様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成　25 年　5 月　16 日現在

機関番号：12601
研究種目：基盤研究(B)
研究期間：2010　～　2012
課題番号：22300014
研究課題名（和文）　超セキュアプロセッサの研究

研究課題名（英文）Study of an Ultra Secure Processor

研究代表者
坂井　修一（SAKAI SHUICHI）
東京大学・大学院情報理工学系研究科・教授
研究者番号：50291290

研究成果の概要（和文）：　マイクロプロセッサのセキュリティを飛躍的に向上させる手法について提案し、試作・シミュレーションなどによって評価した。具体的には、(1)アタック防止技術として、インフォメーションフロー追跡の新手法であるＳＷＩＦＴ、(2)ソフトウェアタンパを防止する認証機構、(3)マイクロプロセッサの統合的セキュリティ管理手法、を提案し、評価した。成果は国際会議・査読付きシンポジウムなどで発表され、一部は汎用ソフトウェアにアドオンされた。

研究成果の概要（英文）：　We have proposed methods to dramatically improve security of microprocessors, and have evaluated them by simulations and prototyping. Actually, (1) SWIFT, an innovative method for tracking information flow, (2) authentication methods for preventing software tampering, (3) integrated security management of a microprocessor have been proposed and evaluated. The results were presented in international conferences and refereed symposiums. Some of the research products are embedded into general purpose software.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2010 年度	6,800,000	2,040,000	8,840,000
2011 年度	5,100,000	1,530,000	6,630,000
2012 年度	2,100,000	630,000	2,730,000
総　計	14,000,000	4,200,000	18,200,000

研究分野：　情報学
科研費の分科・細目：　計算基盤・情報セキュリティ
キーワード：情報セキュリティ、コンピュータウィルス、侵入検知、認証、マイクロプロセッサ、アーキテクチャ、オペレーティングシステム、ハイパバイザ

１．研究開始当初の背景

第３期科学技術基本計画の中では、第２期に引き続いて「安心・安全で質の高い生活のできる国」が最重要の基本理念の１つとして据えられている。今日、この理念を実現するためには、情報インフラが「安心・安全」であることが必須である。このことは、日本だけでなく地球規模で日々その重要性を増している。

情報技術の最も基本にあるのがコンピュータ技術と通信網の技術であり、端的にそれはマイクロプロセッサとインターネットの技術である。すなわち、「安心・安全なマイクロプロセッサとインターネットを構築すること」が、今の情報技術に最大の課題といってよい。ここでは、このうち前者の「安心・安全なマイクロプロセッサ」の構築をめざし、アーキテクチャとソフトウェア技術を中心

に研究する。

具体的には「安心・安全」は、安全性・可用性・堅牢性などの複合したものである。これらの性質をまとめて、ここではセキュリティと総称する。過去においてもマイクロプロセッサのセキュリティは重要なテーマであり、暗号技術や耐タンパ技術によってその向上が図られてきたが、(1)ソフトウェアの複雑化とブラックボックス化、(2)ゲート規模の爆発的増加による LSI の複雑化、(3)インターネットの急激な普及による攻撃の多様化と社会的影響の増大、(4)通信やソフトウェアの信頼性におけるベストエフォート文化の浸透、の 4 点によって、近年になってさらに問題が複雑化・大規模化・多様化している。

このような現状を受けて、国内外の多くの研究機関で、マイクロプロセッサのセキュリティをアーキテクチャ技術・ソフトウェア技術によって向上させる試みがなされている。代表例を示せば、(1)タンパ耐性をもつアーキテクチャとしてスタンフォード大学の XoM、MIT の AEGIS、東芝 L-MSP、(2)アタック耐性をもつアーキテクチャとして MIT の DIFT、プリンストン大学の RIFLE、九州大学の実行監視方式、(3)インジェクション・アタックを防ぐ方式としてのスタンフォード大学の Raksha などがこれである。これらは、プログラムレベルやスレッドレベル、命令レベル、アーキテクチャレベルにおいてセキュリティを向上させる優れた要素技術を提案しているが、セキュリティ阻害要因を系統的に分析し、プロセッサの制御方式を含めて総合的・網羅的な方策を得るには至っていない。

２．研究の目的

本研究では、アタック耐性やタンパ耐性を飛躍的に高めた超セキュアプロセッサを提案し、これを検証することを目的とした。そのために、データ保護や命令動作の正当性検証などの新しい要素技術を提案した。さらに、これらを統合して効率的に動作させるための超セキュアプロセッサの全体アーキテクチャを検討・提案し、アーキテクチャおよびソフトウェアのテストベッドを試作して動作を実証することをめざした。最終的には産業界と協力して、サーバ用からユビキタス用まであらゆるマイクロプロセッサに本研究の成果が取り入れられることが目標となる。日本の産業界では現在、採算性が良いとは言えないマイクロプロセッサの分野において、近未来に新たに国際的ヘゲモニーを得ることに貢献することが期待される。

３．研究の方法

セキュリティ要素技術として、(1) 文字列ごとの情報フロー追跡手法 SWIFT、(2) メモリデータ暗号化や認証などによる耐タンパ技術のそれぞれの研究を行い、提案者が開発したソフトウェアシステムやシミュレータに新規機能を組み込んで評価し、さらに実装などによって実証的に検証する。さらに、これらを統合管理するセキュリティマネジャの機能・機構を明らかにし、超セキュアプロセッサの全体を設計する。超セキュアプロセッサについては、詳細シミュレータを作成するなどし、FPGA テストベッドによる実装などを行い、さらに必要に応じて VDEC などを使ったカスタム VLSI 実装を試みる。以上によって、スタンドアロンなプロセッサとしての超セキュアプロセッサの動作検証、達成されるセキュリティの検証などを行い、本研究成果の有効性を検証する。

４．研究成果

(1) アタック防止技術

近年、クロス・サイト・スクリプティングやSQL インジェクションといったWebアプリケーションの脆弱性を突いたインジェクション・アタックによる被害が深刻化している。インジェクション・アタックを検出する方法としてDTP(Dynamic Taint Propagation) が研究されている。DTP では、外部からの入力にテイント（汚染）をつけ、演算の入力から出力に伝播させ、最後にテイントのついたデータが’危険な使われ方をしないかチェックする。従来のDTP では、命令間のデータの依存関係に基づいてテイント情報を伝播していたため、検出漏れと誤検出のトレードオフに陥り、伝播精度が十分ではなかった。

そこで我々は、以前、load/store 命令のメモリアクセスから文字列操作を識別し、文字列から文字列へテイント情報を伝播させる SWIFT を提案した。SWIFT はこのようにローカルでない追跡を行うため、その伝播精度は従来のDTP より高いことが示されている。しかし、SWIFT はハードウェア上に実装するため普及のハードルが高い。

本研究では、SWIFT をWeb 用スクリプト言語として現在最も多く用いられているPHP 上に実装した（図１）。

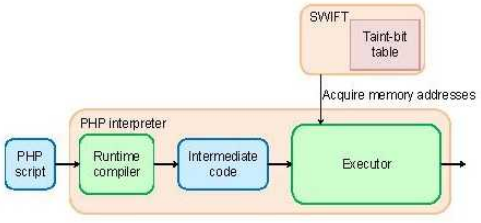

図１．SWIFTのPHP上の実装

表１．SWIFT評価環境

	PHP-SWIFT	PHP-taint	Raksha
Host OS	Windows Vista x64		
Emulator	VMware Player 3.1.4		Bochs 2.3.5
Guest OS	Ubuntu 10.04		Red Hat Linux 8.0
Web Server	Apache 2.2.14		Apache 2.2.11
SQL Server	MySQL 5.1.37		MySQL 4.1.22
PHP	modified PHP-5.3.1	PHP-taint 20080622 package	PHP-5.2.5

これを、表１のような環境で評価した。ここでは比較対象として、旧来の手法であるPHP-taintおよびRakshaを用いた。

表２．基本的な文字列操作と情報伝搬

Operation	PHP-SWIFT FN	PHP-SWIFT FP	PHP-taint FN	PHP-taint FP	Raksha−a FN	Raksha−a FP	Raksha+a FN	Raksha+a FP
(1) concatenation								
(2) *substr()*								
(3) *ereg_replace()*								
(4) *ereg()*								
(5) *strtoupper/tolower()*					√			
(6) *urlencdoe/decoder()*			√		√		√	
(7) *base64_encode/decode()*			√		√			
(8) untaint table								√
(9) taint table			√					√
(10) toupper (switch-statement)			√		√		√	

FN : false negative　FP : false positive

表２に、基本的な文字列操作に対するテイント情報の伝搬についての評価結果を示す。PHP-taint、Rakshaは、それぞれ検出漏れ、誤検出が２～４つの操作についてあったのに対して、我々の提案手法ではこれが皆無であった。

表３．ベンチマークを用いた評価

Program	Attack	PHP-SWIFT FN	PHP-SWIFT FP	PHP-taint FN	PHP-taint FP	Raksha−a FN	Raksha−a FP	Raksha+a FN	Raksha+a FP
phpSysInfo 2.3	Cross-site scripting								√
QwikiWiki 1.4.1	Directory traversal			√					√
phpBB 2.0.8	Cross-site scripting			√					√
PHP-Nuke 7.5	SQL injection			√					√
CubeCart 3.0.3	Cross-site scripting			√		√			√
PHP-Nuke 7.1	Cross-site scripting			√		√			√
PHP-Nuke 7.1	SQL injection			√		√			√

FN false negative　FP false positive

表３にじっさいのプログラムを用いた評価を示す。ここでは、典型的な文字列操作を行うプログラム、および脆弱性の確認されているWeb アプリケーションをベンチマークとした。これらについて、従来手法では存在した検出漏れ、誤検出が、SWIFTを組み込んだPHPでは全く無くなり、テイント情報が完全に正しく伝搬されていることが確認された。

表４．実行時間

ŁProgram	ŁAttack	Unmodified PHP(ms)	PHP-SWIFT(ms)	Overhead
phpSysInfo 2.3	Cross-site scripting	1.10	1.16	6%
QwikiWiki 1.4.1	Directory traversal	9.25	18.08	95%
phpBB 2.0.8	Cross-site scripting	13.96	23.99	72%
PHP-Nuke 7.5	SQL injection	20.58	27.73	35%
CubeCart 3.0.3	Cross-site scripting	45.59	84.05	84%
PHP-Nuke 7.1	Cross-site scripting	25.27	38.66	53%
PHP-Nuke 7.1	SQL injection	21.31	29.04	36%

表４は、SWIFTを組み込むことによってPHPの実行時間がどれぐらい長くなるかを調べたものである。時間オーバヘッドは、６%から95%までであり、その平均は55%であった。

(2) 耐タンパ技術

信頼できないOS の上で、アプリケーションが扱う情報が意図せず漏洩することを防ぐためには、以下の3点が保証されている必要がある。①起動しているアプリケーションが正規のものであること、②主記憶上の実行中のアプリケーションのデータが漏洩しないこと、③二次記憶に保存されたアプリケーションのデータが漏洩しないこと、である。従来の情報漏洩を防止するための手法として、TPM を用いたTrusted Boot やセキュアプロセッサ等が存在するが、それらは3 点全てを保証できているわけではない。

そこで本研究では、信頼できないOS の上でアプリケーション認証を行い、この3 点全てを保証するシステムを提案した。

本手法では、暗号化保存されたアプリケーションをハードウェアによる強制アクセス制御の行われた領域で起動し、強制アクセス制御の下でアプリケーションに対して認証を行うことで、情報の漏洩を防止する。

① 実行中のアプリケーションの保護

最初に、実行中のアプリケーションの情報が漏洩することを防ぐため、TLB とDMA の機能を拡張したSecureMMU を導入し強制アクセス制御を行う。また、強制アクセス制御により、特権プロセスの権限が制限され生じる問題に対し対策を行う。

通常のプロセッサのメモリ保護機能では、ページやセグメントを所有しているプロセスが決まっており、所有しているプロセス以外のプロセスがアクセスしようとするとメモリ保護違反となる。しかし、OSなどの特権プロセスは、すべてのメモリにアクセスすることができてしまう。そこで、図２ に示すように特権プロセスの権限の及ばないユーザープロセスのメモリ領域を確保し、特権プロセスの権限を制限するため、メモリ保護の拡張を行う。

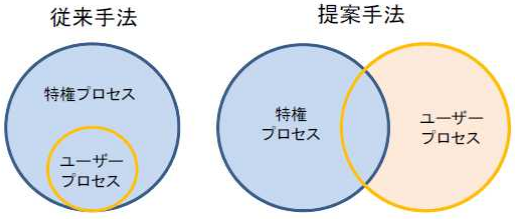

図２．特権プロセスからのメモリ領域保護

また、ページテーブルエントリに含まれる所有者によるアクセス権限の読み込み(RD)/書き込み(WR)/実行(EX) に加えて、特権プロ

セスによるアクセス権限のRD/WR/EX ビットを用意することにより、特権プロセスのアクセス権限をOSによらず、強制的に制御することができるようにする。

本手法では、OS の信頼性によらないアプリケーションの認証、及び実行を行う。そのため、OS がデータを扱う際にすり替え等の攻撃が行われないよう対策を講じる。

(a) ページテーブルエントリのすり替え防止

ページテーブルはOS が管理するが、これがすり替えられてしまうとアクセス権限が無効化されたり、ページをすり替えられたりする危険性がある。そこで、ページテーブルエントリにもハッシュ値を用いた認証を行う。TLBからページテーブルエントリが追い出されるとき。ページテーブルエントリとプロセスIDからハッシュ値を生成し、復元時にハッシュ値の照合を行うことでOS によるページテーブルエントリのすり替えを防止する。

(b) 入出力に関するすり替え防止

入出力を行う際、OS が様々な処理を行う。この際、アプリケーションの扱う情報がOS に漏洩しないよう対策を講じる必要がある。

図３．スワップ発生時のデータすり替え防止

図３ にSecureMMU を用いたスワップ時のデータのすり替え防止の流れを示す。DMA に暗号化・復号化機能を追加し、スワップアウト時にデータの暗号化を行い、暗号化したデータのハッシュ値を記録する。スワップイン時にハッシュ値の比較を行うことですり替えを防止する。また、スワップアウト時にOS はアドレスマッピングを行うが、この際、不正なプロセスによるすり替えや解析が行われる危険性が生じる。これを防ぐため本手法においては逆引きページテーブルを用いる。すなわち、ひとつの物理アドレスに複数のプロセスの仮想アドレスがマッピングされていないかページフォールトが起こるたびにSecureMMU がチェックを行う。

(c) DMA 転送後のページテーブルの書き換え防止

DMA 転送が発生した際はページテーブルを書き換える必要がある。しかし、ページテーブルの書き換えをOS が行った場合、ページテーブルの改竄が行われる危険性がある。

これを防ぐため、 DMA転送後のページテーブルの書き換えはOS を介さずにSecureMMUが行う。

(d) コンテキストスイッチ

コンテキストスイッチに伴うレジスタ退避/復元をOS が行うと、レジスタ値をすり替えられたり、メモリ上のデータに他プロセスからアクセスされる危険性がある。そのため本手法では、各プロセスにレジスタ退避/復元用の機構を実装し、レジスタ退避/復元をプロセス自身に行わせる。

② アプリケーションの認証

非正規のアプリケーションが正規のアプリケーションに成りすますことを防止するため、SecureMMU の機能を利用し、特権プロセスの権限の制限された領域でアプリケーションを起動し、アプリケーションのハッシュ値をSecureMMU が直接取得し認証を行う。認証完了後、アプリケーションは引き続き特権プロセスの権限の制限されたメモリ領域で実行される。従って、アプリケーションが終了するまでOS からの影響は受けない。

図４．情報漏洩を実現するプロセッサの認証と実行の流れ

本手法の認証と実行の流れを図４に示す。図に示すようにアプリケーションのハッシュ値に電子署名・暗号化を行った上で認証サーバーに送信し、認証サーバーはアプリケーションを利用する際必要となるトークンを返す。

(3) 統合技術

要素技術としてこの他に、効率化されたデータタグによるセキュリティ情報の伝搬方式などを行ってきた。さらに、これらを統合する高セキュリティプロセッサについて、ハードウェア機構とハイパバイザの機構を明らかにしている。

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計０件）

〔学会発表〕（計８件）

① 山田 剛史、 五島 正裕、 坂井 修一: 信頼できない OS 上でアプリケーション認証を行うシステム、電子情報通信学会技術報告 CPSY2012-11、 Vol. 112, No. 173 pp. 13-18 (2012).
② 山田 剛史、早川 薫、都井 紘、五島 正裕、坂井 修一:情報漏洩防止プロセッサ、情報処理学会 第74回全国大会 pp. 1-65-1-66(2012).
③ 坂井 修一: 安全・安心社会と電子情報技術 ～ 何をどうやって守るか～、電子情報通信学会 CPSY2011-42, pp. 1-2 (2011).
④ 早川 薫、 塩谷 亮太、 五島 正裕、 坂井 修一:プラットフォーム部分認証、電子情報通信学会技術報告 CPSY2011-12 , pp. 19-24 (2011).
⑤ Hiroshi TOI, Ryota SHIOYA, Masahiro GOSHIMA, and Shuichi SAKAI: Yet Another Taint Mode for PHP,先進的計算基盤システムシンポジウム SACSIS2011, Vol. 2011, pp. 160-169 (2011).
⑥ 早川 薫、都井 紘、塩谷 亮太、五島 正裕、 坂井 修一:プラットフォーム遠隔認証、情報処理学会 第73回全国大会、pp. 3-559-3-560 (2011).
⑦ 都井 紘、 塩谷 亮太、 五島 正裕、 坂井 修一:文字列ごとの情報フロー追跡手法の PHP への実装と評価、情報処理学会研究報告 2010-OS-115, No. 4, pp. 1-11 (2010).
⑧ Hiroshi Toi, Ryota Shioya, Masahiro Goshima, and Shuichi Sakai: Yet Another Taint Mode for PHP,IEEE Int'l Symp. on Pacific Rim Dependable Computing (PRDC) (2010).

〔図書〕（計０件）

〔産業財産権〕
○出願状況（計０件）

○取得状況（計１件）
名 称 : INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, AND COMPUTER,
発明者：S. Katsunuma, M. Goshima, R. Shioya, H. Irie and S. Sakai:
権利者：M. Gosima and R. Shioya
種類：米国特許
番号：米国公開番号：US 2010/9983379A1)
取得年月日：2013 年 2 月 15 日
国内外の別：国外

〔その他〕
○受賞（計１件）
早川 薫（指導学生）:プラットフォーム遠隔認証、情報処理学会第 73 回全国大会、情報処理学会推奨卒業論文認定 (2011).

○ホームページ
http://www.mtl.t.u-tokyo.ac.jp/

６．研究組織
(1)研究代表者
坂井 修一（Shuichi Sakai）
東京大学・大学院情報理工学系研究科・教授
研究者番号：50291290

(2)研究分担者
五島 正裕（Masahiro Goshima）
東京大学・大学院情報理工学系研究科・准教授
研究者番号：90283639