一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SITE2016-1(2016-06)

RTTの分散を考慮した中間者攻撃検知手法の提案

佐藤　隼人†　　宮田　純子†　　加島　宜雄†

† 芝浦工業大学　〒 135-8548 東京都江東区豊洲 3-7-5
E-mail: †{af13045,sumiko,kashima}@shibaura-it.ac.jp

あらまし　情報窃取における脅威の一つとして，中間者攻撃 (man-in-the-middle-attack: MITM attack) が挙げられる．この MITM attack の検知手法において，往復遅延時間 (round-trip-time: RTT) に着目している研究があるが，その検知基準に関する検討は不十分であり，実験環境も現実的とはいえなかった．そこで本研究では，RTT の分散を考慮した新たな MITM attack 検知手法を提案する．提案手法では，既存手法では考慮されていなかった模擬攻撃の考慮や観測点の追加などによって，より現実的な環境での検証実験を行う．さらに，当該検証実験によって提案手法の有効性を示す．

キーワード　中間者攻撃，RTT，分散，検知

A detection method of man-in-the-middle attack based on the variance of the RTT

Hayato SATO†, Sumiko MIYATA†,

and Norio KASHIMA†

† Shibaura Institute of Technology　3-7-5 Toyosu, Kotoku, Tokyo, 135-8548, Japan
E-mail: †{af13045,sumiko,kashima}@shibaura-it.ac.jp

Abstract　A man-in-the-middle (MITM) attack is one of the threats to information security. In order to detect the MITM attack, a conventional method which detects the attack by using round-trip-time (RTT) is proposed. However, in the conventional method, the consideration of the detection threshold is not sufficient and the experimental environment is unreal. In this study, we propose a new MITM attack detection method by considering a variance of the RTT. We conduct a verification experiment with more realistic environment than the conventional research by considering simulated attacks and increasing observational days. The experimental results show the effectiveness of our proposed method.

Key words　MITM attack, RTT, variance, detection

1. はじめに

情報窃取における脅威の一つとして，中間者攻撃 (man-in-the-middle attack: MITM attack) が挙げられる．RFC 2828 によると，MITM attack は，攻撃者が通信を確立している者になりすまし，データの盗聴及び改竄を行う攻撃である [1]．Conti らは，このなりすまし技術に基づいた MITM attack を大きく 4 つに分類し，その中で最も基本的な技術として spoofing-based MITM attack を挙げている [2]．これは，なりすましを行い正規の二者間の通信を傍受する攻撃であり，具体的な手段の一つとして ARP spoofing (ARP poisoning) が存在する．ARP poisoning とは不正な ARP request により，正規の通信をしているホストの ARP テーブルを書き換え，正規の二者間の通信を取得する攻撃である．

ARP poisoning を用いた MITM attack に対する検知手法は，多くの研究者によって研究されている [2]．しかし，一般にルータやネットワークの予期しない要因により，発生する大きな遅延は，ネットワークで頻繁に起こりえないもの [3] であるにも関わらず，往復遅延時間 (round-trip-time: RTT) に基づいた検知手法は数少なく，筆者の知るかぎりでは文献 [4] のみである．一方，MITM attack が行われる状況では，通常よりもホップ数が多くなるため，RTT のばらつき (分散) が通信に影響する可能性もあるが，このことは文献 [4] では考慮されていない．

そこで本研究では，RTT の分散を考慮した新たな MITM attack 検知手法を提案する．提案手法では，既に提案されている RTT に基づく MITM attack の検知手法 [4] を改善することを考える．この既存手法では，前述の RTT の分散を考慮していない点以外にも，検証実験において攻撃者をプロキシと仮定しているこ

とや，日によって RTT のばらつきが存在すると考えられるにも関わらず観測期間が 1 日であることなど，改善すべき点が複数あげられる．本研究では，これらの問題点を改善させた既存手法の再現実験を行うことで，MITM attack の被害を受けているホスト (犠牲者ホスト)，被害を受けていないホスト (通常ホスト) の検知精度を考察する．なお，MITM attack を行うホストを攻撃者ホストと定義する．さらに，この再現実験を行うことで，RTT の分散により一定数の誤検知が発生することも定量的に明らかにする．

本稿の流れは以下の通りである．2 章で本研究における事前知識について説明を行う．次に 3 章で提案手法，4 章で検証実験，5 章で実験結果について述べる．最後に 6 章でまとめを行う．

2. 事 前 知 識

本章は，本研究が提案する手法を説明するにおいて，事前知識となる事項について解説を行う．2.1 節に ARP poisoning の原理を説明をし，2.2 節に MITM attack の基本についてまとめ，2.3 節に本研究のベースとなった研究についてまとめる．

2.1 ARP poisoning

Address resolution protocol (ARP) は，コンピュータが IP アドレスから MAC アドレスの情報を取得するためのプロトコルである．例えば，同一ネットワーク上に存在しているホスト A が別のホスト B の MAC アドレスを取得したい場合，ホスト A はホスト B の IP アドレスが記載された ARP request をブロードキャストする．その後，該当ホスト B のみが自身の MAC アドレスを含んだ ARP reply を，ホスト A に向けて送る．このようにして，取得したホスト B の IP アドレスと MAC アドレスの組を，ホスト A の ARP テーブルに保存する．しかし，ARP は信用されていることが前提のため，攻撃者に対する対策がとられていない．したがって，攻撃者は容易に ARP テーブルの改竄を行うことができてしまう．この攻撃は ARP poisoning と呼ばれる．以下に ARP poisoning の原理を説明する．

図 1 に Eve (IP=192.168.10.12, MAC=EE:EE:EE:EE:EE:EE), Alice (IP=192.168.10.10, MAC=AA:AA:AA:AA:AA:AA), Bob (IP=192.168.10.11, MAC=BB:BB:BB:BB:BB:BB) を示す．図中に示すように，Alice の APR テーブルには，Bob の IP アドレスと MAC アドレスの組，Eve の IP アドレスと MAC アドレスの組が記載されている．また，Bob の ARP テーブルには Alice の IP アドレスと MAC アドレスの組，Eve の IP アドレスと MAC アドレスの組が記載されている．このとき，Eve が図中の Alice, Bob に対して，ARP poisoning を行うとする．

Eve は，Alice に向かって Bob の IP アドレスと Eve の MAC アドレスが記載された ARP reply を送信する．前述した通り，ARP は信用されていることが前提であるため，Alice は ARP reply の正当性を確認せずに APR テーブルを更新する．そのため，Alice の ARP テーブルには，Bob の IP アドレス (192.168.10.11) と Eve の MAC アドレス (EE:EE:EE:EE:EE:EE) の組が保存されることになる．同様に．Eve は，Bob にも不正な ARP reply を送信することで，Bob の ARP テーブルを更新させる．このようにして ARP の応答を偽装することにより，LAN 上でなりすましを行なうことができる．以上が ARP poisoning の原理である．

2.2 ARP poisoning を用いた MITM attack

図 1 より，ARP poisoning 成功後は，Alice-Bob 間の通信は Eve を中継にして行われることとなるため，Eve は Alice と Bob に気づかれることなく盗聴を行うことが可能となる．したがって，パスワード等の重要な情報が流出してしまう恐れがある．この攻撃は MITM attack と呼ばれ，この攻撃を事前に検知する研究が提案されている [4].

2.3 RTT を用いた MITM attack 検知手法

Vallivaara らは，ホスト間の RTT を計算することで，MITM attack を検知する方式を提案した [4]. 図 2 に，この検知で想定しているシナリオを示す．この方式では，Alice が Boogle へアクセスする際に，攻撃者である Eve が MITM attack をすでに行っていることを想定している．Alice から Boogle へコネクションをスタートしたときの時間を t_s秒，Boogle からの応答を Alice が受け取る時間を t_r秒 とする．このときの $t_r - t_s$ を，コネクション i 毎の RTT τ_i とし，この RTT の総コネクション数を n 回測定することで，平均値 $\mu_\tau = \frac{1}{n}\sum_{i=1}^{n}\tau_i$ 及び，標準偏差 $\sigma_\tau = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tau_i - \mu_\tau)^2}$ を導出する．

この手法では，Alice-Boogle 間及び，Boogle-Alice 間を Eve が傍受するとそれぞれ t_1秒，t_2秒 追加で時間がかかると仮定していた．そのため，検知基準である閾値 $\theta_{ave} = \mu_\tau + \sigma_\tau$ 以上の RTT が観測された場合，そのホストは攻撃者ホストによって MITM attack がなされていると判定される．

3. 提 案 手 法

本章では，本研究の提案手法について述べる．3.1 節に既存手法の再現実験について述べ，3.2 節に本研究の提案手法について述べる．

3.1 再 現 実 験

Vallivaara らが対象としている Web サイトの一つである

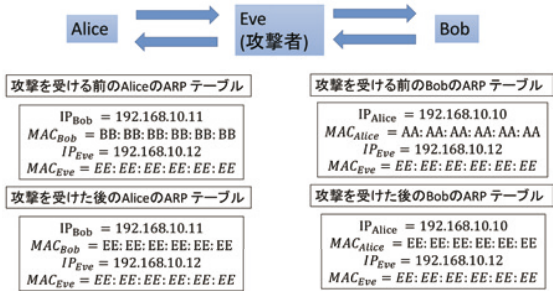

図 1　ARP poisoning を用いた MITM attack

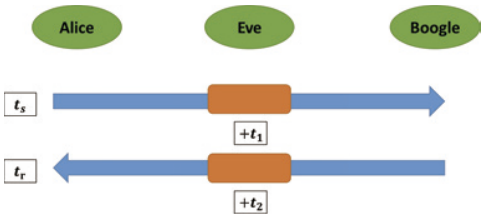

図 2　先行研究のシナリオ

www.sebgroup.com について，検知基準である $\theta_{ave} = \mu_\tau + \sigma_\tau$ に基づく判定法の再現実験を行った [4](注1)．

図 3 に，Vallivaara らが設定した閾値を適用した結果を示す．この図の縦軸は，RTT [ms]，横軸は実験開始からの経過時刻 [分] を表す．ただし，この検証実験は，2016 年 3 月 14 日 9:00〜17:00 間に計測した値とする．また，同様の計測値に対して，図 4 に通常ホスト，犠牲者ホストがそれぞれ www.sebgroup.com へアクセスを行った結果を示す．この図の縦軸は該当する RTT の頻度を，横軸は RTT [ms] を示している．

図 4 より，犠牲者ホストの RTT のばらつきは，通常ホストの RTT のばらつきに比べて大きいことがわかる．また，図 3 より，Vallivaara らの閾値 θ_{ave} は，一定数の誤検知が含まれることがわかる．

しかし，Vallivaara らが対象としていた Web サイトは日本を起点に考えると海外であるため，距離による RTT の大小の変化など，ほかの要因が絡んでくると推測できることから，閾値 θ_{ave} で上手く検知できない可能性が考えられる．その検証のため，国内の Web サイトである三菱 UFJ 銀行 (direct.bk.mufg.jp) に対しても再現実験を行った．

図 3 と同様に，図 5 に Vallivaara らが設定した閾値を適用した結果を示す．計測期間は，2016 年 3 月 15 日 9:00〜17:00 とする．図 5 は，図 3 に比べて，RTT のばらつきは減ったが，同じように閾値 θ_{ave} は，一定数の誤検知が含まれることがわかる．

以上の観測に基づき，本研究では RTT のばらつきを表現する手段として RTT の分散を利用し，この分散に着目した検知閾値して使用することで，通常ホストと犠牲者ホストの検知を行うことを検討する．

3.2 RTT の分散を用いた MITM attack 検知手法

本節では，RTT の分散に着目した新たな MITM attack 検知手法について説明する．この提案手法における想定シナリオにおいては，図 2 と同様に，あるホストが web サイトを閲覧する際に，MITM attack が行われているかどうかを閾値を用いて検知することを考える．検知を行うための閾値は，既存手法と同様に，RTT に基づいて設定する．ただし，以下の理由から，RTT の算出方法は既存手法と異なるものを使用するものとする．

RTT の算出方法として，文献 [5] では TCP の SYN とそれに対する応答の差，既存手法では TCP のタイムスタンプを使用していた [4]．しかし，実ネットワーク上でタイムスタンプを測定値として利用可能か調査したところ，ほとんどのホストでタイムスタンプを利用できないことが明らかになっている [7]．そこで本研究では，TCP の SYN とそれに対する応答である SYN ACK との差を RTT として計測する．

本研究の提案手法の閾値 θ_{var} を RTT の分散 V_τ を用いて，$\theta_{var} = \alpha V_{tau}$ と定義する．ただし，TCP のコネクション i 毎の RTT を $\tau_i = t_r - t_s$，α を閾値における係数とする．分散 V_τ は，総コネクション数 n およびその平均値 μ_τ を用いて，$V = \frac{1}{n}\sum_{i=1}^{n}(\tau_i - \mu_\tau)^2$ で導出されるとする．また，既存手法の閾値は，$\theta_{ave} = \mu_\tau + \sigma_\tau$ と定義する．

既存手法では，コネクション毎に閾値を超えているかどうか判定し，MITM attack を検知していたが，本提案手法では，複数のコネクション n [本] に対して導出した 分散 V_{tau} を計算することで，判定を行う．すなわち，$\theta_{var} \leq V_{tau}$ となった場合には，当該複数コネクションが MITM attack を受けていると判定する．このように，既存手法と異なり，複数コネクションをまとめて判定することで，検知負荷を下げつつ MITM attack 検知が実現可能となる．

4. 検 証 実 験

本章では，提案手法の検証実験について記述する．初めに，4.1 節で実験環境を示し，4.2 節で実験方法についてまとめる．

4.1 実 験 環 境

本研究の提案手法の検証実験では，6 台のコンピュータを使用し，そのうち 1 台が攻撃者ホスト，1 台が通常ホスト，1 台が犠牲者ホストの計 3 台の 2 セットで実験を行った．なお，検証実験の結果は，コンピュータの性能に大きく依存してしまうと考えられるため，通常ホストと犠牲者ホストは同一のコンピュータを使用した．

表 1 に攻撃者ホストの環境，表 2 に 1 セット目の通常ホスト，犠牲者ホストの環境，表 3 に 2 セット目の通常ホスト，犠牲者ホストの環境を示す．本実験では，攻撃者ホストが Ettercap [6] よ

(注1)：ただし，この再現実験における実験環境は，5 章の実験環境と同一とする．

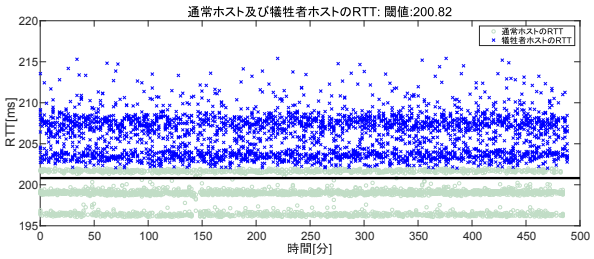

図 3　www.sebgroup.com に対する既存手法に基づいた検知結果

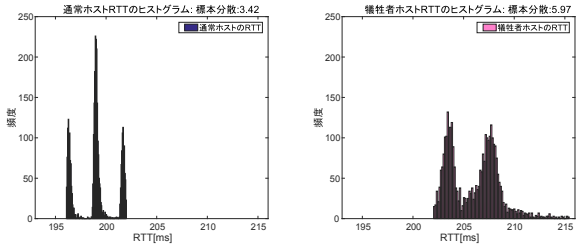

図 4　www.sebgroup.com に対する RTT のヒストグラム

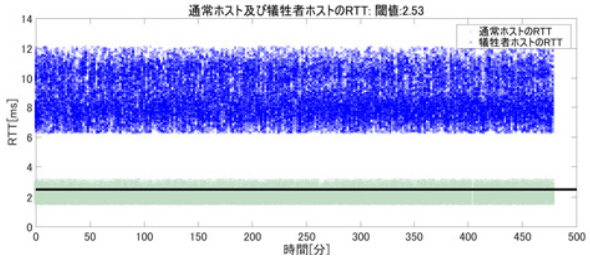

図 5　direct.bk.mufg.jp に対する既存手法に基づいた検知結果

表 1　攻撃者ホストの環境

項目	性能
Product	GB-BXi7-550
OS	Ubuntu 14.04 LTS
CPU	Intel(R)Core(TM)i7-5500U CPU@2.40GHz
Memory	16GB

表 2　1 セット目の通常ホストと犠牲者ホストの環境

項目	性能
Product	Endeavor MR4000
OS	Ubuntu 14.04 LTS
CPU	Intel(R)Core(TM)i5 CPU650@3.20GHz
Memory	4GB

表 3　2 セット目の通常ホストと犠牲者ホストの環境

項目	性能
Product	FMVNN6H3M
OS	Ubuntu 14.04 LTS
CPU	Intel(R)Core(TM)i5-3320M CPU@2.60GHz
Memory	4GB

り MITM attack を行うため，通常ホストはルータと直接通信を行うことに対し，犠牲者ホストは攻撃者ホストを挟みルータと通信を行うことになる．その様子を図 6 に示す．また，Ettercap は多くのプロトコル解析をサポートしており，ネットワークやホスト分析のための豊富な機能が含まれているフリーソフトウェアである [6].

4.2 実 験 方 法

既存手法の検証実験においては，金融系の Web サイトを対象としていた．これにならい，本研究では日本でメガバンクと呼ばれている 3 つの銀行サイト，三菱 UFJ 銀行 (direct.bk.mufg.jp)，みずほ銀行 (www.mizuhobank.co.jp)，三井住友銀行 (www.smbc.co.jp) を対象に実験を行った．

また，本研究では時間ごとで RTT に差がでると推測したため，キャプチャ時間範囲を，24:00 ～ 7:00 (midnight)，9:00 ～ 17:00 (morning)，19:00 ～ 22:00 (night) の大きく 3 つに分け観測を行った．

以下に，実験の流れを示す．

（1） dig コマンドより，対象とする Web サイトの IP アドレスを把握する．

（2） 同一の IP アドレスに対する通信を観測するため，犠牲者ホスト，通常ホストそれぞれの hosts ファイルに IP アドレスとドメイン名の組を記載する．

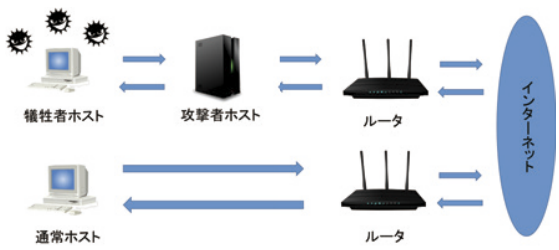

図 6　実 験 環 境

（3） 攻撃者ホストで Ettercap [6] を起動し，犠牲者ホストへ MITM attack を行う

（4） 通常ホストと犠牲者ホストで，指定の Web サイトにアクセスを行い，2 分後ごとにブラウザのリロードを指定回数行うプログラムを起動する．

（5） 同時に，それぞれのホスト上で tcpdump を起動し，pcap ファイルを出力する．なお，tcpdump のオプションより，ポート番号を 80 番に設定する．

（6） コネクション i における通常ホストの RTT τ_i^{nor} および犠牲者ホストの RTT τ_i^{vic}，さらにコネクション i における通常ホストの分散 V_τ^{nor} および犠牲者ホストの分散 V_τ^{vic} を計算し，既存手法と提案手法それぞれについて検知精度の比較を行う．ここで，$T_{nor} = \{\tau_1^{nor}, \tau_2^{nor}, \cdots, \tau_n^{nor}\}$, $T_{vic} = \{\tau_1^{vic}, \tau_2^{vic}, \cdots, \tau_n^{vic}\}$ とする．この際，外れ値を考えないようにするため，T_{nor} のうち，上位 4 %，下位 1 %，，T_{vic} のうち上位 1 %，下位 4 %を削る．また，本研究では基礎検討として (1) で取得した IP アドレス以外については考慮しないこととする．

4.3 比 較 方 法

前節の (6) に記述した比較方法について説明する．まず，T_{nor} のデータを k 個に分割し，その中からランダムに選ばれた 1 個のデータを訓練データとし，残りの $k-1$ 個のデータをテストデータとする．この訓練データに対して，θ_{ave} および θ_{var} を計算する．その後，T_{vic} のデータも k 個に分割し，これらのデータをテストデータとする．分割した T_{nor} および T_{vic} のテストデータについて，先ほど導出した θ_{ave} および θ_{var} を用いて閾値判定を行う．ただし，閾値判定方法は以下の基準に従うものとする．

- $\tau_i^{nor} > \theta_{ave}$ の場合

  →誤検知とみなし，不正解とする．

- $\tau_i^{vic} < \theta_{ave}$ の場合

  →検出漏れとみなし，不正解とする．

- $v_\tau^{nor} > \theta_{var}$ の場合

  →誤検知とみなし，不正解とする．

- $V_\tau^{vic} < \theta_{var}$ の場合

  →検出漏れとみなし，不正解とする

上記以外の場合には，正解としてカウントする．既存手法における正解した数および提案手法において正解した数をそれぞれ c_{con}, c_{pro} とし，以下のように定義する．

$$a_{con} = \frac{100c_{con}}{\frac{n(k-1)}{k}} \tag{1}$$

$$a_{pro} = \frac{100c_{pro}}{k-1} \tag{2}$$

次章では，この実験方法に基づき，既存手法および提案手法の正解率の比較を行う．

5. 実験結果

本章では，既存手法と提案手法による検知率の比較結果を示す．5.1 節に direct.bk.mufg.jp による実験結果を示す．次に 5.2 節に www.mizuhobank.co.jp，5.3 節に www.smbc.co.jp による実験結果を示す．

5.1 対象 Web サイトに対する RTT の分散計算

表 4 に国内サイトにおける通常ホストと犠牲者ホストのそれぞれの RTT の分散を示す．なお，これらの観測時刻は原稿のスペースの都合上，morning のみとする．表 4 より通常ホストと犠牲者ホストの RTT 分散には，明確な差があることが確認できる．本実験では，この分散の結果から，閾値の係数 $\alpha = 1.5$ と設定した．

表 4　国内サイトに対する通常ホストと犠牲者ホストの RTT 分散

	通常ホスト			犠牲者ホスト		
	3/15 (金)	3/18 (月)	3/22 (金)	3/15 (金)	3/18 (月)	3/22 (金)
direct	0.11	0.12	0.11	2.00	1.94	1.98
	3/16 (土)	3/21 (木)	3/23 (土)	3/16 (土)	3/21 (木)	3/23 (土)
mizuho	0.25	0.23	0.25	1.54	1.74	1.61
	3/15 (火)	3/18 (月)	3/22 (金)	3/15 (火)	3/18 (月)	3/22 (金)
smbc	0.08	0.03	0.04	3.15	2.18	2.17

5.2 direct.bk.mufg.jp

表 5 に direct.bk.mufg.jp を対象としたときの既存手法と提案手法による検知率をそれぞれ示す．表 5 より，既存手法の検知率より提案手法の方が正解率が高い場合が多いことがわかる．しかし，3/22 と 3/29 の night は提案手法の方が検知率が低い結果となった．これは，V_τ^{nor} が閾値より大きくなったためである．V_τ^{nor} と閾値の比較は図 7 の通りに行っているが，区分けされたテストデータのうち V_τ^{nor} が閾値を超えてしまう割合が多かったため検知率が下がってしまったと考えられる．

表 5 に示したデータのうち，4/1 morning について考察する．この日時のデータにおいて既存方式の検知率は 94 %であったことに対して，提案方式の検知率は 100 %であった．このデータに対して，通常ホストおよび犠牲者ホストの RTT のヒストグラムと，RTT の時間に対する散布図を図 8, 図 9 にそれぞれ示す．図 8 の縦軸は該当する RTT の頻度を，横軸は RTT [ms] を表す．図 9 の縦軸は RTT [ms]，横軸は実験開始からの経過時刻 [分] を表す．

図 8 より，犠牲者ホストの RTT のばらつきは通常ホストの場合と比較して大きいことがわかる．また，このときの分散を計算すると，それぞれ $V_\tau^{nor} = 0.11$，$V_\tau^{vic} = 1.97$ となっているため，分散を用いた提案検知法が検知精度の向上につながったと考えられる．一方で，図 9 に示すように，既存手法では，平均値に着目しているため，閾値が通常ホストの RTT を超える値になってしまい，検知精度が下がっていることもわかる．

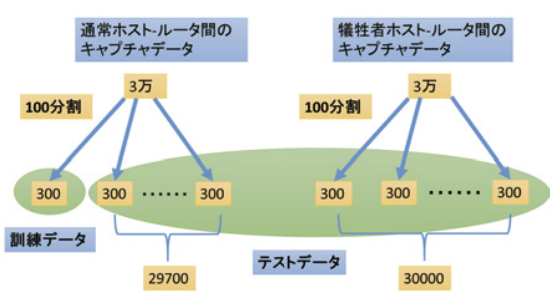

図 7　比較方法

表 5　direct.bk.mufg.jp に対する既存手法と提案手法の正解率比較

	既存手法			提案手法		
2016 年	midnight	morning	night	midnight	morning	night
03/15	91 %	93 %	96 %	100 %	100 %	100 %
03/18	94 %	93 %	95 %	100 %	100 %	100 %
03/22	92 %	93 %	94 %	100 %	100 %	93 %
03/29	93 %	89 %	95 %	100 %	96 %	88 %
04/01	94 %	94 %	93 %	100 %	100 %	100 %

5.3 www.mizuho.co.jp

表 7 に www.mizuho.co.jp を対象としたときの既存手法と提案手法による検知率をそれぞれ示す．表 5 に比べ，表 7 では，提案手法の正解率が極端に低い値が存在していることがわかる．

以下に考察を述べる．表 6 に www.mizuho.co.jp における通常ホストと犠牲者ホストのそれぞれの RTT 分散の値を示す．表 6 より，3/28 と 3/30 の V_τ^{nor} が V_τ^{vic} より大きくなっている．提案手法の閾値は $\theta_{var} = \alpha V_{tau}$ とし，閾値の係数 α は 1.5 と設定した．図 7 より，V_τ^{nor} より閾値を設定するため，仮に V_τ^{nor} が V_τ^{vic} より大きい値となる場合は閾値が V_τ^{vic} を超えてしまい，正解率が非常に低くなる．表 7 の 3/28 と 3/30 それぞれの morning における正解率も 53 %, 49 %と悪くなっていることがわかる．morning 同様，ほかの時間帯においても，V_τ^{nor} が V_τ^{vic} より大きくなったため，正解率は低くなった．しかし 3/30 night のみ，V_τ^{nor} が V_τ^{vic} より小さかったため正解率は高くなった．

表 7 に示したデータのうち，3/30 morning について考察する．この日時のデータにおいて既存手法の検知率は 91 %であった

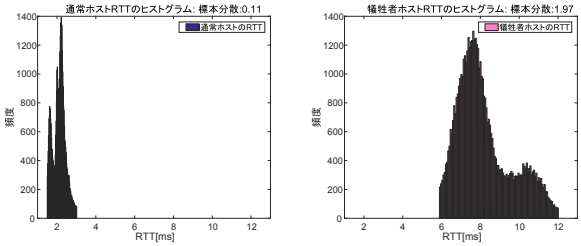

図 8　direct.bk.mufg.jp に対する RTT のヒストグラム

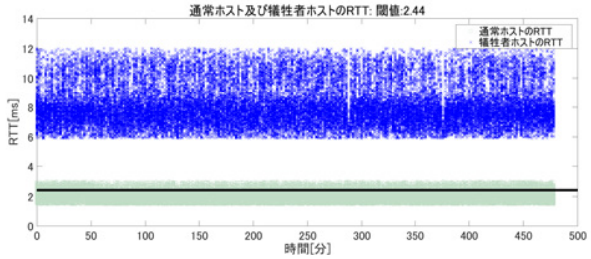

図 9　direct.bk.mufg.jp に対する既存手法に基づいた検知結果

ことに対して，提案手法の検知率は 49 %であった．このデータに対して，通常ホスト及び犠牲者ホストの RTT のヒストグラムと，RTT の時間に対する散布図を図 10, 図 11 にそれぞれ示す.（軸は図 8，図 9 と同様）．図 10 に示すように，犠牲者ホストのばらつきに比べて通常ホストのばらつきが大きいことがわかる．また，このときの RTT 分散を計算すると，それぞれ $V_\tau^{nor} = 2.78$，$V_\tau^{vic} = 2.09$ となっているため，分散を用いた提案検知法の検知精度が下がったと考えられる．一方で，図 9 では通常ホストと犠牲者ホストの RTT のプロットが一部重なっており，既存手法のように RTT の平均値を用いて分類する場合でも，閾値の設定法にかかわらず完全な検知は不可能であることがわかる．

表 6　www.mizuho.co.jp に対する通常ホストと犠牲者ホストの RTT 分散

	通常ホスト		犠牲者ホスト	
	3/28 (日)	3/30 (水)	4/3 (日)	3/30 (水)
mizuho	2.77	2.78	2.54	2.09

表 7　www.mizuho.co.jp に対する既存手法と提案手法の正解率比較

既存手法				提案手法		
2016 年	midnight	morning	night	midnight	morning	night
03/16	88 %	88 %	86 %	100 %	100 %	100 %
03/21	88 %	87 %	93 %	100 %	100 %	100 %
03/23	88 %	89 %	87 %	100 %	100 %	100 %
03/28	91 %	92 %	92 %	50 %	53 %	47 %
03/30	91 %	91 %	91 %	50 %	49 %	100 %

5.4　www.smbc.co.jp

表 8 に www.smbc.co.jp を対象としたときの既存手法と提案手法による検知率をそれぞれ示す．この表より，既存手法の検知率より提案手法の方が正解率が高い場合が多いことがわかる．

しかし，3/22 night と 4/3 morning では提案手法の方が検知率が低い結果となった．これは提案手法の閾値 $\theta_{var} = \alpha V_{tau}$ が訓練データに依存していることが原因であると考えられる．本研究では，図 7 に示すように全データを事前に区分けしたものの一部を訓練データとしているが，これが閾値計算において適切な値かどうかの考慮は一切行っていない．そのため，訓練データの選び方によっては適切な判定が行えず検知率が下がったものと考えられる．

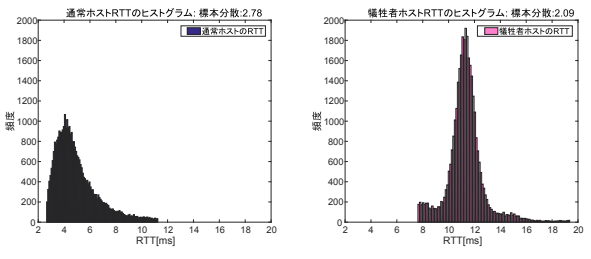

図 10　www.mizuho.co.jp に対する RTT のヒストグラム

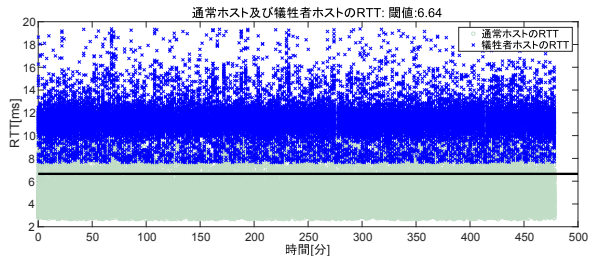

図 11　www.mizuho.co.jp に対する既存手法に基づいた検知結果

この考察を検証するため，4/3 morning において図 7 に示すように初めに区分けされたものを訓練データとせず二番目に区分けされたものを訓練データとして正解率の比較を行ったところ，正解率は既存手法が 94 %，提案手法が 99 %となった．表 8 より 4/3 morning では既存手法が 90 %，提案手法が 74 %であることから大きく改善されたことがわかる．したがって，訓練データの選択方法の検討が今後の課題となる．

表 8　www.smbc.co.jp に対する既存手法と提案手法の正解率比較

既存手法				提案手法		
2016 年	midnight	morning	night	midnight	morning	night
03/15	93 %	94 %	98 %	100 %	100 %	100 %
03/18	93 %	94 %	96 %	100 %	100 %	100 %
03/22	94 %	94 %	90 %	100 %	100 %	88 %
04/01	95 %	94 %	91 %	99 %	100 %	96 %
04/03	94 %	90 %	88 %	99 %	74 %	92 %

6.　ま　と　め

本研究では，RTT の分散を考慮した新たな MITM attack 検知手法を提案した．提案手法では，既存手法で考慮されていなかった，模擬攻撃の考慮や観測点の増加を考えることで，より現実的な環境で検証実験を行った．この検証実験から，正解率が 100 %になる場合が幾つか存在することを確認し，提案手法の有効性を示した．

今後の課題としては，検証実験から明らかになった，訓練データによる正解率変動の改善が挙げられる．また，実ネットワーク環境下を想定した拡張として，SSL を考慮している文献 [5] を踏まえた検知手法拡張，攻撃者ホストかプロキシを判別しつつ検知可能な手法への拡張を行う予定である．

文　　献

[1] RFC 2828, https://www.ietf.org/rfc/rfc2828.txt

[2] M. Conti, N. Dragoni, and V. Lesyk, “A Survey of Man In The Middle Attacks”, Proc. IEEE Communications Surveys & Tutorials, pp. 1–26, early access article, March 2016.

[3] B.-Y.Choi, S.Moon, Z.-L.Zhang, K.Papagiannaki, and C.Diot, “Analysis of Point-To-Point Packet Delay In an Operational Network”, Proc. IEEE INFOCOM, vol. 3, pp. 1797–1807, March 2004.

[4] V. Vallivaara, M. Sailio, and K. Halunen, “Detecting Man-in-the-Middle Attacks on Non-Mobile Systems”, Proc. ACM CODASPY Data and Application Security and Privacy, pp. 131–134, March 2014.

[5] K. Benton, and T. Bross, “Timing Analysis of SSL/TLS Man in the Middle Attacks”, Proc. Computer Science Cryptography and Security, August 2013.

[6] Ettercap, https://ettercap.github.io/ettercap/

[7] H. Ding, and M. Rabinovich, “TCP Stretch Acknowledgements and Timestamps: Findings and Implications for Passive RTT Measurement”, Proc. Computer Communication Review, vol. 45, no. 3, pp. 20–27, July 2015.