An Efficient Direct Anonymous Attestation Scheme with Forward Security

DENG-GUO FENG, JING XU, XIAO-FENG CHEN
State Key Laboratory of Information Security
Institute of Software, Chinese Academy of Sciences
4# South Fourth Street, Zhong Guan Cun, Beijing 100190
CHINA
feng@is.iscas.ac.cn, xujing@is.iscas.ac.cn, chenxiaof@is.iscas.ac.cn

Abstract: - Direct Anonymous Attestation (DAA) is a cryptographic mechanism adopted by the Trusted Computing Group in its specifications for trusted computing platforms (TCP). In this paper, we propose a new DAA scheme and prove it is secure under the strong RSA assumption and the decisional Diffie-Hellman assumption. While satisfying all the security properties proposed in previous DAA schemes, our scheme provides a new desired security property, forward security: compromise of the current private key of TPM does not enable an adversary to forge signatures pertaining to the past. Such forward security is important to mitigate the damage caused by private key exposure.

Key-Words: - Trusted computing platform, Direct anonymous attestation, Forward security, Trusted platform module

1 Introduction

Trusted computing platforms have been proposed as a promising approach to enhance the security [1-4] of general-purpose computing systems. A trusted computing platform is a computing device integrated with a cryptographic chip called a trusted platform module (TPM), which is designed and manufactured in a way such that all parties can trust cryptographic computing results from this TPM. When a TPM is manufactured, a unique asymmetric keypair, called the Endorsement Key (EK), is created and stored in the protected area of the TPM. If the authentication of a TPM is directly based on its EK, all transactions by the same TPM can be linked and the TCP may suffer a loss of privacy. In order to address privacy concerns resulting from routine use of an EK, two approaches have been proposed in the TPM specifications.

The first approach in the TPM Specification Version 1.1 is based on a trusted third party, referred to as a Privacy Certification Authority (Privacy CA). when a TPM needs to authenticate itself to a verifier, it generates a RSA key pair called an Attestation Identity Key (AIK), sends the AIK public key to the Privacy CA, and authenticates this public key using a valid EK. The Privacy CA will check whether it finds the EK in its list and, if so, issue a certificate on the TPM's AIK. The TPM can then forward this certificate to the verifier and authenticate itself. In this way, the TPM hides its identity during the transaction. This approach has the obvious drawback that the Privacy CA needs to be involved in every transaction. Moreover, the compromise of the Privacy CA (or a dishonest Privacy CA) can destroy all privacy guarantees.

The second approach, Direct Anonymous Attestation (DAA), was introduced to counteract this drawback. In the DAA scheme, there exists an issuer who creates a group public key. Later on, the issuer issues a DAA credential to each TPM. To authenticate as a group member, the TPM generates a signature using his credential such that the signature can be verified by a verifier using the group public key. The first DAA scheme was proposed by Brickell *et al*. [6] and later was standardized in TPM Specification Version 1.2 [5]. However, their scheme is too inefficient to be suitable for embedded devices with limited computing capabilities. In 2007, Ge *et al*. [7] presented a new construction with more efficient sign and verify protocols, which is more attractive for embedded devices. Both scheme are provably secure in the random oracle model under the strong RSA and the Decisional Diffie-Hellman (DDH) assumptions. Recently, Chen *et al*. [8] also proposed an efficient DAA scheme based on the Strong Diffie-Hellman (SDH) and DDH assumptions. Their scheme is more efficient in signature length and in computational complexity.

In fact, DAA can be seen as a group signature[9-11] without the capability to open signatures, i.e.,

the anonymity is not revocable. Moreover, DAA allows for detection of "known" keys: if the DAA secret keys are extracted from a TPM and published, a verifier can detect that a signature was produced using these secret keys. In prior DAA schemes, if a private key of TPM is exposed to an attacker, all previously obtained signatures will be invalid, because one cannot distinguish whether a signature is generated by an attacker after it obtained the private key or by the legitimate signer before the attacker obtained the private key.

In this paper, we propose to use the concept of forward security to reduce the damage of exposure of TPM's private key, i.e. even when a private key is exposed, previously generated signatures remain valid and do not need to be re-signed. In particular our construction is built up from the group signature scheme due to Ateniese *et al*. [11]. The concept of forward secure signatures was first proposed by Anderson [12] for traditional signatures, and the first forward secure group signature scheme was presented by Song [13] in 2001. Similarly, our forward secure DAA scheme shares many properties with the forward secure group signature schemes. However, unlike those schemes, the issuer can not open the signer's identity and thus the signature should also be forward secure for the issuer. In a forward secure DAA scheme, the private key of TPM evolve over time: in time period i, TPM's private key evolves from f_{i-1} to f_i using a public one-way function, and then f_{i-1} is erased from the system.

The remainder of this paper is organized as follows. The next section reviews some cryptographic assumptions and building blocks of our proposed scheme. Section 3 introduces the model of our construction. We then propose a forward secure DAA scheme in Section 4, whose security and performance are analyzed in Section 5. Section 6 concludes.

2 Preliminaries

This section reviews some cryptographic assumptions and introduces the building blocks necessary in the subsequent design of our DAA scheme.

2.1 Cryptographic Assumptions

The security of our DAA scheme relies on the Strong-RSA assumption and the Decisional Diffie-Hellman (DDH) assumption.

Definition1 (Strong RSA Assumption) The Strong RSA Assumption states that it is computational infeasible, on input a random RSA modulus n and a random element $u \in \mathbb{Z}_n^*$, to compute values $e > 1$ and v such that $v^e \equiv u \pmod{n}$.

Definition2 (DDH Assumption) There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distribution D and R, where $D = \left(g, g^x, g^y, g^z\right)$ and $R = \left(g, g^x, g^y, g^{xy}\right)$ with $x, y, z \in_R \mathbb{Z}_u$.

2.2 Signature of Knowledge

In our scheme we will use the signature of knowledge that allows a prover to demonstrate the knowledge of a secret w.r.t. some public information such that no other information is revealed in the process. To describe these protocols, we use notation introduced by Camenisch *et al*. [14] for various proofs of knowledge. For example, $PK\left\{(a,b): y_1 = g_1{}^a h_1{}^b \wedge y_2 = g_2{}^a h_2{}^b\right\}$ denotes a proof of knowledge of integers a and b such that $y_1 = g_1{}^a h_1{}^b$ and $y_2 = g_2{}^a h_2{}^b$ holds, where $y_1, g_1, h_1, y_2, g_2, h_2$ are elements of some groups $G_1 = \langle g_1 \rangle = \langle h_1 \rangle$ and $G_2 = \langle g_2 \rangle = \langle h_2 \rangle$. Such proof of knowledge protocols can be turned into signature of knowledge using the Fiat-Shamir heuristic [15, 16]. We use the notation $SPK\left\{(a): y = z^a\right\}(m)$ to denote a signature on a message m obtained in this way.

Definition 3 An $(l+1)$ tuple $(c, s_1, \cdots, s_l) \in \{0,1\}^k \times \mathbb{Z}_n^{*l}$ satisfying the equation $c = H(m \| y \| g \| e \| t_1 \| \ldots \| t_l)$

with $t_i = \begin{cases} g^{(s_i^e)} & if \quad c[i] = 0 \\ y^{(s_i^e)} & otherwise \end{cases}$

is a signature of the knowledge of an e-th root of the discrete logarithm of y to the base g, and is denoted $SKROOTLOG\{\alpha : y = g^{\alpha^e}\}(m)$.

Such a signature can be computed if the e-th root α of the discrete logarithm of y to the base g is known. One first computes the values

$$t_i^* = g^{r_i^e}$$

for $i = 1,2,...l$ with randomly chosen $r_i \in \mathbb{Z}_n^*$. Then, c is set to $H(m \| y \| g \| e \| t_1^* \| ... \| t_l^*)$, and finally, $s_i = \begin{cases} r_i & if \quad c[i] = 0 \\ r_i / x (\mathrm{mod}\, n) & otherwise \end{cases}$ for $i = 1,2,...l$. It can easily be seen that the resulting tuple $(c, s_1, ..., s_l)$ satisfies the verification equation.

Definition 4 A signature of the knowledge of representations of $y_1, y_2, \cdots, y_w$ with respect to the bases $g_1, g_2, \cdots, g_y$ on the message m is denoted as follows

$$SKREP\left\{(\alpha_1, \cdots, \alpha_u) : \left(y_1 = \prod_{j=1}^{l_1} g_{b_{1j}}{}^{\alpha_{e_{1j}}} \right) \wedge \cdots \wedge \left(y_w = \prod_{j=1}^{l_w} g_{b_{wj}}{}^{\alpha_{e_{wj}}} \right) \right\}(m),$$

where the indices $e_{ij} \in \{1, \cdots, u\}$ refer to the elements $\alpha_1, \cdots, \alpha_u$ and the indices $b_{ij} \in \{1, \cdots, v\}$ refer to the base elements $g_1, \cdots, g_v$. The signature consists of an $(u+1)$ tuple $(c, s_1, \cdots, s_u) \in \{0,1\}^k \times \mathbb{Z}_n{}^u$ satisfying the equation

$$c = H\left(m \| y_1 \| \cdots \| y_w \| g_1 \| \cdots \| g_v \| \left\{ \left\{ e_{ij}, b_{ij} \right\}_{j=1}^{l_i} \right\}_{i=1}^{w} \| y_1{}^c \prod_{j=1}^{l_1} g_{b_{1j}}{}^{s_{e_{1j}}} \| \cdots \| y_w{}^c \prod_{j=1}^{l_w} g_{b_{wj}}{}^{s_{e_{wj}}} \right).$$

SKREP can be computed if a u-tuple $(\alpha_1, \cdots, \alpha_u)$ is known which satisfies the given equations. One first chooses $r_i \in_R \mathbb{Z}_n$ for $i = 1, \cdots, u$, computes c as

$$c = H\left(m \| y_1 \| \cdots \| y_w \| g_1 \| \cdots \| g_v \| \left\{ \left\{ e_{ij}, b_{ij} \right\}_{j=1}^{l_i} \right\}_{i=1}^{w} \| \prod_{j=1}^{l_1} g_{b_{1j}}{}^{r_{e_{1j}}} \| \cdots \| y_w{}^c \prod_{j=1}^{l_w} g_{b_{wj}}{}^{r_{e_{wj}}} \right),$$

and then sets $s_i = r_i - c\alpha_i \left(\mathrm{mod}\, n\right)$ for $i = 1, \cdots, u$.

Definition 5 An efficient signature of the knowledge of the e-th root of the g-part of a representation of y to the bases h and g, denoted $E - SKROOTREP\left\{(\alpha, \beta) : y = h^\alpha g^{\beta^e}\right\}(m)$, consists of an $(e-1)$-tuple $(y_1, \cdots, y_{e-1}) \in G^{e-1}$ and of a signature of knowledge

$$U = SKREP\left\{(\gamma_1, \cdots, \gamma_e, \delta) : y_1 = h^{\gamma_1} g^\delta \wedge y_2 = h^{\gamma_2} y_1{}^\delta \cdots \wedge y_{e-1} = h^{\gamma_{e-1}} y_{e-2}{}^\delta \wedge y = h^{\gamma_e} y_{e-2}{}^\delta\right\}(m)$$

The signature of knowledge can be verified by checking the correctness of U.

2.3 Commitment Scheme

In [17], Pederson proposed a computationally binding and unconditionally hiding scheme based on the discrete logarithm problem. The commitment scheme is following: Given are a group G of prime order q and two random generator g and h such that $\log_g h$ is unknown and computing discrete logarithms is infeasible. A value $a \in Z_q$ is committed to as $c_a := g^a h^r$, where r is randomly chosen from Z_q.

3 The Model of DAA Scheme

A forward secure DAA scheme has four types of participants and six procedures. The participants are the certificate issuer, a trusted platform module (TPM), a host that has TPM "built in", and a verifier. The six procedures are described in the following:

- **Setup:** This probabilistic polynomial time (PPT) algorithm takes as input a security parameter and outputs system-wide public parameters and secret keys for group membership certificate generation.
- **Join:** This interactive protocol between a TPM and the issuer allows TPM to obtain a group membership certificate and become a group member. The public certificate and the TPM's identity information are stored by the issuer in a database for future use.
- **Evolve:** This deterministic polynomial time (DPT) algorithm takes as input TPM's private key for time period i and outputs the corresponding private key for time period $i+1$.
- **Sign:** This PPT algorithm takes as input group membership certificate, TPM's private key and a message and returns an anonymous group signature of the message.
- **Verify:** This DPT algorithm takes as input the group public keys, a message and its candidate signature and returns either accept or reject. A signature is verified to make sure it originates from a legitimate TPM without knowledge of which particular one.

- **Rogue tagging:** This DPT algorithm can identify and exclude a rogue TPM for the group.

Conventionally, a forward secure DAA scheme must satisfy the following requirements:

- **Correctness:** Any valid signature can be correctly verified by the Verify protocol.
- **Unforgeability:** A valid group membership certificate can only be created by a TPM and the issuer through the Join protocol. Without the knowledge of a group membership certificate and the corresponding private key, it is computationally impossible to produce a signature that can be accepted by the VERIFY algorithm.
- **Anonymity:** It is infeasible to identify the real TPM of a signature unless this TPM is on the revocation list.
- **Unlinkability:** It is infeasible to link two different signatures of the same TPM.
- **Forward Security:** Given the TPM's private key for the current period, the signature produced in previous time periods is still anonymous and unlinkable.

4 Forward Secure DAA Scheme

In this section, we describe our method for implementing direct anonymous attestation. As mentioned earlier, our construction is based on Ateniese et al.'s group signature scheme [11]. However, the sign and verify protocols are re-designed.

4.1 Setup

Let $\varepsilon > 1$, k, and l_p be security parameters and let $\lambda_1, \lambda_2, r_1, r_2$ denote lengths satisfying $\lambda_1 > \varepsilon(\lambda_2 + k) + 2$, $\lambda_2 > 4l_p$, $r_1 > \varepsilon(r_2 + k) + 2$, and $r_2 > \lambda_1 + 2$. Let H be a collision-resistant hash function $H: \{0,1\}^* \to \{0,1\}^k$.

The certificate issuer generates the group public key and his secret key as follows:

1) Choose random secret l_p-bit primes p', q', such that $p = 2p' + 1$ and $q = 2q' + 1$ are prime. Set the modulus $n = pq$.
2) Choose random elements $a, b, g, h \in QR(n)$ (of order $p'q'$).Compute $\phi(n) = (p-1)(q-1)$. Choose a random integer s such that $\gcd(s, \phi(n)) = 1$ (e.g. $s = 3$). Divide the time during which the group public key is valid into T time period and makes the time intervals public.
3) The group public key is $Y = (n, a, b, g, h, s)$ and the corresponding secret key is $S = (p', q')$.

In DAA scheme, it is infeasible to reveal real identity of signer even if for the issuer. Thus our DAA scheme does not need the revocation public key and corresponding secret key that are required in all group signature schemes.

4.2 Join Protocol

The join protocol is a protocol runs between the issuer and a TPM. We assume that the TPM and the issuer have established an private and authentic channel. The join protocol takes the following steps:

1) TPM chooses a random private key $f_0 \in (0, 2^{\lambda_1})$ and keeps it secret.
2) TPM chooses a random secret $t' \in (0, 2^{\lambda_2})$, computes $C = a^{f_0 s^T} b^{t'} \bmod n$, and sends C to the issuer. Additionally, TPM proves to the issuer knowledge of f_0 and t', it executes as prover the protocol $U = E - SKROOTREP\{(f_0, t') : C = a^{f_0 s^T} b^{t'} \bmod n\}(' \ ')$ with the issuer as the verifier.
3) The issuer checks that $C \in QR(n)$. If this is the case and U is correct, the issuer selects a random $t'' \in (0, 2^{\lambda_2})$ and a random prime $e \in (2^{r_1} - 2^{r_2}, 2^{r_1} + 2^{r_2})$ and computes $A = (Cb^{t''} d)^{1/e} \bmod n$. Finally, the issuer sends (A, e, t'') to the host.
4) The host stores (A, e) and forwards (A, e, t'') to the TPM. The TPM computes $t = t' + t''$, verifies that
$$A^e = a^{f_0 s^T} b^t d \bmod n$$
and stores t.

As a result of the protocol, the TPM will have obtained secret values f_0 and t, and the host will have values A and e. Moreover, the TPM is

allowed to re-run the Join protocol with the same secret f_0 value many times.

4.3 **Evolve**

The TPM can evolve his private key using a public one-way function $f(x) = x^s \bmod n$. Assume the TPM has membership private key (A, e, t, f_i) at time period i. Then at time period $i+1$, his private key becomes (A, e, t, f_{i+1}), where $f_{i+1} = f_i^s \bmod n$, and the verification equation is

$$A^e = a^{f_i^{s^{T-i}}} b^t d \bmod n.$$

4.4 **Sign**

Let m be the message to be signed. The TPM has secret key (f_i, t) and a credential (A, e), whereas the host only knows the credential (A, e). The signing algorithm takes the following steps:

1) The host picks random integer $w \in_R \{0,1\}^{2\ell_p}$ and computes $T_1 = Ag^w \bmod n$, $T_2 = g^e h^w \bmod n$. The TPM computes $T_3 = a^{f_i^{s^{T-i}}} \bmod n$ and sends T_3 to the host.

2) The TPM and host together produce a signature of knowledge that T_1 and T_2 are commitments to the credential and T_3 is computed using the TPM's private key f_i. That is, they compute the signature of knowledge

$$SPK\{(e, f_i, t, w) : T_1^e g^{-ew} = T_3 b^t \bmod n \wedge$$
$$T_2 = g^e h^w \bmod n \wedge T_2^{-e} g^{ee} h^{ew} = 1 \bmod n \wedge$$
$$T_3 = a^{f_i^{s^{T-i}}} \bmod n \wedge e \in (2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2})\}(m)$$

Actually, it equals two signatures of knowledge:

$$V_1 = SPK\{(e, w, t) : T_1^e g^{-ew} = T_3 b^t \bmod n \wedge$$
$$T_2 = g^e h^w \bmod n \wedge T_2^{-e} g^{ee} h^{ew} = 1 \bmod n \wedge$$
$$e \in (2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2})\}(m)$$

and

$$V_2 = SKROOTLOG\{f_i : T_3 = a^{f_i^{s^{T-i}}} \bmod n\}(m)$$

Most of the secrets involved are actually known by the host; in fact only the values involving f_i and t need to be computed by the TPM, as the reader can see below.

a) The TPM picks random integer $r_t \in \{0,1\}^{\varepsilon(\lambda_2 + k)}$ and computes $\tilde{R}_1 = b^{-r_t} \bmod n$. The TPM sends $\tilde{R}_1$ to the host.

b) The host picks random integers $r_e \in \{0,1\}^{\varepsilon(\gamma_2 + k)}$, $r_w \in \{0,1\}^{\varepsilon(2\ell_p + k)}$, $r_{\delta_1} \in \{0,1\}^{\varepsilon(2\gamma_1 + k + 1)}$, $r_{\delta_2} \in \{0,1\}^{\varepsilon(\gamma_1 + 2\ell_p + k + 1)}$ and computes $R_1 = \tilde{R}_1 T_1^{r_e} g^{-r_{\delta_2}} \bmod n$, $R_2 = g^{r_e} h^{r_w} \bmod n$, $R_3 = T_2^{-r_e} g^{r_{\delta_1}} h^{r_{\delta_2}} \bmod n$.

c) The host also computes $c_h = H(g \| h \| n \| s \| T \| a \| b \| d \| T_1 \| T_2 \| T_3 \| R_1 \| R_2 \| R_3)$ and sends c_h to the TPM.

d) The TPM chooses a random $n_t \in \{0,1\}^k$, computes $c = H(H(c_h \| n_t) \| m)$ and $s_t = r_t + ct$, and sends c, n_t, s_t to the host.

e) The host computes $s_e = r_e + c(e - 2^{\gamma_1})$, $s_w = r_w + cw$, $s_{\delta_1} = r_{\delta_1} + c \cdot e \cdot e$, $s_{\delta_2} = r_{\delta_2} + c \cdot e \cdot w$. The signature $V_1 = (T_1, T_2, T_3, c, n_t, s_e, s_t, s_w, s_{\delta_1}, s_{\delta_2})$

f) The TPM computes the values

$$t_j^* = a^{x_j^{s^{T-i}}}$$

for $j = 1, \cdots, l$ with randomly chosen $x_j \in \mathbb{Z}_n^*$ and security parameter $l \le k$. Then, c' is set to $H\left(m \| T_3 \| a \| s^{T-i} \| t_1^* \| \cdots \| t_l^*\right)$, and finally,

$$w_j = \begin{cases} x_j & \text{if } c'[j] = 0 \\ x_j / f_i \pmod n & \text{otherwise} \end{cases}$$

for $j = 1, \cdots, l$. Then $V_2 = (c', w_1, \cdots, w_l)$ and the host outputs the signature $\sigma = (V_1, V_2)$.

4.5 **Verify**

A signature $\sigma = (T_1, T_2, T_3, c, n_t, s_e, s_t, s_w, s_{\delta_1}, s_{\delta_2}, c', w_1, \cdots, w_l)$ on a message m w.r.t the public key (n, a, b, d, g, h, s) is as verified as follows.

1) Compute

$$R_1' = T_1^{s_e + c2^{\gamma_1}} g^{-s_{\delta_2}} b^{-s_t} (T_3)^{-c} \bmod n,$$

$$R_2^{'} = T_2^{-c} g^{s_e + c2^{\eta}} h^{s_w} \bmod n,$$

$$R_3^{'} = T_2^{-(s_e + c2^{\eta})} g^{s_{\delta_1}} h^{s_{\delta_2}} \bmod n.$$

2) Verify that

$$c \stackrel{?}{=} H(H(H(g \| h \| n \| s \| T \| a \| b \| d \| T_1 \| T_2 \| T_3 \| R_1^{'} \| R_2^{'} \| R_3^{'}) \| n_t) \| m),$$

$$s_e \stackrel{?}{\in} \{0,1\}^{\varepsilon(\gamma_2+k)+1},\ s_t \stackrel{?}{\in} \{0,1\}^{\varepsilon(\lambda_2+k)+1},$$

$$s_w \stackrel{?}{\in} \{0,1\}^{\varepsilon(2\ell_p+k)+1},\ s_{\delta_1} \stackrel{?}{\in} \{0,1\}^{\varepsilon(2\gamma_1+k+1)+1},$$

$$s_{\delta_2} \stackrel{?}{\in} \{0,1\}^{\varepsilon(\gamma_1+2\ell_p+k+1)+1}.$$

3) Compute

$$t_i = \begin{cases} a^{(w_i^e)} & if \quad c'[i] = 0 \\ T_3^{(w_i^e)} & otherwise \end{cases}$$

and verify $c' \stackrel{?}{=} H\left(m \| T_3 \| a \| s^{T-i} \| t_1 \| \cdots \| t_l\right)$

4) If all the above verifications succeed, the verifier outputs succeed, otherwise outputs fail.

4.6 Sign with Variable Anonymity

Let variable anonymity [12] is a conditionally linkable anonymous authentication, in which the signatures signed by the same TPM in a certain time interval are linkable. To achieve variable anonymity, each signature will belong to a "linkability class" that is identified using a Solely Signature Identifier (SSID). All signatures made by the same TPM with the same SSID are linkable. If complete anonymity is desired, the signer can simply pick a random SSID.

To implement variable anonymity, we add the following computations to the Sign protocol:

$$\eta = H_1(SSID),\ T_4 = \eta^{f_i} \bmod n,$$

$$V_2 = SPK\{f_i : T_3 = a^{f_i^{s^{T-i}}} \bmod n \wedge T_4 = \eta^{f_i} \bmod n\}(m)$$

where $H_1 : \{0,1\}^* \to Z_n^*$, and outputs the signature $\sigma = (\eta, V_1, V_2)$. The computation of V_2 can be found in the Appendix A.

4.7 Rogue TPM Tagging

If TPM's private key f_i at time period i is compromised, a verifier should be able to identify the attestation request from rogue TPMs. To do so, the secret of a corrupted TPM should be published on the revocation list. For each private key f_i on the revocation list, a verifier checks $T_4 \stackrel{?}{=} \eta^{f_i} \bmod n$. If the equation holds, the request comes from a revoked TPM.

5 Analysis of Proposed DAA Scheme

In this section, we analyze the security and performance of our proposed DAA scheme.

5.1 Security Analysis

Lemma 6 Let n be a special RSA number, $u, g \in QR(n)$ and g be a generator of $QR(n)$. Under the strong RSA assumption, if $g^x \equiv u^y (\bmod n)$ for given $x, y \in Z_n$, then $\gcd(x, y) = y$。

Proof: Let $r = \gcd(x, y)$, by the extended Euclidean algorithm, there exist α, β s.t. $\alpha x + \beta y = r$. Hence,

$$g \equiv g^{(\alpha x + \beta y)/r} \equiv (u^{\alpha} g^{\beta})^{y/r} (\bmod n)$$

Note that we cannot have $y > r$ because otherwise $u^{\alpha} g^{\beta}$ is a y/r-th root of g, which contradicts the strong RSA assumption. Thus, we have $y = r$, i.e. $\gcd(x, y) = y$. ■

Theorem 7 Under the strong RSA assumption, the interactive protocol underlying the Sign and Verify protocol is a statistical zero-knowledge (honest-verifier) proof of knowledge of a membership credential (A, e) and the corresponding secret key (f_i, t).

Proof: The proof that the interactive protocol is statistical zero-knowledge is quite standard. We restrict our attention the proof of knowledge part.

Now we show that the knowledge extractor is able to recover the membership credential and the corresponding secret key once it has found two accepting tuples. Let $(V_1^{'} = (T_1, T_2, T_3, R_1, R_2, R_3, n_t, c', s_e^{'}, s_t^{'}, s_w^{'}, s_{\delta_1}^{'}, s_{\delta_2}^{'}), V_2^{'})$ and $(V_1'' = (T_1, T_2, T_3, R_1, R_2, R_3, n_t, c'', s_e'', s_t'', s_w'', s_{\delta_1}'', s_{\delta_2}''), V_2'')$ be two accepting tuples. Then we have

$$R_1 = T_1^{s_e^{'} + c_1 2^{\eta}} g^{-s_{\delta_2}^{'}} b^{-s_t^{'}} T_3^{-c'}$$

$$= T_1^{s_e'' + c'' 2^{\eta}} g^{-s_{\delta_2}''} b^{-s_t''} T_3^{-c''} \bmod n \qquad (1)$$

$$R_2 = T_2^{-c'} g^{s_e^{'} + c' 2^{\eta}} h^{s_w^{'}}$$

$$= T_2^{-c''} g^{s_e''+c''2^{\eta}} h^{s_w''} \bmod n \quad (2)$$

$$R_3 = T_2^{-s_e'-c'2^{\eta}} g^{s_{\delta_1}'} h^{s_{\delta_2}'}$$

$$= T_2^{-s_e''-c''2^{\eta}} g^{s_{\delta_1}''} h^{s_{\delta_2}''} \bmod n \quad (3)$$

It follows that

$$T_2^{c'-c''} = g^{(s_e'-s_e'')+(c'-c'')2^{\eta}} h^{s_w'-s_w''} \bmod n \quad (4)$$

$$T_1^{(s_e'-s_e'')+(c'-c'')2^{\eta}} = g^{s_{\delta_1}'-s_{\delta_1}''} h^{s_{\delta_2}'-s_{\delta_2}''} \bmod n \quad (5)$$

Let $\Delta c = c' - c''$, $\Delta e = (s_e' - s_e'') + (c' - c'')2^{\eta}$, $\Delta\delta_1 = s_{\delta_1}' - s_{\delta_1}''$, $\Delta\delta_2 = s_{\delta_2}' - s_{\delta_2}''$, $\Delta t = s_t' - s_t''$, and $\Delta w = s_w' - s_w''$. Due to Lemma 6, we find $\gcd(\Delta c, \Delta e) = \Delta c$, $\gcd(\Delta c, \Delta w) = \Delta c$ and $\gcd(\Delta c, \Delta t) = \Delta c$. Then from the equation (3) we have $T_2^{\Delta e} = g^{\Delta\delta_1} h^{\Delta\delta_2} \bmod n$. Since the equation (4) holds, we further obtain $g^{\Delta e\Delta e} h^{\Delta w\Delta e} = g^{\Delta\delta_1\Delta c} h^{\Delta\delta_2\Delta c}$, and then $\Delta\delta_2 = \dfrac{\Delta w\Delta e}{\Delta c}$.

From the equation (1), we have $T_1^{\Delta e} = g^{\frac{\Delta w\Delta e}{\Delta c}} b^{\Delta t} (T_3)^{\Delta c}$, and further obtain $(\dfrac{T_1}{g^{\frac{\Delta w}{\Delta c}}})^{\frac{\Delta e}{\Delta c}} b^{-\frac{\Delta t}{\Delta c}} = T_3$. Finally, we recover $(A, e, t) = (\dfrac{T_1}{g^{\frac{\Delta w}{\Delta c}}}, \dfrac{\Delta e}{\Delta c}, \dfrac{\Delta t}{\Delta c})$. Likewise, we also can recover f_i such that $T_3 = a^{f_i s^{T-i}}$ according to [14], which completes the proof. ■

Theorem 8 Under the decisional Diffie-Hellman assumption over subset of QR_n, it is infeasible to link the transactions by a TPM with different SSID.

Proof. To decide whether two transactions are linked to a TPM, one needs to decide whether two signatures $\sigma = (\eta, V_1, V_2)$ and $\sigma' = (\eta', V_1', V_2')$ are produced from the same TPM.

Since the interactive protocol is statistically zero knowledge, no information is statistically revealed by (V_1', V_2') in the random oracle. For the different SSID, we have $\eta \neq \eta'$. Therefore under the DDH assumption, it is infeasible to decide whether or not there exists an f_i such that $T_4 = \eta^{f_i} \bmod n$ and $T_4' = \eta'^{f_i} \bmod n$. ■

Corollary 9 The DAA scheme presented in Section 4 is secure under the strong RSA assumption and the decisional Diffie-Hellman assumption in the random oracle model.

Proof. We have to show that our scheme satisfies all the security properties listed in Section 3.

Correctness: By inspection.

Unforgeability: It can be easily proven from the Theorem 7.

Unlinkability: It can be easily proven from the Theorem 8.

Anonymity: If the SSID is chosen randomly, it is infeasible to identify the actual signer for a valid signature $\sigma = (\eta, V_1, V_2)$. The reason is that the underlying interactive protocol is statistically zero-knowledge, and no information is statistically revealed in the random oracle model.

Forward Security: The key evolve function is $f(x) = x^s \bmod n$. As only the issuer knows p, q such that $n = pq$, even if the TPM is compromised and the private key f_i in time period i is known by an attacker, he still cannot compute the past private key $f_j\,(j < i)$. Therefore, the signature produced in previous time periods is still anonymous and unlinkable. Moreover, the key evolve procedure is finished by the TPM and no interactive protocol transmitted in the network is needed. In other words, our key-evolve features self-healing. ■

5.2 Performance Analysis

Suppose we set the modulus n be about 2048 bits. We further choose the security parameters $k = 160$, $\varepsilon = 4/3$, $s = 2$, $T = 3$, $\lambda_1 = \lambda_2 = 1024$. For the signature with total anonymity, the total bit-length is 44128. To generate a DAA signature, the host needs to perform 10 modular exponentiations and the TPM 16 modular exponentiations. To verify a signature, 26 modular exponentiations is needed.

In addition, the DAA scheme without forward security (see Appendix B) does not need the signature of knowledge of the e-th root in the Sign phase and Verify phase. Compared to the DAA scheme with forward security, the scheme (Appendix B) requires less computation cost and the signature length improves about 78.5%.

6 Conclusion

In this paper, we present the first forward secure DAA scheme. We believe that forward security provides really useful features of direct anonymous attestation mechanism. Our scheme satisfies forward security as well as all the traditional security properties shared with previous DAA schemes. However, the efficiency of our scheme should be improved and it is left as our future work.

Acknowledgements: The research was supported by the National Grand Fundamental Research (973) Program of China under Grant No. 2007CB311202 and the National Natural Science Foundation of China (NSFC) under Grants No.60673083 and No.60873197.

References:

[1] J. HAJNÝ, T. PELKA, V. ZEMAN, Flexible authentication framework with bound authentication and authorization, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol.8, No.1, 2009, pp. 143-52.

[2] C.I. HSU, The Benefits of PKI Application and Competitive Advantage, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol.7, No.7, 2008, pp. 776-785.

[3] N.C.Wang, J.S. Chen, Y.F. Huang, and T.W. Chan, An IPSec-Based Key Management Algorithm forMobile IP Networks, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol.7, No.8, 2008, pp. 892-901.

[4] N. Doukas, N.V. Karadimas, A blind source separation based cryptography scheme for mobile military communication applications, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol.7, No.12, 2008, pp. 1235-1245.

[5] Trusted Computing Group. TCG TPM specification 1.2 (2003), http://www.trustedcomputinggroup.org

[6] E. Brickell, J. Camenisch, L. Chen, Direct anonymous attestation, *In: Proceedings of the 11th ACM Conference on Computer and Communications Security*, ACM Press, New York, 2004, pp. 132-145.

[7] H. Ge, S. R. Tate, A Direct Anonymous Attestation Scheme for Embedded Devices, *In: Public Key Cryptography*, LNCS 4450, 2007, pp.16-30.

[8] X. F. Chen, D. G. Feng, Direct Anonymous Attestation for Next Generation TPM, *Journal of Computer*, Vol. 3, No.12, 2008, pp.43-50.

[9] D. Chaum and E.V. Heyst, Group signatures, *In: EUROCRYPT'91*, LNCS 547, Springer-Verlag, 1991, pp. 257-265.

[10] D. Boneh, X. Boyen, and H. Shacham, Short group signature, *In proceedings of Crypto '04*, LNCS 3152, 2004, pp. 41-55.

[11] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, A Practical and Provably Secure Coalition-Resistant Group Signature Scheme, *In: Proceedings of CRYPTO 2000*, pp. 255-270.

[12] R. Anderson, Invited Lecture, *4th ACM Computer and Communications Security*, 1997.

[13] D. Song, Practical forward secure group signature schemes, *In: Proceedings of the 8th ACM conference on Computer and Communications Security*, Pennsylvania, USA, 2001, pp. 225-234.

[14] J. Camenisch, M. Stadler, Efficient group signature schemes for large groups, *In: CAIP 1997*. LNCS 1296, Springer, Heidelberg, 1997, pp. 410-424.

[15] A.Fiat, A.Shamir, How to prove yourself: Practical solutions to identification and signature problems, *In: Odlyzko, A.M. (ed.) CRYPTO 1986*, LNCS 263, Springer, Heidelberg 1987, pp. 186-194.

[16] M.Chase, A.Lysyanskaya, On signatures of knowledge, *In: Crypto 2006*, LNCS 4117, Springer, Heidelber 2006, pp.78-96.

[17] T. Pedersen, Non-interactive and information theoretic secure verifiable secret sharing, *In: CRYPTO 1991*, LNCS 576, Springer Verlag, 1992, pp. 129-140.

[18] TCG. TPM V1.2 Specification Changes: A summary of changes with respect to the v1.1b TPM specification, 2003.

Appendix A: The Signature of Knowledge

$$V_2 = SPK\{f_i : T_3 = a^{f_i^{s^{T-i}}} \bmod n \wedge T_4 = \eta^{f_i} \bmod n\}(m)$$

Such a signature can be computed if f_i is known. One first computes the values

$$t_j^* = a^{r_j^{s^{T-i}}}, \quad \overline{t}_j^* = \eta^{r_j}$$

for $j = 1, \cdots, l$ with randomly chosen $r_j \in \mathbb{Z}_n^*$ and security parameter $l \le k$. Then, c is set to $H(m \| T_3 \| T_4 \| \eta \| a \| s^{T-i} \| t_1^* \| \cdots \| t_l^* \| \overline{t}_1^* \| \cdots \| \overline{t}_l^*)$, and finally,

$$s_j = \begin{cases} r_j & if \quad c[j] = 0 \\ r_j / f_i (\bmod n) & otherwise \end{cases}$$

for $j = 1, \cdots, l$. The signature is $(c, s_1, \cdots, s_l)$.

Given the signature $(c,s_1,\cdots,s_l)$, the verification procedure is as follows: One first computes

$$t_j=\begin{cases}a^{s_j s^{T-i}} & if\ \ c[j]=0\\ T_3^{s_j s^{T-i}} & otherwise\end{cases}$$

$$\overline{t}_j=\begin{cases}\eta^{s_j} & if\ \ c[j]=0\\ T_4^{s_j} & otherwise\end{cases}$$

and verify $c=^{?} H(m\|T_3\|T_4\|\eta\|a\|s^{T-i}\|t_1\|\cdots\|t_l\|\overline{t}_1\|\cdots\|\overline{t}_l)$.

Appendix B: Direct Anonymous Attestation Scheme without Forward Security

B.1 Setup

Let $\varepsilon>1$, $k,l_p,\lambda_1,\lambda_2,r_1,r_2$ be security parameters and let H be a collision-resistant hash function $H:\{0,1\}^*\to\{0,1\}^k$ .The certificate issuer generates the group public key and his secret key as follows:

(1) Choose random secret l_p -bit primes p', q', such that $p=2p'+1$ and $q=2q'+1$ are prime. Set the modulus $n=pq$.

(2) Choose random elements $a,b,g,h\in QR(n)$ (of order $p'q'$).

(3) The group public key is $Y=(n,a,b,g,h)$ and the corresponding secret key is $S=(p',q')$.

B.2 Setup

(1) TPM chooses a random private key $f\in(0,2^{\lambda_1})$ and keeps it secret.

(2) TPM chooses a random secret $t'\in(0,2^{\lambda_2})$, computes $C=a^f b^{t'}\bmod n$, and sends C to the issuer. Additionally, TPM proves to the issuer knowledge of f and t', it executes as prover the protocol $SPK\{(f,t'):C=a^f b^{t'}\bmod n\}(\ '\ )$ with the issuer as the verifier.

(3) The issuer checks that $C\in QR(n)$. If this is the case and U is correct, the issuer selects a random $t''\in(0,2^{\lambda_2})$ and a random prime $e\in(2^{\gamma_1}-2^{\gamma_2},2^{\gamma_1}+2^{\gamma_2})$ and computes $A=(Cb^{t''})^{1/e}\bmod n$. Finally, the issuer sends (A,e,t'') to the host.

(4) The host stores (A,e) and forwards (A,e,t'') to the TPM. The TPM computes $t=t'+t''$, verifies that

$$A^e=a^f b^t \bmod n$$

and stores t.

B.3 Sign

Let m be the message to be signed. The TPM has secret key (f,t) and a credential (A,e), whereas the host only knows the credential (A,e). The signing algorithm takes the following steps:

(1) The host picks random integer $w\in_R\{0,1\}^{2\ell_p}$ and computes $T_1=Ag^w\bmod n$, $T_2=g^e h^w\bmod n$.

(2) The TPM and host together produce a signature of knowledge. That is, they compute the signature of knowledge

$$SPK\{(e,f,t,w):T_1^e g^{-ew}=a^f b^t\bmod n\wedge T_2=g^e h^w\bmod n\wedge T_2^{-e}g^{ee}h^{ew}=1\bmod n\}(m)$$

a) The TPM picks random integer $r_t\in\{0,1\}^{\varepsilon(\lambda_2+k)}$ and computes $\tilde{R}_1=b^{-r_t}\bmod n$. The TPM sends $\tilde{R}_1$ to the host.

b) The host picks random integers $r_e\in\{0,1\}^{\varepsilon(\gamma_2+k)}$, $r_w\in\{0,1\}^{\varepsilon(2\ell_p+k)}$, $r_{\delta_1}\in\{0,1\}^{\varepsilon(2\gamma_1+k+1)}$, $r_{\delta_2}\in\{0,1\}^{\varepsilon(\gamma_1+2\ell_p+k+1)}$ and computes

$$R_1=\tilde{R}_1 T_1^{r_e}g^{-r_{\delta_2}}\bmod n$$

$$R_2=g^{r_e}h^{r_w}\bmod n,$$

$$R_3=T_2^{-r_e}g^{r_{\delta_1}}h^{r_{\delta_2}}\bmod n.$$

c) The host also computes $c_h=H(g\|h\|n\|a\|b\|T_1\|T_2\|R_1\|R_2\|R_3)$ and sends c_h to the TPM.

d) The TPM chooses a random $n_t\in\{0,1\}^k$, computes $c=H(H(c_h\|n_t)\|m)$ and $s_t=r_t+ct$, and sends c,n_t,s_t to the host.

e) The host computes $s_e=r_e+c(e-2^{\gamma_1})$, $s_w=r_w+cw$, $s_{\delta_1}=r_{\delta_1}+c\cdot e\cdot e$, $s_{\delta_2}=r_{\delta_2}+c\cdot e\cdot w$. The signature is

$$\sigma = (T_1, T_2, c, n_t, s_e, s_t, s_w, s_{\delta_1}, s_{\delta_2}).$$

B.4 Verify

A signature $\sigma = (T_1, T_2, c, n_t, s_e, s_t, s_w, s_{\delta_1}, s_{\delta_2})$ on a message m w.r.t the public key (n, a, b, g, h) is as verified as follows.

(1) Compute

$$R_1^{'} = T_1^{s_e + c2^{\eta}} g^{-s_{\delta_2}} b^{-s_t} (T_3)^{-c} \bmod n,$$

$$R_2^{'} = T_2^{-c} g^{s_e + c2^{\eta}} h^{s_w} \bmod n,$$

$$R_3^{'} = T_2^{-(s_e + c2^{\eta})} g^{s_{\delta_1}} h^{s_{\delta_2}} \bmod n.$$

(2) Verify that

$$c \stackrel{?}{=} H(H(H(g \| h \| n \| a \| b \| T_1 \| T_2 \| R_1^{'} \| R_2^{'} \| R_3^{'}) \| n_t) \| m)$$

$$s_e \stackrel{?}{\in} \{0,1\}^{\varepsilon(\gamma_2 + k)+1}, \; s_t \stackrel{?}{\in} \{0,1\}^{\varepsilon(\lambda_2 + k)+1},$$

$$s_w \stackrel{?}{\in} \{0,1\}^{\varepsilon(2\ell_p + k)+1}, \; s_{\delta_1} \stackrel{?}{\in} \{0,1\}^{\varepsilon(2\gamma_1 + k + 1)+1},$$

$$s_{\delta_2} \stackrel{?}{\in} \{0,1\}^{\varepsilon(\gamma_1 + 2\ell_p + k + 1)+1}.$$