Applied Mathematics & Information Sciences
An International Journal

Research on the encryption and digital signatures in remote attestation from elliptic curve group

Ting Chen[1,2] *and Huiqun Yu*[1,3]

[1] Department of Computer Science and Engineering, East China University of Science and Technology Shanghai 200237, China
[2] Informatization Office, East China University of Political Science and Law Shanghai 201620, China
[3] Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China

Received: 23 Sep. 2012, Revised: 17 Dec. 2012, Accepted: 1 Jan. 2013
Published online: 1 May 2013

Abstract: Bilinear pairings based on the Weil and Tate pairings over elliptic curves have been applied for constructive applications in cryptography protocol for years. Most protocol can be proved to be simplified or expanded using the mathematical structures of different types of pairings.In this paper,we applied parings to a remote attestation model,namely cloud based remote attestation(CBA).We give all the detailed algorithms of it and it can guarantee the private of cloud service and solve authorization auditing mechanisms in cloud environment. The bilinear pair can shorten the required key length and reduced bandwidth usage. It meets the requirements of the trusted computing remote attestation and cloud environment at the same time and the virtual TPM structure fulfills the need of standard cloud computing secure measure, such as duty separation. What's more, we prove the scheme is correct and secure under the LRSW assumption, and give the costs comparison between the classic remote attestation, BPBA and CBA which show CBA costs lowly.

Keywords: bilinear paring;elliptic curve group; Trusted Computing; Cloud Computing; remote attestation.

1. Introduction

A new age dawned with the invention of the Cloud computing. Cloud Security Alliance (CSA) describes Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services).Gartner defines cloud computing as "a style of computing where scalable and elastic IT-related capabilities are provided 'as a service' to external customers using Internet technologies [22]." Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services [23,24]. There are three typical cloud service delivery models: Software as a service (SaaS) and platform as a service (PaaS) and infrastructure as a service (IaaS). SaaS offerings are typically implemented as Web applications, while PaaS offerings provide development and runtime environments for Web applications and services. For IaaS offerings, administrators typically implement associated services and APIs, such as the management access for customers, using Web application/service technologies. Cloud services are often utilized in conjunction with virtualization technologies. Virtualization is one of the key elements of IaaS cloud offerings and private clouds, and it is increasingly used in portions of the back-end of PaaS and SaaS providers as well. Virtualization is also, naturally, a key technology for virtual desktops, which are delivered from private or public clouds.However, when the existing computing environment changes to cloud environment, there are some issues to be solved. [5,29] have research on the cloud computing vulnerabilities. The most important factor for Cloud Computing is the security interoperability which needs proper authorization and auditing mechanisms for each cloud service. Since the service provides and the customers in the Cloud Computing architecture are different, as well as each cloud service have different role for different participant, duty separation should be well done in Cloud Computing.

Remote attestation is the process of vouching for the accuracy of information. It's a sound approach to relieve some Cloud Computing vulnerabilities. [2] gave the detail standard of this technology. It is based on the trustworthy computing which should provide some external entities

* Corresponding author e-mail: yhq@ecust.edu.cn

such as trusted platform module (TPM) and some special key (include attestation identity key, AIK). TPM 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. 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.

2. 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].Since all the attestation model mentioned above should reveal the system specific configuration, it may lead to privacy violations and discrimination against the underlying system since the remote party may exclude them from his/her business model. 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.

Property-based remote attestation model, which we called classic remote attestation in the last paper, conceals the configuration information of the system platform [9, 10], thus avoiding leakage of the system configuration. Because of security needs, the required RSA key length is too long, and it must to spend a great deal computation of RSA key for TPM consequently. [27,28] propose a remote attestation solution based on the properties of bilinear pairings named BPBA, with respect to the RSA key, the bilinear pairings can use shorter key length, so that model can use smaller bandwidth and memory requirements. But it cannot meet the need of cloud.

[25] designed and implemented a system that provides trusted computing functionality to every virtual machine on a virtualized hardware platform. Since the vTPM manage has full control of all the vTPM, it cannot meet the cloud security.

[26] provided a multi-tenancy trusted computing environment model for IaaS delivery model which conclude the attestation mechanism, but it didn't give a detail here.

[1] introduced attestation of trust platform into cloud computing service using the thought of property-based remote attestation. But it doesn't provide the concrete algorithm for its protocol and think about the virtual environment of cloud computing.

In this paper, firstly, a new construction of the cloud based remote attestation model (CBA) is proposed. CBA can guarantee remote attestation policies enforcement strictly by providing and maintaining trusted working environment in cloud environment. And in our method, the TPM function is executed by vcTPM which is running in Guest OS and vcTPM manage which is running in the cloud infrastructure, the vcTPM is associated with vcTPM manage but doesn't be controlled by the later completely and each tenant OS has different vcTPM, so the secure duty separation for cloud computing can be executed well here. Secondly, we give detailed description of all the algorithms in the protocol, which costs lowly and can satisfy the Cloud Computing requirements. The scheme has some modification on the classic remote attestation [10] and BPBA [27,28], and meets the requirements of the Trusted Computing and cloud environment at the same time. We prove the scheme is correct and secure under the LRSW assumption. Lastly, the prototype implementation of the model is presented and we give the costs comparison between the classic remote attestation [10], BPBA [27,28] and CBA which are all based on property remote attestation. CBA is designed for IaaS delivery model, and its purpose is to assure a trusted cloud infrastructure to customers.

3. Definitions and building blocks

A. Bilinear Pairing

We let $\hat{t} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ denote a pairing between three groups of prime order $q : \mathbb{G}_1$ (resp. $\mathbb{G}_2$) is a cycle additive group while $\mathbb{G}_T$ is a cycle multiplicative group. We let the generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) be denoted by P_1 (resp. P_2).

(1) Bilinearity

For all $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $a,b \in \mathbb{Z}_q$, we have $\hat{t}(aP,bQ) = \hat{t}(P,Q)ab$ and $\hat{t}(P_1+P_2,Q) = \hat{t}(P_1,Q)\hat{t}(P_2,Q)$.

(2) Non-degeneracy

• For all $P \in \mathbb{G}_1$, with $P \neq 0$, there is some $Q \in \mathbb{G}_2$ such that $\hat{t}(P,Q) = 1$.

• For all $Q \in \mathbb{G}_2$, with $Q \neq 0$, there is some $P \in \mathbb{G}_1$ such that $\hat{t}(P,Q) = 1$.

Bilinear pairings can be derived from the general elliptic curve of Weil or Tate while $\mathbb{G}_1 \neq \mathbb{G}_2$. It needs to build in three different groups, the application is very inconvenient. At the condition $\mathbb{G}_1 = \mathbb{G}_2$, 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 $\hat{t}$. Maintaining the Integrity of the Specifications.

B. 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 $\hat{t}:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ denote a pairing between three groups of prime order q. We let the generator of $\mathbb{G}_1$ 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 $(X,Y,Z)\in\mathbb{G}_1\times\mathbb{G}_1\times\mathbb{G}_1$ 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 $a\leftarrow G$.

• 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: $\hat{t}(a,Z)=\hat{t}(P_1,A)$.

2. b and B were formed correctly: $\hat{t}(a,Y)=\hat{t}(P_1,b)$ and $\hat{t}(A,Y)=\hat{t}(P_1B)$.

3. c was formed correctly: $\hat{t}(X,a)\cdot\hat{t}(X,b)^m\cdot\hat{t}(X,B)^r=\hat{t}(P_1,c)$.

Note that the values $(mZrP_1,a,A,b,B,c)$ 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

$\hat{t}(a,Z)=\hat{t}(a,P_1)^z=\hat{t}(P_1,a)^z=\hat{t}(P_1,A)$.

The two second ones hold as

$\hat{t}(a,Y)=\hat{t}(a,P_1)^y$

$=\hat{t}e(P_1,B)$ and $\hat{t}(A,Y)$

$=\hat{t}(a,P_1)^{zy}$

$=\hat{t}(P_1,Ay)$

$=\hat{t}(P_1,B)$.

The third one holds because

$\hat{t}(X,a)\cdot\hat{t}(X,b)^m\cdot\hat{t}(X,B)^r$

$=\hat{t}(P_1,a)^x\cdot\hat{t}(P_1,a)^{myx}\cdot\hat{t}(P_1,a)^{zrxy}$

$=\hat{t}(P_1,a)^{x+myx+zrxy}$

$=\hat{t}(P_1,ax+myx+zrxy)$

$=\hat{t}(P_1,ax+myxAxyr)$

$=\hat{t}(P_1,c)$.

[16] proved 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 $\Re$ 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.

C. Cloud Computing Architecture

Fig. 1 shows a cloud reference architecture that makes the most important security-relevant cloud components explicit and provides an abstract overview of cloud computing for security issue analysis. Encompass one or more service components. Here, service might be both material (such as shelter, power, and hardware) and immaterial (such as a runtime environment). For two layers, the cloud software environment and the cloud software infrastructure, the model makes the layers' three main service components-computation, storage, and communication-explicit. Top layer services also can be implemented on layers further down the stack, in effect skipping intermediate layers. For example, a cloud Web application can be implemented and operated in the traditional way-that is, running on top of a standard OS without using dedicated cloud software infrastructure and environment components. Layering and compositionality imply that the transition from providing some service or function in-house to sourcing the service or function can take place between any of the model's layers. The reference architecture is based on work carried out at the University of California, Los Angeles, and IBM.

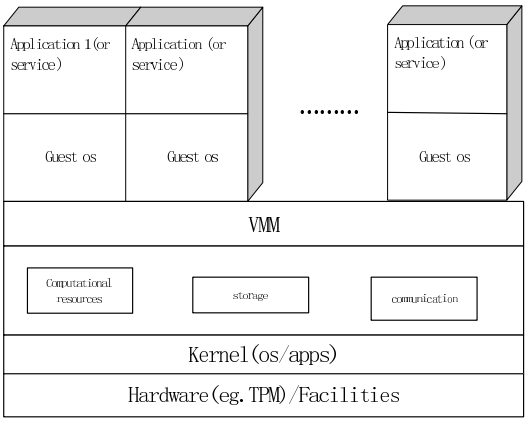

Figure 1 Cloud Computing architecture.

4. Cloud based remote attestation(CBA)

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 1^t the algorithm executes the following:

1. Generate the Commitment Parameters par_c: sufficiently large prime order q for $\mathbb{G}_1$. Random generator

is selected such that $\mathbb{G}_1 =< P_1 >$ along with a pairing $\hat{t} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Next a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$. Par_c is set to be $(\mathbb{G}_1, \mathbb{G}_T, \hat{t}, P_1, q, H_1)$.

2. Generate the Rogue List Parameters par_s: Two has functions $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ are selected. par_s is set to be (H_2, H_3).

3. Generate the Issuer Parameters par_I: For each I_k the following is performed. Three integers are selected $x, y, z \leftarrow \mathbb{Z}_q$, the issuer secret key isk_k is assigned to be $(x, y, z) \cdot X = x \cdot P_1 \in \mathbb{G}_1, Y = y \cdot P_1 \in \mathbb{G}_1, Z = z \cdot P_1 \in \mathbb{G}_1$, the issuer public key ipk_k is set to be (X, Y, Z).

4. Publish public parameters par: Finally, system public parameters par are set to be (par_c, par_s, par_I) and published.

The group order q is selected so that solving the decisional Diffie-Hellman problem in $\mathbb{G}_1$, $\mathbb{G}_T$, as does solving the appropriate bilinear Diffie-Hellman problem with respect to the pairing.

B. The Improved Cloud Computing Architecture With vcTPM Structure

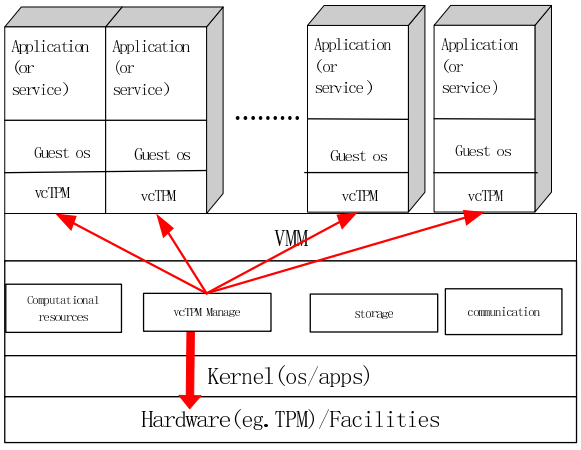

Figure 2 Our Cloud Computing architecture with vcTPM, the vcTPM manage.

[25] present the design and implementation of a system that enables trusted computing for an unlimited number of virtual machines on a single hardware platform and virtualized the Trusted Platform Module(TPM). As a result, the TPM's secure storage and cryptographic functions are available to operating systems and applications running in virtual machines. The owner of the virtual machine possesses the vTPM manage which manages all the vTPMs that disobey standard cloud computing secure measure, such as duty separation. We provide a new structure that the tenant controls their own vcTPM and the provider owns the vcTPM manage which distribute a vcTPM instance to some tenant when a new virtual system is built in the cloud. Fig. 2 shows our architecture where TPM functionality for all VMs is provided by a vcTPM manage sevice running in the cloud software infrastructure layer. TPM functionality for this VM is provided by the hardware TPM, and is used in the same way as in a system without a hypervisor where the operating system owns the hardware TPM. Every VM owns a unique vTPM instance. The vTPM instance number is prepended on the vcTPM manage so that virtual machines cannot forge packets and try to get access to a vTPM instance that is not associated with them. A command's originating virtual machine can be determined from the unique interrupt number.

C. AIK Certification

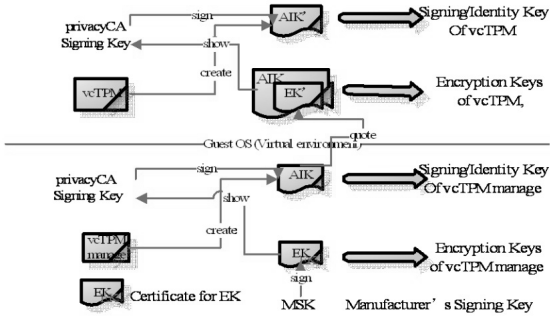

Figure 3 Certification of Endorsement Key using an AIK.

A certificate authority, i.e., a privacy CA, bases its decision to certify an AIK of a vcTPM manage on the certificate of the EK that a manufacturer provides along with the device. This certificate vouches for the vcTPM manage being a hardware device and that it is firmly attached to the firmware of the cloud infrastructure layer. Since the availability of an EK certificate plays this important role in receiving a certificate for AIKs, the EK certificate should also be available to a virtual TPM vcTPM instance even if it does not stand for the same security guarantees as those provided by a vcTPM manage. However, vcTPM can be dynamically created whenever a new VM is created, and therefore requests for EK certificates can become more frequent and their management becomes much more dynamic. [25] have found several solutions for the creation of EK certificates, each having advantages and disadvantages. We use the solution as follow: creates a certificate chain by connecting the certificate issued for the EK of a virtual TPM instance to that of an AIK of the hardware TPM. Fig. 3 depicts this relationship. It shows that a privacy CA

Issuer(I):	$\mathrm{n}_Z \leftarrow \{0,1\}^t$; $\mathrm{Comm}_{\mathtt{req}} \leftarrow \mathrm{n}_Z$; str$\leftarrow 1\|X\|Y\|Z\|\mathrm{n}_Z$
Issuer(I) $\rightarrow$ vcTPM(vm):	$\mathrm{Comm}_{\mathtt{req}}$
vcTPM(vm):	str$\leftarrow 1\|X\|Y\|Z\|\mathrm{n}_Z$; $u \leftarrow \mathbb{Z}_q$; $U \leftarrow u \cdot P_1$; $\mathrm{Comm}_{\mathtt{req}}$; get cert(AIK$\mathtt{pubvm}$); Quote=sign$\{\mathrm{Cs}_{vm}$,$\mathrm{n}_z\}\mathrm{AIK}_{\mathtt{privvm}}$; get SML_{vm};
vcTPM(vm)$\rightarrow$ vcTPM Manage(vmm)	Quote; SML_{vm}; cert($\mathrm{AIK}_{\mathtt{pubvm}}$)
vcTPM Manage(vmm)	If verify(cert($\mathrm{AIK}_{\mathtt{pubvm}}$))=false then abort; Cs=$\mathrm{Cs}_{vmm}\|\mathrm{Cs}_{vm}$; SML=$\mathrm{SML}_{vm} \parallel \mathrm{SML}_{vmm}$; Quote=sign$\{\mathrm{Cs},\mathrm{n}_z\}\mathrm{AIK}_{\mathtt{privvmm}}$; get cert($\mathrm{AIK}_{\mathtt{pubvmm}}$); $F \leftarrow \mathrm{Cs} \cdot P_1$; $s \leftarrow u + c \cdot Cs(mod)q$
vcTPM Manage(vmm)$\rightarrow$ vcTPM(vm)	Quote; SML; str$\|F$; s;
vcTPM(vm)	$c \leftarrow H_1$ (str$\|F\|U$); Comm(F,c,s)
vcTPM(vm)$\rightarrow$ Issuer(I):	Comm(F,c,s); sign$\{$Quote,$n_Z\}\mathrm{AIK}_{\mathtt{privvm}}$; SML; cert(AIK$\mathtt{pubvm}$); cert(AIK$\mathtt{pubvmm}$);
Issuer(I):	$U' \leftarrow sP1 - cF$; if $c \neq H_1(\mathrm{srt}\|F\|U')$ then abort; validate Quote; SML; cert(AIK_{pub}); figure out P_s $r \leftarrow \mathbb{Z}_q$; $a \leftarrow r \cdot P_1$; $A \leftarrow z \cdot a$; $b \leftarrow y \cdot a$;$B \leftarrow y \cdot A$; $c \leftarrow x \cdot a + x \cdot p_s \cdot b + x \cdot y \cdot z \cdot r \cdot F$; $AC \leftarrow (a,A,b,B,c,p_s)$; $\varepsilon \leftarrow E_{ekvm}(E_{ekvmm}(AC))$
Issuer(I)$\rightarrow$ VcTPM(vm):	ε
vcTPM(vm)$\rightarrow$ vcTPM manage(vmm)	$AC' \leftarrow E^{-1}_{ekvm}(\varepsilon)$
vcTPM manage(vmm)$\rightarrow$ vcTPM(vm)	$AC \leftarrow E^{-1}_{ekvmm}(AC')$; $E \leftarrow Cs \cdot B$
vcTPM(vm)$\rightarrow$ Guest OS(go):	AC, E
Guest OS(go):	$\rho_a \leftarrow \hat{t}(X,a)$; $\rho_b \leftarrow \hat{t}(X,b)$; $\rho_B \leftarrow \hat{t}(X,B)$; $\rho_c \leftarrow \hat{t}(P_1,c)$; if $\hat{t}(a,Z) \neq \hat{t}(A,P_1)$ or $\hat{t}(a,Y) \neq \hat{t}(b,P_1)$ or $\hat{t}(A,Y) \neq \hat{t}(B,P_1)$ or $\hat{t}(X,a) \cdot \hat{t}(X,b)^{p_s} \cdot \hat{t}(E,X) \neq \hat{t}(P_1,c)$ then abort

Figure 4 Attribute-Configuration Credential Protocol.

issues certificates for AIKs of a vcTPM based on the certificate of its endorsement key EK'. The advantage of this scheme is that we have preserved the normal procedure of acquiring an AIK' certificate by submitting the certificate of EK' to a privacy CA for evaluation. we are using an (attestation) identity key and the Quote command of vcTPM manage to issue a signature over the current state of PCRs and a user-provided 160bit number. We provide as 160bit number the SHA1 hash of the certificate contents of the EK'. The resulting signature ties this EK' certificate and the vcTPM instance to the underlying platform.

D. Attestation Algorithms

There are four algorithms in our scheme: Attribute-Configuration credential protocol; The Sign protocol; The verification algorithm; Revocation algorithm;

Attribute-Configuration credential protocol proceeds as shown in Fig.4. The protocol is act between vcTPM manage: vmm∈M,vcTPM: vm∈M, the corresponding Guest OS: go∈H and an Issuer: i∈I. There are 4 main stages to an Attribute-Configuration credential protocol. First the vcTPM vm transfer Cs_{vm}(the Cs value of vcTPM) and SML_{vm} to vcTPM manage. In the second stage the vcTPM manage verified AIK certification of the vcTPM and get its own Csvmm and SMLvmm. And then it generates some secret message f using the value Cs ($\mathrm{Cs}_{vm}\|$ Cs_{vmm}) provided by the attestor. The vcTPM

Manage pass the value to vcTPM as well as SML ($SML_{vm} \| SML_{vmm}$). In the third stage vcTPM computes a commitment on this value and passes both the commitment and the value Cs to the Issuer.In the forth stage the issuer performs some checks on the commitment it receives and, if this correctly verify, computes attribute value ps of the platform with some

vcTPM, and achieve attribute credential (AC) by it. AC is encrypted in turn with a public key corresponding to the vcTPM manage endorsement key EK_{vmm} and the vcTPM endorsement key EK_{vm} delivered to vcTPM. The final stage of a Attribute-Configuration credential protocol involves the Guest OS, vcTPM and vcTPM manage working together to verify the correctness of the credential. In our case the Guest OS go first performs some computations and stores some values related to these before passing part of the credential on to the vcTPM prior to verifying the correctness of the credential and then adding this to the list of credentials for that user.

The Sign protocol is a protocol shown in Fig. 5 run between a given vcTPMvcTPM manage and Guest OS, They work together to produce a signature σ of knowledge on some message (such as AC and E). The signature was computed for the value of Cs. Verifier attest whether a platform or an application fulfills the desired value of ps without revealing the specific software or/and hardware configuration by the signature. We note that the Guest OS will know a lot of the values needed in the computation and will be able to take on a lot of the computational workload.

The verification algorithm 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 A, b and$B$. if $\hat{t}(a^{'}, Z) \neq \hat{t}(P_1, A^{'})$, or if $\hat{t}(a^{'}, Y) \neq \hat{t}(P_1, b^{'})$ an $\hat{t}(A^{'}, Y) \neq \hat{t}(P_1, B^{'})$,then return reject.

2. Verify platform identification and verify correctness of Proofs. This is done by performing the following sets of computations:

Figure out s° from the signature by $s^{'}$use AIK and

n_T; $\rho_{a^{\circ}} \leftarrow \hat{t}(X, a^{'})$; $\rho_{b^{\circ}} \leftarrow \hat{t}(X, b^{'})$; $\rho_{B^{\circ}} \leftarrow \hat{t}(X, B^{'})$; $\rho_{c^{\circ}} \leftarrow \hat{t}(P_1, c^{'})$; $\tau^{\circ} \leftarrow (\rho_{B^{\circ}})^{s^{\circ}} \cdot (\rho_{c^{\circ}} / \rho_{a^{\circ}})^{-\omega^{'}} \cdot \rho_{b^{\circ}}^{\omega \cdot p_s}$; $D^{\circ} \leftarrow s^{circ} \cdot B^{'} - \omega^{'} \cdot E^{'}$; $\omega^{\circ} \leftarrow H_2(a^{'} \| A^{'} \| b^{'} \| B^{'} \| c^{'} \| D^{'} \| E^{'} \| \rho_{a^{'}} \| \rho_{b^{'}} \| \rho_{B^{'}} \| \rho_{c^{'}} \| \tau \| n_v \| p_s)$

If $\omega^{'} \neq H_3(\omega^{\circ} n_T)$ then return reject and otherwise return accept.

The revocation algorithm is presented in BPBA[27, 28].

5. A prototype implementation

The research of Digital Rights Management (DRM) has stepped into cloud era. [4] give the model for a TPM-Based DRM, which can also be easily deployed in the cloud architecture. We act the content server and license server as two services in cloud computing architecture, and take the DRM agent as an attestor to request for remote attestation of the license service and content service.

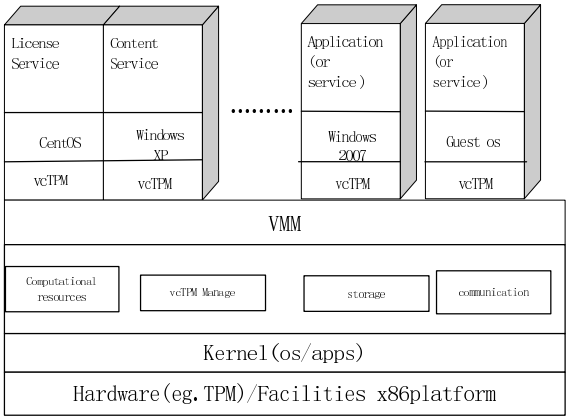

Figure 6 A Prototype of CBA.

We have implemented a prototype of CBA shown in Fig. 6. In the prototype system, the host platform is a x86 based server with a configuration of: Core 2 Quad Q8200 CPU, 8G memory and 500G hard disk. Several virtual instances were created on the platform: one used CentOS as guest OS and provided license service for DRM, and the others used Windows XP or Windows 2000/2003/2007 as the guest OS acted as content service to provide content .

The security duty separation is assured by two independent security management channels: one is vcTPM manage to manage infrastructure, and the other is for customers to manage their own virtual instance by vcTPM.

6. 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:

$\hat{t}(a, Z) = \hat{t}(a, z \cdot P_1) = \hat{t}(a, P_1)^z = \hat{t}(z \cdot a, P_1) = \hat{t}(A, P_1)$;
$\hat{t}(a, Y) = \hat{t}(a, y \cdot P_1) = \hat{t}(a, P_1)^y = \hat{t}(a \cdot y, P_1) = \hat{t}(b, P_1)$;
$\hat{t}(A, Y) = \hat{t}(A, y \cdot P_1) = \hat{t}(y \cdot A, P_1) = \hat{t}(B, P_1)$;
$\hat{t}(P_1, c) = \hat{t}(P_1, x \cdot a + x \cdot p_s \cdot b + x \cdot y \cdot z \cdot r \cdot F)$

Guest OS(go)→ vcTPM(vm):	bsn
vcTPM(vm) → vcTPM Manage(vmm)	Csvm
vcTPM Manage(vmm) → vcTPM(vm)	Cs= $Cs_{vmm} \parallel Cs_{vm}$; $r' \leftarrow H_2(Cs)$; $v \leftarrow \mathbb{Z}_q$; $D' \leftarrow (vr') \cdot B$
vcTPM(vm) → Guest OS (go):	r', D'
Guest OS(go):	checkbsn(), if return $\perp$ then stop; $n_v \leftarrow \{0,1\}^t$; $a' \leftarrow r' \cdot a$; $A' \leftarrow r' \cdot A$; $b' \leftarrow r' \cdot b$; $B' \leftarrow r' \cdot B$; $c' \leftarrow r' \cdot c$; $\rho_{a'} \leftarrow \rho_a^{r'}$; $\rho_{b'} \leftarrow \rho_b^{r'}$; $\rho_{B'} \leftarrow \rho_B^{r'}$; $\rho_{c'} \leftarrow \rho_c^{r'}$; $\tau \leftarrow \hat{t}(X, D')$; $E' \leftarrow r' \cdot E$; $\omega \rightarrow H_2(a' \| A' \| b' \| B' \| c' \| D' \| E' \| \rho_{a'} \| \rho_{b'} \| \rho_{B'} \| \rho_{c'} \| \tau \| n_v \| p_s)$
Guest OS(go) ' vcTPM (vm):	ω
vcTPM(vm) → vcTPM Manage(vmm)	ω
vcTPM Manage(vmm) → vcTPM(vm)	$n_T \leftarrow \{0,1\}^t$; $\omega \leftarrow H_3(\omega) \| n_T$; $s \leftarrow v + \omega \cdot Cs(modq)$; $s \leftarrow \mathtt{sign}\{s, n_{n_T}\}_{AIK_{priv}}$
vcTPM (vm) → Guest OS (go):	$\omega', n_T, s'' \leftarrow \mathtt{sign}\{s', n_T\}_{\mathtt{AIK}_{privvm}}$, Cert($\mathrm{AIK}_{pubvmm}$),Cert($\mathrm{AIK}_{pubum}$);
Guest OS(go):	$\sigma \leftarrow (a', A', b', B', c', D', E'), s', n_v, n_T, p_s, \mathtt{cert}(AIK)_{pub}$

Figure 5 Sign Protocol.

$$
\begin{aligned}
&= \hat{t}(P_1, x \cdot a)\hat{t}(P_1, x \cdot p_s \cdot b)\hat{t}(P_1, x \cdot y \cdot z \cdot r \cdot F) \\
&= \hat{t}(P_1, a)^X \hat{t}(P_1, b)^{x \cdot P_s} \hat{t}(P_1, F)^{y \cdot z \cdot r \cdot x} \\
&= \hat{t}(X, a)\hat{t}(X, b)^{p_s} \hat{t}(y \cdot z \cdot r \cdot P_1 \cdot C_s \cdot P_1)^x \\
&= \hat{t}(X, a)\hat{t}(X, b)^{p_s} \hat{t}(y \cdot z \cdot r \cdot P_1, x \cdot P_1)^{C_s} \\
&= \hat{t}(X, a)\hat{t}(X, b)^{p_s} \hat{t}(y \cdot z \cdot r \cdot P_1 \cdot C_s, x \cdot P_1) \\
&= \hat{t}(X, a)\hat{t}(X, b)^{p_s} \hat{t}(C_s \cdot y \cdot A, X) \\
&= \hat{t}(X, a) \cdot \hat{t}(X, b)^{p_s} \cdot \hat{t}(E, X)
\end{aligned}
$$

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.

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 , we are done.

$$
\begin{aligned}
&\tau^\circ = (\rho_{B^\circ})^{s^\circ} \cdot (\rho_{c^\circ} / \rho_{a^\circ})^{-\omega'} \cdot \rho_{b^\circ}^{\omega' \cdot p_s} \\
&= \hat{t}(X, B')^{s^\circ} \cdot (\hat{t}(X, a')^{\omega'} / \hat{t}(P_1, c')^{\omega'}) \hat{t}(X, b')^{\omega' \cdot p_s} \\
&= \hat{t}(X, B')^{s^\circ} (\hat{t}(X, r' \cdot a' \cdot \omega' + r' \cdot b' \cdot \omega' \cdot p_s) / \hat{t}(P_1, r' \cdot x \cdot a + r' \cdot x \cdot p_s \cdot b + r' \cdot x \cdot y \cdot z \cdot r \cdot F)^{\omega'}) \\
&= \hat{t}(X, B')^{s^\circ} (\hat{t}(X, r' \cdot a' \cdot \omega' + r' \cdot b' \cdot \omega' \cdot p_s) / \hat{t}(P_1, a + r' \cdot \omega' \cdot p_s \cdot b + r' \cdot \omega' \cdot y \cdot z \cdot r \cdot F)^x) \\
&= \hat{t}(X, B')^{s^\circ} (\hat{t}(X, r' \cdot a' \cdot \omega' + r' \cdot b' \cdot \omega' \cdot p_s) / \hat{t}(x \cdot P_1, r' \cdot \omega' \cdot a + r' \cdot \omega' \cdot p_s \cdot b + r' \cdot \omega' \cdot y \cdot z \cdot r \cdot F)) \\
&= \hat{t}(X, B')^{s^\circ} (\hat{t}(X, r' \cdot a' \cdot \omega' + r' \cdot b' \cdot \omega' \cdot p_s) / \hat{t}(X, r' \cdot \omega' \cdot a + r' \cdot \omega' \cdot p_s \cdot b + r' \cdot \omega' \cdot y \cdot z \cdot r \cdot F)) \\
&= \hat{t}(X, B')^{s^\circ} / \hat{t}(X, r' \cdot \omega' \cdot y \cdot z \cdot r \cdot F)) \\
&= \hat{t}(X, (v + \omega' \cdot C_s) \cdot r \cdot B) / \hat{t}(X, r' \cdot \omega' \cdot y \cdot z \cdot r \cdot C_s \cdot P_1)) \\
&= \hat{t}(X, v \cdot r' \cdot B) / \hat{t}(X, \omega' \cdot C_s \cdot r' \cdot B) / \hat{t}(X, r' \cdot \omega' \cdot C_s \cdot B) \\
&= \hat{t}(X, v \cdot r' \cdot B) \\
&= \hat{t}(X, D') \\
&= \tau
\end{aligned}
$$

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.

7. Conclusions

Table 1. presents the costs comparison between the classic remote attestation [10], BPBA[27,28] and CBA, with respect to each player.

An entry of the form: $1 \cdot \mathbb{Q}_N + 2 \cdot \mathbb{Q}_N^2 + 3 \cdot \mathbb{Q}_\Gamma$ implies that the cost is about one exponentiation modulo N, two modulo Γ and three multiexponentiations with two exponents modulo N, i.e. three operations of the form $g^a \cdot h^b \pmod N$. Note, that a multiexponentiation with m exponents can often be performed significantly faster than m separate exponentiations.

In the paper, $\mathbb{Q}_1$ denotes the cost of a exponentiation computation in $\mathbb{G}_1$; $\mathbb{Q}_1^m$ denotes the cost about a multiexponentiation of m values in $\mathbb{G}_1$;$\mathbb{Q}_T$ denotes the cost of a exponentiation computation in $\mathbb{G}_T$; $\mathbb{Q}_T^m$ denotes the cost about a multiexponentiation of m values in $\mathbb{G}_T$;

Table 1 THE COSTS COMPARISON BETWEEN THE CLASSIC REMOTE ATTESTATION, BPBA AND CBA

Operation	Party	Old one[10]	BPBA [27,28]	CBA	
Attribute-Configuration protocol	TPM		$3\cdot\mathbb{Q}_1$	vcTPM	$1\cdot\mathbb{Q}_1$
				vcTPM manage	$2\cdot\mathbb{Q}_1$
	Issuer	$3\cdot\mathbb{Q}_N^4$	$4\cdot\mathbb{Q}_1+1\cdot\mathbb{Q}_1^2+1\cdot\mathbb{Q}_1^3$	$4\cdot\mathbb{Q}_1+1\cdot\mathbb{Q}_1^2+1\cdot\mathbb{Q}_1^3$	
	Host		$11\cdot\mathbb{Q}_L+1\cdot\mathbb{Q}_L^2$	$11\cdot\mathbb{Q}_L+1\cdot\mathbb{Q}_L^2$	
Sign Porotol	TPM	$1\cdot\mathbb{Q}_P^2$	$1\cdot\mathbb{Q}_1$	vcTPm	
				vcTPM manage	$1\cdot\mathbb{Q}_1$
	Host	$2\cdot\mathbb{Q}_N+1\cdot\mathbb{Q}_N^3+1\cdot\mathbb{Q}_P^2$	$6\cdot\mathbb{Q}_1+4\cdot\mathbb{Q}_T+1\cdot\mathbb{Q}_L$	$6\cdot\mathbb{Q}_1+4\cdot\mathbb{Q}_T+1\cdot\mathbb{Q}_L$	
Verification Algorithm	Verifier	$1\cdot\mathbb{Q}_N^4+1\cdot\mathbb{Q}_P^3$	$10\cdot\mathbb{Q}_L+1\cdot\mathbb{Q}_1^2+1\cdot\mathbb{Q}_T^3$	$10\cdot\mathbb{Q}_L+1\cdot\mathbb{Q}_1^2+1\cdot\mathbb{Q}_T^3$	
Revocation algorithm	Host	$3\cdot\mathbb{Q}_P^1+1\cdot\mathbb{Q}_P^2$	$6\cdot\mathbb{Q}_1+4\cdot\mathbb{Q}_T+1\cdot\mathbb{Q}_L$ or 0	$6\cdot\mathbb{Q}_1+4\cdot\mathbb{Q}_T+1\cdot\mathbb{Q}_L$ or 0	
	Verifier	$2\cdot\mathbb{Q}_P^2+2\cdot\mathbb{Q}_P^3$	$\mathbb{Q}_1^2$	$\mathbb{Q}_1^2$	

$\mathbb{Q}_L$denotes the cost of a pairing computation, such as $\tau \leftarrow (X, D^{'})$; $\mathbb{Q}_L^m$ denotes the cost of m pairings computation; $\mathbb{Q}_N$denotes the cost about one exponentiation modulo N, such as g^a mod N; and $\mathbb{Q}_N^m$denotes the cost of a multiexponentiation with m exponents modulo N; $\mathbb{Q}_P^m$denotes the cost of a multiexponentiation with m exponents modulo P, where P is a large prime number, such as $g^a h^b$ mod P;

Operations in $\mathbb{G}_T$ can be made slightly more efficient than those G_N as in $\mathbb{G}_T$. What's more the operations in $\mathbb{G}_1$ are about 1/4 cost of operations in $\mathbb{G}_T$ [21]. Table 1 presents the performance performance analysis of our optimized version of the pairing based remote attestation protocol.

• Due to DDH being hard in $\mathbb{G}_1$ we can remove a number of the checks and masks in the classic property based remote attestation protocol. And a number of important values are stored for later use by the Guest OS in the Attribute-Configuration protocol. This improves the performance by avoiding recomputation of various pairing values. τ is executed by Guest OS not vcTPM or vc TPM manage, so the operation cost for vcTPM and vcTPM manage is much less then the classic version[10].

• We defined vcTPM and vcTPM manage to control the virtual system and the infrastructure of the cloud. Since vcTPM running in the virtual machine and vcTPM manage running in the cloud infrastructure layer, and the vcTPM manage manages P_1 which vcTPM doesn't know, while the guest OS is tampered, nobody can forge F to issuer. And the vcTPM manage doesn't know the parameters X, Y, Z issuer send to vcTPM , so the cloud provider can't pretend the vcTPM to complete all the algorithms. It fullfilled the need of standard cloud computing secure measure, such as duty separation. And we use the solution mentioned in [25] to produce EK and AIK of vcTPM.

• Our model makes full use of the special keys in vcTPM and vcTPMmanage, 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.

What's more, our protocol can be proved secure in the random oracle model under LRSW assumption. Our future work will focus on how the scheme be expanded to the mobile industry.

Acknowledgement

This work is partially supported by the NSF of China under grants No. 61173048, Shanghai Shuguang Program under grant No. 07SG32.

References

[1] Liqun Chen, Paul Morrissey, and Nigel P. Smart, Paring in trusted computing. S.D. Galbraith and K.G. Paterson (Eds.): Pairing 2008, LNCS 5209, pp. 1-17, 2008. Springer-Verlag, Berlin, Heidelberg 2008.

[2] Trusted Computing Group. TCG Architecture Overview (April 2004). http://www.trustedcomputing group.org.

[3] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, Design and implementation of a TCG-based integrity measurement architecture. In: Proceedings of the 13th USENIX Security Symposium. USENIX, Aug, 2004, pp. 223-238.

[4] Aimin Yu, Dengguo Feng and Ren Liu, TBDRM: A TPM-Based Secure DRM Architecture. Computational Science and Engineering, 2009, Vol(2), pp:671-677.

[5] Hyokyung Chang and Euiin Choi, Challenges and Security in Cloud Computing. In: 2010, Part II, CCIS 120, pp. 214-217, 2010. Springer,Heidelberg (2010).

[6] Zhang Huanguo and Wang fan, Behavior-Based Remote Trust Attestation Model. Wuhan University Journal of Nature Sciences, 2006, vol. **11**, no.7, pp.1819-1822.

[7] Liqun Chen., Z. Cheng, and N. Smart, N.P.: Identity-based key agreement protocols from pairings. Int. Journal of Information Security **6**, pp. 213-242. (2007)
[8] Vivek Haldar, Deepak Chandra, and Michael Franz, emantic Remote Attestation - Virtual Machine Directed Approach to Trusted Computing. In: Proceedings of the 3rd Virtual Machine Research and Technology Symposium, May 6-7, 2004, San Jose, CA, USA,2004, pp. 29-41.
[9] A. adeghi and C.Stble, Property-based attestation for computing platforms: caring about properties, not mechanisms. In:C. Hempelmann and V. Raskin, editors, Proceedings of the New Security Paradigms Workshop 2004, September 20-23, 2004, Nova Scotia, Canada, 2004, pp. 67-77.
[10] Liqun Chen, A Protocol for PropertyBased Attestation. In: Proceedings of the 1st ACM Workshop on Scalable Trusted Computing (STC), November 3, 2006, Alexandria, Virginia, USA. ACM. Nova Scotia Canada, 2006, pp.7-16.
[11] J. Camenisch and A. Lysyanskaya, A signature scheme with efficient protocols. In: Third Conference on Security in Communication Networks - SCN '02, volume **2576** of LNCS, pp. 268-289. Springer-Verlag, Berlin Germany, 2002.
[12] Torben Pryds Pedersen, Non-interactive, and information-theoretic secure verifiable secret sharing. In: J. Feigenbaum, editor, Advances in Cryptology - CRYPTO '91, volume 576 of LNCS, pp. 129-140. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1992. Extended abstract
[13] A. J. Menezes, Elliptic curve public key cryptosystems. Kluwer Academic Publishers, 1993.
[14] A. J. Menezes, T. Okamoto and S.Vanstone, Reducing elliptic curve logarithm to logarithms in a finite field.IEEE Transaction on Information Theory, vol. **39**, pp. 1639-1646, 1993.
[15] Anna Lysyanskaya, Ron Rivest, Amit Sahai, and Stefan Wolf, Pseudonym systems. InHoward Heys and Carlisle Adams, editors, Selected Areas in Cryptography, volume 1758 of Lecture Notes in Computer Science. Springer Verlag, 1999.
[16] J.Camenisch and A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. **3152**, pp. 56-72. Springer, Heidelberg. (2004)
[17] Li Xiaoyong, Zuo Xiaodong, and Shen Changxiang.. System Behavior Based Trustworthiness Attestation for Computing Platform. Chinese Journal of Electronics, 2007, vol. **35**, no. 7, pp. 1234-1239.
[18] Trusted Computing Group. TPM main specification. Main Specification Version 1.2 rev. 85, Trusted Computing Group,Feb. 2005.
[19] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, Oct. 2004. ACM Press.
[20] Dries Schellekens, Brecht Wyseur, and Bart Prenee.Remote Attestation on Legacy Operating Systems With Trusted Platform Modules. Electronic Notes in Theoretical Computer Science vol. **197**, pp. 59-72, 2008
[21] Barreto, P.S.L.M., and Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. **3897**, pp. 319-331. Springer, Heidelberg (2006)
[22] Gartner Says Cloud Computing Will Be As Influential As E-business (June 2008), http: //www.gartner.com/it/page.jsp?id=707508
[23] Armbrust, M., et al.: Above the Clouds: A Berkeley View of Cloud Computing. In: Technical Report No. UCB/EECS-2009-28 (2009), doi: http://www.eeec.berkeley.edu/Pubs/TechRpts/2009/ EEEC-2009-28.html.
[24] http://www.cloudtech.org/2010/07/19/cloud-computing-%E2%80%93-the-emerging-computing-technology/.
[25] BERGER SCACERES RGOLDMAN K A. vTPMvirtualizing the trusted platform module[R]. In Proc. of USENIX-SS'06, Berkeley, CA, USA, 2006.
[26] Xiao-Yong Li, Li-Tao Zhou, Yong Shi, Yu Guo: A trusted computing environment model in cloud architecture. ICMLC 2010: 2843-2848.
[27] Ting Chen, Huiqun Yu. Bilinear parings in property-based attestation. Journal of Computers, **6(2)**: 297-304, 2011.
[28] Ting Chen, Huiqun Yu. The improved research on property-based remote attestation. Proceedings of the International Conference on Computer Application and System Modeling (ICCASM), v. **14**, p.320-324, 2010
[29] Grobauer, B. Walloschek andT. Stocker, E Understanding Cloud Computing Vulnerabilities. Security & Privacy, IEEE, Vol. **9** Issue: 2, 2011, pp.50 - 57
[30] Liqun Chen, Paul Morrissey, and Nigel P. Smart, Paring in trusted computing. S.D. Galbraith and K.G. Paterson (Eds.): Pairing 2008, LNCS 5209, pp.1-17, 2008. Springer-Verlag, Berlin, Heidelberg 2008.

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, Hua qiao 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.