ブロックチェーンベースの協調型マルウェア検知システムの研究

藤　竜成 [a)]・岡崎　直宣 [b)]・山場　久昭 [c)]・油田　健太郎 [d)]

A Study on a Blockchain-based Collaborative Malware Detection System

Ryusei FUJI, Naonobu OKAZAKI, Hisaaki YAMABA, Kentaro ABURADA

Abstract

Since malware has a serious adverse effect not only on computers but also on modern society based on the Internet, the detection is important. To prevent malware infection, we usually install anti-virus software developed by anti-virus vendors on our computers. However, generally, we use only single anti-virus software, which is insufficient as a countermeasure for malware. In this study, we propose a collaborative malware detection system that shares information about malware detection results among several anti-virus vendors and performs malware detection using the shared information. Each anti-virus vendor collects its information on malware independently, so it is expected that each anti-virus vendor has a different field of expertise for malware detection. Therefore, if several anti-virus vendors with different fields of expertise share and use information about malware detection results, we believe that malware detection accuracy can be improved. In this study, we propose information sharing of malware detection results among several anti-virus vendors. The information shared by the proposed system is the malware detection results from each anti-virus vendor and is extremely important information that must not be falsified. Therefore, in this study, we adopted blockchain technology as the infrastructure of the information sharing. In the evaluation experiments, we mainly evaluated the malware detection accuracy of the proposed system through simulations. As a result of the evaluation experiment, we confirmed that the malware detection accuracy of the proposed system was significantly improved compared to the malware detection accuracy of a single malware detection system.

Keywords: blockchain technology, collaborative security, collaborative malware detection system, smart contract

1. はじめに

マルウェアはコンピュータだけでなく、インターネットが基盤となっている現代社会に甚大な悪影響を及ぼすことから、その検知は重要である。マルウェアは、その存在を検知されずに不正な動作を行うことにより、コンピュータ内部に保存されている情報の窃取や破壊を行う。マルウェア感染による被害の例として、ランサムウェアが挙げられる。2017 年には、WannaCry と呼ばれるランサムウェアが猛威を振るい、世界中のコンピュータがそのランサムウェアに感染した [1)]。現在も 100 万台を超えるコンピュータが WannaCry の感染の危険性がある [2)]。また、文献 [3)] によると、2017 年のランサムウェアによる金銭的被害額は 50 億ドルであると言われており、ランサムウェアによる被害は甚大である。さらに、近年では Internet of Things (IoT) 機器を標的としたマルウェアも確認されている。例えば、2016 年にはマルウェア「Mirai」による IoT 機器を悪用した、最大で 1.5Tbps の DDoS 攻撃が観測されている [4)]。以上のような、マルウェア感染の被害は今後も増加していくことが予測される。AV-TEST の Security Report 2017/2018 [5)] によると、近年の新種マルウェアの観測数は 1 年あたり 1 億以上、一秒あたりに換算すると約 4 個のマルウェアが観測されている。明らかに、マルウェアによる被害はインターネットを社会基盤とする現代社会に悪影響を及ぼしていることから、それらの検知は重要である。

マルウェア感染を防止するため、一般に、マルウェア検知システムではシグネチャ方式が広く採用されている。しかしながら、シグネチャ方式では新種マルウェアの検知が困難である。そこで、ヒューリスティック方式やビヘイビア方式のマルウェア検知システムが、シグネチャ方式のマルウェア検知システムに加えて利用されているが、これらのマルウェア検知システムは、一般に誤検知、すなわち False positive や False negative を引き起こす。我々は普段利用するコンピュータに、以上のようなマルウェア検知システムを内包するアンチウイルスソフトウェアを導入しているが、通常単一のアンチウイルスソフトウェアしか導入していない。しかしながら、新種マルウェアが急増している現代において、単一のアンチウイルスベンダでは急増するマルウェアへの対策に限界がある。したがって、単一のアンチウイルスソフトウェアを導入するのみでは、マルウェア対策として不十分であり、マルウェ

[a)] 工学専攻機械・情報系コース大学院生
[b)] 情報システム工学科教授
[c)] 情報システム工学科助教
[d)] 情報システム工学科准教授

ア検知精度を向上させるための仕組みが望まれる。

そこで本研究では、複数のアンチウイルスベンダの間でマルウェア検知結果に関する情報を共有し、それらの情報を利用したマルウェア検知を行う、協調型マルウェア検知システムを提案する。アンチウイルスベンダはそれぞれ、独自にマルウェアに関する情報の収集を行っている。そのため、アンチウイルスベンダそれぞれでマルウェア検知に関する「得意領域」が異なると考えられる。したがって、「得意領域」が異なる複数のアンチウイルスベンダ間でのマルウェア検知結果に関する情報を共有し活用すれば、マルウェア検知精度の向上が期待できると考えている。

本研究では、複数のアンチウイルスベンダ間でマルウェア検知結果に関する情報共有を行うために、ブロックチェーン技術を採用する。一般に、複数アンチウイルスベンダ間の情報共有は、分散型データベースを含む一般的なデータベースを利用しても実現可能である。しかしながら、これらのデータベースは、一般に単一の組織によって管理されるため、攻撃者によるデータベースの改ざんに対して脆弱である。本提案システムで共有する情報は、各アンチウイルスベンダのマルウェア検知結果である。これらの情報は、ファイルが悪性か否かを判断するために利用されるため、改ざんされてはならない極めて重要な情報である。さらに、本研究における提案システムの場合、どのアンチウイルスベンダがそのデータベースの管理を主体となって行うのかといった問題が考えられる。仮に、ある単一のアンチウイルスベンダがデータベースの管理を行うことになった場合、そのアンチウイルスベンダに対して多大な負担が掛かる。そこで、本研究では、複数のアンチウイルスベンダ間でマルウェア検知結果に関する情報共有のために、ブロックチェーン技術を採用した。

評価実験では、主に提案システムのマルウェア検知精度の評価をシミュレーションを通して行った。また、マルウェア検知結果の検索性能とブロックチェーンサイズに関して性能評価を行い、提案システムの有用性について議論する。

2. 関連技術

本節では、本研究において関連する技術について概説する。最初にマルウェア検知システムに関して説明し、その後ブロックチェーン技術について述べる。

2.1 マルウェア検知システム

最初に、マルウェア検知手法について説明する。その後、従来のマルウェア検知システムにおける課題点について述べる。

2.1.1 マルウェア検知手法

マルウェア検知のため、一般に以下の 3 種類の方式のマルウェア検知手法が利用されている。

1. シグネチャ方式
2. ヒューリスティック方式
3. ビヘイビア方式

それぞれの方式について説明する。

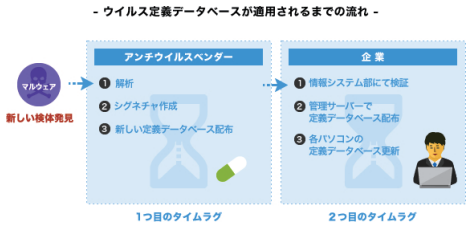

図 1. シグネチャ更新までのタイムラグ ([9] より引用)

2.1.1.1 シグネチャ方式

シグネチャ方式のマルウェア検知は、連続するバイトシーケンスを検査対象から抽出し（これをシグネチャと呼ぶ)、それが既知のマルウェアのシグネチャと一致する場合はマルウェアと判断される。シグネチャ方式は、マルウェア検知において一般的に利用されている方式である [6]。シグネチャの例としては、オープンソースのアンチウイルスソフトである Clam AntiVirus[7] では、16 進数文字列や MD5 シグネチャ、16 進数文字列に関する正規表現などが利用されている [8]。

シグネチャ方式の利点として、シグネチャが登録されているマルウェア、すなわち既知マルウェアを確実に検知可能である点が挙げられる。一方、シグネチャが登録されていないマルウェア、すなわち未知マルウェアについては検知が困難である点が欠点として挙げられる。

また、シグネチャ方式のもう一つの欠点として、新種マルウェアが発生してからそのシグネチャが作成・配布されるまでに、時間がかかってしまうことが挙げられる。シグネチャは、シグネチャ方式のマルウェア検知システムの提供者によって作成される。一般に、シグネチャ方式のマルウェア検知システムの提供者はアンチウイルスベンダである。したがって、アンチウイルスベンダは、日々出現する新種マルウェアを検知するためにシグネチャを生成、配布し続けなければならないが、それらには必然的にタイムラグが存在する。図 1 は、シグネチャが一般ユーザのコンピュータに適用されるまでに発生するタイムラグを示している。図 1 では、2 つのタイムラグが存在するとされているが、そもそもシグネチャを生成するためには、マルウェア検体そのものを入手しなければならない。すなわち、シグネチャが一般ユーザのコンピュータに適用されるまでに発生するタイムラグは、以下の 3 つのタイムラグの和である。

1. アンチウイルスベンダが新種マルウェアの検体を発見・入手するまでに要する時間
2. アンチウイルスベンダが検体を解析し、シグネチャを生成するまでに要する時間
3. 企業（一般ユーザ）がシグネチャを入手し、コンピュータに適用するまでに要する時間

新種マルウェアに対応するために、シグネチャが一般ユーザのコンピュータに適用されるまでの期間は、一般ユーザにとって新種マルウェアに感染するリスクがある。これら新種マルウェアに対応するため、アンチウイルスベンダはヒューリスティック方式やビヘイビア方式に相当するマルウェア検知システムを一般ユーザへ提供している。

2.1.1.2 ヒューリスティック方式

ヒューリスティック方式のマルウェア検知は、OS に対して

発行される API コール列や機械語のオペコードなどの特徴を利用して、マルウェア検知を行う。この検知手法では、データマイニング技術や機械学習技術が利用される [10]。例えば、商用のアンチウイルスソフトウェアでは、ヒューリスティックエンジン [11] などが存在する。

ヒューリスティック方式の利点としては、シグネチャ方式では検知が困難であるマルウェア、すなわち未知マルウェアが検知可能である点が挙げられる。例えば、亜種マルウェアは元となるマルウェアが存在し、そのマルウェアをベースに作成される。したがって、API コール列や機械語のオペコードといった特徴は亜種マルウェアと元となったマルウェアで類似するため、それらの類似性をベースに亜種マルウェアが検知可能である。

一方、ヒューリスティック方式では、マルウェア検知のためにデータマイニング技術や機械学習技術が利用される。そのため、良性ファイルを誤ってマルウェアとして検知してしまう False positive や、マルウェアを誤って良性ファイルとして判定してしまう False negative が発生してしまう点が欠点として挙げられる。

2.1.1.3 ビヘイビア方式

ビヘイビア方式のマルウェア検知は、検査対象となるファイルを実際に実行し、その挙動を分析することによってマルウェアを検知する方式である [6]。ビヘイビア方式もヒューリスティック方式と同様に、シグネチャ方式の補完を目的としてアンチウイルスソフトウェアに導入されている。例えば、商用のアンチウイルスソフトウェアでは、System Watcher[12] などが存在する。ビヘイビア方式の利点や欠点はヒューリスティック方式と同様である。

2.1.2 従来のマルウェア検知システムにおける課題点

一般ユーザは、従来のマルウェア検知システムとして、アンチウイルスベンダが開発するアンチウイルスソフトウェアを利用してきた。アンチウイルスベンダは、2.1.1 で説明した 3 種類の検知手法を利用することで、マルウェア検知を行っているが、単一のアンチウイルスベンダでは急増するマルウェアへの対策に限界がある。

文献 [13] では、アンチウイルスソフトウェアでは検知できないマルウェア（未検知検体）に関する情報をそのベンダに提供し、その後のマルウェアの検知率を算出することにより、アンチウイルスソフトウェアの評価を行っている。図 2 に、未検知検体全体の情報提供後におけるマルウェア検知率を示す。この図から分かるように、アンチウイルスベンダへマルウェアの情報提供後（30 日）の当該マルウェアの検知率は、最大でも 40% であった。

文献 [14] では、商用のアンチウイルスソフトウェアのマルウェア検知率を評価している。評価用のデータセットとして、Arbor Network's Arbor Malware Libraly の 7220 個のマルウェア検体が利用されている。その検体のキャプチャ日時に応じて、検体を 3 ヶ月以内に取得したマルウェアグループ、1 ヶ月以内に取得したマルウェアグループ、1 週間以内に取得したマルウェアグループに属させ評価を行っている。図 3 は、アンチウイルスソフトフェアのマルウェア検知率を示す。この

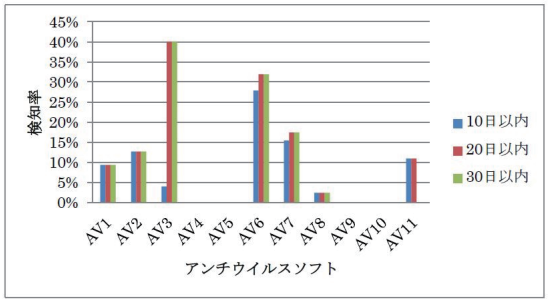

図 4 未検知検体全体の情報提供後の検知率

図 2. 未検知検体全体の情報提供後におけるマルウェア検知率（文献 [13] より引用）

AV Vendor	Version	3 Months	1 Month	1 Week
Avast	4.7.1043	62.7%	45.8%	39.6%
AVG	7.5.503	83.8%	78.6%	72.2%
BitDefender	7.1.2559	83.9%	79.7%	78.5%
ClamAV	0.91.2	57.5%	48.8%	46.8%
CWSandbox	2.0	N/A	N/A	N/A
F-Prot	6.0.8.0	70.4%	49.6%	46.0%
F-Secure	8.00.101	80.9%	74.4%	60.3%
Kaspersky	7.0.0.125	89.2%	84.0%	78.5%
McAfee	8.5.0i	70.5%	56.7%	53.9%
Norman	1.8	N/A	N/A	N/A
Symantec	15.0.0.58	60.8%	38.8%	45.2%
Trend Micro	16.00	79.4%	74.6%	75.3%

図 3. アンチウイルスソフトフェアのマルウェア検知率（文献 [14] より引用）

図から、各アンチウイルスソフトウェアで差異があるものの、検知漏れが発生していることが分かる。

以上から、新種マルウェアが急増している現代において、単一のアンチウイルスベンダでは急増するマルウェアへの対策に限界があり、アンチウイルスソフトウェアを利用する一般ユーザは、依然としてマルウェアの感染リスクがあると考えられる。そのため、マルウェア検知精度を向上させるための仕組みが望まれる。

2.2 ブロックチェーン技術

次に、ブロックチェーン技術について概説する。ブロックチェーン技術は、ビットコインをはじめとする様々な仮想通貨の基礎となる技術であり、近年注目を集めている。また、ロジスティクス分野や医療分野をはじめとする様々な産業分野での応用が検討されており [15]、その応用範囲は今後も拡大していくと予想される。本節では、ブロックチェーン技術について説明する。最初に、ブロックチェーン技術の概要について説明する。その後、ブロックチェーン技術の特徴について述べたあと、ブロックチェーンプラットフォームについて紹介する。本研究では、ブロックチェーン技術を情報共有の基盤として利用しているが、ブロックチェーン技術を採用する動機については 3.3.4 で述べる。

2.2.1 ブロックチェーン技術の概要

ブロックチェーン技術は、2008 年にサトシ・ナカモトによって発表された論文 [16] において、ビットコインと呼ばれる暗号通貨（暗号資産）を実現するために提案された技術であり、分散型台帳技術とも呼ばれている。サトシ・ナカモトは、従来、銀行が担っていた通貨発行や取引の仲介などの機能を、ブロックチェーン技術をはじめとする複数の技術を組み合わせることにより分散化した。機能の分散化により、ビットコインは、銀行などの中央となる第三者機関を介すことなく、通

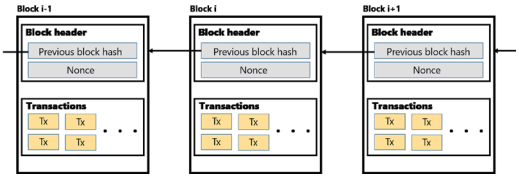

図 4. ブロックチェーンの一例

貨の発行やユーザ間での直接取引が可能となった。

ユーザ間で送金を行う際、送金側のユーザは送金量や宛先のアドレス、自分自身のディジタル署名などを含む、価値の移転を記したトランザクションを発行する。発行されたトランザクションは、相互に接続されているノード間で送受信される。トランザクションを受け取ったノードは、トランザクションの検証を行い、そのトランザクションが有効であれば、次のノードに送信する。以上のトランザクションの送受信により、発行されたトランザクションはブロックチェーンネットワーク全体に伝搬される。最終的には、マイナーによってブロックに取り込まれ、ブロックチェーンの一部になることにより送金処理が完了する。ブロックチェーン上のデータの完全性の維持や合意形成のために、Proof of Work (PoW) や Proof of Stake (PoS)、Delegated Proof of Stake (DPoS) などの合意形成アルゴリズムが利用される。例えば、ビットコインでは、合意形成アルゴリズムとして、PoW が利用されている。PoW では、ブロックのハッシュ値がある特定のハッシュ値を満たすようにある値 Nonce を定め、ブロックを生成する合意形成アルゴリズムである。PoW によって生成されたブロックは、ノード間のブロックの送受信によりビットコインネットワーク全体に伝搬され、独立にその有効性が検証される。検証の結果、そのブロックが有効であれば、ビットコインネットワークに受け入れられ、ブロックチェーンの一部となる。ブロックチェーンの一例を図 4 に示す。

2.2.2 ブロックチェーン技術の特徴

ブロックチェーン技術の主な特徴として、以下の 4 つが挙げられる [17]。

Decentralized
: ブロックチェーン技術では、データ共有やトランザクションの正当性検証のために、中央となる第三者機関が存在しない。例えば、ビットコインでは、従来銀行が担っていた通貨の発行やトランザクションの仲介、二重取引の防止などの機能を複数の技術を組み合わせることにより分散化に成功した。

Persistency
: ブロックチェーン上のデータの完全性の維持やネットワーク全体での合意形成のために、PoW や PoS などの合意形成アルゴリズムが利用される。これらの合意形成アルゴリズムにより、ブロックチェーンに含まれるトランザクションを改ざんすることは困難である。例えば、ビットコインでは合意形成アルゴリズムとして、PoW が利用されており、ブロックを生成するためには膨大な計算資源が必要となる。例えば、あるブロックに含まれるトランザクションを改ざんしようとした場合、ブロック間の整合性を取るために、そのブロックよりも過去に生成されたブロックを再度生成しなければならない。ブロックを生成するためには膨大な計算資源が必要となるため、ブロックチェーンの改ざんは困難である。

Anonymity
: ブロックチェーンネットワークにおいて、あるノードは一つ以上のアドレスで表現される。例えば、ビットコインネットワークでは、「1NYXCdriKyqLjgSffmmC6xysxCMg3LVvpW」のような形式のアドレスを複数所有することが可能である。これらのアドレスは一般に、個人に結びつかない。そのため、ブロックチェーンネットワークでは、匿名でトランザクションを発行することが可能である。

Auditability
: ブロックチェーンネットワークでは、不正なブロックやトランザクションを排除するため、ブロックチェーンネットワーク参加者である各ノードが検証する必要がある。そのためブロックやトランザクションは「検証可能」である。

Table 1 Comparisons among *public blockchain*, *consortium blockchain* and *private blockchain*

Property	*Public blockchain*	*Consortium blockchain*	*Private blockchain*
Consensus determination	All miners	Selected set of nodes	One organisation
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralised	No	Partial	Yes
Consensus process	Permissionless	Permissioned	Permissioned

図 5. ブロックチェーンの形態と特徴（文献 [18] より引用）

2.2.3 ブロックチェーンの形態

ビットコインでは、任意のユーザがブロックチェーンネットワークに参加可能である。すなわち、任意のユーザがマイニングやトランザクションの発行、検証などを行うことが可能である。このようなブロックチェーンをパブリック型のブロックチェーンと呼ぶが、後にパブリック型のブロックチェーンとは異なる形態のブロックチェーンが考案されている。ブロックチェーンの形態は、その管理主体によってパブリック型、コンソーシアム型、プライベート型の 3 種類の形態が存在する。図 5 に、それぞれのブロックチェーン形態の特徴を示す。

図 5 から、それぞれの形態によってブロックチェーンの特徴の「度合い」が異なることがわかる。例えば、2.2.2 で説明した、Persistency に関しては、パブリック型のブロックチェーンにおいては改ざんはほぼ不可能とされているが、コンソーシアム型やプライベート型のブロックチェーンでは、改ざんの可能性があるとされている。したがって、ブロックチェーンを活用するシステムの特徴や要件に沿ってブロックチェーンの形態を決定し、ブロックチェーンプラットフォームを選定しなければならない。

2.2.4 ブロックチェーンプラットフォーム

ブロックチェーンプラットフォームにはビットコインをはじめとして、Ethereum[19] や Hyperledger Fabric[20] などの様々なプラットフォームが存在する。Ethereum はスマート

コントラクト利用する、Decentralized applications (Dapps) を構築するためのブロックチェーンプラットフォームであり、オープンソースで開発が進められている。スマートコントラクトとは、プログラム化された契約であり、ブロックチェーン上でプログラムとして実行されることにより、ブロックチェーンを利用した様々なアプリケーションが開発可能となっている。Ethereum のスマートコントラクトの活用事例としては、デジタル ID サービスである uPort[21] が挙げられる。Ethereum は一般にパブリック型のブロックチェーンに分類される。

一方、Linux Foundation によってオープンソースで開発が進められている、Hyperledger Fabric はコンソーシアム型のブロックチェーンに分類される。Hyperledger Fabric はコンソーシアムブロックチェーンを構築されるためのプラットフォームとして広く活用されている。3. で詳細は説明するが、本研究では協調型マルウェア検知システムにおける情報共有の基盤としてブロックチェーン技術を採用する。本提案システムのブロックチェーンネットワークの参加者は、特定複数のアンチウイルスベンダである。したがって、本研究ではコンソーシアム型のブロックチェーンプラットフォームである Hyperledger Fabric を採用する。

3. 提案システム

2.1 で説明したように、一般にマルウェア検知のためにはシグネチャ方式のマルウェア検知システムが利用される。しかしながら、シグネチャ方式のマルウェア検知システムのみでは、新種マルウェアや亜種マルウェアの検知は困難である。日々、新種マルウェアや亜種マルウェアが出現しており、シグネチャの作成が追い付かないためである。そのため、これらのマルウェア検知のため、ヒューリスティック方式やビヘイビア方式のマルウェア検知システムが利用される。これらのマルウェア検知システムは新種マルウェアや亜種マルウェアを検知可能な一方、誤検知を引き起こす。すなわち、False positive や False negative を引き起こす。したがって、ユーザは依然としてマルウェア感染の危険にさらされており、マルウェア検知精度を向上させるための仕組みが望まれる。

そこで、本研究では、マルウェア検知精度向上のため、協調型マルウェア検知システムを提案する。提案システムの処理は、ユーザがインターネット上からファイルをダウンロードした際に開始される。シグネチャ方式のマルウェア検知システムで、そのファイルがマルウェアと判定されなかった場合、ユーザは自身のヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムによってそのファイルがマルウェアか否かの判定を試みるとともに、アンチウイルスベンダへそのファイルの調査を依頼する。依頼されたファイルが、アンチウイルスベンダ間で以前に判定が行われていた場合、その検知結果をユーザへ返却する。ユーザは、自分自身のマルウェア検知システムの検知結果とアンチウイルスベンダの検知結果を利用して、「悪性度」と呼ばれる値を算出し、その値に基づいてファイルがマルウェアか否かを判定する。依頼されたファイルが、アンチウイルスベンダ間で以前に判定が行われていなかった場合、かつユーザ自身がマルウェアでないと判定した場合は、そのファイルはマルウェアでないと判定し処理を終了する。ユーザ自身がマルウェアであると判定した場合は、当該ファイルをアンチウイルスベンダへ送信し、そのファイルの調査を依頼する。依頼されたアンチウイルスベンダは、その他のアンチウイルスベンダへ当該ファイルの検査を依頼し、検知結果をアンチウイルスベンダ間で共有する。同じファイルについて問い合わせがあった場合は、その検知結果がレスポンスとしてユーザへ返却される。共有された検知結果はユーザへ送信され、ユーザは自身のマルウェア検知結果とアンチウイルスベンダのマルウェア検知結果を利用して「悪性度」を算出し、マルウェアか否かを判定する。

3.1 キーアイデア

提案システムのキーとなるアイデアは、複数のアンチウイルスベンダのマルウェア検知システムの検知結果を共有・集約し、それらの検知結果をユーザ側でのマルウェア検知に活用するというものである。この時、それぞれのアンチウイルスベンダがマルウェア検知に関する異なる「得意領域」を持つと期待されることが、ユーザ側でのマルウェアの検知精度向上につながると考える。

アンチウイルスベンダは、マルウェア検知のため、様々な手法やサービスを利用してマルウェア情報の収集を行っている。図 6 は、その全体像を示している。アンチウイルスベンダは、オンラインマルウェア検査・解析サービスやハニーポットなどを活用し、マルウェア情報の収集を行っている。収集されたマルウェア情報は、シグネチャの作成やヒューリスティック方式やビヘイビア方式のマルウェア検知システムの作成に役立てられる。

これらのアンチウイルスベンダのマルウェア情報収集活動は、それぞれのアンチウイルスベンダが独自に行っている。すなわち、アンチウイルスベンダが入手し得るマルウェア情報は、アンチウイルスベンダ間で異なる。そのため、あるアンチウイルスベンダが販売するアンチウイルスソフトウェアでは検知できなかったマルウェアが、別のアンチウイルスベンダが販売するアンチウイルスソフトウェアでは検知できる可能性がある。言い換えれば、アンチウイルスベンダそれぞれでマルウェア検知に関する「得意領域」が異なると考えられる。

したがって、「得意領域」が異なるアンチウイルスベンダのマルウェア検知システムの検知結果を集約し、それらの検知結果をユーザ側でのマルウェア検知に活用できれば、「得意領域」の相互補完が可能となり、ユーザ側におけるマルウェア検知精度の向上が期待できる。

3.2 提案システムで必要とされる事項

提案システムの実現にあたり重要な事項として、(1) 各アンチウイルスベンダによるマルウェア検知結果の表現と検知結果の集約、(2) マルウェア検知結果の共有、 (3) マルウェアの疑いのあるファイルの収集、(4) ユーザにおける最終的なマルウェア判定がある。以上の 4 点をどのように実現するかについて説明する。

一般に、マルウェア検知システムの出力は検査対象が「悪性」または「良性」を示す情報である。そのため、各アンチウイルスベンダは検査対象を「悪性」または「良性」としてマルウェア検知結果を表現する。また、提案システムでは、「得意領域」が異なるアンチウイルスベンダのマルウェア検知シス

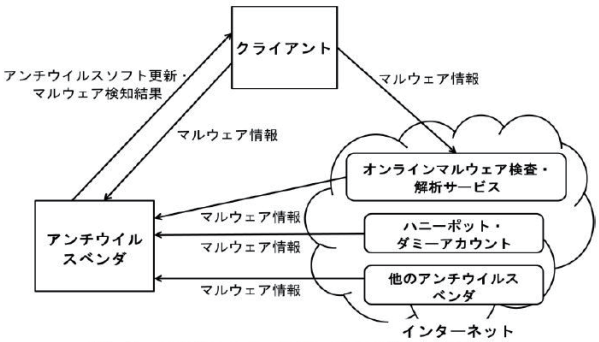

図 2 ベンダによるマルウェア情報収集の全体図

図 6. アンチウイルスベンダによるマルウェア情報収集（文献[13]より引用）

テムの検知結果を共有・集約し、それらの検知結果をユーザ側でのマルウェア検知に活用できれば、ユーザ側におけるマルウェア検知精度の向上が期待できる。したがって、複数のアンチウイルスベンダのマルウェア検知結果の集約には「悪性：XX 票、良性：YY 票」のような「投票」形式が有効であると考える。

アンチウイルスベンダ間のマルウェア検知結果の共有にあたり、我々はその仕組みついて検討しなければならない。本研究では、共有手法としてブロックチェーン技術を採用した。その動機は、3.3.4 で説明する。

また、アンチウイルスベンダがマルウェア検知結果を共有するためには、その検査対象となるファイルそのものを取得する必要がある。しかし、各アンチウイルスベンダがマルウェアの疑いのあるファイルを見つけ出し、検査を実施するのは非効率であると考える。そこで、各ユーザがマルウェアとして検知したファイル（マルウェアの疑いのあるファイル）をアンチウイルスベンダに送信し、各アンチウイルスベンダがそのファイルを検査することにより、マルウェア検知結果を共有する仕組みであれば、より効果的にマルウェアの疑いのあるファイルに対するマルウェア検知結果を得ることが可能であると考える。

ユーザは、最終的なマルウェア判定を下す際に、自分自身のマルウェア検知システムの判定結果とアンチウイルスベンダのマルウェア検知システムの判定結果が利用できる。したがって、それら 2 つの判定結果を総合して、最終的なマルウェア判定を下す仕組みが必要である。本提案システムでは、ユーザは「悪性度」と呼ばれる値を算出し、その値に基づいてファイルがマルウェアか否かを判定する。悪性度については、3.7 で説明する。

3.3 提案システムのモデル

提案システムのモデルを図 7 に示す。提案システムは、アンチウイルスベンダ同士がブロックチェーンネットワーク、ファイル共有ネットワークを構成している。そのため、本研究では、ブロックチェーンの形態として、コンソーシアム型のブロックチェーンを採用する。ユーザから送信されるファイルは、ヒューリスティック方式やビヘイビア方式のマルウェア検知システムで検査可能であるとする。また、アンチウイルスベンダの顧客であるユーザとアンチウイルスベンダのノードはそれぞれ、User components と Anti-virus vendor components を

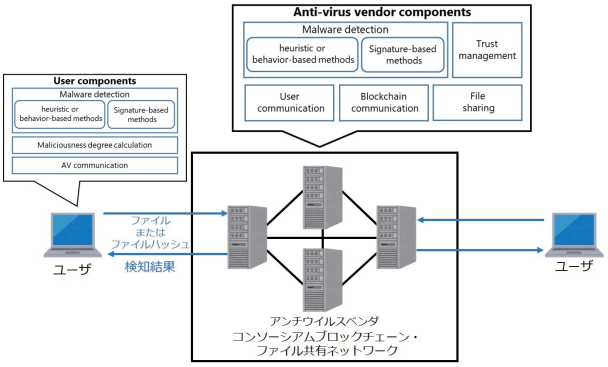

図 7. 提案システムのモデル

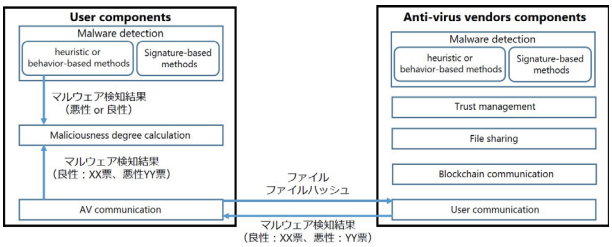

図 8. User components

所有しているとする。以下、アンチウイルスベンダの顧客を単にユーザと表記する。User components と Anti-virus vendor components については、3.3.1 と 3.3.2 でそれぞれ説明する。

3.3.1 User components

User components は、ユーザのコンピュータが所有するコンポーネントである。ユーザはこれらのコンポーネントを通じて、マルウェアの検知やアンチウイルスベンダへのファイルの送信、悪性度の算出などを実施する。図 8 に、User components の詳細を示す。User component を構成するコンポーネントは、以下の 3 つである。

- Malware detection
- Maliciousness degree calculation
- AV communication

Malware detection コンポーネントは、主にマルウェア検知の役割を担うコンポーネントである。Malware detection コンポーネントは、シグネチャ方式のマルウェア検知システムとヒューリスティック方式やビヘイビア方式のマルウェア検知システムから構成される。Malware detection コンポーネントは、アンチウイルスベンダによって提供される、アンチウイルスソフトウェアに相当する。

Maliciousness degree calculation コンポーネントは、悪性度の計算を行い、算出された悪性度に基づき、ファイルがマルウェアか否かを判断する役割を担う。Maliciousness degree calculation コンポーネントは、ユーザの Malware detection コンポーネントからヒューリスティック方式やビヘイビア方式のマルウェア検知システムの検知結果を、AV communication コンポーネントから、アンチウイルスベンダの「投票」結果を受信し、悪性度の計算を行う。

AV communication コンポーネントは、アンチウイルスベンダのノードとの通信を主に行う。ユーザがダウンロードしたファイルや、ファイルハッシュをアンチウイルスベンダの

ノードへ送信したり、アンチウイルスベンダのノードからブロックチェーン上の「投票」結果を受信したりする役割を担う。

3.3.2 Anti-virus vendor components

Anti-virus vendor components は、各アンチウイルスベンダのノードが所有するコンポーネントである。アンチウイルスベンダのノードはこれらのコンポーネントを通じて、マルウェアの検知、ブロックチェーンネットワーク上へのマルウェア検知結果の送出などを実施する. 図 9 に、Anti-virus vendor components の詳細を示す。Anti-virus vendor components を構成するコンポーネントは、以下の 5 つである。

- Malware detection
- Blockchain communication
- File sharing
- Trust management
- User communication

Malware detection コンポーネントは、主にマルウェア検知の役割を担うコンポーネントである。Malware detection コンポーネントは、シグネチャ方式のマルウェア検知システムとヒューリスティック方式やビヘイビア方式のマルウェア検知システムから構成される。Malware detection コンポーネントは、各アンチウイルスベンダが開発するアンチウイルスソフトウェアである。

Blockchain communication コンポーネントは、アンチウイルスベンダのノード間のブロックチェーン技術を利用した通信を行う。ブロックチェーン上から各アンチウイルスベンダの「投票」結果を取得したり、ブロックチェーンネットワーク上に「投票」をトランザクションとして送出したりする役割を担う。

File sharing コンポーネントは、User communication コンポーネントからファイルを受け取り、他のアンチウイルスベンダのノードへファイルを共有する役割を担う。

Trust management コンポーネントは、アンチウイルスベンダのノードに関する「信頼値」を算出し、各アンチウイルスベンダのノードが信頼できるか否かを判定するコンポーネントである。複数のノードが協調して攻撃に関する情報を共有したり、攻撃を検知したりするネットワークである協調型ネットワーク（Collaborative network）では、一般に Insider attack と呼ばれる攻撃について考慮する必要がある。詳細は 3.9 で述べる。

User communication コンポーネントは、アンチウイルスベンダのノードとユーザとの通信を行う際に利用されるコンポーネントである。ユーザからファイルやファイルハッシュの受信を行ったり、ユーザへ各アンチウイルスベンダのマルウェア検知結果を送信する役割を担う。

3.3.3 以前の提案システムの比較

本提案システムは、アンチウイルスベンダ同士がブロックチェーンネットワーク、ファイル共有ネットワークを構成している。そのため、本研究では、ブロックチェーンの形態としてコンソーシアム型のブロックチェーンを採用している。本項では、以前の提案システム [22)23)24)]（以下、旧提案システムと表記する。）との差異について述べる。

旧提案システムの概略図を図 10 に示す。旧提案システムでは、マルウェア情報を共有したいと考えている一般ユーザによって構成されていた。しかしながら、旧提案システムでは以下のような課題が存在した。

- 十分なマルウェア検知結果（投票）を得るには時間を要する。
- ブロックチェーンの肥大化。

旧提案システムでは、各一般ユーザがファイルをダウンロードした際に、マルウェア検知を行いその検知結果を「投票」として、ブロックチェーンネットワーク上に送出していた。そのため、マルウェア検知精度を向上させるだけの十分な「投票」数を得るためには、時間を要した。つまり、ファイルハッシュがブロックチェーン上に登録された初期段階では、「投票」数が少ないため、マルウェア検知精度の向上には至らなかった。本研究における提案システムでは、一般ユーザがファイルをダウンロードした際は、アンチウイルスベンダへその検知を依頼し、その「投票」結果を受け取ることが可能である。それらの「投票」結果は、コンソーシアムに参加する全てのアンチウイルスベンダのマルウェア検知結果を表すため、ファイルハッシュがブロックチェーン上に登録された初期段階から、マルウェア検知精度の向上が期待できる。

また、旧提案システムでは、パブリック型のブロックチェーンを採用していた。これは、旧提案システムでは任意の一般ユーザのブロックチェーンネットワークへの参加を想定していたためである。しかしながら、ブロックチェーン技術の特徴により、ブロックチェーンのサイズは時間の経過に従って増加していく。そのため、ブロックチェーンを保存するために必要となるストレージ容量も増加していくため、一般ユーザにとっては負担である。本研究における提案システムでは、アンチウイルスベンダ同士がブロックチェーンネットワーク、ファイル共有ネットワークを構成しているため、コンソーシアム型のブロックチェーンを採用している。そのため、ブロックチェーンのサイズ増加に対応すべきノードは、アンチウイルスベンダのノートのみである。したがって、一般ユーザがブロックチェーンそのものを保持しておく必要はなく、ブロックチェーンを保存するためのストレージ容量に関する負担はない。

3.3.4 ブロックチェーン技術を採用する動機

複数アンチウイルスベンダ間の情報共有は、分散型データベースを含む一般的なデータベースを利用しても実現可能である。しかしながら、これらのデータベースは、一般に単一の組織によって管理されるため、攻撃者によるデータベースの改ざんに対して脆弱である。本提案システムで共有する情報は、各アンチウイルスベンダのマルウェア検知結果を表す「投票」である。これらの「投票」は、ファイルが悪性か否かを判断するために利用されるため、改ざんされてはならない極めて重要な情報である。さらに、本提案システムの場合、どのアンチウイルスベンダがそのデータベースの管理を主体となって行うのかといった問題が考えられる。仮に、ある単一のアンチウイルスベンダがデータベースの管理を行うことになった場合、そのアンチウイルスベンダに対して多大な負担が掛かる。

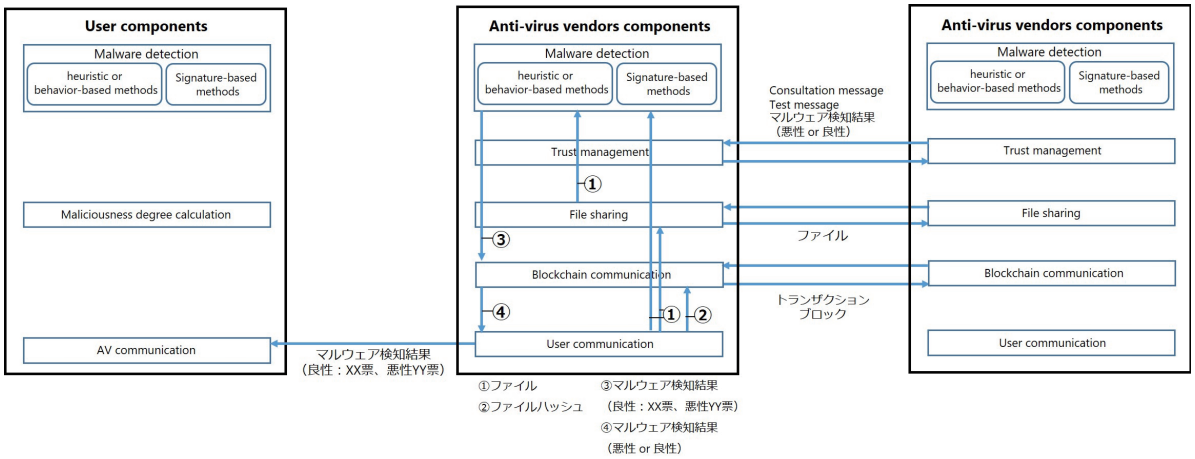

図 9. Anti-virus vendor components

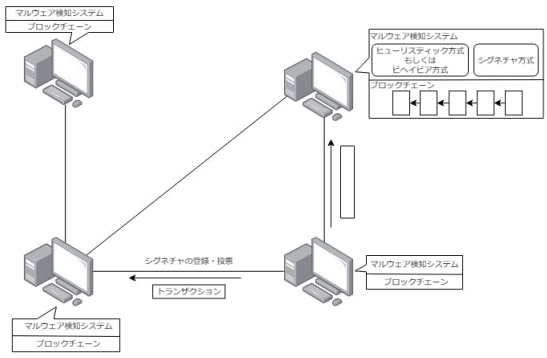

図 10. 旧提案システムの概略図

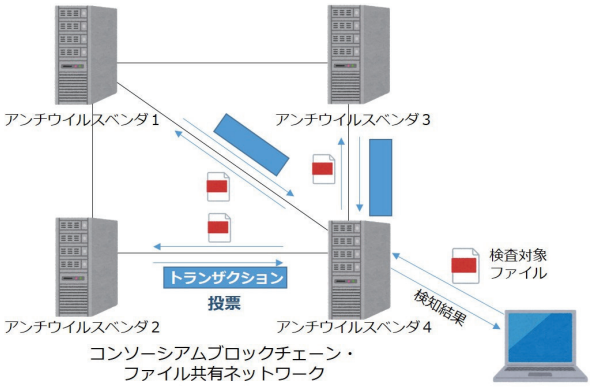

図 11. 提案システムの概略図

我々は上記の問題に対し、ブロックチェーン技術の採用により解決できると考えている。2.2 で説明したように、ブロックチェーン技術では、共有される情報の改ざんは困難である。これは、共有される情報を含むブロックが、ひとつ前に共有されたブロックのハッシュ値を指すように生成されるためである。さらに、ブロックチェーンネットワーク全体で管理されるブロックチェーンは基本的に 1 つであり、各ノードにそのコピーが保存される。ブロックチェーンに対して何らかのデータを追加・更新・削除する場合は、ブロックチェーンネットワーク参加者の合意が必要となる。そのため、ブロックチェーンネットワークの参加者である、全てのアンチウイルスベンダが平等な立場でデータベースを管理・利用することが可能となる。したがって、ある単一のアンチウイルスベンダにデータベースの管理コストが集中するような事態を回避することが可能となる。

3.4 提案システムの概要

提案システムの概略図を図 11 に示す。ファイル共有ネットワークとブロックチェーンネットワークは、アンチウイルスベンダによって構成される。ファイル共有ネットワークでは、ユーザから送信されたマルウェアの疑いのあるファイルが、ブロックチェーンネットワークでは、ファイルハッシュとそれに対応する「投票」結果が共有される。

提案システムの処理は、ユーザがインターネット上からファイルをダウンロードした際に開始される。ユーザの Malware detection コンポーネント構成要素である、ヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムによって、ファイルがマルウェアとして判定された場合、そのファイルはアンチウイルスベンダに送信され、それぞれのアンチウイルスベンダで検査が実施される。その後、マルウェア検知結果がブロックチェーンネットワークを通じて「投票」として共有される。

別のユーザが上記ユーザと同一のファイルをインターネット上からダウンロードした際は、そのファイルハッシュをキーとして、アンチウイルスベンダに問い合わせ、「投票」結果を受け取る。「投票」結果は、コンソーシアムに参加する全てのアンチウイルスベンダのマルウェア検知結果を表す。そのため、それらの「投票」を利用しマルウェア判定を実施することで、マルウェア検知精度の向上が期待できる。「投票」を利用したマルウェア判定は、悪性度を算出することによって行われる。悪性度の算出に関しては、3.7 で説明する。

3.5 ユーザの処理の流れ

ユーザの処理は、インターネット上からファイルがダウンロードされた際に開始される。ユーザの処理を図 12 に示す。ユーザの処理は以下の 4 つのステップに分けられる。ステップの流れと各ステップで利用される User components の対応を図 13 に示す。

1. 既知マルウェア検知ステップ
2. マルウェア検知結果検索ステップ
3. 未知マルウェア検知ステップ
4. マルウェア判定ステップ

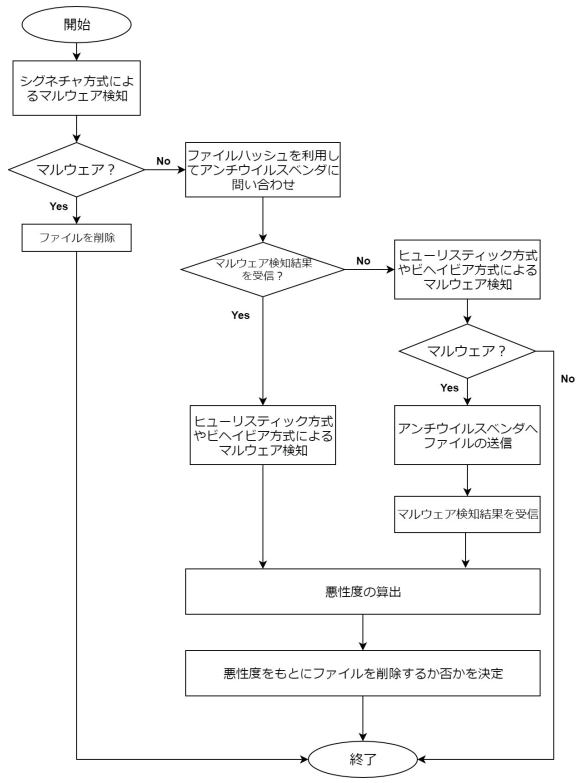

図 12. ユーザの処理

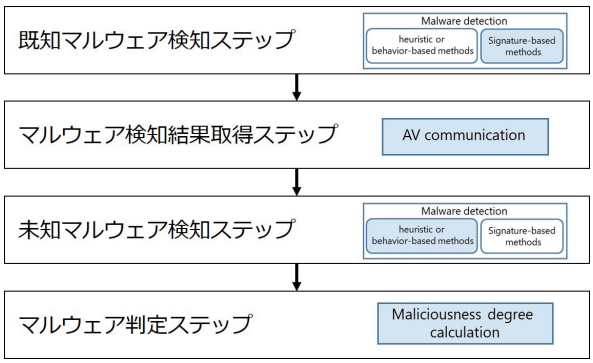

図 13. ステップの流れと各ステップで利用される User components

それぞれのステップについて説明する。

3.5.1 既知マルウェア検知ステップ

既知マルウェア検知ステップでは、Malware detection コンポーネントのシグネチャ方式のマルウェア検知システムを利用して、既知のマルウェア検知が実施される。本研究では、既知マルウェアとは、Malware detection コンポーネントのシグネチャ方式によって検知可能なマルウェアと定義する。すなわち、シグネチャ方式のマルウェア検知システムにおいて、シグネチャが既に登録されているマルウェアを既知マルウェアと呼ぶことにする。

既知マルウェア検知ステップにおいて、ファイルがマルウェアとして判定された場合、ファイルは削除されユーザにおけるすべての処理が終了する。ファイルがマルウェアとして判定されなかった場合、当該ファイルは未知マルウェアまたは良性ファイルのいずれかである。そのため、ファイルハッシュ値を算出し、マルウェア検知結果検索ステップにおいて当該ファイルハッシュに対応するマルウェア検知結果が存在するか否かを確認する。本研究における、未知マルウェアとは、Malware detection コンポーネントのシグネチャ方式によって検知されないマルウェアと定義する。すなわち、シグネチャ方式のマルウェア検知システムにおいて、シグネチャが登録されていないマルウェアを未知マルウェアと呼ぶことにする。

3.5.2 マルウェア検知結果検索ステップ

マルウェア検知結果検索ステップでは、AV communication コンポーネントを利用して、アンチウイルスベンダとの通信が行われる。ファイルハッシュをキーとして、アンチウイルスベンダにマルウェア検知結果の問い合わせを行い、当該ファイルハッシュに対応するマルウェア検知結果が存在するか否かを確認する。

問い合わせの結果、ユーザはアンチウイルスベンダから、マルウェア検知結果もしくは結果なしを表す「No vote」メッセージを受け取る。アンチウイルスベンダから受信するマルウェア検知結果は、「良性：XX 票、悪性：YY 票」で表される。アンチウイルスベンダから、マルウェア検知結果を受け取った場合、未知マルウェア検知ステップにおいて、ユーザのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムを利用してファイルの検査を行う。これは、アンチウイルスベンダのノードにおけるマルウェア検知結果と、ユーザのコンピュータにおけるマルウェア検知結果が、環境の差異などにより異なる可能性があるためである。そのため、最終的なマルウェア判定は、マルウェア判定ステップにおいてマルウェア検知結果とヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知結果を利用して行う。

アンチウイルスベンダから、マルウェア検知結果なしを表す「No vote」メッセージを受け取った場合、当該ファイルは良性ファイルか、ブロックチェーンネットワーク上にもマルウェア検知結果が存在しないマルウェアのいずれかである。そのため、未知マルウェア検知ステップにおいて、ユーザのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムを利用してファイルの検査を行う。その後、マルウェアとして検知された場合、アンチウイルスベンダのノードへ当該ファイルを送信し、マルウェア検知結果を受信する。最終的なマルウェア判定は、マルウェア検知結果を受信した場合と同様に、マルウェア判定ステップにおいて行われる。

3.5.3 未知マルウェア検知ステップ

未知マルウェア検知ステップでは、Malware detection コンポーネントのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムを利用して、未知マルウェアの検知が行われる。インターネット上からダウンロードしたファイルが入力として、「悪性」または「良性」が出力として得られる。各アンチウイルスベンダのマルウェア検知結果と、これらマルウェア検知システムの出力をもとに、マルウェア判定ステップで最終的なマルウェア判定が実施される。

3.5.4 マルウェア判定ステップ

マルウェア判定ステップでは、Maliciousness degree calculation コンポーネントを利用して、悪性度の算出が行われる。悪性度は、ファイルがどの程度悪性かを表す度合いであ

り、詳細は 3.7 で述べる。悪性度は、アンチウイルスベンダのマルウェア検知結果と、ユーザのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知結果をもとに算出される。悪性度がユーザが事前に定めたしきい値を上回った場合は、マルウェアとして判定され当該ファイルが削除される。

3.6 アンチウイルスベンダの処理の流れ

アンチウイルスベンダは、主に以下の 4 つの動作を実施する。

1. マルウェア検知結果検索
2. ファイルハッシュ登録
3. ファイル検査リクエスト
4. マルウェア検知

それぞれについて説明する。

3.6.1 マルウェア検知結果検索

マルウェア検知結果検索は、アンチウイルスベンダの Blockchain communication コンポーネントを利用して行われる。ブロックチェーンネットワーク上から、ユーザから受信したファイルハッシュに対応するマルウェア検知結果を検索する。その後、User communication コンポーネントを利用して、マルウェア検知結果をユーザへ返却する。ブロックチェーンネットワークに、ユーザから受信したファイルハッシュに対応するマルウェア検知結果が存在しない場合、マルウェア検知結果なしを表す「No vote」メッセージをユーザへ返却する。

3.6.2 ファイルハッシュ登録

ファイルハッシュ登録は、アンチウイルスベンダの Blockchain communication コンポーネントを利用して行われる。アンチウイルスベンダがユーザからファイルを受信した際、スマートコントラクトを利用してファイルハッシュをブロックチェーンネットワーク上に登録する。これにより、他のアンチウイルスベンダがファイルの検査結果をスマートコントラクトを利用して共有することが可能となる。

3.6.3 ファイル検査リクエスト

ファイルハッシュ検査リクエストは、アンチウイルスベンダの File sharing コンポーネントを利用して行われる。アンチウイルスベンダがユーザからファイルを受信した際、他のアンチウイルスベンダの検査結果を得るため、他のアンチウイルスベンダへファイル検査リクエストと共にファイルを送信する。

3.6.4 マルウェア検知

マルウェア検知は、アンチウイルスベンダの Malware detection コンポーネントを利用して行われる。ユーザもしくは他のアンチウイルスベンダから受信したファイルが入力として、「悪性」または「良性」が出力として得られる。

3.7 悪性度

悪性度とは、あるファイルがマルウェアであるという疑いの強さを表す度合いである。悪性度は、0 から 1 の範囲の値を取る実数であり、1 に近いほどファイルがマルウェアであるという疑いが強い。この値は、アンチウイルスベンダ群のマルウェア検知結果と、ユーザが所有しているヒューリスティック方式やビヘイビア方式のマルウェア検知結果を利用して計算される。ユーザは悪性度を利用して、最終的にダウンロードしたファイルが悪性であるか否かを判断する。

表 1. 記号の定義

記号	定義
M_d	悪性度
M_t	悪性度に関するしきい値
V_t	「投票」結果を信頼するための投票数
V_b	「良性」の投票数
V_m	「悪性」の投票数
R_v	投票信頼率
R_s	自己信頼率
D_r	マルウェア検知システムの検知結果

ユーザはアンチウイルスベンダからマルウェア検知結果を受信した際、それらのマルウェア検知結果と、ユーザのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知結果に基づき、悪性度を計算する。本節で利用する記号を 1 に示す。

最終的なマルウェアの判定は、悪性度に基づいて判定される。すなわち、駆除判定式 (1) を満たすときには、ファイルをマルウェアとみなし削除する。

$$M_d > M_t \qquad (1)$$
$$0 \leq M_d \leq 1$$

M_d の算出方法は、アンチウイルスベンダの投票総数がユーザが定めるしきい値以上である場合（$V_m + V_b \geq V_t$ の場合）と、そうでない場合（$V_m + V_b < V_t$ の場合）とで異なる。それぞれの場合の算出方法について説明する。

3.7.1 $V_m + V_b \geq V_t$ の場合

アンチウイルスベンダから受信したマルウェア検知結果のみを利用し、式 (2) により悪性度を算出する。

$$M_d = \frac{V_m}{V_m + V_b} \qquad (2)$$

3.7.2 $V_m + V_b < V_t$ の場合

アンチウイルスベンダから受け取ったマルウェア検知結果と、ユーザのヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知結果に基づき、式 (3) により悪性度を算出する。ここで、ユーザが所有するヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムは、検査対象のファイルがマルウェアと判定したときは 1 を、良性と判定したときは 0 を出力するものとする。すなわち、$D_r \in \{0, 1\}$ であるとする。

$$M_d = \frac{V_m}{V_m + V_b} \times R_v + D_r \times R_s \qquad (3)$$

ここで、R_v と R_s は以下の式で定義する。

$$R_v = \frac{V_m + V_b}{V_t} \qquad (4)$$
$$R_s = 1 - R_v \qquad (5)$$

V_t は、アンチウイルスベンダから受けっとたマルウェア検知結果結果にどの程度重み付けを行うのかを規定するパラメータである。すなわち、V_t を増加させることで、ユーザが自分自身のヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知結果に対し、より重み付けを行うことが可能となる。

3.8 提案システムの期待される効果

次に、提案システムの期待される効果について述べる。提案システムの利用により、ヒューリスティック方式やビヘイビア方式のマルウェア検知システムのみを利用場合と比較して、一般にマルウェアの検知精度の向上、すなわち False positive rate と False negative rate の低下が期待できる。しかしながら、ブロックチェーンネットワークにおいてファイルハッシュ値に対応するマルウェア検知結果が存在しない、新規のファイルを入手したユーザと、その後のアンチウイルスベンダの「投票」によりマルウェア検知結果が存在する、ファイルを入手したユーザでは、期待される効果が異なる。

後者は、False positive rate と False negative rate の低下が期待できる。アンチウイルスベンダから受信したファイルハッシュ値に対応するマルウェア検知結果を最終的なマルウェア判定に利用できるためである。

しかしながら、前者は False positive rate のみの低下が期待できる。これは、ブロックチェーンネットワークにおいてファイルハッシュ値に対応するマルウェア検知結果が存在せず、図 14 の青色の部分においてマルウェア検知を行っているためである。例えば、ユーザがダウンロードしたファイルがマルウェアであり、ブロックチェーンネットワークにおいてファイルハッシュ値に対応するマルウェア検知結果が存在しないとする。この時、ユーザは自分自身のヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの検知システムを利用してマルウェアの検知を行う。ここで、このマルウェア検知システムが誤判定、すなわちマルウェアを良性ファイルとして判定した場合は、ユーザの処理は終了する。そのため、False negative rate は向上しない。

一方、例えば、ユーザがダウンロードしたファイルが良性であり、ブロックチェーンネットワークにおいてファイルハッシュ値に対応するマルウェア検知結果が存在しないとする。さらに、自分自身のヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムでは、悪性と判定されたとする。この時、ユーザはアンチウイルスベンダへ当該ファイルを送信し、マルウェア検知結果を受信する。その後、それらのマルウェア検知結果と自分自身のヒューリスティック方式もしくはビヘイビア方式のマルウェア検知システムの判定結果を利用して、最終的なマルウェア判定を行う。そのため、False positive rate の低下が期待できる。

3.9 Trust management

最後に Trust management について説明する。複数のノードやシステムが協調して攻撃に関する情報を共有したり、攻撃を検知したりするネットワークである協調型ネットワーク（Collaborative network）では、一般に Insider attack について考慮する必要がある。本提案システムでは、複数のアンチウイルスベンダのノードがコンソーシアムを成し、マルウェア検知結果を共有する。しかし、何らかの理由でアンチウイルスベンダが悪意のあるノードとなってしまった場合、本来のマルウェア検知結果とは逆の検知結果をネットワーク上に送出してしまう事態が想定される。結果として、提案システムのマルウェア検知精度の低下が懸念される。Trust management では、そのような悪意のあるノードの検知を行う。

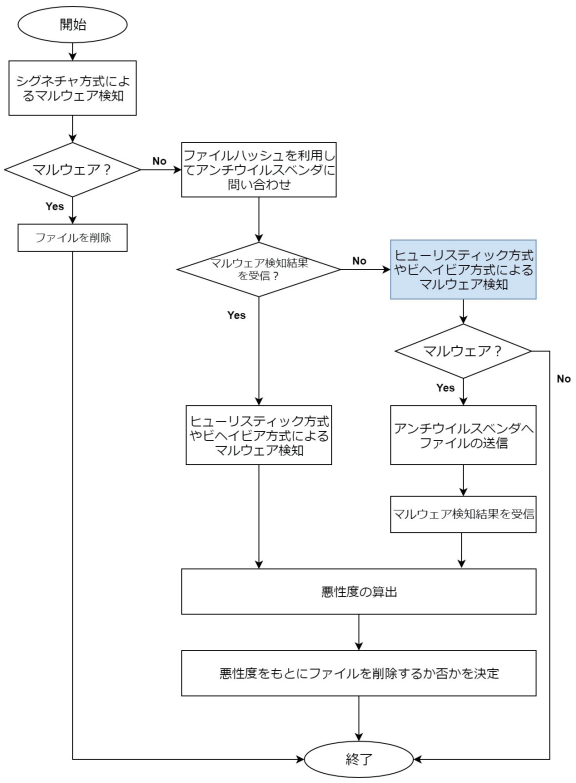

図 14. 各ユーザの処理（再掲）

本小節では、Trust management について説明する。最初に、Trust management の重要性について説明する。その後、提案システムにおける Trust management の流れを説明した後、ノードがどの程度信頼できるかを表す「信頼値」の計算手法について述べる。

3.9.1 Trust management の重要性

協調型ネットワークでは、一般に悪意のあるノードによる Insider attack について考慮する必要がある。Insider attack とは、協調型ネットワーク内の悪意のあるノードが、協調型ネットワークの有効性を低下させるために実施する攻撃の総称であり、Sybil attacks や Betrayal attacks などが存在する。

例えば、本提案システムにおいて、悪性度は各アンチウイルスベンダによる「投票」に基づいて算出される。ここで、何らかの理由でアンチウイルスベンダのノードが悪意のあるノードとなり、故意に本来の検知結果とは逆の検知結果を、「投票」としてブロックチェーンネットワークに送出してしまうと、悪性度の算出に悪影響を及ぼすことになる。すなわち、全てのユーザでマルウェア検知精度の低下を引き起こす。

あるノードが悪意のあるノードとなる要因は様々であるが、一般にはノードのマルウェア感染やノードに対する不正アクセスが要因として考えられる。本提案システムのネットワーク参加者はアンチウイルスベンダであり、マルウェア対策を専門にする企業であるため、ノードのマルウェア感染対策や

不正アクセス対策は十分であるとも考えられる。

しかし、2019年にはロシアのハッカー集団により、大手アンチウイルスベンダが不正アクセスの被害を受けた[25]。この不正アクセスにより、アンチウイルスソフトウェアのソースコードや開発関連のドキュメント、機械学習のモデルといったアンチウイルスソフトウェアに関するデータが窃取された。窃取されたデータは30TB以上であり、30万ドルで販売されているとの報告がある。

以上の報告から、アンチウイルスベンダのノードがマルウェア感染や不正アクセスの被害を受けることも十分に考えられる。したがって、本提案システムにおいても悪意のあるノードによるInsider attackは考慮すべきであり、それらのノードによる悪影響を低減するためにTrust managementは重要である。

3.9.2 Challenge-based trust management

協調型侵入検知システム（CIDS：Collaborative intrusion detection system）では、より複雑で高度な攻撃の検知や検知精度の向上を目的として、複数のノードが攻撃に関する情報の共有を行っている。そのため、本研究と同様にInsider attackへの対策を行う必要がある。文献[26]では、CIDSにおける、Insider attackの一種であるBetrayal attacksやCollusion attacksを行うノードの検知手法を提案している。Betrayal attacksやCollusion attacksは、ある単一もしくは複数のノードが、故意に本来の検知結果とは逆の検知結果をネットワーク上に送出する攻撃であり、攻撃検知に悪影響を及ぼす。これらのノードを検知するため、文献[26]ではChallenge-based trust managementが提案されている。Challenge-based trust managementでは、Test messageと呼ばれるメッセージを検査対象の通信とともに、テスト対象ノードへ送信する。その後、テスト対象ノードの検知結果が自分の期待した検知結果か否かを判断し、その判断をベースにテスト対象の信頼値を算出する。ノード間で交換されるメッセージについては、3.9.3で説明する。

本提案システムでは、各アンチウイルスベンダのマルウェア検知結果がユーザへ送信され、マルウェア判定に利用される。すなわち、各アンチウイルスベンダのマルウェア検知精度は、ユーザにおけるマルウェア検知精度に影響を与える。したがって、Betrayal attacksやCollusion attacksが行われた場合は、提案手法における全てのユーザの検知精度に多大な悪影響を与える。以上から、本研究では文献[26]で提案されたChallenge-based trust managementを利用する。

3.9.3 ノード間で送受信されるメッセージ

Challenge-based trust managementにおいて、ノード間で送受信されるメッセージについて説明する。Challenge-based trust managementでは、以下の2種類のメッセージが利用される。

- Consultation message
- Test message

Consultation messageは、より複雑で高度な攻撃の検知や検知精度の向上を目的として自分自身以外のノードへ、その検知を依頼するためのメッセージである。図15に、Consultation message とその応答を示す。Consultation messageは、検知を依頼したい通信と共に他のノードへ送信される。Consultation messageを受け取ったノードは、その通信がどの程度危険かを表す「危険度」をフィードバックとして返す。

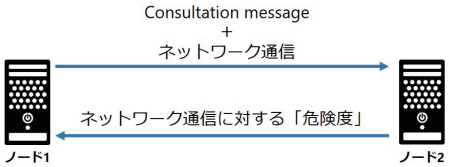

図15. Consultation message

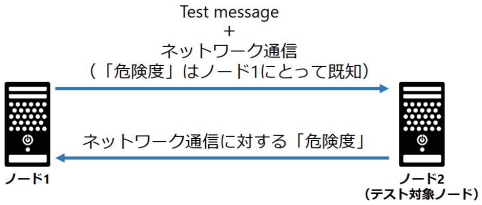

図16. Test message

Test messageは、テスト対象ノードの信頼値を算出するために送信されるメッセージである。図16に、Test messageとその応答を示す。Test messageは、自分自身にとってその「危険度」が既知の通信と共に他のノードへ送信される。また、Test messageはConsultation messageと区別がつかないように、テスト対象ノードへ送信する必要がある。なぜならば、Test messageとConsultation messageの区別がついてしまうと、悪意のあるノードが受信したメッセージによってその動作を切り替えることが可能となるためである。すなわち、Test messageを受信した際は信頼値が増加するようなレスポンスを、Consultation messageを受信した際は故意に本来の検知結果とは逆の検知結果をレスポンスとして返すことが可能となってしまう。Test messageを受け取ったノードは、Consultation messageと同様に、その通信がどの程度危険かを表す「危険度」をレスポンスとして返す。Test messageの送信元は、そのレスポンスと「期待したレスポンス」を比較することによって、信頼値を算出する。信頼値の算出については、3.9.5で説明する。

3.9.4 提案システムにおけるTrust management

文献[26]で提案されたChallenge-based trust managementはCIDSにおける、IDSノードのTrust managementである。しかし、本研究における提案システムでは、アンチウイルスベンダのノードのTrust managementが必要である。したがって、ノード間で交換される情報を決定する必要がある。さらに、文献[26]では、ノード間で送受信されるメッセージに関し、そのプロトコルについては詳述していない。そのため、本研究においてそのプロトコルを規定する必要がある。

提案システムにおけるTrust managementを図17に示す。最初に、Test messageとConsultation messageについて説明する。Challenge-based trust managementでは、Test messageやConsultation messageと共にネットワーク上の通信が他のノードへ送信される。しかし、本研究における提案システムでは、協調型ネットワークの参加者はIDSノードではなくアンチウイルスベンダのノードである。そのため、Test

message と Consultation message と共にファイルを他のアンチウイルスベンダのノードへ送信する。ここで、本研究における提案システムでは、Consultation message はファイル検査リクエストに対応する。

次に、Test message と共に利用されるファイルの制約について説明する。本研究では、情報共有基盤としてブロックチェーン技術を採用しており、ブロックチェーン上のデータは全てのアンチウイルスベンダのノードが閲覧可能である。したがって、ブロックチェーンネットワークにおいて、マルウェア検知結果が既知のファイルを Test message と共に送信しても意味をなさない。なぜならば、マルウェア検知結果が既知である場合、テスト対象ノードがその検知結果をもとにレスポンスを返すことが可能であるためである。そのため、Test message と共に利用されるファイルは、ブロックチェーンネットワークにおいて、マルウェア検知結果が未知のファイルに限定される。したがって、本研究ではユーザが新たにアンチウイルスベンダへ送信するファイルを Test message と共に利用する。

次に、ノード間で送受信されるメッセージに関するプロトコルについて説明する。本研究では情報共有基盤としてブロックチェーン技術を採用しているため、テスト対象のアンチウイルスベンダのノードにのみ Test message を送信しても、Consultation message と区別がついてしまう。なぜならば、テスト対象のアンチウイルスベンダのノードがブロックチェーン上のデータを閲覧することによって、「投票」の有無が確認できてしまうためである。また、Test message をテスト対象ノードに送信すると同時に、Consultation message をテスト対象ノード以外に送信しても、テスト対象ノードはブロックチェーン上の「投票」結果をレスポンスとして利用できてしまう。そこで本研究では、図 17 で示すように、Test message と Consultation message を同時にノードへ送信し、「投票」のタイミングの同期を取ることによって、信頼値の算出を行う。信頼値の算出については、3.9.5 で説明する。

最後に、Test message に対するレスポンスと期待したレスポンスの比較について説明する。本研究における提案システムでは、ブロックチェーンネットワークにおいてマルウェア検知結果が既知のファイルを Test message と共に送信しても意味をなさない。そのため、Test message と共に利用されるファイルはブロックチェーンネットワークにおいてマルウェア検知結果が未知のファイルに限定される。しかしながら、ブロックチェーンネットワークにおいてマルウェア検知結果が未知のファイルは、自分自身にとってもその検知結果が未知であるため、「期待したレスポンス」を作成することが出来ない。そこで、Test message に対するレスポンスが「期待したレスポンス」であるか否かは、テスト対象ノード以外に送信した Consultation message に対応するレスポンスを利用した多数決で決定する。すなわち、Test message に対するレスポンスがテスト対象ノード以外に送信した Consultation message に対応するレスポンスの多数側であれば、当該レスポンスを「期待したレスポンス」とする。

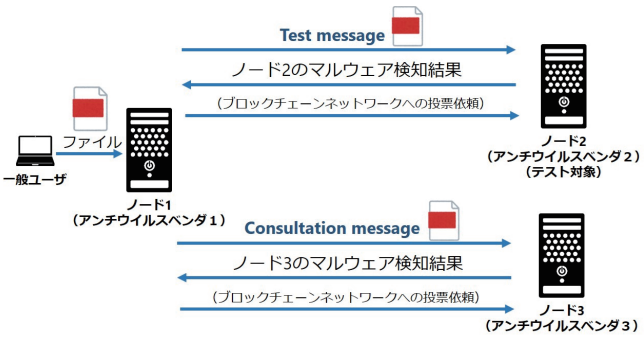

図 17. 提案手法における Trust management

3.9.5 信頼値計算

最後に、信頼値の計算手法について説明する。信頼値の計算方法は、本研究における提案ではなく、文献 [26] にて提案されている手法である。文献 [26] では、「Satisfaction level」と呼ばれる、Test message のレスポンスに対してどの程度「満足」したかを表す値をベースに信頼値の算出を行う。また、悪意のあるノードの迅速な検知のため、最新の Test message のレスポンスに対する Satisfaction level に、より大きな重みを付けを行い信頼値の算出を行う。

あるノード i のノード j に関する信頼値は、以下の式で算出される。

$$t_j^i = \frac{\sum_{k=0}^{n} S_k^{i,j} F^{t_k}}{\sum_{k=0}^{n} F^{t_k}} \tag{6}$$

ここで、n はノード i がノード j から受信した Test message に対するレスポンス数である。$S_k^{i,j}$ は、「Satisfaction level」と呼ばれ、本研究では $S_k^{i,j} \in \{0,1\}$ である。Test message に対するレスポンスが「期待したレスポンス」であれば、$S_k^{i,j} = 1$ である。また、$S_0^{i,j}$ は時間的に最も古い Test message に対するレスポンスの Satisfaction level である。$F(0 \leq F \leq 1)$ は「Fogetting factor」と呼ばれる。t_k は、最新の Test message のレスポンスに対する Satisfaction level に、より大きな重みを付けを行うためのパラメータである。

本研究では、算出された t_j^i が事前に定めたしきい値 T_v を下回った場合、ノード i に対応するアンチウイルスベンダは、ユーザへ送信するマルウェア検知結果にノード j に対応するアンチウイルスベンダからのマルウェア検知結果を含めない。

4. 評価実験

本節では、評価実験について述べる。評価実験では、提案システムのマルウェア検知精度評価と性能評価を行った。最初に、提案システムのマルウェア検知精度評価について説明を行った後、提案システムの性能評価について述べる。

4.1 提案システムのマルウェア検知精度評価

最初に、提案システムの検知精度の評価について述べる。提案システムにおいて最終的なマルウェア判定は、悪性度をもとに行われる。悪性度を算出する際に、ユーザは「投票」結果を信頼するための投票数 V_t を決定し、投票信頼率 R_v と自己信頼率 R_s を定める必要がある。したがって、ユーザが「投票」結果を信頼するための投票数 V_t を決定するために、V_t が提案手法のマルウェア検知精度に与える影響について調査す

る必要がある。また、本提案システムの目的はマルウェア検知精度の向上である。そのため、ユーザが V_t を定め提案システムを利用した場合と、提案システムを利用しない場合、すなわち単一のヒューリスティック方式やビヘイビア方式のマルウェア検知システムを利用した場合のマルウェア検知精度と比較する必要がある。

4.1.1 マルウェア検知精度の評価指標

次に、マルウェア検知精度の評価指標について説明する。マルウェア検知精度の評価指標として、False negative rate (FNR) と False positive rate (FPR) を利用する。FNR はマルウェアを誤って良性ファイルとして誤って判定した割合、FPR は良性ファイルを誤ってマルウェアとして誤って判定した割合である。以下に、FNR と FPR の定義を示す。

$$FNR = \frac{FN}{TP + FN} \tag{7}$$

$$FPR = \frac{FP}{TN + FP} \tag{8}$$

式（7）と式（8）における記号の定義を表 2 に示す。

4.1.2 実験環境と方法

評価実験では、Python スクリプトを利用して提案システムのプロトタイプを作成した。実験環境を図 18 に示す。

評価実験の目的は、投票信頼率もしくは自己信頼率が提案システムのマルウェア検知精度に与える影響について調査し、提案システムを利用する場合と、提案システムを利用しない場合、すなわち単一のヒューリスティック方式やビヘイビア方式のマルウェア検知システムを利用した場合のマルウェア検知精度と比較することである。したがって、今回の評価実験ではユーザにおけるシグネチャ方式のマルウェア検知システムについては考慮していない。

評価実験ではヒューリスティック方式やビヘイビア方式のマルウェア検知システムに関して、擬似的なマルウェア検知システムを実装した。擬似的なマルウェア検知システムとは、パラメータとして FPR と FNR を持つ、ヒューリスティック方式やビヘイビア方式のマルウェア検知システムを模したプログラムである。「マルウェア」とみなすファイル、もしくは「良性」とみなすファイルが入力された際に、これらのパラメータをもとに、当該ファイルがマルウェアか否かを判定する。また、本評価実験では検知精度への影響はないため、ブロックチェーン技術は利用しておらず、Python スクリプトのみを利用してシミュレーションを実施している。

シミュレーションは全てのユーザが、事前に定めた全ての「マルウェア」とみなすファイルと「良性」とみなすファイルに対して「マルウェア」もしくは「良性」と判定するまで実施される。

4.1.3 パラメータ

評価実験におけるパラメータとその値を表 3 に示す。False negative 発生率と False positive 発生率は、4.1.2 で述べた擬似的なマルウェア検知システムのパラメータである。これらの値は文献 [27] をもとに決定した。アンチウイルスベンダ数は VirusTotal のファイルスキャンにおいて利用可能なアンチウイルスエンジン数 [28] をもとに、71 に設定した。また、

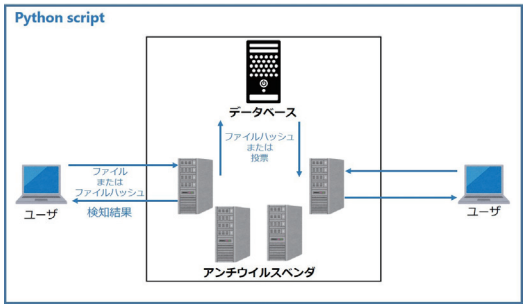

図 18. 実験環境

$M_t = 0.5$ とした。これは、本研究ではファイルが悪性であれば $D_r = 1$、ファイルが良性であれば $D_r = 0$ と定義しているためであり、各ユーザの個別の環境やポリシー等を考慮しない場合は、$M_t = 0.5$ が妥当であると考えたためである。

4.1.4 実験結果と考察

FNR と FPR の実験結果を図 19 と図 20 に示す。図の横軸は「投票」結果を信頼するための投票数を表し、縦軸はそれぞれ FNR と FPR の値を表す。また、図中の青い点線は、提案システムを利用しない場合、すなわち、ユーザが自分自身のヒューリスティック方式やビヘイビア方式のマルウェア検知システムのみを利用した場合の FNR もしくは FPR の値を表す。

実験結果より、「投票」結果を信頼するための投票数が増加するにしたがって、FNR と FPR の値が本来のヒューリスティック方式やビヘイビア方式のマルウェア検知システムの FNR と FPR に近づいていくことが分かる。すなわち、悪性度算出の際に、ユーザが自分自身のヒューリスティック方式やビヘイビア方式のマルウェア検知システムの検知結果に対して、より重みづけを行った場合（自己信頼率 R_s を増加させた場合）は、提案システムの有効性が減少していくことが分かる。したがって、本評価実験におけるパラメータの場合は、ユーザは「投票」結果を信頼するための投票数をアンチウイルスベンダ数に近づける、すなわち「投票」結果に対してより重みづけを行う必要がある。ユーザが「投票」結果に対してより重みづけを行った場合は、FNR と FPR 双方とも 0 に近い値となっており、本来のヒューリスティック方式やビヘイビア方式のマルウェア検知システムにおける FNR と FPR と比較して減少していることから、提案システムの有効性が確認できる。

これらのマルウェア検知精度向上の理由は、自分自身のヒューリスティック方式やビヘイビア方式のマルウェア検知システムの検知結果だけでなく、アンチウイルスベンダの「投票」結果を利用しマルウェア検知を行ったためであると考えられる。自分自身のヒューリスティック方式やビヘイビア方式のマルウェア検知システムだけでは、False positive や False negative が発生しがちであるが、アンチウイルスベンダの「投票」結果を利用し悪性度を算出することで、最終的には良性ファイルやマルウェアを正しく判定、検知することが出来たためであると考えられる。

4.2 提案システムの性能評価

次に、提案手法の性能評価について説明する。本研究では、以下の項目について評価を行った。

表 2. 式（7）と式（8）における記号の定義

記号	定義
TP	マルウェアをマルウェアとして正しく検知できた（True Positive）数
TN	良性ファイルを良性ファイルとして正しく判定できた（True Negative）数
FP	良性ファイルが誤ってマルウェアとして検知された（False Positive）数
FN	マルウェアが誤って良性ファイルとして判定された（False Negative）数

表 3. パラメータ

パラメータ	数値
ユーザ数	1000
「マルウェア」とみなすファイル数	1000
「良性」とみなすファイル数	1000
「投票」結果を信頼するための投票数 (V_t)	71 – 180
悪性度に関するしきい値 (M_t)	0.5
アンチウイルスベンダ数	71
False negative 発生率	0.05
False positive 発生率	0.06

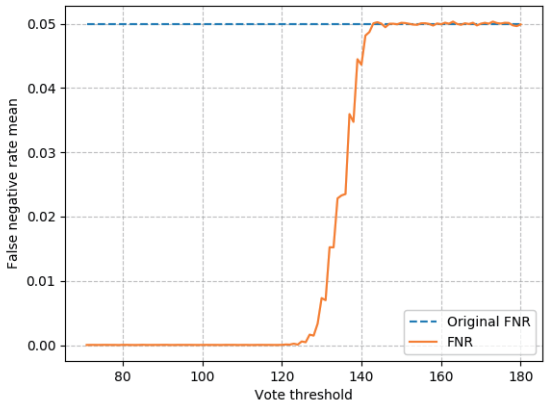

図 19. FNR の評価実験結果

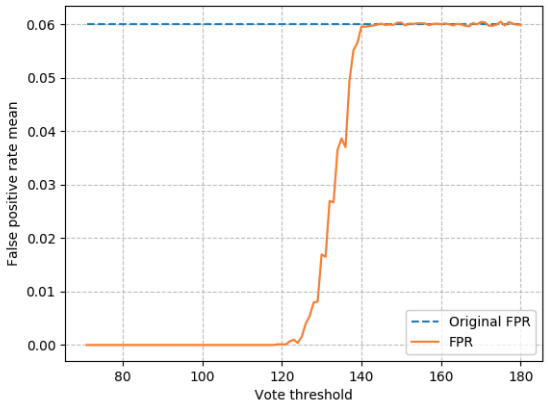

図 20. FPR の評価実験結果

- マルウェア検知結果検索性能
- ブロックチェーンサイズ

マルウェア検知結果検索性能では、アンチウイルスベンダにおけるファイルハッシュに対応する「投票」結果の検索時間を評価する。ブロックチェーンはその性質上、時間の経過とともにブロック数が増加する。また、各ブロックは各アンチウイルスベンダの「投票」を含んでいる。したがって、ユーザへマルウェア検知結果を返却するためには、全てのブロックを辿りファイルハッシュに対応する各アンチウイルスベンダの「投票」を検索し集計する必要があると論理的に考えられる。すなわち、ブロック数の増加に従いその検索時間が増加することが懸念される。本評価実験では、コンソーシアムブロックチェーンのプラットフォームとして広く活用されている Hyperledger Fabric を利用し、ファイルハッシュに対応する「投票」結果の検索時間を評価した。

さらに、本研究ではブロックチェーンサイズについて評価を行った。ブロックチェーンは時間の経過とともにそのブロック数、すなわちブロックチェーンサイズが増加していく。仮にストレージ容量を超えるブロックチェーンサイズとなった場合、新たに追加されるファイルハッシュや、それに対応するアンチウイルスベンダの「投票」をストレージに格納できない事態となる。本評価実験では、マルウェア検知結果検索性能の評価と同様に Hyperledger Fabric を利用し、ブロックチェーンサイズについて評価を行った。

4.2.1 実験環境と方法

本評価実験では、仮想化ソフトウェアとして Oracle VM VirtualBox を利用し、Windows 10 が稼働するホスト PC 上にゲスト OS として Ubuntu Desktop 18.04.3 LTS が動作する仮想環境を構築した。実験環境を表 4 に示す。次にそれぞれの評価方法について説明する。

4.2.1.1 マルウェア検知結果検索性能の評価方法

最初に、ランダムなファイルハッシュを N 件作成し、ブロックチェーン上に登録し「悪性:0、良性：0」で初期化する。

本評価実験では、$N = 1000, 30000, 64758$ とした。次に、ランダムなファイルハッシュ値をキーとして投票を検索する。マルウェア検知結果検索性能は、検索を 100 回実施し、それぞれの検索に要した時間で評価を行う。検索に要する時間の計測はスマートコントラクト上で実施した。

4.2.1.2 ブロックチェーンサイズの評価方法

本提案システムにおいて、各ユーザは個人向けのコンピュータであるパーソナルコンピュータ（パソコン）を利用していると仮定している。すなわち、パソコンの OS として Windows や macOS が利用されていると仮定している。そのため、アンチウイルスベンダのブロックチェーンネットワーク上では Windows や macOS をターゲットとしたマルウェアに関する情報が共有される。本評価実験では、ユーザがパソコンの OS として Windows を利用していると仮定して評価を行った。

AV-TEST の SECURITY REPORT 2017/18[5] によると、2017 年の新種マルウェアの観測数は約 1 億 2,000 万個であり、そのうちの 67.07% が Windows をターゲットとしたマルウェアであった。つまり、Window をターゲットとしたマルウェアは、平均すると 1 秒あたり 2.58 個発生している。したがって、アンチウイルスベンダのブロックチェーンネットワーク上では、マルウェアのファイルハッシュの登録が平均すると 1 秒あたり 2.58 回行われる。

ここで、新種マルウェアのある 1 検体に対して、ブロックチェーンネットワークでは以下の手順でファイルハッシュの登録と各アンチウイルスベンダの「投票」が行われる。

1. あるアンチウイルスベンダから、新種マルウェアのファイルハッシュの登録を表すトランザクションの発行
2. 1. のトランザクションを含むブロックの生成
3. 各アンチウイルスベンダから、「投票」を表すトランザクションの発行
4. 2. のトランザクションを含むブロックの生成

したがって、Hyperledger Fabric のブロック生成時間を 2 秒（デフォルト値）とすると、図 21 のようなブロックチェーンとなり、2 秒毎にブロック数が 1 つずつ増加していく。評価実験の環境では、ブロックのみのサイズが 1858 バイト、投票トランザクションのサイズが 3163 バイト、ファイルハッシュ登録トランザクションのサイズは 3145 バイトであった。

アンチウイルスベンダ数を $n(n \geq 1)$ とすると、2 秒毎に生成されるブロックのサイズ BL_s は以下の式で表される。

$$BL_s = 3145 \times 5 + 3163 \times 5 \times n + 1858 \text{（バイト）} \quad (9)$$

従って、1 年間に増加するブロックチェーンサイズ BC_s は、以下の式で表される。

$$BC_s = 15768000 \times BL_s \text{（バイト）} \quad (10)$$

評価実験では、アンチウイルスベンダ数 n をパラメータとして、1 年間で増加するブロックチェーンサイズについて評価する。

4.2.2 実験結果と考察

マルウェア検知結果検索性能の実験結果を表 5 に示す。表 5 から、ブロックチェーン上のファイルハッシュ数が増加しても、マルウェア検知結果検索性能には影響を及ぼしていないことが分かる。文献 [8] では、オープンソースのアンチウイルスソフトである Clam AntiVirus において、MD5 シグネチャを利用したマルウェア検知に要する時間を測定している。MD5 シグネチャの総数が 64,758 であるとき、MD5 シグネチャを利用したマルウェア検知に要する時間は 0.054 秒であった。提案システムでは、ファイルハッシュ数が 64,758 であるときのマルウェア検知結果検索性能 0.00066 秒であることから、Clam AntiVirus における MD5 シグネチャを利用したマルウェア検知と比較しても十分に高速であることが分かった。ファイルハッシュ数が増加しても、投票検索時間が増加しない理由としては、Hyperledger Fabric が State DB と呼ばれるデータベースを採用しているためであると考えられる。State DB はブロックチェーンの最新の状態を表すデータベースであり [29]、State DB により全てのブロックを辿らなくとも、ファイルハッシュに対するマルウェア検知結果を高速に検索できるためであると考えられる。したがって、提案システムを Hyperledger Fabric で実装した場合はマルウェア検知結果検索性能に関しては問題ないと言える。

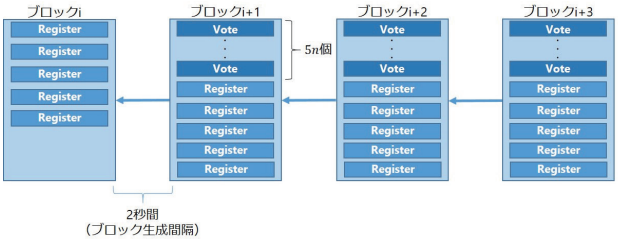

図 21. ブロックチェーンサイズの増加

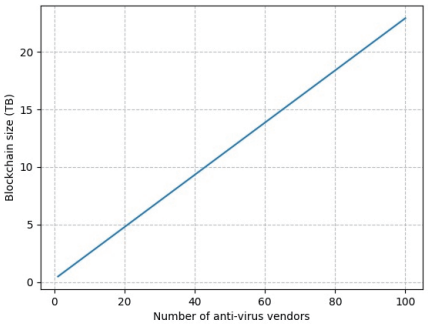

図 22. 1 年間で必要となるストレージ容量

次に、ブロックチェーンサイズの結果を図 22 に示す。式 (10) で説明したように、一年間で必要となるブロックチェーンサイズ BC_s は、アンチウイルスベンダ数 n の一次式で表されることから、グラフは直線となる。VirusTotal では、71 種類のアンチウイルスエンジンを利用してマルウェア検知が可能である [28]。これらのアンチウイルスエンジンを開発する全てのアンチウイルスベンダがブロックチェーンネットワークに参加した場合 ($n = 71$ の場合)、一年間で必要となるブロックチェーンサイズは、約 16TB となる。したがって、現在のストレージ性能を考慮すると、提案システムにおけるブロックチェーンサイズに関しては問題ないといえる。

表 4. 実験環境

ホスト OS	Windows 10
ゲスト OS	Ubuntu Desktop 18.04.3 LTS
ゲスト OS メモリ	8GB
ゲスト OS コア数	2
ブロックチェーンプラットフォーム（バージョン）	Hyperledger Fabric（1.4.0）
スマートコントラクト言語	Go 言語
State DB	LevelDB

表 5. マルウェア検知結果検索性能の評価実験結果

ファイルハッシュ数	投票検索時間（秒）	標準偏差
1000	0.00091	0.00132
30000	0.00126	0.00057
64758	0.00066	0.00095

5. 関連研究

本節では、関連研究について説明する。最初に、協調型セキュリティにおける文献を紹介した後、セキュリティ分野においてブロックチェーン技術を活用している文献を紹介する。最後に、紹介した文献と本研究との差異について述べる。

5.1 協調型セキュリティ

協調型セキュリティ（Collaborative security）は、「Instead of centrally managed security policies, nodes may use specific knowledge (both local and acquired from other nodes) to make security-related decisions」と定義されている [30)31)]。平易に言えば、複数の組織やネットワーク、ノードからの情報を利用して、あるセキュリティ関連の判定、例えば通信やファイルが悪性か否かの判定を行う手法である。協調型セキュリティでは、協調型侵入検知システム（CIDS：Collaborative intrusion detection）を筆頭に様々な分野が存在するが、本研究において特に関連が深い協調型マルウェア検知システム（CMDS：Collaborative malware detection system）の文献を紹介する。

5.1.1 CloudAV

協調型マルウェア検知システムでは、新種マルウェアの検知やマルウェア検知精度の向上を目的として、複数のアンチウイルスソフトウェアのマルウェア検知結果を利用、もしくは複数のノードがマルウェア情報の共有を行う。

Oberheide ら [14)] は、複数のアンチウイルスエンジンを利用しマルウェア検知を行う CloudAV を提案している。背景としては、単一のアンチウイルスベンダでは新種マルウェアへの対処が困難であること、またアンチウイルスソフトウェアの複雑化に伴うアンチウイルスソフトウェア自身の脆弱性が挙げられている。そこで、CloudAV ではクラウド上で複数のアンチウイルスエンジンを利用しマルウェアの検知を行う形態をとっている。アンチウイルスエンジンとしては、10 種類のシグネチャ方式のアンチウイルスソフトウェア、2 種類のビヘイビア方式のアンチウイルスソフトウェアが利用されている。CloudAV は host agent、network service、archival and forensics service の 3 種類のコンポーネントから構成される。各ユーザのコンピュータには、ファイルを一意に識別する ID の算出や network service へのファイル送信の役割を担う host agent がインストールされる。network service では、複数のアンチウイルスエンジンによるマルウェア検知が実施される。また、forensics service にはマルウェアの挙動など、マルウェアに関する情報が保存される。各ユーザは、host agent を介して network service にファイルを送信することでマルウェア検知を行うことが出来る。

評価実験では、評価用のデータセットとして、Arbor Network’s Arbor Malware Libraly の 7220 個のマルウェア検体が利用されている。1 日前に取得されたマルウェアにおいて、単一のアンチウイルスエンジンを利用した場合では、マルウェア検知率は 52%であった。一方、10 種類のアンチウイルスエンジンを利用した場合では、その検知率は 98%であった。

5.1.2 VirusTotal

CloudAV と同様に複数のアンチウイルスエンジンを利用して、マルウェア検知を行うことが可能なサービスが実際に出現してきている。VirusTotal[32)] では、約 70 種類程度のアンチウイルスエンジンを利用したマルウェア検知が可能である [28)]。ユーザは、Web サイトやファイルの検査、URL やファイルハッシュなどを利用した検査結果の検索を行うことができる。さらに、そのファイルの詳細や挙動、他ユーザからのコメントを確認できるとともに、当該ファイルが悪性か良性かを「投票」することが可能である。

5.1.3 CLOUDMARK

複数のアンチウイルスエンジンをベースにしたマルウェア検知ではなく、複数ユーザによるファイルの「Nomination」をベースにマルウェア検知を行うシステムが提案されている。O’Donnell ら [33)] は、複数のユーザのマルウェアの疑いのあるファイルの「Nomination」をベースに、当該ファイルがマル

ウェアか否かを判断するCLOUDMARKを提案している。例えばユーザがファイルを含むe-mailを受信し、そのe-mailがスパムであると判断した場合、そのメールはCLOUDMARKのバックエンドシステムへ送信、すなわち「Nomination」される。バックエンドシステムでは、送信されたメールからファイルを抽出しそのシグネチャを作成する。その後、複数ユーザから当該ファイルが「Nomination」された場合、そのファイルをマルウェアと判定する。

評価実験では、CLOUDMARKにおいてマルウェアと判定されたシグネチャを利用し、オープンソースのアンチウイルスソフトウェアであるClamAVと比較を行っている。評価実験の結果、CLOUDMARKにおいてマルウェアと判定されたシグネチャの80%は、ClamAVでは検出されなかった。すなわち、ClamAVでは検知されないマルウェアを検知できた。また、それらシグネチャの50%は、2日後にはClamAVにおいて検出された。同様の傾向が商用のアンチウイルスソフトウェアでも見られたと報告している。

5.2 セキュリティ分野におけるブロックチェーン技術の活用

近年、ブロックチェーン技術は、ロジスティクス分野や医療分野をはじめとする様々な産業分野での応用が検討されている[15]。同様に、セキュリティ分野においても情報共有の基盤として、その活用が期待されている。以下に、幾つかの文献を紹介する。

5.2.1 CB-MDEE

Jingjingら[34]は、モバイルデバイスを対象として、マルウェア検知と分類を行うConsortium Blockchain for Malware Detection and Evidence Extraction (CB-MDEE) フレームワークを提案している。CB-MDEEは、パブリックブロックチェーンPBとコンソーシアムブロックチェーンCBの2つのブロックチェーンで構成される。PBに所属するユーザは、Sensitive Behavior GraphやInstallation Packageなどから作成される、モデルを利用してマルウェアを検知・分類し、それらの情報をPB上に保存する。CBに属するマルウェア検知組織のメンバーは、それらの情報を利用してモデルを更新するために、fact-baseと呼ばれるデータベースを作成する。評価実験では、アンドロイドのマルウェアにおいて、CB-MDEEは94%の分類精度を達成した。

5.2.2 スマートコントラクトを利用したセキュリティアナリスト支援

スマートコントラクトとは、プログラム化された契約であり、ブロックチェーン上でプログラムとして実行されることにより、ブロックチェーンを利用した様々なアプリケーションが開発可能となっている。現在は、一部のブロックチェーンプラットフォーム上で利用可能である。

Romanら[35]は、セキュリティアナリスト支援のため、ブロックチェーン技術とスマートコントラクト上に実装したオートエンコーダを利用して、セキュリティインシデント報告を分類・管理するためのシステムを提案している。セキュリティインシデント報告がスマートコントラクトに入力として渡されると、システムはその報告を分類し、ブロックチェーン上に保存されている、過去の類似するセキュリティインシデント報告を出力する。提案システムにより、分類・管理が自動的に実施されるため、セキュリティインシデントに対して迅速な対応が可能となる。評価実験では、提案システムの有効性検証のため、訓練用データとして5,850のセキュリティインシデント報告が、テスト用データとして584のセキュリティインシデント報告が利用されている。評価実験の結果「fulldisclosure」カテゴリでは、True Positive Rateは0.991、FPRは0.059を達成した。

5.2.3 PolySwarm

PolySwarm[36]は、マルウェア検知のクラウドサービスを提供している。利用者は、検査を依頼したいファイルをPolySwarmを介して複数のセキュリティエキスパートに送信する。それぞれのセキュリティエキスパートは、独自のアンチウイルスエンジンを開発・保有しており、そのアンチウイルスエンジンを利用して、当該ファイルの検査を行う。利用者は、複数のセキュリティエキスパートからの検査結果をもとに、最終的に当該ファイルがマルウェアか否かを判断する。セキュリティエキスパートの検査結果が「正解」であったかは、第三者が判定し評価を行う。「正解」であったセキュリティエキスパートには、利用者が支払う報酬金が、仮想通貨として配布される。PolySwarmにおいて、ブロックチェーン技術は仮想通貨とセキュリティエキスパートの評価結果の保存に活用されている。

5.2.4 CBSigIDS

協調型侵入検知システムの分野においてもブロックチェーン技術が注目を集めている[37]。例えば、Liら[38]はIoT環境における、ブロックチェーン技術を用いた協調型侵入検知システムのフレームワークであるCBSigIDSが提案されている。ブロックチェーン上では、IDSで利用されるシグネチャが保存され、コンソーシアム型のブロックチェーンが採用されている。ブロックチェーン技術を利用することにより、信頼できる中央機関を利用せず、かつ耐改ざん性を保ちつつ、複数のIDSノード間でシグネチャの共有が可能であると主張している。評価実験では、シミュレーションと実環境において、攻撃（Worm attackやFlooding attack）を検知しその通信の遮断に成功したノード数に関して評価を行っている。

5.3 関連研究と本研究との差異

協調型マルウェア検知システムの分野において、複数のアンチウイルスソフトウェアを検知結果を利用しマルウェア検知を行う手法が存在するが、より迅速に、かつ効果的にマルウェア検知を行うためにはアンチウイルスベンダの間の連携は必要不可欠であると考える。しかしながら、複数アンチウイルスベンダ間のマルウェア検知結果に関する情報共有とそれらを利用したマルウェア検知に関しては研究がなされていない。そこで本研究では、複数のアンチウイルスベンダの間でマルウェア検知結果に関する情報を共有し、それらの情報を利用したマルウェア検知を行うシステムを提案した。したがって、「得意領域」が異なる複数のアンチウイルスベンダのマルウェア検知結果に関する情報を活用し、それらを活用してマルウェア検知精度の向上が期待できる点で上記文献[14][32][33]

と異なる。

また、近年ブロックチェーン技術はセキュリティ分野においても注目を集めており、データ共有の基盤としてその活用が期待されている。本研究では、ブロックチェーン技術を活用したアンチウイルスベンダ間のマルウェア検知結果に関する情報共有を提案している。したがって、複数アンチウイルスベンダ間で情報共有を行っている点で上記文献 [34)35)36)38)] と異なる。

6. まとめ

マルウェアはコンピュータだけでなく、インターネットが基盤となっている現代社会に甚大な悪影響を及ぼすことから、その検知は重要である。マルウェア感染の被害は今後も増加していくことが予測される。マルウェア対策のため、我々が普段利用するコンピュータに、アンチウイルスソフトウェアを導入しているが、通常 1 種類のみのアンチウイルスソフトウェアしか導入していない。しかしながら、新種マルウェアが急増している現代において、単一のアンチウイルスベンダでは急増するマルウェアへの対策に限界ある。

そこで本研究では、複数のアンチウイルスベンダの間でマルウェア検知結果に関する情報を共有し、それらの情報を利用したマルウェア検知を行うシステムを提案した。アンチウイルスベンダはそれぞれ、独自にマルウェアに関する情報の収集を行っている。そのため、アンチウイルスベンダそれぞれでマルウェア検知に関する「得意領域」が異なると考えられる。したがって、「得意領域」が異なる複数のアンチウイルスベンダ間でのマルウェア検知結果に関する情報を共有し活用すれば、マルウェア検知精度の向上が期待できる。

評価実験では、提案システムのマルウェア検知精度評価と性能評価をシミュレーションを通して行った。提案システムのマルウェア検知精度の評価実験の結果、従来のマルウェア検知である単一のヒューリスティック方式やビヘイビア方式のマルウェア検知システムを利用した場合と比較して、提案システムの利用によりマルウェアの検知精度は大きく向上することが分かった。また、提案システムの性能評価では、マルウェア検知結果検索性能とブロックチェーンサイズについて評価を行った。評価実験の結果、コンソーシアム型のブロックチェーンプラットフォームとして広く利用されている Hyperledger Fabric を利用して提案システムを実装した場合は、双方とも問題ないことを確認した。

最後に、今後の課題について述べる。本研究では、提案システムにおける Trust management でノード間で送受信される情報や、その流れについて規定した。しかしながら、実際に悪意のあるノードが発見できるかといった評価実験については行っていない。また、アンチウイルスベンダのマルウェア検知結果共有に対するインセンティブについても検討する必要がある。すなわち、共有した情報に対する何かしらの利益を与える仕組みが必要である。アンチウイルスベンダは企業である以上利益を追求する団体である。アンチウイルスベンダにとっては、マルウェアの検知精度の良し悪しが競争優位性に影響を与えるが、マルウェアの検知結果を共有することは競争優位性に悪影響を与える。しかしながら、アンチウイルスベンダ間でマルウェア検知結果を共有することは、ユーザ側でのマルウェア検知精度向上につながる。したがって、アンチウイルスベンダ間においてマルウェア検知結果の共有を促進するような仕組みが必要である。

参考文献

1) S Mohurle and M Patil: A brief study of wannacry threat: Ransomware attack 2017., International Journal of Advanced Research in Computer Science, Vol.8, 2017.

2) https://techcrunch.com/2019/05/12/wannacry-two-years-on/ (accessed 2019/05/17).

3) H Sultan, et al.: A SURVEY ON RANSOMWARE: EVOLUTION, GROWTH, AND IMPACT., International Journal of Advanced Research in Computer Science, Vol 9, 2018.

4) D Barrera, I Molloy, and H Huang: IDIoT: Securing the Internet of Things like it's 1994., arXiv preprint arXiv:1712.03623, 2017.

5) https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2017-2018.pdf (accessed 2019/06/16).

6) A Souri and R Hosseini: A state-of-the-art survey of malware detection approaches using data mining techniques., Human-centric Computing and Information Sciences 8, 2018.

7) http://www.clamav.net/ (accessed 2020/01/06).

8) X Zhou, et al.: MRSI: A fast pattern matching algorithm for anti-virus applications., Seventh International Conference on Networking (icn 2008), pp. 256-261, 2008.

9) https://eset-info.canon-its.jp/malware_info/qa/detail/141204_2.html (accessed 2020/01/06).

10) Z Bazrafshan, et al.: A survey on heuristic malware detection techniques., Information and Knowledge Technology (IKT), 2013 5th Conference, pp.113-120, 2013.

11) https://eset-info.canon-its.jp/malware_info/technology/detail/140626_3.html (2020/01/06 閲覧).

12) https://media.kaspersky.com/pdf/Kaspersky_Lab_Whitepaper_System_Watcher_ENG.pdf (accessed 2020/01/06).

13) 橋本遼太, 吉岡克成, 松本勉: 未検知マルウェアへの対応に基づくアンチウイルスソフトウェアの評価, 研究報告マルチメディア通信と分散処理 (DPS), pp.1-8, 2012.

14) J Oberheide, E Cooke, and F Jahanian: CloudAV: N-Version Antivirus in the Network Cloud., USENIX Security Symposium., pp.91-106, 2008.

15) J. Al-Jaroodi and N. Mohamed.: Blockchain in Industries: A Survey., IEEE Access 7, pp.36500-36515, 2019.

16) S.Nakamoto: Bitcoin: A peer-to-peer electronic cash system., 2008.

17) Z Zheng, et al.: An overview of blockchain technology: Architecture, consensus, and future trends., IEEE 6th International Congress on Big Data, pp.557-564, 2017.

18) Z Zheng, et al.: Blockchain challenges and opportunities: A survey., International Journal of Web and Grid Services 14, pp.352-375, 2018.

19) https://www.ethereum.org/ (accessed 2019/06/16).

20) https://www.hyperledger.org/ (accessed 2019/06/16).

21) https://www.uport.me/ (accessed 2019/06/16).

22) R.Fuji, et al.: Investigation on Sharing Signatures of Suspected Malware Files Using Blockchain Technology., International MultiConference of Engineers and Computer Scientists (IMECS), pp.94-99, 2019.

23) 藤竜成，臼崎翔太郎，油田健太郎，山場久昭，片山徹郎，朴美娘，岡崎直宣, マルウェア検知システムにおけるブロックチェーンベースのマルウェア情報共有手法の検討, 電子情報通信学会技術研究報告, 119(140)，pp.293-298, 2019.

24) R.Fuji, et al.: Blockchain-Based Malware Detection Method Using Shared Signatures of Suspected Malware Files., International Conference on Network-Based Information Systems, pp.305-316, 2019.

25) https://www.advanced-intel.com/post/top-tier-russian-hacking-collective-claims-breaches-of-three-major-anti-virus-companies (accessed 2020/01/19).

26) CJ Fung: Trust management for host-based collaborative intrusion detection, International Workshop on Distributed Systems: Operations and Management, pp.109-122, 2008.

27) Y Fan, Y Ye, and L Chen: Malicious sequential pattern mining for automatic malware detection., Expert Systems with Applications 52 pp.16-25, 2016.

28) https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors (accessed 2020/01/13).

29) https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger.html (accessed 2020/01/25).

30) G Meng, et al.: Collaborative security: A survey and taxonomy., ACM Computing Surveys (CSUR), Vol.48, pp.1-42, 2015.

31) Jean-Marc Seigneur and Adam Slagell: Collaborative Computer Security and Trust Management., IGI Global.

32) https://www.virustotal.com (accessed 2020/01/13).

33) AJ O'Donnell, VV Prakash: Applying collaborative anti-spam techniques to the anti-virus problem., Virus bulletin, 2006.

34) J Gu, et al.: Consortium Blockchain-Based Malware Detection in Mobile Devices., IEEE Access 6, pp.12118-12128, 2018.

35) R Graf, and R King: Neural network and blockchain based technique for cyber threat intelligence and situational awareness., 10th International Conference on Cyber Conflict (CyCon), 2018.

36) https://polyswarm.io/ (accessed 2020/01/13).

37) W Meng, et al.: When intrusion detection meets blockchain technology: a review., Ieee Access 6, pp.10179-10188, 2018.

38) W Li, et al.: Designing collaborative blockchained signature-based intrusion detection in IoT environments., Future Generation Computer Systems 96, pp.481-489, 2019.