(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2015-72683
(P2015-72683A)
(43)公開日　平成27年4月16日(2015.4.16)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/57	(2013.01)	G06F	21/57	350	5J104
H04L 9/32	(2006.01)	H04L	9/00	675D	
G06F 21/60	(2013.01)	H04L	9/00	675B	
		G06F	21/60	320	

審査請求 有　請求項の数 28　ＯＬ　外国語出願　（全 39 頁）

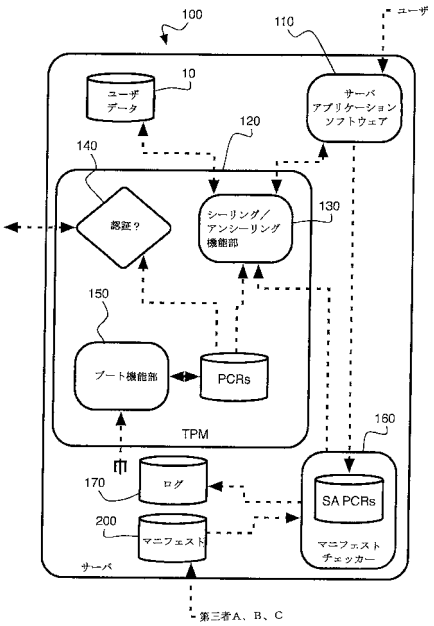
(21)出願番号　特願2014-189808 (P2014-189808)
(22)出願日　平成26年9月18日 (2014.9.18)
(31)優先権主張番号　61/882,321
(32)優先日　平成25年9月25日 (2013.9.25)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　14/283,383
(32)優先日　平成26年5月21日 (2014.5.21)
(33)優先権主張国　米国 (US)

(71)出願人　314016122
マックス　プランク　ゲゼルシャフト　ツール　フォーデルング　デル　ヴィッセンシャフテン
ドイツ連邦共和国　８０５３９　ミュンヘン　ホフガルテンシュトラーセ　８
(74)代理人　100083286
弁理士　三浦　邦夫
(74)代理人　100166408
弁理士　三浦　邦陽
(72)発明者　ポール・フランシス
ドイツ連邦共和国　６７６６１　カイザースラウテルン－ダンゼンベルク　ブーヒェンヘッケン・シュトラーセ　２３

最終頁に続く

(54)【発明の名称】匿名データの第三者の監視を実行するためのシステム及び方法

(57)【要約】
【課題】データ（個人的なまたは保護されたデータ）の第三者の監視を実行するためのシステム及び方法を得る。
【解決手段】改変可能なサーバは、データのシーリング／アンシーリングのための改変可能なサンドボックス要素および該サンドボックス要素の許可・不許可を決める改変不能なチェッカー要素を構築することにより、サーバの測定値（測定結果）に従って、データを信頼性高くシーリング／アンシーリングする。チェッカー要素は、サンドボックス要素が予め定めた標準値に適合しているか否かを判定する。サンドボックス要素が適合していれば、チェッカー要素は、該チェッカー要素の測定値を使用したデータのシーリング／アンシーリングをサンドボックス要素に許可する。サンドボックス要素が適合していなければ、チェッカー要素は、該チェッカー要素の測定値を使用したデータのシーリング／アンシーリングをサンドボックス要素に許可しない。
【選択図】図１



【特許請求の範囲】
【請求項１】
ユーザデータのシーリングとアンシーリングを行うためのサーバであって、
プロセッサとメモリデバイスを有すること；
前記メモリデバイスは、暗号キーと、前記暗号キーにより暗号化されたユーザデータと、それぞれがサーバの認証測定値を有する複数の認証マニフェストと、マニフェストチェッカー要素とサンドボックス要素を有するサーバアプリケーションとを有すること；
前記暗号キーは、前記マニフェストチェッカー要素の測定値を使用して第１のメモリデバイスの内部にシーリングされること；
前記プロセッサは、前記サーバアプリケーションによって、前記マニフェストチェッカー要素を実行することで、サーバの認証測定値に対応するサーバの現在測定値を取得し、サーバの現在測定値と認証測定値とを比較し、且つ、サーバの測定値は、前記マニフェストチェッカー要素のいかなる測定値をも含んでいないこと；及び
前記プロセッサは、サーバの現在測定値と認証測定値が十分に一致しているときには、前記サーバアプリケーションによって、前記サンドボックス要素を実行することで、前記マニフェストチェッカー要素の現在測定値を使用して前記暗号キーをアンシーリングし、アンシーリングした前記暗号キーを使用してユーザデータにアクセスすること；又は
前記プロセッサは、サーバの現在測定値と認証測定値が十分に一致していないときには、前記サーバアプリケーションによって、前記サンドボックス要素を実行しないこと；
を特徴とするサーバ。
【請求項２】
請求項１記載のサーバにおいて、
前記複数の認証マニフェストは、それぞれ、該複数の認証マニフェストとは別の独立した認証部によって認証されるサーバ。
【請求項３】
請求項１記載のサーバにおいて、
前記マニフェストチェッカー要素は、前記プロセッサに、前記メモリデバイスに記憶された複数の認証部に対して前記認証マニフェストを認証させるサーバ。
【請求項４】
請求項３記載のサーバにおいて、
前記複数の認証部は、前記サーバアプリケーションの内部に符号化されているサーバ。
【請求項５】
請求項４記載のサーバにおいて、
前記マニフェストチェッカー要素は、前記複数の認証部を含んでいるサーバ。
【請求項６】
請求項３記載のサーバにおいて、
前記マニフェストチェッカー要素は、前記メモリデバイスに記憶された前記複数の認証部に対して前記認証マニフェストの全てが認証されたとき、及び、前記認証測定値の全てが前記現在測定値と完全に一致しているときにのみ、前記プロセッサに、サーバの現在測定値と認証測定値が十分に一致していると判定させるサーバ。
【請求項７】
請求項３記載のサーバにおいて、
前記マニフェストチェッカー要素は、前記メモリデバイスに記憶された前記複数の認証部に対して前記認証マニフェストの少なくとも一部を残した大部分が認証されたとき、及び、前記認証マニフェストの少なくとも一部を残した大部分が前記現在測定値と一致する前記認証測定値を含んでいるときにのみ、前記プロセッサに、サーバの現在測定値と認証測定値が十分に一致していると判定させるサーバ。
【請求項８】
請求項３記載のサーバにおいて、
前記マニフェストチェッカー要素は、前記メモリデバイスに記憶された前記複数の認証

部に対して前記認証マニフェストの少なくとも主要部分が認証されたとき、及び、前記認証マニフェストの少なくとも主要部分が前記現在測定値と一致する前記認証測定値を含んでいるときにのみ、前記プロセッサに、サーバの現在測定値と認証測定値が十分に一致していると判定させるサーバ。

【請求項９】

請求項３記載のサーバにおいて、

前記マニフェストチェッカー要素は、前記認証マニフェストの少なくとも主要部分が前記現在測定値と一致する前記認証測定値を含んでいるときにのみ、前記プロセッサに、サーバの現在測定値と認証測定値が十分に一致していると判定させるサーバ。

【請求項１０】

請求項１記載のサーバにおいて、

前記マニフェストチェッカー要素は、前記認証マニフェストの少なくとも１つが、前記現在測定値の１つと一致する無効測定値を含んでいるとき、前記プロセッサに、サーバの現在測定値と認証測定値が十分に一致していないと判定させるサーバ。

【請求項１１】

請求項１記載のサーバにおいて、

前記マニフェストチェッカー要素は、該マニフェストチェッカー要素の測定値の一部ではない第三者のリストを含んでおり、この第三者のリストは、前記認証マニフェストを第三者が認証した旨の証明を含んでいるサーバ。

【請求項１２】

請求項１１記載のサーバにおいて、

前記マニフェストチェッカー要素は、前記第三者リストの改変を許容された評価者の認証部を含んでおり、この評価者の認証部は、前記マニフェストチェッカー要素の測定値の一部であるサーバ。

【請求項１３】

請求項１記載のサーバにおいて、

前記マニフェストチェッカー要素は、当該サーバとは別のサーバの現在測定値と前記メモリデバイスに記憶されたマニフェストの認証測定値とを比較することにより、前記プロセッサに、前記別のサーバを遠隔認証させるサーバ。

【請求項１４】

請求項１記載のサーバにおいて、

前記マニフェストチェッカー要素は、認証データマニフェストがサーバアプリケーションの現在測定値と一致しているときにのみ、前記プロセッサに、入力したユーザデータを認証データによりチェックさせるとともに、認証データにより認証されたデータをアンシーリングさせるサーバ。

【請求項１５】

請求項１４記載のサーバにおいて、

前記マニフェストチェッカー要素は、前記プロセッサに、認証マニフェストにより認証された第三者のリストのための認証データをチェックさせるサーバ。

【請求項１６】

請求項１５記載のサーバにおいて、

前記マニフェストチェッカー要素は、サーバの現在測定値と認証測定値が十分に一致しているか否かを判定するための基準を決めるために、前記プロセッサに前記認証データをチェックさせるサーバ。

【請求項１７】

改変可能なサーバを使用して、当該サーバの測定値に従って、データを信頼性高くシーリング及びアンシーリングするための方法であって、

前記サーバに、データのシーリング及びアンシーリングのための改変可能なサンドボックス要素と、このサンドボックス要素の許可又は不許可を決める改変不能なチェッカー要素とを構築するステップ；

前記チェッカー要素を介して、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定するステップ；及び
前記サンドボックス要素が予め定めた標準値に適合していると前記チェッカー要素が判定したときに、チェッカー要素の測定値を使用したデータのシーリング／アンシーリングを前記サンドボックス要素に許可するステップ；又は
前記サンドボックス要素が予め定めた標準値に適合していないと前記チェッカー要素が判定したときに、チェッカー要素の測定値を使用したデータのシーリング／アンシーリングを前記サンドボックス要素に許可しないステップ；
を有することを特徴とする方法。
【請求項１８】
請求項１７記載の方法において、
前記チェッカー要素は、前記サンドボックス要素の現在測定値と、第三者により認証されたマニフェストに含まれる認証測定値とを比較することにより、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項１９】
請求項１８記載の方法において、
前記チェッカー要素は、当該チェッカー要素内に記憶された複数の認証部に対してそれぞれのマニフェストを認証する方法。
【請求項２０】
請求項１９記載の方法において、
前記チェッカー要素の測定値は、データのシーリングとアンシーリングのために使用されるものであり、且つ、前記複数の認証部を含んでいない方法。
【請求項２１】
請求項１９記載の方法において、
前記チェッカー要素は、前記複数の認証部のうち、評価者により認証された第三者リスト内に含まれる認証部だけを受け入れる方法。
【請求項２２】
請求項１８記載の方法において、
前記チェッカー要素は、現在測定値と認証測定値が十分に一致しているか否か、メモリデバイスに記憶された複数の認証部に対して全ての認証マニフェストが認証されているときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否か、及び、現在測定値と認証測定値が完全に一致しているときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否かに基づいて、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項２３】
請求項１８記載の方法において、
前記チェッカー要素は、現在測定値と認証測定値が十分に一致しているか否か、メモリデバイスに記憶された複数の認証部に対して前記認証マニフェストの少なくとも一部を残した大部分が認証されたときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否か、及び、前記認証マニフェストの少なくとも一部を残した大部分が認証測定値を含んでいるときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否かに基づいて、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項２４】
請求項１８記載の方法において、
前記チェッカー要素は、現在測定値と認証測定値が十分に一致しているか否か、メモリデバイスに記憶された複数の認証部に対して前記認証マニフェストの少なくとも主要部分が認証されたときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否か、及び、前記認証マニフェストの少なくとも主要部分が認証測定値を含んでいるときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否かに基づいて、前記

サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項２５】
請求項１８記載の方法において、
前記チェッカー要素は、現在測定値と認証測定値が十分に一致しているか否か、及び、前記認証マニフェストの少なくとも主要部分が認証測定値を含んでいるときにだけ現在測定値と認証測定値が十分に一致していると判定されるか否かに基づいて、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項２６】
請求項１８記載の方法において、
前記チェッカー要素は、現在測定値と認証測定値が十分に一致しているか否か、前記認証マニフェストの少なくとも１つが前記現在測定値の１つと一致する無効測定値を含んでいるときに現在測定値と認証測定値が十分に一致していないと判定されるか否かに基づいて、前記サンドボックス要素が予め定めた標準値に適合しているか否かを判定する方法。
【請求項２７】
請求項１７記載の方法において、
前記チェッカー要素は、前記サンドボックス要素が、入力データに添付された認証データに対応する標準値に適合しているか否かを判定する方法。
【請求項２８】
請求項２７記載の方法において、
前記標準値は、改変可能なサンドボックス要素の認証測定値として設定されており、前記チェッカー要素は、前記認証データにより認証された１またはそれ以上の認証マニフェストから前記認証測定値を取得する方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、２０１３年９月２５日に出願された米国仮出願第６１／８８２３２１号に基づく優先権の利益を主張するものであり、当該仮出願の全体が参照として含められている。
【０００２】
本発明は、個人的なユーザデータを取り扱うサーバを信頼できる第三者機関（以下では単に「第三者」という）により監視するための方法及びシステムに関する。
【背景技術】
【０００３】
ユーザがその個人的なデータを遠隔サーバに供給する（預ける）という機会が増えてきている。サーバの運営組織（サーバ組織）では、ユーザのデータを１または他の方法（例えば個人情報の保護方針について説明した文書による）で保護することを規定している。ユーザは、サーバの運営組織（サーバ組織）を全面的に信頼することはできず、サーバの運営組織（サーバ組織）による保護の規定に応えられるような信頼できる第三者の保証を欲している。しかし今日ではこれが脆弱な手法で行われているのが実情である。例えば、第三者は、サーバソフトウェアを調査して、ソフトウェアが保護の規定を忠実に遂行している旨を証明しなければならない。しかしながら、サーバの運営組織（サーバ組織）は、そのソフトウェアを事後的に簡単に改変して保護の規定に背くことが可能である。このため、信頼できる第三者がサーバソフトウェアを制御することにより、当該第三者の認証が無い限りにおいてサーバの運営組織（サーバ組織）がそのソフトウェアを改変することができないようにしたシステム及び方法が要求されている。さらに、ユーザは、サーバに対して個人的なデータを伝送する前に、このような制御が存在している（実行されている）ことについて保証（認証）を得なければならない。
【発明の概要】
【発明が解決しようとする課題】
【０００４】

本発明の一態様では、独立した複数の第三者が、当該複数の第三者のうちの一部または全部がソフトウェアを認証しない限りにおいて、特定のサーバアプリケーションソフトウェアの動作を停止する（無効にする）ことができる。サーバに対して個人的なデータを供給している（預けている）ユーザは、この個人的なデータを独立した第三者が監視している旨の暗号証明を得ることができる。加えて、サーバは、以前に起動したことがある全てのバージョンのソフトウェアの不正開封防止ログ（tamper-proof log）を生成することができる。これにより、いずれのユーザも、過去にサーバ内で起動したことがあるソフトウェアを正確に把握することができる。このモデルによれば、サーバにデータを送る前に、サーバアプリケーションソフトウェアが当該データを安全に取り扱うことについての保証（信頼）をクライアントに与えることができる。

【図面の簡単な説明】

【０００５】

上述した本発明の特徴および利点ならびに他の特徴および態様については、以下の説明および添付図面を参照することでより良く理解することができる。

【図１】保護されたユーザデータを取り扱うためのサーバを示す概略図である。

【図２】サーバソフトウェアの第三者認証を得るための方法を示す概略図である。

【図３】サーバソフトウェアの第三者認証を確認するための方法を示す概略図である。

【図４】サーバソフトウェアの認証を受けた第三者のリストを管理するための方法を示す概略図である。

【図５】図１のサーバを経由して保護されたユーザデータにアクセス（供給または問い合わせ）するための方法を示す概略図である。

【図６】本発明の一実施形態によるサーバとクライアントとの間における遠隔認証接続モードを示す概略図である。

【図７】本発明の別実施形態によるサーバとクライアントとの間における遠隔認証接続モードを示す概略図である。

【発明を実施するための形態】

【０００６】

添付図面を参照して、本発明の典型的な実施形態について詳細に説明する。添付図面においては、同一または類似の構成要素に同一の符号を使用する。

【０００７】

図１に示すように、サーバ１００は、個人的なまたは保護されたユーザデータ１０を取り扱うためのサーバアプリケーションソフトウェア１１０を起動および実行する。サーバ１００は、その機能的な構成要素として、トラステッド・プラットフォーム・モジュール（Trusted Platform Module）（以下では「ＴＰＭ」と呼ぶ）１２０を有している。以下ではＴＰＭ１２０による機能を「ＴＰＭ機能」と呼ぶことにする。このＴＰＭ機能は、ＴＰＭハードウェアとこれに対応するマザーボードファームウェアにより実現され、またはサーバのＣＰＵの内部に実装されている。ＴＰＭ１２０は、シーリング／アンシーリング機能部（sealing and unsealing functionality）１３０と遠隔認証機能部（remote attestation functionality）１４０を有している。シーリング／アンシーリング機能部１３０と遠隔認証機能部１４０は、例えば、トラステッド・コンピュータ・グループ（Trusted Computing Group）によって定義されている。またサーバ１００は、トラステッドブート機能部（Trusted Boot functionality）（以下では「ブート機能部」と呼ぶ）１５０を有している。

【０００８】

ブート機能部１５０は、サーバ内に設けられた各種のソフトウェアの起動時間（ブート時間）をモニタリングし、これらのソフトウェアの「測定値（測定結果）（measurements）」を記憶する。記憶される測定値は、例えば、ソフトウェアのハッシュ値やその他のファイル（例えばソフトウェアの構成ファイル）として実行可能である。測定値（ハッシュ値等）は、ＴＰＭ１２０の内部に、プラットフォーム・コンフィギュレーション・レジスターズ（以下では「ＰＣＲｓ」と呼ぶ）として記憶される。ブート機能部１５０は、起動

処理の全要素を測定するＰＣＲｓおよび選択的に追加されたソフトウェアのＰＣＲｓを把握する。ブート機能部１５０は、例えば、ＴｒｕｓｔｅｄＧＲＵＢによって実装される。ＰＣＲｓは、起動処理の各要素が選択的に追加されたソフトウェアと同様にサーバ上のソフトウェア動作として実現されることを保証する。

【０００９】

シーリング／アンシーリング機能部１３０は、ＴＰＭ１２０内で、ＰＣＲｓの値に従って、暗号化データと復号化データを参照することができる。特に、ＴＰＭがシーリング機能の一部としてデータを暗号化するとき、シーリング／アンシーリング機能部１３０は、特定のＰＣＲｓの値を記録する。ＴＰＭがデータを復号化（アンシーリング）するとき、シーリング／アンシーリング機能部１３０は、暗号化時（シーリング時）の特定のＰＣＲｓの値と同じ値のときにだけ、データの復号化（アンシーリング）が可能である。実際上、データを暗号化したソフトウェアだけがこの暗号化データを復号化することができる。

【００１０】

ＴＰＭ１２０は、遠隔認証機能部１４０を認証して、遠隔操作デバイスに対して、ＰＣＲの測定値の暗号証明を実行する（させる）。この暗号証明は、ＴＰＭ１２０の製造業者による認証に基づいて、当該製造業者がＴＰＭ１２０内にインストールした永久キーを認証することにより実行される。

【００１１】

本実施形態では、ブート機能部１５０により測定されて選択的に追加されたソフトウェアが、マニフェスト・チェッカー・ソフトウェア（以下では「ＭＣ」または「ＭＣソフトウェア」と呼ぶ）１６０を含んでいる。ブート機能部１５０により把握されたＰＣＲｓのセットは、ＭＣのＰＣＲｓを含んでおり、以下ではこれを「シーリングＰＣＲｓ」と呼ぶ。これは、サーバ上におけるユーザデータのシーリングとアンシーリングがＰＣＲｓのセットが存在するときにだけ実行されるからである。

【００１２】

本実施形態では、ブート機能部１５０とは別の構成要素であるマニフェストチェッカー（ＭＣ）１６０が、サーバ１００を動作不能状態とすることなくサーバアプリケーションソフトウェア（ＳＡ）１１０のアップデートを許容するために、サーバアプリケーションソフトウェア（ＳＡ）１１０を具体的に測定（監視）する。サーバアプリケーションソフトウェア（ＳＡ）１１０は、ユーザデータの保護を保証するためのソフトウェアの全構成要素、例えば、ＯＳ、環境設定ファイル（構成ファイル）、ＳＥＬｉｎｕｘポリシー、アプリケーションバイナリーを含んでいる。しかし、サーバアプリケーションソフトウェア（ＳＡ）１１０は、マニフェストチェッカー（ＭＣ）１６０の測定値を含んでいない。サーバアプリケーションソフトウェア（ＳＡ）１１０の現在測定値を含むＰＣＲまたはＰＣＲｓは「ＳＡ　ＰＣＲｓ」と呼ばれる。マニフェストチェッカー（ＭＣ）１６０は、「ＳＡ　ＰＣＲｓ」と、１つまたはそれ以上のマニフェスト２００の内部に記憶された測定値とを比較する。ここで、マニフェスト２００とその内部に記憶された測定値は、図２を参照して詳述する信頼できる第三者により認証されたものである。

【００１３】

図２に示すように、サーバ１００は、それぞれ異なる複数の第三者（Ａ、Ｂ等）に由来する複数のマニフェスト（２００Ａ、２００Ｂ等）を記憶している。それぞれの第三者は、自らのマニフェスト２００（当該マニフェストを構成するソフトウェアの測定値）を認証する。この認証のための方法の１つとして、第三者が、認証部（certificate）２４０によって確認された署名（signature）２３０を用いて、マニフェスト２００に署名することができる。例えば、第三者は、それぞれのプライベートキー２３０（２３０Ａ、２３０Ｂ、２３０Ｃ）を用いて、それぞれのマニフェスト２００（２００Ａ、２００Ｂ、２００Ｃ）に署名することができる。第三者は、それぞれのプライベートキー２３０（２３０Ａ、２３０Ｂ、２３０Ｃ）を用いることで、それぞれのマニフェスト２００（２００Ａ、２００Ｂ、２００Ｃ）を供給するための安全な通信プロトコル、例えばＳＳＬやＩＰＳＥＣを確立および認証することができる。認証部２４０は、第三者のそれぞれのプライベー

トキー２３０（２３０Ａ、２３０Ｂ、２３０Ｃ）に対応する公開キー２４０（２４０Ａ、２４０Ｂ、２４０Ｃ）とすることができる。認証部２４０は、サーバ１００の内部にマニフェストチェッカー（ＭＣ）１６０の一部として記憶されており、マニフェストチェッカー（ＭＣ）１６０の測定値の一部として含まれている。

【００１４】

図３に示すように、マニフェストチェッカー（ＭＣ）１６０は、サーバ１００の一部であり、サーバ１００の動作を制御するものである。マニフェストチェッカー（ＭＣ）１６０は、サーバ１００の起動要求に応じて、サーバアプリケーションソフトウェア（ＳＡ）１１０の少なくとも１つの測定値（ＳＡ　ＰＣＲ）を取得し、次のような処理ステップを実行する。

【００１５】

『ＭＣａ１．（３１０）』

マニフェストチェッカー（ＭＣ）１６０は、記憶されたマニフェスト２００を読み出す。

【００１６】

『ＭＣａ２．（３２０）』

マニフェストチェッカー（ＭＣ）１６０は、読み出したマニフェスト２００を認証（確認）する。マニフェスト２００を認証（確認）するために用いられる認証部２４０は、ブート機能部１５０（図１）により用いられるマニフェストチェッカー測定値に含まれている。マニフェスト・チェッカー・ソフトウェア（ＭＣ）１６０は、認証（確認）していない全てのマニフェスト２００を無視する。

【００１７】

『ＭＣａ３．（３３０）』

マニフェストチェッカー（ＭＣ）１６０は、認証（確認）したマニフェスト２００と、ＰＣＲｓの現在測定値とを比較する。この比較は少なくとも、上述した「ＳＡ　ＰＣＲｓ」を含んでいる。

【００１８】

『ＭＣａ４．（３４０）』

マニフェストチェッカー（ＭＣ）１６０は、マニフェスト２００の大部分が「ＳＡ　ＰＣＲｓ」の現在測定値と一致していれば、サーバアプリケーションソフトウェア（ＳＡ）１１０とのアクセスが可能になる。逆に、マニフェストチェッカー（ＭＣ）１６０は、マニフェスト２００の大部分が「ＳＡ　ＰＣＲｓ」の現在測定値と一致していなければ、サーバアプリケーションソフトウェア（ＳＡ）１１０とのアクセスが不能になる。例えば、マニフェストチェッカー（ＭＣ）１６０は、全てのマニフェスト２００が一致していることをアクセス条件としてもよいし、全てよりも少ない（大部分の）マニフェスト２００が一致していることをアクセス条件としてもよい。

【００１９】

図１を参照して前述したように、保護されたユーザデータ１０は、シーリング／アンシーリング機能部１３０によりシーリング（暗号化）されて、サーバ１００内の不揮発性メモリ（例えばディスクメモリやフラッシュメモリ）に記憶される。この処理は、全てのユーザデータを直接的にシーリング（暗号化）することで行ってもよいし、ユーザデータを暗号化および復号化するためのシーリングキーを使って効率的に行ってもよい。ユーザデータを暗号化および復号化するためのキーは、アンシーリングされてＲＡＭシステムにて使用される。マニフェストチェッカー（ＭＣ）１６０とサーバアプリケーションソフトウェア（ＳＡ）１１０の少なくとも一方は、シーリング／アンシーリング機能部１３０により、保護されたユーザデータ１０のシーリング／アンシーリングを行うか、ＴＰＭ１２０によりこれらのデータのシーリング／アンシーリングを行うべき旨の要求を出す。あるいは、ＴＰＭ１２０がＣＰＵに埋め込まれている場合には、ＣＰＵの内部でキーがアンシーリングされて暗号化および復号化のために使用される。このようにキーがＣＰＵから分離することはあり得ない。サーバアプリケーションソフトウェア（ＳＡ）１１０に代えて、

マニフェストチェッカー（ＭＣ）１６０が、シーリング／アンシーリング機能部１３０によりデータのシーリング／アンシーリングを行ってもよい。

【００２０】

本実施形態ではシーリングＰＣＲｓのみによってデータがシーリングされる。これは、マニフェストチェッカー（ＭＣ）１６０またはブート機能部の何らかの構成要素が改変されたとしても、ＴＰＭ、ＭＣまたはＳＡは、もはや、保護されたユーザデータ１０にアクセス（暗号化や復号化）するためのキーをアンシーリングできないことを意味している。これは、たとえマニフェストチェッカー（ＭＣ）１６０がシーリング／アンシーリング機能部１３０またはサーバアプリケーションソフトウェア（ＳＡ）１１０と繋がっていても同様である。

【００２１】

例えば、図２に示すように、第三者であるＡ、Ｂが、各自で生成したマニフェスト２００Ａ、２００Ｂに署名するために用いるプライベートキー２３０Ａ、２３０Ｂを保持している場合を考える。これらのマニフェスト２００（２００Ａ、２００Ｂ）は、サーバ１００の内部に伝送されて記憶される。マニフェストチェッカー（ＭＣ）１６０は認証部２４０を含んでおり、この認証部２４０は、第三者であるＡ、Ｂとこれに対応するプライベートキー２３０（２３０Ａ、２３０Ｂ）のための公開キー２４０（２４０Ａ、２４０Ｂ）からなる。各マニフェスト２００（２００Ａ、２００Ｂ）は、上述の『ＭＣａ２．（３２０）』において、記憶された認証部としての公開キー２４０（２４０Ａ、２４０Ｂ）に従ってこれに対応するプライベートキー２３０（２３０Ａ、２３０Ｂ）による署名がなされたときに、これが認証される。つまり、認証部としての公開キー２４０（２４０Ａ、２４０Ｂ）はシーリングＰＣＲに含まれる構成要素である。

【００２２】

第三者がそのマニフェストに署名する方法とは別の方法について説明する。この別の方法では、マニフェストチェッカー（ＭＣ）１６０が、第三者（Ａ、Ｂ等）のための公開キーを含む認証部２４０を使用して、ＳＳＬといった安全な通信チャンネルを経由して、第三者（Ａ、Ｂ等）を認証する。認証部２４０は、保護されたユーザデータ１０をシーリングするために用いるマニフェストチェッカー（ＭＣ）１６０の測定値２２０の一部である。

【００２３】

全ての第三者がマニフェストのマッチングを行うことを要求する方法（上述の『ＭＣａ４．（３４０）』）とは別の方法について説明する。この別の方法では、マニフェストチェッカー（ＭＣ）１６０が、第三者の一部だけがマニフェストのマッチングを行うことを要求する。例えば、３つの第三者が存在する場合を考えたとき、マニフェストチェッカー（ＭＣ）１６０は、そのうちの２つの第三者がマニフェストのマッチングを行うことを要求する。さらに別の方法では、マニフェストチェッカー（ＭＣ）１６０が、まず、第三者の一部だけがマニフェストのマッチングを行うことを要求し、その後、追加の第三者がマニフェストのマッチングを行うことを要求する。後者のタイミングまでに追加のマニフェストが供給不能な場合、マニフェストチェッカー（ＭＣ）１６０は、サーバアプリケーションソフトウェア（ＳＡ）１１０を停止（無効）にする。例えば、３つの第三者が存在する場合を考えたとき、マニフェストチェッカー（ＭＣ）１６０は、まず、そのうちの１つの第三者がマニフェストのマッチングを行うことを要求し、その後の一週間以内に、２番目の第三者がマニフェストのマッチングを行うことを要求する。マニフェストのマッチングを行う最小の人数、及び／又は、マッチングのスケジュールは、マニフェストチェッカー（ＭＣ）１６０の測定値の一部である。この場合、マニフェストのマッチングを行う最小の人数が以前と比べて改変されると、ユーザデータ１０のアンシーリング（復号化）が不可能となる。

【００２４】

本実施形態では、マニフェスト２００が以前のマニフェストの測定値を無視する（無効にする）。例えばマニフェスト２００は、１またはそれ以上の測定値（無効測定値）を含

むとともに、この測定値（無効測定値）のセットと一緒のソフトウェアは起動しない（無効である）ことを示す追加情報（文字列）を含んでいる。これは例えば、以前の測定値（無効測定値）と一緒のソフトウェアに安全性の欠陥が見つかったときに、そのソフトウェアを二度と起動できないようにすることを意味しており、有用な作用効果である。

【００２５】

以下では図１－図３を参照して、本実施形態のシステムと方法の１つの使用例について説明する。サーバ１００は、該サーバ１００がユーザデータ１０をどのようにして保護するかという個人情報保護に関する方針を実行する。個人情報保護に関する方針では、その方針で規定されたデータ保護が行なわれていないシステムによってはユーザデータ１０を絶対に取り扱わない旨の取り決めがなされている。第三者であるＡ、Ｂは、サーバアプリケーションソフトウェア（ＳＡ）１１０を調べて、これが個人情報保護に関する方針を満足していることを確認する。これらの第三者はそれぞれ署名キー２３０とこれに対応する認証部２４０を有している。これらの第三者は前もって認証部２４０をサーバ組織に提供しており、サーバ組織は、提供された認証部２４０をマニフェストチェッカー（ＭＣ）１６０にて一括管理する。サーバ１００のソフトウェアが個人情報保護に関する方針に従っていることを第三者が確認したとき、第三者はそれぞれ、マニフェスト２００に対応するキー２３０を生成してこれに署名する。マニフェスト２００とこれに対応するキー２３０は、「シーリングＰＣＲｓ」や「ＳＡ　ＰＣＲｓ」を含んでいる。

【００２６】

本実施形態では、例えば図４に示すように、生成されたマニフェストによって認証された第三者のリストを改変することが望ましい。図３の実施形態では、マニフェストチェッカー（ＭＣ）１６０に認証部２４０を追加または削除することにより、サーバアプリケーションソフトウェア（ＳＡ）１１０によるユーザデータ１０のアンシーリングを不能にすることができる。第三者認証のリストが改変されたときにユーザデータが無くなるのを防止するために、図４では、マニフェストの追加の種類または層を含む第三者リスト４００を使用している。この第三者リスト４００は、マニフェストを生成する認証された第三者（Ａ、Ｂ、Ｃ）の認証部２４０を識別することができる。しかし、この第三者リスト４００に含まれる認証部２４０は、ＴＰＭ１２０により使用されるマニフェストチェッカーの測定値の一部ではない。むしろ、マニフェストチェッカーの測定値は、第三者リスト４００の生成を許可された第三者のための認証部４１０のみを含むものである。

【００２７】

図４を参照して、第三者であるＣとＤが、それぞれ、プライベートキー４２０Ｃと４２０Ｄを使用して、第三者リスト４００Ｃと４００Ｄを生成してこれに署名する場合について説明する。第三者リスト４００は、第三者であるＡ、Ｂ、Ｃのための認証部（公開キー）２４０を含んでいる。マニフェストチェッカー（ＭＣ）１６０は、第三者であるＣとＤのための認証部（公開キー）４１０を保持している。この認証部（公開キー）４１０は「シーリングＰＣＲ」の測定値の一部である。これに対し認証部（公開キー）２４０はもはや「シーリングＰＣＲ」の測定値の一部ではない。

【００２８】

マニフェストチェッカー（ＭＣ）１６０は、次のような処理ステップを実行する。

【００２９】

『ＭＣｂ１．４３０』

マニフェストチェッカー（ＭＣ）１６０は、記憶された第三者リスト４００を読み出す。

【００３０】

『ＭＣｂ２．４４０』

マニフェストチェッカー（ＭＣ）１６０は、認証部４１０に対して第三者リスト４００を認証する。マニフェストチェッカー（ＭＣ）１６０は、認証失敗した全ての第三者リストを無視（破棄）する。

【００３１】

『ＭＣｂ３．』
マニフェストチェッカー（ＭＣ）１６０は、認証成功した第三者リスト４００内の第三者Ａ、Ｂに対応する認証部を使用して、上述した『ＭＣａ１．（３１０）』から『ＭＣａ４．（３４０）』の処理を実行する。
【００３２】
マニフェストチェッカー（ＭＣ）１６０は、第三者リスト４００が高精度に一致しないときは、『ＭＣｂ３．』の処理を実行しない。あるいは、マニフェストチェッカー（ＭＣ）１６０は、全ての第三者リスト４００に共通する第三者により認証されたマニフェストだけを使用して、『ＭＣｂ３．』の処理を実行してもよい。さらには、マニフェストチェッカー（ＭＣ）１６０は、いずれかの第三者リストに含まれる第三者により認証されたマニフェストを使用してもよい。マニフェストチェッカー（ＭＣ）１６０は、第三者リストに十分な数の第三者がリストアップされていないとき、『ＭＣｂ３．』の処理を実行しない。第三者の最小数は、シーリング／アンシーリング機能部１３０のために認証されたコードベースの一部である。あるいは、マニフェストは安全性の高いセッション確立の間に認証されてもよい。
【００３３】
図５に示すように、評価者としての第三者Ｃ、Ｄは、マニフェスト生成者である第三者Ａ、Ｂがサーバソフトウェアを調査する権限を有しているか否かを評価する（５００）。評価者としての第三者Ｃ、Ｄは、それぞれ、署名キー４２０とこれに対応する認証部４１０を生成する（５１０）。評価者としての第三者Ｃ、Ｄは、その認証部４１０Ｃ、４１０Ｄをサーバ組織に提供する（５２０）。サーバ組織は、マニフェストチェッカー１６０内に認証部を含んでいる。評価者としての第三者Ｃ、Ｄは、それぞれの第三者リスト４００Ｃ、４００Ｄを提供することにより、マニフェスト生成者である第三者Ａ、Ｂに対して権限を与える。第三者リスト４００Ｃ、４００Ｄは、第三者Ａ、Ｂのための認証部を含んでいる。
【００３４】
本実施形態では、図１に示すように、サーバ１００が、現在および過去にサーバにより動作されたソフトウェアによる全ての測定値の不正開封防止ログを記憶している。例えばマニフェストチェッカー１６０がログ１７０を生成することができる。これを実現するための構成として、サーバ１００またはマニフェストチェッカー１６０は、そのＰＣＲが時々刻々と変化するログ１７０へのエントリー権（ＰＣＲ値の参照権）を生成することができる。ログエントリーは、過去のログエントリーのハッシュ値に沿ったＰＣＲの測定値（次の過去のログエントリーとなるログエントリーのハッシュ値）を含んでいる。このログは、遠隔認証部２５０（図２）の一部として確立された安全性の高い経路を介して要求および伝送される。同様に、サーバ１００（さらにはマニフェストチェッカー１６０）は、サーバが受け取るすべてのマニフェストの不正開封防止ログを生成する。
【００３５】
図６に示すように、遠隔システム（クライアント６００）は、サーバ１００との接続を確立し、クライアントは、第三者であるＡ、Ｂ、Ｃの少なくとも一部がサーバを認証したこと（第三者認証）を確認する。つまりクライアントがサーバのユーザとなり、クライアントがピアサーバとなり、サーバのセットがクラスターとして動作し、サーバからサーバへユーザデータ１０を伝送することが可能になる。
【００３６】
本実施形態では、クライアント６００が第三者のための公開キー（認証部）２４０を記憶している。サーバ１００の遠隔認証部１４０が遠隔認証処理を行っている間、クライアントは、サーバのためのＰＣＲ値（「シーリングＰＣＲｓ」と「ＳＡ　ＰＣＲｓ」）を取得する。遠隔認証部１４０は、クライアント６００に対して、サーバ１００が有効なＴＰＭ１２０を有していること及びＰＣＲｓが適切に生成されていることについての確認を行う。クライアント６００は、サーバからマニフェスト２００を取得する。クライアントは、該クライアントが記憶した第三者の認証部２４０を使用してマニフェストを認証する（

３２０）。またクライアントは、該クライアントが記憶した第三者の認証部２４０を使用して、マニフェストＰＣＲ値と、サーバの遠隔認証部１４０により提供されたＰＣＲ値とを比較する（３３０）。マニフェスト２００が認証され、且つ、比較の結果として両ＰＣＲ値が一致すると（３２０、３３０）、クライアントが適切に認証されたサーバと通信することができるようになる。比較処理においては、「シーリングＰＣＲｓ」と「ＳＡ　ＰＣＲｓ」の両方が比較の対象となる。クライアント（ユーザまたはピアサーバ）とサーバ１００とが適切に認証された接続関係になると、ユーザデータの伝送が可能になる。

【００３７】

本実施形態では、遠隔認証部１４０による遠隔認証処理の間、クライアント６００が第三者のマニフェスト２００を取得する。この処理はサーバハッシュ６０２により安全性が確保された状態で実行される。サーバハッシュ６０２は、遠隔認証（トラステッドコンピュータグループによって特定される）を目的としてクライアントにより発行されるチャレンジ６１０を含んだマニフェスト２００からなる。マニフェストのハッシュとチャレンジは、ＰＣＲｓのような他の値と一緒に、ＴＰＭ１２０が署名すべき新たなチャレンジ６２０を生成する。マニフェスト２００は、新チャレンジのＴＰＭの署名６３０およびその他の値と同様に（一緒に）クライアントに戻される。クライアントは、ＴＰＭにより署名された元のチャレンジとマニフェストのハッシュに基づいて、戻された新チャレンジと同じものを再生成する。そして、再生成した新チャレンジが適切に署名されると（新チャレンジ６２０と一致すると）、クライアントは、マニフェストがサーバによって伝送されたことを確認することができる。

【００３８】

図７は、別の実施形態を示している。この別の実施形態では、クライアント６００が、公開キー６３０を使用して、サーバ１００との間で安全性の高い接続経路６４０を確立する。公開キー６３０は、遠隔認証部１４０による遠隔認証処理の間にサーバから伝送される。公開キー６３０は、図６のマニフェスト２００のときと同じ方法（すなわちＴＰＭにより署名されたチャレンジで公開キーをハッシュする）により認証される。サーバ１００は、安全性の高い接続経路６４０を介して、クライアントに対してマニフェスト２００を伝送する。

【００３９】

本明細書は、ベストモードを含む幾つかの実施形態を例示的に開示しており、当業者であれば、本発明の実施形態に様々な装置やシステムや方法を組み込んでこれを実施することができる。本発明の技術的範囲は、クレームの記載及びこれに当業者が適宜の設計変更を付加した範囲にまで及ぶものとする。つまり、このような設計変更を付加した他の例は、これがクレームの文言から相違しない限りにおいて、あるいは、これがクレームの文言と実質的な相違がなく等価である限りにおいて、クレームの範囲に含まれるのである。

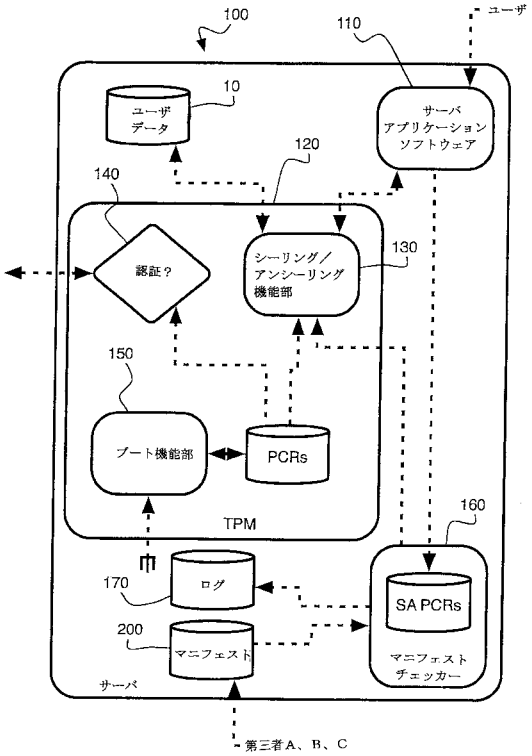
【００４０】

本明細書で使用する構成要素またはステップは、単数形で記載し、“ａ”または“ａｎ”の記載は、その構成要素またはステップが複数あることが発明の必須の構成要件として明示的に述べられない限りにおいて、その構成要素またはステップが複数あることを除外しないように理解するべきである。また、本発明の「一実施形態（one embodiment）」という記載は、これに付加的な特徴を組み込んだ実施形態を排除して解釈されることを意図するものではない。さらに、明確にそうではない旨の記載がない限りにおいて、“ｃｏｍｐｒｉｓｉｎｇ”、“ｉｎｃｌｕｄｉｎｇ”または“ｈａｖｉｎｇ”の記載は、たとえ本発明の１または複数の構成要素に付加的な特徴を組み込んだとしても、本発明の技術的範囲に含まれるという意味で使用する。

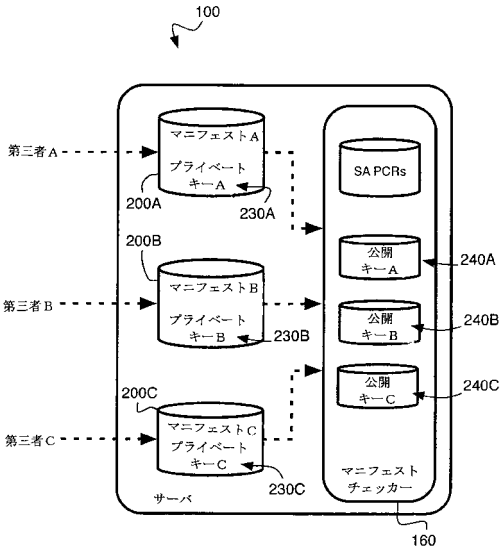
【００４１】

上述した方法およびシステムにおいては、本発明の思想および範囲から逸脱することなく、各種の変更を行うことができる。このため、本明細書と図面に示されているのは、本発明のコンセプトのほんの一例にすぎないと解釈するべきであり、本発明の範囲がこれに限定されるべきものと解釈してはならない。

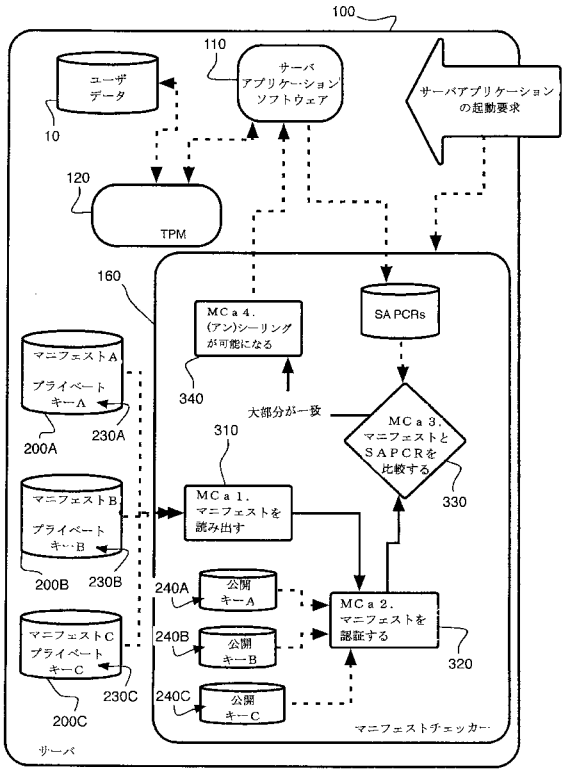
【図１】


100
110
ユーザ
10
ユーザデータ
サーバアプリケーションソフトウェア
120
140
認証？
シーリング／アンシーリング機能部
130
150
ブート機能部
PCRs
TPM
160
170
ログ
SA PCRs
200
マニフェスト
マニフェストチェッカー
サーバ
第三者A、B、C

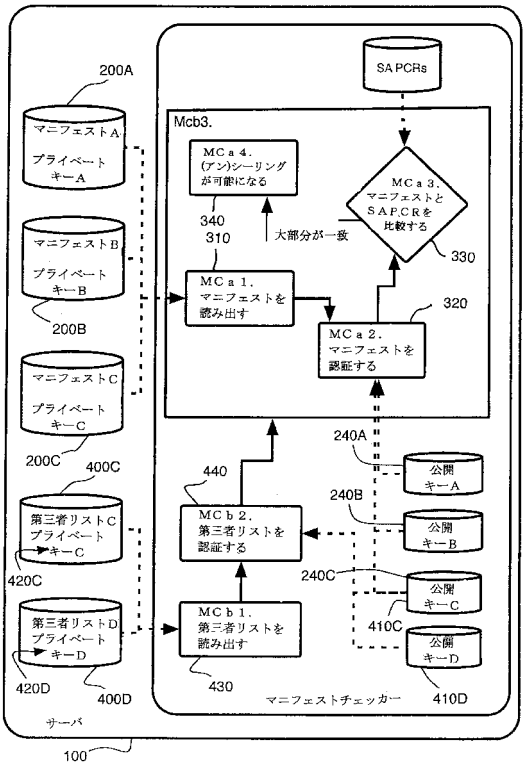
【図２】


100
第三者A
マニフェストA
プライベートキーA
200A
230A
SA PCRs
公開キーA
240A
第三者B
200B
マニフェストB
プライベートキーB
230B
公開キーB
240B
公開キーC
240C
200C
第三者C
マニフェストC
プライベートキーC
230C
マニフェストチェッカー
サーバ
160

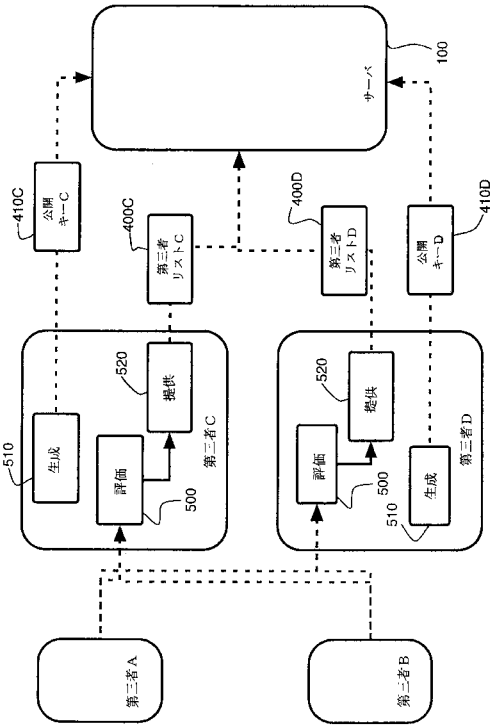
【図３】


100
110
ユーザデータ
10
サーバアプリケーションソフトウェア
サーバアプリケーションの起動要求
120
TPM
160
MCa4.(アン)シーリングが可能になる
SA PCRs
340
大部分が一致
310
MCa3.マニフェストとSAPCRを比較する
330
マニフェストA
プライベートキーA
200A
230A
マニフェストB
プライベートキーB
200B
230B
MCa1.マニフェストを読み出す
240A
公開キーA
240B
公開キーB
MCa2.マニフェストを認証する
320
マニフェストC
プライベートキーC
230C
200C
240C
公開キーC
マニフェストチェッカー
サーバ

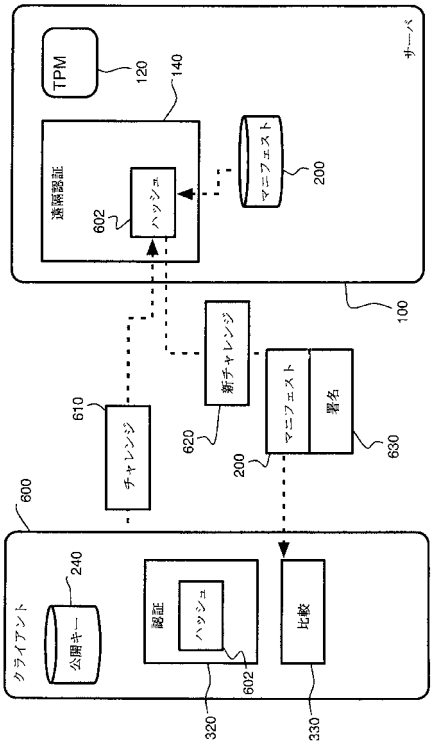
【図４】


200A
マニフェストA
プライベートキーA
SA PCRs
Mcb3.
MCa4.(アン)シーリングが可能になる
340
310
大部分が一致
MCa3.マニフェストとSAPCRを比較する
330
マニフェストB
プライベートキーB
200B
MCa1.マニフェストを読み出す
MCa2.マニフェストを認証する
320
マニフェストC
プライベートキーC
200C
400C
240A
公開キーA
440
240B
公開キーB
第三者リストC
プライベートキーC
420C
MCb2.第三者リストを認証する
240C
公開キーC
410C
第三者リストD
プライベートキーD
420D
400D
MCb1.第三者リストを読み出す
430
公開キーD
410D
マニフェストチェッカー
サーバ
100

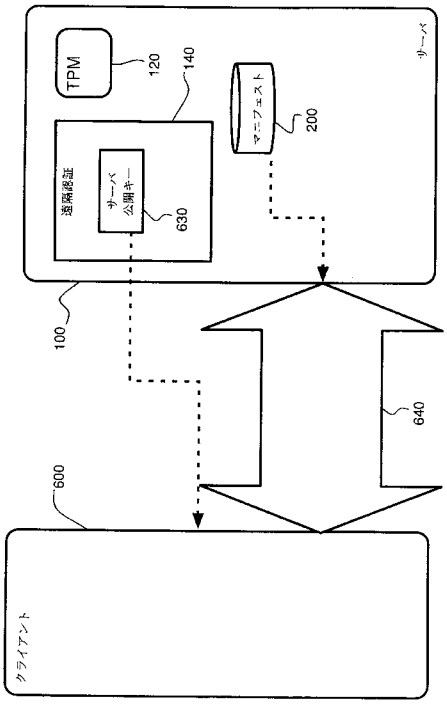
【図５】


サーバ
100
410C
公開キーC
400C
第三者リストC
400D
第三者リストD
410D
公開キーD
520
提供
510
生成
評価
500
第三者C
520
提供
評価
500
510
生成
第三者D
第三者A
第三者B

【図６】


TPM
120
140
遠隔認証
602
ハッシュ
マニフェスト
200
サーバ
100
新チャレンジ
620
マニフェスト
署名
630
200
610
チャレンジ
600
クライアント
240
公開キー
認証
ハッシュ
602
比較
320
330

【図７】


TPM
120
140
遠隔認証
サーバ公開キー
630
マニフェスト
200
サーバ
100
640
600
クライアント

フロントページの続き

(72)発明者　フェリックス・バウアー
ドイツ連邦共和国　６７６５５　カイザースラウテルン　ガウシュトラーセ　１５
(72)発明者　セバスチャン・プロプスト・アイデ
ドイツ連邦共和国　６７６５５　カイザースラウテルン　ガウシュトラーセ　１５
(72)発明者　マティアス・クレッチナー
ドイツ連邦共和国　５３７５７　サンクト・オーガスティン　リヒトーフェンシュトラーセ　４６Ａ
(72)発明者　クリスチャン・ダニエル・ベルネアヌ
ルーマニア　０３２１１５　ブカレスト　セクトル３　アパートメント１７　アリーア・モスティシュテア　ナンバー３９Ａ

Ｆターム(参考) 5J104 AA08 AA12 JA21 LA03 NA02 NA37 PA07

【外国語明細書】

1

SYSTEMS AND METHODS FOR ENFORCING THIRD PARTY OVERSIGHT OF DATA ANONYMIZATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Serial No. 61/882321, filed on September 25, 2013, and hereby incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] The present invention relates generally to a method and system for giving trusted third parties oversight of servers that handle private user data.

BACKGROUND OF THE INVENTION

[0003] There are many cases where users submit private data to remote servers. The organization operating the servers (the serving organization) may claim to protect the user's data in one way or another (for instance in a privacy statement). The users may not fully trust the organization operating the servers (the serving organization), and may wish to have some strong assurance from a trusted third party that the serving organization will honor its claim of protection. Today this is done in a weak fashion. For instance, the third party may examine the server software and provide a certification that the software faithfully executes the protection claim. However, the serving organization may easily modify its software

afterwards and remove the claimed protections. There is a need for systems and methods that enforce trusted third party control over server software, so that the serving organization must be incapable of modifying its software without approval from the third party. Furthermore, users must have strong assurance that this enforced control exists before transmitting private data to the server.

SUMMARY OF THE INVENTION

[0004] In accordance with one aspect of the present invention, independent third parties are able to prevent operation of specific server application software unless some or all of the third parties approve that software. Users submitting private data to the servers can obtain cryptographic proof that the independent third parties have this oversight. In addition, a server is able to produce a tamper-proof log of all software versions that have run previously. This allows anyone to know exactly what software has run in a server in the past. This model provides assurance to clients before sending data to a server that the server application software treats the data safely.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The aforementioned features and advantages, and other features and aspects of the present invention, will become better understood with regard to the following description and accompanying drawings, wherein:

[0006] FIG. 1 shows in schematic view a server for handling protected user data.

[0007] FIG. 2 shows in schematic view a method for third party approval of server software.

[0008] FIG. 3 shows in schematic view a method for verifying third party approval of server software.

[0009] FIG. 4 shows in schematic view a method for managing lists of third parties trusted to approve server software.

[00010] FIG. 5 shows in schematic view a method of accessing (providing or querying) protected user data via the server shown in FIG. 1.

[00011] FIG. 6 shows in schematic view a mode of authenticating remote connection between a server and a client, according to one aspect of the invention.

[00012] FIG. 7 shows in schematic view a mode of authenticating remote connection between a server and a client, according to another aspect of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

[00013] Reference will be made below in detail to exemplary embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference characters used throughout the drawings refer to the same or like parts.

[00014] As shown in FIG. 1, a server 100, which runs server application software 110 for handling private or protected user data 10, contains the functionality of a Trusted Platform Module (TPM) 120. This TPM functionality may be based on TPM hardware and associated motherboard firmware, or may be implemented within the server's CPU. The TPM may provide sealing and unsealing functionality

130 and remote attestation functionality 140, for instance as defined by the Trusted Computing Group. Also, the server 100 contains Trusted Boot functionality 150.

[00015] Essentially, the Trusted Boot 150 monitors at boot time some of the software resident on the server, and internally stores "measurements" of this software. The stored measurements are, for example, in the form of hashes of software executables and other files (i.e. software configuration files). These measurement hashes are stored within the TPM as Platform Configuration Registers (PCRs). The Trusted Boot establishes the PCRs that measure all of the components of the boot process, as well as optionally the PCRs of selected additional software. The Trusted Boot may be implemented for instance by TrustedGRUB. These PCRs provide proof that each component of the boot process, as well as the selected additional software, is indeed the software operating on the server.

[00016] Sealing and unsealing 130 refers to encryption and decryption of data according to the values of the PCRs in the TPM. In particular, when the TPM encrypts data as part of the sealing function, it records the values of specified PCRs. When the TPM is asked to decrypt the data (unseal), it will do so only if the previously specified PCRs have the same values. In effect, only the software that encrypted data can subsequently decrypt that data.

[00017] The TPM also can attest 140 to a remote device a cryptographic proof of its PCR measurement values. This proof derives from a certificate signed by the TPM manufacturer, which certifies a permanent key installed in the TPM by the manufacturer.

[00018] According to embodiments of the invention, the selected additional software measured by the Trusted Boot includes "manifest checker" software 160 (the "MC"). The set of PCRs established by Trusted Boot, including the PCRs of the MC,

are called Sealing PCRs. This is because the sealing and unsealing of user data on the server is accomplished only when this set of PCRs is present.

[00019] According to certain embodiments, the MC 160 specifically measures the server application software (the "SA") 110, which is excluded from the Trusted Boot functionality in order to permit updates of the SA without rendering the server 100 inoperable. The SA includes all software components that are responsible for ensuring that user data is protected, and can include for instance the OS, configuration files, SELinux policies, and application binaries; but do not include measurements of the manifest checker 160. The PCR or PCRs that contain the current measurements of the SA are called the SA PCRs. The MC 160 compares the SA PCRs to stored measurements within one or more manifests 200, which are authenticated by trusted third parties as further discussed with reference to FIG. 2.

[00020] Referring to FIG. 2, the server 100 may store multiple manifests 200A, 200B, etc., which come from respective different third parties A, B, etc. Each third party authenticates its own manifest 200 (and the software measurements that compose the manifest). One method for authentication is for the third party to sign the manifest 200 using a signature 230 that can be verified by a certificate 240. For example, a third party may sign its respective manifest 200 by using a private key 230. A third party may also use its private key 230 to establish and authenticate a secure communications protocol such as SSL or IPSEC for submitting the manifest 200. The certificate 240 then would be the public key corresponding to the third party's private key 230. The certificates 240 are stored within the server 110 as part of the MC 160, and are included as part of the MC measurement..

[00021] Referring now to FIG. 3, the manifest checker 160 is a component of, and controls the operation of, the server 100. In response to each request to launch the

server 100, the manifest checker 160 takes at least one measurement (SA PCR) of the server application 110, and takes the following steps:

[00022] MCa1. Read 310 the stored manifests 200.

[00023] MCa2. Authenticate 320 the manifests 200. Certificates 240 that are used to authenticate the manifests 200, were included in the manifest checker measurement used by the Trusted Boot 150 in FIG. 1. The manifest checker software 160 ignores any manifests 200 that are not authenticated.

[00024] MCa3. Compare 330 the authenticated manifests 200, against presently measured PCRs. This comparison 330 includes at least the SA PCRs.

[00025] MCa4. If a sufficient number of the manifests 200 match to the presently measured SA PCRs, then enable 340 the server application 110. Otherwise, do not enable the server application 110. For example, the manifest checker 160 may be configured such that all the manifests 200 must match; alternatively, as further discussed below, fewer than all the manifests may suffice.

[00026] As mentioned before with reference to FIG. 1, protected user data 10 is sealed 130 while stored in non-volatile memory on the server 100 (e.g., on a disk or flash memory). This may be done by directly sealing all of the user data, or more efficiently by sealing a key that encrypts and decrypts the user data. The key that encrypts and decrypts the user data then may be unsealed and used in system RAM. Either the MC 160 itself, or the SA 110, may perform sealing or unsealing 130 of protected user data 10, or may request sealing and unsealing 130 of data by the TPM 120. Alternatively, in the case where the TPM is embedded in the CPU, the key may be unsealed 130 within the CPU, and used for encryption and decryption while in the CPU. In this way, the key never leaves the CPU. In certain

embodiments, instead of enabling the server application 110, the MC 160 may simply enable sealing / unsealing 130 of data (e.g., by the server application 110).

[00027] In certain embodiments, the data is sealed, according to only the sealing PCRs. This means that if the manifest checker 160 or any of the trusted boot components are modified, the TPM, MC, or SA cannot any longer unseal 130 the key for accessing (encrypting or decrypting) protected user data 10, even if the manifest checker 160 enables sealing / unsealing 130 or enables the server application 110.

[00028] For example, as shown in FIG. 2, third parties A and B hold private keys 230A, 230B that can be used to sign the manifests 200A, 200B that they respectively produce. These manifests 200 are transmitted and stored in the server 100. The manifest checker 160 contains certificates 240, for example the public keys for third parties A and B corresponding to their respective private keys 230. Each manifest 200 is authenticated Mca2 when its signature 230 is valid according to the corresponding stored certificate 240. The certificates 240 are included in the sealing PCR.

[00029] As an alternative method, instead of having a third party sign its manifest, the manifest checker 160 may authenticate the third parties A, B, etc. over a secure communications channel like SSL, using the certificates 240 that contain the public keys for the third parties. The certificates 240 are part of the measurement 220 of the manifest checker 160 that is used for sealing the protected user data 10.

[00030] As another alternative, instead of requiring that all third parties have a matching manifest (step MCa4), the manifest checker 160 could require that only some of the third parties have a matching manifest. For instance, if there were three third parties, the manifest checker might require that only two of them provide a

matching manifest. As another alternative, the manifest checker 160 could require that some of the third parties have a matching manifest, and that at a later time, additional third parties have a matching manifest. In case the additional manifests were not supplied by the later time, the manifest checker would disable the server application software. For instance, if there are three third parties, the manifest checker may require that initially only one of them has a matching manifest, but that within one week a second matching manifest must be supplied. The minimum number of matching manifests, and/or the matching schedule, may be made a part of the manifest checker measurement, in which case, modifying the minimum number from what was previously approved, would make it impossible to unseal the user data 10.

[00031] In certain embodiments, a manifest 200 may negate the measurements of previous manifests. For instance, a manifest 200 may contain one or more measurements ("disabled measurements"), as well as an additional value or string indicating that software with this set of measurements must not be enabled. This is useful, for instance, if software with a previous measurement was found to have a security flaw, and so should never be run again.

[00032] Thus, referring to FIGS. 1-3, one use of the system and methods is as follows. The organization operating the server 100 makes a privacy policy, which describes how the server should protect user data 10. The privacy policy also states that user data 10 will never be handled by a system that does not protect data at least as well as stated in the privacy policy. Third parties A, B then inspect the server application 110 to insure that it enforces the privacy policy. These third parties each have a signing key 230 and corresponding certificate 240. They previously have submitted their certificates 240 to the server organization, which incorporates the certificates

into the manifest checking application 160. When the third parties are satisfied that the server software 100 follows the privacy policy, they produce and sign with their keys 230 respective manifests 200 that contain the measurements for the sealing PCRs and SA PCRs.

[00033] In certain embodiments, of which FIG. 4 shows an example, it may be desirable to modify the list of third parties that are authorized generate manifests. In the embodiment of FIG. 3, adding or removing certificates 240 from the manifest checker 160 would result in the server application 110 not being able to unseal the user data 10. To avoid having to lose user data when the list of third party certificates changes, FIG. 4 provides for using an additional kind or layer of manifest, a third party list 400. The third party list 400 identifies certificates 240 of the authenticating third parties A, B, C that may produce a manifest. However, the certificates 240 contained in the third party list 400 are not part of the manifest checker measurement used by the TPM 120. Rather, the manifest checker measurement includes only the certificates 410 for the third parties that are allowed to produce third party lists 400.

[00034] Referring to FIG. 4, third parties C and D produce and sign third party lists 400C, 400D with their respective private keys 420C, 420D. The third party lists 400 contain the certificates (public keys) 240 for third parties A, B, C. The manifest checker 160 holds the certificates (public keys) 410 for third parties C and D, which are part of the sealing PCR measurement. The certificates 240 no longer are part of the sealing PCR measurement.

[00035] Thus, the manifest checker 160 takes the following steps:

[00036] MCb1. Read 430 the stored third party lists 400.

[00037] MCb2. Authenticate 440 the third party lists 400 against the certificates 410. Ignore any third party lists whose authentication fails.

[00038] MCb3. Using the certificates that correspond to the third parties A, B in the authenticated third party lists 400, execute steps MCa1 through MCa4.

[00039] The manifest checker 160 may refuse to execute step MCb3 if the third party lists 400 do not exactly match. Alternatively, the manifest checker may execute step MCb3 using only manifests authenticated by third parties that are common to all third party-lists 400. Alternatively, the manifest checker may use manifests authenticated by any of those third parties included in any of the third party lists. The manifest checker 160 may refuse to execute step MCb3 if there are not enough third parties listed in the third party lists. This minimum number of third parties may be part of the attested code base for sealing 130. Alternatively, the manifests may be authenticated during secure session establishment.

[00040] Thus, referring to FIG. 5, evaluating third parties C, D can evaluate 500 whether the manifest-producing third parties A, B are qualified to inspect server software. These evaluating third parties C, D then each produce 510 a signing key 420 and corresponding certificate 410. They submit their certificates 410C, 410D to the server organization, which incorporates the certificates into the manifest checker 160. The evaluating third parties C, D then can qualify the manifest producing third parties A, B by providing respective third party lists 400C, 400D that include the certificates for the third parties A, B.

[00041] In certain embodiments, as shown for example in FIG. 1, the server 100 may store a tamper-proof log of all measurements for software operated by the server currently and in the past. For example, the manifest checker 160 may produce the log 170. One way to do this is for the server 100 or the manifest checker 160 to create

an entry in the log 170, each time a PCR differs from its previous value. The log entry may contain the measurement PCR, along with a hash of the previous log entry (including the previous log entry's hash of the next previous log entry). The log may be requested and transmitted via a secure connection that is established as part of a remote attestation 250 (as shown in FIG. 2). Similarly, the server 100 (and more particularly, the manifest checker 160) also may produce a tamper-proof log of all manifests received by the server.

[00042] Referring to FIG. 6, when a remote system (a "client" 600) establishes a connection with the server 100, the client may wish to confirm that at least some of the third parties A, B, C have approved the server. The client may be a user of the server. The client may also be a peer server, for instance in the case where a set of servers operate as a cluster, and where for instance one server transmits user data 10 to another server.

[00043] In an embodiment, the client 600 has stored the public keys (certificates) 240 for the third parties. During remote attestation 140 of the server 100, the client obtains the PCR values for the server (both the sealing PCRs and the SA PCRs). Remote attestation 140 confirms to the client 600 that the server 100 has a valid TPM 120, and that the PCRs were generated properly. The client 600 also obtains the manifests 200 from the server. Using its own stored third party certificates 240, the client authenticates 320 the manifests, and then compares 330 the manifest PCR values with those provided by the server's remote attestation 140. If the manifests 200 are authenticated 320, and the PCR values match 330, then the client is assured that it is communicating with a legitimate approved server. Both the sealing PCRs and the SA PCRs may be compared. Once the client (user or peer server) legitimizes the server 100, it may transmit user data to the server.

[00044] In an embodiment, the client 600 obtains the third party manifests 200 during the remote attestation 140. This may be done securely by the server hashing 602 the manifests 200 with a challenge 610 that is issued by the client for the purpose of remote attestation (as specified by the Trusted Computing Group). The hash of the manifests and the challenge produces a new challenge 620, which the TPM 120 must sign (along with other values like the PCRs). The manifests, as well as the TPM's signature 630 of the new challenge and other values, are then transmitted back to the client. The client recreates the same new challenge by hashing the original challenge and the manifests signed by the TPM. If the recreated new challenge is properly signed (matches the new challenge 620), then the client knows that the manifests have been transmitted by the server itself.

[00045] In another embodiment, shown in FIG. 7, the client 600 uses a public key 630 (transmitted from the server during remote attestation 140) to establish with the server 100 a secure connection 640. This public key 630 may be authorized in the same way as were the manifests 200 in FIG. 6 (i.e., hashing the public key with the challenge signed by the TPM). The server 100 then transmits the manifests 200 to the client via the secure connection 640.

[00046] This written description uses examples to disclose several embodiments of the invention, including the best mode, and also to enable one of ordinary skill in the art to practice the embodiments of invention, including making and using any devices or systems and performing any incorporated methods. The patentable scope of the invention is defined by the claims, and may include other examples that occur to one of ordinary skill in the art. Such other examples are intended to be within the scope of the claims if they have structural elements that do not differ

from the literal language of the claims, or if they include equivalent structural elements with insubstantial differences from the literal languages of the claims.

[00047] As used herein, an element or step recited in the singular and proceeded with the word "a" or "an" should be understood as not excluding plural of said elements or steps, unless such exclusion is explicitly stated. Furthermore, references to "one embodiment" of the present invention are not intended to be interpreted as excluding the existence of additional embodiments that also incorporate the recited features. Moreover, unless explicitly stated to the contrary, embodiments "comprising," "including," or "having" an element or a plurality of elements having a particular property may include additional such elements not having that property.

[00048] Since certain changes may be made in the above-described method and system, without departing from the spirit and scope of the invention herein involved, it is intended that all of the subject matter of the above description or shown in the accompanying drawings shall be interpreted merely as examples illustrating the inventive concept herein and shall not be construed as limiting the invention.

CLAIMS

WHAT IS CLAIMED IS:

1. A server for sealing and unsealing user data, said server comprising:

a processor; and

a memory device, which stores: an encryption key, user data encrypted by the encryption key, a plurality of authenticated manifests each comprising authenticated measurements of the server, and a server application comprising a manifest checker component and a sandbox component, the encryption key being sealed in the first memory device using a measurement of the manifest checker component;

the processor being adapted by the server application to:

implement the manifest checker component to obtain current measurements of the server corresponding to the authenticated measurements of the server, and to compare the current measurements to the authenticated measurements, wherein the measurements of the server do not include any measurement of the manifest checker component; and,

implement the sandbox component to unseal the encryption key using a current measurement of the manifest checker component, and to access the user data using the encryption key, if the current measurements of the server sufficiently match the authenticated measurements of the server, or

prevent implementation of the sandbox component, if the current measurements of the server do not sufficiently match the authenticated measurements of the server.

2. The server as claimed in claim 1, wherein each of the plurality of authenticated manifests has been authenticated by a certificate different from certificates authenticating each other of the plurality of authenticated manifests.

3. The server as claimed in claim 1, wherein the manifest checker component configures the processor to authenticate the authenticated manifests against a plurality of certificates stored in the memory device.

4. The server as claimed in claim 3, wherein the plurality of certificates are encoded into the server application.

5. The server as claimed in claim 4, wherein the manifest checker component incorporates the plurality of certificates.

6. The server as claimed in claim 3, wherein the manifest check component configures the processor such that the current measurements sufficiently match the authenticated measurements only if all of the authenticated manifests can be authenticated against the plurality of certificates stored in the memory device and only if all of the authenticated measurements match the current measurements.

7. The server as claimed in claim 3, wherein the manifest check component configures the processor such that the current measurements sufficiently match the authenticated measurements only if at least all but one of the authenticated manifests can be authenticated against the plurality of certificates stored in the

memory device and only if at least all but one of the authenticated manifests contains authenticated measurements that match the current measurements.

8. The server as claimed in claim 3, wherein the manifest check component configures the processor such that the current measurements sufficiently match the authenticated measurements only if at least a majority of the authenticated manifests can be authenticated against the plurality of certificates stored in the memory device and only if at least a majority of the authenticated manifests contain authenticated measurements that match the current measurements.

9. The server as claimed in claim 3, wherein the manifest check component configures the processor such that the current measurements sufficiently match the authenticated measurements only if at least a majority of the authenticated manifests contain authenticated measurements that match the current measurements.

10. The server as claimed in claim 1, wherein the manifest check component configures the processor such that the current measurements cannot sufficiently match the authenticated measurements if at least one of the authenticated manifests comprises a disabled measurement that matches one of the current measurements.

11. The server as claimed in claim 1, wherein the manifest checker component includes a third party list that is not a part of the measurement of the manifest checker component, and the third party list comprises the certificates of third parties authorized to authenticate manifests.

12. The server as claimed in claim 11, wherein the manifest checker component also includes a certificate of an evaluating party who is authorized to modify the third party list, and the certificate of the evaluating party is a part of the measurement of the manifest checker component.

13. The server as claimed in claim 1, wherein the manifest checker component configures the processor to remotely attest another server by comparing current measurements of the other server to authenticated measurements of the manifests stored in the memory device.

14. The server as claimed in claim 1, wherein the manifest checker component configures the processor to check incoming user data for a data originator certificate, and to unseal data certified by the data originator only if a data originator manifest matches current measurements of the server application.

15. The server as claimed in claim 14, wherein the manifest checker component configures the processor to check the data originator certificate for a list of third parties that are approved to provide authenticated manifests.

16. The server as claimed in claim 15, wherein the manifest checker component configures the processor to check the data originator certificate in order to determine a standard for whether the current measurements of the server sufficiently match the authenticated measurements of the server.

17. A method for using a modifiable server to reliably seal and unseal data according to a measurement of said server, said method comprising:

structuring the server to have a modifiable sandbox component for sealing and unsealing the data, and a non-modifiable checker component for enabling or disabling said sandbox component;

determining via the checker component whether the sandbox component complies with pre-determined standards; and

enabling the sandbox component to seal and unseal the data using a measurement of the checker component, if the checker component has determined that the sandbox component complies with the pre-determined standards; or

disabling the sandbox component, if the checker component has determined that the sandbox component does not comply with the pre-determined standards.

18. The method as claimed in claim 17, wherein the checker component determines whether the sandbox component complies with pre-determined standards by comparing current measurements of the sandbox component to authenticated measurements contained in manifests certified by authorized third parties.

19. The method as claimed in claim 18, wherein the checker component authenticates each manifest against a plurality of certificates stored in the checker component.

20. The method as claimed in claim 19, wherein the measurement of the checker component, used for sealing and unsealing the data, does not include the plurality of certificates.

21. The method as claimed in claim 19, wherein the checker component accepts only those certificates that are included in a third party list that has been certified by an evaluating party.

22. The method as claimed in claim 18, wherein the checker component determines that the sandbox component complies with the pre-determined standards based on whether the current measurements sufficiently match the authenticated measurements, and a sufficient match is made only if all of the authenticated manifests can be authenticated against the plurality of certificates stored in the memory device and only if all of the authenticated measurements match the current measurements.

23. The method as claimed in claim 18, wherein the checker component determines that the sandbox component complies with the pre-determined standards based on whether the current measurements sufficiently match the authenticated measurements, and a sufficient match is made only if at least all but one of the authenticated manifests can be authenticated against the plurality of certificates stored in the memory device and only if at least all but one of the authenticated manifests contains authenticated measurements that match the current measurements.

24. The method as claimed in claim 18, wherein the checker component determines that the sandbox component complies with the pre-determined standards based on whether the current measurements sufficiently match the authenticated

measurements, and a sufficient match is made only if at least a majority of the authenticated manifests can be authenticated against the plurality of certificates stored in the memory device and only if at least a majority of the authenticated manifests contain authenticated measurements that match the current measurements.

25. The method as claimed in claim 18, wherein the checker component determines that the sandbox component complies with the pre-determined standards based on whether the current measurements sufficiently match the authenticated measurements, and a sufficient match is made only if at least a majority of the authenticated manifests contain authenticated measurements that match the current measurements.

26. The method as claimed in claim 18, wherein the checker component determines that the sandbox component complies with the pre-determined standards based on whether the current measurements sufficiently match the authenticated measurements, and the current measurements cannot sufficiently match the authenticated measurements if at least one of the authenticated manifests comprises a disabled measurement that matches one of the current measurements.

27. The method as claimed in claim 17, wherein the checker component determines whether the sandbox component complies with standards corresponding to a data originator certificate that accompanies incoming data.

28. The method as claimed in claim 27, wherein the standards are set as authenticated measurements of the modifiable sandbox component, and the checker component obtains the authenticated measurements from one or more authenticated manifests identified by the data originator certificate.

ABSTRACT OF THE INVENTION

A modifiable server is utilized to reliably seal and unseal data according to a measurement of the server, by structuring the server to have a modifiable sandbox component for sealing, unsealing the data, and a non-modifiable checker component for enabling or disabling said sandbox component. The checker component determines whether the sandbox component complies with pre-determined standards. If the sandbox component is compliant, the checker component enables the sandbox component to seal and unseal the data using a measurement of the checker component. Otherwise, the checker component disables the sandbox component.

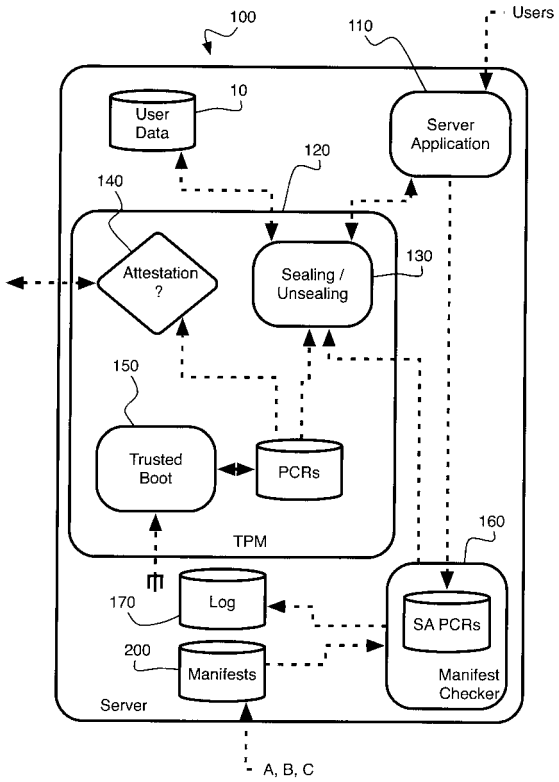


FIG. 1

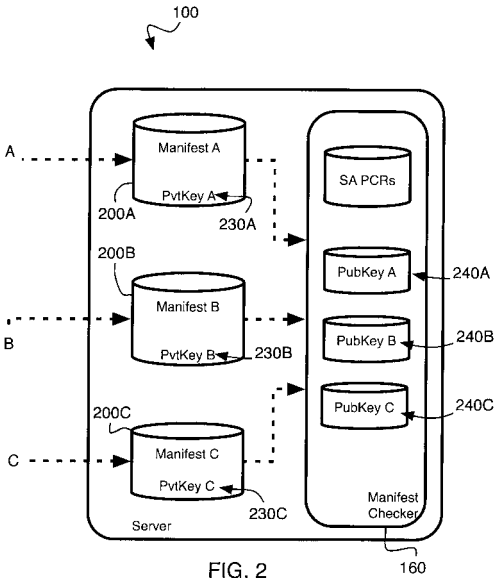


FIG. 2

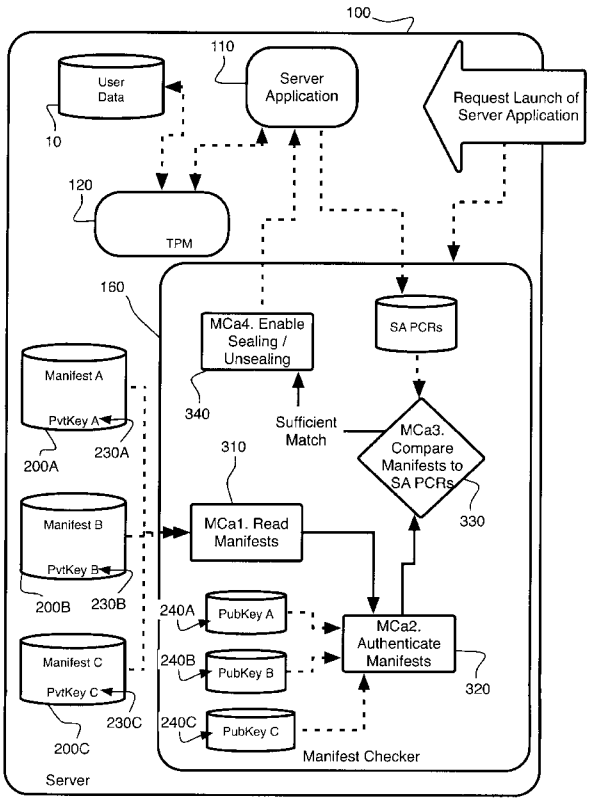


FIG. 3

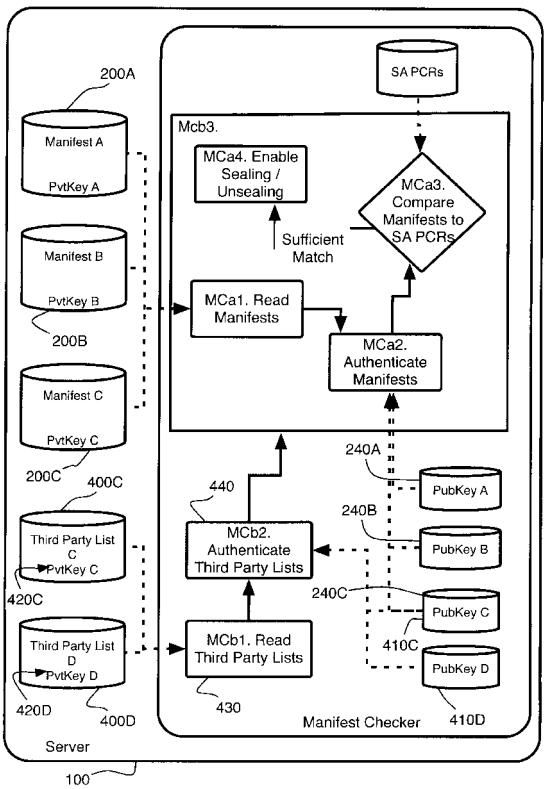


FIG. 4

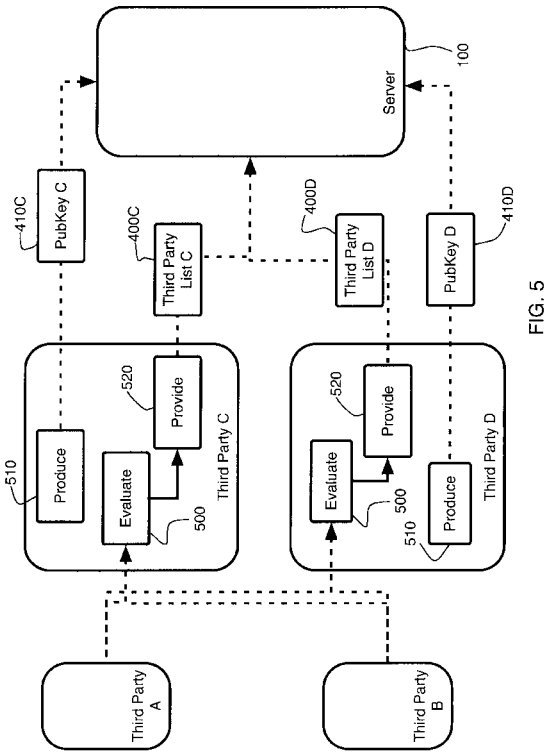


FIG. 5

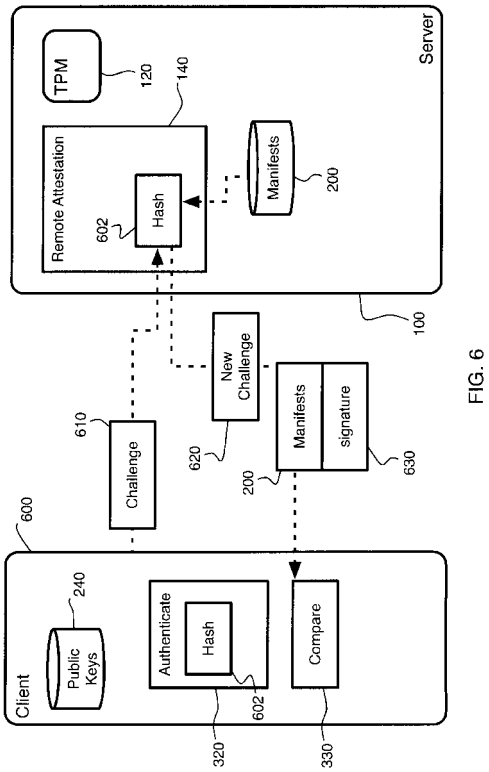


FIG. 6

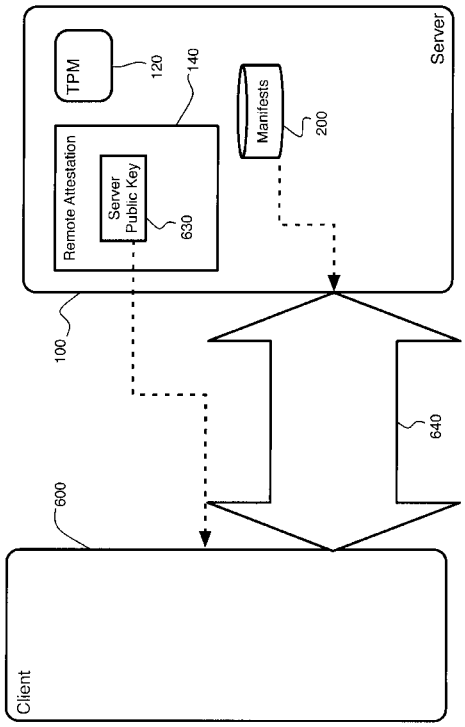


FIG. 7