Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

An Optimized Direct Anonymous Attestation for Trusted Computing Platform

Cheng Song, Wei-ping Peng, Zong-pu Jia, Hai-ru Guo

School of Computer Science and Technology, Henan Polytechnic University,
Jiaozuo, Henan, 454000, China
songcheng@hpu.edu.cn

Received July, 2017; revised January, 2018

Abstract. *Direct Anonymous Attestation (DAA) mechanism not only can prove its legal identity to the remote entity, but also protect the privacy of the trusted attestation platform. However, the DAA scheme is designed complexity and has a large computational complexity, so it is not suitable for resource-constrained embedded or mobile equipment. To solve the computing bottleneck problem, this paper proposed a DAA scheme based on bilinear pairings. This scheme is provably secure under the q-strong Diffie-Hellman assumption and decision Diffie-Hellman assumption. The analysis shows that our scheme is more efficient than previous schemes, with a similar level of security. In addition, our scheme requires even fewer TPM resources. Consequently, the proposed scheme has great theoretical significance and application value in in field of the trusted computing research.*

Keywords: Trusted computing platform, Direct Anonymous Attestation (DAA), Bilinear pairings, Knowledge proof.

1. **Introduction.** With the rapidly development of the computer network and communication technology, the equipment reliability is a matter of great concern. To solve the problem of equipment reliability, the trusted computing technology [1] has always been a hot research topic both in the academic circles and business circles. Its core technology is a specialized chip which is generally called Trusted Platform Module (TPM) [2]. Generally, the platform embedded TPM is called the trusted computing platform(TCP)[3]. TCP can enable remote authentication of a trusted computer whilst preserving privacy of the platform's user based DAA which is a cryptographic primitive.

Generally a DAA scheme involves three types of entities: an issuer, signers and verifiers. The issuer has two functions. One is in charge of verifying the legitimation of signers and the other is in charge of issuing a membership credential to each signer. A signer can prove membership anonymously to a verifier by creating a DAA signature. The verifier can verify the membership of the signer from the DAA signature but he cannot learn the identity of the signer. One interesting feature of DAA is that the signer role of DAA is split between a TPM and a host that has the TPM attached. There are two reasons. One is that the TPM has limited storage, bandwidth, and computational capability and the other is that the host is less trusted. The TPM is the real signer and holds the private signing key. The host is a helper with more computational power. The host helps the TPM to compute DAA signatures udder the credential, but is not allowed to learn the private signing key or forge a DAA signature without the involvement from the TPM. Brickell, Camenisch, Chen, etc. [4] propose the first concrete instance of a Direct Anonymous Attestation scheme. Their scheme is based upon RSA and support for this scheme is mandated by the TPM specification version 1.2 which has been defined as an ISO/IEC international standard. Because the design of this scheme in[4] is complicated, the costs of storage and computation is large. This scheme is inefficient. Especially it is restricted for the embedded or mobile equipments which have limited resource. In[5], the analysis discovers vulnerability in the RSA-based scheme which can be exploited by a passive adversary and, under weaker assumptions, corrupt issuers and verifiers. After DAA was first introduced, it has drawn a lot of attention from both industry and cryptographic community, so lots of improved schemes [6-15] are proposed. The schemes [4][6][15] are based on RSA cryptosystem which is relatively complicated in computation. The schemes [7-13] are based on the bilinear pairings in Elliptic Curve Cryptosystem(ECC). ECC is more effective than RSA. Under LRSW assumption[16], the schemes [7,8,11] are security. The scheme [9] need less computational overhead, but in this scheme there is a potential security problem that the verifier can't identify the malicious TPM because of a design fault in the signing process. The security of schemes [10, 12-13] is under q-SDH assumption[16].

In general, DAA schemes under q-SDH-based assumption is more effective than those under LRSW-based assumption, but for the mobile equipment or embedded equipment which have low computational power, there is still a problem of computation bottleneck. The schemes [14, 15] are specially designed for low computational power equipment such as Mobile, M2M. The scheme [14] can't verify the correctness of join process. Although the design of the scheme[15] is simplified, it is based on RSA. So the computation costs is relatively large. Recently, several improved schemes were presented. in[17], an Provably Secure Anonymous Attestation scheme was presented to fix the TPM2.0, but the scheme is more complex. In[18], a universally composable Direct Anonymous Attestation was presented and gave a comprehensive security definition. In[19], an efficient Direct Anonymous attention scheme based strong Diffie-Hellman assumption was presented for privacy-protecting authentication. In[20], the Direct Anonymous attention scheme was used in Vehicular ad-hoc network(VANETs) to enhance privacy.

In order to ensure the security of the attestation and increase efficiency, we present an improved DAA scheme which is based on bilinear pairings. We give the proof on the security of our protocol under q-SDH and DDH assumption. We also give the proof on the correctness and the analysis on the efficiency. The further analysis shows that the proposed scheme is correct, security and effective. So The scheme is not only applicable to PC equipment with strong computing power, but also suits embedded equipment with low computing power, such as Mobile, Internet of Things, etc.

The rest of this paper is organized as follows: In section 2 we introduce security models and notions. Section3 we introduce the preliminary knowledge. We describe the improved DAA scheme which we present in section4. We discuss the correctness, security, efficiency of the improved scheme in section5. Finally, conclusions are presented in Section6.

2. **Formal Security Models and Notions of DAA.** In this section, we review the security model and notions of DAA proposed in[8]. There are four types of players in the security of a DAA scheme: an issuer I, a TPM M_i , a host H_i , a verifierV_j and a revocation oracle O. M_i and H_i form a platform in the trusted computing environment and share the role of a DAA signer. A basename bsn is used for controlling the linkability. These operations in the ideal system are as follow.

Setup: All players first indicate themselves to T whether or not they are invalid, where T is a trusted third party. Each TPM M_i sends its unique identity id_i to T who forwards it to the respective host H_i .

Join: The host H_i requests T to become a member. T contacts the corresponding TPM M_i and confirm whether the TPM wants to become a member. Then, T contacts and asks the issuer I whether the TPM M_i with identity id_i is valid. If M_i was valid and the issuer agreed to become a member, T admits the host H_i to become a member and tells that it has become a member. If M_i was tagged rogue, T tell the issuer I this and aborts the request.

Sign: On input of sk_i ,cre_i , a basename bsn_j (the name string of v_j or a special symbol $\perp$), and a message m that includes the data to be signed and the verifiers nonce n_V for freshness, M_i and H_i use this randomized algorithm to produce a signature σ on m under (sk_i , cre_i) associated with bsn_j . The basename bsn_j is used for controlling the linkability.

Sign/Verify: On input of m, bsn_j , a candidate signature σ for m , and a set of revoked secret keys RL , v_j uses this deterministic algorithm to return either 1 (accept) or 0 (reject). How to build the revocation list is out the scope of the DAA scheme.

Link: On input of two signatures σ_0 and σ_1 , $v_j$uses this deterministic algorithm to return 1 (linked), 0 (unlinked) or $\perp$) (invalid signatures). Link will output$\perp$) if, by using an empty , either Verify(σ_0) = 0 or Verify(σ_1) = 0 holds. Otherwise, Link will output 1 if signatures can be linked or 0 if the signatures cannot be linked.

3. **Preliminaries.**

3.1. **Bilinear Pairings.**

The DAA scheme uses bilinear pairings as a fundamental building block. Firstly, we briefly describe some background on bilinear pairings.

Definition1[17]: Let G_1 , G_2 be two multiplicative groups of prime order p. Let g_1 denote a generator of G_1 and g_2 denote a generator of G_2 . We say $e : G_1 \times G_2 \to G_T$ is an admissible bilinear pairing, if it satisfies the following properties:

Bilinear: For all $u \in G_1, v \in G_2$ and all $a, b \in Z_p$,$e(u^a, v^b) = e(u, v)^{ab}$.

Non-degenerate: $e(g_1, g_2) \neq 1$ and is a generator of G_T ;

Computabe: For any $u \in G_1, v \in G_2$, there is an effective algorithm for computing $e(u, v)$.

We call the two groups (G_1, G_2) in the above a bilinear group pair. In the rest of the paper, we consider bilinear pairing $e : G_1 \times G_2 \to G_T$ where G_1 , G_2 and G_T are multiplicative groups of prime order p.

More details about bilinear pairings can refer to the work[22-24]

3.2. **q-strong Diffie-Hellman (q-SDH) assumption.**

The security of our DAA scheme is related to the hardness of the q-SDH problem. In the following, we briefly introduce q-SDH question:

Definition2[17]: Let G_1 and G_2 be two multiplicative cyclic groups of prime order p. Let g_1 denote a generator of G_1 and g_2 denote a generator of G_2 . The q-SDH problem in (G_1, G_2) is defined as follows: Give a (q+2)-tuple of $(g_1, g_2, g_2^r, g_2^{(r^2)}, ...g_2^{(r^q)}$) as input, output a pair ($g_1^{1/(r+x)}$, x), where $x \in Z_p^*$. An algorithm A has advantage ε in solving q-SDH question in (G_1, G_2) if $\Pr[A(g_1, g_2, g_2^r, g_2^{(r^2)}, ...g_2^{(r^q)}) = (g_1^{1/(r+x)}, x)] \geq \varepsilon$, where the probability is based on the random choice of $r \in Z_p^*$ and the random choice of polynomial time algorithm A.

3.3. **Decision Diffie-Hellman (DDH) assumption.**

Definition3[17]: Let G be a multiplicative cyclic group of prime order p ; Let g denote a generator of G. The Decisional DiffieHellman (DDH) problem in G is defined as follows: Give four elements $g, g^a, g^b, g^c \in$

G as input, output 1 if $ab = c$ and 0 otherwise, where $a, b, c \in Z_p^*$. An algorithm A has advantage ε in solving DDH question in G If $\begin{array}{l} |\Pr[g \in G, a, b \in Z_p^* : A(g, g^a, g^b, g^{ab}) = 1] \\ -\Pr[g \in G, a, b, c \in Z_p^* : A(g, g^a, g^b, g^c) = 1]| \geq \varepsilon \end{array}$, where the probability is based on the random choice of $a, b, c \in Z_p^*$ and the random choice of polynomial time algorithm A. DDH question is hard to be solved if probability can be ignored.

4. **Improves DAA Scheme based on Bilinear Pairings.** Now we present an improved DAA scheme based on bilinear pairings. In the DAA scheme, there are three types of entities: an issue, TPM, Host and verifiers. The DAA mechanism involves four phases: setup phase, DAA join phase, DAA sign phase and verify phase. We have the following four operations.

4.1. **Setup.** Give a security parameter as input, the operation steps of setup algorithm are as follows:

Step1: Let G_1 and G_2 be two multiplicative groups of prime order p . Let g_1 denote a generator of G_1 and g_2 denote a generator of G_2 . Let $e : G_1 \times G_2 \to G_T$ be an admissible bilinear pairing, where (g_1, g_2) is generator of G_T .

Step2: The issuer chooses $x \in Z_p^*$,$y \in Z_p^*$ uniformly at random and computes $X : = g_2^x ; Y := g_2^y$

Step3: The issuer chooses 5 collision resistant hash functions:$H_1 : \{0, 1\}^* \to \mathrm{Z_p}$,$H_2 : \{0, 1\}^* \to \mathrm{Z_p}$, $H_3 : \{0, 1\}^* \to \mathrm{Z_p}$, $H_4 : \{0, 1\}^* \to \mathrm{Z_p}$, $H_5 : \{0, 1\}^* \to \mathrm{Z_p}$.

Step4: The issuer computes $A := {g_1}^y$, and sets the group public key as PK and its private key as ISK :

$$\mathrm{PK} : =\{\mathrm{G_1,G_2,G_T,X,Y,A,p,e,g_1,g_2,H_1,H_2,H_3,H_4}\}$$

$$\mathrm{ISK} := \{x, y\}$$

4.2. **Join.** The join protocol phase involves three types of entities: TPM, host and issuer. Before this phase, We assume that the issuer and TPM have established a one-way authentic channel. The issuer needs to be sure that it talks to the right signer(TCP) . The authentic channel can be achieved in various ways. The one TCG recommended is that every message sent from the issuer to the signer is encrypted under the TPM endorsement key[2]. The protocol steps are as follows:

Step1: The issuer randomly chooses a string $\mathrm{n_I} \in \{0, 1\}^{\mathrm{t}}$, and sends it to TPM.

Step2: After receiving n_I , TPM computes privacy key $f := \mathrm{H_1(DAAseed||cnt||K_I)}$ and $D := A^f$, where $DAAseed$ is the seed to compute the secret key of TPM, cnt is a counter for tracking the times that TPM executes the join protocol, and K_I is a long-term public key of issuer for authenticating the public key PK. TPM then randomly chooses $l_{\mathrm{f}} \leftarrow \mathrm{Z_p}$, computes $R := A^{l_{\mathrm{f}}}$,$c := \mathrm{H_2(PK||n_I||D||R)}$,$k_f := l_f + cf(\mathrm{mod} p)$, and then TPM sends ($\mathrm{D}, c, \mathrm{k_f}, \mathrm{n_I}$) to the issuer.

Step3: After receiving ($\mathrm{D}, c, \mathrm{k_f}, \mathrm{n_I}$), the issuer firstly verifies the correctness of n_I and checks its record(rogue TPM list RL) and policy to find out whether the value D should be rejected or not. If D belongs to a rouge TPM or doesn't pass the policy, the issuer aborts the protocol. Otherwise, the issuer computes $\hat{\mathrm{R}} : = \mathrm{A}^{k_{\mathrm{f}}} \cdot D^{-c}$ and verifies that $c \overset{?}{=} H_2(PK \parallel n_I \parallel D \parallel \widehat{R})$. If the verification succeeds, the issuer computes $C := {g_1}^x D^x$ and let cre : $=(C)$ to be the credential for TPM.

Step4: TPM sends cre to the host.

Step5: The host verifies the $\mathrm{e}(g_1 \mathrm{D}, X) \overset{\mathrm{e}}{=} (C, g_2)$. If the verification fails, the protocol is aborted.

4.3. **Sign.** The sign protocol phase involves two types of entities: TPM and host. The protocol steps are as follows:

Step1: Depending on whether $bsn = \perp$)or not. if $bsn = \perp$), TPM chooses $B \leftarrow \mathrm{G}_1$ at random; otherwise, TPM computes $B := \mathrm{H}_3(\mathrm{bsn})$. Where bsn is a basename associated with verifier. In[9] there is an introduction about why and how to choose bsn . TPM chooses $\mathrm{r_f} \leftarrow \mathrm{Z_p}$ at random and computes $K := \mathrm{A}^{r_f}$, and then sends (K, B) to the host.

Step2: After receiving the message(K, B) , the host chooses $a \leftarrow Z_p$ at random and computes $T_1 := {g_1}^a$,$T_2 := A^a$,$T_3 := C^a$, $T_4 := D^a$, $c_h := H_4(PK||B||K||T_1||T_2||T_3||T_4||n_V)$, and sends c_h to TPM, where n_V is a random nonce provided by the verifier.

Step3: After receiving the message c_h , TPM chooses $n_T \in \{0,1\}^t$ at random and computes $c := H_5(c_h||n_T||m)$, $s_f := r_f + cf(\mathrm{mod} p)$, where m is the message to be signed. Then TPM sends (c, n_T, s_f) to the host.

Step4: The host computes the signature $\sigma := (B, K, T_1, T_2, T_3, T_4, c, n_T, s_f)$.

4.4. **Verify.** The input to this signature verification algorithm includes a message m , the basename bsn , the noncen_V , DAA public key PK , a candidate signature $(B, K, T_1, T_2, T_3,$ $T_4, c, n_T, s_f)$ and a list RL of secrete keys for rogue TPM. and the verification process is as follows:

Step1: The verifier verifies $\mathrm{B,K,T_1} \overset{?}{\in} G_1$ and $s_f \overset{?}{\in} Z_p$.

Step2: For each $f' \in RL$, the verifier check that ${T_2}^{f'} \overset{?}{=} T_4$. If it holds, the verifier outputs 0 and aborts the protocol, otherwise, outputs 1 and executes the next step.

Step3: Compute $K' := A^{s_f} D^{-c}$.

Step4: The verifier verifies that $c \overset{?}{=} H_5(H_4(PK||B||K'||T_1||T_2||T_3||T_4||n_V)||n_T||m)$. If the verification succeeds, the verifier outputs 1 and executes the next step, otherwise, outputs 0 and aborts the protocol.

Step5: The verifier verifies that $e(T_1T_2, X) \overset{?}{=} e(T_3, g_2)$ and $e(T_1, Y) \overset{?}{=} e(T_2, g_2)$. If the verification succeeds, the verifier outputs 1 and accepts the signature, otherwise, outputs 0 and rejects it.

5. **Discussion.** In this section, we discuss the correctness, security, efficiency of the optimistic DAA scheme.

5.1. **Correctness.** To show our scheme is correct, we prove that the issuer is talking to a right signer and a signature can be successfully verified by any verifier. So the correctness of scheme is reflected in two aspects: one is the correctness that the issuer verifies the trusted platform in join protocol, the other is the correctness that the verifier verifies the trusted platform in sign protocol.

(1) Correctness proof of join protocol

Proof: The correctness proof of join protocol is to verify that $R \overset{?}{=} \hat{R}$.

Because:

$$\begin{aligned} \hat{\mathrm{R}} &= \mathrm{A}^{k_\mathrm{f}} \cdot D^{-c} \bmod p \\ &= \mathrm{A}^{k_\mathrm{f}} \cdot \mathrm{A}^{-c\mathrm{f}} \bmod p \\ &= A^{sf-cf} \bmod p \\ &= A^{r_f} \bmod p \\ &= R \end{aligned}$$

it is correct.

(2) Correctness proof of sign protocol

Proof: The correctness proof of sign protocol is to verify that $K \overset{?}{=} K'$, $e(T_1T_2, X) \overset{?}{=} e(T_3, g_2)$ and $e(T_1, Y) \overset{?}{=} e(T_2, g_2)$.

Because:

$$\begin{aligned} K' &= A^{s_f} D^{-c} \\ &= A^{s_f} A^{-cf} \bmod p \\ &= A^{s_f - cf} \bmod p \\ &= A^{r_f} \bmod p \\ &= K \end{aligned}$$

$$\begin{aligned} & e(T_1 T_2, X) = e({g_1}^a A^a, {g_2}^x) \bmod p \\ & = e({g_1}^{a+ayf}, {g_2}^x) \bmod p \\ & = e({g_1}^{(x+xyf)a}, g_2) \bmod p \\ & = e({g_1}^{ax} D^{ax}, g_2) \bmod p \\ & = e(C^a, g_2) \bmod p \\ & = e(T_3, g_2) \end{aligned}$$

$$\begin{aligned} & e(T_1, Y) = e({g_1}^a, {g_2}^y) \bmod p \\ & = e({g_1}^{ay}, g_2) \bmod p \\ & = e(A^a, g_2) \bmod p \\ & = e(T_2, g_2) \end{aligned}$$

So it is correct.

5.2. **Security.** Under the secure q-Strong DH assumption and Decisional DH assumption, our scheme is provable secure. We discuss the security of the scheme from the following four aspects: confidentiality of f , anti-rogue TPM, anonymity and unforgeability.

(1) Confidentiality of f

Proof: The confidentiality of private key f is crucial. The private key f is secretly kept in TPM always. If a DAA scheme can't resist the private key f compromise impersonation attack, it can't ensure the scheme is reliable. Considering that the security of host is weaker than that of TPM, all computations involving f must be executed in TPM. If necessary, TPM shall prove that it possesses the private key f by adopting zero-knowledge proof. Therefore, the scheme ensures the absolute confidentiality of the secret information f .

(2) Anti-rogue TPM

A DAA scheme shall be able to resist spoofing attacks from rogue TPM. In our scheme the legitimation of TPM is verified in both the join phase and verify phase. Proof In the join phase, after receiving $(\mathrm{D}, c, \mathrm{k_f}, \mathrm{n_I}\ )$, the issuer firstly verifies the correctness of n_I and checks its record(rogue TPM list RL) and policy to find out whether the value D should be rejected or not. If there is some $f_i \in RL$ to let that $D{=}A^{f_i}$, the TPM is regarded as a rogue TPM and rejected.

In the verify phase, after receiving the signature $\sigma := (B, K, T_1, T_2, T_3, T_4, c, n_T, s_f)$, the verifier verifies that ${T_2}^{f'} \overset{?}{=} T_4$ for each $f' \in RL$ to find out whether the TPM is valid or not. If there is some $f_i \in RL$ to let that ${T_2}^{f_i}{=}T_4$, the TPM is regarded as a rogue TPM and rejected.

(3) Anonymity

Anonymity is one of the most important functions of DAA scheme, so a DAA scheme must be able to fulfill anonymity.

Proof: In the sign phase of our scheme, the host does not directly send the credentials cre : =(C) issued by the issuer to TPM. The host first computes:

$$\begin{aligned} T_1 &:= {g_1}^a \\ T_2 &:= A^a \end{aligned}$$

$$T_3 := C^a$$
$$T_4 := D^a$$
$$c_h := H_4(PK||B||K||T_1||T_2||T_3||T_4||n_V)$$

.

TPM computes:

$$c := H_5(c_h||n_T||m) \bmod p$$
$$s_f := r_f + cf(\text{mod} p)$$

Then sends (c, n_T, s_f) to the host.

The host can get the signature $\sigma := (B, K, T_1, T_2, T_3,\ T_4, c, n_T, s_f)$ and send it to the verifier. Obviously, there is not any common information between the verifier and the issuer. So they cannot recognize the specific TPM even if they collude. therefore, this scheme meets the security features such as non-traceability, anonymity.

(4) Unforgeability In this subsection we will discuss the protocol security about unforgeability in ROM (random oracle model). In[25] Bellare and Rogaway first proposed ROM as a non-standard computation model. In the model any concrete object such as hash function is regard as a random object. The query to hash function is changed into an oracle outputting a random response in the uniform distribution field. The reduction method is adopted to prove the protocols security in ROM. It proves that there exists an adversary compromising the cryptographic protocol with non-negligible probability. Another algorithm can be constructed to solve the public mathematical hard problem by invoking the protocol adversary with non-negligible probability.

Theorem 5.1. *(unforgeability). In the random oracle model and under the q-strong Diffie-Hellman assumption and decision Diffie-Hellman assumption, the optimized DAA scheme provides the property of unforgeability; more exactly, if adversary A can forge the DAA signature with the non-negligible probability, there exists a simulator S solving the q-strong Diffie-Hellman assumption and decision Diffie-Hellman assumption with non-negligible probability in the polynomial time.*

Proof: If the adversary A can forge the DAA signature and the credentials in the attestation, we can make use of A to construct an algorithm B to solve the q-strong Diffie-Hellman assumption and decision Diffie-Hellman problems. We will illustrate the construction of the simulator S: S interacts with the adversary playing the attacking game in the join protocol and sign protocol phases.

Without loss of generality, we assume that the challenger B receives the random instance $(\mathrm{G_1,G_2,G_T,X = g_1^x, Y = g_2^y, A, p, e,g_1,g_2,H_1,H_2\ ,H_3,H_4})$ of computing q-strong Diffie-Hellman assumption and decision Diffie-Hellman problems. By interacting with the adversary A, the challenger B will return (x, y) in the following game. Initialization. the challenger B runs the KeyGen algorithm to obtain the system public key $(\mathrm{G_1,G_2,G_T,X = g_1^x, Y = g_2^y, A\ , p, e,g_1,g_2,H_1,H_2\ ,H_3,H_4})$ and the system private key (x, y) of the scheme. $X\ :=g_2^x$ let (g_2, g_2^x) is a challenge to the q-strong Diffie-Hellman and decision Diffie-Hellman assumption. Also, the adversary A is given the parameters $y\ ,Y\ :=g_2^y$. the adversary A is able to query the help oracle H and the target oracle T under the q-strong Diffie-Hellman and decision Diffie-Hellman assumption

Oracle query: the adversary A can access the target oracle T to get a random element $g \in G_2$. And the help oracle H to obtain g_a for some input $g \in G_2$, respectively, in the q-strong Diffie-Hellman and decision Diffie-Hellman assumption. Then the adversary A will oracle query.

the adversary A can make two kinds of oracle queries. hash query from the oracle Oh and signing queries from oracle Os, the detail process is as follows:

1) Oh query

the adversary A queries Oh for the hash value on the input cre : $=(C)$, the challenger B will check whether cre : $=(C)$ had been queried or not.

If cre : $=(C)$ had been queried, retrieve g from the list Lh by taking cre : $=(C)$ as the search index. Otherwise, the challenger B will query H to obtain a random element $g \in G_2$ and stores (C, g) in Lh for preserving consistency. Return g to the adversary A.

2) Os query

In order to get the signature, the adversary A uses n_T ,r_f as input. the challenger B inputs n_T ,r_f to T to get the output $c_h := H_4(PK||B||K||T_1||T_2||T_3||T_4||n_V)$,$c := H_5(c_h||n_T||m)$ and $s_f := r_f{+}cf(\mod p)$. the challenger B calculation $\sigma := (B, K, T_1, T_2, T_3, T_4, c, n_T, s_f)$. Send the output σ to A.

Forgery and problem: In the DAA-signing protocol, the simulator forges signatures by using the zero-knowledge simulator and the power over the random oracle. In cases where the adversary manages to do any of the followings: to sign a signature on behalf of a honest signer, or to tag an honest TPM as a rogue, or to create a signature since they are associated with the same basename and signing key but the forgeability does not exist, the simulator fails. We can show that these cases cannot occur; otherwise the adversary can forge a signature, which is contradiction to the q-strong Diffie-Hellman assumption and decision Diffie-Hellman problems. We can show that these cases cannot occur. We then show that if the simulator does not fail, under the DBDH assumption, the environment will not be able to tell whether or not it is run in the real system interacting with the adversary or the ideal system interacting with the simulator.

Table 1. Calculation Consumption Comparison of DAA Scheme based on Bilinear Pairings

DAA scheme	Join Protocol (Join)			Sign Protocol (Sign)		Verify (Verify)
Participants	TPM	Host (H)	Issuer (I)	TPM	Host (H)	Verifier(V)
Scheme of [7]	$3G_1$	$6P$	$(n+2)G_1+2G_1^2$	$3G_T$	$3G_1+G_T+3P$	$G_T^2+G_T^3+5P+(n+1)G_T$
Scheme of [10]	$2G_1$	G_1+2P	$(n+1)G_1+G_1^2$	$2G_1+G_T$	$G_1+G_T^3$	$G_1^2+G_2^2+G_T^4+P+nG_1$
Scheme of [11]	$3G_1$	$4P$	$(n+2)G_1+G_T$	$2G_1+G_T$	$3G_1+P$	$G_1^2+G_2^2+5P+nG_1$
Scheme of [12]	$3G_1^2+2P$	$2P$	$nG_1+G_1^2+G_1^3$	$2G_1+G_1^2$	$G_1+2G_1^2+G_1^3+G_T^3$	$G_1^2+2G_1^3+G_T^5+3P+nG_1$
Scheme of [13]	$2G_1$	G_1+2P	$(n+1)G_1+G_1^2$	$3G_1$	$G_1+G_1^2+G_T+P$	$G_1^2+G_2^2+G_T^4+P+nG_1$
Our Scheme	$2G_1$	$2P$	$(n)G_1+2G_1^2$	G_1	$4G_1$	$G_1^2+4P+nG_1$

5.3. **Efficiency.** We analyze the efficiency of the optimized scheme by comparing the typical schemes based on bilinear pairings only. The scheme[7] has the same efficiency as the scheme[8]. The efficiency the scheme [9] is obviously improved, but there is a potential security problem in the sign phase. The correctness of join phase can't be verified in the scheme [14]. Therefore, we only compare the schemes [7, 10-13] with our scheme in the Table1. The computation of the setup phase is only calculated once, and can be performed off-line ahead. Therefore, we just consider the computation costs of join phase (including TPM, Host and Issuer), sign phase (including TPM and Host) and verify phase in Table1. For comparing the efficiency of our scheme with other DAA schemes in the following table, we use G_i $(i \in \{1, 2, T\})$ to denote an exponentiation operation in groupG_i ,G_i^m to denote a multi-exponentiation of m exponent in group G_i, P to denote a pairing exponentiation, and n to denote the numbers of keys in the rogue TPM list. We assume that the rogue TPM will be detected by the issuers and verifiers in each scheme.

When analyzing the efficiency of a DAA scheme, we give priority to concern the computation costs of TPM, because it has limited storage, bandwidth, and computational capability. It shows that a decided advantage of our scheme is that TPM has less computation costs than other schemes.

6. **Conclusion.** In this paper, we have introduced an optimized DAA scheme, which is based on bilinear pairings. Analysis shows that this scheme is more efficient than all the existing DAA schemes, in particular, this scheme requires very few TPM resources compared to the existing DAA schemes. We believe the scheme has important theoretical significance and application value in field of the trusted computing research.

Acknowledgement This work was supported by the Natural Science Foundation of China project (61300124, 61300216); Henan Provincial Department of Science and Technology Research Project (132102210123).

REFERENCES

[1] TCG: TCG Specification Architecture Overview, *[S]. TCG Specification Version 14. August* 2007.

[2] TCG: TPM Main, Part 1: Design Principles, *[S]. TCG Specification Version 1.2 Revision103.* July 2007.

[3] Pearson S, Balacheff B, Chen L., Plaquin D, Proudler G. Trusted Computing Platforms: TCPA Technology in Context*[M]. Prentice Hall PTR, Upper Saddle River*, NJ, 2002.

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

[5] B. Smyth , M. D. Ryan , Liqun Chen. Formal analysis of privacy in Direct Anonymous Attestation schemes *Journal of In: Science of Computer Programming*, vol. 111, pp300-317, 2015.

[6] H. Ge and S. R. Tate A Direct anonymous attestation scheme for embedded devices *Journal of In Public Key Cryptography-PKC 2007, Springer-Verlag LNCS 4450*, 2007.

[7] E. Brickell, L. Chen ,J. Li, A new direct anonymous attestation scheme from bilinear maps *Journal of In: Lipp P., Sadeghi A.-R., Koch, K.-M. (eds.) Trust 2008. LNCS*, vol. 4968, pp. 166178. Springer, 2008.

[8] E. Brickell , L. Chen ,J. Li, Simplified security notions of direct anonymous attestation and a concrete scheme from pairings *Journal of International Journal of Information Security*, vol. 8, no. 5, pp. 315330, 2009.

[9] L. Chen, P. Morrissey , Smart N.P. Pairings in trusted computing *Journal of In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS,* vol. 5209, pp. 117. Springer, Heidelberg (2008).

[10] L. Chen, A DAA scheme requiring less TPM resources, *[C]. In: Proceedings of the 5th China International Conference on Information Security and Cryptology, LNCS. Springer, Heidelberg* 2009.

[11] L. Chen, P. Morrissey , N.P. Smart DAA: Fixing the pairing based protocols, *[Z]. Cryptology ePrint Archive, Report 2009/198* 2009.

[12] X. Chen ,D. Feng, Direct anonymous attestation for next generation TPM *Journal of Journal of Computers,* vol. 3, no. 12, pp. 4350, 2008.

[13] E. Brickell ,J. Li, A Pairing-Based DAA Scheme Further Reducing TPM Resouses*[C]. In: Acquisti A., Smith S. W. and Sadeghi A.-R. (eds.) Trust 2010. LNCS 6101*, pp. 181-195. 2010.

[14] Y. Y, He, L. Chen, L. L. Wang. An Improved Direct Anonymous Attestation Scheme for M2M Networks *Journal of Elsevier Procedia Engineering*,vol.5, pp.1481-1486, 2011.

[15] D. D. Zhang ,Z. F. Ma, X, X, Niu, Anonymous authentication scheme of trusted mobile terminal under mobile Internet *Journal of The Journal of China Universities of Posts and Telecommunications*,vol.20, no. 1, pp. 58-65, 2013.

[16] A. Lysyanskaya, R.L. Rives, A. Sahai , Wolf S. Pseudonym systems *Journal of In: Heys H.M., Adams C.M. (eds.) SAC 1999. LNCS*, vol. 1758, pp. 184199. Springer, Heidelberg 2000.

[17] J Camenisch L Chen M Drijvers etc. One TPM to Bind Them All: Fixing TPM 2.0 for Provably Secure *Anonymous Attestation. Security Privacy ,* pp. 901-920, 2017.

[18] J Camenisch M Drijvers A Lehmann. Universally Composable Direct Anonymous Attestation*. Iacr International Workshop on Public Key Cryptography*, pp. 234-264, 2016.

[19] J. Camenisch M Drijvers A Lehmann, Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited. International Conference on Trust Trustworthy Computing, pp. 1-20, 2016.

[20] J. Whitefield L. Chen A. Giannetsos, Privacy-Enhanced Capabilities for VANETs using Direct Anonymous Attestation In: 2017 IEEE Vehicular Networking Conference (VNC), 2017.

[21] J. Li A. Rajan, An Anonymous Attestation Scheme with Optional Traceability [C]. In: International Conference on Trust Trustworthy Computing - 2010 , Trust 2010. LNCS 6101, pp. 196210.(2010).

[22] T. Y. Wu, Y.M. Tseng, An efficient user authentication and key exchange protocol for mobile client-server environments, *Computer Networks*, Vol. 54, no. 9, pp. 1520- 1530, June 17 2010.

[23] T. Y. Wu, Y.M. Tseng, Publicly verifiable multi-secret sharing scheme from bilinear pairings, IET Information Security, Vol. 7, no. 3), pp. 239-246, 2013.

[24] T. Y. Wu, T.T. Tsai, Y. M. Tseng, Efficient searchable ID-based encryption with a designated server, *Annals of telecommunications*, Vol. 69, no. 7, pp. 391-402, 2014.

[25] M. Bellare, R. Rogaway,. Random oracles are practical: A paradigm for designing efficient protocols. *In: Proceedings of the 1st CCS. New York: ACM Press*, pp. 6273, 1993.