Mobile PPC における移動端末の認証

瀬下　正樹[†]　竹内　元規[†]　渡邊　晃[†]
[†]名城大学大学院理工学研究科

Authentication Mechanisms in Mobile PPC

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

1. はじめに

ノートパソコンや PDA(Personal Digital Assistance)などのモバイル端末の普及と，無線ネットワーク環境の広がりにより，端末が自由に移動しながらインターネットに接続するという利用形態が増えつつある．そのような状況下では，端末が移動しても通信を継続することが要求されるが，移動に伴い IP アドレスが変化するため，この要求を満たすことが難しい．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行われている[1].

移動透過性は大きくプロキシ方式とエンドツーエンド方式に分類できる．プロキシ方式は通信相手ノード（以下 CN）からのパケットをプロキシサーバが中継し，移動ノード（以下 MN）へパケットを転送する．エンドツーエンド方式は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．

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．

Mobile IP は，プロキシ方式であり，MN の位置を管理する Home Agent(以下 HA)が導入され，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．CN から MN への通信は直接行われる．Mobile IP は，HA という特殊な装置が必須であり導入するための敷居が高いということと，通信経路の冗長，ヘッダの追加によるオーバーヘッド，HA による一点障害などの問題点が指摘されている．

LIN6 は，エンドツーエンド方式であり，LIN6 ID と呼ばれるノード識別子と IP アドレスのネットワークプリフィクスの対応関係を保持する Mapping Agent(以下 MA)と呼ばれる位置管理サーバを設け，ノード識別子と位置指示子の機能を分離させることで, IP アドレス変化時の問題を解決している．LIN6 は，ノード識別子にグローバルユニークなアドレスを使用するためアドレス体系の整備が必要である．また，IPv6 アドレスを前提としているため IPv4 には適用できない.

Mobile IPv6 は，プロキシ方式からエンドツーエンド方式へ遷移可能な方式である．プロキシ方式である Mobile IP の考え方に基づいて設計されているが，MN が新しく取得した IP アドレスを直接 CN へ通知することができるため, エンドツーエンドでの通信も可能となっている．しかし，通信開始時には HA を経由するルーティングを行うため，HA が必須となっている．

著者らは，移動透過性保証プロトコルで発生するこれらの問題を解決するために, P2P 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 IP アドレスが変化しても通信を継続できる方法(継続 IP アドレスの解決)を異なるアプローチによって解決する．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 Dynamic DNS(DDNS)[9],[10]を利用する．これにより，ホスト名を識別子として通信開始時における端末の IP アドレスを知ることが可能となる. DDNS は DNS の延長であり，既に実用化されている技術である．一方，継続 IP アドレスの解決には，エンドツーエンド方式である Mobile PPC を用いる．Mobile PPC では，MN の IP アドレスが変化すると，その直後に MN から通信中の CN に対して，移動後の IP アドレスを Binding Update(以下 BU)により通知する．BU により，エンド端末間では新旧 IP アドレスの対応関係を示すテーブルが作成され, 以後の通信ではパケット送受信時に IP 層でこのテーブルを参照してアドレス変換を行う．これにより，TCP/IP プロトコルスイートを含む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 ではプロキシサーバを使用しないため，通信経路の冗長や一点障害などの問題がない．また，特殊なアドレス体系を必要とせず, 原理的に IPv4 と IPv6 のどちらにも適用可能な方式である．

しかし，Mobile PPC において，CN が BU パケットを受信する際，セキュリティの観点から MN を確実に認証する必要があるが，これまでの Mobile PPC には認証機構が定義されていなかった．

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 の Return Routability と，それと同様の手法を LIN6 に適用した方式[7]が提案されている．しかし Return Routability では HA , LIN6 では MA のような第三の機器を利用するため, 特殊なネットワーク機器を使用しない Mobile PPC において，これらの認証機構は適用することができない．そこで本稿ではエンド端末間のみで実現可能な Mobile PPC に適した認証方式の提案を行う．なお，本研究は移動透過機をシステムに導入することにより新たに発生する脅威として通信の切替時における乗っ取りを防ぐことを目的としている．従って, 通信開始時における相手認証は検討の対象外である．

以下, 2 章で既存技術の代表として Mobile IPv6 と Return Routability，3 章で Mobile PPC とその認証方式，4 章で実装， 5 章で評価，6 章でむすびについて述べる．

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．初期 IP アドレスの解決にはプロキシ方式である Mobile IP と同様に 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ことによって移動透過性を保証する．一方，継続 IP アドレスの解決には，基本仕様として新たに追加された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 にホームアドレスと気付けアドレスの対応関係を示すテーブル（Binding Cache ; 以下 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 へ新しい IP アドレスを登録し，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ことによりエンドツーエンドでの移動透過性を保証する．CN が BC を保持していないときは初期 IP アドレスの解決と同様に HA を利用した移動透過な通信が行われる．

セキュリティの観点から，経路最適化時において，新しい IP アドレスを登録する際，CN は MN を確実に認証する必要がある．Mobile IPv6 では，この際に次に述べる 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 で導入されている認証機構である．Return Routability は MN と HA 間の信頼関係を利用することと，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とにより MN と CN 間で共有鍵を登録直前に保持させ，登録時にこの共有鍵を使用した認証を行う．Mobile IPv6 では，CN が BC を保持している際に，通信開始時において CN から送信されたパケットが HA を経由して MN へ送られた場合と，MN が移動し IP アドレスが変化した場合に MN から CN へ登録パケットが送信され，そのたびに Return Routability が実行される．なお，Return Routability は経路最適化機能を使用することにより発生する通信の乗っ取りを防ぐことを目的としており，通信開始時における認証は検討の対象外である．

Return Routability の動作を図 2 示す．ここで，MN と HA 間は信頼関係を期待できるため事前に共有鍵を保持させることが可能であると考え，この区間では IPsec[8]を使用し安全な通信路での通信が行われる． MN は CN へ Return Routabhility を開始する Home Test Init (以下 HoTI)（①）および Care-of Test Init (以下 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(以下 HoT)(③)および Care-of Test(以下 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

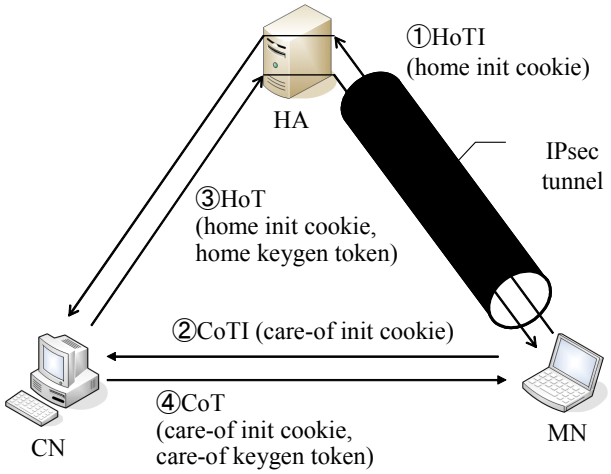


図 1　Return Routability の動作

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 はこの共有鍵から認証データを作成し，これを登録パケットに付加して CN へ送信する．CN は自身が生成した home keygen token と care -of keygen token により共有鍵を作成し，登録パケットに付加された認証データの検証を行い MN の認証を行う．

2.2.2. Return Routability の課題

Return Routability は HA のような特殊なネットワーク機器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使用できない．また，Return Routability では Init cookie の組で MN は CN の認証を行い，keygen token の組で CN は MN の認証を行っているため，Init Cookie の組 や keygen token の組 を攻撃者が得ることができれば CN や MN への成りすましが可能となる．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ため Return Routability は CN 近傍の攻撃者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 はエンド端末以外の特殊なネットワーク機器を不要とし, P2P で移動透過性を実現するプロトコルである. Mobile PPC では初期 IP アドレスの解決と継続 IP アドレスの解決を異なるアプローチによって解決しており，後者が Mobile PPC 特有の機能である. 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 Dynamic DNS(DDNS)[9],[10]を利用する．これにより，ホスト名を識別子として通信開始時における端末の IP アドレスを知ることが可能となる．一方，継続 IP アドレスの解決には，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通信の識別情報を図 2 のように Binding UPDATE(以下 BU)により通知する．エンド端末には新旧 IP アドレスの対応関係を示すテーブル（Connection ID Table ; 以下 CIT）を保持させ，BU により CIT を更新し, 以後の通信では図 3 のようにパケット送受信時にネットワーク層でこの CIT を参照してアドレス変換を行う．これにより， TCP/IP プロトコルスイートを含む上

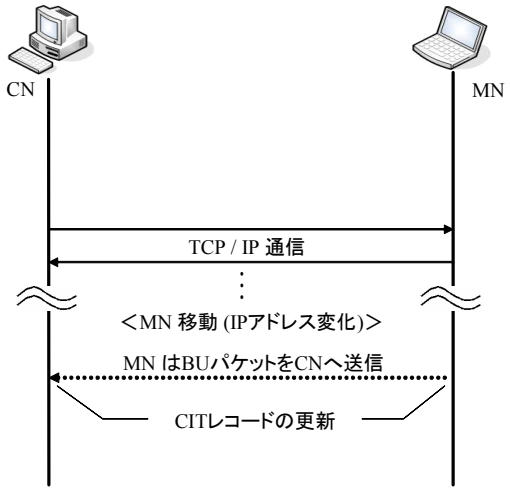


図 2　移動情報の通知

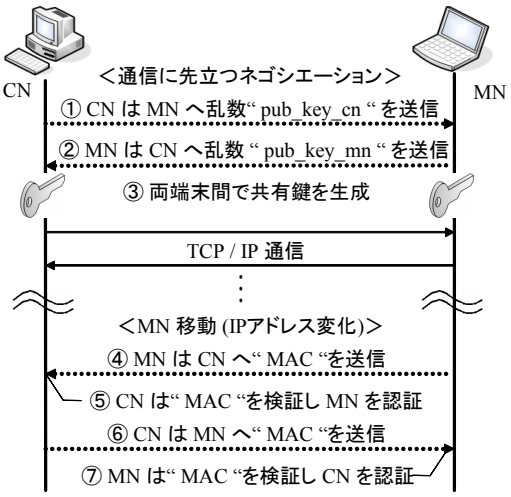


図 4　Mobile PPC における認証方式

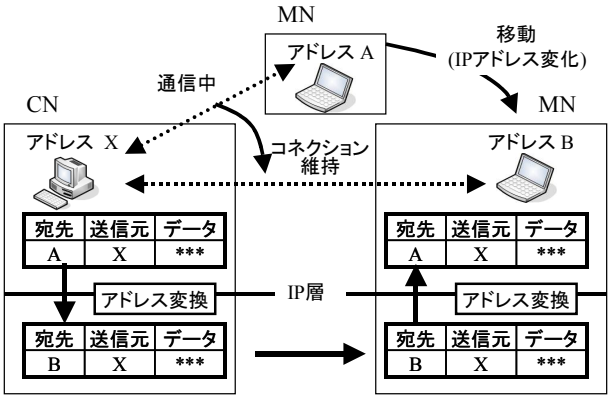


図 3　アドレス変換の例

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 HA を経由することによる通信経路の冗長および一点障害などの問題がない．また原理的に IPv4 と IPv6 のどちらにも適用可能な方式である．

現状の Mobile PPC はエンド端末間で事前に共有鍵を保持している環境を前提に検討されているため，一般の環境では BU を認証する機能がない．このため通信の乗っ取りの懸念があり汎用性に欠けている．Mobile PPC をグローバルな環境で使用する場合，セキュリティの観点から BU パケットの確実な認証が必要である．

3.2.　Mobile PPC における認証方式

本稿では特殊な第三の装置を使用することなく，P2P で移動時の成りすましを防止するための機構として Diffie-Hellman 鍵交換[11]を利用した認証方式を提案する．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したがい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．本提案方式では通信に先立って端末間でネゴシエーションを行う機構を Mobile PPC へ追加し，その機構を用いて Diffie-Hellman 鍵交換を行うことにより 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

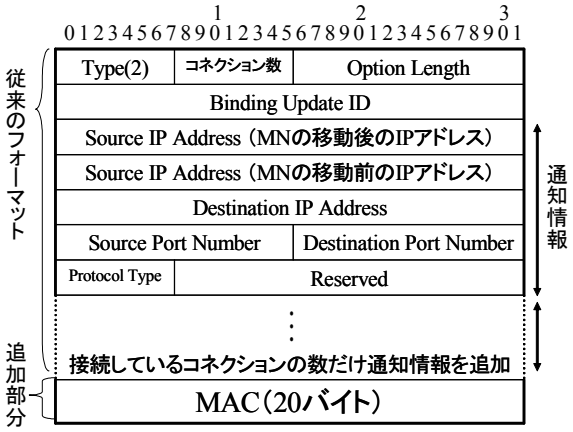


図 5　BU パケットフォーマット

を行う．

提案方式の流れを図 4 に示す．はじめに，CN は priv_key と呼ぶ乱数を生成し，その乱数から pub_key と呼ぶ値を算出する．ここで，CN が生成した priv_key を priv_key_cn, pub_key を pub_key_cn と呼ぶ．CN は TCP/IP 通信に先立ち MN へ pub_key_cn を送信する（①）. CN から pub_key _cn を受信した MN は priv_key を生成し, priv_key から pub_key を算出する．ここで，MN が生成した priv_key を priv_key_mn，pub_key を pub_key_mn と呼ぶ．MN は CN へ pub_key_mn を返信する（②）．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（③）．

以上の Diffie Hellman 鍵交換の動作が完了した後，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は，BU パケットに共有鍵で作成した MAC(Message Authentication Code) を付加し CN へ送信する（④）．CN は BU パケットを受信すると付加された MAC を検証し MN の認証を行う（⑤）．CN は MN を認証すると，BU 応答パケットを共有鍵で作成した MAC を付加し送信する（⑥）．MN は CN から BU の応答パケットを受信すると付加された MAC を検証し CN の認証を行う（⑦）．

BU パケットのフォーマットを図 5 に示す．BU パケット

表 1　Mobile PPC のモジュール機能

モジュール	機能
アドレス変換	IPアドレスを変換するモジュール. 送信／受信パケット毎に呼び出される.
移動管理	移動の通知処理を行うモジュール.
CIT 操作	CITを管理するモジュール.
CIT 削除デーモン	CITを監視し，無通信状態のレコードを削除する.

表 2　Mobile PPC へ追加するモジュール機能

モジュール	機能
乱数交換	通信に先立つネゴシエーションを実行するモジュール.
NIT操作	NITを管理するモジュール.
BU応答	BU応答パケットの送信を行うモジュール.
認証	端末間の認証を行うモジュール.

は ICMP Echo Request をベースに定義されており，移動先における IP アドレス取得処理をトリガーとして MN が生成・送信する．ICMP のデータ部分には，移動情報が格納されている．通知する情報は，MN が移動後に取得した IP アドレスとエンド端末間で行われていた全通信のコネクションの識別子である．本実装では従来の BU パケットのトレーラに MAC を付加する．MAC の計算には HMAC_SHA1 を使用し，次式に従って生成する．

MAC = HMAC_SHA1 (msg ， 共有鍵)　　(1)

msg は MAC による認証の対象範囲であり BU パケット全体を示す．共有鍵は Diffie Hellman 鍵交換によって作成した共有鍵を示す．

4.　実装

4.1.　モジュール構成

Mobile PPC の機能を実現するためのモジュールを表 1 に示す．Mobile PPC は FreeBSD 上に実装されており，IP 層に組み込まれるものとして CIT レコードの内容にしたがってアドレス変換処理やそれに伴うチェックサムの再計算を行う「アドレス変換モジュール」，自端末の IP アドレス変更時に，BU パケットを生成し，通信相手に移動情報を通知する「移動管理モジュール」，CIT レコードの検索・生成・更新を行う「CIT 操作モジュール」，アプリケーションレベルで動作するものとして「CIT 削除デーモン」があり，既存の TCP/IP の処理に変更を加えないようにパケット受信時には IP 入力関数である ip_input，パケット送信時には IP 出力関数である ip_output 内で Mobile PPC を呼び出し，処理を終えたら差し戻す形を取っている．

提案方式を実現するために，Mobile PPC へ追加するモジュールを表 2 に示す．追加モジュールはすべて IP 層に組み込む．通信に先立ち Diffie Hellman 鍵交換により共有鍵の生成を行う「乱数交換モジュール」，端末間ごとの通信の状況と共有鍵を記録するテーブル(Node ID Table；以下 NIT)の検索・生成・更新を行う「NIT 操作モジュール」，受信した BU パケットの認証完了時に BU 応答パケットを生成し通信相手へ送信する「BU 応答モジュール」，MAC の生成・付加・検証を行う「認証モジュール」がある．

5.　評価

本提案方式において，攻撃者が BU を使用し CN と MN 間の通信を乗っ取るためには，CN と MN が通信に先立って作成する共有鍵を取得する必要がある．通信開始に先立つネゴシエーションを盗聴し共有鍵を作成するには 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 を取得する必要がある．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．このため，攻撃者が BU を使用し CN と MN 間の通信を乗っ取ることは不可能である．

6.　むすび

Mobile PPC における認証方式の提案を行った．本提案方式を Mobile PPC へ適用することにより BU における通信の成りすましを防止することが可能となる．本提案方式は特殊な第三のネットワーク機器を必要とせずエンド端末間のみで実現可能であるということと，ネットワーク層ですべての処理を行うため上位のソフトウェアに影響を与えないという利点がある．また，Mobile IPv6 で導入されている Return Routability に比べて安全性が高い．

Mobile PPC の実装は完成しており，現在は本提案方式を Mobile PPC へ組み込む作業を行っている．今後は提案方式の実装を通して有効性の確認を行う．

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，渡邊晃，“モバイル端末の移動透過性を実現する Mobile PPC の提案,” 情報処理学会研究報告，2004-MBL-30, pp.17-24, Sep. 2004.

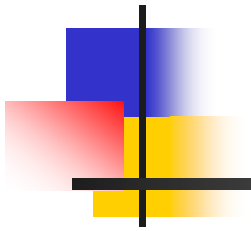
[7] 田中康之，國司光宣，石山政浩，寺岡文男，“LIN6 および HLIN6 における認証機構,” 電気情報通信学会論文誌, vol.J87-D-I No.5, pp.497-507, May.2004.

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

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

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

[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.



Mobile PPCにおける認証方式の提案

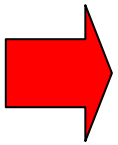
名城大学大学院　理工学研究科

瀬下正樹　竹内元規　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

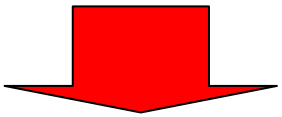
  ⇒端末が自由に移動しながらネットワークに接続するというニーズが増加
- 目的
  - 移動中にIPアドレスが変化しても通信を継続する



ノード移動透過性の実現

研究背景

- 我々は，Mobile PPCの研究を行なっている
  - エンドツーエンドで移動透過性を保証
- Mobile PPCの課題
  - 移動時の成りすまし

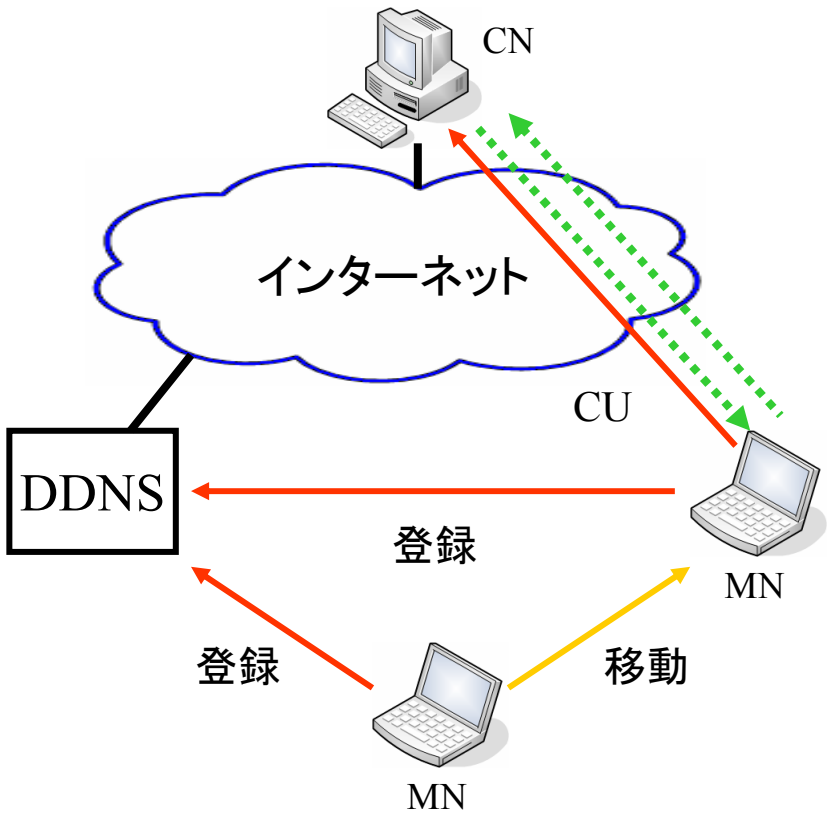


Mobile PPCにおける認証機構を提案

Mobile PPC
(Mobile Peer to Peer Communication)

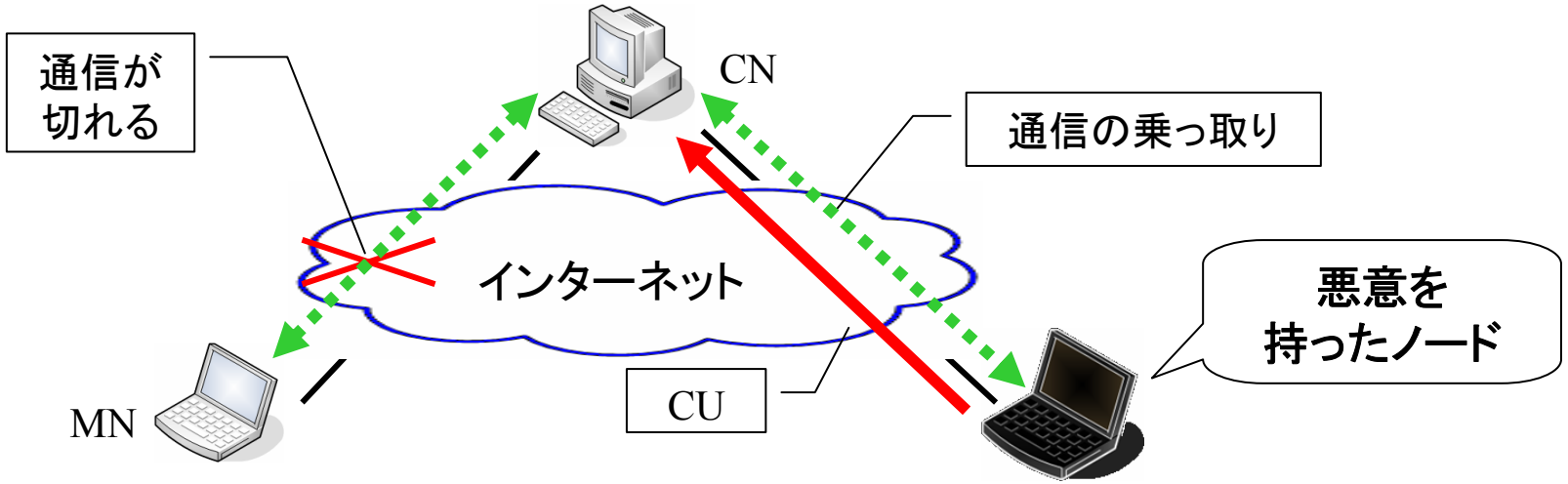
独自技術

- 動作概要
  - 初期IPアドレスの解決にはダイナミックDNS (DDNS)を使う
    - ホスト名とIPアドレスを動的に管理
  - 通信中のIPアドレスの変更はCU(CIT Update)を用いて**エンドエンドで通知**
    - 移動前後の対応関係を示すテーブル(CIT)を生成
    - IP層でアドレス変換を行うことにより上位層にIPアドレスの変化を隠蔽



CUの課題
（エンドエンドでIPアドレスを通知する課題）

- MNとCNが通信中
  - 悪意を持ったノードがアドレス登録 → 通信の乗っ取りが起こる

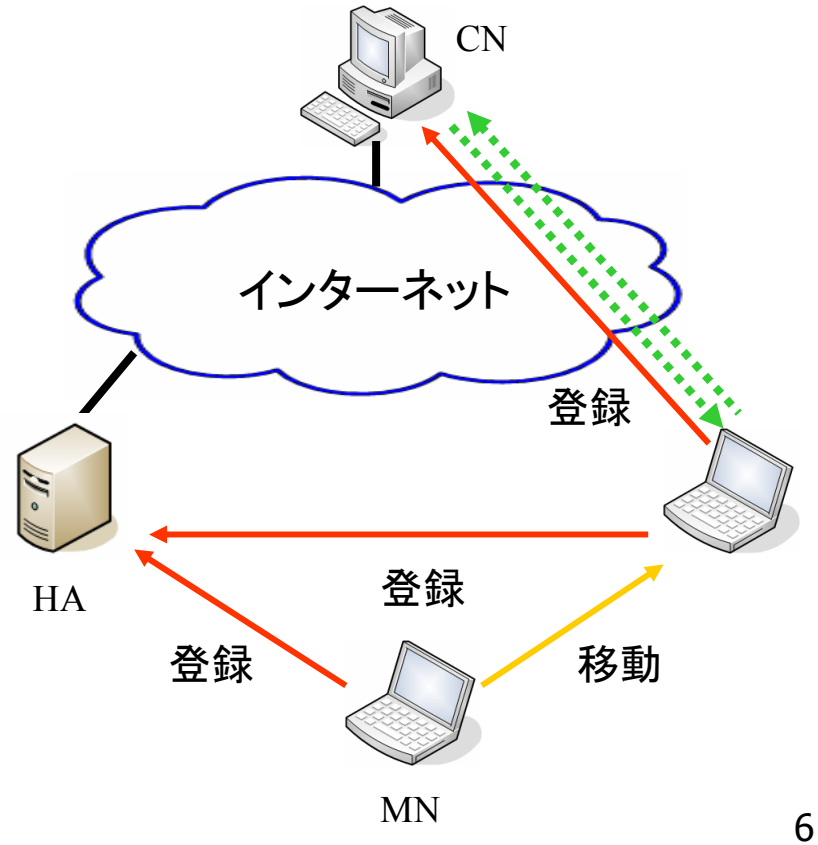


- CNにアクセスしてくるMNは不特定多数
  - 事前に認証に必要なMNの鍵を持つことは難しい
  - PKI の利用は現在の普及状況では現実的でない → 新たな認証機構が必要

関連研究：Return Routability

- Return Routability
  - Mobile IPv6の経路最適化時に使用される認証機構
- Mobile IPv6
  - IPv6においてノード移動透過性を実現するプロトコル
  - 経路最適化
    - 通信中の<u>IPアドレスの変更をエンドエンドで通知</u>
    - IP層で拡張ヘッダを利用したアドレス変換

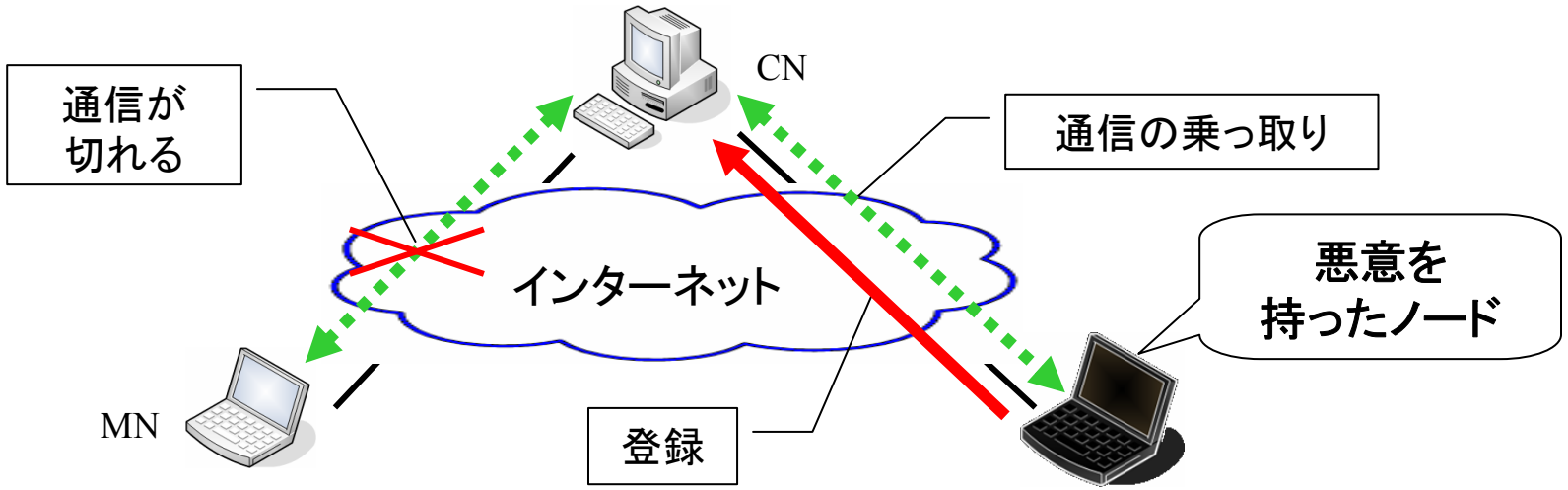
    ⇒CNとMNが直接通信



経路最適化の課題

- MNとCNが通信中
  - 悪意を持ったノードがアドレス登録

通信の乗っ取りが起こる



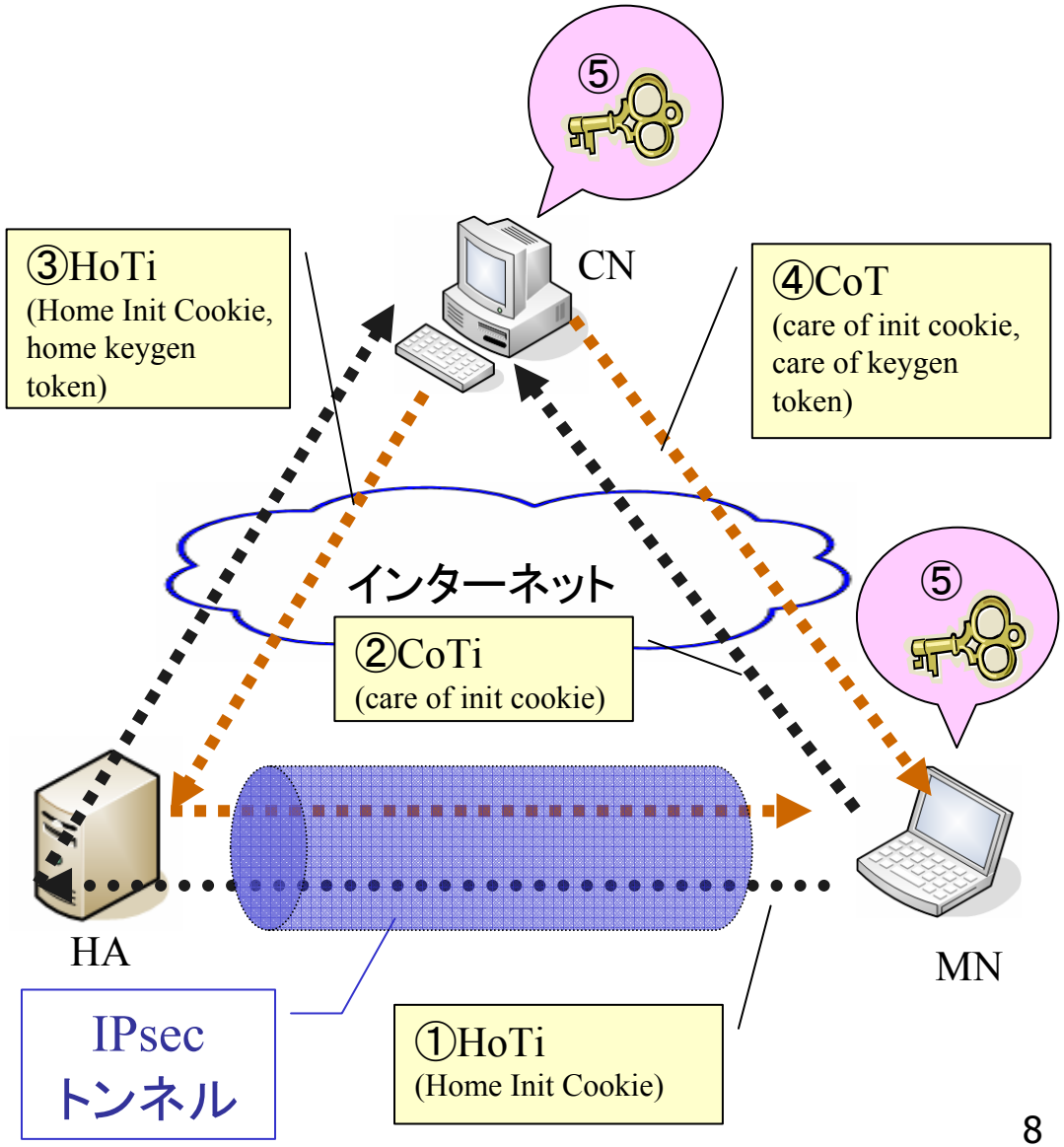
Mobile IPv6では，この問題をReturn Routabilityで解決

Return Routabilityの仕組み

- Return Routability
  - 前提条件
    - HAと呼ぶ第三の機器の導入
    - HAとMNは信頼関係にあることが前提

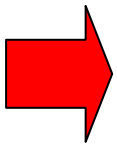
    ⇒HAとMN間はIPsecで保護できると考える
- 動作概要
  - アドレス登録直前
    - 共有鍵を二つに分け，異なる経路から配送（①から④）

    ⇒共有鍵を安全に生成（⑤）
  - アドレス登録時に共有鍵を用いた認証行う



Return Routabilityの問題点

- 問題点
  - HAのような特殊な装置を利用する
  - 盗聴の問題
    - CNと同一セグメント上
      - 2つの鍵が平文のまま流れる
        ⇒容易に共有鍵の盗聴が可能
    - CN～HA間，MNからCN間
      - 2点を同時に盗聴した場合，共有鍵の盗聴可能

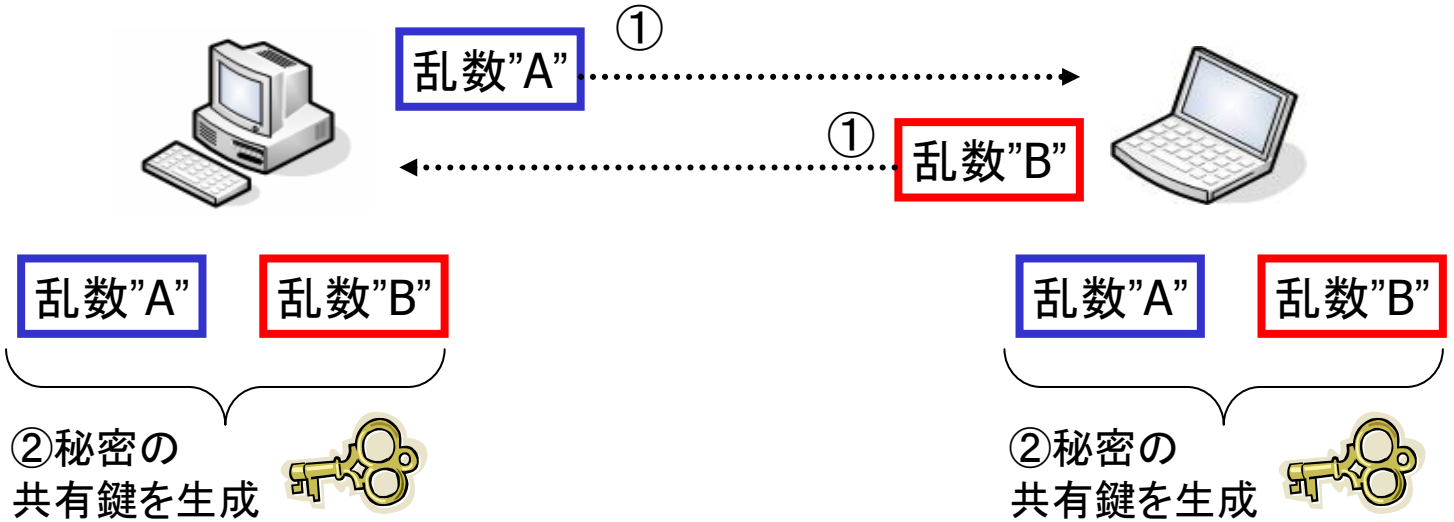


Mobile PPCのようにエンドエンドで移動透過性を保証するプロトコルには適していない

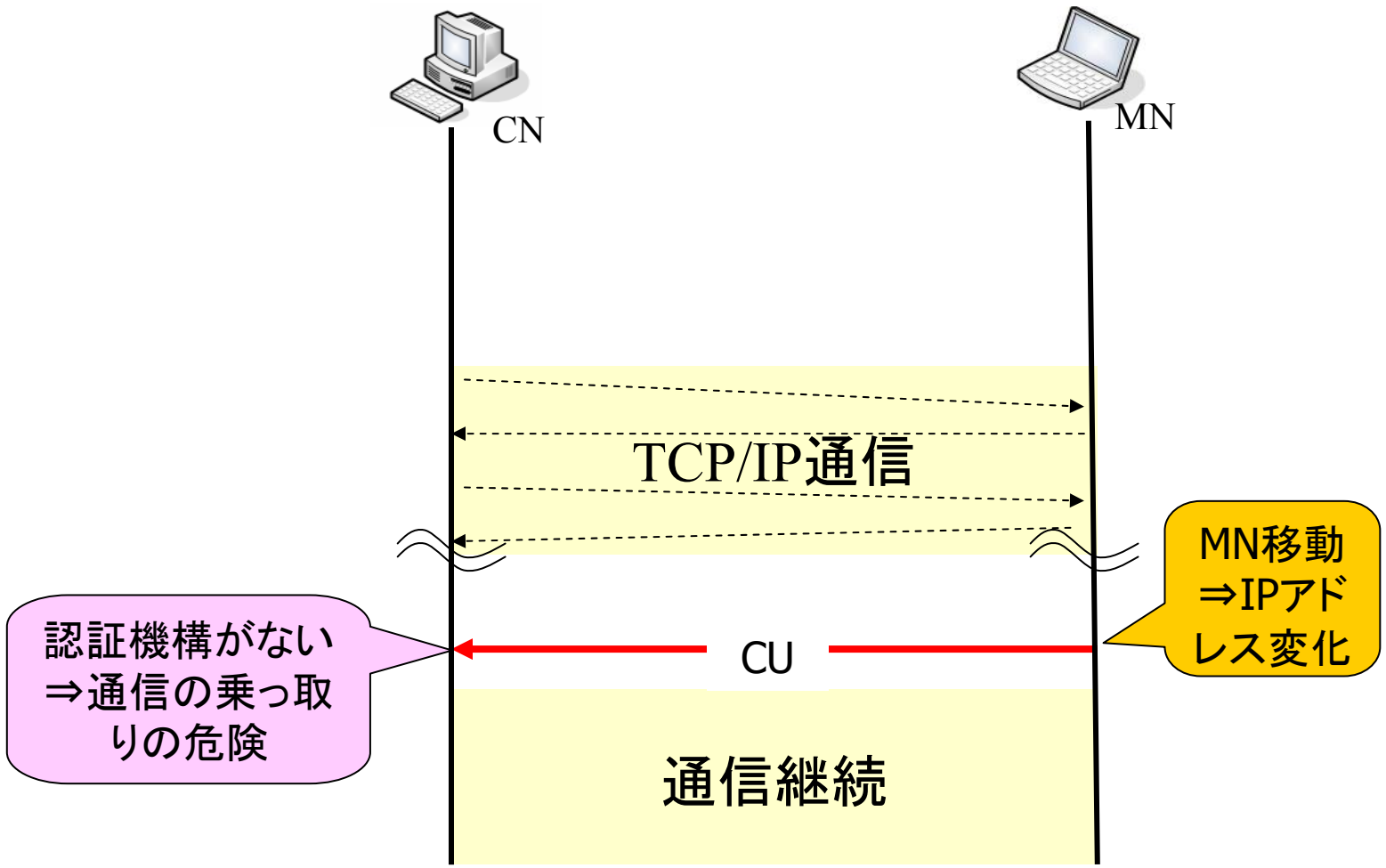
エンドエンドで実現可能な Mobile PPCにおける認証方式

提案技術

- 認証機構として，Diffie-Hellman鍵交換を利用
  - Diffie Hellman鍵交換
    - 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技術（②）
    - ✧ 離散対数問題を利用



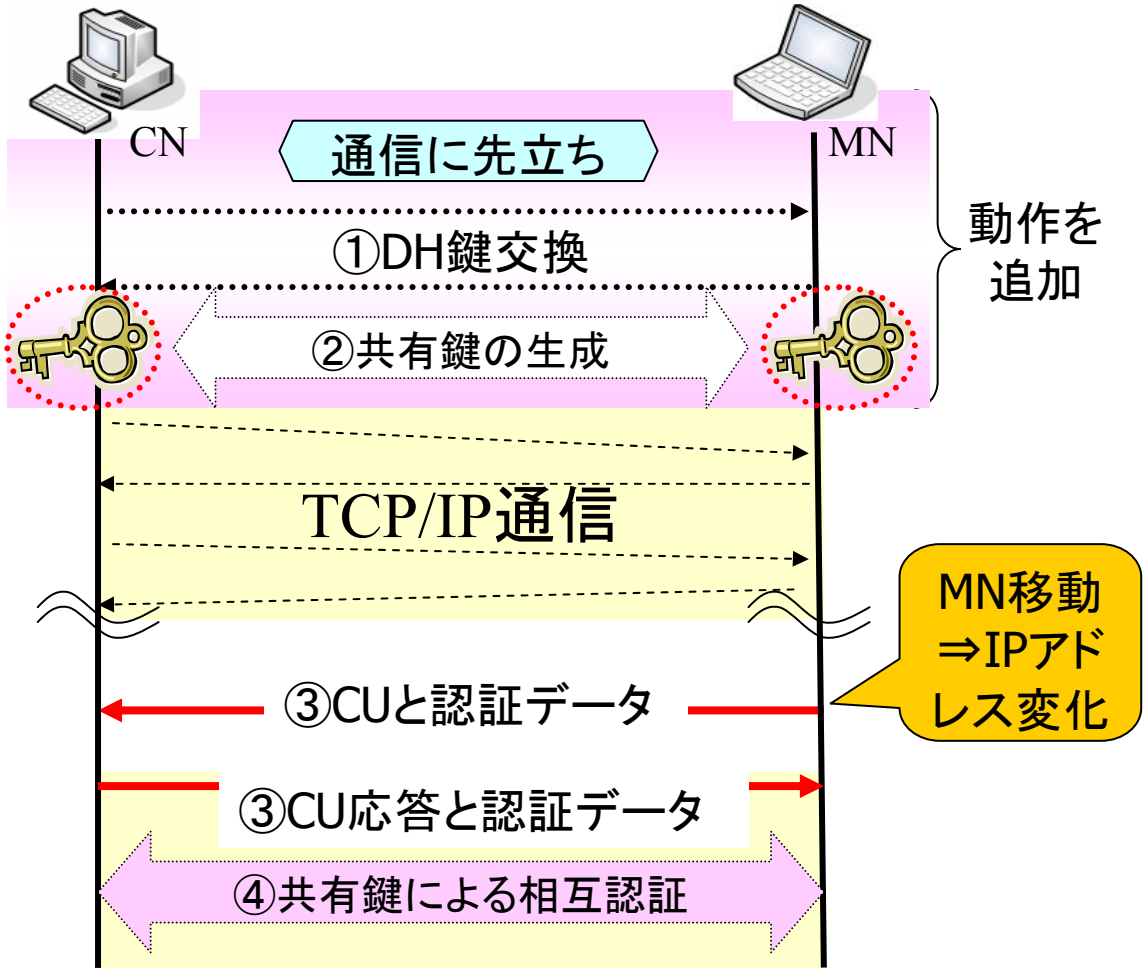
Mobile PPCのシーケンス



提案方式を追加した Mobile PPCのシーケンス

動作概要

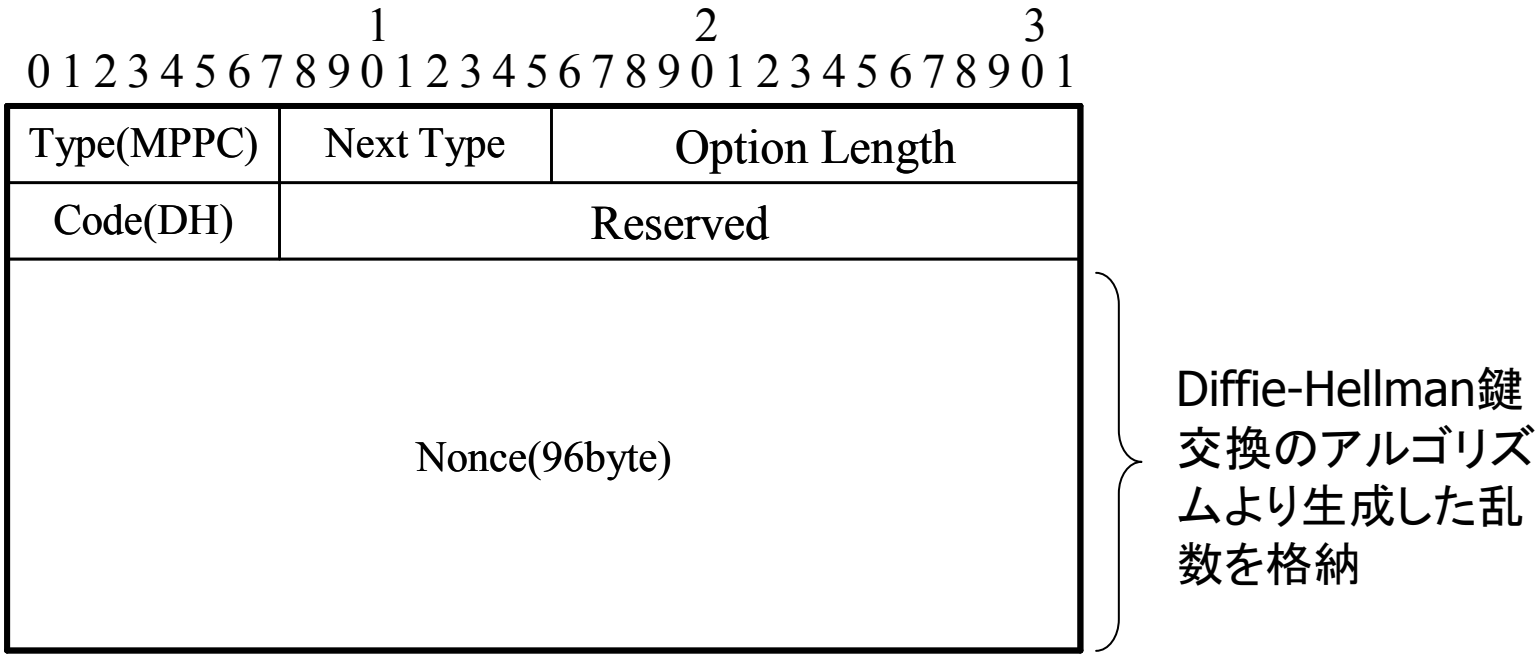
- 通信に先立ち
  - ①Diffie-Hellman鍵交換
  - ②共有鍵を生成
- *その後，通常のTCP/IP通信*
- MN移動
  - ③MNはCUと認証データをCNへ送信，CNはCU応答と認証データをMNへ送信
  - ④共有鍵を使用して相互認証



CUにおける通信の乗っ取りを防止することが可能

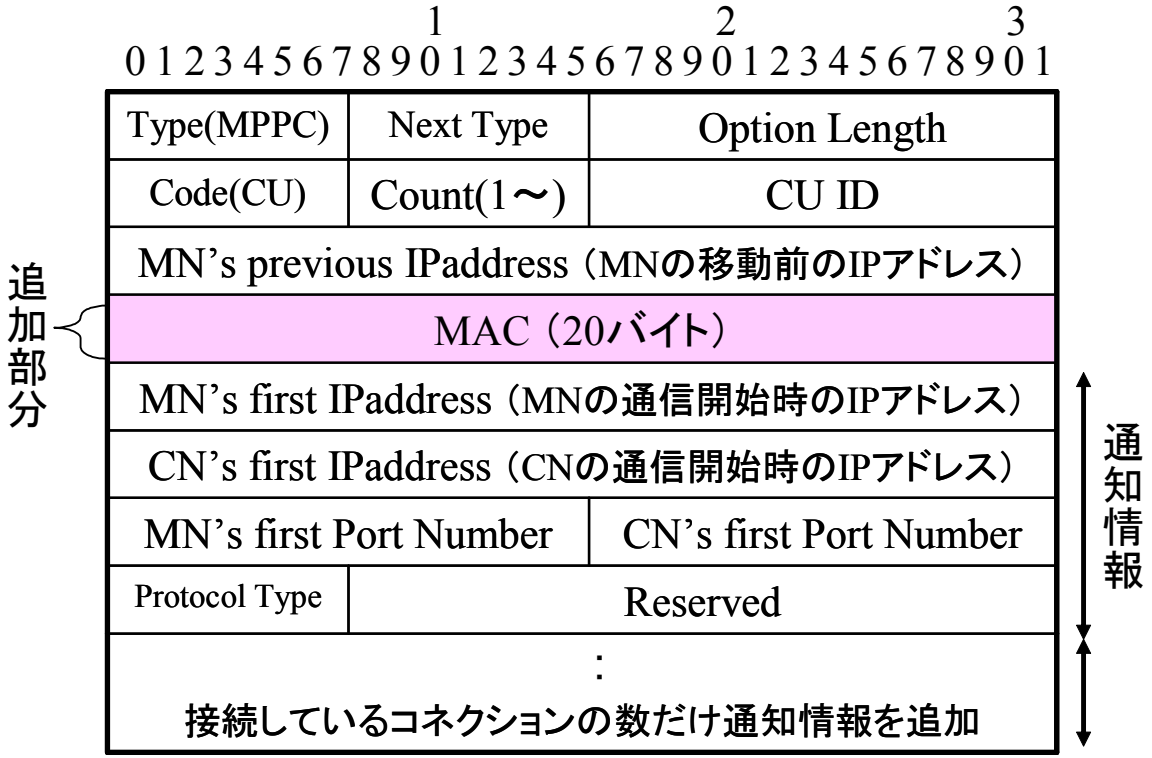
パケットフォーマット

- 通信に先立つネゴシエーションパケットのフォーマット
  - ICMP Echo Requestをベースの定義



パケットフォーマット

- CUパケットフォーマット
  - 従来のフォーマットに新たに認証データ（MAC）を追加
  - MACは一方向ハッシュ関数を用いて共有鍵とCUパケット全体から計算

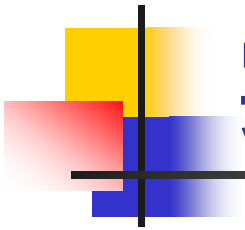


提案方式の評価

- CUパケット受信時における通信の乗っ取り
  - 提案方式は通信に先立って安全に共有した共有鍵を使用して認証を行う.

通信の乗っ取りは不可能

- 通信に先立つネゴシエーションにおける認証
  - 本提案方式では提供していない.
    - 移動透過により発生する通信の乗っ取りの防止することを目的
  - 通信開始時に特定の通信相手を認証したい場合は，通常のTCP/IP通信と同様に，アプリケーションレベルでのパスワードによる認証や，あらかじめ秘密共有鍵を設定することによる認証を行うことを推奨



実装

- 現状のMobile PPC
  - FreeBSDのカーネル部にモジュールを組み込むことで実現
    - IP層の入出力時に呼び出し，処理を終えたら差し戻す
    - IP層で行われる既存の処理へ影響を与えない
- 本実装
  - これまでのMobile PPCにモジュールを追加することで認証方式を実現

実装

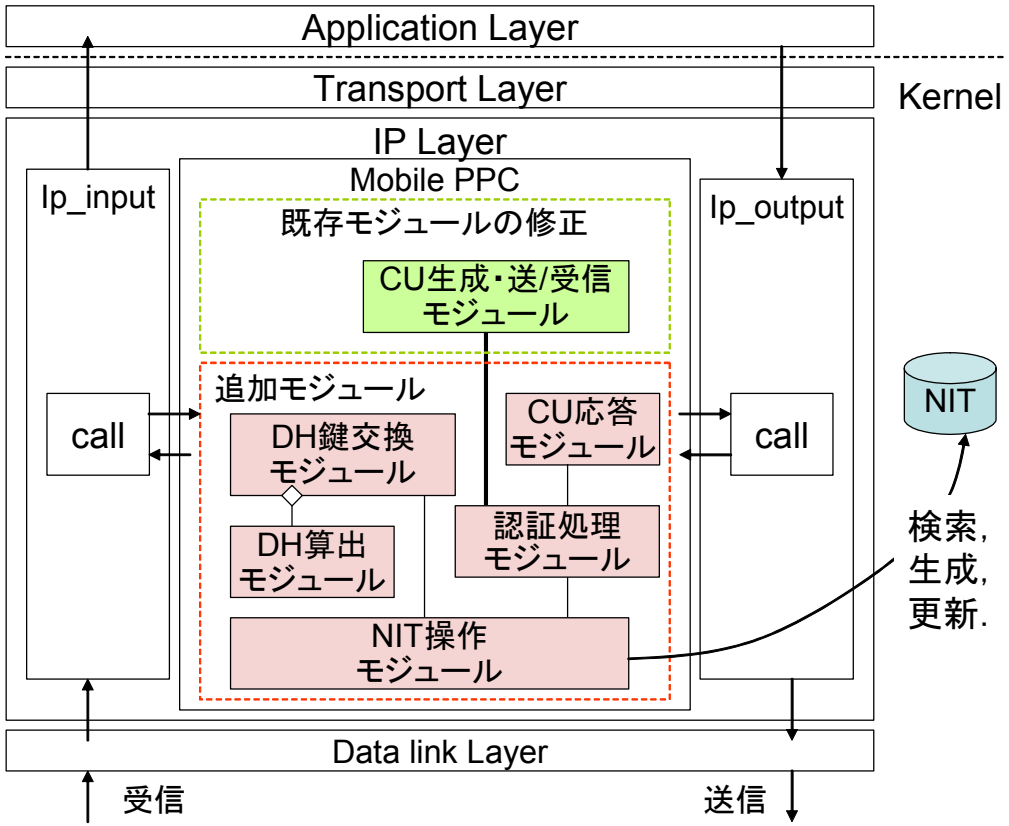
- Mobile PPCにおける認証方式
  - Diffie-Hellman鍵交換を端末単位での通信に先立ち実行

  ⇒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が必要
- NIT (Node Information Table)
  - 端末間での通信の有無を判断する情報を格納
  - 共有鍵に関連する情報も格納
- NITフォーマット

検索キー

自端末 IP	相手端末 IP	自端末DH 乱数	相手端末 DH 乱数	共有鍵	state
CN	MN1	A	B	Key1	Done
CN	MN2	C	D	Key2	Done

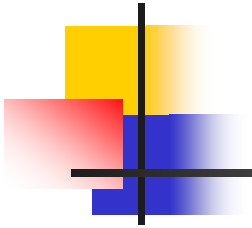
モジュール構成



- 追加モジュール
  - DH鍵交換モジュール
    - 通信に先立ちネゴシエーションを行なう

    ⇒当研究室で別途研究しているDPRPを流用
  - DH算出モジュール
    - DHアルゴリズムにより乱数と共有鍵の算出

    ⇒オープンソースライブラリであるOpenSSLを利用
  - NIT操作モジュール
    - NITレコードの検索・生成・更新を行なう
  - 認証処理モジュール
    - 認証データの生成・検証を行う
  - CU応答モジュール
    - CU応答の生成・送/受信を行う
- 修正モジュール
  - CU生成・送/受信モジュールから認証処理モジュールを呼び出す

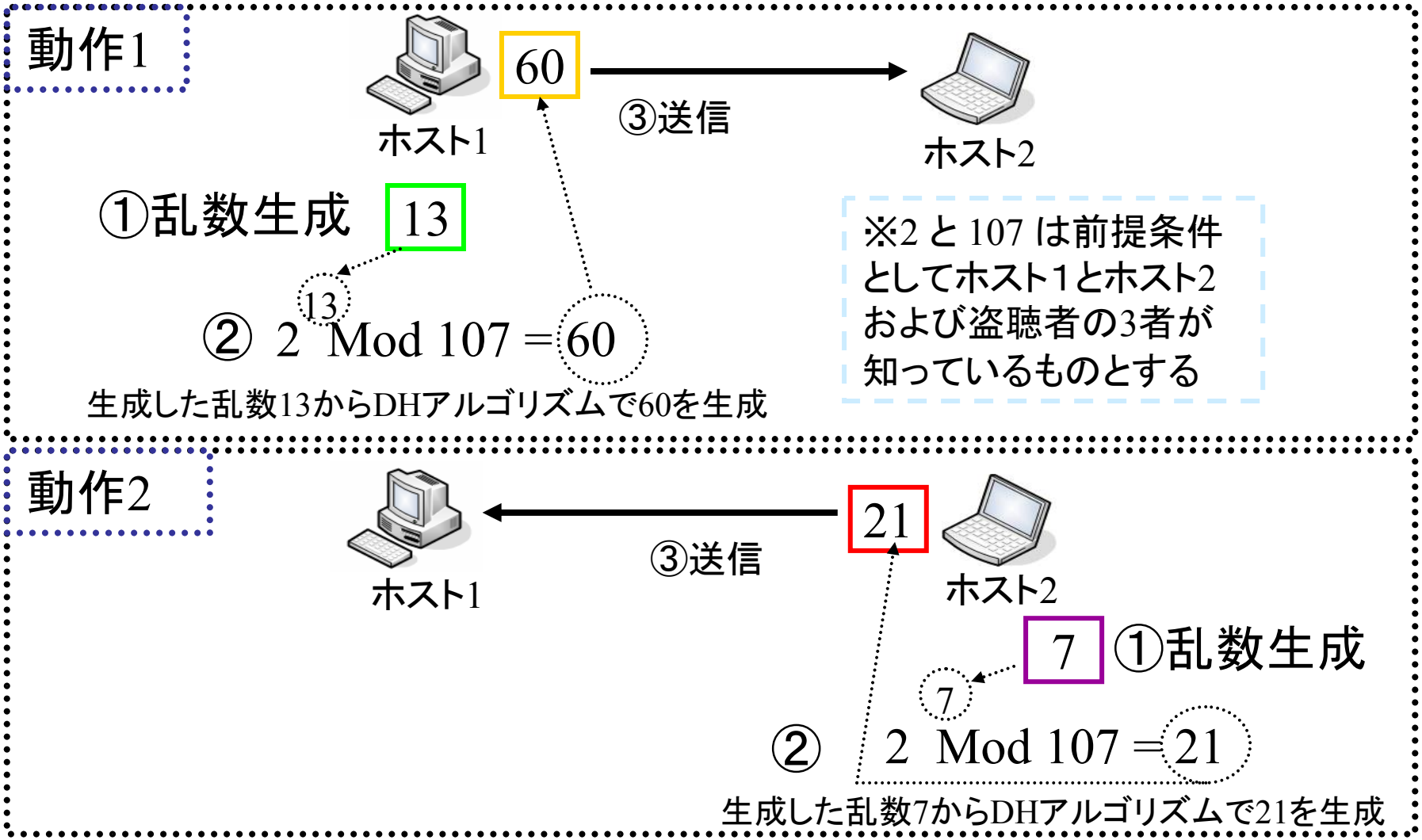


むすび

- Mobile PPCにおける認証方式を提案
- 今後は提案方式を実装し，有効性の確認を行う

おわり

付録. Diffie Hellman鍵交換の詳細例（その1）



付録. Diffie Hellman鍵交換の詳細例（その2）

動作3

ホスト1

ホスト2

$21^{13} \text{ Mod } 107 = 70$　　$60^{7} \text{ Mod } 107 = 70$

計算結果が一致

•上記したDiffie Hellman 交換は以下の式が成り立つことを利用

ホスト2が送信した値　ホスト1が送信した値

ホスト1の乱数　ホスト2の乱数

$B^{x} \bmod n = g^{x \cdot y} \bmod n = A^{y} \bmod n$

※g=2 と n=107 は前提条件としてホスト1とホスト2および盗聴者の3者が知っているものとする

•盗聴者が流れた乱数を盗聴したとして，共通鍵「70」を知るには以下の計算が必要

$$21^{x} \text{Mod } 107 = 2^{x \cdot y} \bmod 107 = 60^{y} \bmod 107$$

⇒この式から x , y を導き出すことは事実上不可能