KY, V_{R_L}, R_L)$ を記憶。

以上の操作により、パブリッシャー側の経路が完成する。以降の通信に関して、この経路では、パブリッシャーが送信するメッセージに対しては、各中継ノードが持つ共通鍵で暗号を復号しながら転送し、逆向きのメッセージに対しては各中継ノードが持つ共通鍵で暗号化を施しながら転送する。

#### 3.7.3 イニシエータ側経路の作成と、パブリッシャー側経路との接続

1. パブリッシャー側と同様にイニシエータは、 $M = (KY, Payload)$ として、ペイロードを構築
2. パブリッシャー側と同じくオニオンルーティングの中継ノード終点までメッセージを転送
3. オニオンルーティングの終点ノードは分散ハッシュテーブルを利用して、ペイロードに含まれるキーを探索し、探索されたノードより、キーに対応する経路情報 $(V_{R_L}, R_L)$ を取得
4.  $(V_{R_L}, M)$ を $R_L$ に送信し、経路情報として $(V_{R'_{L-1}}, R'_{L-1}, V_{R_L}, R_L, K'_L)$ を記憶

以上の操作により、イニシエータ側経路の作成と、パブリッシャー側経路との接続が完了する。

以降の通信に関しはこの経路でも、イニシエータが送信するメッセージに対しては、各中継ノードが持つ共通

鍵で暗号を復号しながら転送し、逆向きのメッセージに対しては各中継ノードが持つ共通鍵で暗号化を施しながら転送する。

### 3.8 双方向通信を行う仕組み

図2のように、メッセージは多重暗号化され中継される。第3.7.2節や第3.7.3節の最後で述べた様に、各中継ノードはメッセージの向きによって共通鍵を利用して暗号化を施すか、復号を行うかを決めるため、送受信されるメッセージは向きにかかわらず図に示す様に位置によって常に同じ多重暗号化を施された状態になる。

また、メッセージの中継も乱数  $V_{R_i}$  と、送信元情報によって経路を識別し、次のノードへ正しく中継できるようにしている。

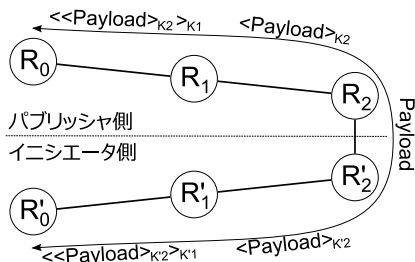


図 2: 多重暗号化

## 4 考察

本節では、本提案手法における匿名性や秘匿性、メッセージサイズについて考察する。

まず、第4.1節にて本提案手法の匿名性に関して考察し、匿名性が破綻する条件である中継ノード全てが結託ノードで構成される確率を求め、確保したい匿名性の強度と中継ノード数との関係について述べる。

そして、第4.2節にて通信内容の秘匿性に関して、第4.3節にてメッセージサイズについて考察する。

### 4.1 匿名性

#### 4.1.1 送信者の匿名性

送信者は、オニオンルーティングを経由してアクセスするため、中継ノード全てが結託ノードで構成されない限りは匿名性は確保される。

#### 4.1.2 受信者の匿名性

受信者は、パブリッシュ時にオニオンルーティングの終端ノードに、自分のキーを平文で渡すが、そのキーに対応するIPアドレスはオニオンルーティングによって、秘匿されるため、オニオンルーティングの中継ノード全てが結託ノードで構成されない限りは匿名性は確保される。

#### 4.1.3 送受信者の繋がり匿名性

送信者はオニオンルーティングの終端ノードに、宛先となるキーを渡し、分散ハッシュテーブルで探索を行うが、渡すのは宛先となるキーだけで、送信者のキーは知らせないので、送信者と受信者両方のキーの繋がりが漏れることはない。

#### 4.1.4 オニオンルーティングの中継ノードとして結託ノードを選択する確率

オニオンルーティングにおいては、前提とする仮定より、中継ノード全てが結託しない限りは匿名性が確保できる。また、本提案では各ノードのルーティングテーブルに含まれる悪意を持つノードの数は一様に分布するという仮定をおいているため、ランダムにノードを選択し、そのノードのルーティングテーブルからエントリをランダムに取得する場合、悪意のあるノードを含む確率  $p$  は結託ノード数を  $m$  と置くと以下ようになる。

$$p = \left[ \left( \frac{N-m}{N} \right) \left( \frac{m}{N} \right) + \left( \frac{m}{N} \right) \cdot 1 \right]$$

そのため、中継ノード全てに結託ノードを選択してしまう確率は、 $p^L$  となり、グラフに示すと図3のようになる。この図は横軸が結託ノードの割合、縦軸が  $p^L$  の値となっている。この図より、結託ノードが全体の2割程度なら  $L \geq 4.51$  において1%未満の値をとるが、結託ノードが4割の場合は  $L \geq 6.84$ 、5割の場合だと  $L \geq 16.0$  という様に、同じ安全性を確保するために  $L$  の値が指数関数的に増加するため、想定されるネットワークサイズや中継コストなどを考慮して、 $L$  を定める必要がある。

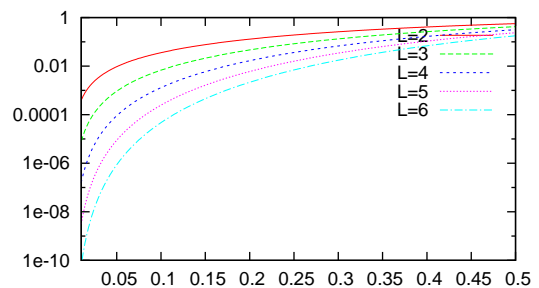


図 3: 全中継ノードが結託ノードで構成される確率

以上より、本提案手法では適切なパラメータを設定することにより、キーとIPアドレスの組み合わせが漏洩する可能性が低くなり、匿名性が確保できることを考察した。

### 4.2 秘匿性

本提案手法では図2の様に、パブリッシャ側とイニシエータ側の境界ノード間での通信は平文でやりとりさ

れる。そのため、匿名通信路を通して平文をやりとりした場合、この2つの境界ノードに内容を読み取られる危険性がある。

そのため、匿名通信路を確立後、パブリッシャーとイニシエータ間で相互認証と鍵交換を行い、平文をやりとりしないようにする必要がある。

### 4.3 メッセージサイズ

本提案手法では、各ノードがパブリッシュするキーの平均数を  $AVG_p$ 、イニシエータとなる平均数を  $AVG_i$  とし、バックアップ用の経路として各経路あたり  $MR$  個の経路を用意したとすると、ノードあたりの平均経路数は  $(L+2)(AVG_p+AVG_i)(1+MR)$  となる。そのため、匿名性が破綻する確率が結託ノードが2割未満の時1%程度となる  $L=4$ 、各ノードが1つの他ノードと匿名通信を行い ( $AVG_p = AVG_i = 1$ )、バックアップ用経路を1つ用意 ( $MR=1$ ) した場合を想定すると、最小構成に近いにもかかわらず、平均経路数は24に達する。しかし、パブリッシュ側オニオンルーティングやバックアップ用の経路は、常にデータのやりとりが発生するというわけではなく、要求があったときのみデータのやりとりが発生するので、その経路を常にTCPコネクションで張り続けるのはコストが高い。そのため、本提案手法ではコネクションレスプロトコルを利用しても動作するような構造になっている。

しかし、UDPではデータグラム長が限られているため、中継ノード数に上限ができてしまう。そこで、メッセージサイズを考察することで、UDPを利用した場合の最大中継ノード数について考察する。

利用する楕円曲線暗号の鍵長を  $ES$ 、共通鍵暗号の鍵長を  $KS$ 、ノード情報を  $NI$ 、送信するペイロード長を  $PL$ 、経路識別用の乱数長を  $RS$  とすると、中継ノード数が  $L$  の時のメッセージサイズ  $MS$  は次のようになる。

$$MS = L \cdot (ES + KS + NI) + PL + RS$$

楕円曲線暗号の鍵長として192bitを採用し、IPv4環境を想定した場合、鍵長が192bitより  $ES=24$ 、公開鍵暗号の強度と見合った共通鍵暗号の鍵長は128bitであるので、 $KS=16$ 。IPv4環境なのでIPアドレスが4バイト、ポート番号が2バイトとなり、 $NI=6$ 。適当な乱数長として  $RS=8$  とすると、UDPのデータグラム長を1000バイトと仮定すると  $L$  の範囲はおおよそ、 $L \leq 21 - \frac{PL}{46}$  となる。つまり、送信するデータが46バイト未満なら、最大中継ノード数は21に達することができる。中継ノード数を5に設定すれば、736バイトのデータを配送することができる。

以上より、中継ノード数に伴うメッセージサイズの増加は上記の環境を想定した場合、中継ノード1つあたり

46バイトであるため、最大データグラム長を1000バイトと想定したUDPを利用した場合でも十分な中継ノード数・配送データサイズを確保することが可能である。

## 5 おわりに

分散ハッシュテーブルだけでは、仮名をキーとした場合、本人到達性の高いIPアドレスが仮名と結びつけられてしまうため、匿名性が確保できないという問題があった。本研究では、分散ハッシュテーブルとオニオンルーティングを組み合わせることで、仮名をキーとして利用した場合でも、匿名性を確保することが出来る双方向匿名通信路の構築手法を提案した。

そして、匿名性や秘匿性、メッセージサイズに関する考察を行い、適切なパラメータを設定することにより、十分な性能を発揮できることを確認した。

今後の予定としては、本提案手法を実装し、可用性や遅延についての評価などを行う予定である。

## 参考文献

- [1] Li Zhuang, Feng Zhou, Ben Y. Zhao, and Antony Rowstron. Cashmere: resilient anonymous routing. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*, 2005.
- [2] 正基近藤, 彰一齋藤, 啓志松尾. DHTを用いた双方向匿名通信路の提案. 情報処理学会研究報告. CSEC, [コンピュータセキュリティ], Vol. 2008, No. 71, pp. 195–202, 2008.
- [3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. pp. 149–160, 2001.
- [4] Ben Y. Zhao, Ben Y. Zhao, John Kubiawicz, John Kubiawicz, Anthony D. Joseph, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, 2001.
- [5] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. pp. 53–65, 2002.
- [6] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In *Information Hiding*, pp. 137–150. Springer-Verlag, 1996.