# USB トークン認証を用いた OS の安全な起動制御

高田真吾[†] 佐藤聡[‡] 新城靖[‡,¶] 中井央[§] 板野肯三[‡,¶]

筑波大学情報学類[†] 筑波大学システム情報工学研究科[‡] 筑波大学図書館情報メディア研究科[§] 科学技術振興機構[¶]

## 1 はじめに

ノート PC を持ち歩くことで, どこでも自分専用の計算機環境を利用できることは大変便利である. しかし, 近年では重要な情報が格納されているコンピュータを持ち出している際の盗難による, 情報の漏洩を防ぐということが重要な課題になっている. 組織においてコンピュータの持ち出し禁止などの対策を行うと, 自分専用の環境をどこに行っても利用できるということが困難になってしまう.

ハードウェアを持ち歩かずに自分の環境を使う方法として, 目前にあるハードウェア上で起動している OS にログインし, リモートデスクトップ機能を用いて自分専用の計算機環境を遠隔から使うという方法がある. しかし, この方法では通信遅延による操作性の悪化などの問題が発生する.

一方, リムーバブルメディアに OS を格納し, そのメディアを利用者の目前にあるハードウェアに接続し, 格納された OS を起動させることにより, 操作性の悪化もなく自分用の環境を利用する方法もある. しかし, 利用者によるリムーバブルメディアによる OS 起動を許すと, 管理者はハードウェア上で起動している OS の管理ができなくなる.

本研究では, 認可された利用者だけがその人用に特化された OS を起動でき, かつ同時刻にその OS がネットワーク上で 1 つしか起動していないことを保証する仕組みを設計することを目的とする. またこの仕組みは, 起動される OS についてのユーザやハードウェア情報を用いた, 管理者による起動制御を行う.

## 2 提案方式

### 2.1 起動制御

本研究では,1 章で述べた目的を達成するために, 適切な物理的認証デバイスが挿入されることにより起動を行おうとしている利用者を特定し, その利用者用の OS を起動し, デバイスが抜かれた場合に OS を停止する仕組みを提案する.

提案方式では, OS の起動を制御するために, ハードウェアと OS の間に起動制御レイヤーを配置する. そして, そのレイヤーが USB トークンや IC カードなどの物理的認証デバイスの挿抜監視や認証処理も行う (図 1).

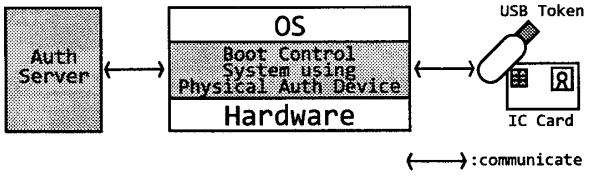


図 1 提案システム概要

提案方式ではハードウェアの電源を投入しただけでは OS を起動しない. 起動制御レイヤーは物理的認証デバイスが挿入されたことを検知すると, 適切な OS を起動する. また, 物理的認証デバイスが抜かれたことを検知すると, 起動した OS を停止させる. ハードウェアと利用者, 利用する OS との関連および OS イメージを管理するために認証サーバを用いる.

従来のコンピュータにおいて, ハードウェア上に搭載されている BIOS が OS の起動を行っている. したがって, 提案方式では本起動制御レイヤーが BIOS に替わって起動制御を行うものであり, 従来のコンピュータシステムとの親和性が高い. また, ハードウェアや OS に対する変更点が少ないという利点がある.

### 2.2 ユーザとハードウェアの認証

OS の起動制御を行う上では, どの利用者がどのハードウェア上で OS の起動を行おうとしているかをシステムが認識できなければならない. 利用者の認証は, 物理認証デバイスを用いて本人とシステムのユーザを結びつけることにより実現可能である. ハードウェアの認証は, そのハードウェアに固有な何らかの情報を用いて実現可能となる. 認証サーバは, 提案システム上のユーザを証明するユーザ用証明書と, システムが実行されるハードウェアの証明書をもとに認証処理を行う. これにより, 特定のユーザと特定のハードウェアについて柔軟な起動制御を行うことができる.

## 3 実装

提案方式の有効性を迅速に確認するために, 本研究では仮想計算機モニタ (VMM) におけるゲスト OS を制御対象の OS として扱い, ホスト OS でその制御を行うことで起動制御レイヤーの機能を実装する. また, 物理認証デバイスとしては USB トークン [3] を対象とする.

本実装では VMM として Xen[1] を用いる. Xen における Domain0 で起動制御レイヤーの主な機能を実装し, 利用者用 OS は Xen における DomainU として起動する.

---

Secure OS Boot Control using USB Token Authentication
Shingo TAKADA, Akira SATO, Yasushi SHINJO, Hisashi NAKAI, Kozo ITANO

† University of Tsukuba, College of Information Science
‡ University of Tsukuba, Department of Computer Science
§ University of Tsukuba, Graduate School of Library, Information and Media Studies
¶ Japan Science and Technology Agency

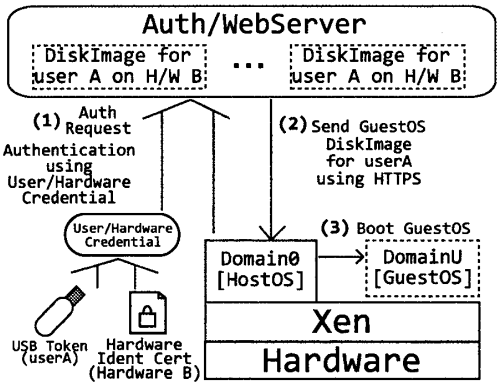


図2 システム起動の流れ

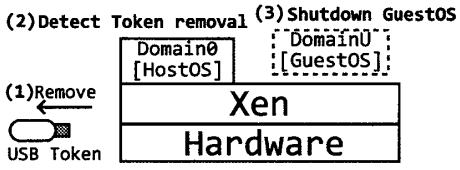


図3 USB トークンが抜かれた場合

### 3.1 利用者用 OS の制御方法の流れ

提案システムにおける利用者用 OS の起動の流れを図 2 に示す. 電源が投入されると, 本システムは, 利用者に対してまず USB トークンの挿入を要求する. USB トークンが挿入されたことを検知すると, 本システムは USB トークンから取得したユーザ名と, あらかじめ Domain0 内に保存されているハードウェア用証明書を用いて認証を伴う利用者用 OS イメージの取得処理を行う.

認証サーバからの利用者用 OS イメージの取得が完了すると, そのイメージを仮想マシン上で起動し, 利用者がその OS を利用できるようにする. また, 同時に USB トークンの抜取りを監視するデーモンを実行する.

利用者用 OS の稼働中に USB トークンが抜かれた場合, 監視デーモンがそれを検知し VMM のシャットダウン命令を利用して利用者用 OS を停止させる (図 3). これにより, USB トークンが挿さっていない場合には利用者用 OS は実行できないため, 利用者用 OS が複数の環境で同時に実行されることはなく, ユニーク性が保証される.

### 3.2 認証を伴う利用者用 OS イメージの取得方法

USB トークンを用いてユーザ認証を行う方法として, 内部に格納された X509 証明書を用いる方法がある. しかし, これだけでは USB トークンの所有者を識別することはできても, どのハードウェア上で認証要求が出されているかを識別することができない.

そこで, あらかじめハードウェアごとに固有の証明書 (ハードウェア用証明書) を持たせておき, その証明書による SSL クライアント認証を用いてサーバとの間にセッションを確立する. その上で, USB トークンに格納されたユーザ用証明書を用いた認証処理を行い, ユーザ・ハードウェアの両方について識別する.

ユーザ用証明書を用いた認証処理に関しては, チャレンジアンドレスポンス方式に RSA 公開鍵暗号を加えた独自プロトコルを実装した. その認証処理の流れを次に示す. 以下で, クライアントとは起動制御レイヤーが配置されているハードウェアのことを言う.

1. クライアントはサーバに対し, ハードウェア用証明書を用いた SSL クライアント認証セッションを確立し, そのセッション上で認証処理を要求し, ユーザ名を名乗る.
2. サーバはユーザ名とハードウェア用証明書の情報からアクセス可否を決定し, 1 の返値としてランダムに生成したチャレンジデータを返す.
3. クライアントは受け取ったチャレンジデータに対し USB トークンを用いて署名を行い, レスポンスデータを作成し, 新しいセッションを確立してサーバに送る.
4. サーバは送られてきたレスポンスデータを, あらかじめ持っているそのユーザの公開鍵を用いて復号化し, 認証要求時に保存しておいたチャレンジデータと一致すれば認証成功とし, 3 の返値として OS のイメージを送信する.

## 4 関連研究

VMKnoppix[2] は, 1CD OS である Knoppix に仮想計算機モニタの機能を持たせたものである. これが起動できるようにハードウェアを設定すると, 他の 1CD OS も起動することが可能になる. また, この仮想計算機モニタ上ではゲスト OS の起動に関してユーザという概念がないため, どのようなゲスト OS でも起動することが可能になる.

## 5 おわりに

本稿では, USB トークンを用いた OS の起動制御の設計・実装方法について述べた. 今後は, この実装を用いて評価を行う. また, より強固なハードウェア認証方法として, TPM(Trusted Platform Module)[4] を用いた手法について検討・実装していく. そのほか, 本システムの具体的な利用モデルのひとつとして, SaaS(Software as a Service) 的な利用方法についても検討していく.

## 参考文献

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand,T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *ACM Symposium on Operating Systems Principles*. 2003, pp. 164-177.

[2] VMKNOPPIX: Collection of Virtual Machine, http://unit.aist.go.jp/itri/knoppix/vmknoppix/ Accessed 07 January 2008.

[3] 飛天ジャパン株式会社 - USB トークン・ドングル, http://www.ftsafe.co.jp/products/epass2000.htm Accessed 26 December 2007.

[4] Sundeep Bajikar. Trusted Platform Module(TPM) based Security on Notebook PCs-White Paper. 2002, pp. 1-20. http://www.intel.com/design/mobile/platform/downloads/Trusted_Platform_Module_White_Paper.pdf Accessed 10 January 2008.