EasyChair Preprint

№ 1253

An efficient property-based authentication scheme in the standard model

Xiaohan Yue, Xin Wang, Xibo Wang, Wencheng Cui and Yuan He

June 30, 2019

An efficient property-based authentication scheme in the standard model

Xiaohan Yue[1][0000-0002-9978-9560], Xin Wang[1], Xibo Wang[1], Wencheng Cui[1], Yuan He[1]

1 School of information science and engineering, Shenyang University of Technology, Shenyang 110870, China
E-mail: yxh21@yeah.net

Abstract. In order to solve the problem of platform configuration information leakage that are caused by the traditional platform authentication in the trusted computing environment, this paper proposes a novel property-based authentication (PBA) scheme. In this paper, we design the framework and define the security model of our scheme. Then we give the detail construction of our scheme. Comparing with existing PBA schemes, our PBA scheme is more effective than other schemes. At the aspect of security, this paper proves that our scheme meets correctness, unforgeability and configuration privacy in the standard model.

Keywords: Property-based attestation; Trusted computing; Bilinear pairing; Standard model

1 Introduction

In today's open distribute network environment, the spread of malicious code has caused huge losses to users and service providers. Therefore, it is necessary to establish a distributed trusted computing environment to ensure the predictable behavior of all parties. To achieve this goal, the computer industry has established a trusted computing organization TCG[1]and developed a trusted platform module TPM to ensure the integrity, confidentiality and authentication of the platform. In terms of platform authentication, TCG provides a solution for platform authentication, called binary authentication [2], that is, TPM as the verifier sends the measurement results (usually binary hash value) of software and hardware on the platform, also called integrity report, to the verifier, who checks the integrity report and evaluates its security. However, this method will destroy the privacy of the platform. As the authenticator reports the identification of the software and hardware in his own system, the adversary obtains the characteristics of the platform through the identification, which will lead to the following situations: first, differentiated services, remote service providers may refuse to provide services due to the business model adopted by the other party, for example, denial of service for Linux, or denial of service for certain platform configuration information by some chat software, etc.; Second, attacks on configuration. If the adversary knows that some hardware or software configurations exist on a large number of platforms by collecting platform configuration information, the adversary can implement targeted

attacks according to these configurations. Third, anonymity is destroyed, and adversaries can uniquely determine a platform based on platform configuration information. Aim at the above problems, in 2004, Sadeghi et al. [3] proposed the definition of property-based attestation (PBA), but there is no construction of PBA in this paper. The advantage of PBA is that different platforms may have different configuration specification (cs), but they have the same property specification (ps) to meet the same secure requirements. Compared with the binary attestation, the property-based attestation converts the original binary attestation into the attestation of the platform property, and the attestor can give the attestation of satisfying the property according to the target property that the verifier needs to verify. In terms of PBA scheme research, Chen et al[4] first proposed an property-based attestation scheme, and then proposed another property attestation scheme without a trusted third party[5]; Due to the efficiency bottleneck of the above schemes, Feng Dengguo et al [6] proposed a more efficient PBA scheme based on bilinear pairings. In recent years, Abir et al. [19] presented a secure cloud monitoring system by using PBA scheme; Nazanin et al. [20] proposed platform property certificate, based on the current certificates of the system as the model's property, and designed a practical PBA protocol.

However, these schemes are proved to be security in the random oracle model. But this model is an ideal model, which was proposed by Bellare and Rogaway in 1993 [7]. In this model, any object, such as hash function, can be regarded as a completely random object. However, in the actual scheme, because the hash function we used usually is specified, the output of hash function for each time is not really random, which may lead to the insecurity of the scheme. In fact, the defects of the random oracle model have been pointed out in several papers[8,9]. The proof under the standard model can clearly show that a provable secure cryptographic scheme cannot be corrupted unless the underlying mathematic problem is solved. Therefore, designing a PBA scheme which security can be proved in the standard model is the main research work of this paper.

Based on bilinear pairing, this paper uses group signature [10,11,12,13,14] technology and Groth-Sahai proof system [15,16] to propose a novel property-based attestation scheme that is provable secure and efficient in the standard model. The scheme has security properties such as unforgeability of attestation and configuration privacy. In terms of performance cost, compared with the existing schemes [4,5,6], our scheme has higher efficiency and shorter attestation value length.

2 Preliminaries

2.1 Bilinear pairing

Let g and G_T be two cyclic groups of order n, where n is a prime number and g is the generator of G. Bilinear mapping on two groups is defined as $e: G \times G \rightarrow G_T$, and satisfies the following properties:

Bilinear: $e(g^a, g^b) = e(g,g)^{ab}$, holds for all $a, b \in \mathbb{Z}_n^*$;

Non-degeneracy: $e(g,g) \neq 1_{G_T}$, where 1_{G_T} is a unit of G_T;

Computability: there is an effective algorithm to calculate $t = e(g,h)$.

2.2 Assumptions

Assumption 1. Subgroup Decision Assumption (SDA): Given that $n = pq$, $h \in G_n$, and $h \in G_p$, it is indistinguishable in polynomial time to determine which group h belongs to. The formal expression is as follows:

$Pr[\,(p,q,G,G_T,e,g) \leftarrow \text{BilinearSetup}(1^k); n = pq; h \in G_n: A(n,G,G_T,e,g,h) = 1] \approx Pr[\,(p,q,G,G_T,e,g) \leftarrow \text{BilinearSetup}(1^k); n = pq; h \in G_p: A(n,G,G_T,e,g,h) = 1]$

Assumption 2. q-Hidden Diffie-Hellman(q-HSDH) Assumption: Given that $g, g^x, u \in G_1$, $h, h^x \in G_2$ and $\{g^{\frac{1}{x+c_l}}, h^{c_l}, u^{c_l}\}_{l=1\ldots q}$, it is difficult to calculate $\left(g^{\frac{1}{x+c}}, h^c, u^c\right)$ in polynomial time, so there is a negligible function v that:

$Pr[\,(p,G_1,G_2,G_T,e,g,h) \leftarrow BilinearSetup(1^k); u \in G_1; x, \{c_l\}_{l=1\ldots q} \leftarrow Z_p; (A,B,C) \leftarrow A(p,G_1,G_2,G_T,e,g,g^x,h,h^x,u,\left\{g^{\frac{1}{x+c_l}}, h^{c_l}, u^{c_l}\right\}_{l=1\ldots q}): (A,B,C) = \left(g^{\frac{1}{x+c}}, h^c, u^c\right) \Lambda c \notin \{c_l\}_{l=1\ldots q}) < v(k)$

2.3 property-based attestation

Property-based attestation (PBA) scheme involves three entities, namely: attestor P (including host Hand trusted platform module PTM), verifier $\mathcal{V}$ and trusted third party issuer $\mathcal{I}$. In the whole PBA scheme, Attestor P asks issuer I for an property certificate of the current platform configuration information, and then proves to verifier $\mathcal{V}$ that the current platform configuration information is consistent with the property certificate

and has corresponding property. The issuer main work consists of two parts: issuing property certificates for platform configuration information and checking whether the property certificates are revoked; Verifier $\mathcal{V}$ verify the certificate given by attestor $\mathcal{P}$. In general, PBA scheme is mainly composed of five algorithms:

(1)Setup: Input the security parameter 1^k, issuer $\mathcal{I}$ use random algorithm to generate a pair of keys(ppk, tsk) , where tsk is the private key of issuer and ppk is the public key.

(2)Join: TPM collects platform configuration information cs and sends cs to issuer $\mathcal{I}$, who evaluates the cs as ps, then signs (cs, ps) with its own private key γ to generate property certificate cre and sends the certificate and (cs, ps) to attestor P.

(3)Attest: This process is to prove to verifier $\mathcal{V}$ that attestor P has a certificate on the property ps and that the current relevant platform configuration information is consistent with that in the certificate. Firstly, TPM carries out commitment calculation on the current platform configuration information cs to obtain a commitment value C_{cs} and signs it to obtain a signature value σ_M; Then host H blinds the certificate cre and calculates relevant witness values; Finally, attestor P obtains the property-based attestation σ_{PBA} according to the calculation results of TPM and host, and sends σ_{PBA} to verifier $\mathcal{V}$.

(4)Verify: Verifier $\mathcal{V}$ obtains the property-based attestation σ_{PBA} from attestor P. First verifier $\mathcal{V}$ verifies TPM's signature σ_M to ensure that the attestation information comes from a real TPM； Then verifier $\mathcal{V}$ checks correctness of the other parameters of σ_{PBA}. If positive, it indicates that attestorPhas a valid property certificate cre and that attestor attestor P 's current platform configuration information cs is consistent with the certificate cre. Finally verifier $\mathcal{V}$ sends C_{cs} and ps to issuer $\mathcal{I}$ to verify whether the property

certificate for (cs, ps) has been revoked. If all the above checks pass, verifier $\mathcal{V}$ outputs accept, otherwise reject.

(5)Check: On receiving the query from verifier $\mathcal{V}$, issuer uses its private key tsk to obtain the corresponding (cs, ps) pair, then checks the configuration-property database whether the (cs, ps) pair is existed in the revocation list RL, and forwards the result to verifier $\mathcal{V}$.

2.4 PBA security model

If a PBA scheme is secure, it will satisfy the following security properties:

(1) Correctness. If both the attestor and verifier are honest, (cs, ps) is not in the revocation list RL, then the attestation generated by the attestor will be regarded as valid by the verifier with overwhelming probability. This means that the PBA scheme must meet the following consistency requirements.

$((ppk, tsk) \leftarrow \text{Setup}(1^k)$, $(cs, cre) \leftarrow \text{Join}(ppk, tsk)$, $\sigma_{PBA} \leftarrow \text{Attest}(cs, ps, cre, ppk)) \Rightarrow 1 \leftarrow \text{Verify}(ps, \sigma_{PBA}, ppk, \text{RL})$

(2) Configuration Privacy. This PBA scheme has configuration privacy, that is, no adversary can win the following games in polynomial time.

-initialization: Challenger $\mathcal{C}$ runs Setup$(1^k)$and sends public key ppk and private key tsk to adversary $\mathcal{A}$.

-Queries: Adversary $\mathcal{A}$ adaptively queried challenger $\mathcal{C}$ in the following method:

Join: $\mathcal{A}$ sends the i-th Join request to challenger $\mathcal{C}$, challenger $\mathcal{C}$ selects $cs_i \in CS = \{cs_1, cs_2, \ldots, cs_n\}$, where *CS* is the same property set, and runs join algorithm to obtain certificate cre_i about property ps , then sends cre_i and ps to adversary $\mathcal{A}$.

Attest: Adversary $\mathcal{A}$ sends the i-th Attest request to challenger $\mathcal{C}$, and challenger $\mathcal{C}$ runs attest algorithm to generate certificate $\sigma_{PBA}^{(i)}$ and takes it as a response to Adversary $\mathcal{A}$.

Corrupt: Taking the index i as input, challenger $\mathcal{C}$ outputs cs_i.

-Challenging response: At this stage, challenger $\mathcal{C}$ randomly selects a cs from CS set and generates the corresponding attestation σ_{PBA} as the query on adversary $\mathcal{A}$. At this time, adversary $\mathcal{A}$ needs to output the index j as the response. If $cs_j = cs$, then the query is successful, otherwise fails.

Definition 1(Configuration Privacy). Let $Adv[\mathcal{A}_{PBA}^{anon}] = |Pr[\mathcal{A}\ wins] - 1/n|$ denotes that the advantage of the adversary $\mathcal{A}$ wins the above game. If $Adv[\mathcal{A}_{PBA}^{anon}]$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$, then the PBA scheme meets configuration privacy.

(3) unforgeability This PBA scheme is unforgeable, that is, no adversary can win the following games in polynomial time.

-initialization: Challenger $\mathcal{C}$ runs *Setup*(1^k) and the adversary $\mathcal{A}$ only knows the public key ppk.

-Queries: Adversary $\mathcal{A}$ adaptively query challenger $\mathcal{C}$ in the following manner:

Join: Adversary $\mathcal{A}$ sends the i-th Join request to challenger $\mathcal{C}$, challenger $\mathcal{C}$ selects a attestor's platform configuration information $cs_i (i \in \{1, \dots, q-1\})$ to run the join algorithm to create a certificate cre_i bout the property ps_i for the attestor, and sends cre_i to adversary $\mathcal{A}$;

Attest Query: This query is divided into two cases. The first is that adversary $\mathcal{A}$ issues the i-th Attest query, challenger C runs Attest algorithm to generate the unblinded

attestation $S_i$and returns it to $\mathcal{A}$; The second is that when $i = i^*$, challenger $\mathcal{C}$ will run the Attest algorithm to generate an attestation s*, and use it as a response to the $\mathcal{A}$.

Corrupt Query: adversary $\mathcal{A}$ sends the i-th Corrupt request to challenger $\mathcal{C}$. $i \neq i^*$. challenger $\mathcal{C}$ will respond to $\mathcal{A}$ with cs_i corresponding to the index i. challenger C will not respond when $i = i^*$.

-Forgery: Adversary $\mathcal{A}$ outputs attestor's property-based attestation $\mathcal{S}$ and a challenge value N_v. If Verify$(N_v, S, ppk,$RL$)$=1(ACCEPT) and adversary $\mathcal{A}$ has not made a Corrupt query to cs corresponding to attestation $\mathcal{S}$, the attack is successful, otherwise failed.

Definition 2(Unforgeability**).** Adversary $\mathcal{A}$ as the adversary in the above-mentioned attack game, and use $Adv\left[\mathcal{A}_{PBA}^{unforgery}\right] = Pr[\,\mathcal{A}\ wins]$ to represent the advantage of $\mathcal{A}$ against the above-mentioned unforgeable game. If adv $Adv\left[\mathcal{A}_{PBA}^{unforgery}\right]$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$, then the PBA scheme is said to have proved unforgeability.

3 Our Scheme

3.1 Setup Algorithm

(1). Input the secure parameter 1^k to generate a bilinear cyclic group G of order n, where$n = p \cdot q$, p and q are prime numbers, G_p and G_q are subgroups of group G, and bilinear map is e: $G \times G \rightarrow G_T$. Select generator g and h from G and G_p respectively;

(2). Randomly select a number γ from $\mathbb{Z}_n$ and calculate $\omega := g^{\gamma}$ and $T := e(g, g)$;

(3). Selecting l generators $\tau_1, \ldots, \tau_l$and u, τ' from group G;

(4). Output PBA public key ppk and TTP private key tsk:

$(ppk, tsk) := \left((G, G_T, n, e, g, h, u, \tau', \tau_1, \ldots \tau_l, \omega, T), \gamma\right)$

3.2 Join Algorithm

(1). TPM collects the platform configuration information cs, then sends cs to issuer I through a secure channel and asks for an property certificate. When issuer I receive TPM's platform configuration information verifier $\mathcal{V}$, issuer I evaluate the property of cs. if the evaluated property are ps, then issuer I issues an property certificate $cre := (g^{cs}, (g^{ps})^{\frac{1}{\gamma+cs}}, u^{cs})$) and sends the property certificate and $(cs, ps)$to TPM, then issuer I restore cre and (cs, ps) into the configuration-property certificate database, which is convenient for later verification and query.

(2). TPM receives the property certificate cre and (cs, ps) and checks them as follows: let $U_1 := g^{cs}$, $U_2 := (g^{ps})^{\frac{1}{\gamma+cs}}$, $U_3 := u^{cs}$, and check whether $e(U_1 \cdot \omega, U_2) = T^{ps}$ and $e(U_1, u) = e(U_3, g)$ are valid. If it passes the check, TPM saves the property certificate cre.

3.3 Attest Algorithm

(1). Verifier $\mathcal{V}$ query TPM with a challenge value $N_v = (m_1 \cdots m_l) \in \{0,1\}^l$;

(2). After TPM obtains the challenge value N_v, it randomly selects $r \leftarrow \mathbb{Z}_n$, calculates the commitment value $C_{cs} = g^{cs} \cdot h^r$ to cs, and generates an anonymous authentication signature (The signature algorithm uses AIK signature method in TCG standard, the corresponding private key is sk_M) $\sigma_M := \text{Sign}(sk_M, C_{cs} \parallel N_v)$;

(3). TPM sends $(\sigma_M, C_{cs}, U_2, U_3, r, N_v)$ to host;

(4). Host randomly selects $t \leftarrow \mathbb{Z}_n$ and computes:

$S := (S_1, S_2, S_3, S_4) = (C_{cs}, U_2, U_3 \cdot (\tau' \cdot \prod_{i=1}^{l} \tau_i^{m_i})^t, g^{-t})$

(5). To ensure privacy, host $\mathcal{H}$ needs to re-randomize attestation $\mathcal{S}$

Select $r_1, r_2, r_3 \leftarrow \mathbb{Z}_n$, and compute:

$\Omega := (\Omega_1, \Omega_2, \Omega_3, \Omega_4) = (S_1, S_2 \cdot h^{r_1}, S_3 \cdot h^{r_2}, S_4 \cdot h^{r_3})$

(6). Host computes the corresponding proof as follows:

$\pi_1 := h^{r \cdot r_1} \cdot (\Omega_1 \cdot \omega)^{r_1} \cdot (\Omega_2)^r$

$\pi_2 := u^r \cdot g^{-r_2} \cdot (\tau' \cdot \prod_{i=1}^{l} \tau_i^{m_i})^{-r_3}$

(7). Host $\mathcal{H}$ sends $\sigma_{PBA} := (\sigma_M, \Omega_1, \Omega_2, \Omega_3, \Omega_4, \pi_1, \pi_2)$ to verifier.

3.4 Verify Algorithm

(1). After receiving σ_{PBA}, the verifier checks whether $\Omega_1, \Omega_2, \Omega_3, \Omega_4, \pi_1, \pi_2$ is belong to G and verifies the validity of σ_M;

(2). Verifier $\mathcal{V}$ computes and checks:

$$e(\Omega_1 \cdot \omega, \Omega_2) \cdot T^{-ps} = e(h, \pi_1)$$

$$e(\Omega_1, u) \cdot e(\Omega_3, g)^{-1} \cdot e(\Omega_4, \tau' \cdot \prod_{i=1}^{l} \tau_i^{m_i})^{-1} = e(h, \pi_2)$$

(3). If the above equations are hold, Verifier $\mathcal{V}$ sends(Ω_1, ps) to issuer $\mathcal{I}$ to judge whether the platform configuration-property pair (*cs,ps*) is in the revocation list;

(4). If all the checks pass, then Verifier $\mathcal{V}$ outputs accept.

3.5 Check Algorithm

On receiving a request (Ω_1, ps) from verifier, issuer should check whether the platform configuration information cs about(Ω_1, ps) is in the revocation list. The issuer computes $\rho := (\Omega_1)^p = g^{cs \cdot p}$, then research the certificate database with ρ and ps to check whether $\rho^{\frac{1}{p}}$ is equal to a certain g^{cs^*} value, where $cs * \in CS$(*CS* is the set of platform configuration specification). If the corresponding certificates exists, issuer inform Verifier $\mathcal{V}$ that the property certificate for platform configuration information cs is valid; Otherwise, the platform configuration information cs was revoked.

4 Security and Performance Analysis

4.1 Security Proof

In this section, under the standard model we prove the security of our scheme. An PBA protocol must satisfy the following security properties: correctness, unforgeability and configuration privacy. According to the security model in the section 2, the following theorems is proved.

Theorem 1 (Correctness) The PBA scheme proposed in Section 3 is correct.

Proof: To prove the correctness of the proposed PBA scheme, it is necessary to prove that the signature generated by the valid signer can be successfully verified by any verifier.

$$
\begin{aligned}
&\mathrm{e}(\Omega_1 \cdot \omega, \Omega_2) \cdot T^{-ps} \\
&= e(g^{cs} \cdot h^r \cdot g^{\gamma}, (g^{ps})^{\frac{1}{(cs+\gamma)}} \cdot h^{r_1}) \cdot e(g,g)^{-ps} \\
&= e(g^{cs+\gamma}, (g^{ps})^{\frac{1}{(cs+\gamma)}}) \cdot e(g^{cs+\gamma}, h^{r_1}) \cdot e(h^r, \left(g^{ps}\right)^{\frac{1}{(cs+\gamma)}}) \cdot e(h^r, h^{\gamma_1}) \\
&= e(g,g)^{ps} \cdot e(h, (g^{cs+\gamma})^{r_1}) \cdot e\left(\left(g^{ps}\right)^{\frac{1}{(cs+\gamma)}}\right)^{\gamma}) e(h, h^{\gamma \cdot r_1}) \cdot e(g,g)^{-ps} \\
&= \mathrm{e}(\mathrm{h}, h^{\gamma \cdot r_1} \cdot (\Omega_1 \cdot \omega)^{r_1} \cdot (\Omega_2)^{\Upsilon}) \\
&= e(h, \pi_1) \\
&\mathrm{e}(\Omega_1, u) \cdot e(\Omega_3, g)^{-1} \cdot e\left(\Omega_4, \tau' \cdot \pi_{i=1}^{\tau} \tau_i^{m_i}\right)^{-1} \\
&= \mathrm{e}(g^{cs} \cdot h^r, u) \cdot \mathrm{e}\left(u^{cs} \cdot h^{r_2} \cdot \left(\tau' \cdot \pi_{i=1}^{\tau} \tau_i^{m_i}\right)^t, g\right)^{-1} \cdot e\left(g^{-t} \cdot h^{r_3}, \tau' \cdot \pi_{i=1}^{\tau} \cdot \tau_i^{m_i}\right)^{-1} \\
&= \mathrm{e}\left(h, u^r \cdot g^{-r_2} \cdot \left(\tau' \cdot \pi_{i=1}^{\tau} \tau_i^{m_i}\right)^{-r_3}\right) \\
&= \mathrm{e}(h, \pi_2)
\end{aligned}
$$

Theorem 2 (Unforgeability) Based on *q*-HSDH assumption, PBA protocol has the attestation unforgeability. Under adaptive choosen message attack, if an adversary $\mathcal{A}$ can forge a valid attestation with non-negligible probability in probability polynomial

time, then there is an algorithm attestation $\mathcal{S}$ that can solve q-HSDH assumption with non-negligible probability in probability polynomial time.

Proof: The idea of proof here is based on papers [12,13,14]. In addition, since TPM is physically secure, adversary $\mathcal{A}$ can only control the behavior of host $\mathcal{H}$ for attestor $\mathcal{P}$. Assuming that an Adversary $\mathcal{A}$ can forge an unblinded PBA with non-negligible probability, then a polynomial time simulator attestation $\mathcal{S}$ can be constructed to solve the q-HSDH problem through interaction with adversary $\mathcal{A}$. It is worth noting that if an adversary can forge an unblinded PBA, it can also forge σ_{PBA}. First, an example of attestation $\mathcal{S}$ is given: $g, u \in G$, $h \in G_p$, $w = g^{\gamma}$ and $q-1$ $(A_i = g^{\frac{1}{\gamma+cs_i}}, B_i = g^{cs_i}, C_i = u^{cs_i})_{i=1,\dots,q-1}$ q-HSDH example, where γ value is unknown. The interaction process between adversary $\mathcal{A}$ and attestation $\mathcal{S}$ is as follows:

Setup. Attestation $\mathcal{S}$ executes the Setup(1^k) algorithm as follows: first, attestation $\mathcal{S}$ selects random numbers $\mu \in \mathbb{Z}_l, t \in \mathbb{Z}_n$and a series of random numbers $(x', x_1, \dots, x_l) \in \mathbb{Z}_{2q-1}^{l+1}$; Then, S randomly selects $(z', z_1, \dots, z_l) \in \mathbb{Z}_n^{l+1}$, so that $(v' = g^{z'}, v_1 = g^{z_1}, \dots, v_l = g^{z_l}) \in G$ for ease of analysis, the following three parameters are defined: $X = -2\mu q + x' + \Sigma_{i=1}^{l} x_i m_i, Y = z' + \Sigma_{i=1}^{l} z_i m_i, Z = \tau' \prod_{j=1}^{l} \tau_j^{m_j}$.

Attestation $\mathcal{S}$ constructs PBA system parameters as follows: $f = \omega^{-1} g^t$, $\tau' = f^{x'-2kl} v''$, $\tau_1 = f^{x_1} v_1$, ..., $\tau_l = f^{x_l} v_l$ and $(g, \omega = g^{\gamma}, h, u, \tau', \tau_1, \dots \tau_l, T)$ as PBA system parameters; Finally, attestation $\mathcal{S}$ maintains a list that records the results of the queries and maps the query results regarding platform configuration information cs_i with *indexes* $i \in \{1, \dots, q-1\}$.

Join Queries. When adversary $\mathcal{A}$ applies for the property certificate to $\mathcal{S}$ for the first time, there are two situations to consider:

When $i \neq i^*$, s selects $cre_i = \left(A_i^{ps}, B_i, C_i\right)$ as a response from the q-HSDH instance, saves the certificates $cre_i$and ps in the i item in the list, and uses this as a return value; if the i times query has occurred before, then S will take the content corresponding to the i item in the list as a response to adversary $\mathcal{A}$.

When $i = i^*$, attestation $\mathcal{S}$ will not respond and terminate because s does not know the value of $cs^* = |t - \gamma|$.

Attest Queries. When adversary $\mathcal{A}$ asks $\mathcal{S}$ for a query, there are two cases:

When $i \neq i^*$, in order to answer adversary $\mathcal{A}$'s query i , attestation $\mathcal{S}$ will do the following. If this i times queries has been conducted before, then attestation $\mathcal{S}$ selects the contents of the i table item from the list and uses cre_i as the return value; If the i times query has not occurred before, then attestation $\mathcal{S}$ selects the number i from q-HSDH instances, takes $cre_i = \left(A_i^{ps}, B_i, C_i\right)$ and ps_i as responses, and records them in the list;

When $i = i^*$, Attestation $\mathcal{S}$ randomly selects a $r \leftarrow \mathbb{Z}_n$ and calculates $S = (S_1, S_2, S_3, S_4) = \left(g^{\frac{ps}{t}}, \omega^{-1} g^t, u^{-\frac{Y}{X}} \cdot Z^r, u^{\frac{1}{X}} g^{-r}\right)$ if $X \equiv 0 (mod\, n)$, S terminates and exits.

Because $cs^* = t - \gamma$, Therefore, $r' = r - \log_g^u X$, $S_3 = u^{-\frac{Y}{X}} \cdot Z^r = u^{-\frac{Y}{X}} \cdot (f^X g^Y)^r = u^{-\frac{Y}{X}} \cdot (f^X g^Y)^{r'} \cdot f^{\log_g^u} \cdot u^{\frac{Y}{X}} = u^{cs^*} \cdot Z^{r'}$ is constructed for the above formula, which is similar to $S_4 = g^{-r'}$, This certificate is:
$S = \left(g^{\frac{ps}{\gamma + cs^*}}, g^{cs^*}, u^{cs^*} \cdot Z^{r'}, g^{-r'}\right)$.It and the corresponding ps^* are taken as the response value to the adversary $\mathcal{A}$ query.

Corrupt Queries. Adversary $\mathcal{A}$ conducts the first time query, where $i \neq i^*$, attestation $\mathcal{S}$ will look up the i-th item from the list and will take cs_i as the return value.

Forgery. adversary $\mathcal{A}$ passed *q-1* inquiries and finally output attestation $S^* = (S_1^*, S_2^*, S_3^*, S_4^*)$.if $S_1^* \neq f$, then attestation $\mathcal{S}$ will terminate, otherwise calculate $X^* = -2\mu q + x' + \Sigma_{i=1}^{l} x_i m_i^*$ and $Y^* = z' + \Sigma_{i=1}^{l} z_i m_i^*$, where $N_v^* = (m_1^* \cdots m_l^*) \in \{0,1\}^l$.If $X^* \neq 0(mod\, n)$ then attestation $\mathcal{S}$ will terminate, because what is obtained is an invalid forgery. If all the conditions are met, the final simulator attestation $\mathcal{S}$ output $\left(S_1^{\frac{1}{ps}}, S_2, S_3 \cdot S_4^{Y^*}\right)$) is taken as the output to the *q*-HSDH problem.

The above proof process describes the simulation process of attestation $\mathcal{S}$, and the success probability of attestation $\mathcal{S}$ is analyzed below. Because the entire simulation algorithm needs to be run completely to solve the *q*-HSDH problem, simulator attestation $\mathcal{S}$ cannot be terminated during the query. According to the above algorithm, three conditions must be met for attestation $\mathcal{S}$ not to terminate: $\Omega_1^* = f$, this probability is $\frac{1}{q-1}$; Secondly, the probability of $X \neq 0(mod\, n)$ is at least 1-1/2q for each attest query of $i = i^*$. if there are at least *q-1* query, then the total probability should be greater than 1/2; Finally, in the forgery stage, the probability of $X^* \equiv 0(mod\, n)$ should be at least 1/2lq. If adversary $\mathcal{A}$ successfully forges the proof in polynomial time with the probability of ε, then the probability of success of attestation $\mathcal{S}$ is $Adv[\mathcal{A}_{PBA}^{unforgery}] \geq \frac{\varepsilon}{4lq(q-1)} \geq \frac{\varepsilon}{4lq^2}$, that is, the problem of *q*-HSDH is solved with the advantage of $Adv[\mathcal{A}_{PBA}^{unforgery}]$ in polynomial time, which contradicts the assumption of *q*-HSDH. Therefore, this PBA scheme has the unforgeability of attestation.

Theorem 3 (Configuration Privacy) Based on SDA assumption, PBA protocol has the property of configuration privacy. Assuming that no polynomial time algorithm can solve the SDA assumption with a probability ε, then for each polynomial time adversary $\mathcal{A}$ there is adv $Adv[\mathcal{A}_{PBA}^{anon}] < 2\varepsilon$.

Proof: in order to prove adv $Adv[\mathcal{A}_{PBA}^{anon}] < 2\varepsilon$, then according to SDA assumption, we first need to prove that the two games when host $\mathcal{H}$ belongs to group G_p or group G, represented Υ_0 and Υ_1 respectively, are indistinguishable from adversary $\mathcal{A}$, namely adv $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_0} - Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_1} < \varepsilon'$ is a negligible value.

In the initialization phase of the game, Attestation $\mathcal{S}$ will receive a subgroup decision assumption instance (n, G, G_T, e, h). As mentioned above, there are two cases for h: case 1. Host $\mathcal{H}$ belongs to group G_p, and then the game is a normal configuration privacy game, marked as Υ_0; case 2. Host $\mathcal{H}$ belongs to group G, then the game is recorded as Υ_1. The remaining parameters of the two games are the same. Then, adversary A and simulator attestation $\mathcal{S}$ play the configuration privacy game described in section 3.4.

In the query phase, adversary $\mathcal{A}$ responds to the query of attestation $\mathcal{S}$, i.e. answers an index j. if the answer is correct, i.e. $cs_j = cs$ then S outputs 1, indicating $h \in G_P$; Otherwise, attestation $\mathcal{S}$ outputs 0, indicating $h \in G$. $Adv[\mathcal{S}_{PBA}^{anon}]$ is used to represent the advantage of simulator attestation $\mathcal{S}$ in subgroup decision game, and $Pr[h \in G_p] = Pr[h \in G] =$ is known, then there is:

$Adv[\mathcal{A}_{PBA}^{anon}]_0 - Adv[\mathcal{A}_{PBA}^{anon}]_1$

$= 2Adv[\mathcal{S}_{PBA}^{anon}] < 2e$

Next, it needs to be further proved that adv $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_1} = 0$, that is to say, when $h \in G$, the query values σ_{PBA} and cs are statistically independent of each other. The proof process is as follows:

In the query phase, given a query value $\sigma_{PBA} = (\sigma_M, \Omega_1, \Omega_2, \Omega_3, \Omega_4, \pi_1, \pi_2)$ (where σ_M is TPM authentication signature, and its security is beyond the scope of this article), it

needs to be proved that σ_{PBA} can match any assumed value $c\tilde{s}$ that the adversary may adopt, that is, σ_{PBA} will not disclose any information about *cs*.

Under the condition of statistical independence, it is necessary to define an adversary $\tilde{A}$ with infinite computing power and obtain the following discrete logarithm:

First of all, for the four commitment values $(\Omega_1, \Omega_2, \Omega_3, \Omega_4)$, they do not disclose any information about *cs* because they are perfectly blinded by four uniformly distributed and independent random values $h^r, h^{r_1}, h^{r_2}, h^{r_3}$. Where Ω_1 and Ω_2 are directly related to *cs*, so no matter the adversary judges *cs* as any assumed value $c\tilde{s}$, for cs̃, there is $\tilde{r}, \tilde{r}_1$ makes $\Omega_1 = g^{c\tilde{s}}h^{\tilde{r}}$, ω $\Omega_2 = g^{1/(\gamma+c\tilde{s})}h^{\tilde{r}_1}$, so blindness does not reveal any information aboutcs.

Secondly, for evidence π_1, which involves Ω_1 and Ω_2, it needs to be proved that the given evidence π_1 value is consistent with the evidence value constructed by the adversary based on the assumed value. Let cs̃, $\tilde{r}, \tilde{r}_1$ be the values assumed by adversary $\tilde{A}$, then there are :$\begin{cases}\omega\Omega_1 = g^{\gamma+cs}h^r = g^{\gamma+cs+\beta\cdot r} \\ \omega\Omega_1 = g^{\gamma+c\tilde{s}}h^{\tilde{r}} = g^{\gamma+c\tilde{s}+\beta\cdot\tilde{r}}\end{cases}$, the two equations are combined to obtain $\tilde{r} = r + (1-\xi)\cdot(\gamma+cs)/\xi\cdot\beta(mod\, n)$.

$\begin{cases}\Omega_2 = g^{ps/(\gamma+cs)}h^{r_1} = g^{ps/(\gamma+cs)+\beta r_1} \\ \Omega_2 = g^{ps/(\gamma+c\tilde{s})}h^{\tilde{r}_1} = g^{(ps/(\gamma+cs))*((\gamma+cs)/(\gamma+c\tilde{s}))+\beta\tilde{r}_1}\end{cases}$

The two equations are combined to obtain $\tilde{r}_1 = r_1 + ps(1-\xi)/\beta(\gamma+cs)$For the evidence $\tilde{\pi}_1 = h^{\tilde{r}\cdot\tilde{r}_1}(g^{c\tilde{s}+\gamma})^{\tilde{r}_1}(g^{ps/(c\tilde{s}+\gamma)})^{\tilde{r}}$constructed by adversary $\tilde{A}$, the above $\tilde{r}, \tilde{r}_1$ and ξ can be substituted into the formula to derive $\tilde{\pi}_1 = \pi_1$, which indicates that the equation holds no matter what the assumed value of adversary $\tilde{A}$ is, it is not helpful to exclude the value of $c\tilde{s}$, and further proves that π_1 does not disclose any information about *cs*.

The same proof method and conclusion apply to evidence π_2.

Through the analysis of $\sigma_{PBA} = (\sigma_M, \Omega_1, \Omega_2, \Omega_3, \Omega_4, \pi_1, \pi_2)$, it is proved that the query attestation values σ_{PBA} and cs are statistically independent of each other, so adversary $\tilde{A}$ has an advantage of 0 in the Υ_1 game, namely $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_1} = 0$

In summary, according to SDA assumption, $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_0} - Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_1} < 2\varepsilon$ holds, and $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_1} = 0$ is obtained according to analysis of various parameters of σ_{PBA} in game Υ_1, thus $Adv[\mathcal{A}_{PBA}^{anon}]_{\Upsilon_0} < 2\varepsilon$ holds, which is proved.

4.2 Performance Analysis

In this section, the scheme proposed in this paper is compared with the existing PBA scheme based on bilinear pairings [6] called PBA-BM scheme. It is worth noting that this paper does not compare with other existing schemes [4,5], because these schemes are not as efficient as PBA-BM schemes based on bilinear pairing.

First, we compare the size of certificate and attestation value. The following parameters are defined: $\mathbb{Z}_n$ denotes the size of the element in $\mathbb{Z}_n$, $\hbar$ denotes the size of the HASH value, G denotes the size of the element in group G, G_T denotes the size of the element in group G_T, and σ_M denotes the size of $|\sigma_M|$. For bilinear mapping satisfying 128-bit security, G_T needs to be about 3072bit [17]. The comparison results are shown in Table

1.The size of certificates and certificates of this scheme are smaller than that of PBA-BM scheme, i.e. the communication cost of this scheme is smaller than that of PBA-BM scheme.

Table 1. Comparison on communication cost

PBA scheme	Certificate(cre) size	Attestation(σ_{PBA}) size
PBA-BM	$2h + 5G$	$\sigma_M + \hbar + 6G + 5\mathbb{Z}_n$
This scheme	$2h + 3G$	$\sigma_M + 6G$

Secondly, we compare the efficiency of the two schemes in proving algorithm, checking algorithm and revocation algorithm. Define the following parameters: p represents a pairing operation, G represents an exponential operation in group G, G^k represents a k times exponential operation, and G_T represents an exponential operation in groupG_T. It is worth noting that a multiple exponential operation is slightly more efficient than an exponential operation [18], and the exponential operation in group g is much more efficient than the exponential operation in group G_T. The comparison results are shown in Table 2.

Table 2. Comparison on computation cost

PBA algorithm	PBA-BM scheme[]	our scheme
Attest(TPM)	$G + G^2$	G^2
Attest(Host)	$8G + G^2 + G^3 + 4G_T$	$3G^2 + G^3 + 2G^4$
Verify	$2G^2 + G^3 + G_T^4 + 4P$	$G_T + G$
Check	G^2	G^2

Because TPM is much less efficient than host, the Attest algorithm is divided into two parts for comparison. Compared with PBA-BM scheme, the Attest algorithm of our scheme is more efficient on both TPM and host. Compared with Verify algorithm, the efficiency of our scheme is better than PBA-BM scheme, while the execution efficiency of Check algorithm is the same. Therefore, our scheme is better than PBA-BM scheme.

5 summary

The proposed property-based attestation scheme solves the problem of privacy leakage caused by the original binary attestation scheme and enhances the configuration privacy of the platform. Compared with the existing schemes on performance, our scheme is better than PBA-BM scheme and other PBA schemes. On the other hand, in terms of security, since the security of the existing property-based attestation schemes is proved under the random oracle model, our scheme can be proved under the standard model based on SDH assumption and SDA assumption, the security is better than other existing schemes. In the future work, the scheme will continue to be further improved to make it more efficient and practical.

6 Reference

1. Trusted Computing Group. TPM Main Part 1, Design Principles Specification, Version 1.2 Revision 62[EB/OL]. [2003-10-2]. https://www.trustedcomputinggroup.org/home
2. Jaeger, T., Sailer, R., Shankar, U.: PRIMA: policy-reduced integrity measurement architecture[C].Proc of the 11th ACM Symposium on Access Control Models and Technologies, New York, 2006: 19–28
3. Sadeghi A, Stuble C. Property-based attestation for computing platforms: caring about properties, not mechanisms [C].Proc of the 2004 Workshop on New Security Paradigms. Nova Scotia: ACM, 2004: 67–77
4. Chen Liqun, Landfermann R, Lohr H, et al. A protocol for property-based attestation[C].Proc of the first ACM workshop on Scalable trusted computing. New York: ACM, 2006: 7–16
5. Chen LiQun, Lohr H, Manulis M, et al. Property-based attestation without a trusted third party[G]. LNCS 5222: Proc of the 11th International Conference on Information Security. Berlin: Springer-Verlag, 2008: 31–46
6. Feng Dengguo, Qin Yu. A property-based attestation protocol for TCM[J]. Science China (Information Sciences), 2010, 53(3): 454-464
7. Bellare M, Rogoway P. Random oracles are practical: A paradigm for designing efficient protocols[C] .Proc of the First Conference on Computer and Communications Security. New York: ACM, 1993: 62-73
8. Bellare M, Boldyreva A, Palacio A. A uninstantiable random oracle-model scheme for a hybrid-encryption problem[G]. LNCS 3027: Conf of EUROCRYPT'2004, Berlin: Springer, 2004: 171-188

9. Canetti R, Goldreich O, Halevi S. The random oracle methodology, Revisited (preliminary version)[C].Proc of the 30th Annual ACM Symposium on the Theory of Computing. STOC'98. New York: ACM, 1998: 209-218
10. David C, Eugene H. Group Signatures [G].LNCS 547: Advances in Cryptology. EUROCRYPT'1991, Berlin: Springer, 1991: 57–65
11. Mihir B, Daniele M, Bogdan W. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions [G].LNCS 2656: Advances in Cryptology, EUROCRYPT'2003, Berlin: Springer, 2003: 14–29
12. Dan B, Xavier B, Hovav S. Short group signatures[G]. LNCS 3152: . Advances in Cryptology, CRYPTO'2004, Berlin: Springer, 2004: 41-55
13. Xavier B, Brent W. Compact group signatures without random oracles[G]. LNCS 4004: Advances in Cryptology, EUROCRYPT'2006, Berlin: Springer, 2006:427-444
14. Xavier B，Brent W. Full-domain subgroup hiding and constant-size group signatures[G]. LNCS 4450: Proc of PKC 2007, Berlin: Springer, 2007: 1-15
15. Jens G , Amit S. Efficient non-interactive proof systems for bilinear groups [G]. LNCS 4965: Advances in Cryptology EUROCRYPT' 2008, Berlin: Springer, 2008: 415-432
16. Jens G. Simulation-sound NIZK proofs for a practical language and constant size group signatures [C]. Proc of ASIACRYPT' 2006, Shanghai, 2006: 444-459.
17. Neal K, Alfred M. Pairing-based cryptography at high security levels [G]. LNCS 3796: Proc of the 10th IMA International Conference on Cryptography and Coding, Berlin: Springer, 2005:13-36
18. Mao Wenbo. Modern Cryptography: Theory and Practice [M]. New Jersey: Prentice Hall Press, 2003
19. Awad A , Kadry S , Lee B , et al. [IEEE 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC) - London, United Kingdom (2014.12.8-2014.12.11)] 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing - Property Based Attestation for a Secure Cloud Monitoring System[J]. 2014:934-940.
20. Qin Y , Feng D G . Component Property Based Remote Attestation[J]. Journal of Software, 2009, 20(6):1625-1641.