ソフトウェアを保護するトラステッドコンピューティング

Trusted Computing for Software System Protection

磯崎　宏
■ ISOZAKI Hiroshi

個人情報や企業情報などの機密情報を扱うシステムは，設計者の意図どおりに動作することが求められる。その解決手段として，ハードウェアの安全性に依存してソフトウェアが不正な動作を行わないようなプラットフォームを実現する，トラステッドコンピューティング（Trusted Computing）という概念が注目されつつある。

東芝は，今後パソコン（PC）だけでなく，コンシューマー機器にもソフトウェアを保護して実行する環境が必要だと考えており，標準化組織TCG（Trusted Computing Group）に参加し，そこで標準化されたセキュリティチップやストレージデバイスを用いることで，各種デジタル機器に対する安全性をよりいっそう向上させることを目指している。

A system that contains privacy-related data or confidential corporate data is required to behave in accordance with the intention of the system designers. In recent years, the concept of trusted computing has been attracting considerable interest as a solution for this. Trusted computing provides a computing platform with robust hardware to ensure that software behavior is not compromised.

Toshiba believes that this concept will be introduced to personal computers and consumer electronics devices in the near future. We are aiming at further enhancing security for digital devices by using security chips or storage devices standardized by the Trusted Computing Group (TCG).

1　まえがき

従来のコンピュータセキュリティは，主として暗号アルゴリズムや暗号プロトコルといった暗号技術によって支えられてきた。暗号技術によってシステムの安全性を理論的に保証することができるが，その前提として，アルゴリズムやプロトコルが設計者の意図どおりに動作する必要がある。現在多くのシステムはソフトウェアによって構築されているが，ソフトウェアが不正に解析されたり，ウィルスやワームなどによって意図せずに改変されたりする被害が後を絶たず，セキュリティシステムもその例外ではない。

これらの問題点に対処するため，トラステッドコンピューティング（Trusted Computing）という概念が注目されつつある。トラステッドコンピューティングでは，ソフトウェアが意図どおりに動作することをある一定のセキュリティレベルで保証するようなプラットフォーム（以下，トラステッドプラットフォームと呼ぶ）の構築を目指している。

ここでは，トラステッドプラットフォームを実現するための標準化組織TCG（Trusted Computing Group）を紹介し，TCGで規格化されている技術と応用例について述べる。

2　TCGの概要

TCGは，ハードウェアの安全性に依存したコンピュータプラットフォームを提供するためのハードウェアモジュールや，そのモジュールにアクセスするためのソフトウェアインタフェースの標準化と普及を目的とした非営利団体である。PCベンダーや，HDD（ハードディスク装置）ベンダー，チップベンダーなど約140社が加盟している。

わが国では，TCGの国内における活動の支援と普及を目的としたJRF（Japan Regional Forum）と，（社）電子情報技術産業協会（JEITA）のTCG専門委員会が主に活動している。

TCGには，セキュリティチップの仕様を決めるTrusted Platform Module（TPM）技術部会やストレージ技術部会など，利用シーンに合わせて11の技術部会があり，それぞれの部会で技術仕様を策定している。

3　TCGで規定されているハードウェアモジュール

ここでは，TPMとトラステッドストレージ（Trusted Storage）それぞれの仕様について述べる。

3.1　TPM

TPMとは，ハードウェアによる暗号機能を備えたセキュリティチップである。東芝製ノートPCのRXシリーズをはじめとして，数多くのビジネス向けPCに搭載されている。2009年2月時点でVersion 1.2仕様が公開されており，以下に示すような様々な特徴を備えている[(1)]（**図1**）。

(1)　乱数生成や，セキュアハッシュ計算，暗号鍵生成，RSA（Rivest-Shamir-Adleman）暗号といった暗号処理をチップ内部で処理する暗号処理機能

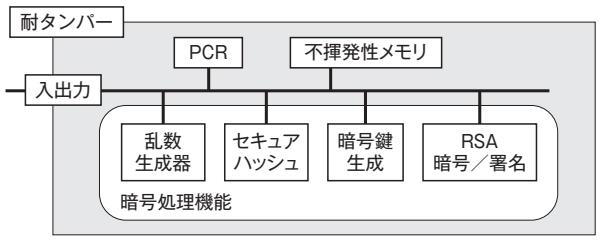

図1. TPMの機能ブロック図 ― TPMにはハードウェアとして暗号処理機能が実装されている。

Functional block diagram of Trusted Platform Module (TPM)

(2) プログラムコードやデータなどある一時点のメモリの状態を保存する，PCR (Platform Configuration Register) と呼ばれるレジスタ

入力データ (data) は入力前のPCRの値 (PCR_{i-1}) と結合されSecure Hash Algorithm (SHA-1) のハッシュ値が新しいPCRの値 (PCR_i = SHA-1 (PCR_{i-1} || data)) となる。PCRに直接任意の値を設定したり，任意のタイミングでリセットしたりすることはできない。

(3) チップに固有のID (Identification) や暗号鍵を保存するための不揮発性メモリ

(4) 現在のPCRの値に対してチップの秘密鍵で署名を生成する署名生成機能

(5) 保護対象の値をPCRの値とともにチップ固有の秘密鍵で暗号化することで，特定の状態のときにしか復号できないようにするシールドストレージ (Sealed Storage) 機能

(6) PCRや不揮発性メモリの値を外部から解析されないようにするハードウェアレベルの耐タンパー機能

TPMでは，仕様で規定されているコマンド，すなわちインタフェース以外の経路でTPM内のデータや処理内容を操作することはできず，またこれらの機能は，ホストCPU上で動作するソフトウェアとは独立に動作する。つまり，ソフトウェアがTPMの処理内容を不正に変更することはできない。前述のように，TCGではソフトウェアレベルの攻撃からシステムを保護するために，ハードウェアをトラステッドプラットフォームにおける信頼の基礎 (Root of Trust) としているが，このTPMを信頼の基礎として利用することができる。

3.2 トラステッドストレージ

トラステッドストレージは，SSD (Solid State Drive) やHDDなどのストレージデバイスにハードウェアレベルでのセキュリティ機能を持たせることで，プラットフォームの安全性を強化するアプローチである。コンシューマー向けストレージデバイスの仕様書では，以下のような機能を定義している[2), (3)]。

(1) パスワード認証によるメディアのロック機能　　パスワードによってストレージデバイスへの読込み又は書込みをロックする。

(2) HDD暗号化機能　　暗号鍵を設定してストレージデバイスのデータを暗号化する。データを暗号化する鍵はストレージデバイスに隠ぺいされ，外部のソフトウェアがその値を参照することはできない。

ストレージデバイスそのものがこれらの機能を持つことで，ソフトウェアによる解決手段と比較して以下のような利点がある。

(1) 暗号鍵の漏えいリスクが低い　　一般的にハードウェアはソフトウェアに比べて解析が困難であり，暗号鍵の漏えいリスクを抑えることができる。

(2) OSなどの環境に非依存　　OS (基本ソフトウェア) やアプリケーションによるファイル暗号化は異種環境下で使うことはできないが，そのような問題がない。

(3) CPUにオーバヘッドが掛からない　　暗号化と復号はストレージデバイス内で行われるため，CPUにオーバヘッドは掛からない。

4 TPMとトラステッドストレージの応用例

ここでは，TPMの従来の応用例として，データの保護及びシステム状態の検証について述べる。更に，当社も一部研究に参加した，TPMを適用したソフトウェアの安全な実行環境の構築についても述べる。

4.1 従来の応用例

4.1.1 データの保護　　TPMのもっとも単純な用途はデータの保護である。前述のとおり，TPMには外部からアクセス不可能な暗号鍵を保持する機能がある。例えば，パスワードなどのデータをTPMに固有の鍵で暗号化することで保護して蓄積することができる。

TPMは通常マザーボードに搭載されているため，パスワードをある固有のPCでしか復号できないように暗号化することもできる。また，トラステッドストレージと組み合わせ，TPMの中に保護対象のデータを保存することで，特定のシステムでしかストレージデバイスにアクセスできないように制限することもできる。

4.1.2 システム状態の検証　　TPMはシステムの状態の検証に利用することもできる[4)]。一般に，PCシステムが起動する際，初期命令 (INIT)，BIOS (Basic Input Output System)，ブートローダ，OS，アプリケーションの順でプログラムが実行される。そこで，**図2**に示すように，初期命令から順々に，次に実行するプログラムのハッシュ値をTPMのPCRに格納していくようプログラムを構成する。PCRにはこれまで実行されたプログラムの積算値が蓄積されることになるので，実行されるべきプログラムとその実行順序がわかっていれば，PCRに格納される予測値をあらかじめ計算しておくことができる。実行中のプログラムは予測値と現在のPCRの値が一致するかどうかを確認することにより，システムが起動されてからどのよ

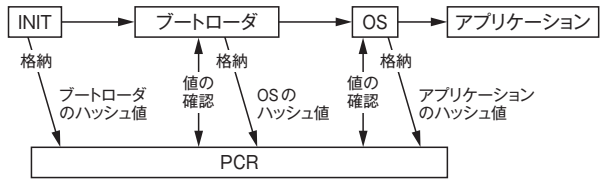

図2. トラステッドブートの概念 — トラステッドブートでは実行したプログラムのハッシュ値がTPMに積算されていく。

Concept of trusted boot

うなプログラムが実行されたかを知ることができる。システム起動時以外にPCRの値をリセットすることができず，任意の値を設定することもできないので，途中のプログラムが不正に改変された場合にはPCRが予測値と異なる値となり，プラットフォームが改変されたことを検出できる。更に，TPMの署名生成機能を使うことで，外部ホストからもPCRの値を署名によって検証し，システムが安全に起動したかどうかを確認することができる。

このように，システム設計者の意図どおりにシステムを起動させる仕組みをトラステッドブートと呼ぶ。トラステッドブートは，CD-ROMやネットワーク経由でOSをダウンロードして起動するような場合に，利用者の意図したシステムで起動したかどうかを確認するといった用途で利用することができる[5]。

4.2 TPMを適用したソフトウェアの安全な実行環境の構築

トラステッドブートでは，検証者が被検証システムの構成を事前に把握しておく必要があり，検証者の負担が大きい。また，検証者が検証するときのPCRの値と，実行時点でのPCRの値にはタイムラグがある。このため，PCRの値を取得した時点からシステムが攻撃を受けると，検証した値と現在のシステムの状態が必ずしも一致するとは限らないという問題もある。より本質的な問題として，トラステッドブートは信頼すべき対象が大きいという欠点がある。つまり，検証対象のプログラムより前に実行したプログラムすべての安全性に依存してしまう。BIOSやOSなどに一つでもバグが存在すると，トラストチェーン（信頼の連鎖）が切れてしまい，それ以降に実行するプログラムの安全性を保障することができないという問題がある。

このように，トラステッドブートは限定された環境においては有用であるものの，OSやアプリケーションが頻繁にバージョンアップするような汎用のPCシステムに適用することは困難である。また，昨今のシステムでは，OSやライブラリにオープンソースなどの自社製ではないソフトウェアを利用する場合が多いが，トラステッドブートでは，それら自社製以外のソフトウェアも信頼の対象に含める必要がある。しかし，理想的には信頼すべき対象ができるだけ小さく，かつ保護対象のプログラムに限定して実行したかどうかを検証できるようなプラットフォームであることが望ましい。

この問題を解決するための一つのアプローチとして，カーネギーメロン大学Adrian Perrig准教授の研究プロジェクトでは，TPMを使って保護対象のアプリケーションだけを安全に実行するためのプラットフォームに関する研究を行っている[6]。当社は，2007年1月から2008年6月までこのプロジェクトに参加し，Intel® (注1) アーキテクチャのCPUを使って，以下のような特徴を持つプラットフォームの実装に携わった。

(1) BIOSやOSを信頼の対象から外すことができるため，信頼すべき対象は非常に小さい。
(2) ほかのソフトウェアやハードウェアにじゃまされることなく，保護対象のアプリケーションを独立して実行する環境を整えることにより，アプリケーションを実装者の意図どおりに実行することができる。
(3) アプリケーションが実行されたことをPCRの値を使って他者が検証することができる。
(4) 従来のLinuxプラットフォームを利用することができる。

プラットフォームの実装例を**図3**に示す。この実装ではTPM Version 1.2を利用する。TPM Version 1.2では，CPUの特殊な命令（SENTER命令）を利用することで，再起動することなくPCRの値をリセットすることが可能となった。SENTER命令は，DMA（Direct Memory Access）転送や割込み，デバッガからのアクセスを無効化し，そのプラットフォームが動作するプログラムが外部からの干渉を受けないようなプログラムの実行環境を整える[7]。その後，PCRの値をリセットして指定された範囲のメモリの内容をPCRに格納し，指定されたプログラムの実行を開始する。TPMとSENTER命令を

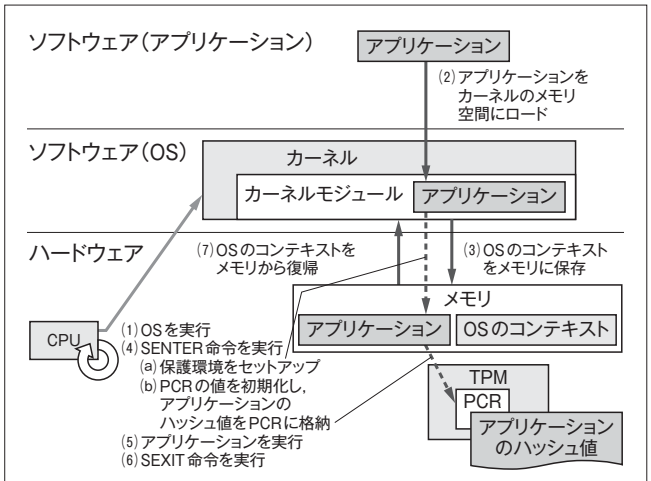

図3. TPMを適用したソフトウェアの安全な実行を実現するプラットフォームの構成 — OSを一時停止させて，ソフトウェアを保護環境下で実行させる。

Structure of trusted platform implementing TPM

（注1） Intelは，米国又はその他の国における米国Intel Corporation又は子会社の登録商標又は商標。

使い，以下のような手順で処理するシステムをLinux上で実装した。

(1) 通常の手順でOSを起動する。
(2) アプリケーションをカーネルが管理するメモリ空間にロードする。
(3) OSのコンテキストをメモリに保存する。
(4) SENTER命令を実行する。
  (a) OSの処理は一時停止し，SENTER命令の中で保護環境がセットアップされる。
  (b) PCRの値が初期化され，メモリ上に展開されたアプリケーションのハッシュ値がPCRに格納される。
(5) 保護対象のアプリケーションを実行する。
(6) アプリケーションの中でSEXIT命令を実行し，保護環境から離脱する。
(7) メモリ上に保存されたOSコンテキストを復帰させ，OSの処理を再開する。

この一連の処理により，保護対象のアプリケーションはほかのプログラムからの干渉を受けることなく，独立して実行することができる。また，PCRの値を確認することで，アプリケーションが実行されたことも検証できる。更に，通常のLinuxアプリケーションと保護対象のアプリケーションを，一つのLinuxシステムで共存させることが可能になる。

5 コンシューマー機器への展開

PC環境ではこれまで，共通して利用可能な信頼の基盤となるモジュールが存在しなかったため，TPMやトラステッドストレージは，トラステッドコンピューティングを構成するハードウェアモジュールとして普及していくと予想される。一方，コンシューマー機器では内部仕様は一般に公開されず，また，汎用ではない部品を使っていることもあり，PCと比較して攻撃に対するリスクが低かったため，これまではハードウェアモジュールの必要性は低かった。しかし，近年，デジタルテレビやハードディスク&DVDレコーダなどコンシューマー機器でも，多機能化の要求によりシステムが複雑化し，システムに占めるソフトウェアの割合も増加する傾向にあり，内部的にはPCシステムに近い構成となりつつある。更に，個人の嗜好（しこう）情報などのプライベートデータや，エンターテインメントコンテンツの保護に使われるDRM（Digital Right Management）など，保護すべきデータやプログラムを扱う機会も多くなってきている。

例えば，コンシューマー機器でも，バグ修正などを目的としたシステムのアップデートをオンラインで実行することが一般的になりつつあるが，モジュールに前述のような機密情報を含む場合もあるため，アップデートモジュールを保護して配布したり，アップデートを確実に実行したりする必要がある。そこで，3.2節で示した手法を適用して解決することができる。むしろコンシューマー機器のほうがPCに比べシステムの自由度が低い分，適用しやすいとも言える。

また，DRMシステムでは，コンテンツを保護するために様々な暗号処理を行うが，それらの処理の一部に3.2節で示した手法を用いて安全な環境で実行したり，トラステッドストレージで暗号化してコンテンツを蓄積したり，特定のユーザーの機器や特定のアプリケーションでしかコンテンツにアクセスできないような仕組みを構築したりすることもできる。

このように，今後はコンシューマー機器でトラステッドプラットフォームの概念を導入する機会が増えていくものと考えられる。

6 あとがき

TPMやストレージデバイスなどTCGでの標準化により，トラステッドプラットフォームを実現するためのハードウェアコンポーネントが徐々に普及しつつある。単なるセキュアストレージだけでなく，ソフトウェアの保護方式としても利用することができる。トラステッドプラットフォームの概念がPCだけでなく，今後はコンシューマー機器にも適用されていくことを予想し，プラットフォームやソフトウェアを保護するアーキテクチャの実現を目指した研究開発を進めていきたい。

文 献

(1) TCG. TPM Specification Version 1.2 Revision 103: Part 1 - 3. <http://www.trustedcomputinggroup.org/developers/trusted_platform_module/specifications>, (accessed 2009-05-18).
(2) TCG. Work Group Storage Architecture Core Specification—Version 1.0, Revision 0.9. <http://www.trustedcomputinggroup.org/developers/storage/specifications>, (accessed 2009-05-18).
(3) TCG. Work Group Storage Security Subsystems Class: Opal—Version 1.0, Revision 1.0. <http://www.trustedcomputinggroup.org/developers/storage/specifications>, (accessed 2009-05-18).
(4) Sailer, R., et al. "Design and implementation of a TCG-based integrity measurement architecture". 13th USENIX Security Symposium. San Diego, CA, USA, 2004-09, USENIX Association. 2004, p.223 - 238.
(5) Suzaki, K., et al. "Trusted Boot of HTTP-FUSE KNOPPIX". Linux-Kongress 2006. Nurnberg, Germany, 2006-09, German Unix User Group.
(6) McCune, J. M., et al. "Flicker: An Execution Infrastructure for TCB Minimization". The European Conference on Computer Systems (EuroSys) 2008. Glasgow, Scotland, 2008-04, EuroSys. p.315 - 328.
(7) Grawrock, D. The Intel Safer Computing Initiative. Intel Press, 2006, 295p.

磯崎　宏　ISOZAKI Hiroshi
研究開発センター　コンピュータ・ネットワークラボラトリー
研究主務。ホームネットワーク及びセキュリティ技術に関する研究・開発に従事。
Computer & Network Systems Lab.