Mobile PPC における認証方式の実装

瀬下　正樹[†]　渡邊　晃[†]
[†]名城大学大学院理工学研究科

Implementation of Authentication Mechanisms in Mobile PPC

Masaki Sejimo[†]　Akira Watanabe[†]
[†]Graduate School of Science and Technology, Meijo University

1. はじめに

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行われている[1].

ネットワーク層における移動透過性保証プロトコルとして Mobile IP[2],Mobile IPv6[3]，LIN6(Location Independent Networking for IPv6)[4],[5]などが提案されているが，Mobile IP と Mobile IPv6 では Home Agent(HA)，LIN6 では Mapping Agent(MA)と呼ばれる特殊な第三の装置を用意する必要があり，導入するための敷居が高い．著者らは，これらの特殊な第三の装置を必要とすることなく，エンドツーエンド方式で移動透過性を実現する Mobile PPC(Mobile Peer to Peer Communication)[6]の研究を行っている．

Mobile PPC では Dynamic DNS(DDNS)[7],[8]を利用して通信を開始する．通信中に移動して MN(Mobile Node)の IP アドレスが変化すると，MN から CN(Correspondent Node)に対して変化情報を直接報告し，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．この方法により，IP アドレスの変化は上位ソフトウェアから隠蔽することができ，移動透過性を容易に実現することが出来る．ここで，MN から CN に対して変化情報を通知する際にはセッションの乗っ取りを防ぐため CN は MN を確実に認証する必要がある．

インターネットにおいて，端末間で認証を行う方法として，共通鍵暗号を利用する方式と公開鍵暗号を利用する方式がある．共通鍵暗号を利用する方式は，認証したい相手端末と共有鍵を事前に設定しておく必要がある．しかし，CN と通信する MN は任意であるため，一般的にこのような設定をしておくことは難しい．公開鍵暗号を利用した認証は，PKI のしくみを適用することで認証が可能であるが，現在の PKI が未整備である状況を考慮すると現実的でない．このため，Mobile PPC における端末間の認証では，CN と MN 間において認証に使用する鍵を，いつ，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．

移動透過性に伴う認証を行うための認証機構として Mobile IPv6 で提案されている Return Routability と，それと同様の手法を LIN6 に適用した方式[9]がある．しかし Return Routability では HA ，LIN6 では MA のような特殊な第三の装置を利用するため，Mobile PPC のようにエンドツーエンドで移動透過性を保証するプロトコルには適していない．

著者らは第三の装置を必要とせずエンドツーエンドで認証を行う Mobile PPC における認証方式[10]の提案を行なっている．Mobile PPC における認証方式は，Diffie-Hellman 鍵交換[11]を利用する．Mobile PPC に，通信に先立ち端末間でネゴシエーションを行う機構を追加する．このネゴシエーションにより MN と CN に Diffie-Helman の共有鍵を保持させておき，移動時にこの共有鍵を用いて MN の認証を行う．これにより第三の装置を使用することなくエンドツーエンドで認証機能を実現する．

本稿では，Mobile PPC における認証方式を FreeBSD 上に実装し，動作確認を実施したので報告する．

2. Mobile IPv6 と Return Routability

2.1. Mobile IPv6

Mobile IPv6 は IPv6(IP version6)において移動透過性を保証する．Mobile IPv6 では MN の位置を管理する HA が必要である．通信開始時においては，CN から MN 宛のパケットを HA が受信し，MN の移動先に届くように，パケットに IP ヘッダを追加するトンネリング転送を行う．一方，MN が移動し IP アドレスが変化した際は，エンドツーエンドで移動透過性を保証する経路最適化機能を使用する．経路最適化は，CN にも MN の移動前後の IP アドレスの対応関係を示すテーブル(Binding Cache; BC)を保持させ，MN が移動し IP アドレスが変化した直後に CN へ Binding Update(BU)により新しい IP アドレスを登録する．以後の通信では，パケットの送受信時に両端末の IP 層において BC と IPv6 拡張ヘッダを利用したアドレス変換を行う．CN が BC を保持していないときは通信開始時と同様に HA を利用した移動透過な通信を行なう．

セキュリティの観点から，移動時において，CN は MN から BU を受信する際，MN を確実に認証する必要がある．Mobile IPv6 では，以下に述べる Return Routability を用いて認証を行う．

2.2. Return Routability

2.2.1. Return Routability の概要

Return Routability は Mobile IPv6 で提案されている認証機構である．この方式は，MN と HA 間の信頼関係を利用する．共有鍵となる情報を二つに分解し，それぞれ異なる経路から配送する．これにより，CN と MN 間で安全に共有鍵を生成する．

Mobile IPv6 では，BU パケット送信の直前に Return Routability を行い，これにより MN と CN に共有鍵を保持させ，BU 時にこの共有鍵を用いた認証を行う．

Return Routability の動作を図 1 に示す．ここで，MN と HA 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[9]による通信を行なう．MN は CN へ Return Routabhility 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 HA を経由し，MN と HA 間は IPsec で保護され 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 HA を経由し，HA と MN 間は IPsec で保護され 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 から共有鍵を作成する．

MN は BU パケットと共有鍵から認証データを作成し，これを BU パケットに付加し CN へ送信する．CN は BU パケットを受信すると，自身が生成した home keygen token と care -of keygen token により共有鍵を作成し，BU パケットに付加された認証データの検証し MN の認証を行う．

2.2.2. Return Routability の課題

Return Routability は HA のような特殊な第三の装置に依存した認証機構であるため Mobile PPC のようにエンドツーエンドで移動透過性を保証するプロトコルには適していない．

また，Return Routability では Init cookie の組で MN が CN の認証を行い，keygen token の組で CN が MN の認証を行っているため， Init Cookie の組 や keygen token の組を攻撃者が入手することができれば CN や MN への成りすましが可能となる．CN と HA 間と CN と MN 間の二つの経路上に攻撃者が存在した場合，init cookie の組や keygen token の組を盗聴することができる．特に，攻撃者が CN と同一セグメント上に接続した場合，init cookie の組や keygen token の組を容易に盗聴することができる．以上より，Return Routability は CN と HA 間と 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．

3. Mobile PPC とその認証方式

3.1. Mobile PPC

Mobile PPC は，通信開始時において相手の IP アドレスを知る方法（初期 IP アドレスの解決）と通信中に変化した相手の IP アドレスを知る方法（継続 IP アドレスの解決）を明確に分離する．初期 IP アドレスの解決には，ホスト名と IP アドレスの関係を動的に管理する DDNS を利用する．継続 IP アドレスの解決には Mobile PPC を用いる．Mobile PPC はエンド端末に新旧 IP アドレスの対応関係を示すテーブル(Connection ID Table; CIT)を保持させる．IP アドレスが変化すると，その直後に MN から CN に対して，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 CIT UPDATE(CU)により通知し，CN は MN に対して CU 応答を返信する（図 2）．このネゴシエーションによりエンド端末で 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（図 3）．これにより，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 では拡張ヘッダや HA を使用しないため，ヘッダオーバヘッドや通信経路の冗長化などの問題が発生しない．原理的に IPv4 と IPv6 のどちらにも適用可能な方式である．

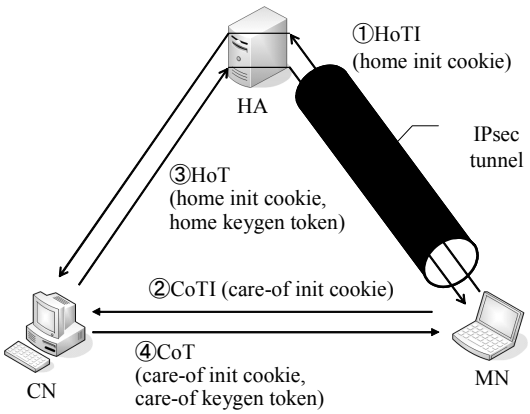


図 1　Return Routability の動作

Mobile PPC では，CN が CU を受信する際，MN を確実に認証する必要があるが，Return Routability のような第三の装置を利用する手法は適していない．この問題を解決するために，次に述べる Mobile PPC における認証方式を提案する．

3.2. Mobile PPC における認証方式

Mobile PPC における認証方式は，Diffie-Hellman 鍵交換を利用し，エンドツーエンドで認証機能を実現する．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乱数を交換することにより，その乱数を盗聴されたとしても盗聴者には知ることのできない共有鍵を生成する鍵交換方式である．

Mobile PPC における認証方式では，通信に先立ち端末間でネゴシエーションを行う機構を追加し，その機構を用いて cookie の交換と Diffie-Hellman 鍵交換を行う．cookie 交換の目的は，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 攻撃を防止することである．これにより，不本意な Diffie-Hellman 鍵交換による端末の計算資源の浪費を防止することができる．通信に先立つネゴシエーションにより MN と CN に共有鍵を保持させておき，移動時にこの共有鍵を用いて MN の認証を行う．

Mobile PPC における認証方式の流れを図 4 に示す．通信

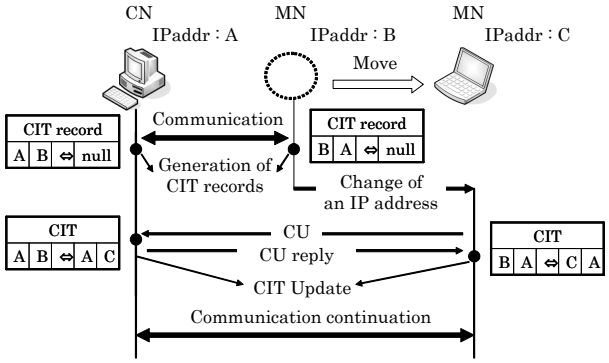


図 2　移動情報の通知

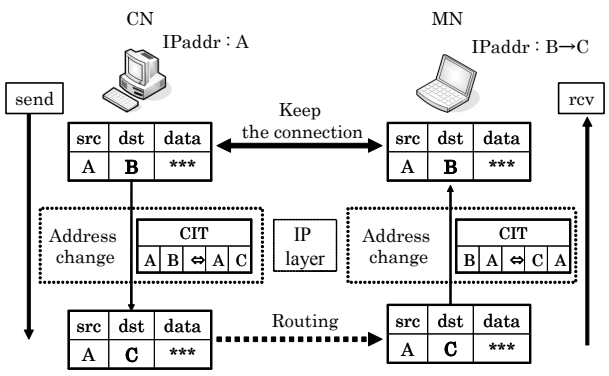


図 3　IP アドレス変換処理

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，CN から MN へ通信を開始した際の説明をする．通信に先立ち，端末間で cookie 交換を行なう．CN は cookie(cookie_cn)を生成し，MN へ送信する（①）．MN は cookie_cn を受信すると，自身の cookie(cookie_mn)を生成し，CN から受信した cookie_cn と自身が生成した cookie_mn を CN へ送信する(②)．CN は MN から cookie_cn と cookie_mn を受信すると，MN から受信した 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 MN 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．CN は priv_key(priv_key_cn)と呼ぶ乱数を生成し，その乱数から pub_key（pub_key_cn）と呼ぶ値を算出する．CN は MN へ pub_key_cn と cookie_mn を送信する（③）．MN は CN から pub_key _cn と cookie_mn を受信すると，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．MN は CN の簡易認証に成功すると MN は priv_key(priv_key_mn)を生成し，priv_key_mn から pub_key(pub_key_mn)を算出する．MN は CN へ pub_key_mn を返信する（④）．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（⑤）．

以上の通信に先立つネゴシエーションが完了した後，通常の TCP/IP 通信が行われる．

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)(MAC_mn)を作成し，CU パケットにこれを付加し CN へ送信する（⑥）．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（⑦）．CN は MN を認証すると，CU 応答パケットと共有鍵から MAC(MAC_cn)を作成し，CU 応答パケットにこれを付加し送信する（⑧）．ここで，MN は CN から CU の応答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（⑨）．

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

ここで，f()は一方向ハッシュ関数を表す．式(1) の msg は CU または CU 応答メッセージ全体を表す．key は Diffie-Hellman 鍵交換によって生成した共有鍵を示す．

3.3. Mobile PPC における認証方式の検証

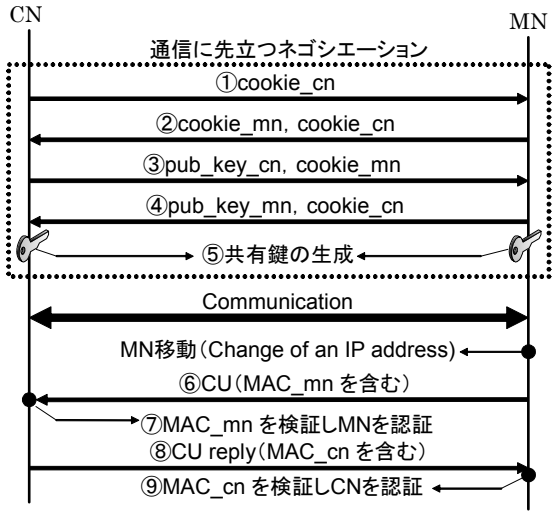


図 4　Mobile PPC における認証方式

共有鍵が CN と MN 間で安全に生成できているか検証する．攻撃者が MN に成りすまして CU を送信するには，MN と CN で生成される共有鍵を入手する必要があり，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．

攻撃者が Diffie-Hellman 鍵交換で交換される情報を入手するために使用する主な攻撃手法として，盗聴と成りすまし考えられる．始めに，盗聴について検証する．CN と MN の通信経路上に攻撃者がおり，攻撃者は CN から MN に開始された通信に先立つネゴシエーションを盗聴する．pub_key_cn や pub_key_mn は平文で通信路を流れているため盗聴することが可能だが，priv_key_mn や priv_key_cn は通信路を流れないため盗聴することができない．よってこの場合，共有鍵は安全に生成される．

次に，成りすましについて検証する．CN と MN の通信経路上に攻撃者がおり，CN から MN に対して開始される通信に先立つネゴシエーションを攻撃者が MN に成りすます．これにより，CN と攻撃者間で通信に先立つネゴシエーションが実行され，CN は攻撃者と共有鍵を生成することになる．また，攻撃者が CN と MN 間で行われる通信に先立つネゴシエーションの間に割り込み，両者が交換する pub_key を自分のものとすりかえた場合，CN と MN は攻撃者と共有鍵を生成することになる．これらの場合，通信に先立つネゴシエーションの後，CN と MN 間で通常の TCP/IP 通信が行なわれた際，攻撃者は CN に対して CU を送信することで CN と MN 間のセッションを乗っ取ることができる．

以上より，Mobile PPC における認証方式は，MN と CN 間の通信経路上における成りすましに対して脆弱性があるといえる．しかし，インターネット上の無制限な攻撃者からのコネクションの乗っ取りは防止することができることや PKI，IPsec，HA などの前提が必要ないので導入が非常に容易であるという利点がある．

4. 実装

Mobile PPC における認証方式を FreeBSD5.2.1 上で実装を行った．既存の Mobile PPC に下記モジュールを追加することにより Mobile PPC における認証方式を実現した．MAC の計算には HMAC_SHA1 を使用した．

4.1. NIT

Mobile PPC における認証方式では Diffie-Hellman 鍵交換を端末単位での通信に先立って実行するため，出力されるパケットが端末単位で 1 回目であるかどうかの判断を行なうための情報と共有鍵やそれに係わる 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する．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．

4.2. 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 層以外の部分には一切変更を加えない．

Mobile PPC を実現するモジュールは CIT 操作モジュール, IP アドレス変換モジュール, 移動管理モジュールの 3 つがあり，既に実装と評価を終えている．Mobile PPC における認証方式を実現するために Mobile PPC に追加するモジュールは, Diffie-Hellman 鍵交換モジュール, NIT 操作モジュール，認証モジュールの 3 つである．Diffie-Helllman 鍵交換モジュールは, 通信に先立ち cookie の交換および Diffie-Hellman 鍵交換を実行する．NIT 操作モジュールは，NIT レコードの検索・生成・更新を実行する．また，NIT の状態を監視し，無通信状態にある NIT レコードを削除する. 認証モジュールは，CU および CU 応答パケットに対して MAC の生成・付加・検証を行う．

5. 評価

本章では，試作システムの性能を測定し，通信に先立つネゴシエーションの処理時間と移動通知処理時間に関するオーバヘッドを測定した．また，Return Routability との定性的な比較を行なった．

5.1. 実験環境

通信に先立つネゴシエーションの処理時間と CU パケットおよび CU 応答パケットの処理時間を図 5 に示す測定環境で測定した．MN と CN の OS は FreeBSD5.2.1 を採用した．MN の CPU は Celeron 2GHz，メモリが 256MB で IEEE802.11b で実験ネットワークに接続した．CN の CPU は Pentium2.4GHz，メモリが 256MB，NIC は 100BASE-T である．

5.2. 通信に先立つネゴシエーションの処理時間

通信に先立つネゴシエーションの処理時間の測定結果は，2.92 s であった．このうち，DH 鍵交換に係わる鍵演算の時間は 2.89 s で，全体の 99%を占めている．この時間を短縮するには Diffie-Hellman 鍵交換の演算に FFT(高速フーリエ変換)[13]を使用する方法が考えられる．

5.3. 移動通知処理時間

CU パケットおよび CU 応答パケットの処理時間の測定結果は，0.33 ms であった．同一条件下において，認証のない Mobile PPC の CU および CU 応答の処理時間の測定結果は 0.29 ms であった．これらの測定結果より，処理時間の差は 0.04 ms であり，CU および CU 応答に追加された認証処理によるオーバヘッドは無視できる．

5.4. Return Roubatility との比較

始めに，セキュリティ面での比較を行なう．Return Routability は CN と HA 間と 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．Mobile PPC における認証方式は CN と MN 間の通信経路上での成りすましに対して脆弱性がある．両方式とも脆弱性が存在するが，インターネット上の無制限な攻撃者からのコネクションの乗っ取りは防止することができる．このため，両方式とも認証の効果は大きいといえる．

次に，運用面での比較を行なう．本提案方式は，Return Routablity と比較して，HA のような特殊な第三の装置や設定の複雑な IPsec が必要ないため，導入が容易である．

6. むすび

エンドツーエンドで認証機能を実現する Mobile PPC における認証方式を提案した．本提案方式は，インターネット上の無制限な攻撃者からのコネクションの乗っ取りは防止することができるという点で有効である．運用面においては，

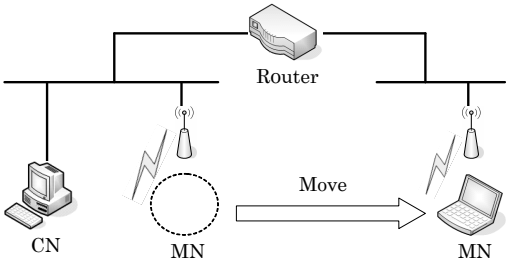


図 5 実験環境

Return Routability と比較して HA のような特殊な第三の装置や設定の複雑な IPsec が必要ないため, 導入が容易である．また，測定評価の結果，移動通知処理における認証処理で発生するオーバヘッドは小さいことが分かった．しかし，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大きいことが分かった．今後は，Diffie-Hellman 鍵交換の演算に FFT を使用することで，通信に先立つネゴシエーション時間の短縮を行う．また，CN が移動端末である場合の検証も行なう．

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-D-I, No.3, pp.308-328, March.2004.

[2] C. E. Perkins, “IP Mobility Support for IPv4,” RFC 3344, Aug. 2002.

[3] D. Johnson, C. Perkins, J. Arkko, “Mobility Support in IPv6,” RFC3775, June. 2004.

[4] M. Kunishi, M. Ishiyama, K. Uehara, H. Esaki, and F. Teraoka, “LIN6: A new approach to mobility support in IPv6, ” Third International Symposium on Wireless Personal Multimedia Communications, pp.1079-1084, Nov. 2000.

[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, Feb.2002.

[6] 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 の実装と評価, ” DICOMO2005 シンポジウム論文集, Vol.2005, No.6, pp.125-128, Jul.2005.

[7] R. Droms, “Dynamic Host Configuration Protocol,” RFC2131, March 1997.

[8] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, “Dynamic Updates in the Domain Name System,” RFC 2136, April 1997.

[9] 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, May.2004.

[10] 瀬下正樹，竹内元規，渡邊晃，“Mobile PPC における移動端末の認証, ” DICOMO2005 シンポジウム論文集, Vol.2005，No.6，pp129-132，Jul.2005.

[11] W.Diffie, M.E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp.644-654, Nov.1976.

[12] S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol,” RFC 2401, 1998.

[13] J.W.Cooley, J.W.Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” Math of Comput,Vol.9,pp.297-301,1965.

Mobile PPCにおける認証方式の実装

名城大学大学院　理工学研究科

瀬下正樹　渡邊晃

研究背景

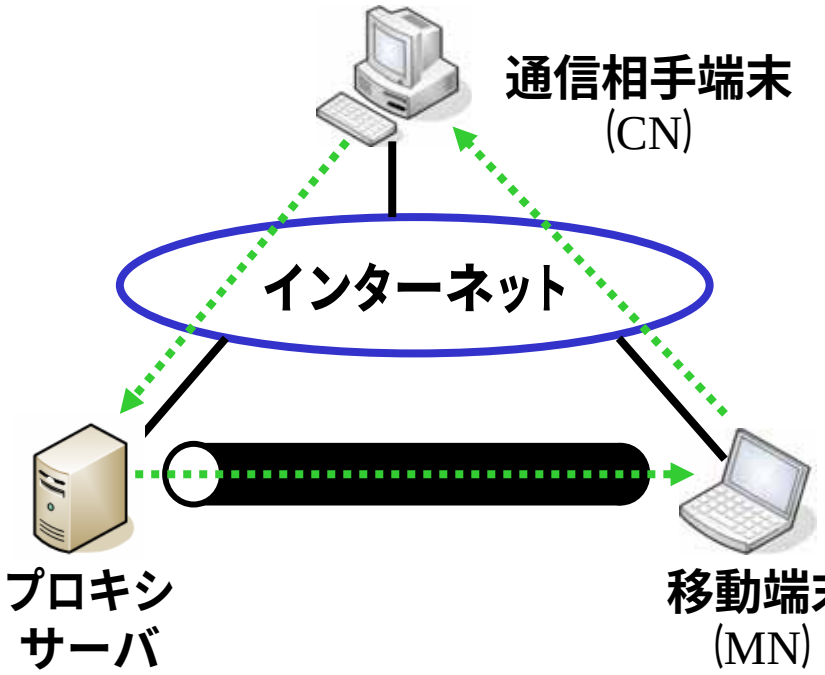
- **研究背景**
  - **モバイル端末の普及**
  - **無線ネットワーク環境の普及**

  ⇒端末が自由に移動しながらネットワークに接続するというニーズが増加
- **目的**
  - **移動中にIPアドレスが変化しても通信を継続する**

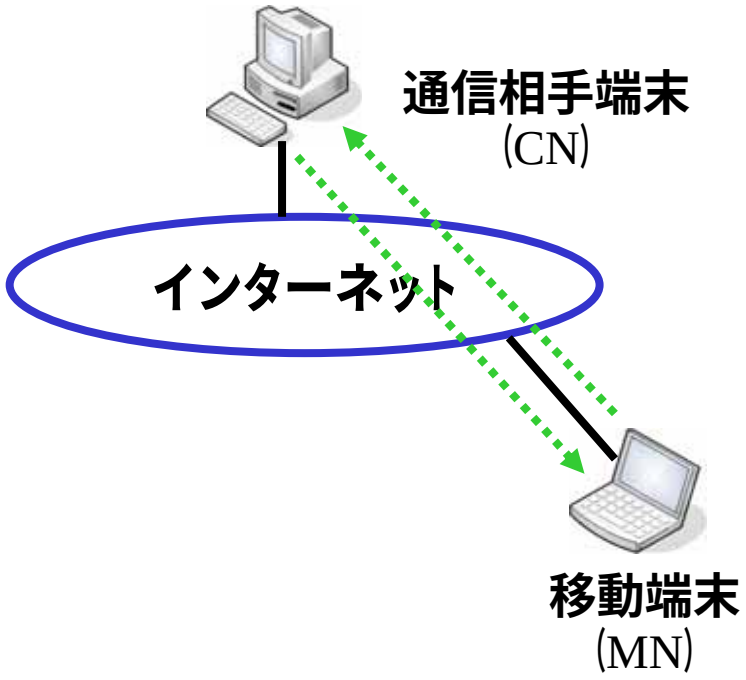
移動透過性の実現

移動透過性保証プロトコル

- **プロキシ方式**
  - **通信相手端末 (CN) からのパケットをプロキシサーバが中継し，移動端末 (MN) へパケットを転送**



- **エンドツーエンド方式**
  - **プロキシサーバを用いず，エンド端末間により移動透過な通信を行う**

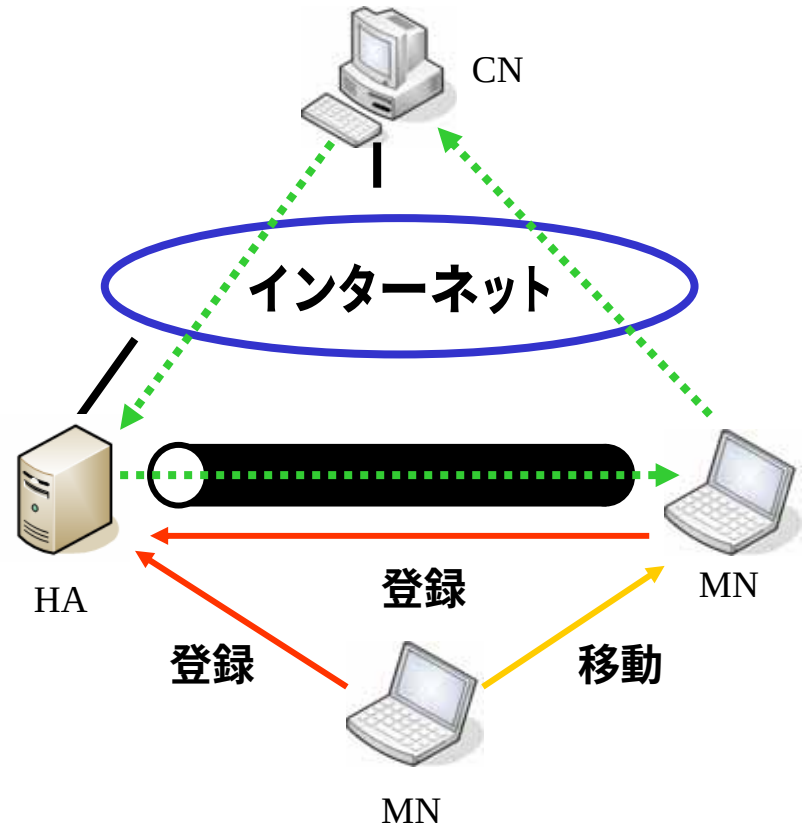


既存技術　Mobile IPv4

- Mobile IPv4
  - **プロキシ方式**
    - **プロキシサーバ**:
      HA (Home Agent)

- **動作概要**
  - MN**は現在のアドレスを**HA **へ登録**
  - CN**から**MN**宛のパケットは**HA**が代理受し，**HA**はパケットを**MN**へ転送**
  - MN**から**CN**への通信は直接行う**

- **課題**
  - **通信経路が三角経路**
  - HA**と**MN**間はトンネル転送**
  - **特殊な装置**(HA)**が必要**

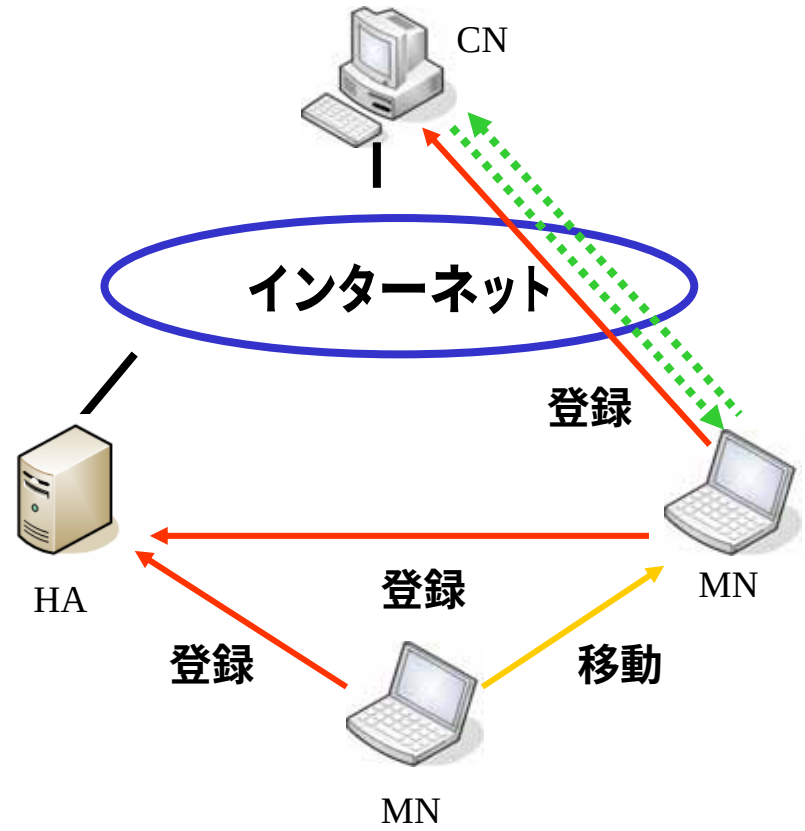


既存技術　Mobile IPv6

- Mobile IPv6
  - **プロキシ方式からエンドツーエンド方式に遷移可能**

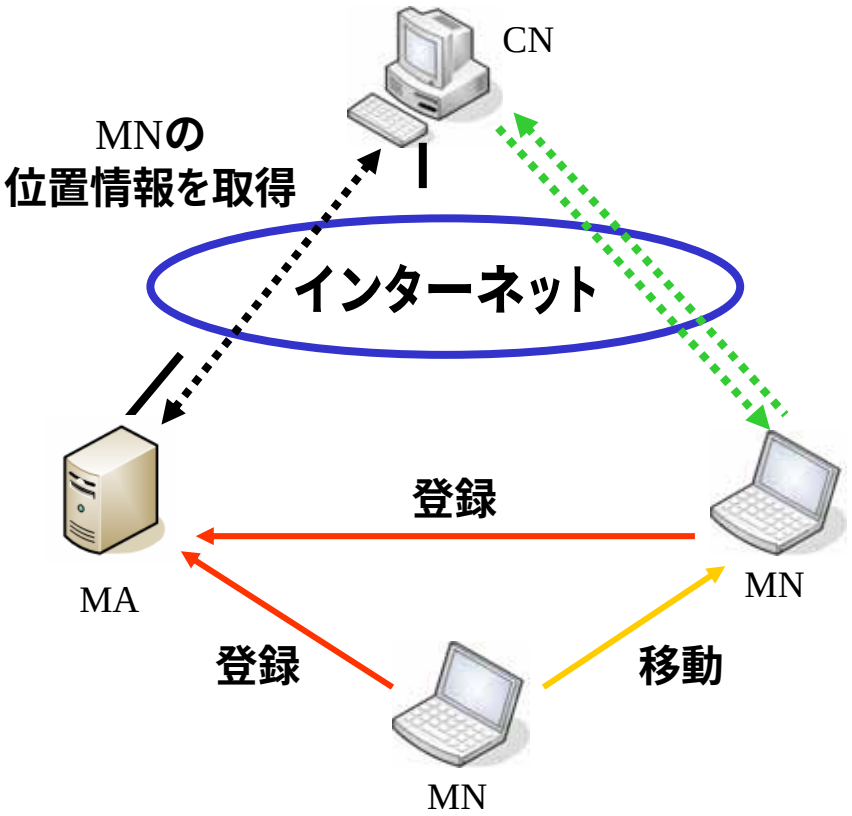
- **動作概要**
  - **基本的な動作は**Mobile IPv4**と同じ**
  - **経路最適化機能**
    - CN**とMNが直接通信**

- **課題**
  - **特殊な装置**(HA)**が必要**
  - IPv6**拡張ヘッダを使う**



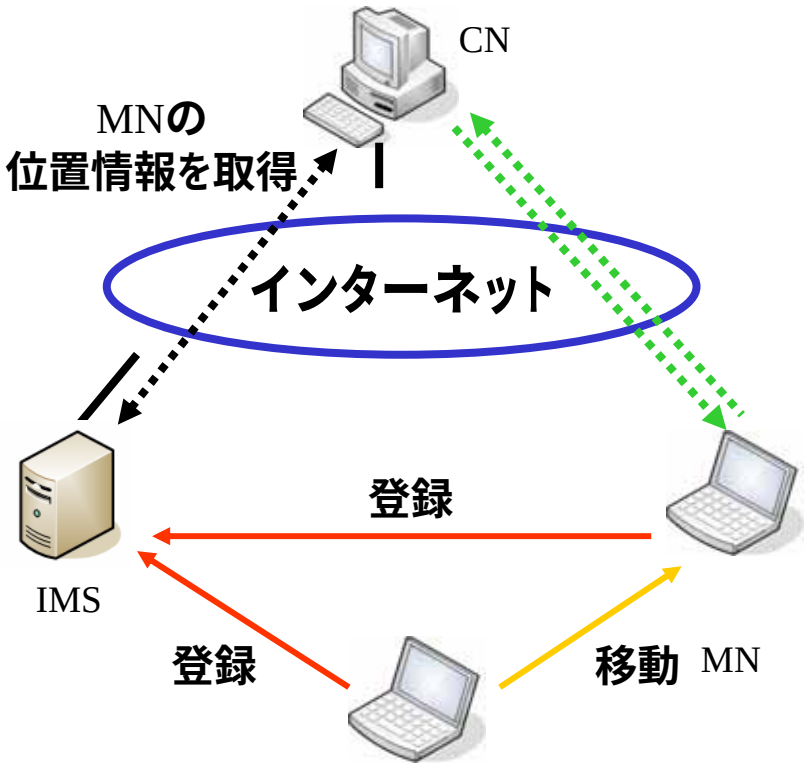
既存技術　LIN6

- LIN6
  - **エンドツーエンド方式**
- **仕組み**
  - **縮退アドレスモデル**
    - IPv6**アドレス空間を位置指示子とノード識別子に分離**
  - MA(Mapping Agent)**でMNの位置指示子とノード識別子の対応関係を管理**
- **課題**
  - **アドレス体系の整備が必要**
  - **独自の位置管理装置**(MA)**の導入**



既存技術　MAT (Mobile IP with Address Translation)

- MAT
  - **エンドツーエンド方式**
- **仕組み**
  - LIN6**と同じ設計思想だがアドレス空間の分割はしない**
  - IMS(IP address Mapping Server)**でMNの位置指示子とノード識別子の対応関係を管理**
- **利点**
  - **既存のアドレス体系が使える**
- **課題**
  - **位置管理装置**(IMS)**の導入**
  - DNS**の改造**



- 既存技術のまとめ
  - プロキシ方式
    - 柔軟性やリアルタイム性が失われる
  - 既存のエンドツーエンド方式
    - 特殊な位置管理サーバが必要

    ⇒普及に至るまで機能が発揮できない

    ⇒サーバの二重化の対策が必須，管理負荷が大きい
- 今後のユビキタス社会
  - ネットワークを最大限に活かせるP2P通信の要求
    - P2P通信は個人間の通信が主体

    ⇒エンドツーエンド方式かつ，特殊な位置管理サーバを必要としない方式が望まれる．
  - 実装レイヤはTCP/IPの区別なく利用可能なネットワークレイヤでの実現方法が有利

Mobile PPC (Mobile Peer to Peer Communication)の提案

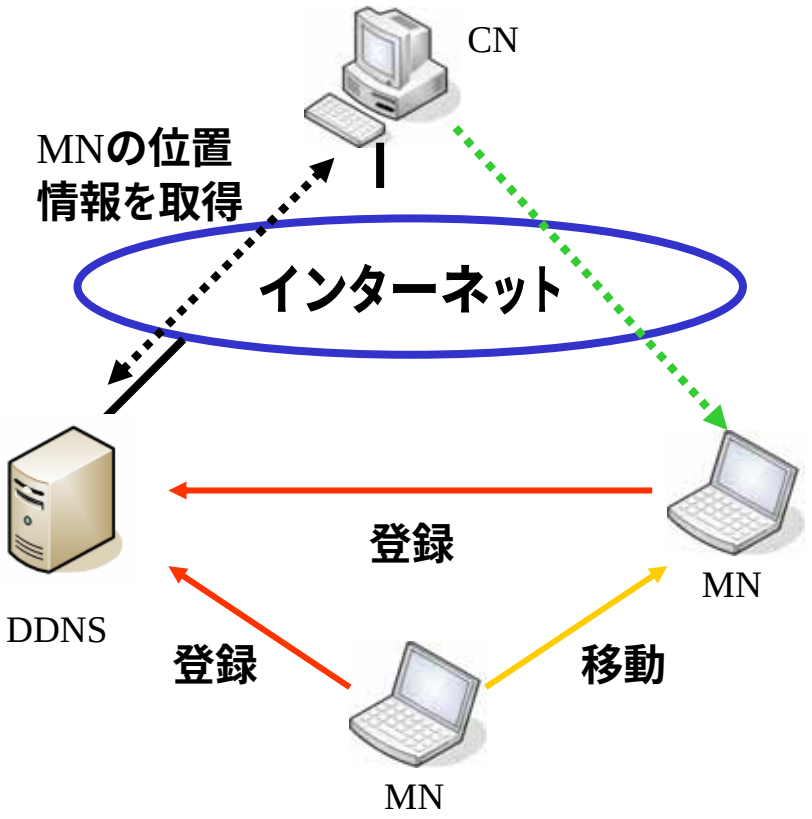
Mobile PPC (Mobile Peer to Peer Communication)

- **移動透過性を実現する機能**
  - **通信開始時において相手のIPアドレスを知る方法**
    - **“初期IPアドレスの解決”と呼ぶ**
  - **通信中にIPアドレスが変わった場合に変更時のIPアドレスを知る方法**
    - **“継続IPアドレスの解決”と呼ぶ**
- **Mobile PPCの設計方針**
  - **初期IPアドレスの解決と継続IPアドレスの解決を実現する仕組みを明確に分ける**

Mobile PPC (Mobile Peer to Peer Communication) 独自技術

- Mobile PPC
  - エンドツーエンド方式
  - 両端末間においてアドレス変換処理を実行
- 初期IPアドレスの解決
  - DDNS (Dynamic DNS)
    - ホスト名とIPアドレスを動的に管理

⇒ホスト名を識別子としてMNへ通信を開始

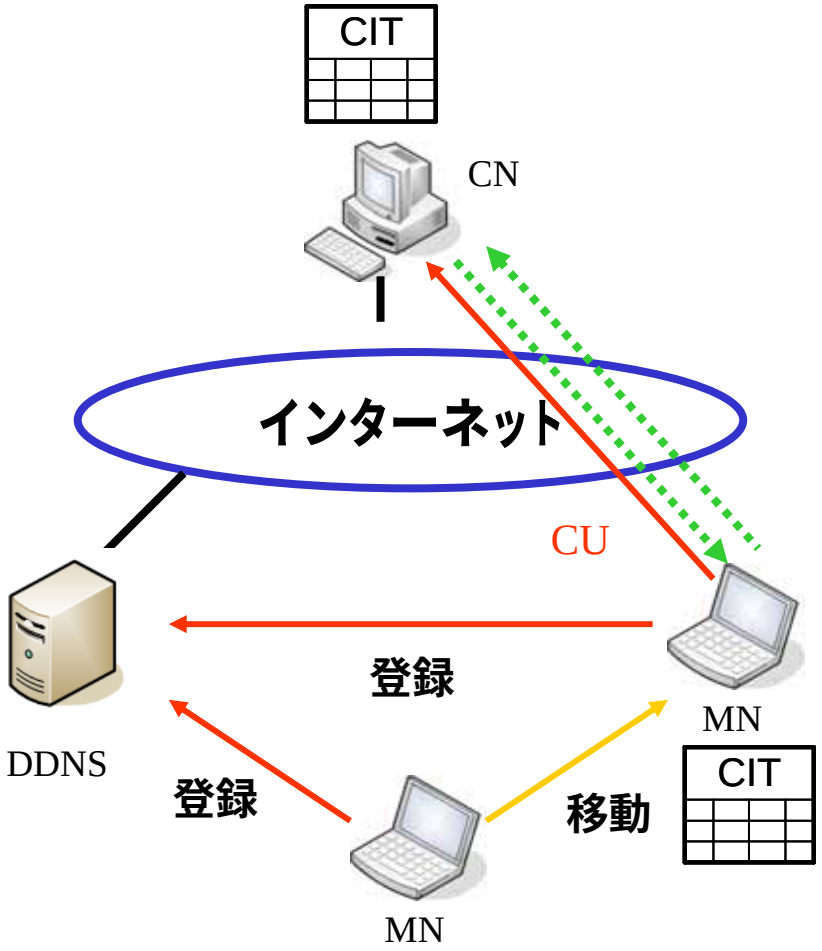


継続IPアドレスの解決

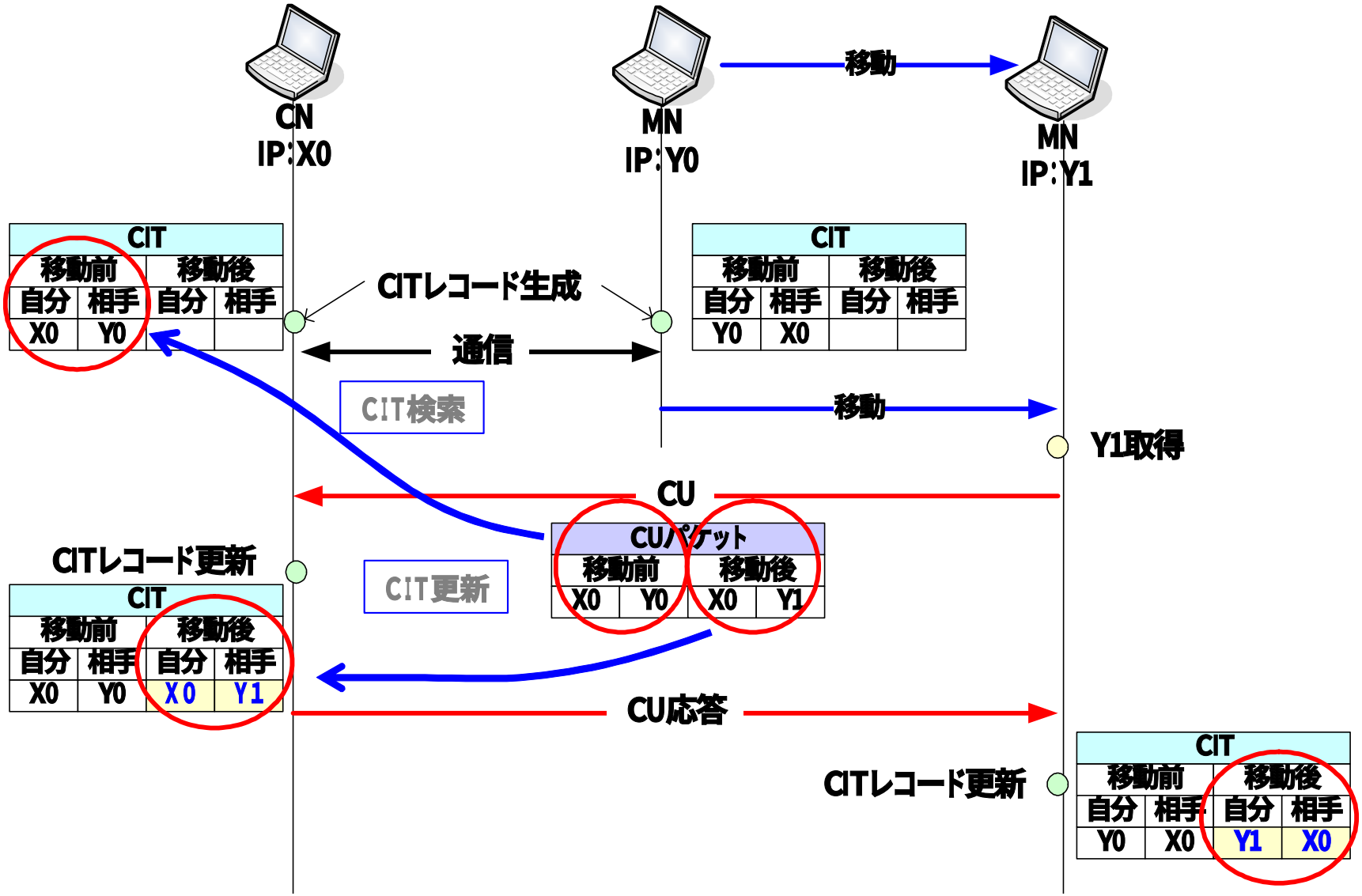
- IP層：Mobile PPC機能を追加
  - 端末はCIT（Connection ID Table）を保持

    CIT：移動前後の通信の識別情報（IPアドレスなど）の対応関係を示すテーブル
  - IPアドレスが変化するとCU (CIT UPDATE)パケットによりP2Pで移動情報を通知

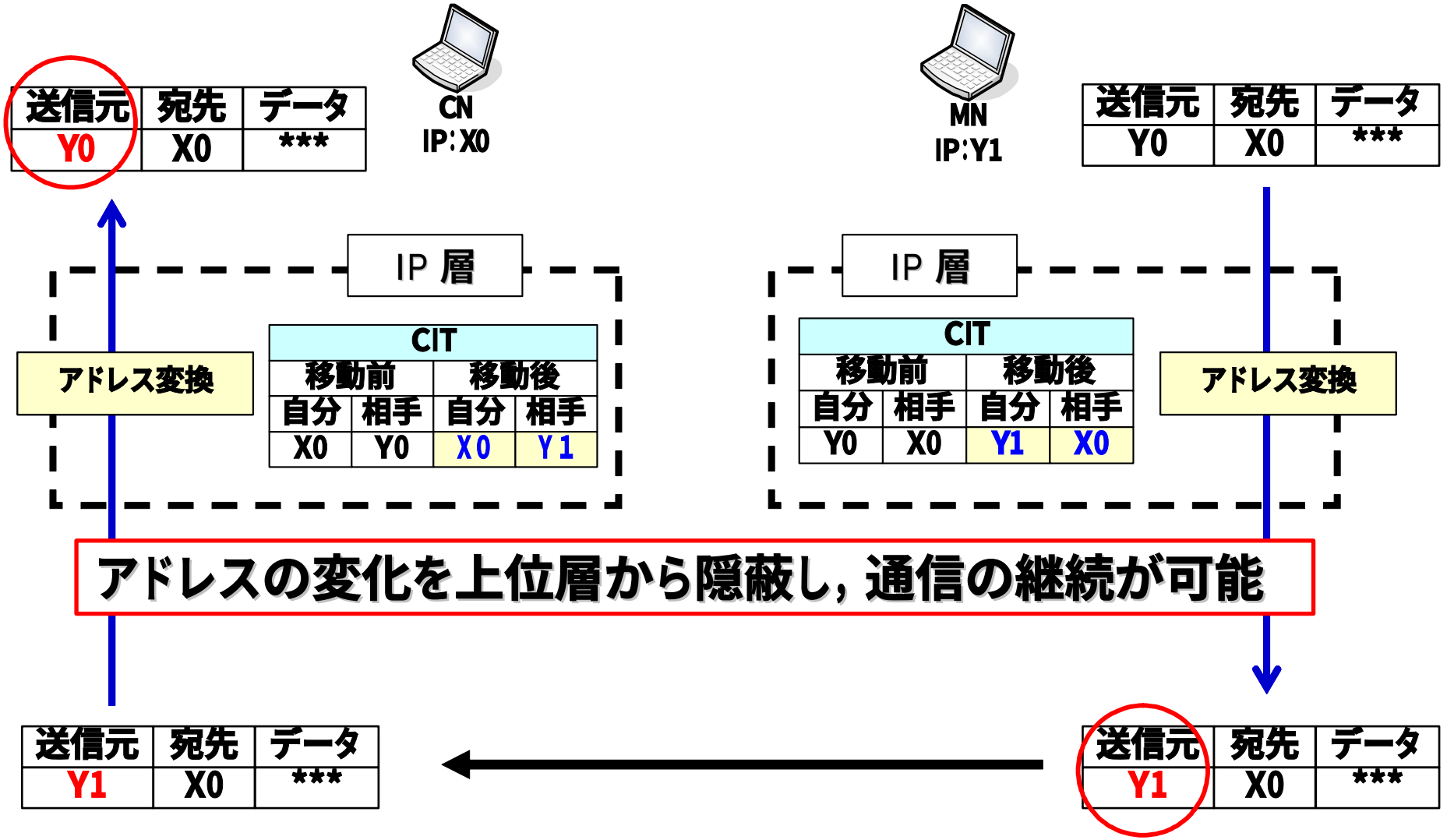
    ⇒CITの情報の更新
  - エンド端末で通信パケットに対し，CITに基づいたアドレス変換処理を行う



Mobile PPCによる通信［1］



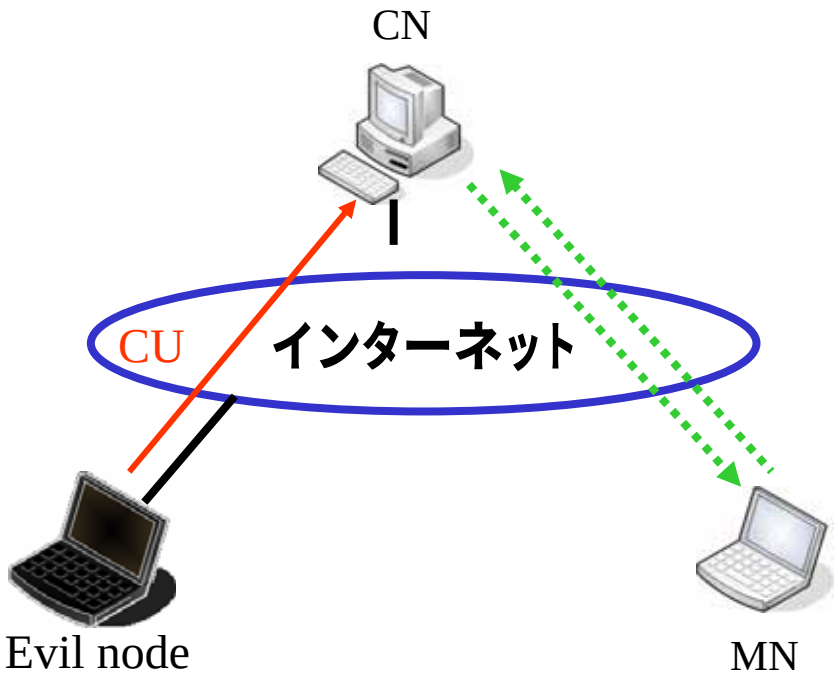
Mobile PPCによる通信［2］



Mobile PPCの課題

- MNとCNが通信中
  - 悪意を持ったノードがCNへCUパケット送信
    - CITが不正に書き換えられる

⇒ **通信の乗っ取りが発生**



インターネットにおける認証

- **インターネットにおける端末間の認証**
  - A) **共通鍵暗号を利用する方式**
  - B) **公開鍵暗号を利用する方式**

A) **共通鍵暗号を利用する方式**

– **認証したい相手端末と共有鍵を事前に設定が必要**

⇒ CN と通信するMN は任意であるため難しい.

B) **公開鍵暗号を利用した認証**

– PKI の仕組みを適用することで認証が可能

– **現在のPKI が未整備である状況を考慮すると非現実的**

- Mobile PPC **における端末間の認証**

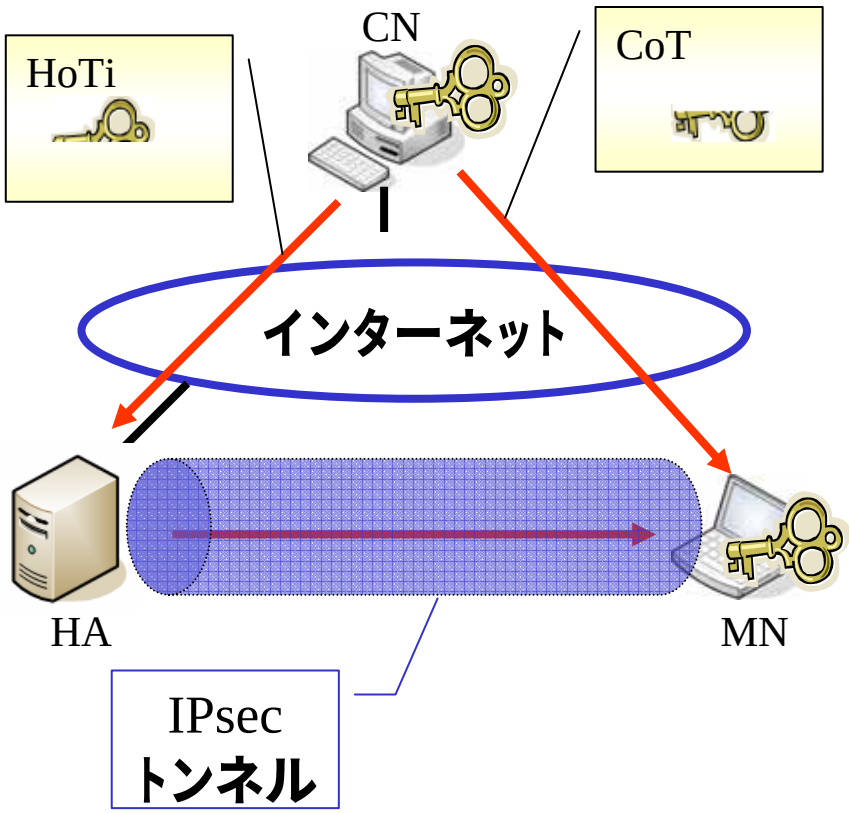
–CN とMN 間において認証に使用する鍵を，いつ，どのようにして安全に交換するかが解決すべき課題

従来技術：Return Routability

- Return Routability
  - Mobile IPv6の経路最適化時における認証機構
  - 共通鍵暗号方式による認証

  ※特殊な装置(HA)を利用
- 前提条件
  - HAとMNは信頼関係にある

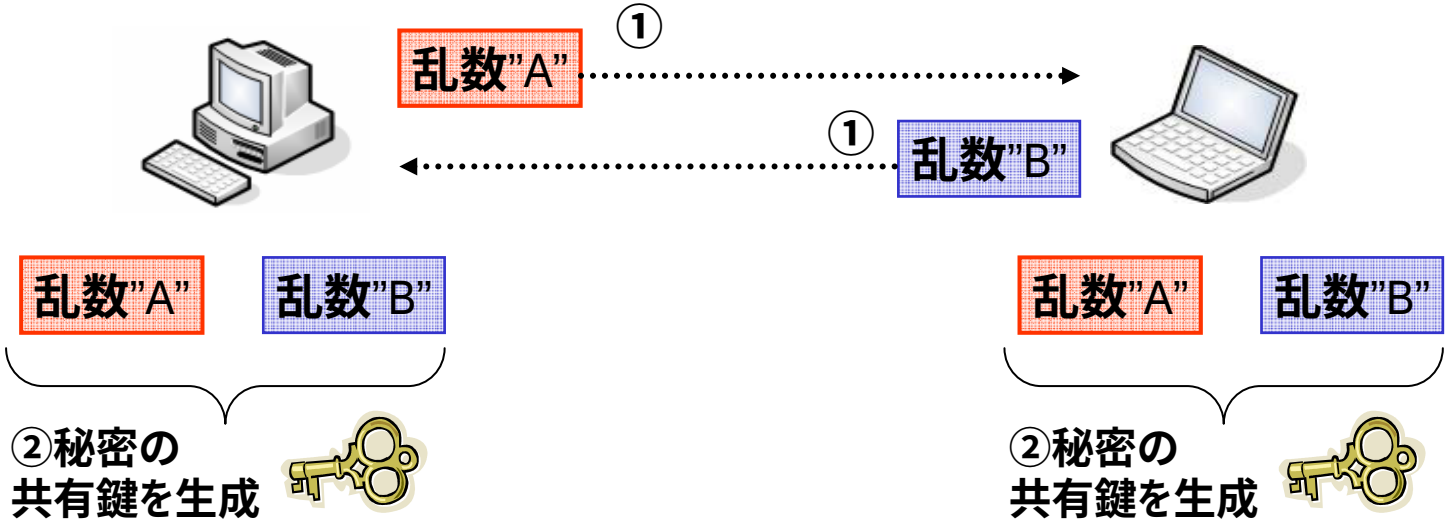
  ⇒ HAとMN間はIPsecで保護
- 共有鍵生成方法
  - 共有鍵を二つに分け，異なる経路から配送することで安全に共有鍵を生成



Mobile PPCのようにエンドエンドで移動透過性を保証するプロトコルには適していない

Mobile PPCにおける認証方式

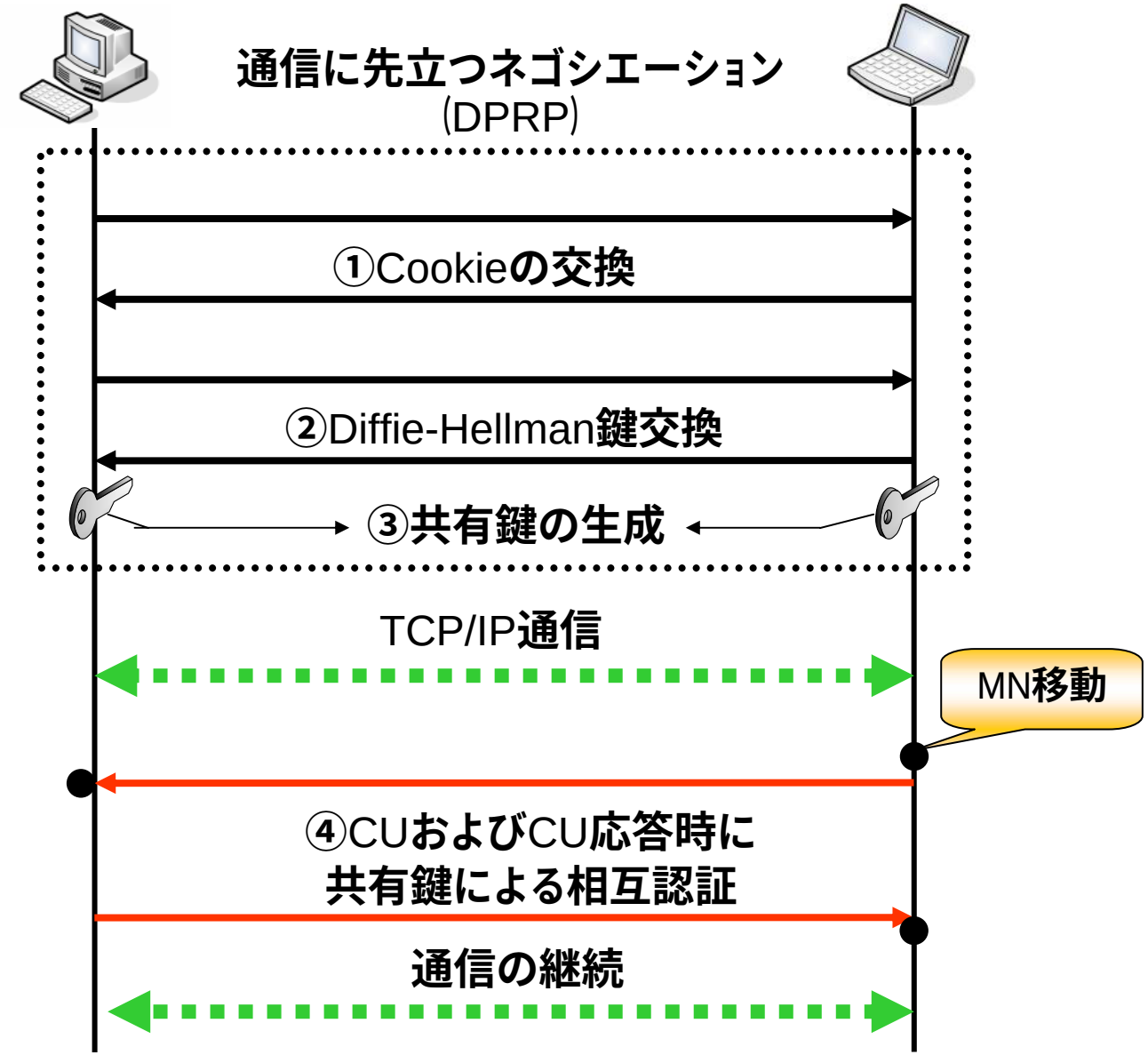
- **Mobile PPCにおける認証方式**
  - Diffie-Hellman**鍵交換を利用する**
    - **ある乱数を交換することによって(①)**
    - **盗聴者がいても端末間で共有鍵を安全に生成する(②)**
    - ✧**<u>離散対数問題を利用</u>**



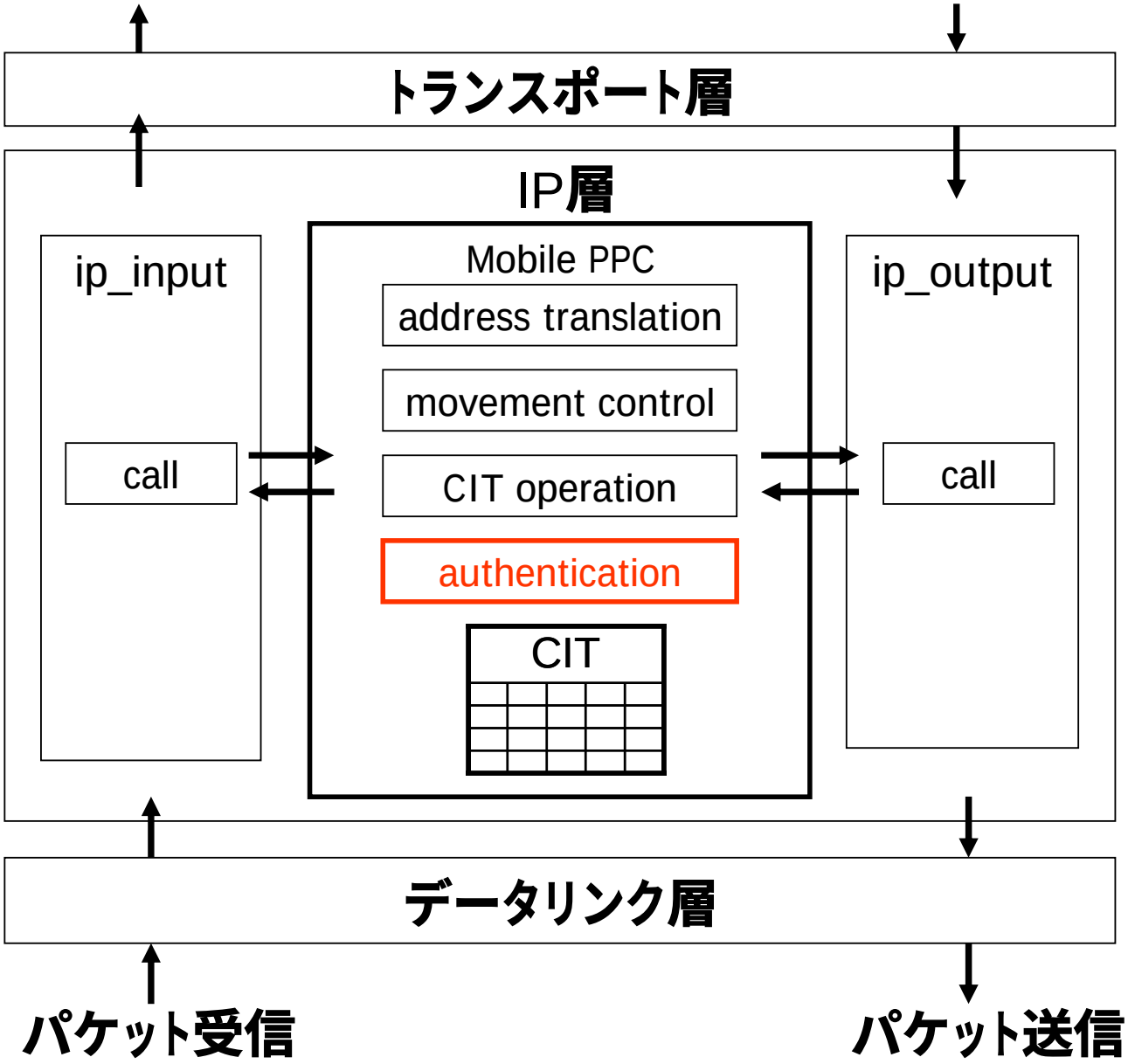
Mobile PPCにおける認証方式

1. **端末間の通信に先立ち**
   - Cookie**交換**
     - **通信相手端末が**Mobile PPC**実装端末であるかの判別**
     - **送信元**IP**アドレスを偽造した成りすましによる**DOS**攻撃の防止**
   - Diffie-Hellman**鍵交換**
     - **秘密共有鍵の作成**
   - ✓ DPRP (Dynamic Process Resolution Protocol) **を利用**
     - **端末間の通信に先立ちネゴシエーションを実行**
2. MN**移動時**
   - **通信に先立ち生成した共有鍵による認証**

Mobile PPCにおける認証方式のシーケンス

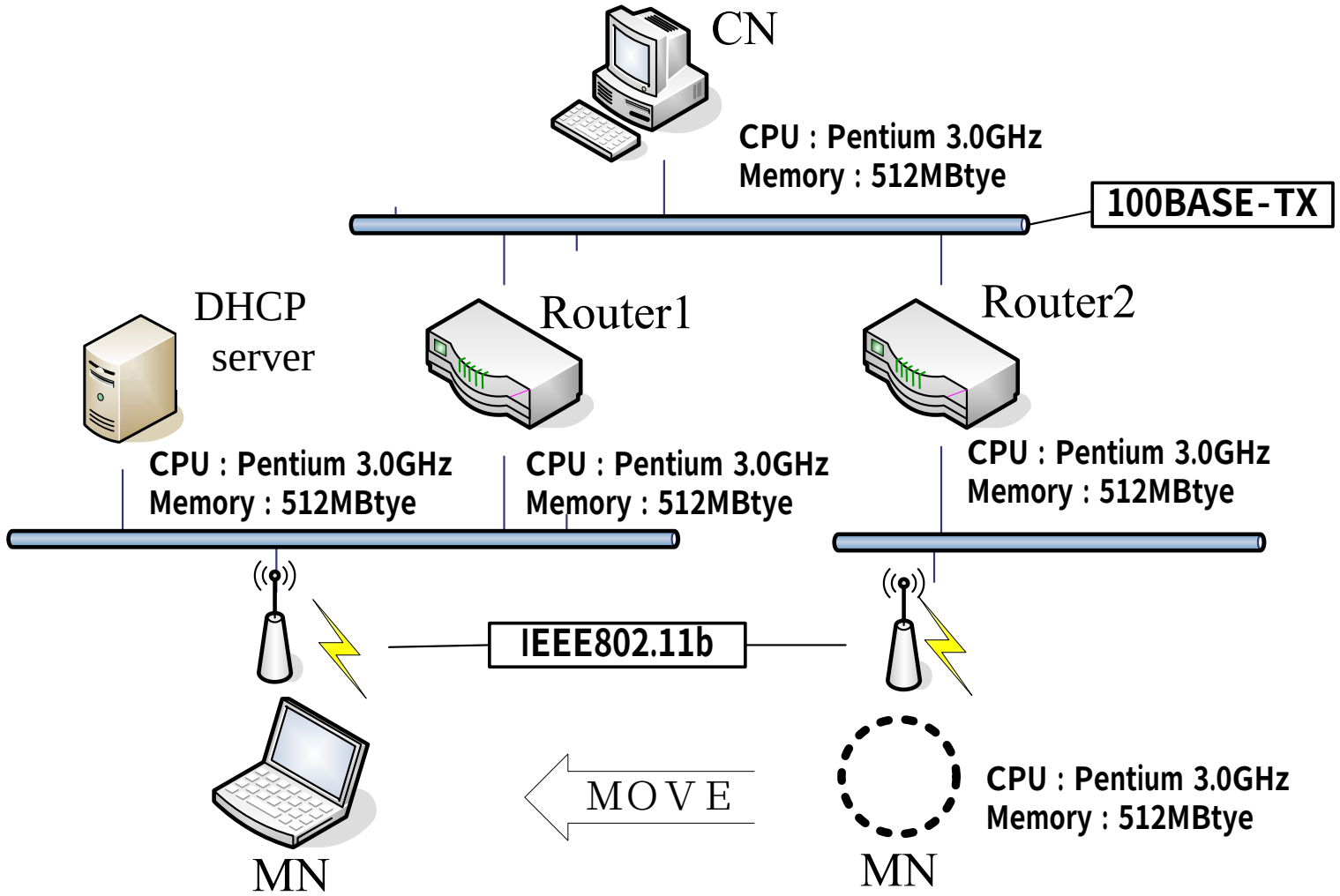


実装　モジュール構成


トランスポート層
IP層
ip_input
Mobile PPC
address translation
movement control
call
CIT operation
call
ip_output
authentication
CIT
データリンク層
パケット受信
パケット送信

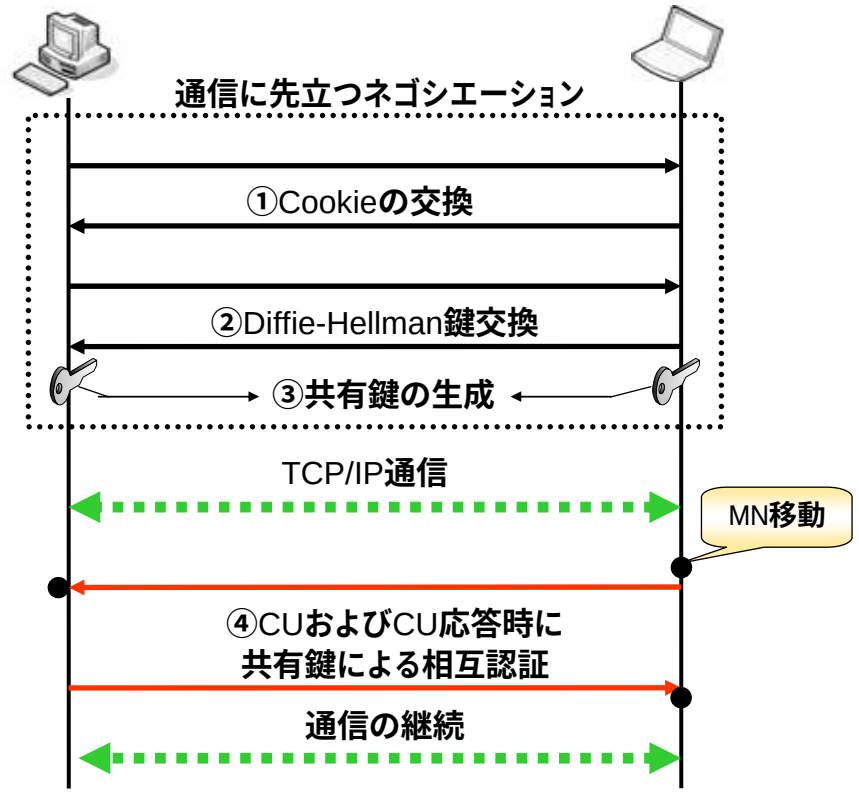
評価： 実機環境

➢ 実機環境


CN
CPU : Pentium 3.0GHz
Memory : 512MBtye
100BASE-TX
DHCP
server
Router1
Router2
CPU : Pentium 3.0GHz
Memory : 512MBtye
CPU : Pentium 3.0GHz
Memory : 512MBtye
CPU : Pentium 3.0GHz
Memory : 512MBtye
IEEE802.11b
MOVE
CPU : Pentium 3.0GHz
Memory : 512MBtye
MN
MN

評価：性能測定

- **通信に先立つネゴシエーションの処理時間**
  - 2.92[s]
    - DH**鍵交換の演算時間は**2.89[s]**で全体の**99%
- **移動通知処理時間（CU**および**CU**応答パケットの処理時間）
  - 0.33[ms]
    - **認証処理のない場合**：0.29[ms]
    - **増加した時間**：0.04[ms]



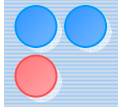
Return Routabilityとの比較

- **セキュリティ面での比較**

	Return Routability	Mobile PPCにおける認証方式
簡易認証	○	○
盗聴	△ CNとHA間とCNとMN間 特にCN近傍	○
中間者攻撃	○	△ CNとMN間

両方式とも脆弱性が存在
しかし，インターネット上の無制限な攻撃者からのコネクションの乗っ取りは防止は可能

⇒導入の効果は大きい



むすび

- Mobile PPCの概要と課題
- Mobile PPCにおける認証方式の提案
  - 特殊な第三の装置が不要で導入が容易
- 性能測定
  - 通信に先立つネゴシエーション
    - オーバーヘッドは大きい
- 今後
  - 通信に先立つネゴシエーション処理時間の短縮

おわり