Zic and Hardjono *Journal of Cloud Computing: Advances, Systems and Applications* 2013, **2**:4
http://www.journalofcloudcomputing.com/content/2/1/4

Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH Open Access

Towards a cloud-based integrity measurement service

John Zic[1*] and Thomas Hardjono[2]

Abstract

The aim of this paper is to propose the use of a cloud-based integrity management service coupled with a trustworthy client component – in the form of the *Trust Extension Device* (TED) platform – as a means to to increase the quality of the security evaluation of a client. Thus, in addition to performing authentication of the client (e.g. as part of Single Sign-On), the Identity Provider asks that the integrity of the client platform be computed and then be evaluated by a trustworthy and independent *Cloud-based Integrity Measurement Service* (cIMS). The TED platform has been previously developed based on the Trusted Platform Module (TPM), and allows the integrity measurement of the client environment to be conducted and reported in a secure manner. Within the SSO flow, the portable TED device performs an integrity measurement of the client platform, and sends an integrity report to the cIMS as part of the client authentication process. The cIMS validates the measurements performed by the TED device, and reports a *trust score* to the Identity Provider (IdP). The IdP takes into account the reported trust score when the IdP computes and issues a Level of Assurance (LOA) value to the client platform. In this way the Service Provider obtains a greater degree of assurance that the client's computing environment is relatively free of unrecognized and/or unauthorized components.

Introduction

Today there is a strong interest within Enterprises to move some or all their IT infrastructure and services to the cloud, with the aim among others of reducing the cost of IT operations as a whole. However, there are a number of security and privacy issues relating to cloud-based services, including the issues relating to providing access to external entities.

Two of the common cloud deployment scenarios faced by many Enterprises today are as follows:

- *Employee access to cloud-based applications*: An Enterprise seeks to make cloud-based productivity-applications available to its employees. The employees should not notice any differences between accessing the application in the cloud versus the same application running on a local machine. This scenario has a number of security implications. One implication is the need for the employee authentication status and authorization data to be conveyed from the Enterprise to Cloud-based provider. In this scenario the Enterprise remains being the authoritative source of all employee identity. Typically employee identities and privileges are managed through the corporate directory service, which itself may or may not be in the cloud (though this is tangential to the case of the cloud-based applications scenario).
- *Enterprise with in-bound institutional customers*: Another scenario is one in which an Enterprise with its applications running in the cloud is sharing this application with another organization. Thus, consider an Enterprise-A that has as its customer another institution called Enterprise-B with its own employees. For example, Enterprise-A could be financial services company offering retirement fund (e.g. U.S. 401K) management for employees of Enterprise-B. Here Enterprise-A has customer-facing applications that are operating in the Cloud (e.g. SaaS application). A key aspect of this scenario is that Enterprise-B is the authoritative source of identity for its employees, all of whom are accessing the cloud-based application belonging to Enterprise-A. Thus, authentication and authorization data must be

*Correspondence: john.zic@csiro.au
[1]CSIRO ICT Centre, PO Box 76, Epping, NSW 1710, Australia
Full list of author information is available at the end of the article

> conveyed from Enterprise-B into the cloud-based application within the domain or realm of Enterprise-A.

An additional aspect of these scenarios is that Enterprises have already invested in and deployed strong authentication infrastructure and identity management services. Many seek to extend and re-use these infrastructures to address the needs of the new cloud environment.

Parallel to the recent developments in cloud-based services is that of the development of loosely connected federated identity and authentication services in the low-value consumer space. These low-value federated identity services have been exemplified by social networks where one simple password-based login to one social network allows the user to access other social networks without needing to perform further authentications. Although Enterprises have been interested in using this model to expand their customer base, one stumbling block remains that of the low security-quality of these loosely connected federated identity services.

Partly in response to this poor security quality, several organizations have emerged with the aim of defining the so-called standard *trust frameworks* as a means to bootstrap trust among entities in the identity ecosystem [1,2]. These trust frameworks provide a foundation for entities to transact based on an agreed common legal contract, thereby overcoming the limitations and non-scalability of bilateral agreements. One common aspect of many of these trust frameworks is the use of a *Level of Assurance* (LOA) as a means to denote the quality of authentication performed by (and therefore confidence in) an identity provider. The LOA is a way to express the quality of the authentication event from the perspective of security. Thus, for example, a user wielding a hardware token in a two-factor authentication event will obtain a higher LOA value compared to a user that authenticates merely using a password. The US National Institute for Standards and Technology (NIST) has issued a publication defining a granular level of LOA values [3].

In this paper we argue that in addition to strong authentication with high LOA values, identity-based services in the cloud need to also perform access decisions based on the quality of the computing platforms or devices from which clients (e.g. employees, customers, or users) perform remote access to these cloud-based applications and services. We believe that a measured trustworthiness indicator or the so-called *trust score* of a given computing platform should be part of the authentication and authorization of that platform when it seeks access to services in the cloud. In so far as possible, the computing platform elements being measured include all softwares and firmwares, and also the hardware component identifiers [4].

We also argue that in order for new cloud-based services to be acceptable by Enterprises today, a high degree of interoperability with existing "Enterprise-grade" authentication and authorization infrastructures is required.

In this paper we propose an architecture for a *cloud-based Integrity Measurement Service* (cIMS). The cIMS performs the evaluation of the integrity measurements received about the client, and issues a trust score reflecting its evaluation against one or more predetermined profiles for client measurements. We use the classic SAML 2.0 ecosystem [5] as a means to illustrate usage of the cIMS. In this model, a client seeking access to a *Service Provider* (SP) must first be authenticated by an *Identity Provider* (IdP), who issues SAML assertions pertaining to the client. Here we extend the SAML2.0 model by having the IdP request also from the client an *integrity measurement report*. In order to satisfy the need for a high LOA level, we propose the use of a trustworthy portable computing platform, the *Trust Extension Device* (TED) to provide the client-side trusted computing environment capable of performing integrity measurements of the client-side components.

Since the NIST Recommendations [3] already point to the need of hardware tokens to achieve a Level of Assurance (LOA) Level-3 or higher, we believe that the TED device offers a flexible and portable computing environment that satisfies the NIST Requirements. We believe the TED device represents a strong token for the subscribers within the e-Authentication model defined within the NIST Recommendation.

Background: TED and integrity measurements

There has been a growing interest by commercial organisations in providing portable, trusted and secure computing platforms that may be used in the scenarios such as those outlined in the Introduction section. A number of solutions have been proposed from a variety of vendors, ranging from IronKey [6], and Gemalto [7], to the Singapore Government's DIVA [8]. Some of these solutions are strongly tamper resistant and locally tamper evident.

The Trust Extension Device (TED) was developed with similar goals of providing a portable, secure and trusted computing platform. However, its key differentiator is that it adopted and implemented the Trusted Computing Group's (TCG) [9] standards and architectures into a small, portable device. In particular, the TED provides an issuing enterprise a truly *trusted computing platform* whose root of trust and associated functionality is based on the Trusted Platform Module (TPM v1.2b [10]) cryptographic microcontroller hardware. The TPM becomes a root of trust for the TED platform, and allows the *remote validation* of its hardware and software through the use of cryptographically secure integrity measurement and attestation protocols.

In the TCG architecture, a specialised Privacy Certifying Authority service is required to participate and provide supporting validation of credentials and keys that are used by the TPM to encrypt and sign messages (including integrity measurements) between the TED and the service provider's cloud infrastructure. Together, the TED, Privacy Certifying Authority and the service provider's cloud infrastructure, develop and maintain a provable, measurable trust relationship between themselves.

It should be noted here that there are two significant points need to be addressed when implementing and ultimately deploying such a system.

First, in every TPM enabled enterprise system, the enterprise application servers need to have *complete* knowledge of the hardware and software characteristics of *all* their client computers in order to engage, and successfully complete, the integrity measurements and attestation protocol. Each variation from a standard, known environment needs to be identified, captured and maintained within the application server so that the attestation protocols can continue to operate correctly. However, the variety of software images and hardware configurations, the rate at which these change, coupled with the typically large number of computers connecting to the enterprise server makes the task of maintaining and managing this information difficult and challenging.

The TED and associated infrastructure addresses the management issue by (i) reducing the complexity of the device and associated operating system and application software, (ii) having the device issued by a controlling enterprise/authority and (iii) being sufficiently cheap and portable for a new one to be easily re-issued if required. The TED's environment (drivers, operating system and applications) was specifically designed and optimised for execution speed and offers a restricted and controlled set of applications and services. It is completely under the control of the issuer. By design, the TED cannot be altered or modified once it has been configured and issued by the enterprise. Any changes or deviation from expected configuration are remotely detected by the application server through the TPM integrity measurements and attestation protocols. Should a change be detected, the issuer can take appropriate action, such as not engaging in the critical transaction, or notifying the client that their TED has been compromised and will be revoked, etc.

Second, data and services are now available (for example, when enabled to utilize cloud computing infrastructures) to a wider cross-section of users, operating under unknown and unpredictable computing environments that lie beyond the control of a single enterprise. In many cases, the users themselves operate beyond a single organizational boundary.

These uncertainties are addressed by TED being able to be plugged into a USB port of any host computer, without the need for specialised hardware interfaces or readers, to create a known (to the issuing enterprise), trusted computing platform and associated environment and applications that are isolated (from the host's hardware up through to its operating system and applications) from the host computer.

The design and implementation of the TED prototype (both the hardware and software) are presented in detail in another paper ([11]). We summarise its salient features here.

An overview of the TED hardware and software

For completeness, the design requirements for the TED prototype (and its associated software components/system) were as follows:

- DR-1 Should be small and cheap enough to be portable and easily re-issued to a client by the owner enterprise or cloud service provider.
- DR-2 Must be able to physically plug into any host PC with a USB 2.0 compliant port.
- DR-3 Must use the USB port *solely* for power and for establishing a secure network connection (tunnel) to well-known servers after it has successfully booted.
- DR-4 Must be able execute owner enterprise or cloud service provider developed applications on an embedded operating system.
- DR-5 Must include and use a TPM v1.2 compliant cryptographic microcontroller.
- DR-6 Must be able to implement and participate in attestation and identity management protocols as per TCG specifications.
- DR-7 Must *not* require that the host needs to be rebooted for correct operation.
- DR-8 Must *not* rely on the host PC being interrupted from normal operation.
- DR-9 Must be able to be inserted or removed from the host PC *at any time* without causing either the host or the TED to enter an error state.

The selection of the USB 2.0 connection was based on pragmatic reasons - USB 2.0 reduced the hardware costs, complexity and software development time substantially over other design options.

In principle, a TED may be regarded as a stand-alone networked device. According to the above design requirements, the TED really only requires the host PC for power and network connectivity (using either wired Ethernet or WiFi). It operates its own isolated memory address space, running on a separate CPU, all of which may be attested as operating within the strictly controlled environment provided by the issuing enterprise (which may be a cloud service provider). There were no requirements

for having any traditional user screen or keyboards to interface to it; any such interfaces were to be provided through appropriately secured connections to the host PC.

As such, we did consider other designs, including having a keypad and single LCD screen mounted on the device that the user may interact with independent of the host PC (which again was only providing power and secure network connectivity). This design was never implemented due to time and project resourcing issues. Another design was a natural development from the "small keypad single LCD screen" version, where the TED was connected to a cellular phone platform providing power and that utilised 3G data network for secure network connectivity. However the initial design investigations revealed that the power requirements of this TED variant exceeded the capacity of a cellular phone's supply, and so this option was never pursued.

Figure 1 shows the TED prototype hardware that consisted of two boards: a motherboard containing the CPU, memory and associated control logic (out of view) and a daughterboard, carrying the TPM chip, USB interface and associated power supplies for both boards.

The TED internal software architecture is shown in Figure 2. It is based on an embedded Linux operating system, and required the development of dedicated extensions and drivers to accommodate the TPM device and USB interface. Above the operating system, the TED utilised a light-weight TCG Software Stack (TSS) Library and TSS server that allows applications to interface to the TPM functions.

A user's applications execute on the TED (again, that are under the control of the issuing cloud service provider) to access external systems. Its execution and memory spaces are isolated from the host PC, and only uses the host PC for power and to establish secured Internet connection to the cloud provider's service. The secured Internet connection is tunneled from the TED's USB port, through the host PC's network connections and finally onto the cloud provider's service.

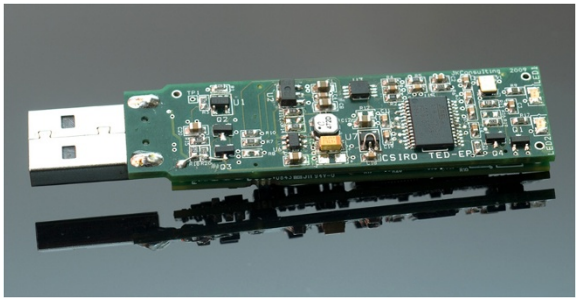

Figure 1 TED engineering prototype.

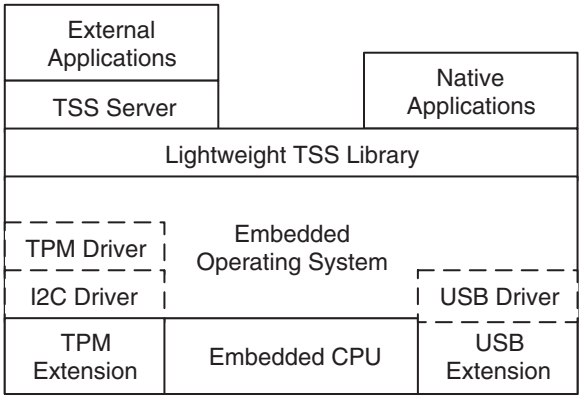

Figure 2 TED software architecture.

Integrity measurement and attestation

Without entering into details of the TCG's recommended integrity measurement and attestation (of the TED platform and its complete operational environment) [12], most (except for the Direct Anonymous Attestation protocols) follow the similar structure of the protocol given in Figure 3. Section 4.2 in the TCG Specification Architecture Overview [4] gives an excellent overview of this topic.

At step (1), a challenge message with a fresh nonce is issued to a platform (in this case, the TED) to attest its identity and integrity with the application service. Once this message is received, the TED platform at step (2) calls a TPM function to generate an Attestation Identity Key (AIK) and the TPM credentials. These credentials, along with other credentials[a] the public part of the AIK and the original challenge nonce are signed by the private TPM Endorsement Key. The resulting signed credential is then encrypted using the public part of the certifying authority's key. This is then sent at step (3) to the certifying authority as a request to validate its credentials. If successful, the certifying authority creates a signed, encrypted credential that is sent back to the platform at step (4). Step (5) is used to produce an encrypted summary of measurements of the environment held in sealed storage (the Platform Configuration Registers, or PCR) in the TPM chip. (e.g. one measurement may be a list of loaded and running processes just after boot, another measurement may be a new list of running processes several hours after boot). This encrypted measurement information is then sent back to the challenger, along with the identity credential received from the certifying authority at step (6). Upon reception of the message, the challenging application service now validates at step (7) the measured environment and compares it to its own expected measurements that it holds. During this process, the challenging application service checks the identity

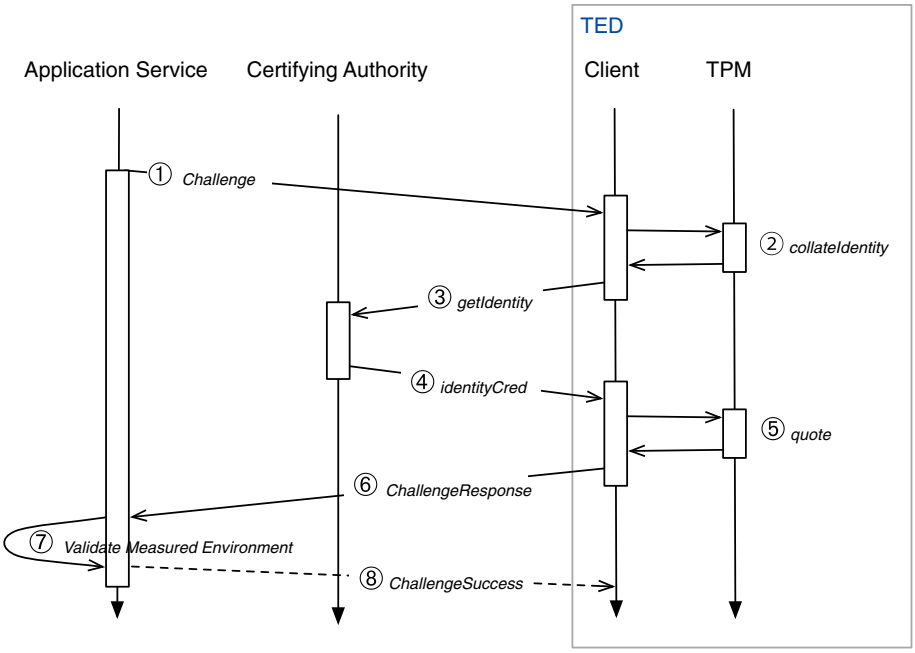

Figure 3 An attestation protocol.

credential to determine that it was signed by the certifying authority, as well as verifying the signature of the measured values and that it contains the original challenge nonce. Depending on the application requirements, step (8) may optionally be used to signal to the application client the success of the measurement and attestation process.

Architecture for cloud-based integrity measurement services using TED

As mentioned previously, we argue that another dimension of trust in identities on the Internet that must be accounted for is that of the state or condition of the platform from which a given identity is seeking services. To this end we propose a *cloud-based Integrity Measurement Service* architecture (Figure 4) coupled with TED that adds the dimension of state measurement to the process of authentication of clients by the IdP when they seek access to SPs. In the following we describe the entities, assumptions and functions within the architecture, followed by a description o the steps taken by the User/Client located within the Enterprise.

Entities, assumptions and functions

Figure 4 shows the entities operating in the ecosystem:

- *TED platform, with client and User*: The User is assumed to be an employee within the Enterprise, using the client software operating on the TED platform. The TED device is able to perform the measurement of the client-side environment, including all application software.
- *Enterprise Directory Service* (EDS): In the architecture we assume that some form of directory services exist under the domain of control of the Enterprise. For authentication performed by the IdP, the EDS is assumed to be the authoritative source of identities for all employees in the enterprise. Although Figure 4 shows the directory service as being a separate entity located within a cloud external to the Enterprise, in practice an implementation can place the directory service (i) within the Enterprise (as has been the case in the past), (ii) within a private cloud inside or outside the physical boundary of the Enterprise (as is often proposed today), or (iii) within a hosted service in an external cloud. Regardless of the configuration of the directory service, in this architecture we assume that the Enterprise has full control of the directory, with some degree of sharing of certain employee attributes and permissions with the IdP.
- *Identity Provider* (IdP): The IdP is as understood broadly in the identity literature, and as defined more specifically by the SAML core [13] and SAML profiles [5] specifications.
  When the user performs an SSO to a Service provider (SP) and is redirected by the SP to the IdP for authentication, the IdP will perform an additional step of seeking the measurement of the Client's platform. After a successful authentication by the IdP,

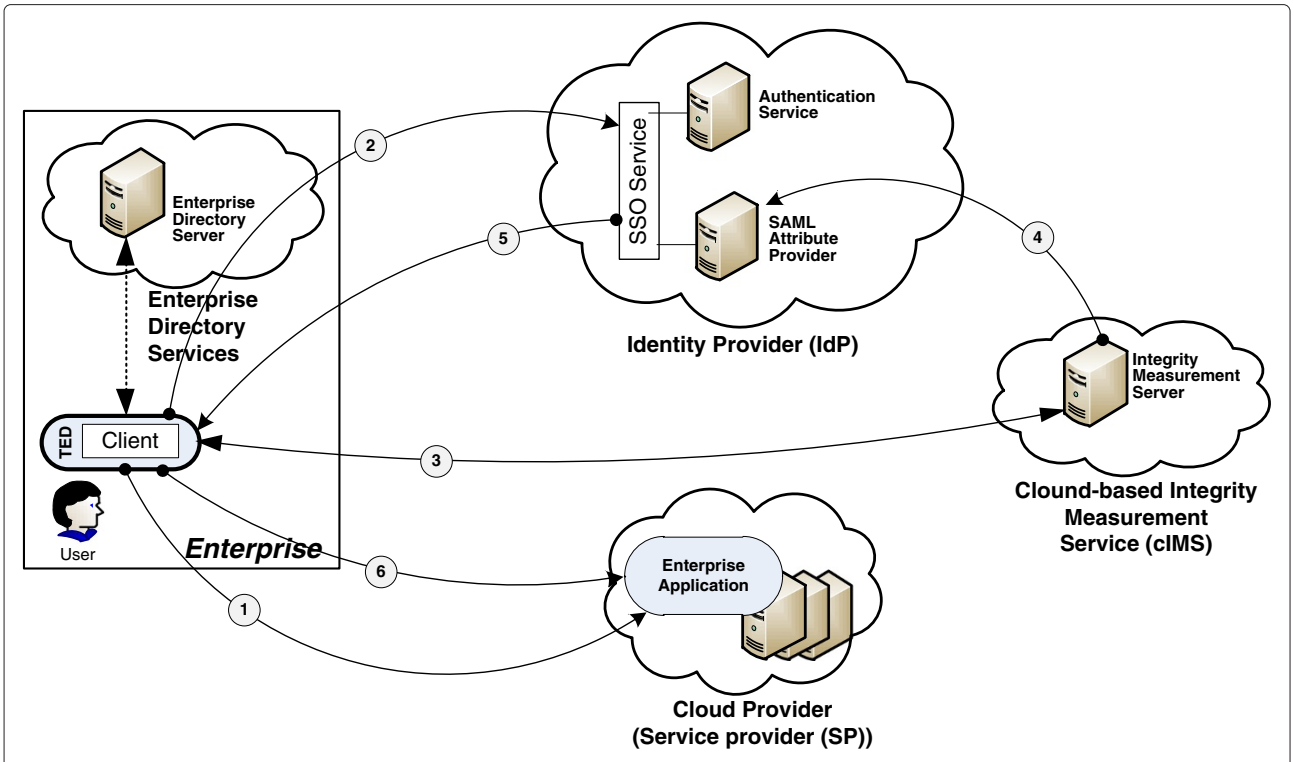

Figure 4 Overview of the cIMS architecture.

the IdP redirects the Client to the cIMS service (see below) in order to perform an integrity measurement using the TCG Trusted Connect Model (TNC) [14]. As a result of a successful authentication and measurement of the client and TED, the IdP will issue the relevant signed assertions containing among others the identity of the client, the LOA level achieved (according the defined assurance policies set at the IdP), and the protocol type used to authenticate the User and Client.
We note that the IdP may take on a similar role of the TCG's *Privacy Certifying Authority* (Privacy CA), of creating, managing and interacting with the TPM chip onboard the TED. Although another party could take on the role of a Privacy CA in principle, implementing and deploying within a SAML architecture would present difficulties due to policies, accountability requirements and responsibilities that come with operating such an authority as a separate entity.

- *Cloud-based Integrity Measurement Service* (cIMS): The cIMS is the service that performs the integrity measurement and evaluation of the Client platform. In general, we assume that the IdP has a trust relationship and a contractual business agreement with the cIMS based on a mature trust framework. There are a number of possible outputs from an evaluation by the cIMS. Here we assume that at the very least the cIMS returns an *integrity score* based on some measurement scale agreed upon with the IdP. Although the cIMS is shown in Figure 4 as separate from the IdP, the cIMS could in fact be a service operating within the IdP. This approach would allow the IdP to offer a wider set of services while operating under a single trust framework.
- *Cloud Provider* (CP): The CP is the cloud provider, which corresponds to the *Service Provider* in the SAML2.0 terminology [5]. The CP is the relying party who depends entirely on the IdP for correct authentication and integrity evaluation of the TED platform. The CP uses the signed SAML assertions (containing the LOA values) from the IdP in order to grant/deny access to the User to resources or services at the CP (either provided by the CP itself or gated by the CP).

It is important to note that Figure 4 *does not* explicitly show the additional entities and protocols involved required to fully support the TPM functionality such as the various key and credential creation and management functions. Where required, these are presented in the text of the paper.

Overview of protocol flows

Figure 4 provides a high level view of the interactions between the client running on the TED platform with the IdP. In the following discussion, we use the terminology and scenario of the SAML2.0 Single Sign-ON (SSO) as defined in [5]. This well known SSO scenario is purposely selected in order to highlight the introduction of TED and the cIMS service.

For clarity, the Privacy Certifying Authority has been left off the diagram. As explained in a previous section, the Privacy CA service could be run by the IdP, or if required, by a separate trusted third party.

We use the term "Client" to denote the application software running on the TED platform that performs the SSO to the IdP. The Client software is the piece of software that integrates the authentication client and is entity that interacts with the SP and IdP on behalf of the user. It is also the software that triggers the integrity measurements performed by TED, and as such must always be included in any platform measurements and in integrity self-checks.

The following summarizes the interaction between the Client on TED and the entities in the ecosystem:

(1) *The Client (on TED) requests access to Cloud Provider*: Following the classic SAML 2.0 SSO scenario, the Client requests access to resources and services at the Cloud Provider (which is called the Service Provider in the SAML2.0 glossary). The CP redirects the Client to the IdP for authentication of the Client.
Within the re-directon message the CP includes a signed *integrity schema* document (or a pointer to it) [12]. The integrity schema represents the components within the client's platform that is of interest to the CP. Thus, for example, the CP may be interested in the status-information regarding the client's firmwares (e.g. BIOS, drivers, etc), in the OS patch level and in the Anti-Virus (AV) condition of the client machine.
Note that if the CP and the IdP has a back-channel established by virtue of their business relationship, the CP can communicate its set of preferred integrity schemas to the IdP via the back-channel.

(2) *Client Authenticates to IdP*: Here the Client has been redirected by the Cloud Provider (ie. SP) to the Identity Provider (IdP) for user authentication. The method for user authentication is out-of scope in this paper, and has been well treated elsewhere (for example, see SAML2.0 SSO profile).
After successful authentication by the IdP, the Client is further re-directed to the cloud-based Integrity Measurement Service (cIMS) selected by the IdP. The cIMS is assumed to be a trusted third party operating under the same trust framework as the IdP and Cloud provider.
In requesting that an integrity measurement be performed, the IdP needs to indicate which components in the client's environment must be measured. One possible approach is for the IdP to include (in its re-direction of the Client) an *integrity schema* [12] which indicates to the cIMS which platform components are of interest to the IdP.
In this way the Client can initiate the TED platform to perform measurements following the integrity schema. Later, in compiling the integrity report, TED will format the report also following the integrity schema, and return it The integrity schema should be a core part of the client-profile that the IdP maintains for that Client.

(3) *Client is integrity-evaluated by the cIMS*: Here the Client has been re-directed by the IdP to the cIMS for the purpose of measurement of the Client and evaluation by the cIMS.
Upon receiving the indication from the cIMS that an integrity report is required, the Client performs the measurement of the components as indicated by the cIMS. Assuming that an integrity schema was returned (or pointed to) by the IdP, the TED performs the measurements following the schema. The integrity schema provides a uniform and standard manner in which to indicate which software components to measure. Examples include the measurement of the Client itself, other applications, the kernel, the firmwares, anti-virus measurements, and others. Note that a complete measurement of the entire platform could also be requested [12].

(4) *The cIMS forwards a Trust Score to IdP*: Upon completing the evaluation of the integrity report obtained from TED in the previous step, the cIMS generates a *trust score* for the Client. The trust score reflects the judgement of the cIMS (regarding the Client) as compared against some *Integrity Measurement Policies* stored at the cIMS. The cIMS logs all the measurement and evaluation events, and archives all received integrity reports and resulting trust scores in order to maintain audit and accountability information.
Note that in the multi-host situation, the IdP may have an account with the cIMS within which it defines the set of integrity measurement policies for all clients that the IdP re-directs to the cIMS. As such, the cIMS becomes a provider to multiple IdPs.

(5) *The IdP issues assertions with LOA*: Upon receiving the trust score from the cIMS, the IdP compares the trust score against the access control policies (e.g. belonging to the Enterprise) stored at the IdP.

(Alternatively, the IdP could access the relevant Enterprise access control policies from the EDS). The IdP then issues a signed assertion or claim containing the LOA value. The LOA is the result of the IdP successfully authenticating the User and the evaluation of the trust score (about the Client/TED) as received from the cIMS.
Here it is worthwhile noting that the IdP could in fact issue multiple LOAs, each refering to one aspect of the authentication event. Thus, for example, an additional LOA value could also be issued by the IdP conveying the second factor of authentication used by the user (e.g. User wielded an OTP token or biometric device).

(6) *The Client sends request to the Cloud Provider*: Upon the signed assertion (containing the LOA value) from the IdP, the Client forwards the assertion to the Cloud provider in its re-attempt to access the resources at the Cloud Provider.

Identity and trustworthy collaborations

Trustworthiness and being able of assure identity claims with high degree of confidence also arises in broader collaborative situations, where organizations have to partner with each other over a specific period of time each to achieve a set of mutually desirable outcomes. Typically, each participant will have their own respective policies in place about control and sharing of information, as well as policies that cover exceptions. When brought together, the establishment of identity, who has access to information and the control of access and the information itself forces each participant to consider how to evaluate the trustworthiness of the collaboration and the respective partners.

We believe that trustworthy collaborations are enabled if the partners have the following in place:

1. *Agreed upon contracts.* The collaborating partners must first formalise an agreement (a contract) that allows them to understand how information within the collaboration will be used between partners. This includes who has access, how information may be shared, how long does the shared information persist, what happens to shared collaborative information should a partner leave the collaboration prematurely or if a new partner joins the collaboration later, and so on. For cloud service providers, this typically constitutes an explicit SLA.
2. *Demonstrable adherance to the contract* Once a contract between participants is agreed upon, each partner and their organisations must be able to prove that they are able to behave according to the collaborative agreement, and in particular, for those transactions that are critical within the collaboration. In this paper, we identify one particular technique that applies to cloud service providers: the assurance of the integrity of the systems that are used are within bounds of the collaboration agreement, as described in "Background: TED and integrity measurements".
3. *Resolving exceptions and disputes* Establishing that the partners demonstrated compliance against a contract and behaviour (given in points (1) and (2), above) also requires the use of a variety of differing pieces of irrefutable evidence gathered about the behaviour of each of the collaborators. This evidence needs to be irrefutable so that it can be used to settle any disputes about whether or not partners have behaved according to the collaboration agreement. From another point of view, each of the partners is held accountable for their actions. With this in mind, we developed an Accountability Service, proposed in [15], that is directly applicable to cloud service providers. This service may be used to collect and manage the evidence of critical transactions within a collaborative environment, where each participant belongs to a separate organisational domain (with its own policies).

In terms of the identity claims that are made within a collaboration, adopting the approach taken in this paper would be of benefit to each individual participant. Further, as the cloud service provider is key in establishing and maintaining the collaborative "infrastructure", the addition of services such as accountability and provenance increases the level of trust in the system and between the participants by providing (if required) additional evidence of "good" behaviour as well as irrefutable evidence of "bad" behaviour.

Conclusion and further work

The goal of this paper is to proposed the use of a cloud-based integrity management service coupled with a trustworthy client component in the form of the portable trust extension device (TED), as a means to to increase the quality of the security evaluation of a client. In addition to performing authentication of the client (e.g. as part of Single Sign-On), the IdP asks that the integrity of the client platform be computed and then be evaluated by a trustworthy and independent Cloud-based Integrity Measurement Service (cIMS).

The portable TED device performs an integrity measurement of the client platform, and sends an integrity report to the cIMS the cIMS validates the measurements performed by the TED device, reports a *trust score* to the Identity Provider (IdP). The IdP takes into account the reported trust score when the IdP computes and issues a Level of Assurance (LOA) value to the client platform.

This approach provides a path forward for Service Providers to obtain better picture of the state of client endpoints, and thereby providing them better assurance of the quality of the client's computing environment.

It is our intention to demonstrate a prototype system in the near future to elaborate upon, and evaluate, the ideas presented in this paper. Of particular interest to the authors is using concepts behind this paper in extending standard authentication systems such as Kerberos to offer a higher level of assurance of identity claims made to cloud service providers.

Endnote

[a]These that are unique to each TPM hardware chip and "burnt in" during its manufacture.

Competing interests
The authors declare that they have no competing interests.

Authors' contribution
TH and JZ proposed the new system architecture and protocol extensions. TH drafted the sections on Abstract; Introduction; Architecture for Cloud-based Integrity Measurement Services using TED. JZ drafted the sections Background: TED and Integrity Measurements; Identity and Trustworthy Collaborations. TH and JZ jointly drafted the Conclusions and further work section. All authors read and approved the final manuscript.

Acknowledgement
We thank Stephen Buckley at the MIT Kerberos Consortium and John Taylor and Dimitrios Georgakopoulos from CSIRO for supporting this work.

Author details
[1]CSIRO ICT Centre, PO Box 76, Epping, NSW 1710, Australia. [2]MIT Kerberos and Internet Trust Consortium, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

Received: 9 July 2012 Accepted: 22 January 2013
Published: 13 February 2013

References

1. Federal Identity Credentialing and Access Management (2009) Trust Framework Provider Adoption Process (TFPAP). [Online]. Available: http://www.idmanagement.gov/documents/TrustFrameworkProviderAdoptionProcess.pdf
2. Open Identity Exchange (OIX) (2011) The Respect Trust Framework. Public Review Draft [Online]. Available: http://openidentityexchange.org/frameworks
3. Burr WE, Dodson DF, Perlner RA, et al. (2008) DRAFT i Draft Special Publication 800-63-1 Electronic Authentication Guideline. [Online]. Available: http://csrc.nist.gov/publications/drafts/800-63-rev1/SP800-63-Rev1_Dec2008.pdf
4. Trusted Computing Group (2007) TCG Specification Architecture Overview Specification Revision 1.4. [Online]. Available: http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14
5. Security Services Technical Committee (2005) Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
6. Ironkey (2013) Company home web page. [Online]. Available: http://www.ironkey.com/
7. Gemalto - security to be free (2013) Company home web page. [Online]. Available: http://www.gemalto.com/
8. Gratzer V, Naccache D (2007) Trust on a nationwide scale. IEEE Secur Privacy 5(5): 69–71
9. Trusted Computing Group (2013) Trusted Computing Group Home. [Online]. Available: http://www.trustedcomputinggroup.org
10. TPM Main Specification (2011). [Online]. Available: http://www.trustedcomputinggroup.org/resources/tpm_main_specification
11. Nepal S, Zic J, Liu D, Jang J (2010) Trusted computing platform in your pocket. In: Embedded and Ubiquitous Computing, IEEE/IFIP International Conference on, pp 812–817
12. TCG Infrastructure Working Group (2006) TCG Infrastructure Architecture Part II - Integrity Management, TCG Standard. [Online]. Available: http://www.trustedcomputinggroup.org/resources/
13. Security Services Technical Committee (2005) Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
14. TCG Trusted Network Connect Working Group (2012) TNC Architecture for Interoperability, Version 1.5, TCG Standard. [Online]. Available: http://www.trustedcomputinggroup.org/resources/
15. Yao J, Chen S, Wang C, Levy D, Zic J (2010) Accountability as a service for the cloud. In: Services Computing, IEEE International Conference on, pp 81–88

doi:10.1186/2192-113X-2-4
Cite this article as: Zic and Hardjono: **Towards a cloud-based integrity measurement service.** *Journal of Cloud Computing: Advances, Systems and Applications* 2013 **2**:4.