SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

March 2010 Vol. 53 No. 3: 454–464
doi: 10.1007/s11432-010-0057-1

A property-based attestation protocol for TCM

FENG DengGuo[1,2*] & QIN Yu[1,2]

[1]*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Science, Beijing* 100190*, China;*
[2]*National Engineering Research Center of Information Security, Beijing* 100190*, China*

Received June 30, 2009; accepted February 1, 2010; published online March 1, 2010

Abstract This paper presents a property attestation protocol for the security chip TCM (trusted cryptographic module) via analyzing the problems of the current property attestation, which is built on the property attestation model with the online trust third party. In the protocol the prover utilizes the zero-knowledge proof by the attribute certificates, configuration commitment and TCM signature, and attests its configuration and status which are compliant with the declarative security property. The protocol is characterized by shorter signature length and lower computations. The security of the protocol is proved at the random oracle model. The protocol can help extend application and improve standard for security chip TCM, and it also has practical value and immediate significance.

Keywords trust computing, trust cryptographic module (TCM), property attestation, signature of knowledge, configuration commitment

Citation Feng D G, Qin Y. A property-based attestation protocol for TCM. Sci China Inf Sci, 2010, 53: 454–464, doi: 10.1007/s11432-010-0057-1

1 Introduction

The security requirements for the platform and the data become higher and higher in the current distributed applications, particularly in internet open network environment. The different participants have different security requirements for their own interest, so it is absolutely critical to provide a security mechanism for multi-side security in the distributed environment. Trusted computing technology develops rapidly under this background, which leverages the hardware and the software to enhance the security from the computer architecture. TCG (trusted computing group) which consists of many international IT enterprises makes a series of trusted computing specifications [1, 2] with the security chip TPM (trust platform module) [3] at the core. TPM is a tamper-resistant chip which provides cryptographic operations, integrity measurement, storage protection, remote attestation, etc. TPM and its supporting software TSS (trust software stack) are the foundation for trusted computing technology. Now many PCs, notebooks, servers and even mobile phones, are equipped with TPM security chip providing stronger security.

The integrity measurement, sealing storage, remote attestation, etc. are TPM's feature function. Remote attestation is one of the important security mechanisms for building multi-side security in distributed environment, and it is used to establish the trust between different participants by mutually attesting its

*Corresponding author (email: feng@is.iscas.ac.cn)

platform configuration. Many research centers at home and broad have made deepgoing study on this problem, and presented a lot of remote attestation methods. The study covers the direct binary attestation complying with TCG specification [4, 5], the semantic attestation based on high level language [6], the software attestation on the embedding device [7], and the web service attestation [8]. Remote attestation is applied widely, and it has a variety of attestation approaches. Among all the attestation approaches the property-based attestation is the one growing fast and having many uses. It overcomes the drawbacks of attestation complexity, privacy leaking, and attestation discrimination. The early study on the property-based attestation was focused on the attestation model and architecture. Sadeghi et al. [9] of Ruhr-University Bochum presented property-based attestation first in 2004. Then IBM research center gave the attestation architecture [10]. Chen et al. [11] proposed the first property-based attestation protocol in 2006 (PBA protocol for short). Another property attestation protocol (PBA-RS protocol [12] for short) was proposed without the trust third party issuing property certificates; it employed the ring signature to prove that the target platform's configuration and status satisfy the verifier's requirements. Without the trust third party, it also protected the platform configuration privacy. The conventional method of the property attestation uses the zero knowledge proof to attest platform configuration commitment on the property certificate. But the above protocols rely on the offline trust third party, and they have the revocation complexity of the properties and low performance. Using the property attestation, Ruhr-University Bochum realized the property transformation of binary measurement at bootstrap by online trust third party [13] as well as the system attestation and sealing. Based on the trust third party, the scheme easily supports the integrity management and properties revocation by CRL (certificate revocation list), but it lacks a practical protocol to protect the platform configuration privacy in the scheme.

In the mean time, China has been developing her own trusted computing technology and industry. China State Password Administration Committee has published Functionality and Interface Specification of Cryptographic Supporting Platform for Trusted Computing [14] in December, 2007. It marks a new stage for Chinese trusted computing technology, products and standards with our self-owned intellectual property. Today China has successfully developed TCM (trust cryptographic module), TSM (TCM service module), security PC, etc. The security chip TCM has similar function to TPM, but supports China's own cryptographic algorithms, especially supports symmetric algorithm and asymmetric elliptic curve algorithm. Though TCM and its service software are developing fast in China, the study on sealing storage, remote attestation, etc. drops behind. Therefore it is necessary to speed up the research on these key technologies, especially research on the property-based attestation to promote and improve trusted computing standards.

We present a new property-based attestation protocol on bilinear map (PBA-BM for short) against the relative problems, which makes full use of the TCM's cryptographic feature. Our attestation protocol is built on the integrity management and the online trust third party. It relies on the security of configuration commitment and CL-LRSW signature, ensures the attestation authenticity and configuration privacy, and proves secure in random oracle model. PBA-BM uses ECC and bilinear map to attest the platform configuration commitment for the verifier's security requirement. It simplifies the verification of the properties revocation, and has shorter signature length and far less computation cost. The research helps extend the TCM application fields, and advances TCM application in high security field.

The paper is organized as follows. Section 2 describes the property attestation model abstractly including security properties and attestation processes. Section 3 introduces the relative cryptographic preliminaries for our attestation protocol. Section 4 illustrates the attestation protocol in detail, analyzes its security and proves it. The protocol performance is compared with other schemes in section 5. The conclusion is summarized in the last section.

2 Property attestation model

The property-based attestation reduces the computation on attesting and verifying phases. The security chip TCM attests the security property towards the remote verifier which the platform configuration

satisfies. In property attestation model, there are three parties involved: Prover $\mathcal{P}$ (host $\mathcal{H}$ and security chip $\mathcal{M}$), Verifier $\mathcal{V}$, Property Authority $\mathcal{T}$. The prover attests whether platform configuration and property certificate are consistent. Assuming the security chip $\mathcal{M}$ is honest and tamper-resistant in the prover's platform. The verifier determines whether the attestation meets the security requirement. The property authority $\mathcal{T}$ as an online trust third party issues the platform's property certificate, and check the validation of the certificate. The property authority and TCM are completely trusted in the model. There are two kinds of the potential attacks in the model: attestation forgery and observation on configuration privacy. The corrupted host $\mathcal{H}$ makes the attestation forgery to convince the remote verifier that the platform is in trust by interpreting and forging the attestation messages. The malicious verifier $\mathcal{V}$ may observe the configuration privacy of attested platform by analyzing the attestation proof, and then make the corresponding attack to obtain the benefit against the platform holes. The attestation processes are illustrated in property attestation model figure (see Figure 1).

The property attestation has the following phases in the model.

1. Setup. Set up the parameters of the property attestation system. Generate the keypair for the participants $\mathcal{M}$ and $\mathcal{T}$ with the algorithm $G(1^k) \rightarrow (sk, vk)$. (Msk, Mvk) is the keypair made up for TCM, (sk, vk) is for $\mathcal{T}$.

2. Issue. $\mathcal{P}$ requests $\mathcal{T}$ to generate the property certificate for the current configuration cs, $\mathcal{T}$ evaluates the configuration cs to output the security property ps. $\mathcal{T}$ signs the pair (cs, ps) with its private key sk, and it is also the issuing process of the property certificate $\mathfrak{cre}$. Next $\mathcal{T}$ sends back the certificate to $\mathcal{P}$, and then $\mathcal{P}$ saves $\mathfrak{cre}$ for the later attestation.

3. Attest. Verifier $\mathcal{V}$ challenges the prover $\mathcal{P}$ with the random number N_v, and the prover attests whether the current configuration cs satisfies the same property ps. The security chip $\mathcal{M}$ first computes the commitment C on cs, and then signs the commitment C. $\mathcal{H}$ attests the platform property by the knowledge signature [15] with the commitment C and the certificate $\mathfrak{cre}$, and $\mathcal{P}$ outputs signature of property-based attestation σ_{PBA}, and sends σ_{PBA} to verifier for verification.

4. Verify. The verifier $\mathcal{V}$ receives the attestation data from $\mathcal{P}$, checks whether the random number N_v is fresh, then verifies the signature δ of security chip, as well as the knowledge signature on the commitment C and the certificate $\mathfrak{cre}$, and the verifier finally checks wether the property ps is revoked. If all of verifications are passed, $\mathcal{P}$ successfully attests its security property, otherwise fails.

There are several methods to check the property revocation for $\mathcal{V}$: the zero knowledge proof on $cs \notin CS_{\text{revoked}}$ is used in PBA scheme; the negotiation on the configuration set CS is used in PBA-RS scheme, which makes sure the configuration set is not revoked during the attestation; the verification of the online trust third party is used in our scheme for the property revocation (see Check phase).

5. Check. $\mathcal{V}$ forwards the property attestation data to $\mathcal{T}$. $\mathcal{T}$ decrypts (cs, ps) from the certificate $\mathfrak{cre}$ with its private key sk, then queries the property database to decide whether the pair (cs, ps) is revoked, and sends the checking result to $\mathcal{V}$ through the secure channel.

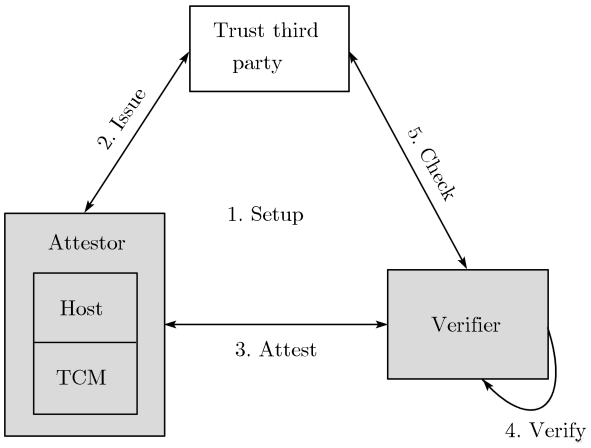

Figure 1 Property attestation model.

The property attestation for $\mathcal{P}$'s platform needs to meet the $\mathcal{V}$ security requirement on the condition that the both interest are protected between $\mathcal{P}$ and $\mathcal{V}$. The property attestation model must achieve the following two security attributes:

1) Unforgeability. Let $Game_{\mathcal{A}}^{att-fg}(1^k)$ be attacking interaction of the forged attestation among the multiple participants $\mathcal{P}$, $\mathcal{V}$, $\mathcal{T}$ and $\mathcal{A}$. $\mathcal{A}$ chooses a valid security chip $\mathcal{M}$, and attests the configuration property pair (cs, ps) towards $\mathcal{V}$, where $(cs, ps) \notin CS$, CS is the configuration property set accepted by $\mathcal{T}$. Assuming that $\mathcal{A}$ can intercept, modify, and forward any participant's messages. When all the honest participants carry out the protocol correctly, $\mathcal{A}$ sends the messages denoted by $send(E, m)$ $(E \in \{\mathcal{H}, \mathcal{M}, \mathcal{V}\})$, then queries the oracle $\mathcal{O}$ to attack the system, and finally $\mathcal{A}$ outputs the PBA signature σ_{PBA}. If σ_{PBA} is accepted by $\mathcal{V}$ without querying $\mathcal{O}$ on (cs, ps), $\mathcal{A}$ wins the game. Let $\mathrm{Adv}[\mathcal{A}_{PBA}^{att-fg}(1^k)] = \Pr[Game_{\mathcal{A}}^{att-fg}(1^k) = win]$ be the advantage probability that $\mathcal{A}$ wins $Game_{\mathcal{A}}^{att-fg}(1^k)$. If $\mathrm{Adv}[\mathcal{A}_{PBA}^{att-fg}(1^k)]$ is negligible on security parameter k, the property attestation is unforgeable.

2) Configuration privacy. Let $Game_{\mathcal{A}}^{cf-prv}(1^k)$ denote attacking interaction of breaking the configuration privacy among the multiple participants $\mathcal{P}$, $\mathcal{V}$, $\mathcal{T}$ and $\mathcal{A}$. In the n pairs (cs_i, ps) for the same property, $\mathcal{A}$ can compromise the configuration privacy of attestation with the advantage probability $\mathrm{Adv}[\mathcal{A}_{PBA}^{cf-prv}(1^k)] = |\Pr[b = j] - 1/n|$. If $\mathrm{Adv}[\mathcal{A}_{PBA}^{cf-prv}(1^k)]$ is negligible for the probability polynomial time adversary $\mathcal{A}$, we say that the property attestation meets the property of configuration privacy.

3 Cryptographic preliminaries

This section will introduce the cryptographic preliminaries later used in attestation protocol.

1) Bilinear maps. Our scheme uses the same bilinear map $e : G_1 \times G_2 \to G_T$ as in IBE scheme [16] and CL-LRSW signature scheme [17], where $G_i (i = 1, 2)$ and G_T denote the group of prime order q. The bilinear map e satisfies the following properties:

a) Bilinear. For all $P \in G_1$, $Q \in G_2$, any $a, b \in Z_q^*$, $e(aP, bQ) = e(P, Q)^{ab}$. For all $P_1, P_2 \in G_1$, $Q \in G_2$, $e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q)$.

b) Non-degenerate. There exists some $P \in G_1$, $Q \in G_2$ such that $e(aP, bQ) \neq I_{G_T}$, where I_{G_T} is the identity of group G_T.

c) Computable. There exists an efficient algorithm for computing $e(P, Q)$.

We use the symmetric pairing $(G_1 = G_2)$ in the attestation protocol, where G_1 and G_2 are the cycle group, because our protocol is built on CL-LRSW signature scheme by making use of symmetric pairing.

2) LRSW assumption [18]. Let $G = \langle g \rangle$ be a cycle group, $X, Y \in G$, $X = g^x$, $Y = g^y$. Suppose that there exists an oracle $\mathcal{O}$, on input $m \in Z_q$, and randomly choose $a \in G$, output (a, a^y, a^{x+mxy}). Then there exists no efficient adversary querying oracle $\mathcal{O}$ in polynomial times, and output (m, a, b, c) where m has not been queried before such that $m \neq 0$, $b = a^y$, $c = a^{x+mxy}$.

3) CL-LRSW signature. Our property attestation protocol is built on CL-LRSW signature scheme. Unlike the most signature scheme, it makes use of bilinear map, and efficiently proves the knowledge of signature for building cryptographic protocol. Due to the CL-LRSW signature scheme on message (m, r) used in this paper, we directly introduce the CL-LRSW signature on (m, r):

a) Key generation. Generate the keypair for signature scheme. Setup algorithm generates the public parameters (q, G, G_T, g, g_T, e), chooses $x \leftarrow_R Z_q$, $y \leftarrow_R Z_q$ and $z \leftarrow_R Z_q$ at random, and computes $X = g^x$, $Y = g^y$, $Z = g^z$. Set $sk = (x, y, z)$, $pk = (q, G, G_T, g, g_T, e, X, Y, Z)$.

b) Signature. On input message (m, r), private key sk and public key pk, randomly choose $a \in_R G$, compute $A = a^z$, $b = a^y$, $B = A^y$, $c = a^{x+xym} A^{xyr}$, and output signature $\sigma = (a, A, b, B, c)$.

c) Verification. On input pk, message (m, r) and signature σ, verify whether $e(a, Z) = e(g, A)$, $e(a, Y) = e(g, b)$, $e(A, Y) = e(g, B)$, $e(X, a) \cdot e(X, b)^m \cdot e(X, B)^r = e(g, c)$ hold.

Ref. [17] has proved that CL-LRSW signature scheme is secure under LRSW assumption against adaptively chosen message attack. In our protocol the signature scheme is employed to issue property certificate.

4) Commitment. We adopt Pedersen commitment scheme [19] in the property attestation. Let r be a random commitment key. Then the commitment is $y = g^m h^r \bmod p$ for message m, where p is a large prime, h is the generator of the prime order q cycle subgroup $G_q \subseteq Z_p^*$, and g is randomly chosen from $\langle h \rangle$. Since the receiver does not know $\log_h g$, Pedersen commitment scheme has perfect hidden property under the discrete logarithm assumption.

4 Property attestation protocol

4.1 Protocol phases

1) Setup. Generate the security parameters in the attestation system: l_q, l_H, l_ϕ, where l_q is the order length of the prime q order group; l_H is output length of hash function for Fiat-Shamir heuristic [20]; l_ϕ is security parameter length used to control statistical zero knowledge property. Let H be a strong collision-resistant hash function, $H : \{0,1\}^* \to \{0,1\}^{l_H}$. $\mathcal{T}$ chooses two groups $G = \langle g \rangle$, $G_T = \langle g_T \rangle$ of prime q order and admissible bilinear map $e : G \times G \to G_T$, and then set up private key $sk = (x, y, z)$ and public key $pk = (q, G, G_T, g, g_T, e, X, Y, Z)$ as the CL-LRSW signature scheme.

2) Issue. $\mathcal{P}$ carries out the integrity collection for the platform configuration, and then requests the property certificate to $\mathcal{T}$ with the current platform configuration cs. $\mathcal{T}$ evaluates the configuration information for the security property. Suppose the evaluated property is ps. $\mathcal{T}$ issues the configuration property certificate $\mathfrak{cre}$ for (cs, ps), where the signature is (a, A, b, B, c) in the certificate, $a \in_R G$, $A \leftarrow a^z$, $b \leftarrow a^y$, $B \leftarrow A^y$, $c \leftarrow a^{x+x\cdot y\cdot cs} A^{x\cdot y\cdot ps}$. Let $\sigma = (a, A, b, B, c)$, and define the configuration property certificate as $\mathfrak{cre} = ((cs, ps), \sigma)$. $\mathcal{T}$ sends the property certificate to $\mathcal{P}$; $\mathcal{P}$ checks the validation of certificate by the following equations, and saves the certificate for the further attestation.

$$e(a, Z) \stackrel{?}{=} e(g, A), \quad e(a, Y) \stackrel{?}{=} e(g, b), \quad e(A, Y) \stackrel{?}{=} e(g, B), \quad e(X, a) \cdot e(X, b)^{cs} \cdot e(X, B)^{ps} \stackrel{?}{=} e(g, c).$$

The CL-LRSW signature issued by $\mathcal{T}$ has the randomization property. Choose $r' \in Z_q^*$ at random, and compute $a' = a^{r'}$, $b' = b^{r'}$, $A' = A^{r'}$, $B' = B^{r'}$, $c' = c^{r'}$. Then $\sigma' = (a', A', b', B', c')$ is also the CL-LRSW signature for (cs, ps) after randomization.

3) Attest. $\mathcal{V}$ challenges $\mathcal{P}$ with a random number $N_v \in_R \{0,1\}^{l_H}$ for the property attestation. The host $\mathcal{H}$ invokes TCM chip($\mathcal{M}$) to attest the platform configuration after $\mathcal{P}$ receives the request. $\mathcal{M}$ generates random number $r_h, r_0 \in_R Z_q^*$, $N_t \in_R \{0,1\}^{l_\phi}$ with its inside RNG (random number generator), $h_T = g_T^{r_h} \in G_T$, commitment $C = g_T^{cs} h_T^{r_0}$, and signs the commitment with PIK (platform identity key) in TCM chip. The final signature is defined as $\delta = Sig_{\mathcal{M}}(C, N_v \| N_t)$.

$\mathcal{M}$ sends $g_T, h_T, C, r_0, \delta, N_t$ to host $\mathcal{H}$ for the property attestation signature.

$\mathcal{H}$ randomizes the signature σ on the property certificate, and gets $a' = a^{r'}$, $b' = b^{r'}$, $A' = A^{r'}$, $B' = B^{r'}$, $c' = c^{r'r^{-1}}$, where $r', r \in_R Z_q^*$. $\mathcal{H}$ chooses random number $t_1, t_2 \in_R Z_q^*$, computes $\sigma_0 = \sigma \cdot g^{t_1+t_2}$, and then computes the following equations:

$$v_x = e(X, a'), \quad v_{xy} = e(X, b'), \quad v_s = e(g, c'), \quad v_{xyz} = e(X, B').$$

We can optimize the protocol computation at this step. Host $\mathcal{H}$ can precompute the pairing according to property certificate, $\bar{v}_x = e(X, a)$, $\bar{v}_{xy} = e(X, b)$, $\bar{v}_s = e(g, c)$, $\bar{v}_{xyz} = e(X, B)$. When $\mathcal{H}$ receives the attestation challenge, $\mathcal{H}$ can easily compute $v_x = (\bar{v}_x)^{r'}$, $v_{xy} = (\bar{v}_{xy})^{r'}$, $v_s = (\bar{v}_s)^{r'}$, $v_{xyz} = (\bar{v}_{xyz})^{r'}$.

Next $\mathcal{H}$ executes the following steps on the proof of the knowledge signature on (N_v, N_t):

$$SPK\{(cs, r_0, r, t_1, t_2) | v_x v_{xy}^{cs} v_{xyz}^{ps} = v_s^r \wedge C = g_T^{cs} h_T^{r_0} \wedge d_1 = X^{t_1} \wedge d_2 = Y^{t_2}\}(N_v, N_t).$$

a) $\mathcal{H}$ chooses the random number $R_1, R_2, R_3, R_4, R_5 \in_R Z_q^*$, and computes $\tilde{T}_1 = v_s^{R_3} (v_x v_{xy}^{R_1} v_{xyz}^{ps})^{-1}$, $\tilde{T}_2 = g_T^{R_1} h_T^{R_2}$, $\tilde{d}_1 = X^{R_4}$, $\tilde{d}_2 = Y^{R_5}$.

b) $\mathcal{H}$ computes

$$c_H = H(q \| g \| X \| Y \| a' \| b' \| c' \| A' \| B' \| g_T \| h_T \| C \| \sigma_0 \| d_1 \| d_2 \| v_x \| v_{xy} \| v_{xyz} \| v_s \| \tilde{d}_1 \| \tilde{d}_2 \| \tilde{T}_1 \| \tilde{T}_2 \| N_v \| N_t).$$

c) $\mathcal{H}$ computes $s_1 = R_1 - c_H \cdot cs \bmod q$, $s_2 = R_2 - c_H \cdot r_0 \bmod q$, $s_3 = R_3 - c_H \cdot r \bmod q$, $s_4 = R_4 - c_H \cdot t_1 \bmod q$, $s_5 = R_5 - c_H \cdot t_2 \bmod q$.

At last $\mathcal{H}$ outputs the property attestation signature $\sigma_{PBA} = (\delta, C, a', A', b', B', c', c_H, s_1, s_2, s_3, s_4, s_5)$, and sends the attestation result to $\mathcal{V}$.

4) Verify. $pk = (q, G, G_T, g, g_T, e, X, Y, Z)$ and ps are all publicly known to the participants $\mathcal{P}$ and $\mathcal{V}$. When $\mathcal{V}$ receives the property attestation signature σ_{PBA} on (N_v, N_t), $\mathcal{V}$ verifies the result in the following steps:

a) $\mathcal{V}$ verifies $h_T \stackrel{?}{\in} G_T$, and then uses TCM public key to verify commitment signature: $Verf_{\mathcal{M}}(\delta, C, N_v \| N_t) \stackrel{?}{=} true$;

b) $\mathcal{V}$ verifies $e(a', Z) \stackrel{?}{=} e(g, A')$, $e(a', Y) \stackrel{?}{=} e(g, b')$, $e(A', Y) \stackrel{?}{=} e(g, B')$;

c) $\mathcal{V}$ computes

$$\hat{v}_x = e(X, a'), \quad \hat{v}_{xy} = e(X, b'), \quad \hat{v}_s = e(g, c'), \quad \hat{v}_{xyz} = e(X, B');$$
$$\hat{T}_1 = v_s^{s_3} v_{xy}^{-s_1} (v_x v_{xyz}^{ps})^{c_H - 1}, \quad \hat{T}_2 = g_T^{s_1} h_T^{s_2} C^{c_H}, \quad \hat{d}_1 = X^{s_4} d_1^{c_H}, \quad \hat{d}_2 = Y^{s_5} d_2^{c_H};$$

d) Next $\mathcal{V}$ verifies

$$c_H \stackrel{?}{=} H(q\|g\|X\|Y\|a'\|b'\|c'\|A'\|B'\|g_T\|h_T\|C\|\sigma_0\|d_1\|d_2\|v_x\|v_{xy}\|v_{xyz}\|v_s\|\hat{d}_1\|\hat{d}_2\|\hat{T}_1\|\hat{T}_2\|N_v\|N_t);$$

e) $\mathcal{V}$ sends (σ_0, d_1, d_2, ps) to $\mathcal{T}$, and requests $\mathcal{T}$ to check whether the certificate on (cs, ps) is revoked.

f) If all of above verifications are passed, $\mathcal{V}$ outputs ACCEPT, otherwise outputs REJECT.

5) Check. $\mathcal{V}$ requests $\mathcal{T}$ to check whether the property on cs is revoked with (σ_0, d_1, d_2, ps). $\mathcal{T}$ computes $\sigma = \frac{\sigma_0}{d_1^{1/x} d_2^{1/y}}$, and queries the property pair in certificate database by ps and σ. If $\mathcal{T}$ gets the relevant certificate, it indicates that the property on cs is still valid. Otherwise $\mathcal{T}$ notifies $\mathcal{V}$ to deny the attestation when (cs, ps) has been revoked. $\mathcal{T}$ can improve the efficiency of revocation verification when precomputing $1/x$ and $1/y$.

4.2 Security analysis

In this subsection we will discuss the protocol security in ROM (random oracle model). ROM was first proposed as a non-standard computation model by Bellare and Rogaway [21] in 1993. In the model any concrete object like hash function is treated 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 protocol's 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 1 (Unforgeability)**.** PBA-BM protocol provides the property of unforgeability under the LRSW assumption; more exactly, if adversary $\mathcal{A}$ can forge the PBA-BM signature with the non-negligible probability, there exists a simulator $\mathcal{S}$ solving the LRSW assumption or discrete logarithm hard problem with non-negligible probability in the polynomial time.

Proof. If adversary $\mathcal{A}$ can forge the PBA-BM signature in the attestation, we can make use of $\mathcal{A}$ to construct an algorithm $\mathfrak{B}$ to solve LRSW problem or discrete logarithm problem. We will illustrate the construction of the simulator $\mathcal{S}$: $\mathcal{S}$ interacts with the adversary playing the attacking game on the PBA-BM protocol phases. $\mathcal{S}$ sets up $pk = (q, G, G_T, g, g_T, e, X, Y, Z)$, $sk = (x, y)$ for $\mathcal{T}$; but $\mathcal{S}$ does not know the private key; $\mathcal{S}$ sends the system parameters to adversary $\mathcal{A}$.

$\mathcal{S}$ simulates each protocol step. The certificate issuing oracle, the attestation oracle and property revocation oracle must be required in the simulation. $\mathcal{S}$ maintains the following lists for keeping the consistent with the oracle queries. L_H stores the recorded data of query and response by hash oracle for knowledge signature $SPK\{(cs, r_0, r, t_1, t_2)|\ldots\}$. L_I stores the record data of query and response in the certificate issuing. The record in L_I is $(cs, ps, \mathfrak{cre}, \mathfrak{s})$, where $\mathfrak{cre} = (a, A, b, B, c)$, $\mathfrak{s} = 1$ means that the certificate on (cs, ps) has been revoked, otherwise $\mathfrak{s} = 0$. L_S stores the record data of the query

and response in attestation phase, and the record in L_S is $(\mathcal{P}_i, N_v, \mathfrak{cre}, \sigma_{PBA}, \mathfrak{c})$, where $\mathfrak{c} = 1$ means that prover $\mathcal{P}_i$ has been corrupted by adversary $\mathcal{A}$, that is, the host $\mathcal{H}_i$ in platform $\mathcal{P}_i$ controlled by adversary $\mathcal{A}$, otherwise $\mathfrak{c} = 0$.

Simulator: H(m). If $(m, h) \in L_H$, $\mathcal{S}$ returns h, otherwise $\mathcal{S}$ chooses the random number $h \in_R \{0,1\}^{l_H}$ in uniform distribution field, adds (m, h) into list L_H, and then returns h.

Simulator: Issuing(cs). The configuration cs is given to adversary $\mathcal{A}$. The simulator $\mathcal{S}$ (playing the role of $\mathcal{T}$) evaluates the security property for cs. Suppose the evaluation property is ps on cs given by $\mathcal{T}$. The simulator queries the oracle $\mathcal{O}$, and $\mathcal{O}$ responses with the certificate $\mathfrak{cre} = (a, A, b, B, c)$, and then $\mathcal{S}$ adds the record $(cs, ps, \mathfrak{cre}, 0)$ into the list L_I.

Simulator: Revoke(cs). Suppose the adversary $\mathcal{A}$ must query the configuration property certificate in Issuing(cs) before $\mathcal{A}$ revokes (cs, ps), otherwise $\mathcal{S}$ must execute Issuing(cs) first. The simulator $\mathcal{S}$ queries record $(cs, ps, \mathfrak{cre}, 0)$ in list L_I when revoking the property, then returns $\mathfrak{cre}$, and finally updates the record to $(cs, ps, \mathfrak{cre}, 1)$.

Simulator: Attest(cs). Let $N_v \in \{0,1\}^{l_H}$ be a random number chosen by $\mathcal{A}$, let $N_t \in \{0,1\}^{l_\phi}$ be the random number chosen by $\mathcal{S}$. The adversary chooses $(N_v, cs, ps, \mathfrak{cre})$ and requests the prover $\mathcal{P}$ for attestation. $\mathcal{S}$ queries the record $(cs, ps, \mathfrak{cre}, 0/1)$ in the list L_I, and then computes property attestation signature σ_{PBA}. We consider two cases in the property attestation:

Case 1: The security chip TCM is in physical security, the prover $\mathcal{P}$(Host $\mathcal{H}$) is an honest participant. $\mathcal{S}$ first computes configuration commitment, and then computes PBA-BM signature according to the attestation protocol.

1) The adversary $\mathcal{A}$ challenges the simulator $\mathcal{S}$ with random number N_v; $\mathcal{S}$ randomly chooses $f, k \in Z_q^*$; $N_t \in \{0,1\}^{l_\phi}$, then computes $h_T = g_T^f$, $C = g^k$, where C is the configuration commitment of security chip. TCM signs the commitment, $\mathcal{S}$ obtains the signature δ. and $\mathcal{S}$ sends g_T, h_T, N_t, C, δ to host $\mathcal{H}$.

2) The simulator $\mathcal{S}$ randomly chooses $s_1, s_2 \in_R Z_q^*$.

3) The simulator $\mathcal{S}$ randomly chooses $t_1, t_2, s_4, s_5 \in_R Z_q^*$, and computes $d_1 = X^{t_1}$, $d_2 = Y^{t_2}$.

4) The simulator $\mathcal{S}$ randomly chooses $r', s_3 \in_R Z_q^*$, $c' \in_R G$, and computes $a' = a^{r'}$, $b' = b^{r'}$, $A' = A^{r'}$, $B' = B^{r'}$, $\sigma_0 = \sigma \cdot g^{t_1+t_2}$.

5) The simulator $\mathcal{S}$ computes $v_x = e(X, a'), v_{xy} = e(X, b'), v_s = e(g, c'), v_{xyz} = e(X, B')$.

6) The simulator $\mathcal{S}$ randomly chooses $c_H \in_R \{0,1\}^{l_H}$, and queries c_H in list L_H. If c_H exists in L_H, $\mathcal{S}$ continues to execute this step.

7) The simulator $\mathcal{S}$ computes $\tilde{T}_1 = v_s^{s_3} v_{xy}^{-s_1} (v_x v_{xyz}^{ps})^{c_H - 1}$, $\tilde{T}_2 = g_T^{s_1} h_T^{s_2} C^{c_H}$, $\tilde{d}_1 = X^{s_4} d_1^{c_H}$, $\tilde{d}_2 = Y^{s_5} d_2^{c_H}$.

8) Set $w = q\|g\|X\|Y\|a'\|b'\|c'\|A'\|B'\|g_T\|h_T\|C\|\sigma_0\|d_1\|d_2\|v_x\|v_{xy}\|v_{xyz}\|v_s\|\tilde{d}_1\|\tilde{d}_2\|\tilde{T}_1\|\tilde{T}_2\|N_v\|N_t$. $\mathcal{S}$ queries (c_H, w) in list L_H, and decides whether the record is in the list. If yes, back to first step of the simulation, otherwise add (c_H, w) into list L_H.

9) The simulator $\mathcal{S}$ outputs $\sigma_{PBA} = (\delta, C, a', b', c', A', B', c_H, s_1, s_2, s_3, s_4, s_5)$.

10) The simulator $\mathcal{S}$ adds $(N_v, \mathfrak{cre}, \sigma_{PBA}, 0)$ into list L_S.

Case 2: The security chip is physically secure, and the prover $\mathcal{P}$ (Host $\mathcal{H}$) is controlled by adversary. $\mathcal{S}$ simulates the property attestation in this case. If Attest phase is successfully completed, $\mathcal{S}$ obtains the property attestation signature σ_{PBA} on cs; otherwise $\mathcal{S}$ obtains the forged signature for the property attestation by adversary.

The protocol outputs $\delta, C, a', A', b', B', c', \tilde{T}_1, \tilde{T}_2, c_H, s_1, s_2, s_3, s_4, s_5$ in Attest phase, which are indistinguishable between the simulation and real running. After the simulation of the property attestation ends, if adversary $\mathcal{A}$ can forge the attestation signature with the non-negligible probability ε, then we can construct the algorithm $\mathfrak{B}$ for solving the LRSW assumption problem at least $\varepsilon/2$ probability, or solve the discrete logarithm hard problem at least $\varepsilon/2$. LRSW instance $(q, G, G_T, g, g_T, e, X, Y)$ is given to algorithm $\mathfrak{B}$, where $G = \langle g \rangle$ is the cycle group, $X, Y \in G$, $X = g^x$, $Y = g^y$, admissible bilinear map $e : G \times G \to G_T$, $g_T = e(g, g)$ is the identity of group G_T. On input $m \in Z_q$, with the non-negligible probability the algorithm $\mathfrak{B}$ outputs (m, a, b, c) which has not been queried to LRSW oracle $\mathcal{O}$, where (m, a, b, c) satisfies $a \in G$, $b = a^y$, $c = a^{x+mxy}$, so the algorithm $\mathfrak{B}$ solves the LRSW assumption problem.

Suppose that $\mathcal{A}$ outputs the forged property attestation signature on the pair (cs, ps), and that the simulator $\mathcal{S}$ randomly chooses $z \in_R Z_q$, and computes $Z = g^z$. Algorithm $\mathfrak{B}$ constructs the system parameters of PBA-BM scheme $pk = (q, G, G_T, g, g_T, e, X, Y, Z)$, $sk = (x, y, z)$ like the parameters of LRSW assumption instance. Assume that the adversary has queried q_s Attest Oracle before forging the property attestation signature, and define the queried configuration property pairs as (cs_i, ps_i), $i = 1, \ldots, q_s$. Because $\mathcal{A}$ forges the property attestation signature on pair (cs, ps) which has not been queried to Attest Oracle, $(cs, ps) \neq (cs_i, ps_i)$. We discuss the property attestation protocol unforgeability in two cases:

1) For the configuration property pair (cs_i, ps_i), $i = 1, \ldots, q_s$, there exists some i at least $\varepsilon/2$, which satisfies $cs + ps \cdot z = cs_i + ps_i \cdot z$. The adversary $\mathcal{A}$ can compute $z = (cs_i - cs)/(ps - ps_i) \mod q$ in this case, so the adversary can solve the discrete logarithm hard problem when z is set at the target value $\log_g Z$.

2) The configuration property pair (cs_i, ps_i), $i = 1, \ldots, q_s$, has at least $\varepsilon/2$ probability that $cs + ps \cdot z \neq cs_i + ps_i \cdot z$. If simulator $\mathcal{S}$ (plays in the role of $\mathcal{V}$) gets some valid property attestation signature σ_{PBA} from the adversary $\mathcal{A}$, the signature is not in the attestation list L_S. $\mathcal{S}$ rewinds adversary $\mathcal{A}$ to the point when attestation oracle is invoked to generate c_H in the signature, so the different c_H is provided to $\mathcal{A}$, $\mathcal{S}$ can extract two signatures on $(\delta, C, \tilde{T}_1, \tilde{T}_2, a', b', c', A', B')$, which have different c_H and s_1, s_2, s_3, s_4, s_5. The two signatures are defined as

$$(\delta, C, \tilde{T}_1, \tilde{T}_2, a', b', c', A', B', c_H^{(0)}, s_1^{(0)}, s_2^{(0)}, s_3^{(0)}, s_4^{(0)}, s_5^{(0)}),$$
$$(\delta, C, \tilde{T}_1, \tilde{T}_2, a', b', c', A', B', c_H^{(1)}, s_1^{(1)}, s_2^{(1)}, s_3^{(1)}, s_4^{(1)}, s_5^{(1)}).$$

Let $a' = g^\alpha$, $b' = g^\beta$, $c' = g^\gamma$, $\alpha, \beta, \gamma \in Z_q^*$. $c_H^{(0)} \neq c_H^{(1)}$, $s_X^{(0)} \neq s_X^{(1)} (X = 1, 2, \ldots, 5)$ in the two signatures. Let $\Delta c_H = c_H^{(0)} - c_H^{(1)}$, $\Delta s_X = s_X^{(0)} - s_X^{(1)}$. The two signatures both satisfy the verification condition:

$$\tilde{T}_1 = v_s^{s_3^{(0)}} v_{xy}^{-s_1^{(0)}} (v_x v_{xyz}^{ps})^{c_H^{(0)} - 1}, \quad \tilde{T}_1 = v_s^{s_3^{(1)}} v_{xy}^{-s_1^{(1)}} (v_x v_{xyz}^{ps})^{c_H^{(1)} - 1},$$

where $v_x = e(X, a'), v_{xy} = e(X, b'), v_s = e(g, c'), v_{xyz} = e(X, B')$ such that $v_s^{\Delta s_3} = v_{xy}^{-\Delta s_1} (v_x v_{xyz}^{ps})^{\Delta c_H}$. Because of $\Delta c_H \neq 0$, the squaring of Δc_H^{-1} can be executed on both sides of the equation. Let $\hat{s}_X = \Delta s_X / \Delta c_H$ $(X = 1, 2, \ldots, 5)$. Then we get $v_s^{\hat{s}_3} = v_x v_{xy}^{\hat{s}_1} v_{xyz}^{ps}$. From the attestation process of rewinding $\mathcal{A}$ by simulator, we can know $cs = \Delta s_1 / \Delta c_H = \hat{s}_1$. Let $m = \hat{s}_1 + ps \cdot z \mod q$, and let $(\hat{s}_1, ps)$ be queried in attest oracle. We can know from the verification algorithm of PBA-BM scheme that $\sigma_{PBA} = (\delta, C, a', b', c', A', B', *, *, *, *, *, *)$ statisfies: 1) $e(a', Y) = e(g, b')$; 2) $v_s^{\hat{s}_3} = v_x v_{xy}^{\hat{s}_1} v_{xyz}^{ps}$, that is, $e(g, c')^{\hat{s}_3} = e(X, a') \cdot e(X, b')^{\hat{s}_1} \cdot e(X, B')^{ps}$.

1)

$$e(a', Y) = e(g, b'), \quad e(g^\alpha, g^y) = e(g, g^\beta), \quad g^{\alpha y} = g_T^\beta,$$

such that $\beta = \alpha y \mod q$.

2) Because $e(X, B') = e(X, A'^y) = e(X, a'^{yz}) = e(X, b')^z$,

$$\begin{aligned}
&e(g, c')^{\hat{s}_3} = e(X, a') \cdot e(X, b')^{\hat{s}_1} \cdot e(X, B')^{ps},\\
&e(g, c')^{\hat{s}_3} = e(X, a') \cdot e(X, b')^{\hat{s}_1} \cdot e(X, b')^{ps \cdot z},\\
&e(g, c')^{\hat{s}_3} = e(X, a') \cdot e(X, b')^{\hat{s}_1 + ps \cdot z},\\
&e(g, g)^{\hat{s}_3 \cdot \gamma} = e(g^x, g^\alpha) \cdot e(X, b')^m,\\
&g_T^{\hat{s}_3 \cdot \gamma} = g_T^{x\alpha} \cdot g_T^{mxy\alpha} = g_T^{(x + mxy)\alpha},
\end{aligned}$$

such that $\hat{s}_3 \cdot \gamma = (x + mxy)\alpha \mod q$.

To sum up, input $m = cs + ps \cdot z \mod q$, algorithm $\mathfrak{B}$ outputs $a = a' = g^\alpha$, $b = b' = g^\beta$, $c = c'^{\hat{s}_3} = g^{\hat{s}_3 \cdot \gamma}$, such that (m, a, b, c) is the LRSW assumption instance satisfying: $b = g^\beta = g^{\alpha y} = a^y$; $c = g^{\hat{s}_3 \cdot \gamma} = g^{(x + mxy)\alpha} = a^{x + mxy}$. If $\mathcal{A}$ can win the $Game_{\mathcal{A}}^{att-fg}(1^k)$ with a non-negligible probability, the algorithm $\mathfrak{B}$ either solves the LRSW problem with non-negligible probability, or solves discrete logarithm hard problem with non-negligible probability.

Theorem 2 (Configuration privacy)**.** PBA-BM protocol provides the security property of configuration privacy; more exactly, if there exists an adversary $\mathcal{A}$ distinguishing the different configuration from the same property with a non-negligible probability, there must exist a polynomial time simulator $\mathcal{S}$ that can break the perfect hidden property of the commitment scheme with a non-negligible probability.

Proof. We construct simulator $\mathcal{S}$ playing the protocol participants, which plays the $Game_{\mathcal{A}}^{cf-prv}(1^k)$ interacting with adversary. Even if the adversary $\mathcal{A}$ is computationally unbounded, the advantage probability $\mathrm{Adv}[\mathcal{A}_{PBA}^{cf-prv}(1^k)]$ for $\mathcal{A}$ wining the game is negligible on the security parameter k. If $\mathcal{A}$ can break the configuration privacy of PBA-BM scheme, the simulator can open the secret from the commitment.

The commitment $C = g_T^{cs} h_T^{r_0}$ is given to $\mathcal{S}$, where $cs \in CS = \{cs_1, cs_2, \ldots, cs_n\}$. Then $\mathcal{S}$ plays $Game_{\mathcal{A}}^{cf-prv}(1^k)$ with adversary $\mathcal{A}$.

When receiving the challenge from $\mathcal{A}$, $\mathcal{S}$ uses the commitment C to execute PBA-BM attestation protocol, but $\mathcal{S}$ does not know the secrets cs and r_0 hidden in commitment. $\mathcal{S}$ gets the TCM signature $\delta = Sig_{\mathcal{M}}(C, N_v \| N_t)$ from security chip. Because the adversary has the unbounded computation ability, $\mathcal{S}$ can compute α, k, which satisfies $h_T = g_T^{\alpha}$, $C = g_T^k = g_T^{cs+\alpha r_0}$. $\mathcal{S}$ finds out the configuration certificates $(cs_i, ps, \mathfrak{cre}_i)$, $i = 1, \ldots, n$, which has similar attested properties. For the property certificate $\mathfrak{cre}_i$, $\mathcal{S}$ constructs the pair (cs_i, r_i), $k = cs_i + \alpha r_i$. According to the simulation of Attest(cs), the simulator computes the property attestation signature $\sigma_{PBA}^{(i)} = (\delta, C, a'^{(i)}, b'^{(i)}, c'^{(i)}, A'^{(i)}, B'^{(i)}, c_H^{(i)}, s_1^{(i)}, s_2^{(i)}, s_3^{(i)}, s_4^{(i)}, s_5^{(i)})$, and then sends $\sigma_{PBA}^{(i)}$ to adversary $\mathcal{A}$.

At the end of the game, the adversary $\mathcal{A}$ outputs index j. If $cs_j = cs$, $\mathcal{S}$ can open the commitment C with (cs_j, r_j). The probability for $\mathcal{S}$ to open the commitment is the one for $\mathcal{A}$ to decide $cs_j = cs$. The adversary $\mathcal{A}$ can win the $Game_{\mathcal{A}}^{cf-prv}(1^k)$ with the non-negligible probability $\mathrm{Adv}[\mathcal{A}_{PBA}^{cf-prv}(1^k)]$, certainly implying that the simulator $\mathcal{S}$ can open the commitment with the non-negligible probability.

5 Performance analysis

TCM computations in our scheme are fully supported according to China's TCM design and implementation specification [14], because TCM adopts the SM2 elliptic curve algorithm. The current TCM chip uses the 256 bits ECC key to sign message, so the TCM signature length is $|\delta| = 512$ bits. NTL cryptographic library [22] and PBC bilinear map library [23] provided by Stanford University are suitable for the host cryptographic computations in the property attestation. The system implementation can choose the elliptic curve $E(Fq)(|q| = 170)$ recommended in [24], where the point in groups G and G_T is represented by 171 and 1020 bits bitstring respectively. The integrity and confidentiality of the attestation data between prover and verifier can be ensured through the trust channel established by Openssl.

Our scheme (PBA-BM) has the following parameters setting: $l_q(170)$, $l_H(160)$, $l_\phi(80)$. Because the property attestation signature contains 5 elements in G, 5 elements in Z_q and 1 element in G_T, the total signature length is 2885+$|\delta|$=3397 bits.

According to the parameters of PBA scheme [12]: $l_{ps}(160)$, $l_{cs}(160)$, $l_\phi(80)$, $l_P(1632)$, $l_Q(208)$, $l_n(2048)$, $l_e(368)$, $l_{e'}(120)$, $l_v(2536)$, such that the length of PBA signature is 7162+$|\delta|$=7162+2048=9210 bits.

According to the parameters setting of PBA-RS scheme [13]: $l_{ps}(160)$, $l_{cs}(160)$, $l_P(1632)$, $l_Q(208)$, $l_n(2048)$, such that the signature length is $(n+1)$*208 + $|\delta|$=$(n+1)$*208+2048 bits.

We estimate the computation cost based on the technique for general exponentiation, where the exponentiation computations can be measured by squarings and multiplications for simplicity. For example $y = g^x$, $x \in \{0,1\}^{160}$ such that the average counts of the bit 1 is 80 in x. The computation cost on g^x includes 160 squarings and 80 multiplications, denoted by 160S+80M for short. We denote pairing computation by P, and summarize computation comparison of property attestation protocol in Table 1.

PBA scheme relies on the trust third party, and it takes too much computations to check the property revocation. The Attest/Verify computations are highly efficient in PBA-RS scheme, but it increases the extra property management for negotiating the configuration between Prover and Verifier. Table 1 shows that PBA-BM has the less signature length, and more efficient computation cost in TCM Attest phase than other schemes.

Table 1 Computation cost comparison of property attestation schemes[a)]

	PBA	PBA-RS	PBA-BM
Issuing	3224S+1612M	—	850S+426M
Attest	5201S+2606M	(528+208n)S+(265+105n)M	3540S+1777M
Verify	4665S+2337M	(368+208n)S+(185+105n)M	1830S+922M+4P
Check	k*(5520S+2771M)	—	340S+172M
Signature length (bit)	9210	(n+1)*208+2048	3397
Assumption	Strong-RSA assumption	Ring-signature	LRSW assumption

a) n is counts of negotiating configurations accepted by both side. k is counts of property revocation.

6 Conclusions

The paper sums up the problems on the property attestation, and builds the property attestation model. A new property attestation based on bilinear map is proposed, which makes full use of TCM's cryptographic feature. TCM can attest the configuration commitment safely with trust third party. The protocol prevents the adversary's forgery of property attestation signature as well as the compromise of the platform configuration from verifier. We will research the high load on the trust third party's verification, and make the protocol more practical in future work.

Acknowledgements

This work was supported by the National Basic Research Program of China (Grant No. 2007CB311202), and the National Natural Science Foundation of China (Grant No. 60673083).

References

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 Trusted Computing Group. TCG Software Stack (TSS) Specification, Version 1.10[EB/OL]. [2003-8-20]. https://www.trusted-computinggroup.org.
3 Trusted Computing Group. TCG Glossary Specification, Revision 0.1[EB/OL]. [2004-7-22]. https://www.trusted-computinggroup.org/home.
4 Sailer R, Zhang X L, Jaeger T, et al. Design and implementation of a TCG-based integrity measurement architecture. In: 13th Usenix Security Symposium. San Diego: USENIX Association, 2004. 16–16
5 Safford D, Zohar M. A Trusted Linux Clent (TLC). http://www.research.ibm.com/gsal/tcpa/tlc.pdf
6 Haldar V, Chandra D, Franz M. Semantic remote attestation: A virtual machine directed approach to trusted computing. In: Proceedings of USENIX Virtual Machine Research and Technology Symposium, Long Beach: California State University, 2004. 145–154
7 Seshadri A, Perrig A, Doorn L V, et al. SWATT: Software-based Attestation for embedded devices. In: Proceedings of the IEEE Security & Privacy Conference, Oakland: IEEE, 2004. 272–282
8 Yoshihama S, Ebringer T, Nakamura M, et al. WS-Attestation: Efficient and fine-grained remote attestation on web services. In: Proceedings of International Conference on Web Services. Washington, DC: IEEE, 2005. 743–750
9 Sadeghi A, Stüble C. Property-based attestation for computing platforms: caring about properties, not mechanisms. In: Proceedings of the 2004 Workshop on New Security Paradigms. Nova Scotia: ACM Press, 2004. 67–77
10 Poritz J, Schunter M, Herreweghen E V, et al. Property attestation—scalable and privacy-friendly security assessment of peer computers. IBM Research Report RZ 3548. 2004
11 Chen L Q, Landfermann R, Löhr H, et al. A protocol for property-based attestation. In: Proceedings of the first ACM workshop on Scalable trusted computing. New York: ACM Press, 2006. 7–16
12 Chen L Q, Löhr H, Manulis M, et al. Property-based attestation without a trusted third party. In: Proceedings of the 11th International Conference on Information Security. LNCS, vol. 5222. Berlin: Springer-Verlag, 2008. 31–46
13 Kuehn U, Selhorst M, Stueble C. Realizing property-based attestation and sealing with commonly available hard- and software. In: Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing. New York: ACM, 2007. 50–57
14 China State Password Administration Committee. Functionality and Interface Specification of Cryptographic Supporting Platform for Trusted Computing, 2007. http://www.oscca.gov.cn/

15 Camenisch J, Stadler M. Efficient group signature schemes for large groups. In: CAIP 1997. LNCS, vol. 1296. Heidelberg: Springer, 1997. 410–424
16 Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Kilian J, ed. CRYPTO 2001. LNCS, vol. 2139. Heidelberg: Springer, 2001. 213–229
17 Camenisch J, Lysyanskaya A. Signature schemes and anonymous credentials from bilinear maps. In: Franklin M, ed. CRYPTO 2004. LNCS, vol. 3152. Heidelberg: Springer, 2004. 56–72
18 Lysyanskaya A, Rivest R L, Sahai A, et al. Pseudonym systems. In: Heys H M, Adams C M, eds. SAC 1999. LNCS, vol. 1758. Heidelberg: Springer, 2000. 184–199
19 Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum J, ed. Advances in Cryptology-CRYPTO '91, LNCS, vol. 576. Berlin: Springer-Verlag, 1992. 129–140
20 Fiat A, Shamir A. How to prove ourself: Practical solution to identification and signature problems. In: Advances in Cryptology-Crypto'86. LNCS 263. London: Springer-Verlag, 1987. 186–199
21 Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st CCS. New York: ACM Press, 1993. 62–73
22 NTL: A Library for doing Number Theory[EB/OL]. http://www.shoup.net/ntl/
23 The Pairing-Based Cryptography Library[EB/OL]. http://crypto.stanford.edu/pbc/times.html
24 Miyaji A, Nakabayashi M, Takano S. New explicit conditions of elliptic curve traces for FRreduction. IEICE Trans, 2002, E85-A: 481–484