様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　４ 年　 ６ 月 １３ 日現在

機関番号：１３９０３
研究種目：基盤研究(C)（一般）
研究期間：2019～2021
課題番号：１９Ｋ１１９６２

研究課題名（和文）仮想化実行基盤の遠隔認証によるIoT環境の高信頼化

研究課題名（英文）Highly Reliable IoT Environments by Remote Attestation of Virtualized Execution Infrastructure

研究代表者
斎藤　彰一（SAITO, Shoichi）

名古屋工業大学・工学（系）研究科（研究院）・教授

研究者番号：７０３０４１８６

交付決定額（研究期間全体）：（直接経費）　　3,400,000円

研究成果の概要（和文）：IoTデバイスの普及により、これらが悪用される場面も増えている。このような状況を改善するために、IoTデバイスの安全性向上が重要である。本研究では、IoTデバイスの課題である計算性能が低いことを克服するために、クラウド等の強力な計算機にセキュリティ機能を委託する方法について研究を行った。この結果、IoTデバイスに負荷をかけずにプログラムの実行状況を取得する方法と、外部委託する方法のそれぞれ一部について解決することができた。しかし、より軽量な方法についての引き続き検討が必要である。

研究成果の学術的意義や社会的意義
IoTデバイスはパソコンやスマートフォンのように高性能な計算能力を持たず、必要最低限の能力だけを持っている。このため、セキュリティ機能を実現する余裕がない。しかし、セキュリティ機能の需要が増している。これらの問題を共に解決する手法が求められている。本申請では、低負荷でIoTデバイスの実行状態を取得しつつ、高性能な計算機でセキュリティ機構を実現する手法の研究を行った。これらは相反する課題であり、この解決は、インターネットをより安全にすることに大きく貢献できる。

研究成果の概要（英文）：With the proliferation of IoT devices, these have become targets of attacks. It is essential to secure IoT devices. To overcome the low computational performance of IoT devices, I studied how to outsource security functions to powerful computing devices such as the cloud. As a result, I solved a low-load method of obtaining the execution status of the IoT device's program and a method of outsourcing the security mechanism, respectively, in part. However, I need to continue to study the realization of a more lightweight method.

研究分野：情報工学

キーワード：IoTセキュリティ　異常検知

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

IoT デバイスの利用が進み，さらに通信網も 5G へと進化し高速大容量・低遅延が実現することで，これまで以上にネットワークに接続されたデバイスが社会の隅々にまで広がると考えられる．さらに，エッジコンピューティングと言われる，ネットワークの末端（デバイス）に近い場所に設置された機器（本研究ではゲートウェイという）でのコンピューティングの検討も進められている．これらに対するセキュリティ向上が必要である．

エッジコンピューティングではデバイスのネットワーク的近傍において各種処理を行うことから，デバイスやゲートウェイ等の機器が信頼できることが必要である．しかし，これら機器が他社製等でブラックボックスの場合は詳細を完全に把握できないため，情報流出や攻撃を受ける可能性が高まる等のセキュリティの確保が問題となる．この状況は，5G における通信経路上でのエッジゲートウェイから家庭におけるホームゲートウェイまで広範囲の問題であると申請者は考える．さらに，デバイスの低コスト化は今後も続くと予想されることから，高性能プロセッサや大量のメモリを搭載できないデバイスが利用され続けると考える．このため，セキュリティのためのリソース量の削減は今後も重要な問題である．

以上より，限られたリソースしか持たず，信頼できないデバイスやゲートウェイで実行するプログラムに対する高度なセキュリティ確保が今後さらに重要となる．このため，この問題を解決する技術の構築が必要であり，この解決は社会への大きな貢献になる．

２．研究の目的

本研究では，デバイスを提供する事業者（開発製造業者）がクラウドからゲートウェイを経てデバイスの実行状態を把握して安全に実行できる基盤を構築し，背景で述べた問題点を解決する．このために，次の 4 点を有するセキュリティ基盤を構築する．

1) 信頼できない実行環境（ゲートウェイ）におけるプログラム内容の信頼性確保方法
2) 実行履歴に基づく信頼性を遠隔（他の計算機）から確認する方法
3) リソースが限られたデバイスでの本提案機構の実現方法
4) IoT デバイスを狙ったマルウェアの調査

本研究は，1) により個々の実行環境におけるプログラムコードの信頼性を確認し，2) により実行履歴に基づく信頼性を遠隔から確認できるようにすることで信頼の連鎖（トラストチェーン）を繋げる．つまり，自社や第三者監査によって信頼できるクラウドを信頼の起点とし，それぞれの接続機器同士の信頼を繋げることで全体における信頼を実現する．また，プログラム実行の信頼性は Control Flow Integrity(CFI)に代表される実行履歴に基づく方法と機械学習の組み合わせで確認する．さらに，3) により省リソースデバイスにも本方式を適用して IoT の隅々にまで信頼の連鎖を繋ぎ，End-to-end の信頼できる実行基盤を実現する．さらに，4) として IoT デバイスに対する攻撃の詳細を理解するために，実際のマルウェアの調査を行う．

３．研究の方法

目的で述べた 1) から 4) の項目毎の方法を示す．

1) 信頼できない実行環境におけるプログラム内容の信頼性確保方法

本方式の特徴は Intel SGX を用いたプログラムの保護と，2) で述べた実行履歴の確認の実施である．CPU レベルのセキュリティ機構である SGX を用いることで，他のプログラムからの攻撃を防ぎ安全な実行環境を構築する．さらに，このプログラムを開発業者がクラウドから提供することで，プログラムの初期状態の信頼性を確保する．

2) 実行履歴に基づく信頼性を遠隔から確認する方法

保護対象のデバイスの実行状況の保障に CFI を使用する．実行状態が正しいか否かの確認方法には，何も感染していない状況で正しい制御フローグラフ（Control Flow Graph：CFG）を作成し，これと実際の実行履歴との比較で行う．このために，プログラムの実行状態の収集とその実行状態が正しい状態を示していることの確認の 2 点が必要となる．実行状態の収集にはプログラムの実行記録をどの程度の粒度で収集するかによって精度が定まる．しかし，粒度が細かいほど収集のためのコストが多くなるため実行速度が遅くなる．粒度が細かい単位では Basic Block (BB) を単位とする方法がある．粒度を粗くすると，関数を単位とする方法や，より荒くシステムコールを単位とする方法もある．これらの方法において必要な精度と実行コストを考慮して適切な方法を選ぶ．

正常であるか否かの確認のための CFG は, 実行アドレスの遷移を用いて作成する. これには, 既存研究である C-FLAT[1]の成果を活用する. C-FLAT の方法は, 現在の実行アドレスと, 直前までの実行アドレスの履歴のハッシュ値を合わせて新しいハッシュ値を計算することで, 現在までの実行履歴を含んだユニークなハッシュ値を求めることができる. このハッシュ値は, 他の実行経路のアドレスを含んだ場合には違う値になることから, ハッシュ値によって実行履歴の同一性を確認できる. これを基に CFG を作成する. 作成した CFG を低コストで正常値と比較して正常判定を行う.

3) リソースが限られたデバイスでの本提案機構の実現方法

CPU の性能やメモリが限られた IoT デバイスで実行履歴の確認などを行う場合, IoT デバイスでの処理時間に対する影響を考慮しなければならない. 一般に, IoT デバイスの性能は, 当該製品にとって必要最低限の性能となっていることが多いと考えられる. そのため, セキュリティ機構を実現する余力はあまりない. ここで, 提案手法のようなセキュリティ機構を実現するには, 可能な限りの機構をデバイス外部で実現できるようにする方法が考えられる.

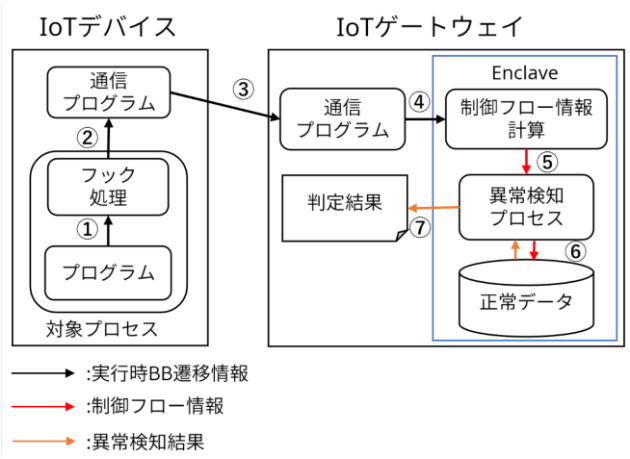
本研究では, デバイスで取得した実行履歴を外部のゲートウェイに送り, 1) および 2) の手法を活用して当該ゲートウェイにて CFG など用いて正常性確認を行う手法を実現する. また, ゲートウェイに実行履歴を送ることによって発生する負荷上昇を抑える方法についても検討する.

4) IoT デバイスを狙ったマルウェアの調査

提案手法の評価のために, 実際の IoT マルウェアについての調査を行った. 提案手法を実際に IoT デバイスに搭載するには, 様々な実装のための制限があると思われる. このような制限を理解することは提案手法を実デバイスに適用するために必要なことである. このために, 1) ハニーポットによるマルウェアの収集および, 2) IoT デバイスのファームウェアの調査を行う.

4. 研究成果

本研究の主な問題は提案手法 1), 2), 3) についてとなる. これらは関連しているために, 一体的に研究を行った. 提案手法のシステム概要を図に示す. IoT デバイスは実行履歴（図中の実行時 BB 遷移情報）取得のためのフック処理が追加された監視対象プロセスと通信プログラムで構成される. IoT デバイスでは, 対象プロセスの実行履歴情報の取得と, IoT ゲートウェイへの送信を行う. 初めに対象プロセスに対して追加するフック処理について述べる. 対象プロセスのフック処理①では 実行履歴情報の取得と解析, 通信プログラムへの出力②を行う. 実行履歴情報の取得は, BB 単位および関数単位での取得では C-FLAT の手法を用いて遷移元 BB の末尾アドレスと遷移先 BB の先頭アドレスを取得する. なお, システムコール単位での取得については後述する. 次に, 実行履歴情報の解析ではループの開始や終了の判定や, 制御フロー情報の計算コスト削減のための BB の取捨選択を行う. 取捨選択の基準は, 自明な履歴情報は捨てることである. 特に検討した点は, 条件分岐時にどちらの条件を経由したかの情報である. 条件分岐した場合にどの経路が実行された判定するには経路途中の 1 つの BB の実行が判定できれば良く, すべての BB を記録することは不要である. よって, 条件分岐後の実行経路判定に必要最小限の情報を残してそれ以外は捨てる. これらにより, BB 取得数を削減し, BB の各処理の負荷を軽減する. 最後に, 通信プログラムへ出力する. 次に通信プログラムでの動作について述べる. 通信プログラムではフック処理から受け取った実行履歴情報を IoT ゲートウェイへ送信する③. IoT ゲートウェイへの通信回数を抑えるため送信間隔や一度に送信する最大のデータ数を設定し, この条件を満たす場合のみ送信を行う. このとき送信間隔や一度に送信するデータ数は対象プロセスからの実行履歴情報の出力頻度によって適宜調整する.



図：提案手法のシステム概要

IoT ゲートウェイは, 通信プログラムと制御フロー情報計算プロセス, 異常検知プロセスで構成される. 各部の動作について述べる. 初めに通信プログラムでは IoT デバイスから実行履歴情報を取得③し, SGX Enclave 内部の制御フロー情報計算プロセスに受け渡す④. Enclave 内部への入力にかかる処理コストは大きいことからコストを削減するため, 実行履歴情報はまとめて Enclave 内部へ受け渡し, Enclave 内部で実行履歴を 1 つずつに分割する. 次に制御フロー情報の計算プロセスでは, 受け取った実行履歴情報から制御フロー情報を計算する. また制御フロー情報の計算に必要となる前ノードのハッシュ値は, Enclave 内の制御フロー情報の計算プロセスで保存する. 最後に異常検知プロセス⑤で計算結果と, 正常な制御フロー情報を比較⑥し,

異常の場合には判定結果を出力⑦する処理を行う．この時に用いる正常な制御フロー情報は，対象プロセスのプログラムを事前に解析し求めたプログラム全体の CFG から計算する．本研究では CFG 内の全パスを正常なフローとして正常データを求めたが，検知したいフローが存在する場合にはそのフローを異常として扱うために正常データから除外する．また本研究では異常検知の処理コストの削減のため，異常検知を行う制御フロー情報を限定し，異常検知処理の回数を削減する．本研究で用いる制御フロー情報はその時点までの実行フローを表すという特徴を持つ．このため，ある地点での制御フロー情報に対して異常検知を行うと，この地点に至るまでの実行フローの異常検知が可能である．この特徴を利用し本提案では一定間隔毎，もしくは異常検知の要所となる実行履歴に対してのみ異常検知を行い，処理コストを削減することができる．

本システムを実装し，各種評価を行った．まず，プログラムが異常な実行遷移を記録した場合には，正しく異常検知を行うことができた．これによって，最低限の機能の実現は達成できた．次に処理性能を計測した．結論としては十分な実行速度を得ることはできなかった．本システムの提案による BB の履歴情報削減によって，BB 処理コストを半減することはできたが，実処理においては 1 つの BB の実行に要するオーバーヘッドに 3〜4 us が必要であり，アプリの実行時間は数百倍に至った．さらに，IoT ゲートウェイにおける処理でも，SGX Enclave に情報を送る際の処理時間が大きい上にハッシュ値計算の処理時間も予想外に大きな負担となった．ハッシュ値計算を比較的軽量な MD5 を採用したが，十分な高速化は達成できなかった．

さらなる軽量を行うために，より粒度が荒いシステムコール単位での実行履歴情報の取得について研究を行った．図 1 の①に相当する処理の負荷軽減である．本研究は，Linux カーネルが標準で備える seccomp 機能を使用した．Seccomp を本研究で使用するために必要となる事項は，システムコール発行時点での実行履歴の取得である．一般に，システムコールを呼び出したユーザプログラム内の実行アドレスは取得されないため，本研究で新たに取得する必要がある．本研究では，呼び出したプロセスのコールスタックを辿ることで，ライブラリ関数を除いたユーザプログラム内の実行アドレスを取得し，実行履歴を得た．この結果，seccomp 適用時のシステムと比較して，1%から 5%程度の速度低下であることが判明した．このオーバーヘッドは，コールスタックを辿るために発生している．BB による履歴情報取得と比較して十分に軽量である．ただし，実行履歴の粒度は荒く，システムコールが発行されない条件分岐や関数では実行経路の確認ができない．このため，この条件下における安全性確保にはさらなる研究が必要である．

次に，IoT マルウェアの調査のために，IoT ハニーポットの構築および IoT ファームウェアの構造調査を行った．これらの研究では，Wi-fi ルータのファームウェアを仮想計算機上で実行し，クライアントを想定したアクセスを行う．そのアクセス結果から応答内容を機械学習する．インターネットからの接続には，機械学習した結果を応答する．このため，実デバイスや実際のファームウェアが直接攻撃を受けることはなく安全なハニーポットとなっている．さらに，実デバイスを購入する必要がないため，安価にハニーポットを構築できるシステムである．システムの開発は成功し，オープンソース[2]として公開している．本ハニーポットを大学で運用した結果，攻撃に対する反応を正しく行うことができた．しかし，新たな攻撃手法の発見やマルウェアの収集には至っていない．今後は，ハニーポットを大学以外に設置して評価を行う必要がある．

本申請による研究では，IoT デバイスのセキュリティ機構を外部で実現することで低負荷なセキュリティ機構を目指した．しかし，実行履歴の粒度と取得コストの相反する関係を十分に解決することはできなかった．この問題を解決するために継続して研究を進める必要がある．

[1] Abera, Tigist, et al. "C-FLAT: control-flow attestation for embedded systems software." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.
[2] FirmPot, https://github.com/SaitoLab-Nitech/FirmPot

５．主な発表論文等

〔雑誌論文〕　計0件

〔学会発表〕　計5件（うち招待講演　0件／うち国際学会　1件）

１．発表者名 山本 萌花，掛井 将平，齋藤 彰一
２．発表標題 ファームウェアの挙動を事前収集するIoTハニーポットの提案および基礎調査
３．学会等名 情報処理学会 コンピュータセキュリティシンポジウム 2020
４．発表年 2020年

１．発表者名 西村 賢太，山本 萌花，掛井 将平，瀧本 栄二，毛利 公一，齋藤 彰一
２．発表標題 IoTゲートウェイで動作するコンテナの異常検知手法の提案
３．学会等名 コンピュータセキュリティシンポジウム2019
４．発表年 2019年

１．発表者名 吉野 貴史，掛井 将平，瀧本 栄二，毛利 公一，齋藤 彰一
２．発表標題 IoTデバイス向けの制御フローベース遠隔認証手法の軽量化の検討
３．学会等名 情報処理学会　コンピュータセキュリティ研究会
４．発表年 2020年

１．発表者名 山本萌花，掛井将平，齋藤彰一
２．発表標題 IoT機器のWebUIを模したハニーポットの自動生成フレームワーク
３．学会等名 情報処理学会　コンピュータセキュリティ研究会
４．発表年 2021年

１．発表者名 Moeka Yamamoto, Shohei Kakei, and Shoichi Saito
２．発表標題 FirmPot: A Framework for Intelligent-Interaction Honeypots Using Firmware of IoT Devices
３．学会等名 The 8th International Workshop on Information and Communication Security (WICS 2021)（国際学会）
４．発表年 2021年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関