On Device-based Identity Management in Enterprises

Marco Casassa Mont, Boris Balacheff
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2007-53
April 18, 2007*

device, device management, identity management, device identity, trust, trusted computing, identity provisioning, access control

This paper focuses on the management of device-based identities within enterprises. This is a key requirement in enterprises where the identities of platforms and devices have become as important as the identities of humans to grant access to enterprise resources. In this context, access control systems need to understand which devices with what properties are being used to access resource, by whom and in which contexts. Trust in managed devices' identities is an important first step to enable this. No effective commercial solution is currently available. We investigate requirements and related issues. We introduce an initial approach to: model devices' identities; enable their provisioning in heterogeneous enterprise systems; provide support for making and enforcing related access control decisions; leverage trusted computing capabilities of modern devices to deal with aspects of trust management. We describe a related solution where access control is based on policies that take into account: device identities in addition to traditional human-based identities; protected resources; additional constraints on contextual information. A working prototype (proof-of concept) has been fully implemented by HP Labs by leveraging and extending HP OpenView Identity Management solutions and using trusted computing-enabled devices. This is work in progress: we aim at setting the context and discussing our current status and next steps.

* Internal Accession Date Only Approved for External Publication

On Device-based Identity Management in Enterprises

Marco Casassa Mont, Boris Balacheff

Hewlett-Packard Labs, Trusted Systems Lab
Bristol, UK
{marco.casassa-mont, boris.balacheff}@hp.com

Abstract. This paper focuses on the management of device-based identities within enterprises. This is a key requirement in enterprises where the identities of platforms and devices have become as important as the identities of humans to grant access to enterprise resources. In this context, access control systems need to understand which devices with what properties are being used to access resource, by whom and in which contexts. Trust in managed devices' identities is an important first step to enable this. No effective commercial solution is currently available. We investigate requirements and related issues. We introduce an initial approach to: model devices' identities; enable their provisioning in heterogeneous enterprise systems; provide support for making and enforcing related access control decisions; leverage trusted computing capabilities of modern devices to deal with aspects of trust management. We describe a related solution where access control is based on policies that take into account: device identities in addition to traditional human-based identities; protected resources; additional constraints on contextual information. A working prototype (proof-of concept) has been fully implemented by HP Labs by leveraging and extending HP OpenView Identity Management solutions and using trusted computing-enabled devices. This is work in progress: we aim at setting the context and discussing our current status and next steps.

1 Introduction

Computing devices are becoming more and more pervasive in today's society. Laptops, PDAs, mobile phones, etc. are used by employees in enterprises to fulfill their jobs, enable mobile communications and engage in local/remote connections to enterprise services, applications and information.

The separation between work, public and private aspects of people's life is becoming more and more blurred. In particular, some devices are not only used for work-related matters but also for personal matters, such as accessing the web to retrieve information and make transactions, exchanging personal e-mails, making personal phone calls, storing and keeping track of personal information, calendars, etc.

From a user (individual) perspective, this trend further simplifies their day-to-day life by avoiding any unnecessary duplication of devices, tools and related efforts. There are of course implicit risks: for example, by doing this, aspects of "private" life (e.g. personal information, personal transactions, web browsing habits, etc.) could be potentially "assessed/disclosed" in the working environment, with potential negative consequences. This is an important issue, involving privacy aspects.

From an enterprise perspective, the fact that devices are used by employees for a variety of purposes, introduces additional risks and threats, in particular about the integrity of these devices and their trustworthiness to access enterprise intranets and networked resources. An additional risk is that private devices (e.g. personal laptops, etc.) could also be used at work - with potential lower security and assurance levels (e.g. about installed software, patch control, local access control settings, etc.) than the ones mandated by the enterprise.

In this paper we specifically focus on devices (e.g. laptops) owned by enterprises (and other organisations) and used by employees to carry out their daily work (and potentially their private activities).

Current enterprise services, applications and information are mainly protected by traditional access control systems that usually only take into account human-based identities (via login/passwords, digital certificates, etc.) or (in more advanced situations) only human-based identities that are strongly bound to a given device. To have better control of accessed resources, it is becoming more and more important for enterprises also to explicitly identify what the identity of devices are, along with its properties i.e. consider the identity of a device as a self-standing entity or the identity of a device as one of a group of known entities. Furthermore, trust and assurance is required about the authenticity and validity of a device's identity.

Dealing with devices' identities and their associations to human identities is not trivial. It involves making decisions on how to model these identities, how to provision them to enterprise systems and solutions, how to deal with their lifecycle, how to set proper access control policies and enforce them, how to deal with trust aspects.

The goal of this paper is to explore this space and propose an initial approach and solution to address (aspects of) the management of devices' identities in enterprises. This is work in progress, involving ongoing R&D activities at HP Labs, in collaboration also with HP business groups.

2 Scenario of Interest

We consider an enterprise scenario where users (i.e. employees, business partners, other workers, etc.) use enterprise devices (laptops, PCs, mobile appliances, etc.) to access enterprise resources, such as web services, shared file systems, document repositories, legacy applications and services – either by connecting to the enterprise intranet or occasionally via the Internet, e.g. by means VPN connections, etc. Occasionally these devices can be used for personal matters.

Users usually authenticate to their devices via their login and password: their "identities" are known by the enterprise. To perform their day-to-day work, users might access printers, FAX machines, shared file system areas, etc. that do not require further levels of authentications.

However, a user might also need to access specific enterprise services (e.g. an HR application, SAP user management tools, research and marketing tools, etc.) which require further access rights and credentials. In today's enterprise scenarios this is achieved by requiring the user to further authenticate, for example via an additional login/password or via digital certificates. In more advanced cases, single-sign-on mechanisms are used to allow access based on previous authentications, users' roles and specific access control configurations about the required resources. In all these cases it is common practice that just the identity of users is kept into account, their authentication tokens and/or knowledge of secrets shared between them and an authentication service.

In the enterprise scenario under analysis, the identity of a device is also important and it is taken into account. A user could be granted access to a resource purely based on the identity of a device (it is using) and its properties, for example if it has been checked and vouched by enterprise IT administrators. In this context the user could be anonymous or authenticated. Alternatively, access could be granted only if a specific combination/association of user identity and device identity is available. The same device could be shared by multiple users, at different times and for different purposes.

In this scenario, enterprise IT and security administrators protect enterprise resources by defining appropriate fine-grained access control policies that are derived from business and security needs and can be based on any combination of devices' identities, users' identities, device properties and other contextual information.

In this scenario we also consider the case where trusted computing components, such as Trusted Computing Group (TCG) TPM modules [1], are deployed within devices and leveraged to further increase security aspects of devices' identities and their overall trustworthiness. Specifically, they can be used to strongly protect cryptographic identities (that can be encrypted before being locally stored).

In the remaining part of this paper we will use the terms "human-based identity" and "user identity" to refer to people's identities whilst we will use the terms "device-based identity" and "device identity" to specifically refer to identities of devices.

3 Problem Statement

Our work aims at addressing the problem of the management of device-based identities in enterprises. Specifically, this paper sets the context and suggests an initial approach to deal with it.

Section 4 highlights a few important requirements that need to be taken into account. Section 5 provides further analysis on the implications for the enterprise to satisfy these requirements. Section 6 introduces our approach and solution, followed by a discussion on related work, current status and next steps.

4 Important Requirements

A few important requirements have been identified by investigating our enterprise scenario and by focusing on aspects and processes involved in device-based identity management. An initial set of these requirements follows:

- **Define a model and explicitly represent a device identity**;
- **Be able to "assess" and "certify" a device identity to deal with trust issues**;
- **Securely store and protect a device identit**y;
- **Be able to associate users' identities to devices' identities**;

- **Be able to provision devices' identities (along with users' identities) within enterprise systems and IT security systems, such as access control systems**;
- **Deal with the lifecycle management (inclusive of modification and disposal) of devices' identities, in addition to the lifecycle management of traditional users' identities**;
- **Define and manage fine-grained access control policies that can keep into account any combination of users' identities, devices' identities, device properties and other contextual informatio**n.

All these requirements have an impact on current devices, on how enterprises assess, manage and use identities, their lifecycle and how resources are protected. Based on the stated problem and related requirements, this paper further discusses our approach and proposal to deal with the following aspects: (1) How to enable device-based identity representation and provisioning within enterprises; (2) How to explicitly model device identities in complex access control scenarios also involving human-based identities and resources; (3) How to allow/disallow fine-grained access to enterprise resources based on the device identity, independently by the human identity; (4) How to allow/disallow fine-grained access based also on other contextual and system properties, including device properties; (5) How to allow/disallow fine-grained access based on any combination of device identities, human identities and contextual information. The following section provides further analysis and discussions about these aspects.

5 Our Analysis

This section further analyses the stated problem, by taking into account requirements, addressed enterprise scenario and current enterprise identity management solutions. This analysis focuses on three key aspects: (1) processes that an enterprise needs to put in place to deal with device-based identities; (2) modelling and storage of a device identity; (3) definition of fine-grained access control policies involving devices' identities and users' identities.

5.1 Enterprise Processes for Device-based Identity Management

The management of devices' identities in enterprises has to comply with enterprises' current identity management processes – in particular the ones that have already been deployed to deal with users' identities [2]. Figure 1 provides a high-level overview of these processes:

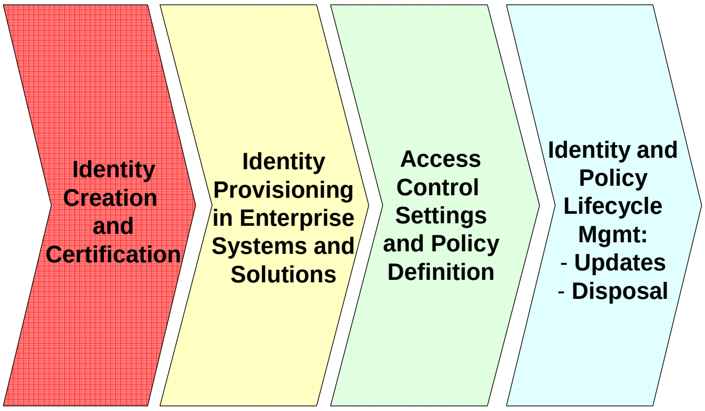

Identity Creation and Certification
Identity Provisioning in Enterprise Systems and Solutions
Access Control Settings and Policy Definition
Identity and Policy Lifecycle Mgmt:
- Updates
- Disposal

Fig. 1. Enterprise Identity Management Processes

A more detailed description of these processes and related implications for devices' identities follow:

- **Identity creation and certification:** this is the process where identity information is collected by an enterprise, its provenance is checked and verified. An identity can be certified and potentially vouched by another entity, for example a trusted third party. In case of a device's identity this stage requires collecting properties and information about the device itself and the definition of what its unique identity is. This step might require as well a certification of this information, according to Enterprise IT security standards. An additional step, also related to devices' identities, consists in storing these identities (in a secure and safe way) directly on the devices, for their future usage. As anticipated, trusted computing features can be leveraged to securely achieve this goal. Section 5.2 further expands on this point;

- **Identity provisioning**: once an identity (of a user or a device) has been created, it has to be "provisioned" within enterprise systems. The provisioning phase (currently in terms of users' identities) usually involves: (1) further processing of an identity's attributes; (2) storing identity attributes in relevant enterprise data repositories (e.g. databases, LDAP directories, etc.); (3) creating user accounts; (4) setting and configuring applications and services to recognise these new identities. Many Identity Management enterprise solutions (e.g. [3]) currently provide these provisioning and account management features – however they mainly focus on users' identities. These steps are also required for devices' identities: these identities need to be provisioned not only to devices but also to the relevant enterprise systems to enable their future processing, authentications and access to resources;
- **Access Control Setting and Policy Definition:** This part of the process requires the definition and settings of fine-grained access control policies (by security administrators) to protect enterprise resources and allow/disallow accesses based on rights and credentials. This step can be partially carried out (and automated) during the provisioning phase and/or by intervention of administrators. The introduction of devices' identities increases the degree of control and richness of access control policies but, at the same time, poses problems on how to effectively capture all these aspects within access control policies. Section 5.3 further expands on this aspect;
- **Identity Lifecycle Management:** once identities are provisioned and access control policies are in place, they can be subject to modification and updates - during their entire lifetime. Eventually these identities need to be disposed and access rights removed. The same process applies both to users' identities and devices' identities with the further implication that the lifecycle of devices' identities (and related policies) not only has implications on enterprise resources but also on the affected devices. This introduces further complexity to the lifecycle management that needs to be addressed.

5.2 Modeling, Representing and Protecting a Device Identity

A device identity consists of a set of information (attributes) that uniquely identifies a device and describes its properties in a given context – in this case an enterprise context. In general a device identity can include: device unique identifier; logical name of the device; product properties of the device (including manufacturer, production date, etc.); expected "location" of the device (in case of static device); intended usage of the device/business purposes of the device; potential list of device's owners, etc.

The list of attributes composing a device identity can vary depending on the context. There is currently no agreement in the industry on exactly what a device identity (i.e. which attributes) consists of. There are a few initiatives carried on in this space to further explore this aspect, including the one mentioned in [4].

Key requirements for device identity (as well as for user identity) include being able to explicitly represent it, safely store it and then limit its use to the intended device – for example for purposes of authenticating and authorizing operations carried out from the device.

At the current stage there are two main options for representing devices' identities, reflecting what happens for users' identities:

- **A basic device identity**: this is a collection of identity attributes, with which a device will be associated. It will typically include one specific identifier used to demonstrate that the device is what it claims to be, such as a MAC address, a shared secret, or any other unique identifier such as information collected about the physical device hardware. Such an identity can be represented in a variety of ways, including XML representation, simple list of attributes, etc. The main disadvantage of this more typical approach is that it is difficult to prove that only the intended device is going to be able to use this identity to authenticate itself to a third-party. This is typical when MAC addresses are used to identify a device on the network, where there is little guarantees that another device won't be able to claim the same MAC address.
- **A cryptographic device identity:** By leveraging public-key cryptographic schemas, e.g. [5], a "private key" and a correspondent "public key" are associated to a device. This type of identity can still include a collection of attributes and it will typically be certified, for example by using digital certificates, XML-based signature schemas, etc. A "cryptographic device identity" certificate (e.g. a signed XML digital credential [6] or an X509 identity/attribute certificate [5]) contains a statement about the device's public key and it is signed, to allow checking the integrity of identity attributes when authentication is performed. . In the case of the enterprise scenario, the enterprise itself can vouch for and issue these certificates: this simplifies the overall certification process and avoids dependencies on third parties. However a certification infrastructure has to be put in place within enterprises to deal with the lifecycle management of these certificates.

A problem that remains for any type of device identity is that of being able to claim a unique binding between a device and its identity. Different types of device identities will require different types of protection mechanisms to achieve that. In the case of a "basic device identity" it is typically hard to prove that a device cannot spoof the identity of another device. Even if the basic identity consists in unique attributes of the device, those could be copied and reused by another device. In the case of a "cryptographic device identity" the focus lies on protecting the use of the private component of the cryptographic key associated to the device identity. Traditionally this has been done using cryptographic software libraries and operating system access control to restrict usage of the key to platform authentication only, and in some cases to make it difficult to export the key from the system. Methods using hardware cryptographic modules exist to enforce access control on the usage of the key in hardware, but these can be costly, and non-standard, and they can fail to meet the unique binding requirement between device and identity (for example in the case of client systems where the hardware module may end up being external and removable, rather than internal and immutable).

To address these issues, we consider the use of emerging trusted computing technologies [1] that are more and more pervasive in business computing systems. Specifically Trusted Platform Modules – or TPMs – are currently available on a broad range of devices including laptops and PCs. Typically, a TPM is a tamper-resistant cryptographic module. The advantage of using TPM is that it provides hardware access control over the use of a cryptographic key, that it is uniquely bound to the device (it is embedded in) and that it is an industry standard component that is increasingly found in business computing devices. Additionally, TPM hardware is designed to ship with a built in endorsement (cryptographic) credential installed by its manufacturer. This endorsement credential can be used to implement a device identity provisioning solution after the system has been provisioned to its end-user, whilst still being able to remotely identify that the device has the appropriate trusted computing capabilities to protect its device identity with hardware and to bind it physically to the device via the TPM.. More generally, we will use the TPM to generate a cryptographic key in a secure way, with the assurance that this cryptographic key can only be used on the device where it was provisioned to represent the device's identity. We do not go into the details of the use of state-of-the-art TPM mechanisms and protocols in this paper. Details can be found in TCG specifications [1,16] and in literature on this topic such as [15].

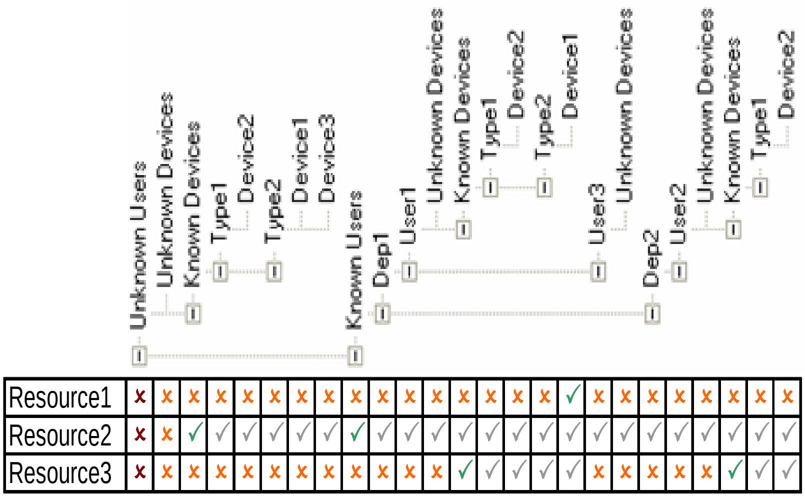
5.3 Access Control Policies involving Device Identity and User Identity

Devices' identities that have been provisioned in enterprises can be used by enterprise security administrators to define access control policies and dictate how other enterprise resources (e.g. services, applications, etc.) can be accessed. These policies involve authentications and authorization aspects.

At a "conceptual level", *user-based* access control policies can be represented via a "***Resources x Users***" access control matrix [7] - for a given set of resources and users. This matrix describes - for each protected resource and for each user - which access rights a user has on a given resource. Of course this conceptual approach allows discriminating between *known* and *unknown (anonymous)* users, based on the knowledge of their identities. These access rights could be as simple as allowing/disallowing particular operations (read/write/execute, etc.) or include more complex policy constraints, dictating for example levels of required authentication, time-based constraints, conditions on specific (contextual) attributes, etc.

This is a good starting point also to explore how to factor in devices' identities. We analysed a few different models of access control policies by leveraging the matrix model:

a) **Representation of devices as a "special types" of Users** and classify them as "*Unknown Devices*" and "*Known Devices*", based on the fact their identity is unknown/known – within the classification of "Known/Unknown" users. The latter type contains an explicit declaration of devices' identities. Specifically this allows the representation of: *Unknown Users AND Unknown Devices*; *Unknown Users AND Known Devices*; *Known Users AND Unknown Devices*; *Known Users AND Known Devices*. Hierarchies of groups of known devices can be provided. Access control on resources can be expressed by keeping into account either users' identities or devices' identities - see Figure 2. This representation constrains administrators to represent devices in the context of (known, unknown) users that might use/be associated to these devices.



	Unknown Users	Unknown Devices	Known Devices	Type1	Device2	Type2	Device1	Device3	Known Users	Dep1	User1	Unknown Devices	Known Devices	Type1	Device2	Type2	Device1	User3	Unknown Devices	Dep2	User2	Unknown Devices	Known Devices	Type1	Device2
Resource1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Resource2	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Resource3	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓

Fig. 2. Representing devices as a “special type” of users

As shown in Figure 2, fine-grained access control rules/constraints can be defined in the intersection of a user/device with a resource. Rules in the access control matrix can be used to deal with joint authentication – *AND* - of both a user and a device and related access constraints. This includes:

- Constraints for authenticating users with traditional authentication mechanisms (e.g. login/password, credentials);
- Constraints for authenticating devices, for example by requiring certified identities;
- Check properties of the involved user and device and impose additional constraints on other contextual information.

This approach allows enterprise security administrator to express a policy such as *“only a known user using a specific known device, with, for example, an identity underpinned by a TPM module, can access the specified resource”*.

b) **Representation of devices along with their identities as Resources**. In this approach, devices are listed (either separately or within hierarchies) as “Resources”. A device can authenticate itself with its own identity. A negative aspect of this approach is that is not clear how to deal with unknown devices and, in general, how to associate overall access control rights. In this model devices are just represented as resources: we still want to enable their access to other resources purely based on their identities.

The main limitation of the above two approaches is that users and devices are classified in the same matrix – hence constraints applies to the identity of a user *AND* the identity of a device (unless one of the two or both are unknown). In our analysis we also explores additional technical approaches to handle access control based on ***combinations*** of devices, users and resources to handle authentication and access of “users *AND* resource” or “users *OR* devices”. This allows administrators to describe fine-grained access control by keeping into account not only joint authentication of users and devices but also authentication of only one of the two categories. Alternative approaches include:

- **Usage of multiple matrices**, including a “***Resources x Users”*** matrix and a “***Resources x Devices”*** matrix. The main issue is defining a consistent way to express access rules across these matrices: is could be an *AND* constraint (a device AND a user must be authenticated) or an *OR* constraint (e.g. a device OR a user must be authenticated). This really depends on the type of resource and nature of the policy. For example, a dial-up connection accessing a company intranet could use an *OR* constraint: if the device belongs to the company, the access is granted. If not, the user can authenticate with a login/password. But accessing a resource that contains “secrets” or valuable data might require using an *AND* constraint across matrices: the device must be authenticated and the user must log in.

  If an *AND* constraint is used, the two matrices show the policy very clearly: devices and users who can access a resource are well identified. But if an *OR* constraint is also needed, it is hard to understand the semantic. This problem can be solved with one matrix and rules, but it is not very simple to fill in the matrix in this case and specifically identify in a systematic way witch rules to use;

- Usage of **three-dimensional matrix.** A more precise way to have a policy matrix involving users and devices is to use a three-dimensional matrix. In one dimension there are resources, in the second users and in the last devices. But it may be impossible to fill in such a matrix or very hard to understand it. In addition hierarchies/groups are very important when rules come from resources, users and devices. Which rule has priority? Such a view could involve lots of representational problems during the creation of the policy. To simplify, only **two "Users x Resources"** matrices could be used: one matrix (a) including unknown devices and one (b) including known devices. With that, it is possible to specify: *AND* constraints between matrices - deny access in the first matrix (a), and allow in the second (b) for known users; *OR* constraints between matrices - allow access in the first matrix (a) for known users and in the second (b) to everybody. However this approach requires duplication of information (e.g. resource representation) across matrices and it does not easily scale in case of enterprises where large amounts of users, devices and resources need to be managed.
- Usage of **a "Tree of Matrices"** where: devices are classified in a hierarchical way, via a tree; each leaf of the tree – i.e. a device – define an access control policy consisting of a **"Resources x Users"** matrix. This is equivalent to a three-dimensional matrix, though focused on devices and their hierarchies. It allows delegated administration of subset of trees. However, with such an approach, it is complex for an administrator to understand exactly what is allowed by a policy. For example, a device can have special rights: they are set in the matrix of the device (a leaf of the tree). If several devices have some modifications in their own matrices, it will be really hard for an administrator to see the global policy. Administrator-friendly schemas (e.g. based on colors, icons) in the access control matrix can be used to help administrators to understand policies by differentiating if a policy applies to all sub-nodes of a given node or, it just applies to the current node and it does not to the sub-nodes or, if there are differences between the current node and some of the sub-nodes. However, more work and analysis has to be done to fully understand the implications of this approach.

In our opinion - at the current stage - the most realistic and feasible approach (based also on current enterprise identity management solutions for users' identities) consists in using **a "Resources x Users" matrix** and representing in the "**Users Dimension**" all the possible combinations of users' identities and associated devices' identities. As anticipated, this happens by considering all combinations of ***Known/Unknown Users*** with ***Known/Unknown Devices –*** as shown in Figure 2. In this context, Allow/Deny policies or fine-grained access control policies can be set at the intersections of managed resources and each of the above categories, to obtain the required level of control.

6 Our Approach

Our current approach to device-based identity management in enterprises is pragmatic. To move towards its adoption, we want to leverage as much as possible state-of-the-art identity provisioning and access control solutions and extended them to manage devices' identities. We also choose to use strong cryptographic identities for devices, to allow proper security bindings to be established with infrastructure protocols that will make use of device authentication for access control. In this context, we recognise the value of trusted computing technology to help provide proper binding of device identity to the device, and we make use of certification for these identities. Based on key requirements and our analysis, we propose a solution that consists of a system and related mechanisms to:

a. Explicitly certify and protect cryptographic devices' identities by leveraging (when available) trusted computing capabilities [1] of a device (e.g. its TPM module) and an enterprise "Identity Certification Service";
b. Allow for a flexible association of human identities to device-identities – when this is required;
c. Provision and manage the lifecycle of device identities (and other associated information) into enterprise management systems;
d. Support fine-grained, policy-driven access control on enterprise resources taking into account different types of identities (device and/or human-based) and contextual information.

We consider the significant case where enterprise devices are configured by enterprise administrators. In our model, a "self-registration" web service is used to authenticate administrators, before starting the process of registering and provisioning a device identity. This web service mediates and coordinates all these steps. Attributes qualifying a device identity (e.g. name, manufacturer, configuration attributes, etc.) are inserted by an administrator via a related web form. An optional approach for the "self-registration" service is to allow individual users to register a device and provision a device identity within the enterprise infrastructure.

Managed devices may or may not have trusted computing capabilities e.g. TPM modules. In either case, a unique (cryptographic) private key is associated to a device and a device identity is "certified" for that key via the

"Identity Certification Service". Specifically, a device identity is in the form of a signed certificate of an RSA key pair.

In case of a device being TPM enabled, the concept of identity is further strengthened and security is introduced as the private key associated to the "device identity" certificate can be generated by the TPM and can be bound permanently for use by that TPM in that device.

The "Identity Certification Service" is a "Certification Authority" specialised in handling and certifying devices' identities with a format that can be flexibly configured by the enterprise: current formats (we experimented with) are based on X.509 certificates and digital-signed XML credentials. Importantly the "Identity Certification Service" will be able to identify in the device identity certificates it issues whether those are issued to (1) strong cryptographic device identities rooted in hardware TPM, or (2) to software-protected cryptographic device identities protected by a user credential.

In the case where the user "self-registration" option is used for the provisioning service, we cannot rely on a human administrator to verify whether the device identity will be appropriately protected by TPM hardware. In this case it is possible to take advantage of the TPM endorsement credentials (as defined in the TCG specifications [16]) for the self-registration web service to identify remotely that the device identity to be certified is indeed related to a cryptographic key that was generated by a hardware TPM - produced by a known manufacturer. We will not discuss the details of these mechanisms in this paper, they are standard applications of TPM technology. This "Identity Certification Service" can be operated by either internally to an enterprise or by a trusted third party.

If required, the "Identity Certification Service" also allows an administrator to associate a human-based identity (e.g. his/her login name or his/her identity certificate) to the device identity and certifies this binding.

Devices' identities are then provisioned to enterprise systems via our enterprise identity provisioning solution. In our approach, the same identity provisioning solution is used to provision also "human-based identities" and their associations to "device-identities". This provisioning phase including configuring an access control system to be aware that a new device identity has been provisioned so that access control polices can be set: alternatively, if the device identity is part of an existing hierarchy of devices, it will obey to the associated policies.

Our access control system is driven by fine-grained access control policies, targeting enterprise resources (e.g. systems, web services, applications, etc.). It consists of:

a. a ***Policy Authoring Point (PAP)*** to author fine-grained access policies keeping into account the nature of the remote device (e.g. with or without TPM), a broad variety of identities (including "devices' identities", "users' identities", association of devices to users), their hierarchical organisation, etc;
b. a ***Policy Decision Point (PDP)*** that makes decisions based on the context and the above policies;
c. a ***Policy Enforcement Point (PEP)*** that intercepts runtime attempts to access enterprise resources, gather contextual information (such as a device identity) and enforces decisions made by the PDP component. We considered the significant case of an enforcement point deployed within a web server (providing enterprise web services/applications).

Particularly important is the flexibility that we introduce in modelling device identities and related access control rules. Our solution uses an access control "*matrix*" model similar to the one described in section 5.3 based on a "*Representation of devices as a "special types" of Users*" and shown in Figure 2. This matrix represents (a) *controlled resources* and (b) all combination of Known/Unknown "*entities*" that can (or cannot) access resources.

Entities can be devices, users and any grouping of them based on their identities. The intersection in the matrix of a resource with an entity (or a set of hierarchical entities) can be set to allow/disallow accesses via fine-grained rules (based on time, contextual parameters, certificate properties, TPM-based device authentication, etc.). Hierarchies of groups of devices are supported. In this context, a device can authenticate itself with its own identity, potentially underpinned by its TPM (when present).

A full working prototype has been implemented to demonstrate the actual registration, certification and provisioning of devices' identities and the actual definition and enforcement of access control policies. As a proof-of-concept, this has been achieved by leveraging and extending three HP Identity Management solutions:

- **HP ProtectTools Embedded Security** [8]: this is a state-of-the-art application that includes standard software interfaces as defined by the TCG, as well as cryptographic service providers to access TPM functionalities in a device, generate cryptographic keys, deal with data encryption and manage access control to TPM-protected keys;
- **HP OpenView Select Identity** [3]: this is a state-of-the-art identity provisioning and user account management solution. This solution – currently used to handle users' identities provides native "self-registration" capabilities to collect identity attributes, process them and then automatically provision them by creating user accounts, storing information in predefined data repositories and other related system, such as access control systems.
- **HP OpenView Select Access** [9]: this is a state-of-the-art role-based access control system to describe fine-grained access control policies involving users' identities, managed resources and related access constraints.

Figure 3 illustrates the main components and steps involved in our prototype.

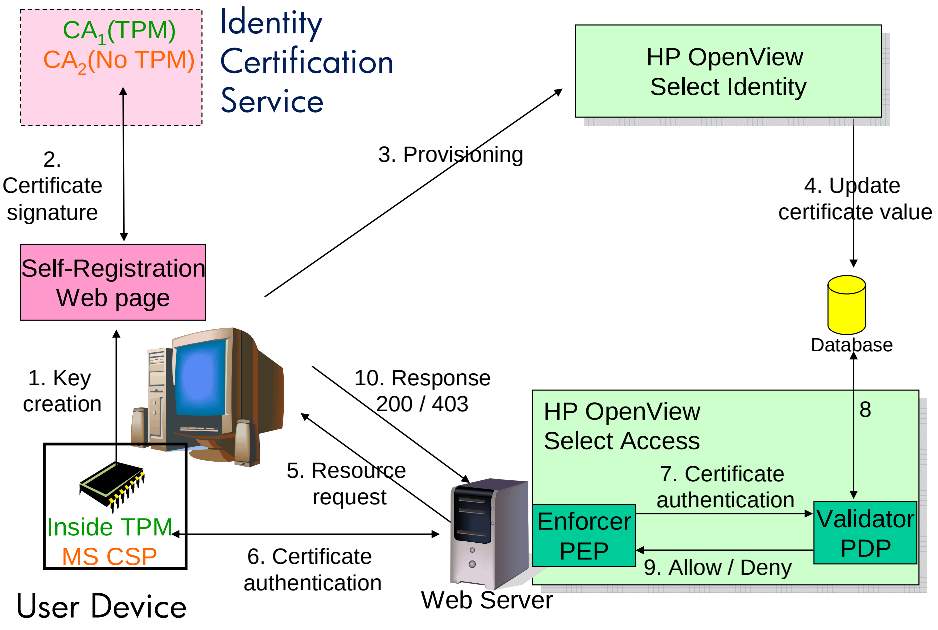


Fig. 3. Device-identity Management Demonstrator

The “self-registration” web service has been implemented by extending “self-registration” capabilities offered by HP OpenView Select Identity. Before starting the process of registering a device identity, an administrator logs in to this service.

In our prototype, a device identity is cryptographic key pair that is associated to the device: if available, this is achieved by using a TPM component installed on the device. Otherwise we use standard cryptographic libraries. We explicitly considered the case where we create an identity for a TPM-enabled device. In our current implementation, enforcing the use of a TPM requires an administrator to use HP ProtectTools to generate a TPM cryptographic key - and issue a certificate request (Step 1 – Figure 3) to the “self-registration” service. The invocation of HP ProtectTools is made by the “self-registration” service, transparently to the administrator. This model can be naturally extended for the self-registration service to challenge a newly provisioned device to establish that it is indeed using a TPM key as a device identity as described earlier.

The “self-registration” web service then interacts with the “Identity Certification Service” to generate and retrieve a related identity certificate (Step 2 – Figure 3).

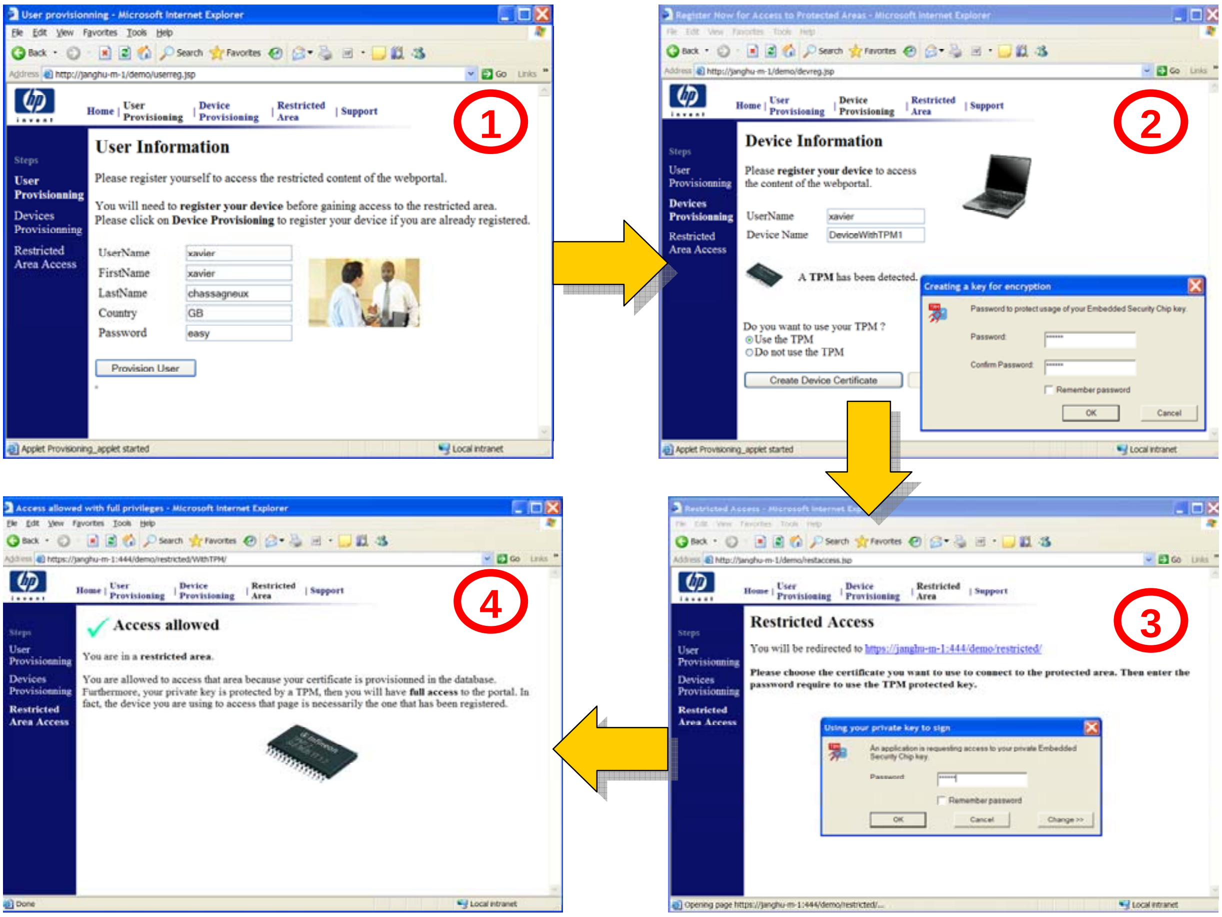
At this stage the “self-registration” service interacts with our enterprise identity provisioning solution – specifically HP OpenView Select Identity (Step 3- Figure 3). This solution has been configured to store related information in a data repository (Step 4 – Figure 3) shared with our access control system – HP OpenView Select Access – and notify this system that a new device identity has been added.

Our prototype fully implements the access control policies and related access control mechanisms described at the beginning of this section, by leveraging and extending HP OpenView Select Access.

Devices attempting to access controlled resources – such as web services, applications, etc. (Step 5 – Figure 3) are challenged, for device authentication, by this access control system, to “authenticate” with their device-based identities (certificates) by demonstrating their “knowledge/ownership” of associated private keys (Step 6 – Figure 3). Additional controls can be made on associated “user-identities” – if required.

An access control decision is made by the PDP component of the access control system, based on identity information and context (Steps 7-8 - Figure 3). Selective access to enterprise resources is therefore enforced (Step 9 – Figure 3) based on security criteria for example, to differentiate which enterprise web services can be accessed by authorized users from any device; by authorized users from enterprise-owned devices; by authorized users from one (or a small number of) specific authorized devices, or even by any users from a specific set of authorized devices. Based on this enforcement step, access to requested resources is allowed/denied (Step 10 – Figure 3).

Figure 4 shows a few screenshots of some of these steps carried out by a demonstrator built on top of our prototype: (1) “self-registration” web service collecting a user identity; (2) “self-registration” web service collecting a device identity (in this example just the device name) and potentially associating it to a user identity. Administrators can also choose to use a TPM (if installed): in this case, the administrator is challenged by HP ProtectTools to access TPM functionalities; (3) a device with a TPM (potentially associated to a user) tries to access a protected web resources. The user is challenged by HP ProtectTools to demonstrate it can access the private key protected by the TPM; (4) device is authenticated and access is granted.



1
User provisionning - Microsoft Internet Explorer
Home | User Provisioning | Device Provisioning | Restricted Area | Support
Steps
User Provisionning
Devices Provisionning
Restricted Area Access

http://janghu-m-1/demo/userreg.jsp

User Information

Please register yourself to access the restricted content of the webportal.

You will need to **register your device** before gaining access to the restricted area.
Please click on **Device Provisioning** to register your device if you are already registered.

UserName: xavier
FirstName: xavier
LastName: chassagneux
Country: GB
Password: easy

2
Register Now for Access to Protected Areas - Microsoft Internet Explorer
Home | User Provisioning | Device Provisioning | Restricted Area | Support
Steps
User Provisionning
Devices Provisionning
Restricted Area Access

http://janghu-m-1/demo/devreg.jsp

Device Information

Please **register your device** to access the content of the webportal.

UserName: xavier
Device Name: DeviceWithTPM1

A **TPM** has been detected.

Do you want to use your TPM ?
● Use the TPM
○ Do not use the TPM

Creating a key for encryption

Password to protect usage of your Embedded Security Chip key.

Password: *******
Confirm Password: *******
□ Remember password

3
Restricted Access - Microsoft Internet Ex
Home | User Provisioning | Device Provisioning | Restricted Area | Support
Steps
User Provisionning
Devices Provisionning
Restricted Area Access

http://janghu-m-1/demo/restaccess.jsp

Restricted Access

You will be redirected to https://janghu-m-1:444/demo/restricted/

Please choose the certificate you want to use to connect to the protected area. Then enter the password require to use the TPM protected key.

Using your private key to sign

An application is requesting access to your private Embedded Security Chip key.

Password: ******
□ Remember password

Opening page https://janghu-m-1:444/demo/restricted/...

4
Access allowed with full privileges - Microsoft Internet Explorer
Home | User Provisioning | Device Provisioning | Restricted Area | Support
Steps
User Provisionning
Devices Provisionning
Restricted Area Access

https://janghu-m-1:444/demo/restricted/WithTPM/

✓ **Access allowed**

You are in a **restricted area**.

You are allowed to access that area because your certificate is provisionned in the database. Furthermore, your private key is protected by a TPM, then you will have **full access** to the portal. In fact, the device you are using to access that page is necessarily the one that has been registered.

Fig. 4. Demonstrator Screenshots

7 Related Work

The idea of being able to identify devices is not new and has been pursued extensively with weakly bound identifiers such as software protected cryptographic identities, MAC addresses, or even statically allocated IP addresses. The idea of using a Hardware Security Module to protect device identities is described in [10]. But device identities have so far been used exclusively in the network infrastructure, such as in [11,12], rather than to protect access to application level resources in the enterprise. As such, most of the existing enterprise identity and access control management software do not include features to support device identities and authentication.

There are currently multiple initiatives to standardise device identities, as mentioned in [4], including ePC, PDML, ONS, EAN, GTAG and ISO 15693: for the time being it is not obvious how these initiatives will evolve and if they will converge.

Relevant work is also going on in the context of Liberty Alliance [13] in terms of *Identity Capable Devices*: this is about provisioning devices (potentially using trusted computing devices) with “identity tokens” and autonomously using them in federated identity management context [14]. This work can influence the way devices’ identities are certified and locally associated/stored to devices - an important part of the overall device-based identity management space. We are currently involved in this initiative and related activities.

8 Current Status and Next Steps

We currently have a working prototype integrated with HP OpenView Identity Management solutions. We are refining our technology by adding further expressiveness to our access control policies. Work has also been done to include additional aspects of the usage of “device” authentication, such as reconciling network-based and application-based access control management.

Next steps include further research in this space, in particular with regards to the representation of devices in access control policies and the full lifecycle management of devices’ identities.

9 Conclusions

This paper focused on the problem of dealing with device-based identity management in enterprises. This is becoming more and more important for enterprises, where access to protected resources depends not only on users’ identities but also on the type and properties of used devices. Current enterprise identity management solutions lack support for integrated management of devices’ identities and users’ identities, in particular in terms of provisioning, access control policies and authentications. We explored requirements and analysed approaches to: represent devices’ identities as certified identities; assess their trustworthiness by leveraging trusted computing components; provision them within enterprises; represent access control policies that are flexile enough to keep into account users’ identities, devices’ identities, contextual information and additional constraints. We introduced our approach and discussed a related proof-of-concept to deal with device-based identity management that leverages state-of-the-art HP Identity Management solutions. A full working prototype has been implemented along with a related demonstrator leveraging TPM-enabled devices. This is work in progress: further work is required to fully explore the implications of devices’ identities on access control policies and how to deal with their overall lifecycle management in enterprises.

Acknowledgements

We would like to thank Xavier Chassagneux for his hard work and key contributions made whilst he was an HP Labs intern - in researching this area and implementing our prototype and demonstrator.

References

1. TCG: TCG TPM Specification, Available via http://www.trustedcomputinggroup.org, Version 1.2, 2003
2. Casassa Mont, M. Bramhall, P., Pato, J.: On Adaptive Identity Management: The Next Generation of Identity Management Technologies, HPL-2003-149, 2003
3. Hewlett-Packard (HP): HP OpenView Select Identity: Overview and Features, http://www.openview.hp.com/products/slctid/index.html, 2007
4. Wills, T.: The Identities of Electronic Devices, Digital ID World article, http://www.digitalidworld.com/modules.php?op=modload&name=News&file=article&sid=96, 2002
5. IETF: IETF PKIX Working Group, http://www.ietf.org/html.charters/pkix-charter.html, 2005
6. W3C: XML Signature WG, http://www.w3.org/Signature/, 2003.
7. Lampson, B.W.: Protection, Proc. 5th Princeton Symposium on Information Sciences and Systems, Princeton University, pp. 18-24, 1974
8. Hewlett-Packard (HP): HP ProtectTools Security Manager, http://www.hp.com/sbso/security/protecttools.html,, 2007
9. Hewlett-Packard (HP): HP Openview Select Access: Overview and Features - http://www.openview.hp.com/products/select/, 2005
10. Baldwin, A., Shiu, S.: “Hardware Encapsulation of Security Services”, ESORICS 2003, 2003
11. iPass: DeviceID, http://www.ipass.com/services/services_deviceid.html, 2007
12. INTEL: Network Access Control: User and Device Authentication, http://www.intel.com/it/pdf/network-access-control.pdf, 2005
13. Liberty Alliance: The Liberty Alliance Project, http://www.projectliberty.org/, 2007
14. Liberty Alliance: ID-WSF Advanced Client 1.0 DRAFT Specifications, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_advanced_client_1_0_draft_specifications, 2007
15. Pearson, S. (ed): Trusted Computing Platforms, Prentice Hall, 2002
16. TCG: Trusted Computing Group, http://www.trustedcomputinggroup.org, 2007