ネットワーク処理性能を考慮したMTD手法の設計

田邊優人[1] 前田香織[1] 大石恭弘[2] 高野知佐[1]

概要：静的なIPアドレスの付与やネットワーク構成はDDoSなど様々な攻撃に晒される．また，多くのプロトコルに見られる最短経路によるルーティングは盗聴など中間者攻撃に脆弱である．これらの対策としてMTD(Moving Target Defense)がある．MTDは保護対象の識別子を予測不能なものにすることによって攻撃の対象となる確率を減少させる防御技術である．本稿では，既存研究である移動透過通信アーキテクチャMATをMTDに用いるMAT MTDを改良したMTDシステムの開発について述べる．開発したMTDシステムでは複数の伝送路を用いて通信を行うルートホッピングを追加し，伝送路上の中間者攻撃からの防御機能の強化する．また，MTDシステムを使うことで，サーバのネットワーク処理性能が低下しないように高速パケット処理機構であるDPDK(Data Plane Development Kit)を用いる．ルートホッピングによるオーバヘッドを調べるために，通信性能差のあるネットワーク間をルートホッピングする場合のスループットを測定実験で行いその結果も述べる．

Moving Target Defense Method with Consideration of Network Processing Performance

YUTO TANABE[1] KAORI MAEDA[1] YASUHIRO OHISHI[2] CHISA TAKANO[1]

1. はじめに

現在，インターネット上のサーバの多くは静的にIPアドレスが付与されており，インフラストラクチャでは伝送路などのネットワーク構成は静的なものがほとんどである．静的なIPアドレスは偵察やDDoSなどの攻撃に対して脆弱であり，攻撃者に大きな優位性がある．また，多くの通信プロトコルでは，経路選択は最短経路に基づいて行われており，伝送路上のデータへの攻撃もしやすい状況である．

攻撃の複雑さや難易度を高めることは保護対象の被攻撃確率を低減させることとなり，有効な防御手段である．特にインターネット上のサーバにおいて，被攻撃確率を低減させる目的としてパケットの伝送経路やIPアドレス，ポート番号といった識別子などを予測不能なものに変更するMTD(Moving Target Defense)[1]の研究が活発に議論されている．著者らもIPアドレスを予測不能なものに変更してサーバを防御するMAT MTD[2]を提案，実装している．

本研究ではMAT MTDに伝送路上のパケット傍受など中間者攻撃に対する防御機能を追加し，MTDの防御機能の強化とサーバのネットワーク処理性能が低下しないMTDシステムの実装を目指す．

MAT MTDの処理負荷に加えて追加のMTD手法によるサーバ処理性能の低下が想定されるため，MAT[3]の実装に高速パケット処理機構であるDPDK(Data Plane Development Kit)[4]を用いたDPDK-MAT[5]を使用する．

2. 関連研究

2.1 MTD(Moving Target Defense)

MTD(Moving Target Defense)[1]とは，保護対象の識別子を予測不能なものに変更することで，攻撃を困難にし，被攻撃確率を減少させるセキュリティ技術である．特にネットワークにおいて識別子とはIPアドレスやポート番号などを指し，伝送路など識別可能なものも含まれる．本稿では識別子を変更することをホッピングと呼ぶ．

IPアドレスホッピングには，サーバとクライアントの間にゲートウェイを設置してそこでアドレス変更するゲートウェイ型と，サーバやクライアントそのものでアドレス変更するエッジ型がある．本稿で提案するMTD方式はエッジ型である．エッジ型の既存研究としてMT6D[6]ではIPv6アドレスのインタフェース識別部の生成に疑似乱数を用い，これを定期的に変更する．サーバとクライアント間の通信にUDPパケットによるカプセル化を行い，ヘッダ部のIPアドレスを変更することでアドレスホッピングを実現している．MT6Dではカプセル化によるトンネルのオーバヘッドが生じる上，マルチホームでのホッピングができない．また，アドレスホッピング時の通信途絶に対する対策がなく，盗聴など中間者（Man-In-The-Middle）攻撃に対する防御もできない．通信品質劣化の問題を解決するためモビリティ機構による通信である，IPアドレスホッピングが提案され，MTM6D[7]やMAT MTD[2]がある．これについては次節で詳述する．

識別子として伝送路が変わるルートホッピングとして

1 広島市立大学大学院情報科学研究科
2 株式会社インターネットイニシアティブ(IIJ)

DHC(Double Hopping Communication)[8]が提案されている．DHC は SDN 上で転送されるパケットの伝送経路やパケットに含まれる IP アドレスやポート番号を動的に変更することで盗聴攻撃や特定のフローに対する DoS 攻撃を防ぐ MTD 手法である．DHC は伝送路上のデータ保護手法に関してのみ述べられており，特定ノードに対する攻撃の保護手法や通信に与えるオーバヘッドに関しては述べられていない．

2.2 モビリティ機構を用いる IP アドレスホッピング

モビリティ機構として Mobile IPv6[9]を用いて IP アドレスホッピングを行うものに MTM6D[7]や MTM6D II[10]がある．また，モビリティ機構に MAT[3]を用いて行う MAT MTD[2]がある．

Mobile IPv6 は異なるネットワークを移動可能なモバイルノード(Mobile Node : MN)に，ネットワークを移動しても変化しないホームアドレス(Home Address : HoA)とネットワークを移動することで変化する気付けアドレス(Care-of Address : CoA)をもたせ，ホームエージェント(Home Agent : HA)と呼ばれる HoA と CoA の対応情報を管理するノードを配置することでモビリティをサポートする．MTM6D では CoA を動的に変更することで IP アドレスホッピングを実現している．しかし，MTM6D では最適化通信をするために Mobile IPv6 では全パケットの IPv6 のルーティングヘッダのホームアドレスオプションに HoA が格納され、中間者攻撃の対象になったり，通信する 2 つのホストのプライバシーに関わる他の攻撃対象となるリスクをもつ．これらを解決するために MTM6D II では IKE ver2[11]を用いる IPsec をホスト間の通信に追加している．これにより中間者攻撃からの防御が可能になるが，モビリティ機構によるアドレス変換に加え，IPsec による処理など MTD のためのオーバヘッドがより大きくなる．しかし，そのための通信性能低下への対策は講じられてない．

MAT はネットワーク上の位置識別子であるモバイルアドレス(Mobile Address : MoA)と端末識別子であるホームアドレス(Home Address : HoA)を相互変換することでモビリティをサポートする．HoA と MoA は全 MAT 対応ノードがもち，両アドレスの対応情報は，各ノードがもつ IMT(IP Address Mapping Table)と両 IMT の更新に用いられる IMS(IP Address Mapping Server)が管理する．IMS が保持する対応情報は MAT 対応ノードが新たなネットワークへ移動し，IMT 更新時に IMS に通知することで更新される．HoA と MoA の変換は図 1 に示すようにネットワーク層で行う．ノードから送信されるパケットは IMT を参照することで，HoA から MoA に変換し，下位層に渡す．パケットは MoA によってルーティングされるので，最短経路で通信が行われる．MAT はカーネル空間に実装されたアーキテクチャであるが，MAT をユーザ空間のアプリケーションとして実装した

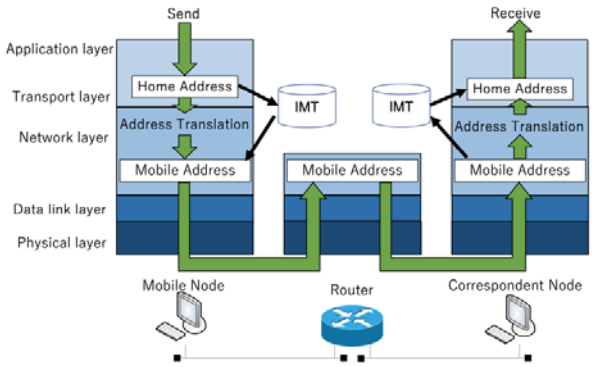

図 1：MAT のアドレス変換

DPDK-MAT[5]もある．DPDK-MAT は高速パケット処理機構である DPDK[4]を用いており，スループットの向上や導入，開発を容易にしている．

MAT MTD は，MAT の MoA を動的に変更することで IP アドレスホッピングを行う．ホッピング前のアドレスを残しておくことで，パケットロスを防ぐ仕組みになっている．

保護対象のノードと通信するノードは事前共有共通鍵をもっている必要がある．MAT MTD は HA を経由して通信を行う必要やパケットのカプセル化によるオーバヘッドがない．しかし，MAT MTD では，特定ノードに対する攻撃の防御にのみ述べられており，伝送路上のデータ保護手法に関しては述べられていない．

3. 提案方式

前述のとおり MTD にはノード（サーバ）そのものの攻撃耐性の強化とノード間の伝送路上のデータ保護が期待されるが，これらの機能を実装することによる通信品質の劣化やオーバヘッド増加のための通信性能低下を回避することもあわせて必要である．本稿ではこれらの期待される項目を備えるものを DPDK-MAT MTD として提案する．

3.1 設計方針

DPDK-MAT MTD は既存研究の MAT MTD に動的に伝送路の変更を行うホッピングであるルートホッピングを加えた MTD 手法であるが，MAT アーキテクチャの実装として DPDK-MAT を用いている点で MAT MTD とは異なる．DPDK-MAT を用いることで求められる通信性能低下を回避しようとするものである．

これらを考慮し，DPDK MAT MTD を以下の方針で設計する．

(1) IP アドレスホッピング時のセッション途絶がない
(2) 複数のネットワークをまたぐ IP アドレスホッピングが可能
(3) 複数のネットワークをまたぐシグナリング処理のないルートホッピングが可能
(4) 保護対象サーバおよびクライアントの属するネットワークの構成変更を必要としない

(5) パケットは IPsec によって標準で暗号化される
(6) 保護対象サーバは FQDN による名前解決が可能
(7) 提案方式の導入によって通信スループットが低下しないこと

(1)(2)は IP モビリティ技術を使用することで実現可能であり, (3)(4)を満たすためにエンド間でアドレス変換を行う MAT を使用する．(5)は事前共有鍵を用いる DPDK の IPsec 機能を使用する．(6)は SSL 通信を行うために必要であり，MAT の HoA を FQDN と対応させる．また，DPDK-MAT を用いることによって(7)を満たすことを目指す．DPDK-MAT は MAT のカーネル空間実装と比較してスループットが上回っていることが[5]により確認済みであり, IP アドレスホッピングやルートホッピング, IPsec による暗号化処理が加わった場合のスループットを考慮して採用した．

3.2 IP アドレスホッピングの方式

ホッピング先のアドレス空間やアドレスの選択，アドレスの共有方法は MAT MTD と同様である．ただし，ホッピング周期は異なり，MAT MTD の固定周期(60 秒)を可変長に変更した．また，IPv6 アドレスのみに対応している．

IP アドレスホッピングは MAT MTD 同様，MAT の MoA を変更することで実現する．MAT MTD ではホッピング前のアドレスを残しておくことで，ホッピングによるパケットロスを防ぐ仕組みになっており，DPDK-MAT MTD でも同じである．このことから，１つのインタフェースにつき 2 つのアドレスをもつことになる．以降，ホッピング先の新たなアドレスのことを Current MoA，ホッピング前の古いアドレスを Outdated MoA とする．

IP アドレスホッピングの処理フローは以下のとおりである．

1） Current MoA を保護対象サーバにある全てのインタフェースに対してそれぞれ生成する．
2） 全てのインタフェースの中で最大 2 つまで選択したのち，IMS に Current MoA と HoA のマッピング情報を登録し，自身の IMT と物理 NIC の管理情報を更新する．
3） クライアントに IP アドレスホッピングが完了したことを MUO(Mapping Update Option)を用いて通知する．

ホッピング前まで使用していた Current MoA はホッピング後 Outdated MoA となり，今まで使用していた Outdated MoA は削除される．保護対象サーバへの Outdated MoA 宛のパケットは攻撃者からの追跡を防ぐために MAT 対応ノードまたは IMS から送信されたもののみ受信する．

サーバと事前共有した共通鍵を持つクライアントは IMS への問い合わせにより，HoA に対応した有効な MoA を得ることが可能である．しかし，共通鍵を持たないクライアントはサーバの HoA に対応した MoA を得ることが困難であるためポートスキャンなどの偵察や DDoS 攻撃の困難性を高めることが可能である．

3.3 ルートホッピングの方式

ルートホッピングは HoA に複数の MoA を対応付け，アドレス変換する際に変換先アドレスを変更することでホッピングを行う．IP アドレスホッピングとは異なり，アドレス変更の通知などホッピング時のシグナリング処理が不要で，内部処理のみで切り替えが可能な設計となっている．そのため，オーバーヘッドが小さく，両ノードの通信に用いられるパケットを頻繁に異なる伝送路で送信することができ，盗聴攻撃などの困難性を高めることが可能である．

DPDK-MAT MTD のルートホッピングの方式は以下のとおりである．

(a) ルートホップ先の選択

DPDK-MAT MTD では，保護対象サーバは 2 つ以上のプレフィックスの異なるネットワークに接続されている状態を想定している．ルート選択は，ユーザが接続されているネットワークの中から 2 つユーザが選択し，ポッピング先ルートとする．

(b) ホッピング周期

ホッピング周期はユーザが任意で決定可能であるが，ルートホップは１セッション内の通信パケットを異なる経路で伝送することで通信データを保護することが目的である．そのため，１セッションの通信に要する時間より小さい値にすることが望ましい．

3.4 アーキテクチャ

DPDK-MAT MTD のアーキテクチャの構成を図 2 に示す．本提案アーキテクチャはアドレス変換やホッピング処理，IPsec による暗号化，物理 NIC のアドレス管理などを行う DPDK-MAT とホッピング先のアドレスの生成や通知, 指示を行うホッピングコントローラ(Hopping Controller)の 2 つから構成されている．

アプリケーション(App)が送信するパケットは DPDK-MAT によって生成された仮想インタフェース(mat_vif)を通じて DPDK-MAT へ引き上げられる．図 2 の DPDK-MAT では, IPsec による暗号化を施した後, HoA に対応する MoA を IMT から探し，アドレス変換を行った後，ネットワークへ送信する．

ホストホッピング(Host Hopping)はホッピングコントローラで生成された Current MoA を受け取り，IMS へ登録，

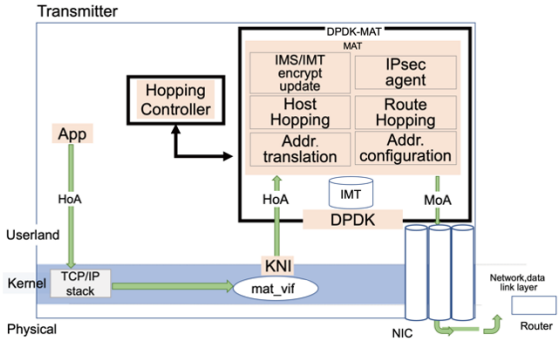

図 2：DPDK-MAT MTD の構成

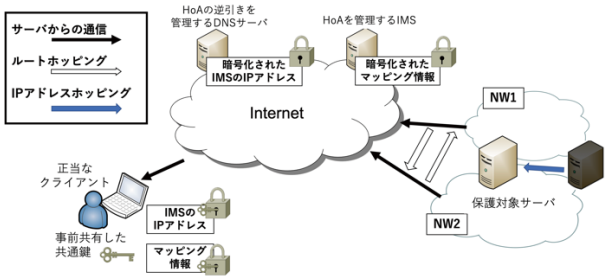

図 3：DPDK-MAT MTD の概観

IMT と物理 NIC の情報更新を行う．IMS への登録は事前共有した共通鍵で暗号化したものを用いる(IMS/IMT encrypt update)．

ルートホッピング(Route Hopping)はホッピングコントローラから通知された周期に従って通信に用いる物理 NIC の切り替えや，ルーティングテーブルの変更処理を行う．

図 3 に本提案の概観を示す．保護対象サーバは 2 つ以上のネットワーク接続されており，ルートホッピングにより伝送路を切り替える．IP アドレスホッピングは接続されたネットワーク内でアドレスのホスト部を変更することで行われる．

4. 実装

実装環境の仕様を表 1 に示す．現在実装が完了しているのは図 2 で示した構成図の「Addr translation」「Addr configuration」「Route Hopping」の箇所であり，MAT による通信やルートホッピング機能の確認ができている．今後は IP アドレスホッピングを行う「Host Hopping」やホッピング時のシグナリングパケットの暗号化を行う「IMS/IMT encrypt update」, IPsec によるパケットの暗号化を行う「IPsec agent」，ホッピング先 IP アドレスの生成やホッピング周期の通知を行う「Hopping Controller」の実装を進めていく予定である．

5. 通信性能に関する実験

5.1 ルートホッピングに要する時間の計測

ルートホッピングは単一セッション上のパケットを別経路で送ることで，パケット傍受によるセッションの復元や特定のフローに対する DoS 攻撃を防ぐ目的で行う．そのため，ルートホッピングの間隔はできるだけ小さい方が良いが，1 回のホッピングが完了するまでの時間より小さくすることはできない．

そこで，ルートホッピングの完了に要する時間の最小値を計測する．ルートホッピングの完了とは，送信する物理 NIC が切り替わり，正しいルーティングテーブルが作成されるまでと定義する．ルートホッピングの完了時間は DPDK-MAT MTD のプログラムのソースコードで上記の処理が行われる箇所に時間を計測する関数を挿入して計測する．実験に用いたノードは表 2 のサーバである．

表 1：開発環境の仕様

	開発環境
OS	Ubuntu 18.04.5 LTS
CPU	Intel(R) Core(TM) i7 4790K 4.00GHz
Memory	DDR3 24GB
Kernel	Linux 4.15.0-139-generic
DPDK ver	19.05
Language	C
Compiler	gcc 7.5.0

表 2：測定に用いたノードの仕様

	サーバ	クライアント
OS	Ubuntu 18.04.5 LTS	Ubuntu 18.04.5 LTS
CPU	Intel(R) Core(TM) i7 4790K 4.00GHz	Intel(R) Core(TM) i9 9900X 3.50GHz
Memory	DDR3 24GB	DDR4 32GB
Kernel	Linux 4.15.0-139-generic	Linux 4.15.0-139-generic
DPDK ver	19.05	19.05

表 3：ルートホッピングに要する時間

	所要時間[μ sec]
物理 NIC 切り替え時間	0.088
ルーティングテーブルの変更時間	110.41
合計	110.50

実験結果を表 3 に示す．計測は 5 回行い，結果はその平均値である．

5.2 ルートホッピングがスループットに与える影響

ルートホッピングの機能を有効にした場合，RTT の異なる経路でパケットが伝送されるため，TCP の順序制御や再送制御による処理遅延など通信のオーバーヘッドが生じる．このオーバヘッドがどの程度サーバの通信スループットに与えるのかを調べる．実験は RTT やスループットなど通信性能が異なるネットワーク間をルートホッピングした場合と同程度のネットワーク間のルートホッピングの 2 つの場合で行った．

スループットの測定実験環境を図 4 に示し，測定に用いたノードの仕様を表 2 に示す．経路 A と C におけるサーバとクライアント間の RTT は 0.1msec 程度，経路 B におけるサーバとクライアント間の RTT は 25msec 程度である．サーバからクライアントへ 1000MB のファイル転送を行った際のスループットを計測する．ファイルの転送は sftp を用い，その転送終了後の結果をスループットとした．スループットの計測は 1) ルートホッピングを 3 秒間隔で実行した場合，2) ルートホッピングを 1 秒間隔で実行した場合，

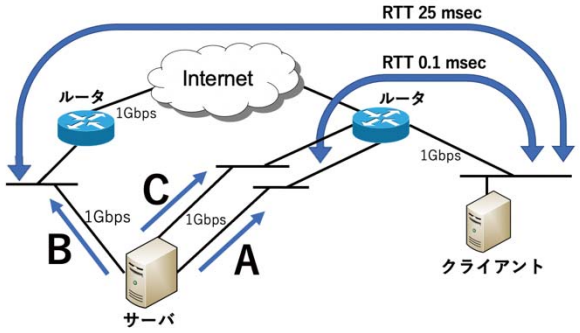

図 4：スループット測定実験環境

3) ルートホッピングを実行せず，図 4 の経路 A のみを使用した場合，4) 経路 B のみを使用した場合，5) 経路 C のみを使用した場合の 5 つで行った．このとき，ルートホッピングは 3 秒間隔と 1 秒間隔で実行した．また，1)と 2)について RTT の差のない経路 A と C のホッピング時の計測も行った．

通信に用いられるパケットはルートホッピングのタイミング次第で，往復する経路が異なる場合がある．例えば経路 A と経路 B を用いた場合，送信パケットは経路 A を使用するが，それに対する Ack パケットは経路 B を使用し，返ってくる場合もある．

実験結果を表 4 に示す．なお，計測は 5 回行い，結果はその平均値である．

5.3 考察

表 2 の結果より， 1 回のルートホッピング完了時間の最小値は 0.1msec 程度であることがわかった．このことより，ホッピング間隔は 0.1msec 以上に設定することが可能で，ホッピング完了時間がホッピング間隔の制約になることは無いと言える．

表 4 より，経路 A と経路 C とのルートホッピングの場合，ホッピング間隔に関係なくスループットは経路 A や経路 C のみを使った場合と同一であることがわかる．これは表 2 で示したようにルートホッピング自体のオーバヘッドが 0.1msec と非常に小さく，ホッピング間隔が小さくなった(ホッピングの頻度が高い)としてもスループットに反映されないからである．

一方，RTT が異なる経路 A と経路 B 間のルートホッピングでは，ホッピング間隔が小さくなるとスループットが低下していることがわかる．これは実験の通信で用いた sftp で使用されるトランスポートプロトコルである TCP の順序制御や再送制御などが，ホッピング間隔が小さくなると多発し，その制御によるオーバヘッドが大きくなるからだと考えられる．実際に 3 秒間隔と 1 秒間隔の場合でパケットキャプチャを行い, TCP の Out of Order や Duplicate Ack などのエラー通知パケットが通信で用いられたパケット全体の何割を占めているかを確認すると，3 秒間隔の場合は

表 4：ホッピングの有無によるスループット

ホッピング			経路 (RTT 差)	スループット(Mbps)
有り	ホッピング間隔	3 秒	A,B(25ms)	423.20
			A,C(0ms)	777.76
		1 秒	A,B(25ms)	399.68
			A,C(0ms)	787.20
無し	A のみ			787.04
	B のみ			77.60
	C のみ			773.11

10.3%，1 秒間隔の場合は 11.7%を占めていた．

以上より，RTT やスループットに差があるがあるネットワーク間でルートホッピングをする場合，ルートホッピングにおけるオーバヘッドは上位層の通信プロトコルの影響を受けて生じ, 今回のような TCP 通信ではホッピング間隔が小さいほどオーバヘッドが大きくなることがわかった．

UDP を使ったファイル転送の実験は行っていないが，TCP を使った実験の結果を踏まえると, UDP には再送制御や順序制御などが無いため，RTT の差があるネットワーク間のルートホッピングでもスループット低下は生じないと予想されるが，今後追加実験により検討する．

6. おわりに

本稿では MTD に必要なノード（サーバ）そのものの攻撃耐性の強化とノード間の伝送路上の情報の保護機能をもつ DPDK-MAT MTD の開発について述べた．これらの機能の実装にあたって通信品質の劣化やオーバヘッド増加のための通信性能低下を回避するための設計方針を述べた．具体的には MAT MTD の IP アドレスホッピングの機能に追加して，伝送路上の情報を防御する MTD 手法であるルートホッピングを DPDK-MAT を用いてユーザ空間アプリケーションとして実装するものである．

ルートホッピングの実装によるオーバヘッドがどの程度かを確認する実験を行い，ホッピング完了にかかる時間がオーバヘッドとなることはなく，通信性能が同程度のネットワーク間のルートホッピングはスループットへ与える影響はほとんど無いことを確認した．また，RTT が異なるネットワーク間のホッピングでは上位層プロトコルの処理の影響を受け，ホッピング間隔が小さいほどオーバヘッドが大きくなることを確認した．

現在プロトタイプシステムの実装を進めており，今後は IPsec による暗号化や，ホッピングコントローラの実装，ルートホッピングによってどの程度，攻撃の困難性を高めることが可能か, IPsec がスループットに与える影響など実装，評価を進めていく．

謝辞

本研究の一部は JSPS 科研費 19K11929，21H03432，及び 18K11271 の支援を受けて実施しました.

参考文献

[1] National Cyber Leap Year Summit 2009(online), available from <https://www.nitrd.gov/nitrdgroups/index.php?title=File:National_Cyber_Leap_Year_Summit_2009_CoChairs_Report.pdf> (accessed 2021-05-08).

[2] 大畠史也，前田香織，大石恭弘，相原玲二,:IP モビリティを用いた IP アドレスホッピングによる MTD の提案，*信学技報*，Vol.117，No.294，pp.7-12(2017).

[3] 相原玲二，藤田貴大，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.43.12，pp.3889-3897(2002).

[4] Home-DPDK(online)，available from <https://www.dpdk.org/> (accessed 2021-05-08).

[5] Yuto Tanabe，Kaori Maeda，Yasuhiro Ohishi，Reiji Aibara，: A Userland Implementation of an IP Mobility Support Function using Data Plane Development Kit, Proc. *IEICE Tech. Rep*, Vol.119，No.291，pp.33-37(2019).

[6] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront,: MT6D:A moving target IPv6 defense, Proc. *AFCEA/IEEE MILCOM*, pp.1321–1326(2011).

[7] Vadid Heydari,: IP hopping by mobile IPv6, *Handbook Cyber-Development, Cyber-Democracy, Cyber-Defense,* pp.1–28(2017).

[8] Zheng Zhao, Daofu Gong, Bin Lu, Fenlin Liu, Chuanhao Zhang,: SDN-Based Double Hopping Communication against Sniffer Attack, *Hindawi Publishing Corporation Mathematical Problems in Engineering* Vol.2016, No.8927169, pp.13, (2016).

[9] Johnson，D. B.，Arkko, J. and Perkins，C. E.: Mobility Support in IPv6 RFC 6275 Proposed Standard 2011(online)，available from <http://www.rfc-editor.org/rfc/rfc6275.txt> (accessed 2021-05-08).

[10] Vahid Heydari,: Moving Target Defense for Securing SCADA Communications, Proc. *IEEE Access*, pp.33329-33343(2018).

[11] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. KivinenInternet: KeyExchange Protocol Version 2 (IKEv2) document RFC 7296 Oct. 2014(online), available from <http://www.rfc-editor.org/rfc/rfc7296.txt> (accessed 2021-05-08).