Mobile PPC における認証方式の提案と評価

053432015　瀬下正樹
渡邊研究室

1. はじめに

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，一般には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盛んに行われている．我々は，移動透過性の一方式として，特別な位置管理装置（第三の装置）を不要とし，常時 P2P 通信をおこなうことができる Mobile PPC[1]の研究を行っている．しかし，これまでの Mobile PPC には移動ノード(以下 MN)が移動した際に通信相手ノード(以下 CN)との間で成りすましを防止するための認証機構が定義されていなかった．そこで，本研究では Mobile PPC における認証方式についての提案を行う．

2. Mobile PPC とその課題

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．DDNS は DNS の延長技術であり，既に実用化されている．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．MN が移動すると，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 CU(CIT Update)により通知し，CN は MN に対して CU REPLY を返信する（図 1）．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．以後の通信では全ての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．しかし，現状の Mobile PPC は CU ネゴシエーションの際に通信を乗っ取られる危険があり，セキュリティの観点から確実な認証が必要であるという課題があった．

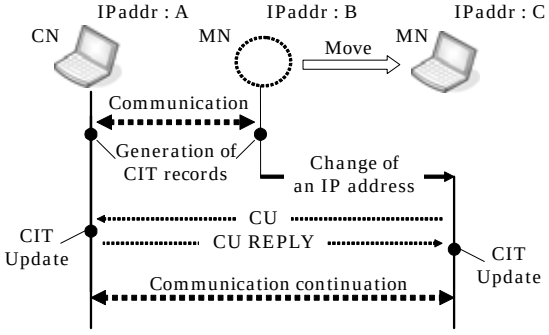


図 1　Mobile PPC の動作

2.1 Mobile PPC における認証方式の提案

Mobile PPC における認証方式では，Diffie-Hellman 鍵交換[2]を利用する．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（以下 DH 鍵）を交換することにより，その DH 鍵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，従来の Mobile PPC に対して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，cookie 交換と Diffie-Hellman 鍵交換を行う．cookie 交換では，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．これにより，不要な Diffie-Hellman 鍵交換による端末の計算資源の浪費を防止する．このネゴシエーションにより通信開始時に MN と CN の間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 における認証方式では，導入が容易なエンド端末間のみで認証を実現する MPPC ダイレクト認証と，第三の装置を利用してセキュリティ強度を高めた MPPC アドバンスト認証の 2 つの方式を提案する．MPPC アドバンスト認証における第三の装置としては改造した DDNS を用いる．また，2 つの方式は端末側で選択が可能である．

2.2 MPPC ダイレクト認証

MPPC 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．MPPC ダイレクト認証による共有鍵の生成方法を図 2 に示す．通信に先立ち，端末間で cookie 交換を行い（(1)），次に Diffie-Hellman 鍵交換により共有鍵を生成する（(2)）．移動時にはこの共有鍵を用いて認証を行う．

本方式は第三の装置が必要ないので従来の Mobile PPC への適用が非常に容易であるという利点がある．しかし，CN と MN 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．ただし，実際に中間者攻撃を実行するには高度な技術と準備が必要であり，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．

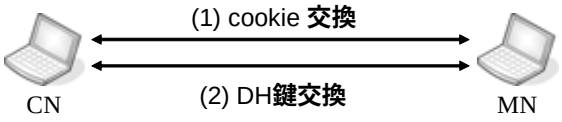


図 2　MPPC ダイレクト認証による
共有鍵の生成方法

2.3 MPPC アドバンスト認証

MPPC アドバンスト認証は通信に先立ち端末間で第三の装置を利用して Diffie-Hellman 鍵交換を実行する．本方式では DH 鍵を 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．第三の装置としては改造した DDNS を用いる． MPPC アドバンスト認証による共有鍵の生成方法を図 3 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 による通信を行う．通信に先立ち，端末間で cookie 交換を行なう．cookie は二つの異なる経路から配送する（(1)）．次に，DH 鍵も 2 つに分解し，二つの異なる経路からそれぞれ配送する（(2)）．エンド端末は DH 鍵を合成して共有鍵を生成する．移動時にはこの共有鍵を用いて認証を行う．

本方式は DDNScn と DDNSmn 間，CN と MN 間の二つの通信経路上における同時中間者攻撃に対して脆弱性があるが，これを実行する極めて困難であり，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．

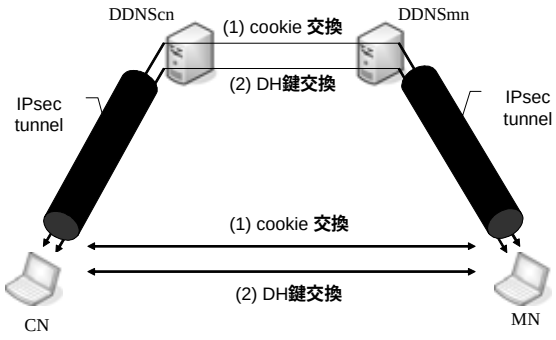


図 3　MPPC アドバンスト認証による
共有鍵の生成方法

3. 実装と評価

3.1 実装方法と評価内容

Mobile PPC における認証方式の 1 つである MPPC ダイレクト認証の実装を FreeBSD5.2.1 上で行った．MPPC 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．

MPPC ダイレクト認証における通信に先立つネゴシエーションの処理時間と移動情報通知処理時間の測定を行った．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，MPPC ダイレクト認証の測定結果より MPPC アドバンスト認証の処理時間を推測した．

3.2 通信に先立つネゴシエーション処理時間

(1) MPPC ダイレクト認証の処理時間

MPPC ダイレクト認証による通信に先立つネゴシエーションの処理時間は 501μ秒であった．この結果より，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ほとんどないといえる．

(2) MPPC アドバンスト認証の処理時間

MPPC アドバンスト認証は MPPC ダイレクト認証の結果から処理時間を推測した．それによると，MPPC アドバンスト認証による先立つネゴシエーションの処理時間は 1620μ秒である．このことより，MPPC アドバンスト認証においても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ほとんどないといえる．

3.3 移動情報通知処理時間

表 1 に移動情報通知処理時間（CU パケットおよび CU REPLY パケットの処理時間）を示す．従来の Mobile PPC における移動通知処理時間は 0.644 m 秒であり，MPPC ダイレクト認証における移動通知処理時間は 0.685 m 秒であった．処理時間の差は 0.041 m 秒（約 6%の劣化）であり，従来の Mobile PPC の移動通知処理に対してほとんど性能が劣化せず認証処理を実現することができた．

表 1　移動情報通知処理時間

	処理時間[m秒]
従来のMobile PPC	0.644
認証処理を追加したMobile PPC	0.685

4. むすび

Mobile PPC における認証方式を提案した．MPPC ダイレクト認証は第三の装置が不要であり，Mobile PPC への適用が容易である．MPPC アドバンスト認証は改造した DDNS の導入が必要となるが，MPPC ダイレクト認証よりもセキュリティ強度が高い．また，本提案方式を実装し，処理時間の測定をした結果，両方式ともに，通信に先立つネゴシエーションおよび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十分小さいことが実証された．

参考文献

[1] 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).

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

Mobile PPCにおける認証方式の提案と評価

A Proposal of an Authentication Mechanism for Mobile PPC and its evaluation.

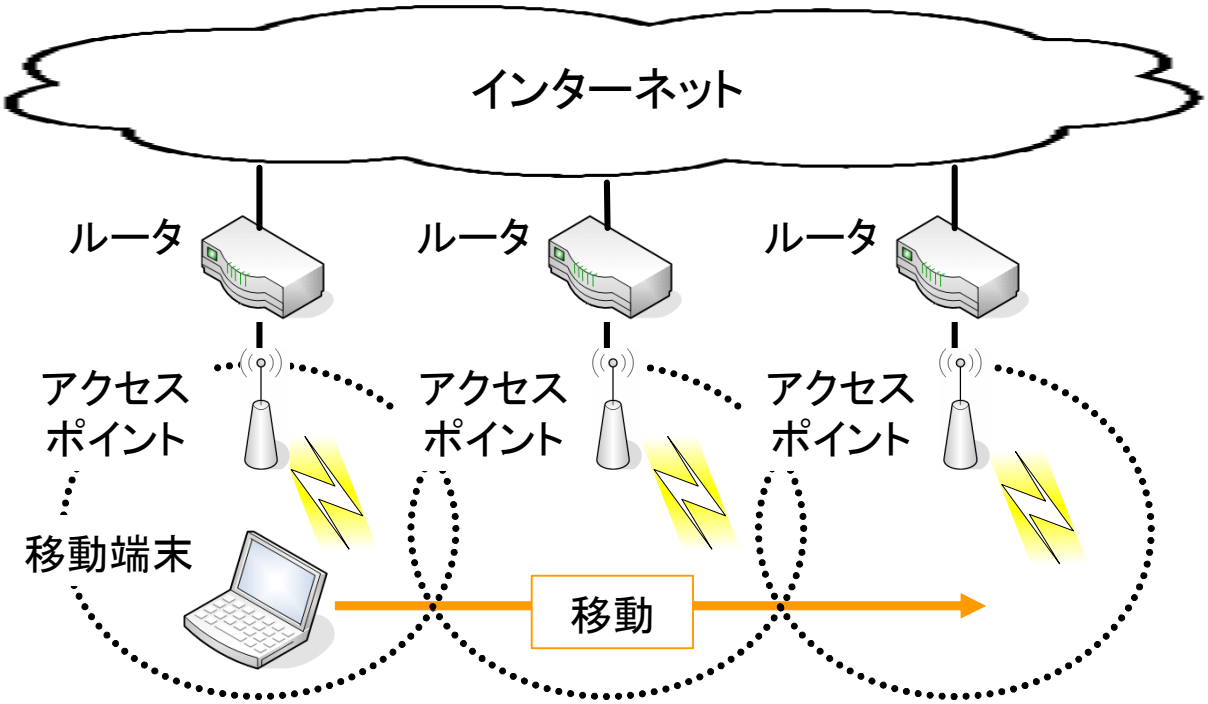
渡邊研究室

053432015　瀬下正樹

研究背景

- 研究背景
  - 移動端末の普及
  - 無線ネットワーク環境の普及

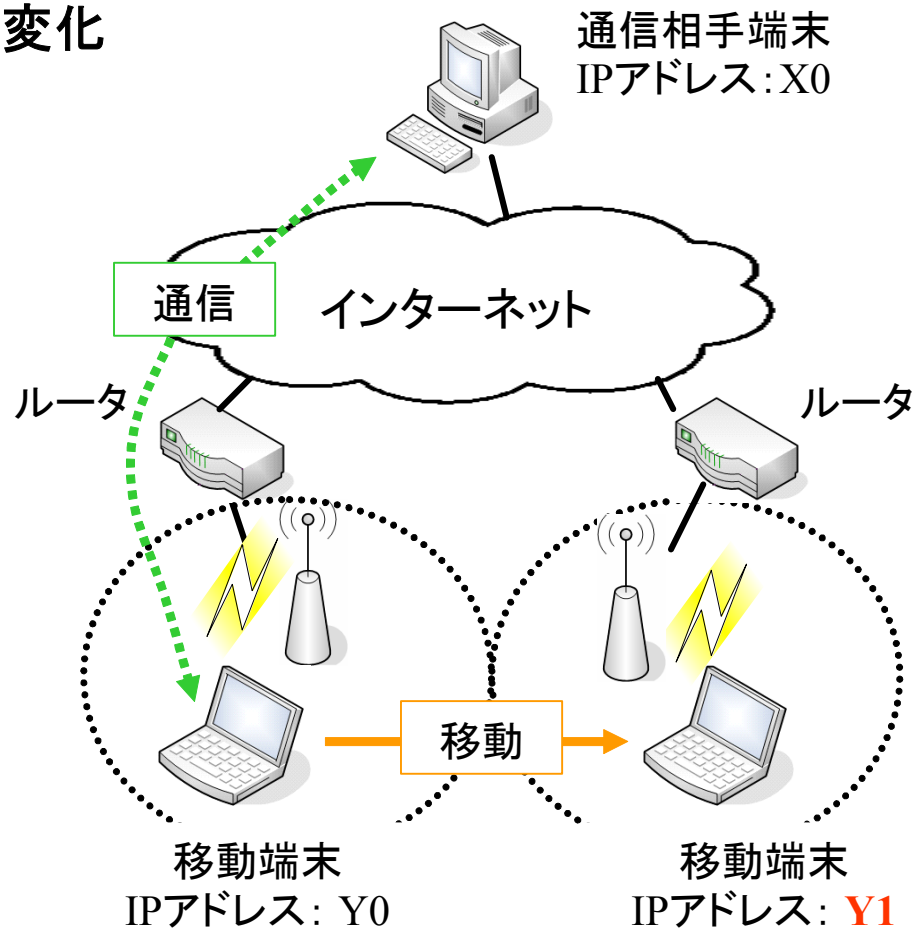
⇒端末が自由に移動しながらネットワークに接続するというニーズが増加



研究背景

- インターネット
  - 通信はIPアドレスとポート番号によって識別
  - 端末が移動するとIPアドレスが変化
- 通信中に移動
  - パケットが正しくルーティングされない
  - 別の通信とみなされ通信が切断される
- 目的
  - **通信中にIPアドレスが変化しても通信を継続**

移動透過性の実現

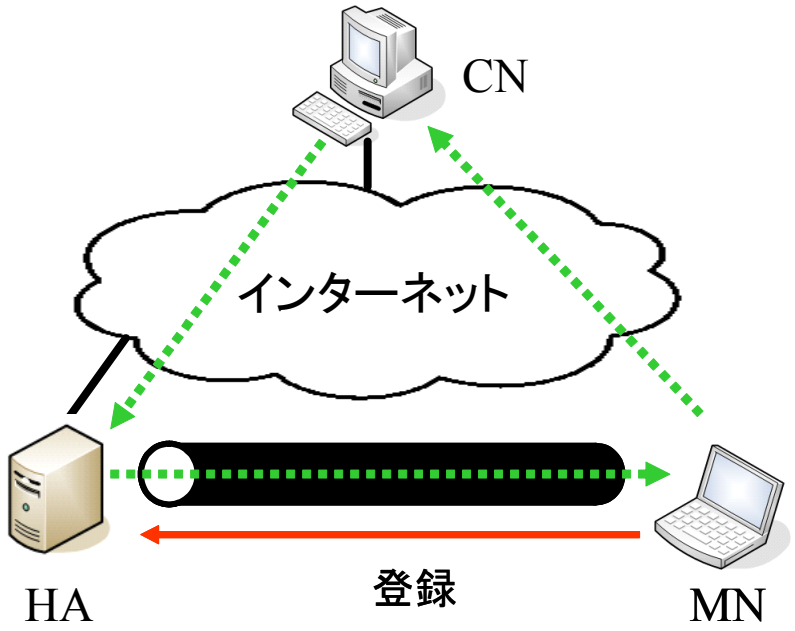


既存技術　Mobile IP

- Mobile IP
  - 第三の装置が導入：
    HA (Home Agent)

移動端末
（Mobile Node；MN）
通信相手端末
（Correspondent Node；CN）

- 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
  - 冗長な通信経路
  - HAとMN間はトンネル転送
  - 第三の装置(HA)が必要

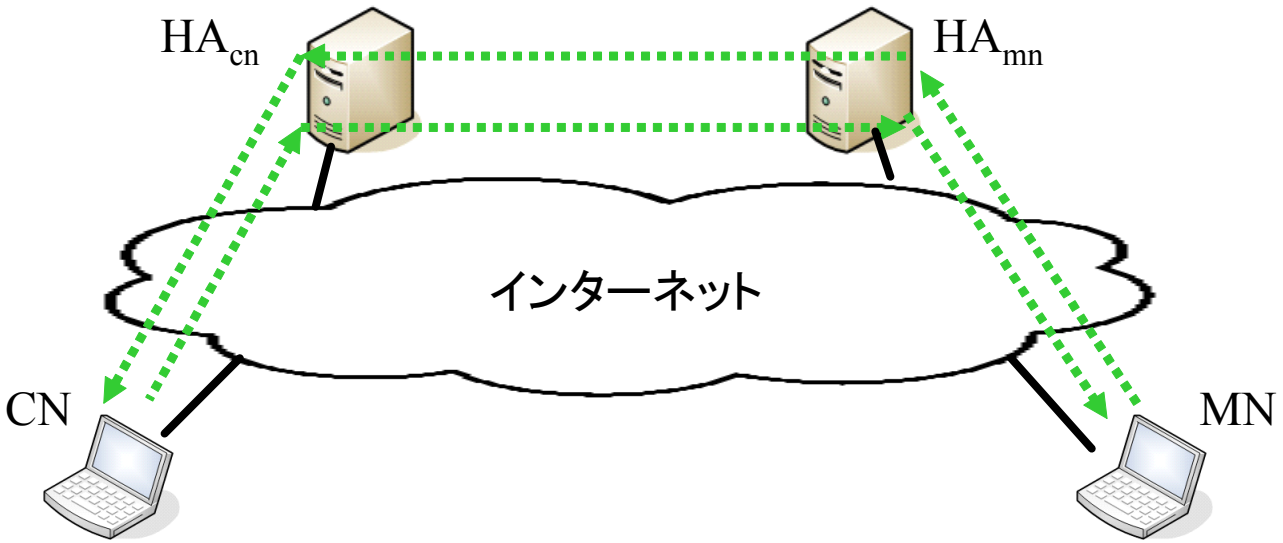


既存技術　Mobile IP

- CNも移動する場合
  - CNにもHAが必要

  ⇒導入の敷居がより高くなる

  ⇒通信経路がより冗長となる



Mobile IPのまとめ

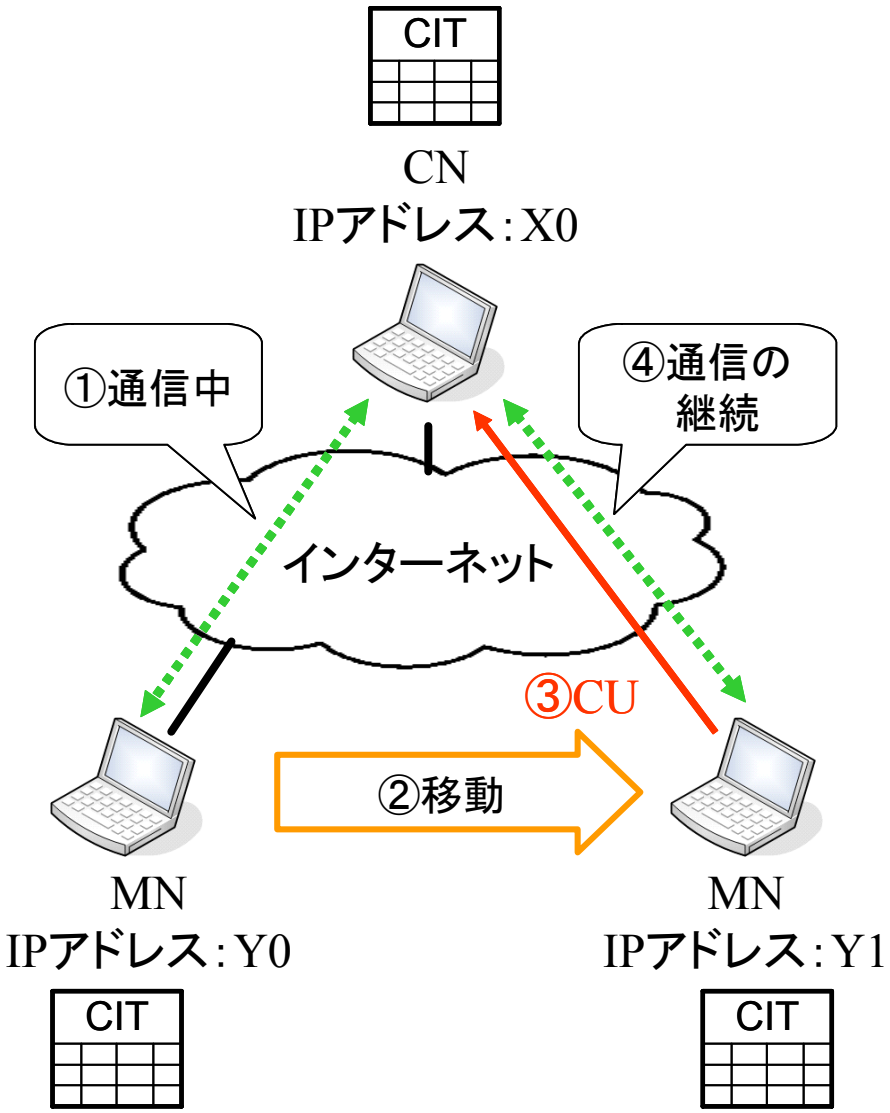
- Mobile IP
  - HAが必須
    - 導入の敷居が高い
    - サーバの二重化の対策が必須，管理負荷が大きい
  - 通信経路の冗長
    - リアルタイム性が失われる
- 今後のユビキタスネットワーク社会
  - 全ての端末に対して移動をサポート
  - IP電話などのリアルタイム性が要求されるアプリケーションへの対応
  - 個人間の通信が主体，第三の装置は極力排除

Mobile PPC (Mobile Peer to Peer Communication)の提案

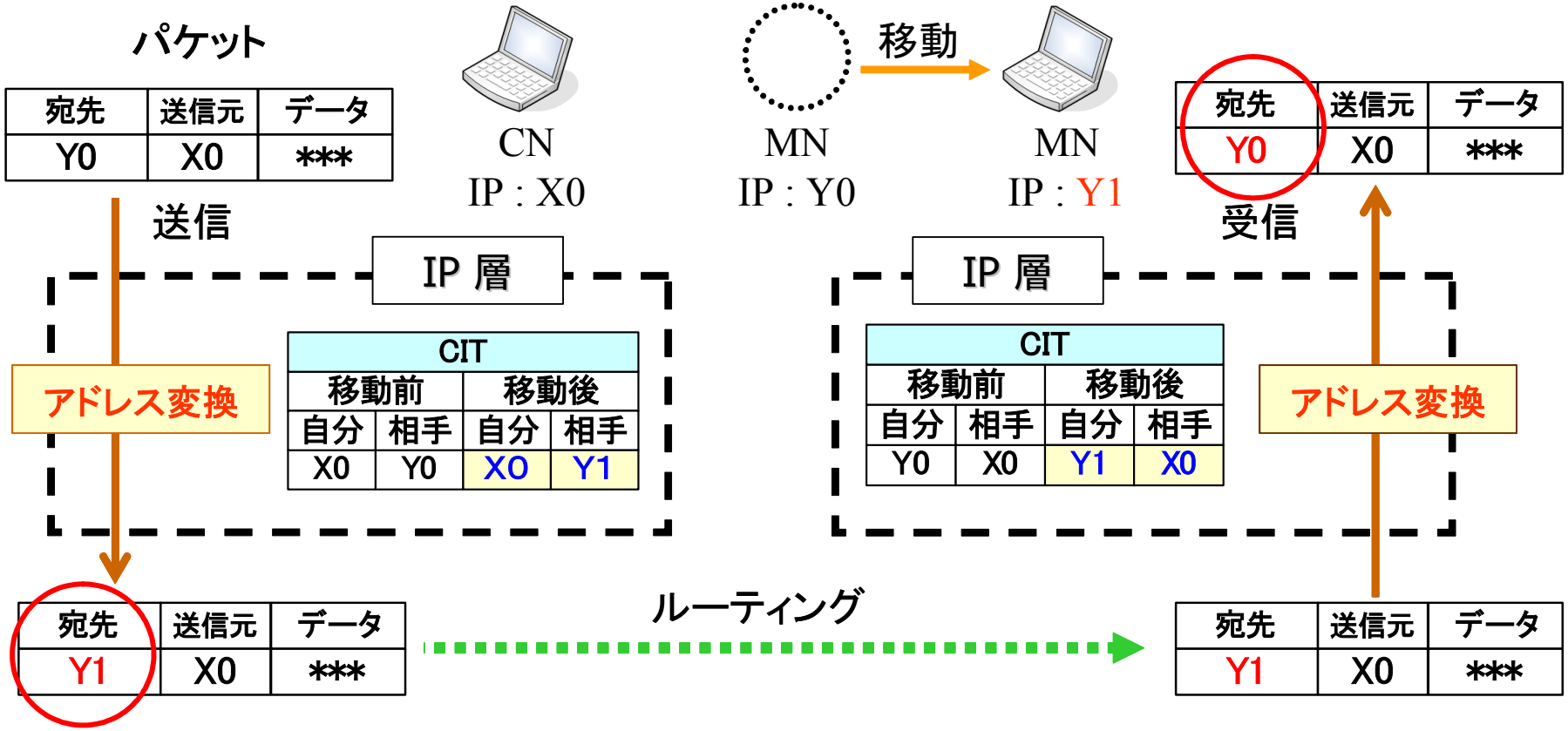
Mobile PPC(Mobile Peer to Peer Communication) 独自技術

- Mobile PPC
  - 第三の装置がいらない
  - 常に最適経路で通信
- 動作概要
  - 端末は移動前後のIPアドレスの対応関係を示すテーブル**CIT**（Connection ID Table）をIP層で保持
  - 通信中にIPアドレスが変化
    - **CU**(CIT Update)によりエンドエンドで新しいIPアドレスを通知しCIT情報を更新
    - パケット送受信時にIP層でCITを参照し，**アドレス変換**を行う

  ⇒IP層より上位層に対してIPアドレスの変化を隠蔽し通信継続



アドレス変換の詳細

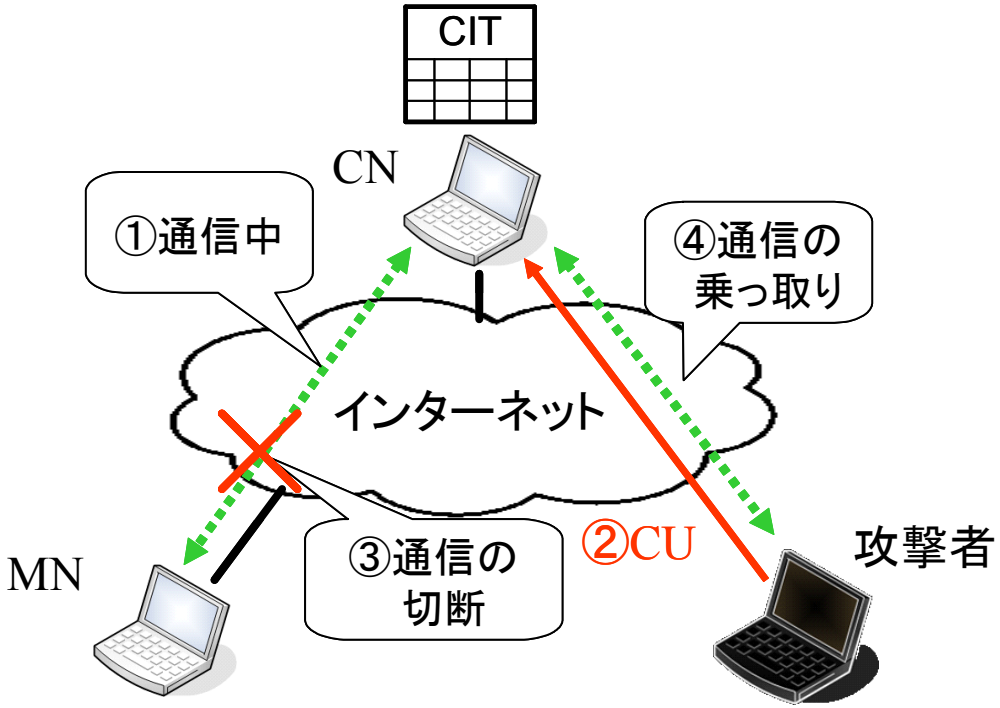


- CNからMNへ送信される宛先IPアドレスをCITを参照し“Y0” から “Y1”へアドレス変換 ⇒パケットが正しくルーティングされる
- MNはパケットを受信すると宛先IPアドレスをCITを参照し“Y1” から “Y0”へアドレス変換 ⇒上位層に対してIPアドレスの変化を隠蔽

Mobile PPCのセキュリティ上の課題

- MNとCNが通信中
  - 攻撃者がCNへCUパケット送信した場合，CITが不正に書き換えられる

通信の乗っ取りが発生



移動時における端末間の認証は必須

端末間の認証

- 端末間の認証
  - 共通の鍵（**共有鍵**）があれば可能

  ⇒共有鍵を両端末に安全に保持させる必要性
- 共有鍵の生成方法
  - 事前に両端末に共有鍵を設定

    ⇒CN と通信するMN は任意であるため難しい
  - CNからMNへ共有鍵を配送する

    ⇒何らかの工夫をしなければ簡単に盗聴される

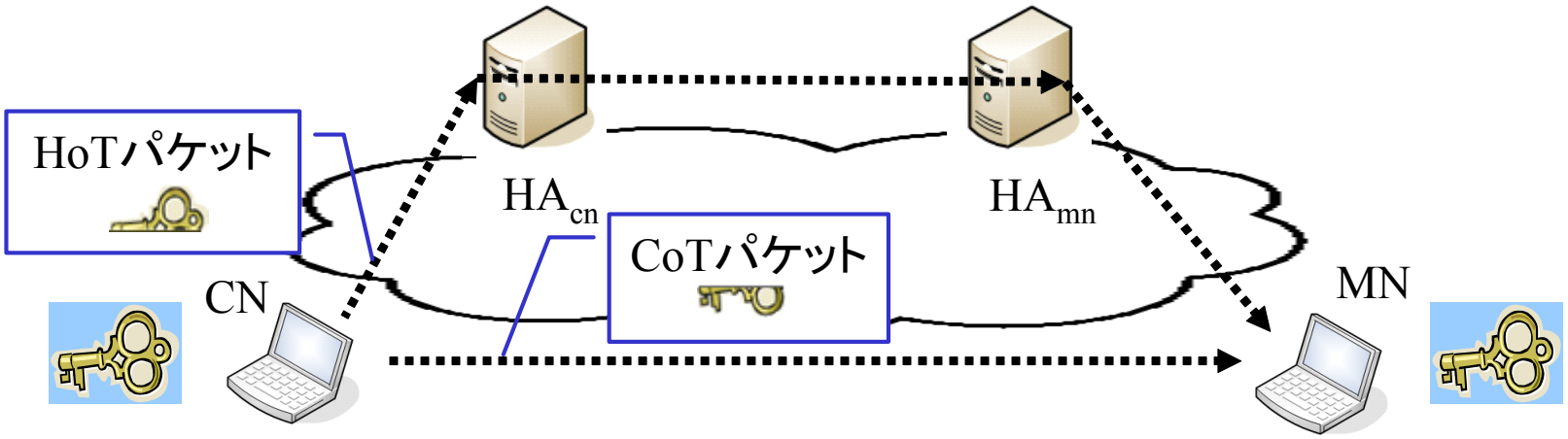
➢ Mobile PPC における端末間の認証

–CN とMN 間でいつ，どのようにして安全に共有鍵を生成するかが解決すべき課題

従来技術　Return Routability

- Return Routability
  - Mobile IPで導入
  - 通信に開始時にCNとMNで共有鍵を安全に生成
  - 第三の装置(HA)を利用
- 共有鍵の生成方法
  - 共有鍵を二つに分割
    - HAを経由したルート，経由しないルート，それぞれから配送

⇒二つの通信経路の同時盗聴されない限り，共有鍵を安全に生成可能



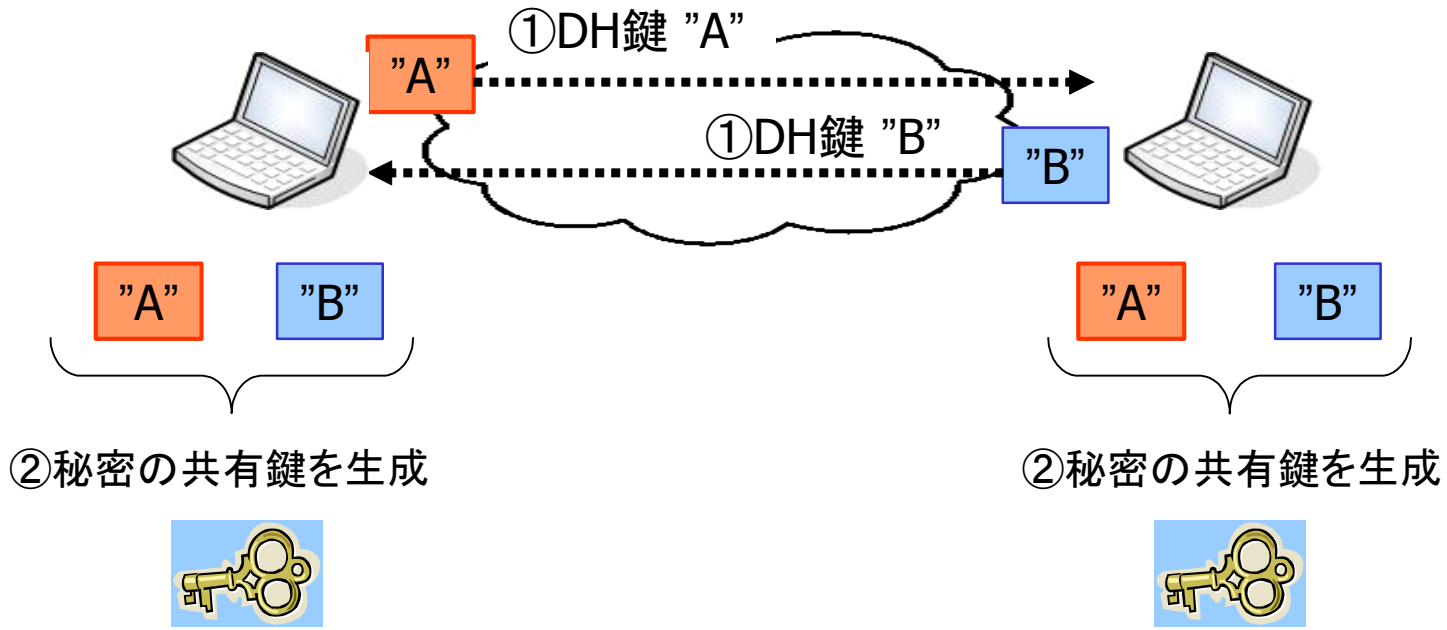
第三の装置(HA)を利用するためMobile PPCには適していない

Mobile PPCにおける認証方式

提案技術

- Diffie-Hellman（DH）鍵交換を利用する
  - DH鍵交換
    - DH鍵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）

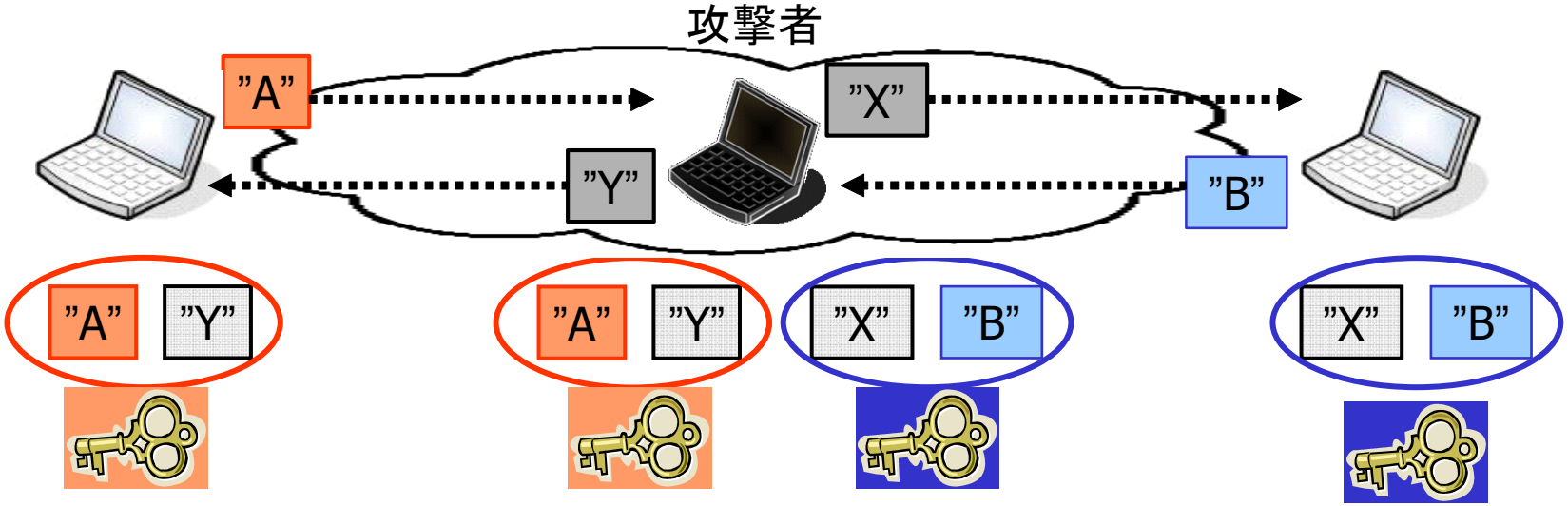
✧公開鍵暗号技術を利用



Mobile PPCにおける認証方式

- DH鍵交換
  - 盗聴には強いが中間者攻撃に対して脆弱性
- 中間者攻撃
  - 攻撃者が通信を行なう二者の間に割り込む
  - 両者が交換する公開情報をすりかえる

⇒攻撃者が共有鍵を入手



- 盗聴よりも高度な技術が必要

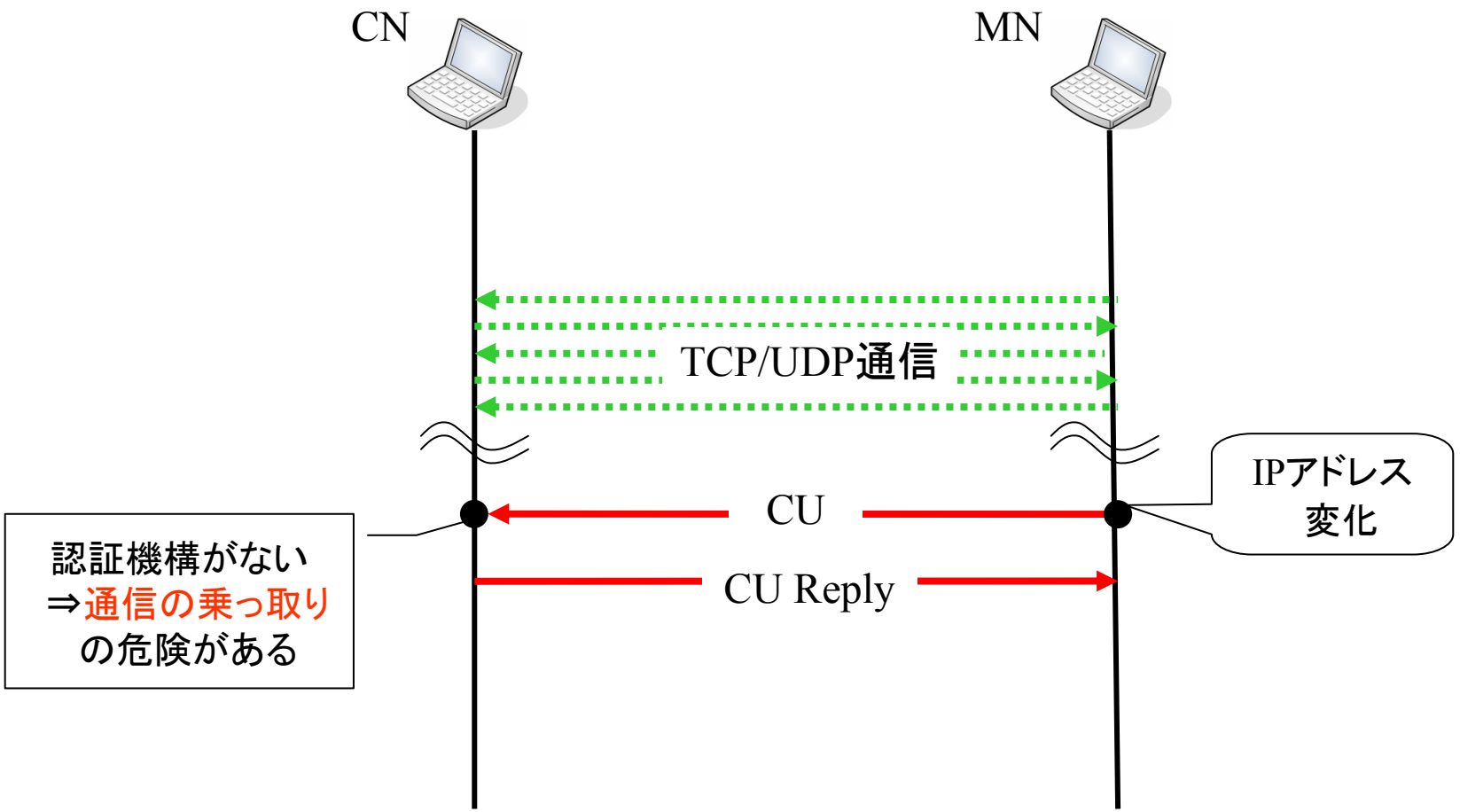
⇒実際の攻撃は極めて困難

Mobile PPCにおける認証方式

1. 端末間の通信に先立ち
   - Cookie交換
     - 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)の防止

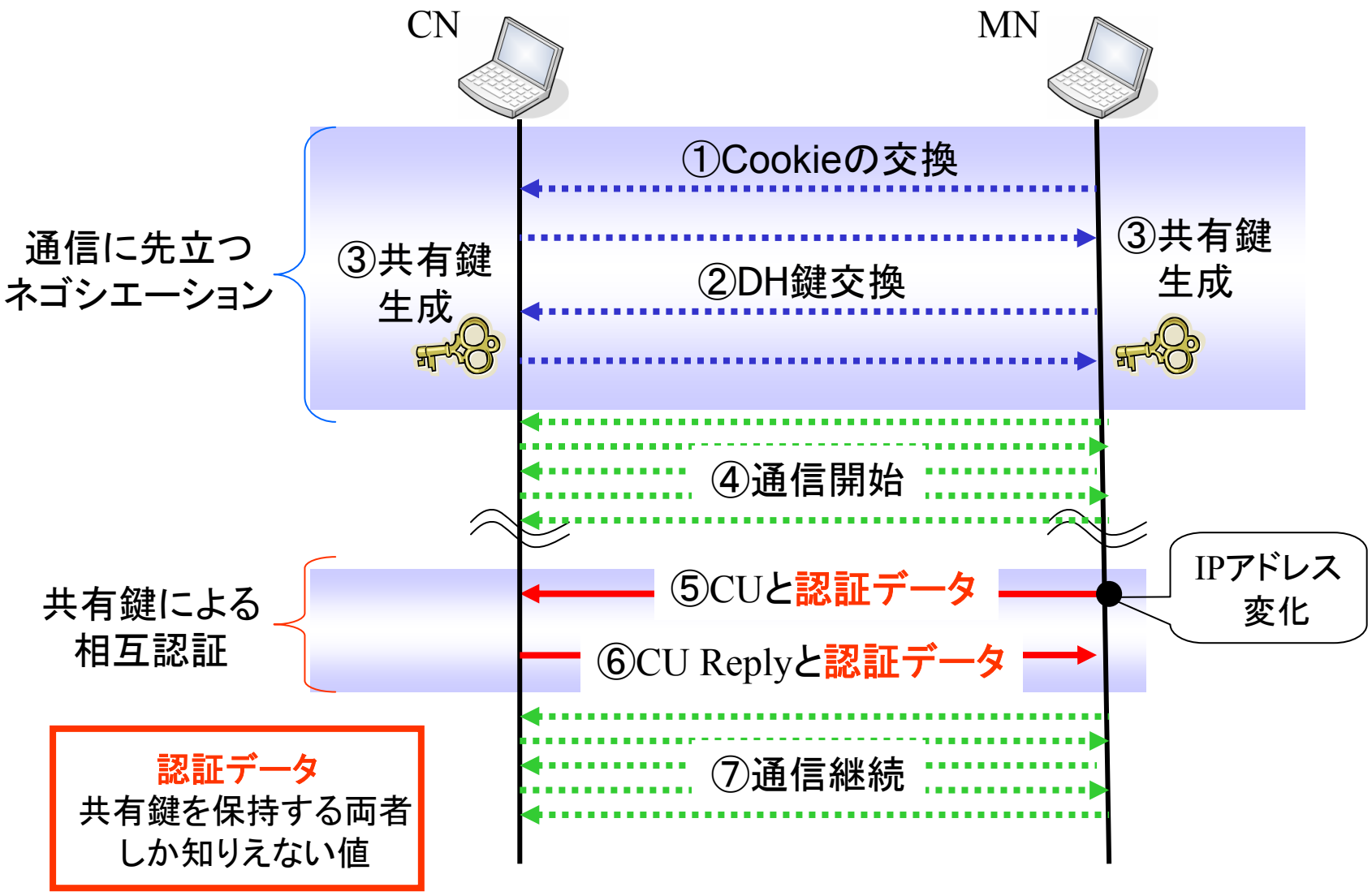
     ⇒DH演算は時間がかかるため，不要なDH演算による端末の計算資源の浪費防止.
   - Diffie-Hellman鍵交換
     - 秘密共有鍵の作成
2. MN移動時
   - 通信に先立ち生成した共有鍵による認証

従来のMobile PPCのシーケンス



Mobile PPCにおける認証方式

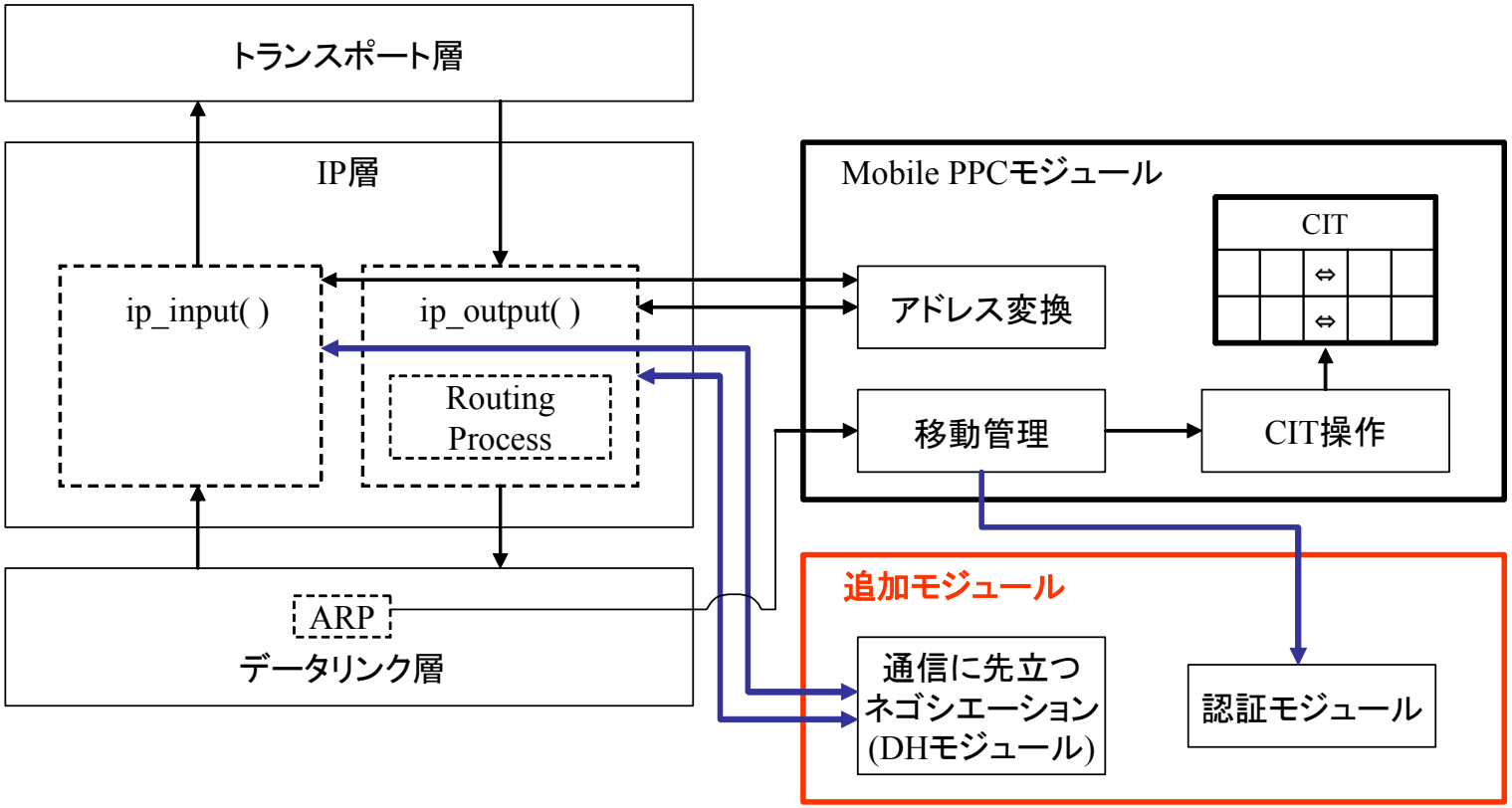
提案技術



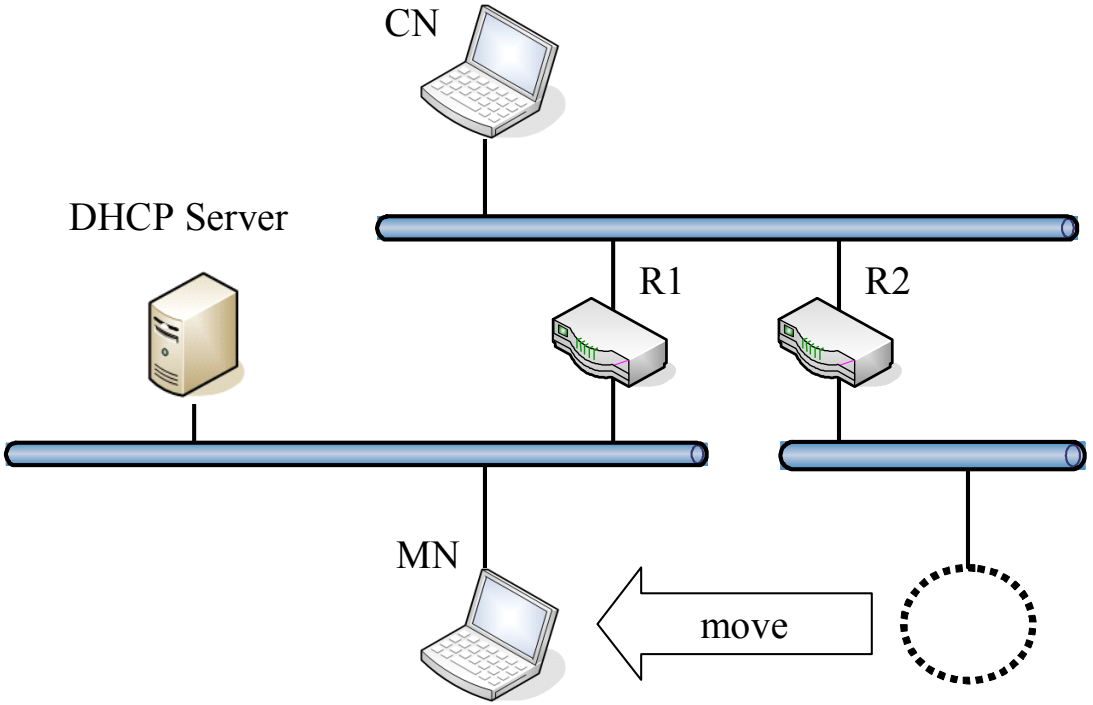
実装： モジュール構成

- 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
- 従来の Mobile PPCにモジュールを追加
  - 通信に先立つネゴシエーション（DHモジュール）

⇒DPRP(Dynamic Process Resolution Protocol)を流用



検証実験環境



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

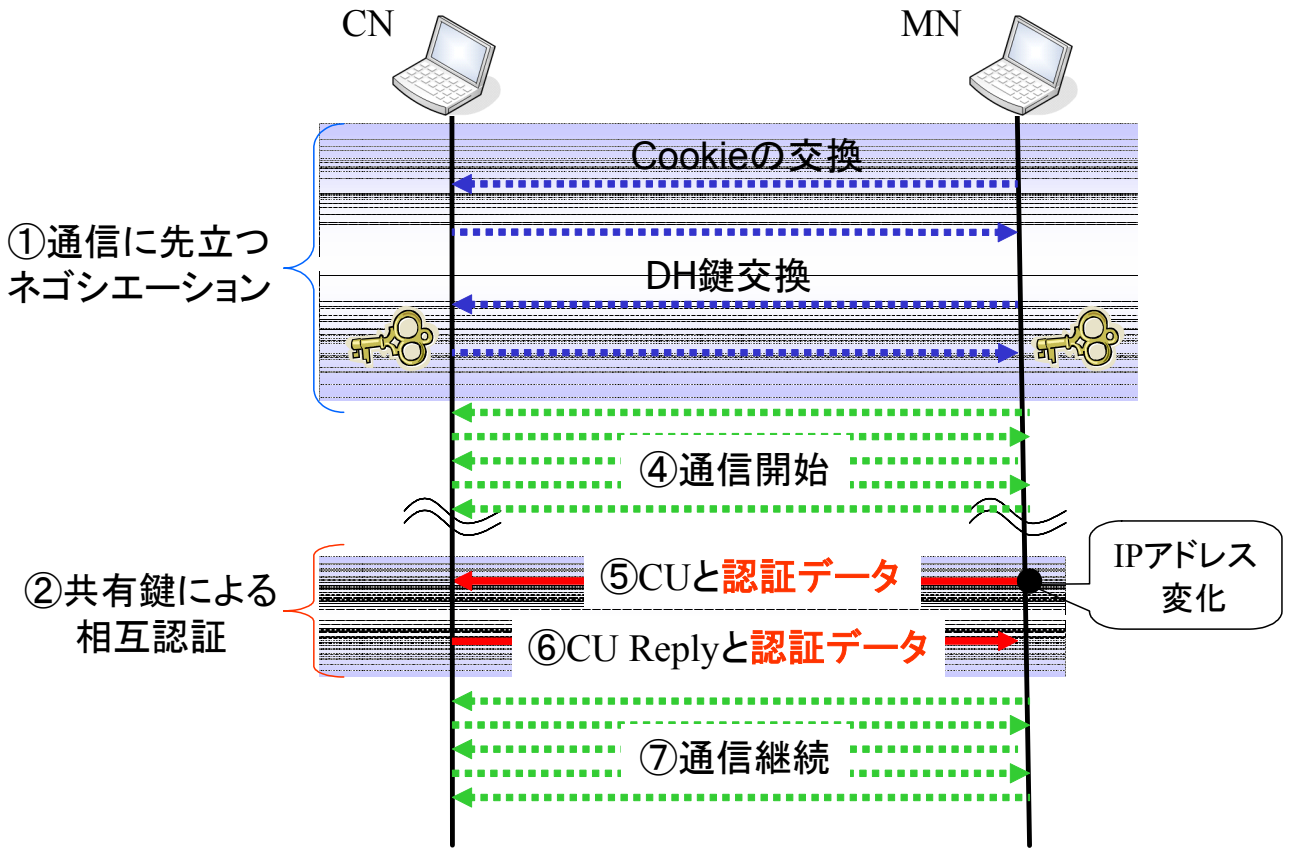
測定内容

提案方式の内部処理時間を測定

①通信に先立つネゴシエーション(cookie交換とDH交換)処理時間

②移動情報通知（CU）処理時間

- 従来のMobile PPCの処理時間と比較

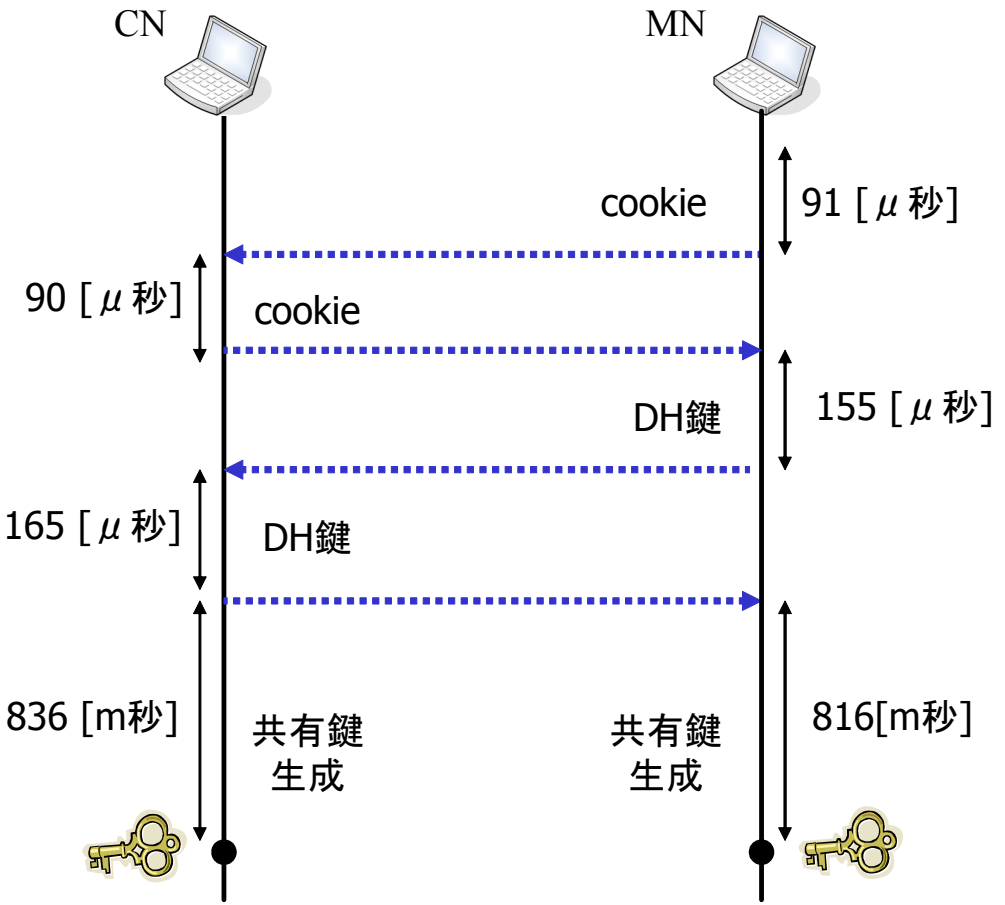


①通信に先立つネゴシエーションの処理時間

- 通信に先立つネゴシエーション(cookie交換とDH交換)処理時間

- 時間短縮の工夫
  - DH鍵は事前に生成が可能
    ⇒ 端末起動時に生成しておく
  - 共有鍵生成処理と並列して通信の開始が可能
    ⇒共有鍵生成時間は処理時間から省く

- 測定結果
  - 501μ秒
    ⇒極めて小さい



②移動情報通知(CU)処理時間

- 移動情報通知(CU)処理時間

	処理時間[μ秒]
従来のMobile PPC	644
認証処理を追加したMobile PPC	685

– 処理時間の差
  - 41μ秒(約6%の増加)

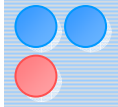
  ⇒性能はほとんど劣化していない

Return Routabilityとの比較

	Return Routabiltiy	提案方式
盗聴	△	○
中間者攻撃	○	△
第三の装置	×	○
処理時間	○	○

- 提案方式
  - セキュリティ
    - 中間者攻撃に対して脆弱性
      - 中間者攻撃には高度な技術と準備が必要

      ⇒実用上十分有効
  - 第三の装置が必要ない



むすび

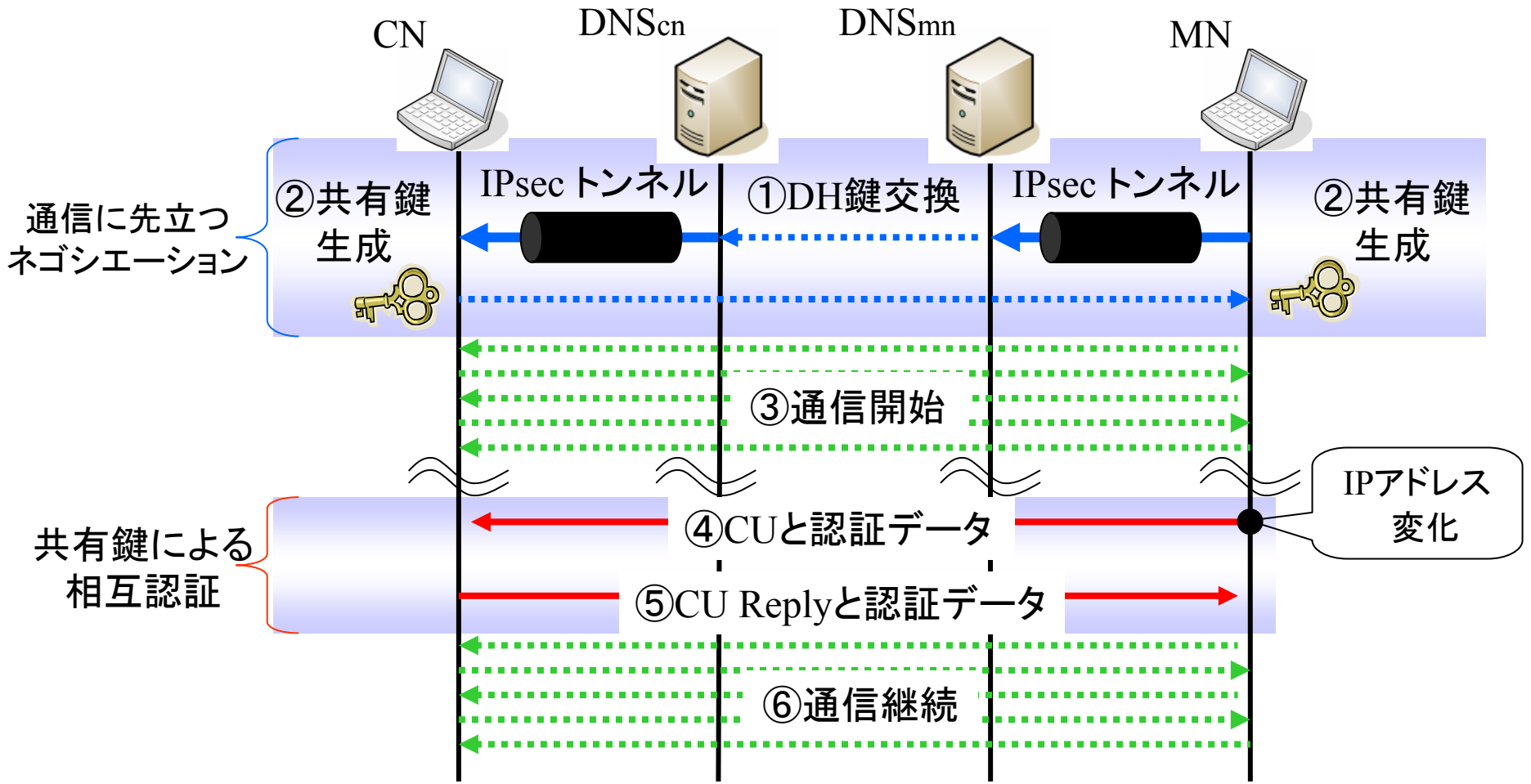
- Mobile PPCにおける認証方式の提案
  - Mobile PPCのセキュリティ課題を克服
    ⇒Mobile PPCが実使用可能
  - 第三の装置が不要で導入が容易
- 性能測定
  - オーバーヘッドはほとんどない
- 今後
  - セキュリティ強度をさらに高めた高度な認証方式を検討

付録

さらにセキュリティ強度の高い方式

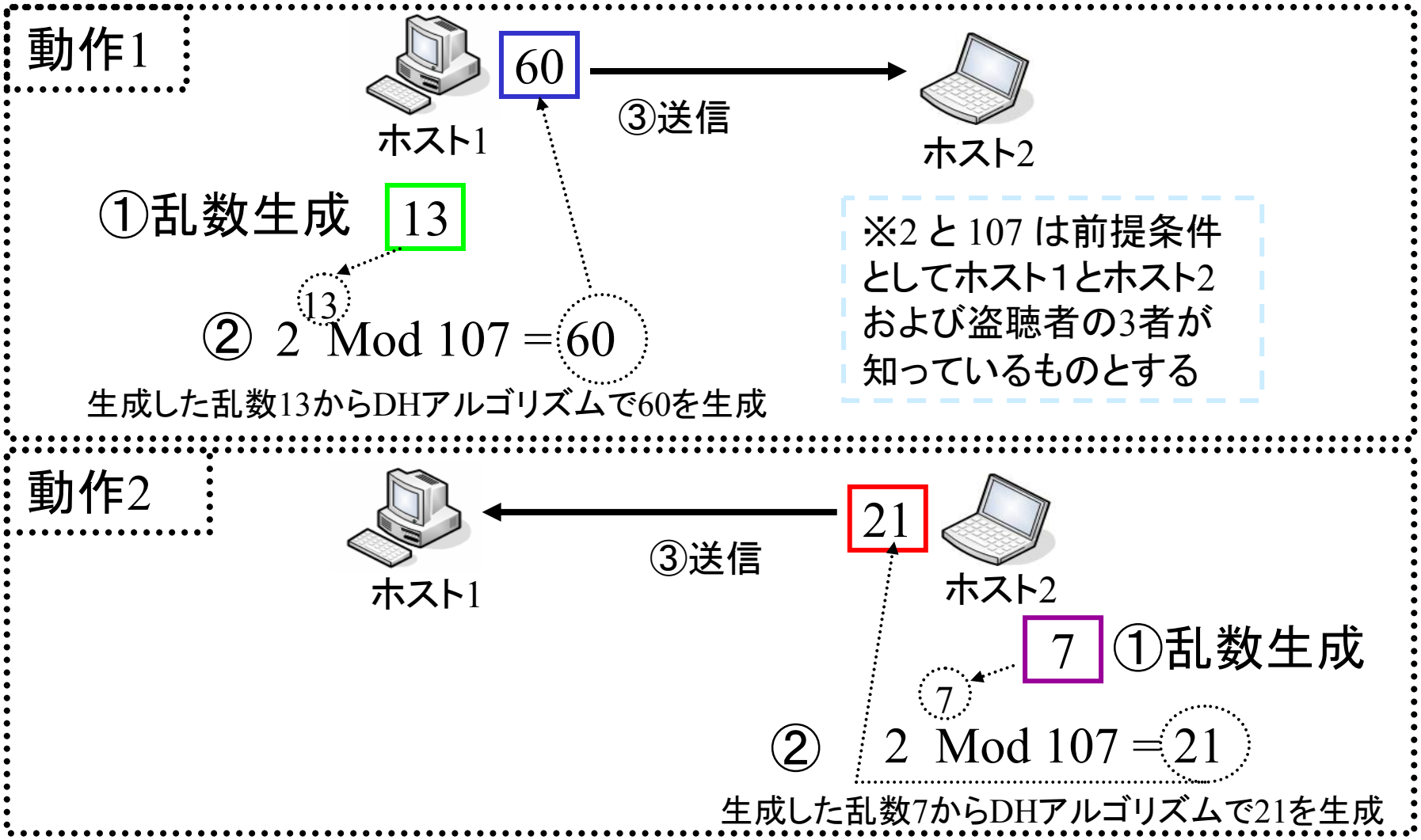
- MPPCアドバンスト認証
  - 第三の装置を利用してセキュリティ強度を強化
  - 第三の装置としてDNSサーバを利用

MPPCアドバンスト認証



- 第三の装置が必要.
- 中間者攻撃:DDNScnとDDNSmn間，CNとMN間を同時攻撃
  ⇒この攻撃の実行はほぼ不可能

Diffie-Hellman鍵交換の詳細例（その1）



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

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

※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

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

⇒この式から x , y を導き出すことは事実上不可能