自律分散型インターネットセキュリティ基盤のための信頼基盤構成法の提案

堤智昭[1] 石川毅[1] 芦川大地[1] 米﨑直樹[1] 君山博之[1] 八槇博史[1]
上野洋一郎[1] 佐野香[1] 佐々木良一[2] 小林浩[1]

概要： 近年の IoT 機器の発達・増加に伴い，これらを悪用したサイバー攻撃が深刻な問題となっている．特に，DDoS 攻撃は，攻撃を受ける側がセキュリティ対策を行っていても，防ぐことができない．それに対応するために，我々は，インターネット全体で取り組むセキュリティを実現するための“自律分散型インターネットセキュリティ（AIS）基盤”を提案している．本稿では，この AIS 基盤のトラストを確立するために，AIS 基盤を構築する中核機器や AIS 基盤内でやり取りされる制御通信を安全に運用するための信頼基盤を提案する．また，信頼基盤の実現可能性を検証するために TPM シミュレータを用いてプロトタイプシステムを実装し，動作確認を行った．

A Proposal of Trust Architecture for Autonomous Internet Security Infrastructure

TOMOAKI TSUTSUMI[1] TSUYOSHI ISHIKAWA[1] DAICHI ASHIKAWA[1] NAOKI YONEZAKI[1] HIROYUKI KIMIYAMA[1] HIROFUMI YAMAKI[1] YOICHIRO UENO[1] KAORU SANO[1] RYOICHI SASAKI[2] HIROSHI KOBAYASHI[1]

1. はじめに

過激化・組織化するサイバー攻撃が大きな脅威となっている．中でも DDoS 攻撃の規模拡大が深刻な問題となっている．DDoS 攻撃には，ネットワークに接続している多数の端末を用いて通信帯域を圧迫することによって，正常な通信を不能とする攻撃がある．こうした攻撃の多くは近年爆発的に増加し，至る所に存在する IoT デバイスを悪用して行われている．そのため，攻撃を受ける側のユーザ（主に企業や行政府など）がセキュリティソフトを導入したり，システムのセキュリティをアップデートしたりしても，帯域圧迫攻撃は防ぐことができない．実際に，Akamai Technology 社のレポート[1]によると，2016 年第 3 四半期の DDoS 攻撃においては単一のサーバに向け 600Gbit/s を超える攻撃トラヒックが観測されている．これらの攻撃の多くでは，脆弱な IoT デバイスを総当たり攻撃（探索パケット）でマルウェア感染させた多数のデバイスが用いられており，約 24,000 もの送信元 IP アドレスが観測された例もある．IoT 時代を迎え，このような大量の通信フローに対して，一つひとつフローの特定を行いエンドポイントで攻撃を遮断するという従来の DDoS 緩和ソリューションを適用することは，今後難しくなっていくと考えられる．

1 東京電機大学情報環境学部
2 東京電機大学未来科学部

そこで我々は，これまでの“自分を守るセキュリティ”から“インターネット全体で取り組むセキュリティ”への転換が必要であると考え，その発想に基づいて“自律分散型インターネットセキュリティ（AIS）基盤”を提案している[2]-[12]．AIS 基盤は，各ネットワークが自律分散的に攻撃パケットをインターネットに流入／流出させないというセキュリティ対策により，インターネット全体の安全性を高めることを目標としている．

しかし，この AIS 基盤自体に脆弱性があれば，攻撃に悪用されインターネット全体のサービスを脅かすことにもなりかねない．そこで本稿では，この AIS 基盤のトラストを確立するため，AIS 基盤を構成する機器や攻撃を遮断するための廃棄要請などのプロトコルを対象としたサイバー攻撃を成立させないための信頼基盤を提案する．また，提案する信頼基盤の実現可能性を検証するため，プロトタイプシステムの実装を行った結果について報告する．

2. 自律分散型インターネットセキュリティ基盤の概要と課題

2.1 DDoS 攻撃などのサイバー攻撃対策における課題

DDoS 攻撃など帯域圧迫型のサイバー攻撃に対する対策は，これまでも様々な研究が行われ実装されてきている．ここでは，サイバー攻撃に使用されるパケットの送信元 IP アドレスが詐称されているか，詐称されていな

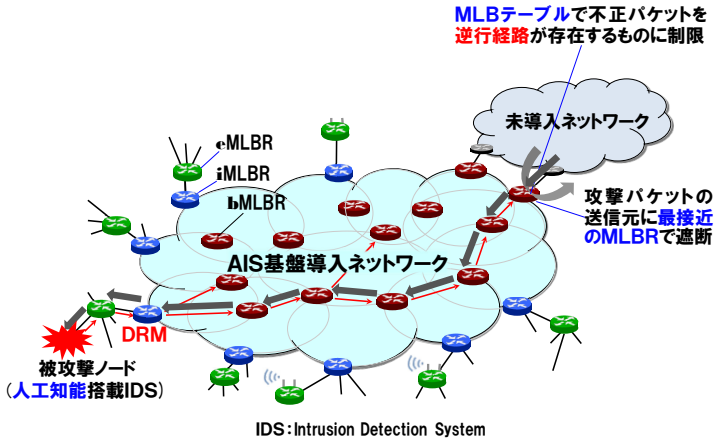

図 1　自律分散型インターネットセキュリティ基盤を導入したネットワークの例

いかの 2 つに分類し，既存の対策の課題について考察する．

まず初めに，送信元 IP アドレスが詐称されているパケットに対して行われる対策として，uRPF（unicast Reverse Path Forwarding）機能[13] を用いたフィルタリングが挙げられる．uRPF はインターネットの入口に設置されるエッジルータに適用し，通過するパケットの送信元 IP アドレスを検査し，ルータの経路表に存在しないアドレスか否かによってフィルタリングを行う[14]．アドレス詐称パケットに対して一定のフィルタリング効果を期待できるが，ルータの経路表に記載のネットワークアドレス空間内で詐称した不正パケットは遮断できない．また，宛先 IP アドレスと送信元 IP アドレス両方で経路表を検索することによるルータの過負荷回避などから，ISP の多くは uRPF 機能を設定していないのが実情で，攻撃側が圧倒的に有利な一因となっている．

次に，送信元 IP アドレスを詐称していないパケットに対しては，攻撃を検知して自動でフィルタ設定を行う仕組みを利用した対策が行われている．具体例として，契約ユーザからの対処要請や ISP のサービス継続が危うくなったとき，攻撃検知装置が検出した攻撃パケットの L3/L4 ヘッダ情報を BGP フロースペック・メッセージ[15] に載せて，他 ISP との境界ルータに送信し，遮断する方式がある[16]．ただし，パケットの詳細な属性情報を指定できないため，C&C サーバからの攻撃指令や探索パケットなどを遮断できない．また，エンドユーザから直接メッセージを送信することを想定した仕様でない，あるいは偽装メッセージをチェックする手段を持たないなどの課題がある．

2.2 自律分散型インターネットセキュリティ基盤の概要

我々が提案している MLBR（Multi-Layer Binding Router）を中核機器とする AIS 基盤は，次の 6 項目からなる[12]．

1. ユーザネットワークの出口と，インターネットの入口，および ISP 間の境界に，各々 **e**MLBR, **i**MLBR, **b**MLBR を配備し，適応型ファイアウォールとして機能する AIS 基盤を形成する．
2. MLBR は，MLB テーブルを用いて（uRPF のように）アドレス詐称パケットをフィルタリングし，不正パケットを逆行経路（送信元 IP アドレスへ到達可能な経路）が存在するもののみに制限する．
3. 被害ノードからの逆行経路に向けた廃棄要請メッセージ（DRM）により，攻撃パケットの送信元 IP アドレスに最も接近した MLBR で遮断する．
4. ソフトステート型の認証・検疫などにより，すべてのパケットの送信元をセキュリティ評価し，結果を IP ヘッダの TOS フィールドに附す．パケットを受信するか否かの判断は受信者に委ねる．
5. AIS 基盤は，人の作為が入らないよう機械的運用を軸に，公平・中立・公正性をベストエフォートで実現する．
6. ISP やエンドユーザに自発的な導入を促すインセンティブ・メカニズムを導入する

上記 1 項から 3 項を表した AIS 基盤の概要図を図 1 に示す．同図において，AIS 基盤導入ネットワークは，未導入ネットワークと **b**MLBR を介して接続している．同 **b**MLBR の MLB テーブル（e-uRPF aggregate mode†）をすり抜けた（逆行経路が存在する）攻撃パケットが，ターゲットの被攻撃ノードに到達する．人工知能搭載 IDS（学習済みデータを搭載した攻撃検知エンジン）は，

† 経路障害などにより最善でない経路から流入する正常なパケットを誤って廃棄しないよう，他 ISP から流入し得るすべての L3 アドレスを対象とする．ただし，逆行経路上で uRPF が設定されていれば，その分のエントリを省くことができる．MLB テーブルのサイズを限りなく小さくするには，ISP に uRPF の導入を促す（例えば，**b**MLBR の MLB テーブルでアドレス詐称パケットを廃棄した ISP は，uRPF 未設定の IPS からペナルティとして廃棄処理料金を徴収する）ように，6 項のインセンティブ・メカニズムをデザインする．

攻撃を検知すると，L1～L7 のヘッダ情報の中から攻撃パケットの遮断に必要な属性情報を抽出して廃棄要請メッセージ（DRM）を生成し，逆行経路に向けて送信する．DRM を最初に受信した eMLBR は，この 2 者間でのみ共有する秘密のナンスと共有鍵を用いて，eMLBR が改ざんされていないということを前提に，DRM が偽装されたものでないことを確認してから，DRM を廃棄テーブルに反映し，攻撃パケットの遮断を開始する．DRM を発する状況においては，帯域が圧迫されており，後で定義される双方向通信を必要とする統合性検証プロトコルは利用できない．eMLBR は予め iMLBR と相互認証を行い交換しておいた共有鍵を用いて認証ヘッダ（AH）を生成し，DRM に添えて iMLBR へ転送する．iMLBR では AH の正当性を検証後，DRM を廃棄テーブルに反映し，攻撃パケットの遮断を開始する．ただし AH は相手認証を行うと同時に，IP データグラムの改ざんや再送を防止するものであり，アプリケーションレイヤーでの中間者攻撃や再送攻撃を防止することができない場合があることに注意が必要である．以後，同様の手順で逆行経路に向けて DRM が転送され，攻撃パケットの送信元アドレスに最も接近した bMLBR で攻撃を遮断，すなわち攻撃パケットの AIS 基盤導入ネットワーク内への流入を阻止する．

AIS 基盤には，ほかに IoT 機器のネットワークレベルでのセキュリティ対策（ダークネットでの定点観測による探索パケットの遮断や通信相手限定による IoT 機器のマルウェアの感染予防と汚染 IoT 機器の無害化），上記 4 項に記載のすべてのパケットにセキュリティ評価を附すなどの機能がある．

2.3 AIS 基盤へのサイバー攻撃に対する課題

上記の AIS 基盤がインターネットのセキュリティ基盤として広く浸透するためには，AIS 基盤のトラストが問題となる．AIS 基盤を装備したネットワークでは，MLBR は流れるパケットに対して転送先の設定や廃棄といった大きな権限を持つ．したがって，その自律的な判断が公正・公平なものかどうか，また AIS を構成する機器や端末そして通信プロトコルが攻撃耐性をもっているかが問題となる．前者については，様々な手法をアンサンブルして使うことによって対応する．後者が本稿の主題である．

AIS 基盤を制御するための通信には，サイバー攻撃を検知したエンドポイントから送られる DRM や，定期的に更新される攻撃判定を自動化するための学習済みデータなどが含まれる．AIS 基盤の構成機器や端末は，常にサイバー攻撃に晒される一方，これらの機器間の通信も中間者攻撃や再送攻撃の対象とされる危険性が存在するため，これらのセキュリティ確保が課題となる．

この課題を解決するため本稿では，セキュリティモジュールを用いて AIS 基盤を構成する機器や制御通信を対象としたサイバー攻撃を成立させないための信頼基盤構築技術の提案を行う．さらに，プロトタイプ実装により実現可能性の検証を行う．

3. セキュリティ確保のための信頼基盤

3.1 信頼基盤が守る安全性

本稿では，AIS 基盤の脆弱性をついた攻撃や，通信における中間者攻撃に対して確保すべきトラストに係わる性質として，以下の 3 要素を満たす信頼基盤を提案する．

A) 構成機器を守るために，MLBR のソフトウェアが改ざんされていないこと（統合性:Integrity）を確認する手段を持っていること．

B) 署名や暗号化に使われる鍵を安全に管理する機能を持っていること．

C) AIS 基盤内でメッセージをやり取りする AIS 基盤制御管理用プロトコルが，プロトコル自身に対する様々な攻撃を検知する機能を持っていること．より具体的には，この機能により次の 4 点を保証すること．

C-1)悪意を持った装置による MLBR などへの成りすましを防ぎ，通信相手の真正性(Authenticity)を保証する

C-2)メッセージが改ざんされていないこと（データ統合性）を保証する

C-3)メッセージの作成者とメッセージの作成タイミングが想定されているものであること（データオリジン真正性）を保証する

C-4)秘密とすべき情報が第 3 者に渡らないこと（秘匿性）を保証する

3.2 信頼基盤の構成

各通信者間はまず共有鍵交換を行うことを基本とする．AIS 基盤内での通信参加者間の真正性の保証はこの共有鍵やナンスを用いて行い，システム統合性の確認や，攻撃検知に必要なデータの統合性の確保には，認定機関に

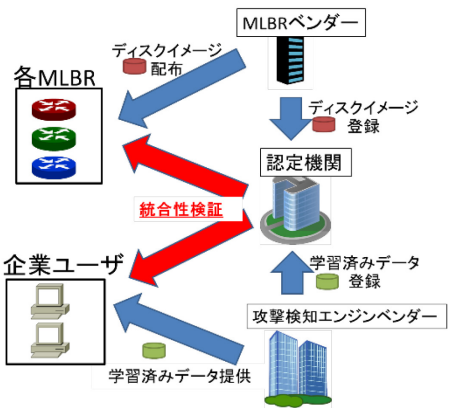

図 2　信頼基盤構築を構成する機関と手続き

登録されるプログラム群やデータのハッシュ値を用いることにより行う．これらを基に，システムの改ざんやメッセージのすり替え，再送攻撃の検知を可能とし，誤った振る舞いをしないシステムを人間の手を介さずに自動で実行させることにより，セキュリティが保たれた信頼基盤を確立する．

図 2 に信頼基盤を構成する参加者とそれが行う手続きの概要を示す．本信頼基盤は，MLBR，MLBR ベンダー，企業ユーザ，攻撃検知エンジンベンダー，認定機関の 5 つによって構成される．MLBR は攻撃パケットの遮断などの直接的なセキュリティ対策を行うための機器である．MLBR ベンダーは，MLBR の動作を規定するディスクイメージの配布を行う．企業ユーザは，導入している攻撃検知エンジンを用いてサイバー攻撃を検知し，MLBR に廃棄要請を行う．攻撃検知エンジンベンダーは，企業ユーザが用いる攻撃検知エンジンのマルウェア検出や，侵入検知のための学習済みデータの提供を行う．認定機関は，MLBR のディスクイメージや攻撃検知エンジン用の学習済みデータのハッシュ値，メッセージ認証用共有鍵の交換を行うための公開鍵などの登録・管理を行う第三者機関である．

3.3 暗号鍵の管理

暗号鍵を用いてシステムのセキュリティを確保するためには，適切に暗号鍵を保管し暗号鍵に対する不正な読み書きを防止しなければならない．そのため，比較的容易に不正アクセスが行われる恐れのある HDD 上やメモリ上ではなく，耐タンパー性のあるセキュアモジュールである TPM（Trusted Platform Module）[17]を用いた管理を行うこととする．TPM によって管理された暗号鍵を，信頼基盤のトラストルートとして活用する．

TPM は，2014 年にバージョン 2 がリリースされた比較的安価なセキュアモジュールである．汎用のコンピュータ，IoT デバイス，サーバなど多くのプラットフォームで導入しやすく，2016 年 8 月以降に販売される Windows10 を搭載したコンピュータでは TPM 搭載が必須条件となるなど，近年広く普及しつつある．

TPM には，数種類の特有の鍵があるが，本提案では次の 3 種類を用いる．

a) EK: Endorsement Key という TPM の識別と，署名または復号時に使用することを目的とした RSA 鍵である．EK は TPM の製造時に固有に生成される．
b) AIK: Attestation Identity Key という署名に使用される鍵である．AIK は使用時にその都度生成される．
c) STK: Storage Key といい，暗号化に使用される鍵である．STK は使用時にその都度生成される．

TPM は，不揮発性メモリと PCR（Platform Configuration Register）と呼ばれる揮発性メモリを持つ．不揮発性メモリは，EK を格納するのに使用され，揮発性メモリは，AIK や STK, TPM を搭載している機器構成情報の記憶に使用する．

3.4 MLBR のソフトウェア改ざん検知

3.4.1 MLBR の統合性検証概要

MLBR は，初期インストールやアップデート時に，まず自身のディスクイメージのハッシュ値（計測値と呼ぶ）を計算し，認定機関の持つハッシュ値と比較する．これはディスクイメージの電子的な配布において，改ざんされたプログラムにすり替えられていないかを検証するためである．もし検証に合格すれば，そのハッシュ値を改ざんされ難い場所に保管する．また MLBR は定期的にリブートし，その時計測するハッシュ値と，格納されているハッシュ値と比較し，自身の統合性を検証する．

MLBR においては，以下 3 点を保証する仕組みが必要であるが，これらを TPM の機能を用いてどう実現するかについては，別途検討を要する重要な課題である．

1) ハッシュ値を計算し比較するプログラムが改ざんされないこと．
2) 認定機関にハッシュ値を送るプログラムが改ざんされ，初期値にすり替えられることや，不正な値にすり替えられないこと．
3) あるいはハッシュ値の保管場所の内容が改ざんされないこと．

ここでは，MLBR がその機能を実行するためのソフトウェア群がインストールされた直後に，それが改ざんされていないことを認定機関に確認すると同時に，改ざんが認められなければ，MLBR が提示する AIK 公開鍵に対してその証明書を認証機関が発行するためのプロトコルを提案する．本確認手続きを実現するプロトコルを，以後，統合性検証プロトコルと称する．図 3 に，提案する MLBR のインストールから，MLBR 間で共有鍵を設定するまでの一連の手続きを示す．なお，ここではディスクイメージとはシステムのメインメモリに展開されたプロ

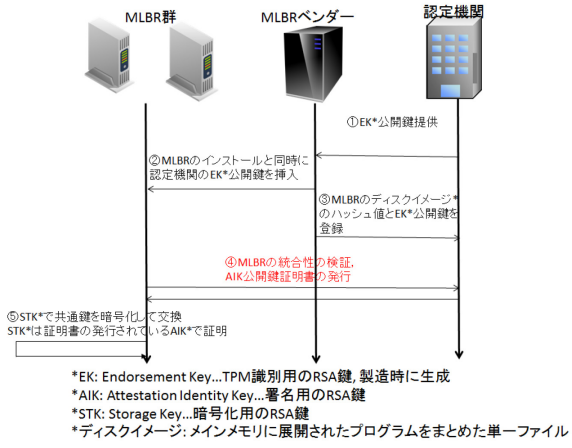

図 3 MLBR のインストールから MLBR 間で共有鍵を設定するまでの一連の手続き

グラム群をまとめた単一ファイルを指すものとする．

本手続きでは，図 3 の①～⑤の手続きを安全に実行し，MLBR の統合性検証を行うと同時に AIK の証明書を入手し，それをもとに MLBR 間で共有鍵の交換を行う．

① 認定機関 A はベンダーへ，認定機関 A の公開鍵EK_p^Aを提供する．

② ベンダーは MLBR M の製造時に，M に認定機関から提供された公開鍵EK_p^Aと M の秘密鍵EK_s^Mを挿入する．

③ ベンダーは MLBR M の製造時に秘密鍵EK_s^Mとその公開鍵EK_p^Mを生成し，M のディスクイメージのハッシュ値$h(D^M)$）と公開鍵EK_p^Mを認定機関 A に登録する．

④ MLBR M は起動時に，認定機関 A へメインメモリ上に展開したプログラム群から計算したハッシュ値と公開鍵AIK_p^Mを送り統合性の検証を行う．ここで MLBR M が正しいと判定されれば，認定機関 A は MLBR M に，受信した公開鍵AIK_p^Mのデジタル証明書を発行する．

⑤ 統合性を検証し改ざんがないと判定された MLBR M は，他の MLBR（ここでは L とする）にデジタル証明書と公開鍵STK_p^M，署名（STK_p^Mのハッシュ値をAIK_S^Mで暗号化したデータ）を送る．受信した L はデジタル証明書と署名の検証を行い，検証結果が正しければ，共有鍵を公開鍵STK_p^Mで暗号化して M に送信することで共有鍵の交換を行う．

ここでは，①～③の手続きを安全に実行するプロトコルが存在するとして，④の統合性検証を安全に実行するプロトコルについて特に考察する．MLBR の統合性や攻撃検知のための学習済みデータの統合性検証は，DRM を送る直前に行うことが理想であるが，DRM 送信が必要となっている状況では，帯域が圧迫されており，認定機関を介した統合性検証ができない可能性があるため起動時に行い，その後の統合性検証は MLBR 自身が，自身の TPM が保持するディスクイメージのハッシュ値と，その時点ごとの計測値を比較して行うものとする．ただし，DRM 送信時には，その時点でのディスクイメージのハッシュ値なども安全性向上のために添付する．

3.4.2 プロトコルで用いるメッセージの定義

MLBR の統合性検証プロトコルで使用可能なメッセージを定義する．以下で用いる論理式は，通常の集合に関する演算や述語を含む 1 階述語論理式である．小文字は変数であり，項目ごとに全称束縛されているものとする．

定義１：利用可能なメッセージ集合

このプロトコルで使用可能なメッセージの集合 S は，以下で帰納的に定義される最小集合 W の部分集合である．

i) $N_0, N_1, \cdots \in W$

ii) $\{EK_s^M, EK_p^M, AIK_s^M, AIK_p^M, EK_s^A, EK_p^A\} \subseteq KEY$

iii) $KEY \subseteq W,\ \{"M", "D^M"\} \subseteq W$

iv) $w \in W \rightarrow h(w) \in W$

v) $w_1,\ w_2 \in W \rightarrow < w_1,\ w_2 > \in W$

vi) $k \in KEY,\ w \in W \rightarrow \{w\}_k \in W$

ここで，Niはナンスと呼ばれる乱数を表す定数記号である．ii)は全ての鍵は集合KEYの要素であることを示す．iii）は，KEYの要素である鍵自身もメッセージであることを示し，また，MLBR の識別子，MLBR で実行するプログラム，“M”，“D^M”もメッセージであることを示す．iv）は，任意のメッセージにハッシュ関数を施した結果もメッセージであることを示している．v）は，w_1，w_2が共にメッセージであるならば，w_1，w_2のペア$< w_1, w_2 >$もメッセージであることを示す．この構成子は結合則を満たすとして，＜$w_1, w_2, \cdots, w_n$＞で，任意のペアの入れ子構造を表すものとする．vi）は任意のメッセージを，持っている鍵で暗号化した結果もメッセージであることを示している．

ここでメッセージ集合Sはこのプロトコルで使われる可能性のあるメッセージ全てを含むように定義しているが，実際にプロトコルの参加者が使用できるメッセージは，メッセージ送信時に保持している知識に依存し，知識は，メッセージを受信する毎に単調に増えてゆく．このプロトコルの各ステップiの終了後の各参加者が持つ知識は，ステップiで送信されるメッセージをm_iとするとき，以下のように帰納的に定義される．

定義２：参加者の持つ知識

ステップ i の通信が正常に行われた後の参加者 x の持つ知識 $K(x, i)$ は，以下で定義される最小集合 X である．

i) $K(x,\ i-1) \subseteq X,\ m_i \in X$

ii) $< w_1,\ w_2 > \in X \rightarrow w_1,\ w_2 \in X$

iii) $(k \in X \cap K \wedge \{w\}_k \in X) \rightarrow w \in X$

初期知識 K（x, 0）については，すでに確定していると仮定する．

参加者は得られたメッセージをもとに，それ自身およびそれがペアであればその要素を知識に追加する．メッセージが暗号化メッセージでありかつその復号鍵を知識として所有しているならば，平文も知識に追加することができる．また，インターネットを流れるメッセージは盗聴可能であるので，攻撃者についても，公開情報を初期知識として同様に知識が増えていく．

定義３：参加者が送信可能なメッセージ集合

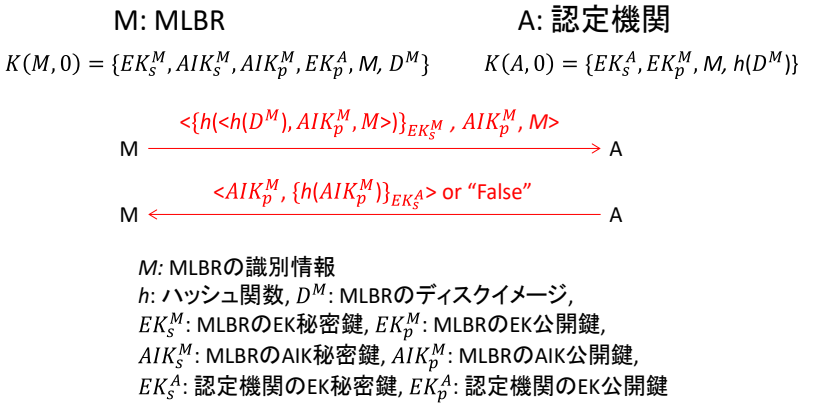

図 4　MLBR 統合性検証プロトコルの Alice Bob 記法による表記

参加者 x がステップ i で送信側であったとき，そこで x が送信可能なメッセージの集合 $M(x, i)$ は，その時点で持ち得る知識 $K(x, i\text{-}1)$ を基にして以下で定義される最小集合 X である．

i) $N_0, N_1, \cdots \in X$

ii) $K(x, i-1) \subseteq X$

iii) $m \in X \rightarrow h(m) \in X$

iv) $m_1, m_2 \in X \rightarrow <m_1, m_2> \in X$

v) $(k \in K \cap X \wedge m \in X) \rightarrow \{m\}_k \in X$

参加者は持っている知識をもとに，ハッシュ関数，ペアリング関数，そしてもし鍵を持っているならば暗号化アルゴリズムを用いてメッセージを構成することができる．そのとき必要な実引数となるメッセージは知識として所有していなければならない．ただし，実装上は TPM の暗号化能力に制限があるため，任意の長さのメッセージを暗号化できるわけではないため，注意が必要である．

3.4.3 MLBR の統合性検証プロトコル

MLBR の統合性検証プロトコルにおけるメッセージのやりとりを Alice Bob 記法で記述すると，その基本は図 4 のようになる．MLBR の統合性検証のプロトコルは 2 回の通信で行われる．1 回目の MLBR M から認定機関 A へのメッセージ送信では，M の秘密鍵EK_s^Mで署名を施したハッシュ値$\{h(<h(D^M)), AIK_p^M, M>)\}_{EK_s^M}$，M の公開鍵$AIK_p^M$，M の識別子 M をタップルにして送信する．このとき用いる$h(D^M)$は，MLBR の中で送信時点での内部プログラム群の計測値であるが，ベンダーから遠隔でインストールされたものであるため，そのインストールのためのプロトコルに瑕疵があった場合には，マルウェアを含むプログラムがインストールされている可能性がある．そのため計測値 $h(D^M)$を認定機関に送信し，統合性を検証する．

メッセージを受信した認定機関 A は，識別情報の値から送信元の MLBR を M であると仮定し，予め登録されている M の公開鍵EK_p^Mを用いて，M から送られてきた$\{h(<h(D^M), AIK_p^M, M>)\}_{EK_s^M}$を復号し，M のハッシュ値 $h(<h(D^M), AIK_p^M, M>)$を得る．そして，ベンダーから送られ登録されている M のディスクイメージのハッシュ値$h(D^M)$と，受信したAIK_p^Mと識別子 M をタップルにしてそのハッシュ値を求め，受信したハッシュ値と照合する．データオリジン真正性が確かめられた場合は，送信してきた M のこの通信における真正性とシステムとしての統合性が同時に検証できたことになる．その場合は，2 回目の通信でデジタル証明書として M の公開鍵

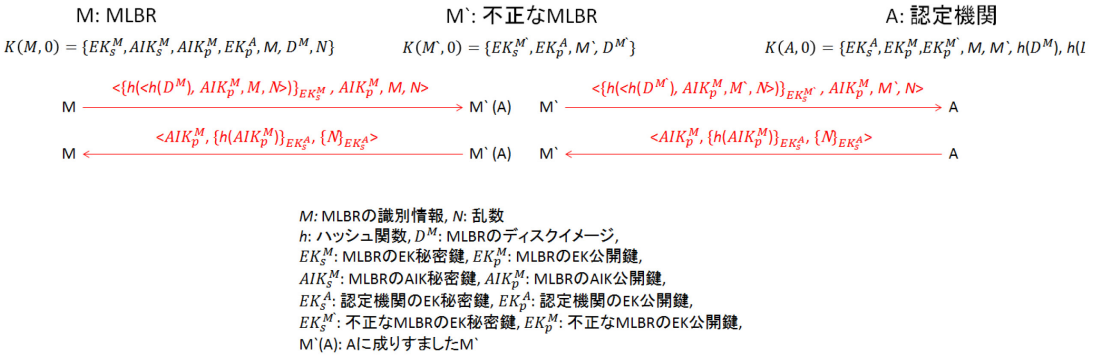

図 5　想定される中間者攻撃

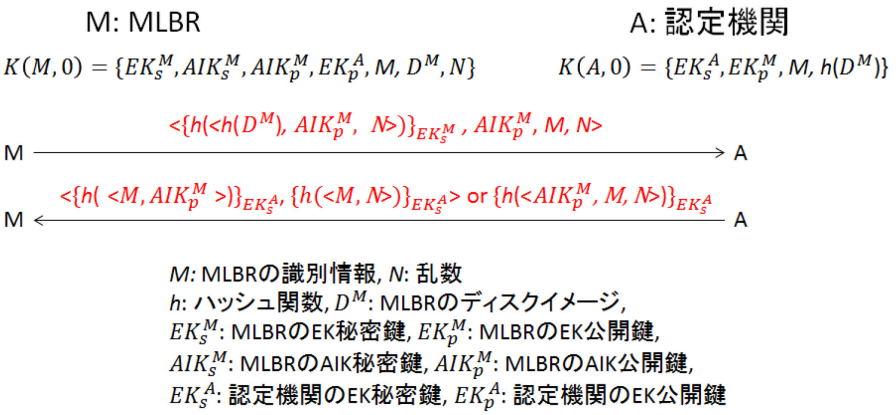

図 6　改良した MLBR 統合性検証プロトコルの Alice Bob 記法による表記

AIK_p^Mとそれのハッシュ値に認定機関の秘密鍵EK_S^Aを用いて署名した結果を送信する．一致しなければ検証失敗と判定しその結果を M に送信する．

しかしながら，このやり取りに対して図 5 に示すような中間者攻撃が考えられる．MLBR M が初めに送信したメッセージを，マルウェアが埋め込まれ認定機関に成りすました不正な MLBR M’ が受け取り，M’ はそのメッセージを改ざん・流用し，自らの統合性検証要求として認定機関に送信する．定義 3 にしたがえば，このようなメッセージを構成することが可能であることが分かる．システム M’ 自身は改竄されているが，認定機関に登録されている MLBR であるので，その正しい計測値を保持し続けている可能性がある．この正しい計測値$h(M')$を流用して，認定機関へ M の提供するAIK_p^Mと合わせて，認証機関に統合性検証と鍵証明書を要求し，その結果を M に転送することにより，M が改ざんされていたとしても，M の統合性検証が成功していると M に思わせてしまう可能性が考えられる．また逆に，M が改ざんされていない場合に，M が改ざんされているという報告を送ることも同様に可能である．

これらに対応するため，マルウェアに感染した MLBR による中間者攻撃の防止を目的とし，認定機関から MLBR への検証成功メッセージ，失敗メッセージのそれぞれに MLBR の識別情報を付与したものにサインをする．これにより，認定機関から攻撃者へのメッセージに攻撃者の識別情報 M’ が改ざんできない形で埋め込まれる．攻撃者はサインされた情報に手を加えることができないため，M にそのままの情報しか送ることができず，M は中間攻撃者が介在していることに気づくことが可能となる．

一方 AIK が複数セッションで流用される場合にはそこに流れるメッセージを記録・流用する再送攻撃もまた可能であると考えられる．例えば，本プロトコルでは，MLBR からの検証要求の後に，認定機関から MLBR へ検証結果が送信されるが，この時攻撃者が統合性検証失敗時のメッセージを観測可能である場合，認定機関から正規の検証結果にすり替えて，MLBR に統合性検証失敗時の認定機関のサインした結果を再送することで，MLBR の起動を妨害するという攻撃が成立する．また同様に失敗しているにも係わらず，以前成功したときのメッセージを再送することで，感染した MLBR を使い続けさせることも可能である．

そこで，通信時にナンス N を加えた通信内容とすることで，再送攻撃を防ぐことが可能なプロトコルとする．具体的には，初めの MLBR から認定機関へのメッセージに N を含め，$\{h(<h(D^M)), AIK_p^M, N>)\}_{EK_S^M}$とする．その後の統合性検証成功時に認定機関から MLBR へ送るメッセージに，N と MLBR の識別情報 M に対して A の署名を施したデータを追加する．検証失敗時に送信するメッセージは M の公開鍵AIK_p^Mのハッシュ値と，MおよびNを認定機関の秘密鍵EK_S^Aで署名を施したデータとした．これにより，A から M へのメッセージが現在のセッションのメッセージであるかを確認することができ，データオリジン真正性が保たれる．これらの対策を施した手続きを Alice Bob 記法で示すと，図 6 のようになる．

これに続く一連の手続きを行うことで，最終的に共有鍵を交換した状態になる．以降の通信では，この共有鍵を用いて IPsec における認証ヘッダ AH をパケットに付加する．

3.5　廃棄要請メッセージ生成・送信機構の検証機構

3.5.1 攻撃検知エンジンの統合性検証

廃棄要請メッセージ（DRM）は，サイバー攻撃を検知するための攻撃検知エンジンを持つサイトによって生成され，eMLBR に送信される．ここでは初めに，DRM を生成する攻撃検知エンジンの安全性を確認するための，統合性検証プロトコルを提案する．

提案プロトコルは図 7 に示す手順で行われる．なお，本プロトコルで用いるハッシュ関数 h は，各参加者が事前に安全な方法で共有しているものとする．

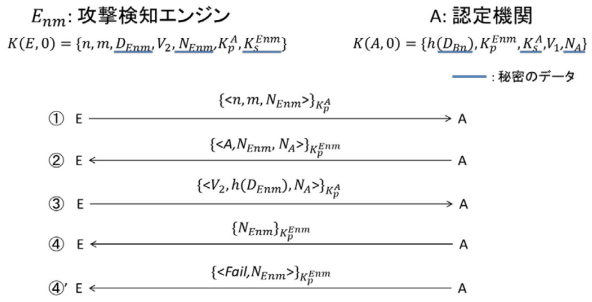

図 7 攻撃検知エンジンの統合性検証プロトコル

① 攻撃検知エンジンは自身を提供しているベンダー名とマシン識別子（ｎはマシンの型式番号，ｍはシリアル番号）とナンスを暗号化したものを検証要求として送信する

② 認定機関は自身の名前と受信したナンス，自身が生成したナンスを攻撃検知エンジンの STK 公開鍵で暗号化し，攻撃検知エンジンへ送信する

③ 攻撃検知エンジンは受信したものを TPM 内で復号すると同時に，学習済みデータからハッシュ値を計算し，そのハッシュ値と認定機関が生成したナンス，攻撃検知エンジンの学習済みデータのバージョン番号V_2を認定機関の STK 公開鍵で暗号化し認定機関へ送信する

④ 認定機関では受信したものを復号し，送られてきた学習済みデータのバージョンと自身が持っているバージョン番号V_1が一致しているか，また学習済みデータのハッシュ値が一致しているかを検証する．バージョンとハッシュ値が同じであった場合，検証が成功したことを送信する

⑤ バージョンが異なる，またはハッシュ値が異なる場合は検証失敗のメッセージを送信する

3.5.2 廃棄要請メッセージ（DRM）の送信プロトコル

攻撃検知エンジンから送られてきた DRM が偽装された物でないかを，eMLBR 側で確認するため，図 8 に示す DRM の送信プロトコルを提案する．ここで使われるメッセージは，鍵の集合として$\{AIK_s^A, AIK_p^A\}$，原子メッセージとして$\{D^A, DRM\}$を用いて，MLBR の統合性検証プロトコルの場合と同様に定義される．

プロトコルでは，初めに攻撃検知エンジンで DRM を生成し，そのハッシュ値$h(DRM)$を計算する．次に攻撃検知エンジンの統合性検証時に用いた学習済みデータのハッシュ値$h(D^A)$を TPM の PCR から取得し，$h(h(D^A), h(DRM))$を計算する．求めたハッシュ値にAIK_s^Aで署名を行って認証コードを生成し，DRM に付加して eMLBR に送信する．DRM と認証コードを受け取った eMLBR は受信した DRM からハッシュ値$h(DRM)$を求め，登録されている学習済みデータのハッシュ値を用いて認証コード$h(h(D^A), h(DRM))$を求める．その後攻撃

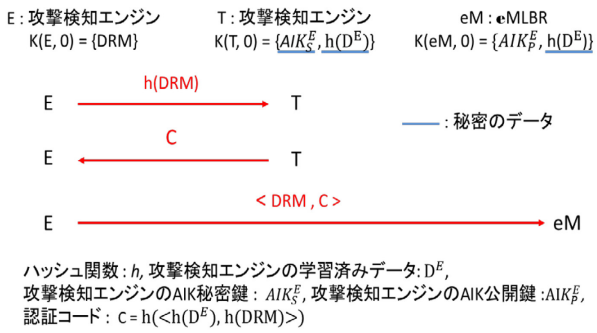

図 8 DRM 送信プロトコルの Alice Bob 記法による表記

検知エンジンの公開鍵AIK_p^Aを用いて署名を復号する．最後に受信した DRM から計算した$h(h(D^A), h(DRM))$と署名復号で得られた認証コード$h(h(D^A), h(DRM))$を比較して認証コードの検証を行う．検証が正常に実行されれば，受信した DRM が偽装されていないことが確認できる．

一方，この手続には中間者攻撃が成立する．そこで以下の手続きの検討を行っている．手続きの Alice Bob 記法による表記を図 9 に示す．検討を行っているプロトコルを実行するための前提として，企業ユーザ E と eMLBR 間で共有鍵CK^{EeM}の交換をし，そのときに使用したナンス $N1$ をそれぞれ保持しているものとする．企業ユーザ E に搭載されている攻撃検知エンジンが攻撃を検知すると，E は攻撃の属性情報を用いて DRM を生成する．さらに，E は，自身の統合性を示すためのディスクイメージのハッシュ値，正しく攻撃を検知したことを示すための攻撃検知エンジンの学習済みデータ，平文で送る DRM の改ざん防止のための DRM のハッシュ値，これら 3 つのデータからハッシュ値を求める．加えて，再送攻撃対策のため，新たなナンス $N2$ を生成し，共有鍵交換時に使用したナンス $N1$ とペアにして，共有鍵で暗号化する．E は，これらの DRM，ハッシュ値，暗号文と E の識別情報を eM に送信する．受信した eM は，識別情報から E であることを判断し，登録されているディスクイメージ，学習済みデータ，受信した DRM からハッシュ値を求め，受信したハッシュ値と比較し検証する．検証結果が正しかった場合は，暗号文を復号してナンス $N1$，$N2$ を得る．ここで，復号して得たナンス $N1$ と所持していたナンスが等しければ，送られてきた DRM は，再送攻撃ではないと判断し，DRM の処理を行う．その

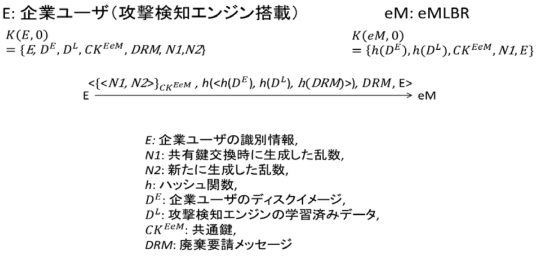

図 9 改良した DRM 送信プロトコルの Alice Bob 記法による表記

後 eM は，ナンス $N1$ を消去し，ナンス $N2$ を記憶する．これにより攻撃下でのチャレンジレスポンス方式や送られてきたナンスを全て記憶するといった方式を用いることなく再送攻撃対策を行うことができる．

4. 実装実験

4.1 MLBR のソフトウェア改ざん検知プロトコルの検証

3.4 で述べたプロトコルを実装するにあたって，実装上の問題点を検討するために，プログラムによる実装実験を行った．本実験では，TPM を C#で作成したプログラムから使用するためのライブラリである TSS.NET を利用しプログラミングを行った．また，実験に用いる TPM モジュールには，TPM シミュレータ[18]を使用した．

実装プログラムの具体的な処理シーケンスを図 10 に示す．初めに，MLBR でディスクイメージD^Mの計測を行い，そのディスクイメージのハッシュ値$h(D^M)$を計算する．次にAIK_p^Mを生成し，ディスクイメージのハッシュ値$h(D^M)$と MLBR の公開鍵AIK_p^Mおよびナンス N からハッシュ値$h(< h(D^M), AIK_p^M, N >)$を求める．求めたハッシュ値に MLBR の秘密鍵$EK_S^M$で署名を行い，その結果と$AIK_p^M, M$をタップルにして検証要求として認定機関に送信する．

検証要求を受け取った認定機関は，登録されている MLBR の EK 公開鍵EK_p^Mで署名の復号を行う．また，登録されているディスクイメージのハッシュ値 $h(\mathrm{D^M})$と受

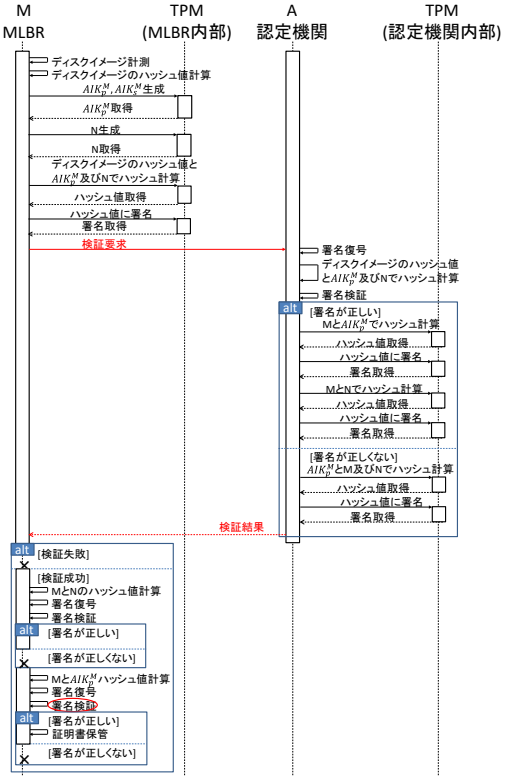

図 10　実装プログラムのシーケンス図（MLBR）

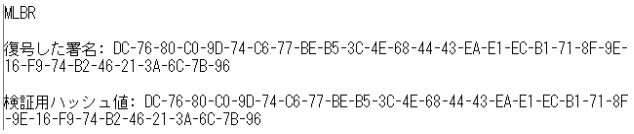

図 11 実験結果（MLBR）

け取ったAIK_p^Mと Nからハッシュ値$h(< h(D^M), AIK_p^M, N >)$を計算し，署名を復号して得た値と計算したハッシュ値$h(< h(D^M), AIK_p^M, N >)$を比較し，一致しているかを判断する．一致していれば，受信したAIK_p^Mと M のハッシュ値にEK_S^Aを用いて署名を行った結果，および M と N のハッシュ値を計算しEK_S^Aで署名を行った結果をタップルし MLBR に送信する．一致していなかった場合は，AIK_p^Mと M，および N のハッシュ値を計算し，それに認定機関の公開鍵EK_S^Aを用いて署名を行う．その後，検証結果として署名を行ったデータを MLBR に送信する．

メッセージを受け取った MLBR は，受信内容が検証成功か，失敗かを確認する．確認のために MLBR は受信したデータを登録されている認定機関の公開鍵EK_p^Aで署名の復号を行う．検証結果が失敗であった場合は，プログラムを終了する．成功であった場合は，受信した M と N のハッシュ値と，MLBR が所持している M と N のハッシュ値を比較し，現在の検証要求に対する応答であるかどうかを確認する．その後所持しているAIK_p^Mと M のハッシュ値を求め，署名を復号して得た値と求めたハッシュ値が一致しているか確認し，証明書の検証を行う．

以上の実験を行った結果を，図 11 に示す．署名を復号して得た値と，検証のために求めた値が一致し正しいことが確認でき，提案プロトコルによる認証が適切に行われていることが確認できた．

4.2 廃棄要請プロトコルの実装

3.5 で述べたプロトコルを実装するにあたって，実装上の問題点を検討するために，廃棄要請メッセージ（DRM）送信システムのプロトタイプの実装実験を行った．実装には 4.1 と同様に，TSS.NET と，TPM シミュレータを使用した．

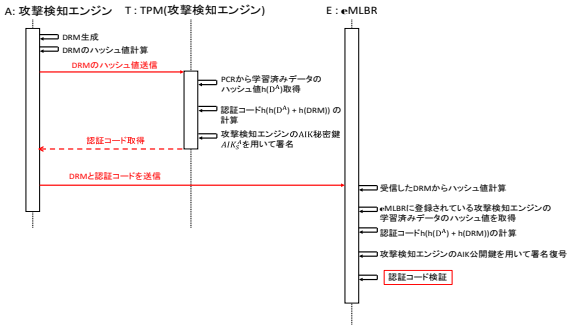

図 12 実装プログラムのシーケンス図（DRM）

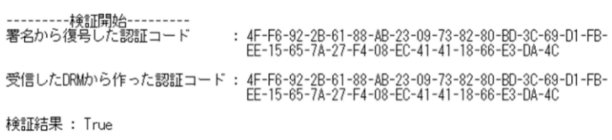

図 13　実験結果（DRM）

実装プログラムの具体的な処理シーケンスを図 12 に示す．実験の結果，図 13 に示すとおり，署名復号によって得られた認証コードと DRM から生成した認証コードが一致していることが確認でき，提案プロトコルの TPM による実装が動作したことを確認した．

5. まとめ

本稿では，“インターネット全体で守るセキュリティ”の実現を目指した“自律分散型インターネットセキュリティ基盤”を信頼に足るものとするための信頼基盤の提案と検討を行った．AIS 基盤を支える機器や攻撃を遮断するために AIS 基盤内で用いられるプロトコルなどを対象としたサイバー攻撃を成立させないための，核となる要素の統合性検証プロトコルを提案し，プロトコルに対するサイバー攻撃を想定し，その攻撃が成立しないようにプロトコルを設計した．さらにプロトタイプシステムを用いて実現可能性を検証した．その結果，セキュリティモジュールに TPM を用いることで，プロトタイプシステムが実現可能であることを確認した．また提案するプロトコルによって，MLBR，DRM の統合性検証が行えることを確認した．

今後の課題として，提案プロトコルでは攻撃が成立しないことの形式検証を行うことや，DRM 送信プロトコルの改良とプロトタイプ実装実験，プロトコルを AIS 基盤に実装し形式手法による実システムの脆弱性検出や実験により安全性を検証することが必要である．また，今回はセキュアモジュールに TPM を用いたが，それ以外のセキュアモジュール，例えば HSM（Hardware Security Module）を用いた実装の検討を行うことも課題の一つと考えられる．

参考文献

[1] "2016 年第 3 四半期「インターネットの現状／セキュリティ」レポート,",https://content.akamai.com/PG7516-Q3-2016-SOTI-Security-Report.html?gclid=COOyht7H4dECFYSUvQod2-UD_Q, （オンライン）,2016.
[2] 小林浩，八槇博史，上野洋一郎，佐野香，佐々木良一，“自律分散型インターネットセキュリティ基盤の実用性検討，”信学技報， ISEC2015-24/ SITE2015-33/ LOIS2015-40， （2015）.
[3] 小林浩，八槇博史，末廣友貴，上野洋一郎，佐野香,佐々木良一,“マルチレイヤ・バインディング・ルータによるサイバー攻撃対策の提案と，OpenFlow を用いた実装評価，” 情処研報，Vol.2014-CSEC-66，No.51，pp.1-8，（2014）.
[4] 小林浩，八槇博史，末廣友貴，上野洋一郎，佐野香，佐々木良一，“マルチレイヤ・バインディング・ルータによるサイバー攻撃対策の提案，”信学技報，IA2014-14，pp.1-6， （2014）.
[5] 江口健，長友勝太，横山裕士，佐野香，八槇博史，上野洋一郎，小林浩，“マルチレイヤ・バインディング（MLB）ルータによるサイバー攻撃対策技術 -廃棄要請プロトコル（DRP）の提案と，OpenFlow を用いた実装評価-，”信学技報，ICSS2014-65，pp.13-18，（2015）.
[6] 小林浩，八槇博史，上野洋一郎，佐野香，佐々木良一，“自律分散型インターネットセキュリティ基盤の実現性検討，” 信学技報，ISEC2015-24/ SITE2015-33/ LOIS2015-40，pp. 67-74，（2015）.
[7] 山本優人,川下壮周,田中伸吾,八槇博史,小林浩，“MLB ルータを用いた通信相手限定によるマルウェア汚染 IoT の無害化，”信学技報，IA2015-86，pp.43-48，（2016）.
[8] 岸有哉,江口健,石川毅,鈴木竜生,宮口侑己,大澤一生,與五澤守,佐野香,八槇博史,上野洋一郎,佐々木良一，小林浩，“自律分散型インターネットセキュリティ基盤を模擬したテストベッドでの DDoS 攻撃の遮断実験，”信学技報, IA2016-9, pp.45-50, (2016).
[9] 宮口侑己，江口健，岸有哉，與五澤守，工藤渉，鈴木竜生，大澤一生，石川毅，芦川大地，堤智昭，佐野香，八槇博史，上野洋一郎，佐々木良一，小林浩，“自律分散型インターネットセキュリティ基盤を模擬したテストベッドでの帯域幅攻撃の遮断実験，” 信学技報，IA2016-14，pp.7-12，（2016）.
[10] 與五澤守,宮口侑己,江口健,岸有哉,石川毅,堤智昭,佐野香,八槇博史,上野洋一郎,君山博之,米崎直樹,佐々木良一,小林浩,”攻撃パケットの送信元アドレスと属性情報の特定による廃棄要請メッセージの生成”, 信学技報, vol. 116, no. 362, IA2016-69, pp. 35-40, （2016）.
[11] 君山博之，江口健，堤智明，佐野香，八槇博史，上野洋一郎，米崎直樹，佐々木一郎，小林浩，"自律分散型インターネットセキュリティ基盤とその攻撃遮断法，"電子情報通信学会信学技報,NS2016-251, pp.541-546 (2017).
[12] 小林浩,八槇博之,米崎直樹,君山博之,上野洋一郎,堤智昭,佐野香,佐々木良一, "自律分散型インターネットセキュリティ基盤の概要と実現課題," サイバーセキュリティシンポジウム 2017 in TDU, （2017）.
http://web.dendai.ac.jp/souken/about/index_1.html
[13] F. Baker, and P. Savola, "RFC 3704 - Ingress Filtering for Multihomed Networks,", https://tools.ietf.org/html/rfc3704, Mar. 2004.
[14] Internet Initiative Japan Inc., "送信元検証「Source Address alidation」最新の技術・取り組み| IIJ," http://www.iij.ad.jp/company/development/tech/activities/sav/, Jul. 2012.
[15] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch,and, D. McPherson, "RFC 5575 - Dissemination of FlowSpecification Rules," https://tools.ietf.org/html/rfc5575.
[16] "NTT 西日本のネットワークサービス網に適用した DDoS 攻撃対策," NTT 技術ジャーナル, Vol.28, No.3, （2016）.
[17] TCG EK Credential Profile For TPM Family 2.0; Level 0, Specifiation Version 2.0, Revision 14, 4 November, 2014.
[18] TPM 2.0 Simulator ,https://www.microsoft.com/en-us/download/details.aspx?id=52507