Vol.2019-ITS-78 No.7
2019/9/6

Pattern Match Accelerator を用いた Edge 向け Malware Cyber-Security の検討

柏山　正守*，関　弘翔，細野　裕行（日本大学）

A Study on Malware Cyber-Security for Edge utilizing Pattern Match Accelerator

Masamori Kashiyama, Hiroto Seki, Hiroyuki Hosono (Nihon University)

Malware detection algorithm to be embedded in edge computing is proposed and validated by the emulator. This algorithm with pattern match accelerator reduces the computing cost while maintaining the relatively high detection accuracy. There efficiencies were identified as the performance of approximately 80%. In comparison with the characteristic extraction using the AI, computing cost is reduced, and these processes allow realization of the edge computing with high cybersecurity characteristics.

キーワード：サイバーセキュリティ，マルウェア，エッジコンピューティング，パターンマッチアクセラレータ，テクスチャイメージ，高次局所自己相関
(Cyber-Security, Malware, Edge Computing, Pattern Match Accelerator, Texture Image, Higher-order Local Auto Correlation)

1. はじめに

近年，Edge(IoT を含む)を標的とした Cyber-Security のリスクが増大している。Cyber-Security は，Malware 感染，不正アクセス，DDoS 攻撃などからシステムを防御することであり，これらの脅威は年々増加傾向にある[(1)(2)]。具体的な数字で見ると，IoT 機器の数は 2020 年時点で約 400 億個にまで急激に増加すると予想されている。このような拡大の中で，2017 年，IoT 機器を狙ったサイバー攻撃は，700 億パケットに上り，2015 年比で 5.7 倍に達した[(1)(2)]。このような変化の中で，Edge を標的とした Malware の急激な増加は，Connected Car にも大きな脅威であり，ICT 高度化する自動車を狙った Malware の増加も懸念される。

将来，自動車は，自動運転制御や先進安全運転支援などの高度な IT 機器を多数内蔵することから，組み込まれているプログラム量も膨大なステップになり，OTA(On the Air)による機能アップデートやディーラー作業による不具合修正などの組み込みソフトウェア更新も頻繁に実行される。このような技術進化のもと，ソフトウェア開発や工場生産現場におけるプログラム注入作業の過程において，Malware の混入や悪意のある故意の組み込み，それらが数年後に活動を開始するような潜伏型 Malware が忍んでいることなど，全体の製造工程，修理メンテナンス，中古利用などのライフサイクルの中で生じる Malware 混入防止のセキュリティ脆弱性を完全に担保することは難しい。

Cyber-Security のためには，①ネットワークを介した侵入検知，②組み込み済みプログラムのスクリーニングパトロール処理を Edge で実行することが必要である。例えば，Malware 侵入防御・車内探索を目的として Security Gateway(Fig.1)を車載する手段が考えられているが，これらの Malware 検知手段は，システム運用設計段階でファイアウォールや検知機能を追加する手法や Cloud 側の AI(Artificial Intelligence)を用いた統計的解析処理による手法が一般的である。しかし，これら後付けの対策や複雑な AI 処理等を用いた手法は，計算コスト(処理時間・消費電力と熱・プログラムサイズと複雑さ・物理的サイズ)の問題からリアルタイム自律検出処理に限界がある。特に Edge においては，予め組み込んだ Cyber-Security 対策が重要であり，ハードウェアとソフトウェアが協調した検出実行環境を持つ

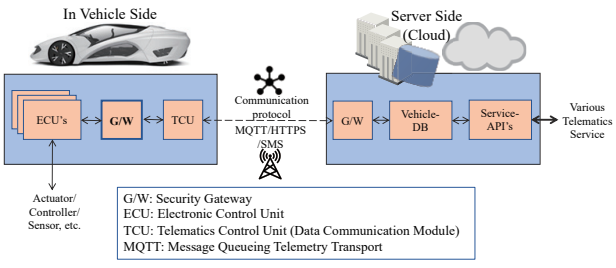


図 1 車載セキュリティゲートウェイ

Fig. 1. Security Gateway Embedded in the Vehicle.

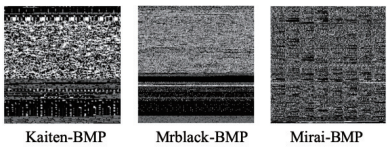


図 2 Malware Texture Image 例 (Kaiten, Mrblack, Mirai)

Fig. 2. Examples of the Malware Texture Images.

ことが必要である。

そこで我々は，Malware を Texture Image (Fig.2)として抽象化し，Pattern Match Accelerator を用いた多重テクスチャ解析を行うことで，Edge への適用が可能で低い計算コストを実現するシンプルな検出手法を発想した。

従来，パターンマッチングは非常に高負荷な処理と考えられてきたが半導体技術の進化により，汎用プロセッサに Pattern Match Accelerator を組み合わせたヘテロジニアスコンピューティング(Heterogeneous Computing)環境は，非常に低計算コストで Malware Texture Image の全スキャン実行が可能である。

提案手法では，高次局所自己相関(HLAC: Higher-order Local Auto Correlation)から得られるマスクパターンを用いて Malware Texture Image の構造レベル解析を Pattern Match Accelerator で行い，パターンマッチングから得られる Malware Texture Image 固有のパワースペクトル特徴量(Power Spectrum Features)を主成分解析するアルゴリズムを用いて Malware 識別を実現した。提案手法の Malware 検出システムは，80%程度の識別性能を持つ。これらの性能評価は，6 種類の Malware ファミリー群，641 サンプルを用いてエミュレータシステムで確認した。さらに，既往研究結果から導出した計算コスト概算比較では，一つの Malware ファイルの識別処理は，数 100μsec オーダー程度にまで縮退させることが可能である。提案手法の実証結果より，自動車などに搭載される Edge システムが求める重要な要件が実現できる。具体的には，Pattern Match Accelerator の適用と新しい提案アルゴリズムの採用により，Malware 検出を目的とした低消費電力かつ高速処理の組み込みシステムの実現が可能になる。

2. 提案手法

〈2･1〉 Malware 特性を考慮した構造的解析 Edge での Malware 検出は，リアルタイムで Malware ファイルの画像化を行い Texture Image(Fig.2)として，パターンマッチやパワースペクトル解析を組み合わせる手法を選択した。また Malware ファイルの機械語命令列は，自然な Texture Image と比較し，明確な配列の規則性を持つことから，構造レベルの解析手法を用いて特徴量を抽出することが有効であると考えた。

各々の Malware ファミリーは，原種のプログラムコードの特質を引き継いでおり，Malware コードのコアとなる機械語命令列は，プログラムの何処かに必ず記述されている。難読化の順序入れ替えや，データ挿入などのテクニックを用いても，プログラムコードの中の何処かのアドレス位置に存在する。従って，Malware の画像化後の配列規則は，プログラムコードの機械語命令列群が，ある種の目的を持った処理を行うための固有コードとして局所的に類似な Texture Image を形成していると考えた。このことは即ち，局所的なパターンとして Texture Image の中に存在すると言うことであり，逆アセンブラーを用いて，この機械語命令列が何を意味しているのか解析することと，このセクションのパターンは何かという特徴を見つける方法と等価である。さらに，ファミリー毎各々に原型となる固有パターンのレイアウト(位置)とテクスチャー(パターン)を比較すれば，ファミリー毎の分類が可能であると仮定した。

〈2･2〉 高次局所自己相関特徴によるマスクパターン創出 Malware 原種のプログラムコードからどの機械語命令列が最適なマスクパターンとなるコードか特定することは困難である。さらに，大きな課題として，正常ファイルとの識別や他の Malware ファミリーとの分類は不可能であり，より汎用性のあるマスクパターンを考える必要がある。

本提案では，高次局所自己相関特徴(HLAC Features)の Eq.(1)を次数 M=1 から M=7 まで拡張し，算出される 3×3 サイズのマスクパターン 221 種(Fig.3: HLAC マスクパターン 221 種)を用いて，Malware の Texture Image 全体をパターンマッチ処理することにより，局所的類似性と大局的類似性を比較する構造レベルの解析手法を導入する[4]。ここで Eq.(1)は，着目点rとrからの相対的な変位方向を$(a_1, \ldots, a_M)$として定義したものである。

$$v_M(a_1, \ldots, a_M) = \int_{\mathcal{R}} I(r)I(r+a_1)\ldots I(r+a_M)dr \quad \cdots (1)$$

Malware Texture Image の構造レベル解析における識別能力向上には，多様なパターンが有効であるが，機械語命令列の並びを表現するためには，3×3 程度の局所的な構造解析が適していると判断し，このサイズを導入した。

一方，構造解析による大域的特徴は，加法性と位置不変性の性質から HLAC のマスクパターンで Malware Texture Image 全体をスキャンしたことと等価であり，パターンマッチ結果の頻度(HLAC マスクパターン種毎の累積度数)をヒストグラム化したものは，その画像が持つ性質を数値として表現することが可能である。

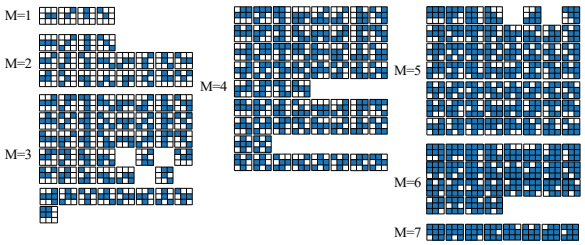


図 3 パターンマッチングのためのマスクパターン

Fig. 3. Mask Patterns for Malware Pattern Matching.

(221 Mask Patterns, 1st-order to 7th-order HLAC Features, 3×3 pixels)

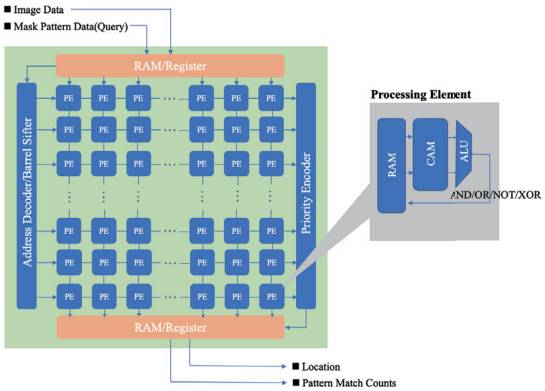


図 4 Pattern Match Accelerator 概略

Fig. 4. Overview of the Pattern Match Accelerator.

〈2･3〉 Pattern Match Accelerator を適用したアルゴリズム Malware Texture Image 全体を HLAC マスクパターン 221 種でパターンマッチさせる処理は，非常に大きな計算コストが発生する。処理系システム全体は，シンプルな構成と計算アルゴリズムを用い，負荷の重い処理を Pattern Match Accelerator に担わせるアルゴリズムを導入する必要がある。

本提案では，文献(5)〜(7)で Inoue, Pham らが提案した Pattern Match Accelerator(Fig.4)を高負荷処理へ導入し，極めてシンプルかつ短時間で完了させる。ここで，システムの計算コストは，目的の処理時間と消費電力である。

Malware Texture Image 全体を HLAC マスクパターン 221 種でパターンマッチさせた結果は，HLAC マスクパターン種毎の累積度数をヒストグラム化させ，Fig.5 で示す相関に従い Malware の解析を行う。構造レベルの解析①(Structural Analysis①)は，テクスチャを構成する Primitive を命令列群と考えて HLAC パターンを用いて構成要素の形と量に分解する。①で算出した形と量は，統計レベルの解析②(Statistical Analysis②)へ入力する数値としてヒストグラムを適用し求める。このヒストグラムの分布は，パワースペクトル/時系列データの統計量となっており，ベクトル解析することでデータが持つ性質の分類と識別を行う。これは，HLAC が持つ加法性と位置不変性の性質からヒストグラムは，改変亜種を含む Malware ファミリー群の Malware コードのコアとなる機械語命令列がプログラムコードの何処かに必ず記述されていることを特徴として抽出し，ヒストグラムが表現する数値的分布は，同じ傾向を示すと考えたからである。即ち，Malware Texture Image の局所的な構造解析データの統計結果は，類似する機械語命令列の分布に等しく，画像全体が持つ大局的特徴は，パワースペクトル/時系列データに類似した特徴を持つと推測する(Fig.5)。よって，本提案のアルゴリズムにより計算されたマッチングデータは，パワースペクトル/時系列データの個別ベクトルの相関解析を適用することで Malware ファミリーの分類と Malware 判別が可能になる。

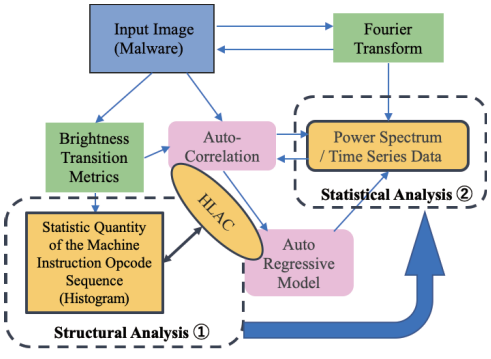


図 5 Malware 解析に適用した統計の関係

Fig. 5. Relationship of Statistics applied to Malware Analysis.

表 1 Malware ファミリーの評価サンプル

Table 1. The Malware Family Samples for Validation.

File		Feature	Bitmap type	Quantity
Malware Family Name	Gafgyt	Botnet (The most famous).	.png	142
	Grip	Distributed Denial Service Attack.	.bmp	41
	Kaiten	Botnet for IoT. Closely Related to the Mirai.	.bmp	109
	Mrblack	Linux Spike Trojan Malware.	.bmp	99
	Mirai	Botnet (The most famous).	.png	134
	Ganiw	Create a Backdoor. Springboard of DDoS Attacks.	.bmp	116
Not Malware	Normal	A Normal Application Running on Linux.	.bmp	110

3. 有効性検証

〈3･1〉 評価方法 提案の目的を達成するためには，Pattern Match Accelerator を適用した Malware 検出技術の新しいアルゴリズムが，Edge の要件を満たす Malware 識別性能を有するのか検証する必要がある。そのため，表 1(Table 1)に示した複数の代表的な Malware サンプルファミリーのファイルを被検体データとして用意した。さらに，正常ファイルとの比較による識別も必要なことから，Linux 上で動作する通常のアプリケーションを Normal ファイルとして用意した。これらの Malware サンプル(6 種 641 個)は，Symantec 社のウイルススキャンを用いてシグネチャによるファミリー分類を施した検体サンプルである。いずれのサンプルも，IoT や Edge デバイスを標的とした Linux 上で動作する Malware ファイルである。

Fig.6 は，評価方法のプロセスフローを示したチャート図である。第 1 次の抽象化(Class1)として，Malware バイナリ

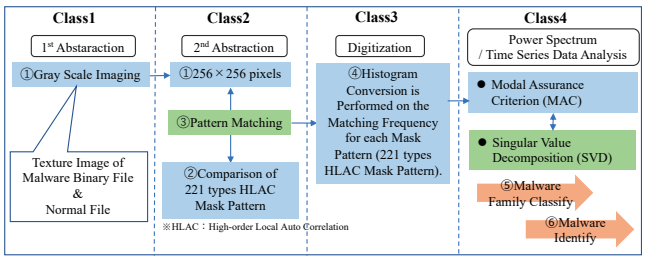


図 6 アルゴリズムのプロセスフローチャート

Fig. 6. Process Flow Chart of the Algorithm.

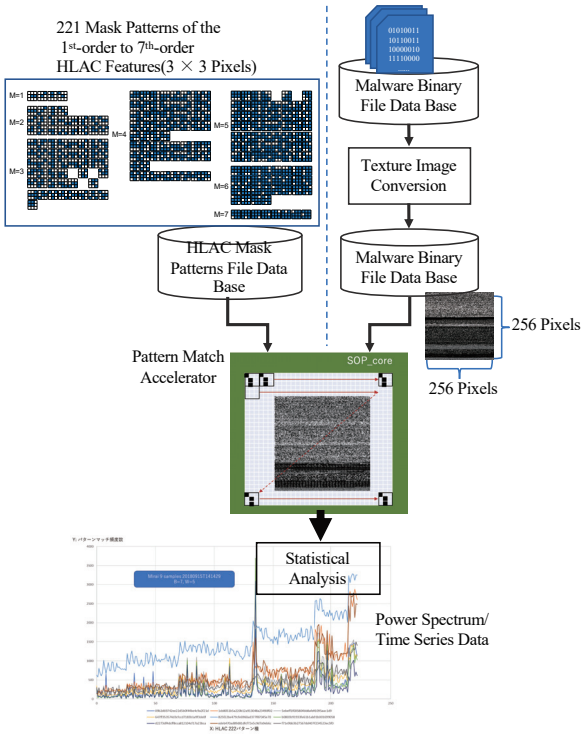


図 7 実験システムの構成

Fig. 7. The Configuration of the Experimental Systems.

ーファイルを 256×256 Pixels のグレースケール画像化(Texture Image 化)を行いデータベース化する。一方，パターンマッチのマスクパターンは，HLAC から算出された M=1 から M=7 次元のマスクパターン 221 種をハードコーディングで用意し，定数テーブル化する。第 2 次の抽象化(Class2)では，Pattern Match Accelerator を用いて Malware サンプルファイル毎にパターンマッチの演算を実行する。

パターンマッチ演算は，1Pixel 毎の Bin で照合処理を実行し，HLAC マスクパターン 221 種毎に対する Malware ファイルのマッチ結果が，ファイル固有の数値データとして生成される。Class3 では，この得られた数値データに基づいて X 軸(横軸)にマスクパターン 221 種をとり，Y 軸(縦軸)に個々のマスクパターンに対応した出現頻度をとる。このようにして得られたヒストグラムは，Malware 固有のパワースペクトル/時系列データとして表現できる。Class4 において，これらパワースペクトル/時系列データの個別ベクトル相関解析および主成分ベクトル抽出による相互相関解析を経て，Malware ファミリーの分類，識別処理を行う。

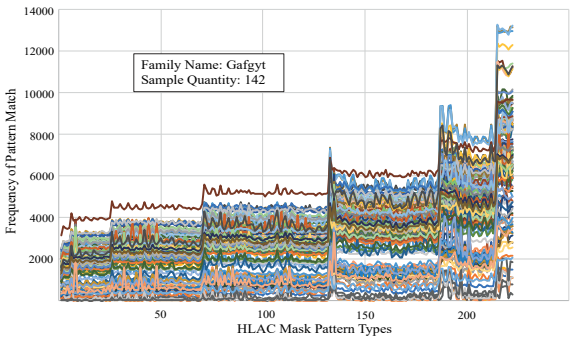
〈3･2〉 実験システムの構成 Fig.7 は，本提案の有効性検証システム概要を示したものである。検証システムの Pattern Match Accelerator は，提案処理方式のアルゴリズム検証を目的としたため，検証システム全体のエミュレータを構築する手段を導入し検証実験を行っている。

エミュレータに適用した Pattern Match Accelerator(Fig.4)は，文献(5)～(7)で提案されているエミュレータコア(C#で記述)を適用した。既往研究結果[6][7]から，Malware 画像全体 256×256 Pixels 画像に対する 1 Query(マスクパターン)の照合処理性能は，実 ASIC(Application Specific Integrated Circuit)チップでは，～5μsec という非常に高速な処理が実現可能である。

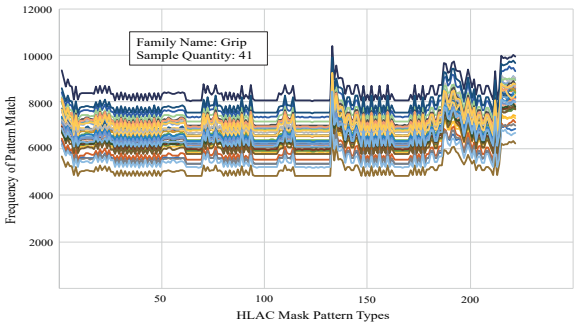
Pattern Match Accelerator を制御する周辺の Class1 グレースケール化画像処理部，HLAC マスクパターンテーブルからのデータ読みだし処理部，照合指示発行部，パターンマッチ処理結果の統計量処理部などのサブシステムは，Python で記述しており，実装段階では ARM 系および RISC-V などの汎用プロセッサで処理することを想定した構成である。

〈3･3〉 実験結果解析 Fig.7 に示した Malware 識別システムが Edge に必要な識別精度 80%を有することを，検証システム全体のエミュレータを用いた実験で示す。

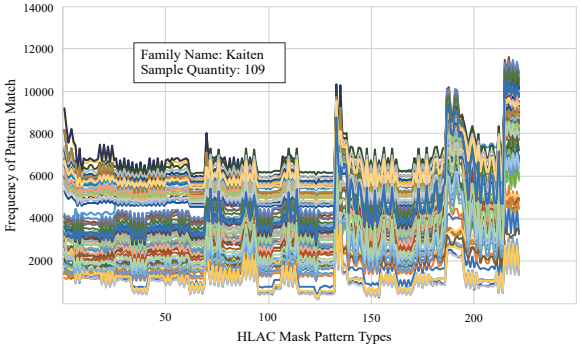
Fig.2 のようにグレースケール画像化した 751 個のサンプル(641 個の Malware ファイル，110 個の Normal ファイル)に対して Fig.3 に示す 221 種類の HLAC マスクパターンでパターンマッチングした結果の頻度をヒストグラム化したものを Fig. 8 に示す。横軸は HLAC マスクパターンの番号，縦軸はパターンマッチ頻度で，Malware ファミリー毎にまと



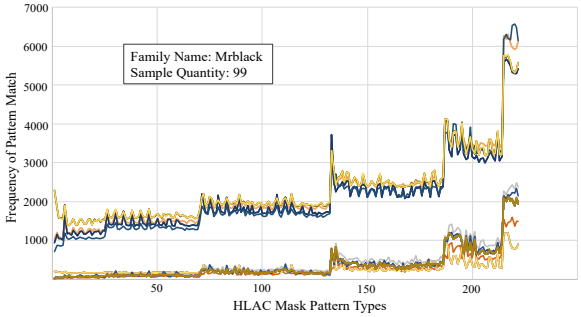
(a) Gafgyt Samples.



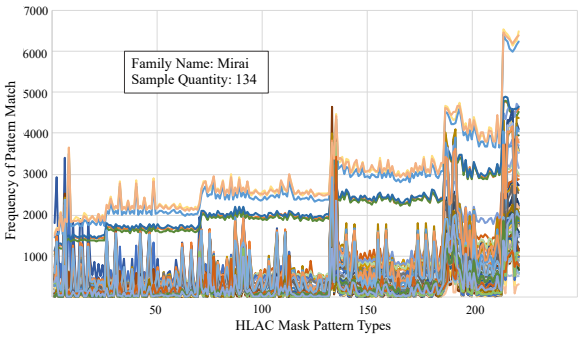
(b) Grip Samples.



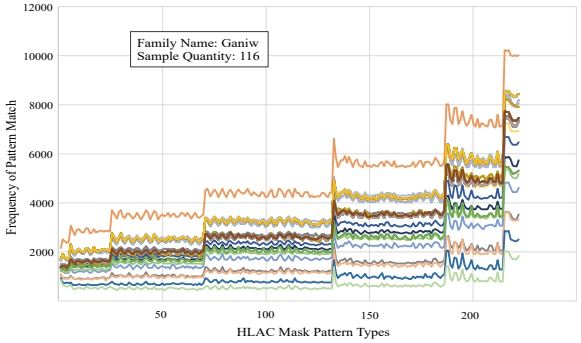
(c) Kaiten Samples.



(d) Mrblack Samples.



(e) Mirai Samples.



(f) Ganiw Samples.

図 8 Malware のパターンマッチ結果ヒストグラム

Fig. 8. Histogram of Pattern Match Frequency of Malware.

めている。Fig. 8 のヒストグラムから，それぞれのファミリー毎に波形やピークに特徴があること，少数ではあるが所属ファミリー内の特徴とは異なる波形を有するサンプルが

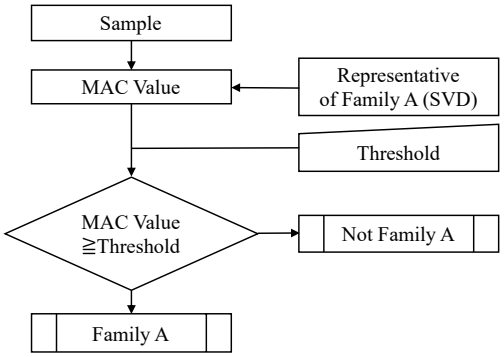


図 9 Malware 識別フローチャート

Fig. 9. Flow Chart of Malware Identification Program.

存在することが視覚的に理解できる。そこで，我々はヒストグラムの波形をパワースペクトルとみなし，モード信頼性評価基準(MAC: Modal Assurance Criterion)によってサンプル間の相関関係を定量的に評価する。自作した Malware 識別プログラムのフローチャートを Fig.9 に示す。本プログラムは，マルウェアがファミリー(Gafgyt, Grip, Kaiten, Mrblack, Mirai, Ganiw, Normal)のいずれか，もしくは何にも所属しないと識別する。具体的には，サンプルのヒストグラムの生データとあるファミリー(ファミリーA)の代表データのヒストグラムの MAC 値を算出する。ここで，ファミリーの代表データは，ファミリーのサンプルを特異値分解(SVD: Singular Value Decomposition)によって特徴を抽出し，かつその中で MAC 値が最大のものを選定している。算出した MAC 値と手動で設定した閾値を比較することで，MAC 値が閾値と等しいか以上，つまりファミリーA との相関関係が基準と等しいか以上ならばサンプルはファミリーA に所属と判断し，MAC 値が閾値以下，つまりファミリーA との相関関係が基準以下ならばサンプルはファミリーA に非所属と判断する。以上の処理をファミリー全てに対して行う。ここで，全てのサンプルにおいて，所属していないファミリーへの所属判定が出ないように閾値設定を行った。つまり，本プログラムでは全てのサンプルは所属ファミリーもしくは何にも所属しない，と判定され，非所属ファミリーに所属している，とは判定されない。正しい所属ファミリーに所属していると判定されたサンプルの数の割合が識別性能となる。

以上で設定したファミリー代表データと先述した閾値を用いて，Fig. 9 に示すプログラムによってサンプルのマルウェア識別を行う。

〈3･4〉 結果と考察　表 2(Table 2)は，各 Malware 種 641 個のサンプルを 2 分割し，322 個のサンプル(Extract)から先述のアルゴリズムに従い特徴を抽出し，残りのサンプル(Verify)319 個+Normal サンプル 110 個を混ぜ合わせた 429 個の未知データグループを作成し，これらのデータグループに対して識別率を導出したものである。Grip は，ファミリー内の相関が非常に高く，識別率が 100%であった。一方，Mirai と Gafgyt は，識別率が 62.7%と 74.6%であった。総合して，319 個の Malware から 256 個の Malware を正しく識別でき，識別率は，79.6%であった。識別率の低い Mirai と Gafgyt に

表 2 Malware 識別性能

Table 2. Malware Detection Performance.

File		Extract	Verify	Detectable	Detection Rate
Malware Family Name	Gafgyt	71/142	71/142	53/71	74.6%
	Grip	21/41	20/41	20/20	100.0%
	Kaiten	55/109	54/109	44/54	81.5%
	Mrblack	50/99	49/99	43/49	87.8%
	Mirai	67/134	67/134	42/67	62.7%
	Ganiw	58/116	58/116	52/58	89.7%
Not Malware	Normal	-	110	-	-
Sum			429	254/319	79.6%

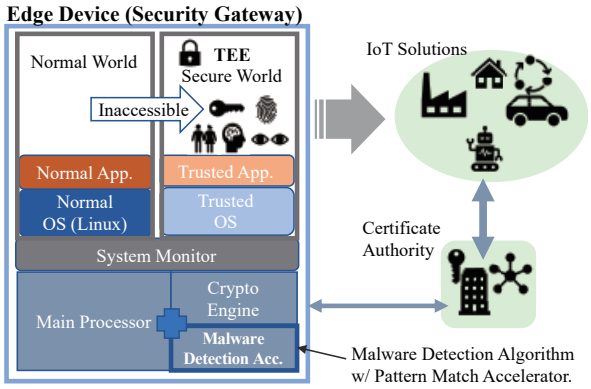


図 10 Pattern Match Accelerator を用いた Malware 検出器を TEE Edge デバイスへインテグレートしたコンセプト図

Fig. 10. Cyber-Security for Edge Computing.

関して，要因解析を行った結果，2 つの Malware 間の特徴類似度が高いことが原因であり，Mirai が Gafgyt と識別される誤検知が発生していた。さらに，Gafgyt に関しては，Normal と識別される誤検知が発生していた。

Fig.10 は，自動車などの Security Gateway に適用が進む TEE(Trusted Execution Environment)という暗号鍵生成・管理・認証機能を備えた Edge デバイスである。通常の動作領域(Normal World）からセキュリティ保護領域(Secure World)へのアクセスが不可能な構成となっており，クリティカルデータの格納や処理を提供できる。このような Cyber-Security 性に優れたシステムにおいても，システムの立ち上がりシーケンスや通常稼働中の攻撃検知など，定常的にセキュリティチェックを実行する機能実装が求められている。特に NHTSA(米国運輸省道路交通安全局)からは，自動車の Cyber-Security 保護要求 (Fundamental Vehicle Cybersecurity Protection)という指令が発効されており，クリティカルなファームウェアの不正更新に対する Cyber-Security 対策(監視と記録)を強く求めている。

これらの必要性は，メンテナンスアップデートなどのライフサイクルに渡り，ファームウェアなどへの Malware 混入を完全に担保することが難しいからである。さらに，クルマの Cyber-Security 要件として，これら TEE 機構のセキュアブート時間は，500msec 以内に完結することが求められており，ブートシーケンスに組み入れられる Malware の検知処理もリアルタイムに近い性能が必要である。

本研究の Malware 検出システムは，Security Gateway に求められる処理スピード，サイズ，消費電力を備えることから，Fig.10 の TEE 機構を支えるメインプロセッサに連結するコプロセッサとして活用できると考える。

4. まとめ

本研究で提案したアルゴリズムを用いプロトタイプによる実証実験を実施し，以下の 3 点の効果を確認した。

（1） HLAC 理論を適用したマスクパターンと Pattern Match Accelerator を用いた構造的解析手法のアルゴリズムにより画像化 Malware(Texture Image)から Malware 固有のパワースペクトル/時系列データの抽出ができることを確認した。

（2） Malware 固有のパワースペクトル/時系列データをモード信頼性評価基準(MAC: Modal Assurance Criterion)を用いて識別処理を行った結果，高い精度で Malware ファイルを検出可能であると確認できた。

（3） Pattern Match Accelerator の既往研究結果[6][7]から導出した計算コスト概算によると，本研究のアルゴリズムを適用する Pattern Match Accelerator は，一つの Malware ファイルの識別処理を，1.1msec以下(～5μsec×221 マスクパターン)で実行する。汎用プロセッサ単体を用いたパターンマッチングに比較し，高速処理可能であると考える。さらに，処理時間の短縮により，電力使用量の低減が可能である。

以上，3 点の効果により，高い Cyber-Security 能力を持った低計算コスト(処理時間，消費電力，簡素なプログラム，システムサイズ)の自動車向け Malware 検出システム，例えば Security Gateway(Fig.1)の実現が可能となる。

文　献

（1） Y. Taniwaki : “Cybersecurity in the Age of Digital Economy – Toward Establishing a Foundation for Digital Transformation -: 4. Spreading IoT Devices and Cybersecurity Policy”, *IPSJ Magazine,* Vol.59, No.12, pp.1090-1094 (2018-11) (in Japanese)
谷脇康彦:「ディジタルエコノミー時代のサイバーセキュリティ –ディジタルトランスフォーメーション促進の基盤確立に向けて-：4. IoT 機器の普及とサイバーセキュリティ政策」，情報処理学会誌, Vol.59, No.12, pp.1090-1094 (2018-11)

（2） M. Iwamura, M. Itoh, and Y. Muraoka : “Automatic Malware Classification System Based on Similarity of Machine Code Instructions” , *IPSJ Journal*, Vol.51, No.9, pp.1622-1632 (2010-9) (in Japanese)
岩村誠 ・ 伊藤光恭 ・ 岡村洋一：「機械語命令列の類似性に基づく自動マルウェア分類システム」，情報処理学会論文誌, Vol.51, No.9, pp.1622-1632 (2010-9)

（3） L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang :”A Comparative Assessment of Malware Classification using Binary Texture Analysis and Dynamic Analysis”, *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence,* USA, (2011-10)

（4） T. Toyoda, and O. Hasegawa :“Feature Extraction for Texture Classification Using Mask Patterns”, *IPSJ SIG Technical Reports*, Vol.2004, No.91, pp.77-84 (2004-9) (in Japanese)
豊田崇弘・長谷川修 ：「テクスチャ識別のためのマスクパターンによる特徴抽出法」，情報処理学会研究報告コンピュータビジョンとイメージメディア, Vol.2004, No.91, pp.77-84 (2004-9)

（5） K. Inoue, and C.-K. Pham : “The Memorism Processor: Towards a Memory-Based Artificially Intelligence Complementing the von Neumann Architecture”, *SICE Journal of Control, Measurement, and System Integration*, Vol.10, No.6, pp.544-550 (2017-11)

（6） K. Inoue, D.-H. Le, M. Sowa, and C.-K. Pham : “Set Operating Processor(SOP) : Application for Image recognition”, *The Institute of Electronics, Information and Communication Engineers, IEICE Technical Report*, Vol.113, No.236, pp.35-40 (2013-10) (in Japanese)
井上克己・レ ドゥクフン・曽和将容・範公可：「集合演算プロセッサー(SOP) – 画像認識への応用－」，電子情報通信学会技術研究報告, Vol. 113, No. 236, pp.35-40 (2013-10)

（7） D.-H. Le, T.-B.-T. Cao, K. Inoue, and C.-K. Pham : “A CAM-Based Information Detection Hardware System for Fast Image Matching on FPGA”, *IEICE Trans Electron.,* Vol. E97-C, No.1, pp.65-76 (2014-1)