社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－CSEC－34 (31)
2006／7／20

トラステッド・コンピューティングによる HTTP-FUSE KNOPPIX クライアントのセキュリティ強化

中村 めぐみ† 宗藤 誠治† 須崎 有康‡ 飯島 賢吾‡ 八木 豊志樹‡ 大澤 一郎‡

†日本アイ・ビー・エム株式会社 東京基礎研究所 〒242-8215 神奈川県大和市下鶴間 1423-3
‡独立行政法人 産業技術総合研究所 〒101-0021 東京都千代田区外神田 1-18-13

E-mail: †{nakamegu, munetoh}@jp.ibm.com, ‡{k.suzaki, k-iijima, yagi-toshiki}@aist.go.jp, ichiro@ni.aist.go.jp

あらまし ルートファイルシステムをインターネット経由で取得して起動する HTTP-FUSE KNOPPIX では、取得したファイル群の信頼性が端末の信頼性を大きく左右する。トラステッド・コンピューティングの技術を活用することで、サービス提供者から HTTP-FUSE KNOPPIX クライアントの完全性検証を行なうことが可能となる。本稿ではトラステッド・コンピューティングの活用による安全なシンクライアント環境の構築に向けて、HTTP-FUSE-CLOOP による分割圧縮ブロックの配信の仕組みと、トラステッド・コンピューティングの完全性検証の仕組みを組み合わせることにより、セキュリティチップ(TPM)にクライアントの完全性を記録する手法を提案し、その問題点と課題を考察する。

キーワード: シンクライアント, トラステッド・コンピューティング

Security Enhancement of HTTP-FUSE KNOPPIX Client by Trusted Computing

Megumi NAKAMURA† Seiji MUNETOH† Kuniyasu SUZAKI‡
Kengo IIJIMA‡ Toshiki YAGI‡ and Ichiro OSAWA‡

† Tokyo Research Laboratory, IBM Japan Ltd. 1423-2 Shimo-tsuruma, Yamato-shi, Kanagawa, 242-8125 Japan
‡ National Institute of Advanced Industrial Science and Technology
1-18-13 Sotokanda, Chiyoda-ku, Tokyo, 101-0021 Japan

E-mail: †{nakamegu, munetoh}@jp.ibm.com, ‡{k.suzaki, k-iijima, yagi-toshiki}@aist.go.jp, ichiro@ni.aist.go.jp

Abstract HTTP-FUSE KNOPPIX starts by getting the root filesystem by way of the Internet. The reliability of this KNOPPIX is greatly controlled by the reliability of the files acquired from the server. The service provider can verify the integrity of the HTTP-FUSE KNOPPIX client by using the technology of Trusted Computing. To construct the secure thin client environment using Trusted Computing, we combine the delivery mechanism of the split-compressed loopback device of HTTP-FUSE-CLOOP and the mechanism of the integrity verification of Trusted Computing, and then, we propose how to record the integrity information in security chip (TPM).

Keyword: Thin Client, Trusted Computing

1. はじめに

機密情報などの情報漏えい対策としてシンクライアントを用いる場合、そのシンクライアント自体が正しい動作をすること、たとえば、不正な改ざんやなりすましへの十分な対策がなされていることが必要である。このようにユーザーやサービス提供者が期待するようなコンピュータを実現することをトラステッド・コンピューティングといい、シンクライアントの環境としてそのようなトラステッドなクライアント環境が求められる。

シンクライアントの構成としては様々な形態が考えられるが、その1つとして CD-ROM から起動する Linux[1]である KNOPPIX[4]を利用するという方法があり、CD-ROM だけでなく USB メモリキーから起動するものがある。また、通常は動作時に使用するファイル群はそれら CD-ROM や USB メモリに格納されるが、それ以外の方法の1つとして、不特定多数のユーザーを対象に世界中のどこからでもインターネットさえ接続できればシンクライアントを構成することができる

[1] Linux は Linus Torvalds の米国およびその他の国における商標

HTTP-FUSE KNOPPIX[2]がある。これは、圧縮されたブロックファイルをまとめて1つのループバックデバイスに見せるHTTP-FUSE CLOOPを用いることで、オンデマンドでクライアント環境を構成する。この方法は、ローカルマシン内にあらかじめ必要なデータを保持していなくてもシンクライアントを構成することができるという利点がある。しかし、シンクライアントを構成するために必要なデータをインターネット経由で入手するため、インターネット経由で入手したデータが正しいものであるかどうかを検証することが重要となる。

そこで、HTTP-FUSE KNOPPIXを用いたシンクライアント環境の特性を考慮し、そのセキュリティについて脅威分析を行う。そして、HTTP-FUSE-CLOOPによる分割圧縮ブロックの配信の仕組みと、トラステッド・コンピューティング[1]の完全性検証の仕組みを組み合わせることにより、セキュリティチップ(TPM: Trusted Platform Module)にクライアントの完全性を記録する手法を提案すると共に、トラステッド・コンピューティングの活用による安全なシンクライアント環境の構築について、その問題点と課題を考察する。

2. HTTP-FUSE CLOOP[2]

HTTP-FUSE CLOOPとはネットワークに対応したループバックブロックデバイスである。通常のループバックデバイスは一つの巨大なブロックファイルを使うが、HTTP-FUSE CLOOPでは複数に分割され、圧縮されたブロックファイルをまとめて1つのループバックデバイスに見せる。ブロックファイルは現在もっとも汎用なファイル配信インフラとして使われているHTTPでのダウンロードを想定している。

HTTP-FUSE CLOOPは、ICD LinuxであるKNOPPIX[4]で使われているCLOOP (Compressed Loopback Device)を元に開発した。CLOOPではブロックデバイスを固定サイズ(標準で64KB)毎に切り出し、zlibで圧縮して一つのループバックファイルに収納している。HTTP-FUSE CLOOPでは、元のブロックデバイス上で使われているファイルシステム(iso, ext2, etc.)に依存せず、固定サイズで切り出して圧縮した後はそれぞれのファイルに保存する。ファイル名はブロックの内容のSHA1ハッシュ値とする。同一内容の場合、ハッシュ値が同じになりファイル数を減らすことができる。

CLOOPではブロックファイルのヘッダに位置情報のテーブルが付け加えられている。CLOOPにアクセスがあるとドライバがこのテーブルから該当ブロックを展開して割り当てている。HTTP-FUSE CLOOPでは位置情報とブロックファイル名（SHA1のハッシュ値）のテーブルをインデックスファイルに格納し、管理している。

HTTP-FUSE KNOPPIXはCLOOPファイルをHTTP-FUSE CLOOPに置き換えたものである。つまり、ルートファイルシステムが収録されているCLOOPがCDから取り除かれて、代わりにHTTP-FUSE CLOOPドライバがminirt.gzファイルに収録されている。CDには起動に必要なブートローダ(isolinux)、Linuxカーネル、minirt.gzファイルのみの構成になり、サイズが700MBから6MBに減少した。ルートファイルシステムは起動後、HTTP-FUSE CLOOPを介してオンデマンドで取得する。

CD版のKNOPPIXでは、ルートファイルシステムにわずかな更新であっても、CLOOP全体の作り直しとなり、非効率であった。HTTP-FUSE-CLOOPでは各ブロックをファイルとして扱えるようにしたために、変更の生じたファイルとその位置情報であるインデックスファイルの変更で差分更新に対応可能となった。また、ルートファイルシステムの変更は、ブロックファイルをHTTPサーバーへ追加するだけで済むため、セキュリティパッチなどの対処が簡単になった。

3. 脅威と対策

HTTP-FUSE KNOPPIXにおける脅威として、クライアント端末やその構成ファイルの改ざんやなりすまし、モニタリングが挙げられる。

改ざん：不正な配信サーバーにより、改ざんされたルートファイルシステムが配信されることが考えられる。この対策としては、ルートファイルシステムのインデックスであるインデックスファイルの配布元を確認するために署名を添付する、HTTPSによるサーバーの認証を行う、などが挙げられる。この場合は証明書を改ざんされる可能性を考えなければならない。証明書の改ざんに対する対策としては、ブートメディアに保存して安全な手続きでユーザーへ配布し、ユーザーが安全に管理するといった方法が挙げられる。

なりすまし：不正な端末がクライアント端末になりすましてサーバーに接続することが考えられる。根本的な対策としては、クライアント環境上に保護対象となる資産を置かないこと、資産がある場合には暗号化を行うこと、暗号化にはTPMを用いて特定のクライアント環境でしか利用できないようにすることなどが挙げられる。

モニタリング：不正に仮想マシンやエミュレータを利用することにより、意図しないモニタリングをされることが考えられる。この対策としては、TPMを活用して検証を行うという方法がある。

また、TPMを活用したTrusted Bootとリモートアテ

ステーションにより、サーバー側からクライアントを検証するという方法は、改ざんやなりすましへの対策となる。詳しくは次章より説明する。

4. トラステッド・コンピューティング

4.1. 完全性検証

クライアント端末の完全性を検証するため、Trusted Computing Group (TCG)[1]によって定められたトラステッド・コンピューティングの技術を用いる。TCGでは、耐タンパ性のあるハードウェアモジュールであるTPM (Trusted Platform Module)を用いたプラットフォームの完全性測定と通知について定義している。図 1に完全性測定と、チャレンジ・レスポンスによる検証者への通知について示す。TPMが持つPCR (Platform Configuration Register)と呼ばれるレジスタの値は、式1によって示されるExtend と呼ばれる操作でしか更新できない。

$$PCR(i) = SHA1(PCR(i + 1) + Digest)$$

式 1：Extend

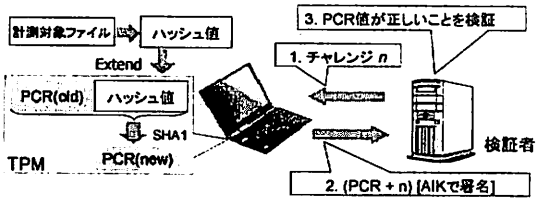

図 1：完全性測定と通知

TCGが定めるTrusted Bootstrapでは、CRTM (Core Root of Trust for Measurement)というBIOSの中で書き換え不能な領域がBIOSの完全性を測定し、その結果を用いてPCRをExtendする。続いてBIOSからブートローダをExtendする。

このようにしてシステムが起動した段階でPCRの値は特定のBIOSバージョン、OSバージョンに依存する値となるため、この値を調べることでプラットフォームがどのようなBIOS、カーネルを使って起動したかを調査することができる。また、外部の検証者に対してこのPCRの値をTCS (TCG Core Service)を介して知らせることができる。

5. 関連研究

5.1. CLOOPファイルの完全性検証

文献[3]では、TCGの技術を用いてKNOPPIXの完全性測定を行なっている。Trusted Bootstrapの機能を持つ端末がシステム起動時にBIOSを計測し、続いてBIOSからブートローダ (Trusted GRUB) を計測する。そしてブートローダが、KNOPPIXにおいて重要なファイルであるカーネル、暫定的なルートファイルシステムであるminirt.gzファイル、ルートファイルシステムを計測する。

ルートファイルシステムは、マウントして伸張した後に計測を行なうと、測定対象となるファイル数が非常に多くなってしまい煩雑である。そこで起動時にルートファイルシステム全体を測定することで、実行ファイルや設定ファイルの種類を問わず、システム全体の完全性を検証することができる。このとき、ブートローダがルートファイルシステム全体の測定をしてから起動した場合、非常に多くの時間が必要である。

そこで効率よく測定するため、図 2に示すように、ルートファイルシステムの各ブロックに対してハッシュ値をあらかじめ計算し、全ブロック分のハッシュ値をリストファイルに保持する。ブートローダからはルートファイルシステムを計測せず、代わりにハッシュ値のリストを計測する。実際にファイルがアクセスされたときに読み出されたブロックに対してハッシュ値を求め、リストに記録された値との一致を確認することでルートファイルシステムの検証を行なう。

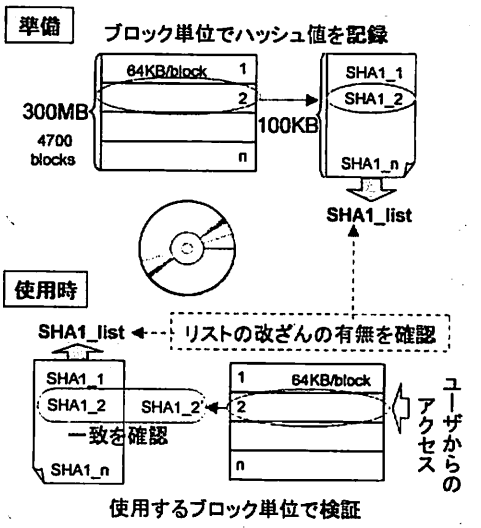

図 2：CLOOPファイルの完全性検証[3]

5.2. Integrity Measurement Architecture[8]

Integrity Measurement Architecture (IMA)は、Linux Security Module (LSM)に対応しており、実行ファイルやモジュールなどを実行時に測定してTPMに記録する。IMAのメカニズムではシステム起動後の動作を詳細に記録することができるが、PCRに記録される値は動的に変化する。そのため、検証にはどのコンポーネントの値をどの順番で計測したかというログが必要と

なり、正しい値であるかどうかの検証作業が複雑になる。

6. セキュリティ強化

6.1. HTTP-FUSE KNOPPIX クライアントに求められるセキュリティ機能

HTTP-FUSE KNOPPIX を用いたシンクライアントでは、図 3に示すような3つの役割が考えられる。実際にシンクライアント環境を使用するユーザー（以下、ユーザー）、シンクライアント環境を構築するためのイメージを配布するサーバー（以下、配布サーバー）、シンクライアントへサービスを提供するサービスプロバイダー（以下、プロバイダー）である。

ユーザーは配布サーバーからルートファイルシステムをダウンロードしてシンクライアント環境を起動し、プロバイダーからサービスを得る。また、プロバイダーは、配布サーバーがセキュリティ機能を搭載した正しい KNOPPIX を配布していることを求める。配布サーバーはプロバイダーに対して、配布しているルートファイルシステムに含まれる機能についての情報を提供する。

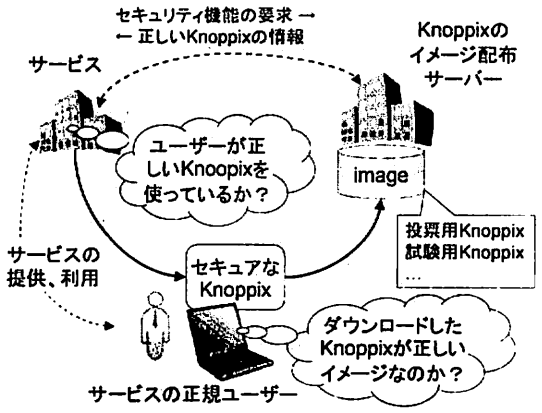

図 3：HTTP-FUSE KNOPPIX セキュリティ

本稿ではこの3者のうち、ユーザーにおけるセキュリティ強化について検討する。

ユーザーの立場から考えると、まずシンクライアント環境を構築する際に、配布サーバーから正しいルートファイルシステムを入手できることが重要である。そのためには、正しいサーバーへ接続していること、改ざんのない正しいルートファイルシステムを入手できることを確認したい。次に、構築したシンクライアント環境がプロバイダーからサービスを受ける場合などに、プロバイダーに対してユーザー自らが正しい構成をしていることを主張することができる必要がある。

クライアントが起動してからルートファイルシステムを取得するまでの流れと、そのセキュリティ強化のために付加する機能を図 4に示す。

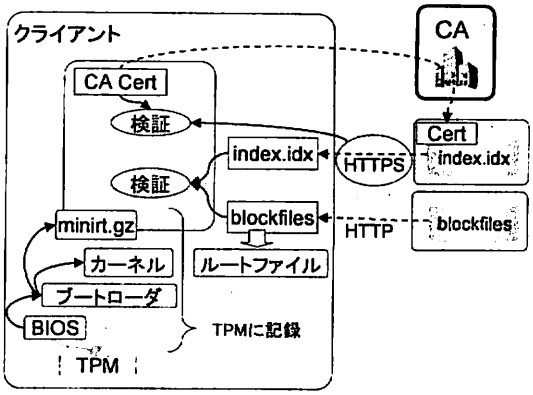

図 4：クライアントにおけるセキュリティ機能を TPM へ記録

まず、クライアントの完全性を TPM に記録する。ここでは Trusted Boot により起動時の BIOS の状態から、HTTP-FUSE を利用するまでに使用する重要なコンポーネントを TPM に記録する。こうすることで、プロバイダーなどからリモートアテステーションによりクライアントの完全性を検証することができる。また、TPM の暗号鍵を利用して、クライアント上に保管する資産の暗号化を行なうこともできる。

次に、HTTP-FUSE によりルートファイルシステムを取得し、構成する。配布サーバーへ接続する機能は minirt.gz ファイルに含まれている。まずインデックスファイルを取得し、それに従ってルートファイルシステムを取得していく。そのため、インデックスファイルの信頼性を確保することが重要である。このファイルをクライアント側で保管している場合は、起動時にインデックスファイルの完全性を TPM に記録し、外部接続時に接続先から検証を行なう。インデックスファイルをサーバーに接続して入手する場合、サーバー側があらかじめ署名を付加しておき、クライアント側で署名の検証を行なう。署名の検証に必要な証明書は、起動用の CD-ROM や USB メモリに保管しておく。

ルートファイルシステムを取得するときは、前の段階でインデックスファイルの信頼性を確保しているので、その情報に従ってハッシュ値を検証する。

これらを次節より詳しく説明する。

6.2. Knoppix のトラステッド・コンピューティング対応

Trusted Boot に対応した PC を使用することを考え、BIOS からブートローダ（GRUB）、OS という一連の起動シーケンスを TPM へ記録する。

残念ながら現在市販されている TPM 搭載 PC の全て

が Trusted Boot をサポートしているわけではないが、TCG では BIOS がサポートするべき Trusted Boot の仕様が決められている。Trusted Boot に対応した BIOS は自身の完全性情報を TPM に記録し、続いて、次の段階であるブートローダ（MBR）の計測を行う。またブートローダが TPM へアクセスするための API も提供する。BIOS 以降、ブートローダの TCG 対応のため、TCG 対応のための GRUB パッチを適用した GRUB（Trusted GRUB）[5]を使用する。しかし、現在公開されている TCG 対応のための GRUB パッチでは OS のセキュリティの完全性を計測する上で機能が不完全なため、修正を行なった。

具体的には、OS の起動コマンドラインオプションや、GRUB 内部で発生する各種イベントの詳細を TPM へ記録する。GRUB での計測は BIOS API を経由して TPM に記録されるため、計測のイベント情報も BIOS が管理する ACPI テーブル中に記録される。そのため、BIOS からブートローダ、OS までの起動の過程はこのテーブルに一元管理されることになる。

KNOPPIX 5.0.1 ではカーネル 2.6.17 が使用されており、これには TPM v1.1, v1.2 両方の TPM ドライバが含まれている。TCG の機能を使用するためのソフトウェアである TCG Software Stack (TSS)として、オープンソースとして公開されている TrouSerS[6]を使用する。しかし TrouSerS では、どのコンポーネントを計測したかといったイベントログ処理の実装が不十分だった。イベントログを処理するためには、まず TPM ドライバの持つ BIOS イベントログへのアクセス機能を利用する必要があるが、2.6.16 までのバイナリ・フォーマットではイベント情報が十分に得られない問題があり、その修正を行った。最新の 2.6.17 では BIOS のイベントログ（ACPI テーブル）の情報がそのままユーザー空間に開示されるようになっている。TrouSerS での対応は現在作業中である。その他に、個々の PC 端末が持つ TPM 関連の情報を管理するためのツールが必要である。

6.3. 完全性検証

主要なコンポーネントを Trusted Boot によって TPM に記録する。ここでは、HTTP-FUSE-CLOOP によるブロックのチェック機能を持たせ、そのプログラムと設定を minirt.gz ファイルに含める。この minirt.gz ファイルを GRUB から完全性を計測し、TPM に記録する。

HTTP-FUSE-CLOOP における完全性検証の方法として、まず登録された正規のサーバーからのインデックスファイルを取得したことを確認する。そしてそのインデックスファイルで示される圧縮ブロックのハッシュ値を基に、各ブロックファイルの検証を行なうことで、ルートファイルシステムの正当性を確認する。

6.3.1. インデックスファイルの検証

インデックスファイルを外部より取得する場合、SSL を用いて取得先を特定し、インデックスファイルの配布サーバーの証明書をクライアント側で確認する。この場合、証明書とインデックスファイルの対応を確認することは可能であるが、使用する証明書が正しいかどうかを別途検証する必要がある。

使用する配布サーバーを特定できる場合は、あらかじめ証明書を minirt.gz ファイル内に含めて保存しておき、起動時に minirt.gz ファイルを測定して TPM に記録するという方法がある。この場合は minirt.gz ファイルのサイズが大きくなってしまうが、HTTP-FUSE のツールと同時に完全性を検証可能である。しかし、ネットワークなど外部から minirt.gz ファイルを入手する場合は、改ざんされたファイルを入手してしまう可能性がある。これについては後に6.3.3節で述べる。

配布サーバーをあらかじめ特定できない場合、サーバーをカーネルのオプションとして指定する。この場合、配布サーバーの URL の情報を GRUB により TPM に記録し、PCR の値とイベントログの照合を行うことで、使用した配布サーバーの URL を確認することができる。

6.3.2. 各ブロックの検証

登録された正規のサーバーからインデックスファイルを取得し、そのインデックスファイルに記載された内容を用いてブロックの完全性検証を行なう。

HTTP-FUSE-CLOOP では、インデックスファイルに分割ファイルのファイル名が記載されている。ファイル名は分割ブロックのハッシュ値と等しいため、分割ブロックを入手した後、ハッシュ値を計算してファイル名との比較を行なうことで、分割ファイルの正当性を確認する。

この仕組みを KNOPPIX に適用するため、HTTP-FUSE の wrapper にブロックのハッシュ値を検証する機能を持たせる。この Wrapper は minirt.gz ファイルに含まれ、起動時にブートローダからその完全性が TPM に記録される。

6.3.3. 初期ルートファイルシステムの検証

minirt.gz ファイルは、分割ルートファイルシステムを取得する前の状態で用いられる暫定的なルートファイルシステムである。ファイル内には、分割ルートファイルシステムを取得して環境を移行するために必要なファイル群が含まれる。それらは主にライブラリ、バイナリであり、限られた役割しか持たないため、構成するファイルの種類は限られている。

そのため、事前に必要最低限のファイル類を調査し、

それらのみから構成されるminirt.gzファイルのハッシュ値を測定しておくことができる。ここではあらかじめ配布したminirt.gzファイルと同一か否かを調査する。

6.3.4. 暗号ライブラリの活用

HTTP-FUSE では、分割ファイルを取得しない限りKNOPPIX 環境を構築することができないので、分割ファイルを取得する時間を短縮することが望まれる。そこで、OpenSSL[7]などに付属する高速な暗号実装を利用し、署名検証やハッシュの計算を高速化する。

7. プロトタイプシステムの評価

HTTP-FUSE KNOPPIX において、使用するブートローダを Trusted GRUB[5]とし、6章に示したようにカーネルと minirt.gz ファイルの完全性検証を行なう。これにより起動時間は増加する。

また、HTTP-FUSE-CLOOP を改良し、OpenSSL ライブラリを使用したハッシュ値の検証機能を加えたものを実装した。機能を追加する前後で異なるのは、minirt.gz ファイルのサイズ、取得したブロックファイルの検証時間の 2 点である。まず、minirt.gz のファイルサイズの増加は、OpenSSL ライブラリを追加したことによるものである。追加前は 2.9MB であったものが、追加後は 3.6MB であった。このサイズの増加により、GRUB が minirt.gz ファイルを読み込む時間が増加する。これは Trusted GRUB による計測に要する時間にも影響するものである。

ブートローダを TCG 対応の Trusted GRUB に変更したことと、取得したブロックファイルの検証による時間の増分を調べるため、機能追加の前後において KNOPPIX が起動するまでの時間を測定した。

クライアント端末としてノート PC（ThinkPad X31、メモリ 1GHz、CPU 1.5GHz）を用いて動作テストを行なった。搭載されている TPM は TCG v1.1 準拠[1]のものである。使用したカーネルは 2.6.17.1（1.5MB）である。ルートファイルシステム全体では 240MB であり、256KB ごとにブロックファイルを作成して 2394 個のファイルに分割されている。カーネルと minirt.gz ファイルを USB メモリキー(Lenovo 社 ThinkPlus 1GB USB 2.0) に格納して使用した。

サーバー端末はメモリ 1.5GHz、CPU 1.66GHz であり、クライアント・サーバー間のネットワーク環境は LAN（100Mbps）である。

クライアント端末における HTTP-FUSE KNOPPIX の起動時間を表 1に示す。時間は、起動画面を映像で取得し、フレーム単位の粒度で測定したものである。GRUB メニュー画面において HTTP-FUSE オプションの設定を終え、起動開始のキーを押下した時間を 0 とし、カーネル読み込み以降の時間を測定した。

表 1：機能追加による起動時間の増加

(分:秒:1/10 秒)

	機能追加前	機能追加後	差分
カーネル読込*	00:02:0	00:05:8	00:03:8
minirt.gz 読込*	00:03:9	00:09:6	00:05:7
カーネル起動	00:03:7	00:03:9	00:00:2
minirt 起動	01:17:3	01:18:2	00:00:9
X Window 起動	00:24:7	00:26:5	00:01:8
合計			00:12:4

* 機能追加後は Extend(式 1)も行なう

この起動が完了するまで間に使用したブロックファイルの数は約 700 であった。

表 1から、GRUB による完全性検証に、カーネル、minirt.gz ファイルとも 4,5 秒を要しており、これがセキュリティ機能を加えたことで最も大きい差が出た部分である。HTTP-FUSE-CLOOP によりハッシュ値を検証する部分に関しては、特に X Window 起動の部分は HTTP-FUSE によりファイルを取得しながら起動しているが、増加時間は 7%であり、起動時間全体に与える影響は約 2 秒である。そのため、クライアント起動時には Trusted GRUB による検証のために 10 秒程度の時間が余計に必要となるが、その後の使用時には HTTP-FUSE-CLOOP を用いてファイルを取得する時間の増加は数パーセントに抑えられている。

8. 今後の課題

8.1. ライフサイクル管理

本方式を適用した HTTP-FUSE KNOPPIX が正しく利用されるためには、図 3に示したような 3 者間の連携が必要である。

その中で、ユーザーが正しい KNOPPIX を使用しているかどうかを検証するために、TPM を使用している。そのため、あらかじめユーザーが使用する PC 端末で TPM を初期化すること、使用する AIK を生成することが必要となる。

また、ユーザーは正しい配布サーバーであることを確認するために配布サーバーの証明書を入手しておくことが必要である。

ユーザーがサービスを使用するためには、サービス提供者があらかじめユーザーの情報として、使用される PC 端末の情報を知っておく必要がある。またどのようなイメージが配布されているかという情報を配布サーバーから入手しておく必要がある。

このように情報の登録や削除について、3 者が連携しないとこのシステムを有効に利用することができない。

8.2. 完全性情報の管理

ユーザーの環境で起動している KNOPPIX が正しいものであるかどうかを検証するために、配布されているルートファイルシステムと同一か否かの判断だけではなく、その中身がどの程度のセキュリティ機能を有しているかを確認したい。また、セキュリティの強さはユーザーが使用する PC 端末の種類や設定によっても変化するため、その点についても検討する必要がある。

そこで、どのような構成であればセキュアであるのかを調べるために、脆弱性情報やパッケージ情報などを格納したデータベースを用意し、その情報を参照しながら実際のユーザーの環境のセキュリティの強度を算出するサービスを提供することを考えている。

8.3. セキュア OS の完全性保証

Linux では SELinux をはじめとして各種のセキュリティ機能を強化するモジュールやパッチを使用することができ、強制アクセス制御による高度なセキュリティを実現できる。さらに、TPM による完全性計測によって、こうしたセキュリティ機能が正しく有効化されていることを保証することが可能になる。今回、Grub の機能拡張によって、OS の起動コマンドラインオプションなど、OS 起動時からの詳細な情報を TPM に記録することを可能にした。

実際には、セキュリティ機能は OS の各種設定やポリシーの管理方法など、複雑な依存性を持つが、今後さらに、各セキュア OS のそうした特徴を踏まえた完全性の記録手法について検討を進める必要がある。

8.4. ネットワークブート対応

現在の HTTP-FUSE KNOPPIX ではルートファイルシステムをネットワーク経由で入手するが、ルートファイルシステムだけでなくユーザー側のマシンにファイルが何もない状態からシンクライアントを構成するなど、ネットワークブート時のセキュリティについて考察する。

ネットワークブートには、PXE (Preboot eXecution Environment) 機能を用いる。DHCP により IP アドレスを取得したクライアントは、PXE サーバーに対して NBP (Network Bootstrap Program) を要求する。この NBP の完全性計測を行なう場合を想定し、セキュリティ上の問題点を考える。

現状では、PXE 機能による起動では、BIOS OptionROM が TCG に対応していないため、Trusted Boot をすることができない。

また別の問題として、配布サーバーの正当性を確認するための証明書を minirt.gz ファイル内に含めて保持する場合には、その minirt.gz ファイル自体をネットワーク経由で入手することになってしまうので、偽の minirt.gz ファイルで改ざんされる可能性がある。そのため minirt.gz ファイル内部に置く情報の信頼性が下がることへの対応を考慮しなければならない。

8.5. 仮想化環境への対応

Xenoppix のように、KNOPPIX に仮想マシンモニター Xen を追加したものでは、1 台のホスト OS 上で複数の仮想マシンを設定し、複数のゲスト OS を動かすことができる。この場合、仮想化することによりトラスト・チェーンが切れないよう考慮する必要がある。

9. おわりに

ルートファイルシステムをインターネット経由で取得して起動する HTTP-FUSE KNOPPIX では、取得したファイル群の信頼性が端末の信頼性を大きく左右するため、トラステッド・コンピューティングの技術を活用することで、安全なシンクライアント環境の構築に向けて、HTTP-FUSE KNOPPIX のセキュリティを強化する方法を提案した。

ブートローダによる完全性検証を行うことでクライアントの起動に要する時間は増加し、また、HTTP-FUSE CLOOP への機能追加のために起動時、使用時におけるルートファイル取得に要する時間は少々増加するが、起動時に BIOS、カーネル、minirt.gz ファイルの完全性について TPM に記録し、クライアント端末のセキュリティ情報として外部へ提出することができるようになった。さらに起動時に記録された minirt.gz ファイルに含まれている HTTP-FUSE CLOOP ツールを用いることで、ルートファイルの完全性を検証しながら取得することができる。

セキュリティ強化の目的で使用するシンクライアントにおいては、このようなトラステッド・コンピューティングの技術が有用である。

謝　辞

本研究の一部は、経済産業省、新世代情報セキュリティ研究開発事業の研究として行われたものである。

文　献

[1] Trusted Computing Group, https://www.trustedcomputinggroup.org/

[2] 須崎有康, 八木豊志樹, 飯島賢吾, 北川健司, 田代秀一, "ネットワークに対応した分割圧縮ループバックデバイス HTTP-FUSE-CLOOP とそれから起動する Linux,"インターネットコンファレンス 2005 (IC2005), 2005 年 10 月.

[3] 中村めぐみ, 宗藤誠治, 吉濱佐知子, "シンクライアントにおける完全性検証とその効率化,"

Symposium on Cryptography and Information Security (SCIS2006), 2B1-1, 2006年1月.

[4] KNOPPIX, http://www.knopper.net/

[5] GRUB TCG Patch to support Trusted Boot, http://trousers.sourceforge.net/grub.html

[6] TrouSerS, http://trousers.sourceforge.net/

[7] OpenSSL Project, http://www.openssl.org/

[8] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, Leendert van Doorn, Design and Implementation of a TCG-based Integrity Measurement Architecture. USENIX Security Symposium 2004, pp.223-238.