IEICE TRANS. INF. & SYST., VOL.E99–D, NO.4 APRIL 2016

PAPER Special Section on Cyberworlds

SSL Client Authentication with TPM

Shohei KAKEI[†a)], ***Nonmember***, **Masami MOHRI**[††], **Yoshiaki SHIRAISHI**[†], ***and*** **Masakatu MORII**[†], ***Senior Members***

SUMMARY TPM-embedded devices can be used as authentication tokens by issuing certificates to signing keys generated by TPM. TPM generates Attestation Identity Key (AIK) and Binding Key (BK) that are RSA keys. AIK is used to identify TPM. BK is used to encrypt data so that specific TPM can decrypt it. TPM can use for device authentication by linking a SSL client certificate to TPM. This paper proposes a method of an AIK certificate issuance with OpenID and a method of the SSL client certificate issuance to specific TPM using AIK and BK. In addition, the paper shows how to implement device authentication system using the SSL client certificate related to TPM.

key words: *Trusted Platform Module, public key certificate, OpenID, SSL, client certificate*

1. Introduction

In the password-based user authentication which is used in general Web systems, it is hard to notice to plagiarize the password, and it is difficult to notice to lose the one before accessing a service. If not "*something you know*" but "*something you have*" is used as a factor of the authentication, it is easy to notice to plagiarize and lose the password.

The SSL client authentication is an authentication method using "something you have". A client signs data using a secret key corresponding to a public key contained in a client certificate. The client sends a signature of the data and the client certificate to the server. The server authenticates the client by verifying the signature using the client certificate. It is possible to improve security if the signing key is stored in a hardware token such as IC card or USB token. The SSL client authentication is used primarily for services in which an advanced security is needed.

This paper aims to establish the SSL client authentication infrastructure so as to replace the password-based user authentication. It is desirable to use different certificates by each service from the view point of privacy. The certificate that includes a personal data or a unique data causes to compromise a user anonymity. In the proposed system, the service provider acts as a certificate authority (CA), and issues the certificate that is valid only in a specific service to the user. It is unnecessary to use of the password, if the client certificate can be used. However, some other mechanism of personal identity verification is needed in order to issue the client certificate. OpenID [1] is open and simple web single sign-on protocol. If each server uses not the password-based user authentication but an OpenID-based one, it is possible to reduce the password that the user manages by OpenID Provider storing the password of the user. Additionally, a cost of managing the password is eliminated. Moreover, a leakage risk of the password because of dispersing the password can be reduced, and the user anonymity is protected.

A security chip TPM (Trusted Platform Module) [2] is specified by Trusted Computing Group [3]. TPM has been embedded in laptop computers, mobile devices and so on and it is widely used as a leading security chip. Because TPM has a feature of securing a secret key corresponding to a public key contained in a client certificate, it is possible to use TPM just like IC card or USB token without additional devices such as a card reader or a USB device. If the SSL client authentication uses TPM, an assurance that the signing key is secured by TPM is needed. A certificate-based authentication can use two TPM keys: one is a RSA signing key AIK (Attestation Identity Key) and the other is a RSA encrypting key BK (Binding Key). AIK is the signing key that is related a certain TPM, and the signed data by AIK is proved that the data is generated by TPM for a third party.

This paper proposes a scheme for SSL client authentication, which consists of two methods: one is an OpenID-enabled issuing and the other is TPM-enabled SSL client authentication. In the proposed scheme, at first, an AIK certificate is issued to the user that is authenticated with OpenID. Secondly, the client sends AIK and BK that is signed by AIK. The server verifies AIK for checking that AIK is related to a certain TPM, and verifies the signature of BK with AIK for checking that BK is related to a certain AIK. According to the above verification, BK is related to a certain TPM, and it is possible to check that the signing key is secured in that TPM. By encrypting the client certificate and the signing key, it is possible to issue the client certificate and the signing key to only the certain TPM.

The paper is organized as follows. Section 2 describes about TPM and the keys. Section 3 indicates issues for applying TPM to SSL client authentication. Section 4 summarizes assurance levels of client certificate. Section 5 proposes an AIK certificate issuance method using OpenID, and Sect. 6 proposes the client certificate issuance method that is related the certain TPM. Section 7 shows

Manuscript received August 24, 2015.
Manuscript revised December 1, 2015.
Manuscript publicized January 28, 2016.

[†]The authors are with Kobe University, Kobe-shi, 657–8501 Japan.
[††]The author is with Gifu University, Gifu-shi, 501–1193 Japan.
a) E-mail: kakei.shohei@nitzlab.com

DOI: 10.1587/transinf.2015CYP0012

implementations of these methods and Sect. 8 shows a device authentication system using TPM. Section 9 evaluates these methods and Sect. 10 concludes the paper.

2. Keys of TPM and Certificates

2.1 Keys of TPM

TPM has tamper resistance that prevents tampering of data in TPM and a function for generating a RSA key. The RSA key has a usage limit, and the limit is determined at time of generating the key. All keys managed by TPM have an attribute designation of "migratable" and "non-migratable". A private part of a migratable key can migrate to another TPM. On the other hand, a private part of a non-migratable key is related permanently to TPM that generates the non-migratable key. The following four TPM keys are handled in this paper.

EK (Endorsement Key): EK is a RSA key that is generated by a TPM vendor at time of shipping, and is only one in one TPM. A private part of EK is securely stored in a tamper area against leakage. A public key certificate of EK (EK certificate) secured in TPM proves that the private part of EK is installed in TPM. One of purposes of using EK is to prove that AIK is generated by a certain TPM, and is used to encrypt a public key certificate of AIK (AIK certificate) to enable the only TPM that is used to generate the AIK to decrypt the AIK certificate.

SRK (Storage Root Key): SRK is secured in a tamper area of TPM against leakage. RSA keys that are generated with TPM can be used as a non-migratable key by encrypting with SRK. At that time, SRK is called a parent key for these RSA keys. SRK is only one in one TPM. TPM has a feature of resetting SRK. The keys related in old SRK are unusable if SRK is reset.

AIK (Attestation Identity Key): AIK is a non-migratable signature key, and is wrapped by SRK as the parent key. AIK cannot be taken out from TPM, and is exclusively used to sign data originated by TPM. In the specification of TPM, it is impossible to use AIK until the public key certificate of the corresponding AIK (AIK certificate) is issued. It is hard to analyze AIK because AIK is stored in an external storage under encryption with TPM.

BK (Binding Key): BK is an encryption key for encrypting and decrypting data. BK is only available as the key for decryption in the specified TPM if BK is the non-migratable key.

It is impossible to use TPM in another device because TPM is not a separate hardware token [4], then AIK is secured in TPM. This implies that a device can be identified by verifying the signature that is generated by AIK. It is also able to verify BK that is generated by TPM because BK is related to AIK by using a TPM command TPM_CertifyKey [5] which signs BK with AIK.

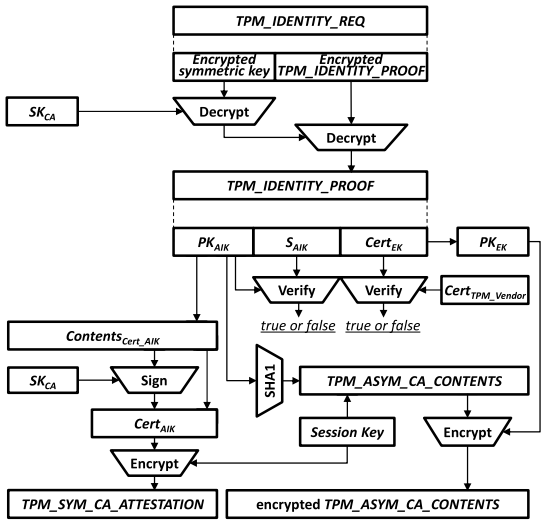

Fig. 1 The process of generating the AIK certificate

2.2 Issuing the AIK Certificate

In a protocol specified by TCG, the way for CA to verify that AIK is related in TPM is standardized as shown in Fig. 1.

In the following paper, let SK_A, PK_A, S_A, and $Cert_A$ be a secret key, a public key, a signature and a public key certificate, respectively. These symbols are identified by A. Let *SHA1*, *Encrypt*, *Decrypt*, *Sign*, and *Verify* be cryptographic algorithms. $SHA1(A)$ computes a sha-1 hash value of A, and let H_A be a hash value of A. $Encrypt(A, PK_B)$ encrypts A with PK_B, and E_A indicates an encrypted A. $Decrypt(A, SK_B)$ decrypts A with SK_B. $Sign(A, SK_B)$ signs A with SK_B, and let S_A be a signature of A. $Verify(S_A, H_A, PK_B)$ verifies S_A with H_A and PK_B. A symbol "||" is data concatenation. For example, $A||B$ indicates that A and B are concatenated.

A user generates *TPM_IDENTITY_REQ* that contains an encrypted symmetric key with PK_{CA} and an encrypted *TPM_IDENTITY_PROOF* with the symmetric key. The CA gets *TPM_IDENTITY_PROOF* with a secret key SK_{CA} that contains a public key of AIK (PK_{AIK}), a signature of PK_{AIK} (S_{AIK}) by a secret key of AIK (SK_{AIK}) and an EK certificate ($Cert_{EK}$). The CA verifies S_{AIK} by PK_{AIK} and verifies $Cert_{EK}$ by a public key certificate of a TPM vendor ($Cert_{TPM\_Vendor}$). If the verifications are success, it is proved that the SK_{AIK} and a secret key of the EK (SK_{EK}) are secured in TPM owned by the user. The CA generates an AIK certificate ($Cert_{AIK}$) that corresponds to PK_{AIK} and a session key. Then, The CA encrypts $Cert_{AIK}$ by the session key and generates *TPM_SYM_CA_ATTESTATION* that contains the encrypted $Cert_{AIK}$. Additionally the CA generates *TPM_ASYM_CA_CONTENTS* that consists of the session key and a SHA1-digest of PK_{AIK} and make encrypted *TPM_ASYM_CA_CONTENTS* by encrypting the *TPM_ASYM_CA_CONTENTS*. The CA sends the *TPM_SYM_CA_ATTESTATION* and the encrypted

TPM_ASYM_CA_CONTENTS to the user. The user decrypts the encrypted *TPM_ASYM_CA_CONTENTS* by SK_{EK} to get the session key and decrypts *TPM_SYM_CA_ATTESTATION* to get $Cert_{AIK}$.

In the following paper, *TPM_IDENTITY_REQ* is referred to as *AIK_CERT_REQ*, and *TPM_SYM_CA_AT TESTATION* and encrypted *TPM_ASYM_CA_CONTENTS* are referred to as *ENC_AIK_CERT*.

3. SSL Client Authentication and TPM

SSL client authentication can be used as one of the authentication in Web services. A server is able to authenticate the client by verifying a signature. A secondary storage such as hard disk drive is used as a storage device of the secret key for signing.

TPM can be used as a device for securely storing a client certificate and a secret key as same as IC card and USB token. A leading PC vendor provides a tool for storing the client certificate to TPM [7]. This feature allows the secret key to be stored more securely than secondary storages. If follows from this that a spoofing caused by a leakage of the secret key is prevented. However, it is not clear that the server can identify whether the client has TPM or not. That is the reason the client certificate can be used without storing in TPM. If the server can confirm that the certificate is stored in TPM, TPM can use to SSL client authentication.

3.1 Privacy

SSL client is validated with an X.509 certificates. The certificate may include personal information such as a user name or an e-mail address. Even if these information are not contained, some unique information such as a serial number is contained. Flowing the information through public networks can increase risks of crawling and leakage. That is, even if a personal information is not included in a certificate, the user activity could be tracked by collecting client certificates. For example, TLS protocol has a disclosure issue of the client certificate during the initial handshake. IETF Network Working Group publishes a draft to solve it by encrypting the client certificate [6]. Due to compromise user anonymity, one client certificate should be limited to one specific service and the user should use different client certificates for each services. Then, a certificate issued by a global CA service is not suitable, because the service provider cannot know whether the certificate is used in another service.

In initial issuance of a certificate, identity validation of the client is needed. In general web services, the password-based user authentication is one of the major methods. However, it is known that password list attack increases a risk such that the user anonymity is compromised by sharing the same ID and password among several services [8]. The leaked ID and password from a service can be used to login the other services. If a service does not store pairs of ID and password, there is no privacy risk of leaking account information. OpenID can provide a mechanism of the user authentication without registering the ID and password to the service.

3.2 Authenticity

A client should be securely stores a private signing key corresponding to a SSL client certificate includes a public verification key, because a SSL server authenticates a SSL client on trust that the private signing key is not disclosed. If this trust is compromised, the server is not able to determine whether the client is valid or not.

A private signing key is generally stored in a hard disk drive. Due to a removal medium, it may be copied to another medium. Then, it is not guaranteed that the key is under the control of the client. There is TPM as a suitable hardware for securing the key as IC card and USB token. The data in TPM is protected by strict access control, and TPM is kept under hardware-based protection. If a server can verify whether the client has TPM and the private signing key is installed in TPM, a SSL client certificate protected by TPM can be used for client authentication even on a public network with a large and unspecified number of users. Unlike IC card and USB token, since TPM is directly mounted on a mother board, the use of the signing key is restricted in environments of the owner of a TPM-installed computer.

Generating a signing/verification key pair in issuing a client certificate can take two cases; one is in a client, the other is in a CA. In case of the former, the client certifies ownership of the signing key by requesting a CSR to the CA. In case of the latter, the client only receives the encrypted signing key that is packed with the SSL client certificate into a PKCS#12 file encrypted with a password. If a client is to avoid a possession of a raw singing key, the service provider with the CA should generate a key pair. But there is no guarantee whether a SSL client is valid if we would take a naive exchange method. The certificate can be available by only the valid client who has TPM by encrypting the certificate with BK signed by AIK.

The data encrypted by the public key of BK is decrypted by only the secret key of BK. If it is verified that BK is secured in a certain TPM, the server can issue the client certificate under encryption to only the certain TPM. AIK is available to verify whether BK is secured in the certain TPM or not because AIK is exclusively used to sign data so as to originate in TPM. From the reason TPM has the feature that is unable to use the other devices, TPM is available to authenticate a device if the client certificate can be issued only to the specified TPM.

Towards implementation of a TPM-enabled SSL client authentication, this paper focuses on how to use BK properly. Before the proposal, the first point that we have to discuss is that the proposed scheme uses what kind of client certificate.

4. Assurance Levels of Certificates of a Client Certificate Issuance Service

According to operation rules [9], [10] of a client certificate issuance service for individual, the client certificate has to be issued after verifying the existence of the user. In [9], [10], the way of verifying existence and authenticating the client is stipulated depending on the assurance levels of the issued certificate.

There are Class 1 (lowest class) to Class 3 as shown in Table 1. Class 1 does not need a formal individual name since CA issues the client certificate without verifying the individual name of the user. The client certificate of Class 2 and Class 3 is issued after verifying information that is registered in a credit agency or that is an identification that is issued by a government.

Because this paper aims to establish the SSL client authentication infrastructure so as to replace the password-based user authentication, the proposed scheme takes the use of certificate corresponding to Class 1 account.

5. The Proposed Method of Issuing the AIK Certificate Comparable with the Class1 Certificate

From the point of view of privacy, it is desirable to use different certificates for each services. In each of the classes, the password that the service provider and the user manage is increased by a number equal to a number of issuing the client certificate.

Table 1 Class of certification and assurance level

Class of Certificate	Assurance Level	Available Subscriber	How to Verify and Authenticate Applicant
Class 1	Low	Anonymity or Pseudonymity	The applicant is required to demonstrate control of any e-mail address to which the certificate relates.
Class 2	Middle	True Personal or Organizational Name	In addition to the way of Class 1, a legible copy of a valid government issued national identity document or photo ID are required.
Class 3	High	True Personal or Organizational Name	In addition to the way of Class 1 and Class 2, a face to face meeting is required.

Then, this paper proposes a method using OpenID for issuing the AIK certificate comparable with the class1 certificate without increasing the password managed by the PCA and the user. The following presents the role of each entity in this method and the process of issuing the AIK certificate.

The Role of Each Entity

User: It is a person who operates UA.
User Agent (UA): It operates TPM and requires PCA to issue the AIK certificate.
TPM: It is a security chip that generates an AIK.
Privacy CA (PCA): It is a Trusted Third Party and a Relying Party (RP) in OpenID. PCA pays attention to the user anonymity and issues the AIK certificate. RP is a server to support login using OpenID and provides a service to an UA depending on a result of authentication that is received from OP. PCA executes processing for issuing the AIK certificate since requesting from the UA.
OpenID Provider (OP): It is a server that authenticates UA. OP notifies PCA that UA is authenticated.

The Process of Issuing the AIK Certificate

As a preparation, the user makes an account of OP and installs a program to operate TPM. The following presents the process of issuing the AIK certificate that is shown in Fig. 2.

1. User accesses a web site of PCA using UA and selects OP that is used at login process.
2. PCA redirects UA to the selected OP, and returns a login result to UA if OP authenticates User.
3. User executes requesting the AIK certificate.
4. UA generates a symmetric key (K_{sym}) and $TPM\_IDENTITY\_PROOF$, and generates AIK_CERT_REQ by encrypting $TPM\_IDENTITY\_PROOF$ with K_{sym}.

$$H_{AIK} := SHA1(PK_{AIK})$$
$$S_{AIK} := Sign(H_{AIK}, SK_{AIK})$$
$$TPM\_IDENTITY\_PROOF := PK_{AIK} || S_{AIK} || Cert_{EK}$$
$$E_{TPM\_IDENTITY\_PROOF} := Encrypt(TPM\_IDENTITY\_PROOF, K_{sym})$$
$$E_{Ksym} := Encrypt(K_{sym}, PK_{PCA})$$

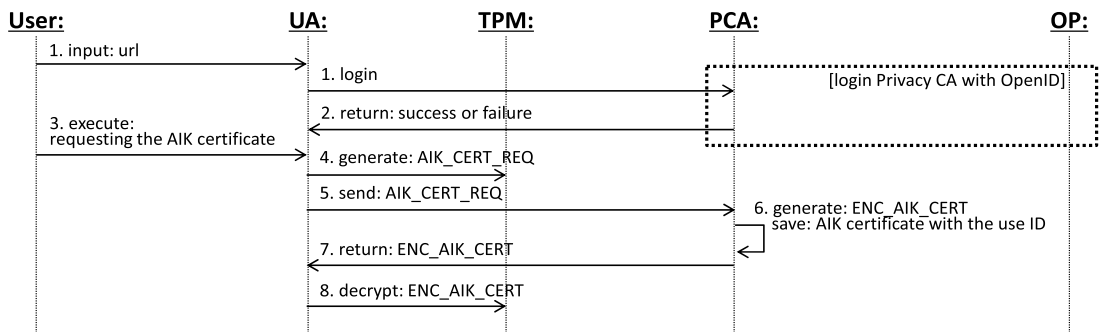

Fig. 2 The sequence of issuing the AIK certificate with OpenID

$$AIK\_CERT\_REQ := E_{TPM\_IDENTITY\_PROOF} || E_{Ksym}$$

5. UA sends *AIK_CERT_REQ* to PCA, and requires issuing the AIK certificate.
6. PCA generates *ENC_AIK_CERT*.

$$TPM\_SYM\_CA\_ATTESTATION := Encrypt(Cert_{AIK}, Session\_Key)$$
$$H_{AIK} := SHA1(PK_{AIK})$$
$$TPM\_ASYM\_CA\_CONTENTS := H_{AIK} || Session\_Key$$
$$E_{TPM\_ASYM\_CA\_CONTENTS} := Encrypt(TPM\_ASYM\_CA\_CONTENTS, PK_{EK})$$
$$ENC\_AIK\_CERT := TPM\_SYM\_CA\_ATTESTATION || E_{TPM\_ASYM\_CA\_CONTENTS}$$

7. PCA sends *ENC_AIK_CERT* to UA.
8. UA decrypts *ENC_AIK_CERT* with TPM.

$$Session\_Key := Decrypt(E_{TPM\_ASYM\_CA\_CONTENTS}, SK_{EK})$$
$$Cert_{AIK} := Decrypt(TPM\_SYM\_CA\_ATTESTATION, Session\_Key)$$

ENC_AIK_CERT is encrypted by the session key, and the session key is encrypted by PK_{EK}. Additionally, Because of verifying $Cert_{EK}$ by $Cert_{TPM\_Vendor}$, it is proved that SK_{EK} is secured in TPM. For this reason, *ENC_AIK_CERT* encrypted by *Session_Key* is decrypted by only TPM that secures EK used in this process.

Since the authentication processing is executed by OP, UA can log in to PCA without registering the password to PCA. In addition, since the existence of the user is verified at registering for OP, PCA treats UA as if the existence is verified.

6. The Proposed Method of Issuing the SSL Client Certificate for Specific TPM

By combining AIK with BK, an encrypted data that is assured to be decrypted by only a specific TPM can be generated. In this section, User requests a SSL client certificate with AIK certificate. It is unnecessary to prepare the same number of AIK certificate as the SSL client certificate and User obtains the AIK certificate in advance.

The following presents the role of the each entity in this method and the process of issuing the SSL client certificate that is shown in Fig. 3.

The Role of Each Entity

User: It is a person who operates UA.
User Agent (UA): It operates TPM and requires SP to issue the SSL client certificate.
TPM: It is a security chip for distinguishing individually the user by CA. TPM secures the SSL client certificate.
Service Provider (SP): It is a server that provides services for User. In this issuing the SSL client certificate process, SP acts as a CA to issue a SSL client certificate to User.
OpenID Provider (OP): It is a server that authenticates UA. OP notifies SP that UA is authenticated.

A https channel is used in the following processes.

The Process of Issuing the SSL Client Certificate

1. User accesses a web site of SP using UA and selects OP that is used at login process.
2. SP redirects UA to the selected OP, and returns a login result to UA if OP authenticates User.
3. User executes generating BK and generating S_{BK} of

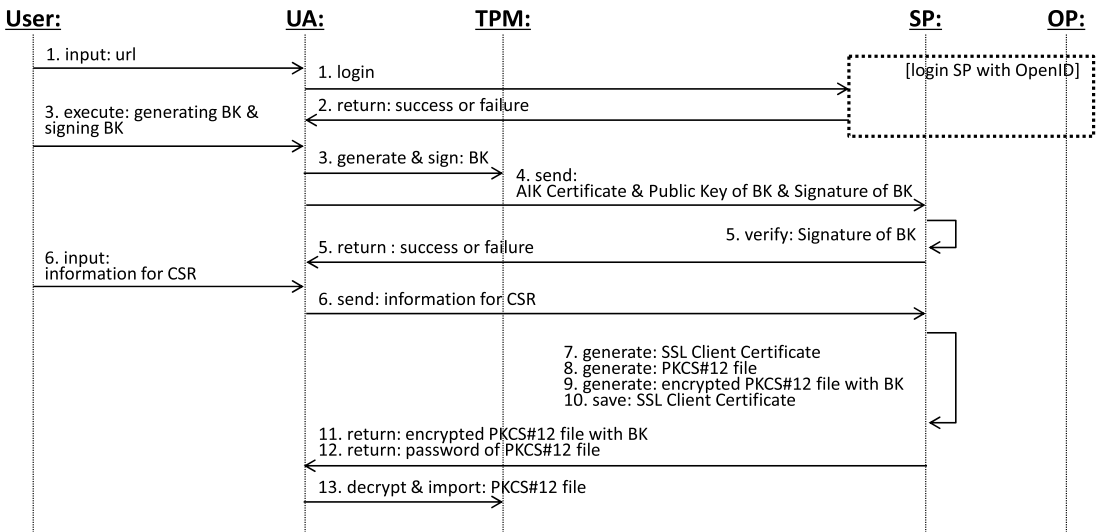

Fig. 3 The sequence of issuing the SSL client certificate

PK_{BK} with AIK using UA.

$$H_{BK} := SHA1(PK_{BK})$$
$$S_{BK} := Sign(H_{BK}, SK_{AIK})$$

4. UA sends PK_{BK}, S_{BK} and $Cert_{AIK}$ to SP.
5. SP verifies S_{BK} with PK_{BK} and $Cert_{AIK}$, and returns the verification result.

$$H_{BK} := SHA1(PK_{BK})$$
$$Verify(S_{BK}, H_{BK}, PK_{AIK})$$

6. If the verification is success, User inputs information for CSR (Certificate Signing Request), and UA sends them to SP.
7. SP generates a key pair (PK_{SSL}/SK_{SSL}) and CSR for the SSL client certificate with the information received from UA, and generates the SSL client certificate ($Cert_{SSL}$) including PK_{SSL}.
8. SP generates a password (PW) randomly, and combine the $Cert_{SSL}$ and SK_{SSL} into a file of PKCS#12 format ($F_{PKCS\#12}$) with the password.
9. SP encrypts the PKCS#12 file with PK_{BK}.

$$E_{SSL} := Encrypt(F_{PKCS\#12}, PK_{BK})$$

10. SP saves $Cert_{SSL}$ in the storage for future reference.
11. SP returns E_{SSL} to UA.
12. SP returns PW to UA.
13. UA gets $F_{PKCS\#12}$ by decrypting E_{SSL} with SK_{BK}, and imports $F_{PKCS\#12}$ to TPM with PW.

$$F_{PKCS\#12} := Decrypt(E_{SSL}, SK_{BK})$$

SP issues the SSL client certificate to only User who is verified the existence and has TPM. In step.1 and step.2, SP verifies the existence of User with OpenID. In step.4 and step.5, SP verifies that User has TPM with the AIK certificate issued by PCA in advance. SP can verify whether User is valid or not, because SP trusts PCA and OP that are Trusted Third Parties.

7. Implementation of the Proposed Methods

The methods of issuing the AIK certificate and the SSL client certificate showed in Sect. 5 and Sect. 6 are implemented in the architecture shown in Fig. 4. The following presents implemented modules.

TPM Access Program (TAP): TAP is UA to access TPM, and is implemented as a GUI application that provides features for requesting the AIK certificate and the SSL client certificate. TAP calls each sub module and stores data as a file in the storage. TAP uses IAIK jTSS [11] that is a Java [12] implementation of TCG Software Stack (TSS) [13] for accessing TPM. TSS is a software specification for providing standardized APIs for accessing features of TPM. TSS command list that are used in TAP are showed in Table 2. TAP uses IAIK TCCert [14] that provides functions for using extended X.509 format certificates such as the AIK certificate, the EK certificate and so on. In the process of encrypting and signing, IAIK JCE [15] and IAIK CMS [16] are used.

Entry Point (EP): EP is a web page group that provides operations allowed to be executed for a client. EP is composed an Apache Server [17] that processes user authentication by OpenID and a GlassFish Server [18] that processes issuing certificates. The client accesses EP and requests user authentication or issuing the AIK certificate or the SSL client certificate. According to the process requested by the client, EP calls each module.

User Authentication Module (UAM): UAM authenticates the user in accordance with OpenID protocol. UAM redirects the user to OP selected by the user, and accepts authentication result of OP. In this implementation, Google and Yahoo can be used as OP. This module is implemented as a PHP [19] code that runs on the Apache server.

AIK Certificate Issuing Module (ACIM): ACIM generates

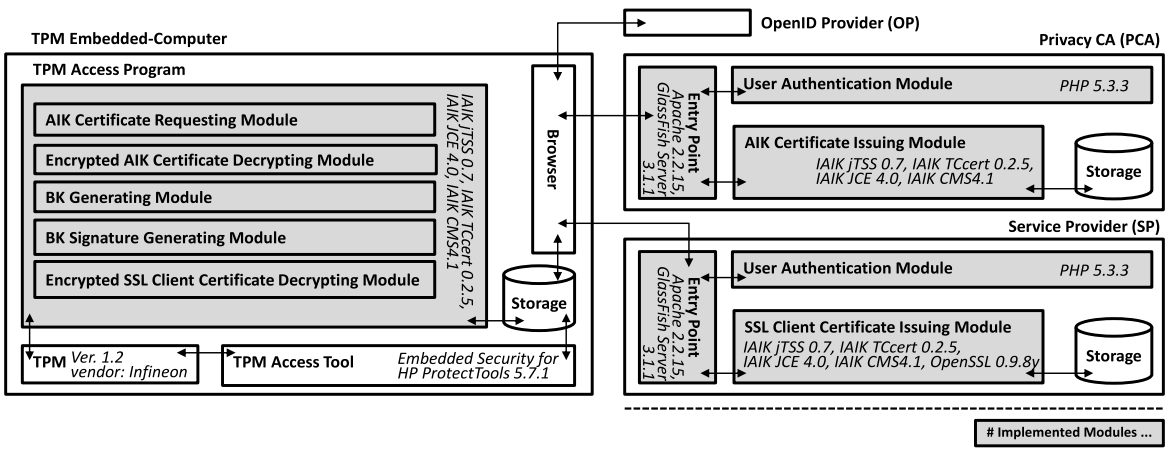

Fig. 4 The sequence of issuing the SSL client certificate

Table 2 TSS command list used by the TPM access program

	Tspi_Context_Connect	Tspi_Context_CreateObject	Tspi_Context_GetKeyByUUID	Tspi_Context_GetTPMObject	Tspi_Context_RegisterKey	Tspi_GetAttribData	Tspi_Key_CertifyKey	Tspi_Key_CreateKey	Tspi_Key_GetPubKey	Tspi_Key_LoadKey	Tspi_Policy_AssignToObject	Tspi_Policy_SetSecret	Tspi_SetAttribData	Tspi_TPM_ActivateIdentity	Tspi_TPM_CollateIdentity	Tspi_Data_Unbind
AIK Certificate Requesting Module	✓	✓	✓	✓	✓	✓					✓	✓	✓		✓	
Encrypted AIK Certificate Decrypting Module		✓	✓							✓	✓	✓		✓		
BK Generating Module		✓						✓	✓	✓	✓					
BK Signature Generating Module							✓									
Encrypted SSL Client Certificate Decrypting Module		✓											✓			✓

ENC_AIK_CERT from *AIK_CERT_REQ* in accordance with the process showed in Fig. 1. *ENC_AIK_CERT* is downloaded by the client through EP. PCA stores lists of the AIK certificate and the user for future reference. In this module, encrypting processing, signing processing, hash value computing processing and certificate generating processing are needed. This module is implemented as a Java code that runs on the GlassFish server using IAIK jTSS, IAIK TCCert, IAIK JCE and IAIK CMS.

SSL Client Certificate Issuing Module (SCCIM): SCCIM generates the SSL client certificate using OpenSSL and encrypts it by BK of the client. In the proposed method, the BK-owner is identified by the signature of BK by AIK and the list of AIK and the user. Information of the user that is needed for generating the SSL client certificate is input in a web form, and OpenSSL command is executed using the information. The encrypted SSL client certificate is downloaded through EP.

In the both methods, the user generates files uploaded to PCA and SP by TAP and uploads the files by the browser. TAP and the browser are UA. These modules showed in this section are implemented in accordance with the process showed in Fig. 2 and Fig. 3.

In issuing the AIK certificate, at first, the user accesses PCA using the browser and is authenticated with OpenID. If the authentication process is success, PCA displays an upload page for *AIK_CERT_REQ*. Secondly, the user generates *AIK_CERT_REQ* using TAP. If *AIK_CERT_REQ* is uploaded in the upload page, *ENC_AIK_CERT* is downloaded. At last, the user decrypts *ENC_AIK_CERT* using TAP, and TAP stores the AIK certificate as a file.

In issuing the SSL client certificate, the user executes the process of generating BK using TAP. TAP generates BK, and signs the public key of BK by AIK, and stores the public key of BK and the signature as files in the storage. The user accesses SP, and uploads the stored files, and inputs the information used for generating the SSL client certificate through the web form, then the SSL client certificate encrypted by the public key of BK is downloaded. The user executes decrypting the encrypted SSL client certificate, and the decrypted SSL client certificate is stored in the storage.

8. Construction of the SSL Client Authentication System Using a Key Stored in TPM

The SSL client certificate can be issued upon verifying the owner of TPM by using AIK and BK. In this section, the way of the SSL client authentication using TPM is shown. The architecture of the SSL client authentication system is showed in Fig. 5.

On the service user side, the SSL client certificate is used upon securing it by TPM. For this purpose, TPM Professional Package that is provided by leading PC vendors can be used. This system uses Embedded Security for HP ProtectTools 5.7.1 that is TPM Professional Package provided by HP. Embedded Security for HP ProtectTools is referred to as TPM Access Tool in the following paper. In installation of the SSL client certificate, it is only necessary to execute according to TPM Access Tool. Under an intermediation of TPM Access Tool, the SSL client authentication can be done while the SSL client certificate is secured by TPM. Internet Explorer or Google Chrome that support SSL, MS-CAPI and PKCS#11 can be used for the browser.

On the service provider side, a device authentication server can be built by adding a SSL client authentication module such as mod_ssl [20] to httpd for enabling a function of the SSL client authentication. The special configurations are not needed.

After installation of the SSL client certificate, as the service user accesses the service provider, the browser displays lists of the available SSL client certificate. If the certificate is selected, TPM Access Tool requests a password for using the secret key of the SSL client certificate. When the service user inputs the password registered at time of activating TPM, the SSL client authentication is executed.

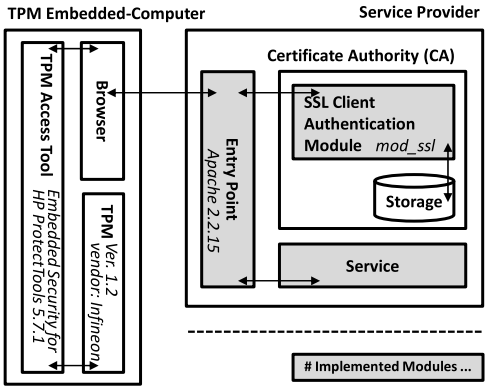

Fig. 5 The architecture of the SSL client authentication system

9. Evaluations

9.1 User Anonymity

The comparison of privacy risks among authentication schemes is showed in Table 3. We evaluate these schemes by assuming that an attacker steals the user information managed by the service provider. In order to steal the user information, the attacker identifies the user at first. Then, the attacker steals the user information of the identified user.

Firstly, in the password-based user authentication, if there are user accounts among several services that can be accessed with the one pair of ID and password, it is considered that these user accounts are owned by one user. In the assumption, since the attacker is able to collect the ID and password, the attacker can conjecture the owner of the user account. In addition, the attacker is able to take over the user account and steal the personal information.

Secondly, in the basic SSL client authentication, since a client certificate includes a personal information or a unique information, the attacker identify a user by checking a personal information or collecting a unique information. However, the attacker cannot login the user account because the verification key is owned by the user. Consequently, the attacker can identify which services are used by the user but cannot steal the personal information managed by the service provider.

At last, in the proposed scheme, a client certificate includes valid information for only a service provider because a service provider issues a client certificate instead of a global CA. Even if an attacker gets a client certificate, the personal information and services used by the user are not known by the attacker.

9.2 Issuance of the Certificates Using OpenID

In some client certificate issuance service of Class 1, CA verifies the existence of the user by reachability of a mail address. In this paper, this method is referred as a mail address reachability verification method. In the mail address reachability verification method, CA sends a password by E-mail for every application of the client certificate issuance. CA verifies the existence of the user by making the user to input the same password.

Table 3 Comparison of privacy risks among authentication schemes

Schemes	Information Managed by Service Provider	Privacy Risks
Password-Based User Authentication	ID, Password	Conjecturing an owner of ID and Takeover of a user account
Basic SSL Client Authentication	ID, Client Certificate (including personal info.)	Identifying a specific individual or Conjecturing an owner of ID
The Proposed Scheme	ID, Client Certificate (not including personal info.)	----

In contrast, PCA and SP of the proposed method entrust the verification of the existence of the user and the authentication of the user to OP. The assurance level of the certificate depends on the way of the verification of the existence of the user since PCA and SP do not verify directly the existence of the user in the proposed method. In the proposed method, these two methods are the same assurance level because OP verifies the existence of the user and authenticates the user using equality method with the mail address reachability method and PCA and SP trust the authentication result of the user notified by OP.

In Google, the user accesses an individualized URL that is sent by OP in the creation of an account [21]. This OpenID method is the equivalent to the mail address reachability verification method since the URL is known only by the user who receives the E-mail.

The comparison of the mail address reachability verification method and the proposed method using OpenID are shown in Table 4. In the proposed method, the existence of the user is just verified at the creation of the user account since OP manages the user account. In contrast to the comparison method in which the password increases in proportion to number of the client certificate, it is only necessary for the user to manage the one password for OP. Additionally, a number of times of inputting the password can be reduced by a session management of the browser.

In the comparison method, CA verifies directly the existence of the user and authenticates the user and issues the client certificate. In contrast, the proposed method only needs the feature of the issuance of the client certificate since the features of verifying the existence of the user and authenticating the user are entrusted to OP. Additionally, in the proposed method, it is unnecessary for PCA and SP to manage user secrets. Since OP manages the user secrets for the authentication, the security of the user secrets does not depend on each PCA and SP.

9.3 SSL Client Certificate Issuance Method Using TPM

The service provider in the proposed method authenticates the client on a trust that BK is related to AIK and the client installs the SSL client certificate to TPM.

Table 4 Comparison of client certificate issuing method

Evaluation Item		The e-Mail Address Reachability Verification Method	The Proposed Method
N-times Issuing of the Certificate	Number of Verifying the Existence	N times	One time
	Information Used Authentication	N Different Passwords	One Password
	Number of Inputting the Authentication Information	N times	N times at most (The number can be reduced by session management)
	The Secret Managed by the User	N Different Passwords	One Account
The Required Feature for CA	Certificate Issuing Feature	Yes	Yes
	Authentication Feature	Yes	No (It is delegated to OP)
	Verifying the Existence Feature	Yes	No (It is delegated to OP)
	The User Information and the Certificate Management Feature	Yes (Name and e-Mail Address)	Yes (Name and OpenID)
	The Certificate and the Password Management Feature	Yes	No

At the step 5 in the Sect. 6, SP verifies the relation of AIK and BK by verifying the signature of the public key of BK using the AIK certificate. Because AIK is allowed to signing only the non-migratable key, this verification indicates that the secret key of BK is stored in TPM. For this reason, the encrypted SSL client certificate can be decrypted only by TPM that is used to requesting the certificate. Even if the encrypted SSL client certificate is leaked at the step 11 in the Sect. 6, it is impossible that a third person decrypts the certificate. In the step 4 in the Sect. 6, leakage of the secret information is prevented by encrypted communication.

10. Conclusion

In the password authentication system, the user login to the web service using the password that is registered at sign up time. It is desirable to be used a different password in each web services in terms of security. However, the password is reused in most cases since to memorize of two or more passwords bothers the user.

In this paper, we focus on TPM embedded in a computer as a physical device without carrying unlike IC card and USB token, and we proposes the SSL client authentication using the client certificate and the signing key which are secured in TPM. By using OpenID for user authentication, the proposed method reduces the password managed by the user, the cost of managing the password by the server and the risk of leaking the password.

In case of using TPM for the SSL client authentication, the server cannot distinguish the client certificate issued to the TPM-embedded computer from other certificates in a simple method of issuing the client certificate. Then, this paper proposed the method combining two types of TPM keys that are AIK and BK.

At first, this paper proposed the method of issuing the AIK certificate using OpenID. AIK can be used distinguishing TPM, and the existence of the user having AIK is verified by OpenID. Accordingly, the user can be related to TPM. Secondly, this paper proposed the method of issuing the SSL client certificate using BK related to AIK by the TPM command TPM_CertifyKey and using OpenID. According to these two methods, the SSL client certificate can be issued to the specific TPM having AIK. Additionally, the implementation of these methods was shown. At last, this paper showed to be able to authenticate the user using the SSL client certificate that is installed in TPM.

In the client side, the TPM-embedded computer, the major browser and software that link between TPM and the browser are needed. In the server side, a module of SSL client-authentication is needed. Microsoft publishes to support TPM in the hardware certification requirements [22].

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 25330151.

References

[1] “OpenID Authentication 2.0 – Final,” http://openid.net/specs/openid-authentication-2_0.html, accessed May 26, 2015.
[2] Trusted Computing Group, “Trusted Platform Module (TPM) Summary,” http://www.trustedcomputinggroup.org/resources/trusted_platform_module_tpm_summary, accessed May 26, 2015.
[3] Trusted Computing Group, http://www.trustedcomputinggroup.org/, accessed May 26, 2015.
[4] “Trusted Platform Modules Strengthen User and Platform Authenticity,” Trusted Computing Group, Jan. 2005.
[5] “TPM Main Part 3 Commands,” Trusted Computing Group, Specification Version 1.2, Level 2 Revision 116, March 1, 2011.
[6] A. Langley, “Transport Layer Security (TLS) Encrypted Client Certificates draft-agl-tls-encryptedclientcerts-00,” Network Working Group Internet-Draft, April 26, 2012.
[7] Infineon, “TPM Professional Package 4.3,” http://www.infineon.com/cms/en/product/security-ic/trusted-computing/tpm-professional-package-43-br-windows-8-ready!/channel.html?channel=ff80808112ab681d0112ab69225e0121, accessed May 26, 2015.
[8] “JPCERT/CC Activities Overview [July 1, 2014 – September 30, 2014],” Japan Computer Emergency Response Team Coordination Center, http://www.jpcert.or.jp/english/doc/ActivityTopic2014Q2_en.pdf, accessed Nov. 22, 2015.
[9] GlobalSign, “GlobalSign CA Certification Practice Statement – 3.2.3 Authentication of Individual Identity,” Version v7.8, Sept. 2, 2014.
[10] VeriSign, “VeriSign Japan K.K. Certification Practice Statement – 3.2.3 Authentication of Individual Identity,” Version 3.8.7, Oct. 21, 2012.
[11] IAIK TU Graz, “IAIK jTSS – TCG Software Stack for the Java (tm) Platform,” http://trustedjava.sourceforge.net/index.php?item=jtss/readme, accessed May 26, 2015.

[12] Oracle, "Java Software," https://www.oracle.com/java/index.html, accessed May 26, 2015.
[13] Trusted Computing Group, "TCG Software Stack (TSS) Specification Version 1.2 Level 1 Errata A, Part1: Commands and Structures," Trusted Computing Group, March 7, 2007.
[14] IAIK TU Graz, "IAIK/OpenTC TCcert – Trusted Computing certificate tool," http://trustedjava.sourceforge.net/index.php?item=tccert/readme, accessed May 26, 2015.
[15] IAIK TU Graz, "JCA/JCE," https://jce.iaik.tugraz.at/sic/Products/Core-Crypto-Toolkits/JCA-JCE, accessed May 26, 2015.
[16] IAIK TU Graz, "CMS-S/MIME," https://jce.iaik.tugraz.at/sic/Products/Communication-Messaging-Security/CMS-S-MIME, accessed May 26, 2015.
[17] The Apache Sofware Foundation, "Apache HTTP SERVER PROJECT," http://httpd.apache.org/, accessed May 26, 2015.
[18] Oracle, "GlassFish," https://glassfish.java.net/, accessed May 26, 2015.
[19] The PHP Group, "PHP: Hypertext Processor," http://php.net/, accessed May 26, 2015.
[20] R.S. Engelschall, "mod_ssl: The Apache Interface to OpenSSL," http://www.modssl.org/, accessed May 26, 2015.
[21] Google, "Verify your Google Account," https://support.google.com/accounts/answer/63950?hl=en, accessed May 26, 2015.
[22] Microsoft, "Windows Hardware Certification Requirements for Client and Server Systems," http://msdn.microsoft.com/en-us/library/windows/hardware/jj128256.aspx, accessed May 26, 2015.

Shohei Kakei received the B.E. and M.E. degrees from Gifu University in 2011 and 2013, respectively. He is currently doctoral student at Kobe University. His current research interests include digital forensics and security applications.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa junior college, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, the University of Tokushima, Japan. From 2003 to 2008 she was a lecturer of the same department. Since 2008, she has been an associate professor at the Information and Multimedia Center, Gifu University, Japan. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kinki University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.

Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering, Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronic Engineering, Faculty of Engineering, Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.