An improved trusted cloud computing platform model based on DAA and Privacy CA scheme

Wang Han-zhang[1], Huang Liu-sheng[2]
National High Performance Computing Center,
Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei, China
e-mail: [1]:fishspy@mail.ustc.edu.cn, [2]lshuang@ustc.edu.cn

***Abstract*—Security and privacy are two prime barriers to adoption of the cloud computing. To address this problem on Infrastructure-as-a-Service model, a trusted cloud computing platform model has been proposed to provide a closed box execution environment that guarantees confidential execution of guest virtual machines. However this model has significant drawbacks that it relies on the trusted third party outside of the cloud circumstance too much. In this paper we show how to address this issue based on the neutral feature of the Trusted Platform Module. By moving the responsibility of managing trusted platforms from the trusted third party to the trusted platforms of Infrastructure-as-a-Service model, our improved TCCP model achieves higher availability, reliability and safety.**

Keywords-trusted computing; cloud computing; privacy preserving; DAA scheme; anonymity

I. INTRODUCTION

Cloud computing, which is considered to be the next big trend of information age by many people, offers great benefits including: low upfront IT investments, pay-for-use model allows for reduced operating expenses, reduced complexity, etc. However organizations or companies have to upload their data or programs to the cloud, obviously security and privacy will be two significant barriers to adoption. Fig. 1 illustrates the impact of cloud computing on the governance structure of IT organizations [1], shows that organization loses the control of their resources increasingly from IaaS (Infrastructure-as-a-Service) model to SaaS (Software-as-a-service) model.

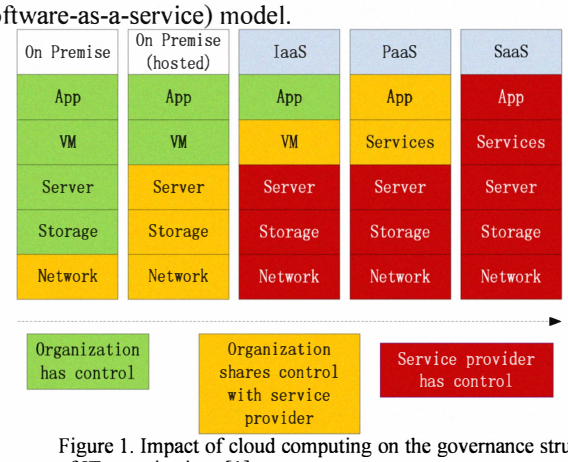


Figure 1. Impact of cloud computing on the governance structure of IT organizations [1].

N. Santos, K. P. Gummadi, and R. Rodrigue[2] showed how to leverage the advances of trusted computing technologies to design a *Trusted Cloud Computing Platform* (TCCP). They argued in IaaS cloud services such as Amazon's EC2, insiders that administer the software systems at the provider backend have the privilege or technical means such as using Xenaccess [3] to access the memory of a customer's VM, thus there is a need for ensuring the confidentiality and integrity of computation that are outsourced to IaaS services. Fig. 2 shows the components of the TCCP including untrusted *Cloud Manager* (CM) that offers the cloud services, *Trusted Nodes (TN)*, the backend of which run a *Trusted Virtual Machine Monitor* (TVMM) to prevent insiders from inspecting or modifying them and *Trusted Coordinator* (TC) that manages the set of trusted nodes in *External Trusted Entity* (ETE) maintained by a third party with little relevance to IaaS *Cloud Service Providers* (CSPs).

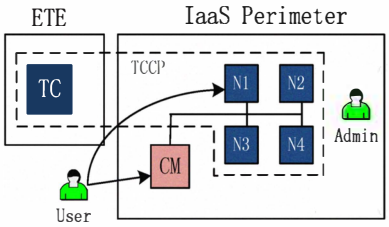


Figure 2. The components of the TCCP [2].

In this solution a node must register with the TC by certifying its Endorsement Key (EK) and Measurement List (ML_N). The TC maintains a list of trusted nodes. When launching a VM or in live migration, the TC gets involved in the protocol to make sure the node is trusted so that a customer's VM is always running on a trusted platform , it's the manifest drawback that every transaction needs the TC making it become a bottleneck (availability and trust in the TC). Moreover, if the TC and the insiders collude, the TPM will be identified uniquely by some means.

To overcome these problems, in this paper we show how to use Direct Anonymous Attestation (DAA) and Privacy CA scheme to improve the anonymity and availability of the TCCP model. Section II introduces these technologies associated with our model. Section III presents our model in detail.

978-1-4244-7237-6/$26.00 © 2010 IEEE

II. BACKGROUND

A. Eucalyptus architecture

IaaS providers such as Amazon EC2 allow customers to allocate entire virtual machines on demand. Eucalyptus, an open-source software implementation of IaaS, will be used to describe our model. Fig.3 depicts the architecture of Eucalyptus including a *Cloud Controller* (CLC), *Cluster Controllers* (CC) and *Node Controllers* (NC) and *Clients.* The CLC is the user-visible entry point and global decision-making component [4]. The CC is responsible for managing the clusters of nodes which is described as a "zone" (In Amazon EC2 or Eucalyptus a "zone" is correlated to a vague geographic location). The NC is the component that executes on the physical resources that host VM instances.

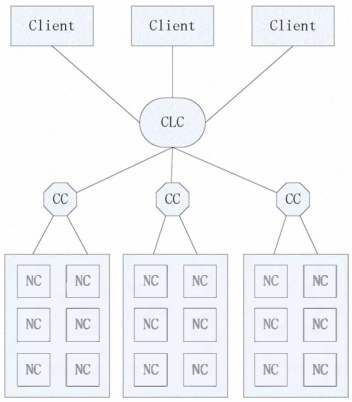


Figure 3. Eucalyptus employs a hierarchical design to reflect underlying resource topologies [4].

B. Direct Anonymous Attestation scheme

A Trusted Platform Module (TPM) chip can be used to authenticate hardware devices, as each TPM chip has a unique and secret RSA key named Endorsement Key (EK) generated for the device at time of manufacture. TCG firstly use a trusted third party named Privacy Certification Authority (Private CA). Just like the TC in the TCCP model, Private CA get involved in every transaction. Therefore E. Brickell, J. Camenisch, Liqun Chen proposed a DAA scheme [5] to protect the privacy of the owner of a trusted computing platform. The DAA scheme was adopted by TCG in the new specification 1.2 of TPM later.

The DAA scheme involves three types of entities: issuers, signers, and verifiers. An issuer is responsible for verifying the legitimation of signers and issuing a DAA credential to each signer. A signer interacts with the issuer in order to obtain an anonymous credential on a secret value f, and then can prove the membership to a verifier by providing a DAA signature; meanwhile he can hide his own identity to the verifier. Because of the limited storage space and computation capability of TPM, the role of the signer is spilt into a TPM and the TPM's host. The TPM is the actual signer and holds the secret signing key, while the host helps the TPM to compute the signature under the credential, but is not allowed to learn the secret signing key and to forge such a signature without the TPM involvement. The rationale behind this is that the host can always reveal a TPM's identity.

J. Camenisch [9] showed how to modify the DAA attestation protocol as to provide the same level of privacy as the TCG's initial Privacy CA solution while avoiding all the drawbacks of it. In our paper we will modify the protocols in [9] to serve for a different purpose.

C. Remote Attestation

Remote attestation mechanism works as follows: The platform builds a chain of trust from the Core Root of Trust for Measurement (CRTM) up to the OS by measuring integrity metrics of modules and holds them in Platform Configuration Registers (PCRs). When challenged, The TPM is able to report on these metrics in an authenticated way. This requires a list of Reference Integrity Measurements (RIMs) contained in a Reference Manifest Database. However in the DAA scheme using this mechanism will limit the anonymity by revealing the configuration information of the platform to the verifier [11].

D. Trusted platform architecture

Due to the rapid development of hardware virtualization for reducing the cost of ownership of systems, several trusted platforms based on virtualization technologies have been proposed to deal with the security concerns, such as Terra [7], NGSCB [6], vTPM [8], etc. Terra realizes the TVMM that allows the tamper-resident platform partitioned into multiple isolated VMs, of which there is a type of so called "closed-box" VMs that provide the functionality of running on a dedicated closed platform. Through the protection of memory and storage, the content in the "closed-box" VMs cannot be inspected or manipulated by the platform owner. The vTPM is a prototype system that provides trusted computing functionality to every VM on a virtualized platform.

In this paper our TVMM is combining features of vTPM with Terra, differences between these two prototype systems are:

1. The TVMM of Terra is using VMware GSX Server 2.0.1 while the vTPM is based on the Xen hypervisor. The latter one is open-source software while the former one is not. Moreover the latter one supports paravirtualization which would gain less performance loss of the platform.
2. The publications of Terra recognize the availability of TPM 1.1b, yet the DAA scheme hasn't been adopted until new specification 1.2.

So the vTPM prototype is more suitable for our research, however the concept of "closed-box" of Terra is needed.

III. THE MODEL

We present the improved trusted cloud computing platform model based on the DAA and Privacy CA scheme. This model guarantees the confidentially and the integrity of a customer's VM as well, and is able to solve the dependence issue on the TC. Section A gives an overview of our model, Section B presents a detailed design, and Section C gives an informal security analysis.

A. Overview

The TCCP [2] includes two components: a TVMM and a TC. A TVMM adopts the technologies from trusted system, hence is able to protect its own confidentiality and integrity. However the authors didn't go into their details about the design of the TVMM. In this paper our TVMM is mainly based on the combination of vTPM and Terra. According to different demands of customers, OS can choose to run in a standard virtual machine ("open-box untrusted VM") that provides the semantics of open platforms, a standard virtual machine associates with a unique vTPM instance ("open-box trusted VM") which has the trusted computing functionality, or a closed-box virtual machine associates with a unique vTPM instance("closed-box trusted VM"). Fig. 4 shows the TVMM architecture.

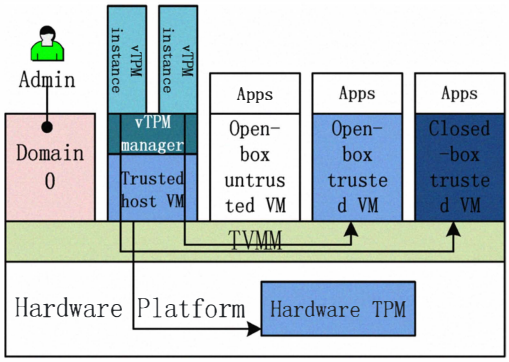


Figure 4. The TVMM architecture

The management VM, corresponding to Domain 0 in Xen, is determined by the administrator. If vTPMs exist in the Domain 0, then all of them are subject to subversion. Moreover in the DAA scheme, the fact that a signer is spilt into a TPM and the TPM's host is the host can always reveal a TPM's identity that we don't want to expose to the administrator. In order to eliminate the dependence of vTPMs on the Domain 0 and hide the host from the administrator, we have a dedicated trusted host VM (THVM) which runs at the highest privilege level that can prevent the tampering even by the administrator. The THVM is enforced to be launched by the TVMM when the hardware platform starts or restarts; it has cryptographic capabilities for generating asymmetric keys, handling encryption and decryption of data, and migration of asymmetric keys between virtual TPMs. It includes a vTPM manager that makes the mapping between VM and vTPM instances, and acts as the host in the attestation transactions. Platforms with the TVMM are called Trusted Nodes (TNs).

We retain the TC in the ETE; however the role won't go on managing the trusted nodes any more. The TC in this paper is to act as a trusted issuer and feedback the information of rogue TPMs to the CSP; hence an eligible ETE most likely is a manufacturer of TPMs. We employ a hierarchical design of the TC so that each sub-node associates with one zone. The TC Manager (TCM), as depicted in Fig. 5, makes the connection between the TC and the CC.

The role of managing the nodes will be transferred to the TN itself based on the neutral feature of the TPM. The topology in a zone is simplified as below, as every node can link with any other nodes in the same zone through the CC. TNs in light blue are ordinary trusted nodes, while green TNs are selected as Privacy CAs by the TC, each of them manages a portion of TNs. Green TNs contact with each other through the secretive dark blue TN. The reason why we use multiple green TNs is that a zone in the cloud contains a very large number of nodes. This solution is able to improve the efficiency of node management and increase the security through isolation. One green TN gets compromised, only a portion of TNs get affected.

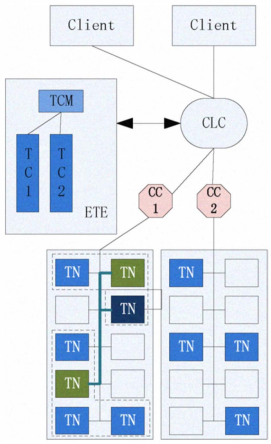


Figure 5. The TCCP architecture

B. Preliminaries

- In this paper if the remote attestation mechanism has to be adopted, then the TN will offer two configurations, one from the CRTM up to Domain 0, the other from the CRTM up to the THVM. Considering the violation of privacy information, we don't adopt this mechanism in this paper. However property attestation [11], which is out of the scope of this paper, can address the privacy issue.
- There are two roles for TNs, one is an ordinary role, and the other one is a role of the Privacy CA. The TN of Privacy CA (Privacy CA for short) is an ordinary TN which has additional functionalities to manage ordinary TNs, such as the "closed-box trusted VM" migration operation.
- Two different DAA schemes are employed on ordinary TNs and Privacy CAs. Protocols [5] and [9] will be modified to satisfy the need of our research. However more special and exclusive protocols are preferred as interaction processes between TNs and TCs require to be hidden.
- We suppose all TNs are aware of the public key of the TCM in a private way. When the TN starts or restarts, it will contact with the TCM first, and then will be connected with the corresponding TC. This solution deals with TN' zone transfer issue.
- The same parameters will be employed as the DAA protocol [5], i.e. ℓ_e, $\ell_{e'}$, ℓ_f, etc.

C. Detailed Design

This section describes our mechanisms in detail. We describe the protocols how Privacy CAs are selected, how they manage a list of TNs and contact with each other to deal with the VM management issues. To compare our model with the original TCCP, we will also make use of the issues namely launching and migrating VMs as examples.

1) Node management

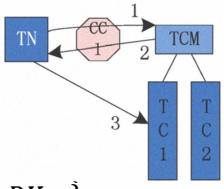
As mentioned before, in the original TCCP model the TC manages the set of TNs by maintaining a directory containing EKs identifying nodes' TPMs, while our TC primarily acts as an issuer and select TNs to be Privacy CAs randomly.

▫ Establishment of trust management relationship

This stage happens when TNs demand to build up the trust management relationship after the TCCP's or a zone's initialization. When the relationship is stable, and Privacy CAs have selected an *internal trusted coordinator* (ITC), the TC will trandfer a part of duty to the ITC. However the content in this section still hold true in the stable stage with only a few changes.

In order to obtain an anonymous DDA credential, a node must interact with the TC. Fig. 6 shows the preceding three messages during the communication. Suppose the TN in zone 1 this time, firstly it will send a challenge n_N and its PK_{EK} to the TCM in step 1, the TCM will check whether this PK_{EK} exists through querying a database of PKs_{EK}, then the TC1 will be notified to decide which role it will offer.

Assume the TN ā is the first one in zone 1 that has ever contacted with the TC1. The TC1will offer a chance to let it be a Privacy CA. Let M in step 2 be a byte describing the decision, that is, M = 0 means that a role of Privacy CA is going to be offered and M = 1 means that an ordinary role is going to be offered. Each value of M is corresponding to a DAA scheme, and that makes PK_{TC1} varied. One for the role of Privacy CA is $(n, g', g, h, S, Z, R_0, R_1, R_2, \gamma, \Gamma, \rho)$ [9], one for the ordinary role is $(n, g', g, h, S, Z, R_0, R_1, \gamma, \Gamma, \rho)$ [5]. However the TCM must ensure that the TC only generates two public keys for each role to prevent the Rudolph attack [10]. The TC1 will also choose a random $n_{TC1} \in \{0,1\}^{\ell_\emptyset}$ to help detecting message replays. TABLE 1 demonstrates the differences clearly between two roles in the following interaction processes.



1. $\{n_N, PK_{EK}\}_{PK_{TCM}}$
2. $\{\{ML_{TC1}, PK_{TC1}, M, n_{TC1}\}_{PK_{EK}}, n_N\}_{pK_{TCM}}$
3. $U, N_I, n_{TC1}, H(U \parallel n_{TC1})$

Figure 6. Message exchange in the beginning of the attestation

Carry on our illustration, the TC1 replies with $\{k_s, K_p\}_{PK_c}$ in step 9, K_p is a session key for the communication among Privacy CAs here. And then ā generates $< PK_c, pK_c >$ and sends PK_c to the TC1 for the interaction with ordinary TNs later. If the protocol ends successfully, ā will be the first Privacy CA in zone 1.

Assume that ƃ is the second one which wants to obtain a DDA credential, and then we will go through the DAA signing scheme for ordinary TNs (M = 1). The THVM of ƃ will be given PK_c, PK_p, bsn_P of ā and k_s that corresponding to ā in step 9.

If the protocol ends successfully as well, in next stage ƃ will prove to ā that its TPM/THVM has obtained the DAA attestation as follows (ƃ can also require ā proving its DAA certificate):

a) ƃ sends $\{PK_P, k_s, n_ƃ\}_{PK_c}$ to ā.

b) ā checks whether the k_s and PK_P are correct. If yes, ā replies with $\{n_ƃ, start\}_{pK_c}$.

c) Here we adopt the frequency certification protocol in [9]. The Privacy CA acts as a verifier in this scene, so in step 2(b), we modify k_0, k_1 as follows: $k_0 := LSB_{\ell_f}(H(bsn_P \parallel ran))$, $k_1 := CAR_{\ell_f}(H(bsn_P \parallel ran))$.

d) After step 2 of [9], The THVM of ƃ will not only sends ā its N_P and U, but also its N_{TC1}. The THVM of ā stores a pair of (N_P, N_{TC1}) for rogue TPM tagging in future.

e) Finally the THVM of ƃ stores (A, e, v) together with ran and $exp - date$.

The protocol we apply here is not an original anonymous one-time certificate any more, thus the duration of this Privacy CA certificate should last longer than the original one, and that will affect the generation of the $exp - date$. When the Privacy CA plans to check the validity of the TN after a time, we would apply the signing algorithm with k_0,k_1 modified as indicated above.

Now ƃ gets a DDA certificate and a Privacy certificate, and these two certificates are connected with a common value k_t. Fig.7 illustrates the management sub-zone of ā, ā stores (N_P, N_{TC1}) for each ordinary TN.

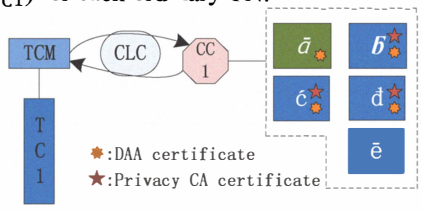


Figure 7. The management sub-zone of the Privacy CA ā.

Notice that there is a random integer $N_P \in [N_{min}, N_{max}]$ picked to appoint the number of TNs managed by ā in step9 of TABLE 1. Assume $N_P = 4$ in the case of Fig. 7, when the following TN ġ interacts with the TC1 successfully, then it will be the second Privacy CA in zone 1.

▫ Election of the *internal trusted coordinator*

Assume that TNs in zone 1 have established a stable trust management relationship after zone 1's initialization. Now Privacy CAs need to elect an internal trusted coordinator from the ordinary TNs which would connect with each management zone. Let us use the illustration in Fig. 8 to show how the election works. There are five trusted management sub-zones in zone 1. Privacy CAs will use the session key K_p generated by the TC1 for communication.

The Privacy CA will prove its DAA certificate to each other using the signing and verification protocols of the original DAA scheme.

The election result will be computed jointly by Privacy CAs. To avoid the situation that the election gets manipulated to some degree in case of that a few Privacy CAs get compromised, our paper introduces the idea of *Secure Multi-Party Computation* (SMC) here.

a) Each Privacy CA picks a random number and sends to the other Privacy CAs (assume ā: 4; ŧ: 2; ƙ: 7; ń: 16; ġ: 3 here).

b) Each Privacy CA calculates the sum of the random numbers, and then mod the number (denoted by n) of Privacy CAs ($(4+2+7+16+4) mod\ 5 = 2$ here). The result i corresponds to the Privacy CA which offered the $(i+1)$ th smallest number (ā will be appointed to select an ordinary TN as the ITC if every Privacy CA agrees the result).

c) ā selects ć as the ITC randomly, then ć will prove to the other Privacy CAs its Privacy CA certificate issued by ā and at the same time authenticate the public key of AIK generated for each Privacy CA (b=0 here).

d) Each Privacy CA divides its k_s into two parts: k_{s0}, k_{s1}. Assume one offered the ith smallest number, it would send k_{s0} to the Privacy CA who offered the $(i-1)(mod\ n)$th smallest number (0 denotes the biggest number), and k_{s1} to the Privacy CA who offered $(i+1)(mod\ n)$th smallest number.

e) Each Privacy CA adds up these two values from two other Privacy CAs, and sends the result to ć encrypted by the PK_{AIK}. ć calculates the sum of all these values, and submits the result to the TC1 when proving both its DAA certificate and Privacy CA certificate to the TC1. Meanwhile, using the DAA-signing protocol the ITC authenticates the PK_{AIK} generated for the communication with the TC later.

f) The TC1 adds all the k_ss generated for zone 1 earlier, and check whether it equals with the value that ć submitted. If yes, then the TC1 is convinced that ć is the legal ITC in zone 1, and would accept the PK_{AIK} generated by the ITC for further communication.

In the protocol above the purpose of step d) and e) is to calculate the sum of all the k_ss in zone 1 without exposing private k_ss of Privacy CAs while the TC1 is aware of all the values of k_s.

After the election of the ITC is finished, several details need to be described:

- As we mentioned before, each management sub-zone has a number range $N_P \in [N_{min}, N_{max}]$, while in this stable stage the ITC would expand the number range into N_{max}, i.e. the number range of the five sub-zones would be N_{max} after the election in Fig. 8. The information about vacant number of each sub-zone would be gathered by the ITC and shared with the TC1, thus the role of Privacy CA wouldn't be offered by the TC1 unless all the trusted management sub-zones are full.
- Once a new Privacy CA is added, the re-election of the ITC is not necessarily needed. The new Privacy CA would be notified the existing ITC. However the election of the ITC might be executed at intervals to change the host of the ITC.

Although the new ITC will do some additional tasks, it will keep playing the role of the ordinary TN to make its activity unnoticed.

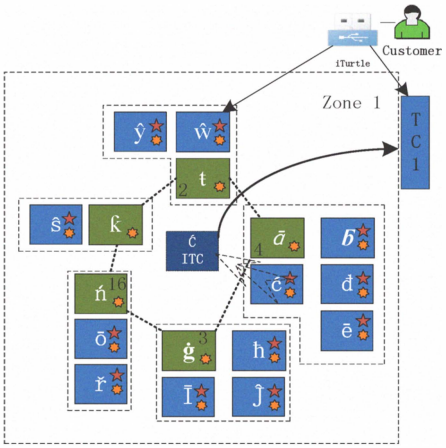


Figure 8. The election of the ITC.

2) Virtual machine management

Firstly the issue which customers concern a lot is discussed, i.e. how to identify their own "closed-box trusted VMs" are running on TNs when they are not aware of the interior of the cloud at all, moreover thin clients may not be trustable. Here we introduce the idea of iTurtle [12]. Our USB iTurtle could be considered as a simplified version of the trusted node. iTurtle firstly apply a DAA certificate from the TC, and prove its credential to the TN owning its VM, then the TN will prove its original credential to the iTurtle.

In order to compare our model with the original TCCP model, we present the protocols to secure the "closed-box trusted VM" launch and migration operations.

▫ VM launch operation

When the customer wants to upload its VM image (VMI) to a TN in zone 1, the protocol is depicted as follows:

a) The iTurtle applies an original DAA certificate from the TC1.

b) The TC1 contacts this issue with the ITC using the PK_{AIK}.

c) The ITC will select a Privacy CA to deal with the issue based on a randomized policy (the information about vacant number of each sub-zone is considered here).

d) The Privacy CA will select a TN based on a randomized policy (The availability of TNs is considered here).According to Fig.8, ŵ is selected to handle with the VMI.

g) The selected TN will require the iTurtle to prove its DAA certificate. Meanwhile using the DAA-signing protocol the iTurtle authenticates the PK_{AIK} generated for the communication with the selected TN later.

h) The selected TN will acquire an anonymous one-time certificate from its Privacy CA using the protocols of [9] without modification. (In step 2(b) of [9], the verifier-name will be the name selected by the iTurtle). And

then using the signing algorithm the TN will prove its certificate to the iTurtle.

i) The iTurtle sends $\{\alpha, H(\alpha), n_U\}_{pK_{AIK}}$ *to the selected TN (*α denotes the VMI here*).*

j) The selected TN will decrypt the message using the PK_{AIK} *which is shared only between these two entities, and then check whether the hash of* α *and* H(α) *equal. If yes, the VM will be booted.*

▫ VM live migration

There are two kinds of migration: the VM migration in a zone and the VM migration across a zone. We will depict the migration in a zone here. Assume the state of an executing VM in Ī is going to be transferred in Fig. 9, the destination TN will be selected based on a randomized policy according to the steps indicated above (ŝ is also an ordinary TN in Fig. 9). In this scenario each TN will acquire an anonymous one-time certificate from its own Privacy CA, and prove it to each other. And the VM states could finally be transferred to the destination encrypted by the AIK. Fig. 10 demonstrates that the migration across a zone relies on the TC.

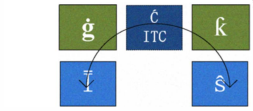


Figure 9. The migration in a zone

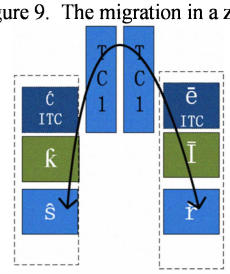


Figure 10. The migration across a zone

3) Rogue TPM tagging

The rogue tagging would be performed by the Privacy CA (Assume in zone 1) when ordinary TNs apply the certificate from the Privacy CA. The Privacy CA checks whether $N_P \not\equiv \left(\zeta^{f_0+f_1 2^{\ell_f}}\right) \bmod \Gamma$, for all (f_0, f_1) on the revocation list; and checks whether N_p has appeared too often lately. If N_P is considered as a name of rogue TPM, the Privacy CA would check a list of (N_P, N_{TC1}), and sends N_{TC1} to the TC1. Then PK_{EK} of the TPM will be discovered.

D. Performance and Security analyses

Providing a formal proof of security is beyond the scope of this paper, we will discuss our model informal as below:

1) We modified two DAA schemes [5] and [9] for our DAA and Privacy scheme. Both schemes have been proved that they are able to implement secure DAA systems under the decisional Differ-Hellman assumption in $< \gamma >$ and the strong RSA assumption in the random oracle model. And our modification wouldn't affect the proofs.
2) A THVM is designed to act as a host in the DAA and Privacy CA scheme to prevent the exposure of the TPM's identity.
3) The responsibility of managing TNs is transferred from the TC to TNs of IaaS based on the neutral feature of the TPM. Privacy CAs and the ITC are invisible to the adversary even by the administrator.
4) Although complicated protocols are executed to build a stable trust management relationship, the reliance on the TC would be lessened in the stable stage. Moreover if we adopt the policy that the Privacy CA wouldn't migrate the VM across its sub-zone unless its sub-zone is full, then the reliance on the ITC will also be lessened.
5) Multiple Privacy CAs divide TNs into several sub-zones, this mechanism will allow the management issues get dealt in parallel, and that would increase the performance significantly. Also this mechanism offers a decent feather of isolation, i.e. one Privacy CA gets compromised, and others would still work well.
6) We introduce the idea of SMC to avoid the situation that the election gets manipulated to some degree in case of a few Privacy CAs get compromised. The ITC submits the sum of all the k_ss to the TC for proving the election result without exposing the private k_s of each sub-zone.

IV. CONCLUSIONS AND SUGGESTIONS

In this paper we have proposed an improved TCCP model based on the DAA and Privacy CA scheme to reduce the reliance on the third trusted party outside of the cloud, meanwhile the confidentially and the integrity of a customer's VM is still guaranteed as well.

Property attestation is suggested to substitute for remote attestation, and the idea of property attestation will be introduced in our further research. Based on our design of the TVMM and TCCP model we will implement a new prototype system in the future.

ACKNOWLEDGMENT

This work was supported by the Major Research Plan of the National Natural Science Foundation of China (No.90818005), the National Natural Science Foundation of China(Nos. 60903217 and 60773032), and the China Postdoctoral Science Foundation funded project (No. 20090450701).

REFERENCES

[1] Tim Mather, Subra Kumaraswamy, and Shahed Latif, Cloud Security and Privacy, September 2009 , pp.30.

[2] N. Santos, K. P. Gummadi, and R. Rodrigue, "Towards Trusted Cloud Computing," In Proc. of the 1st USENIX Workshop on Hot Topics in Cloud Computing, Berkeley, CA, USA, 2009.

[3] B.D.Payne, M.Carbone, and W.Lee. "Secure and Flexible Monitoring of Virtual Machines," In Proc.of ACSAC'07,2007.

[4] D.Nurmi, R.Wolski, C.Grzegorczyk, G.Obertelli, S.Soman, Youseff, and D.Zagorodnov. "Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems," TechnicalReport2008-10,UCSB.

[5] E.Brickell, J.Camenisch, and L.Chen. "Direct anonymous attestation," In 11th ACM Conference on Computer and Communications Security. ACMPress, 2004.

[6] M. Peinado, Y. Chen, P. England, and J. Manferdelli. "NGSCB: A Trusted Open System," In Proc. of 9th Australasian Conference on Information Security and Privacy, *ACISP*, Sydney, Australia, 2004.

[7] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. "Terra: a virtual machine-based platform for trusted computing," In *ACM Symposium on Operating Systems Principles (ASOSP)*, pages 193–206. ACM Press, 2003

[8] S.Berger,R.C´aceres,K.A.Goldman,R.Perez,R.Sailer,and L.vanDoorn. "vTPM: virtualizing the trusted platform module," In Proc.of USENIX-SS'06, Berkeley, CA, USA, 2006

[9] Jan Camenisch. "Better Privacy for Trusted Computing Platforms (Extended Abstract)," ESORICS 2004: 73-88

[10] C. Rudolph. "Covert identity information in direct anonymous attestation (DAA)," In Proc. of the 22nd IFIP TC-11 International Information Security Conference (SEC2007)

[11] J. Poritz, M. Schunter, E. Van Herreweghen, and M. Waidner. "Property attestation – scalable and privacy-friendly assessment of peer computers," IBM research report RZ3548, 2004.

[12] J. M. McCune, A. Perrig, A. Seshadri, and L. van Doorn. "Turtles all the way down: Research challenges in user-based attestation," In Usenix Workshop on HotSec'07

TABLE 1. THE FOLLOWING INTERACTION PROCESSES WITH THE TC1

Interaction processes	**Different details**	
	Role of Ordinary TNs $(\boldsymbol{m = 1})$	***Role of Privacy CA*** $(\boldsymbol{m = 0})$
4. The THVM computes $\zeta_{TC1} := \left(H_\Gamma(1 \parallel bsn_{TC1})\right)^{\frac{\Gamma-1}{\rho}} (mod\ \Gamma)$, and selects cnt and sends them to the TPM.		
5. The TPM checks whether $\zeta_{TC1}^{\rho} \equiv 1(mod\ \Gamma)$, computesU,N_{TC1},$a_U := H(U \parallel n_{TC1})$, f : = H(H(DAAseed ‖ H($PK_{TCM}$)) ‖ cnt ‖ 0) ‖ ⋯ ‖ $H\left(H\left(DAAseed \parallel H(PK_{TCM})\right) \parallel cnt \parallel i\right)$ $(mod\ \rho)$, and sends them to the TC1.		
6. The TC1 verifies if $a_U = (U \parallel n_{TC1})$ to ensure that it's talking to the right TPM, checks whether values of $N_{TC1} \not\equiv \left(\zeta_{TC1}^{f_0+f_1 2^{\ell_f}}\right) mod\ \Gamma$for each (f_0, f_1) on the rogue list.		
7. The TPM proves to the TC1 the knowledge off_0, f_1 and v' :.		
8. The TC1 chooses a prime e $\in_R [2^{\ell_e-1}, 2^{\ell_e-1} + 2^{\ell'_e-1}]$, and computesv'', A, and sends (A, e, v'') to the corresponding THVM.	$A := \left(\frac{Z}{US^{v''}}\right)^{\frac{1}{e}} (mod\ n)$	The TC1 chooses an appropriate and uniquek_t, and computes $A := \left(\frac{Z}{US^{v''}R_2^{k_t}}\right)^{\frac{1}{e}} (mod\ n)$.
9. The TC1 runs the protocol to convince the THVM that A was correctly computed.	SPK{(d): A $\equiv \pm(\frac{Z}{US^{v''}})^d$(mod n)}($n_h$); The TC1 sends the THVM the PK_P, bsn_P, PK_c of the Privacy CA with which the TC1 would like the TN to contact later,and k_s that corresponding to the Privacy CA;	SPK{(d): A $\equiv \pm(\frac{Z}{US^{v''}R_2^{k_t}})^d$(mod n)}($n_h$); The TC1 generates an appropriate and unique k_s, sends $\{k_s, K_P\}_{PK_c}$ to the THVM; The TC1 picks a random integer $N_P \in [N_{min}, N_{max}]$ to appoint the number of TNs managed by the Privacy CA (random number is generated to blind the adversay).
10. The THVM verifies whether e is a prime and lies in $[2^{\ell_e-1}, 2^{\ell_e-1} + 2^{\ell'_e-1}]$, and then forwards $v''$to the TPM.	The THVM stores (A, e, PK_c, k_s) ; the TPM stores (f_0, f_1, v).	The THVM generates an asymmetric key pair< $PK_c, pK_c >$, $PK_P := (n, g', g, h, S, Z, R_0, R_1, R_2, R_3,)$, bsn_P and then sends PK_c , PK_P , bsn_P to the TC1;The THVM stores (A, e, k_t, k_s); the TPM stores (f_0, f_1, v).