Bilinear Parings in Property-based Attestation

Ting Chen
Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
Informatization Office, East China University of Political Science and Law, Shanghai 201620, China
chenting@ecupl.edu.cn

Huiqun Yu
Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China
yhq@ecust.edu.cn

Abstract—One of the objectives of trusted computing is to provide remote attestation method that is able to confirm the status of remote platform or application. Existing property-based attestation is based on the strong-RSA assumption and the required key length is too long. What’s more, a considerable number of RSA-length operations having to be performed which lead to low computational efficiency. Bilinear parings-Based Attestation model, which based on elliptic curve discrete logarithm bilinear paring, can shorten the required key length and reduce bandwidth usage at the same premise of safety performance requirements, as well as ensure platform configurations not to be exposed to the platform while improving operating efficiency. On the other hand, the model includes many trusted computing platform parameters in order to resist replay attacks, and take use of information hiding technology to hide certificates and effectively preventing anyone with a source of certificate misuse of the certificate.

Index Terms—bilinear paring; remote attestation; CL signature; discrete logarithm

I. INTRODUCTION

With the continuous development of network technology, terminals that can participate in the exchange of online information become more abundant. To ensure that the server in the provision of services (or software), as well as the credibility of the various terminal itself, TCG (Trusted Computing Group) has proven remote attestation technology [2], which proved to be as the expansion of local integrity verification. Services will fail when the code was found to be tampered with.

Attestation is the process of vouching for the accuracy of information. External entities can attest to shielded locations, protected capabilities, and Roots of Trust. A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform. All forms of attestation require reliable evidence of the attesting entity. The TPM, attestation to the platform, attestations of the platform and authentication of the platform can understand attestation along several dimensions, attestation attestation by the TPM, attestation to the platform, attestation of the platform and authentication of the platform. Attestation by the TPM is an operation that provides proof of data known to the TPM. This is done by digitally signing specific internal TPM data using an attestation identity key (AIK). The acceptance and validity of both the integrity measurements and the AIK itself are determined by a verifier. The AIK is obtained using either the Privacy CA or via a trusted attestation protocol. Attestation to the platform is an operation that provides proof that a platform can be trusted to report integrity measurements; performed using the set or subset of the credentials associated with the platform; used to issue an AIK credential. Attestation of the platform is an operation that provides proof of a set of the platform’s integrity measurements. This is done by digitally signing a set of PCRs using an AIK in the TPM. Authentication of the platform provides evidence of a claimed platform identity.

The TCG solution for remote attestation measures all the code executed by using certain metrics (currently a cryptographic hash value over the code binary). The result is stored in special registers in the TPM (Trusted Platform Module) before executing the code. This procedure is bootstrapped starting with a kind of pre-BIOS that is trusted by default and measures the bootloader, storing the result, which may be inferred that the current state of the system, as well as follow-up to predict the behavior. This procedure builds the so-called chain of trust, which can then be extended to the operating system components up to applications. Remote attestation mechanism can limit the use of a remote client through the targeted selection of remote applications, and prevent malicious programs from being used or may have defects in the application of the abuse of services to prevent the misuse of Trojans, as well as avoid malicious terminal connection.

TCG attestation protocol consists of several steps:
Remote attestation protocol should have the following characteristics:

- It should be able to confirm the identity of both the communication party, and check out the pretender;
- It should ensure the timeliness of the certificate, taking into account the revocation of the certificate;
- It should ensure that the information in the transmission will not be tampered with, and the integrity of information;
- It should ensure that the information transmission is not leaking;
- It should be able to resist replay attacks, and ensure that the information of the fresh;

II. RELATED WORKS

TCG proposes the remote attestation standard [2], and R. Sailer and others extend the TCG standard to dynamic executable content from the BIOS all the way up into the application layer [3]. Identity-based digital signature is proposed here. What’s more, it needs to protect system specific configuration information. Otherwise, revealing the system configuration may lead to privacy violations and discrimination against the underlying system since the remote party may exclude them from his/her business model. Seshadri and others provide a remote attestation mechanism to embedded devices without the precondition of TPM [4]. They also proposed a remote code integrity verification solution by combination of legacy systems, Pioneer [5]. The model only needs software-based approach to achieve the remote attestation, which taking into account the time factor, using checksum methods, and pseudo-random memory traversal method of detection, from reduced to some extent, the attacker through the memory copy to achieve the purpose of tampering with the software. This method does not require trusted third party involved. However, the program does not involve proof of identity. It is easy to by pretending to attack, and there are difficulties to achieve.

Some scholars put forward the behavior-based model to prove the remote attestation [6, 17]. Contrary to the fact that the model calls for all the acts are known, most acts are unknown, so there are difficult to achieve. The model was proved only in theory, has yet to see the realization of the relevant literature. Haldar, who will be proof of remote attestation with virtual technology, the use of language-based virtual machine technology, a complex and dynamic, high-level process attributes, platform-independent remote attestation mode [8]. The model did not integrate verification platform with identity information that the model does not satisfy the requirement of confirm the identity as mention above. Other than providing system configuration to verifier directly, Sadeghi provided system configuration to the trusted third party (TTP), which derives its attributes, and issuance of the corresponding attribute certificate [9]. Attribute certificate can be used to prove the credibility of the platform during the remote attestation mechanism. Liqun Chen and Sadeghi made on the specific realization of the mechanism above which called property-based remote attestation [10]. It avoids leakage the system configuration information, and reflects the identity of the platform.

Property-based remote attestation model conceals the configuration information of the system platform [9,10], thus avoiding leakage of the system configuration; Verifier can verify the security of authentication system platform through attribute certificate, and do not need to understand the complexity of the systems configuration, such as the system software version number, which reduces the occurrence of a monopoly in the software industry; In particular, this solution also allows a more flexible way of handling system patches and updates, the corresponding attribute certificate can be revoked without a TTP. Because of security needs, the required RSA key length is too long, and it must spend a great deal computation of RSA key for TPM consequently. We propose a remote attestation solution based on the properties of bilinear pairings, with respect to the RSA key, the bilinear pairings can use shorter key length, so that we can use smaller bandwidth and memory requirements.

III. INTRODUCTION TO PROPERTY-BASED ATTESTATION [10]

The TCG attestation protocol is used to give assurance about the platform configuration cs to a remote party. Here the attesting party, the attestor, reports to a remote party, the verifier, the configuration of a machine(s) to be attested, e.g., the configuration of the platform or/and of applications. To guarantee integrity and freshness, this value and a fresh nonce N, must be digitally signed with an asymmetric key called Attestation Identity Key (AIK) that is under the sole control of the TPM. A trusted third party called Privacy Certification Authority (Privacy-CA) is used to guarantee the pseudonymity of the AIKs. However, this party can always link the transactions a certain platform was involved in. To overcome this problem, version 1.2 of the TCG specification [18] defines another cryptographic protocol called Direct Anonymous Attestation (DAA) [19] that, roughly spoken,
provides users with an unlimited number of pseudonyms without requiring a Privacy-CA. Note that the anonymity provided by DAA or Privacy-CAs is completely orthogonal to the stated goals of this paper. Some scholars name the TCG attestation model load-time binary attestation [20]. The TCG attestation protocol proceeds is shown in Fig.1.

**Figure 1. Attestation Protocol and Message Exchange**

In its basic form TCG attestation has some shortcomings. First, a huge number of possible configurations exist, because every new version of a component will have a different binary and hence produces a different hash value. Lastly, load-time attestation provides no run-time assurance as there can be a big time difference between integrity measurement (i.e., startup) and integrity reporting. The platform could be compromised since it has been booted.

To overcoming some of the shortcomings of binary attestation, some scholars provide a more general and flexible solution to the attestation problem, a property-based attestation (PBA) approach [9]. It means that attestation should only determine whether a platform (configuration) or an application has the desired property. This avoids revealing the concrete configuration of software and hardware components. For example, it would not matter whether the application was Webbrowser A or B, as long as both have the same properties. In contrast, the binary attestation function provided by TCG-compliant hardware attests the system configuration of a platform that was determined at system startup. For (nearly) all practical applications, the verifier is not really interested in the specific system or application configuration. Informally, a property, in this context, describes an aspect of the behavior of the underlying object (platform / application) with respect to certain requirements, e.g., a security-related requirement. In general, properties for different abstraction levels are imaginable. For instance, a platform property may state that a platform is privacy preserving, i.e., it has built-in measures conform to the privacy laws, or that the platform provides isolation, i.e., strictly separates processes from each other, or it provides Multi-Level Security (MLS) and so forth.

The property-based attestation model shown in Fig.2 is built on the CL signatures [11], Pedersen commitment algorithm [12] based on the premise that strong RSA assumption and the corresponding algebraic structure of the discrete logarithm assumption. Based on the strong RSA modulus $N = pq$ as factorization. It figures out property configuration certificates protocol, signing algorithm, verification algorithm, and the revocation check protocol.

In Fig.2, a platform PF represents our main IT system, i.e., it consists of all (software and hardware) components of a system. The Trusted Platform Module (TPM) $M$ is one of the main components of a platform PF. TPM $M$ is embedded in a host $H$. The host includes the software running on the platform PF. The TPM can only communicate with other parties (external to the platform) via the host. A verifier $V$ is a party that wants to verify the attestation result of some platform. The certificate issuer $CI$ is the party that certifies mappings between properties and configurations attesting that a given platform configuration $cs$ fulfills a desired property $p$ by means of a property certificate $\sigma_{cI}$.

**Figure 2. Property-based Attestation model**

In Table I we present the computational cost for all the other algorithms, with respect to each player. An entry of the form:

$$1 \cdot Q_a + 2 \cdot Q_c + 3 \cdot Q_s$$

implies that the cost is about one exponentiation modulo $N$, two modulo $\Gamma$ and three multiexponentiations with two exponents modulo $N$, i.e., three operations of the form $g^a h^b \pmod N$. Note, that a multiexponentiation with $m$ exponents can often be performed significantly faster than $m$ separate exponentiations.

In all tables presented in the paper, $Q_a$ denotes the cost of a exponentiation computation in $G_1$; $Q_c$ denotes the cost about a multixponentiation of $m$ values in $G_1$; $Q_s$ denotes the cost of a exponentiation computation in $G_1$; $Q_i$ denotes the cost about a multixponentiation of $m$ values in $G_1$; $Q_r$ denotes the cost of a pairing computation, such as $\tau \leftarrow \langle X, D \rangle$; $Q_p$ denotes the cost of a pairing computation; $Q_m$ denotes the cost about one exponentiation modulo $N$, such as $g^a \pmod N$; and $Q_m^w$ denotes the cost of a multiexponentiation with $m$ exponents modulo $N$; $Q_m^w$ denotes the cost of a multiexponentiation with $m$ exponents modulo $P$, where $P$ is a large prime number, such as $g^a \pmod P$.
### Table I. Cost of the Property-Based Attestation Protocol [10]

<table>
<thead>
<tr>
<th>Operation</th>
<th>Party</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute-Configuration protocol</td>
<td>TPM</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Issuer</td>
<td>3*Q′_s</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>2*Q′_s</td>
</tr>
<tr>
<td>Sign Protocol</td>
<td>TPM</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>2*Q′_x</td>
</tr>
<tr>
<td>Verification Algorithm</td>
<td>Verifer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>3*Q′_y</td>
</tr>
<tr>
<td>Revocation Algorithm</td>
<td>Verifer</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>2*Q′_z</td>
</tr>
</tbody>
</table>

IV. BILINEAR PAIRING

**A. Conception and characteristic**

At first, bilinear pairing is on the form of Weil pairing proposed by Menezes to solve supersingular elliptic curve discrete logarithm problem [13,14].

We let \( G_1 \times G_2 \rightarrow G_3 \) denote a pairing between three groups of prime order \( q \); \( G_1 \) (resp. \( G_2 \)) is a cyclic additive group while \( G_3 \) is a cyclic multiplicative group. We let the generator of \( G_1 \) (resp. \( G_2 \)) be denoted by \( P_1 \) (resp. \( P_2 \)).

1. Bilinearity

For all \( P \in G_1, Q \in G_2 \), a, b \( \in \mathbb{Z}_q \), we have \( \ell (aP, bQ) = \ell (P, Q)^{ab} \) and \( \ell (P_1, P_2.Q) = \ell (P_1, Q) \ell (P_2, Q) \).

2. Non-degeneracy

- For all \( P \in G_1 \), with \( P \neq 0 \), there is some \( Q \in G_2 \) such that \( \ell (P, Q) = 1 \).
- For all \( Q \in G_2 \), with \( Q \neq 0 \), there is some \( P \in G_1 \) such that \( \ell (P, Q) = 1 \).

Bilinear pairings can be derived from the general elliptic curve of Weil or Tate while \( G_1 \neq G_2 \). It needs to build in three different groups, the application is very inconvenient. At the condition \( G_1 = G_2 \), although the bilinear pairings can only be modified on the supersingular elliptic curve of Weil or Tate, we take this type of bilinear pairing for its simplify and convenient application, and sign it with the notation \( \ell \).

**B. Computation problem based on bilinear pairing**

Computational Diffie-Hellman (CDH) assumption: for all random \( a, b \in \mathbb{Z}_q \), given \((P, aP, bP)\), it is computationally intractable to compute the value \( abP \).

Bilinear Diffie–Hellman Problem (BDH): \( G_1 \) is a cycle additive group, and \( \ell \) is a bilinear pairing. We let the generator of \( G_1 \) be denoted by \( P_1 \) and \( aP_1, bP_1, cP_1 \in G_1 \).

Decision Bilinear Diffie–Hellman problem (DBDH): Given \((P, aP, bP, cP)\), test whether \( \ell (P, cP) = \ell (aP, bP)^c \).

LSRW Assumption: let \( \ell : G_1 \times G_2 \rightarrow G_3 \) denote a pairing between three groups of prime order \( q \). Let \( G_1 = \langle g \rangle, G_1 = \langle g \rangle, G_1 = \langle g \rangle \), \( X, Y \in G_1, X = g^x, Y = g^y \). Let \( O_x, y(.) \) be an oracle that, on input a value \( \theta \) outputs a triple \( A = [a, x, ax + mxy] \) for a randomly chosen \( a \). Then for all probabilistic polynomial time adversaries \( \Phi \), \( \text{Adv} (k) \) defined as follows is a negligible function:

\[
\text{Adv} (k) = \Pr [((q, G_1, G_1, g, g, r) \leftarrow \text{Setup}(k)); x \leftarrow Z_q, y \leftarrow Z_q, X = g^x, Y = g^y; (Q, m, a, b, c) \leftarrow \Phi^{O_x, y}(q, G_1, G_1, g, g, r): m \neq Q \land a \in G_1 \land b = a^r \land c = a^{mxy}]
\]

LSRW Assumption was introduced by Lysyanskaya et al. [15], and considered for groups that are not known to admit an efficient bilinear map. It was also shown, that this assumption holds for generic groups, and is independent of the decisional Diffie-Hellman assumption.

V. THE CAMENSICH-LYSYANSKAYA SIGNATURE SCHEME

Our attestation model is based on the Camensich-Lysyanskaya signature scheme [16]. Before introduce the model, we must present the signature scheme at first. There are three CL signature schemes, and the signature scheme B is used here. We let \( \ell : G_1 \times G_1 \rightarrow G_1 \) denote a pairing between three groups of prime order \( q \). We let the generator of \( G_1 \) be denoted by \( P_1 \).

**Key generation.** The private key is a pair \((x, y, z) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q \). The public key is given by the pair \((X, Y, Z) \in G_1 \times G_1 \times G_1 \) where \( X = xP_1 \) and \( Y = yP_1 \) and \( Z = zP_1 \).

**Signature.** On input message \((m, r)\), secret key \( sk = (x, y, z)\), and public key \((X, Y, Z)\) do:

- Choose a random \( a \ll G \).
- Let \( A = za \).
- Let \( b = ya, B = yA \).
- Let \( c = [(x + xym)]a \cdot xyrA \).

Output \( \sigma = (a, A, b, B, c) \).

**Verification.** On input \( pk = (X, Y, Z) \), message \((m, r)\), and purported signature \( \sigma = (a, A, b, B, c) \), check the following:

1. \( A \) was formed correctly: \( \ell (a, Z) = \ell (P_1, A) \).
2. \( b \) and \( B \) were formed correctly: \( \ell (a, Y) = \ell (P_1, b) \) and \( A, Y = \ell (P_1, B) \).
3. \( c \) was formed correctly: \( \ell (X, a) \cdot \ell (X, b)^m \cdot \ell (X, B)^c = \ell (P_1, c) \).

Note that the values \((mZr P_1, a, A, b, B, c)\) are information-theoretically independent of \( m \) if \( r \) is chosen randomly. This will become crucial when using this signature scheme in the context of attestation system.

**Theorem 1.** Signature Scheme B described above is correct and secure under the LRSW assumption.

**Proof.** We will first show correctness. The first verification equation holds as:

\[
\ell (a, Z) = \ell (a, P_1)^x \\
\ell (P_1, a)^y \\
\ell (P_1, A).
\]

The two second ones hold as:

\[
\ell (a, Y) = \ell (a, P_1)^y \\
\ell (A, Y)
\]

© 2011 ACADEMY PUBLISHER
The third one holds because
\[ \ell \left( X, a \right) \cdot \ell \left( X, b \right)^m \cdot \ell \left( X, B \right)^r \]
\[ = \ell \left( P_1, a \right)^x \cdot \ell \left( P_1, a \right)^{mx \cdot \ell \left( P_1, a \right)^{zxy}} \]
\[ = \ell \left( P_1, ax + my + zxy \right) \]
\[ = \ell \left( P_1, ax + myAx + yr \right) \]
\[ = \ell \left( P_1, c \right). \]

[16] proves security of this signature using (1) the fact that Signature Scheme A is secure under the LRSW assumption; and (2) the fact that the LRSW assumption implies that the discrete logarithm problem is hard. It supposes that there is an adversary \( \Psi \) who creates a valid forgery with probability \( \Psi \) (k), and claims two forger types. It shows that both of these types of forgery contradict the LRSW assumption. So the Signature Scheme B is secure under the LRSW assumption.

VI. BILINEAR PAIRING BASED REMOTE ATTESTATION

Three main categories mathematical problems used to structure a safe and effective public-key cryptographic algorithm currently: issue on large integer factorization (RSA system); discrete logarithm problem based on the finite field multiplication group (EIGamal system); discrete logarithm problem based on elliptic curve (ECC system). We have adopted bilinear pairing which is based on the elliptic curve discrete logarithm to overcome the shortcoming of existing remote attestation model [10], and achieve a new remote attestation model by making some modify of it. Remote attestation protocol in this article includes the following entities:

- **Verifier:** verify the platform information (such as platform configuration, attribution and so on).
- **Attestor:** take measurement on the platform, store platform information and verify the information to challenger.
- **Issuer:** party who is fully trusted, i.e. by the attestor and the verifier, confirms the correctness of the correspondence between the platform configuration and certain properties according to defined criteria, and produces corresponding attribute certificate.
- **Attribute Certification—AC:** Description some aspect of an entity (a platform or application), such as security-related needs and to ensure that the form of a certificate of authenticity properties.

A. The setup parameter

To set the system up we need to select parameters for each protocol and algorithm used within our scheme. On input of the security parameter \( \ell \) the algorithm can be described as below. Note that the group order \( q \) is selected so that solving the decisional Diffie–Hellman problem in \( G_1, G_2 \), as does solving the appropriate bilinear Diffie–Hellman problem with respect to the pairing.

1. Generate the Commitment Parameters \( \text{par}_1 \): sufficiently large prime order \( q \) for \( G_1 \). Random generator is selected such that \( G_1 \leftarrow P \_i \) along with a pairing \( \ell : G_1 \times G_1 \rightarrow \Psi \). Next a hash function \( H_1 : \left\{ 0,1 \right\}^* \rightarrow \mathbb{Z}_q \) is set to be \( \left( G_1, G_2, \ell, P_1, q, H_1 \right) \).
2. Generate the Rogue List Parameters \( \text{par}_2 \): Two hash functions \( H_2 : \left\{ 0,1 \right\}^* \rightarrow \mathbb{Z}_q \) are selected.
3. Generate the Issuer Parameters \( \text{par}_3 \): For each \( i \) the following is performed. Three integers are selected \( x, y, z \leftarrow \mathbb{Z}_q \), the issuer secret key \( \text{isk}_i \) is assigned to be \( \left( x, y, z \right) \). \( X = xP_1 \in G_1 \), \( Y = yP_1 \in G_1 \). \( Z = zP_1 \in G_1 \). The issuer public key \( \text{ipk}_i \) is set to be \( \left( X, Y, Z \right) \).
4. Publish Public Parameters \( \text{par} \): Finally, the system public parameters \( \text{par} \) are set to be \( \left( \text{par}_1, \text{par}_2, \text{par}_3 \right) \) and published.

B. Attribute-Configuration credential protocol

Attribute –Configuration credential protocol is a protocol between TPM: \( m \in \mathbb{M} \), the corresponding Host: \( h \in \mathbb{H} \) and an Issuer: \( i \in \mathbb{I} \). There are 3 main stages to an Attribute-Configuration credential protocol. First the TPM \( m \) generates some secret message \( f \) using the value \( C_s \) provided by the attestor. The TPM then computes a commitment on this value and passes both the commitment and the value \( C_s \) to the Issuer. In the second stage the issuer performs some checks on the commitment it receives and, if this correctly verify, computes attribute value \( p \) of the platform with some TPM, and achieve attribute credential (AC) by it. AC is encrypted under a public key corresponding to the TPM endorsement key \( EK \) and delivered to TPM. The final stage of a Attribute-Configuration credential protocol involves the Host and TPM working together to verify the correctness of the credential. In our case the Host first performs some computations and stores some values related to these before passing part of the credential on to the TPM prior to verifying the correctness of the credential and then adding this to the list of credentials for that user.

Our protocol proceeds as shown in Fig. 3. The following notes should be born in mind when examining this protocol.

- If the points \( P_1, X, Y, Z \) are not formed correctly then this could leak information about the value of a given \( C_s \). For example due to small subgroup attacks. To prevent this from happening each TPM needs to verify that \( P_1 \) generates \( G_1 \), and that \( X, Y, Z \subseteq G_1 \). This need be done once for each [7].
- Issuer should validate cert (AIKpub).Quote.nz before achieve AC, and compute the value of \( C_s \) consequently. And then Issuer compare the value of \( C_s \) and SML, determine whether the platform configuration is modified or not and compute the value of Ps according to \( C_s \) if there is no modification occur. Issuer is supposed trusted here, and will not disclose the value of \( C_s \).
- The value of AC is not sent in the clear and only the intended user can obtain the complete credential. This is done by encrypting the value AC.
under a public key corresponding to the TPM endorsement key EK.

- In contrast to the RSA-based remote attestation schemes we do not require a relatively complicated proof of knowledge of the correctness of a given commitment. Instead, the proof of knowledge is provided by a very efficient Schnorr signature, on the value F computed using the secret key Cs.

- The verification algorithm in Host is provided by Camenisch Lysyanskaya signature scheme B [16], signature is independent of the message Ps.

- Note that the Host does not perform any verification on values that are provided by the TPM. Since we assume that it is harder to compromise a TPM.

- The values $\rho_1, \rho_2, \rho_3, \rho_4$ and E are stored for later use by the Host in the Attribute-Configuration protocol. This improves the performance by avoiding recomputation of various pairing values. And even if the Host is compromised, attacker can not make use of the values stored in the Host, just because that the Host have no ideal of the secret value of Cs.

C. The Sign protocol

This is a protocol shown in Fig. 4 run between a given TPM and Host. They work together to produce a signature $\sigma$ of knowledge on some message (such as AC and E).

The signature was computed for the value of Cs. Verifier computation and will be able to take on a lot of the computational workload.

- In most applications of the Sign protocol, the signature is generated as a request from the verifier, and the verifier supplies its own value of $n_V$, to protect against replays of previously requested signatures. If a signature is produced in an offline manner we allow the Host to generate its own value of $n_V$.

- During the run of the signature protocol two nonce are used: one from the verifier $n_V$ and one from the TPM $n_T$. These are used to ensure each signature is different from previous controlled TPM and Host pair or no honest TPM and adversarial controlled Host can predict or force the value of a given signature.

- The value $r'$ is used to mask the signature created from the others in the protocol including the issuer. Without using $r'$ the credential on which the signature is computed would be sent in the clear and hence other parties would be able to link signatures.

D. The verification algorithm

This is an algorithm run by a verifier. Intuitively the verifier checks that a provided signature proves knowledge of a discrete logarithm Cs, and checks that it proves knowledge of a valid credential issued on the same value of Cs. Verify algorithm performs the following steps:

1. Check Correctness of Ab and B. if $\ell = (a, z) = \langle (P, A) \rangle$, or if $\ell = (a, b) = \langle (P, B) \rangle$, then return reject.

2. Verify the platform identification and Verify Correctness of Proofs. This is done by performing the following sets of computations: Figure out $s'$ from the signature $s$ by use AIK and $n_s$;

   $\rho_2' \leftarrow \ell (X, a); \rho_3' \leftarrow \ell (X, b); \rho_b' \leftarrow \ell (X, B', \rho_2'); \tau' \leftarrow \ell (\rho_2' \gamma \sigma (\rho_3' \gamma \sigma (\rho_1')) \pi (D) \leftarrow \ell (X, B' - \omega E);$

   $\omega' \leftarrow H_1(a || b || B || C || D || E || \rho_2 || \rho_3 || \rho_b || \tau || n_s || \rho_1);$

   if $\omega \neq H_1(\omega || n_s)$, then return reject and otherwise return accept.

VII. SECURITY RESULTS

Theorem 2. Attribute-Configuration credential protocol described above is correct and secure under the LRSW assumption.

Proof: We will first show correctness. The first verification equation in the host holds as:

$$\ell (a, Z) = \ell (a, Z || p_s) = \ell (a, p_s) = \ell (A, p_s);$$

$$\ell (a, Y) = \ell (a, Y || p_s) = \ell (a, p_s) = \ell (b, p_s);$$

$$\ell (A, Y) = \ell (A, Z || p_s) = \ell (A, p_s) = \ell (b, p_s);$$

$$\ell (A, c) = \ell (A, x || y || p) = \ell (A, x || y || p) = \ell (b, p);$$

Since our Attribute-Configuration credential protocol is based on Camenisch Lysyanskaya signature scheme B [16], which has proved secure under the LRSW assumption, our scheme is secure under the LRSW assumption.

Theorem 3. Verification algorithm described above is correct and secure under the LRSW assumption.

Proof: The correctness of the verification algorithm. If we can compute $r' = r$, we are done.

$$r' = \ell (P, A) \gamma (P, B) \pi (P, c) \gamma (P, b) \pi (P, y || y || y || z || r || F)$$

Since our Attribute-Configuration credential protocol is based on Camenisch Lysyanskaya signature scheme B [16], which has proved secure under the LRSW assumption, our scheme is secure under the LRSW assumption.
Since the verification algorithm is based on Camenisch Lysyanskaya signature scheme B [16], which has proved secure under the LRSW assumption, our scheme is secure under the LRSW assumption.

VIII. CONCLUSIONS

Table II presents the costs of bilinear parings-based attestation protocol.

Operations in \( G_T \) can be made slightly more efficient than those in \( G_N \) as in \( G_T \). What's more the operations in \( G_1 \) are about 1/4 the cost of operations in \( G_T \) [21]. Table I presents the performance analysis of our optimized version of the pairing based remote attestation protocol

- Due to DDH being hard in \( G_1 \) we can remove a number of the checks and masks in the original property based remote attestation protocol. And a number of important values are stored for later use by the Host in the Attribute-Configuration protocol. This improves the performance by avoiding recomputation of various pairing values. \( \tau \) is executed by Host not TPM, so the operation cost for TPM is much less then the old version.

- Our model satisfies all the characteristics except the condition 2 mentioned above, and makes full use of the special keys in TPM, such EK and AIK, to verify identification; Applies random nonce (such as \( n_T,n_V \)) to resist replay attacks, and ensure that the information of the fresh. Adopts some efficient algorithm to mask the credential, other parties would not be able to link signatures.

- Bilinear pairings is based on elliptic curve cryptography, one of its significant advantages are that with respect to the RSA key, the bilinear pairings can use more shorter key length, so that we can use smaller bandwidth and memory requirements.

<table>
<thead>
<tr>
<th>TABLE II. THE COSTS OF BILINEAR PARINGS-BASED ATTENTION PROTOCOL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Attribute-Configuration protocol</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Sign Porotol</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Verification Algorithm</td>
</tr>
</tbody>
</table>

In conclusion we have presented a remote attestation protocol based on pairings. Our protocol can be proved secure in the random oracle model under LRSW assumption. The signature was computed for the value of \( c_i \) in sign protocol. Verifier attests whether a platform or an application fulfills the desired value of \( p_i \) without revealing the specific software or/hardware configuration by the signature. The host knows a lot of the values needed in the computation and will be able to take on a lot of the computational workload. TPM and host must reduce \( \sigma \) every time the verifier send authentication request. However, if some verifier sends authentication requests to an attester that he has done it before, the latter can be confirmed in accordance with specific condition to determine whether to re-generate a new masked certificate, so as to reduce the system cost calculation. In addition to the protocols provided above, revocation algorithm is necessary. If for any reason, e.g., due to system security updates, a set of configuration specifications becomes invalid, the corresponding certificate needs to be removed, which requires verifier verify the validity of the certificate as well as the platform attributes. These are the work of the next work to be addressed.

REFERENCES

Li Xiaoyong, Zuo Xiaodong, and Shen Changxiang.

Engineer in East China University of Political Science and Law.

Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1992. Extended abstract


Ting Chen Hunan, China, 18 FEB 1979. Ph.D. Candidate, East China University of Science and Technology. Research Interests: Information security, trusted computing ;Master Degree in computer science, Huaqiao University, Fujian, China, 2004; Bachelor Degree in Computer Science, Hunan Normal University, Hunan, China, 2001. She is an engineer in East China University of Political science and Law.

Huiqun Yu Jiangsu, China, 6 February, 1967. Ph.D in Computer Science. His research interests include information security, software architecture, design techniques for real-time reactive systems, formal methods and trusted computing. Dr. Yu is a senior member of the IEEE and CCF, and a member of the ACM.