情報処理学会論文誌 Vol. 50 No. 7 1725–1734 (July 2009)

分割 Diffie-Hellman 鍵交換による移動ノードの鍵共有方式の提案

瀬下正樹[†1,*1] 鈴木秀和[†1,†2]
伊藤将志[†1,*2] 渡邊晃[†1]

通信中に移動しても通信を継続できる移動透過性技術は，今後のユビキタスネットワークに必須の技術である．このとき移動後のエンドノード間で相互認証を行い，通信の横取りを防ぐことは重要な機能である．このため一般に通信に先立ち認証鍵を共有する方法がとられる．これまで認証鍵をエンドノード間で共有する方法として，乱数を 2 つの経路に分割して交換する Return Routability が提案されていた．しかし，この方法では情報が平文であるため，盗聴に対して完全な解決策とはなっていなかった．本論文では，Diffie-Hellman 鍵交換を採用し，さらにこの鍵交換を 2 つの経路に分割して実行する鍵共有方式 Split DH（Split Diffie-Hellman key sharing method）を提案する．Split DH は，盗聴による攻撃を完全に防止するとともに，より高度な攻撃となる中間者攻撃に対しても高いセキュリティを保つことができる．提案方式の基本機能を Mobile PPC（Mobile Peer-to-Peer Communication）へ実装して処理時間の測定を行った．その結果，実装の工夫により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させず実現できることが分かった．

A Proposal of Split Diffie-Hellman Key Sharing Method for Mobile Nodes

Masaki Sejimo,[†1,*1] Hidekazu Suzuki,[†1,†2]
Masashi Ito[†1,*2] and Akira Watanabe[†1]

Mobile transparency that can keep communication when a node moves during communication is an essential technology for the future ubiquitous network. In particular, secure authentication between end nodes at the time of movement is an important function. In this paper, we propose Split Diffie-Hellman key sharing method for mobile nodes (Split DH). Split DH executes Diffie-Hellman key exchange with two routes. This method provides a high level of security, not only against eavesdropping but also man-in-the-middle attacks. We implemented Split DH in Mobile Peer-to-Peer Communication (Mobile PPC) that can realize mobile transparency with only end nodes, and evaluated the system. As a result, it is shown that the overhead does not affect the performance of communication.

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ネットワークを実現するために，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，IP ネットワークではノードが通信中に移動すると 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，ノードの移動による 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さかんに行われている[1)]．

移動透過性を実現する代表技術として，プロキシサーバを経由した通信でこれを実現する 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．一般に移動透過性を実現するには，移動にかかわる情報を相手ノードに伝えるため，アドレスの変化情報を通知するシーケンスを新たに定義しなければならない．このシーケンスを利用して，攻撃者が移動ノードに成りすまして通信相手ノードへ変化情報を通知すると，セッションをハイジャックすることができる．したがって，移動時に移動ノードと通信相手ノードが確実に相互認証する必要がある．

Mobile IP（IPv4 版）では移動時の認証機構は厳密には検討されなかったが，Mobile IPv6 における認証機構は，IETF（Internet Engineering Task Force）において検討が進められた．当初は移動ノードと通信相手ノードとの間に IPsec[8)] を適用する方法が検討された．この方式では認証に使用する共通鍵を，通信に先立って IKE（Internet Key Exchange）[9)] に

†1 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
†2 日本学術振興会特別研究員 PD
Research Fellow of the Japan Society for the Promotion of Science
*1 現在，日本電気株式会社
Presently with NEC Corporation
*2 現在，株式会社東芝
Presently with Toshiba Corporation

より自動生成する．しかし，任意の装置間で IKE に必要となる認証情報を，お互いがあらかじめ保持している必要があり，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 [3] と呼ぶ認証機構が考案された．Return Routability では通信に先立って，認証鍵生成の元となる乱数を 2 つ生成し，一方をエンドノードどうしで，もう一方をホームエージェント経由で交換することによりエンドノード間で認証鍵を共有する．経路を 2 通りに分割することにより，2 カ所で同時に盗聴されない限り安全に認証鍵を共有できる．ノードの移動時には上記の認証鍵を用いて認証コードを生成することにより高速に相互認証を行うことができる．

しかし，Return Routability は通信相手ノード近傍では盗聴が困難であるという前提をおいている．通信相手ノード近傍においてはすべての通信が平文で流れるため，第三者が盗聴することにより共通鍵を生成することが可能である．このような前提は実環境上において必ずしも保証されているとはいえない．攻撃者にとって盗聴は技術的に容易であり，かつネットワーク管理者が盗聴者を発見することはきわめて困難であることから，より安全な方式が望まれる．また，今後は両エンドノードがともに移動ノードであるようなケースにおいても安全に認証鍵を共有できる手段が必要である．

文献 10) では両エンドノードがともに移動することも想定し，LIN6 に対応した認証方式を提案しているが，Return Routability と同様に特定のノード近傍での盗聴はできないという前提はそのままである．

そこで，本論文では盗聴に対しての安全性が計算量的に保証されている Diffie-Hellman（DH）鍵交換[11] を利用し，さらに DH 鍵交換を 2 つの経路に分割して実行する鍵共有方式 Split DH（Split Diffie-Hellman key sharing method）を提案する．Split DH は，盗聴による攻撃に対して計算量的に安全であるとともに，より高度な攻撃となる中間者攻撃に対しても高いセキュリティを保つことができる．一般に DH 鍵交換は認証機能がないので中間者攻撃に弱いとされているが，経路を分割することにより中間者攻撃に対しても十分高い安全性を有することが可能となる．

Split DH の基本機能を Mobile PPC に適用し，動作確認と性能測定を実施した．DH 鍵交換については，演算にかかる時間が問題とされることが多いが，実装の工夫により通信と並行して演算を実行させることにより，ほとんど通信に影響を与えずに実現が可能であることを示した．

以下，2 章で既存技術の例として Return Routability における鍵共有手順とその課題について述べる．3 章で Split DH を提案し，4 章では実装について，5 章で性能測定の結果を示す．最後に 6 章でまとめる．

2. 課題と要件

2.1 議論すべき課題

移動透過性において移動ノードの認証を実現するには，一般にエンドノード間で 2 通りのネゴシエーションを実行する必要がある．**図 1** に移動ノード（Mobile Node；以後 MN）と通信相手ノード（Correspondent Node；以後 CN）が通信している間に，MN が移動した例を示す．MN と CN は通信開始に先立ち，第 1 のネゴシエーションとなる鍵共有ネゴシエーションを実行する．このとき，Return Routability では安全に鍵を共有するため，MN と CN の直接経路と，MN が信頼できる装置 TD（Trusted Device）を中継する 2 つの経路を用いたネゴシエーションを実行する．Mobile IPv6 の場合，TD は HA（Home Agent）である．MN と TD の間は IPsec ESP [12] で暗号化する．この方法により，比較的安全に MN と CN が認証鍵 AK を共有できる．

次に MN が通信中に移動して新しい IP アドレスを取得すると，MN と CN の間で第 2 のネゴシエーションとなる移動通知ネゴシエーションを実行し，移動前後のコネクション ID の情報を交換する．移動通知ネゴシエーションではあらかじめ共有しておいた上記認証鍵 AK を用いて認証情報を生成し相互認証を高速に実行する．

移動透過性にかかわるネゴシエーションは，不特定多数の通信相手に適用できることを想

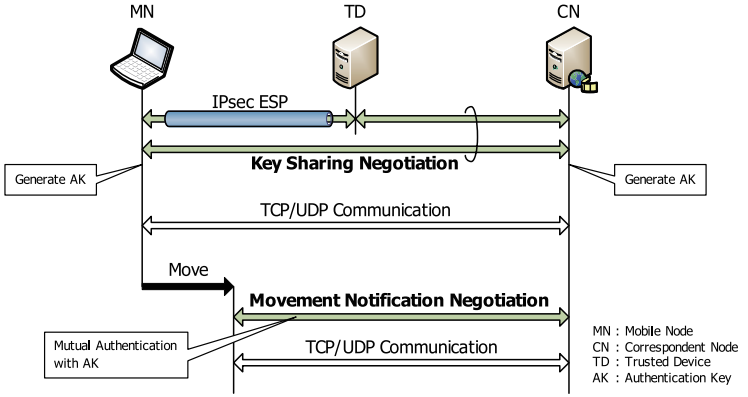


図 1　移動透過性の実現に必要となるネゴシエーション
Fig. 1　The negotiation needed for realizing mobility.

定しているため，あらかじめ通信相手の秘密情報を保持するということは想定しない．もし通信開始時点で確実な認証が必要となる場合は，適切な認証手順を別途準備する必要があり，これは移動透過性の検討範囲外とされている．したがって，鍵共有ネゴシエーションフェーズにおいては，認証鍵の漏えい防止に十分な配慮をとるが厳密な認証は定義しない．

それに対し，移動通知ネゴシエーションフェーズは，移動透過性を実現するために新たに定義したシーケンスであるため，このシーケンスを悪用したセッションのハイジャックは，移動透過性の実現にかかわる新たな脅威である．そのため，このフェーズにおける認証は，移動透過性を実現するために必須の機能とされている．図 1 に示すように，もし通信開始時に鍵共有ネゴシエーションフェーズにより認証鍵を安全に共有していれば，移動通知ネゴシエーションフェーズでの認証は容易にかつ高速に実現できる．

したがって，移動透過性のセキュリティにかかわる課題は，鍵共有ネゴシエーションのフェーズにおいて，いかに認証鍵 AK を安全に共有するかにかかっている．本論文で議論すべき課題は，鍵共有ネゴシエーションにおける安全性を既存方式に比べてさらに向上させることである．

2.2　攻撃モデルと安全性の要件

攻撃者が共通鍵を入手する代表的な方法として，盗聴と中間者攻撃がある．盗聴とは，鍵共有ネゴシエーションの情報を盗聴し，その内容から共通鍵を入手する方法である．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り，かつ盗聴者が存在することを検出することはきわめて困難である．

中間者攻撃とは，攻撃者が鍵共有ネゴシエーションの間に割り込んで，両者が交換する情報を自分のものとすりかえる方法である．中間者攻撃が成立すると，MN と CN はネゴシエーションが正常に終了したように見えるが，攻撃者と MN は偽共通鍵 FK1 を，攻撃者と CN は偽共通鍵 FK2 を共有した状態となり，攻撃者は MN と CN に気付かれることなく通信相手に成りすますことができる．中間者攻撃は，パケットのキャプチャ，解析，改ざん，送信動作などが必要であり，盗聴に比べて難易度が高い攻撃である．

安全性を高めるための要件として，第 1 に盗聴による認証鍵の漏洩が技術的に不可能であることが望ましい．Return Routability では，特定のネットワーク上では管理が厳重であるため，盗聴ができないという仮定をおいている．しかし，このような仮定は管理者が信頼できる存在であることが前提であり，管理上の責任が大きな負担になることも考えられる．そのため，安全性の要件として，原理的に攻撃が不可能な方式であることが望ましい．提案方式では DH 鍵交換を適用することによりこの要件を満たす．

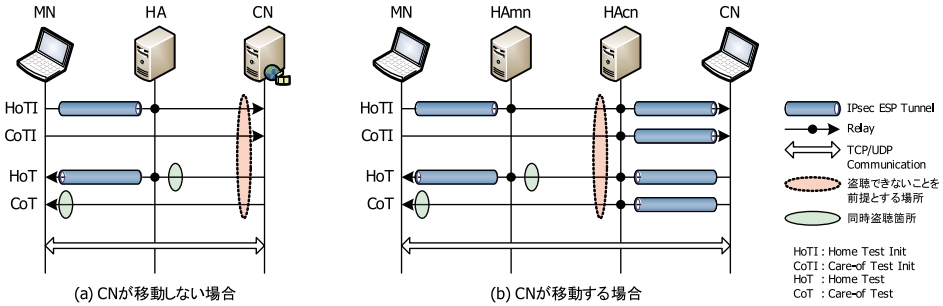


図 2　Return Routability による鍵共有ネゴシエーション
Fig. 2　Key-sharing negotiation with Return Routability.

第 2 に中間者攻撃が困難でなければならない．両エンドノードはお互い不特定多数の相手であることを仮定すると，あらかじめ認証情報を保持しておくという前提をおくことはできない．したがって，確実な認証情報がないので，中間者攻撃を完全に避けることは理論上できない．しかし，Return Routability が盗聴攻撃に対して安全性を向上させることができたように，中間者攻撃に対しても安全性を向上させることは可能である．提案方式では，DH 鍵交換を 2 つの経路を用いて実行することにより，中間者攻撃に対する安全性を向上させる．

2.3　Return Routability とその課題

ここでは，既存技術の代表として Return Routability をとりあげ，具体的な課題を述べる．Return Routability は Mobile IPv6 で検討された鍵共有ネゴシエーションの方式である．Mobile IPv6 は，IPv4 対応の移動透過性技術 Mobile IP を IPv6 用に定義しなおしたものである．このとき経路最適化機能が導入され，移動後においてもエンドエンドで最適な経路による通信が可能となった．しかし，MN の位置を管理する HA は依然として必要な装置として残されている．Return Routability では HA を信頼できる装置として積極的に利用し，安全な鍵共有ネゴシエーションを試みたものである．Return Routability の原理は移動透過性を実現するためのシーケンスとは独立しており，移動ノードにとって信頼できる装置がシステム内に存在すれば，他の移動透過性技術に適用することも可能である．

Return Routability による鍵共有ネゴシエーションの流れを**図 2** に示す．図 2 (a) は CN の位置が固定である場合，図 2 (b) は MN と CN がともに移動ノードである場合の例である．文献 3) の中で記述されている Return Routability は図 2 (a) に示す MN 側が移動し，

CN 側は固定である場合を定義している．

MN は Home Test Init（HoTI）メッセージと Care-of Test Init（CoTI）メッセージを同時に送信する．これらのメッセージには，それぞれ home init cookie および care-of init cookie と呼ばれる乱数が含まれる．HoTI は HA を経由し，CoTI は直接 CN 宛に送られる．MN にとって HA は信頼のおける装置であり，MN-HA 間は IPsec ESP による暗号化通信を行う．

CN は HoTI と CoTI の 2 つの init cookie を受信したら，home keygen token と care-of keygen token と呼ばれる値を生成する．その後，CN は Home Test（HoT）メッセージと，Care-of Test（CoT）メッセージを同時に返信する．これらのメッセージにはそれぞれ home init cookie と home keygen token，および care-of init cookie と care-of keygen token が含まれる．HoT は HA を経由して，CoT は直接経路により MN へ送信される．

MN は HoT と CoT を受信すると，home init cookie と care-of init cookie を検証し，home keygen token と care-of keygen token から認証鍵 AK を作成する．

本論文では図 2 (b) に示すように，MN と CN がいずれも移動することを想定する．そのため，MN が信頼する HAmn のほかに，CN が信頼する HAcn を導入し，MN と HAmn，CN と HAcn 間は IPsec ESP による暗号化通信を行う．MN と CN の通信に先立ち，図 2 (a) と同じ Return Routability を実行するが，MN は CN のホームアドレス宛に対してネゴシエーションを開始するため，シーケンスは図 2 (b) のように，すべてのパケットが HAcn を経由する．CN から通信を開始する場合は，この逆にすべてのパケットが HAmn を経由する．このように通信開始の方向により送信側と受信側の動きが非対象となるが，認証鍵を共有することだけが目的であり，以後の通信とは独立している．

ここで，図 2 (a) における Return Routability では，CN は管理者の管理下におかれるべき装置であり，CN 近傍（図 2 (a) の点線円部分）は盗聴ができないことを仮定している．仮に CN 近傍ですべてのパケットの内容を盗聴することができれば，攻撃者が認証鍵を導出することができる．CN 近傍の盗聴ができないという仮定が正しいとすれば，Return Routability は次の理由で安全性が高い．すなわち，攻撃を成功させようとすると，攻撃者は平文が流れる HA と CN 間，および MN と CN 間の 2 つの経路上（図 2 (a) の実線円部分）で同時に盗聴する必要がある．このような攻撃は困難であることから安全性が高いという判断がなされている．

図 2 (b) のように，MN と CN 両者が移動することを想定したシステムにおいては，CN 近傍はすべて IPsec により暗号化されるため盗聴はできない．そのかわり，HAmn-HAcn 間が平文となるため，HAcn 近傍（図 2 (b) の点線円部分）での盗聴はできないという仮定が必要となる．このような仮定は，実運用では保証されているとはいえず，2.2 節で示した安全性の要件を満たしていない．Return Routability は盗聴だけで攻撃者の目的を達成できるため，中間者攻撃に対する強度が話題になることはなかったが，中間者攻撃を防止するための条件は盗聴と同様で，図 2 の点線円部分での中間者攻撃はできないとの仮定が必要である．

3. 提案方式 Split DH

3.1 Split DH の概要

Return Routability の課題を解決するため，盗聴が技術的に不可能で，かつ中間者攻撃がきわめて困難な方式 Split DH（Split Diffie-Hellman key sharing method）を提案する．Split DH は，信頼できる装置経由と直接経路の 2 つの経路を用いて DH 鍵を分割して交換する．DH 鍵交換はネットワーク上で互いに交換する情報を盗聴されても安全に鍵を共有できる鍵共有手段として確立された技術である．しかし，DH 鍵交換自体には認証の機能がないため，中間者攻撃には弱い．また，DH 鍵演算には多くの CPU パワーが必要であるため，通信開始時に遅延が発生するという新たな課題が発生する．

そこで，本論文では単純に DH 鍵を交換するのでなく，DH 鍵を分割し，一方を信頼できる装置経由，もう一方を直接交換することにより，中間者攻撃に対してもきわめて強い方式とした．また，DH 演算時間が初期遅延に影響を与えないようにするため，実装の工夫により認証鍵の生成を通信開始と並行して実行できるようにした．Split DH は 2 往復のシーケンスで構成され，1 往復目でそれぞれの cookie と DH 鍵の前半部分を交換し，2 往復目で交換済みの cookie と DH 鍵の後半部分を交換する．1 往復目で cookie 交換を行う理由は，送信元 IP アドレスを偽造した DoS 攻撃を防止するためであり，DH 鍵交換を用いるシステムでは一般に用いられる．

3.2 DH 鍵交換アルゴリズムについて

本節では DH 鍵交換アルゴリズムの基本原理を示す．DH 鍵交換は離散対数の演算が困難であることを利用した鍵共有方式である．本節の内容は，提案方式を説明するために必要となる前提知識であり，既知の技術である．本節で使用する記号を以下に定義する．

- p：素数
- g：p の原始根（$g \in \mathbb{Z}_p^*$，$\mathbb{Z}_p^*$ は乗法群）
- R_i：ノード i が生成した使い捨て乱数（$R_i \in \mathbb{Z}_{p-1}$）

- DH_i：ノード i が生成した DH 鍵（$DH_i = g^{R_i} \bmod p$）
- CK：共通鍵

p と g はシステム共通のパラメータとして各ノードがあらかじめ固定値として保持しておく．MN と CN はそれぞれ乱数 R_{mn}，R_{cn} を生成し，さらに DH 交換鍵 DH_{mn}，DH_{cn} を以下のように生成する．

$$DH_{mn} = g^{R_{mn}} \bmod p \quad \text{（MN 側）} \tag{1}$$

$$DH_{cn} = g^{R_{cn}} \bmod p \quad \text{（CN 側）} \tag{2}$$

MN と CN は，DH_{mn} と DH_{cn} をネットワーク上で交換する．これを受信した各ノードは以下の演算を行う．両者の演算結果は等しくなり，共通鍵 CK を共有することができる．

$$CK = \begin{cases} {DH_{cn}}^{R_{mn}} \bmod p = g^{R_{cn} \cdot R_{mn}} \bmod p & \text{（MN 側）} \\ {DH_{mn}}^{R_{cn}} \bmod p = g^{R_{mn} \cdot R_{cn}} \bmod p & \text{（CN 側）} \end{cases} \tag{3}$$

攻撃者が知ることができる情報は DH_{mn}，DH_{cn}，p，g である．攻撃者にとってこれだけの情報から CK を求めることは，p が十分大きければ離散対数問題により計算量的に困難である．

3.3　Split DH による鍵共有の流れ

本節では，提案方式 Split DH による鍵共有の流れを説明する．本節で用いる記号を以下に定義する．

- C_i：ノード i が生成した cookie（十分大きい乱数）
- AK：共通鍵と cookie から生成した認証鍵
- $x \mid y$：データ x と y の連結
- $FH(x)$：データ x の前半部分（Former Half）
- $LH(x)$：データ x の後半部分（Latter Half）
- $h(x)$：データ x のハッシュ値
- $MSGj$：j 番目の交換メッセージ

Split DH による鍵共有ネゴシエーションを**図 3** に示す．交換する DH 鍵（DH_{mn} と DH_{cn}）は新たな通信が開始される前に事前に生成しておく．ここでは適用する移動透過性技術として Mobile PPC をとりあげる[⋆1]．

Mobile PPC はエンドノードのみで移動透過性を実現できるため，Mobile IP における

⋆1 Mobile PPC の原理は付録を参照．

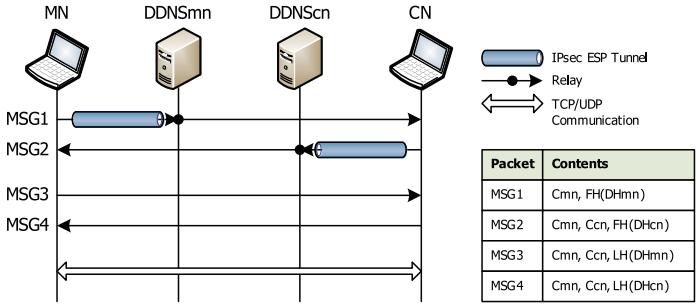


図 3　Split DH による鍵共有ネゴシエーション
Fig. 3　Key-sharing negotiation method with Split DH.

HA に相当する装置がない．そこで Mobile PPC では，MN が自身の位置を登録するために使用する DDNS（Dynamic DNS）[13] サーバを信頼できる装置として利用する．図 3 における DDNSmn と DDNScn はそれぞれ，MN と CN の位置を管理する DDNS サーバである．MN と DDNSmn 間，CN と DDNScn 間は信頼関係を期待できるものとし，この区間では IPsec ESP トンネルモードによる通信を行う．Mobile PPC では，Mobile IP で使用するホームアドレスが不要であり，MN，CN ともに相手のアドレスが直接の通信相手となる．

図 3 に示すように，MN は CN との通信に先立ち Split DH を実行する．MN は cookie（C_{mn}）を生成し，事前に生成しておいた DH 鍵（DH_{mn}）の前半部分 $FH(DH_{mn})$ とともに DDNSmn を経由して CN へ送信する．MN と DDNSmn 間は IPsec ESP でカプセル化される．

$$MSG1 = C_{mn} \mid FH(DH_{mn}) \tag{4}$$

CN はこれを受信すると，自身の cookie（C_{cn}）を生成し，あらかじめ生成しておいた DH 鍵（DH_{cn}）の前半部分 $FH(DH_{cn})$ とともに DDNScn を経由して MN へ返信する．CN と DDNScn 間は IPsec ESP でカプセル化される．

$$MSG2 = C_{mn} \mid C_{cn} \mid FH(DH_{cn}) \tag{5}$$

MN は $MSG2$ を受信すると C_{mn} を検証し，C_{mn}，C_{cn}，および DH 鍵（DH_{mn}）の後半部分 $LH(DH_{mn})$ を CN へ直接送信する．

$$MSG3 = C_{mn} \mid C_{cn} \mid LH(DH_{mn}) \tag{6}$$

CN はこれを受信すると C_{cn} を検証し，C_{mn}，C_{cn}，および DH 鍵（DH_{cn}）の後半部分

表 1　Return Routability と Split DH の比較
Table 1　Comparison between Return Routability and Split DH.

方式	前提条件	盗聴	中間者攻撃
Return Routability（Mobile IPv6）	HA 近傍で不正ができない	△（2 カ所）	△（2 カ所）
Split DH（Mobile PPC）	なし	○	△（2 カ所）

$LH(DH_{cn})$ を MN へ直接返信する．

$$MSG4 = C_{mn} \mid C_{cn} \mid LH(DH_{cn}) \tag{7}$$

MN は $MSG4$ を受信すると，C_{mn} を検証する．以上で両ノード間の DH 鍵の交換が完了する．MN と CN は受信した DH 鍵により，式 (3) を用いて共通鍵 CK を生成する．次に生成した CK と C_{cn}，C_{mn} のハッシュ値を求め，最終的な認証鍵 AK を生成する．

$$AK = h(CK \mid C_{mn} \mid C_{cn}) \tag{8}$$

DH 鍵交換は高い CPU 負荷を必要とするため DoS 攻撃に弱い．DoS 攻撃は一般に送信元アドレスを偽造し攻撃者の位置を隠蔽するのが一般であるが，アドレスを偽造するとクッキー交換ができない．そのため，クッキー交換は DH 鍵交換に対する DoS 攻撃を防止するために有効な手段であり[9]，Split DH にもこの考えを取り入れた．

3.4　安全性に関する考察

本節では Retuen Routability と Split DH の安全性を比較する．**表 1** に両者の比較を示す．Return Routability は HA 近傍は管理が厳しいため，不正ができないという条件がある．ここでいう不正とは，盗聴と中間者攻撃の両者を含む．この条件においては，攻撃者は同時に 2 カ所でネットワークを盗聴しなければ共通鍵を盗むことはできない．これは技術的に困難といえる．しかし，理論的に不可能ではないため，表 1 での評価は △ とした．中間者攻撃についても同様に 2 カ所で同時に実行する必要があるため △ とした．ただし攻撃者は盗聴だけで目的を達成できるため，より難易度の高い中間者攻撃を行う必要はない．

これに対し，Split DH には Return Routability のような前提条件は不要である．また，複数のネットワークを同時に盗聴されても理論的に安全である．その理由は，盗聴情報から鍵を導出するには，離散対数問題を多項式時間で解くことができないという計算量理論に基づき証明されている．中間者攻撃については，2 往復のシーケンスの間に入って，2 往復のパケットの内容をすべてすり変える必要がある．これには，少なくとも MN 近傍と CN 近傍の 2 カ所で同時に連携をとりながら実行する必要がある．これは技術的にきわめて困難であるが，理論的に不可能ではないため評価は △ とした．

Split DH において，MN と CN がもし同一ネットワークに存在すると，攻撃者は 1 カ所で中間者攻撃を実行できる．ただし，MN と CN は不特定多数であることを想定するため，多くの場合は別のネットワークに存在すると考えてよい．Retuen Routability においても MN と HAmn が同一のネットワークに存在すると，盗聴は 1 カ所でよいため Split DH との相対的な評価結果は変わらない．

3.3 節では Mobile PPC を用いて Split DH の原理を説明したが，他のエンドエンド型移動透過技術にも適用可能な方式である．ただし，Mobile IPv6 においては通信開始時がエンドエンドではないため課題が残る．たとえば MN から通信を開始する場合に必ず HAcn を経由する必要があるため，攻撃者は HAcn 近傍の 1 カ所だけで中間者攻撃を行うことができる．したがって，Split DH は Mobile PPC のように通信開始時に MN と CN ともに相手のアドレスが直接の通信相手となる方式に有効である．

4. 実　　装

DH 鍵交換は処理負荷が高いため，通信に影響を与えることが懸念される．DH 鍵の生成は通信が開始される前にあらかじめ生成しておくことができる．したがって，Split DH において最も性能に影響を与える可能性があるのは，DH 鍵を用いた共通鍵 CK の演算が通信開始に与える初期遅延時間であり，式 (3) にあたる部分である．そこで実装上の工夫として，共通鍵 CK の生成を通信の開始と並行して実行することとした．この方法により DH 演算による初期遅延の影響を最小限に抑えることができる．

上記方針に従って Split DH の基本機能を試作し，動作検証と性能評価を行った．すでに FreeBSD 5.2.1-RELEASE 上で開発済みの Mobile PPC モジュールに，Split DH による鍵共有ネゴシエーション機能と，移動通知ネゴシエーションの認証機構を追加することにより実現した．今回の評価では Split DH の検証と DH 鍵生成にかかわる性能評価が目的であるため，Split DH の 1 往復目で DDNS サーバを経由することはせず，直接 MN と CN でシーケンスを交換することとした．

図 4 に Mobile PPC のモジュール構成と Split DH による追加内容を示す．図 4 における CIT（Connection ID Table）は，IP 層で保持する新旧コネクション ID の対応関係を保持したテーブルである．Mobile PPC モジュールはパケット送受信時には IP 入力関数 `ip_input()` から，パケット送信時には IP 出力関数 `ip_output()` から呼び出され，CIT の参照およびアドレス変換処理を行う．通信中に IP アドレスが変化すると，DHCP サーバからアドレス取得後に Gratuitous ARP による二重アドレスチェックを行う．この処理の終了をトリガとして，ARP 関数から Mobile PPC の移動管理処理が呼び出され，移動通知

ネゴシエーションと CIT の更新を行う．呼び出し以外のカーネルの処理にはいっさい変更を加えずに実現されている．

今回 Split DH のために新たに実装したのは，NIT（Node Information Table）操作モジュール，DH 鍵交換モジュール，DH 演算モジュール，認証モジュールである．アドレス変換モジュールと移動管理モジュールに修正を加えることにより，上記追加モジュールを呼び出せるようにした．NIT は今回の認証機構のために新たに定義したテーブルであり，cookie 生成に関する情報や生成された認証鍵などを通信相手ノードごとに記録する．NIT 操作モジュールは，NIT レコードの生成・更新・検索を行う．ノードは通信開始時に，送信元/宛先 IP アドレスをキーとして NIT 検索を行い，必要であれば Split DH を実行する．DH 鍵交換モジュールは MSG1〜4 の送受信時に呼び出され，cookie の生成や検証，共通鍵，認証鍵の生成などを行う．ハッシュ関数には，MD5 を使用した．鍵共有ネゴシエーションは，今回の実装では同様な処理をすでに実現している文献 14) による成果を流用して実現した．DH 演算モジュールは OpenSSL [15] を利用してアプリケーション層に実装し，データ通信と並行して処理を実行できるようにした．認証モジュールは Mobile PPC の移動通知ネゴシエーションに用いる CU（CIT Update）Request/Response メッセージに対して，認証鍵 AK を用いた MAC（Message Authentication Code）の生成・付加・検証を行う．MAC の生成に用いる鍵付きハッシュ関数には，HMAC SHA1 を使用した．なお，認証鍵の生成中にノードが移動した場合は，認証鍵の生成が完了した時点で移動通知ネゴシエーションを実行することとした．DH 交換で使用する素数 p のサイズは 1,024 ビットとした．

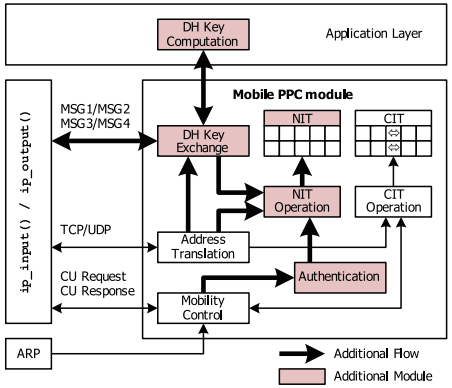


図 4　モジュール構成
Fig. 4　Module structure.

5. 性能評価

5.1 測定環境

通信に先立つネゴシエーション時間と移動情報通知時の認証処理時間の測定を行った．測定環境を**図 5** に，ルータおよびエンドノードの仕様を**表 2** に示す．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．有線 LAN は 100BASE-TX で構成し，ネットワークの移動は LAN ケーブルをつなぎなおすことでエミュレートした．MN から CN へ連続的に FTP によるデータ転送を実行させておき，MN を別のネットワークに移動させ，MN 側で直接コマンドを入力することにより，DHCP サーバから新しく IP アドレスを取得させた．

5.2 鍵共有ネゴシエーションの時間

Split DH による鍵共有ネゴシエーションにおいて，MN 側で測定したモジュール処理時間を**図 6** に示す．カーネルモジュールの処理時間の測定には，RDTSC（Read Time Stamp Counter）[16] を用いた．MN は最初の通信パケットをカーネル内で一時待避しておき，鍵共

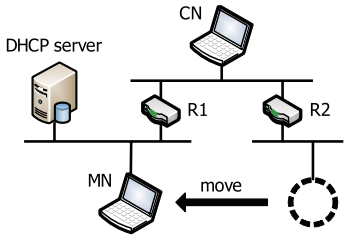
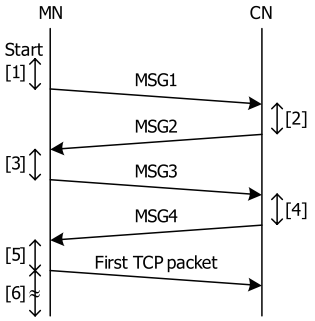


図 5　測定環境
Fig. 5　Measurement environment.

表 2　装置仕様
Table 2　Device specification.

	仕様
CPU	Pentium4 3.0 GHz
Memory	512 MB
NIC	100BASE-TX
OS	FreeBSD 5.2.1-RELEASE



測定箇所	処理内容	処理時間
[1] Kernel	TCP/UDPパケット待避, Cookie生成, MSG1生成	91 μs
[2] Kernel	Cookie生成, MSG2生成	90 μs
[3] Kernel	Cookie検証, MSG3生成	155 μs
[4] Kernel	Cookie検証, MSG4生成	165 μs
[5] Kernel	Cookie検証, パケット復帰	127 μs
[6] Userland	AK生成	72.5 ms

図 6　Split DH の鍵共有ネゴシエーション処理時間
Fig. 6　Processing time in Split DH negotiation.

有ネゴシエーションを開始する．ネゴシエーション完了後，待避していた通信パケットを復帰させ通信を開始する．共通鍵 CK および認証鍵 AK の生成はネゴシエーション終了後，通信の開始と並行してユーザランドで実行する．この方法により，図 6 に示すように，通信開始時に発生する初期遅延はモジュール処理時間の 628 μs[⋆1] と，MSG1〜4 がルータ R2 を中継する 2 往復の RTT（Round Trip Time）の合計となる．モジュール処理時間は，通信開始時に発生するオーバヘッドとしてはほとんど無視できる値である．

実際のインターネットでは RTT が大きくなるため，その分初期遅延は大きくなる．さらに，MSG1，MSG2 は DDNS サーバを経由するため，そのための中継時間と IPsec ESP によるカプセル化時間がさらに必要である．しかし，これらの処理に DH 鍵を用いた公開鍵演算処理は含まれないので，十分小さな値におさまるものと判断できる．そのため，最初の通信パケットを一時待避している間に TCP コネクションのタイムアウトが発生するようなことはない．

なお，通信と並行して実行した認証鍵の生成処理（式 (3) の公開鍵演算処理）の時間を測定したところ，MN 側で 72.5 ms の時間を要していた．

5.3　移動通知ネゴシエーション時間

試作では，鍵共有ネゴシエーションとともに，Mobile PPC の移動通知ネゴシエーションの改造も行った．すなわち，AK を用いて CU Request/Response の MAC を生成し，これによる認証処理を追加した．そこで，MN を移動させて新 IP アドレスを取得した後，CU Request/Response の交換を終え，CIT の更新が完了するまでの時間を，MAC による認証がない場合とある場合で比較した．

⋆1 $91 + 90 + 155 + 165 + 127 = 628\,\mu$s

表 3　移動通知ネゴシエーション時間
Table 3　Negotiation time of move information.

	処理時間（μs）
MAC による認証なし	644
MAC による認証あり（提案方式）	707

図 5 の環境において，MN 側の処理時間を測定した結果を**表 3** に示す．この時間はルータ R1 を中継する RTT の時間を含んでいる．Mobile PPC の移動通知ネゴシエーション時間は，AK による認証を行わない場合は 644 μs，AK による認証を行った場合は 707 μs となり，認証処理を加えたことによる増加率は約 10%となった．このように，Mobile PPC の移動通知ネゴシエーションは，認証処理を含めてもきわめて高速に実現できることを確認した．

6. ま と め

本論文では Diffie-Hellman 鍵を分割して交換することによる鍵共有ネゴシエーション方式 Split DH を提案した．Split DH は通信に先立ち信頼できる装置を用いて，DH 鍵を複数の経路に分割して交換する．この方式は，盗聴はもちろんのこと，中間者攻撃に対しても十分高いセキュリティ強度を有している．Split DH の原理は，他のエンドエンド型移動透過技術にも適用可能である．

Split DH の基本機能を Mobile PPC へ適用し，鍵共有ネゴシエーションによる共通鍵の生成処理，および移動時における移動通知ネゴシエーションの認証処理が正常に動作することを確認した．処理時間の測定を行った結果，認証鍵生成を通信と並行して行わせることにより，DH 演算による通信開始時の遅延はほとんどなくすことができた．また，移動時の認証処理は高速に実現できることが分かった．

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（D-I），Vol.J87-D-I, No.3, pp.308–328 (2004).
2) Perkins, C.: IP Mobility Support for IPv4, RFC 3220, IETF (2002).
3) Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, RFC 3775, IETF (2004).

4) Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H. and Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks, *IEICE Trans. Communications*, 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) Kent, S. and Seo, K.: Security Architecture for the Internet Protocol, RFC 4301, IETF (2005).
9) Harkins, D. and Carrel, D.: The Internet Key Exchange (IKE), RFC 2409, IETF (1998).
10) 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，電子情報通信学会論文誌，Vol.J87-D-I, No.5, pp.497–507 (2004).
11) Diffie, W. and Hellman, M.: New Directions in Cryptography, *IEEE Trans. Information Theory*, Vol.IT-22, No.6, pp.644–654 (1976).
12) Kent, S.: IP Encapsulating Security Payload (ESP), RFC 4303, IETF (2005).
13) Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).
14) 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).
15) The OpenSSL Project: The Open Source toolkit for SSL/TLS. http://www.openssl.org/
16) Intel Corp.: *Using the RDTSC Instruction for Performance Monitoring* (1998). http://developer.intel.com/drg/pentiumII/appnotes/RDTSCPM1.htm

付　　録

A.1 Mobile PPC

Mobile PPC[7] は，エンドエンドで移動透過性を実現する方式であり，移動ノード到達性と通信継続性を明確に分離した点に特徴がある．移動ノード到達性には DDNS（Dynamic DNS）サーバ[13] を利用し，通信継続性にかかわる機能を Mobile PPC で実現する．

図 7 に Mobile PPC の基本処理を示す．MN と CN は DDNS サーバを利用して通信相手の名前解決後，IP アドレス $G1$ と $G3$ で通信を開始するものとする．MN と CN は IP 層内にアドレス変換テーブル CIT（Connection ID Table）を保持している．CIT はパケットの IP アドレスの移動前と移動後の対応関係を記録したテーブルである．通信開始時点では，CIT の内容は以下のようになっており，MN と CN 間で確立したセッション情報が記録される．

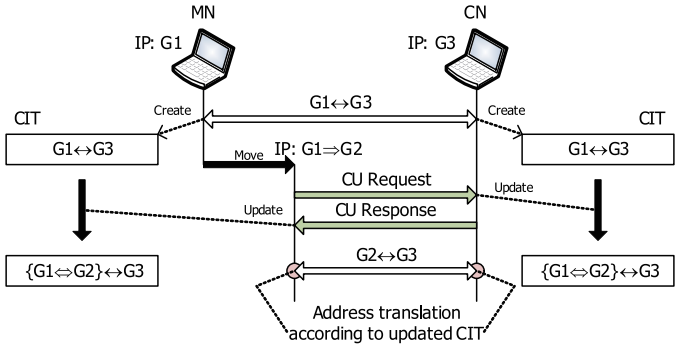


図 7　Mobile PPC の基本処理
Fig. 7　Basic operation of Mobile PPC.

$$G1 \leftrightarrow G3 \tag{9}$$

なお，MN の移動前はアドレス変換を行わない．

MN が通信中に移動して，IP アドレスが $G1$ から $G2$ に変わると，MN と CN は CU Request/Response によるネゴシエーションを行い，CIT を

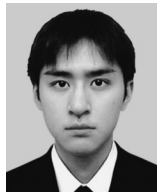
$$\{G1 \Leftrightarrow G2\} \leftrightarrow G3 \tag{10}$$

のように更新する（⇔ は変換関係を示す）．

CIT 更新後は全パケットに対し，MN と CN はそれぞれ IP 層において，CIT に基づきアドレス変換を行う．CN 側で上位層から送信元 $G3$，宛先 $G1$ として送信指示されたパケットが，IP 層において宛先が $G2$ に書き換えられてネットワーク上に送信される．MN 側は IP 層において，受信したパケットの宛先を $G2$ から $G1$ に書き換えてから上位層へ渡す．

これにより，IP 層以下においては実際の IP アドレスによる通信となり，上位層は MN の IP アドレスが変化していないように見えるため，移動透過性を実現できる．

(平成 21 年 1 月 5 日受付)
(平成 21 年 4 月 3 日採録)



瀬下　正樹（正会員）

2005 年名城大学理工学部情報科学科卒業．2007 年同大学大学院理工学研究科情報科学専攻修了．同年日本電気株式会社入社．モバイルネットワーク事業本部に所属．修士（工学）．

鈴木　秀和（正会員）

2004 年名城大学理工学部情報科学科卒業．2006 年同大学大学院理工学研究科情報科学専攻修了．2009 年同大学院理工学研究科電気電子・情報・材料工学専攻博士後期課程修了．2008 年より日本学術振興会特別研究員．ネットワークセキュリティ，モバイルネットワーク，ホームネットワーク等の研究に従事．博士（工学）．2006 年 IEEE 名古屋支部学生奨励賞受賞．2006 年 DICOMO 松下温賞受賞．2007 年情報処理学会東海支部学生論文奨励賞受賞．2007 年，2008 年 DICOMO ヤングリサーチャ賞受賞．電子情報通信学会，IEEE 各会員．

伊藤　将志（正会員）

2004 年名城大学理工学部情報科学科卒業．2006 年同大学大学院理工学研究科情報科学専攻修了．2009 年同大学院理工学研究科電気電子・情報・材料工学専攻博士後期課程修了．同年株式会社東芝入社．VoIP，無線ネットワーク等の研究に従事．博士（工学）．2008 年情報処理学会東海支部学生論文奨励賞受賞．2008 年 DICOMO 優秀プレゼンテーション賞および優秀論文賞受賞．電子情報通信学会会員．

渡邊　　晃（正会員）

1974 年慶應義塾大学工学部電気工学科卒業．1976 年同大学大学院工学研究科修士課程修了．同年三菱電機株式会社入社後，LAN システムの開発・設計に従事．1991 年同社情報技術総合研究所に移籍し，ルータ，ネットワークセキュリティ等の研究に従事．2002 年名城大学理工学部教授，現在に至る．博士（工学）．電子情報通信学会，IEEE 各会員．