IPv6 における着信側のプライバシー向上のための ワンタイムアドレスの実現

桜井 敦史† 蓑原 隆‡ 佐藤 良太†
拓殖大学大学院工学研究科† 拓殖大学工学部‡

概 要

近年インターネットにおけるプライバシー保護が重要になっているため、コンテンツを暗号化などで保護する技術が数多く提案されている。しかしアドレス情報を保護する技術が十分に確保されているとは言い難い。特に通信を開始するノードは送信元アドレスを変更することでプライバシーを確保することができるが、通信を受ける側のアドレスを変更した場合、本来の送信者には変更したアドレスがわかる仕組みが必要となる。そこで本稿は IPv6 通信において通信を受ける側のノードにワンタイムアドレスを実装することでアドレスとノードの関係付けを困難にし、第 3 者からプライバシーを確保する方法を提案する。これを実現するために両ノードで同一のアドレスリストを生成し、通信を開始するノードはアドレスリストを参照することで着信側のワンタイムアドレスにアクセスを行ない、着信側はこれを受け入れる。我々はこれまでにプロトタイプとして Linux カーネルに本システムの実装し、性能を評価した。通信の遅れはほとんどなくワンタイムアドレスの実現によって通信者の特定を困難にすることができた。

1 はじめに

近年、インターネットの利用者および利用目的の拡大に伴い、利用者のプライバシーの保護が重要になってきている[1]。インターネットで送られるメッセージをプライバシーの面からみると、メッセージの内容自体は IPsec[2]などの暗号化技術によって保護できるが、アドレス情報については隠蔽するとメッセージの配信自体が困難になるため容易には保護することができない。

1.1 アドレスにおけるプライバシー保護

アドレスにおけるプライバシーの保護として、複数の通信を関連付けることで新しい情報が引き出せるという問題が挙げられる。例えば、「2 つのメッセージが同一の送信者から送られている」といった情報や、「2 つのメッセージが同一の受信者に送られている」という情報をアドレスの関連付けによって第 3 者が取得できてしまう。このような情報はノードの性質を特定することや行動の追跡を容易にしてしまう恐れがあるため、アドレスにはプライバシー用語でいう「Unlinkability」[3]を持たせることが重要である。すなわち固定のアドレスを使い続けるのではなく、第 3 者に追跡されないアドレスに適宜変更することでプライバシーを確保できる技術が必要である。IPv6 では通信を開始するノード（発信ノード）の送信元アドレスをランダムに変更することで、プライバシーを確保する技術が標準で実装されている[4]。しかし、通信を受け入れるノード（着信ノード）がアドレスをランダムに変更させると正規の発信ノードのアクセスが困難になるため、着信側のプライバシー確保が制限されるという問題がある。

1.2 本研究の目的

本研究は着信側のアドレスに Unlinkability を持たせることでプライバシーを向上させることを目的とする。着信ノードのプライバシーの問題は公開サーバに対する通信よりも、P2P などの対等な通信においてより重要になると考えられる。したがって本研究は P2P 通信と親和性が高い IPv6 を対象とし、広大なアドレス空間を活用することで着信ノードのアドレスをワンタイムに変化させる手法を提案する。

本研究では許可した者同士で構成する Friend-to-Friend（F2F）[5][6]ネットワークのような、あらかじめ暗号キーなどの交換が可能なコミュニティを対象とする。また第 3 者による盗聴は公共のインターネット接続環境のネットワークの入り口など通信経路上（ネットワーク層）に設置されたスニファによって行なわれるものとし、データリンク層（MAC アドレス）によ

[1] One-Time Receiver Address in IPv6 for Protecting Privacy against Eavesdroppers.
[2] Atsushi SAKURAI, Graduate School of Engineering, Takushoku University
[3] Takashi MINOHARA, Department of Computer Science, Takushoku University
[4] Ryota SATO, Graduate of Engineering, Takushoku University

るノードの識別は本稿では考慮しない。

1.3 関連研究

着信ノードのアドレス情報を隠蔽する方法として Onion Routing[7]がある。これはアドレス情報を含むメッセージを多重に暗号化し、中継ノードで復号化・転送を行なうことでメッセージの送信元や最終的な宛先情報を中継ノードを含む第 3 者から隠す匿名通信である。これによって誰と誰の通信であるかを第 3 者が追跡することを困難にすることができる。しかしこの手法は中継ノードにおける演算処理の負担が大きく、また何層にもわたって暗号化を施すため経路長に応じてデータ量が増大するという問題点があり、広範囲に利用する場合にこれがボトルネックになると考えられる。

また着信ノードのプライバシーを確保する研究として、Waters 氏らは Incomparable Public Key[8]という暗号システムを使用する方法を提案している。この方法では暗号化したメッセージをマルチキャストで複数のノードに送ることで、着信ノードのプライバシーを確保する。Incomparable Public Key は公開鍵暗号の一種で正しい着信ノードのみがマルチキャストで送られたメッセージを復号できる。復号に失敗したノードはメッセージを破棄し、正しい着信ノード以外はどのノードが受け取ったかを知ることができないため着信ノードのプライバシーを確保することができる。しかしこの方法で着信ノードの推定を困難にするためには、マルチキャストグループのノード数が多いことが前提となるが、正当な受信者以外においてもメッセージの復号は試行されるため、ノード数が多くなるとその分余計なオーバヘッドが増加すると考えられる。さらにマルチキャストを使用しているため TCP 通信が困難になるという問題が挙げられる。

プライバシーを確保する技術は、従来の通信方法に比べ性能が劣らず、様々なアプリケーションで利用可能なスケーラビリティに優れた技術であることが望ましい。本研究で提案する手法は特別な中継ノードや高負荷な計算を必要とせず TCP を利用する様々なアプリケーションで実現可能である。

2 ワンタイムアドレスによる着信ノードのプライバシーの確保

発信ノードは新しい通信を開始するとき、宛先として今まで使用したことのない新しい着信ノードのアドレスを選択し、着信ノードはそのアドレスへの通信を受け入れ、通信終了時にアドレスをネットワークインタフェースから削除することでワンタイムアドレスを実現する。これによって着信側のアドレスとノードの関連付けを困難にする。

2.1 着信ノードアドレスのワンタイム化

図 1 にワンタイムアドレスの概念について示す。発信ノードは着信ノードに対する 1 回目のアクセスで Address1 を指定し、2 回目のアクセスには Address2 を指定するように 2 回以上同じアドレスを使用しない。

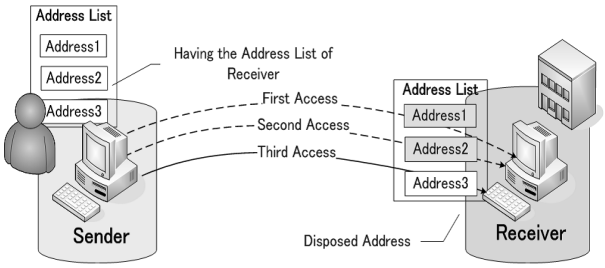

図 1: ワンタイムアドレスの概念

これを実現するために、まず発信ノードは計算で求めた複数のアドレスをアドレスリストとして保持しておく。そして通信を開始する時にリストから新しいアドレスを取得してこれを宛先としてアクセスする。着信ノードも発信ノードが持つ同じアドレスリストを計算で求め保持しておき、リストに存在するアドレスへのアクセスを受け入れる。両ノードはあらかじめ交換した暗号キーを使って独立に同一のアドレスリストを生成する。このアドレスリストを用いたワンタイムアドレスについて次の事項を実現する。

- 盗聴者は着信ノードが次に使用するアドレス及び、これまでに使用された複数のアドレスが同じ着信ノードであることを推測することができない
- 発信ノードと着信ノードはそれぞれ複数のノードとの通信に対応するために、複数の暗号キーから生成されたノードごとのアドレスリストを持つ

2.2 アドレスリストの生成

次に暗号キーを用いたアドレスリストの生成方法について示す。

1. ネットワークプレフィックスと、履歴バッファから前回の計算で求めた値（初回は後述する初期値を使用）に暗号キーを足し合わせた値を結合する
2. ステップ 1 で生成した値を MD5 を使用して 128 ビットのハッシュ値を計算する

3. ステップ 2 で生成した値の左 64 ビットを取り出し 6 ビット目（最も左のビットを 0 ビット目とする）を 0 にセットして、これを新しいインタフェース ID として使用する
4. ステップ 2 で生成した値の右 64 ビットを取り出し、これを履歴バッファに格納し、次回の計算に使用する

このアルゴリズムは最初のアドレス生成時に着信ノードが接続しているネットワークプレフィックスと計算に用いる初期値を両ノードで共有しておく必要がある。そこで着信ノードはルータから受け取る RA(Router Advertisement) メッセージから抽出したネットワークプレフィックスと、ランダムなインタフェース ID を結合したアドレスをダミーアドレスとして図 2 に示すように DNS サーバに登録する。DNS に登録することで発信ノードはダミーアドレスを初期値として取得でき、暗号キーを使うことで着信ノードのアドレスリストを生成することができる。

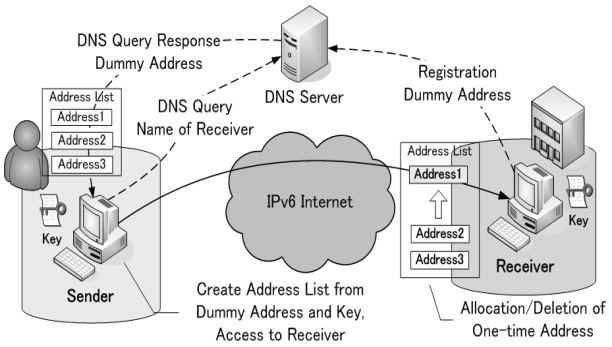

図 2: 発信ノードから着信ノードへのアクセス

2.3 着信側のワンタイムアドレスの管理

着信ノードにおいて、アドレスリストの全てのアドレスをネットワークインタフェースに割り当てることは効率的ではないため、アドレスリストの先頭アドレスのみを割り当て対象とする。着信ノードはアドレスリストを次の方法で管理する。実装の詳細は 3.1 で示す。

1. アドレスリスト生成
   着信ノードは最初の RA メッセージ受信時にダミーアドレスを生成し、発信ノードの数だけアドレスリストを生成する。アドレスリストの初期値は同一のダミーアドレスだが、発信ノードごとに共有した暗号キーを使用することでそれぞれ異なったリストになる。
2. アドレスの割り当て
   アドレスリストの先頭アドレスをインターフェースに割り当てておくことで、発信ノードからの通信を受け入れる。
3. アドレスの削除
   ワンタイムアドレスは通信が終わり次第、アドレスリストとネットワークインタフェースから削除する。通信終了の検知はトランスポート層で判断するか、タイムアウトを用いて削除を行なうことができる。

2.4 発信側のワンタイムアドレスの管理

発信ノードも複数の着信ノードに対応するために、それぞれ異なるキーによって独立したアドレスリストを生成し、それを通信開始時に参照することでアクセスを行なう。発信ノードは通信ごとに異なるワンタイムアドレスを選択する必要があり、さらに様々なアプリケーションでアクセスできることが望ましい。従ってアドレス選択はアプリケーションから意識せずに行なえる仕組みにする必要がある。また一般に IPv6 ではネットワーク層のアドレスを直接指定するのではなくノードの名前を指定することでアクセスすると考えられるので、着信ノードのワンタイムアドレスへのアクセスを名前解決の形で実現する。これは 3.2 で述べるように着信ノードのワンタイムアドレスの解決プログラムを開発し、これを名前解決ライブラリに統合した。

3 ワンタイムアドレスの実装

我々はワンタイムアドレスを実現するために、これまでに示した処理を USAGI プロジェクト[9]が開発した IPv6 をサポートしている Linux に実装した。

3.1 着信ノードの実装

着信ノードはアドレスの割り当て・削除処理を実装するために Linux のプロトコルスタックに新しい処理を追加した。本来 Linux ではネットワークレイヤで受信したパケットの IP ヘッダのチェックを行なった後、宛先が自ノード宛であるか他のノード宛であるかを判断するルーティング処理に移る。我々はルーティング処理の前に着信ノードの処理（アドレスリストの生成処理・アドレスの割り当て・削除処理）を追加した。次に追加した処理の詳細を述べる。

3.1.1 アドレスリストの生成処理

着信ノードは起動時、ルータから RA メッセージを受信して 2.3 で示したように着信ノードごとにアドレスリストを生成する。データ構造を図 3 に示す。アドレスリストの先頭アドレスをネットワークインタフェー

スに割り当てているため、受信したパケットが割り当て中のアドレスであるかを高速に探索できるように、各アドレスリストの先頭をリストとして結合する。通信終了した使用済みのアドレスはリンクを付け替えることでリストから削除し、次に使用するアドレスをリストの先頭アドレスにする。

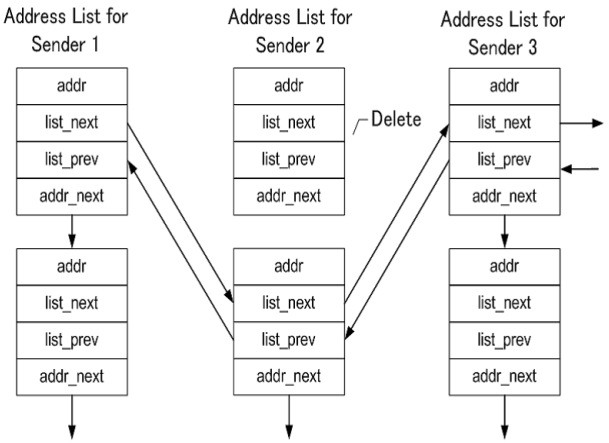

図 3: アドレスリストのデータ構造

3.1.2 アドレスの割り当て・削除処理

着信ノードは通信が終了したときアドレスを削除し、次の新規接続要求に応えるために、次のアドレスをあらかじめ割り当てておく。この処理のアルゴリズムを図 4 に示す。まず受信したパケットのタイプが FIN パケットであった場合、そのターゲットアドレス（packet->target）をネットワークインタフェースから削除するべきか、各アドレスリストの先頭アドレス（addrlist->addr）と比較する。もしターゲットアドレスがアドレスリストに存在した場合は、削除処理を行なう。実際のインターフェースから削除する処理（del_addr）はタイマーを用いて行なう。これは発信ノードに対して FIN/ACK パケットの送信など TCP 切断の処理を行なう必要があるため、カーネルのタイマーを利用して 3 秒後に削除する実装にしている。削除命令を行なった後、次に先頭アドレスとなるアドレス（addrlist->addr_next->addr）をネットワークインターフェースに割り当てる。その後、アドレスリストの付け替え（change_list）を行ない、使用済みのアドレスを図 3 のように削除し、既存のルーティング処理に移行する。また受信した FIN パケットの宛先がアドレスリストに存在しなかった場合は処理を加えずルーティング処理に移行する。

```
if( packet->type == FIN ){

  /* for each senders */
  while( addrlist != NULL ){

    /* Check whether target
       address exists in the list */

    if( packet->target == addrlist->addr ){
      /* Set the timer of function
         that Delete the address
         from the interface */

      timer.expires = jiffies + 3000;
      timer.data = addrlist->addr
      timer.function = del_addr;
      add_timer( &timer )

      /* Assign the address
         into the interface */

      add_addr( addrlist->addr_next->addr );

      change_list( addrlist );
      break;
    }

    addrlist = addrlist->list_next;
  }
}
```

図 4: アドレス割り当て・削除処理のアルゴリズム

3.2 発信ノードの実装

発信ノードのユーザは着信ノードに対して一般的なアプリケーションでアクセスすることができ、その都度変化するワンタイムアドレスを指定する必要がある。そこで一般的に用いられる名前解決処理を拡張し、着信ノードの名前を指定するだけで自動的にワンタイムアドレスを指定できる仕組みを実装した。

3.2.1 名前解決処理

Linux システムの名前解決に使われる標準ライブラリ（glibc-2.4）は容易に拡張できるように設計されている。名前解決の設定ファイル（/etc/nsswitch.conf）に複数のサービスが記述でき、これは記述した順に実行される。我々はアドレスリストを管理するアドレス管理プログラムと、着信ノードのアドレスをアドレス管理プログラムから取得するサービスライブラリを開発した。このライブラリは nsswitch.conf の先頭に記述し、着信ノード以外の名前についてはアドレス管理プログラムがライブラリにエラーを返すことで、ライブラリは処理をスキップし、通常どおり NIS、DNS などのサービスによって名前解決が行うことができる。アドレス管理プログラムは名前解決にあたって次の処理を行なう。

- 着信ノードごとのアドレスリストを生成する

  発信ノードはアドレス管理プログラムの起動時に登録しておいた着信ノードごとのダミーアドレスを DNS から取得して 2.2 のようにアドレスを生成する。アドレスリストの構成は図 3 とほぼ同様であるが、着信ノードの名前を保持しておくメンバを新たに追加している。

- ライブラリに着信ノードのアドレスを返す

  ライブラリとアドレス管理プログラムはソケット通信によってデータ交換を行なう。ライブラリは名前をアドレス管理プログラムに渡し、アドレス管理プログラムは取得した名前が着信ノードであるかそれぞれのアドレスリストを検索する。着信ノードの名前であった場合はアドレスリストの先頭アドレスを返し、異なる場合はエラーを返す。

3.2.2 アドレスの削除処理

発信ノードもアドレスリストの検索を先頭アドレスのみに限定させるため、通信終了時には使用したアドレスをリストから削除する。通信終了の検知はアドレス管理プログラムがプロミスキャスモードでパケットをキャプチャしておき、着信ノードから FIN パケットを受信した時に、アドレスリストからアドレスを削除する。

4 動作実験と性能評価

図 5 のようなネットワークを構築してプロトタイプの動作実験を行なった。発信ノード（Sender: Linux-2.6.17, PentiumD 2.80GHz）が着信ノード（Receiver: Linux-2.6.18, AMD Athlon64 1GHz）に対して ssh で通信を行なう場合、毎回違うワンタイムアドレスを宛先としてアクセスし、着信ノードはそのアクセスを受け入れた。また ping6 で着信ノード以外の名前を指定した場合、新しく追加したライブラリはスキップされ、DNS で名前解決を行ない IPv6 のサイトにアクセスしたことを確認した。発信ノードが着信ノードのワンタイムアドレスに標準的なアプリケーションを用いてアクセスできる処理を、既存のシステムと併用できる形で実現できた。

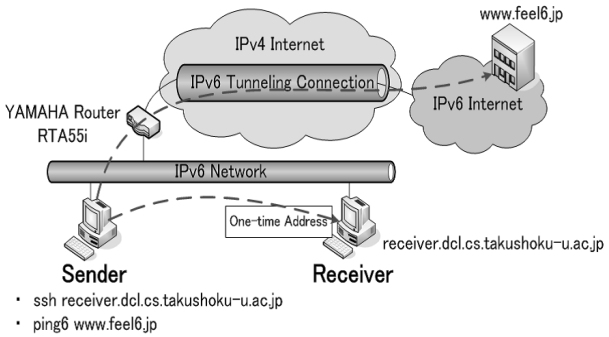

図 5: 実験環境と動作確認

着信ノードの性能評価としてワンタイムアドレスを実装していない標準のカーネルと、ワンタイムアドレスを実装した拡張カーネルにアクセスした場合の RTT の差を ping6 を用いて測定した（表 1）。実験環境は図 5 と同様に同一リンク内で行なった。1 回目の RTT は標準のカーネルに比べ約 0.2ms の遅れが生じ、2 回目以降は約 0.04ms の遅れを確認した。RTT の差はごく小さく、通常の通信をおいて誤差の範囲であるといえる。

5 本研究の効果と今後の課題

本稿で提案した方法は特別なノードの設置や高負荷な処理を必要としないため、通信の遅れがほとんど生じず、TCP 接続を行なう様々なアプリケーションで利用可能である。着信ノードのアドレスをワンタイムにすることでアドレスの関連付けを困難にし、さらに特

表 1: 標準カーネルと拡張カーネルの RTT の差

送信回数	1	2	3	4
標準カーネル (ms)	1.115	0.151	0.149	0.150
標準偏差 (ms)	0.414	0.007	0.009	0.009
拡張カーネル (ms)	1.291	0.190	0.191	0.192
標準偏差 (ms)	0.310	0.011	0.015	0.009

定のノードに対する Man-in-the-middle 攻撃やリプレイ攻撃の対策としても期待できる。次に本研究の問題点や今後検討すべき事柄について述べる。

- MAC アドレスによる識別

  着信側のリンク内で盗聴された場合、MAC アドレスによるノードの識別が可能になる。今後 MAC アドレスをランダムに変化させることで関連付けを困難にする手法を検討していく。

- アドレスリストの同期

  アドレスリストは両ノードで同期していなければならないため、システムや通信の障害によってリストの状態に差が生まれたときの対策が必要である。今後着信側の受け入れるアドレスの範囲を広げる方向で検討し、実用性を高めていく。

- プレフィックス情報への対応

  上位 64 ビットのプレフィックス情報によってどの組織間の通信であるかといった情報を取得できる場合がある。今後プレフィックス情報を含めたアドレスの Unlinkability について検討していく。

6 まとめ

本稿で IPv6 通信における着信側のプライバシーを確保するためのワンタイムアドレスの実現方法について述べた。本研究は発信ノードと着信ノードとの間であらかじめ共有しておいた暗号キーを用いてアドレスリストを生成する。発信ノードは通信開始時にアドレスリストから得た新しいアドレスを送信先に指定し、着信ノードは次に使用するアドレスをネットワークインタフェースに割り当てておくことで通信を受け入れる。

提案した手法の実装として、発信ノードは新しい名前解決ライブラリを追加することで、一般的なアプリケーションで着信ノードの変化するワンタイムアドレスにアクセスするシステムを実現した。着信ノードは Linux の IPv6 プロトコルスタックを改変し、通信終了時にネットワークインタフェースからアドレスを削除し、次回使用するアドレスを割り当てることでワンタイムアドレスを実現した。

性能評価として変更を加えていない標準の Linux とワンタイムアドレスを実装した Linux で RTT の差を測定した。標準の Linux に比べて本システムに大きな遅れはなく、ssh など複数のアプリケーションでワンタイムアドレスを使った通信ができることを確認した。

参考文献

[1] Goldberg, I.: Privacy-enhancing technologies for the Internet,II, five years later. In: Proceeding of the Workshop on Privacy on Privacy Enhancing Technologies. LNCS 2009, San Francisco, CA, USA (April 2002) 1-12

[2] Kent, S., Seo, K.: Security architecture for the Internet protocol. RFC 4301 (December 2005)

[3] Haddad, W., Nordmark, E. : Privacy Aspects Terminology draft-haddad-alien-privacy-terminology-02, Network Working Group, Internet-Draft.

[4] Narten, T., Draves, R.: Privacy extensions for stateless address autoconfiguration in IPv6, RFC3041 (January 2001)

[5] Bricklin, D.: Friend-to-friend networks. http://www.bricklin.com/f2f.htm (August 2000)

[6] Chothia, T., Chatzikokolakis, K.: A survey of anonymous peer-to-peer file-sharing. In: NCUS 2005: Proceedings of the IFIP International Symposium on Network-Centric Ubiquitous Systems. LNCS 3823, Nagasaki, Japan (2005) 744-755

[7] David Goldschlag, Michael Reed, and Paul Syverson.: Onion Routing, Communications of The ACM, February 1999, Vol.42, No.2, pp.39-41,

[8] Waters, B.R., Felten, E.W., Sahai, A.: Receiver anonymity via incomparable public keys. In: CCS '03: Proceedings of the 10th ACM conference on Computer and Communications Security, Washington D.C., USA (2003) 112-121

[9] WIDE Project: USAGI project - linux IPv6 development project. http://www.linux-ipv6.org/