目　次

概　要

本論文では移動透過性における端末移動時の認証機構として Diffie-Hellman 鍵交換を利用したダイレクト認証とアドバンスト認証の 2 つの認証方式を提案する. ダイレクト認証はエンド端末間で直接 Diffie-Hellman 鍵交換を行う. この方式は中間者攻撃に対して課題はあるものの実用上は十分なセキュリティ強度を持つことができる. アドバンスト認証は DDNS サーバを改造し, Diffie-Hellman 鍵を 2 つに分解し, 一方を直接エンド端末同士で, もう一方を DDNS サーバを経由して交換する. この方式は DDNS を改造する必要があるがセキュリティ強度は Return Routability や LIN6 で提案されている認証機構より高い. 提案方式を Mobile PPC へ実装し, 処理時間の測定を行なった. その結果, 通信に影響を与えるようなオーバヘッドをほとんど発生せず実現可能であることが分かった.

1. はじめに

いつでも誰でもどこからでもネットワークへのアクセスが可能なユビキタス社会を実現するために,移動しながら通信を行える環境が要求されている.しかし,インターネットでは端末が通信中に移動をするとIPアドレスが変化するため，通信が継続できないという問題がある．そこで，端末の移動によるIPアドレスの変化を隠蔽し,通信を継続できるようにする移動透過性の研究が盛んに行われている[1].

移動透過性を実現する代表技術として，プロキシサーバを用いるMobile IP[2]，エンドエンドでこれを実現するMobile IPv6[3]，LIN6(Location Independent Networking for IPv6)[4],[5]，MAT(Mobile IP with Address Translation)[6]，Mobile PPC(Mobile Peer to Peer Communication)[7]などがある．現状のネットワークはIPv4が主体であることから移動透過性を早期に普及させるにはIPv4での実装が可能であることが望ましいが，IPv4に対応しているのは現在のところMobile IPとMobile PPCのみである.

Mobile IPは，移動ノードMN(Mobile Node)の位置を管理するHA(Home Agent)を導入し,通信相手ノードCN(Correspondent Node)が自分の通信相手がHAであるかのように見せかける移動透過性技術である．しかし，特殊な装置が必要となるほか，通信経路に冗長が発生したり，トンネル転送時に余分なヘッダが必要となったりするなどの課題があり,P2P(Peer-to-Peer)通信が主流になると想定される今後のユビキタス社会に適している方式とはいえない.

Mobile PPCはこの様な課題を解決するために提案された技術で,IPv4を対象としつつ，エンド端末だけにより移動透過性を実現する．Mobile PPCではDDNS(Dynamic DNS)[8]を利用して通信を開始する.通信中にMNが移動すると，MNからCNに対して変化情報を直接通知し，両端末のIP層の中にアドレス変換テーブルを生成する．以後の通信パケットは上記アドレス変換テーブルに基づき変換するため,IPアドレスの変化を上位ソフトウェアから隠蔽することができる.

ここで，移動透過性を実現するに当たり，必須となるのが移動時の相互認証にかかわる課題である．移動透過性を実現するには，一般に移動に係わる情報を相手端末に伝えるために変化情報を通知するシーケンスを新たに定義する必要がある．このシーケンスを利用して，攻撃者がMNに成りすましてCNへ変化情報を通知すると，通信を乗っ取られる危険性がある．従って，移動時にCNとMNが確実に認証する必要がある.

移動時の認証に関しては，Mobile IPv6では当初，IETF(Internet Engineering Task Force)においてCNとMNとの間にIPsec[9]を適用する方法が検討されて

いた．IPsec では認証に使用する共通鍵を IKE(Internet Key Exchange)[10]により自動生成する．しかし，IKE を実行するには PKI(Public Key Infrastructure)の整備が必須であり，現状の PKI が未整備である状況を考えるとこの方法は現実的でないとして却下された経緯がある．そこで Mobile IPv6 では，Return Routability という認証機構が導入されることとなった．

Return Routability では通信に先立って共通鍵を生成し，更にこれを二つに分解して，一方をエンド端末同士で，もう一方を HA を経由して交換することによりエンド端末間で共通鍵を共有する．経路を 2 分割することにより，同時に盗聴されない限り安全に共通鍵を共有できる．端末の移動時には上記の共通鍵を用いて相互認証を行う．LIN6 においても MA(Mapping Agent)と呼ばれる特殊な位置管理装置を用いることにより Return Routability と類似の認証方式を実現している[11].

[7]では，Mobile PPC の認証方式として Diffie-Hellman 鍵交換により事前に共通鍵を共有する方式が示唆されていたが正確な定義はなされていなかった.

そこで, 本論文では, Diffie-Hellman 鍵交換[12]による認証を厳密に定義し, 2 通りの方式を検証した. まず, 1 つ目は, エンド端末間で直接 Difiie-Hellman 鍵を交換するダイレクト認証である．Diffi-Hellman 鍵交換は中間者攻撃に弱いといわれているが適切な対策をとることにより，実用上十分なセキュリティ強度を持つことができる. 2 つ目は, DDNS サーバを改造し, Diffie-Hellman 鍵を 2 つに分解し，一方を直接エンド端末同士で，もう一方を DDNS サーバを経由して交換するアドバンスト認証である．アドバンスト認証は DDNS を改造する必要があるがセキュリティ強度は Return Routability や LIN6 で提案されている認証機構より高い.

提案方式を FreeBSD5.2.1 上に実装し, 動作確認と性能測定を実施した結果, 通信に影響を与えるオーバヘッドはほとんど発生させずに実現可能であることが分かった. 本提案方式は Mobile PPC への適用時に最も有効な手段となりうるが，他の移動透過性技術に対しても有効な方式である.

以下, 2 章で攻撃手法の定義し, 3 章で従来技術の例として Return Routability と Mobile PPC の概要を記述し，4 章で Mobile PPC における認証方式の概要と詳細を記述する．5 章で Mobile PPC における認証方式の実装，6 章で性能測定結果と従来技術との比較を行う．最後に 7 章でまとめる.

2. 攻撃手法の定義

移動透過性における端末移動時の相互認証では，エンド端末間で認証に使用する共有鍵をどのようにして安全に交換するかが解決すべき課題となる.

CN と MN が鍵交換を行う際，攻撃者が共有鍵を入手する方法として盗聴による入手と，中間者攻撃による入手の 2 つの手法が考えられる．ここで盗聴による入手とは，攻撃者が通信する二者が交換する情報を盗聴し，この情報を用いて共有鍵を入手する方法を意味する．中間者攻撃による入手とは，攻撃者が通信を行なう二者の間に割り込んで，両者が交換する公開情報を自分のものとすりかえることにより，共有鍵を入手する方法を意味する．以降これらの攻撃を単に盗聴，中間者攻撃と呼ぶ．盗聴は攻撃者にとって比較的簡単に実行できる攻撃手法であるが，中間者攻撃を実行するには，攻撃者はパケットのキャプチャ，解析，改ざん，送信などの手順をリアルタイムで行う必要があるため，難易度が極めて高い攻撃手法である．

3. 既存技術

3.1. Return Routability とその課題

Return Routability では, HA を積極的に利用し MN と HA 間に信頼関係があることを前提とする．共通鍵を二つに分解し，それぞれ異なる経路から配送する．

Return Routability の動作を図 1 に示す．CN，MN はともに移動可能なノードであり，CN の位置を管理する HA を HAcn，MN の位置を管理する HA を HAmn と記述する．ここで，MN と HAmn 間は信頼関係を期待できるものとし，事前に共通鍵を保持させ，この区間では IPsec による通信を行なう．CN と HAcn 間にも IPsec が適用可能であるが, Return Routability のパケットは単なる IP-in-IP トンネリングで HAcn を経由する．

MN は CN へ Return Routability を開始する Home Test Init (HoTI)（(1)）および Care-of Test Init (CoTI)（(2)）と呼ばれるパケットを同時に送信する．これらのパケットには HoTI および CoTI に対する CN からの応答を認証するために HoTI には home init cookie，CoTI には care-of init cookie と呼ばれる乱数が含まれる．HoTI は HAmn と HAcn を経由し，CN へ送信される．CoTI は HA を経由せず平文のまま CN へ送信される. CN は HoTI と CoTI の二つの Test Init を受信したら，それぞれに対して Home Test(HoT)（(3)）および Care-of Test(CoT)（(4)）と呼ばれるパケットを同時に送信する. HoT には home keygen token と呼ばれる値と MN から受け取った home init cookie , CoT には care-of keygen token と呼ばれる値と MN から受け取った care-of init cookie が含まる．HoT は HAcn と HAmn を経由し，MN へ送信される．CoT は HA を経由せず平文のまま MN へ送信される．MN は CN から HoT と CoT を受信すると，そ

こに含まれる home init cookie と care-of init cookie を検証し CN の認証を行い，home keygen token と care-of keygen token から共通鍵を作成する．

MN は通信中に移動すると共通鍵を利用して認証データを作成し，これを BU(Binding Update)と呼ばれる移動情報通知パケットに付加し CN へ送信する．CN は，自身が生成した共通鍵を用いて，BU パケットに付加された認証データを検証し MN の認証を行う．

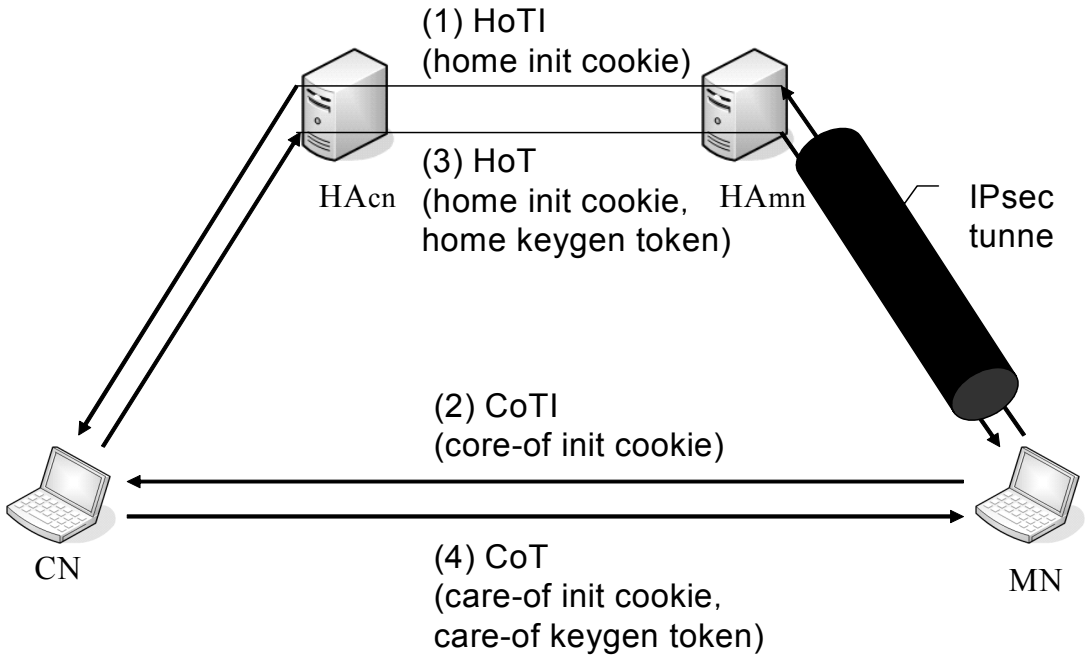

図 1　Return Routability

Return Routability では HAcn と HAm 間，および CN と MN 間の二つの経路上に攻撃者が存在した場合，keygen token の組を盗聴することができるが異なる経路上の同時盗聴は困難であるため実用上大きな問題はないとしている．しかし，HAcn と HAmn 間の経路は特定できるため，必ずしも安全が保障されているわけではない．LIN6 の認証では，第 1 回目の token の交換を MA(Mapping Agent)経由で，第 2 回目の token の交換をエンド端末間で直接行うことにより共通鍵を生成する方式をとるが同時盗聴に関する条件は RR と同じである．

3.2. Mobile PPC とその認証方式

本論文で示す提案方式は Mobile PPC を念頭においたものであるため，

Mobile PPC の概要をまず示し，次に Mobile PPC の認証方式の現状を示す．

Mobile PPC を適用したシステムでは，通信開始時の IP アドレスの解決には，DDNS(Dynamic DND)を利用する．端末移動時の IP アドレスの解決には Mobile PPC を用いる．図 2 に Mobile PPC における移動情報の通知を示す．Mobile PPC ではエンド端末が新旧 IP アドレスの対応関係を示すテーブル(Connection ID Table; 以下 CIT)を保持する．IP アドレスが変化すると，その直後に MN と CN の間で移動後の IP アドレスと継続させるべき通信の識別情報を CU(CIT Update)および CU REPLY により交換する（CU ネゴシエーション）．このネゴシエーションによりエンド端末の CIT が更新される．以後の通信ではパケット送受信時にネットワーク層で CIT を参照してアドレス変換を行う．この方式により，上位ソフトウェアに対し IP アドレスの変化を隠蔽し，通信を継続させることができる．

Mobile PPC は冗長な経路が発生せず，常に最適経路での通信が行われる．パケットサイズの冗長がなく高スループットが実現できる．IP アドレスは従来のアドレス体系をそのまま利用できる．通信開始時に DDNS を利用するが，これは移動透過性を実現するために必要となる特有な装置ではない．このため，既存環境への適用が容易であり，特有の装置を二重化するなどの措置も不要で管理が容易である．さらに，Mobile PPC は既存端末との上位互換性があり，既存端末との通信は移動しない限り可能である．このため，段階的な普及が期待できる．原理的に IPv4 と IPv6 の両者に適用可能である．これらの利点より，今後のユビキタス社会に最も適した方式として期待できる．

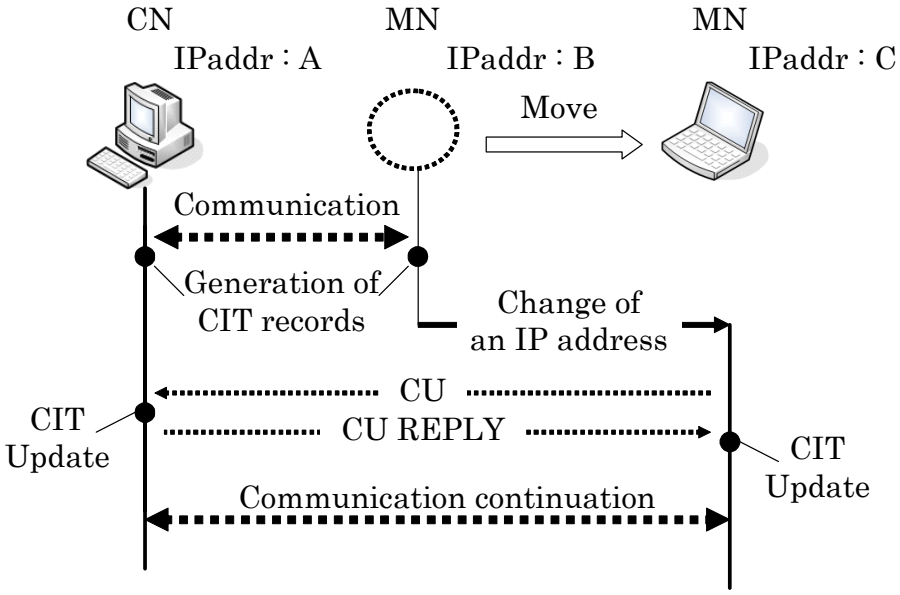

図 2　Mobile PPC における移動情報の通知

Mobile PPC における移動時の認証は，通信開始時に Diffie-Hellman 鍵交換を行うことにより共通鍵を生成しておく方法が示されている[7]．しかし，厳密な定義は行われておらず，詳細は今後の課題となっていた．また通信開始までに大きな遅延が発生することが指摘されていた．

4. 提案方式

本論文では，エンド端末間で直接 Diffie-Hellman 鍵を交換するダイレクト認証と，第三の装置を介して Diffie-Hellman 鍵を交換するアドバンスト認証を提案する．

4.1. 提案方式の概要

Diffie-Hellman 鍵交換とは，両端末間において，離散対数問題を利用したアルゴリズムに従って生成した Diffie-Hellman 鍵を交換することにより，盗聴者には知ることのできない共通鍵を端末間で生成する鍵交換方式である．

提案方式では，通信に先立ち端末間で鍵交換ネゴシエーションを行う機構を追加し，このネゴシエーションにより cookie 交換と Diffie-Hellman 鍵交換を行う．cookie 交換は，通信相手端末 CN が Mobile PPC 実装端末であるかどうかの判別と，送信元 IP アドレスを偽造した成りすましによる DoS 攻撃(Denial of Service attack)を防止するためのものである．MN が移動した際は，通信に先立って生成した共通鍵を用いて端末間で相互認証を行う．

Diffie-Hellman 鍵交換は認証機能がないため，中間者攻撃に弱いという指摘がある．そこで，中間者攻撃の有無をチェックし適切な処置をとることにより安全性を高めた．また，通信開始と併行して共通鍵を生成することにより，通信開始時の遅延を抑えた．

4.1.1. 用語定義

Mobile PPC における認証方式に用いる用語を以下に定義する．CN と MN は通信相手ごとに cookie と呼ぶ乱数を生成する．CN が生成した cookie を cookie_cn と呼び，MN が生成した cookie を cookie_mn と呼ぶ．また，アドバンス認証では CN と MN は通信相手ごとに cookie を 2 つ生成する．CN が生成した 2 つの cookie をそれぞれ cookie_cn_a，cookie_cn_b と呼び，MN が生成した 2 つの cookie をそれぞれ cookie_mn_a，cookie_mn_b と呼ぶ．

CN と MN は cookie 交換の後，Diffie-Hellman 鍵交換を行う．CN と MN は priv_key と呼ぶ秘密鍵と pub_key と呼ぶ公開鍵を生成する．ここで，CN が生成した秘密鍵と公開鍵のペアをそれぞれ priv_key_cn と pub_key_cn と呼び，

MN が生成した秘密鍵と公開鍵のペアをそれぞれ priv_key_mn と pub_key_mn と呼ぶ．また，アドバンス認証では pub_key を 2 つに分割する．2 つに分割した pub_key_cn をそれぞれ pub_key_cn_a，pub_key_cn_b と呼び，2 つに分割した pub_key_mn をそれぞれ pub_key_mn_a，pub_key_mn_b と呼ぶ．また，Diffie-Hellman 鍵交換により生成した共有鍵を K と呼ぶ．

MN が移動し，IP アドレスが変化したとき，MN は CU パケットと共有鍵から MAC(Message Authentication Code)を作成する．ここで MN が生成した MAC を MAC_mn と呼ぶ．MN は CU パケットに MAC_mn を付加し CN へ送信する．CN は CU パケットを受信すると付加された MAC_mn を検証し MN の認証を行う．CN は MN を認証すると，CU REPLY パケットと共有鍵から MAC を作成する．ここで，CN が生成した MAC を MAC_cn と呼ぶ．CN は CU REPLY パケットに MAC_cn を付加し送信する．MN は CN から CU REPLY パケットを受信すると付加された MAC_cn を検証し CN の認証を行う．

$$MAC = f(msg\ ,\ K) \qquad (1)$$

ここで，f()は一方向ハッシュ関数を表し，msg は CU または CU REPLY メッセージ全体を表す．

4.2. ダイレクト認証

ダイレクト認証は通信に先立ちエンド端末間のみで Diffie-Hellman 鍵交換を実行することにより共有鍵を生成する．移動時にはこの共有鍵を使用した認証を行う．ダイレクト認証による認証処理の流れを図 3 に示す．通信に先立つネゴシエーションは，通信を開始した端末から実行する．以下では，MN から通信を開始したものとする．MN は cookie_mn を生成し，CN へ送信

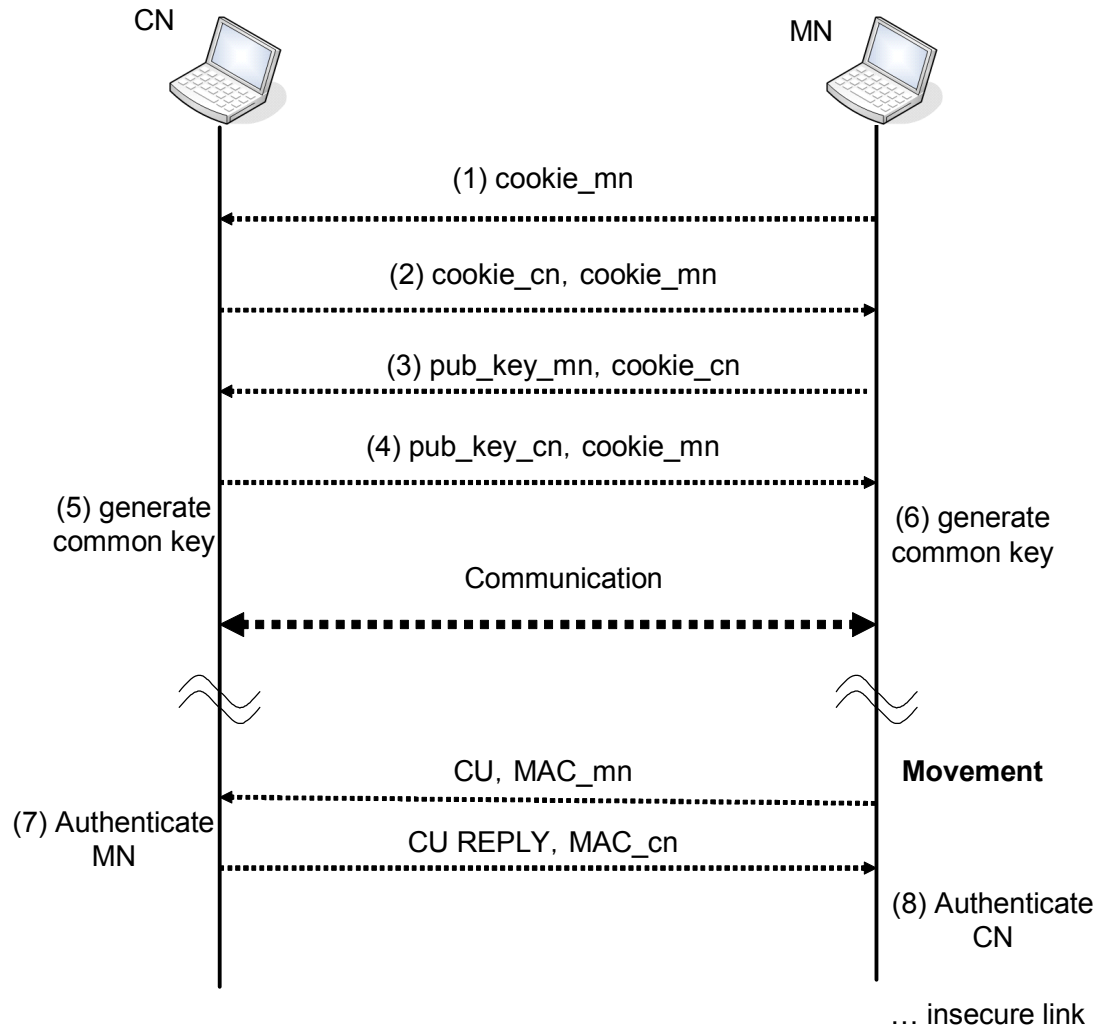

図３　ダイレクト認証

する（(1)）．CN はこれを受信すると，自身の cookie_cn を生成し，cookie_mn と cookie_cn を MN へ送信する（(2)）．MN は CN から受信した cookie_mn と自身が生成した cookie_mn を比較し CN の簡易認証を行う．

簡易認証に成功すると，MN は pub_key_mn と cookie_cn を送信する（(3)）．CN は MN からこれを受信すると，cookie_cn と自身が生成した cookie_cn を比較し MN の簡易認証を行う．CN は簡易認証に成功すると pub_key_cn と cookie_mn を返信する（(4)）．MN はこれを受信すると，cookie_mn と自身が生成した cookie_mn を比較し CN の簡易認証を行う．

両端末間で pub_key の交換が完了すると，端末自身が保持する priv_key と相手端末から受信した pub_key により共有鍵 K を生成する（(5),(6)）．移動時には共有鍵 K を用いて認証を行う（(7),(8)）．

ダイレクト認証で共有鍵 K が安全に交換できているかを以下に検証する．共有鍵を生成するには Diffie-Hellman 鍵交換の特性上 priv_key_mn と pub_key_cn ，または pub_key_mn と priv_key_cn を入手する必要がある．しかし攻撃者は priv_key_mn や priv_key_cn を入手することができないため盗聴に対する脆弱性はない．しかし，CN と MN 間の通信経路上に攻撃者がおり，CN と MN 間で行われる鍵交換ネゴシエーションに割り込み，両者が交換する pub_key を自分のものとすりかえた場合，攻撃者は CN と MN との間に共有鍵を生成することができる．

以上より，ダイレクト認証は，MN と CN 間の通信経路上における中間者攻撃に対して脆弱性があるといえる．しかし，中間者攻撃の実行には極めて高度な技術が必要であり，一般用途での使用においては，本方式の導入はセキュリティ上十分有効であると考えられる．

4.3. アドバンスト認証

アドバンスト認証は通信に先立ち端末間で Diffie-Hellman 鍵交換を第三の装置を介して実行する．本方式では pub_key を 2 つに分解し，それぞれ異なる経路から配送することにより共有鍵を生成する．移動時にはこの共有鍵を使用した認証を行う．第三の装置としては DDNS を改造して用いる．アドバンスト認証による認証の流れを図 4 に示す．CN，MN ともに移動可能なノードであり，CN の位置を管理する DDNS を DDNScn，MN の位置を管理する DDNS を DDNSmn と記述する．ここで，CN と DDNScn 間，MN と DDNSmn 間は信頼関係を期待できるものとし，事前に共有鍵を保持させ，この区間では IPsec による通信を行なう．

MN は cookie_mn_a と cookie_mn_b を生成し，CN へ送信する．cookie_mn_a は DDNSmn と DDNScn を経由し，MN と DDNSmn 間，DDNScn と CN 間は IPsec で保護され CN へ送信する（(1)）．cookie_mn_b は平文のまま CN へ直接送信する（(2)）．CN は cookie_mn_a と cookie_mn_b を受信すると，cookie_cn_a と cookie_cn_b を生成し，MN へ送信する．cookie_cn_a には cookie_mn_a が付加され，DDNScn と DDNSmn を経由し，CN と DDNScn 間，DDNSmn と MN 間は IPsec で保護され MN へ送信する（(3)）．cookie_cn_b には cookie_mn_b が付加され，平文のまま MN へ直接送信する（(4)）．MN は CN から cookie_cn_a と cookie_cn_b を受信すると付加された cookie_mn_a と cookie_mn_b を検証し CN の簡易認証を行う．MN は CN の簡易認証に成功すると，次に，端末間で Diffie-Hellman 鍵交換を行なう．

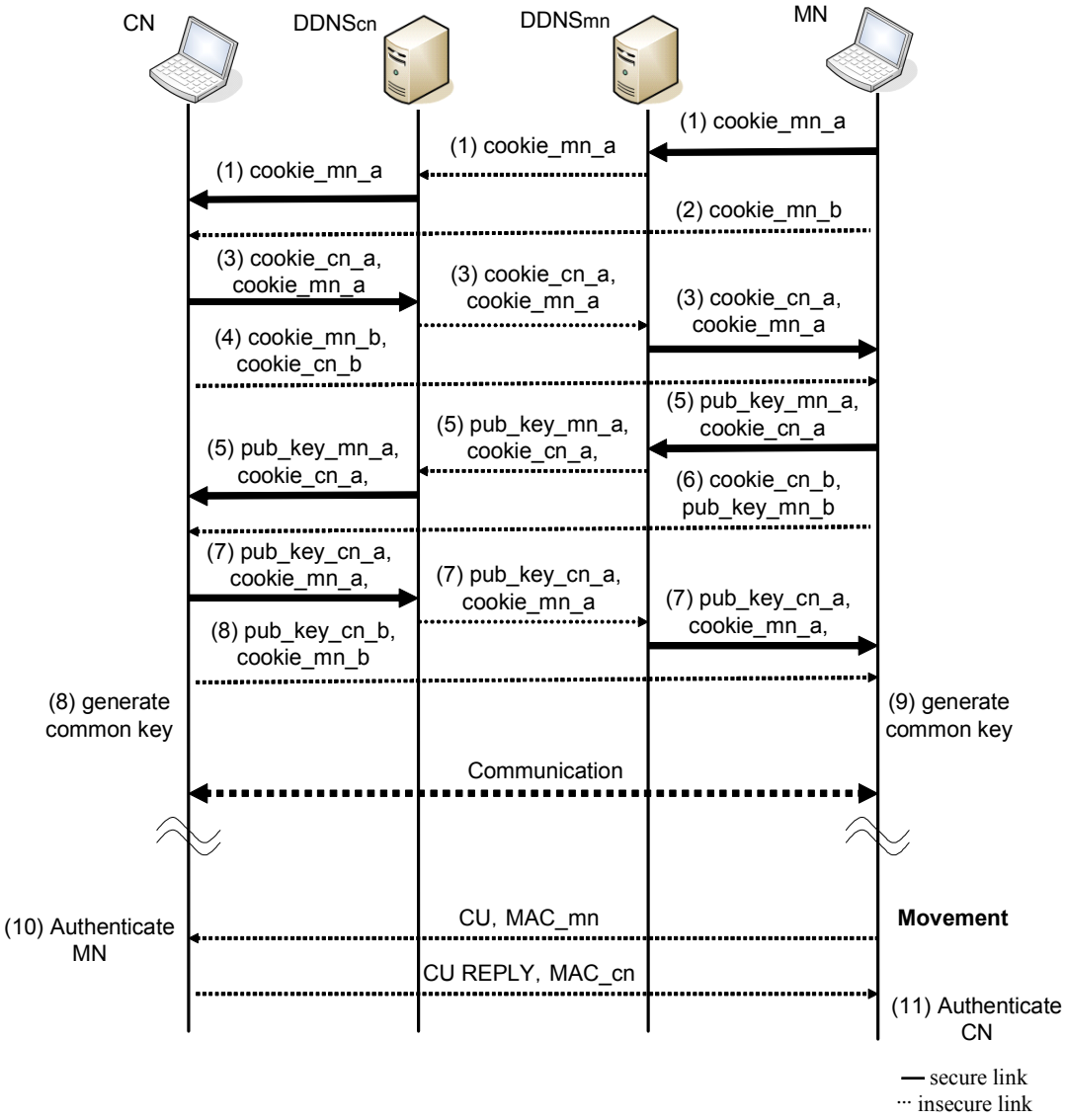

図 4　アドバンスト認証

MN は pub_key_mn を pub_key_mn_a と pub_key_mn_b の 2 つに分解し，CN へ送信する. pub_key_mn_a には cookie_cn_a が付加され, DDNSmn と DDNScn を経由し，MN と DDNSmn 間，DDNScn と CN 間は IPsec で保護され CN へ送信する（(5)）. pub_key_mn_b には cookie_cn_b が付加され，平文のまま CN へ直接送信する（(6)）. CN は MN から pub_key_mn_a と cookie_mn_b を受信すると付加された cookie_cn_a と cookie_cn_b を検証し MN の簡易認証を行う. CN は MN の簡易認証に成功すると，MN から受信した pub_key_mn_a と pub_key_mn_b の二つを合せて pub_key_mn を得る. その後, CN は pub_key_cn を pub_key_cn_a と pub_key_cn_b の 2 つに分解し，MN へ送信する.

pub_key_cn_aにはcookie_mn_aが付加され, DDNScnとDDNSmnを経由し, CNとDDNScn間, DDNSmnとMN間はIPsecで保護されMNへ送信する((7)). pub_key_cn_bにはcookie_mn_bが付加され，平文のままMNへ直接送信する((8)). MNはCNからpub_key_cn_aとpub_key_cn_bを受信すると，付加されたcookie_mn_aとcookie_mn_bを検証しCNの簡易認証を行う. MNはCNの簡易認証に成功すると, CNから受信したpub_key_cn_aとpub_key_cn_bの二つを合せてpub_key_cnを得る. 両端末間でpub_keyの交換が完了すると，端末自身が保持するpriv_keyと相手端末から受信したpub_keyにより共有鍵を生成する((9)). 移動時にはこの共有鍵を用いて認証を行う((10),(11)).

では，共有鍵が安全に交換できているかを検証する. 前述したように共有鍵を生成するにはDiffie-Hellman鍵交換の特性上priv_key_mn とpub_key_cn , または pub_key_mn とpriv_key_cn を入手する必要があるため，本方式の場合においては priv_key_mn と pub_key_cn_a と pub_key_cn_b , またはpub_key_mn_aとpub_key_mn_b とpriv_key_cnを得ることができれば共有鍵を生成することができるといえる.

攻撃者はpriv_key_mnやpriv_key_cnを入手することができないため本手法は盗聴に対する脆弱性がない. しかし，DDNScnとDDNSmn間，CNとMN間の二つの通信経路上に攻撃者がおり，これら二つの通信経路上で行われる鍵交換ネゴシエーションに割り込み，両者が交換するpub_keyを自分のものとすりかえた場合，攻撃者はCNとMNとの間に共有鍵を生成することができる.

以上より，アドバンスト認証は，DDNScnとDDNSmn間，CNとMN間の二つ通信経路上におけるに同時中間者攻撃に対して脆弱性があるといえる. しかし，二つの経路上における同時中間者攻撃の実行には極めて高度な技術が必要であり，一般用途での使用においては，本方式の導入はセキュリティ上十分有効であると考えられる.

5. 実装

Mobile PPCにおける認証方式の 1 つであるダイレクト認証の実装をFreeBSD5.2.1上で行った. ダイレクト認証は従来の Mobile PPCにモジュールを追加することにより実現した. MACの計算にはHMAC_SHA1を使用した.

5.1. NIT

Diffie-Hellman鍵交換は端末ペア単位に1回だけ実行するため, 出力されるパケットが初回であるかどうかの判断を行なうための情報，共通鍵，それに

係わる cookie 交換や Diffie-Hellman 鍵交換などの情報を記録させるテーブルが必要である.このテーブルを NIT(Node Information Table)と呼び Mobile PPC の仕様に新たに追加した．図 5 に NIT フォーマットを示す．NIT は，自端末/通信相手端末の IP アドレス，自端末/通信相手端末の cookie，自端末/通信相手端末の Diffie-Hellman 鍵交換に使用する乱数，共有鍵，状態の 8 つのフィールドから構成される.

自端末 IPアドレス	相手端末 IPアドレス	自端末 cookie	相手端末 cookie	自端末 DH鍵	相手端末 DH鍵	共通鍵	状態
CN	MN1	A	B	E	F	Key1	Done
CN	MN2	C	D	G	E	Key2	Done

図 5　NIT フォーマット

5.2. モジュール構成

Mobile PPC は，パケット送受信時には IP 入力関数である ip_input から，パケット送信時には IP 出力関数である ip_output から Mobile PPC モジュールを呼び出し，アドレス変換処理を終えたら差し戻す形をとっている．IP アドレス変更時には ARP 関数より Mobile PPC モジュールが呼ばれ，移動情報通知処理を行う．既存の処理にはいっさい変更を加えない.

Mobile PPC を実現するモジュールは CIT 操作モジュール，アドレス変換モジュール,移動管理モジュールの 3 つがある,既に実装と評価を終えている.

本実装は，これまでの Mobile PPC にモジュールを追加することにより，認証方式を実現する.モジュール構成を図 6 に示す.追加するモジュールは NIT 操作モジュール，Diffie-Hellman 鍵交換モジュール，認証モジュールの 3 つがある．また，既存の移動管理モジュールから認証モジュールと NIT 操作モジュールを呼び出せるように修正を加える.

NIT 操作モジュールは，NIT レコードの検索・生成・更新を行う．Diffie-Hellman 鍵交換モジュールは,パケットの送信および受信時に呼び出され，入出力パケットの通信識別子をキーとして NIT 検索を行い，必要であれば cookie 交換と Diffie-Hellman 鍵交換を実行する．認証モジュールは CU および CU REPLY パケットに対して MAC の生成・付加・検証を行う.

Diffie-Hellman 鍵交換モジュールは，著者らが別途研究を行っている DPRP (Dynamic Process Resolution Protocol)[13]を流用することで比較的容易に実現することができた．DPRP は通信に先立ち端末間が互いに情報を交換し，端末間の通信に必要な動作処理情報テーブルを動的に生成する機能を持つ.

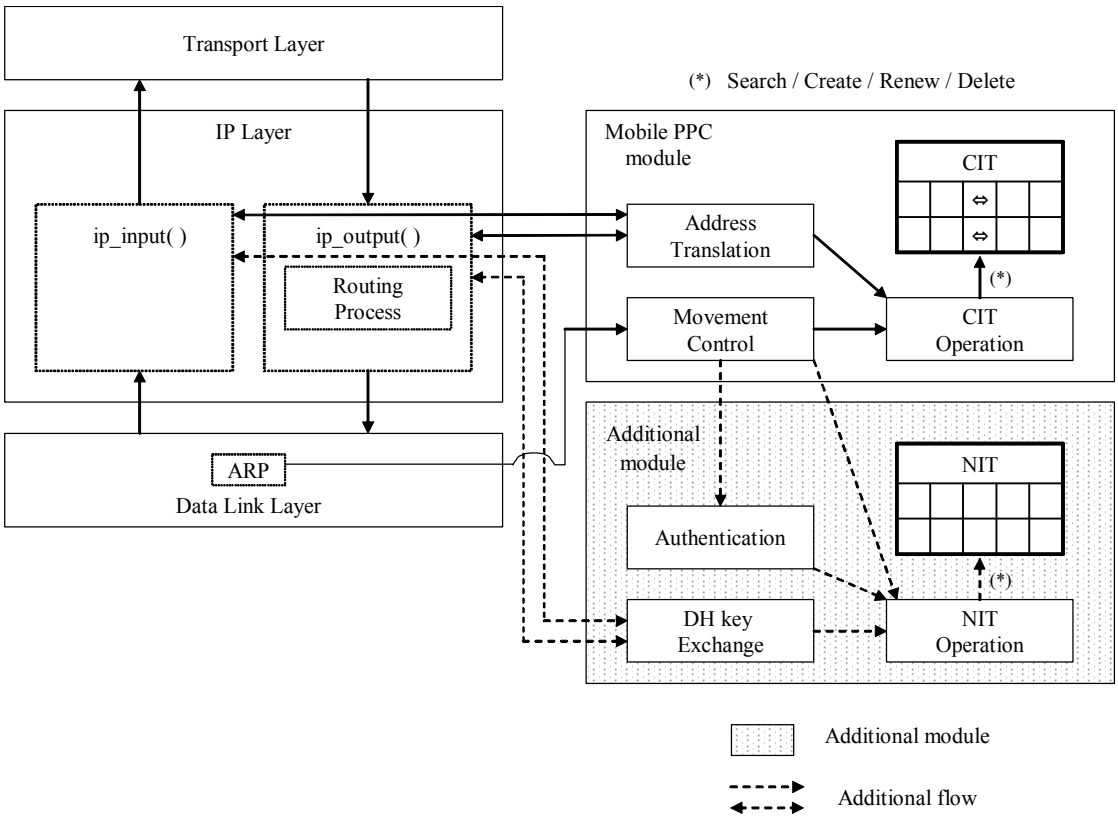

図 6　Mobile PPC のモジュール構成

6. 評価

ダイレクト認証を試作し，両エンド端末間で実行される通信に先立つネゴシエーションによる共有鍵の生成処理および移動時における移動情報通知の認証処理が正常に動作していることを確認した．本章では，ダイレクト認証における鍵交換処理時間と移動情報通知処理時間の測定した．移動情報通知処理時間に関しては従来の Mobile PPC の処理時間と比較した．また，ダイレクト認証の測定結果よりアドバンスト認証の処理時間を推測した．

6.1. 実験環境

ダイレクト認証および従来の Mobile PPC の処理時間を図 7 に示す測定環境で測定した．2 つのルータ R1，R2 によりサブネットが異なる 3 つのネットワークを用意し，MN の移動先となるネットワークには DHCP サーバを設置した．表 1 に装置仕様を示す．

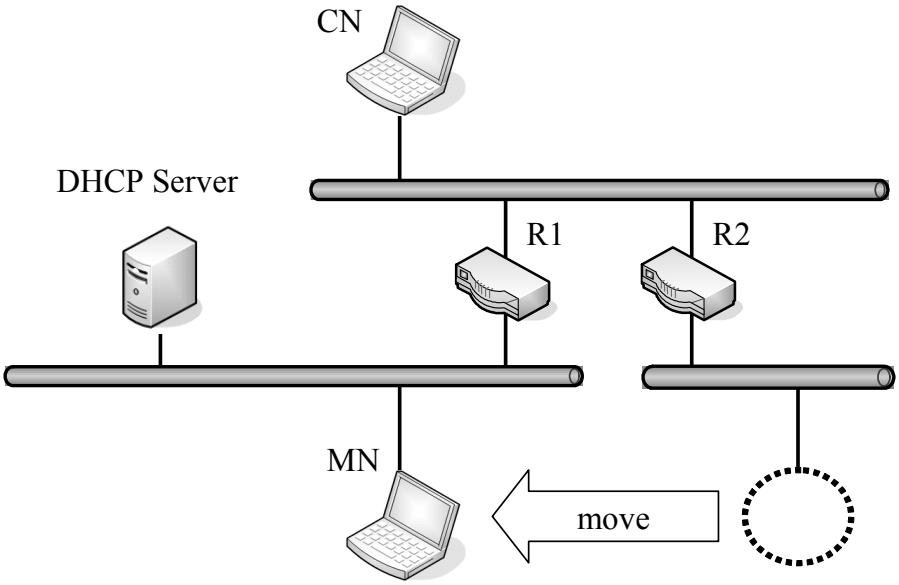

図 7　実験環境

表 1　装置仕様

	MN / CN / R1 / R2
CPU	Penitium 4 3.0GHz
Memory	512 MB
NIC	100BASE-TX
OS	FreeBSD 5.2.1-RELEASE

6.2. ダイレクト認証による鍵交換処理時間

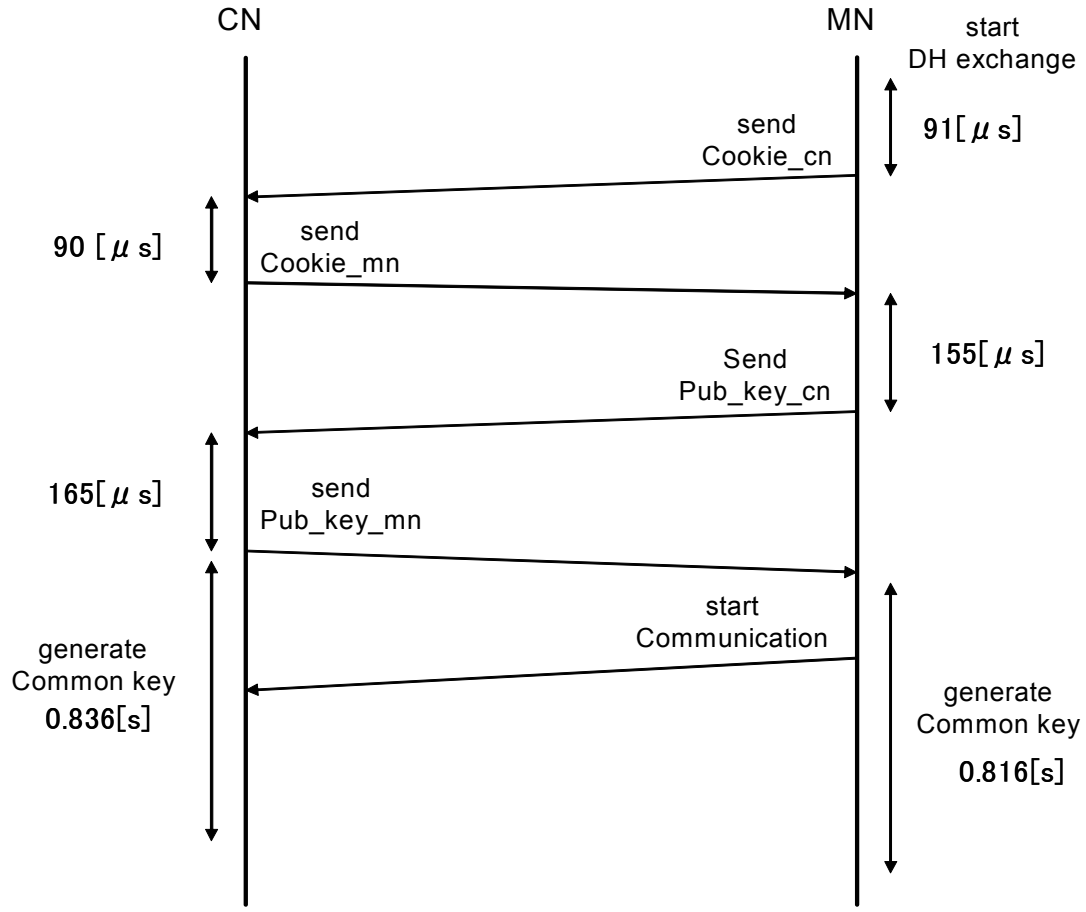

図 8　ダイレクト認証による鍵交換処理時間

ダイレクト認証による鍵交換時間を図 8 に示す．pub_key は事前に作成が可能であるため，端末起動時に生成した．図 8 より処理時間は 501μ秒[☆1]（[☆1] 91+90+155+165=501）である．この結果より通信開始時に発生するオーバヘッドはほとんど無視できることがわかる.共通鍵の生成に MN では 0.816 秒，CN では 0.836 秒の時間を要するが，この処理はデータ通信と並列して行うことができるため，鍵交換時間から除外した．通信直後に MN がすると，共通鍵がまだ生成できていない可能性があるが，CU ネゴシエーションを共通鍵生成後まで待つことにより移動透過性は問題なく実現できる.

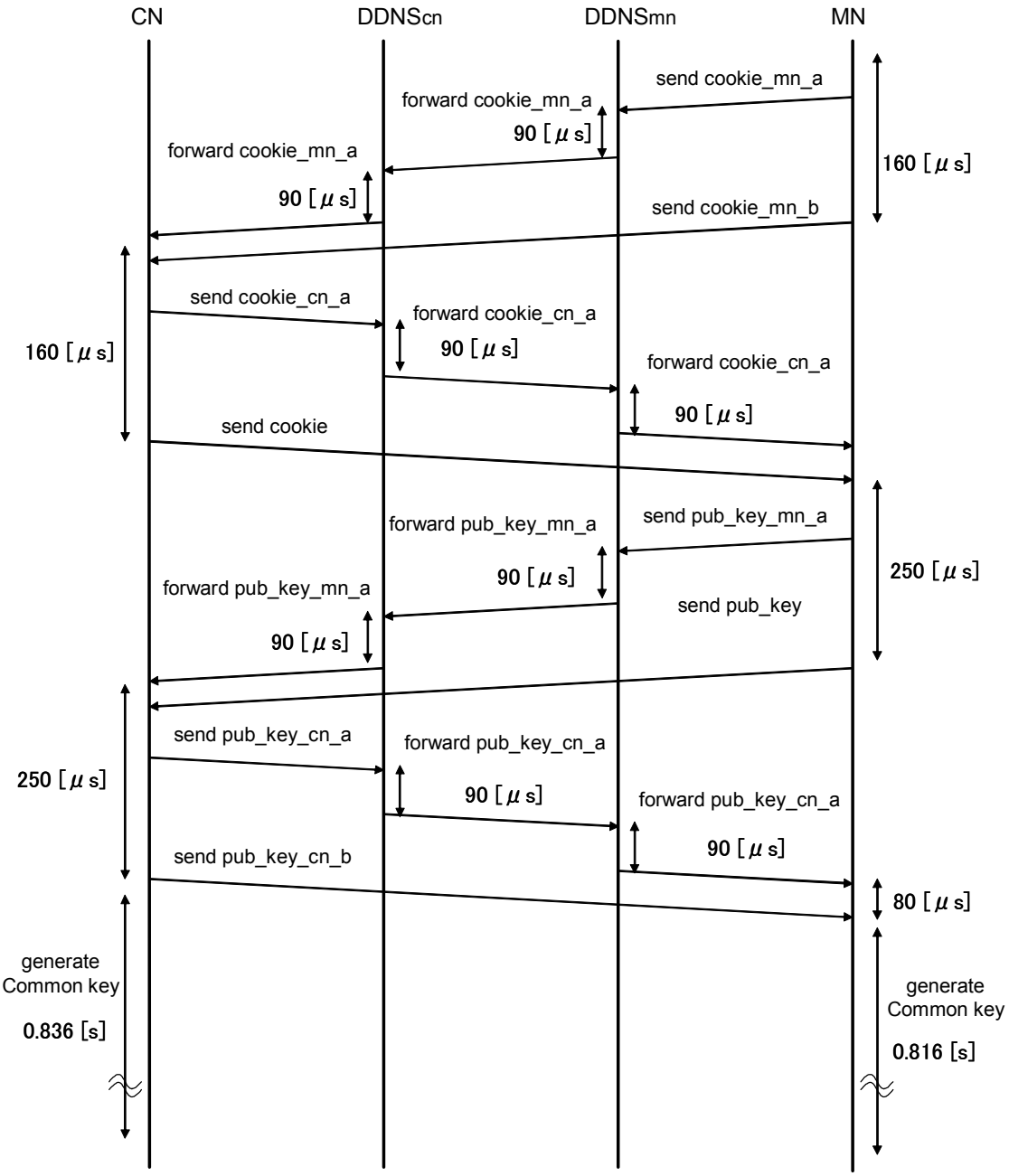

図 9　アドバンスト認証による鍵交換処理時間

6.3. アドバンスト認証による鍵交換処理時間

ダイレクト認証の結果をもとにアドバンスト認証の処理時間を推測した．推測した処理時間を図 9 に示す．図 9 よりアドバンスト認証による鍵交換処

理時間は 1620μ秒 ☆2(☆2 2*(160+90+90)+2*(250+90+90)+80=1620))であり，通信に先立つネゴシエーションで発生するオーバヘッドはダイレクト認証と同様にほとんど無視できる．

6.4. 移動情報通知処理時間

表 2　移動情報通知処理時間

	内部処理時間[ms]
従来のMobile PPC	0.644
認証処理を追加したMobile PPC	0.685

表 2 に認証処理を行わなかった場合の Mobile PPC における CU ネゴシエーション処理時間と認証処理を行った場合の CU ネゴシエーション処理時間を示す．前者は 644μ秒，後者は 685μ秒で，6%の増加となった．両者の差は MAC 演算処理の有無に起因するが，共通鍵暗号の処理であるため極めて高速に実行できる．

6.5. Return Routability との比較

提案技術と Return Routablility の比較を表 3 に示す．Return Routability は CN と HAcn と HAmn 間，CN と MN 間の同時盗聴，特に CN の近傍での盗聴に対して脆弱性がある． LIN6 における認証機構は MAcn と MAmn 間，CN と MN 間の同時盗聴に対して脆弱性がある．ダイレクト認証とアドバンスト認証は盗聴に対して脆弱性はない．

中間者攻撃について比較を行なう．Return Routability は CN と HAcn と HAmn 間，CN と MN 間の同時中間者攻撃，特に CN の近傍での中間者攻撃に対して脆弱性がある．LIN6 における認証機構は MAcn と MAmn 間，CN と MN 間の同時中間者攻撃に対して脆弱性がある．ダイレクト認証は CN と MN 間の中間者攻撃に対して脆弱性がある．アドバンスト認証は DDNScn と DDNSmn 間，CN と MN 間の同時中間者攻撃に対して脆弱性がある．

Return Routablity，LIN6 における認証機構，アドバンスト認証は特殊な第三の装置の導入が必要である．ダイレクト認証は第三の装置の導入が必要ない．また，Return Routablity，LIN6 における認証機構，アドバンスト認証は MN と第三の装置間に設定の複雑な IPsec を導入する必要があり，管理負荷が増加する．ダイレクト認証は IPsec を導入する必要がない．

表 3　Return Routability との比較

	セキュリティ		運用管理
	盗聴	中間者攻撃	
Return Routability	△	△	×
ダイレクト認証	○	△	○
アドバンスト認証	○	○	×

7. まとめ

本論文では移動透過性における端末移動時の認証機構として Diffie-Hellman 鍵交換を利用したダイレクト認証とアドバンスト認証の 2 つの認証方式を提案した.ダイレクト認証はエンド端末間で直接 Diffie-Hellman 鍵交換を行う．この方式は中間者攻撃に対して課題はあるものの実用上は十分なセキュリティ強度を持つことができる．アドバンスト認証は DDNS サーバを改造し, Diffie-Hellman 鍵を 2 つに分解し, 一方を直接エンド端末同士で, もう一方を DDNS サーバを経由して交換する．この方式は DDNS を改造する必要があるがセキュリティ強度は Return Routability や LIN6 で提案されている認証機構より高い．提案方式を Mobile PPC へ実装し，処理時間の測定を行なった．その結果，通信に影響を与えるようなオーバヘッドをほとんど発生せず実現可能であることが分かった．

謝辞

本研究を進めるにあたり，多大なるご指導，ご鞭撻を賜りました名城大学理工学部情報工学科　渡邊晃教授に心より厚く御礼申し上げます．また，お忙しい中，貴重な御意見を頂きました，名城大学理工学部情報工学科　小川明教授，柳田康幸教授，宇佐見庄五講師に心より厚く御礼申し上げます．

そして，有益なご助言，ご検討を頂きました渡邊研究室の皆さんに深く感謝いたします．

参考文献

[1] 寺岡文男：インターネットにおけるノード移動透過性プロトコル，電子情報通信学会論文誌，vol.J87-DI，No.3，pp.308-328 (2004).

[2] Perkins. C.: IP Mobility Support for IPv4, RFC 3344, IETF (2002).

[3] Johnson. D, Perkins. C, Arkko. J.: Mobility Support in IPv6, RFC3775, IETF (2004).

[4] Ishiyama. M, Kunishi. M, Uehara. K, Esaki. H, Teraoka: LINA:A New Approach to Mobiity Support in Wide Area Networks, IEICE Transactions on Communication, vol.E84-B No.8 pp.2076-2086 (2001).

[5] 國司光宣，石山政浩，植原啓介，寺岡文男：移動体通信プロトコル LIN6 の性能評価，情報処理学会論文誌，Vol.43, No.2, pp.398-407 (2002).

[6] 相原玲二，藤田貫大，前田香織，野村嘉洋：アドレス変換方式による移動透過インターネットアーキテクチャ，情報処理学会論文誌，vol.43，no.12，pp.3889-3897 (2002).

[7] 竹内 元規，鈴木 秀和，渡邊 晃：エンドエンドで移動透過性を実現する Mobile PPC の提案と実装，情報処理学会論文誌，Vol.47，No.12，pp.3244-3257 (2006).

[8] Vixie. P., Thomson. S., Rekhter. Y., and Bound. J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).

[9] Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC2401, IETF (1998).

[10] Harkins, D. and Carrel, D.: The Internet key exchange (IKE), RFC2409, IETF (1998).

[11] 田中康之，國司光宣，石山政浩，寺岡文男：LIN6 および HLIN6 における認証機構，電気情報通信学会論文誌，vol.J87-D-I No.5，pp.497-507 (2004).

[12] Diffie, W. and Hellman, M.: New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp.644-654 (1976).

[13] 鈴木秀和，渡邊　晃：フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装と評価，情報処理学会論文誌，Vol.47，No.11，pp.2976－2991 (2006).

研究業績

1. 国際会議

1) Masaki Sejimo and Akira Watanabe, “Implementation of Mobile PPC Realizing Mobility of Mobile Nodes”, International Symposium on Information Theory and its Applications (ISITA) 2006, Oct.2006.
2) Ayako Kanemoto, Masaki Sejimo and Akira Watanabe, “Proposal of a packet-lossless handover in Mobile PPC”, IEEE International Region 10 Conference (TENCON2006), Nov.2006.

2. 口頭発表

1) 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC における認証方式の提案”, 平成 16 年度電気関係学会東海支部連合大会講演論文集, Sep.2004.
2) 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC における認証方式の提案”, 第 67 回情報処理学会全国大会講演論文集, Mar.2005.
3) 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC における認証方式の実装に関する検討”, 平成 17 年度電気関係学会東海支部連合大会講演論文集, Sep.2005.
4) 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC における移動端末の認証”, マルチメディア, 分散, 協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.129-132, Jul.2005.
5) 金本 綾子, 瀬下 正樹, 竹内 元規, 渡邊 晃, “IPv6 環境での移動透過性を実現する Mobile PPCv6 の検討”, 平成 17 年度電気関係学会東海支部連合大会講演論文集, Sep.2005.
6) 金本 綾子, 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC におけるパケットロスなしハンドオーバの提案”, 第 68 回情報処理学会全国大会講演論文集, Mar.2006.
7) 葛谷 章一, 瀬下 正樹, 渡邊 晃, “プロキシを利用した Mobile PPC の検討”, 平成 18 年度電気関係学会東海支部連合大会講演論文集, Sep.2006.
8) 瀬下 正樹, 渡邊 晃, “Mobile PPC における認証方式の実装”, マルチメディア, 分散, 協調とモバイル（DICOMO2006）シンポジウム論文集(II), Vol.2006, No.6, pp.809-812, Jul.2006.
9) 金本 綾子, 瀬下 正樹, 竹内 元規, 渡邊 晃, “Mobile PPC におけるパケットロスなしハンドオーバの提案”, マルチメディア, 分散, 協調とモバイル（DICOMO2006）シンポジウム論文集(II), Vol.2006, No.6, pp.817-820, Jul.2006.