(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5986897号
(P5986897)
(45)発行日　平成28年9月6日(2016.9.6)　(24)登録日　平成28年8月12日(2016.8.12)

(51)Int.Cl.		F I		
G06F 21/57	(2013.01)	G06F	21/57	350
H04L 9/32	(2006.01)	H04L	9/00	675B
H04L 9/10	(2006.01)	H04L	9/00	621Z

請求項の数 7　(全 17 頁)

(21)出願番号　特願2012-249108(P2012-249108)
(22)出願日　平成24年11月13日(2012.11.13)
(65)公開番号　特開2014-98951(P2014-98951A)
(43)公開日　平成26年5月29日(2014.5.29)
審査請求日　平成27年8月6日(2015.8.6)

(73)特許権者　000208891
ＫＤＤＩ株式会社
東京都新宿区西新宿二丁目３番２号
(74)代理人　100106909
弁理士　棚井　澄雄
(74)代理人　100064908
弁理士　志賀　正武
(74)代理人　100146835
弁理士　佐伯　義文
(72)発明者　竹森　敬祐
埼玉県ふじみ野市大原２丁目１番１５号
株式会社ＫＤＤＩ研究所内
(72)発明者　川端　秀明
埼玉県ふじみ野市大原２丁目１番１５号
株式会社ＫＤＤＩ研究所内
最終頁に続く

(54)【発明の名称】端末装置、完全性検証システム、およびプログラム

(57)【特許請求の範囲】
【請求項１】
OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、
カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、
前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、
を含むOSプログラムを記憶する第１の記憶部と、
前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、
前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、
前記サービスプログラムを測定したときに期待される測定結果である第１の期待値と前記第１の測定値とを比較し、前記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値と前記第２の測定値とを比較し、前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値と前記第３の測定値とを比較することにより前記OSの完全性を検証するサーバへ前記第１の測定値、前記第２の測定値

、および前記第３の測定値を送信する送信部と、
を備えたことを特徴とする端末装置。
【請求項２】
前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動する第１のブートローダの機能を実現するための、書き換え可能な第１のブートローダプログラムを記憶する前記第２の記憶部と、
前記第１のブートローダプログラムを測定し、測定結果である第４の測定値を生成した後、前記第１のブートローダを起動する第２のブートローダの機能を実現するための、書き換え不可能な第２のブートローダプログラムを記憶する第３の記憶部と、
前記第１のブートローダプログラムを測定したときに期待される測定結果である第４の期待値と前記第４の測定値とを比較することにより前記第１のブートローダプログラムの完全性を検証する検証部と、
を有することを特徴とする請求項１に記載の端末装置。
【請求項３】
前記検証部は、耐タンパ性を有するTPM（Trusted Platform Module）であって、
前記第４の期待値を記憶する記憶部と、
前記第４の期待値と前記第４の測定値とを比較することにより前記第１のブートローダプログラムの完全性を検証する比較部と、
を有することを特徴とする請求項２に記載の端末装置。
【請求項４】
前記TPMはさらに、前記第１の測定値、前記第２の測定値、および前記第３の測定値の電子署名を生成する署名部を有し、
前記送信部は、前記サーバへ、前記第１の測定値、前記第２の測定値、前記第３の測定値、および前記電子署名を送信する
ことを特徴とする請求項３に記載の端末装置。
【請求項５】
前記サービス測定・起動プログラムは、起動した前記カーネルが最初に実行するプログラムであることを特徴とする請求項１～請求項４のいずれか一項に記載の端末装置。
【請求項６】
端末装置およびサーバを備えた完全性検証システムであって、
前記端末装置は、
OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、
カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、
前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、
を含むOSプログラムを記憶する第１の記憶部と、
前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、
前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、
前記サーバへ前記第１の測定値、前記第２の測定値、および前記第３の測定値を送信する送信部と、
を有し、
前記サーバは、
前記サービスプログラムを測定したときに期待される測定結果である第１の期待値、前

記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値、および前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値を記憶する第２の記憶部と、
前記端末装置から前記第１の測定値、前記第２の測定値、および第３の測定値を受信する受信部と、
前記第１の期待値と前記第１の測定値とを比較し、前記第２の期待値と前記第２の測定値とを比較し、前記第３の期待値と前記第３の測定値とを比較することにより、前記OSの完全性を検証する検証部と、
を有することを特徴とする完全性検証システム。
【請求項７】
OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、
カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、
前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、
を含むOSプログラムを記憶する第１の記憶部と、
前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、
前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、
前記サービスプログラムを測定したときに期待される測定結果である第１の期待値と前記第１の測定値とを比較し、前記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値と前記第２の測定値とを比較し、前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値と前記第３の測定値とを比較することにより前記OSの完全性を検証するサーバへ前記第１の測定値、前記第２の測定値、および前記第３の測定値を送信する送信部と、
としてコンピュータを機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、スマートフォン等の端末装置の完全性（改竄されていないこと）を検証する技術に関する。
【背景技術】
【０００２】
OS（Operating System）から独立した、耐タンパ性を有する専用チップで構成され、暗号化等の処理を安全に行うTrusted Platform Module（TPM）が開発されている。TPMは主に、x86アーキテクチャをベースとした端末、いわゆるパソコンに搭載されている。非特許文献１，２には、TPMを利用して、端末上のファイルが改竄されていない完全な状態であるか否かを検証しながら端末を安全に起動させる「セキュアブート」を可能とする技術が記載されている。
【０００３】
この技術では、パソコン用のOSのプログラムが書き込まれたCD-ROMを起動ディスクとしてOSを起動させる際に、OSの起動を行うブートローダのプログラムに対して、TPMを利用して完全性の検証が行われる。CD-ROMに書き込まれたOSのプログラムを書き換えることはできないので、ブートローダの完全性を検証することで、端末が完全な状態で起動したか否かを確認することができる。
【０００４】

完全性の検証は、例えば以下のように行われる。
（１）検証対象のプログラムに対応する完全な状態のプログラムを構成する各ファイルをハッシュ関数の入力値としてハッシュ値を算出する「測定」を行い、測定結果としてハッシュ値（期待値）を得る。このハッシュ値は、検証対象のプログラムに対して同様の測定を行った場合に測定結果として得られることが期待される値である。
（２）検証対象のプログラムを構成する各ファイルに対して測定を行い、測定結果としてハッシュ値（測定値）を得る。
（３）完全な状態のプログラムを測定して得られた期待値と、検証対象のプログラムを測定して得られた測定値とを比較する。両方の測定値が一致した場合、検証対象のプログラムは完全であり、両方の測定値が異なる場合、検証対象のプログラムは完全ではないと判断できる。

【先行技術文献】
【非特許文献】
【０００５】
【非特許文献１】“KNOPPIXとTPMの組み合わせで「安全なサービス」実現へ”，［online］，［平成２４年１０月２９日検索］，インターネット＜URL:http://www.atmarkit.co.jp/news/200802/07/aist.html＞
【非特許文献２】“Knoppix 5.1.1 for Trusted Computing Geeks”，［online］，［平成２４年１１月０７日検索］，インターネット＜URL:http://unit.aist.go.jp/itri/knoppix/TCGeeks-CD20071105.pdf＞

【発明の概要】
【発明が解決しようとする課題】
【０００６】
パソコンでは、OSの主要な機能を実現するプログラムを含むシステムファイルが保存されるハードディスク上のシステム領域がユーザに開放されている。このため、例えばユーザがパソコンに任意のアプリケーションを追加する等の行為により、システム領域の状態が任意に変化する。このような事情から、パソコンの完全な状態の基準を策定しきれない難しさがあり、2012年7月現在、パソコンのセキュアブートは普及に至っていない。
【０００７】
OSのシステムファイルが保存されるシステム領域と、ユーザが任意にファイルを追加することが可能なユーザ領域とが分離しているスマートフォン等の端末の場合、端末に対してウイルスによる攻撃等の特殊な状況を除いて、基本的には端末メーカや通信キャリアのみがシステム領域を設定することができる。OSの更新やパッチの適用が行われる場合でも、どのような更新あるいは適用が行われるのかを把握できることを前提とすれば、OSの更新あるいはパッチの適用後に想定されるシステム領域の状態を事前に策定することが可能となる。
【０００８】
しかし、OSのシステムファイルのサイズが膨大であるため、TPMの非力な演算回路がOSのシステムファイルの測定を行うと、膨大な時間を要する。そこで、処理を高速に行える端末のCPU（Central Processing Unit）を用いてOSのシステムファイルの測定を行う手法が考えられる。この手法では、OSを起動するブートローダがCPUを制御し、OSのシステムファイルの測定を行うことになる。しかし、ブートローダは、シングルタスク／シングルスレッドで動作するために、マルチタスク／マルチスレッドで動作可能なOSと比較して処理能力が低い。また、メモリ・キャッシュも限定的にしか機能していないためCPUの動作も、OSが機能している場合よりも遅い。このため、膨大なサイズのOSのシステムファイルの測定を行うのに時間を要し、セキュアブートを実現する際に端末の高速起動の障害となるという問題がある。
【０００９】
また、OSのシステムファイルの測定値と期待値との比較をTPMの演算回路が行うためには、TPM内の記憶装置にOSのシステムファイルの期待値を予め保持させておく必要がある

。しかし、OSのシステムファイルのサイズが膨大であるため、OSのシステムファイルの期待値も膨大となり、OSのシステムファイルの期待値を、記憶容量に乏しいTPMの記憶装置に予め保持させておくことが困難である。そこで、十分な記憶容量を有する記憶装置を備えた外部装置（サーバ）にOSのシステムファイルの期待値を予め保持させておき、端末の起動時にOSのシステムファイルの測定値を端末から外部装置へ送信し、外部装置がOSのシステムファイルの測定値と期待値との比較を行う手法が考えられる。

【００１０】

本発明は、上述した課題に鑑みてなされたものであって、端末装置の起動時にOSの完全性の検証を行うと共に端末装置をより高速に起動することを目的とする。

【課題を解決するための手段】

【００１１】

本発明は、上記の課題を解決するためになされたもので、OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、を含むOSプログラムを記憶する第１の記憶部と、前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、前記サービスプログラムを測定したときに期待される測定結果である第１の期待値と前記第１の測定値とを比較し、前記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値と前記第２の測定値とを比較し、前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値と前記第３の測定値とを比較することにより前記OSの完全性を検証するサーバへ前記第１の測定値、前記第２の測定値、および前記第３の測定値を送信する送信部と、を備えたことを特徴とする端末装置である。

【００１２】

また、本発明の端末装置は、前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動する第１のブートローダの機能を実現するための、書き換え可能な第１のブートローダプログラムを記憶する前記第２の記憶部と、前記第１のブートローダプログラムを測定し、測定結果である第４の測定値を生成した後、前記第１のブートローダを起動する第２のブートローダの機能を実現するための、書き換え不可能な第２のブートローダプログラムを記憶する第３の記憶部と、前記第１のブートローダプログラムを測定したときに期待される測定結果である第４の期待値と前記第４の測定値とを比較することにより前記第１のブートローダプログラムの完全性を検証する検証部と、を有することを特徴とする。

【００１３】

また、本発明の端末装置において、前記検証部は、耐タンパ性を有するTPM（Trusted Platform Module）であって、前記第４の期待値を記憶する記憶部と、前記第４の期待値と前記第４の測定値とを比較することにより前記第１のブートローダプログラムの完全性を検証する比較部と、を有することを特徴とする。

【００１４】

また、本発明の端末装置において、前記TPMはさらに、前記第１の測定値、前記第２の測定値、および前記第３の測定値の電子署名を生成する署名部を有し、前記送信部は、前記サーバへ、前記第１の測定値、前記第２の測定値、前記第３の測定値、および前記電子署名を送信することを特徴とする。

【００１５】

また、本発明の端末装置において、前記サービス測定・起動プログラムは、起動した前記カーネルが最初に実行するプログラムであることを特徴とする。

【００１６】

また、本発明は、端末装置およびサーバを備えた完全性検証システムであって、前記端末装置は、OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、を含むOSプログラムを記憶する第１の記憶部と、前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、前記サーバへ前記第１の測定値、前記第２の測定値、および前記第３の測定値を送信する送信部と、を有し、前記サーバは、前記サービスプログラムを測定したときに期待される測定結果である第１の期待値、前記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値、および前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値を記憶する第２の記憶部と、前記端末装置から前記第１の測定値、前記第２の測定値、および第３の測定値を受信する受信部と、前記第１の期待値と前記第１の測定値とを比較し、前記第２の期待値と前記第２の測定値とを比較し、前記第３の期待値と前記第３の測定値とを比較することにより、前記OSの完全性を検証する検証部と、を有することを特徴とする完全性検証システムである。

【００１７】

また、本発明は、OS（Operating System）が提供するサービスの機能を実現するためのサービスプログラムと、カーネルによる制御に従って、前記サービスプログラムを測定し、測定結果である第１の測定値を生成した後、前記サービスを起動するサービス測定・起動部の機能を実現するためのサービス測定・起動プログラムと、前記サービス測定・起動部を起動し、前記サービス測定・起動部によって行われる処理を制御する前記カーネルの機能を実現するためのカーネルプログラムと、を含むOSプログラムを記憶する第１の記憶部と、前記サービス測定・起動プログラムを測定し、測定結果である第２の測定値を生成すると共に、前記カーネルプログラムを測定し、測定結果である第３の測定値を生成した後、前記カーネルを起動するブートローダの機能を実現するためのブートローダプログラムを記憶する第２の記憶部と、前記OSプログラムおよび前記ブートローダプログラムに従って処理を行う処理部と、前記サービスプログラムを測定したときに期待される測定結果である第１の期待値と前記第１の測定値とを比較し、前記サービス測定・起動プログラムを測定したときに期待される測定結果である第２の期待値と前記第２の測定値とを比較し、前記カーネルプログラムを測定したときに期待される測定結果である第３の期待値と前記第３の測定値とを比較することにより前記OSの完全性を検証するサーバへ前記第１の測定値、前記第２の測定値、および前記第３の測定値を送信する送信部と、としてコンピュータを機能させるためのプログラムである。

【発明の効果】

【００１８】

本発明によれば、端末装置が、端末装置のOSの完全性を検証するサーバへ第１の測定値、第２の測定値、および第３の測定値を送信することによって、端末装置の起動時にサーバが端末装置のOSの完全性を検証することができる。また、カーネルによる制御に従って、サービス測定・起動部がサービスプログラムを測定することによって、端末装置をより高速に起動することができる。

【図面の簡単な説明】

【００１９】

【図１】本発明の一実施形態による端末装置の構成を示すブロック図である。
【図２】本発明の一実施形態による端末装置が有するCPUの機能構成を示すブロック図である。
【図３】本発明の一実施形態によるサーバの構成を示すブロック図である。
【図４】本発明の一実施形態におけるOSのファイルシステムを示す参考図である。
【図５】本発明の一実施形態による端末装置の動作の手順を示すフローチャートである。
【図６】本発明の一実施形態による端末装置の動作の手順を示すフローチャートである。
【図７】本発明の一実施形態によるサーバの動作の手順を示すフローチャートである。
【発明を実施するための形態】
【００２０】
以下、図面を参照し、本発明の実施形態を説明する。本実施形態による完全性検証システムは、完全性の検証の対象であるOSを有する端末装置と、端末装置が有するOSの完全性の検証を行うサーバとを含む。図１は、本実施形態による端末装置１の構成を示している。
【００２１】
端末装置１は、通信部１０（送信部）、CPU１１（処理部）、フラッシュメモリ１２、ROM（Read Only Memory）１３、RAM（Random Access Memory）１４、およびTPM１５を有する。通信部１０は、サーバ２と通信を行う通信回路を有する。CPU１１は、各種の処理を行う処理回路を有する。フラッシュメモリ１２は、各種のプログラムを記憶する不揮発性のメモリである。ROM１３は、各種のプログラムを記憶する、書き換えが不可能な不揮発性のメモリである。RAM１４は、各種のプログラムやデータを一時記憶する揮発性のメモリである。フラッシュメモリ１２、ROM１３、およびRAM１４は、以下で説明する各プログラムを記憶する記憶部として機能する。TPM１５（検証部）は、内部にCPU（図示せず）を有しており、耐タンパ性を有する専用チップで構成されている。本実施形態では、OSとして、スマートフォン用のOSとして一般的なAndroid（登録商標）が端末装置１に搭載されている場合を例として説明する。
【００２２】
フラッシュメモリ１２は、サービスプログラム１２０、サービス測定・起動プログラム１２１、カーネルプログラム１２２、および第１ブートローダプログラム１２３を記憶する。ROM１３は第２ブートローダプログラム１３０を記憶する。
【００２３】
サービスプログラム１２０は、OSが提供するサービスの機能を実現するためのシステムファイルを含むプログラムである。サービスプログラム１２０がフラッシュメモリ１２からRAM１４に読み込まれることによりサービス１４０が起動する。サービス１４０は、RAM１４に常駐しているプログラムであり、OSによってはデーモンと呼ばれることもある。起動したサービス１４０は、サービスプログラム１２０で規定されている命令をCPU１１に与えることにより、サービスの機能を実現する。
【００２４】
サービス測定・起動プログラム１２１は、サービスプログラム１２０を測定し、測定値（第１の測定値）を生成した後、サービス１４０を起動する機能を実現するためのプログラムである。サービス測定・起動プログラム１２１がフラッシュメモリ１２からRAM１４に読み込まれることによりサービス測定・起動部１４１が起動する。起動したサービス測定・起動部１４１は、サービス測定・起動プログラム１２１で規定されている命令をCPU１１に与えることにより、上記の機能を実現する。
【００２５】
カーネルプログラム１２２は、メモリ管理、プロセス管理、デバイス管理等のOSとしての基本機能を実現すると共に、本実施形態ではサービス測定・起動部１４１を起動し、サービス測定・起動部１４１によって行われる処理を制御する機能を実現するためのプログラムである。カーネルプログラム１２２がフラッシュメモリ１２からRAM１４に読み込まれることによりカーネル１４２が起動する。起動したカーネル１４２は、カーネルプログ

ラム１２２で規定されている命令をCPU１１に与えることにより、上記の機能を実現する。

【００２６】

サービスプログラム１２０、サービス測定・起動プログラム１２１、およびカーネルプログラム１２２はOSプログラムを構成する。また、RAM１４上で起動したサービス１４０、サービス測定・起動部１４１、およびカーネル１４２はOSを構成する。サービスプログラム１２０、サービス測定・起動プログラム１２１、およびカーネルプログラム１２２は、圧縮された状態でフラッシュメモリ１２に格納されている。

【００２７】

第１ブートローダプログラム１２３は、サービス測定・起動プログラム１２１を測定し、測定値（第２の測定値）を生成すると共に、カーネルプログラム１２２を測定し、測定値（第３の測定値）を生成した後、カーネル１４２を起動する機能を実現するためのプログラムである。第１ブートローダプログラム１２３がフラッシュメモリ１２からRAM１４に読み込まれることにより第１ブートローダ１４３が起動する。起動した第１ブートローダ１４３は、第１ブートローダプログラム１２３で規定されている命令をCPU１１に与えることにより、上記の機能を実現する。本実施形態の第１ブートローダプログラム１２３は書き換え可能なプログラムであり、ブートローダの機能の更新に対応することが可能である。

【００２８】

第２ブートローダプログラム１３０は、第１ブートローダプログラム１２３を測定し、測定値（第４の測定値）を生成した後、第１ブートローダ１４３を起動する機能を実現するためのプログラムである。第２ブートローダプログラム１３０がROM１３からRAM１４に読み込まれることにより第２ブートローダ１４４が起動する。起動した第２ブートローダ１４４は、第２ブートローダプログラム１３０で規定されている命令をCPU１１に与えることにより、上記の機能を実現する。本実施形態の第２ブートローダプログラム１３０は書き換え不可能なプログラムである。

【００２９】

第１ブートローダ１４３および第２ブートローダ１４４はブートローダを構成する。本実施形態では第１ブートローダ１４３および第２ブートローダ１４４が使用されるが、第１ブートローダ１４３を設けず、第２ブートローダ１４４にその機能を持たせてもよい。この場合、ブートローダの全体が書き換え不可能となる。

【００３０】

TPM１５は、記憶部１５０、比較部１５１、および署名部１５２を有する。記憶部１５０は、第１ブートローダプログラム１２３を測定したときに測定結果として期待される期待値（第４の期待値）を記憶する。比較部１５１は、第２ブートローダ１４４が第１ブートローダプログラム１２３を測定して得られた測定値と、記憶部１５０に格納されている期待値とを比較することにより第１ブートローダ１４３の完全性を検証する。

【００３１】

署名部１５２は、第１ブートローダ１４３がサービス測定・起動プログラム１２１を測定して得られる測定値と、第１ブートローダ１４３がカーネルプログラム１２２を測定して得られる測定値と、サービス測定・起動部１４１がサービスプログラム１２０を測定して得られる測定値とのそれぞれに対して電子署名を付加した署名ファイルを生成する。生成された署名ファイルはサーバ２へ送信される。

【００３２】

CPU１１は、RAM１４に読み込まれて起動したサービス１４０、サービス測定・起動部１４１、カーネル１４２、第１ブートローダ１４３、および第２ブートローダ１４４から与えられる各種命令を実行することによって、それらの機能に対応する処理を実行する処理部として機能する。図２はCPU１１の機能構成を模式的に示している。CPU１１は、サービス１４０の機能に対応するサービス処理部１１０、サービス測定・起動部１４１の機能に対応するサービス測定・起動処理部１１１、カーネル１４２の機能に対応するカーネル処

理部１１２、第１ブートローダ１４３の機能に対応する第１ブートローダ処理部１１３、および第２ブートローダ１４４の機能に対応する第２ブートローダ処理部１１４として機能する。

【００３３】

本実施形態では、第１ブートローダ１４３の完全性の検証は端末装置１で行われる。しかし、サービス１４０、サービス測定・起動部１４１、およびカーネル１４２を含むOSの完全性の検証は、測定値と期待値との比較に時間を要するため、サーバ２で行われる。

【００３４】

図３は、本実施形態によるサーバ２の構成を示している。サーバ２は、通信部２０（受信部）、検証部２１、および記憶部２２を有する。通信部２０は、端末装置１と通信を行う通信回路を有する。検証部２１は、端末装置１から受信された署名ファイルに基づいて、測定値が改竄されているか否かを確認し、さらに、署名ファイルに含まれる測定値と、記憶部２２に格納されている期待値とを比較することにより、端末装置１におけるOSの完全性を検証する。記憶部２２は、サービスプログラム１２０を測定したときに測定結果として期待される期待値（第１の期待値）と、サービス測定・起動プログラム１２１を測定したときに測定結果として期待される期待値（第２の期待値）と、カーネルプログラム１２２を測定したときに測定結果として期待される期待値（第３の期待値）と、検証部２１が端末装置１におけるOSの完全性を検証した結果とを記憶する。サーバ２の完全性は予め保証されているものとする。

【００３５】

次に、本実施形態におけるOSのファイルシステムを説明する。図４は、フラッシュメモリ１２上に構築されているOSのファイルシステムを示している。なお、図４は、端末装置１の起動時の状態を示している。「*」はルートディレクトリを示している。“[”と“] ”で囲まれた文字列はディレクトリ（フォルダ）を示している。ルートディレクトリには、ディレクトリである/data、/proc、/sbin、/sys、/systemのほか、プログラムである/init、ファイルであるdefault.prop、init.goldfish.rc、init.rc、tcg-scan.conf等がある。ディレクトリにはファイルをマウント（格納）することが可能であり、例えば/sbinにはadbというファイルがマウントされている。/data、/proc、/sys、/systemには端末装置１の起動時にファイルはマウントされていない。

【００３６】

initは、カーネル１４２が最初に実行するプログラムである。initが処理を実行する順番は、init.rcに規定されている。init.rcには、サービスプログラム１２０を起動して初期化する従来の処理に対して、サービスプログラム１２０を測定する処理が追加されている。また、initがinit.rcに従ってサービスプログラム１２０を測定する際の条件はtcg-scan.confに規定されている。tcg-scan.confには、例えば測定するファイルのパス、測定対象となるエントリーの名称のマッチングパターン、エントリーのタイプ（ファイル、ディレクトリ、ノード、リンク、デバイス等）等が含まれる。

【００３７】

サービス測定・起動プログラム１２１は、少なくともinit、init.rc、tcg-scan.confを含んでいる。また、サービスプログラム１２０は、少なくとも/dataおよび/systemにマウントされるシステムファイルを含んでいる。上述したように、端末装置１の起動時に/dataおよび/systemにはファイルがマウントされていないため、サービスプログラム１２０を構成するシステムファイルが/dataおよび/systemにマウントされた後、サービスプログラム１２０が測定される。なお、カーネルプログラム１２２はファイルシステムに含まれていない。

【００３８】

次に、本実施形態による端末装置１の起動時の動作を説明する。図５および図６は端末装置１の起動時の動作を示している。図５はOSおよびブートローダの動作を示し、図６はTPM１５の動作を示している。RAM１４に読み込まれて起動した各プログラムに従ってCPU１１が処理を行うことによって、図４に示す各処理が行われるが、以下では、RAM１４上

で起動した各プログラムを処理の主体として説明を行う。

【００３９】

端末装置１の電源が投入され、起動が指示されると、図５に示すように、第２ブートローダプログラム１３０がROM１３からRAM１４に読み込まれ、第２ブートローダ１４４が起動する（ステップＳ１００）。第２ブートローダ１４４は、フラッシュメモリ１２からRAM１４に第１ブートローダプログラム１２３を読み込み、第１ブートローダプログラム１２３を測定する（ステップＳ１０５）。第１ブートローダプログラム１２３の測定値はTPM１５へ出力される。

【００４０】

続いて、第２ブートローダ１４４はフラッシュメモリ１２からRAM１４に第１ブートローダプログラム１２３を読み込み、第１ブートローダ１４３を起動する（ステップＳ１１０）。起動した第１ブートローダ１４３は、TPM１５から指示が出力されるのを待つ。

【００４１】

図６に示すように、TPM１５の比較部１５１は、第１ブートローダプログラム１２３の測定値が入力されると、第１ブートローダプログラム１２３の期待値を記憶部１５０から読み出す。比較部１５１は、第１ブートローダプログラム１２３の測定値と期待値とを比較する（ステップＳ２００）。

【００４２】

続いて、比較部１５１は、第１ブートローダプログラム１２３の測定値と期待値とが一致するか否かを判定する（ステップＳ２０５）。第１ブートローダプログラム１２３の測定値と期待値とが一致しなかった場合、比較部１５１は終了指示を出力し（ステップＳ２３５）、TPM１５は処理を終了する。この場合、第１ブートローダ１４３の完全性が確認できなかったことになる。

【００４３】

また、第１ブートローダプログラム１２３の測定値と期待値とが一致した場合、比較部１５１は起動指示を出力する（ステップＳ２１０）。この場合、第１ブートローダ１４３４の完全性が確認できたことになる。続いて、署名部１５２は、TPM１５に測定値が入力されるのを待つ。

【００４４】

図５に示すように、第１ブートローダ１４３は、TPM１５から出力された指示の内容を判定する（ステップＳ１１５）。TPM１５から出力された指示が終了指示であった場合、第１ブートローダ１４３は処理を終了する。この場合、端末装置１の起動は中止される。また、TPM１５から出力された指示が起動指示であった場合、第１ブートローダ１４３は、圧縮されたカーネルプログラム１２２をフラッシュメモリ１２から読み出して測定しながら、カーネルプログラム１２２をRAM１４に展開（伸張）する。また、第１ブートローダ１４３は、圧縮されたサービス測定・起動プログラム１２１をフラッシュメモリ１２から読み出して測定しながら、サービス測定・起動プログラム１２１をRAM１４に展開する（ステップＳ１２０）。カーネルプログラム１２２およびサービス測定・起動プログラム１２１のそれぞれの測定値はTPM１５へ出力される。

【００４５】

続いて、第１ブートローダ１４３は、圧縮されたカーネルプログラム１２２をフラッシュメモリ１２から再度読み出してRAM１４に展開し、カーネル１４２を起動する。また、第１ブートローダ１４３は、圧縮されたサービス測定・起動プログラム１２１をフラッシュメモリ１２から再度読み出してRAM１４に展開し、サービス測定・起動部１４１を起動する（ステップＳ１２５）。

【００４６】

続いて、起動したカーネル１４２が、前述したinitを起動することにより、サービス測定・起動部１４１を起動する（ステップＳ１３０）。起動したサービス測定・起動部１４１のinitは、init.rcで規定されている手順に従って処理を行う。本実施形態のinit.rcでは、サービスプログラム１２０を構成するシステムファイルのマウント、サービスプログ

ラム１２０の測定、サービス１４０の起動の順に処理を行うことが規定されている。したがって、サービス測定・起動部１４１のinitは、圧縮されたサービスプログラム１２０をフラッシュメモリ１２から読み出して展開し、サービスプログラム１２０を構成するシステムファイルをフラッシュメモリ１２中の/dataと/systemにマウントする（ステップＳ１３５）。

【００４７】

続いて、サービス測定・起動部１４１のinitは、tcg-scan.confに従って、/dataと/systemにマウントされたサービスプログラム１２０をフラッシュメモリ１２からRAM１４に読み込み、サービスプログラム１２０を測定する（ステップＳ１４０）。ステップＳ１４０では、サービスプログラム１２０を構成するシステムファイルに含まれるファイル毎に測定が行われ、ファイル毎に測定値が算出される。各ファイルの測定値は１つのファイルにまとめられ、サービスプログラム１２０の測定値を記録したファイルとしてフラッシュメモリ１２に格納される。

【００４８】

続いて、サービス測定・起動部１４１のinitは、/dataと/systemにマウントされたサービスプログラム１２０をフラッシュメモリ１２からRAM１４に読み込み、サービス１４０を起動する（ステップＳ１４５）。起動したサービス１４０は、通信に関する機能が起動した時点で、TPM１５から指示が出力されるのを待つ。

【００４９】

図６に示すように、TPM１５の署名部１５２は、比較部１５１から起動指示が出力された後、カーネルプログラム１２２の測定値が入力されると、この測定値と電子署名とを含む署名ファイルを生成し、フラッシュメモリ１２に格納する（ステップＳ２１５）。この署名ファイルに含まれる電子署名は、カーネルプログラム１２２の測定値のハッシュ値を、TPM１５が保持する秘密鍵で暗号化した情報である。署名ファイルの生成に使用される秘密鍵は、例えば記憶部１５０に格納されている。

【００５０】

続いて、署名部１５２は、サービス測定・起動プログラム１２１の測定値が入力されると、この測定値と電子署名とを含む署名ファイルを生成し、フラッシュメモリ１２に格納する（ステップＳ２２０）。この署名ファイルに含まれる電子署名は、サービス測定・起動プログラム１２１の測定値のハッシュ値を、TPM１５が保持する秘密鍵で暗号化した情報である。

【００５１】

続いて、署名部１５２は、サービスプログラム１２０の測定値を含むファイルをフラッシュメモリ１２から読み出し、この測定値と電子署名とを含む署名ファイルを生成し、フラッシュメモリ１２に格納する（ステップＳ２２０）。この署名ファイルに含まれる電子署名は、サービスプログラム１２０の測定値のハッシュ値を、TPM１５が保持する秘密鍵で暗号化した情報である。

【００５２】

続いて、署名部１５２は、サービス１４０に署名ファイルの送信指示を出力し（ステップＳ２３０）、TPM１５は処理を終了する。

【００５３】

図５に示すように、サービス１４０は、TPM１５から署名ファイルの送信指示が出力されたか否かを判定する（ステップＳ１５０）。署名ファイルの送信指示が出力されていない場合、サービス１４０は、署名ファイルの送信指示が出力されるまで待つ。また、署名ファイルの送信指示が出力された場合、サービス１４０は、フラッシュメモリ１２から署名ファイルを読み出して通信部１０へ出力し、署名ファイルをサーバ２へ送信させる（ステップＳ２２５）。この署名ファイルは、カーネルプログラム１２２の測定値を含む署名ファイルと、サービス測定・起動プログラム１２１の測定値を含む署名ファイルと、サービスプログラム１２０の測定値を含む署名ファイルとからなる。署名ファイルは、これらの各プログラムの測定値と電子署名とを含む１つのファイルであってもよい。署名ファイ

ルの送信後、サービス１４０は、端末装置１の起動に関する処理を完了し、起動中の処理を行う。

【００５４】

第１ブートローダプログラム１２３の書き換えは、必要に応じて、サービス１４０が起動した後に行われる。第１ブートローダプログラム１２３の書き換えが行われた直後に、第１ブートローダプログラム１２３の期待値が算出される。算出された期待値はTPM１５へ出力され、TPM１５の記憶部１５０に格納されている期待値が、TPM１５に入力された期待値に更新される。第１ブートローダプログラム１２３の書き換えが失敗する可能性を想定して、以下のようにしてもよい。例えば、初期状態の第１ブートローダプログラム１２３がROM１３に予め格納されており、TPM１５の比較部１５１は、第１ブートローダプログラム１２３の測定値と期待値とが一致しなかったと判定した場合に、初期状態の第１ブートローダプログラム１２３を使用した起動の指示を出力する。第２ブートローダ１４４は、この起動指示に基づいて、ROM１３からRAM１４に初期状態の第１ブートローダプログラム１２３を読み込み、第１ブートローダ１４３を起動する。起動した第１ブートローダ１４３はステップＳ１２０の処理を行う。

【００５５】

次に、本実施形態によるサーバ２の動作を説明する。図７はサーバ２の動作を示している。

【００５６】

通信部２０は端末装置１から署名ファイルを受信し、検証部２１へ出力する（ステップＳ３００）。検証部２１は、署名ファイルに含まれている電子署名を検証する（ステップＳ３０５）。このとき、検証部２１は、記憶部２２から公開鍵を読み出し、署名ファイルに含まれている電子署名を公開鍵で復号し、得られたハッシュ値と、署名ファイルに含まれている測定値から算出したハッシュ値とが一致するか否かを確認する。両者が一致すれば、署名ファイルの内容が改竄されていないことが保証される。

【００５７】

サーバ２が有する公開鍵は、端末装置１のTPM１５が有する秘密鍵と対になる鍵である。端末装置１の起動時に通信が可能になった時点で、署名ファイルとは別々に、あるいは署名ファイルと一緒に、公開鍵が端末装置１からサーバ２へ送信される。

【００５８】

署名ファイルの内容が改竄されていなければ、検証部２１は、カーネルプログラム１２２の期待値を記憶部２２から読み出し、署名ファイルに含まれるカーネルプログラム１２２の測定値と、記憶部２２から読み出した期待値とを比較する（ステップＳ３１０）。続いて、検証部２１は、サービス測定・起動プログラム１２１の期待値を記憶部２２から読み出し、署名ファイルに含まれるサービス測定・起動プログラム１２１の測定値と、記憶部２２から読み出した期待値とを比較する（ステップＳ３１５）。さらに、検証部２１は、サービスプログラム１２０の期待値を記憶部２２から読み出し、署名ファイルに含まれるサービスプログラム１２０の測定値と、記憶部２２から読み出した期待値とを比較する（ステップＳ３２０）。サービスプログラム１２０の期待値は、サービスプログラム１２０を構成するシステムファイルに含まれるファイル毎の期待値からなる。

【００５９】

続いて、検証部２１は、ステップＳ３１０，Ｓ３１５，Ｓ３２０の処理の結果に基づいて、端末装置１のOSが完全であるか否かを判定する（ステップＳ３２５）。ステップＳ３１０，Ｓ３１５，Ｓ３２０の全てにおいて、全てのファイルの測定値と期待値とが完全に一致した場合、端末装置１のOSが完全であると判定される。また、ステップＳ３１０，Ｓ３１５，Ｓ３２０の少なくともいずれかにおいて、いずれかのファイルの測定値と期待値とが一致しなかった場合、端末装置１のOSが完全でないと判定される。

【００６０】

端末装置１のOSが完全であると判定された場合、検証部２１は、端末装置１のOSが完全であることを示す情報を生成し、端末装置１の識別情報（MACアドレスやTPM１５のID等）

と関連付けて記憶部２２に記録する（ステップＳ３３０）。また、端末装置１のOSが完全でないと判定された場合、検証部２１は、端末装置１のOSが完全でないことを示す情報を生成し、端末装置１の識別情報と関連付けて記憶部２２に記録する（ステップＳ３３５）。ステップＳ３３０，Ｓ３３５で記憶部２２に記録された情報を参照することにより、端末装置１が完全な状態で起動したのか否かを後から確認することができる。ステップＳ３３０，Ｓ３３５のいずれかの処理が終了すると、サーバ２は、端末装置１のOSの完全性を検証する処理を終了する。

【００６１】

本実施形態では、以下の変形が可能である。ステップＳ１１５の判定はステップＳ１１０の処理が行われる前に第２ブートローダ１４４によって行われてもよい。また、ステップＳ２１５，Ｓ２２０，Ｓ２２５の各処理が行われる順番は、図６に示す順番でなくてもよい。また、ステップＳ３１０，Ｓ３１５，Ｓ３２０の各処理が行われる順番は、図７に示す順番でなくてもよい。

【００６２】

上記の説明では、サービス測定・起動部１４１のinitがサービス測定・起動プログラム１２１およびカーネルプログラム１２２の測定を行っているが、サービス測定・起動部１４１のinitが測定用プログラムを実行し、その測定用プログラムがサービス測定・起動プログラム１２１およびカーネルプログラム１２２の測定を行ってもよい。この場合、サービス測定・起動プログラム１２１は測定用プログラムを含む。

【００６３】

上記の説明では、第２ブートローダ１４４が、圧縮されたサービス測定・起動プログラム１２１およびカーネルプログラム１２２の測定および展開を行っているが、第２ブートローダ１４４が、圧縮されたサービス測定・起動プログラム１２１およびカーネルプログラム１２２の測定を行い、カーネル１４２が起動した後、カーネル１４２が、圧縮されたサービス測定・起動プログラム１２１およびカーネルプログラム１２２の展開を行ってもよい。

【００６４】

上記の説明では、サービス測定・起動プログラム１２１およびカーネルプログラム１２２が分離しているが、両者が一体化されていてもよい。

【００６５】

上記の説明では、第１ブートローダプログラム１２３および第２ブートローダプログラム１３０が分離しているが、両者が一体化され、そのプログラムがROM１３に格納されていてもよい。

【００６６】

次に、端末装置１をより高速に起動することができる理由を説明する。本実施形態では、サービス測定・起動部１４１のinitがサービスプログラム１２０の測定を行うことにより、ブートローダがサービスプログラム１２０の測定を行う場合よりも測定時間を短縮し、その結果、端末装置１をより高速に起動することができる。

【００６７】

カーネル１４２が起動する前、ブートローダはカーネル１４２によって提供されるファイルのオープン、ファイルのクローズ、スレッドの生成等の機能を使用することができない。サービスプログラム１２０を構成するシステムファイルの測定はファイル単位で行われるため、ブートローダがサービスプログラム１２０の測定を行うためにはブートローダがファイルを扱えるようにする必要がある。しかし、ブートローダにファイルのオープンやクローズを扱う機能を追加するとコードサイズが大きくなるので、その機能を追加する代わりにブートローダにファイルキャッシュ（ページキャッシュ）の機能を実装しないことが考えられる。

【００６８】

ファイルキャッシュの機能がないと、ファイルをオープンするたびに、ファイルシステムの最上位位置にあるルートディレクトリから、開くファイルがマウントされているディ

レクトリまでディレクトリの検索を行うためにデバイス（本実施形態ではフラッシュメモリ１２）にアクセスする必要があり、ファイルのオープンに要する時間が長くなる。また、ファイルの先読み機能も実装されないことが考えられ、ファイルを読み込むためにデバイスにアクセスし、多くの待ち時間が発生する。さらに、Ｌ２キャッシュ（２次キャッシュ）のハンドリングが面倒であるため、Ｌ２キャッシュを無効にすることが考えられる。これにより、Ｌ１キャッシュ（１次キャッシュ）でのキャッシュミスによるミスペナルティが発生し、処理速度が低下する。

【００６９】

また、前述したように、ブートローダはシングルタスク／シングルスレッドで動作するため、複数のCPUコアが搭載されている場合でも１つのCPUコアしか使用することができない。

【００７０】

一方、サービス測定・起動部１４１のinitが起動したとき、カーネル１４２が起動しているので、サービス測定・起動部１４１のinitは、カーネル１４２によって提供されるファイルのオープン、ファイルのクローズ、スレッドの生成等の機能を使用することができる。これにより、サービス測定・起動部１４１のinitは、マルチタスク／マルチスレッドで動作することができ、測定対象のファイルの読み込みと測定とを並行的に行うことができる。また、ファイルキャッシュの機能により、開くファイルがマウントされているディレクトリの検索を行う際に同じディレクトリを繰り返し無駄に読み込む回数が減る。さらに、ファイルの先読み機能により、デバイスにアクセスする回数が減る。さらに、Ｌ２キャッシュを有効にすることにより、Ｌ１キャッシュでのキャッシュミスによるミスペナルティが減少する。

【００７１】

また、複数のCPUコアが搭載されている場合には、複数のスレッドのそれぞれにCPUコアを割り当てて測定を並行的に行うことができる。

【００７２】

以上のように、サービス測定・起動部１４１のinitがカーネル１４２の各種機能を使用することによって、サービスプログラム１２０の測定に要する時間を短縮し、その結果、端末装置１をより高速に起動することができる。

【００７３】

次に、本実施形態の完全性検証システムを適用した例を説明する。個人が所有している端末装置を会社に持ち込むBring Your Own Device（BYOD）が進んでいる。システム領域が完全である、つまり、マルウエア感染や社員による不正改造の影響を受けていない安全な端末装置であることを検証した後に端末装置を会社のネットワークに接続させる検疫サービスに対して、本実施形態の完全性検証システムを適用することが可能である。例えば、サーバ２は会社の管理サーバであり、端末装置１のOSが完全であることが確認できた場合のみ、端末装置１を会社のネットワークに接続させる。

【００７４】

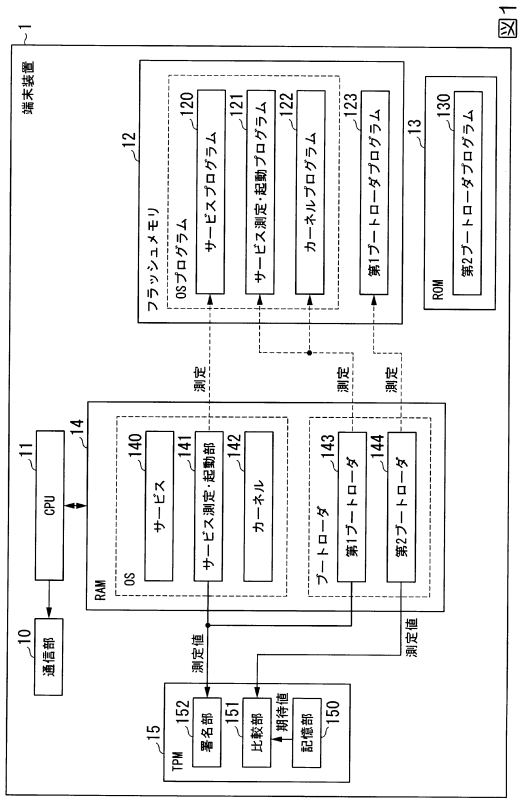
また、銀行決済アプリケーションを利用する際に、端末装置１の完全性の確認を行うようにしてもよい。例えば、端末装置１に銀行決済アプリケーションがインストールされており、銀行決済アプリケーションが銀行のサーバに決済の要求を行った際に、銀行のサーバがサーバ２に端末装置１の完全性の確認を求める。サーバ２は、記憶部２２に格納されている情報に基づいて、端末装置１のOSが完全であるか否かを確認し、確認結果を銀行のサーバに通知する。銀行のサーバは、端末装置１のOSが完全であることが確認できた場合のみ、決済の処理を行う。あるいは、サーバ２が銀行のサーバであってもよい。

【００７５】

上述したように、本実施形態によれば、端末装置１の起動時にサーバ２が端末装置１のOSの完全性を検証することができる。また、カーネル１４２による制御に従って、サービス測定・起動部１４１がサービスプログラム１２０を測定することによって、端末装置１をより高速に起動することができる。

【００７６】
また、サービス１４０の初期化を行う初期化プログラム（init）に対してサービスプログラム１２０の測定を行う機能を追加してサービス測定・起動プログラム１２１を構成することによって、サービス測定・起動プログラム１２１の実装が容易になる。
【００７７】
以上、図面を参照して本発明の実施形態について詳述してきたが、具体的な構成は上記の実施形態に限られるものではなく、本発明の要旨を逸脱しない範囲の設計変更等も含まれる。本実施形態では、測定値および期待値にハッシュ値を用いているが、不可逆な値であれば、ハッシュ値の代わりに用いることが可能である。また、本実施形態では、端末装置１にOSとしてAndroid（登録商標）が搭載されているが、端末装置のディスク領域（フラッシュメモリやハードディスクドライブ等の領域）において、OSのシステムファイルが保存されるシステム領域と、ユーザが任意にファイルを追加することが可能なユーザ領域とが分離していればよく、他のOSが端末装置１に搭載されていてもよい。
【符号の説明】
【００７８】
１　端末装置、２　サーバ、１０，２０　通信部、１１　CPU、１２　RAM、１３　フラッシュメモリ、１４　ROM、１５　TPM、２１　検証部、２２，１５０　記憶部、１１０　サービス処理部、１１１　サービス測定・起動処理部、１１２　カーネル処理部、１１３　第１ブートローダ処理部、１１４　第２ブートローダ処理部、１２０　サービスプログラム、１２１　サービス測定・起動プログラム、１２２　カーネルプログラム、１２３　第１ブートローダプログラム、１３０　第２ブートローダプログラム、１４０　サービス、１４１　サービス測定・起動部、１４２　カーネル、１４３　第１ブートローダ、１４４　第２ブートローダ、１５１　比較部、１５２　署名部

【図１】



【図２】

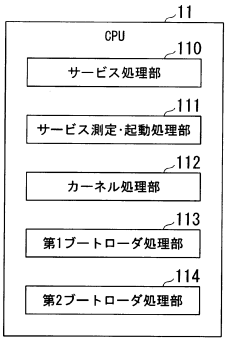


図2

【図３】

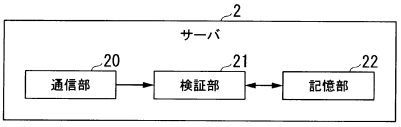


図3

【図４】

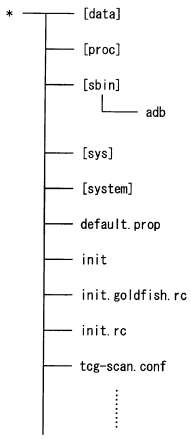


図4

【図５】

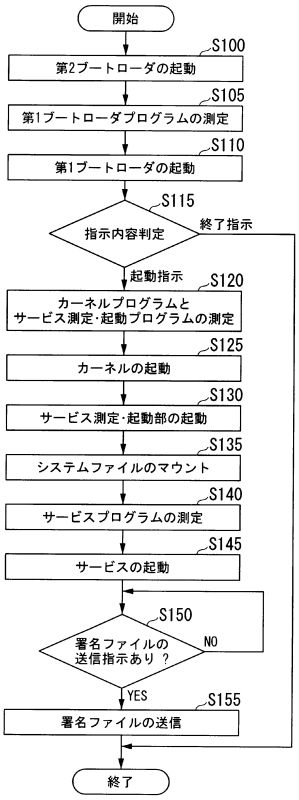


図5

【図６】

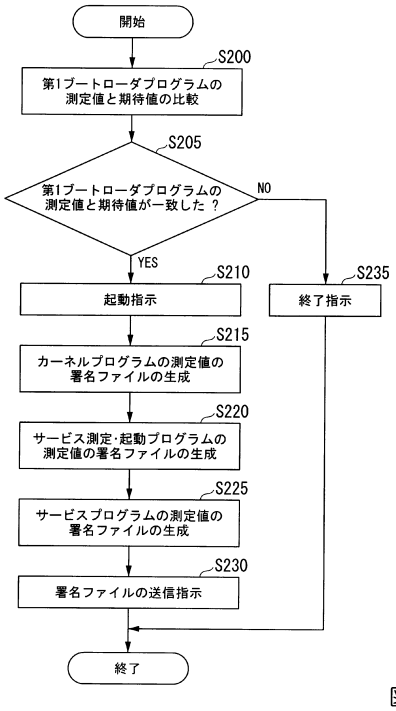


図6

【図７】

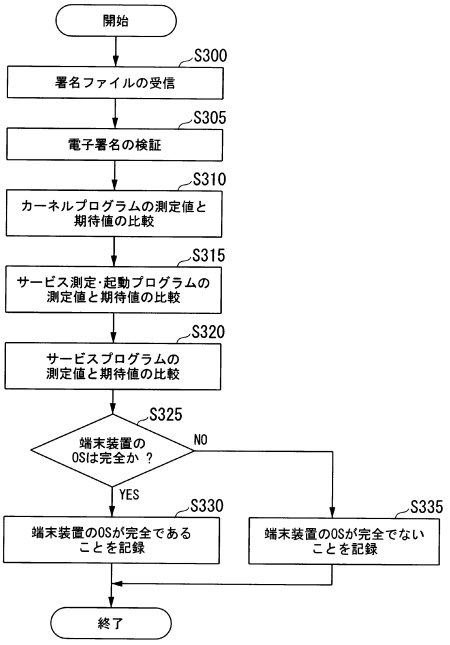


図7

フロントページの続き

(72)発明者　磯原　隆将
埼玉県ふじみ野市大原２丁目１番１５号　株式会社ＫＤＤＩ研究所内
(72)発明者　窪田　歩
埼玉県ふじみ野市大原２丁目１番１５号　株式会社ＫＤＤＩ研究所内

審査官　青木　重徳

(56)参考文献　特開２００９－１２９０６１（ＪＰ，Ａ）
特開２００８－１６５７５８（ＪＰ，Ａ）
特開２００４－２６５２８６（ＪＰ，Ａ）
国際公開第２００８／００４５２５（WO，Ａ１）
竹森　敬祐、他，カーネル保護機能を利用したファイル完全性リモート検証機構，情報処理学会研究報告，日本，社団法人情報処理学会，２００９年　2月２６日，Ｖｏｌ．２００９，Ｎｏ．２０，ｐ．１－６

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５７
Ｈ０４Ｌ　９／１０
Ｈ０４Ｌ　９／３２