TPM を用いた高信頼な REST ベースの Web サービス認証の提案と評価

細野 嵩史† 甲斐 賢‡ 手塚 悟†

†東京工科大学大学院 バイオ・情報メディア研究科 コンピュータサイエンス専攻
192-0982 東京都八王子市片倉町 1404-1
g2111045e7@st.teu.ac.jp tezuka@cs.teu.ac.jp

‡株式会社日立製作所 横浜研究所
244-0817 神奈川県横浜市戸塚区吉田町 292
satoshi.kai.nf@hitachi.com

あらまし 今日では多くのRESTfulなWebサービスが提供されている．しかしそれらWeb サービスの認証はソフトウェアベースで処理がなされ，署名に使う鍵やアクセスキーも数十文字程度の文字列で構成されているだけのことが少なくない．個人情報や資産情報をもWebアプリケーションに渡すWebサービスでは，ますますリソースの保護，責任の所在の明確化，課金処理は重要となり，Webアプリケーションに対してはより高信頼な認証を要求しなくてはならない．本稿ではTrusted Platform Moduleセキュリティチップを使ったハードウェアベースの鍵の管理と署名処理による高信頼なWebサービス認証方式を提案する．

Proposal and evaluation of Trusted REST Web Services Authentication using TPM

Takashi Hosono† Satoshi Kai‡ Satoru Tezuka†

†Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology
1404-1 Katakura-cho, Hachioji, Tokyo 192-0982, Japan
g2111045e7@st.teu.ac.jp tezuka@cs.teu.ac.jp
‡Yokohama Research Laboratory, Hitachi, Ltd.
292 Yoshida-cho, Totsuka-ku, Yokohama, Kanagawa 244-0817, Japan
satoshi.kai.nf@hitachi.com

Abstract In the web services by which the personal information and/or the assets are delivered to the web applications, protecting resources, clarifying the responsibility and billing would be increasingly important, therefore the trusted authentication must be required for the web applications. In this paper, we propose the trusted web services authentication by means of the hardware-based key management and digital signature using TPM security chip.

1　はじめに

近年，RESTful Web サービスや REST-RPC ハイブリッド の Web サービスが広く提供され，それら Web サービスは多くの Web アプリケーションによって利用されている．

Web アプリケーションの中には，Web サービスからエンドユーザの個人情報やデータを引き出して利用するサービスも少なくない．そのため Web サービスを使用した Web アプリケーションは，エンドユーザと Web サービスプロバイダから信頼されるものでなくてはならない．

Web アプリケーションは，エンドユーザが所有する Web サービスの認証 ID とパスワードやそれらと同等の役割を持つ認証トークンを取得することで，エンドユーザの権限で Web サービスを呼び出すことができる．これは悪意のある Web アプリケーションがエンドユーザの権限を利用し，不正に Web サービスからエンドユーザが所有する情報を取得，改ざんすることさえ可能であることを意味する．

また，ほとんどの Web アプリケーションは一般にはソースが公開されず，エンドユーザや Web サービスプロバイダからすると Web アプリケーションそのものの管理状況や運用が明確でないことが多い．Web サービスクライアントならばエンドユーザのコンピュータ上で実行されるため挙動を追い制御をすることはできるが，Web アプリケーションはエンドユーザにとっては信頼に値するかわからないコンピュータ上 で実行される上に挙動を追うことも制御もできない．Web サービスプロバイダにとってみても，信頼できない Web アプリケーション からエンドユーザの権限を持ってしてリクエストをされ，そのリクエストに応えることは Web サービスプロバイダの信用にも関わる問題である．

エンドユーザが Web アプリケーションを制御できない以上 ，Web サービスプロバイダが信頼しうる Web アプリケーション，それを実行するコンピュータ，そしてそれを管理する運営者にのみ，適切に Web サービスへのリクエストの認可を与えるべきである．

本研究では，Web サービスプロバイダと Web アプリケーションとを信頼のもとに結びつけるため，Trusted Platform Module を使った高信頼な REST ベースの Web サービス認証法を提案する．

2　課題と対策の検討

2.1 現状と課題

Web サービスが信頼のもとに Web アプリケーションからのリクエストを認可するためには，アイデンティティの確立と認証が重要である．

従来の Web サービスの認証，認可の方法は，BASIC 認証を使ったものから，Google の AuthSub[8]のように Web サービスによって独自に定められたもの[4][5]や，OAuth[7] のように規格化され多くの Web サービスから使われるものもある．

しかし，いずれの Web サービスの認証と認可の処理も，信頼性のために注目すべき鍵の生成や鍵の登録(保存)とロード，デジタル署名はソフトウェアベースで行われていた．クレデンシャルもまた信頼性が高いとは決していえないソフトウェアトークン(認証 ID やパスワードやアクセスキー)である．

これら従来の Web サービスの認証方法は，認証 ID とパスワードといったクレデンシャルが揃えば必ずしも本人が使っているとは限らないコンピュータからでも認証が可能となり，信頼性の高い認証とは言えない．

2.2 信頼とアイデンティティの確立

高信頼な Web サービスの認証，認可をするには，従来ソフトウェアのみであった認証処理をハードウェアベースのセキュリティチップを利用することで以下の項目をより確実なものとする必要があると考える．

- クレデンシャルが信頼できること
- クレデンシャルが強固であること
- 認証と認可のフローが信頼できること

本提案では，認証に用いるクレデンシャルが信頼でき，かつ強固なものとするためにハード

ウェアベースの鍵の生成と管理を，そして認証と認可のフローをより信頼できるものとするためにハードウェアベースのデジタル署名処理を行う．

また，あらかじめ Web アプリケーションを実行し Web サービスをリクエストすることになるコンピュータをも認証，認可の対象とする．コンピュータにも認可を与えることで，Web サービスがデータを提供し処理を許可するコンピュータを明確にすることができると考える．

3 提案手法

以下では，TPM を用いることで，鍵管理とデジタル署名をハードウェアベースで行った Web サービス認証が実現できることを示す．また，REST と実装に際して使用した TPM の概要を示し，提案する TPM を使った Web サービス認証の動作・フローについて説明を行う．

3.1 REST

REST(Representation State Transfer)は Roy Thomas Fielding 氏が博士論文で発表したアーキテクチャスタイル[1]で，Web サービス[2]では SOAP/WSDL(Simple Object Access Protocol, Web Services Description Language)ベースのものに代わる手段として広く使われるようになっている．本論では Web サービスを SOAP/WSDL ベースのものではなく，REST もしくは RESTful ベースのものを指す．

REST にはいくつかの制約があり，その一つとして”ステートレス”がある．提案する認証システムでも，この制約に従いセッションを使わずリクエストの都度，認証処理を行う．

3.2 Trusted Platform Module

TPM(Trusted Platform Module) は TCG (Trusted Computing Group)[3]がトラステッドコンピューティングを実現するために策定したセキュリティチップである．TPM は演算装置，暗号エンジン，鍵生成，乱数生成装置，メモリ領域，タンパ検出の機能を持ったハードウェアチップである．この TPM を使うことでハードウェアベースのセキュリティを得ることができトラステッドコンピューティングが実現できる．本提案で注目すべきは，TPM が鍵の生成やデジタル署名をハードウェアベースで行えることにある．

3.3 認証のフロー

提案するハードウェアベースの Web サービス認証は，図 1 にある TPM を搭載する Web アプリケーションと Web サービスを動かす 2 台のエンティティ間のシーケンスでなる．

認証のシーケンスは次の通りである．

(1) Web アプリケーションは TPM から Signing Key(RSA 鍵)をロードする
(2) Web アプリケーションはリクエスト内容に基づいたタイムスタンプを含む文字列を作成する
(3) 作成した文字列にロードした Signing Key の秘密鍵を使い RSA-SHA1 署名を作成する
(4) Base64 エンコードした RSA-SHA1 署名をリクエスト文字列に付加して，リクエストを Web サービスに送信する
(5) Web アプリケーションからの RSA-SHA1 署名付きリクエストを受け取った Web サービスは，リクエストに付加された RSA-SHA1 署名を検証する
(6) Web サービスは署名の検証結果に応じて Web アプリケーションに対しレスポンスを行う

3.4 実装の構成

Web サービスと Web アプリケーションは認証に必要な鍵の生成と署名処理を TPM の機能を呼び出して行う．Web サービスと Web アプリケーションで図 1 の認証フローを実現できるように，TPM の機能を呼び出す以下の関数を PHP の extension として実装した．このときのシステムレイヤは図 2 のとおりである．

- TPM を使った RSA 鍵の生成と鍵の登録
- TPM に登録した RSA 鍵の公開鍵の取得
- TPM を使った RSA-SHA1 署名
- TPM を使った RSA-SHA1 署名の検証

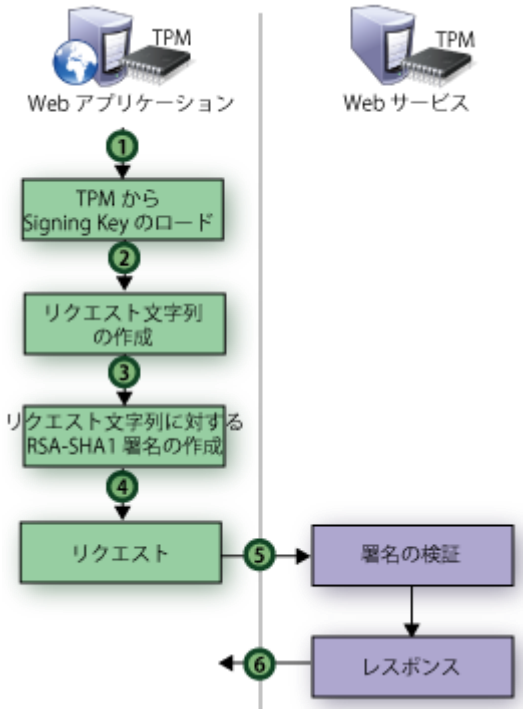

図 1 認証動作のフロー

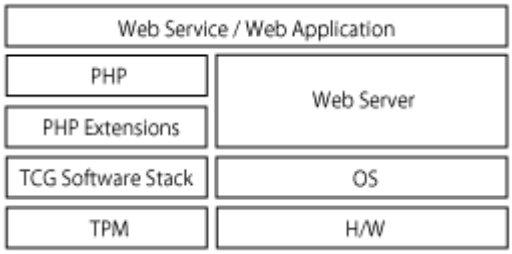

図 2 実装の構成

4 実証実験

Web サービス, Web アプリケーションそれぞれを実行する TPM を搭載したコンピュータを 2 台用意し実験を行った. 表 1 に実験環境を示す.

4.1 実験方法

提案した認証フローに従い以下の手順で実験を行った.

実験の手順

(1) 実装した TPM の機能を呼び出す PHP の関数を使い, Web サービスと Web アプリケーションそれぞれに認証機能を埋め込む

(2) Web アプリケーションを実行するコンピュータ上にある TPM で Attestation Identity Key を生成する

(3) 同じ TPM で Signing Key を生成し, 先ほど生成した Attestation Identity Key で署名する

(4) 署名された Signing Key のうち公開鍵を取り出し, Web サーバ側に渡す

(5) 提案する認証方法に従い, Web アプリケーションは Signing Key による RSA-SHA1 署名を付加したリクエストを Web サーバに送る

(6) Web サーバは Web アプリケーションから署名付きのリクエストを受け取り, 事前に渡された公開鍵とを使って TPM でリクエストの署名検証を行う

表 1 実験環境

Web サービス実行コンピュータ – マシン A	
CPU	Intel Core 2 Duo E7500 2.93GHz
Memory	4GB
TPM	Infineon 1.2.1.2
OS	CentOS 5.5 i386, 2.6.18-194
HTTP Server	apache 2.2.3
PHP	5.1.6
TCG Software Stack	TrouSerS 0.3.1
SSL/TLS Library	OpenSSL 0.9.8e
Web アプリケーション実行コンピュータ – マシン B	
CPU	Intel U1400 1.20GHz
Memory	4GB
TPM	Infineon 1.2.1.0
OS	CentOS 5.5 i386, 2.6.18-194
HTTP Server	apache 2.2.3
PHP	5.1.6
TCG Software Stack	TrouSerS 0.3.1
SSL/TLS Library	OpenSSL 0.9.8e

4.2 実験結果

TPM を使ったハードウェアベースの鍵の管理とリクエストに対しての署名によって高信頼な Web サービス認証ができることを確認できた. また, Signing Key のロードから署名付きのリクエス

トの送信までにかかる時間を 5 回測定し平均値を求めたところ 1.47 秒要した.

5 性能評価実験

従来の署名付きリクエストを使った Web サービス認証では，OpenSSL[6]のライブラリを使うなどして鍵のロードと署名をもソフトウェア処理のみで行い署名付きリクエストを作成していた.

実装したシステムでは，鍵のロードと署名には TPM を使い Web サービス認証を行うための署名付きリクエストを作成している.

Web サービス認証のための署名付きリクエストを作成するにあたって，ソフトウェア処理のみのものと TPM を使ったものとで，署名付きリクエストの作成速度にどれだけ差がでるのかを比較するための評価実験を行った.

5.1 実験環境

CPU 性能の異なる TPM を搭載した 2 台のコンピュータを用意し実験を行った. 表 1 に実験環境を示す.

5.2 実験用プログラムの用意

実装した TPM の機能を使った署名付きリクエスト作成プログラムの鍵のロードから署名までのプロセスは次の通りである.

(1) TPM から署名用の RSA 鍵をロード
(2) 署名の対象となる文字列の作成
(3) TPM を使った RSA-SHA1 署名の作成

評価実験では OpenSSL を使い，TPM の機能を使って実装した鍵のロードから署名までの部分をソフトウェア処理に置き換え比較対象となるプログラムを用意した. この処理を TPM から OpenSSL の関数に置き換えた署名付きリクエスト作成プログラムの鍵のロードから署名までのプロセスは次の通りである.

(1) 証明書のロード
(2) 秘密鍵のロード
(3) 秘密鍵が証明書に対応するかの確認
(4) 署名の対象となる文字列の作成
(5) RSA-SHA1 署名の作成

5.3 実験方法

先に示した 2 つのプログラムを用意し以下の手順で実験を行った.

実験の手順

(1) 鍵のロードから RSA-SHA1 署名までの処理を TPM の機能を使って行うプログラムと，その部分を OpenSSL の関数に置き換えたプログラムとで交互に 5 回ずつ実行速度を測定する.
(2) この作業を用意した 2 台のコンピュータで行う.

5.4 実験結果

提案した TPM の機能を使って鍵のロードから RSA-SHA1 署名までを行った署名付きリクエスト作成プログラムでの測定結果は表 2 のとおりとなり，このとき測定結果の平均値はマシン A(E7500@2.93GHz)で 1,470ms，マシン B(U1400@1.20GHz)で 1,381ms となった.

それと比べ，鍵のロードから RSA-SHA1 署名までを OpenSSL の関数に置き換えた署名付きリクエスト作成プログラムでの測定結果は表 3 のとおりとなり，このとき測定結果の平均値はマシン A (E7500@2.93GHz)で 8ms，マシン B(U1400@1.20GHz) で 49ms となった.

表 2 TPM を使った署名付きリクエスト作成時間

	マシン A [s]	マシン B [s]
1 回目	1.476	1.379
2 回目	1.472	1.384
3 回目	1.463	1.384
4 回目	1.459	1.379
5 回目	1.484	1.379

表 3 OpenSSL を使った
署名付きリクエスト作成時間

	マシン A [s]	マシン B [s]
1 回目	0.012	0.048
2 回目	0.008	0.050
3 回目	0.004	0.048
4 回目	0.008	0.048
5 回目	0.008	0.049

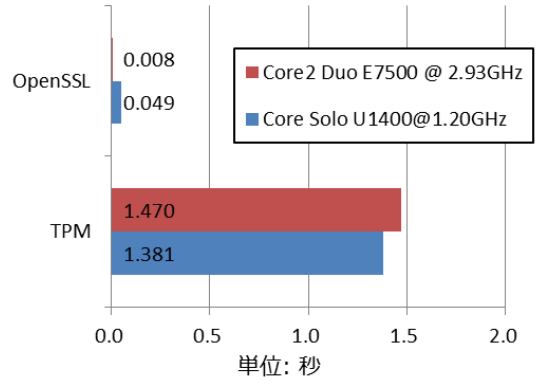

図 3 性能処理速度比較結果

いずれのコンピュータでも TPM を使った署名付きリクエスト作成プログラムの実行速度は，OpenSSL を使った署名付きリクエスト作成プログラムと比べ 1.3 秒以上遅いものとなっている．

提案する認証方式は，リクエストをするたびに 1.3 秒以上のオーバーヘッドがかかるため，リアルタイム性の高いシステムには向かないが，リアルタイム性が低く信頼性を重視すべきシステムでは実運用に耐えうると考える．

6 まとめ

TPM を使うことで高信頼な認証が実現できるとともに，Web サービスは信頼するコンピュータで実行する Web アプリケーションからのリクエストにのみ認可することができる．

Web サービスはデータの提供先をより明確にできることで，より重要なデータのやりとりを Web サービスで行えることが期待できる．またプラットフォームにひもづけたサービスの提供，プラットフォームに対しての課金，ユーザとプラットフォームのひもづけも可能となると考えられる．

7 今後の課題

TPM は高い処理性能を持たないことから，署名などを行う際に大きなオーバーヘッドがかかる．ハードウェアによる信頼性を保ちつつ，このオーバーヘッドをいかに軽減させるかが今後の課題である．

参考文献

[1] Roy Thomas Fielding, Representational State Transfer (REST) http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[2] Web Services Architecture Requirements http://www.w3.org/TR/wsa-reqs/

[3] Trusted Computing Group http://www.trustedcomputinggroup.org/

[4] Amazon Product Advertising API https://affiliate.amazon.co.jp/gp/advertising/api/detail/main.html

[5] NIFTY Cloud REST 共通パラメーターと認証方式 http://cloud.nifty.com/api/rest/authenticate.htm

[6] OpenSSL http://www.openssl.org/

[7] OAuth http://oauth.net/

[8] AuthSub for Web Applications http://code.google.com/intl/ja/apis/accounts/docs/AuthSub.html

[9] R. Sailer, T. Jaeger, X. Zhang, L. van Doorn: Attestation-based Policy Enforcement for Remote Access. 11th ACM Conference on Computer and Communications Security (CCS) 2004, Washington, 2004 年 10 月

[10] 文 栄光，塩谷 亮太，五島 正裕，坂井 修一，情報漏洩防止のためのプラットフォーム認証,電子情報通信学会研究報告，CPSY 2009，pp. 13−18 (2009).